Sparse supervised principal component analysis (SSPCA) for dimension reduction and variable selection

Sharifzadeh, Sara; Ghodsi, Ali; Clemmensen, Line H.; Ersbøll, Bjarne Kjær

Published in:
Engineering Applications of Artificial Intelligence

Link to article, DOI:
10.1016/j.engappai.2017.07.004

Publication date:
2017

Document Version
Early version, also known as pre-print

Sparse Supervised Principal Component Analysis (SSPCA) for Dimension Reduction and Variable Selection

Sara Sharifzadeh[a,*], Ali Ghodsi[b], Line H. Clemmensen[a], Bjarne K. Ersbøll[a]

[a]*Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Lyngby, Denmark (Ph. +45-45253413),* [b]*Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, ON., Canada*

sarasharifzade@yahoo.com, aghodsib@uwaterloo.ca,
clemmensen.line@gmail.com, bke@dtu.dk

Abstract

Principal component analysis (PCA)[1] is one of the main un-supervised pre-processing methods for dimension reduction. When the training labels are available, it is worth using a supervised PCA strategy. In cases that both dimension reduction and variable selection are required, sparse PCA (SPCA) methods are preferred. In this paper, a sparse supervised PCA (SSPCA) method is proposed for pre-processing. This method is appropriate especially in problems where, a high dimensional input necessitates the use of a sparse method and a target label is also available to guide the variable selection strategy. Such a method is valuable in many Engineering and scientific problems, when the number of training samples is also limited. The Hilbert Schmit Independence Criteria (HSIC) is used to form an objective based on minimization of a loss function and an L_1 norm is used for regularization of the Eigen vectors. While the proposed objective function allows a sparse low rank solution for both linear and non-linear relationships between the input and response matrices, other similar methods in this case are

[1]PCA:principal component analysis, SPCA: sparse PCA, SSPCA: sparse supervised PCA, SPLS: sparse partial least squares, PMD: penalized matrix decomposition, SVD: singular value decomposition, HSIC: Hilbert Schmidt independence criterion, RKHS: reproducing kernel Hilbert space, SIMPLS: statistically inspired modification of PLS, SVM: support vector machine, CV: cross validation, RBF: radial basis function, RMSE: root mean square error, ROI:region of interest, NIR: near infrared, SSC: solvable solid content, KNN: K nearest neighbour

only based on a linear model. The objective is solved based on penalized matrix decomposition (PMD) algorithm. We compare the proposed method with PCA, PMD-based SPCA and supervised PCA. In addition, SSPCA is also compared with sparse partial least squares (SPLS), due to the similarity between the two objective functions. Experimental results from the simulated as well as real data sets show that, SSPCA provides an appropriate trade-off between accuracy and sparsity. Comparisons show that, in terms of sparsity, SSPCA performs the highest level of variable reduction and also, in terms of accuracy it is one of the most successful methods. Therefore, the Eigen vectors found by SSPCA can be used for feature selection in various high dimensional problems.

Keywords: Variable selection, Dimension reduction, Sparse PCA, Supervised PCA, Sparse supervised PCA, Penalized matrix decomposition

---

1. Introduction

Principal component analysis (PCA) is a well known dimension reduction technique that is used in many data mining and machine learning problems such as genetics, image and signal processing, chemistry, etc. Given a data matrix $X_{n\times p}$ with n data points and p features, it maps data into an orthogonal space based on the sorted variance of the input data. In the new space, each principal component (PC) is a linear combination of all the original variables. The first PC corresponds to the highest variance and the second to the second highest variance and in the same way all PCs are estimated based on the subsequent orders of variances (Hastie et al., 2009).

However, based on the type of problem, two main limitations can be considered for PCA; First, PCA is not sparse, while in many applications, especially those with a high number of variables, it is important to reduce the number of variables and remove any irrelevant or noisy variable. For example, in spectral imaging applications, each variable might be a wavelength and sparse PCs result in a simpler vision set-up or in biology, each variable might correspond to a specific gene and interpretation of the sparse PCs is easier (Zou et al., 2004). Second, PCA is un-supervised. Although this can be considered as an advantage in many cases, it can also be a limitation when a label or response vector is available (Barshan et al., 2011; Chen et al., 2008). In such cases, it is more efficient to guide the low rank approximation algorithm based on the available target response. This is especially important when the task is regression or classification and it is preferred to map data into a low-rank space based on its maximum dependency on

the response than maximum variance.

To address the first limitation, many researchers have proposed methods and algorithms for sparse PCA (SPCA). Such as simple thresholding of the loadings (Cadima and Jolliffe, 1995), non-negative sparse PCA (Sigg and Buhmann, 2008; Zass and Shashua, 2007; Asteris et al., 2014), greedy algorithms (Moghaddam et al., 2006; A. d'Aspremont and El Ghaoui, 2008), the SCoTLASS method (Jolliffe et al., 2003), an Elastic-Net framework based on the L1-norm (Zou et al., 2004), SPCA based on the penalized matrix decomposition (PMD) (Witten et al., 2009), an augmented Lagrangian method (ALSPCA) (Lu and Zhang, 2009), a regularized singular value decomposition (SVD) (Shen and Huang, 2008), a generalized power method (Journée et al., 2010) and optimized sparse encoding by column subset selection (Magdon-Ismail and Boutsidis, 2016). Most of the solutions to SPCA are non-convex optimization procedures that find a solution close to the optimal point. Some of them such as DSPCA based on semi-definite programming (SD) (d'Aspremont et al., 2007) also guarantee a global convergence.

In addition, several supervised PCA methods have been proposed in the literature (Bair et al., 2006; Barshan et al., 2011). In (Bair et al., 2006), an initial regression step is used to find the features corresponding to high values of regression coefficients. Then, those features were used for PCA. The supervised PCA method proposed in (Barshan et al., 2011) is a generalization of PCA which aims at finding the PCs with maximum dependency to the response variables. In that work, the Hilbert Schmidt independence criterion (HSIC) (Gretton et al., 2005) was used as a dependency function between data and target response.

This work is focused on developing a sparse supervised PCA (SSPCA) algorithm. Such an algorithm is appropriate for pre-processing of high dimensional data sets with an available target response. Such condition necessitates the use of a sparse solution for variable selection or interpretation. In order to handle any type of dependency between input and response matrices, similar to (Barshan et al., 2011), the initial objective function is formed based on the HSIC criterion. In addition, an L_1 constraint is applied on the Eigen vectors in order to find sparse solutions. The resulting optimization problem is bi-convex and can be solved using the PMD algorithm (Witten et al., 2009). The sparse Eigen vectors found by the SSPCA algorithm can be used either for projection of a data set or for feature selection.

The most similar work to our work is the SPLS algorithm that is based on a latent decomposition of both response vector Y and the predictor matrix X (Chun and Keles, 2010). In that work, an L_1 norm was imposed to achieve a sparse solution and the objective was solved iteratively as a biconvex problem. However,

only a linear relationship between the input matrix and the response vector was considered. We will demonstrate that the proposed objective function is a general form of the SPLS objective function and the solution can handle data sets with linear as well as non-linear behaviour.

In this paper, SSPCA is compared with PCA, the SPCA based on the PMD algorithm (Witten et al., 2009) and the supervised PCA based on HSIC (Barshan et al., 2011). Due to the reasons explained above, it is also compared with SPLS (Chun and Keles, 2010). The experiments were conducted on both simulated and real data sets.

A version of this work has been presented in a PhD thesis previously (Sharifzadeh, 2015).

The rest of this paper is organized as follows; Section 2 describes the PCA and HSIC criterion. Section 3 introduces the SSPCA method and its connection to PMD-based SPCA, supervised PCA and SPLS. Then, experimental results are presented in section 4. Finally, discussion and conclusion are given in sections 5 and 6 respectively.

2. Background

Considering a data matrix $X_{n\times p}$ that has n data points and p features, and also a target vector $Y_{n\times 1}$, in a PCA problem, the centred data matrix X_c is projected into a new space with orthogonal directions. The projection of a data vectors x_i along a direction v_k, $k = 1,2,...,p$, is $x_i.v_k$. Then, the variance is:

$$\sigma^2_{v_k} = \frac{1}{n}\sum_i (x_i v_k)^2 = \frac{1}{n}(Xv_k)^T(Xv_k) = v_k^T(\frac{X^TX}{n})v_k = v_k^T\Sigma_x v_k. \tag{1}$$

In a PCA problem, this direction has the maximum variance and unit length:

$$\arg\max_{v_k} \sigma^2_{v_k} = \arg\max_{v_k} v_k^T\Sigma_x v_k \text{s.t.} v_k^T v_k = 1, \tag{2}$$

which can be re-formulated based on a Lagrange multiplier λ and be solved by setting the derivatives to zero:

$$L(v_k,\lambda) = \arg\max_{v_k,\alpha} v_k^T\Sigma_x v_k - \lambda(v_k^T v_k - 1), \tag{3}$$

$$\frac{\partial L}{\partial v_k} = 0, \frac{\partial L}{\partial \lambda} = 0, \tag{4}$$

$$v_k^T.v_k = 1, \Sigma_x v_k = \lambda v_k. \tag{5}$$

The Eigen decomposition of the covariance matrix Σ_x results in the Eigen vectors that maximize the variation of the projected data Xv_k.

However, as mentioned in the previous section, in the presence of a response vector, Y, finding a subspace that maximizes the dependency between the projected data Xv_k and the outcome Y is preferred.

A linear dependency between two variables can be measured based on a correlation criterion. However, to handle any linear or non-linear dependency, a more general criterion is required.

2.1. HSIC

HSIC is an independence criterion, introduced in (Gretton et al., 2005). According to HSIC, the independence of the variables X and Y is possible, if and only if any bounded continuous function of them is uncorrelated. Therefore, dependency is a more general criterion than correlation. If two random variables are independent, their HSIC value will be zero.

HSIC was used previously for a supervised PCA technique in (Barshan et al., 2011). It can be expressed in terms of kernel functions. Let $Z = (x_1, y_1), ..., (x_n, y_n) \subset (X \times Y)$ be a series of n independent observations drawn from $P_{X,Y}$, an empirical practical form of HSIC for independence testing between X and Y is:

$$\text{HSIC}(Z, F, G) = (n-1)^{-2}\text{tr}(KHLH) = (n-1)^{-2}\text{tr}(HKHL), \tag{6}$$

where F and G are separable reproducing kernel Hilbert space (RKHS), containing all continuous bounded real-valued functions of x and y respectively (from X to R and from Y to R), K and L are the corresponding kernels of F and G, $H, K, L \in \mathbb{R}^{n \times n}, K_{ij} = k(x_i, x_j), L_{ij} = l(y_i, y_j)$ and $H_{ij} = I - n^{-1}ee^T$ is the centring matrix (e is a vector of all ones). A high HSIC value shows a high level of dependency between the two kernels.

3. Formulation of SSPCA

We adopt the HSIC criterion to maximize the dependency between the projected data to the new subspace XV and the response Y. For this aim, the input kernel K is defined based on the projected data in the new subspace, $K = XVV^TX^T$. In addition, two constraints are considered for the Eigen vectors; a constraint for unit length and an L_1 norm penalty constraint for sparsity:

$$\arg\max_{V} \ \mathrm{tr}(HXVV^TX^THL) = \arg\max_{V} \ \mathrm{tr}(V^TX^THLHXV) \ \text{s.t.}\ V^TV = I,\ |V| \leq c. \tag{7}$$

Considering $Q = X^THLHX$, Eq.7 is a penalized Eigen value decomposition problem. Since Q is a symmetric and real matrix, it can be decomposed as $Q = \Psi^T\Psi$, so that $L = \Delta\Delta^T$, $\Psi_{n\times p} = \Delta^THX$. Then, the objective function can be rewritten as follows:

$$\arg\max_{V} \ \mathrm{tr}(V^TQV) = \arg\max_{V} \ \mathrm{tr}(V^T\Psi^T\Psi V) \ \text{s.t.}\ V^TV = I,\ |V| \leq c. \tag{8}$$

Two different approaches can be considered for solving this optimization problem. One strategy is penalizing the Eigen vectors matrix and finding all the Eigen vectors simultaneously. This requires different regularization parameters $(c_1, c_2, \ldots, c_p)$ for Eigen vectors, to avoid a sparse Eigen matrix of rank one. That means an increase in the number of parameters which makes the problem more difficult.

Another approach is using the same penalization constraint c for all the Eigen vectors and finding them individually in separate optimization steps. This is a more feasible strategy. Therefore, in our work, we consider the same regularization parameter for all Eigen vectors and solve the problem for each Eigen vector separately. Then, there exist a mathematical solution for this simplified problem based on the PMD algorithm (Witten et al., 2009).

First, the equivalent SVD problem to the objective function is considered for rank-K approximation of Ψ:

$$\Psi = U\Lambda V^T \ \text{s.t.}\ U^TU = I_n\ ,\ VV^T = I_p; \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_K \geq 0. \tag{9}$$

For $r \leq K$, the above SVD problem can be considered as a minimization of a loss function based on the Frobenius norm:

$$\sum_{k=1}^{r} \lambda_k u_k v_k^T = \arg\min_{\hat{\Psi}\in \mathrm{M(r)}} \left\|\Psi - \hat{\Psi}\right\|_F^2 = \arg\min_{\hat{\Psi}\in \mathrm{M(r)}} \left\|\Psi - U\Lambda V^T\right\|_F^2. \tag{10}$$

Where u_k and v_k are the column k of U and V and $M(r)$ is the set of rank-r $n \times p$ matrices. In the case of the Frobenius norm, the following has been demonstrated in (Witten et al., 2009):

$$\frac{1}{2}\left\|\Psi - U\Lambda V^T\right\|_F^2 = \frac{1}{2}\left\|\Psi\right\|_F^2 - \sum_{k=1}^{r} u_k^T \Psi v_k \lambda_k + \frac{1}{2}\sum_{k=1}^{r}\lambda_k^2. \tag{11}$$

Therefore, in the next step, the minimization in Eq. 10 is written as a maximization form for $k = 1$ and the constant terms are ignored. In addition, an L_1 constraint is added on v_k besides the unite length L_2 constraints on u_k and v_k. This makes the v_k sparse. Furthermore, an orthogonality constraint is considered.

$$\arg\max_{u_k v_k} u_k^T \Psi^k v_k \text{ s.t. } \|u_k\|_2 \leq 1, \|v_k\|_2 \leq 1, \|v_k\|_1 \leq c, u_k \perp u_1, u_2, \ldots, u_{k-1}. \tag{12}$$

The equality constraints are changed into inequality to avoid a non-convex problem. This objective function is bi-convex in u_k and v_k. That is, with u_k fixed, it is linear in v_k, and vice versa. As planned before, the same regularization parameter c controls the sparsity of the individual Eigen vectors v_k, $k = 1, 2, \ldots, r$. This optimization problem can be solved based on the PMD$(.,L_1)$ algorithm (Witten et al., 2009) that was used previously for SPCA problem. It is explained in more details in the appendix.

The step by step procedures for SSPCA are shown in Algorithm 1. As can be seen, the row and column vectors u_k and v_k are computed separately. The update equation for u_k, forces orthogonality. $U_{k-1}^{\perp}$ is an orthogonal basis to $U_{k-1} = 1, 2, \ldots, k-1$. This update step yields orthogonal factors. It cannot be used directly for v_k, since it does not result in a sparse solution. However, the v_ks, are not very correlated, since they are associated with orthogonal u_ks, (Witten et al., 2009). The update equation for v_k, utilizes the soft thresholding operator S, so that for $\tau > 0$:

$$S(a,\tau) = \begin{cases} \text{sgn}(a)(|a| - \tau) & |a| > \tau, \\ 0 & |a| \leq \tau. \end{cases} \tag{13}$$

The solution to the above equation, satisfies $v_k = \frac{S(a,\tau)}{\|S(a,\tau)\|_2}$ with $\tau = 0$, if this results in $\|v_k\|_1 \leq c$; otherwise, τ is chosen so that $\|v_k\|_1 = c$. the range of possible values for c is $1 \leq c \leq \sqrt{p}$, (Witten et al., 2009). Further demonstrations for these update formula can be found in (Witten et al., 2009) and also provided in the Appendix.

In fact, the use of soft thresholding inside the convergence loop, reduces the absolute value of the Eigen vector elements so that, some of them will become zero or close to zero. The features that, the kernel is dependent on (relevant features), remain among the non-zero elements and the zero or small elements correspond to the irrelevant and noisy input variables. Especially for the first Eigen

Algorithm 1 Procedures for SSPCA

Input: training data matrix **X**, test data **x**, kernel matrix of target variable **L** and training data size **n**.

Output: Dimension reduced training and test data using sparse Eigen vectors, **Z** and **z**.

1. Decompose L such that $L = \Delta^T \Delta$

2: $H \leftarrow I - n^{-1} ee^T$

3: $\Psi \leftarrow \Delta^T HX$

4: Compute the sparse basis based on the PMD method:

Let $\Psi^1 \leftarrow \Psi$

For $k \in 1, ..., K$:

Find u_k, v_k and λ_k by applying the following single-factor PMD algorithm to Ψ^k:

Initialize v_k to have L_2-norm equal to one.

Repeat (a) and (b) until convergence:

(a) $u_k = \frac{U_{K-1}^{\perp} U_{k-1}^{\perp T} \Psi^k v_k}{\left\| U_{k-1}^{\perp T} \Psi^k v_k \right\|_2}$

(b) $v_k = \frac{S(a,\tau)}{\|S(a,\tau)\|_2}$, where $a = \psi^k u_k$, $\tau = 0$ if $\|v_k\|_1 \leq c$, otherwise an appropriate τ is found so that, the condition is fulfilled.

$\lambda_k \leftarrow u_k^T \Psi^k v_k$.

$\Psi^{k+1} \leftarrow \Psi^k - \lambda_k u_k v_k^T$

5:Encode training data: $Z \leftarrow XV$

6:Encode test data: $z \leftarrow xV$

vector when the original Ψ^1 is used. Because the second Eigen vector is orthogonal to the first one and consequently, high value elements in the first Eigen vector, might be down weighted in the second vector due to the orthogonality issue.

An appropriate kernel (L) is the one that has the highest dependency to the input matrix. Using an appropriate constraint value c, most irrelevant variables are cancelled out and most relevant ones are remained. Both the kernel and c are chosen based on CV.

3.1. Relation to SPCA and Supervised PCA

SSPCA is in fact a general form for SPCA based on the PMD method (Witten et al., 2009) and the supervised PCA (Barshan et al., 2011). If the target kernel

$L = I$, the algorithm 1 solves the unsupervised SPCA problem. On the hand, if the regularization parameter c tend to infinity, the supervised PCA problem is solved.

3.2. Comparison with SPLS

Due to the similarities between the proposed method and SPLS (Chun and Keles, 2010), their main differences are described here. SPLS is a sparse version of the well known supervised regression method PLS. In PLS, the response matrix $Y_{n\times q}$ and the predictor matrix $X_{n\times p}$ are decomposed into latent vectors so that, $Y = TQ^T + F$ and $X = TP^T + E$. $T_{n\times k}$ is a matrix that produces K linear combinations (scores), $P_{p\times k}$ and $Q_{q\times k}$ are matrices of coefficients (loadings) and $E_{n\times p}$ and $F_{n\times q}$ are matrices of random errors. PLS finds the columns of $W = (w_1, w_2, ..., w_K)$ by successive optimization problems and then, the latent component matrix $T = XW$ is computed:

$$w_k = \arg\max_w \mathrm{cor}^2(Y, Xw)\mathrm{var}(Xw) \quad \text{s.t.} \quad w^T w = 1, \quad w^T \Sigma_{XX} w_j = 0 \ , \qquad (14)$$

for $j = 1, ..., k-1$, where Σ_{XX} is the covariance of X. Using the statistically inspired modification of PLS (SIMPLS), the k^{th} estimated direction vector $\hat{w}_k$ is found by solving the following optimization problem:

$$\hat{w}_k = \arg\max_w \ w^T \sigma_{XY} \sigma_{XY} w \quad \text{s.t.} \quad w^T w = 1, \quad w^T \Sigma_{XX} w_j = 0 \ , \qquad (15)$$

Σ_{XX} and σ_{XY} are the populations covariances of X and Y that can be replaced by the samples covariances (S_{XX}, S_{XY}):

$$w_k = \arg\max_w \ w^T X^T Y Y^T X w \quad \text{s.t.} \quad w^T w = 1, \quad w^T S_{XX} w_j = 0 \ . \qquad (16)$$

Using W, the latent components T and loadings Q are computed. Finally, $\hat{\beta}_{PLS}$ is obtained by $\hat{\beta}_{PLS} = \hat{W}\hat{Q}^T$.

In the sparse version of the PLS algorithm, an L_1 penalty is imposed to the PLS objective function:

$$w_k = \arg\max_w \ w^T X^T Y Y^T X w \quad \text{s.t.} \quad w^T w = 1, \quad |w| \leq \lambda \ . \qquad (17)$$

This optimization problem is solved by a bi-convex procedure that is explained in more detail in (Chun and Keles, 2010).

The major difference between the SPLS and SSPCA can be explained by the definition of the correlation and dependency. Similar to PLS, SPLS aims to maximize the covariance between two random variables while SSPCA (similar to supervised PCA) maximizes the dependency between them. In other words, SPLS can detect linear dependence between two variables while in SSPCA any linear or non-linear dependency can be detected. This is performed by the choice of an appropriate kernel. In addition, after finding $\hat{\beta}_{SPLS}$, a linear regression is performed to compute $\hat{Y}$. However, SSPCA is a pre-processing step and can be followed by different regression or classification methods.

4. Experimental results

Five methods including PCA, SPCA based on the PMD method, supervised PCA, SSPCA and SPLS were applied on three simulated and three real data sets and the results are shown in this section. Both regression and classification scenarios exist among these data sets. In data simulations, both linear and non-linear conditions were generated. In all the experiments and for all the methods, at least three Eigen vectors were chosen, so that their corresponding Eigen values explain at least 95% of variance. The models were trained using the cross validation (CV) model selection technique. In both regression and classification problems, the support vector machine (SVM) from the LibSVM toolbox (Chang and Lin, 2011) was used in training over the CV loops and the final tests. For classification problems, the K nearest neighbour (KNN) was also applied using CV for the choice of K and the results are compared with SVM. We have also employed CV loops for selection of the SVM parameters such as kernel type, spread parameter of radial basis function (RBF), degrees of the polynomial kernels etc. For each data set, based on its dimension, an appropriate number of folds was determined. Since in many real problems, the number of data points is less than the number of features, such condition was considered. For example, in cases where the number of samples was much less than the number of variables ($n \ll p$), larger number of folds (e.g. 10 folds) were used to avoid over-fitting.

No model parameters are required for PCA. As mentioned above, at least three components are selected explaining at least 95% of variance. However, for all the other methods, CV loops were used for model selection; In SPCA, CV was used for the choice of the restriction parameter c. As mentioned in section 3, c can be chosen in the range of $1 \leq c \leq \sqrt{p}$. In supervised PCA, CV was used for the choice of the kernel type. The tested kernels were RBF, adaptive (RBF) (Zelnik-manor and Perona, 2004), quadratic and sinusoid kernels. For RBF and quadratic kernels,

the spread parameter σ and degree parameters were respectively chosen based on iterations over a list of candidate values. For the proposed SSPCA method, both c and kernel were found based on CV. The required parameters for SPLS such as λ are also found using a CV loop.

Root Mean Square Error (RMSE) was used as an evaluation criterion for all the methods in the regression problems for both the training (over the CV loops) and final tests. In the case of classification, the percentage of classification performance was considered. In addition, the average number of non-zero rows in the selected Eigen vectors are reported. All analyses were performed using MATLAB (R2013a).

4.1. Simulation results

The first sets of experiments were performed on simulated data sets to evaluate the performance of the proposed SSPCA method and compare it with the other methods. The major difference between the three simulated cases is the type of dependency between the input matrix X and target Y. So that, both linear as well as non-linear dependencies, such as polynomial and exponential relations, are considered. In addition, the number of training samples n_tr versus the number of variables p are different in each case, covering both $n_{tr} > p$ and $n_{tr} < p$. These are important factors for evaluation and comparison of the methods based on their ability in feature selection and extraction for different data conditions.

In these experiments, the first Eigen vector will be plotted. This helps to compare the sparsity level of the tested algorithms as well as their ability to find the relevant features. As mentioned in section 3, in the case of SSPCA, the Eigen vectors elements corresponding to the irrelevant and noisy variables should be zero or small in absolute values, while those corresponding to the relevant features should be higher in absolute values. Specially, when the kernel type and other parameters are chosen appropriately. Generally, a successful method should have small elements (or zero, if it is an sparse method) for irrelevant and noisy variables and higher absolute values where the variables are relevant. That is, the principal directions should mostly be formed by the significant contribution of the relevant features.

In all simulations, the data set was randomly divided into training and test sets five times and the average results were considered.

4.1.1. Simulation 1

In this example, a data matrix $X_{sim1(150\times120)}$ with $n = 150$ random samples and $p = 120$ variables were generated from a standard normal distribution. Then,

a linear function of four variables $X(5,15,25,35)$ was defined:

$$Y_{sim1} = 6X_{sim1}(5) + 5X_{sim1}(15) - 7X_{sim1}(25) - 3X_{sim1}(35). \quad (18)$$

The data set was divided five times randomly into training (100 samples) and test (50 samples) sets and the training sets were used for finding the Eigen vectors. Fig. 1 shows the first Eigen vector for PCA, SPCA, supervised PCA and SSPCA methods as well as the regression coefficients of SPLS (β_{SPLS}). In this example, β_{SPLS} was scaled to be shown on the same plot with the Eigen vectors. For ease of visualization,each method graph is plotted with an offset relative to the other methods and the numbers on the vertical axes are reset for each graph. The big markers with black edges show the relevant features. In the figure, the y axis shows the numerical value of Eigen vector elements. Based on its sign (positive or negative), each element is combined with others to form the principal direction for transforming data into the new space. Table 1 shows the average and standard deviation of regression results. The last row shows the average and standard deviation of number of non-zero rows in the selected Eigen vectors. SPLS obtained the best result in terms of accuracy and sparsity and then the proposed method is the next best method for this linear function.

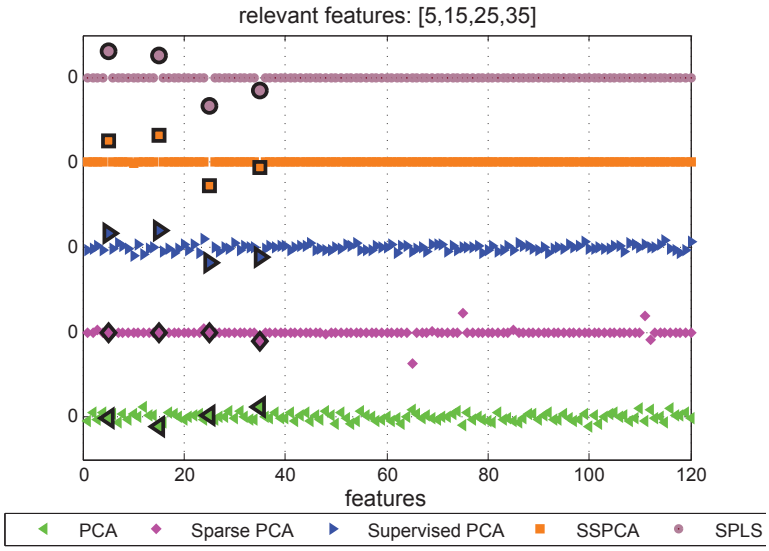


Figure 1: Comparison of the first Eigen vector/regression coefficients of the five tested methods on the first simulated data set. The black edged markers show the relevant features.

Table 1: Regression results for the first simulated data set. The average and standard deviations for five training and test sets are presented.

	PCA	SPCA	Sup. PCA	SSPCA	SPLS
$RMSE_{tr}$	9.57±0.58	9.76±0.83	5.00±0.17	2.14±0.91	0.00±0.00
$RMSE_{ts}$	9.33±1.00	9.75±1.24	7.69±0.65	2.53±1.42	0.00±0.00
Num. of NZ.	120.00±0.00	43.00±17.46	120.00±0.00	12.8±5.40	10.00±1.00

Table 2: Regression results for the second simulated data set. The average and standard deviations for five training and test sets are presented.

	PCA	SPCA	Sup. PCA	SSPCA	SPLS
$RMSE_{tr}$	1.81±0.22	1.78±0.34	1.38±0.30	1.42±0.20	0.50±0.34
$RMSE_{ts}$	2.05±0.14	2.08±0.13	1.99±0.09	1.79±0.17	2.41±0.42
Num. of NZ.	50.00±0.00	10.80±1.30	50.00±0.00	13.40±4.39	22.60±16.62

4.1.2. Simulation 2

The data matrix is $X_{sim2(100\times 50)}$. The non-linear function depends on variables $X(10,40)$:

$$Y_{sim2} = (1+X_{sim2}(10)) \circ (1+X_{sim2}(10)) + X_{sim2}(40) \oslash (0.5+ \\ (1.5+X_{sim2}(10)) \circ (1.5+X_{sim2}(10))). \tag{19}$$

The $\circ$ and $\oslash$ show the element-wise multiplication and division respectively. Five training sets (each consist of 30 samples) and test sets (each consist of 70 samples) were generated randomly. Fig. 2 and table 2 show the results. Each method graph is plotted with an offset from others, similar to the previous simulation. As can be seen, for this non-linear function, SSPCA obtained the best result while the worst result was for the SPLS method. That is, SPLS as a linear regression method, is not an appropriate method for non-linear data sets.

4.1.3. Simulation 3

The data matrix is $X_{sim3(400\times 30)}$. The non-linear function depends on variables $X(5,20)$:

$$Y_{sim3} = \exp(X_{sim3}(5)) - 2X_{sim3}(20) \circ X_{sim3}(20). \tag{20}$$

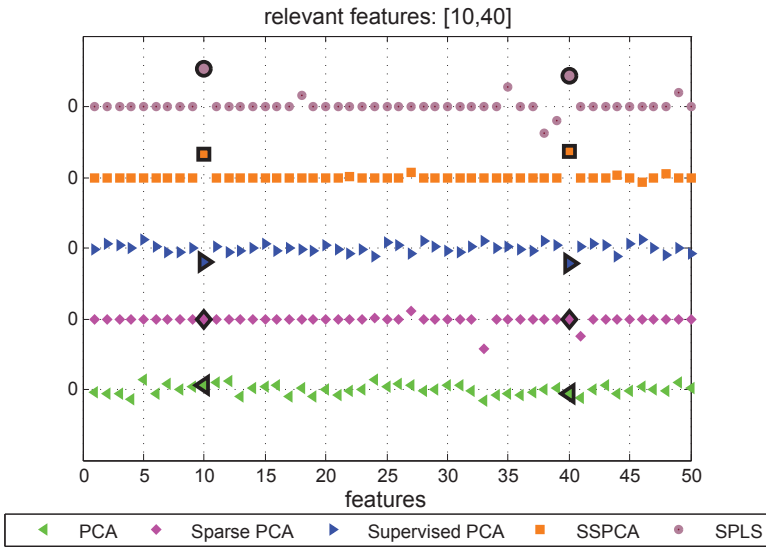


Figure 2: Comparison of the first Eigen vector/regression coefficients of the five tested methods on the second simulated data set. The black edged markers show the relevant features.

Table 3: Regression results for the third simulated data set. The average and standard deviations for five training and test sets are presented.

	PCA	SPCA	Sup. PCA	SSPCA	SPLS
$RMSE_{tr}$	3.55±0.44	3.40±0.54	2.54±0.30	2.59±0.25	3.02±0.35
$RMSE_{ts}$	3.61±1.16	3.51±1.13	2.85±0.70	2.75±0.75	3.36±0.97
Num. of NZ.	30.00±0.00	23.00±1.73	30.00±0.00	10.80±1.79	12.80±6.72

The data set was divided five times randomly into training (300 samples) and test (100 samples) sets. Fig. 3 and table 3 show the results.

4.2. Real data sets results

In this part of the report, three real data sets are considered and the five methods are tested on them. In all cases, the data sets were divided four times into training and test sets and the average results are considered.

4.2.1. Prediction of solvable solid content (SSC) of apple using spectroscopic measurements

The first real data set is the spectroscopic data of an apple type called *Rajka*. This is the same data set used in (Sharifzadeh et al., 2013). Spectroscopic mea-

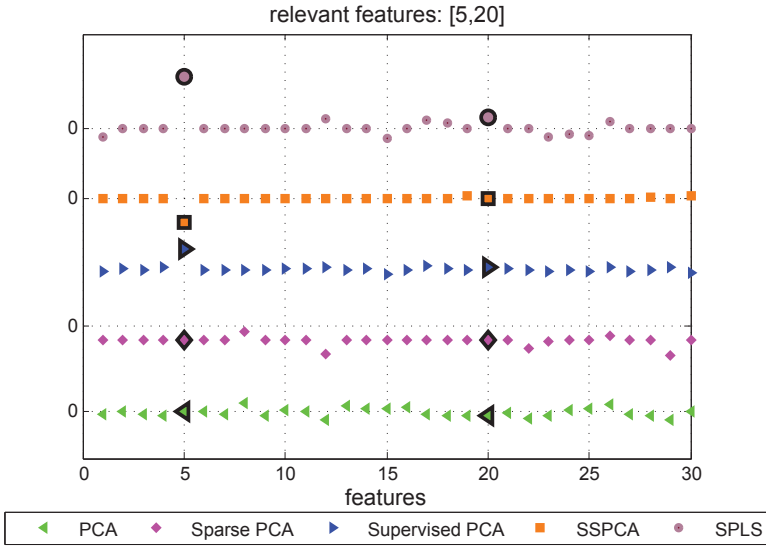


Figure 3: Comparison of the first Eigen vector/regression coefficients of the five tested methods on the third simulated data set.Each method graph is shifted up with an offset for better visualization. The black edged markers show the relevant features.

surements were performed in 825 wavelengths (306 -1130 nm) and there were 185 data points (apple samples) in total. In addition, the SSC (%Brix) value for each apple was available from laboratory measurements. We divided the data into training and test sets four times based on a systematic sampling method called a smooth arrangement or smooth fractionator (Gundersen, 2002). For this aim, the samples were ranked in ascending order according to the SSC level. Then, from every four samples, one was chosen as test (unseen data during training) and the rest as training. By using this method, both training and test sets comprise the original variation of the data.

In Fig. 4, the first three Eigen vectors of the first four methods are shown on the same plot together with the SPLS regression coefficients. The graphs are also shifted relative to each other similar to the previous illustrations. The average and standard deviation of regression results are presented in table 4. As can be seen, the proposed method is the best method in terms of accuracy and sparsity. SPLS and supervised PCA are the second best methods. However their number of used wavelengths are not comparable with the proposed method. All methods have a peak in the red colour area of the visible bands that corresponds to the apple colour.

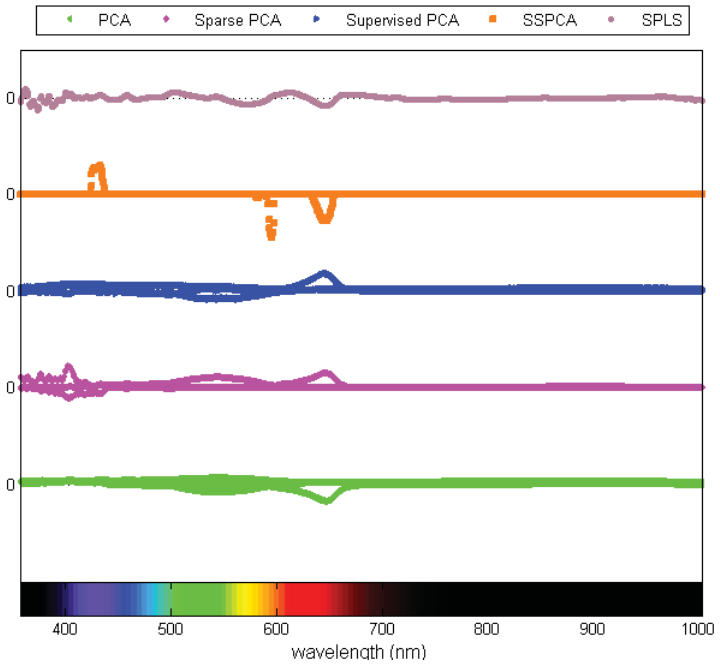


Figure 4: Comparison of the first three Eigen vector/regression coefficients of the five tested methods on the apple data set. Each method graph is shifted up with an offset and the corresponding vertical axis number is reset.

4.2.2. *Prediction of a* colour component for several meat types using multispectral images*

This data set consists of multispectral images of different types of meat, e.g. turkey, chicken, beef, veal and pork. This data was previously used in (Sharifzadeh et al., 2014). Totally, there were spectral images in 20 wavelengths (430-970 nm) and 52 meat samples. The median of the pixel values in a region of interest (ROI) was considered at each wavelength, forming a 52×20 matrix. In addition, the a* colour component of each sample was available from a Minolta colorimeter measurement. The data was divided randomly into training and test sets four times. In each data set, the number of training and test samples were 38 and 14 respectively.

The first three Eigen vectors of the first four methods are shown in the same plot together with the regression coefficients of SPLS in Fig. 5. β_{SPLS} is scaled in this plot. Here also, the graphs are visualized with an offset relative to each other.

Table 4: Regression results for the apple data set. The average and standard deviations for four training and test sets are presented.

	PCA	SPCA	Sup. PCA	SSPCA	SPLS
$RMSE_{tr}$	0.91±0.03	0.92±0.03	0.88±0.02	0.88±0.01	0.79±0.04
$RMSE_{ts}$	0.90±0.07	0.91±0.05	0.88±0.07	0.87±0.06	0.88±0.07
Num. of NZ.	825.00±0.00	439.75±177.86	825.00±0.00	149.00±202.38	778.25±48.93

Table 5: Regression results for the meat data set. The average and standard deviations for four training and test sets are presented.

	PCA	SPCA	Sup. PCA	SSPCA	SPLS
$RMSE_{tr}$	2.32±0.09	2.42±0.51	2.25±0.36	1.93±0.15	1.06±0.07
$RMSE_{ts}$	2.32±0.22	2.36±0.72	2.52±0.14	2.01±0.32	1.60±0.23
Num. of NZ.	20.00±0.00	11.75±6.18	20.00±0.00	9.25±3.20	18.50±1.73

The regression results are presented in table 5. As can be seen, SPLS obtained the best result in terms of accuracy and SSPCA is the second most accurate method. However, SSPCA is the best method in terms of sparsity. SPLS uses most of the 20 wavelengths on average. Reducing the number of wavelengths is important for a vision system design in industrial scale. Both the red colour wavelengths as well as the near infrared (NIR) bands are among the selected bands by the first three Eigen vectors of SSPCA. The red area corresponds to the colour of most meat types and NIR regions are correlated to their chemical characteristics.

4.2.3. Leukemia microarray classification and gene selection

The leukemia data set consist of 7129 genes and 72 samples (Golub et al., 1999). Previously it was used in (Zou and Hastie, 2005). There are two types of leukemia (acute lymphoblastic leukemia and acute myeloid leukemia). The goal is to predict the type of leukemia based on the expression level of those 7219 genes. In microarray analysis, it is important to diagnose the related genes to the disease. In our experiment, we divided the data into training and test sets four times based on the smooth fractionator method (Gundersen, 2002), so that, 75% of samples were chosen for training and the rest were kept for test. The percentages of classification performances using SVM and KNN as well as the number of selected genes are shown in table 6. In the case of SPLS, the predicted labels were assigned to the closest class labels. SVM obtained better results in most cases compared to KNN. The SVM classifier is built based on maximising the

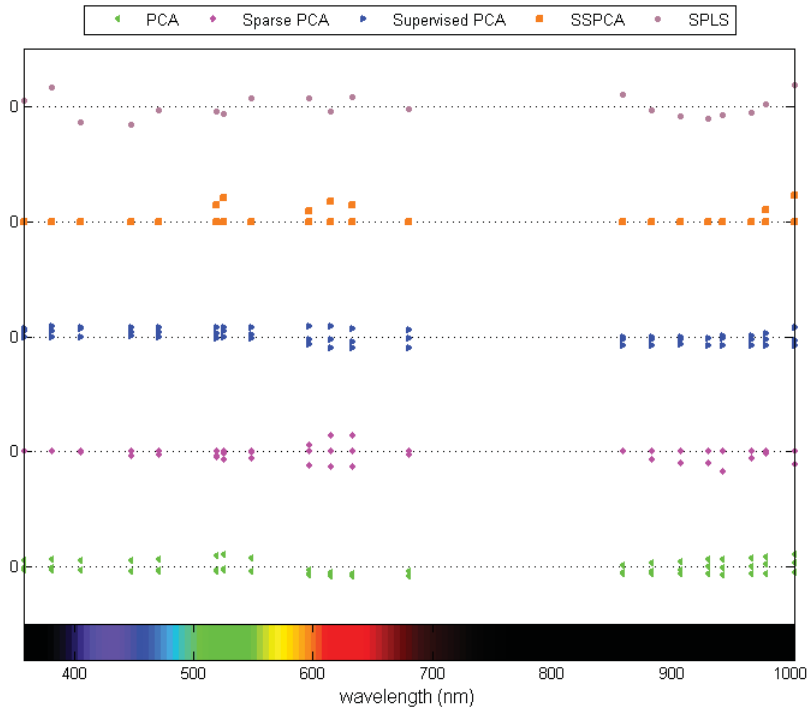


Figure 5: Comparison of the first three Eigen vectors/regression coefficients of the five tested methods on the meat data. Each method graph is shifted up with an offset and the corresponding vertical axis number is reset.

separating margins at the boundaries of classes, while KNN is based on majority vote of the K closest neighbours. Then, depending on the boundaries condition, SVM might be more successful, specially when the samples of classes are close in the boundary area and there is overlap between their features. PCA obtained the best classification rate using SVM classifier and all the genes, while the other methods performances come close to that. However, in terms of gene selection, the proposed method obtained an excellent result compared to the other methods and the performance obtained by SVM is comparable with the other techniques.

5. Discussion

The proposed method has been compared to four other techniques using various simulated and real data sets of different sample and variable sizes including both $N << P$ and $N >> P$ cases, successfully. A systematic assessment of the

Table 6: Classification results for the leukemia data set. The average and standard deviations for four training and test sets are presented.

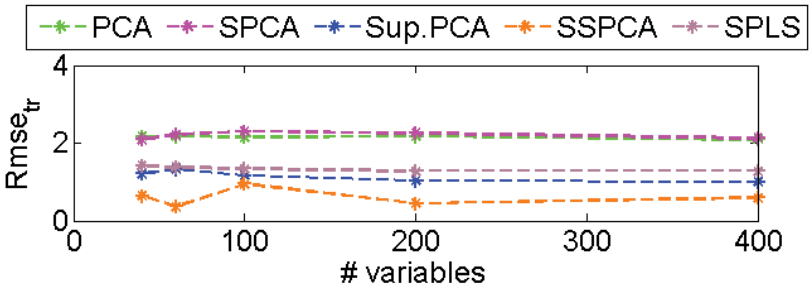
	Num. of NZ		PRF_{tr}	PRF_{ts}
PCA	7129.00±0.00	SVM	98.62±1.77	97.30±3.13
		KNN	95.42±4.36	90.42±9.46
SPCA	2618.25±405.38	SVM	100.00±0.00	94.44±7.86
		KNN	98.18±2.57	86.33±7.07
Sup. PCA	7129.00±0.00	SVM	98.63±0.91	94.52±7.86
		KNN	97.72±0.90	95.98±5.07
SSPCA	30.75±18.34	SVM	98.17±1.48	94.52±4.54
		KNN	98.18±3.67	89.04±4.55
SPLS	1630.50±1671.96	-	100.00±0.00	95.91±5.30

techniques performances based on different number of variables for a given sample size has achieved similar results. Fig. 6 shows such analysis results for the second simulated dataset; while the number of training samples were kept fixed at 100, the number of variables were changed as [40,60,100,200,400], forming ($N << P$, $N = P$ and $N >> P$) cases. The relevant features weren't changed. As can be seen, the behaviour of techniques remains the same, similar to the results shown previously in table 2.

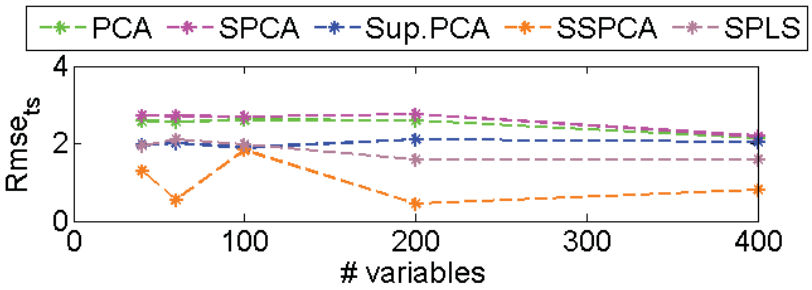
In the case of accuracy and sparsity, the experimental results has demonstrated that the proposed algorithm for SSPCA can make an appropriate trade-off between these two factors; In the first simulation, SPLS was the best method in terms of accuracy and sparsity as there was a pure linear relationship between X and Y. This is due to the linear kernel in its objective function. However, in the case of non-linear relationships, the second and third simulation results showed that SSPCA can perform better in terms of accuracy and sparsity.

The choice of kernel type and penalization parameter play an important role on the accuracy and sparsity of this method. When the kernel is close to data behaviour, the results can improve more. The sparsity of SSPCA was better than SPCA in almost all cases and its accuracy was better than supervised PCA in all experiments due to cancelling the effect of irrelevant and noisy variables. SSPCA also showed excellent sparsity for high dimensional data sets such as the apple and microarray data. The reasons for the success of SSPCA compared to SPCA will be discussed more in section 5.1.

Another important aspect of SSPCA algorithm, is its ability on choosing the



(a)



(b)

Figure 6: Illustration of the effect of change in the number of variables given a fixed number of training sample from the second simulation. (a) training RMSE (b) test RMSE.

Table 7: SVM analysis results on the selected features of the three real data sets.

	output	train	test	Num. of NZ.
apple	RMSE	0.87± 0.02	0.87± 0.07	149.00±202.37
meat	RMSE	1.66±0.17	2.53±0.74	9.25±3.20
lukemia	PRF.	97.68±4.63	93.13±2.83	30.75±18.34

relevant features. This can be used as a criterion to perform feature selection as a pre-processing step for different applications. To demonstrate this, the corresponding features to the non-zero rows of the first three Eigen vectors found by SSPCA are considered. Using the selected features (in original space), SVM was used for the regression and classification problems of the three real data sets. The training as well as test results are shown in table 7. As can be seen, the results are close to the obtained results in the orthogonal domain that have been presented in tables 4,5 and 6. This shows the relevance and dependency of the selected features to the target.

Regarding the complexity of the algorithms, PCA and Supervised PCA have closed form solutions and are less complex. Their complexity is $O(p^3, n^3)$. On the other hand, SPLS, SPCA and SSPCA are all solved based on iterative solutions to optimise biconvex objective functions. SPCA and SSPCA both use PMD and their biconvex optimisation loops are similar.Their complexity can be expressed as $O(K_1K_2(n^2p))$. K_1 is the number of components and is usually less than 5. K_2 is the number of iterations for convergence. For example, given a known kernel and c parameter, the number of iterations to achieve convergence in computations of an Eigen vector for the lukemia data set is 17.71±2.87. This was computed based on the average and standard deviation of required number of iterations in computation of 7 Eigen vectors. The biconvex solution of the sparse SIMSPLS objective function utilises the LARS algorithm at one of the optimisation steps (Chun and Keles, 2010). Then its complexity can be described as $O(K_1K_2(n^2p + p^3 + p^2n))$. In addition, the complexity increases when using a loop to find the best sparsity control parameters in the case of sparse solutions. Besides that, identification of the best kernel in the case of Supervised PCA and SSPCA is an additional complexity source for these algorithms

5.1. Comparison of sparsity between SSPCA, SPCA and SPLS

The objective function of the proposed SSPCA method is different from the objective function of SPCA, although both utilize PMD to solve the optimization. SPCA, is based on the singular value decomposition (SVD) of the input matrix X

while in SSPCA, Ψ is decomposed. In both cases an L_1 norm constraint is applied on the Eigen vectors and the resulting optimization problem is solved.

However, due to the structure of Ψ, SSPCA results in sparser Eigen vectors compared to SPCA. $\Psi = \Delta^T HX$ and $\Delta = U_L \Lambda_L^{\frac{1}{2}}$ where $L_{n\times n} = K(Y,Y) = U_L \Lambda_L V_L$. Based on these relationships, the absolute values of columns of Δ and therefore, the rows of Ψ decrease in a descending order following the descending order of the roots of Eigen values at the diagonal of $\Lambda_L^{\frac{1}{2}}$ as shown in Fig.7. For ease of visualization, a subset of 50×50 of the first simulation data was used for the pictures. As can be seen, due to the multiplication of the input matrix X by the supervision matrix Δ^T, the corresponding elements to the relevant features (5,15,25,35) are enhanced in Ψ and also its covariance matrix as shown in Fig.8(a). Comparison of the covariance matrices of Ψ and X demonstrates that while most of the corresponding elements to the irrelevant features are down weighted in the former, the latter does not show any discrimination for them. As a result, the relevant features in the first Eigen vector of Ψ have higher values compared to the other features. Therefore, the remaining features are numerically closer to zero due to the fact that the Eigen vectors are constrained to have unit length. This increases the chance of such low value elements to become zero in the smooth thresholding step of the PMD algorithm and hence, improves the potential of SSPCA algorithm in terms of sparsity. However, in the case of X, as can be seen, the distribution of values is random among the elements of Eigen vectors and the relevant features are not necessarily among the dominant values.

Besides the numerical values of the Eigen vectors, the sparsity level also depends on the threshold value c which is selected over the CV loops based on the average validation performance. Therefore, in general, with similar c values, SSPCA is sparser than SPCA. With unbalanced values of c (lower for SPCA), SPCA might become closer to SSPCA in terms of sparsity especially when the number of variables p is not very high, as seen in the second simulation and the meat data experiment. However, in the case of high number of variables $p \gg n$ such as apple or microarray experiments, the sparsity level of the proposed method dramatically increases and the differences are more prominent as many close to zero elements fall under the threshold. Fig. 9 illustrates the histogram plots of the first three Eigen vectors for apple data. As can be seen, the distribution of the numerical values of vectors is higher around zero for Ψ than X. That shows the potential of SSPCA Eigen vectors to be sparser than those of X at the smooth thresholding step of PMD.

In the case of SPLS, the objective function includes a linear kernel and the

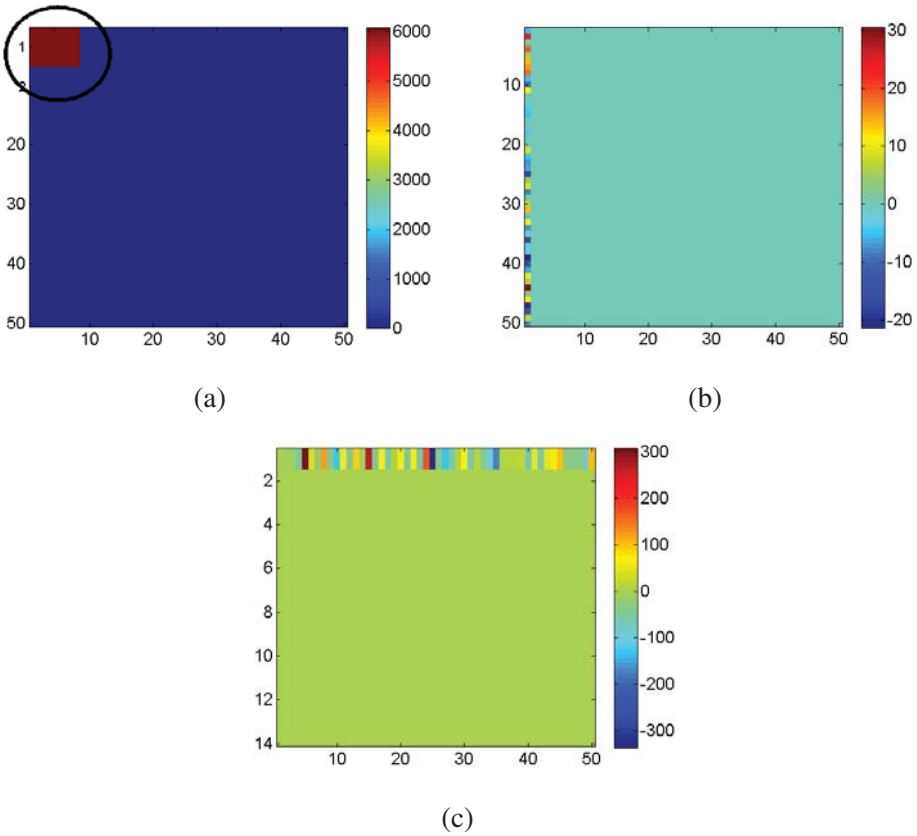


Figure 7: Illustration of (a) $\Lambda^{\frac{1}{2}}$ (b) Δ (c) Ψ matrices. In order to make the highest Eigen value visible, the top left area of the first plot is zoomed in.

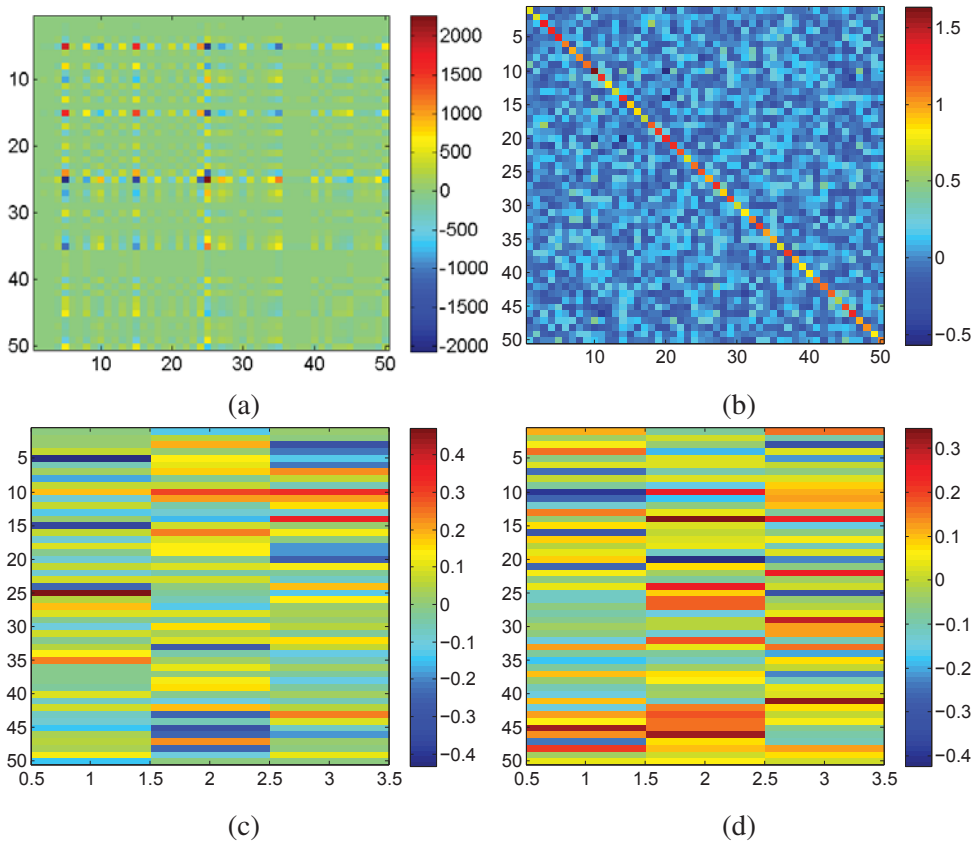
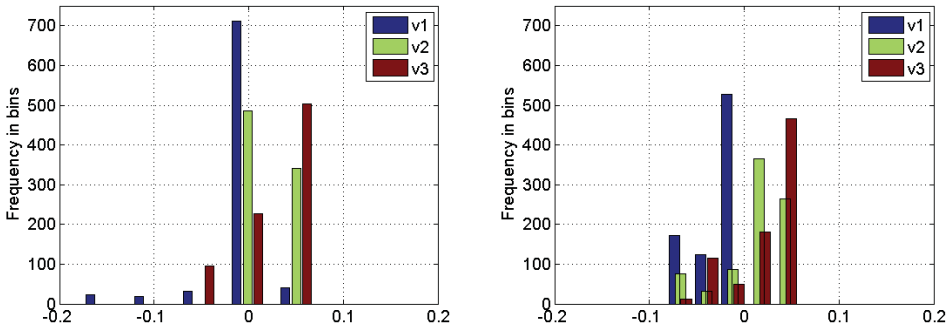


Figure 8: (a) The covariance matrix of Ψ (b) the covariance matrix of X. (c) from left to right the first three Eigen vectors of Ψ (d) the first three Eigen vectors of X.



(a) (b)

Figure 9: The histogram plots of the first three Eigen vectors of Ψ (a) , X (b). The spectroscopy data set of apples was used.

regression is also solved based on a biconvex optimization procedure. The constraint parameter λ is also chosen based on a CV loop. Therefore, the sparsity level is comparable with SSPCA when the relationship between X and Y is linear such as the first simulation.

6. Conclusion

In this paper, an SSPCA method was proposed for pre-processing of data sets with available target vectors. It computes sparse Eigen vectors based on the maximum dependency of the data to the response. The resulting Eigen vectors are almost orthogonal. The HSIC independence criterion was minimized between the input and output and a penalization term was added to make the Eigen vectors sparse. The objective function was solved based on the PMD algorithm. The SSPCA Eigen vectors are sparser compared to the PMD-based SPCA. Due to the use of the HSIC criterion in its objective function, this method can be used for data sets with linear as well as non-linear behaviour. Experimental results showed that SSPCA can make an appropriate compromise between accuracy and sparsity. Comparison of the results from PCA, PMD-based SPCA, supervised PCA, SSPCA and SPLS on both simulated and real data sets showed that SSPCA works best in terms of sparsity. The accuracy was also among one of the two best in all the experiments. In addition, the sparse Eigen vectors can be used as a means of feature selection, since the relevant features are among the non-zero rows.

Acknowledgement

This work was (in part) financed by the Centre for Imaging Food Quality project which is funded by the Danish Council for Strategic Research (contract no 09-067039) within the Program Commission on Health, Food and Welfare.

References

A. d'Aspremont, F. B., El Ghaoui, L., July 2008. Optimal solutions for sparse principal component analysis. Journal of Machine Learning Research 9, 1269–1294.

Asteris, M., Asteris, M., Dimakis, A. G., 2014. Nonnegative sparse pca with provable guarantees. In: Proceedings of 31th International Conference on Machine Learning.

Bair, E., Paul, D., Tibshirani, R., 2006. Prediction by supervised principal components. Journal of the American Statistical Association 101, 119–137.

Barshan, E., Ghodsi, A., Azimifar, Z., Jahromi, M. Z., 2011. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. Pattern Recognition 44 (7), 1357 – 1371.

Cadima, J., Jolliffe, I. T., 1995. Loading and correlations in the interpretation of principle compenents. Journal of Applied Statistics 22 (2), 203–214.

Chang, Ch. Ch.., Lin, Ch. J.., 2011. Libsvm: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 1 – 27.

Chen, X.,Wang, L.,Smith,J. D.,Zhang, B.,2008. Supervised principal component analysis for gene set enrichment of microarray data with continuous or survival outcomes. Bioinformatics 24 (21),2474–2448.

Chun, H., Keles, S., 2010. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. Journal of the Royal Statistical Society, Series B Stat Methodol, 3–25.

d'Aspremont, A., El Ghaoui, L., Jordan, M., Lanckriet, G., 2007. A direct formulation for sparse pca using semidefinite programming. SIAM Review 49 (3), 434–448.

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

Gretton, A., Bousquet, O., Smola, A., Schölkopf, B., 2005. Measuring statistical dependence with hilbert-schmidt norms. In: Algorithmic Learning Theory. Vol. 3734 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 63–77.

Gundersen, H., 2002. The smooth fractionator. Journal of microscopy, 191–210.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learning. Springer.

Jolliffe, I. T., Trendafilov, N. T., Uddin, M., 2003. A modified principal component technique based on the lasso. Journal of Computational and Graphical Statistics 12 (3), 531–547.

Journée, M., Nesterov, Y., Richtárik, P., Sepulchre, R., 2010. Generalized power method for sparse principal component analysis. Journal of Machine Learning Research 11, 517–553.

Lu, Z., Zhang, Y., 2009. An augmented lagrangian approach for sparse principal component analysis.

Magdon-Ismail, M., Boutsidis, C., 2016. Optimal sparse linear encoders and sparse pca. In: Preceedings of the 30th annual conference on neural Information processing systems (NIPS).

Moghaddam, B., Weiss, Y., Avidan, S., 2006. Spectral Bounds for Sparse PCA: Cact and Greedy Algorithms. MIT Press.

Sharifzadeh, S., 2015. Multivariate analysis techniques for optimal vision system design. Ph.D. thesis.

Sharifzadeh, S., Clemmensen, L. H., Borggaard, C., Støier, S., Ersbøll, B. K., 2014. Supervised feature selection for linear and non-linear regression of l*a*b* color from multispectral images of meat. Engineering Applications of Artificial Intelligence 27 (0), 211 – 227.

Sharifzadeh, S., Vega, M. V., Clemmensen, L. H. ., Ersbøll, B. K., 2013. Optimal vision system design for characterization of apples using uv/vis/nir spectroscopy data. In: In Proc. International Conference on Systems, Signals and Image Processing.

Shen, H., Huang, J. Z., Jul. 2008. Sparse principal component analysis via regularized low rank matrix approximation. J. Multivar. Anal. 99 (6), 1015–1034.

Sigg, C. D., Buhmann, J. M., 2008. Expectation-maximization for sparse and non-negative pca. In: Proceedings of the 25th International Conference on Machine Learning. ICML '08. ACM, pp. 960–967.

Witten, D. M., Hastie, T., Tibshirani, R., 2009. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics.

Zass, R., Shashua, A., 2007. Nonnegative sparse pca. In: Preceedings of the 20th annual conference on neural Information processing systems (NIPS).

Zelnik-manor, L., Perona, P., 2004. Self-tuning spectral clustering. In: Advances in Neural Information Processing Systems 17. MIT Press, pp. 1601–1608.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B 67, 301–320.

Zou, H., Hastie, T., Tibshirani, R., 2004. Sparse principal component analysis. Journal of Computational and Graphical Statistics 15, 2006.

Appendices

Considering the $PMD(.,L_1)$ problem for finding the individual sparse Eigen vectors v_k for SPCA:

$$\arg\max_{u_k,v_k} u_k^T X v_k, \text{s.t.} \|u_k\|_2^2 \leq 1, \|v_k\|_2^2 \leq 1, \|v_k\|_1 \leq c, u_k \perp u_1, \ldots, u_{k-1}, \quad \text{(A.1)}$$

a bi- convex optimization procedure can be used to solve this problem. Algorithm A.1 shows the procedure for finding K number of sparse Eigen vectors based on $PMD(.,L_1)$.

Algorithm A.1 Computation of K-factors of $PMD(.,L_1)$

1. Let $X^1 \leftarrow X$
2. For $k \in 1,\ldots,K$:

(a) Find u_k, v_k and d_k by applying the following single-factor PMD algorithm to X^k:

- Initialize v_k to have L_2-norm equal to one.
- Iterate until convergence:
$$\begin{cases} u_k \leftarrow & \arg\max_{u_k} u_k^T X^k v_k, \\ & \text{s.t.} \|u_k\|_2^2 \leq 1 \\ v_k \leftarrow & \arg\max_{v_k} u_k^T X^k v_k, \\ & \text{s.t.} \|v_k\|_1 \leq c \text{ and } \|v_k\|_2^2 \leq 1 \end{cases}$$
- $d_k \leftarrow u_k^T X^k v_k$

(b) $X^{k+1} \leftarrow X^k - d_k u_k v_k^T$

The optimization equations in this algorithm have a closed form solution. The parameters c is restricted to$1 \leq c \leq \sqrt{p}$. The smaller the c value, the more sparse the v_ks.

For the first optimization, v_k is considered as a fixed constant and $a = X^k v_k$, u_k is calculated based on the following steps:

$$\arg\max_{u_k} \left\| u_k^T a \right\| \text{s.t.} \|u_k\|_2^2 \leq 1, u_k \perp u_1, \ldots, u_{k-1}. \quad \text{(A.2)}$$

Then $u_k = U_{k-1}^{\perp}\theta$, so that $U_{k-1}^{\perp}$ is an orthogonal basis to $U_{k-1} = \{u_1, u_2, \ldots, u_{k-1}\}$ and $\|u\|_2 = \|\theta\|_2$:

$$\arg\max_{\theta} \theta^T U_{k-1}^{\perp T} X^k v_k, \text{s.t.} \|\theta\|_2^2 \leq 1, \tag{A.3}$$

and the optimal θ is:

$$\theta_{opt.} = \frac{U_{k-1}^{\perp T} X^k v_k}{\left\| U_{k-1}^{\perp T} X^k v_k \right\|_2}. \tag{A.4}$$

Therefore, u_k is found as:

$$u_k = \frac{U_{K-1}^{\perp} U_{k-1}^{\perp T} X^k v_k}{\left\| U_{k-1}^{\perp T} X^k v_k \right\|_2} = \frac{(I - U_{k-1} U_{k-1}^T) X^k v_k}{\left\| U_{k-1}^{\perp T} X^k v_k \right\|_2}. \tag{A.5}$$

This update step yields orthogonal factors for u_k. Similarly, in the second optimization step of the Algorithm A.1, u_k is considered as a fixed constant so that, $a = (X^k)^T u_k$. Then, we have:

$$\arg\max_{v_k} v_k^T a \quad \text{subject to,} \quad \|v_k\|_2^2 \leq 1, \|v_k\|_1 \leq c \ , \tag{A.6}$$

or the equivalent minimization:

$$\arg\min_{v_k} - v_k^T a \quad \text{s.t.} \quad \|v_k\|_2^2 \leq 1, \|v_k\|_1 \leq c \ . \tag{A.7}$$

The problem can be rewritten based on Lagrange multipliers:

$$-v_k^T a + \lambda \|v_k\|_2^2 + \tau \|v_k\|_1 , \tag{A.8}$$

and by setting the derivatives to zero and considering the Karush–Kuhn–Tucker conditions for optimality:

$$\begin{aligned} 0 &= -a + 2\lambda v_k + \tau \Gamma_k, \\ \lambda(\|v_k\|_2^2 - 1) &= 0, \\ \tau(\|v_k\|_1 - c) &= 0, \end{aligned} \tag{A.9}$$

where $\Gamma_k = sgn(v_k)$ if $v_k \neq 0$; otherwise, $\Gamma_k \in [-1, 1]$. If $\lambda > 0$, then from the first equation:

$$v_k = \frac{S(a, \tau)}{2\lambda}, \tag{A.10}$$

where S is the soft thresholding operator as described in Eq. 13. Generally, $\lambda = 0$ (if this results in a feasible solution) or it must be chosen so that, $\|v_k\|_2 = 1$:

$$v_k = \frac{S(a,\tau)}{\|S(a,\tau)\|_2}. \tag{A.11}$$

Again by the Karush–Kuhn–Tucker conditions, $\tau = 0$ (if this results in a feasible solution) or it must be chosen such that $\|v_k\|_1 = c$. Then, $\tau = 0$ if this results in $\|v_k\|_1 \leq c$; otherwise, it is chosen such that $\|v_k\|_1 = c$.