SPARSE LOADING NOISY PCA USING AN l_0 PENALTY

M.O. Ulfarsson[†] and V. Solo[‡]

[†]University of Iceland, Dept. Electrical Eng., Reykjavik, ICELAND
[‡]University of New South Wales, School of Electrical Eng., Sydney, AUSTRALIA

ABSTRACT

In this paper we present a novel model based sparse principal component analysis method based on the l_0 penalty. We develop an estimation method based on the generalized EM algorithm and iterative hard thresholding and an associated model selection method based on Bayesian information criterion (BIC). The method is compared to a previous sparse PCA method using both simulated data and DNA microarray data.

Index Terms— principal component analysis, sparsity, estimation.

1. INTRODUCTION

Sparse principal component analysis is a relatively recent extension of traditional principal component analysis (PCA). The idea is to seek sparse component loadings while explaining as much variance of the data set as possible.

There are two kinds of sparse PCA: sparse variable PCA (svPCA) [1, 2] and sparse loading PCA (slPCA) [3, 4, 5, 6, 7]. svPCA removes some of the original variables completely by simultaneously zeroing out all their loadings. slPCA keeps all of the original variables but zeroes out some of their loadings. Sparse PCA has, for example, been found useful for genomic data sets [4, 5, 6, 7] and medical imaging data [1, 2].

In this paper we develop a novel model based slPCA method using the l_0 penalty which we call sparse loading noisy PCA (slnPCA). An estimation method based on the generalized EM algorithm is constructed. We modify BIC [8] to choose the associated tuning parameters. The method is compared to the sparse loading PCA method in [4] by using simulations, and a DNA expression microarrays data set.

In section 2 we review noisy PCA (nPCA). The slnPCA method is described in section 3, where we develop an estimation and model selection method. We also give formulas for computing the explained variance of the slnPCA. Section 4 gives simulation examples and a DNA microarray data example. Finally in section 5 conclusions are drawn.

1.1. Notation

Vectors and matrices are denoted by boldface letters. $\boldsymbol{x} \sim N(\boldsymbol{m}, \boldsymbol{C})$ means that the random vector $\boldsymbol{x}$ is Gaussian distributed with mean $\boldsymbol{m}$ and covariance $\boldsymbol{C}$. $I(x \neq 0)$ is the indicator function that is equal to 1 if x is not equal to zero and equal to 0 else. If $\boldsymbol{G}$ is an $M \times r$ matrix we denote its column vectors by $\boldsymbol{g}_{(j)}, j = 1, ..., r$ and its row vectors by $\boldsymbol{g}_i^T, i = 1, ..., M$. $\odot$ is the Hadamard product.

2. NOISY PCA

Noisy PCA (nPCA) [9] is a structured covariance model whose maximum likelihood estimator yields PCA. The model is given by

$$\boldsymbol{y}_t = \boldsymbol{G}\boldsymbol{u}_t + \boldsymbol{\epsilon}_t, \quad t = 1, ..., T, \tag{1}$$

where $\boldsymbol{y}_t$ is a zero-mean $M \times 1$ vector, $\boldsymbol{G}$ is an $M \times r$ loading matrix, $\boldsymbol{\epsilon}_t \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_M)$, $\boldsymbol{u}_t \sim N(\boldsymbol{0}, \boldsymbol{I}_r)$, and $\boldsymbol{\epsilon}_t$ and $\boldsymbol{u}_t$ are uncorrelated. The normalized (divided by T) log-likelihood function is given by

$$l_{\boldsymbol{\theta}}(\boldsymbol{Y}) = -\frac{1}{2}\text{tr}(\boldsymbol{S}_y \boldsymbol{\Omega}^{-1}) - \frac{1}{2}\log|\boldsymbol{\Omega}|,$$

where $\boldsymbol{\Omega} = \boldsymbol{G}\boldsymbol{G}^T + \sigma^2 \boldsymbol{I}_M$, $\boldsymbol{Y} = [\boldsymbol{y}_1, ..., \boldsymbol{y}_T]^T$, $\boldsymbol{S}_y = \frac{1}{T}\boldsymbol{Y}\boldsymbol{Y}^T$ and $\boldsymbol{\theta} = (\text{vec}(\boldsymbol{G})^T, \sigma^2)^T$. It can be shown that the maximum likelihood estimates of $\boldsymbol{G}$ and σ^2 are given by

$$\begin{aligned} \hat{\boldsymbol{G}} &= \boldsymbol{P}_r(\boldsymbol{L}_r - \hat{\sigma}^2 \boldsymbol{I}_r)^{1/2}\boldsymbol{R}^T \\ \hat{\sigma}^2 &= \frac{1}{M-r}\sum_{j=r+1}^{M} l_j, \end{aligned}$$

where $\boldsymbol{L}_r = \text{diag}(l_1, ..., l_r)$ is a diagonal matrix containing the r largest eigenvalues of $\boldsymbol{S}_y$; $\boldsymbol{P}_r$ contains the r first eigenvectors of $\boldsymbol{S}_y$ in its columns; $\boldsymbol{R}$ is an arbitrary rotation matrix.

The estimated noisy principal components (nPCs) $\hat{\boldsymbol{U}} = [\hat{\boldsymbol{u}}_1, ..., \hat{\boldsymbol{u}}_T]^T$ are given by $\hat{\boldsymbol{U}} = \boldsymbol{Y}\hat{\boldsymbol{G}}\boldsymbol{W}^{-1}$, where $\boldsymbol{W} = \hat{\boldsymbol{G}}^T\hat{\boldsymbol{G}} + \hat{\sigma}^2\boldsymbol{I}_r$.

3. SPARSE LOADING NPCA

We obtain sparse solutions by penalizing the number of entries in $\boldsymbol{G}$. The penalized likelihood is then

$$J_{\boldsymbol{\theta}}(\boldsymbol{Y}) = -l_{\boldsymbol{\theta}}(\boldsymbol{Y}) + \frac{h}{2}\|\boldsymbol{G}\|_0,$$

where $\|\boldsymbol{G}\|_0 = \sum_{v=1}^{M}\sum_{j=1}^{r} I(g_{v,j} \neq 0)$ and $g_{v,j}$ is the v,j-th element of $\boldsymbol{G}$. This should not be confused with our earlier work [2] where we zero out rows of $\boldsymbol{G}$ all at once; here the penalty zeros out individual entries in $\boldsymbol{G}$.

3.1. Estimation

We maximize the sparse loading nPCA (slnPCA$_0$) criterion with the generalized EM algorithm. The penalized normalized complete likelihood is

$$J_{\boldsymbol{\theta}}(\boldsymbol{Y},\boldsymbol{U}) = l_{\boldsymbol{\theta}}(\boldsymbol{Y},\boldsymbol{U}) - \frac{h}{2}\|\boldsymbol{G}\|_0,$$

where the complete normalized log-likelihood is given by

$$\begin{aligned} l_{\boldsymbol{\theta}}(\boldsymbol{Y},\boldsymbol{U}) &= -\frac{M}{2}\log\sigma^2 - \frac{1}{T}\sum_{t=1}^{T}\frac{\|\boldsymbol{y}_t - \boldsymbol{G}\boldsymbol{u}_t\|^2}{2\sigma^2} \\ &- \frac{1}{2T}\sum_{t=1}^{T}\boldsymbol{u}_t^T\boldsymbol{u}_t \end{aligned}$$

In the E-step we compute the penalized EM functional

$$\text{EM}_p(\boldsymbol{\theta}_0,\boldsymbol{\theta}) = \text{EM}(\boldsymbol{\theta}_0,\boldsymbol{\theta}) - \frac{h}{2}\|\boldsymbol{G}\|_0,$$

where the EM functional is given by

$$\begin{aligned} \text{EM}(\boldsymbol{\theta}_0,\boldsymbol{\theta}) &= E_{\boldsymbol{\theta}_0}(l_{\boldsymbol{\theta}}(\boldsymbol{Y},\boldsymbol{U})|\boldsymbol{Y}) \\ &= -\frac{\text{tr}(\boldsymbol{S}_y)}{2\sigma^2} + \frac{\text{tr}(\boldsymbol{G}\boldsymbol{B}_0^T)}{\sigma^2} - \frac{\text{tr}(\boldsymbol{G}\boldsymbol{A}_0\boldsymbol{G}^T)}{2\sigma^2} \\ &- \frac{M}{2}\log\sigma^2, \end{aligned}$$

where $\boldsymbol{A}_0 = \sigma_0^2\boldsymbol{W}_0^{-1} + \frac{1}{T}\boldsymbol{U}_0^T\boldsymbol{U}_0$ and $\boldsymbol{B}_0 = \frac{1}{T}\boldsymbol{Y}^T\boldsymbol{U}_0$. In the M-step we maximize the penalized EM functional over $\boldsymbol{G}$ which is equivalent to minimizing

$$J(\boldsymbol{G}) = \sum_{v=1}^{M}\left(\frac{1}{2}\boldsymbol{g}_v^T\boldsymbol{A}_0\boldsymbol{g}_v - \boldsymbol{g}_v^T\boldsymbol{b}_{v,0} + \frac{h\sigma^2}{2}\|\boldsymbol{g}_v\|_0\right),$$

where $\boldsymbol{g}_v^T$ is the v-th row vector of $\boldsymbol{G}$. We use an iterative algorithm (see Algorithm 2 at the end of the paper) to solve this optimization problem, denoting the solution by $\Gamma(\boldsymbol{A}_0,\boldsymbol{B}_0,\sigma^2,h)$. Optimization of the penalized EM functional w.r.t. σ^2 yields

$$\sigma_1^2 = \frac{1}{M}\left[\text{tr}(\boldsymbol{A}_0\boldsymbol{G}_1^T\boldsymbol{G}_1) - 2\text{tr}(\boldsymbol{B}_0^T\boldsymbol{G}_1) + \text{tr}(\boldsymbol{S}_y)\right].$$

We say the algorithm has converged if

$$1 - \min_j\left(\frac{|\boldsymbol{g}_{(j),k}^T\boldsymbol{g}_{(j),k-1}|}{\|\boldsymbol{g}_{(j),k}\|\|\boldsymbol{g}_{(j),k-1}\|}\right) < 1\cdot 10^{-5}.$$

The slnPCA$_0$ algorithm is given in Algorithm 1.

Algorithm 1: The slnPCA$_0$ algorithm

Input: Data matrix $\boldsymbol{Y}$, r, and h
Initialization: $\boldsymbol{G}_0$ and σ_0^2
while *(Not converged)* **do**
- $\boldsymbol{W}_k = \boldsymbol{G}_k^T\boldsymbol{G}_k + \sigma_k^2\boldsymbol{I}_r$
- $\boldsymbol{U}_k = \boldsymbol{Y}\boldsymbol{G}_k\boldsymbol{W}_k^{-1}$
- $\boldsymbol{A}_k = \sigma_k^2\boldsymbol{W}_k^{-1} + \frac{1}{T}\boldsymbol{U}_k^T\boldsymbol{U}_k$
- $\boldsymbol{B}_k = \frac{1}{T}\boldsymbol{Y}^T\boldsymbol{U}_k$
- $\boldsymbol{G}_{k+1} = \Gamma(\boldsymbol{A}_k,\boldsymbol{B}_k,\sigma_k^2,h)$
- $\sigma_{k+1}^2 =$ $\frac{1}{M}\left[\text{tr}(\boldsymbol{A}_k\boldsymbol{G}_{k+1}^T\boldsymbol{G}_{k+1}) - 2\text{tr}(\boldsymbol{B}_k^T\boldsymbol{G}_{k+1}) + \text{tr}(\boldsymbol{S}_y)\right]$

Output: $\boldsymbol{G}$ and σ^2

3.2. Model selection

We use the BIC statistic to select the number of principal components r and the sparseness tuning parameter h.

$$\text{BIC}_{r,h} = -2l_{\hat{\boldsymbol{\theta}}}(\boldsymbol{Y}) + \frac{d_{r,h}}{T}\log T,$$

where $d_{r,h}$ is the number of nonzero elements of $\boldsymbol{G}$; note that this also depends on h. The model corresponding to the global minimum of the BIC surface is selected.

3.3. Variance explained and ordering

We define the variance explained by the sparse noisy principal components as

$$\text{tr}(\boldsymbol{Q}^T\boldsymbol{S}_y\boldsymbol{Q}) \tag{2}$$

where $\boldsymbol{Q} = \boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{G}+\sigma^2\boldsymbol{I}_r)^{-1/2}$. The columns of the sparse loading matrix $\boldsymbol{G}$ are ordered according to their contribution in the sum (2).

4. EXAMPLES

In this section we provide two simulation examples comparing slnPCA$_0$ to the method in [4], which we will call ZHT, using an implementation by [10]. We also provide a DNA microarray data example. In simulation 2 and in DNA microarray data example the $n >> p$ (gene expression array) version of ZHT is used.

4.1. Simulation 1

We simulated $T = 100$ data vectors from (1) with $\boldsymbol{G}_{10\times 2} = [\boldsymbol{g}_{(1)},\boldsymbol{g}_{(2)}]$, where $\boldsymbol{g}_{(1)} = \sqrt{200}\cdot\tilde{\boldsymbol{g}}_{(1)}/\|\tilde{\boldsymbol{g}}_{(1)}\|$, $\boldsymbol{g}_{(2)} = \sqrt{100}\cdot\tilde{\boldsymbol{g}}_{(2)}/\|\tilde{\boldsymbol{g}}_{(2)}\|$,

$$\begin{aligned} \tilde{\boldsymbol{g}}_{(1)} &= [1,1,1,1,0,0,0,0,0,0]^T \\ \tilde{\boldsymbol{g}}_{(2)} &= [0,0,0,0,1,1,1,1,0,0]^T \end{aligned}$$

	angle 1 (deg)	angle 2 (deg)
slnPCA$_0$	2.15	3.10
ZHT	2.49	3.49

Table 1. Simulation 1. The angles (in degrees) between the estimated and true loadings.

and $\sigma^2 = 10$. The simulation was run 10 times. For each run BIC was used to select the number of principal components r and the sparseness tuning parameter h. BIC picked the correct model each time. Fig. 1 shows a sample BIC profile versus the sparseness parameter h. The model corresponding to the local minimum at $h = 0.039$ was selected. The ZHT

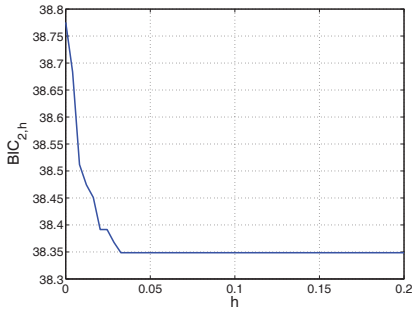
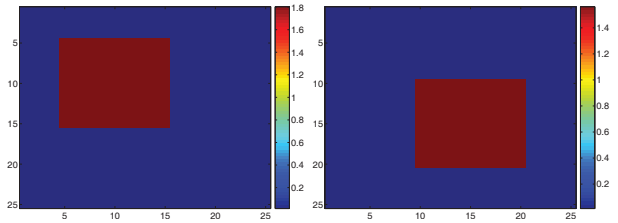


Fig. 1. Simulation 1. The BIC$_{2,h}$ profile for slnPCA$_0$.

method is not model based so the BIC cannot be used to select its tuning parameter. To compare we used the same r and sparsity level as BIC selected for the slnPCA$_0$ method. We computed the mean angle (in degrees) between the estimated loading vectors $\hat{\boldsymbol{g}}_{(1)}$, $\hat{\boldsymbol{g}}_{(2)}$ and the truth over 10 simulations. Table 1 shows that angle is lower for the slnPCA$_0$ method.

4.2. Simulation 2

We simulated $T = 100$ data vectors from the (1) with $\boldsymbol{G}_{625\times 2} = [\boldsymbol{g}_{(1)}, \boldsymbol{g}_{(2)}]$, where the loading vectors $\boldsymbol{g}_{(1)}$ and



(a) $\boldsymbol{g}_{(1)}$ (b) $\boldsymbol{g}_{(2)}$

Fig. 2. Simulation 2. The loading vector $\boldsymbol{g}_{(1)}$ and $\boldsymbol{g}_{(2)}$ depicted as 25×25 images.

$\boldsymbol{g}_{(2)}$ are depicted (as 25×25=625 images) in Fig. 2 and $\sigma^2 = 5$. We simulated the data 44 times and each time used the BIC to select the tuning parameters. Fig. 3 shows a plot of the probability of error for each of those simulations where probability of error was defined as

$$P_E = \alpha P_F + (1-\alpha)(1-P_D)$$

, where $P_D = \Pr(\text{decide signal}|\text{signal})$, $P_F = \Pr(\text{decide signal}|\text{no signal})$, and α is the fraction of zero loadings. For each simulation BIC was used to select the slnPCA$_0$ model. The model corresponding to the lowest probability of error was selected for ZHT. The figure shows that probability of error is almost always much lower for slnPCA$_0$. Fig. 4 shows a sample BIC profile vs h where the model corresponding to $h = 0.28$ was selected.

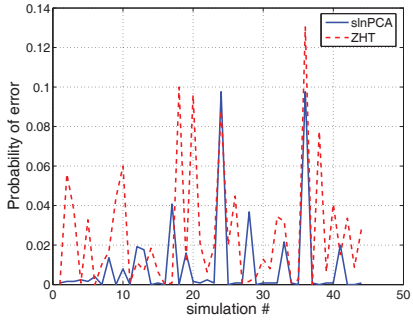


Fig. 3. Simulation 2. Probability of error vs simulation number.

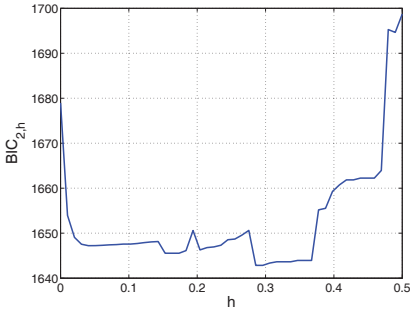
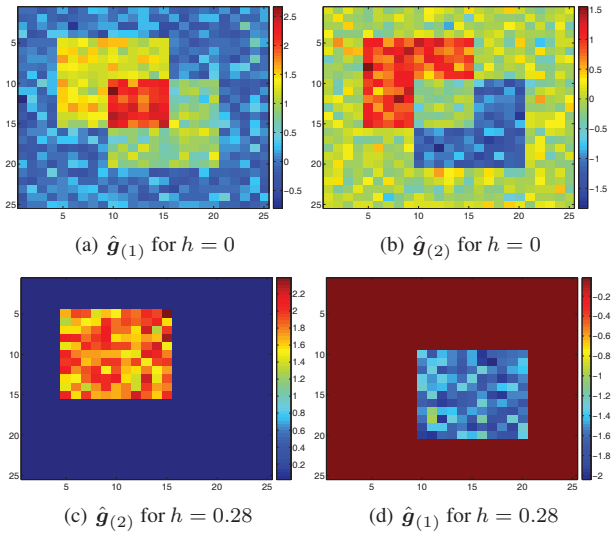


Fig. 4. Simulation 2. The BIC$_{2,h}$ profile for slnPCA$_0$.

Fig. 5 shows the slnPCA$_0$ solution for $h = 0$ (no penalty) which is equal to the nPCA solution described in section 2 and the slnPCA$_0$ solution selected by BIC. It can be seen that BIC selects the correct degree of sparseness. Interestingly, since the true loading vectors are correlated, the nPCA solution is not able to separate them. On the contrary the slnPCA$_0$ solution separates them.

The ZHT method, with the degree of sparseness, and r corresponding to minimum probability of error was used to estimate the loadings. The angle between the loading vectors and the truth can be seen in Table 2. Just as in simulation 1 the slnPCA$_0$ gives lower values.



(a) $\hat{\boldsymbol{g}}_{(1)}$ for $h = 0$ (b) $\hat{\boldsymbol{g}}_{(2)}$ for $h = 0$

(c) $\hat{\boldsymbol{g}}_{(2)}$ for $h = 0.28$ (d) $\hat{\boldsymbol{g}}_{(1)}$ for $h = 0.28$

Fig. 5. Simulation 2. The estimated loadings

	angle 1 (deg)	angle 2 (deg)
slnPCA$_0$	7.6	11.7
ZHT	14.8	23.9

Table 2. Simulation 2. The angles between the estimated and true loadings.

4.3. DNA microarray expression data

In this section we analyze DNA microarray expression data matrix of M=6830 genes and $T = 64$ samples [11]. We extract the first principal component using slnPCA$_0$ and also the ZHT method. Fig. 6 shows the variance explained vs the level of sparsity, the level of sparsity chosen by BIC is indicated with a dot on the figure. slnPCA$_0$ clearly explains more variance for a given level of sparsity than ZHT.

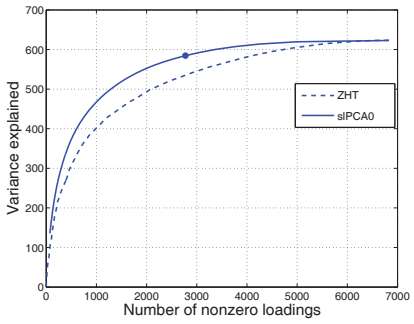


Fig. 6. The variance explained for the microarray expression data.

5. CONCLUSIONS

We have developed a new sparse loading PCA method based on an l_0 penalized likelihood framework. Optimization is facilitated by a generalized EM algorithm which yields a simple thresholding procedure. A simulation study and real data analysis show that the method outperforms the standard procedure.

Algorithm 2: Iterations for $\Gamma(\boldsymbol{A}_0, \boldsymbol{B}_0, \sigma^2, h)$.

Input: Data matrix $\boldsymbol{A}_0 = [A_{ij}]$, $\boldsymbol{B}_0 = [\boldsymbol{b}_{(1)}, ..., \boldsymbol{b}_{(r)}]$, σ^2, h.

Initialization: $\boldsymbol{G}_0 = [\boldsymbol{g}_{(1),0}, ..., \boldsymbol{g}_{(r),0}]$

while *(Not converged)* **do**

  for $i = 1, ..., r$ **do**

    $\boldsymbol{r}_{(i),k+1} = \boldsymbol{b}_{(i)} - \sum_{j<i} A_{ij}\boldsymbol{g}_{(j),k+1} - \sum_{j>i} A_{ij}\boldsymbol{g}_{(j),k}$

    $\boldsymbol{g}_{(i),k+1} = \frac{\boldsymbol{r}_{(i),k+1}}{A_{ii}} I\left(\frac{\boldsymbol{r}_{(i),k+1} \odot \boldsymbol{r}_{(i),k+1}}{A_{ii}} > h\sigma^2\right)$

Output: $\boldsymbol{G}$

6. REFERENCES

[1] M.O. Ulfarsson and V. Solo, "Sparse variable PCA using geodesic steepest descent," *IEEE Transactions on Signal Processing*, vol. 56, no. 12, pp. 5823–5832, 2008.

[2] M.O. Ulfarsson and V. Solo, "Vector l_0 sparse variable pca," *IEEE Trans. Signal Proc.*, vol. 59, no. 5, pp. 1949–1958, 2011.

[3] I.T. Jolliffe and M. Uddin, "A modified principal component technique based on the lasso," *J. Comput. Graphical Stat.*, vol. 12, no. 3, pp. 531–547, 2003.

[4] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graphical Stat.*, vol. 15, no. 2, pp. 265–286, 2006.

[5] A. D'Aspremont, F. Bach, and L.E. Ghaoui, "Optimal solutions for sparse principal component analysis," *J. Mach. Learn. Res.*, vol. 9, pp. 1269–1294, 2008.

[6] J. Huang H. Shen, "Sparse principal component analysis via regularized low rank matrix approximation," *J. Multivariate Anal.*, vol. 99, pp. 1015–1034, 2008.

[7] Y. Guan and J. Dy, "Sparse probabilistic principal component analysis," in *AISTATS 2009*.

[8] G. Schwartz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.

[9] D.N. Lawley, "Tests of significance of the latent roots of the covariance and correlation matrices," *Biometrica*, vol. 43, pp. 128–136, 1956.

[10] K. Sjostrand, "Matlab implementation of LASSO, LARS, the elastic net and SPCA," 2005, Version 2.0.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Verlag, New York, NY, 2001.