SPARSE VARIABLE PCA USING A STEEPEST DESCENT ON A GRASSMAN MANIFOLD

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

Recently there has developed considerable interest in using sparseness with PCA. Almost all previous methods concentrate on zeroing out some loadings. Here we develop a new approach which zeros out whole variables automatically. We formulate a vector l_1 penalized PCA criterion and optimize it by steepest descent along geodesic on a Grassman manifold. This ensures that each step obeys PCA orthogonality as well as an invariance property of the criterion. We show in simulations that it outperforms a previous svPCA algorithm and apply it to a real high dimensional functional Magnetic Resonance Imaging (fMRI) data.

***Index Terms*—** Principal component analysis, fMRI, Optimization on a manifold.

1. INTRODUCTION

Principal Component Analysis (PCA) is useful for analyzing large data sets that, e.g., arise in applications involving functional Magnetic Resonance Imaging (fMRI) data. Given T observations on M variables, PCA expresses the data in terms of new uncorrelated variables named principal components. The first Principal Component (PC) is the linear combination of the data variables that maximizes the variance. The second PCs is orthogonal to the first while maximizing the variance etc. Often the variability of the data can be captured into few PCs leading to substantial data reduction.

A typical high dimensional data set consists of a relatively few observations on very many variables. For instance, fMRI data consists of time series of brain images where the number of observations is on the order of hundreds while the number of volume elements (voxels) or variables is typically on the order of thousands. In this application, some of the variables only represent noise and should be kept out of the analyzes. The purpose of this paper is to develop a PCA method that automatically zeros out irrelevant variables.

We identify two kinds of methods that use sparseness in connection with PCA. Firstly, sparse loading PCA (slPCA) where all variables are retained but each variable may have some of its loading zeroed out. Examples of slPCA are given in [1, 2, 3, 4]. Secondly, sparse variable PCA (svPCA) where whole variables are removed by simultaneously zeroing out all their loadings. Note that slPCA and svPCA thus aim to solve totally different problems. Previously, [5], Johnstone and Lu suggested a simple svPCA algorithm that thresholds low variance variables followed by using a conventional PCA on the rest of the variables. In our previous work [6, 7], an svPCA algorithm called svnPCA based on a penalized likelihood formulation was described. It relies on Lawley's [8, 9] stochastic model for PCA. This optimization was carried out on a Stiefel manifold. These two methods are really aimed at doing totally different things; slPCA is probably best suited to small data problems and svPCA to large data problems. In addition to the slPCA and svPCA methods described above, we mention that [10] develops a sparse PCA method based on rotating PCs, and [11] develops a sparse kernel PCA method.

In this paper we introduce an svPCA method based on solving an orthogonality constrained optimization problem. The cost function possesses a vector l_1 penalty that can produce sparseness and has symmetry that allows for optimization on the Grassman manifold. We propose to use the steepest descent algorithm on the Grassman manifold to solve the optimization problem.

Following introduction in Section 1, we review PCA in Section 2. Section 3 introduces the new svPCA method. Section 4 discusses how we estimate the svPCs, and Section 5 covers the associated tuning parameter selection. In Section 6 the svPCA method is applied to a simulated data set and a real fMRI data set. Finally, in Section 7, conclusions are drawn.

2. PRINCIPAL COMPONENT ANALYSIS

Let us assume the data consists of T observations on M variables and is contained in a $T \times M$ matrix $Y = [y_t^T] = [y_{(v)}]$. Furthermore, assume throughout the paper that the data is mean centered. The PCs are defined as $z_t = P^T y_t, \quad t = 1, ..., T$ where P is the $M \times M$ eigenvector matrix of the data covariance $S_y = \frac{1}{T}\sum_{t=1}^{T} y_t y_t^T$. These eigenvectors can be efficiently computed via the Singular Value Decomposition (SVD) of the data $\frac{1}{\sqrt{T}}Y = QL^{1/2}P^T$ where Q is $T \times T$ orthonormal matrix, P is $M \times M$ orthonormal matrix, and L is $T \times M$ diagonal matrix of singular values.

PCA is posed [12] as the solution to the following problem

$$\begin{array}{ll} \text{minimize} & J(F) = -\frac{1}{2}\text{tr}(F^T S_y F) \\ \text{subject to} & F^T F = I_r \end{array} \quad (1)$$

The optimal $M \times r$ matrix F is the r first columns of the eigenmatrix P. The optimal value of (1) is $J(F) = -\frac{1}{2}\sum_{j=1}^{r} l_r$. In the following we penalize (1) to encourage a sparse variable property.

3. SPARSE VARIABLE PCA

svPCA is defined as the following penalized optimization problem

$$\begin{array}{ll} \text{minimize} & J(F) = -\frac{1}{2c}\text{tr}(F^T S_y F) + \frac{h}{M}\rho(F) \\ \text{subject to} & F^T F = I_r \end{array}$$

where $c = \text{tr}(S_y)$. Before we specify the penalty let us consider what needs to be done to zero out whole variable v. Typically, the data can be represented by few PCs. This means that a variable v can be written as $y_{(v)} \approx Z f_v$ where Z is a $T \times r$ matrix of the PCs and f_v is the v-th row vector of F. The variable y_v is zeroed out when $f_v = 0$. Therefore the penalty should be designed to do that.

978-1-4244-2354-5/09/$25.00 ©2009 IEEE

3.1. The penalty function

As in [6, 7] we propose to use the vector l_1 penalty $\rho(F) = \sum_{v=1}^{M} \|f_v\|_2$ where $\|f_v\|_2 = \sqrt{\sum_{j=1}^{r} f_{v,j}^2}$. Since this penalty has discontinuous derivative (only) for $f_v = 0$ it can produce zeroing of f_v. However, note that the penalties $\sum_{v=1}^{M} \|f_v\|_p$ for $p > 2$ can also provide zeroing. But since 1) $p = 2$ provides rotational invariance, which is a property which will be useful in the estimation section below, and 2) $p = 2$ is the strongest penalty for all $p \geq 2$, we pick $p = 2$. Interestingly, the special case of $p = 1$ also provides sparseness, but it does not simultaneously zero out all elements of f_v. Actually, $p = 1$ leads to slPCA.

In a very different context this kind of penalty has been independently introduced under the name of group Lasso [13]. This kind of penalty is also long known in total variation denoising, but there it is used to regularize derivatives not amplitudes.

Our estimation strategy requires the cost function to be smoothly differentiable which is not the case since the derivative of $\rho(F)$ is discontinuous at zero. We use a standard trick [14] and approximate the non-smooth penalty with a smooth one:

$$\rho_\gamma(F) = \sum_{v=1}^{M} \sqrt{\|f_v\|_2^2 + \gamma^2}$$

where γ is a small constant. Therefore the svPCA problem consists of solving the following optimization problem:

$$\begin{array}{ll} \min & J(F) = -\frac{1}{2c}\text{tr}(F^T S_y F) + \frac{h}{M}\sum_{v=1}^{M} \sqrt{\|f_v\|_2^2 + \gamma^2} \\ \text{s.t} & F^T F = I_r. \end{array} \quad (2)$$

4. OPTIMIZATION BY A GEODESIC STEEPEST DESCENT ON A GRASSMAN MANIFOLD

The optimization problem (2) has no closed form solution except for the special case of $h = 0$, i.e., the PCA problem. Therefore we need to resort to iterative algorithms. A traditional iterative method to solve optimization problems is the steepest descent algorithm. In that method the solution is updated at each step by moving it in the direction of the negative gradient. However in our case this approach does not work since we require the solution to satisfy the orthogonality constraints $F^T F = I_r$ and at each step the update would step off the constraint surface.

Luenberger [15] pioneered a general approach to apply steepest descent to constrained optimization problems or optimization on manifolds. Instead of moving in straight line in the direction of the negative gradient we move along a geodesic on the constraint surface, i.e., satisfy the constraint in each step. A general form of a geodesic steepest descent algorithm is the following:

Geodesic steepest descent algorithm

i. Compute the geodesic $F(\theta)$ emanating from the negative gradient $-\nabla J$.

ii. Compute the θ^* that minimizes $J(F(\theta))$

iii. Update $F = F(\theta^*)$, if not converged return to step i.

This kind of geodesic steepest gradient is, in general, not practical due to a high computation cost. A system of linear differential equations (Euler-Lagrange equations) has to be solved in ii. In this paper we have orthogonality constraints which were treated in general in [16]. In that case the computation of the geodesic in i. is relatively quick and simple.

4.1. Stiefel and Grassman manifolds

The orthogonality constraint $F^T F = I_r$ defines a Stiefel manifold [17]. So the optimization is at least on a Stiefel manifold. However unlike the criterion in [7] our new criterion possesses the additional homogeneity condition $J(FQ) = J(F)$ where Q is an arbitrary $r \times r$ unitary matrix.In this case the constraint surface is called the Grassman manifold.

4.2. Grassman gradient

The Grassman gradient ∇J is the projection of the (unconstrained) gradient J_F onto the tangent plane at F [16]:

$$\nabla J = (I_M - FF^T)J_F \quad (3)$$

where $J_F = -\frac{1}{c}S_y F + \frac{h}{M}DF$ and
$D = \text{diag}((\|f_1\|^2 + \gamma^2)^{-1/2}, ..., (\|f_M\|^2 + \gamma^2)^{-1/2})$.

4.3. Grassman geodesic

A smooth curve $F(\theta), 0 \leq \theta \leq N$ starting at $F(0)$ and terminating at $F(N)$ minimizing the path length between the two points is called a geodesic. The Grassman geodesic $F(\theta)$ starting at F emanating from $-\nabla J$ is given by [16]

$$F(\theta) = (FV\cos(\Sigma\theta) + U\sin(\Sigma\theta))V^T$$

where $U\Sigma V^T$ is the compact SVD of the negative Grassman gradient $-\nabla J$.

4.4. Normalized Grassman gradient

Let s be the path length along the geodesic from the initial point F to $F(\theta)$, it can be shown that $s = \theta\|\nabla J\|$ where $\|A\| = \text{tr}(A^T A)$. Thus we can normalize the geodesic by re-parameterizing it in terms of the path length by writing $\theta = s/\|\nabla J\|$. This normalization helps in the line search below.

4.5. Grassman-steepest descent svPCA algorithm

The Grassman svPCA (GsvPCA) algorithm is by:

i. Initialize: Set $F_0 = P$ where $Y/\sqrt{T} = QL^{1/2}P^T$.

ii. Grassman gradient : Compute ∇J_k using (3).

iii. Stop Condition : Terminate the algorithm if $\frac{\|\nabla J_{k+1}\|}{\|\nabla J_0\|} \leq \delta$.

iv. Line search : Compute the path length s^* that minimizes $J(F(s))$ where $F(s)$ is the Grassman geodesic parameterized by path length.

v. Update : Update the solution to $F_{k+1} = F(s^*)$. Go to step ii.

A little experimentation is needed in practice to select the tolerance parameter δ. An interesting idea is to let $\gamma \rightarrow 0$ in the gradient ∇J. This seems to work in practice, but proving convergence is difficult and is left for future work.

5. TUNING PARAMETER SELECTION

The svPCA cost function depends on two tuning parameters; the number of svPCs and the sparseness parameter h. A traditional way to select tuning parameters is cross-validation. However, for large data sets it is unfeasible due to high computation cost. We suggest the following Cost-Complexity (CC) criterion to select the tuning parameters:

$$\text{CC}_{h,r} = \frac{M}{2}\log(\hat{\sigma}^2) + \frac{1}{2T}d\log(T)$$

where $d = M_h r - r(r-1)/2$ is the number of free parameters to be fitted, M_h is the number of non-zero variables, $\hat{\sigma}^2 = \frac{1}{T}\sum_{t=1}^{T}\|y_t - \mu_t\|^2$ and $\mu_t = FF^T y_t$. The local minimums of the criterion are inspected and h and r are selected accordingly. Because our criterion is not based on a stochastic model we cannot apply traditional criteria such as BIC. However our adhoc criterion has the traditional form of residual sum of squares plus a BIC-like complexity penalty. It seems to behave well as long as T, M are not of similar size. Development of a more formally justified procedure is a non-trivial task and will be pursued elsewhere.

6. RESULTS

6.1. Simulation

We simulated a $T = 100$ times $M = 1024$ data matrix $Y = [y_t^T] = [y_{(v)}]$ where we think of the row vector y_t as a 32×32 image sampled at a time instance t. The data was constructed as follows:

$$\begin{aligned}
y_{t,v} &= \frac{50}{6\sqrt{50}}\cos(\frac{8\pi}{T}t) + \epsilon_{t,v}, \quad t = 0, ..., T-1, v \in A_1 \\
y_{t,v} &= \frac{25}{6\sqrt{50}}\cos(\frac{8\pi}{T}t) + \sqrt{0.6}\epsilon_{t,v}, \quad t = 0, ..., T-1, v \in A_2 \\
y_{t,v} &= \frac{40}{6\sqrt{50}}\sin(\frac{8\pi}{T}t) + \epsilon_{t,v}, \quad t = 0, ..., T-1, v \in A_3 \\
y_{t,v} &= \epsilon_{t,v}, \quad t = 0, ..., T-1, v \in A_4
\end{aligned}$$

where $\epsilon_{t,v} \sim N(0,1)$ and the regions A_1, A_2, A_3 and A_4 are defined in Fig. 1 Left (Region A_4 are the pixels outside the white rectangles). The time series in region A_1 and A_2 are highly correlated. The time series in region A_3 have very low correlation with the rest. Fig. 1. Right shows time instances y_{33} of the data showing that the signal to noise ratio is rather low.

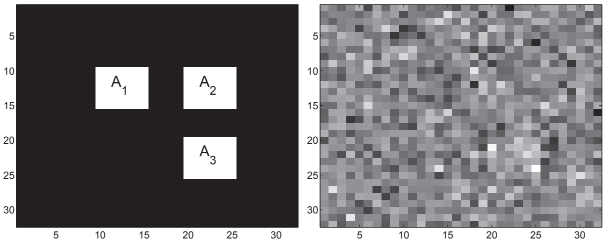

Fig. 1. Left: Regions A_1, A_2 and A_3 defined. Region A_4 is outside the white rectangles. Right: Noisy data samples y_{33}.

Fig. 2 shows the CC statistic. Based on the minimum of the CC statistics $h = 2$ and $r = 2$ are selected for the GsvPCA algorithm corresponding to $CC = 3543$. Fig. 3 show the columns of the GsvPC estimate F showing that the method detected signals correctly in all regions. We compare this result with the simple thresholding svPCA algorithm (JLsvPCA) in [5]:

JLsvPCA algorithm

i. Compute sample variances $\sigma_v^2 = \frac{1}{T}\sum_{t=1}^{T}(y_{t,v} - \bar{y}_v)^2, \quad v = 1, ..., M$ where $\bar{y}_v, v = 1, ..., M$ are the sample means.

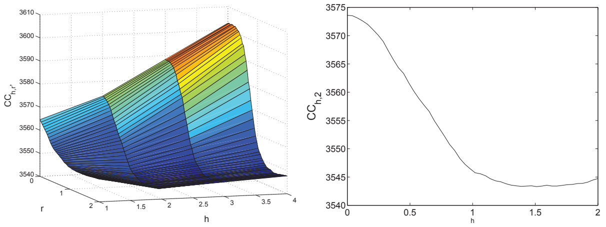

Fig. 2. CC statistics for the simulation. Left: 2 dimensional CC plot. Right: $r = 2$ CC profile.

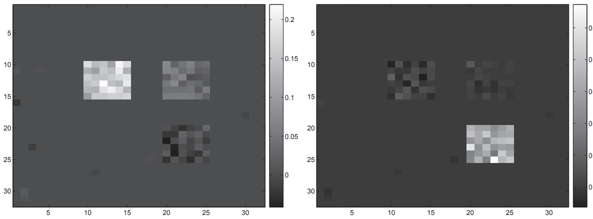

Fig. 3. Loadings from the GsvPCA algorithm. Left: First column of F. Right: Second column of F.

ii. Order the sample variances: $\sigma_{(1)}^2 > \sigma_{(2)}^2 > ... > \sigma_{(M)}^2$.

iii. Keep the variables corresponding to the M_h largest sample variances and zero out the rest.

iv. Compute traditional PCA (Section 2) on the reduced data set.

Actually, in [5] the JLsvPCA was performed in the wavelet domain. But since we are interested in sparsity in the spatial domain (not wavelet domain) we do not wavelet transform the data.

The CC statistic was used to select the tuning parameters for the tsvPCA algorithm which are the number of PCs r and the number of non-zero pixels M_h. The CC statistic took minimum at $(M_h, r) = (97, 2)$ with value $CC = 3545$ which is higher than for our svPCA algorithm. Fig. 4 displays the JLsvPC loadings. There are more noisy pixels than for GsvPCA and the algorithm missed region A_2 completely.

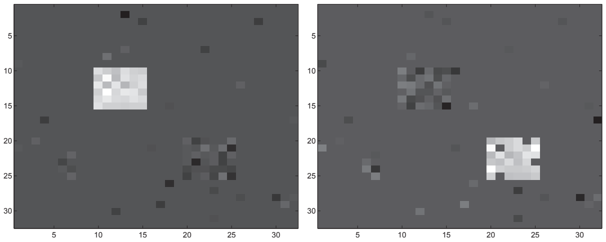

Fig. 4. Loadings from the JLsvPCA algorithm. Left: First column of F. Right: Second column of F.

6.2. GsvPCA vs svnPCA from [7]

Formally comparing GsvPCA is difficult since svnPCA is model based but GsvPCA is not. However, our experiments suggest that the results in practice are similar. But GsvPCA is faster since it is based on optimization on a Grassman manifold instead of the Stiefel optimization needed for svnPCA. In addition, svnPCA has an additional cyclic descent step which GsvPCA has not.

6.3. Real Data

In this section, we apply svPCA on real fMRI data [18] coming from a motor/visual experiment. We analyze a brain slice that includes the visual cortex. The data consists of $M = 1122$ volume elements (voxels) sampled over time with sampling period of 2 seconds. There are $T = 100$ time points in the data set. Fig. 5 shows brain image y_{50} (the mean has not been removed). The CC statistic is depicted on Fig. 6. After inspection of local minimums $r = 8$ and $h = 3.43$ was selected.

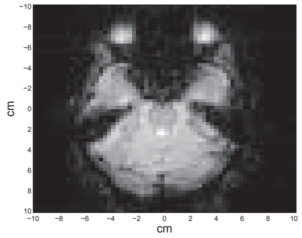

Fig. 5. Brain image y_{50}

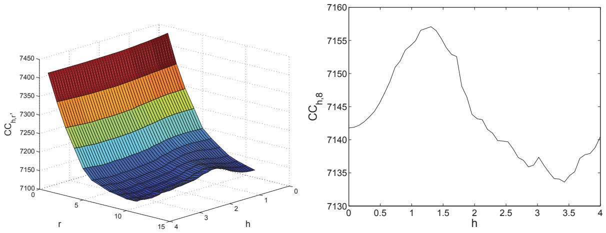

Fig. 6. CC statistics for the real fMRI data. Left: 2 dimensional CC plot. Right: $r = 8$ CC profile.

Fig. 7 shows spatial map number 2 (column 2 of F) and compares it to spatial map number 2 from traditional PCA ($h = 0$). We clearly see that the svPC map is much sparser.

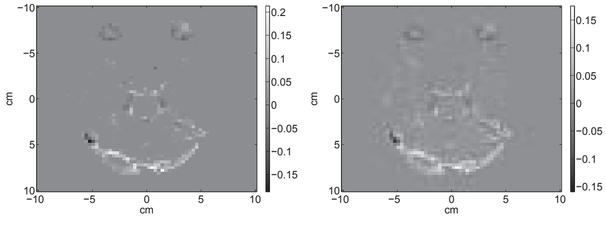

Fig. 7. Left: GsvPCA activation map 2. Right: PCA ($h = 0$) activation map 2.

7. CONCLUSIONS

In this paper we have developed a new sparse variable PCA algorithm. It is based on a geodesic steepest descent optimization of a vector l_1 penalized PCA criterion on a Grassman manifold. We also developed an ad-hoc model selection criterion for choosing jointly the number of principal components and the penalty parameter. The algorithm was demonstrated in a simulation to outperform the previous svPCA algorithm of [5]. In addition it was shown to perform well for real high dimensional fMRI data.

8. REFERENCES

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

[2] I. 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.

[3] A. D’Aspremont, L. E. Ghaoui, M. Jordan, and G. Lanckriet, “A direct formulation for sparse PCA using semidefinite programming,” to be published in SIAM review.

[4] J. H. H. Shen, “Sparse principal component analysis via regularized low rank matrix approximation,” *J. Multivariate Anal.*, 2007.

[5] I. Johnstone and A. Lu, “Sparse principal component analysis,” Statistics department, Stanford University, Tech. Rep., 2004.

[6] M. Ulfarsson and V. Solo, “Sparse variable principal component analysis with application to fMRI,” in *Proc. IEEE International Symposium on Biomedical Imaging (ISBI’07)*, Washington D.C., 2007.

[7] ——, “Sparse variable noisy PCA using geodesic descent,” *IEEE Trans. Signal Proc.*, accepted for publication.

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

[9] M. Tipping and C. Bishop, “Probabilistic principal component analysis,” *J. Royal Stat. Soc., Series B*, vol. 61, no. 3, pp. 611–622, 1999.

[10] C. Chennubhotla and A. Jepson, “Sparse PCA extracting multiscale structure from data,” in *Proc. International Conference on Computer Vision*, Vancouver, Canada, 2001, pp. 641–647.

[11] M. Tipping, “Sparse kernel principal component analysis,” in *Advances in Neural Information Processing Systems 13*, 2001.

[12] G. Seber, *Multivariate Observations*. New York, NY: Wiley, 1984.

[13] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *J.R. Statist. Soc. B.*, vol. 68, pp. 49–67, 2006.

[14] C. Vogel and M. Oman, “Iterative methods for total variation denoising,” *SIAM J. Sci. Comp.*, vol. 17, no. 1, pp. 227–238, 1996.

[15] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. New York, NY: Addison-Wesley, 1973.

[16] A. Edelman, T. Aries, and S. Smith, “The geometry of algorithms with orthogonality constraints,” *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 2, pp. 303–353, 1998.

[17] E. Stiefel, “Richtungsfelder und fernparallelismus in n-dimensionalem mannig faltigkeiten,” *Commentarii Math. Helvetici*, vol. 8, pp. 727–764, 1935.

[18] R. Cox, “AFNI,” http://afni.nimh.nih.gov/afni/, National Institute of Mental Health NIMH, Bethesda, Md.