AN EFFICIENT ALGORITHM FOR L1-NORM PRINCIPAL COMPONENT ANALYSIS

Linbin Yu, Miao Zhang, Chris Ding

Department of Computer Science & Engineering, University of Texas, Arlington, TX 76019

ABSTRACT

Principal component analysis (PCA) (also called Karhunen - Loève transform) has been widely used for dimensionality reduction, denoising, feature selection, subspace detection and other purposes. However, traditional PCA minimizes the sum of squared errors and suffers from both outliers and large feature noises. The L_1-norm based PCA (more precisely $L_{1,1}$ norm) is more robust. Yet, the optimization on L_1-PCA is much harder than standard PCA. In this paper, we propose a simple yet efficient algorithm to solve the L_1-PCA problem. We carry out extensive experiments to evaluate the proposed algorithm, and verify the robustness against image occlusions. Both numerical and visual results show that L_1-PCA is consistently better than standard PCA.

Index Terms— Principal component analysis, robustness, Lagrangian methods, Image processing

1. INTRODUCTION

Principal component analysis (PCA) (also called Karhunen - Loève transform) has been widely used for dimensionality reduction, denoising, feature selection, subspace detection and many other purposes and in many research areas. It is well-known that standard PCA is not robust: it minimizes the sum of squared errors; therefore large errors due to outliers and feature noises such as occlusion, after been squared, dominate the error function and force the low rank approximation to overwhelmingly concentrate on these few data points and features, while nearly ignoring most of other data points.

Over the years, there are a large number of research to solve this problem [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]. There are many different approaches. For example, early work focus on computing covariance matrix [4]. There are work on using $L_{2,1}$-norm to extract robust subspace [3], sparse coding [1]. More recent approach uses trace norm formulation [10]. However, the approach using pure L_1 norm[1] is used widely because it offers an simple and elegant formulation [5] [6] [7] [9]. A difficulty of pure L_1-PCA is that the optimization tends to be hard. Several computational methods have been proposed [5] [6] [7] [9]. These methods are either fairly complicated or difficult to scale to large problems.

In this paper, we propose a simple yet computational efficient algorithm to solve the L_1-PCA optimization. This method also provide some insights to the optimization problem such as the Lagrangian multiplier and KKT condition. We also carry out extensive experiments in face recognition, and verify the robustness of the proposed method to image occlusions. Both numerical and visual results validate the effectiveness of our proposed method.

[1] In this paper, the "L_1-norm" refers to the $L_{1,1}$-norm defined in Eq.(4). This definition of matrix L_1-norm is used in many earlier papers because it is similar to the L_1 norm of a vector. However, for a n-by-m matrix, there already exists an L_1-norm definition: $\|A\|_1 = \max_{v\in\Re^m} \|Av\|_1$, For this reason, we call the norm defined in Eq.(4) as $L_{1,1}$-norm.

2. FROM PCA TO L_1-PCA

In standard principal component analysis, the dimension reduction is on the rank of the matrix. This can be seen through as data compression or approximation. Given p-dimensional data X

$$X \simeq UV \tag{1}$$

where (U, V) are determined by

$$\min_{U,V} \quad \|X - UV\|_F^2,\ s.t.\quad U \in \Re^{p\times k},\ V \in \Re^{k\times n} \tag{2}$$

where for a $m \times n$ matrix A, $\|A\|_F = \sqrt{\sum_i^m \sum_j^n A_{ij}^2}$ is the Frobenius norm. It is well-known that the solution to the above optimization is given by the singular value decomposition (SVD). Let SVD of X be $X = F\Sigma G^T$, where r=rank(X), $F = (f_1 \cdots, f_r)$ are left singular vectors, $G = (g_1 \cdots, g_r)$ are right singular vectors, and $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_r)$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. Then the solution of Eq.(2) is $U = F_k = (f_1 \cdots f_k)$ and $V = \Sigma_k G_k^T$, where $G_k = (g_1 \cdots g_k)$ and $\Sigma_k = \mathrm{diag}(\sigma_1, \cdots, \sigma_k)$.

The orthogonal matrix U is regarded as the projection matrix. The $k \times n$ matrix $V = (v_1 \cdots v_n)$ is the projection of X i.e., representation of X in the new basis U. Because generally $k \ll p$, V has much smaller dimension than original data X, and is generally used for further computation. Thus UV is the low-rank approximation of X.

We may view this approximation as denoising model:

$$X_{ij} = (UV)_{ij} + E_{ij} \tag{3}$$

where the input X is consist of the true signal UV and noise E. Here UV can be viewed as model parameter: $\theta_{ij} =$

$(UV)_{ij}$. In standard PCA, we regard the noise as white-noise described by Gaussian distribution, $p(x) \sim \exp(-(x-\theta)^2/2\sigma^2)$.

Thus the error function is the usual least square of Eq.(2). Although the least square error function is not robust for large errors, this is not a problem for Gaussian white noise because large noises bigger than 3σ are very rare (with probability $p < 0.003$).

For the large noises, the distribution model is the Laplacian distribution, $p(x) \sim \exp(-|x-\theta|/\sigma)$. The error function is the minus log-likelihood

$$-\log[\Pi_{i=1}^{n} p(x_i)] \propto \sum_{i=1}^{n} |x_i - \theta|.$$

Thus the more robust version of the PCA is formulated using matrix $L_{1,1}$ norm:

$$\begin{aligned} &\min_{U,V} \ \|X - UV\|_{1,1} = \sum_{j=1}^{n}\sum_{i=1}^{p} |(X-UV)_{ij}|, \\ &s.t. \quad U \in \Re^{p\times k},\ V \in \Re^{k\times n} \end{aligned} \tag{4}$$

Below we introduce an efficient and stable algorithm to solve Eq.(4). This is the main contribution of this paper.

3. ALGORITHM FOR L_1-PCA

Eq.(4) can be rewritten equivalently as

$$\begin{aligned} &\min_{E,U,V} \ \|E\|_{1,1} \\ &s.t. \ \ E = X - UV, \ \ U \in \Re^{p\times k},\ V \in \Re^{k\times n} \end{aligned} \tag{5}$$

The Augmented Lagrange Multiplier (ALM) method is employed [11] to solve this problem. ALM solves a sequence of sub-problems

$$\min_{E,U,V} \ \|E\|_{1,1} + \langle A, X - UV - E\rangle + \frac{\mu}{2}||X - UV - E||_F^2 \tag{6}$$

where matrix A is the Lagrange multipliers, scalar μ is the penalty parameter, $\langle P, Q\rangle$ is defined as $\sum_{ij} P_{ij}Q_{ij} = \mathrm{Tr}P^TQ$.

The ALM is an iteratively updating algorithm. There are two major parts, solving the sub-problem and updating parameters, which will be presented in the following sections.

3.1. Solving the Sub Optimization Problem

The key step of the algorithm is solving the two sub-program of Eq.(6) for each set of parameter values of A, μ. Fortunately, this can be solved in closed form solutions for E and pair of (U, V).

Solve for E. First, we solve E while fixing U and V. From Eq.(6), it becomes

$$\min_{E} \ \|E\|_{1,1} + \frac{\mu}{2}||E - (X - UV + \frac{A}{\mu})||_F^2 \tag{7}$$

This problem has closed form solution:

$$E_{ij}^* = sign(P_{ij})(|P_{ij}| - 1/\mu)_+, \ P = X - UV + \frac{A}{\mu}. \tag{8}$$

Solve for U, V. Next we solve U and V together while fixing E. From Eq.(6), it becomes

$$\min_{U,V} \ \langle A, X - UV - E\rangle + \frac{\mu}{2}||X - UV - E||_F^2. \tag{9}$$

Which is is equivalent to

$$\min_{U,V} \ \frac{\mu}{2}||Q - UV||_F^2, \ \ Q = X - E + \frac{A}{\mu}; \tag{10}$$

The solution is given by standard PCA. Denote the singular value decomposition (SVD) of Q as

$$Q = F\Sigma G^T \tag{11}$$

Only first k largest singular values and associated singular vectors are needed. Then the solution of U, V are given by

$$\begin{aligned} U &= F_k \\ V &= \Sigma_k G_k^T \end{aligned} \tag{12}$$

3.2. Updating ALM Parameters

In each iteration of ALM, after obtaining consistent E and (U, V), the parameters A and μ are updated as following

$$A \ \Leftarrow \ A + \mu(X - UV - E) \tag{13}$$

$$\mu \ \Leftarrow \ \mu\rho \tag{14}$$

where $\rho > 1$ is a constant.

The complete algorithm is described in Algorithm 1.

Input: X, k
Output: U, V
Initialize $\mu = 1/||X||_F, \rho = 1.2, E = 0, A = 0$
repeat
  Compute U, V using Eq.(12)
  Compute E using Eq.(8)
  $A = A + \mu(X - UV - E)$
  $\mu = \min(\mu\rho, 10^{10})$
until *Converge*

Algorithm 1: L1-PCA Algorithm

4. ANALYSIS

In ALM, the constraint $E = X - UV$ in Eq.(5) is enforced in two ways: (1) using Lagrangian multiplier A by adding $\langle A, X - UV - E\rangle$ in the objective of Eq.(6). (2) additional penalty terms $\frac{\mu}{2}||X - UV - E||_F^2$.

This "double-enforcing" of the constraint $E = X - UV$ will succeed if (1) either Lagrangian multiplier A approach the correct values, or (2) penally parameter μ becomes very large to enforce the constraints directly. In ALM method, these two channels are coherently combined together to achieve the final solution.

We demonstrate this fact below. The Lagrangian function is

$$L = \|E\|_{1,1} + TrA^T(X - UV - E) \tag{15}$$

where A is the Lagrangian multiplier to enforce the constraint $E = X - UV$. The Karush-Kuhn-Tucker (KKT) condition is

$$\frac{\partial L}{\partial E_{ij}} = \frac{\partial \|E\|_{1,1}}{\partial E_{ij}} - A_{ij} = 0 \tag{16}$$

This gives

$$A_{ij} = \begin{cases} sign(E_{ij}) & \text{if } E_{ij} \neq 0 \\ \partial |E_{ij}| & \text{if } E_{ij} = 0 \end{cases} \tag{17}$$

where $\partial|E_{ij}| \in [-1, 1]$ is the subgradient of function $f(x) = |x|$.

A convex function $f(x)$ always satisfies

$$f(x) \geq f(x_0) + f'(x_0)(x - x_0),\ \forall\, x$$

where $f'(x_0)$ is the gradient of $f(x)$ at $x = x_0$. When the gradient $f'(x_0)$ at $x = x_0$ does not exist, such as the gradient of $f(x) = |x|$ at $x = 0$, we define the **subgradient** of $f(x)$ at $x = x_0$ (denoted by $\partial f|_{x_0}$) as any real number that satisfies

$$f(x) \geq f(x_0) + \partial f|_{x_0}(x - x_0),\ \forall\, x.$$

In general, subgradient at $x = x_0$ is not unique. The set of subgradients at $x = x_0$ form a closed interval $[a, b]$, where a, b are directional derivatives, which are calculated using one-side limits (for $f(x) = |x|$ at point $x_0 = 0$)

$$a = \lim_{x \to x_0^-} \frac{f(x) - f(x_0)}{x - x_0} = -1,$$

$$b = \lim_{x \to x_0^+} \frac{f(x) - f(x_0)}{x - x_0} = 1,$$

Thus subgradients of $f(x) = |x|$ at point $x_0 = 0$ is any value in the closed interval $[-1, 1]$. With this introduction of subgradient, we can **verify** if the A obtained in ALM is indeed the Lagrangian multiplier through the KKT condition of Eq.(17). Once the ALM is converged, A should satisfy Eq.(17).

We now demonstrate the correctness of our algorithm. we randomly generate 5×6 matrix X, and set $k = 3$

$$X = \begin{pmatrix} 0.46 & 0.87 & 0.79 & 0.51 & 0.37 & 0.54 \\ 0.45 & 0.05 & 0.45 & 0.20 & 0.94 & 0.65 \\ 0.55 & 0.22 & 0.33 & 0.43 & 0.02 & 0.73 \\ 0.81 & 0.46 & 0.06 & 0.17 & 0.83 & 0.09 \\ 0.70 & 0.96 & 0.74 & 0.75 & 0.63 & 0.88 \end{pmatrix}$$

Using our algorithm, the converged parameter A is

$$A = \begin{pmatrix} -1.00 & 0.82 & 1.00 & -1.00 & 0.44 & -0.39 \\ -0.70 & 0.56 & 0.70 & -0.68 & 0.31 & -0.29 \\ 1.00 & -0.76 & -1.00 & 0.89 & -0.43 & 0.41 \\ 1.00 & -0.83 & -1.00 & 1.00 & -0.47 & 0.44 \\ 0.22 & -0.20 & -0.21 & 0.27 & -0.09 & 0.06 \end{pmatrix}$$

and residue values are

$$E = X - UV = \begin{pmatrix} -0.11 & 0 & 0.29 & -0.08 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.56 & 0 & -0.19 & 0 & 0 & 0 \\ 0.10 & 0 & -0.08 & 0.02 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

One can clearly see that for non-zero elements E_{ij}, the corresponding elements in A equal to $sign(E_{ij})$. For zero elements E_{ij}, the corresponding elements A_{ij} are subgradients satisfying $A_{ij} \in [-1, 1]$. In other words, A obtained my our algorithm is indeed the Lagrangian multiplier, and the solution satisfies the KKT condition.

5. EXPERIMENTS

We perform experiments on AT&T and Extended YaleB face data sets. The AT&T face data contains 400 upright face images of 40 individuals, collected by AT&T Laboratories Cambridge. Each image is resized to 28x23 pixels in this experiment. The extended Yale Face Database B contains 2414 cropped frontal face image of 38 individuals. We select 10 images with different brightness per individual and resize them to 32x28 pixels.

To check the convergence of the proposed ALM algorithm, in Figure 1 we plot objective function of each iteration on two datasets. Clearly, the algorithm takes about 60 iterations to converge.

According to Algorithm 1, one can find that in the first iteration $A = 0, E = 0, Q = X$, so the first U, V is actually the solution of standard SVD of X. Once started, the algorithm wonders around for about 30 iterations for the parameters of A to settle down to the proper values, i.e., becoming close to the true Lagrangian multiplier values. From there, the overall algorithm converges very quickly, in essentially 10 iterations and then reach the asymptotic values of the objective function.

We generate sizes 1×1, 2×2, 3×3 occlusions in each image. Denote X as the original images, O as the noise due to occlusion, then $X - O$ is the occluded images, which are the input of L_1-PCA and PCA.

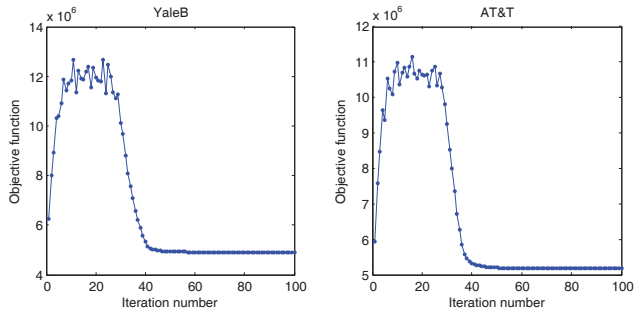

Fig. 1. Convergence of the proposed algorithm on YaleB and AT&T datasets. Shown are values of $\|X - UV\|_1$. The first point is $\|X - U_0V_0\|_1$ where U_0, V_0 are the SVD of X, according to Algorithm 1.

Data	d	m	$\|O\|_F$	R_{PCA}	R_{L1PCA}
YaleB	1	100	19495	13240	8104
YaleB	2	25	18674	13997	8120
YaleB	3	10	17173	14230	8153
AT&T	1	100	24452	16826	7924
AT&T	2	25	23623	17987	9050
AT&T	3	10	22257	18856	14613

Table 1. Comparison of denoising reconstruction error. m is the number of size $d \times d$ occlusions in one image. R_{PCA} and R_{L1PCA} are the noise-free reconstruction errors.

To see the noise reduction effects, we compute the **noise-free** reconstruction error is defined as

$$\begin{aligned} R_{PCA} &= \|Y_{PCA} - X\|_F \\ R_{L1PCA} &= \|Y_{L1PCA} - X\|_F \end{aligned} \tag{18}$$

where Y_{PCA} is the reconstructed images from PCA and Y_{L1PCA} is the reconstructed images from L_1-PCA

Note that these are not the usual reconstruction errors which are defined as $\|Y_{PCA} - (X - O)\|_F$ and $\|Y_{L1PCA} - (X - O)\|_F$, because $(X - O)$ is the input to the algorithms. These two quantities do not measure the denoising effects.

The results are listed in Table 1. From Table 1, R_{L1PCA} is 22% to 50% less than R_{PCA} on the same occluded image data. Figure 2 shows some example of denoised images.

Conclusions. In this paper, we propose a computational efficient algorithm to solve L_1-norm based PCA. Extensive experiments are carried out to evaluate the proposed algorithm. Both numerical and visual results are consistently better than standard PCA, which validate the effectiveness of the proposed L_1-PCA method.

Acknowledgements. This work is partially supported by NSF-CCF-0939187, NSF-CCF-0917274, NSF-DMS-15228.

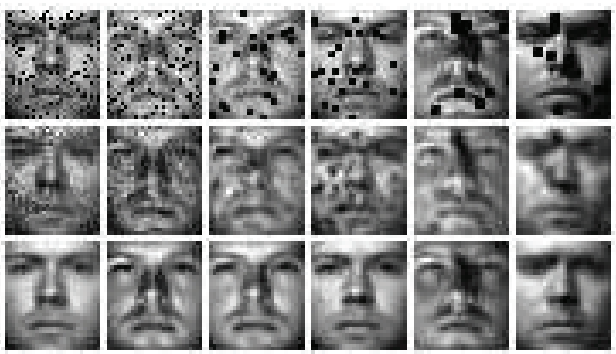

Fig. 2. Examples of denoising. The top row lists the corrupted images. The middle row lists reconstructed images by standard PCA. The bottom row lists reconstructed images by the proposed L1-PCA.

6. REFERENCES

[1] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Computational and Graphical Statistics*, vol. 15, pp. 265–286, 2006.

[2] F. D.A. Torre and M. J. Black, "A framework for robust subspace learning," *Int'l J. Computer Vision*, pp. 117–142, 2003.

[3] C. Ding, D. Zhou, X. He, and H. Zha, "R1-pca: Rotational invariant l1-norm principal component analysis for robust subspace factorization," *Proc. Int'l Conf. Machine Learning (ICML)*, June 2006.

[4] J.S. Galpin and D.M. Hawkins., "Methods of l1 estimation of a covariance matrix," *Computational Statistics and Data Analysis*, vol. 5, pp. 305–319, 1987.

[5] A. Baccini, Ph. Besse, and A. De Falguerolles, "An L1-norm PCA and a heuristic approach," *Ordinal and Symbolic Data Analysis, edited by E. Diday, Y. Lechevalier and O. Opitz, Springer*, 1996.

[6] J. Bolton and Wojtek J. Krzanowski, "A characterization of principal components for projection pursuit," Mar. 26 2001.

[7] Q. Ke and T. Kanade, "Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2004, pp. 592–599.

[8] N. Kwak, "Principal component analysis based on L1-norm maximization," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 30, no. 9, pp. 1672–1680, 2008.

[9] J. Gao, "Robust L1 principal component analysis and its bayesian variational inference," *Neural Computation*, vol. 20, no. 2, 2008.

[10] E. J. Candes, Xiaodong Li, Yi Ma, and John Wright, "Robust principal component analysis?," Dec. 18 2009.

[11] D. P. Bertsekas, *Nonlinear Programming, 2nd Ed.*, MIT Press, 1998.