Proceedings of 2009 APSIPA Annual Summit and Conference, Sapporo, Japan, October 4-7, 2009

Fast Method of Principal Component Analysis Based on L1-Norm Maximization Algorithm

Nobuhiro Funatsu* and Yoshimitsu Kuroki*
* Kurume National College of Technology
1-1-1, Komorino, Kurume-shi, Fukuoka 838-8555 Japan
Tel: +81-942-35-9392 Fax: +81-942-35-9398
E-mail: kuroki@kurume-nct.ac.jp

***Abstract*—In data-analysis problems with a large number of dimension, principal component analysis based on L2-norm (L2-PCA) is one of the most popular methods, but L2-PCA is sensitive to outliers. Unlike L2-PCA, PCA-L1 is robust to outliers because it utilizes the L1-norm, which is less sensitive to outliers. Furthermore, the bases obtained by PCA-L1 is invariant to rotations. However, PCA-L1 needs long time to calculate bases, because PCA-L1 employs an iterative algorithm to obtain each basis, and requires to calculate an eigenvector of autocorrelation matrix as an initial vector. The autocorrelation matrix needs to be recalculated for each basis. In this paper, we propose a fast method to compute the autocorrelation matrices. In order to verify the proposed method, we apply L2-PCA, PCA-L1, and the proposed method to face recognition. Simulation results show that the proposed method provides same recognition performance as PCA-L1, and is superior to L2-PCA, while the execution time is less than PCA-L1.**

I. Introduction

In data-analysis problems with a large number of dimension, principal component analysis (PCA) is one of the most popular methods. PCA is an operation that finds orthonormal bases to project in a subspace among multivariable data. Various methods in order to obtain the bases are proposed, and the most popular method is a PCA based on L2-norm (L2-PCA). Projection values of data onto the bases derived from L2-PCA have the greatest number of variance. Although L2-PCA has been successful for many problems, the influence of outliers on the principal bases are significant due to the L2-norm criterion. The influence seems to be reduced by the PCA based on L1-norm (L1-PCA). Unlike L2-PCA, L1-PCA is robust to outliers because it utilizes the L1-norm, which is less sensitive to outliers. However, it is difficult to calculate exact solutions of L1-PCA. To solve this problem, Kwak proposes a scheme employing a substitute formula based on L1-norm, designated as PCA-L1 [1], to obtain principal bases easily. Furthermore, the bases obtained by PCA-L1 is invariant to rotations. The detail of PCA-L1 is described in Section Ⅱ; PCA-L1 employs an iterative algorithm to compute each basis, and requires to calculate an eigenvector of autocorrelation matrix as an initial vector. The autocorrelation matrix is computed by the projected data onto the orthogonal complement of already calculated eigenvectors. Thus, autocorrelation matrix needs to be recalculated for each basis. This paper proposes a fast method to compute the autocorrelation matrices. In order to verify the proposed method, we apply L2-PCA, PCA-L1, and the proposed method to face recognition. The rest of this paper is organized as follows: In Section Ⅱ, PCA-L1 algorithm is formulated. The proposed method is explained in Section Ⅱ I. In Section Ⅳ, we mention face recognition technique. The performance of the proposed method is compared with the conventional methods in Section V and the conclusion in Section Ⅵ.

II. PCA-L1 algorithm

Let $X = [x_1, \ldots, x_n] \in R^{d\times n}$ be the given data, where n and d denote the number of samples and the dimension of the original input space, respectively. Without loss of generality, ${x_i}_{i=1}^{n}$ is assumed to have zero mean.

In L2-PCA, one tries to find an $\boldsymbol{W}^*$ which is $m(< d)$ dimensional linear subspace. The $\boldsymbol{W}^*$ is the solution of the following dual problem:

$$\boldsymbol{W}^* = \underset{\boldsymbol{W}}{\operatorname{argmax}} \parallel \boldsymbol{W}^T\boldsymbol{S}\boldsymbol{W} \parallel_2 = \underset{\boldsymbol{W}}{\operatorname{argmax}} \parallel \boldsymbol{W}^T\boldsymbol{X} \parallel_2, \quad \text{Subject to} \quad \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_m, \tag{1}$$

where $\boldsymbol{W} \in R^{d\times m}$ is the projection matrix whose columns ${\boldsymbol{w}_k}_{k=1}^{m}$ constitute the bases of the m-dimensional linear subspace (feature space), $\boldsymbol{S} = \boldsymbol{X}^T\boldsymbol{X}$ is the autocorrelation matrix of $\boldsymbol{X}$, $\boldsymbol{I}_m$ is the $m \times m$ identity matrix, and $\parallel \cdot \parallel_2$ denotes the L2-norm of a matrix or a vector. The methods based on the L2-norm are sensitive to outliers, so we use the methods based on the L1-norm which is robust to outliers than the L2-norm.

In PCA-L1, one tries to find an $\boldsymbol{W}^*$ which is $m(< d)$ dimensional linear subspace. The $\boldsymbol{W}^*$ is the solution of the following dual problem:

$$\boldsymbol{W}^* = \underset{\boldsymbol{W}}{\operatorname{argmax}} \parallel \boldsymbol{W}^T\boldsymbol{X} \parallel_1, \quad \text{Subject to} \quad \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_m. \tag{2}$$

Here, the constraint $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_m$ ensures the orthonormality of the projection matrix. The solution of (2) is invariant to rotations because the maximization is done on the subspace and it is expected to be more robust to outliers than the L2 solution.

As a downside, finding a global solution of (2) for $m > 1$ is very difficult. To ameliorate this problem, Kwak simplify (2) into series of $m = 1$ problems using a greedy search method.

If we set $m = 1$, (2) becomes the following optimization problem:

$$\boldsymbol{w}^* = \underset{\boldsymbol{W}}{\operatorname{argmax}} \parallel \boldsymbol{w}^T \boldsymbol{X} \parallel_1 = \sum_{i=1}^{n} \left| \boldsymbol{w}^T \boldsymbol{x}_i \right|, \quad \text{Subject to} \quad \parallel \boldsymbol{W} \parallel_2 = 1. \tag{3}$$

In the following, an algorithm to solve (3) is presented.
Algorithm: PCA-L1 ($m = 1$)

1) Initialization: Pick any $\boldsymbol{w}(t = 0)$. Set $\boldsymbol{w}(0) \leftarrow \boldsymbol{w}(0)/\parallel \boldsymbol{w}(0) \parallel_2$.
2) Polarity check: For all $i \in \{1, \ldots, n\}$, if $\boldsymbol{w}^T \boldsymbol{x}_i < 0, p_i(t) = -1$, otherwise $p_i(t) = 1$.
3) Flipping and maximization: Set $t \leftarrow t+1$ and $\boldsymbol{w}(t) = \sum_{i=1}^{n} p_i(t-1)\boldsymbol{x}_i$. Set $\boldsymbol{w}(t) \leftarrow \boldsymbol{w}(t)/\parallel \boldsymbol{w}(t) \parallel_2$.
4) Convergence check:
   a) if $\boldsymbol{w}(t) \neq \boldsymbol{w}(t-1)$, go to Step 2.
   b) Else if there exists i such that $\boldsymbol{w}^T \boldsymbol{x}_i = 0$, set $\boldsymbol{w}(t) \leftarrow (\boldsymbol{w}(t) + \Delta\boldsymbol{w}) / \parallel \boldsymbol{w}(t) + \Delta\boldsymbol{w} \parallel_2$ and go to Step 2. Here $\Delta\boldsymbol{w}$ is a small nonzero random vector.
   c) Otherwise, set $\boldsymbol{w}^* = \boldsymbol{w}(t)$ and stop.

The projection vector $\boldsymbol{w}$ converges to $\boldsymbol{w}^*$, which is a local maximum point of $\sum_{i=1}^{n} \left| \boldsymbol{w}^T \boldsymbol{x}_i \right|$, and there is a possibility that it may not be global solution. The first principal base of L2-PCA is used as an initial vector $\boldsymbol{w}(0)$ and presents a greedy search algorithm for $m > 1$. The greedy search algorithm is as follows:

Greedy search algorithm

1) For all $i \in \{1, \ldots, n\}$, $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \bar{\boldsymbol{x}}$, where $\boldsymbol{x}_i$ is the image for study, and $\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$
2) Set a data-set $\boldsymbol{X} = (\boldsymbol{x}_1 \ldots \boldsymbol{x}_n)$.
3) Compute a autocorrelation matrix $\boldsymbol{S} = \boldsymbol{X}^T \boldsymbol{X}$.
4) Processing from 5) to 7) is repeated m times.
5) The first principal base of L2-PCA is assumed to be the initial value and a principal base $\boldsymbol{w}$ is calculated by the PCA-L1 algorithm.
6) Set $\boldsymbol{X} \leftarrow \boldsymbol{X} - \boldsymbol{w}(\boldsymbol{w}^T \boldsymbol{X})$. By the processing, $\boldsymbol{X}$ is projected into orthogonal complementary space of $\boldsymbol{w}$.
7) Compute $\boldsymbol{S} = \boldsymbol{X}^T \boldsymbol{X}$.

We should notice that the solution of the greedy algorithm is not necessarily the optimal solution of (2). However, it is expected to provide a set of good projections that maximizes L1 dispersion.

III. Proposed Method

The algorithms shown in the previous section requires much execution time to calculate the new data set and the autocorrelation matrix, the procedure 6) and 7) in the greedy search algorithm. Then, this section describes a fast method of the processing. Now, we would like to focus explanation for the computation of the mth autocorrelation matrix. At this time, it is assumed that the $m-1$th data-set:

$$\begin{aligned} \boldsymbol{X} &= \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_n \end{bmatrix} \\ &= \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & \cdots & x_{dn} \end{bmatrix}, \end{aligned} \tag{4}$$

autocorrelation matrix:

$$\begin{aligned} \boldsymbol{S} &= \boldsymbol{X}^T \boldsymbol{X} \\ &= \begin{bmatrix} \sum_{k=1}^{d} x_{k1}^2 & \sum_{k=1}^{d} x_{k1}x_{k2} & \cdots & \sum_{k=1}^{d} x_{k1}x_{kn} \\ \sum_{k=1}^{d} x_{k2}x_{k1} & \sum_{k=1}^{d} x_{k2}^2 & \cdots & \sum_{k=1}^{d} x_{k2}x_{kn} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=1}^{d} x_{kn}x_{k1} & \sum_{k=1}^{d} x_{kn}x_{k2} & \cdots & \sum_{k=1}^{d} x_{kn}^2 \end{bmatrix}, \end{aligned} \tag{5}$$

and the principal base $\boldsymbol{w}$ are given. To calculate the mth principal base, first, the data-set $\boldsymbol{X}$ is projected into orthogonal complementary space of the vector $\boldsymbol{w}$. The new data-set $\boldsymbol{X}'$ is given by

$$\begin{aligned} \boldsymbol{X}' &= \begin{bmatrix} \boldsymbol{x}'_1 & \ldots & \boldsymbol{x}'_n \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{x}_1 - (\boldsymbol{w}^T\boldsymbol{x}_1)\boldsymbol{w} & \ldots & \boldsymbol{x}_n - (\boldsymbol{w}^T\boldsymbol{x}_n)\boldsymbol{w} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{x}_1 - \alpha_1\boldsymbol{w} & \ldots & \boldsymbol{x}_n - \alpha_n\boldsymbol{w} \end{bmatrix}, \end{aligned} \tag{6}$$

where α_i is a inner product between $\boldsymbol{w}$ and $\boldsymbol{x}_i$.
Second, a new autocorrelation matrix $\boldsymbol{S}'$ is calculated form $\boldsymbol{X}'$. $\boldsymbol{S}'$ is given by

$$\begin{aligned} \boldsymbol{S}' &= \boldsymbol{X}'^T \boldsymbol{X}' \\ &= \begin{bmatrix} \sum_{k=1}^{d} {x'}_{k1}^2 & \sum_{k=1}^{d} x'_{k1}x'_{k2} & \cdots & \sum_{k=1}^{d} x'_{k1}x'_{kn} \\ \sum_{k=1}^{d} x'_{k2}x'_{k1} & \sum_{k=1}^{d} {x'}_{k2}^2 & \cdots & \sum_{k=1}^{d} x'_{k2}x'_{kn} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=1}^{d} x'_{kn}x'_{k1} & \sum_{k=1}^{d} x'_{kn}x'_{k2} & \cdots & \sum_{k=1}^{d} {x'}_{kn}^2 \end{bmatrix}. \end{aligned} \tag{7}$$

Based on (7), we can derive

$$
\begin{aligned}
s'_{ij} &= \sum_{k=1}^{d} x'_{ki} x'_{kj} && (8)\\
&= \sum_{k=1}^{d} (x_{ki} - \alpha_i w_k)(x_{kj} - \alpha_j w_k) && (9)\\
&= \sum_{k=1}^{d} (x_{ki}x_{kj} - \alpha_i w_k x_{kj} - \alpha_j w_k x_{ki} + \alpha_i \alpha_j w_k^2) && (10)\\
&= \sum_{k=1}^{d} (x_{ki}x_{kj} - \alpha_i \alpha_j - \alpha_j \alpha_i + \alpha_i \alpha_j w_k^2) && (11)\\
&= \sum_{k=1}^{d} (x_{ki}x_{kj}) - 2\alpha_i \alpha_j + \sum_{k=1}^{d} \alpha_i \alpha_j w_k^2 && (12)\\
&= s_{ij} + \alpha_i \alpha_j (\sum_{k=1}^{d} w_k^2 - 2), && (13)
\end{aligned}
$$

where s'_{ij} is the (i, j) element in $\boldsymbol{S}'$. Therefore, to calculate $\boldsymbol{S}'$, we only have to add $\alpha_i \alpha_j (\sum_{k=1}^{d} w_k^2 - 2)$ to s_{ij}. By the proposed method, we expect that the same result as the conventional method can be obtained in a short time.

IV. Face recognition

In the face recognition, face images are divided into two groups; for study and for test. The images are considered to be column vectors by raster scan order, and study vectors are embedded in a data-set matrix $\boldsymbol{X}$. A low-dimensional linear subspace of $\boldsymbol{X}$ is calculated. As for how to calculate the projected bases, various techniques are proposed. Eigenface is one of the face recognition techniques, and L2-PCA is used to reduce dimension [2]. In this study, PCA-L1 is used instead of L2-PCA. To identify a person, the all study images are projected into the subspace in advance; then, a test image is projected into the subspace. The study image having the smallest distance from the test image in the subspace is considered to be the most suitable person.

V. Experimental results

A Yale Face Database [3] is used for the experiment. The database consists of 65 gray-scale images of 10 individuals.

Fig. 1. A part of images used to experiment

And the size of images is 170 × 120 pixels. Fig. 1 shows a part of the images.

Specifications of PC we used is shown as follows:OS:Linux-2.6.18, CPU:Intel Xeon(R) 2.33GHz[Dual CPU], Memory:2GB. To calculate eigenvectors and eigenvalues, we employed the GNU Scientific Library (GSL) [4].

Here, n images for study are selected at random, and the rest is assumed to be images for the test. The images for study projected into subspace are $\boldsymbol{v_i} = (v_{1i}\ v_{2i}\ ...\ v_{mi})^T \quad (i = 1, 2, ..., n)$ and the image for test is $\boldsymbol{x} = (x_1\ x_2\ ...\ x_m)^T$. The L1-norm distance function is assumed to be an evaluation function. The evaluation for $\boldsymbol{v}_i$, denoted by g_i is shown as:

$$g_i = \sum_{k=1}^{m} \mid x_k - v_{ki} \mid . \tag{14}$$

In the experiment, the bases are calculated by the conventional PCA-L1 and the proposed method. We measured a computing time and a recognition rate for each method. As a comparison, we also measure the recognition rate using the bases derived by L2-PCA. If $\boldsymbol{x}$ and $\boldsymbol{v}_i$ that minimizes (14) are the same persons, recognition is assumed to be a success. The recognition rate a is defined by

$$a = \frac{b}{c} \times 100, \tag{15}$$

where b and c mean a number of success recognition and a number of test images, respectively.

In the experiment, parameters are changed as follows.
Experiment 1 : The number of extracted features is fixed to 20, and the number of the images for study is varied from 30 to 100.
Experiment 2 : The number of the images for study is fixed to 60 pieces, and the number of extracted features is varied form 10 to 50.

Each experiment is exected 10 times, and we compare the averages. The results of computing time are shown in Table I and Table II. From Table I and Table II, the computing time of the proposed method is faster than the conventional method. It seems that the proposed method is more effective as both the number of study images and the number of features increase. Table III and Table IV signify the results of recognition rate. Because the recognition rate of PCA-L1 tends to rise more than PCA-L2, it seems that PCA-L1 is useful as the base reduction method of Eigenface.

VI. Conclusion

In this paper, we proposed the fast method of the PCA-L1 algorithm. The effect of the proposed method was verified by applying PCA-L1 to the dimension reduction of Eigenface in the facial recognition. In the PCA-L1 algorithm, it is necessary to solve the eigenvalue problems of the number of extracted bases, and it takes many time to calculate the autocorrelation matrix. In the proposed method, the procedure of the autocorrelation matrix calculation and the data-set generation is shorten. The proposed method keeps recognition rate, in

TABLE I
COMPUTING TIME OF THE EXPERIMENT 1

The number of the images for study	Average of computing time [s]		rate [%] [a]
	proposed method	PCA-L1	
30	4.44	11.67	▽ 61.91
40	5.62	18.88	▽ 70.23
50	7.40	29.09	▽ 74.57
60	9.10	40.16	▽ 77.34
70	11.60	55.26	▽ 79.00
80	13.39	66.96	▽ 80.01
90	16.16	91.99	▽ 82.43
100	21.06	101.23	▽ 79.20

[a] △ is increase, and ▽ is decrease.

TABLE II
COMPUTING TIME OF THE EXPERIMENT 2

The number of the extracted features	Average of computing time [s]		rate [%] [a]
	proposed method	PCA-L1	
10	5.78	20.04	▽ 71.18
20	9.25	40.69	▽ 77.28
30	12.68	58.99	▽ 78.51
40	16.67	77.34	▽ 78.45
50	21.09	103.21	▽ 79.56

[a] △ is increase, and ▽ is decrease.

addition, PCA-L1 achieves higher recognition performance than L2-PCA.

In future works, we would like to apply the proposed method to color image procession.

TABLE III
RECOGNITION RATE OF THE EXPERIMENT 1

The number of the images for study	Average of recognition rate [%]		
	proposed method	PCA-L1	L2-PCA
30	64.44	64.44	63.61
40	73.34	73.34	71.69
50	76.48	76.48	75.90
60	75.51	75.51	75.42
70	79.52	79.52	78.66
80	79.91	79.91	79.21
90	81.45	81.45	80.02
100	84.11	84.11	83.29

TABLE IV
RECOGNITION RATE OF THE EXPERIMENT 2

The number of the extracted features	Average of recognition rate [%]		
	proposed method	PCA-L1	L2-PCA
10	64.34	64.34	62.90
20	75.69	75.69	74.20
30	84.00	84.00	84.20
40	82.32	82.32	82.34
50	86.83	86.83	86.88

REFERENCES

[1] P.N.Belhumeur, J.P.Hespanha, and D.J.Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific Linear projection", IEEE Transactions on pattern analysis and machine intelligence, vol.19, no.7, pp.711-720, 1997.

[2] Nojun Kwak, "Principal component analysis baced on L1-norm maximization", IEEE Transactions on pattern analysis and machine intelligence, vol.30, no,9, Sept. 2008.

[3] Yale Face Database
http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html

[4] GNU Scientific Library
http://www.gnu.org/software/gsl/