Principal Component Analysis based on L1-norm Maximization

Nojun Kwak, *Member, IEEE,*

N. Kwak is an assistant professor at the Division of Electrical & Computer Engineering, Ajou University, Suwon 443-749, KOREA.

Abstract

A method of principal component analysis (PCA) based on a new L1-norm optimization technique is proposed. Unlike conventional PCA which is based on L2-norm, the proposed method is robust to outliers because it utilizes L1-norm which is less sensitive to outliers. It is invariant to rotations as well. The proposed L1-norm optimization technique is intuitive, simple, and easy to implement. It is also proven to find a locally maximal solution. The proposed method is applied to several datasets and the performances are compared with those of other conventional methods.

Index Terms

PCA-L1, L1-norm, optimization, principal component analysis, robust.

I. Introduction

In data analysis problems with a large number of input variables, dimensionality reduction methods are typically used to reduce the number of input variables to simplify the problems without degrading performances. Among them, the principal component analysis [1] is one of the most popular methods. In PCA, one tries to find a set of projections that maximize the variance of given data. These projections constitute a low-dimensional linear subspace by which the data structure in the original input space can effectively be captured.

Although the conventional PCA based on L2-norm (L2-PCA[1]) has been successful for many problems, it is prone to the presence of outliers, because the effect of the outliers with a large norm is exaggerated by the use of L2-norm. In order to alleviate this problem and achieve robustness, many researches have been performed [2] – [7].

In [5], [6] and [7], each component of the error between a projection and the original data point was assumed to follow a Laplacian distribution instead of Gaussian and L1-norm PCA (L1-PCA) was formulated by applying maximum likelihood estimation to the given data. In order to obtain a solution of L1-PCA, a heuristic estimates for the general L1 problem was used in [5], while in [6] and [7], the weighted median method and convex programming methods were proposed. Despite the robustness of L1-PCA, it has several drawbacks. First of all, it is computationally expensive because it is based on linear or quadratic programming. Secondly, it is not invariant to rotations.

[1] In order to prevent confusion, the conventional PCA will be referred to as L2-PCA hereafter.

In [4], Ding *et al.* proposed R1-PCA which combines the merits of L2-PCA and those of L1-PCA. Unlike L1-PCA, it is rotational invariant while it successfully suppresses the effect of outliers as L1-PCA does. However, the method is highly dependent on the dimension m of a subspace to be found. For example, the projection vector obtained when $m = 1$ may not be in a subspace obtained when $m = 2$. Moreover, because it is an iterative algorithm based on a successive use of the power method [8], for a large dimensional input space, it takes a lot of time to achieve convergence.

The above methods try to minimize the error between a projection and the original data point in the original input space. If L2-norm is used as a distance measure, this goal can be achieved by SVD (singular value decomposition) [8] which is also equivalent to finding projections by which variances are maximized in the feature space[2]. In this paper, instead of maximizing variance which is based on L2-norm, a method that maximizes L1-norm in the feature space is presented to achieve robust and rotational invariant PCA. The proposed L1-norm optimization algorithm is intuitive, simple, and easy to implement. It is also proven to find a locally maximal solution.

The rest of this paper is organized as follows. In Section II, the problem is formulated. A new algorithm for the L1-norm optimization problem is presented and the local optimality of the algorithm is proven in Section III. The proposed method is applied to several pattern recognition problems and the performances are compared with those of other conventional methods in Section IV and conclusion follows in Section V.

II. Problem formulation

Let $X = [\boldsymbol{x}_1, \cdots \boldsymbol{x}_n] \in \Re^{d \times n}$ 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, $\{\boldsymbol{x}_i\}_{i=1}^n$ is assumed to have zero mean.

In L2-PCA, one tries to find an $m(< d)$ dimensional linear subspace by minimizing the error function:

$$E_2(W, V) = ||X - WV||_2^2 = \sum_{i=1}^{n} ||\boldsymbol{x}_i - \sum_{k=1}^{m} \boldsymbol{w}_k v_{ki}||_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{d}(x_{ji} - \sum_{k=1}^{m} w_{jk} v_{ki})^2 \qquad (1)$$

where $W \in \Re^{d \times m}$ is the projection matrix whose columns $\{\boldsymbol{w}_k\}_{k=1}^m$ constitute the bases of the m-dimensional linear subspace (i.e., feature space), $V \in \Re^{m \times n}$ is the coefficient matrix whose

[2] We use the term *feature space* for the space spanned by projections to differentiate it from the original input space.

(i,j)-th component v_{ij} corresponds to the i-th coordinate of $\boldsymbol{x}_j$ in the m-dimensional feature space spanned by W, and $||\cdot||_2$ denotes the L2-norm of a matrix or a vector. By projection theorem [9], for a fixed W, V that minimizes (1) is uniquely determined by $V = W^T X$.

The global minimum of (1) is provided by SVD [8] whose solution is also the solution of the following dual problem:

$$W^* = \underset{W}{\operatorname{argmax}} ||W^T S_x W||_2 = \underset{W}{\operatorname{argmax}} ||W^T X||_2, \quad \text{subject to} \quad W^T W = I_m, \tag{2}$$

where $S_x = XX^T$ is the covariance matrix of X and I_m is the $m \times m$ identity matrix. Note that (2) searches for a projection matrix W^* by which the variances of $W^T X$ are maximized.

It is known that L2-norm is sensitive to outliers and several approaches were presented to resolve this problem. From statistical point of view, the methods based on L1-norm are more robust to outliers than the methods based on L2-norm and in [5], [6] and [7], instead of L2-norm, L1-norm was used in the error function. In this case, the problem becomes finding W that minimizes the following error function:

$$E_1(W,V) = ||X - WV||_1 = \sum_{i=1}^{n} ||\boldsymbol{x}_i - \sum_{k=1}^{m} \boldsymbol{w}_k v_{ki}||_1 = \sum_{i=1}^{n}\sum_{j=1}^{d} |x_{ji} - \sum_{k=1}^{m} w_{jk} v_{ki}|. \tag{3}$$

Here, $||\cdot||_1$ denotes the L1-norm of a matrix or a vector.

Although (3) reduces the effect of outliers, it is not invariant to rotations and the shape of a equidistance surface $\{\boldsymbol{x} : ||\boldsymbol{x}||_1 = const\}$ becomes very skewed [4]. It is because the above L1-norm is calculated on the input space. Moreover, the exact solution of (3) is very hard to achieve.

To resolve this problem, in [4], R1-norm was defined and the approximate solution that minimizes the following error function was proposed:

$$E_{R1}(W,V) = ||X - WV||_{R1} \triangleq \sum_{i=1}^{n} \left(\sum_{j=1}^{d} (x_{ji} - \sum_{k=1}^{m} w_{jk} v_{ki})^2 \right)^{\frac{1}{2}}. \tag{4}$$

However, the solution of (4) depends on the dimension m of subspace to be found. In other words, the optimal solution W^{1*} when $m = 1$ is not necessarily a subspace of W^{2*} when $m = 2$. In addition, the minimization of (4) is also very difficult and a subspace iteration algorithm based on L1 norm estimation techniques such as Huber's M-estimator was used in [4].

In this paper, motivated by the fact that the L2 solutions of (1) and (2) are the same, to obtain a subspace which is not only robust to outliers but also invariant to rotations, instead of minimizing the L1 error function (3) in the original d-dimensional input space, we would like to maximize the L1 dispersion using L1-norm in the feature space as the following:

$$W^* = \underset{W}{\operatorname{argmax}} ||W^T X||_1 = \underset{W}{\operatorname{argmax}} \sum_{i=1}^{n} \sum_{k=1}^{m} |\sum_{j=1}^{d} w_{jk} x_{ji}| \quad \text{subject to} \quad W^T W = I_m. \tag{5}$$

Here, the constraint $W^T W = I_m$ is to ensure the orthonormality of the projection matrix. Note that the solutions of (3) and (5) are different because they are not dual to each other. However, the two solutions are equally important in representing data as (1) and (2) are equally important. In a sense, they are like two sides of a coin.

The solution of (5) is invariant to rotations because the maximization is done on the feature space and it is expected to be more robust to outliers than the L2 solution of (2).

As a downside, the optimal i-th projection $\boldsymbol{w}_i^\star$ of (5) varies with different m's as in R1-PCA and finding a global solution of (5) for $m > 1$ is very difficult. To ameliorate this problem, we simplify (5) into a series of $m = 1$ problems using a greedy search method. If we set $m = 1$, (5) becomes the following optimization problem:

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmax}} ||\boldsymbol{w}^T X||_1 = \underset{\boldsymbol{w}}{\operatorname{argmax}} \sum_{i=1}^{n} |\boldsymbol{w}^T \boldsymbol{x}_i| \quad \text{subject to} \quad ||\boldsymbol{w}||_2 = 1. \tag{6}$$

Although the successive greedy solutions of (6) may differ from the optimal solution of (5), it is expected to provide a good approximation for (5). In the following section, an algorithm to solve (6) and a greedy search algorithm for $m > 1$ are presented.

III. Solution: PCA based on L1-norm maximization

A. Algorithm: PCA-L1

From now on, we derive a new algorithm to solve (6). The optimization of this objective function is difficult because it contains absolute value operation, which is nonlinear. In order to find the projection vector $\boldsymbol{w}$ that maximizes this L1 objective function, the following algorithm is presented. We refer to the algorithm as PCA-L1 to differentiate it from the L1-PCA in [6] and [7].

(Algorithm: PCA-L1)

1) Initialization: Pick any $\boldsymbol{w}(0)$. Set $\boldsymbol{w}(0) \leftarrow \boldsymbol{w}(0)/||\boldsymbol{w}(0)||_2$ and $t = 0$.
2) Polarity check: For all $i \in \{1, \cdots, n\}$, if $\boldsymbol{w}^T(t)\boldsymbol{x}_i < 0$, $p_i(t) = -1$, 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 $\boldsymbol{w}(t) \leftarrow \boldsymbol{w}(t)/||\boldsymbol{w}(t)||_2$.
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 $\boldsymbol{w}^T(t)\boldsymbol{x}_i = 0$, set $\boldsymbol{w}(t) \leftarrow (\boldsymbol{w}(t) + \Delta\boldsymbol{w})/||\boldsymbol{w}(t) + \Delta\boldsymbol{w}||_2$ and go to Step 2. Here, $\Delta\boldsymbol{w}$ is a small nonzero random vector.
   c. Otherwise, set $\boldsymbol{w}^\star = \boldsymbol{w}(t)$ and stop.

Theorem 1: With the above PCA-L1 procedure, the projection vector $\boldsymbol{w}$ converges to $\boldsymbol{w}^\star$, which is a local maximum point of $\sum_{i=1}^{n} |\boldsymbol{w}^T\boldsymbol{x}_i|$.

Proof: Firstly, we can show that $\sum_{i=1}^{n} |\boldsymbol{w}^T(t)\boldsymbol{x}_i|$ is a non-decreasing function of t as the following:

$$\begin{aligned} \sum_{i=1}^{n} |\boldsymbol{w}^T(t)\boldsymbol{x}_i| &= \boldsymbol{w}^T(t)(\sum_{i=1}^{n} p_i(t)\boldsymbol{x}_i) \geq \boldsymbol{w}^T(t)(\sum_{i=1}^{n} p_i(t-1)\boldsymbol{x}_i) \\ &\geq \boldsymbol{w}^T(t-1)(\sum_{i=1}^{n} p_i(t-1)\boldsymbol{x}_i) = \sum_{i=1}^{n} |\boldsymbol{w}^T(t-1)\boldsymbol{x}_i|. \end{aligned} \tag{7}$$

In the above, the first inequality is due to the fact that $\{p_i(t)\}_{i=1}^{n}$ is the set of optimal polarity corresponding to $\boldsymbol{w}(t)$, such that for all i, $p_i(t)\boldsymbol{w}^T(t)\boldsymbol{x}_i \geq 0$. Note that the inner product of two vectors is maximized when the two vectors are parallel. Hence, the second inequality holds because $||\boldsymbol{w}(t)||_2 = ||\boldsymbol{w}(t-1)||_2$ $(=1)$ and the vectors $\boldsymbol{w}(t)$ $(= \frac{\sum_{i=1}^{n} p_i(t-1)\boldsymbol{x}_i}{||\sum_{i=1}^{n} p_i(t-1)\boldsymbol{x}_i||})$ and $\sum_{i=1}^{n} p_i(t-1)\boldsymbol{x}_i$ are parallel.

Because the objective function is non-decreasing and there are finite number of data points, the PCA-L1 procedure converges to a projection vector $\boldsymbol{w}^\star$.

Secondly, we show that the objective function has a local maximum value at $\boldsymbol{w}^\star$. This can be shown as follows.

Because $\boldsymbol{w}(t)$ converges to $\boldsymbol{w}^\star$ by the PCA-L1 procedure, $\boldsymbol{w}^{\star T} p_i(t)\boldsymbol{x}_i \geq 0$ for all i. Since the number of data points is finite and $\boldsymbol{w}^{\star T}\boldsymbol{x}_i \neq 0$ for all i which is ensured by Step 4b, there exists a small neighborhood $N(\boldsymbol{w}^\star)$ of $\boldsymbol{w}^\star$ such that if $\boldsymbol{w} \in N(\boldsymbol{w}^\star)$, then $\boldsymbol{w}^T p_i(t)\boldsymbol{x}_i \geq 0$ for all i. Since $\boldsymbol{w}^\star$ is parallel to $\sum_{i=1}^{n} p_i(t)\boldsymbol{x}_i$, the inequality $\sum_{i=1}^{n} |\boldsymbol{w}^{\star T}\boldsymbol{x}_i| > \sum_{i=1}^{n} |\boldsymbol{w}^T\boldsymbol{x}_i|$ holds for all $\boldsymbol{w} \in N(\boldsymbol{w}^\star)$ and $\boldsymbol{w}^\star$ is a local maximum point.

Therefore, the PCA-L1 procedure finds a local maximum point $\boldsymbol{w}^\star$. ■

Because the projection vector is a linear combination of data points $\boldsymbol{x}_i$'s, i.e., $\boldsymbol{w}(t) \propto \sum_{i=1}^{n} p_i(t-1)\boldsymbol{x}_i$, it is naturally invariant to rotations.

The computational complexity of the proposed algorithm is $\mathcal{O}(nd) \times n_{it}$ where n_{it} is the number of iterations for convergence. It is clear that the number of iterations does not depend on the dimension d of input space but depends only on the number of samples n. Therefore, PCA-L1 can be applied to problems with a large number input variables without adding much computational complexity.

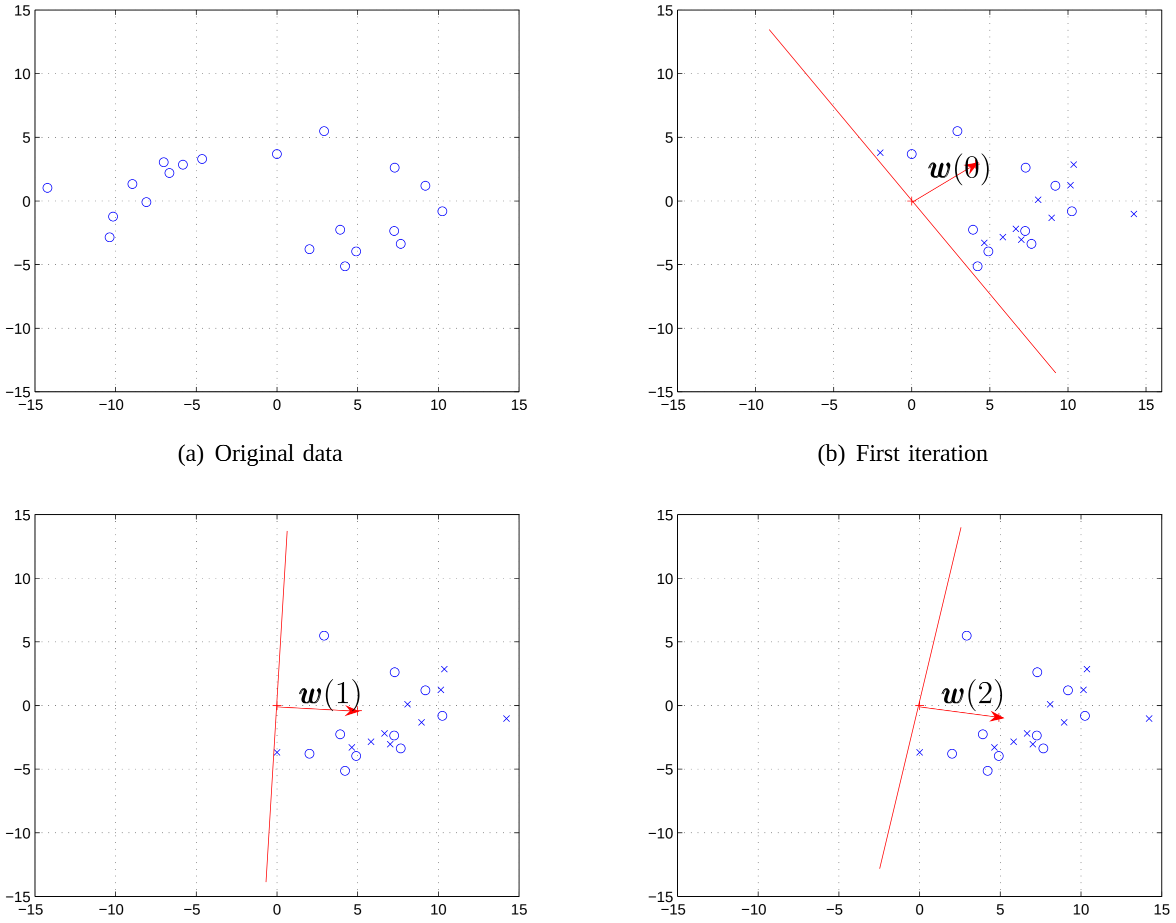
Note that this procedure tries to find a local maximum solution and there is a possibility that it may not be the global solution. However, considering that the initial vector $\boldsymbol{w}(0)$ can be set arbitrarily, by setting $\boldsymbol{w}(0)$ appropriately, e.g., by setting $\boldsymbol{w}(0) = \mathrm{argmax}_{\boldsymbol{x}_i} ||\boldsymbol{x}_i||_2$ or by setting it to the solution of the L2-PCA, we expect to find the global maximum point with higher probability in fewer iterations. In other approach, we can run the PCA-L1 procedure several times with different initial vectors and output the projection vector that gives the maximum L1 dispersion.

B. Examples

The PCA-L1 procedure is depicted in Fig. 1. In this example, Fig. 1 (a) is the original dataset which has been created as follows. Firstly, 20 random data points $\{(a_i, b_i)\}_{i=1}^{20}$ were generated in a two dimensional space with the mean distance of 5 from the origin and the variance of 1 with a uniform random orientation. And the point (a_i, b_i) is transformed to $(2a_i, b_i)$ for all i.

If we set the initial projection $\boldsymbol{w}(0) = [0.8151, 0.5794]^T$ randomly as shown in Fig. 1 (b), the polarities of the points which are below the line orthogonal to $\boldsymbol{w}(0)$ are set to -1 in the polarity checking step and these are flipped across the origin and marked as 'x'. By summing up all the points marked as 'o' and 'x' and normalizing it, we get a new projection vector $\boldsymbol{w}(1) = [0.9967, -0.0812]^T$ as shown in Fig. 1 (c). By the same procedure, we get $\boldsymbol{w}(2) = [0.9826, -0.1859]^T$ as shown in Fig. 1 (d). After this, the polarity of each point does not change and the convergence condition is fulfilled. Thus, $\boldsymbol{w}(2)$ becomes $\boldsymbol{w}^{\star}$, which is the global maximum point in this case.

We initialized the sample points as well as the projection vector $\boldsymbol{w}(0)$ randomly for this example 1,000 times. The solution was found in 3.26 iterations on average with the standard



(a) Original data

(b) First iteration

(c) Second iteration

(d) Third iteration (Final solution)

Fig. 1. PCA-L1 procedure

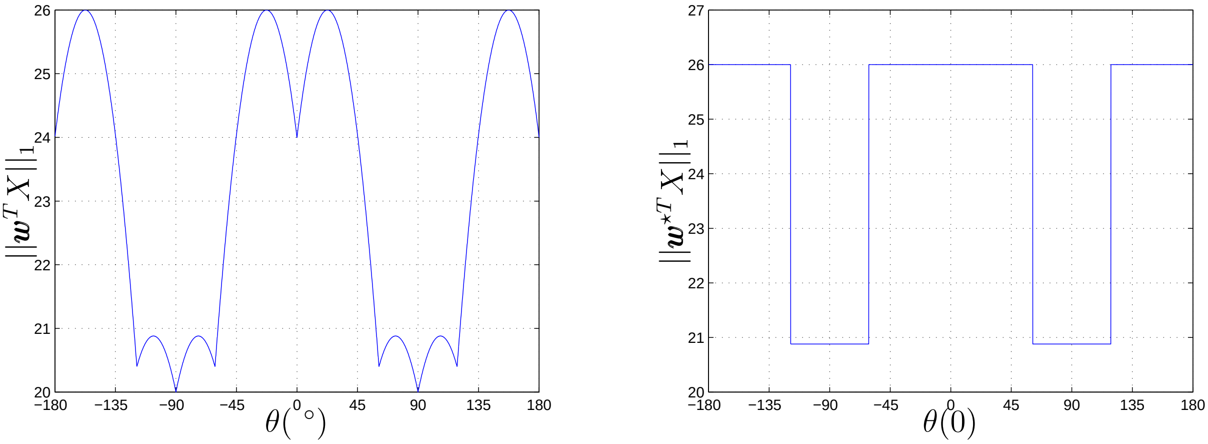
deviation of 0.8 for this 1,000 experiments. On the other hand, it required only 3 iterations in all 1000 experiments when we set the initial vector as $\boldsymbol{w}(0) = \text{argmax}_{\boldsymbol{x}_i} ||\boldsymbol{x}_i||_2$.

Step 4.b plays a crucial role in avoiding the solution to be stuck to a point which is not locally maximum. Consider the following two-dimensional dataset which consists of 5 data points:

$$X = \begin{bmatrix} 0 & 9 & -9 & 3 & -3 \\ 10 & -5 & -5 & 0 & 0 \end{bmatrix}.$$

Figure 2 (a) is the objective function $||\boldsymbol{w}^T X||_1 (= \sum_{i=1}^{5} |\boldsymbol{w}^T \boldsymbol{x}_i|)$ with respect to $\boldsymbol{w} = [\cos\theta, \sin\theta]^T$ for $\theta \in \{-180^\circ, 180^\circ\}$.

If the initial projection vector was set to $\boldsymbol{w}(0) = [0, 1]^T$, the polarities of the five data points will be $\{1, -1, -1, 1, 1\}$ and $\boldsymbol{w}(1)$ becomes the same as $\boldsymbol{w}(0)$. Therefore, if there have not been



(a) Objective function

(b) Dependency on initial vector $\boldsymbol{w}(0)$ ($\theta(0)$)

Fig. 2. Local optimality of the PCA-L1 algorithm

Step 4.b, the resulting projection vector would have been $\boldsymbol{w}^\star = [0, 1]^T$ ($\theta = 90^\circ$) which is actually the minimum point as can be seen from Fig. 2 (a). With Step 4.b, we can avoid setting the initial vector as $\boldsymbol{w}(0) = [0, 1]^T$.

Figure 2 (b) shows the local optimality of the PCA-L1 algorithm. In the figure, the horizontal axis represents the angle $\theta(0)$ of the initial projection $\boldsymbol{w}(0)$ and the vertical axis is the final objective function $||\boldsymbol{w}^{\star T} X||_1$ corresponding to the initial projection $\boldsymbol{w}(0)$. Comparing it to Fig. 2 (a), we can see that regardless of the initial projection $\boldsymbol{w}(0)$, the local optimality of the algorithm is guaranteed as *Theorem 1* states. However, we can also see that in the figure, the global optimality is not achieved. With the initial angle around $\pm 90^\circ$ it only converges to a local maximum point but not to a global maximum point. If we set the initial projection as $\boldsymbol{w}(0) = \mathrm{argmax}_{\boldsymbol{x}_i} ||\boldsymbol{x}_i||_2 = [9, -5]^T$ ($\theta(0) \simeq -30^\circ$), it converges to a global maximum point.

C. Extracting Multiple Features ($m > 1$)

Until now, we have shown that we can extract one best projection or feature[3] that maximizes the L1 objective function (6). The proposed method can be easily extended to extract arbitrary number of features by applying the same procedure greedily to the remainder of the projected samples as follows:

[3] Given a projection vector $\boldsymbol{w}$, the corresponding feature is defined as $f = \boldsymbol{w}^T \boldsymbol{x}$ where $\boldsymbol{x}$ denotes a sample point.

(Greedy search algorithm)

```
$\boldsymbol{w}_0 = \mathbf{0}, \{\boldsymbol{x}_i^0 = \boldsymbol{x}_i\}_{i=1}^n$.
For $j = 1$ to $m$,
    For all $i \in \{1, \cdots, n\}$, $\boldsymbol{x}_i^j = \boldsymbol{x}_i^{j-1} - \boldsymbol{w}_{j-1}(\boldsymbol{w}_{j-1}^T \boldsymbol{x}_i^{j-1})$.
    In order to find $\boldsymbol{w}_j$, apply the PCA-L1 procedure to $X^j = [\boldsymbol{x}_1^j, \cdots, \boldsymbol{x}_n^j]$.
end
```

By this procedure, the orthonormality of the projection vectors are guaranteed as follows:

1) Because $\boldsymbol{w}_j$ is a linear combination of the samples X^j, it is in the subspace spanned by X^j.
2) By multiplying $\boldsymbol{w}_{j-1}^T$ to the right side of X^j, we get $\boldsymbol{w}_{j-1}^T X^j = \boldsymbol{w}_{j-1}^T X^{j-1} - \boldsymbol{w}_{j-1}^T \boldsymbol{w}_{j-1} \boldsymbol{w}_{j-1}^T X^{j-1} = \boldsymbol{w}_{j-1}^T X^{j-1} - \boldsymbol{w}_{j-1}^T X^{j-1} = 0$.
3) From 2), $\boldsymbol{w}_{j-1}$ is orthogonal to X^j, which again shows that $\boldsymbol{w}_j$ is orthogonal to $\boldsymbol{w}_{j-1}$ by 1). ■

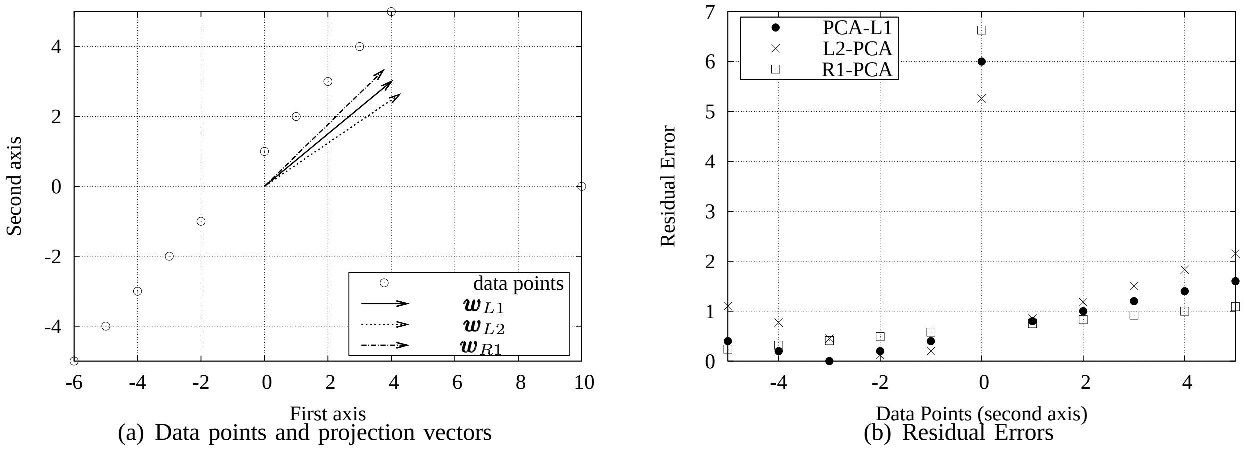
Note that even if this greedy algorithm does not provide the optimal solution of (5), it is expected to provide a set of good projections that maximizes L1 dispersion.

In conventional L2-PCA, the relative importance of a feature is usually computed by the corresponding eigenvalue of the covariance matrix S_x in (2) because the i-th eigenvalue is equivalent to the variance of the i-th feature. Since the total variance of a dataset is the same as the sum of all the variances of each feature, the number of extracted feature m is usually set by comparing the sum of variances up to m features and the total variance, i.e., if the sum of variances $s(j) = \sum_{i=1}^j \lambda_i$ exceeds e.g. 95% of the total variance, m is set to j. Here, λ_i is the i-th largest eigenvalue of S_x.

Likewise, in PCA-L1, once $\boldsymbol{w}_j$ is obtained, the variance of the j-th feature $\chi_j = \frac{\sum_{i=1}^n (\boldsymbol{w}_j^T \boldsymbol{x}_i)^2}{n}$ can be computed and the sum $s(j) = \sum_{i=1}^j \chi_i$ can be compared with the total variance $s = \frac{\sum_{i=1}^n ||\boldsymbol{x}_i||_2^2}{n}$ to set an appropriate number of extracted features.

IV. Experimental Results

In this section, we applied the proposed PCA-L1 algorithm to several pattern recognition problems and compared the performance with those of R1-PCA [4] and L2-PCA. In all the experiments, Huber's M-estimator was used for R1-PCA and the convergence condition for



(a) Data points and projection vectors

(b) Residual Errors

Fig. 3. A toy problem with an outlier

R1-PCA was set if the difference between the norms of Lagrangian multipliers in successive iterations was less than 10^{-3} or maximum number of iterations of 50 was reached [4]. All the experiments were performed by *MATLAB* on a Pentium D 3.40GHz processor.

A. A toy problem with an outlier

Consider the following measurement matrix X consisting of eleven data points in a two dimensional space.

$$X = \begin{bmatrix} -6 & -5 & -4 & -3 & -2 & \mathbf{10} & 0 & 1 & 2 & 3 & 4 \\ -5 & -4 & -3 & -2 & -1 & \mathbf{0} & 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

It is obvious that the sixth data point in boldface is an outlier and if we discard the data point, the projection vector should be $\boldsymbol{w} \propto [1, 1]^T(\theta = 45°)$.

For this data, L2-PCA, R1-PCA and PCA-L1 were applied and the projection vectors $\boldsymbol{w}_{L2} = [0.8507, 0.5257]^T$ $(\theta_{L2} = 31.7°)$, $\boldsymbol{w}_{R1} = [0.7483, 0.6634]^T$ $(\theta_{R1} = 41.6°)$ and $\boldsymbol{w}_{L1} = [0.8, 0.6]^T$ $(\theta_{L1} = 36.9°)$ were obtained respectively as shown in Fig. 3(a). In this experiment, PCA-L1 was randomly initialized and only 2 iterations were taken for convergence. On the other hand, R1-PCA converged in 7 iterations.

Figure 3(b) shows the residual error e_i of each data point where it was calculated as $e_i = ||\boldsymbol{x}_i - \boldsymbol{w}\boldsymbol{w}^T\boldsymbol{x}_i||_2$. The average residual errors of PCA-L1, L2-PCA and R1-PCA were 1.200, 1.401, and 1.206 respectively. With this result, we can see that L2-PCA was much influenced by the

TABLE I

UCI DATASETS USED IN THE EXPERIMENTS

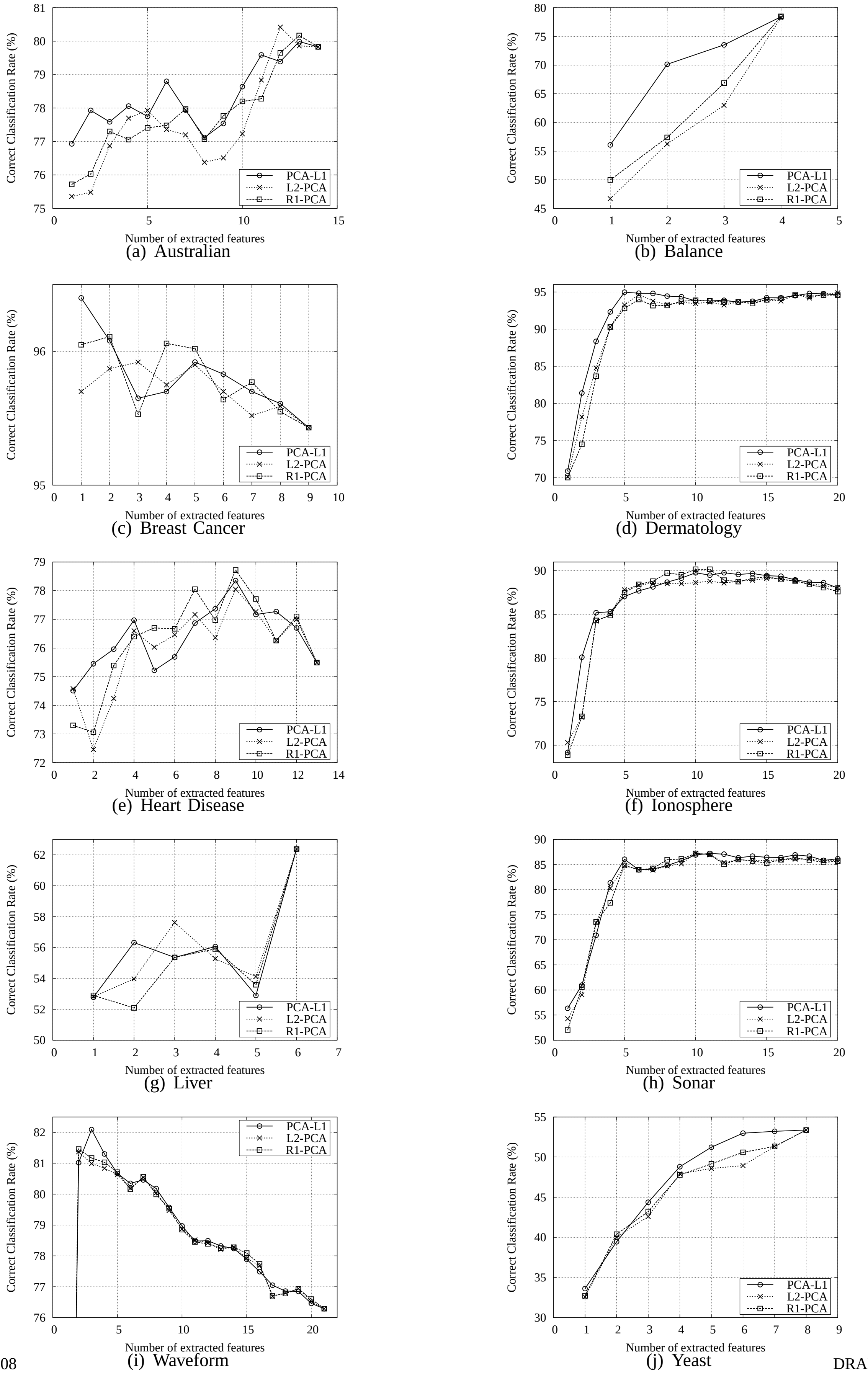
Data set	No. of variables (d)	No. of classes	No. of instances
Australian	14	2	690
Balance	4	3	625
Breast cancer	9	2	683
Dermatology	34	6	358
Heart disease	13	2	297
Ionosphere	33	2	351
Liver	6	2	345
Sonar	60	2	208
Waveform	21	3	4999
Yeast	8	10	1484

outlier while R1-PCA and PCA-L1 suppressed the effect of the outlier efficiently. Considering that the object of R1-PCA is to minimize the average residual error, it is quite impressive that the average residual error of PCA-L1 is smaller than that of R1-PCA. The reason is that R1-PCA does not solve the exact L1-norm minimization problem in (4) but an approximated one using L1-norm estimation techniques. Although it cannot be proven in this paper, this example shows a clue that the minimum residual error problem is closely related to maximizing L1 dispersion.

B. UCI datasets

We also applied PCA-L1 to several datasets in UCI machine learning repositories [10] and compared the classification performances with those of L2-PCA and R1-PCA. In all the experiments, the initial projection of PCA-L1 was set to the sample with the largest L2-norm, i.e., $\boldsymbol{w}(0) = \mathrm{argmax}_{\boldsymbol{x}_i} ||\boldsymbol{x}_i||_2$.

Table I shows a brief summary of the datasets used in this paper. These datasets have been used in many studies [11] [12] [13]. As a classifier, one nearest neighborhood (1-NN) classifier was used. For each dataset, we performed 10-fold cross validation (CV) 10 times and computed the average classification rate. Before training, each input variable in the training set was normalized to have zero mean and unit variance. The variables in the test set were also normalized using the means and the variances of the training set.



(a) Australian

(b) Balance

(c) Breast Cancer

(d) Dermatology

(e) Heart Disease

(f) Ionosphere

(g) Liver

(h) Sonar

(i) Waveform

(j) Yeast

Fig. 4. Correct Classification Rates for UCI datasets

TABLE II

AVERAGE CLASSIFICATION RATES ON UCI DATASETS (%). THE LAST COLUMN IS THE AVERAGES OF THE BEST CLASSIFICATION RATES AMONG THE CASES WHERE THE NUMBER OF EXTRACTED FEATURES WAS ONE TO HALF THE NUMBER OF ORIGINAL INPUTS

No. of extracted features	1	2	3	4	Best performance
L2-PCA	62.49	68.59	73.36	76.79	76.46
R1-PCA	62.44	68.49	73.63	76.52	76.47
PCA-L1	63.94	71.88	74.90	77.43	78.15

Figure 4 shows the average correct classification rates of each dataset with various numbers of extracted features. The number of extracted features m was varied from one to the dimension of original input space d. For datasets with a large number of input variables such as 'Dermatology', 'Ionosphere', and 'Sonar' datasets, the numbers of extracted features in the figure were truncated at 20 for clear view.

Comparing the performance of PCA-L1 and other methods, we can see that in many cases, PCA-L1 outperformed L2-PCA and R1-PCA when the number of extracted features was small. This phenomenon is clear in Table II which shows the average classification rate of these 10 datasets for a fixed number of extracted features from one to four. The last column of the table shows the averages of the best classification rates among the cases where the number of extracted features was one to half the number of original inputs. In the table, we can see that PCA-L1 outperformed other methods by more than 1% on average when the number of extracted features was one to three.

Regarding the computational cost, Table III shows the average time taken for L2-PCA, R1-PCA and PCA-L1. For R1-PCA and PCA-L1, average numbers of iterations are also shown. Because the i-th projection vector of R1-PCA varied with different numbers of extracted features, the reported time and iterations for R1-PCA are the average values of different numbers of extracted features. On the other hand, the time and iterations for L2-PCA and PCA-L1 were obtained when the number of extracted features is equal to the number of input variables. For example, in obtaining Fig. 4(d), R1-PCA took 25,500 *ms* (750 *ms* $\times$ 34), while L2-PCA and PCA-L1 took 62 *ms* and 750 *ms* on average respectively. In the table, we can see that the PCA-L1

TABLE III

COMPUTATIONAL COST (TIME & AVERAGE NUMBER OF ITERATIONS FOR UCI DATASETS)

	Average time (msec)			Average number of iterations	
Data sets	L2-PCA	R1-PCA	PCA-L1	R1-PCA	PCA-L1
Australian	0	583	343	42.29	7.14
Balance	0	36	47	2.00	1.00
Breast cancer	0	547	266	45.44	9.78
Dermatology	62	750	750	45.47	12.82
Heart disease	0	239	141	36.61	6.53
Ionosphere	64	816	625	47.30	10.15
Liver	0	125	79	24.67	5.67
Sonar	47	1533	734	34.50	10.30
Waveform	16	6480	24063	46.00	52.52
Yeast	0	340	531	14.63	10.75

was faster than R1-PCA in many cases and PCA-L1 converged in less than 15 iterations except for ‘waveform’ dataset. For ‘waveform’ dataset, the time and average iterations were greatly increased because of the large number of samples (4,999).

C. Face reconstruction

In this part, the proposed PCA-L1 algorithm was applied to face reconstruction problems and the performances were compared with those of other methods. As in the previous subsection, the initial projection of PCA-L1 was set to the sample with the largest L2-norm.

The Yale face database consists of 165 gray-scale images of 15 individuals. There are 11 images per subject with different facial expressions or configurations. In [14], the authors report two types of databases: a closely cropped set and a full face set. In this paper, the full face set whose size is 100 $\times$ 80 pixels was used. Each pixel was regarded as an input variable which constitutes an 8,000 dimensional input space.

In the first experiment, among 165 images, 20% were randomly selected and occluded with a rectangular noise consisting of random black and white dots whose size was at least 15 $\times$ 10 located at a random position. The left column of Fig. 5 shows typical examples of occluded images.

To these image set, we applied L2-PCA (eigenface [15]), R1-PCA and PCA-L1 and extracted various numbers of features. By using only a fraction of features, we could reconstruct images such as the ones shown on the second to the fourth columns of Fig. 5 and computed the average reconstruction error with respect to the original unoccluded images as follows:

$$\bar{e}(m) = \frac{1}{n}\sum_{i=1}^{n} ||\boldsymbol{x}_i^{org} - \sum_{j=1}^{m} \boldsymbol{w}_k \boldsymbol{w}_k^T \boldsymbol{x}_i||_2. \tag{8}$$

Here, n is the number of samples which is 165 in this case, $\boldsymbol{x}_i^{org}$ and $\boldsymbol{x}_i$ are the i-th original unoccluded image and the i-th image used in the training respectively, and m is the number of extracted features.

Figure 6(a) shows the average reconstruction errors for various numbers of extracted features. In the figure, when the number of extracted features was small, the average reconstruction errors for different methods were almost the same. However, from around 10 features, the difference among different methods became apparent and PCA-L1 started to be better than the other methods. Figure 5 shows the original and the reconstructed images using 20 projection vectors respectively. In the figure, we can see that the reconstructed images by L2-PCA have lots of dots compared to those of other methods resulting in bad quality. Although the qualities of the reconstructed images by PCA-L1 and those of R1-PCA are not distinct in the figure, average reconstruction error of PCA-L1 was smaller than that of R1-PCA when 20 projection vectors were used.

Note that this problem is a typical example of small sample size problems where the dimension of input space is higher than the number of samples. For this kind of problems, there exists a high dimensional null space where all the samples are projected to the origin (zero). Considering that PCA-L1 involves only summation and negation of given samples, it can easily be shown that the result of PCA-L1 does not change whether it is performed on the original input space or on the subspace excluding the null space. Therefore, to expedite PCA-L1, in this experiment, L2-PCA was first performed to exclude the null space and then PCA-L1 procedure were performed. By doing this, the operation time of PCA-L1 was reduced from 11,342 *ms* to 2,453 *ms* (which includes 1.719 *ms*, the operation time of L2-PCA). Note that for such methods as L1-PCA which are not invariant to rotations, this kind of preprocessing cannot be performed because the solution will be altered. The average number of iterations for PCA-L1 was 7.51 regardless whether the data were preprocessed by L2-PCA or not. For this problem, R1-PCA was also preprocessed by

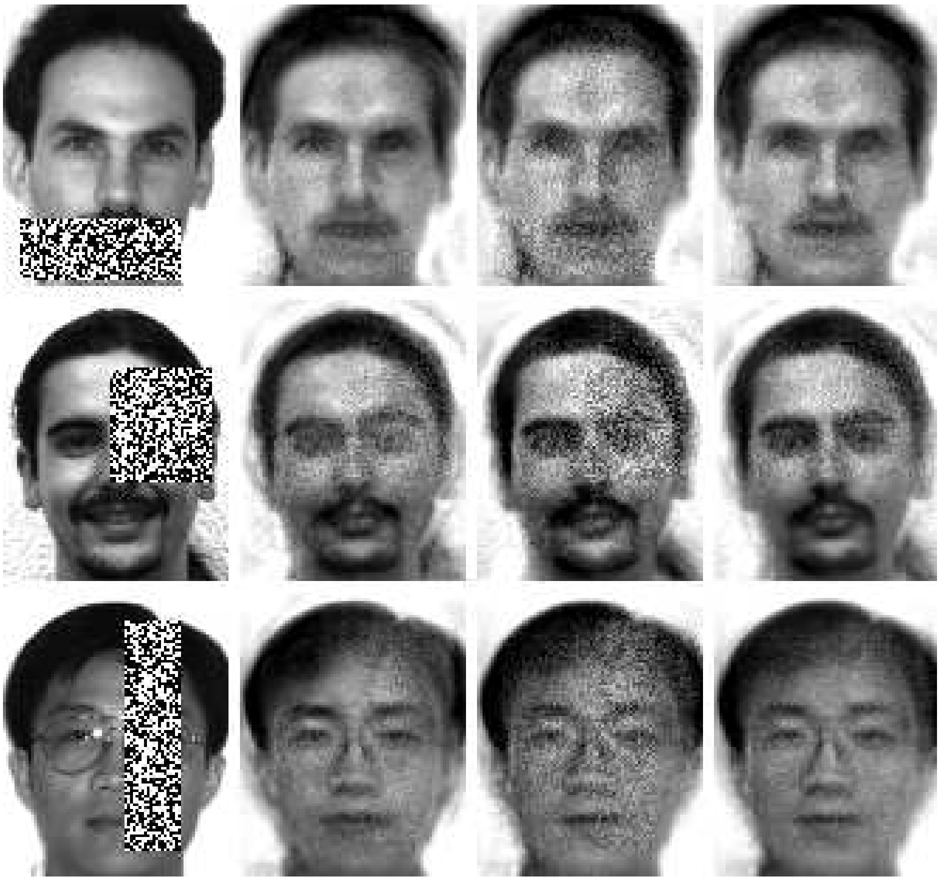
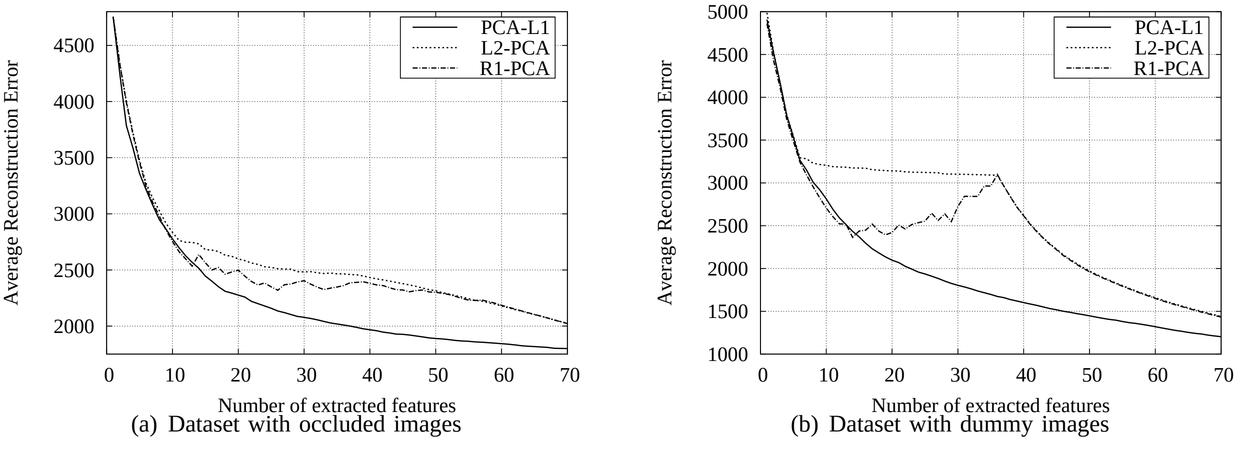


Fig. 5. Face images with occlusion and the reconstructed faces: 1st column: original, 2nd column: PCA-L1, 3rd column: L2-PCA, 4th column: R1-PCA. (reconstructed with 20 projection vectors)



(a) Dataset with occluded images

(b) Dataset with dummy images

Fig. 6. Average reconstruction errors for Yale dataset

L2-PCA and it took 14,898 *ms* on average, i.e., to obtain Fig. 6(a), R1-PCA took 1,042,860 *ms* (= 14,898 *ms* $\times$ 70).

As a second experiment, to the original 165 Yale images, we added 30 dummy images which consist of random black and white dots and performed L2-PCA, R1-PCA and PCA-L1. Figure 6(b) shows the average reconstruction error of each method with various numbers of extracted features. In the computation of average reconstruction error, (8) was used with $n = 165$, i.e., 30 dummy images were excluded. In this case, $\boldsymbol{x}_i^{org}$ and $\boldsymbol{x}_i$ were the same.

In the figure, when the number of extracted features is from 6 to 36, the error of L2-PCA is almost constant. This shows that the dummy images affected the 6th up to the 36th projection vectors seriously and these vectors were tuned to explain the dummy data. For R1-PCA, this phenomenon started later at around 13th projection vector while R1-PCA did not suffer from this phenomenon and the reconstruction error was smallest of the three after the 14th projection vector. The fluctuation of R1-PCA might be due to the fact that the whole projection vectors were replaced as the number of extracted features was varied.

Figure 7 shows the reconstructed images with 20 projection vectors as well as the original face images. The figure clearly shows that PCA-L1 is better than other methods in reconstructing original images when there are outliers.

The average number of iterations of PCA-L1 was 7.61 and it took 3,078 *ms* including 2,172 *ms* which was the time took for preprocessing by L2-PCA. For this problem, R1-PCA took 26,555 *ms* on average.

V. Conclusion

In this paper, we proposed a method of principal component analysis based on L1-norm optimization. The proposed PCA-L1 tries to find projections that maximizes L1-norm in the projected space instead of the conventional L2-norm. In due course, a new method of L1-norm optimization was introduced and was proven to find a local maximum point. The proposed L1-norm optimization technique is intuitive, simple, and easy to implement. In addition, it not only successfully suppresses the negative effects of outliers but also is invariant to rotations.

The computational complexity of the proposed method is proportional to the number of samples, the dimension of input space, and the number of iterations. Considering that the number

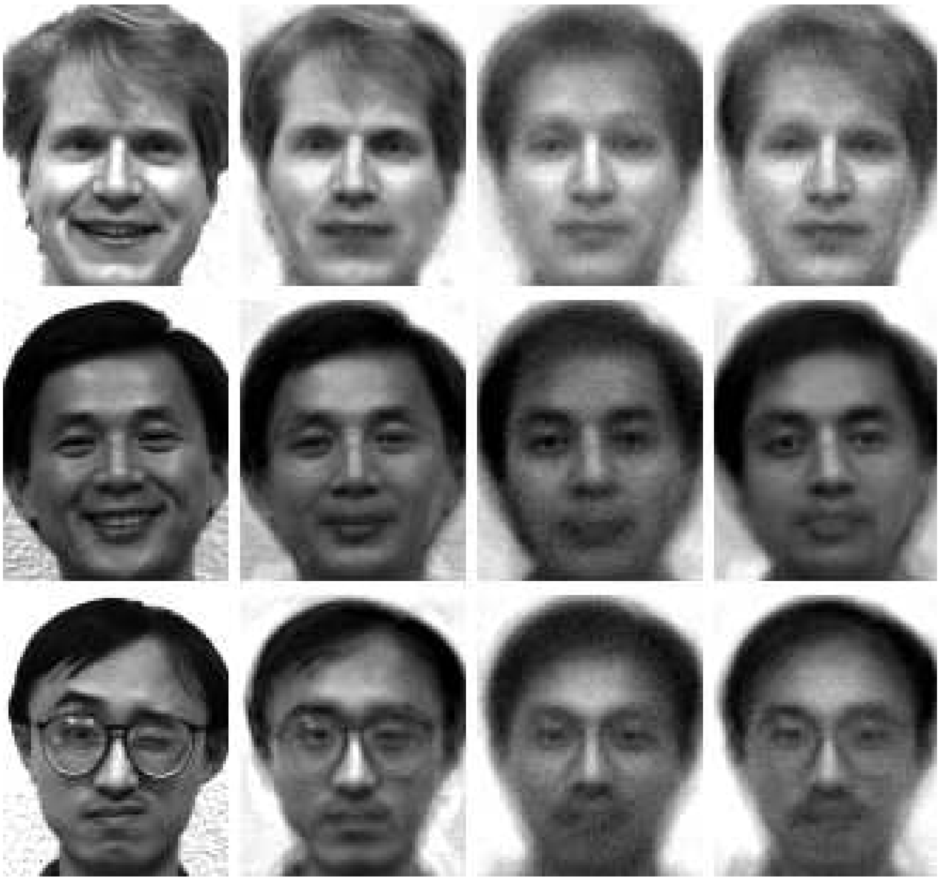


Fig. 7. Face images trained with dummy images and the reconstructed faces: 1st column: original, 2nd column: PCA-L1, 3rd column: L2-PCA, 4th column: R1-PCA. (reconstructed with 20 projection vectors)

of iterations does not depend on the dimension of input space, it is expected to perform well for the problems with large input dimension such as the ones that deal with images.

The proposed method was applied to several pattern recognition problems including face reconstruction problems and the performances were compared with those of the conventional L2-PCA and R1-PCA. The experimental results show that the proposed method is usually faster than R1-PCA and robust to outliers.

ACKNOWLEDGMENTS

This work was supported by the Korea Research Foundation Grant funded by Korean Government (KRF-2007-313-D00610).

REFERENCES

[1] I.T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, 1986.

[2] F. De la Torre and M.J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 117–142, Aug. 2003.

[3] H. Aanas, R. Fisker, K. Astrom, and J. Carstensen, "Robust factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1215–1225, Sep. 2002.

[4] C. Ding, D. Zhou, X. He, and H. Zha, "R1-pca: rotational invariant l1-norm principal component analysis for fobust subspace factorization," in *Proc. International Conference on Machine Learning*, Pittsburgh, PA, June 2006.

[5] A. Baccini, P. Besse, and A.D. Falguerolles, "A l1-norm pca and a heuristic approach," in *Ordinal and Symbolic Data Analysis*, E. Diday, Y. Lechevalier, and P. Opitz, Eds. 1996, pp. 359–368, Springer.

[6] Q. Ke and T. Kanade, "Robust subspace computation using l1 norm," Tech. Rep. CMU-CS-03-172, Carnegie Mellon University, Aug. 2003, http://citeseer.ist.psu.edu/ ke03robust.html.

[7] Q. Ke and T. Kanade, "Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2005.

[8] G. Golub and C.V. Loan, *Matrix Computation*, Johns Hopkins University Press, 3 edition, 1996.

[9] D.G. Luenberger, *Optimization by Vector Space Methods*, Wiley, New York, 1969.

[10] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz, "Uci repository of machine learning databases," 1998, For more information contact ml-repository@ics.uci.edu or http://www.ics.uci.edu/ mlearn/MLRepository.html.

[11] Marco Loog and Robert P.W. Duin, "Linear dimensionality reduction via a heteroscedastic extension of lda: The chernoff criterion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 732–739, June 2004.

[12] H. Xiong, M.N.S. Swamy, and M.O. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 460–474, March 2005.

[13] C.J. Veeman and M.J.T. Reinders, "The nearest subclass classifier: A compromise between the nearest mean and nearest neighbor classifier," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1417–1429, Sep. 2005.

[14] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, July 1997.

[15] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.