R_1-PCA: Rotational Invariant L_1-norm Principal Component Analysis for Robust Subspace Factorization

Chris Ding CHQDING@LBL.GOV
Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720

Ding Zhou DZHOU@CSE.PSU.EDU
Dept. of Computer Science and Engineering, The Pennsylvania State University, University Park, PA, 16801

Xiaofeng He XHE@LBL.GOV
Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720

Hongyuan Zha ZHA@CSE.PSU.EDU
Dept. of Computer Science and Engineering, The Pennsylvania State University, University Park, PA, 16801

Abstract

Principal component analysis (PCA) minimizes the sum of squared errors (L_2-norm) and is sensitive to the presence of outliers. We propose a *rotational invariant* L_1-norm PCA (R_1-PCA). R_1-PCA is similar to PCA in that (1) it has a unique global solution, (2) the solution are principal eigenvectors of a robust covariance matrix (re-weighted to soften the effects of outliers), (3) the solution is rotational invariant. These properties are not shared by the L_1-norm PCA. A new subspace iteration algorithm is given to compute R_1-PCA efficiently. Experiments on several real-life datasets show R_1-PCA can effectively handle outliers. We extend R_1-norm to K-means clustering and show that L_1-norm K-means leads to poor results while R_1-K-means outperforms standard K-means.

1. Introduction

Principal component analysis (PCA)(Jolliffe, 2002) is a widely-used method for dimension reduction. When data points lie in a low-dimensional manifold and the manifold is linear or nearly-linear, the low-dimensional structure of data can be effectively captured by a linear subspace spanned by the principal PCA directions.

In this paper, we address the issue of robustness of PCA in the presence of outliers, which we define as the points that deviates significantly from the rest of the data. Traditional PCA minimizes the sum of squared errors, which is prone to the presence of outliers, because large errors squared dominate the sum. Several robust PCA have been proposed(Torre & Black, 2003; Aanas et al., 2002).

Another approach uses the L_1-norm or the least absolute deviance, which is less sensitive to outliers compared to the Euclidean metric (L_2-norm). This has been proposed for K-means clustering and is recently (Ke & Kanade, 2004) extended to PCA.[1]

In this paper we propose the rotational invariant L_1-norm (we call it R_1-norm) for the objective functions of PCA. In R_1-norm, distance in spatial dimensions (attribute dimensions) are measured in L_2, while the summation over different data points uses L_1. Let $X = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ be n data points in d-dimensional space. In matrix form $X = (x_{ji})$, index j sum over spatial dimensions, $j = 1, \cdots, d$ and index i sum over data points, $i = 1, \cdots, n$. R_1-norm is defined as

$$||X||_{R_1} = \sum_{i=1}^{n} \Big(\sum_{j=1}^{d} x_{ji}^2 \Big)^{\frac{1}{2}}, \tag{1}$$

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

[1] L_1-norm originates from LASSO (Tibshirani, 1996), and has caught some interest in machine learning (Ng, 2004) and statistics. Besides the *robustness against outliers* context in this paper, L_1-norm is also used as a penalty/regularization term on model parameters to enforce *sparsity* , or parameter/feature selection, such as sparse PCA (Jolliffe, 2002; Zou et al., 2004), logistic-regression(Ng, 2004). In addition, L_0-norm (the number of nonzero) is also used (d'Aspremont et al., 2004; Zhang et al., 2004). L_1 robustness is different from L_1 sparsification: in sparsification L_1 is a constraint to the objective function while in robustness L_1 is on the main objective function itself.

while the Frobenius and L_1-norms are defined as[2]

$$||X||_{_F} = \Big(\sum_{i=1}^{n}\sum_{j=1}^{d} x_{ji}^2\Big)^{\frac{1}{2}},\ ||X||_{_{L_1}} = \sum_{i=1}^{n}\sum_{j=1}^{d}|x_{ji}|. \quad (2)$$

R_1-norm is indeed a *norm*: for any two matrices A, B, we can show that the triangle inequality holds, i.e., $||A+B||_{_{R_1}} \leq ||A||_{_{R_1}} + ||B||_{_{R_1}}$.

Rotational invariance is a fundamental property of Euclidean space with L_2-norm. It has been emphasized in the context of learning algorithms (Ng, 2004). For any orthogonal coordinate rotation R (an orthogonal matrix), and data point transformation $R : \mathbf{x}_i \leftarrow R\mathbf{x}_i$, the L_2-norm is invariant $||R\mathbf{x}_i|| = ||\mathbf{x}_i||$. In many applications, the dimension is high and we use PCA to project data into a low-dimensional subspace which reduces the noise at same time. A subspace is not uniquely determined up to an orthogonal transformations. Therefore, we prefer to model data with distributions that satisfy rotational invariance.

Another reason against *pure* L_1-norm PCA and *pure* L_1-norm K-means is the shape of the equi-distance surface of a given norm. In K-means the assignment of data points to centroids determines the shape of clusters which is the equi-distance surface. This surface in L_2-norm $||\mathbf{x}-\mu||^2 = const$ is a sphere, which is the same in R_1-norm. However in L_1-norm, the equi-distance surface $||\mathbf{x}-\mu||_1 = const$ is a simplex surface centered at coordinate origin. In high p-dimensional space, the simplex has very skewed surface. This can be seen from the ratio of longest direction vs. shortest direction which is $p/\sqrt{p} = \sqrt{p}$. For the newsgroups data (see §5) at $p = 500$, the ratio is $\sqrt{p} = 22.4$. Thus the clusters described by the L_1-K-means is far away from a Gaussian distribution. This is the reason why the L_1-K-means performs poorly (see §5). This motives us to propose the R_1-norm.

Our main results on the rotational invariant L_1-norm PCA (we call it R_1-PCA) are (1) The principal components in R_1-PCA are the principal eigenvectors of a robust (R_1) covariance matrix (re-weighted to soften outliers); (2) The solutions are rotational invariant. (3) An efficient subspace iteration based algorithm iteratively solve the nonlinear eigenvector problem of R_1-PCA. We show several experimental results on 4 real-life datasets, which illustrate the usefulness of the R_1-PCA in handling outliers. Properties (1) and (2) are shared by standard PCA, but not by L_1-PCA.

We further extend R_1-norm to to K-means and compare it with L_1-norm K-means in §5. Experiments on internet newsgroup data show that L_1-K-means clustering has very poor performance This poor performance is found to be caused by a key weakness of L_1-K-means , i.e., the assignment of a point data to nearest cluster centroid using L_1 distance. On datasets with noises, R_1-K-means performs slightly better than standard K-means .

2. Covariance, L_1-PCA and R_1-PCA

2.1. Two Formulations for PCA

Let $U = (\mathbf{u}_1, \cdots, \mathbf{u}_k)$ contains the principal *directions* and $V = (\mathbf{v}_1, \cdots, \mathbf{v}_k)$ contains the principal *components* (data projects along the principal directions). There are two formulations for PCA.

(a) Covariance based approach. Compute the covariance matrix $C = \sum_i (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T = XX^T$. Here we assume the data are already centered, $\bar{\mathbf{x}} = 0$, and we drop the factor $1/(n-1)$ which does not affect U. The principal directions are obtained as

$$\max_{U^TU=I} \mathrm{Tr}\ U^TXX^TU \quad (3)$$

(b) Matrix low-rank approximation based approach. Let $X = UV^T$. We solve

$$\min_{U,V} J,\ J = ||X - UV^T||_F^2 = \sum_{ij}[X_{ij} - (UV^T)_{ij}]^2. \quad (4)$$

For standard PCA, the solutions to these two approaches are identical, thanks to SVD.

The standard generalization to L_1-norm PCA is to solve (Ke & Kanade, 2004).

$$\min_{U,V} J,\ J = ||X - UV^T||_{_{L_1}} = \sum_{ij}|X_{ij} - (UV^T)_{ij}|. \quad (5)$$

There are several drawbacks of L_1-PCA: (1) Computationally expansive; (2) It is not clear whether the solution U relates to the covariance matrix; (3) Questions relating to use L_1 in clustering (see §5).

A common feature of previous approaches using Frobenius norm and L_1-norm is that they treat the two indexes i and j in the same way. However, these two indexes have different meaning: i runs through data points, while $j = 1$ run through the spatial dimensions. In strict matrix format, this subtle distinction is easy to get lost. R_1-norm captures this subtle distinction.

[2] The L_p-norm of a vector $\mathbf{x}$ in d-dimensional space is $||\mathbf{x}||_p = (\sum_{j=1}^d |x_j|^p)^{1/p}$. By convention, $||\mathbf{x}|| \equiv ||\mathbf{x}||_2$.

2.2. Rotational Invariant L_1-norm PCA

We first express R_1-norm in vector format. Let $V = (\mathbf{v}_1, \cdots, \mathbf{v}_n) \in R^{k \times n}$ and we write

$$X \simeq UV \tag{6}$$

in contrast to $X \simeq UV^T$. The standard PCA can be formulated as

$$\min_{U,V} J_{\text{SVD}} = ||X - UV||_F^2 = \sum_{i=1}^{n} ||\mathbf{x}_i - U\mathbf{v}_i||^2. \tag{7}$$

In R_1-PCA, we use R_1-norm,

$$\min_{U,V} J_{\text{R1-PCA}} = ||X - UV||_{R_1} = \sum_{i=1}^{n} ||\mathbf{x}_i - U\mathbf{v}_i||. \tag{8}$$

An algorithm can be developed for alternatively updating U (while fixing V) and V (while fixing U). Here we develop a more efficient algorithm to solve this problem. It uses the covariance matrix, thus is statistically more interesting.

First, we can require U to be orthonormal without losing generality. Second, given a fixed U, we solve for V according to Eq.(8). Different column vectors $\mathbf{v}_i$ of V can be solved independently. Solving $\min ||\mathbf{x}_i - U\mathbf{v}_i||^2$, the solution is $\mathbf{v}_i = (U^TU)^{-1}U^T\mathbf{x}_i$. Applying to all columns, we obtain the solution $V = (\mathbf{v}_1, \cdots, \mathbf{v}_n) = (U^TU)^{-1}U^TX$. Now since we require U to be orthonormal, $V = (U^TU)^{-1}U^TX = U^TX$. Thus

$$||\mathbf{x}_i - U\mathbf{v}_i|| = \sqrt{\mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^TUU^T\mathbf{x}_i} \equiv s_i. \tag{9}$$

The approximation error s_i is the distance of $\mathbf{x}_i$ to the subspace. Thus the R_1-PCA optimization problem is simplified to

$$\min_{U^TU=I} J_{\text{R1-PCA}} = \sum_{i=1}^{n} \sqrt{\mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^TUU^T\mathbf{x}_i}. \tag{10}$$

The standard PCA (SVD) can be similarly written as the solution to the optimization problem

$$\min_{U^TU=I} J_{\text{PCA}} = \sum_{i=1}^{n} (\mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^TUU^T\mathbf{x}_i). \tag{11}$$

Clearly, $J_{\text{R1-PCA}}(U)$ and $J_{\text{PCA}}(U)$ are convex functions of UU^T, since each term in both $J_{\text{R1-PCA}}$ and J_{PCA} is a convex function of UU^T. Thus we have
Proposition 0. Both PCA and R_1-PCA have a unique global optimal solution. [3] For PCA, this is well-known. For R_1-PCA, this ensures a unique and well-behaved solution.

For PCA, U is the principal eigenvectors of the covariance matrix $C = XX^T = \sum_i \mathbf{x}_i\mathbf{x}_i^T$. For R_1-PCA, we have a similar result (the main result of this paper):
Theorem 1. The solution to R_1-PCA are the principal eigenvectors of the R_1-covariance matrix

$$C_r = \sum_i w_i\mathbf{x}_i\mathbf{x}_i^T,\ w_i^{(L_1)} = \frac{1}{||\mathbf{x}_i - UU^T\mathbf{x}_i||}, \tag{12}$$

This is a weighted version of the covariance matrix.

2.3. Rotational Invariance of the Solutions of R_1-PCA and PCA

In previous sections, "rotational invariance" is w.r.t. to the objective function. But "rotational invariance" can also be w.r.t. to the solution. This means that under a rotational transformation of the feature space $R : \mathbf{x}_i \leftarrow R\mathbf{x}_i$, the solution of PCA satisfy: (1) principal directions (columns of U) are rotated accordingly, $\mathbf{u}_k \leftarrow R\mathbf{u}_k$; (2) principal components V remains fixed. PCA solution has the rotational invariance property.

Theorem 2. R_1-PCA solution has the rotational invariance property, while L_1-PCA does not.

Proof. Since R is orthogonal, i.e., $R^TR = I$. The L_2-norm of a vector has

$$||\mathbf{x}_i - U\mathbf{v}_i|| = ||(R^TR)(\mathbf{x}_i - U\mathbf{v}_i)||$$

$$= ||R^T(R\mathbf{x}_i - RU\mathbf{v}_i)|| = ||R\mathbf{x}_i - RU\mathbf{v}_i||$$

This show under the transformation of $X \leftarrow RX$, $U \leftarrow RU$; and V remains unchanged. Thus PCA and R_1-PCA have the rotational invariance, because they use L_2-norm in spatial dimensions. For L_1-norm, in general $||R^T(R\mathbf{x}_i - RU\mathbf{v}_i)||_1 \neq ||R\mathbf{x}_i - RU\mathbf{v}_i||_1$. Thus L_1-PCA does not has rotational invariance. □

Proposition 0 and Theorems 1,2 show that R_1-PCA is very similar to standard PCA. Furthermore, R_1-PCA can be solved by an efficient subspace iteration algorithm.

3. R$_1$-PCA Algorithm

3.1. R_1-PCA Using Generic Robust Estimator

We first generalize rotational invariant L_1-PCA of Eq.(10) using a generic *loss* function $\rho(\cdot)$ as

$$\min_{U^TU=I} J_r = \sum_{i=1}^{n} \rho(\sqrt{\mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^TUU^T\mathbf{x}_i}\ ) \tag{13}$$

[3] Although UU^T is unique, U is unique up to an orthogonal transformation R. In Theorem 3, once C_r is computed, the solution is unique.

Many forms for the loss function are possible. $\rho(s) = |s|$ recovers the rotational invariant L_1 measure. Another popular robust estimation is Huber's M-estimator,

$$\rho_H(s) = \begin{cases} s^2 & \text{if} \quad |s| \le c \\ 2c|s| - c^2 & \text{if} \quad |s| > c \end{cases} \qquad (14)$$

We call the parameter c "cutoff" for its regularization effect of the weights in the R_1 covariance matrix (see §3.4). Another robust estimation is Cauchy function

$$\rho_C(s) = c^2 \log(1 + s^2/c^2) \qquad (15)$$

Like M-estimator, at small distance $s \ll c$, $\rho_C(s) = s^2$, reducing to Euclidean metric.

Let us define the R_1 covariance matrix

$$C_r = \sum_i w_i \mathbf{x}_i \mathbf{x}_i^T \qquad (16)$$

where the weight is, for Huber's M-estimator,

$$w_i^{(H)} = \begin{cases} 1 & \text{if } ||\mathbf{x}_i - UU^T\mathbf{x}_i|| \le c \\ c/||\mathbf{x}_i - UU^T\mathbf{x}_i|| & \text{otherwise} \end{cases} \qquad (17)$$

which reduces to the L_1 form of Eq.(12) for $c \to 0$ (more precisely, $\rho_H(c)/c \to ||s||_1$). For the Cauchy robust function, the weight is

$$w_i^{(C)} = \left(1 + ||\mathbf{x}_i - UU^T\mathbf{x}_i||^2/c^2\right)^{-1} \qquad (18)$$

The main difference between this R_1 covariance and the usual covariance is to reduce the weight or contribution from those "outlying" points (whose distance to its projection in the subspace s_i are larger than cutoff c).

Theorem 3. The global optimal solution for R_1-PCR are given by the principal eigenvectors of C_r i.e.,

$$C_r \mathbf{u}_k = \lambda_k \mathbf{u}_k.$$

Proof. We follow the standard theory of constrained optimization and introducing the Lagrangian function

$$L = \sum_{i=1}^n \rho(\sqrt{\mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^T UU^T \mathbf{x}_i}\,) + \mathrm{Tr}\Lambda(U^TU - I), \qquad (19)$$

where the Lagrangian multipliers $\Lambda = (\Lambda_{k\ell})$ for enforcing the orthonormal constraints $U^TU = I$. The KKT condition for optimal solution specifies that the gradient of L must be zero:

$$\frac{\partial L}{\partial U} = -2\sum_i w_i \mathbf{x}_i \mathbf{x}_i^T U + 2U\Lambda = 0, \qquad (20)$$

where the specific form of w_i depends on the robustness function $\rho(\cdot)$, and are given by Eqs.(12,17,18) for L_1, Huber and Cauchy functions. Eq.(20) gives the fixed point relation

$$C_r U = U\Lambda. \qquad (21)$$

Left multiply by U^T, we obtain the Lagrangian multipliers as

$$\Lambda = U^T C_r U. \qquad (22)$$

Generally speaking, Lagrangian multipliers Λ could take any values. In particular, the off-diagonal elements of Λ does not have to be zero.

However, we recognize that there is an unique solution to Eq.(21), which are the eigenvectors of the symmetric positive definite matrix C_r. And the Lagrangian multipliers Λ becomes a diagonal matrix: $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_k)$. Now, according to KKT Theory, the solution to Lagrangian multipliers are unique under general conditions. Therefore, the eigenvector solutions to Eq.(21) must be the unique and global solution. □

3.2. Subspace Iteration Algorithm

Now we provide an efficient algorithm to compute the solution to Eq.(21). First, we recognize this is a nonlinear eigenvalue problem, since R_1 covariance matrix C_r is dependent on U in a non-trivial way. Fortunately, all we need are the k eigenvectors corresponding to the k large eigenvalues. This is precisely the principal subspace of C_r. There exists a well-known subspace iteration algorithm in matrix theory (Golub & Van Loan, 1996) that can efficiently compute the principal subspace.

The basic idea is the following. We start with an initial guess $U^{(0)}$, which we take as the principal directions of the standard covariance matrix. From $U^{(0)}$, we compute the R_1-covariance $C_r(U^{(0)})$. U is updated using the power method and while maintaining orthogonality:

$$U^{(t+\frac{1}{2})} = C_r(U^{(t)})U^{(t)}, \qquad (23)$$

$$U^{(t+1)} = \mathrm{orthogonalize}(U^{(t+\frac{1}{2})}) \qquad (24)$$

This update reduces L in each step of the way. At convergence, $C_r(U^{(t)})$ converges to its asymptotic value: $C_r(U^{(\infty)}) \equiv C_r$. $U^{(t)}$ converge to the eigenvectors of C_r. The Lagrangian multiplier $\Lambda = U^T C_r U$ converge to the diagonal matrix containing eigenvalues.

3.3. Effects of Cutoff in Huber's Estimator

Now we discuss the effects of the cutoff c in Eqs.(14,17) and how to specify it. First, we consider the case when

$c = 0$, which is equivalent to use the rotational invariant L_1-norm as robustness function and is given in Eq.(12).

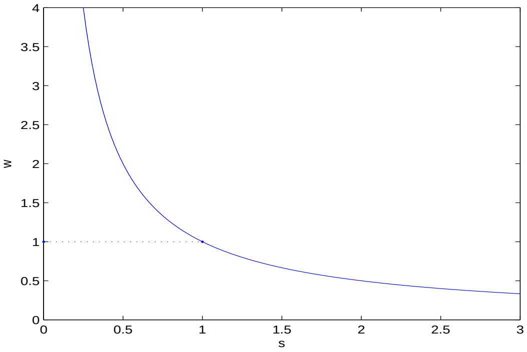


Figure 1. The weights in covariance matrix. The singularity (blow-up of the weight $w(s)$ near $s = 0$) of L_1-norm weight of Eq.(12) is being cutoff at 1 by the L_2-norm weight of Eq.(17) assuming $c = 1$.

Using this L_1 form of the R_1-covariance (we call it C_1), the subspace iteration algorithm of §3.3 works as well. However, because the weights are directly proportional to $1/s_i$, so C_1 is dominated by the data points with near-zero distance to subspace $s_i = ||\mathbf{x}_i - UU^T\mathbf{x}_i|| \simeq 0$. We have run the algorithm on a number of datasets. The solution of U always pass some of the data points. On these data points, the denominator is near zero. (This singularity problem can be temporarily prevented by adding a small number ϵ of the smallest quantity a computer can represents: $w_i^{(L_1)} = 1/(s_i + \epsilon)$.

To prevent this, we incorporate the cutoff c in Eqs.(14,17) The effects of the cutoff can be seen in Figure 1. When s_i is larger than the cutoff, we use L_1-norm weight of $1/s$. Otherwise, we reverse back to L_2-norm weight of 1.

How to determine the cutoff? From Figure 1, we see that as long as the singularity (blow-up nero $s = 0$) is cutoff, the weight curve is not particularly sensitive to the exact value of c. This is crucial for the stability of our R_1 approach — if the final subspace obtained is very sensitive to cutoff, then it is not well defined.

This important stability property also makes the choice of c easy. A general motivation and also quantitative goal for the choice of c is to cut off outliers, that is, using L_1 distance on them. In most applications, the number of outliers are small. Therefore, a reasonable choice is to set c at median of $(s_1, \cdots, s_n)$. We can estimate this median by using U from the standard SVD.

3.4. R_1-PCA Algorithm

. Here we outline the concrete algorithm

```
R_1- PCA algorithm:
Input: data matrix $X$, the subspace dimension $k$
Initialize:
    compute standard PCA and obtain $U_0$
    compute residue $s_i = \sqrt{\mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^T U_0 U_0^T \mathbf{x}_i}$
    compute $c = median(s_i)$
Set $U = U_0$.
Update $U$ according to Eqs.(23,24)
    iterate until convergence
Compute $V = U^T X$
Compute $\Lambda = U^T C_r U$. Check deviation from diagonal
Output $U, V$
```

Starting from the initial guess $U^{(0)} = U_0$ the algorithm iteratively converges to the optimal solution. At convergence, $U^{(t)}$ converges to the eigenvectors of C_r and the Lagrangian multiplier $\Lambda^{(t)}$ converges to the eigenvalues of C_r. The off-diagonal elements of $\Lambda^{(t)}$ is a measure of the accuracy of the algorithm.

4. Experiments on R_1-PCA

We apply R_1-PCA to a synthetic dataset and four datasets from UCI repository [4].

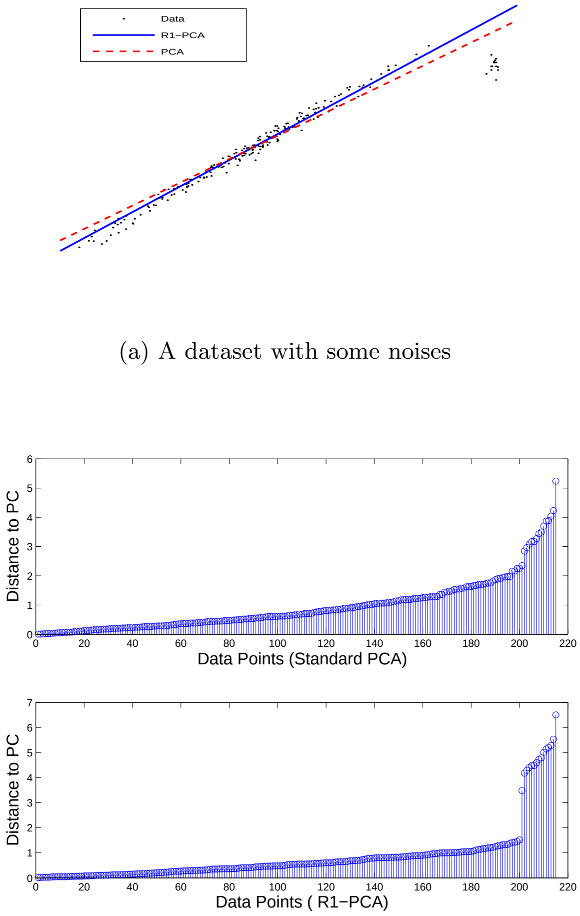
Synthetic dataset. We test the sensitivity of PCA results to the presence of noises. 200 points near a straight line are generated with 12 outliers (see Figure 2). We apply PCA and R_1-PCA to this dataset. The results are shown in Fig. 2(a). We can see that standard PCA is significantly affected by the noises while R_1-PCA is affected much less.

In Fig. 2(b), we plot $\{s_i\}$ (the distances to PCA and R_1-PCA principal subspaces). The horizontal axises are data points in the sorted order. In PCA (top panel of Fig. 2(b)) the noise points are very clearly distinct from the normal points. In R_1-PCA (lower panel of Fig. 2(b)), we see a sharp jump near the 12 rightmost points (index 201-212): outliers become obvious. This indicates R_1-PCA has a capability of detecting outliers.

UCI datasets. We run R_1-PCA on four real world datasets in UCI repository: *glass*, *diabetes*, *mfeat* (hand writing recognition), and *isolet*. A summary of the four datasets is give in Table 4.

Fig. 3 shows the convergence curves of J_r in Eq.(13) (at a stopping threshold 10^{-8}). The algorithm typically converges to the asymptotic limit in 6 iterations.

[4] http://www.ics.uci.edu/∼mlearn/



(a) A dataset with some noises

(b) Distance to principal subspace

Figure 2. PCA vs R_1-PCA on a synthetic dataset

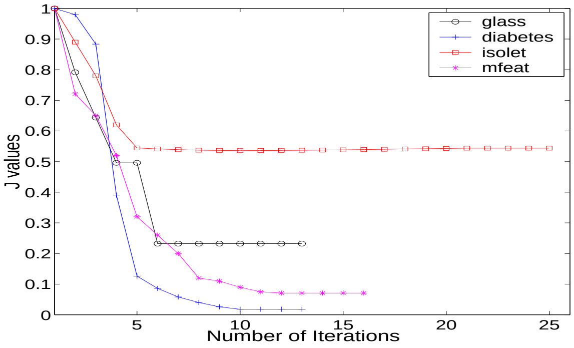


Figure 3. Convergence of J_r towards their asymptotic limits in normalized way.

Table 1. Summary of UCI datasets and PCA / R_1-PCA subspace dimension K.

dataset	# instance	dimensions	# class	K
glass	214	9	6	5
diabetes	768	8	2	5
mfeat	2000	216	10	15
isolet	1559	617	26	15

Next we show the quality of convergences by looking at the initial value of Lagrangian multiplier $\Lambda^{[0]}$ and the converged $\Lambda^{[t]}$ values for R_1-PCA. We list the results for *glass* and *isolet* (the rests are very similar).

$$\Lambda^{[0]}_{glass} = \begin{pmatrix} 2.3993 & 0.0496 & 0.1833 & 0.0578 & -0.0519 \\ 0.0496 & 1.9984 & -0.0072 & 0.0177 & 0.0860 \\ 0.1833 & -0.0072 & 0.7579 & -0.0675 & 0.1204 \\ 0.0578 & 0.0177 & -0.0675 & 1.0250 & -0.0387 \\ -0.0519 & 0.0860 & 0.1204 & -0.0387 & 0.7847 \end{pmatrix}$$

$$\Lambda^{[t=10]}_{glass} = \begin{pmatrix} 2.3889 & -0.0000 & 0.0000 & 0.0000 & 0.0000 \\ -0.0000 & 2.0501 & -0.0000 & -0.0000 & -0.0000 \\ 0.0000 & -0.0000 & 1.0662 & 0.0000 & 0.0000 \\ 0.0000 & -0.0000 & 0.0000 & 0.9204 & 0.0000 \\ 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.6181 \end{pmatrix}$$

$$\Lambda^{[0]}_{isolet} = \begin{pmatrix} 121.6442 & -0.0624 & -0.1388 & -0.0434 & 0.3064 \\ -0.0624 & 52.5410 & -0.1280 & -0.0690 & 0.1302 \\ -0.1388 & -0.1280 & 31.9051 & -0.0282 & 0.0795 \\ -0.0434 & -0.0690 & -0.0282 & 25.8461 & -0.0864 \\ 0.3064 & 0.1302 & 0.0795 & -0.0864 & 24.4433 \end{pmatrix}$$

$$\Lambda^{[t=20]}_{isolet} = \begin{pmatrix} 121.6139 & -0.0001 & 0.0001 & -0.0000 & 0.0000 \\ -0.0001 & 52.5283 & -0.0002 & 0.0001 & -0.0002 \\ 0.0001 & -0.0002 & 31.8918 & -0.0003 & 0.0000 \\ -0.0000 & 0.0001 & -0.0003 & 25.8902 & 0.0186 \\ 0.0000 & -0.0002 & 0.0000 & 0.0186 & 24.4808 \end{pmatrix}$$

Clearly the Lagrangian multipliers converge to the diagonal form. The relative magnitudes of off-diagonal elements reflect the accuracy of the convergence. Because of the convexity of R_1-PCA (Proposition 0), the solutions are well-behaved.

Subspaces. We discuss the computed R_1-PCA subspace (principal directions) $U' = (\mathbf{u}'_1, \cdots, \mathbf{u}'_K)$ and compare to the PCA subspace U=$(\mathbf{u}_1, \cdots, \mathbf{u}_K)$. The inner products (cosine of angels) are given in Table 4 for *glass* and Table 4 for *isolet* (for *isolet* $K = 15$, we show the first 5 dimensions due to space limitation).

We observe that most of the principal directions are different. Some of them have large differences while others have smaller differences.

The *principal angle* $\theta \in [0, \pi/2]$ between two subspace A, B is defined (Golub & Van Loan, 1996) as

$$cos(\theta) = \max_{a \in A} \max_{b \in B} a^T b, \quad s.t.\ ||a|| = ||b|| = 1.$$

This is a generalization of the *angle* between two vectors and characterizes the distance between two subspaces in a comprehensive way. The computed principal angles between PCA-subspace and R_1-PCA subspace are given below.

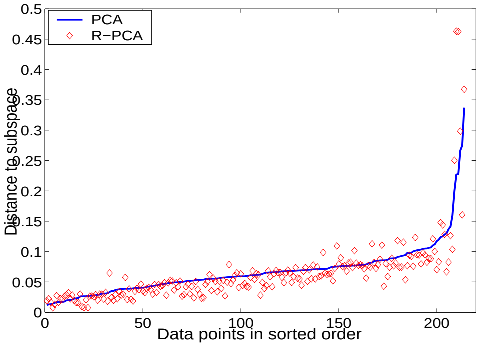


Figure 4. Distance to principal subspace for *glass* data.

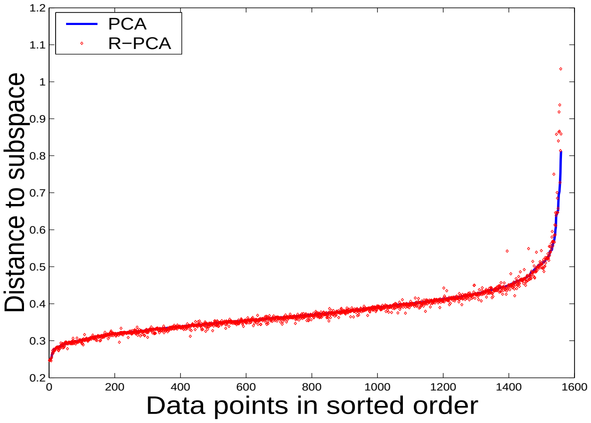


Figure 5. Distance to principal subspace for *isolet* data.

dataset	*glass*	*diabetes*	*mfeat*	*isolet*
θ	0.4539	0.2188	0.1457	0.5918

These results indicate the difference between R_1-PCA and PCA is small for *mfeat* and *diabetes* while the difference is large for *glass* and *isolet*. For this reason, we present the distance-to-subspace results on *glass* and *isolet* in Figures 4 and 5. To save space, we put PCA and R_1-PCA results in one figure, and plot them in the sorted order for PCA results. In both *glass* and *isolet*, we see the same trends as in Figure 2b: a few outlying points move away from the subspace while most stay or move slightly towards the subspace.

K-means on Subspaces. If the R_1-PCA-subspace better captures the data manifold than the PCA-subspace, we hope data clustering on the R_1-PCA-subspace is improved compared to that on the PCA-subspace. Here we compare the clustering accuracy of K-means algorithm on these subspaces. Results for averages over 10 runs are shown in Table 4. The results indicate that R_1-PCA-subspace outperforms PCA-subspace for clustering; K-means in both subspaces improve over the full-space K-means .

Table 2. Inner products between PCA and R_1-PCA principal directions for *glass* data.

$\mathbf{u}_p \cdot \mathbf{u}'_q$	$\mathbf{u}'_1$	$\mathbf{u}'_2$	$\mathbf{u}'_3$	$\mathbf{u}'_4$	$\mathbf{u}'_5$
$\mathbf{u}_1$	0.9873	-0.0679	-0.0848	0.0143	-0.0605
$\mathbf{u}_2$	0.0705	0.9878	0.0130	0.0146	-0.0422
$\mathbf{u}_3$	0.1152	-0.0293	0.7458	0.2829	0.5094
$\mathbf{u}_4$	-0.0464	-0.0311	-0.0868	0.9161	-0.3849
$\mathbf{u}_5$	-0.0120	0.0761	-0.5076	0.2814	0.7477

Table 3. Inner products between PCA and R_1-PCA principal directions for *isolet* data.

$\mathbf{u}_p \cdot \mathbf{u}'_q$	$\mathbf{u}'_1$	$\mathbf{u}'_2$	$\mathbf{u}'_3$	$\mathbf{u}'_4$	$\mathbf{u}'_5$
$\mathbf{u}_1$	0.9999	0.0026	0.0017	0.0013	0.0046
$\mathbf{u}_2$	-0.0027	0.9999	0.0012	0.0084	-0.0016
$\mathbf{u}_3$	-0.0016	-0.0008	0.9969	-0.0649	-0.0271
$\mathbf{u}_4$	-0.0018	-0.0085	0.0670	0.9952	0.0633
$\mathbf{u}_5$	-0.0046	0.0020	0.0238	-0.0661	0.9967

5. Rotational Invariant L_1-norm K-means Clustering (R_1-Kmeans)

We generalize R_1-norm from PCA to K-means , and discuss the L_1-norm K-means . PCA relates to K-means in that the relaxed solution of cluster membership indicators are given by principal components, and the subspace spanned by the cluster centroids are given by the PCA principal subspace (Ding & He, 2004; Zha et al., 2002). We perform experiments and show that L_1-K-means performs poorly compared to standard K-means , while R_1-K-means outperforms standard K-means for the cases where outliers exist.

The K-means clustering minimizes the objective

$$J_{\ell_2} = \sum_{k=1}^{K} \sum_{i \in C_k} ||\mathbf{x}_i - \mu_k||^2, \tag{25}$$

where C_k is the k-th cluster and μ is the centroid. The L_1-norm K-means clustering minimizes

$$J_{\ell_1} = \sum_{k} \sum_{i \in C_k} ||\mathbf{x}_i - \mu_k||_1. \tag{26}$$

The R_1-norm K-means clustering minimizes

$$f_{r_1} = \sum_{k} \sum_{i \in C_k} ||\mathbf{x}_i - \mu_k||. \tag{27}$$

K-means is closely related to the spherical Gaussian

Table 4. Clustering accuracy of K-means on subspaces.

method	**glass**	**diabetes**	**mfeat**	**isolet**
PCA+K-means	0.7043	0.5490	0.9111	0.9480
R_1-PCA +K-means	0.7922	0.5608	0.9438	0.9512
Fullspace+K-means	0.6851	0.5463	0.9088	0.9248

distribution

$$g(\mathbf{x}\,;\sigma,\mu) = \frac{1}{\left[\sqrt{2\pi}\sigma\right]^d}\exp(-\frac{1}{2}\|\frac{\mathbf{x}-\mu}{\sigma}\|^2) \qquad (28)$$

where σ is the standard deviation. L_1-norm K-means relates to the Laplace distribution,

$$f_{\ell_1}(\mathbf{x}\,;\sigma,\mu) = \frac{1}{[4\sigma]^d}\exp(-\frac{1}{2}||\frac{\mathbf{x}-\mu}{\sigma}||_1) \qquad (29)$$

We generalize this to *rotational invariant* Laplace distribution as the underlying distribution for R_1-norm K-means ,

$$f_{r_1}(\mathbf{x}\,;\sigma,\mu) = \frac{1}{v(d)\sigma^d}\exp(-\frac{1}{2}||\frac{\mathbf{x}-\mu}{\sigma}||) \qquad (30)$$

where $v(d) = 2\pi^{d/2}/\Gamma(d/2)$ is the volume of unit sphere.

With these distributions, we can derive the K-means algorithms. They are easily generalized to the Expectation-Maximization (EM) algorithm for the mixture

$$g(\mathbf{x}) = \pi_1 p_1(\mathbf{x}) + \cdots + \pi_K p_K(\mathbf{x}) \qquad (31)$$

where $p_k(\mathbf{x})$ is a one of the above distributions.

In those algorithms, the key is to compute the centroids μ. We compute it by gradient descent and obtain an iterative algorithm:

$$\mu \leftarrow (1-\beta)\mu + \beta\sum_i \frac{\mathbf{x}_i}{||\mathbf{x}_i-\mu||}\Big/\sum_i \frac{1}{||\mathbf{x}_i-\mu||} \qquad (32)$$

The iteration starts with μ as the mean of $\{\mathbf{x}_i\}$. We use $\beta = 0.5$ in all datasets. The convexity of $J(\mu)$ ensures the convergence. Here is the outline.

R_1-K-means algorithm

```
Initialization: centroids $\{\mu_k\}$.
Iterate the following two steps until convergence:
    (E) Re-assign $\{x_i\}$ to closest centers using $L_2$-norm;
    (M) Update centroids $\mu_k$ according to Eq.(32)
```

Experiment. We apply R_1-K-means and L_1-K-means on the widely-used 20-newsgroup dataset. We use five newsgroups: *comp.graphics rec.motorcycles rec.sport.baseball sci.space talk.politics.mideast.* 200 documents are randomly sampled from each newsgroup, with a total of 1000 documents. To simulate the outliers, we randomly pick 80 documents from the rest 15 newsgroups and merge them with 5-newsgroups dataset. The word-document matrix X is constructed with 500 words selected according to the mutual information between words and documents. `tf.idf` term weighting is used. Clustering accuracy are computed using the known class labels. Results of on 5 random samples are given below

K-means	0.618	0.848	0.634	0.770	0.835
L_1-K-means	0.332	0.239	0.286	0.259	0.276
R_1-K-means	0.756	0.846	0.786	0.748	0.869

L_1-K-means performs very poorly; the reason is due to the the assignment of data points to centroids using L_1-norm which defines very skewed of cluster shape. R_1-K-means perform better than standard K-means for this dataset with some outliers.

6. Summary

R_1-PCA is a natural extension of PCA. R_1-PCA solutions are eigenvectors of the R_1-covariance matrix that softens the contributions from outliers. It arises from the optimization of the R_1-norm objective function. R_1-norm is extended to K-means clustering. Experiments show R_1-K-means is a better robust K-means than the L_1-norm K-means .

Acknowledgment. Work partially funded by the Department of Energy (Office of Science) DE-AC03-76SF00098 and the National Science Foundation CCF-0305879.

References

Aanas, H., Fisker, R., Astrm, K., & Carstensen, J. (2002). Robust factorization. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, *24*, 1215 – 1225.

d'Aspremont, A., Ghaoui, L. E., Jordan, M., & Lanckriet, G. (2004). A direct formulation for sparse pca using semidefinite programming. *UC Berkeley Tech Report.*

Ding, C., & He, X. (2004). K-means clustering and principal component analysis. *Int'l Conf. Machine Learning.*

Golub, G., & Van Loan, C. (1996). *Matrix computations, 3rd edition.* Johns Hopkins, Baltimore.

Jolliffe, I. (2002). *Principal component analysis.* Springer. 2nd edition.

Ke, Q., & Kanade, T. (2004). Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming. *IEEE Conf. Computer Vision and Pattern Recognition* (pp. 592–599).

Ng, A. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. *Proc. Int'l Conf. Machine Learning.*

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, *58*, 267–288.

Torre, F. D., & Black, M. J. (2003). A framework for robust subspace learning. *Int'l J. Computer Vision*, 117–142.

Zha, H., Ding, C., Gu, M., He, X., & Simon, H. (2002). Spectral relaxation for K-means clustering. *Advances in Neural Information Processing Systems 14 (NIPS'01)*, 1057–1064.

Zhang, Z., Zha, H., & Simon, H. (2004). Low-rank approximations with sparse factors ii: Penalized methods with discrete newton-like iterations. *SIAM J. Matrix Analysis Applications*, 901–920.

Zou, H., Hastie, T., & Tibshirani, R. (2004). Sparse principal component analysis. *Standard Statistics Tech Report.*