A pure L_1-norm principal component analysis

J.P. Brooks [a,*], J.H. Dulá [b], E.L. Boone [a]

[a] Department of Statistical Sciences and Operations Research, Virginia Commonwealth University, Richmond, VA 23284, United States
[b] Department of Management, Virginia Commonwealth University, Richmond, VA 23284, United States

ARTICLE INFO

Keywords:
Principal component analysis
Linear programming
L_1 regression

ABSTRACT

The L_1 norm has been applied in numerous variations of principal component analysis (PCA). An L_1-norm PCA is an attractive alternative to traditional L_2-based PCA because it can impart robustness in the presence of outliers and is indicated for models where standard Gaussian assumptions about the noise may not apply. Of all the previously-proposed PCA schemes that recast PCA as an optimization problem involving the L_1 norm, none provide globally optimal solutions in polynomial time. This paper proposes an L_1-norm PCA procedure based on the efficient calculation of the optimal solution of the L_1-norm best-fit hyperplane problem. We present a procedure called L_1-PCA* based on the application of this idea that fits data to subspaces of successively smaller dimension. The procedure is implemented and tested on a diverse problem suite. Our tests show that L_1-PCA* is the indicated procedure in the presence of unbalanced outlier contamination.

1. Introduction

Principal component analysis (PCA) is a data analysis technique with various uses including dimensionality reduction, quality control, extraction of interpretable derived variables, and outlier detection (Jolliffe, 2002). Traditional PCA, hereafter referred to as L_2-PCA, is based on the L_2 norm. Principal component analysis using L_2-PCA possesses several important properties such as: the loadings vectors are the eigenvectors of the covariance matrix, the loadings vectors are the successive orthogonal directions of maximum (or minimum) variation in data, and the principal components define the L_2-norm best-fit linear subspaces to the data (Jolliffe, 2002). The simultaneous occurrence of these properties is unique to L_2-PCA, and is a reason why it is widely used. The term "pure" in this paper is used to reflect the fact that the proposed L_1-norm PCA shares an analogous property to L_2-PCA in that the principal components are defined by successive L_1-norm best-fit subspaces.

L_2-PCA is sensitive to outlier observations. This sensitivity is the principal reason for exploring alternative norms. Procedures for PCA that involve the L_1 norm have been developed to increase robustness (Galpin and Hawkins, 1987; Baccini et al., 1996; Ke and Kanade, 2003; Agarwal et al., 2006; Choulakian, 2006; Ding et al., 2006; Gao, 2008; Kwak, 2008). Galpin and Hawkins (1987) develop a robust L_1 covariance estimation procedure. Others have considered robust measures of dispersion for finding directions of maximum variation (Croux and Ruiz-Gazen, 2005; Choulakian, 2006; Kwak, 2008). The approaches in Croux and Ruiz-Gazen (2005) and Choulakian (2006) are based on the projection pursuit method introduced in Li and Chen (1985).

Several previous works involve the L_1 norm in subspace estimation with PCA. Baccini et al. (1996) and Ke and Kanade (2003) consider the problem of finding a subspace such that the sum of L_1 distances of points to the subspace is minimized, and propose heuristic schemes that approximate the subspace. In light of the camera resectioning problem from computer vision, Agarwal et al. (2006) formulate the problem of L_1 projection as a fractional program and give a branch-and-bound algorithm for finding solutions. Gao (2008) proposes a probabilistic Bayesian approach to estimating the best L_1 subspace under the assumption that the noise follows a Laplacian distribution.

Measuring error for PCA using the L_1 norm can impart desirable properties besides providing robustness to outlier observations. For example, the L_1 norm is the indicated measure in a noise model where the error follows a Laplace distribution (Agarwal et al., 2006; Gao, 2008). In special applications such as cellular automata models, where translations can only occur along unit directions, the fit of a subspace can be measured using the L_1 norm (Brooks and Dulá, 2013; Kier et al., 2005).

In this paper, we propose a new approach for robust PCA, L_1-PCA*, based on a "pure" application of the L_1 norm in the sense that it uses globally optimal subspaces that minimize the sum of L_1 distances of points to their projections. L_1-PCA* generates a sequence of subspaces, projecting data down one dimension at a time, in a manner analogous to L_2-PCA. The procedure makes use of a polynomial-time algorithm for projecting m-dimensional data into an L_1-norm best-fit $(m-1)$-dimensional subspace. The algorithm is based on results concerning properties of L_1 projection and best-fit hyperplanes (Späth and Watson, 1987; Martini and Schöbel, 1998; Mangasarian, 1999; Brooks and Dulá, 2013). The provable optimality of the projected subspace ensures that interesting properties are inherited. The polynomiality of the algorithm makes it practical. We establish experimentally conditions under which the proposed procedure is preferred over L_2-PCA and other previously-investigated schemes for robust PCA.

2. L_1 regression, geometry of the L_1 norm, and best-fit subspaces

Linear regression models seek to find a hyperplane of the form

$$y = \beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}$$

for a dependent variable y and independent variables $\boldsymbol{x}$. In L_2 regression, the hyperplane of best fit is determined for a given dataset $(y_i, \boldsymbol{x}_i)$, $i = 1, \ldots, n$ by minimizing the residual sum of squared errors given by

$$\sum_{i=1}^{n} (y_i - (\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}_i))^2.$$

In L_2 regression, the distance of points to the fitted hyperplane is given by the L_2 norm. L_1 linear regression is analogous to L_2 regression in that they both find a hyperplane that minimizes the sum of distances of points to their projections along the unit direction defined by the dependent variable. In the case of L_1 regression, the distances are measured using the L_1 norm. The sum of absolute errors is minimized. A standard result about L_1 regression is that the hyperplane can be found by solving a linear program (LP) (Charnes et al., 1955; Wagner, 1959). The L_1 regression LP is as follows.

$$\min_{\beta_0, \boldsymbol{\beta}, \boldsymbol{e}^+, \boldsymbol{e}^-} \sum_{i=1}^{n} e_i^+ + e_i^-, \tag{R}$$

subject to

$$\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}_i + e_i^+ - e_i^- = y_i, \quad i = 1, \ldots, n,$$
$$e_i^+, e_i^- \geq 0, \quad i = 1, \ldots, n.$$

The input data are $\boldsymbol{x}_i \in \mathbb{R}^m$, $i = 1, \ldots, n$ which are rows of the data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times m}$, and the dependent variable values y_i, $i = 1, \ldots, n$. The vector $\boldsymbol{\beta}$ in (R) are the regression coefficients and the variable β_0 is the level value that are to be determined by the optimal solution to the LP. The variables e_i^+ and e_i^- are the errors measured as the distance from either side of the hyperplane to observation i. Note that at most one of each pair e_i^+, e_i^- is positive at optimality, and the sum of absolute errors is $\sum_{i=1}^{n} (e_i^+ + e_i^-)$.

Given an $m \times q$ matrix $\boldsymbol{A}$, the internal (linear combination of points) representation of a subspace defined by $\boldsymbol{A}$ is the column space of $\boldsymbol{A}$, given by $\mathcal{C}(\boldsymbol{A}) = \{\boldsymbol{z} \in \mathbb{R}^m | \boldsymbol{A}\boldsymbol{\lambda} = \boldsymbol{z} \text{ for all } \boldsymbol{\lambda} \in \mathbb{R}^q\}$. The external (intersection of hyperplanes) representation is $\mathcal{C}'(\boldsymbol{A}) = \{\boldsymbol{z} \in \mathbb{R}^m | \boldsymbol{A}^T \boldsymbol{z} = 0\}$.

The projection of a point $\boldsymbol{x} \in \mathbb{R}^m$ on a set S is the set of points $P \subseteq S$ such that the distance between $\boldsymbol{x}$ and points in S is minimized. Distance for PCA is usually measured using the L_2 norm. The L_2 norm of a vector $\boldsymbol{x}$ is

$$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{m} x_i^2}.$$

When S is an affine set, the L_2 projection of $\boldsymbol{x}$ is a unique point and the direction from P to $\boldsymbol{x}$ is orthogonal to S.

The L_1 norm of a vector $\boldsymbol{x}$ is

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{m} |x_i|.$$

Using the L_1 norm for measuring distance results in different projections. Fig. 1 illustrates this difference for the case when $\boldsymbol{x} \in \mathbb{R}^2$ and S is a line. The figure represents the level sets using the L_1 and L_2 norms. The L_2 norm level sets in two

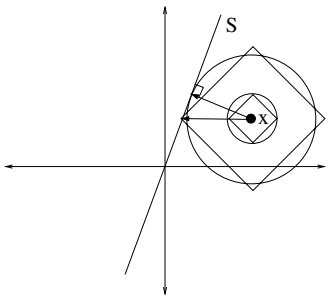


Fig. 1. Level sets for the L_1 and L_2 norms.

dimensions are circles; the L_1 norm level sets are diamonds. Notice that the L_1 projection occurs along a horizontal direction. The property that projections occur along a single unit direction generalizes to multiple dimensions (Mangasarian, 1999; Brooks and Dulá, 2013). Further, the direction of a projection depends only on the orientation of the hyperplane and not on the location of the point (Mangasarian, 1999). These two properties lead to the following result about *L_1-norm best-fit* hyperplanes; that is, hyperplanes for which the sum of L_1 distances of points to their projections is minimized.

Lemma 1. *Given a set of points $\boldsymbol{x}_i \in \mathbb{R}^m$, $i = 1, \ldots, n$, the projections into an L_1-norm best-fit $(m-1)$-dimensional hyperplane occur along the same unit direction for all of the points.*

Proof. See Mangasarian (1999). □

Lemma 1 implies that an L_1-norm best-fit hyperplane is found by computing the m hyperplanes that minimize the sum of absolute errors along each of the m dimensions and selecting the hyperplane with the smallest sum of absolute errors. Identifying the hyperplane that minimizes the sum of absolute errors along a given dimension is the L_1 linear regression problem presented above where the dependent variable corresponds to the dimension along which measurements are made. This discussion is the proof of the following Proposition 1.

Proposition 1. *A best-fit hyperplane in $\mathbb{R}^m$ is found by computing m L_1 linear regressions, where each variable takes a turn as the dependent variable, and selecting the regression hyperplane with the smallest sum of absolute errors.*

Recall that a hyperplane containing the origin is an $(m-1)$-dimensional subspace. PCA assumes that data are centered around the mean and fits subspaces accordingly. The analogy for the L_1 measure is that the data are centered around the median and that the fitted hyperplanes contain the origin. This assumption will be applied later in numerical experiments.

The L_1-norm best-fit $(m-1)$-dimensional subspace $\mathcal{C}'(\boldsymbol{\beta}^*)$ can be found by the following procedure.

Algorithm for finding a L_1-norm best-fit subspace of dimension $m-1$.

Given a data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ with full column rank.

Set $j^* = 0$, $R_0(\boldsymbol{X}) = \infty$. /* Initialization. */

for $(j = 1; j \le m; j = j + 1)$ **do**

  Solve $R_j(\boldsymbol{X}) = \min_{\boldsymbol{\beta}, \boldsymbol{e}^+, \boldsymbol{e}^-} \sum_{i=1}^{n} e_i^+ + e_i^-$, /*Find the L_1 regression with variable j as the dependent variable.*/

  subject to

$$\begin{aligned} \boldsymbol{\beta}^T \boldsymbol{x}_i + e_i^+ - e_i^- &= 0, \quad i = 1, \ldots, n, \\ \beta_j &= -1, \\ e_i^+, e_i^- &\ge 0, \quad i = 1, \ldots, n. \end{aligned}$$

  if if $R_j(\boldsymbol{X}) < R_{j^*}(\boldsymbol{X})$, /* if the fitted subspace for variable j is better than that for j^{**}/

  then

    $j^* = j$, $\boldsymbol{\beta}^* = \boldsymbol{\beta}$. /* Update the coefficients defining the best fit subspace */

  end if

end for

The procedure finds the L_1-norm regression subspace ($\beta_0 = 0$) where each variable in turn serves as the response, as enforced by the constraint $\beta_j = -1$. In Brooks and Dulá (2013), results concerning L_1 projection on hyperplanes using the L_1 norm are proved and an LP-based algorithm for finding the L_1-norm best-fit hyperplane is presented. This paper adapts the algorithm in Brooks and Dulá (2013) as a subroutine in the design of L_1-PCA*, explains correspondences between L_1-PCA* and traditional PCA, and demonstrates the effectiveness of the procedure on simulated and real-world datasets.

The fitted subspace inherits the following well-known properties (Appa and Smith, 1973; Baccini et al., 1996; Agarwal et al., 2006):

1. At least $(m-1)$ of the points lie in the fitted subspace;

2. The subspace corresponds to a maximum likelihood estimate for a fixed effect model with noise following a joint distribution of m independent, identically distributed Laplace random variables.

In the development below, the normal vector of a best-fit $(m-1)$-dimensional subspace for points in an m-dimensional space is given by $\boldsymbol{\beta}^m$. We will employ the notation $(\boldsymbol{I}^{j^*})^m$ to denote the $m \times m$ identity matrix modified such that row j^* has entries

$$(\boldsymbol{I}^{j^*}_{j^*\ell})^m = \beta^m_\ell / \|\boldsymbol{\beta}^m\|_2 \quad \ell \neq j^*,$$
$$(\boldsymbol{I}^{j^*}_{j^*j^*})^m = 0.$$

In the next section, we apply these results to develop the new PCA procedure based on the optimality of the fitted subspaces.

3. The L_1-PCA* algorithm

Proposition 1 and the procedure for finding the L_1-norm best-fit subspace motivate algorithm L_1-PCA* where points are iteratively projected down from the initial space $\mathbb{R}^m$ of the data, to an $(m-1)$-dimensional subspace, then to an $(m-2)$-dimensional subspace, and so on.

The algorithm takes as input a data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ and generates a sequence of subspaces, each one dimension less than the previous one, defined by their orthogonal vectors $\boldsymbol{\alpha}^k \in \mathbb{R}^m$, $k = m, m-1, \ldots, 1$. The projection into the best $(k-1)$-dimensional subspace is determined by applying the algorithm for finding the L_1-norm best-fit subspace by finding the best of k L_1 regressions. The $(k-1)$-dimensional subspace has an external representation given by $\mathcal{C}'(\boldsymbol{\beta}^k)$. The vector $\boldsymbol{\beta}^k \in \mathbb{R}^k$ is the optimal value of $\boldsymbol{\beta}$ returned by the algorithm above. The corresponding vector $\boldsymbol{\alpha}^k$ is the normalized representation of $\boldsymbol{\beta}^k$ in the original m-dimensional space and is the kth principal component loadings vector.

Each subspace is determined by its normal vector $\boldsymbol{\beta}^k$. Applying $\boldsymbol{\beta}^k$ to the current data matrix $\boldsymbol{X}^k \in \mathbb{R}^{n\times k}$ produces the projections in the $(k-1)$-dimensional subspace. An internal representation of the subspace, needed for the next iteration, requires a set of spanning vectors of the space containing the projections. The spanning vectors form the columns of the projection matrix $\boldsymbol{V}^k \in \mathbb{R}^{k\times(k-1)}$. Obtaining an internal representation can be done in any number of ways. Algorithm L_1-PCA* uses the singular value decomposition (SVD) of the projected points $\boldsymbol{Z}^k \in \mathbb{R}^{n\times k}$. The product of the $(m-1)$ projection matrices is the vector of loadings for the first principal component $\boldsymbol{\alpha}^1$.

The pseudocode for the L_1-PCA* algorithm is given next.

Algorithm L_1-PCA*

Given a data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ with full column rank.

1: Set $\boldsymbol{X}^m = \boldsymbol{X}$; set $\boldsymbol{V}^{m+1} = \boldsymbol{I}$; set $(\boldsymbol{I}^{j^*})^{m+1} = \boldsymbol{I}$. /* Initialization. */
2: **for** $(k = m; k > 1; k = k-1)$ **do**
3: Set $j^* = \text{argmin}_j R_j(\boldsymbol{X}^k)$ and $\boldsymbol{\beta}^k = \boldsymbol{\beta}^*$ using the algorithm for finding a L_1-norm best-fit subspace of dimension $m-1$. /* Find the best-fitting L_1 subspace among subspaces derived with each variable j as the dependent variable. */
4: Set $\boldsymbol{Z}^k = (\boldsymbol{X}^k)((\boldsymbol{I}^{j^*})^k)^T$. /* Project points into a $(k-1)$-dimensional subspace. */
5: Calculate the SVD of $\boldsymbol{Z}^k$, $\boldsymbol{Z}^k = \boldsymbol{U}\Lambda\boldsymbol{V}^T$, and set $\boldsymbol{V}^k$ to be equal to the $(k-1)$ columns of $\boldsymbol{V}$ corresponding to the largest values in the diagonal matrix Λ. /* Find a basis for the $(k-1)$-dimensional subspace. */
6: Set $\boldsymbol{\alpha}^k = \left(\prod_{\ell=m+1}^{k+1} \boldsymbol{V}^\ell\right)\boldsymbol{\beta}^k / \|\boldsymbol{\beta}^k\|_2$. /* Calculate the k^{th} principal component. */
7: Set $\boldsymbol{X}^{(k-1)} = \boldsymbol{Z}^k\boldsymbol{V}^k$. /* Calculate the projected points in terms of the new basis. */
8: **end for**
9: Set $\boldsymbol{\alpha}^1 = \prod_{\ell=m+1}^{2} \boldsymbol{V}^\ell$. /* Calculate the first principal component. */

Notes on L_1-PCA*

1. The algorithm generates a sequence of matrices $\boldsymbol{X}^m, \ldots, \boldsymbol{X}^1$. Each of these matrices contains n rows, each row corresponding to a point. All of the points in $\boldsymbol{X}^k$ are in a k-dimensional space. The normal vectors of the successive subspaces are mutually orthogonal.
2. The sequence of vectors $\boldsymbol{\alpha}^k$ in Step 6 represent the principal component loadings vectors. The vector $\boldsymbol{\alpha}^k$ is orthogonal to the subspace $\mathcal{C}(\boldsymbol{V}^k)$.
3. Any $(k-1)$ vectors spanning the projected points can form the columns of $\boldsymbol{V}^k$. In this respect, the algorithm is indeterminate. Different choices for this set will lead to different projections at successive iterations because the L_1 norm is not *rotationally invariant* (Ding et al., 2006). One way to make the algorithm determinate is to always use singular value decomposition to define a new coordinate system as in Step 5.

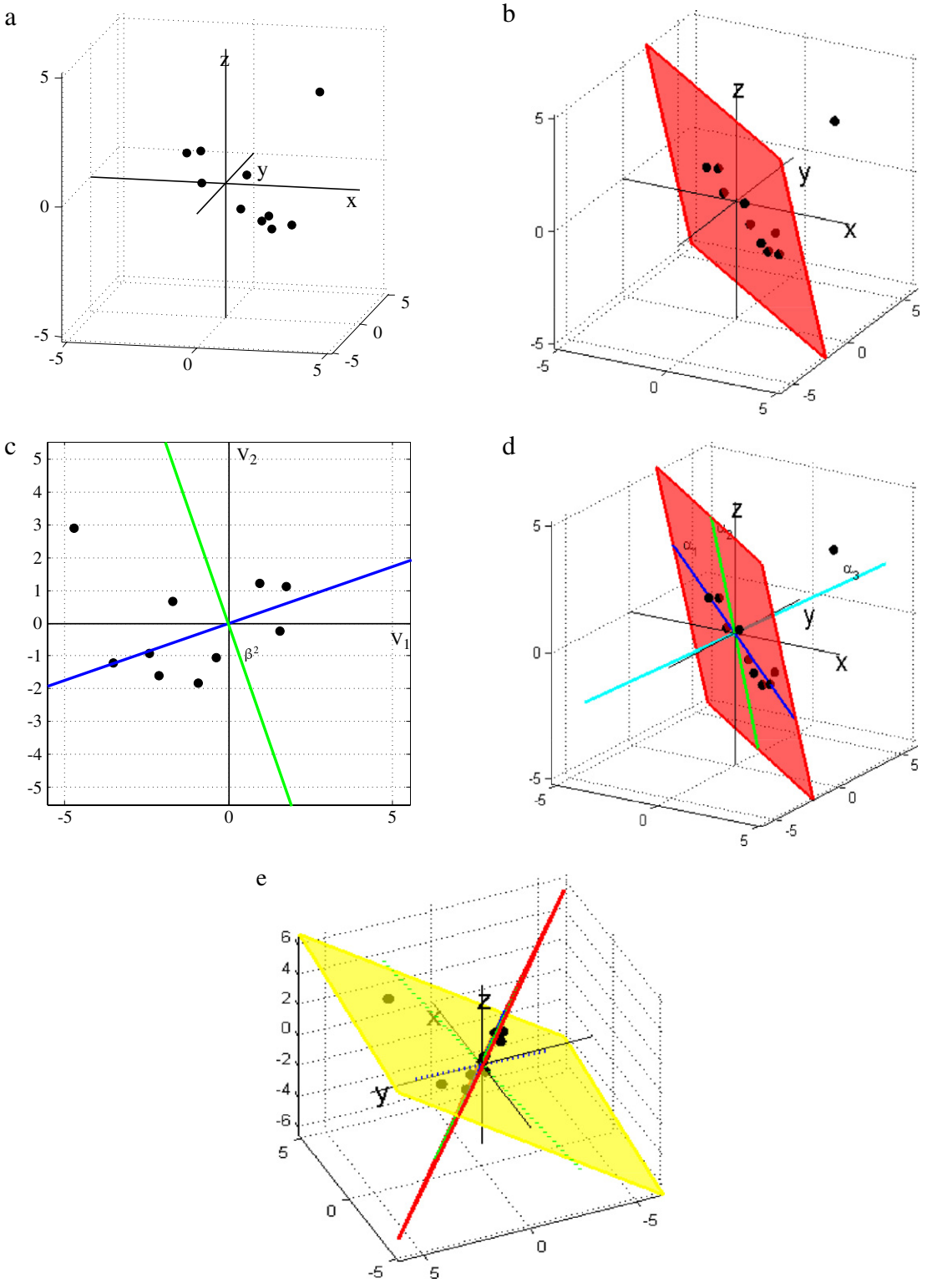


Fig. 2. L_1-PCA* implementation for a 3-dimensional example. (a) The point set in 3-D. (b) The L_1 best-fit plane. (c) The projection in the best-fit plane with best-fit line. (d) L_1-PCA* results. (e) Comparison of L_1-PCA* versus L_2-PCA.

4. The solution of linear programs is the most computationally-intensive step in each iteration. A total of $\sum_{k=1}^{m} k = \frac{m(m+1)}{2}$ linear programs are solved. Each linear program has $2n + k$ variables and n constraints. The algorithm has a worst-case

Table 1
Correspondences between L_1-PCA* and L_2-PCA for estimating the k-dimensional best-fit subspace.

	Concept	Formula
1	kth principal component loadings vector $\boldsymbol{\alpha}^k$, $k = 2, \ldots, m$ (set $\boldsymbol{\alpha}^1$ orthogonal to $\boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^m$)	$\left(\prod_{\ell=m+1}^{k+1} \boldsymbol{V}^\ell\right) \frac{\boldsymbol{\beta}^k}{\left\|\boldsymbol{\beta}^k\right\|_2}$
2	Score of observation i (from Step 4 of algorithm L_1-PCA*)	$\boldsymbol{x}_i^k$
3	Projection of point $\boldsymbol{x}_i$ for observation i (in terms of original coordinates)	$\left(\prod_{\ell=m+1}^{k+1} \boldsymbol{V}^\ell\right) \boldsymbol{x}_i^k$
4	Score of a new point $\boldsymbol{x}_{n+1}$	$\left(\prod_{\ell=k+1}^{m+1} (\boldsymbol{V}^\ell)^T (\boldsymbol{I}^{j^*})^\ell\right) \boldsymbol{x}_{n+1}$
5	Projection of a new point $\boldsymbol{x}_{n+1}$ (in terms of original coordinates)	$\left(\prod_{r=m+1}^{k+1} \boldsymbol{V}^r\right) \left(\prod_{\ell=k+1}^{m+1} (\boldsymbol{V}^\ell)^T (\boldsymbol{I}^{j^*})^\ell\right) \boldsymbol{x}_{n+1}$

running time of

$$O\left(\sum_{k=1}^{m} k\mathcal{P}(2n+k, n)\right),$$

where $\mathcal{P}(r, s)$ is the complexity of solving a linear program with r variables and s constraints. Since the complexity of linear programming is polynomial, the complexity of L_1-PCA* is polynomial.

5. The algorithm produces an L_1 best-fit subspace at each iteration. Accordingly, this procedure performs well for outlier-contaminated data so long as the accrued effect of the L_1 distances of the outliers does not force the optimal solution to fit them directly.

The following three-dimensional example will help to illustrate L_1-PCA*. Consider the data matrix

$$(\boldsymbol{X}^3)^T = \begin{bmatrix} -1.17 & 0.53 & -1.02 & 1.12 & 2.08 & -1.61 & 1.17 & 2.00 & 3.00 & 3.00 \\ 1.20 & 0.24 & 0.40 & 1.36 & -1.82 & 0.53 & -1.52 & -1.03 & -2.00 & 3.00 \\ -0.30 & -1.00 & 1.11 & -1.69 & -0.76 & 0.99 & 0.71 & -1.44 & -1.00 & 3.00 \end{bmatrix}.$$

These points are displayed in Fig. 2(a). The L_1 best-fit plane obtained in Step 3 by applying Proposition 1 and after solving three LPs is defined by $\boldsymbol{\beta}^3\boldsymbol{x} = 0$ where $\boldsymbol{\beta}^3 = (-0.80, -1.00, -0.39)$. The plane can be see in Fig. 2(b). This plane minimizes the total sum of L_1 distances of the points to their projections at 9.75. Notice that all projections occur along $j^* = 2$ (y-axis). The projected points $\boldsymbol{Z}^3$ from Steps 4 and 5 are

$$(\boldsymbol{Z}^3)^T = \begin{bmatrix} -1.17 & 0.53 & -1.02 & 1.12 & 2.08 & -1.61 & 1.17 & 2.00 & 3.00 & 3.00 \\ 1.05 & -0.03 & 0.38 & -0.22 & -1.36 & 0.90 & -1.21 & -1.03 & -2.00 & -3.58 \\ -0.30 & -1.00 & 1.11 & -1.69 & -0.76 & 0.99 & 0.71 & -1.44 & -1.00 & 3.00 \end{bmatrix}.$$

Notice that only the second component of each point changes. The singular value decomposition of $\boldsymbol{Z}^3$ is $\boldsymbol{U}\Lambda\boldsymbol{V}^T$, where

$$\boldsymbol{U}^T = \begin{bmatrix} -0.212 & 0.051 & -0.130 & 0.124 & 0.325 & -0.237 & 0.228 & 0.285 & 0.474 & 0.633 \\ 0.052 & 0.232 & -0.264 & 0.397 & 0.200 & -0.245 & -0.144 & 0.352 & 0.266 & -0.633 \\ 0.858 & -0.267 & -0.021 & 0.123 & -0.092 & 0.200 & -0.084 & 0.198 & 0.225 & 0.175 \end{bmatrix},$$

$$\Lambda = \begin{bmatrix} 7.46 & 0.00 & 0.00 \\ 0.00 & 4.59 & 0.00 \\ 0.00 & 0.00 & 0.00 \end{bmatrix}, \qquad \boldsymbol{V} = \begin{bmatrix} 0.77 & 0.22 & 0.59 \\ -0.63 & 0.20 & 0.75 \\ 0.05 & -0.95 & 0.29 \end{bmatrix}.$$

The first two columns of $\boldsymbol{V}$ span the plane and comprise the 3 by 2 matrix $\boldsymbol{V}^3$. The third principal component loadings vector is $\boldsymbol{\alpha}^3 = (-0.59, -0.75, -0.29)$ using the formula in Step 6. Notice that $\boldsymbol{\alpha}^3$ is orthogonal to the best-fit subspace, and that for this first iteration, $\boldsymbol{\alpha}^3$ points in the same direction as $\boldsymbol{\beta}^3$. The next iteration will use the dataset $\boldsymbol{X}^2 = \boldsymbol{Z}^3\boldsymbol{V}^3$ which is calculated in Step 7 and corresponds to a re-orientation of the axes so that the plane becomes the two-dimensional space where the data resides:

$$(\boldsymbol{X}^2)^T = \begin{bmatrix} -1.58 & 0.38 & -0.97 & 0.92 & 2.43 & -1.77 & 1.70 & 2.13 & 3.54 & 4.73 \\ 0.24 & 1.07 & -1.21 & 1.82 & 0.92 & -1.13 & -0.66 & 1.61 & 1.22 & -2.91 \end{bmatrix}.$$

Fig. 2(c) is a plot of the data points in the matrix $\boldsymbol{X}^2$, embedded in the two-dimensional best-fit subspace. The figure also shows the resulting best-fit subspace found in the next iteration in terms of the new axes. Fig. 2(c) illustrates how our method remains insensitive to outliers despite the use of SVD as a method for determining the basis for projected points at each iteration. The choice of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ by using SVD is adversely affected by the outlier; however, L_1-PCA* overcomes the poor choice of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ and rotates the axes so that the blue and green lines are the principal components. Traditional PCA would identify $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ as the principal component loadings vectors for the projected points. The line in the $\boldsymbol{v}_1$–$\boldsymbol{v}_2$ plane perpendicular to the line labeled $\boldsymbol{\beta}^2$ is the L_1 best-fit subspace for the two-dimensional problem solved in the next iteration and represents the projection of the first principal component in this plane. Notice that the presence of the outlier

Table 2
Calculated values for numerical example using formulas from Table 1.

$(\boldsymbol{x}_{n+1} = (-2.0, 3.0, 1.0))$		
Formula 1	$k = 3$	$\boldsymbol{\alpha}^3 = (-0.59, -0.75, -0.29)$
Principal	$k = 2$	$\boldsymbol{\alpha}^2 = (0.04, -0.40, 0.92)$
Components	$k = 1$	$\boldsymbol{\alpha}^1 = (0.80, -0.53, -0.27)$
Formula 2		
Scores	$k = 3$	$(\boldsymbol{X}^3)^T = \begin{bmatrix} -1.17 & 0.53 & -1.02 & 1.12 & 2.08 & -1.61 & 1.17 & 2.00 & 3.00 & 3.00 \\ 1.20 & 0.24 & 0.40 & 1.36 & -1.82 & 0.53 & -1.52 & -1.03 & -2.00 & 3.00 \\ -0.30 & -1.00 & 1.11 & -1.69 & -0.76 & 0.99 & 0.71 & -1.44 & -1.00 & 3.00 \end{bmatrix}$
	$k = 2$	$(\boldsymbol{X}^2)^T = \begin{bmatrix} -1.58 & 0.38 & -0.97 & 0.92 & 2.43 & -1.77 & 1.70 & 2.13 & 3.54 & 4.73 \\ 0.24 & 1.07 & -1.21 & 1.82 & 0.92 & -1.13 & -0.66 & 1.61 & 1.22 & -2.91 \end{bmatrix}$
	$k = 1$	$(\boldsymbol{X}^1)^T = \begin{bmatrix} -1.67 & 0.40 & -1.03 & 0.98 & 2.57 & -1.87 & 1.80 & 2.25 & 3.74 & 5.00 \end{bmatrix}$
Formula 3		
Projections	$k = 3$	$\begin{bmatrix} -1.17 & 0.53 & -1.02 & 1.12 & 2.08 & -1.61 & 1.17 & 2.00 & 3.00 & 3.00 \\ 1.20 & 0.24 & 0.40 & 1.36 & -1.82 & 0.53 & -1.52 & -1.03 & -2.00 & 3.00 \\ -0.30 & -1.00 & 1.11 & -1.69 & -0.76 & 0.99 & 0.71 & -1.44 & -1.00 & 3.00 \end{bmatrix}$
	$k = 2$	$\begin{bmatrix} -1.17 & 0.53 & -1.02 & 1.12 & 2.08 & -1.61 & 1.17 & 2.00 & 3.00 & 3.00 \\ 1.05 & -0.03 & 0.38 & -0.22 & -1.36 & 0.90 & -1.21 & -1.03 & -2.00 & -3.58 \\ -0.30 & -1.00 & 1.11 & -1.69 & -0.76 & 0.99 & 0.71 & -1.44 & -1.00 & 3.00 \end{bmatrix}$
	$k = 1$	$\begin{bmatrix} -1.34 & 0.32 & -0.83 & 0.78 & 2.06 & -1.50 & 1.45 & 1.81 & 3.00 & 4.01 \\ 0.89 & -0.22 & 0.55 & -0.52 & -1.37 & 1.00 & -0.96 & -1.20 & -2.00 & -2.67 \\ 0.45 & -0.11 & 0.28 & -0.26 & -0.69 & 0.50 & -0.48 & -0.60 & -1.00 & -1.34 \end{bmatrix}$
Formula 4	$k = 3$	$(-2.00, 3.00, 1.00)$
Scores	$k = 2$	$(-2.26, -1.16)$
New point	$k = 1$	-2.39
Formula 5	$k = 3$	$(-2.00, 3.00, 1.00)$
Projections	$k = 2$	$(-2.00, 1.20, 1.00)$
New point	$k = 1$	$(-1.92, 1.28, 0.64)$

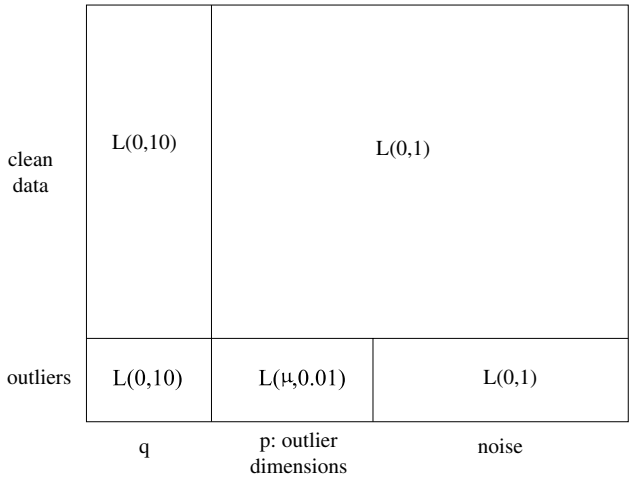
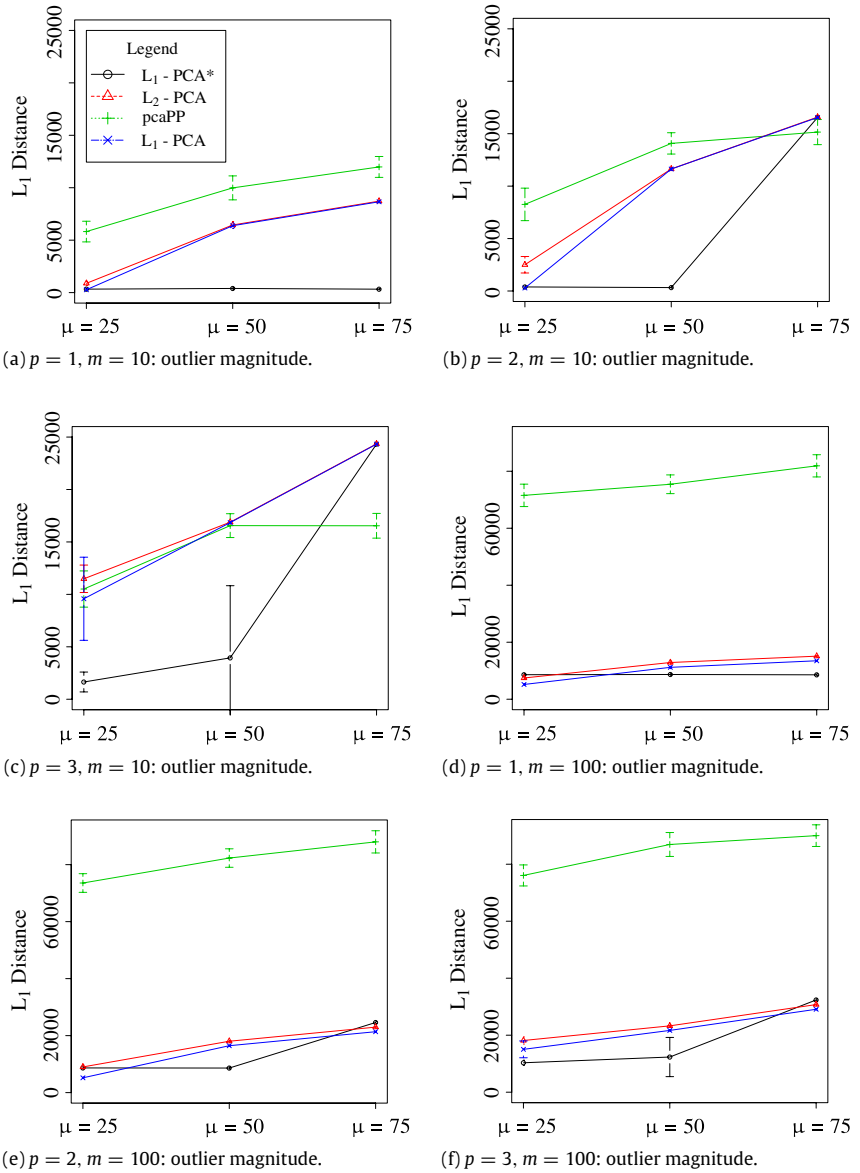


Fig. 3. Dataset design for simulation experiments. Observations are represented by rows and variables are represented by columns. For each instance, outliers comprise 10% of the dataset.

in the second quadrant does not significantly affect the fit. The line labeled $\boldsymbol{\beta}^2$ is the projection of the second L_1 principal component in this plane.

Fig. 2(d) depicts the two best-fit subspaces and the three principal component axes in terms of the original coordinates of the data. The two subspaces are the red plane and the line labeled $\boldsymbol{\alpha}_1$. The three principal component axes are the lines labeled $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$, and $\boldsymbol{\alpha}_3$.

Fig. 2(e) compares the plane defined by the first two principal component axes from L_1-PCA* to that derived using L_2-PCA. The view is oriented so that the plane from L_1-PCA* is orthogonal to the plane of the page. Most of the data points are near the plane given by L_1-PCA* indicating a good fit despite the presence of the outlier $\boldsymbol{x}_{10} = (3.0, 3.0, 3.0)$. The outlier has clearly affected the location of the plane derived using L_2-PCA and does not fit the rest of the data well. This comparison between L_1- and L_2-based methods for PCA is formalized in the experiments below.



(a) $p = 1$, $m = 10$: outlier magnitude.

(b) $p = 2$, $m = 10$: outlier magnitude.

(c) $p = 3$, $m = 10$: outlier magnitude.

(d) $p = 1$, $m = 100$: outlier magnitude.

(e) $p = 2$, $m = 100$: outlier magnitude.

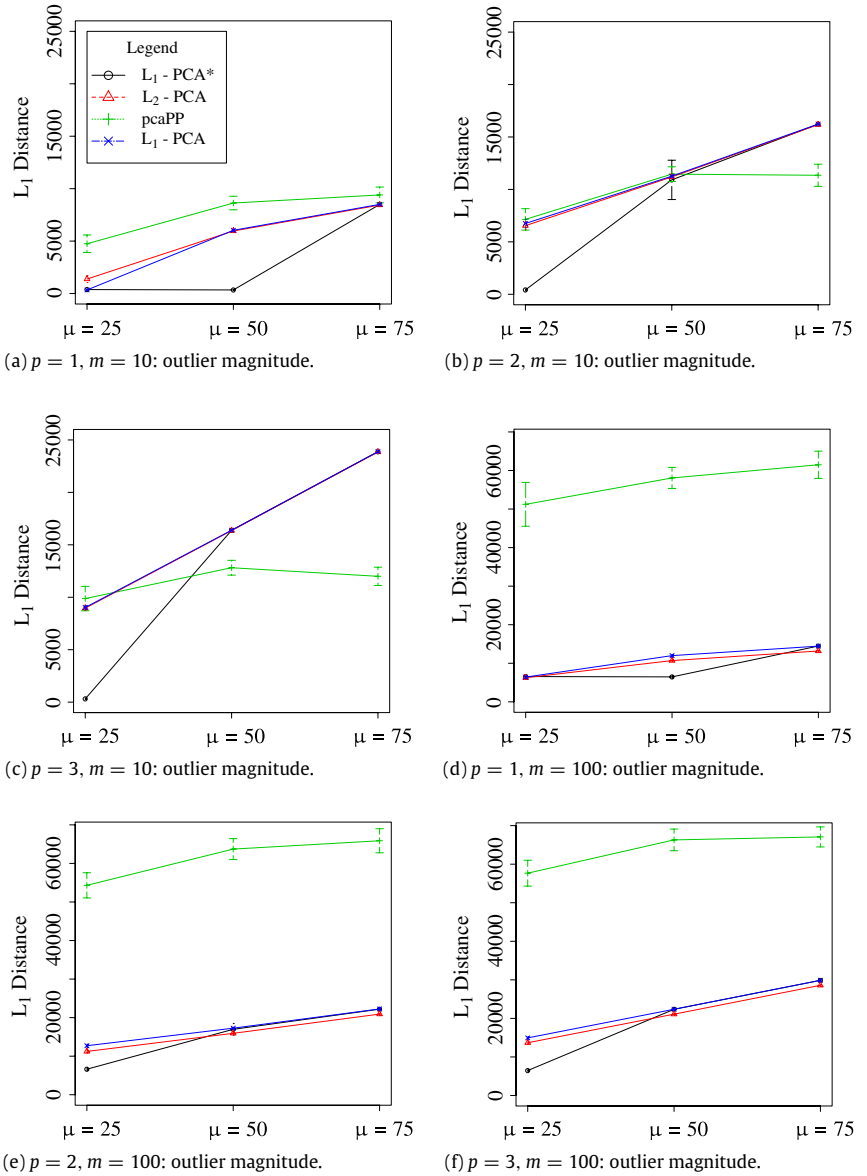
(f) $p = 3$, $m = 100$: outlier magnitude.

Fig. 4. *Laplacian noise.* The sum of errors, the sum of L_1 distances of projected points in a 5-dimensional subspace to the "true" 5-dimensional subspace of the data, versus outlier magnitude with Laplacian noise, for dimensions $m = 10$ and $m = 100$, and $p = 1, 2, 3$. The average sum of errors over 100 iterations is plotted. Error bars represent one standard deviation. The parameter p is the number of outlier-contaminated dimensions.

4. Correspondences between L_1-PCA* and L_2-PCA

When algorithm L_1-PCA* is applied to a dataset, values analogous to those used in an application of L_2-PCA are obtained. These values permit an analysis with all the functionality of L_2-PCA. Table 1 collects these results along with their explicit formulas in L_1-PCA*. We include correspondences for principal components and scores, and projections of new points into fitted subspaces. Below we explain how to obtain these correspondences. These are summarized in Table 1 and their results using the numerical example are in Table 2.

Extracting principal components and scores. As algorithm L_1-PCA* iteratively projects points into lower-dimensional subspaces, we can collect the normal vectors as principal component loadings vectors. The vector that is orthogonal to projected points is unique at each iteration k. This vector is precisely $\boldsymbol{\beta}^k$. Also, when the singular value decomposition of the projected points is calculated in Step 5, $\boldsymbol{Z}^k = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^T$, the principal component loadings vector $\boldsymbol{\beta}^k/\|\boldsymbol{\beta}^k\|$ is the column of $\boldsymbol{V}$ corresponding to the smallest value in the diagonal matrix $\boldsymbol{\Lambda}$. This direction defined by $\boldsymbol{\beta}^k$ is a k-dimensional vector in the current subspace. Formula 1 in Table 1 presents $\boldsymbol{\alpha}^k$, the kth L_1 principal component loadings vector in terms of the original m-dimensional space.



(a) $p = 1$, $m = 10$: outlier magnitude. (b) $p = 2$, $m = 10$: outlier magnitude.

(c) $p = 3$, $m = 10$: outlier magnitude. (d) $p = 1$, $m = 100$: outlier magnitude.

(e) $p = 2$, $m = 100$: outlier magnitude. (f) $p = 3$, $m = 100$: outlier magnitude.

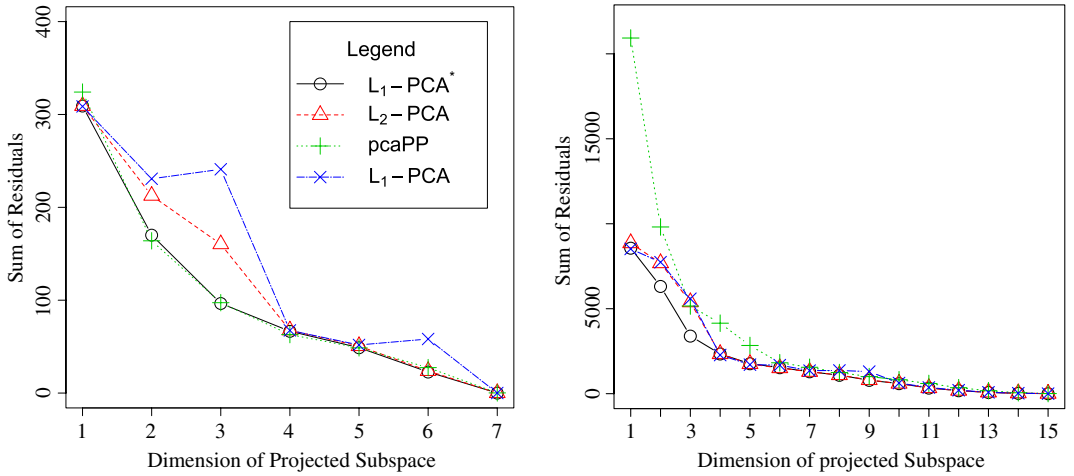
Fig. 5. *Gaussian noise.* The sum of errors, the sum of L_1 distances of projected points in a 5-dimensional subspace to the "true" 5-dimensional subspace of the data, versus outlier magnitude with Gaussian noise, for dimensions $m = 10$ and $m = 100$, and $p = 1, 2, 3$. The average sum of errors over 100 iterations is plotted. Error bars represent one standard deviation. The parameter p is the number of outlier-contaminated dimensions.

The rows of the matrix $\boldsymbol{X}^k$, $\boldsymbol{x}_i^k$ in Table 1, are the principal component scores. These are the projected points in the projected coordinate system. For an observation i, the projection into the k-dimensional subspace in terms of the original coordinates is calculated using Formula 3.

Projecting new points. The principal component loadings obtained with Formula 1 of Table 1 define the subspace(s) into which observations are projected. In L_2-PCA, the matrix the columns of which are the first k principal component loadings vectors is the rotation matrix and is used to project points into the k-dimensional fitted subspace. The projection of a point using L_1-PCA* depends on the sequence of intermediate subspaces so that the matrix of principal component loadings vectors should not be used as a projection matrix. L_1-PCA* projects optimally one dimension at a time in a unit direction that may not coincide with the normal vector $\boldsymbol{\beta}^k$.

For a new point $\boldsymbol{x}_{n+1}$, the projection into the best-fit $(m-1)$-dimensional subspace is given by $(\boldsymbol{I}^{j^*})^m \boldsymbol{x}_{n+1}$. The projected point in terms of the original coordinates is given by $\boldsymbol{V}^m (\boldsymbol{I}^{j^*})^m \boldsymbol{x}_{n+1}$. To project the point into a k-dimensional subspace, use Formulas 4 and 5.

Table 2 applies the formulas from Table 1 to the numerical example from the previous section for three subspaces: (1) for $k = 3$, the space $\mathbb{R}^3$ where the original data reside, (2) for $k = 2$, the fitted plane in Fig. 2(b), and (3) for $k = 1$, the line labeled $\boldsymbol{\alpha}^1$ in Fig. 2(d).



(a) Milk data. (b) McDonald and Schwing data.

Fig. 6. Sum of residuals of non-outlier observations versus the dimension of the fitted subspace for (a) the Milk dataset and (b) the McDonald and Schwing dataset.

5. Computational results

L_1-PCA* is implemented and its performance on simulated and real-world data is compared to L_2-PCA, pcaPP, and L_1-PCA. The R package pcaPP is a publicly-available implementation of the L_1-norm-based PCA procedure developed by Croux and Ruiz-Gazen (2005). The approach implemented in pcaPP maximizes an L_1 measure of dispersion in the data to find successive locally-optimal directions of maximum dispersion. We implemented L_1-PCA, an algorithm developed by Ke and Kanade (2003) that approximates L_1-norm best-fit subspaces directly which stands in contrast to the successive approaches L_1-PCA* and pcaPP.

L_1-PCA* is implemented in a C program that uses ILOG CPLEX 11.1 Callable Library (ILOG, 2009) for the solution of the linear programs required for Step 3. The singular value decomposition in Step 5 is calculated using the function `dgesvd` in LAPACK (Anderson et al., 1999). The L_2-PCA and pcaPP implementations are publicly available in the `stats` (R Development Core Team, 2008) and `pcaPP` (Filzmozer et al., 2009) packages for the R language for statistical computing (R Development Core Team, 2008). The function used for L_2-PCA is `prcomp`. L_1-PCA is implemented as part of an R package that will be released at a later date (Jot, 2011). All experiments are performed on machines with 2.6 GHz Opteron processors and at least 4 GB RAM.

Tests with simulated data. The implementations are tested on simulated data. Simulated data provide a controlled setting for comparison of data analysis algorithms and reveal trends that help make generalizable conclusions. The objectives for the data are to explore the impact of outliers on the procedures by varying the dimensionality and magnitude of outlier contamination. The data are generated such that a predetermined subspace contains most of the dispersion. The dimension of this "true" subspace is varied to assess the dependence on this data characteristic.

Each dataset consists of $n = 1000$ observations with m dimensions. The first q dimensions define the predetermined subspace and the remaining dimensions contain noise. The first q dimensions are sampled from a Laplace(0, 10) distribution; the remaining dimensions are sampled from a Laplace(0, 1) distribution. Outliers are introduced in the dataset by generating additional points where the first q dimensions are sampled from a Laplace(0, 10) distribution, the next p dimensions are sampled from a Laplace(μ, 0.01) distribution so that the outliers are on the same side of the true subspace, and the remaining $m - p - q$ dimensions are sampled from a Laplace(0, 1) distribution. The problem suite also includes control datasets ($p = 0$, $\mu = 0$) without outlier observations. Outlier observations comprise ten percent of each dataset. We refer to the parameter p as the number of outlier-contaminated dimensions and the parameter μ as the outlier magnitude. Datasets are also generated by replacing distribution Laplace(0, 10) with $N(0, 10)$, replacing Laplace(0, 1) with $N(0, 1)$, and replacing Laplace(μ, 0.01) with $N(\mu, 0.01)$. Fig. 3 is a schematic of the roles of the data components in the simulated data.

Tests are conducted for the configurations that result when $m = 10, 100$; $q = 2, 5$; $p = 1, 2, 3$; and $\mu = 25, 50, 75$; in addition to the control datasets; for a total of 60 configurations. We define the error for an observation as the L_1 distance from its projected point in the best-fitting q-dimensional subspace determined by the four methods to the predetermined q-dimensional subspace.

The problem suite is processed by the L_1-PCA*, L_2-PCA, pcaPP, and L_1-PCA implementations. The results for Laplacian noise are reported in Tables 3–4 and Figs. 4 and 7 and the results for Gaussian noise are reported in Tables 5–6 and Figs. 5 and 8. These tables contain the mean and standard deviation of the sum of errors for 100 replications of each configuration.

Our experiments include eight controls ($m = 10, 100$, $q = 2, 5$, Laplacian and Gaussian noise) when $\mu = 0$ and $p = 0$ for each distribution of noise. In these datasets, there are no outliers. L_2-PCA outperforms L_1-PCA* in seven of the eight con-

trol experiments. As expected, L_2-PCA outperforms L_1-PCA* in the control experiments when the noise is Gaussian. In the presence of Laplacian noise, L_1-PCA* performs better than L_2-PCA when the dimension of the underlying subspace is higher ($q = 5$) and the dimension of the original data is lower ($m = 10$). The explanation is that the fitted subspaces of successively smaller dimension derived by L_1-PCA* are optimal with respect to the projected data at each iteration and do not necessarily coincide with the L_1 best-fit subspace with respect to the original data points. Therefore, there is a degradation in the performance of L_1-PCA* as the dimension of the true subspace decreases. L_1-PCA approximates subspaces directly, and performs best in the control experiments in the presence of Laplacian noise. For the control experiments, pcaPP is not competitive.

Fig. 4 illustrates the results for $q = 5$ and Laplacian noise. The figure compares the performance of the four implementations with respect to outlier magnitude μ, number of outlier-contaminated dimensions p, and number of total dimensions m. For small contamination ($\mu = 25$ and $p = 1$), there is little discernible difference among L_1-PCA*, L_1-PCA, and L_2-PCA, and these three methods outperform pcaPP. For moderate levels of contamination ($\mu = 50$), L_1-PCA* provides a clear advantage in the presence of outliers. The other methods fit the outlier observations, while L_1-PCA* ignores them and fits the clean data. In the cases with extreme outlier contamination ($\mu = 75$ and $p \geq 2$), all of the methods break down and fit the outlier observations with the exception of pcaPP when $m = 10$, $\mu = 75$, and $p = 3$. This advantage for pcaPP for $\mu = 75$ and $p = 3$ is not present when $m = 100$. This adverse reaction of pcaPP to an increase in dimension can be explained by a dimensionality curse effect since the algorithm relies on a grid search procedure. Similar patterns are observed in the experiments when the noise are Gaussian, except that the increased noise in the data causes each method to break down sooner (Fig. 5).

For each method, as μ and p are increased, the breakdown point is reached where the methods begin to fit the outlier observations better than the non-contaminated data. For Laplacian noise and for $p = 1, 2, 3$, L_1-PCA* does not break down, even as μ is increased to 50, while significant increases in the sum of errors are seen for L_2-PCA, pcaPP, and L_1-PCA. The approach of the breakdown point for L_1-PCA* and L_2-PCA as p and μ increase is signaled in Tables 3–6 by an increase in the standard deviations of the sum of errors. For the configurations with large standard deviations; such as $m = 10, q = 2, p = 3$, and $\mu = 50$ for L_1-PCA*; the methods fit the non-contaminated data for some samples and fit the outlier observations for other samples which result in drastically different sums of errors. The standard deviations of the sums of errors for pcaPP are larger than those for the other methods for almost every configuration, an indication that pcaPP is sensitive to small changes in the data. For $q = 5$ and Laplacian noise, we can see in Fig. 4 that L_1-PCA* is less susceptible to breakdown and only breaks down when $p \geq 2$ and $\mu = 75$. In summary, L_1-PCA* is competitive with other methods in the presence of low outlier contamination and provides substantial benefits in the presence of moderate outlier contamination. Our experiments show that L_1-PCA* is the indicated procedure in the presence of unbalanced outlier contamination. These experiments validate the intuition that L_1-PCA* is robust to outliers because of the underlying reliance on optimally-fitted L_1 subspaces.

Tests with real data. The four implementations; L_1-PCA*, L_2-PCA, pcaPP, and L_1-PCA; are applied to real-world data that are known to contain outliers. The "Milk" dataset is introduced by Daudin et al. (1988) and used by Choulakian (2006) for tests with an L_1 projection-pursuit algorithm for PCA. The "McDonald and Schwing" dataset is introduced by McDonald and Schwing (1973) and is used by Croux and Ruiz-Gazen (2005) for tests with an L_1 projection-pursuit algorithm for PCA. For each dataset, the data are centered by subtracting the attribute medians.

Fig. 6 contains plots of the sum of residuals of non-outlier observations against the dimension of the fitted subspace for the two datasets. The residual for an observation is measured as the L_1 distance from the original point to the projected point in the fitted subspace. If a method is properly ignoring the outlier observations and fitting the non-contaminated data, then the plotted values should be small.

Fig. 6(a) contains the results for the Milk dataset. Observations 17, 47, and 70 are identified as outliers in previous analyses (see Choulakian (2006)). When the outliers are removed the correlation of variables 4 and 6, 4 and 7, 5 and 6, and 5 and 7 increase by more than 0.3 each when outliers are removed, indicating that the outlier contamination is present in more than one dimension. The sum of residuals for the non-outlier observations for L_1-PCA* and pcaPP are almost identical and are less than that for L_2-PCA and L_1-PCA for two and three dimensions.

The residuals for the non-outliers in the McDonald and Schwing dataset are depicted in Fig. 6(b). Observations 29 and 48 are identified as outliers in previous analyses (see Croux and Ruiz-Gazen (2005)). When the outliers are removed, the correlation of variables 5 and 12, 5 and 13, 12 and 15, 13 and 14, 13 and 16 increase by at least 0.3 each, indicating that the outlier contamination is present in more than one dimension. The sum of residuals for non-outlier observations for pcaPP is larger for smaller-dimensional fitted subspaces when compared to L_1-PCA*, L_2-PCA, and L_1-PCA. Procedure pcaPP appears to be pushing the fitted subspace away from the outlier observations at the expense of the fit of non-outlier observations.

The expectation that L_1-PCA* performs well in the presence of outliers is validated for these datasets. For the two real datasets, Fig. 6 shows that L_1-PCA* generates subspaces that are competitive in terms of having low residuals for non-outliers. The diminished advantage of using L_1-PCA* for these real datasets when compared to the simulated data can be attributed to the fact that the outliers in the real data are more balanced in that they are not located on one side of the best-fitting subspace for the non-contaminated data. Real data with outlier contamination are likely to have some imbalance in the location of the outliers. L_1-PCA* is well-suited in these situations.

6. Conclusions

This paper proposes a new procedure for PCA based on finding successive L_1-norm best-fit subspaces. The result is a pure L_1 PCA procedure in the sense that it uses the globally optimal solution of an optimization problem where the sum of L_1

distances is minimized. Several "pure" L_1 PCAs are possible by taking any one of the properties of traditional L_2 PCA and using the L_1 norm. We describe a complete PCA procedure, L_1-PCA*, based on this result, and include formulas for correspondences with familiar L_2-PCA outputs. The procedure is tested on simulated and real data. The results for simulated data indicate that the procedure can be more robust than L_2-PCA and competing L_1-based procedures, pcaPP and L_1-PCA, for unbalanced outlier contamination and for a wide range of outlier magnitudes. Experiments with real data confirm that L_1-PCA* is competitive in datasets with outliers. L_1-PCA* represents an alternative tool for the numerous applications of PCA.

Acknowledgments

The first author was supported in part by NIH-NIAID awards UH2AI083263-01 and UH3AI08326-01, and NASA award NNX09AR44A. The authors would also like to acknowledge the Center for High Performance Computing at VCU for providing computational infrastructure and support.

Appendix A

Table 3
Average (standard deviation) of L_1 distance to true subspace for $m = 10$ Laplacian noise/error 100 replications for each configuration.

q	p	μ	L_1-PCA*	L_2-PCA	pcaPP	L_1-PCA
2	0	0	339.8 (66.0)	329.1 (60.2)	4600.2 (978.6)	244.5 (46.2)
2	1	25	328.5 (60.0)	640.7 (185.9)	5556.0 (1216.1)	253.1 (48.7)
2	2	25	371.5 (73.7)	1515.7 (681.4)	7456.1 (1711.5)	253.9 (47.1)
2	3	25	872.4 (427.8)	8974.9 (3298.1)	9145.1 (1852.6)	5520.1 (4965.3)
2	1	50	358.4 (75.6)	6521.5 (393.8)	8085.5 (1045.0)	6211.4 (248.6)
2	2	50	337.4 (60.4)	11,637.9 (111.2)	10,010.6 (1091.2)	11,611.9 (107.5)
2	3	50	3644.8 (6662.5)	16,912.9 (80.8)	11,097.6 (1090.0)	16,850.1 (63.8)
2	1	75	329.3 (61.8)	8698.4 (88.5)	8835.1 (940.1)	8606.3 (58.0)
2	2	75	16,599.0 (75.1)	16,584.4 (71.7)	9940.4 (1130.2)	16,538.7 (75.5)
2	3	75	24,372.2 (61.9)	24,371.8 (68.2)	10,288.8 (1297.9)	24,329.6 (56.6)
5	0	0	347.5 (51.8)	370.4 (56.0)	4555.6 (716.1)	273.3 (39.4)
5	1	25	330.1 (50.2)	884.1 (191.1)	5819.6 (986.1)	271.3 (40.9)
5	2	25	386.3 (72.1)	2503.5 (785.3)	8258.3 (1549.5)	286.0 (44.6)
5	3	25	1644.9 (947.7)	11,485.5 (1308.1)	10,510.0 (1727.2)	9581.7 (3963.4)
5	1	50	395.5 (75.8)	6447.3 (308.8)	9985.7 (1142.8)	6379.1 (269.2)
5	2	50	326.1 (52.5)	11,636.4 (129.0)	14,075.7 (1017.2)	11,629.6 (110.4)
5	3	50	3944.9 (6881.0)	16,882.8 (65.8)	16,555.9 (1136.5)	16,845.3 (78.1)
5	1	75	325.3 (55.1)	8719.7 (76.3)	11,983.1 (996.1)	8659.6 (85.6)
5	2	75	16,554.3 (60.0)	16,584.5 (63.3)	15,153.7 (1198.3)	16,546.4 (67.5)
5	3	75	24,330.9 (64.1)	24,351.5 (56.6)	16,541.6 (1179.0)	24,327.3 (59.0)

q: dimension of "true" underlying subspace.
p: number of outlier-contaminated dimensions.
μ: outlier magnitude.

Table 4
Average (standard deviation) of L_1 distance to true subspace for $m = 100$ Laplacian noise/error 100 replications for each configuration.

q	p	μ	L_1-PCA*	L_2-PCA	pcaPP	L_1-PCA
2	0	0	5012.0 (294.1)	4054.7 (223.0)	42,876.7 (2733.3)	2965.2 (167.5)
2	1	25	5079.1 (306.8)	4379.2 (276.9)	43,127.7 (2088.2)	2991.8 (164.3)
2	2	25	5055.7 (272.7)	5210.7 (754.1)	45,059.6 (2644.3)	3001.6 (161.7)
2	3	25	6069.4 (727.1)	13,567.6 (2811.4)	46,975.9 (2926.5)	8774.4 (4563.1)
2	1	50	5127.4 (295.5)	9842.0 (500.7)	46,602.8 (2507.1)	8628.0 (453.7)
2	2	50	5052.1 (305.0)	14,850.5 (237.8)	49,793.2 (2478.7)	13,998.4 (339.3)
2	3	50	6668.7 (4793.3)	20,106.7 (217.6)	52,352.1 (2922.9)	19,140.3 (144.6)
2	1	75	5036.1 (260.9)	11,868.0 (213.0)	49,244.6 (2666.6)	10,923.4 (165.8)
2	2	75	20,649.4 (262.5)	19,790.3 (217.9)	52,503.2 (2703.2)	18,913.6 (155.2)
2	3	75	28,369.1 (262.8)	27,549.7 (211.9)	54,090.3 (3307.9)	26,720.3 (436.8)
5	0	0	8590.3 (278.1)	6933.6 (202.9)	70,283.2 (3071.5)	5182.5 (187.5)
5	1	25	8603.1 (285.1)	7451.2 (326.7)	71,539.1 (3933.4)	5178.2 (171.3)
5	2	25	8663.4 (314.3)	9014.1 (712.9)	73,553.8 (3275.1)	5194.2 (213.9)

(*continued on next page*)

Table 4 (*continued*)

q	p	μ	L_1-PCA*	L_2-PCA	pcaPP	L_1-PCA
5	3	25	10,326.2 (922.5)	18,144.2 (1251.5)	76,066.9 (3690.2)	15,035.6 (2951.7)
5	1	50	8682.2 (291.9)	12,879.5 (439.0)	75,432.8 (3286.9)	11,186.4 (420.0)
5	2	50	8623.1 (306.7)	18,030.7 (251.3)	82,312.4 (3241.5)	16,459.9 (242.5)
5	3	50	12,340.6 (6879.0)	23,276.8 (257.7)	86,927.0 (4196.8)	21,677.5 (231.0)
5	1	75	8564.1 (287.2)	15,107.4 (213.0)	81,899.0 (3897.0)	13,488.4 (192.7)
5	2	75	24,608.9 (292.8)	22,979.6 (216.2)	87,993.7 (3899.2)	21,373.5 (203.4)
5	3	75	32,371.0 (299.4)	30,743.9 (242.8)	90,029.9 (3800.5)	29,097.8 (194.9)

q: dimension of "true" underlying subspace.
p: number of outlier-contaminated dimensions.
μ: outlier magnitude.

Table 5
Average (standard deviation) of L_1 distance to true subspace for $m = 10$ Gaussian noise/error 100 replications for each configuration.

q	p	μ	L_1-PCA*	L_2-PCA	pcaPP	L_1-PCA
2	0	0	313.8 (61.6)	253.1 (50.6)	3312.2 (637.9)	323.9 (64.9)
2	1	25	342.2 (64.0)	921.6 (408.7)	4104.5 (870.0)	323.4 (58.4)
2	2	25	368.2 (75.4)	6540.2 (373.4)	5663.8 (1261.7)	6675.4 (347.8)
2	3	25	315.2 (56.2)	9014.0 (161.8)	7188.9 (1276.1)	9019.2 (157.6)
2	1	50	327.5 (66.1)	5945.0 (60.8)	6342.9 (923.5)	5979.1 (68.3)
2	2	50	4569.8 (5343.7)	11,188.4 (53.3)	7725.0 (887.7)	11,262.7 (72.0)
2	3	50	16,428.7 (55.0)	16,400.3 (48.2)	8622.2 (991.6)	16,433.2 (54.6)
2	1	75	8464.7 (55.3)	8420.7 (43.4)	6804.2 (847.4)	8470.5 (64.4)
2	2	75	16,217.4 (58.2)	16,181.2 (50.1)	7650.8 (774.2)	16,230.9 (60.9)
2	3	75	23,914.2 (54.6)	23,886.2 (46.1)	7743.1 (837.0)	23,923.6 (51.1)
5	0	0	342.9 (43.3)	272.7 (37.2)	3323.9 (525.1)	337.0 (46.9)
5	1	25	378.1 (53.4)	1381.1 (337.0)	4747.7 (837.0)	340.8 (51.2)
5	2	25	416.6 (72.6)	6544.6 (390.3)	7142.7 (1021.1)	6751.3 (402.9)
5	3	25	309.7 (48.3)	8961.2 (166.0)	9876.1 (1164.5)	9023.1 (155.9)
5	1	50	340.6 (56.7)	5963.8 (59.9)	8629.7 (645.3)	6030.6 (79.1)
5	2	50	10,909.0 (1875.4)	11,189.1 (46.9)	11,456.6 (699.8)	11,262.6 (58.8)
5	3	50	16,383.4 (49.1)	16,357.3 (37.6)	12,814.8 (705.6)	16,392.5 (53.4)
5	1	75	8495.9 (48.2)	8438.6 (40.6)	9399.0 (755.7)	8501.7 (52.6)
5	2	75	16,221.3 (49.0)	16,178.6 (39.4)	11,348.0 (1056.0)	16,229.8 (47.0)
5	3	75	23,878.3 (38.5)	23,854.2 (37.3)	11,995.8 (871.1)	23,884.5 (45.6)

q: dimension of "true" underlying subspace.
p: number of outlier-contaminated dimensions.
μ: outlier magnitude.

Table 6
Average (standard deviation) of L_1 distance to true subspace for $m = 100$ Gaussian noise/error 100 replications for each configuration.

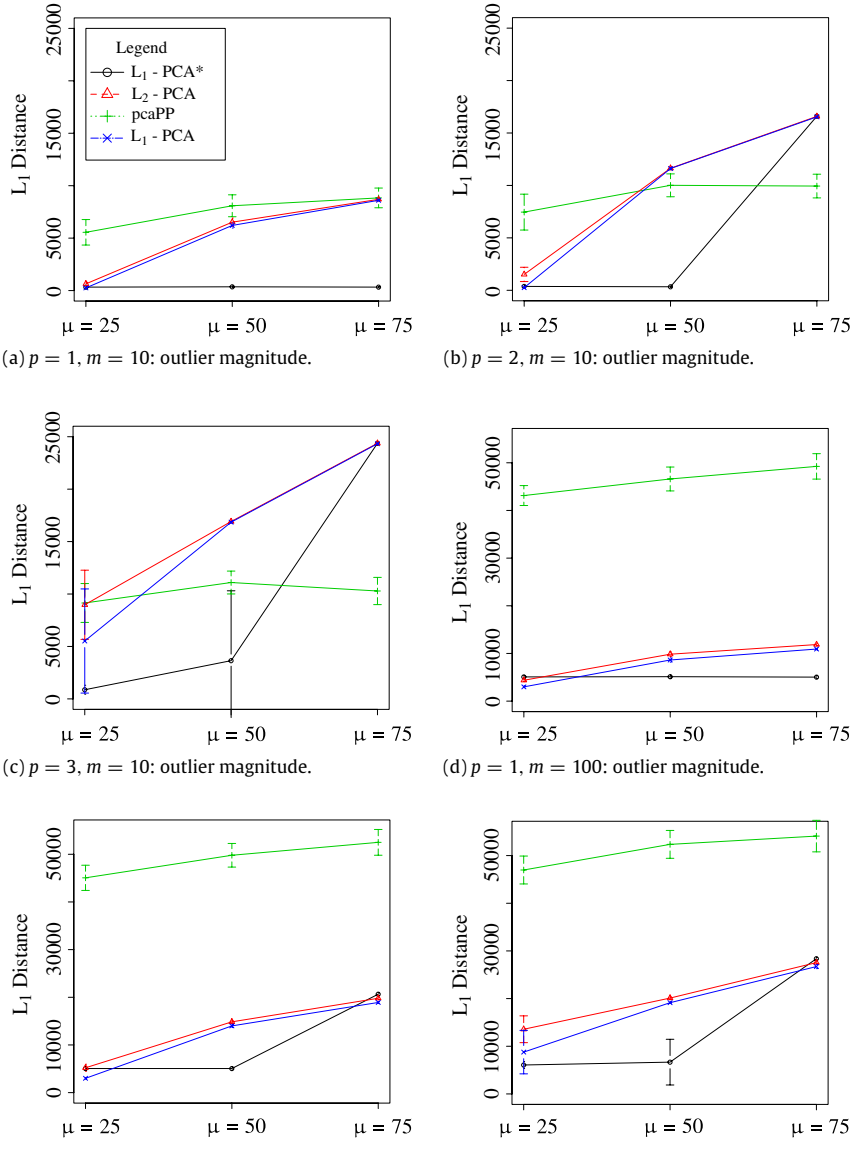
q	p	μ	L_1-PCA*	L_2-PCA	pcaPP	L_1-PCA
2	0	0	3941.4 (242.1)	3130.7 (171.0)	30,524.2 (1520.4)	3873.4 (222.0)
2	1	25	3995.1 (221.7)	3841.3 (404.3)	31,296.6 (1782.1)	3919.1 (205.5)
2	2	25	4007.9 (236.1)	8910.4 (421.6)	33,394.3 (2204.3)	9657.8 (342.9)
2	3	25	3969.2 (229.6)	11,418.2 (228.3)	35,798.6 (2353.9)	11,982.3 (244.1)
2	1	50	3948.1 (208.7)	8348.5 (145.6)	35,591.2 (2142.0)	8964.0 (184.5)
2	2	50	7585.4 (4947.6)	13,599.2 (155.6)	38,132.4 (2254.6)	14,294.2 (174.5)
2	3	50	19,447.2 (190.7)	18,787.3 (148.9)	39,397.2 (2142.2)	19,504.4 (545.8)
2	1	75	11,435.0 (780.0)	10,826.2 (144.5)	37,157.3 (2191.3)	11,465.6 (209.1)
2	2	75	19,260.1 (197.1)	18,565.4 (135.0)	39,185.0 (2395.3)	19,254.7 (202.4)
2	3	75	26,936.7 (169.2)	26,268.0 (149.6)	39,902.9 (2575.0)	26,891.1 (178.7)
5	0	0	6513.4 (215.0)	5153.2 (184.8)	49,849.0 (3663.3)	6399.8 (215.9)
5	1	25	6556.8 (224.8)	6252.6 (417.5)	51,223.5 (5684.2)	6408.6 (208.2)
5	2	25	6624.2 (247.0)	11,230.7 (358.5)	54,312.2 (3270.9)	12,712.2 (399.8)
5	3	25	6474.8 (197.4)	13,683.9 (217.7)	57,664.4 (3355.7)	14,944.9 (270.4)
5	1	50	6465.6 (240.3)	10,696.2 (188.0)	58,066.3 (2745.5)	11,983.2 (310.2)
5	2	50	16,971.2 (1523.5)	15,931.6 (159.6)	63,716.0 (2714.6)	17,261.5 (201.8)

(*continued on next page*)

Table 6 (*continued*)

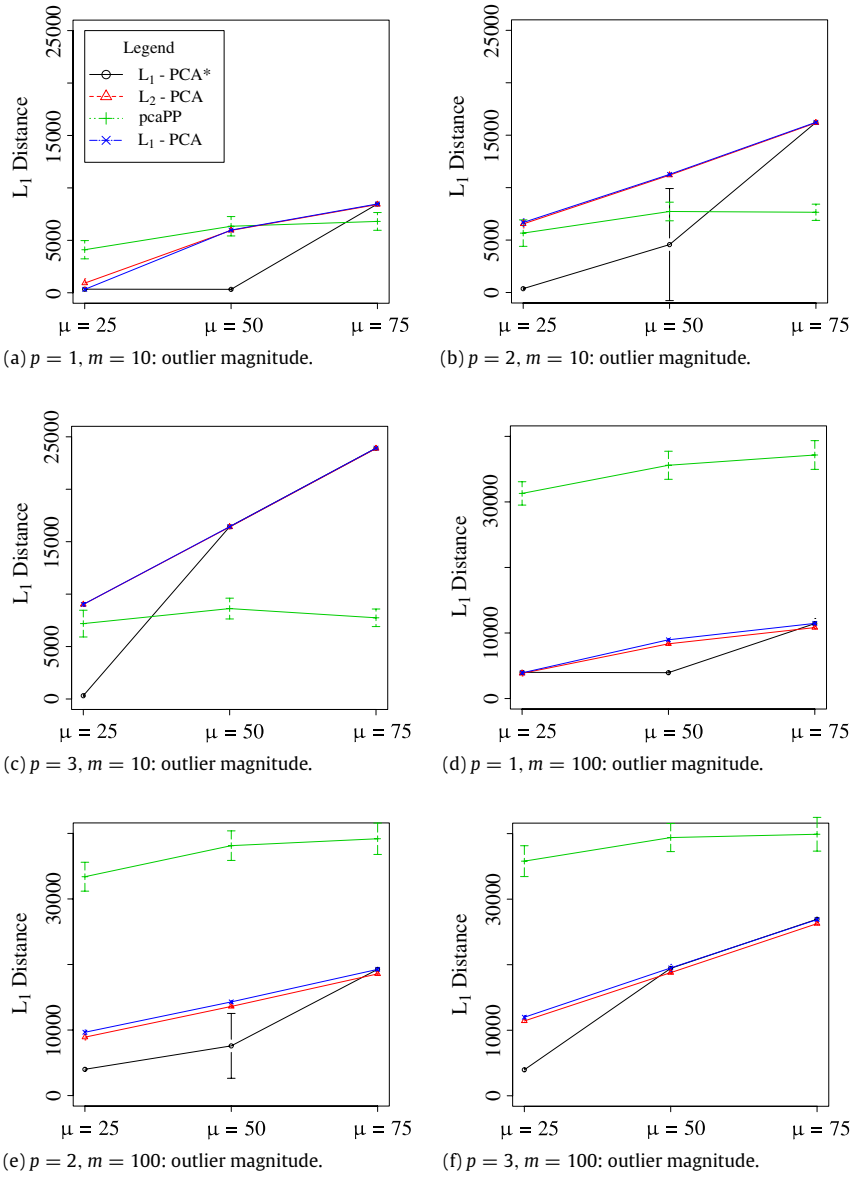
q	p	μ	L_1-PCA*	L_2-PCA	pcaPP	L_1-PCA
5	3	50	22,360.3 (227.1)	21,095.7 (161.0)	66,299.2 (2812.8)	22,370.1 (231.8)
5	1	75	14,467.8 (219.5)	13,184.2 (174.3)	61,481.8 (3534.9)	14,444.7 (259.4)
5	2	75	22,192.6 (233.2)	20,917.8 (174.6)	65,874.3 (3126.6)	22,254.4 (316.3)
5	3	75	29,864.4 (226.1)	28,586.3 (177.4)	67,072.6 (2599.8)	29,877.5 (226.1)

q: dimension of "true" underlying subspace.
p: number of outlier-contaminated dimensions.
μ: outlier magnitude.



(a) $p = 1$, $m = 10$: outlier magnitude.
(b) $p = 2$, $m = 10$: outlier magnitude.
(c) $p = 3$, $m = 10$: outlier magnitude.
(d) $p = 1$, $m = 100$: outlier magnitude.
(e) $p = 2$, $m = 100$: outlier magnitude.
(f) $p = 3$, $m = 100$: outlier magnitude.

Fig. 7. *Laplacian noise.* The sum of errors, the sum of L_1 distances of projected points in a 2-dimensional subspace to the "true" 2-dimensional subspace of the data, versus outlier magnitude with Laplacian noise, for dimensions $m = 10$ and $m = 100$, and $p = 1, 2, 3$. The average sum of errors over 100 iterations is plotted. Error bars represent one standard deviation. The parameter p is the number of outlier-contaminated dimensions.



(a) $p = 1$, $m = 10$: outlier magnitude. (b) $p = 2$, $m = 10$: outlier magnitude.

(c) $p = 3$, $m = 10$: outlier magnitude. (d) $p = 1$, $m = 100$: outlier magnitude.

(e) $p = 2$, $m = 100$: outlier magnitude. (f) $p = 3$, $m = 100$: outlier magnitude.

Fig. 8. *Gaussian noise.* The sum of errors, the sum of L_1 distances of projected points in a 2-dimensional subspace to the "true" 2-dimensional subspace of the data, versus outlier magnitude with Gaussian noise, for dimensions $m = 10$ and $m = 100$, and $p = 1, 2, 3$. The average sum of errors over 100 iterations is plotted. Error bars represent one standard deviation. The parameter p is the number of outlier-contaminated dimensions.

Appendix B. Supplementary data

At the URL http://www.people.vcu.edu/~jpbrooks/l1pcastar are a script for L_1-PCA* for R (R Development Core Team, 2008), data for the numerical example of Sections 3 and 4, and simulated data for the experiments in Section 5. Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.csda.2012.11.007.

References

Agarwal, S., Chandraker, M.K., Kahl, F., Kriegman, D., Belongie, S., 2006. Practical global optimization for multiview geometry. Lecture Notes in Computer Science 3951, 592–605.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D., 1999. LAPACK Users' Guide, third ed. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Appa, G., Smith, C., 1973. On L_1 and Chebyshev estimation. Mathematical Programming 5, 73–87.

Baccini, A., Besse, P., de Faguerolles, A., 1996. A L_1-norm PCA and heuristic approach, In: Proceedings of the International Conference on Ordinal and Symbolic Data Analysis, pp. 359–368.

Brooks, J.P., Dula, J.H., 2013. The L_1-norm best-fit hyperplane problem. Applied Mathematics Letters 26, 51–55.

Charnes, A., Cooper, W.W., Ferguson, R.O., 1955. Optimal estimation of executive compensation by linear programming. Management Science 1, 138–150.
Choulakian, V., 2006. L_1-norm projection pursuit principal component analysis. Computational Statistics and Data Analysis 50, 1441–1451.
Croux, C., Ruiz-Gazen, A., 2005. High breakdown estimators for prinicpal components: the projection-pursuit approach revisited. Journal of Multivariate Analysis 95, 206–226.
Daudin, J.J., Duby, C., Trecourt, P., 1988. Stability of principal component analysis studied by the bootstrap method. Statistics 19, 241–258.
Ding, C., Zhou, D., He, X., Zha, H., 2006. R_1-pca: rotational invariant L_1-norm principal component analysis for robust subspace factorization. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 281–288.
Filzmozer, P., Fritz, H., Kalcher, K., 2009. pcaPP: robust PCA by projection pursuit.
Galpin, J.S., Hawkins, D.M., 1987. Methods of L_1 estimation of a covariance matrix. Computational Statistics and Data Analysis 5, 305–319.
Gao, J., 2008. Robust L1 principal component analysis and its Bayesian variational inference. Neural Computation 20, 555–572.
ILOG,, 2009. ILOG CPLEX Division. 889 Alder Avenue. Incline Village, Nevada.
Jolliffe, I.T., 2002. Principal Component Analysis, second ed. Springer.
Jot, S., 2011. pcaL1: an R package of principal component analysis methods using the $L1$ norm, Master's Thesis, Statistical Sciences and Operations Research, Virginia Commonwealth University, Richmond, Virgina.
Ke, Q., Kanade, T., 2003. Robust subspace computation using $L1$ norm, Technical Report CMU-CS-03-172, Carnegie Mellon University, Pittsburgh, PA.
Kier, L.B., Seybold, P.G., Cheng, C.-K., 2005. Cellular Automata Modeling of Chemical Systems: A Textbook and Laboratory Manual. Springer.
Kwak, N., 2008. Principal component analysis based on $L1$-norm maximization. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 1672–1680.
Li, G., Chen, Z., 1985. Projection-pursuit approach to robust distpersion matrices and principal components: primary theory and Monte Carlo. Journal of the American Statistical Association 80, 759–766.
Mangasarian, O.L., 1999. Arbitrary-norm separating plane. Operations Research Letters 24, 15–23.
Martini, H., Schöbel, A., 1998. Median hyperplanes in normed spaces—a survey. Discrete Applied Mathematics 89, 181–195.
McDonald, G.C., Schwing, R.C., 1973. Instabilities of regression estimates relating air pollution to mortality. Technometrics 15, 463–481.
R Development Core Team, 2008. R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.
Späth, H., Watson, G.A., 1987. On orthogonal linear l_1 approximation. Numerische Mathematik 51, 531–543.
Wagner, H.M., 1959. Linear programming techniques for regression analysis. Journal of the American Statistical Association 54, 206–212.