Algorithms **2014**, *7*, 429-443; doi:10.3390/a7030429

OPEN ACCESS

algorithms

ISSN 1999-4893

www.mdpi.com/journal/algorithms

Article

ℓ_1 Major Component Detection and Analysis (ℓ_1 MCDA) in Three and Higher Dimensional Spaces

Zhibin Deng [1,2], John E. Lavery [2,3], Shu-Cherng Fang [2] and Jian Luo [2,*]

[1] School of Management, University of Chinese Academy of Sciences, Beijing 100190, China;
E-Mail: zhibindeng@ucas.ac.cn

[2] Department of Industrial and Systems Engineering, North Carolina State University, Raleigh,
NC 27695-7906, USA; E-Mails: john.e.lavery4.civ@mail.mil (J.E.L.); fang@ncsu.edu (S.-C.F.)

[3] Mathematical Sciences Division and Computing Sciences Division, Army Research Office,
Army Research Laboratory, P.O. Box 12211, Research Triangle Park, NC 27709-2211, USA

* Author to whom correspondence should be addressed; E-Mail: jluo3@ncsu.edu;
Tel.: +1-919-917-5967.

*Received: 21 May 2014; in revised form: 21 July 2014 / Accepted: 23 July 2014 /
Published: 19 August 2014*

Abstract: Based on the recent development of two dimensional ℓ_1 major component detection and analysis (ℓ_1 MCDA), we develop a scalable ℓ_1 MCDA in the n-dimensional space to identify the major directions of star-shaped heavy-tailed statistical distributions with irregularly positioned "spokes" and "clutters". In order to achieve robustness and efficiency, the proposed ℓ_1 MCDA in n-dimensional space adopts a two-level median fit process in a local neighbor of a given direction in each iteration. Computational results indicate that in terms of accuracy ℓ_1 MCDA is competitive with two well-known PCAs when there is only one major direction in the data, and ℓ_1 MCDA can further determine multiple major directions of the n-dimensional data from superimposed Gaussians or heavy-tailed distributions without and with patterned artificial outliers. With the ability to recover complex spoke structures with heavy-tailed noise and clutter in the data, ℓ_1 MCDA has potential to generate better semantics than other methods.

Keywords: multidimensional heavy-tailed distribution; ℓ_1-norm; major component; n-dimensional; outlier; pattern recognition; robust principal component analysis

1. Introduction

In the analysis of human-based activities, including those in social media, one increasingly encounters "irregular" data, that is, data that follows a star-shaped statistical distribution with many irregularly positioned "spokes" and "clutters". The "spokes" are the data from distributions whose level surfaces of the probability density functions extend much further out in certain directions from the central point than other directions; and the "clutters" often consist of patterned outliers. The spokes and clutters often have special meaning but are currently difficult to recover. For example, the spokes in point clouds representing the intersection of roads and fences encountered in urban modeling contain the information about the roads and fences, while clutters in such data may represent obstructions between the sensing mechanism and the roads or fences. Analyzing the actual irregular structure in the point cloud is important for recovering semantic information out of the data. Classical principal component analysis (PCA) [1], which is based on the ℓ_2 norm, assumes few outliers and assumes a light-tailed statistical distribution similar to the elliptically shaped Gaussian, is not a meaningful approach for determining the major components of such data. However, point clouds obtained from reality often contain a large number of outliers and the points may be incomplete or follow a heavy-tailed distribution. To deal with this case, various types of "robust principal component analysis" (robust PCA) involving the ℓ_1 norm have been proposed and successfully developed [2–9]. This shift from the ℓ_2 norm to the ℓ_1 norm is part of a larger tendency in recent research that has taken place also in signal and image processing [10,11], compressive sensing [12,13], shape-preserving geometric modeling [14,15] and other areas.

The robust PCAs that have been developed so far have been successful for many classes of problems but are not able to handle the data that (1) follows a star-shaped statistical distribution with multiple irregularly positioned spokes and (2) includes many statistical and patterned outliers. Our goal in this paper is to create a method for identifying the major components of such data in three and higher dimensions. In [16], Ye *et al.* created an ℓ_1 major component detection and analysis (ℓ_1 MCDA) for 2D data that can handle data from distributions with irregularly positioned spokes. The ℓ_1 MCDA in [16] consists of the following steps:

1. Calculate the central point of the data and subtract it out of the data.
2. Find a local neighbor of a direction.
3. Calculate the quadratic surface that best fits the data in the current local neighbor in ℓ_1 norm.
4. Determine the location of the maximum of the quadratic surface over the local neighbor. If the location of the maximum is strictly inside the local neighbor, go to Step 5; otherwise move to the direction on the boundary of the local neighbor and return to Step 2.
5. Calculate a new best-fit quadratic surface on a larger neighbor of the direction identified in Step 4. Output the maximum of the new quadratic surface.

In this paper, we extend the 2D ℓ_1 MCDA of [16] to higher dimensions. Just like the 2D ℓ_1 MCDA of [16], the ℓ_1 MCDA for higher dimensions that we propose here involves a fundamental reformulation of all steps of PCA in a framework based on the assumption that the data points follow a heavy-tailed statistical distribution. The rest of this paper is arranged as follows. In Section 2, ℓ_1 MCDA in the n-dimensional space is derived. Computational results for different types of light-tailed and heavy-tailed distributions are presented in Section 3. Section 4 gives conclusions and future work.

2. High Dimensional ℓ_1 Major Component Detection and Analysis (MCDA)

The ℓ_1 MCDA that we propose includes algorithms to do the following:

(1) Calculate the central point of data and subtract it out of the data.

(2) Calculate the major directions of the point cloud resulting from Step 1.

(3) Calculate the radial spread of the point cloud in various directions such as directions where the radial spread is maximal.

The distance measure in the data space can be any norm that is appropriate for the data, while the distance measure in the data space was the ℓ_1 norm in [16].

2.1. Calculation of the Central Point

Our ℓ_1 MCDA assumes that the data is symmetrically distributed around a central point. This assumption will be eliminated in the future but is needed for now so that the calculation of the central point is accurate. Let the data be $\{\bar{\mathbf{x}}_m\}_{m=0}^{M-1}$. The central point of the data in Step 1 is to find the $\hat{\mathbf{x}}$ that minimizes

$$\sum_{m=0}^{M-1} d(\hat{\mathbf{x}}, \bar{\mathbf{x}}_m) \tag{1}$$

where $d(\hat{\mathbf{x}}, \bar{\mathbf{x}}_m)$ is the distance function for data points $\hat{\mathbf{x}}$ and $\bar{\mathbf{x}}_m$ in the data space. The distance function d can be any ℓ_p norm or p-th power of the ℓ_p norm or other norm function that the user wishes to choose. After subtracting the central point out of the data, the point cloud is centered at the origin of the space. For simplicity, we still denote the dataset after this change by $\{\bar{\mathbf{x}}_m\}_{m=0}^{M-1}$.

In our experiment, we chose the ℓ_1 norm as the distance function, *i.e.*, the multidimensional median is the central point of the dataset. The multidimensional median is not guaranteed to be an appropriate central point unless the data comes from a distribution that is symmetric with respect to the origin. For example, a point cloud in 2D representing a corner is a distribution in which the data are densest along the sides of a "V". In this case, the central point should be the vertex of the V rather than the multidimensional median of the data. We do not consider this issue in this present paper but will handle it in future research.

2.2. Calculation of the Major Directions and Median Radii in those Directions

The "major directions" are the directions in which the multidimensional distribution locally spreads farthest. We measure this spread by the "median radius" in each direction. To be more precise, assume that $f(r|\boldsymbol{\theta})$ is the conditional probability density function of radius r in the given direction $\boldsymbol{\theta}$. The median radius in the direction $\boldsymbol{\theta}$ is defined as the median of $f(r|\boldsymbol{\theta})$, *i.e.*, the value of $r^*(\boldsymbol{\theta})$ such that $\int_0^{r^*(\boldsymbol{\theta})} f(r|\boldsymbol{\theta})dr / \int_0^{+\infty} f(r|\boldsymbol{\theta})dr = 0.5$. According to this definition, a direction $\hat{\boldsymbol{\theta}}$ is one major direction if $r^*(\hat{\boldsymbol{\theta}})$ is a local maximum in the angular space $\boldsymbol{\theta}$. For a finite sample, the median radius in a given direction is estimated by the two-level median of the sample points over a small angular neighborhood around that direction. The details about the two-level median estimation will be described in the following algorithm.

The algorithm for calculating the major directions and median radii in those directions is described as follows:

1. For $m = 0, \cdots, M-1$, transform $\bar{\mathbf{x}}_m = (\bar{x}_m^1, \bar{x}_m^2, \cdots, \bar{x}_m^n)$ into "polar" form by the relations

$$\bar{r}_m = \sqrt{(\bar{x}_m^1)^2 + \cdots + (\bar{x}_m^n)^2} \tag{2}$$

$$\bar{\theta}_m^i = \frac{\bar{x}_m^i}{\bar{r}_m}, \ i = 1, \cdots, n \tag{3}$$

   Here, all $\bar{r}_m$ are nonnegative and the $\bar{\boldsymbol{\theta}}_m = (\bar{\theta}_m^1, \bar{\theta}_m^2, \cdots, \bar{\theta}_m^n)$ are on the unit sphere in $\Re^n$.
2. Choose a direction $\bar{\boldsymbol{\theta}}_{\bar{m}}$ to start, where $\bar{m} \in \{0, \ldots, M-1\}$.
3. Choose a positive integer I that represents the size of the neighbor of a point (including the given point itself). Determine the neighbor $\mathcal{N}_{\bar{m}}$ of $\bar{\boldsymbol{\theta}}_{\bar{m}}$ by choosing I directions $\bar{\boldsymbol{\theta}}_{\bar{m}_i}$, $\bar{m}_i \in \{0, \cdots, M-1\}$, $i = 1, 2, \ldots, I$, that are nearest to $\bar{\boldsymbol{\theta}}_{\bar{m}}$ in a given angular measure (ℓ_1-norm or ℓ_2-norm, *etc.*).
4. Determine the neighbor $\mathcal{N}_{\bar{m}_i}$ for each direction $\bar{\boldsymbol{\theta}}_{\bar{m}_i} \in \mathcal{N}_{\bar{m}}$ by choosing I directions $\bar{\boldsymbol{\theta}}_{\bar{m}_{i_j}}$, $\bar{m}_{i_j} \in \{0, \cdots, M-1\}$, $j = 1, 2, \ldots, I$, that are nearest to $\bar{\boldsymbol{\theta}}_{\bar{m}_i}$. For each direction $\bar{\boldsymbol{\theta}}_{\bar{m}_{i_j}} \in \mathcal{N}_{\bar{m}_i}$, obtain the neighborhood $\mathcal{N}_{\bar{m}_{i_j}}$ of I directions $\bar{\boldsymbol{\theta}}_m$, $m \in \{0, \ldots, M-1\}$, that are nearest to $\bar{\boldsymbol{\theta}}_{\bar{m}_{i_j}}$ and calculate the median $\tilde{r}_{\bar{m}_{i_j}}$ of the lengths $\{\bar{r}_m | \bar{\boldsymbol{\theta}}_m \in \mathcal{N}_{\bar{m}_{i_j}}\}$. Then, calculate the median $\hat{r}_{\bar{m}_i}$ of the lengths $\{\tilde{r}_{\bar{m}_{i_j}} | \bar{\boldsymbol{\theta}}_{\bar{m}_{i_j}} \in \mathcal{N}_{\bar{m}_i}\}$.
5. Determine the maximum of the median lengths $\hat{r}_{\bar{m}_i}$, $i = 1, 2, \ldots, I$. Let $i^* = \mathrm{argmax}_{i=1,2,\ldots,I}\{\hat{r}_{\bar{m}_i}\}$. If the maximum is achieved at $\bar{m}_{i^*} = \bar{m}$, go to Step 6. Otherwise, set $\bar{m} = \bar{m}_{i^*}$ and return to Step 3 .
6. Refine the location and value of the local maximum of the median radius. Calculate a neighbor $\mathcal{N}$ of $\lceil I/2 \rceil$ directions $\bar{\boldsymbol{\theta}}_m$ of the local maximal direction identified in Step 5, where $\lceil I/2 \rceil$ is the smallest integer no less than $I/2$. The median of $\bar{\boldsymbol{\theta}}_m$ in $\mathcal{N}$ is the calculated major direction of the distribution. The median of the lengths $\{\bar{r}_m | \bar{\boldsymbol{\theta}}_m \in \mathcal{N}\}$ is the estimate of the median radius in the major direction.

Remark 1. *In Step 4, we used two-level median fit process. This is equivalent to a local weighted median process except that the weight is not fixed or predetermined, but adaptive to the local neighbor information. A higher-level median fit process could be developed similarly. However, according to our computational experiment, the accuracy of the estimation is not obviously improved while the computational cost dramatically increases.*

Remark 2. *In Steps 4 and 6, one may choose the neighbor by collecting all the directions within a given distance from a direction. However, adopting this method may result in bad estimation because some neighbors may contain so few points that the median of these points is extremely biased from the theoretical value.*

The above procedure will definitely terminate because the algorithm will traverse all directions $\bar{\boldsymbol{\theta}}_m$ under the worst case. The procedure is for calculating one maximum (one spoke). To calculate all local maxima for a distribution with several major directions, one can choose different starting points for multiple implementations of this algorithm. For example, we can randomly find a direction that is orthogonal to the major directions obtained so far, and then choose the $\bar{\boldsymbol{\theta}}_{\bar{m}}$ that is nearest to this direction as the new starting point.

In the 2D version of ℓ_1 MCDA described in [16], the algorithm fits quadratic surfaces in the ℓ_1 norm to the data in the neighborhoods (Steps 3 and 5 of the algorithm in [16]) instead of calculating many two-level medians (Steps 4 and 6 of the algorithm described above in this paper). However, fitting a quadratic surface is not linearly scalable as the dimension increases, because both the size of the local neighborhood and the computing time increase quadratically. For this reason, the two-level medians were used here instead of quadratic surface fitting in order to develop a scalable procedure for high dimensional spaces.

3. Computational Experiments

In this section, we present comparisons of our ℓ_1 MCDA with Croux and Ruiz-Gazen's robust PCA [5] and Brooks *et al.*'s L_1 PCA [17]. Since both Croux and Ruiz-Gazen's and Brooks *et al.*'s PCA methods are also widely used for the data with outliers, comparisons of our ℓ_1 MCDA with their methods are imperative.

The following eight types of distributions were used for the computational experiments:

- n-Dimensional ($n \geq 3$) Gaussian without and with additional artificial outliers;
- n-Dimensional ($n \geq 3$) Student t (degree of freedom = 1) without and with additional artificial outliers;
- Four superimposed n-dimensional ($n \geq 3$) Gaussians without and with additional artificial outliers;
- Four superimposed n-dimensional ($n \geq 3$) Student t (degree of freedom = 1) without and with additional artificial outliers.

All computational results were generated using MATLAB codes in [18] and MATLAB R2012a [19] on a 2.50 GHz PC with 4 GB memory. Samples from Gaussian and Student t distributions were generated using the MATLAB mvnrnd and mvtrnd modules, respectively, with the covariance/correlation matrix

$$\sigma(n) = \begin{bmatrix} 1 & b & \cdots & b \\ b & 1 & \cdots & b \\ \vdots & \vdots & \ddots & \vdots \\ b & b & \cdots & 1 \end{bmatrix}$$

where $0 < b < 1$ and n is the size of the covariance/correlation matrix. In our experiment, the computational time for each sample was restricted to 3600 s.

For the one-major-direction situation, the ratio of the length of the longest major direction to that of other major directions (which are equal for the covariance/correlation matrix $\sigma(n)$) is set to be a constant C for all n. To accomplish this, we set the ratio of the maximum eigenvalue (α_{max}) to the minimal eigenvalue (α_{min}) to be C^2, *i.e.*, $\frac{\alpha_{max}}{\alpha_{min}} = \frac{1+(n-1)b}{1-b} = C^2$. Hence, $b = \frac{C^2-1}{n-1+C^2}$ in the covariance/correlation matrix $\sigma(n)$. In our experiment, we chose $C = 10$, which requires that $b = \frac{99}{n+99}$. Therefore, for the one-major-direction situation, the major direction is $(\frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}})^T$ and the ratio of the length of the longest major direction to that of other major directions is 10. In Figures 1 and 2, we present the examples of the datasets with 8000 data points (red dots) for one Gaussian distribution and one Student t distribution, respectively. In order to exhibit the major component clearly, the plot range in Figure 2 is the same as the one in Figure 1. There are many points outside the plot range of Figure 2. The directions and magnitudes of median radii in the major directions are indicated by blue bars emanating from the origin.

Figure 1. Sample from one Gaussian distribution.

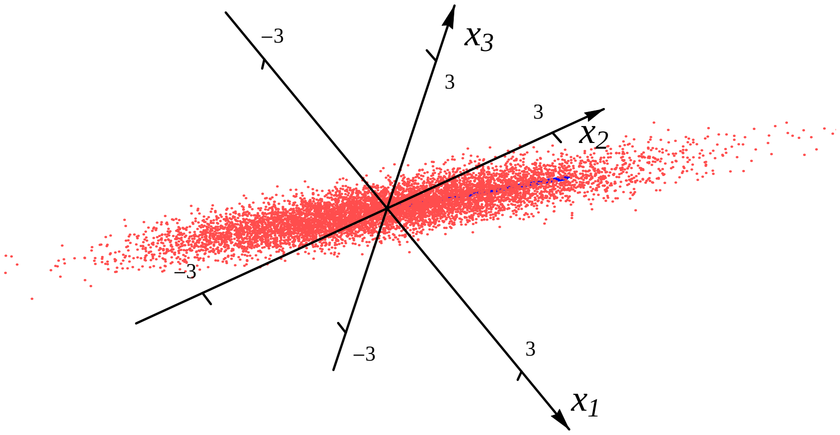
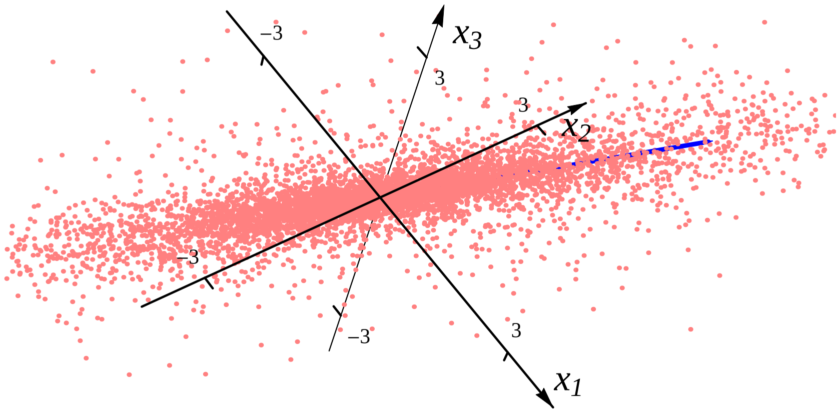


Figure 2. Sample from one Student t distribution.



For the four-major-directions situation, we overlaid four distributions rotated to the directions $(\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}})^T$, $(1, 0, 0, \ldots, 0)^T$, $(0, 1, 0, \ldots, 0)^T$ and $(0, 0, 1, \ldots, 0)^T$. The samples for each major direction are first generated with correlation/covariance matrix $\sigma(n)$, and then rotated in ℓ_2-norm to other major directions after multiplying by a factor (to make the lengths of the "spokes" different from each other). For example, for the 3D four-major-directions distribution created by overlaying four Gaussians, the median radii of the four major directions are 2.643, 3.964, 1.586 and 7.928. Figures 3 and 4 show the examples of the datasets with 32,000 data points (red dots) for four superimposed Gaussian distributions and four Student t distributions, respectively. The blue bars indicate the major directions and the magnitudes of median radii in the major directions.

Figure 3. Sample from four overlaid Gaussian distributions.

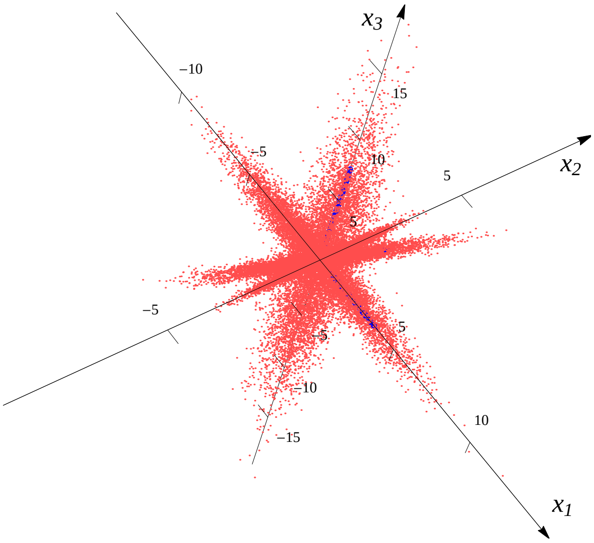
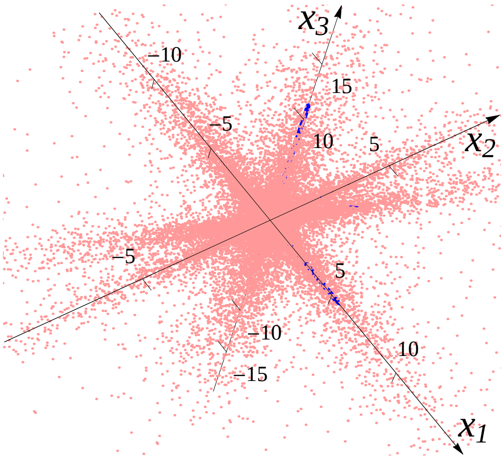


Figure 4. Sample from four overlaid Student t distributions.



The artificial outliers are generated from a uniform distribution on a simplex. The n vertices of the simplex are randomly chosen from the intersection of the ball $x_1^2 + x_2^2 + \ldots + x_n^2 = 1500^2$ with the hyperplane $a_1x_1 + a_2x_2 + \ldots + a_nx_n = 1000$. For each distribution in the data, the angle between the normal of the hyperplane $(a_1, a_2, \ldots, a_n)^T$ and the major direction of the distribution is 45°. In Figures 5–8, we present examples with 10% artificial outliers (depicted by blue dots) for one Gaussian distribution, one Student t distribution, four superimposed Gaussian distributions and four superimposed Student t distributions (points from distributions depicted by red dots), respectively.

Figure 5. Sample from one Gaussian distribution with 10% artificial outliers.

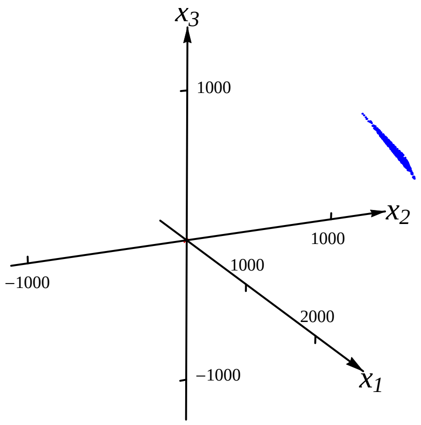


Figure 6. Sample from one Student t distribution with 10% artificial outliers.

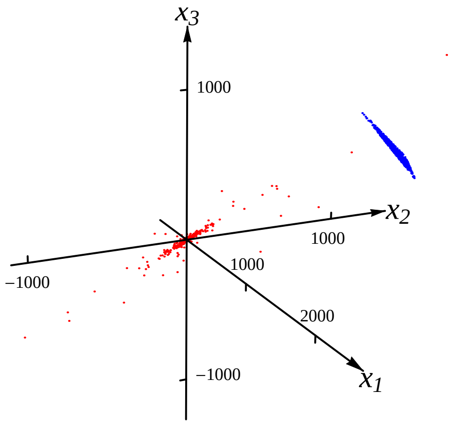


Figure 7. Sample from four Gaussian distributions with 10% artificial outliers.

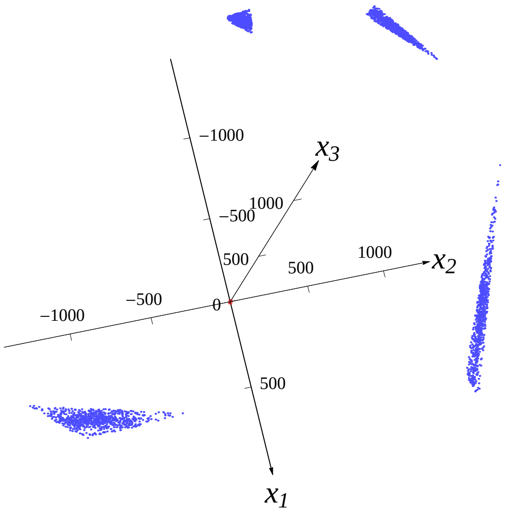
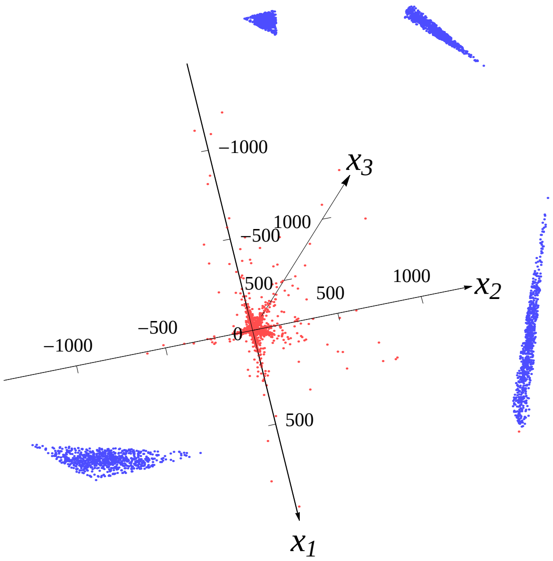


Figure 8. Sample from four Student t distributions with 10% artificial outliers.



For each type of data, we carried out 100 computational experiments, each time with a new sample from the statistical distribution(s), including the uniform distributions that generated the outliers. The neighbors required in Steps 4 and 6 of the proposed ℓ_1 MCDA were calculated by the k-nearest-neighbors (kNN) method [20]. Other methods for identifying neighbors may also be used.

To measure the accuracy of the results, we calculated the average over 100 computational experiments of the absolute error of each major direction and the average of the relative error of the median radius in that direction *vs.* the theoretical values of the direction of maximum spread and the median radius in that direction of the distribution. The theoretical value of major direction for the one-major-direction case is $(1/\sqrt{n}, \cdots, 1/\sqrt{n})^T$ and the theoretical values of major directions for the four-major-directions case are $(1/\sqrt{n}, \cdots, 1/\sqrt{n})^T$, $(1, 0, \cdots, 0)^T$, $(0, 1, 0, \cdots, 0)^T$, $(0, 0, 1, 0, \cdots, 0)^T$, respectively. The theoretical value of median radius for each major direction is calculated as the value of r^* such that $\int_0^{r^*} f(r|\boldsymbol{\theta})dr / \int_0^{+\infty} f(r|\boldsymbol{\theta})dr = 0.5$ by numerical integration, where $f(r|\boldsymbol{\theta})$ is the conditional probability density function of radius r along the major direction $\boldsymbol{\theta}$.

In Tables 1–4, we present computational results for the sets of 8000 points with local neighborhood of size $I = 160$ (2% of total points) for the one-major-direction situation. We compared our method with the methods that appeared in the literature [5,17]. Specifically, the MATLAB version of pcaPP [21] package of the R language for statistical computing [22] was used to implement the robust PCA proposed by Croux and Ruiz-Gazen in [5]. Brooks *et al.*'s L_1 PCA [17] was implemented by using MATLAB and IBM ILOG CPLEX 12.5 [23]. The computational results for those two PCA methods are also summarized in Tables 1–4. Table 5 shows the average computational time for each method when $n = 10$. For higher dimensions, the average computational time of ℓ_1 MCDA is shorter than the other two PCA methods. In Tables 6–9, we present the computational results for 32,000 points with local neighborhood of size $I = 320$, 160 and 80 (*i.e.*, 2%, 1% and 0.5% of total points, respectively) for the four-major-directions situation.

Table 1. The results of ℓ_1 MCDA and Robust PCAs on one Gaussian distribution.

n	ℓ_1 **MCDA**		**Croux + Ruiz-Gazen**		**Brooks *et al.***	
	av. abs_err of angle	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**
3	0.0112	0.0314	0.0289	0.0681	0.0556	0.0675
8	0.0233	0.2498	0.0399	0.2177	0.0118	0.1586
10	0.0277	0.3129	0.0308	0.2683	0.0090	0.2256
30	0.0315	0.5772	0.0112	0.5308	0.0052	0.5099
50	0.0363	0.6539	0.0136	0.6269	–	–

Table 2. The results of ℓ_1 MCDA and Robust PCAs on one Student t distribution.

n	ℓ_1 **MCDA**		**Croux + Ruiz-Gazen**		**Brooks *et al.***	
	av. abs_err of angle	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**
3	0.0276	0.0921	0.0346	0.1224	0.0557	0.1310
8	0.0423	0.2035	0.0483	0.2113	0.0118	0.1800
10	0.0441	0.2717	0.0455	0.2625	0.0090	0.2352
30	0.0446	0.5367	0.0113	0.5163	0.0053	0.4869
50	0.0448	0.6350	0.0140	0.6100	–	–

Table 3. The results of ℓ_1 MCDA and Robust PCAs for one Gaussian distribution with 10% artificial outliers.

n	ℓ_1 **MCDA**		**Croux + Ruiz-Gazen**		**Brooks *et al.***	
	av. abs_err of angle	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**
3	0.0119	0.0373	0.0318	0.0759	0.0560	0.0654
8	0.0236	0.2512	0.0499	0.2292	0.0211	0.1588
10	0.0279	0.3134	0.0549	0.2781	0.0170	0.2289
30	0.0322	0.5785	0.0990	0.5376	0.0072	0.5077
50	0.0375	0.6601	0.1393	0.6308	–	–

Table 4. The results of ℓ_1 MCDA and Robust PCAs for one Student t distribution with 10% artificial outliers.

n	ℓ_1 **MCDA**		**Croux + Ruiz-Gazen**		**Brooks *et al.***	
	av. abs_err of angle	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**
3	0.0276	0.0969	0.0366	0.1140	0.0554	0.1515
8	0.0424	0.2038	0.0556	0.2054	0.0214	0.1714
10	0.0441	0.2725	0.0616	0.2775	0.0170	0.2314
30	0.0467	0.5376	0.1009	0.5238	0.0072	0.4833
50	0.0449	0.6351	0.1390	0.6142	–	–

Table 5. Average computational times (in seconds) to generate the results in Tables 1–4 for $n = 10$.

	ℓ_1 **MCDA**	**Croux + Ruiz-Gazen**	**Brooks *et al.***
Table 1	3.77	4.20	236.03
Table 2	3.64	4.50	239.32
Table 3	3.55	4.46	286.29
Table 4	3.21	4.49	278.69

Table 6. The results of ℓ_1 MCDA on four superimposed Gaussian distributions.

n	**knn = 640 (2%)**		**knn = 320 (1%)**		**knn = 160 (0.5%)**	
	av. abs_err of angle	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**
3	0.0124	0.0499	0.0089	0.0374	0.0121	0.0428
10	0.0226	0.0904	0.0294	0.0666	0.0364	0.0805
50	0.0269	0.5323	0.0352	0.4931	0.0448	0.4518

Table 7. The results of ℓ_1 MCDA on four superimposed Student t distributions.

n	**knn = 640 (2%)**		**knn = 320 (1%)**		**knn = 160 (0.5%)**	
	av. abs_err of angle	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**
3	0.0120	0.0552	0.0155	0.0794	0.0211	0.1420
10	0.0225	0.3225	0.0318	0.2458	0.0437	0.1777
50	0.0250	0.6220	0.0335	0.5814	0.0430	0.5355

Table 8. The results of ℓ_1 MCDA on four superimposed Gaussian distributions with 10% artificial outliers.

n	**knn = 640 (2%)**		**knn = 320 (1%)**		**knn = 160 (0.5%)**	
	av. abs_err of angle	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**
3	0.0152	0.0396	0.0099	0.0406	0.0171	0.0493
10	0.0271	0.2614	0.0208	0.3044	0.0173	0.3615
50	0.0379	0.6275	0.0308	0.6566	0.0261	0.6920

Table 9. The results of ℓ_1 MCDA on four superimposed Student t distributions with 10% artificial outliers.

n	**knn = 640 (2%)**		**knn = 320 (1%)**		**knn = 160 (0.5%)**	
	av. abs_err of angle	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**	**av. abs_err of angle**	**av. rel_err ra. of length**
3	0.0279	0.1639	0.0184	0.0941	0.0173	0.0501
10	0.0452	0.1671	0.0336	0.2570	0.0238	0.3308
50	0.0499	0.5333	0.0388	0.5851	0.0293	0.6198

Remark 3. *In Tables 1–4, the radius information for the robust PCA methods [5,17] was generated by Step 6 of the algorithm described in this paper with the major directions returned by their own. Some results are not available because the computational time exceeded the limit of 3600 s.*

Remark 4. *In Tables 6–9, no results for either Croux and Ruiz-Gazen's robust PCA or Brooks* et al.*'s L_1 PCA are presented, because these two methods provide only one major direction for the superimposed distributions and do not yield any meaningful information about the individual major direction.*

The results in Tables 1 and 2 indicate that, when there is only one major direction in the data, ℓ_1 MCDA obtains comparable results to Croux and Ruiz-Gazen's robust PCA in accuracy. Although Brooks *et al.*'s L_1 PCA outperforms the proposed ℓ_1 MCDA, it requires much longer computational time as shown in Table 5. The results in Tables 3 and 4 indicate that ℓ_1 MCDA outperforms Croux and Ruiz-Gazen's robust PCA in all cases, especially when the dimension is larger than 30. Brooks *et al.*'s L_1 PCA obtained similar results as ℓ_1 MCDA but consumed much more CPU time as shown in Table 5. It is noticeable that the accuracy of the proposed method decreases as the dimension increases. The reason is that for a given data size, the number of points falling into a fixed neighbor of the major direction decreases and the two-level median estimation deteriorates as the dimension grows higher. In contrast, the accuracy of the robust PCAs increases as the dimension increases because both robust PCAs used project-pursuit method, whose performance degrades when the underlying dimension decreases [17].

The results in Tables 6–9 indicate that our method can obtain good accuracy by using a small neighbor size (up to 2% of the size of given data) for the four-major-directions case. It is worth to point out that the four major directions are not orthogonal to each other, and the proposed ℓ_1 MCDA is very robust by noting that the accuracy of ℓ_1 MCDA for data with the artificial outliers is as good as the one of ℓ_1 MCDA without outliers. Although the distributions designed in the experiment are symmetrically around some center, the proposed algorithm can also deal with the data from asymmetric distributions.

The computational results show that, for the types of data considered here, ℓ_1 MCDA has the marked advantage in accuracy and efficiency when comparing with robust PCAs. Moreover, ℓ_1 MCDA can deal with cases of multiple components for which standard and robust PCAs perform less well or even poorly.

4. Conclusions and Future Work

The assumptions of the distribution underlying ℓ_1 MCDA are much less restrictive than those underlying most robust PCAs. The ℓ_1 MCDA that we have developed has the following advantages:

(1) it allows use of various distance functions in the data space (although we used ℓ_2 norm in this paper); (2) it works well for both heavy-tailed and light-tailed distributions; (3) it is applicable to data that have multiple spokes, contain patterned outliers, and do not necessarily have mutually orthogonal major directions; (4) it does not require the assumption of sparsity of the major components or of the error, and (5) it utilizes local information, instead of global information, in each iteration to improve the efficiency. These advantages allow ℓ_1 MCDA to be used in many circumstances in which robust PCAs cannot be used. For many geometric problems and for, perhaps, most "soft" problems (face and pattern recognition, image analysis, social network analysis, *etc.*), irregularly positioned "spokes" with many outliers are likely to be commonly encountered. Identifying these spokes is part of a process of generating semantics from raw data. With the ability to recover complex spoke structures in spite of the presence of heavy-tailed statistical noise and of clutter, ℓ_1 MCDA shows its potential to generate better semantics from data than other methods.

Topics that require further investigation include:

- Use of alternative principles for calculation of the central point (for example, for V-shaped distributions or more complicated asymmetric distributions with spokes for which the median of the data is not a meaningful central point);
- Properties of ℓ_1 MCDA for high dimensions (10^4 to 10^6);
- Ability of ℓ_1 MCDA to deal with missing data.

Irregular, high-dimensional heavy-tailed distributions are likely to describe a large number of financial, economic, social, networking, and physical phenomena. The ℓ_1 MCDA proposed in this paper is a candidate for a new, fully robust procedure that can be used for going from data to semantics for such phenomena.

Acknowledgments

This work was generously supported by United States Army Research Office Grant # W911NF-04-D-0003, by the North Carolina State University Edward P. Fitts Fellowship and by United States National Science Foundation Grant # DMI-0553310. It is the policy of the Army Research Office (ARO) that university personnel do not need to do joint work with ARO personnel in order to receive grants from ARO. We also thank to the two anonymous reviewers whose comments improved the quality of this paper.

Author Contributions

The authors contributed equally to this paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Jolliffe, I.T. *Principal Component Analysis*, 2nd ed.; Springer: New York City, NY, USA, 2002.

2. Candès, E.J.; Li, X.; Ma, Y.; Wright, J. *Robust Principal Component Analysis*; Technical Report No. 13; Department of Statistics, Stanford University: Stanford, CA, USA, 2009.
3. Choulakian, V. L_1-Norm projection pursuit principal component analysis. *Comput. Stat. Data Anal.* **2006**, *50*, 1441–1451.
4. Croux, C.; Filzmoser, P.; Fritz, H. Robust sparse principal component analysis. *Technometrics* **2013**, *55*, 202–214.
5. Croux, C.; Ruiz-Gazen, A. High breakdown estimators for principal components: The projection-pursuit approach revisited. *J. Multivar. Anal.* **2005**, *95*, 206–226.
6. Ke, Q.; Kanade, T. Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming. *IEEE Conf. Comput. Vis. Pattern Recognit.* **2005**, *1*, 739–746.
7. Kwak, N. Principal component analysis based on L_1-norm maximization. *IEEE Trans. Pattern Anal.* **2008**, *30*, 1672–1680.
8. Lin, Z.; Chen, M.; Wu, L.; Ma, Y. The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrix. Available online: http://arxiv.org/abs/1009.5055 (accessed on 18 October 2013).
9. Xu, H.; Caramanis, C.; Mannor, S. Outlier-robust PCA: the high-dimensional case. *IEEE Trans. Inf. Theory* **2013**, *59*, 546–572.
10. Gribonval, R.; Nielsen, M. Sparse approximations in signal and image processing. *Signal Process.* **2006**, *86*, 415–416.
11. Lai, M.-J.; Wang, J. An unconstrained ℓ_q minimization with $0 < q \leq 1$ for sparse solution of under-determined linear systems. *SIAM J. Optim.* **2010**, *21*, 82–101.
12. Candès, E.J.; Wakin, M.B. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30.
13. Chartrand, R. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.* **2007**, *14*, 707–710.
14. Auquiert, P.; Gibaru, O.; Nyiri, E. Fast $L_1^kC^k$ polynomial spline interpolation algorithm with shape-preserving properties. *Comput. Aided Geom. Des.* **2011**, *28*, 65–74.
15. Yu, L.; Jin, Q.; Lavery, J.E.; Fang, S.-C. Univariate cubic L_1 interpolating splines: Spline functional, window size and analysis-based algorithm. *Algorithms* **2010**, *3*, 311–328.
16. Tian, Y.; Jin, Q.; Lavery, J.E.; Fang, S.-C. ℓ^1 major component detection and analysis (ℓ^1 MCDA): Foundations in two dimensions. *Algorithms* **2013**, *6*, 12–28.
17. Brooks, J.P.; Dulá, J.H.; Boone, E.L. A pure L_1-norm principal component analysis. *Comput. Stat. Data Anal.* **2013**, *61*, 83–98.
18. Deng, Z.; Luo, J. FANGroup-Fuzzy And Neural Group at North Carolina State University. Available online: http://www.ise.ncsu.edu/fangroup/index.htm (accessed on 8 August 2014).
19. *MATLAB Release 2012a*; The MathWorks Inc.: Natick, MA, USA, 2012.
20. Friedman, J.H.; Bentely, J.; Finkel, R.A. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.* **1997**, *3*, 209–226.
21. Filzmozer, P.; Fritz, H.; Kalcher, K. pcaPP: Robust PCA by Projection Pursuit. Aailable online: http://cran.r-project.org/web/packages/pcaPP/index.html (accessed on 22 March 2013).

22. R Development Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2011.
23. IBM ILOG CPLEX Optimization. Available online: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/ (accessed on 16 March 2014).