A MULTIVARIATE MEDIAN IN BANACH SPACES AND APPLICATIONS TO ROBUST PCA

DAVID BRUCE
ADVISOR: PROF. ROMAN VERSHYNIN

Abstract. With the rise in prominence of high dimensional data, multivariate measures of center have become very important. In this paper we focus on one multivariate measure of center - the geometric median, which is defined as the minimizer of the sum of distances to the data points. We study the quantitative robustness of the geometric median. Showing that for a non-degenerate distribution of N points, altering k points can only change the median by at most $O(k/N)$. Taking advantage of this robustness we introduce a robust form of Principle Component Analysis (PCA), which is based on what we call the median covariance matrix. Since there are several natural matrix norms, we look at the notion of the geometric median in general Banach Spaces. We conclude by conjecturing that the geometric median is robust in all uniformly convex Banach spaces.

1. Background

Given a data set $\{X_1, \ldots, X_n\}$ in $\mathbb{R}$ one can use various measures of center to describe the data. A very common measure of center is the mean of a data set, denoted μ, which is calculated by summing up all the data points and dividing by the number of observations.

$$\mu = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

Although very nice computationally the mean does have the large drawback of being very sensitive to outliers. For example, if we had the data set $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ we see that the mean is exactly 5. However, if our data set were to contain an outlier like 50 the mean becomes 9.5, which is already outside the bulk of our data. One way of describing the robustness of a measure of center is by how many points must be changed arbitrarily in a data set of n points in order to arbitrarily change the measure of center. This is called the breakdown point, and it takes values between $1/n$ and $1/2$. The breakdown point cannot be higher than $1/2$ because once half the data is changed outliers then become indistinguishable. As we began to notice with our example above changing one point can arbitrarily change the mean of a data set, and thus the mean has a breakdown point of $1/n$. So by this measure of robustness the mean is as sensitive to outliers as possible.

Date: Summer 2011
Funded by the NSF through an REU program.

Another common measure of center is the median, which is calculated by ordering all the elements of the data set and then finding which data point is in the middle. So if we again consider the data set $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ then the median is also 5. However, the nice thing about the median is that it is much less sensitive to outliers than the mean. If our data set were to contain the outlier 50 again we see the median only become 5.5, and in fact the breakdown point of the median is .5. Meaning by this measure of robustness the median is as robust as a statistic can be.

Now that we have reviewed the behavior of various measurers of center in one dimension it would be nice to try and generalize the mean and median to higher dimensions. The mean generalizes very nicely to higher dimensions since we can simply use the same formula we did for one dimension. However, generalizing the median is slightly more complicated since there is no way to order $\mathbb{R}^n$. One approach to generalizing the median would be to fix a basis and then calculate the median coordinatewise, but this generalization is not rotationally invariant and so another might be better.

To generalize the median to higher dimensions we need first note alternative characterizations of the mean and the median in one dimension. Namely the mean μ is the number which minimizes the mean error squared and the median M is the number, which minimizes the mean error.

$$M = \arg\min_{M \in \mathbb{R}} \sum_{i=1}^{N} |X_i - M|.$$

$$\mu = \arg\min_{\mu \in \mathbb{R}} \sum_{i=1}^{N} |X_i - \mu|^2.$$

Using this definition of the median it becomes easy to generalize the median to higher dimensions:

Definition 1 (1)**.** *The median of points $X_1, X_2, \ldots, X_N \in \mathbb{R}^n$ is defined as:*

$$\mathrm{Med}(X_1, \ldots, X_N) = \arg\min_{M \in B} \sum_{i=1}^{N} \|X_i - M\|.$$

This generalization of the median often called the spatial or geometric median is nice in that just like the median in one dimension this has breakdown point of $1/2$, and unlike the cordinatewise median this is rotationally invariant. [2]

2. Euclidean Space

The first part our project focused on studying the properties of the spatial median in Euclidean spaces namely on trying to quantitatively describe the robustness. However, before we get to describing the robustness of the median we need to find an alternative characterization of the spatial median. Since the median is the point which minimizes the

sum of the distances from the points it will be useful to calculate the gradient in order to help find the minimizing value.

Theorem 1. *Given a set of points, $X_1, X_2, \ldots, X_N \in \mathbb{R}^n$, define a functional f by:*

$$f(Y) = \sum_{i=1}^{m} \|X_i - Y\|_2.$$

Then the gradient of f is:

$$\nabla f(Y) = -\sum_{i=1}^{N} \frac{(X_i - Y)}{\|X_i - Y\|_2}.$$

Proof. Let us calculate the partial derivative of f with respect to an arbitrary Y_j. Using the definition of the Euclidean Norm we see that:

$$\frac{\partial}{\partial Y_j} \sum_{i=1}^{N} \|X_i - Y\|_2 = \frac{\partial}{\partial Y_j} \sum_{i}^{N} \left(\sum_{j=1}^{n} (X_{ij} - Y_j)^2 \right)^{1/2},$$

which is equal to:

$$\sum_{i=1}^{n} \frac{-(X_{ij} - Y_j)}{\left(\sum_{j=1}^{n} (X_{ij} - Y_j)^2 \right)^{1/2}} = \sum_{i=1}^{n} \frac{-(X_{ij} - Y_j)}{\|X_i - Y\|_2}.$$

So with a few simplifications we see that the gradient of f is:

$$\nabla f(Y) = -\sum_{i=1}^{N} \frac{(X_i - Y)}{\|X_i - Y\|_2}.$$

□

Now by the definition of the spatial median we know that the median M of $X_1, \ldots, X_N$ minimizes the function f, and so from this proposition we find an alternative characterization of the median.

Corollary 1.1. *Given a set of points, $X_1, X_2, \ldots, X_N \in \mathbb{R}^n$, then $M \in \mathbb{R}^n$ is the median for this data set if and only if:*

$$\sum_{i=1}^{N} \frac{(X_i - M)}{\|X_i - M\|_2} = 0.$$

Proof. Assume M is the median for this data set then by definition M minimizes $f(Y)$, and therefore we know that $\nabla f(M) = 0$. So from Theorem 1 we see that since $\nabla f(M) = 0$, then:

$$\sum_{i=1}^{N} \frac{(X_i - M)}{\|X_i - M\|_2} = 0.$$

Now assume that:

$$\sum_{i=1}^{N} \frac{(X_i - M)}{\|X_i - M\|_2} = 0.$$

Again by Theorem 1 we can conclude that $\nabla f(M) = 0$. Since f is a convex function we know then that M minimizes f, and therefore by definition M is the median of $X_1, X_2, \ldots, X_N$. □

This characterization of the spatial median points to the interesting fact that the median is based on not necessarily on how far apart points distributed, but instead on how the are distributed. For example, the next corollary shows that you can move a point along a ray that goes from the median through point without changing where the median is.

Corollary 1.2. *Given a set of points, $X_1, X_2, \ldots, X_n \in \mathbb{R}^n$, let M be the median for this data set. If we create a new data set by replacing any X_i by $t(X_i - M) + M$, where $t > 0$,then the median of this new data set will remain M.*

Proof. Let M be the median of $X_1, X_2, \ldots, X_N$ then because of Corollary 1.1 we know that:

$$\sum_{i=1}^{N} \frac{(X_i - M)}{\|X_i - M\|_2} = 0. \tag{1}$$

With out loss of generality let us create a new set by replacing an arbitrary X_N by $t(X_N - M) + M$ where $t > 0$. Again thanks to Corollary 1.1 to show that M is the median for this new set it suffices to show that:

$$\sum_{i=1}^{N} \frac{(X_i - M)}{\|x_i' - M\|_2} = 0. \tag{2}$$

In order to show this note that:

$$\frac{((t(X_N - M) + M) - M)}{\|(t(X_N - M) + M) - M\|_2} = \frac{(t(X_N - M)}{\|(t(X_N - M)\|_2} = \frac{(X_N - M)}{\|X_N - M\|_2}.$$

Therefore, equation 1 implies equation 2, and thus by Corollary 1.1 we know that M is the median for this new set of data. □

Now that we have explored a few properties of the geometric median, and have characterized it based on its gradient we can return to the question of quantitatively describing its robustness. Namely we can answer the question of given N points by how much can the median change if we arbitrarily alter k of these points? In order to answer this question we will want to find two bounds. First we will want to find the most the that the gradient of the median function can become unbalanced by because this will tell us the worst that will happen if we alter k points. Second we want to find the smallest amount the gradient can change in a given direction. This is because if we have N points we can consider each

point being able to absorb a certain amount of the unbalance in the gradient, and so by minimizing this we can find the most the median will have to move.

This first lemma begins to answer this second question of how little the gradient will change in a given direction assuming that the point is not in line with direction the gradient is changing in. Meaning we are saying that in general our data set cannot be essentially one dimensional.

Lemma 1. *Consider a euclidean ball $B(0,R)$ and a cylinder C with infinite dimension in the u direction and base $B'(0,\beta R)$, where $0<\beta<1$ and $\|u\|_2=1$. Let $x\in B\setminus C$ and $\epsilon>0$ then:*

$$\delta = P_u\left(\frac{x+\varepsilon u}{\|x+\varepsilon u\|_2}\right) - P_u\left(\frac{x}{\|x\|_2}\right) \geq \frac{\varepsilon}{R+\varepsilon}\left(1-\sqrt{1-\beta^2}\right).$$

Proof. Since we are projecting one unit vector onto another unit vector we can define δ as:

$$\delta = cos(\theta') - cos(\theta) = \frac{\langle x,u\rangle + \langle \varepsilon u,u\rangle}{\|x+\varepsilon u\|_2} - \frac{\langle x,u\rangle}{\|x\|_2},$$

and so we will now try and find a lower bound for this formula. To do this we can apply the triangle inequality to obtain:

$$\|x+\varepsilon u\|_2 \leq \|x\|_2 + \|\varepsilon u\|_2 = \|x\|_2 + \varepsilon,$$

which means we can bound δ by:

$$\delta = \frac{\langle x,u\rangle + \langle \varepsilon u,u\rangle}{\|x+\varepsilon u\|_2} - \frac{\langle x,u\rangle}{\|x\|_2} \geq \frac{\langle x,u\rangle + \langle \varepsilon u,u\rangle}{\|x\|_2+\varepsilon} - \frac{\langle x,u\rangle}{\|x\|_2}.$$

Simplifying this expression by using the fact that $\|u\|_2 = 1$, and then combining the fractions we see that:

$$\delta \geq \frac{\langle x,u\rangle + \langle \varepsilon u,u\rangle}{\|x\|_2+\varepsilon} - \frac{\langle x,u\rangle}{\|x\|_2} = \frac{\|x\|_2\varepsilon - \varepsilon\langle x,u\rangle}{\|x\|_2\left(\|x\|_2+\varepsilon\right)}.$$

Factoring this expression we obtain:

$$\delta \geq \frac{\varepsilon}{\|x\|_2+\varepsilon}\left(1-\frac{\langle x,u\rangle}{\|x\|_2}\right).$$

In order to obtained the desired lower bound we will now find a lower bound for $\frac{\varepsilon}{\|x\|_2+\varepsilon}$ and a lower bound for $\left(1-\frac{\langle x,u\rangle}{\|x\|_2}\right)$. To find the first lower bound we simply must note that because we assumed that $x_i\in B\setminus C$ then $\|x\|_2\leq R$, and so:

$$\frac{\varepsilon}{\|x\|_2+\varepsilon} \geq \frac{\varepsilon}{R+\varepsilon}.$$

Finding the bound for $\left(1-\frac{\langle x,u\rangle}{\|x\|_2}\right)$ will be slightly more challenging. We will first note that:

$$\left(1-\frac{\langle x,u\rangle}{\|x\|_2}\right) = 1-cos(\theta).$$

By our assumption that $x \in B \setminus C$, we know that $\langle x, v \rangle \geq \beta R$, where $\langle u, v \rangle = 0$, and again we know that $\|x\|_2 \leq R$. Combining these two facts with the definition of sin we know that:

$$sin(\theta) = \frac{\langle x, v \rangle}{\|x\|_2} \geq \frac{\beta R}{R} = \beta.$$

Now that we know a bound for $sin(\theta)$ we can easily solve for a bound for $1 - cos(\theta)$ by using the Pythagorean Identity, which will yield:

$$1 - cos(\theta) = \left(1 - \frac{\langle x, u \rangle}{\|x\|_2}\right) \geq \sqrt{1 - \beta}.$$

So now that we have our two desired lower bounds we can combine them to obtain the desired result of:

$$\delta \geq \frac{\varepsilon}{\|x\|_2 + \varepsilon} \left(1 - \frac{\langle x, u \rangle}{\|x\|_2}\right) \geq \frac{\varepsilon}{R + \varepsilon} \left(1 - \sqrt{1 - \beta^2}\right).$$

□

With this lemma and our previous work in hand we can combine all of this together to give us the desired bound on the movement of the median, and which shows that the median is in fact quite robust.

Theorem 2. *Consider a set of N points, $X := \{x_1 \dots x_N\}$, such that $x_i \in \mathbb{R}^n$, $\|x_i\|_2 \leq R$, and γN points lie outside of any given cylinder that has 1 infinite dimension and a base with radius βR, where $0 < \beta < 1$. Adding, removing, or albitrarly altering k will change the median of X by at most:*

$$c(\gamma, \beta) \frac{kR}{N} = \left(\frac{4}{\left(1 - \sqrt{1 - \beta^2}\right)\gamma}\right) \frac{kR}{N},$$

provided that $k < N/2$ and $(1 - \sqrt{1 - \beta^2})\frac{N}{k}\gamma > 2$.

Proof. Let M be the median of X. We will first begin by translating our data so that the median, M, lies at the origin. In order to do this we will create a new data set, $X' := \{x'_i = x_i - M\}$. Since the median must reside in the convex hull of X, we know that $\|M\|_2 \leq R$, and so we know that for each x'_i:

$$\|x'_i\|_2 = \|x_i - M\|_2 \leq \|x_i\|_2 + \|M\|_2 \leq R + R = 2R.$$

Applying Lemma 1 we see that for each x'_i:

$$\delta_i \geq \frac{\varepsilon}{2R + \epsilon} \left(1 - \sqrt{1 - \beta^2}\right).$$

This means that the total possible change in one direction is bounded by:

$$\frac{\varepsilon}{2R + \varepsilon} \left(1 - \sqrt{1 - \beta^2}\right) \gamma N \leq \sum_{i=1}^{\gamma N} \delta_i.$$

According to Corollary 1.1 since M is the median for X we know that the gradient at M equals 0. Since gradient for the median function as described in Theorem 1 is just adding up a series of unit vectors by altering k points we know that:

$$\sum_{1}^{\gamma N} \frac{x_i'}{\|x_i'\|_2} \leq k.$$

Combining all of this we know that:

$$\frac{\varepsilon}{2R+\varepsilon}\left(1-\sqrt{1-\beta^2}\right)\gamma N \leq k.$$

So this implies that the most the median can change by is ε, and so we simply must solve the above inequality for ε. Solving this for ε we obtain:

$$\varepsilon \leq \frac{2R}{\left(1-\sqrt{1-\beta^2}\right)\frac{N}{k}\gamma - 1}.$$

Since we assumed that $(1-\sqrt{1-\beta^2})\frac{N}{k}\gamma > 2$ we know that:

$$\varepsilon \leq \frac{4Rk}{\left(1-\sqrt{1-\beta^2}\right)N\gamma}.$$

□

So we see that in Euclidean space the spatial median is robust not only in the sense that it has a breakdown point of $1/2$, but also in the quantitative sense that give certain conditions to assure non-degeneracy that arbitrarily altering k points out of data set of N points will only cause the median to change by at most $O(\frac{k}{N})$.

3. Banach Spaces

Since we defined the median of a data $X_1, X_2, \ldots, X_N$ as the minimizer of:

$$\sum_{i=1}^{N} \|X_i - M\|.$$

We can in fact discuss the median not only in $\mathbb{R}^n$, but also in any Banach space. In particular we studied the spatial median in uniformly convex Banach spaces. A Banach space B is uniformly convex if $\forall \varepsilon \in (0,2)$, $\delta(\varepsilon) > 0$, where $\delta(\varepsilon)$ is the modulus of convexity, which is defined as:

$$\delta(\varepsilon) := inf\{1 - \frac{\|x+y\|}{2} | \|x-y\| \geq \varepsilon\}.$$

Intuitively a uniformly convex space is one whose unit ball has no flat surfaces. We chose to study the spatial median in uniformly convex Banach spaces because we believe that in non-uniformly convex Banach spaces the geometric median is not in fact quantitatively robust. However, before we discuss this in further detail we must again characterize the

median by its gradient. Since the gradient points in the direction of steepest descent we believe that the gradient of arbitrary norm is the unit dual vector.

Conjecture 3. *Let B be a Banach space with norm $\|\cdot\|$ then $\nabla\|v\| = v^*$.*

This agrees with the gradient of the Euclidean norm we calculated in section 2, which lends support to the validity of this conjecture. Assuming this holds true we know then that in a non-uniformly convex space the gradient does not necessarily change at all in certain directions, and thus the spatial median is not quantitatively robust in these spaces. However, assuming the space is uniformly convex then the gradient of the median function will change, and thus be able to absurd so of the disturbance in the median caused by moving points about. So we are led to another conjecture, which extends our previous results on the robustness of the spatial median to uniformly convex spaces.

Conjecture 4. *Consider a set of N points in a uniformly convex Banach space B bounded by $O(1)$, which is not essentially one dimensional. Then arbitrarily altering k points will cause the median to change by at most $O(\frac{k}{N})$.*

This conjecture if true would be vary useful because there are many applications such as the one discussed in the following section that might use the spatial median in spaces other than Euclidean spaces. Therefore, knowing that in uniformly convex spaces the median is in fact quantitatively robust would be very beneficial.

4. Applications to PCA

One very useful statistical method is principle component analysis, which seeks to find dependencies amongst variables by analyzing the covariance of a data set, which is defined as follows:

Definition 2 (Covariance Matrix)**.** *The **covariance matrix** of points $X_1, \ldots, X_N$ in $\mathbb{R}^n$ is*

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} X_i X_i^T.$$

Once the covariance matrix is found by it is possible to find the principle components, which point in the directions of dependencies amongst the variables. So for examples, if we consider the data set of 500 points shown bellow we clearly see that the data is distributed about some line with positive slope with some fairly small variance. The red arrows on figure 1 are the principle components, which clearly show these two dependencies. However, principle component analysis is exceptionally sensitive to outliers, and in fact one outlier can arbitrarily corrupt the statistic. [3] This sensitivity arises from the fact that when calculating the covariance of a data set we are in a sense taking an average, which as we discussed in section one is very sensitive to outliers. To overcome this we propose the idea of using the robustness of the spatial median to decrease the sensitivity of principle

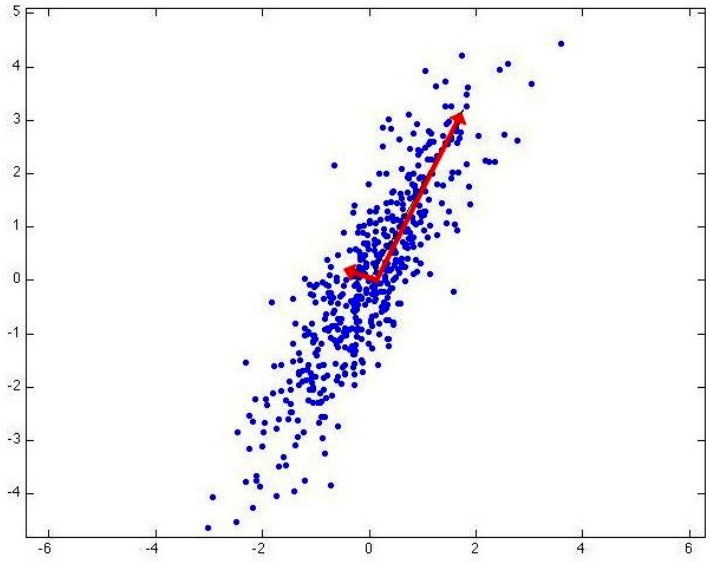

FIGURE 1. Example of Principle Component Analysis (PCA)

component analysis to outliers. In order to accomplish this we define what we an analogue to the the standard covariance matrix we call the median covariance matrix.

Definition 3 (Median Covariance Matrix)**.** *The* ***median covariance matrix*** *of points* $X_1, \ldots, X_N$ *in* $\mathbb{R}^n$ *is:*

$$\mathbf{\Sigma_M} = \text{Med}(X_i X_i^T | i \in \{1, \ldots N\}).$$

In order to test whether the using median covariance matrix would give the desired robustness to principle component analysis we created a program in MatLab capable of both calculating the spatial median in any dimension and preforming principle component analysis using the median covariance matrix. Using this program we compared the behavior of the median based principle component analysis verse the standard principle component analysis on data sets that were artificially created to show the sensitive of standard principle component analysis.

The first data set is extremely artificial consisting of 10 points distributed evenly on the line $y = x$ between -5 and 5, and one point on a line $y = -x$. Clearly the major principle component of this data should be on the $y = x$ line, which the other point being an outlier. However, when we perform standard principle component analysis seen bellow in figure 2 in red we see that the major principle component is on the $y = -x$ line with the minor principle component being on the $y = x$ line. So the one outlier was completely able to corrupt the standard principle component analysis. In comparison is we look at the results for our median based principle component analysis we see that as desired the major principle component is by far on the $y = x$ line. Thus, we see that this new form of

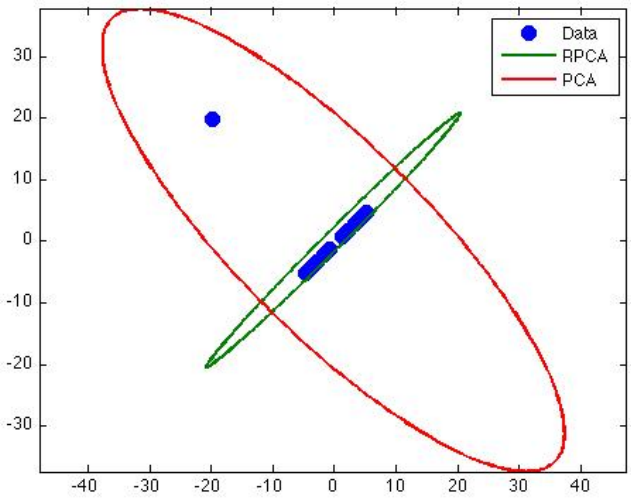

FIGURE 2. Standard PCA vs. median based PCA

principle component analysis was little effected by the presence of the outlier, and so hints that this is much more robust than standard principle component analysis.

The second example we created to test whether or not the new median based principle component analysis is more robust than standard principle component analysis is similar to first. It has two distributions one of 500 points distributed along a the line $y = x$ with some variation, and a circular distribution of 200 points roughy perpendicular to the other. When we preform standard principle component analysis on this data set we again see that the outliers complete corrupt the statistic and the major component in fact points roughly perpendicular to the distribution of 500 points. However, when we use our median based principle component analysis on this data set we see again that the outliers have little effect on the principle components. The major principle component points largely in the direction of the main distribution.

These two examples show that using the median covariance matrix instead of the standard covariance matrix seems to drastically increase the robustness of principle component analysis. Currently, the program we used to calculate the median covariance matrix is incapable of computing the median covariance matrix for high dimensional data, however, doing so is computationally tractable and so should be doable. This gives us hope that this could in fact prove to be a viable more robust alternative to principle component analysis.

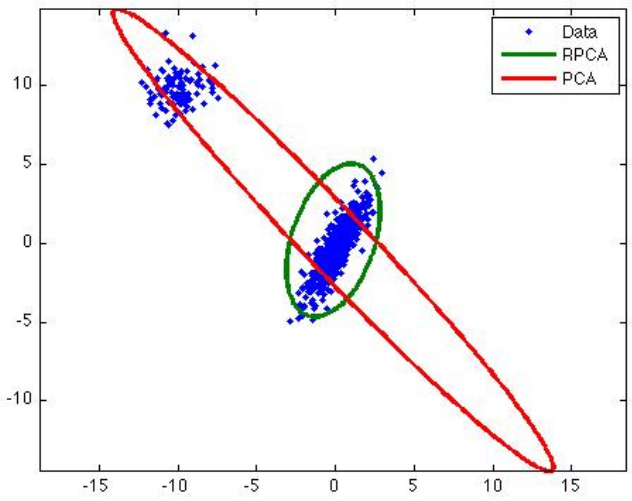

FIGURE 3. Standard PCA vs. median based PCA

5. APPENDIX

As we discussed in Section 4 we created two programs in order to computational explore whether using the median covariance matrix did in fact produce a more robust form of principle component analysis. The first program is written in MatLab and calculates the spatial median of a set of data points by minimizing the distance between each point. In order to accomplish this program uses one of MatLab's built-in minimization functions to search for said minimum.

```
%OVERVIEW: This function takes a n x p data matrix where n is the number of
%points, and p is dimmesionion the points lie in, and then calculates the
%spatial median for these data points.

%INPUTS: In order to run this function one must first define pne variable,
%A.  A is a n x p data matrix where each of the n rows of the matrix
%represent a point in that lies in R^p.

%OUTPUTS: This function will return two outputs, X and fval.  X will be the
%spatial median for the data set A, and fval will be the median function
%evaluated at X.  *SEE HOW IT WORKS*

%STRUCTURE: This program is broken into two functions.  The first function
%entitled 'runMedian' is the minimization part of the funcition, and the
%second function 'Median' is the actual median function.  The reason for
```

```
%this structure is to pass the extra parameter A needed for the median
%function.

%HOW TO RUN: In order to run this program first define the data matrix A,
%and the initialation value X0.  After having defined these use the
%following line to calculate the median:
% [X, fval] = runMedian(X0, A)

%HOW IT WORKS: This program utilizes the fact that the L-1 median is the
%minimizer of the function:
%               f(X) = SUM(|A_{i} - X|)
%In order to do this first the Median function breaks the matrix A in its
%indvidual data points.  It does this by the arrayfun and A(nc,:) commands.
%Once this is done the program calculates the distance that X is from each
%data point by doing: norm(X-A(nc,:)).  Note the norm can be changed:
%
%      n = norm(X,2) returns the 2-norm of X.
%      n = norm(X) is the same as n = norm(X,2).
%      n = norm(X,1) returns the 1-norm of X.
%      n = norm(X,Inf) returns the infinity norm of X.
%      n = norm(X,'fro') returns the Frobenius norm of X.
%
%After doing this the Median function sums up all these distances, and
%returns this values as Y.  After this the second function runMedian kicks
%in and minimizes the vedian function.  This is done using either the
%fminsearch or the fminunc commands.

%GRAPHICS: If you are using either two or three dimensional data the data
%and median can be plotted in a scattering plot by uncommenting the last
%two lines of the runvedian program. ***NOTE: To plot in 3-dimensions you
%must comment out the line labeled 2D Plot and uncomment the one
%labeled 3D Plot.***

function[X,fval] = runMedian(A)
  X0 = mean(A);
  [X,fval] = fminunc(@(X) Median(X,A), X0);
  %plot(A(:,1),A(:,2), 'r+', X(:,1),X(:,2), 'bX') %2D Plot
  %plot3(A(:,1),A(:,2),A(:,3), 'r+', X(:,1),X(:,2),X(:,3), 'bX') %3D Plot
end
function[Y] = Median(X,A)
  Y = sum(arrayfun(@(nc) norm(X-(A(nc,:))),1:size(A,1)));
end
```

This second program, also written in MatLab, utilizes the first program to calculate the median covariance matrix and then calculates its principle components. In addition to doing this this program is capable of preforming standard principle component analysis on the data set so that the two methods can be compared. Also if the data set resides in $\mathbb{R}^2$ then this program can graphically present the data as well as the principle components for both the standard principle component analysis and the median based principle component analysis. In low dimensions this program is capable of handling fairly large data sets, however, as the dimension grows beyond four or five the speed of this program is drastically reduced. Calculating the geometric median is a convex optimization problem, and so is computationally tractable, thus, we believe that our optimization procedure use in this and the previous program are not optimal, and that with some work this program can be made to run efficiently even for high dimensions.

```
%OVERVIEW: This function takes a n x p data matrix where n is the number of
%trials, and p is the number of parameters measured, and then performs a
%type of princpled component analysis that is based on the spatial median,
%which hopefully is more robust than classical PCA.

%INPUTS: In order to run this function one must first define one variable,
%A. A is a n x p data matrix where each of the n rows of the matrix
%represent a point in that lies in R^p.

%OUTPUTS: This function will return six outputs, V, D, COEFF, SCORE, X and
%fval.
%       - V = Is the matrix that contains the eigenvectors of the M-Covariance
%       matrix.  Note: The eigenvectors are represented as columns.
%       - D = Is a diagonal matrix that contains the eigenvalues for the
%       the eigenvectors in matrix V.
%       - COEFF = Is a p x p matrix that contains the coefficeints for the
%       princple components found using classical PCA.
%       - SCORE = Is a matrix that contains for the scores found using
%       classical PCA. Rows of SCORE correspond to observations, columns to
%       components.
%       - X = Is the M-Covariance matrix, that is the covariance matrix
%       found using the new RPCA
%       fval = Is the value of the median function evaluated at X.

%STRUCTURE: This program is broken into two functions.  The first function
%entitled 'runRPCA' is the minimization part of the funcition and performs
%the RPCA analysis.  The second function 'RPCA' is the actual median function.  The reason for
%this structure is to pass the extra parameter A needed for the median
%function.
```

```
%HOW TO RUN: In order to run this program first define the data matrix A,
%and the initialation value X0.  After having defined these use the
%following line to calculate the median:
% [X, fval] = runRPCA(X0, A)

%MATHEMATICAL BACKGROUND: Classical PCA is an incredibly useful statsitical
%tool for analyzing high dimensional data, however, it is incredibly
%sensative to outliers and poor data.  We feel that the cause of
%sensativity is due to the fact that in clasical PCA the covariance matrix
%is the mean of each covariance matrix.  In order to overcome this and make
%PCA more robust our RPCA use seeks to find the covariance matrix by using
%the spatial median instead of mean since the spatial median is robust when
%it comes to outliers.

%HOW IT WORKS: This program utilizes the fact that the L-1 median is the
%minimizer of the function:
%               f(X) = SUM(|A_{i} - X|)
%In order to do this first the RPCA function breaks the matrix A in its
%indvidual data points.  It does this by the arrayfun and A(nc,:) commands.
%Once this is done the program normalizes the data by substracting off the
%means from each point.  Then the program calculates the "distance" that X
%is from each covariance matrix by doing:
%       norm(X-((A(nc,:)-mean(A))'*(A(nc,:)-mean(A))).  \
%Note the norm can be changed:
%
%      n = norm(X,2) returns the 2-norm of X.
%      n = norm(X) is the same as n = norm(X,2).
%      n = norm(X,1) returns the 1-norm of X.
%      n = norm(X,Inf) returns the infinity norm of X.
%      n = norm(X,'fro') returns the Frobenius norm of X.
%
%After doing this the RPCA function sums up all these distances, and
%returns this values as Y.  After this the second function runRPCA kicks
%in and minimizes the RPCA function.  This is done using either the
%fminsearch or the fminunc commands.  This minimization thus produes our
%M-Covariance matrix.
%
%Now that this is done the runRPCA begins analyzing the M-Covariance
%matrix. It does this by calculating the eigenvalues and eigenvectors for
%the M-Covariance matrix.  It then performs classical PCA on A so that one
%can compare how our RPCA compares to classical PCA.

%GRAPHICS: If you are using either two or three dimensional data the data
```

```
%and median can be plotted in a scattering plot by uncommenting the last
%lines of the runRPCA program. There are roughly 3 graphics options, the 2D
%case, the 3D case, and the biplot for 2D case.
%   2D CASE: This graphic is created by uncommenting lines 94-96.  When this
%   is done this will create a scatterplot of the data found in A.  Then
%   it will plot the first to eigenvectors of the M-Covariance matrix, with
%   the length of the arrow proporanate to the eigenvector of the
%   respective eigenvector.
%   3D CASE: This graphic is created by uncommenting lines 98-100.  It does the
%   same thing as the 2D CASE, however in 3D.
%   Biplot: This creates a 2-D biplot of data, and is turned on by
%   uncommenting line 97.

function[X,fval] = runRPCA(A)
  T = runMedian(A);
  B = A - repmat(T,size(A,1),1); %ADJUSTS DATA TO 0 MEDIAN
  X0 = cov(A);
  [X,fval] = fminunc(@(X) RPCA(X,B), X0);
  [V,D] = eig(X) %#ok<NOPRT>
  [V1, D1] = eig(X0) %#ok<NOPRT>
  [COEFF] = princomp(A) %#ok<NASGU,NOPRT>
  %plot(B(:,1),B(:,2), 'b*','MarkerSize',10) %2D Plot
  %arrow([0 0],2.2* D1(1,1)*[V1(1,1) V1(2,1)])%2D Plot
  %arrow([0 0], D1(2,2)*[V1(1,2) V1(2,2)])%2D Plot
  %ellipse(1.7*D(1,1),1.7*D(2,2),atan(V(2,1)/V(1,1)),T(1,1),T(1,2),'g')
  %ellipse(.7*D1(1,1),.7*D1(2,2),atan(V1(2,1)/V1(1,1)),T(1,1),T(1,2),'r')
  %axis('equal')
  %biplot(COEFF,'scores',A)
  %plot3(B(:,1),B(:,2),B(:,3) 'r+') %3D Plot
  %arrow([0 0 0],D(1,1)*[V(1,1) V(2,1) V(3,1)],D(2,2)*[V(1,2) V(2,2)
  %V(3,2)],D(3,3)*[V(1,3) V(2,3) V(3,3)]) $3D Plot
end
function[Y] = RPCA(X,B)
  Y = sum(arrayfun(@(nc) norm(X-((B(nc,:))'*(B(nc,:)))),1:size(B,1)));
end
```

6. References

1. Weber, Alfred (1909). ber den Standort der Industrien, Erster Teil: Reine Theorie des Standortes. Tbingen: Mohr

2. Rousseeuw, P. (1985), Multivariate Estimation with High Breakdown Point, in Math-

ematical Statistics and Applications, edited by W. Grossmann, G. Pflug, I. Vincze, and W. Wertz, Reidel Publishing Company, Dordrecht (co-published with Akad?emiai Kiad?o, Budapest), 283297.

3. Candes, E. Li, X. Ma, Y. Wright, J. (2009), "Robust Principle Component Analysis?"