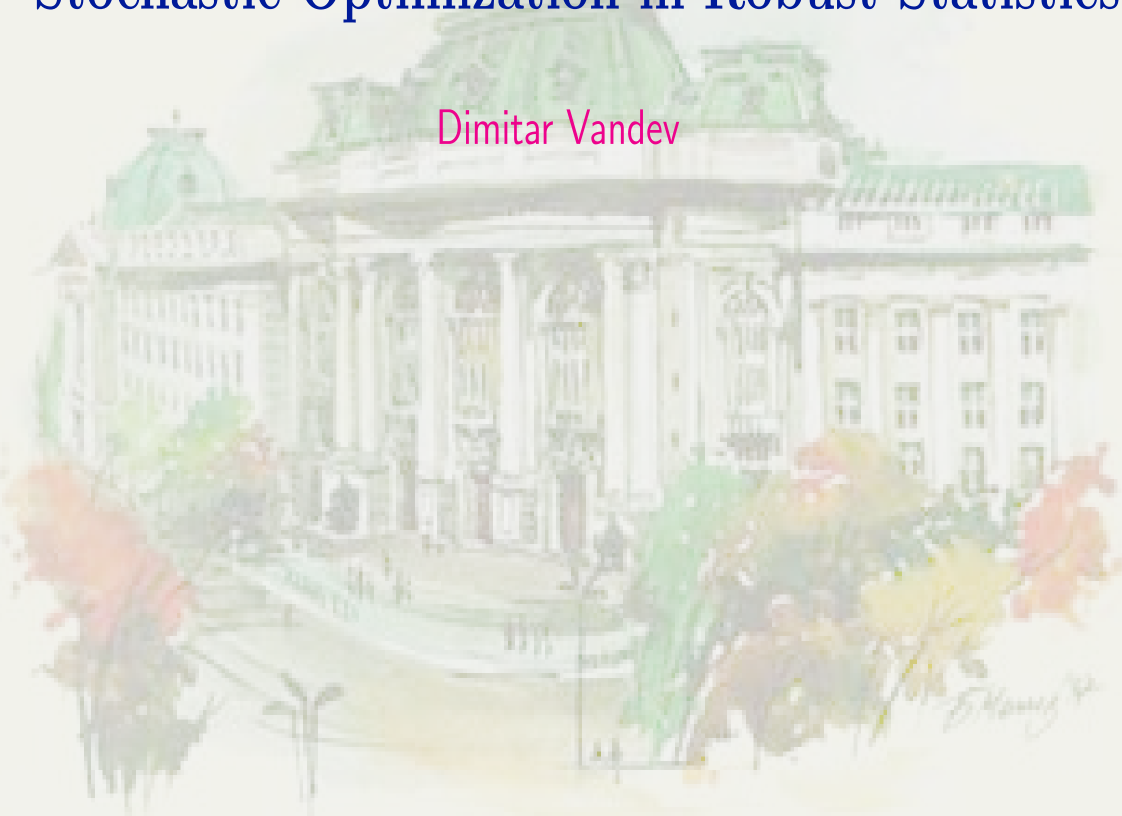
Софийски Университет „Св.Климент Охридски“

Stochastic Optimization in Robust Statistics

Dimitar Vandev



Софийски Университет „Св.Климент Охридски“

Stochastic Optimization in Robust Statistics

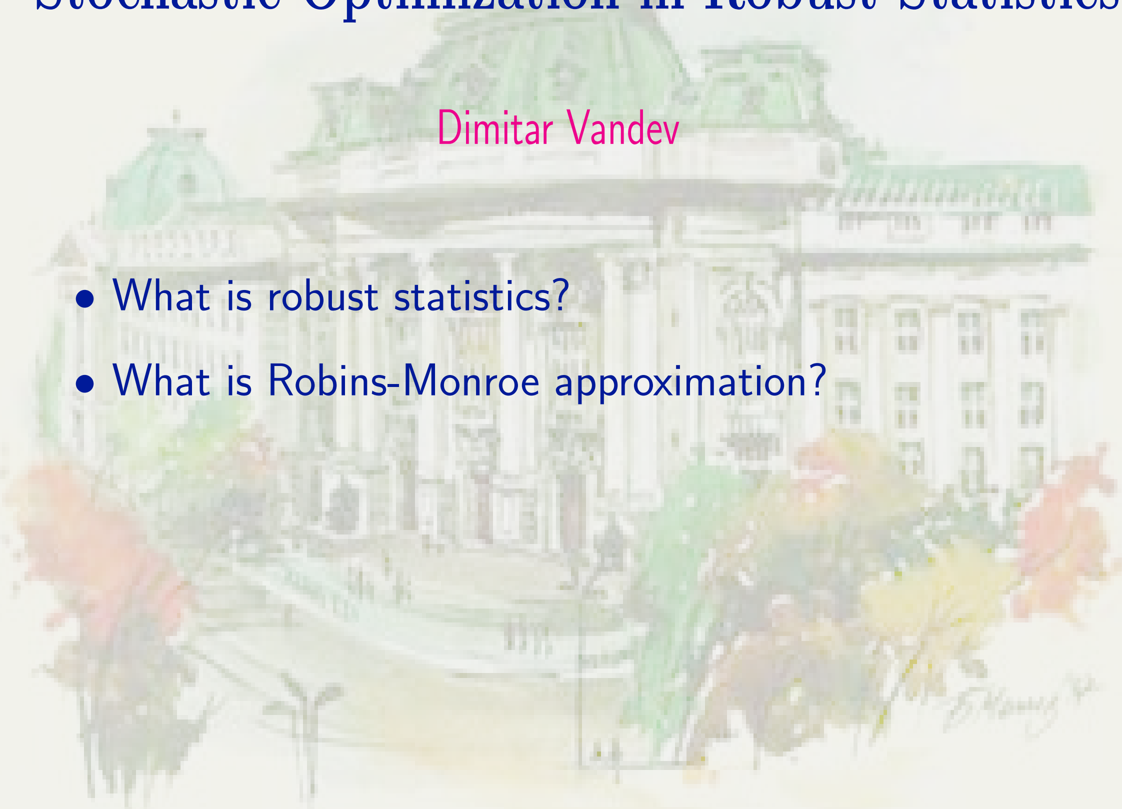
Dimitar Vandev

- What is robust statistics?

Софийски Университет „Св.Климент Охридски“

Stochastic Optimization in Robust Statistics

Dimitar Vandev



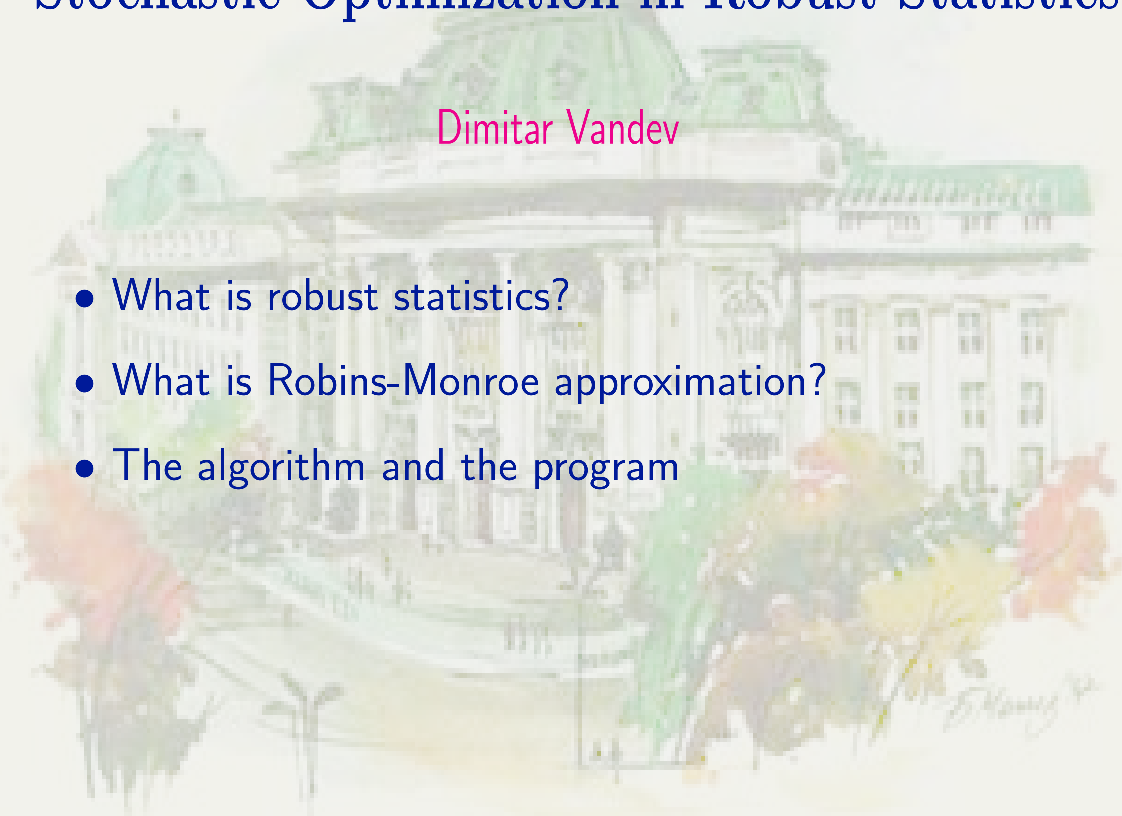
- What is robust statistics?
- What is Robins-Monroe approximation?

Home Page
Title Page
Go Back
Full Screen
Close
Quit

Софийски Университет „Св.Климент Охридски“

Stochastic Optimization in Robust Statistics

Dimitar Vandev



- What is robust statistics?
- What is Robins-Monroe approximation?
- The algorithm and the program

Home Page
Title Page
Go Back
Full Screen
Close
Quit

Софийски Университет „Св.Климент Охридски“

Stochastic Optimization in Robust Statistics

Dimitar Vandev

- What is robust statistics?
- What is Robins-Monroe approximation?
- The algorithm and the program
- Successful Examples in Regression

Софийски Университет „Св.Климент Охридски“

Stochastic Optimization in Robust Statistics

Dimitar Vandev

- What is robust statistics?
- What is Robins-Monroe approximation?
- The algorithm and the program
- Successful Examples in Regression
- The problems with covariance

1. Robust statistics

For modeling gross errors and outliers in the sample, the most popular is the Tukey supermodel (Tukey, 1960) based on the Gaussean low:

$$\mathcal{F} = \left\{ F : F(x) = (1-\varepsilon)\Phi(x) + \varepsilon\Phi(\frac{x-\theta}{k}), \quad 0 < \varepsilon < 1 \right\}.$$

1. Robust statistics

For modeling gross errors and outliers in the sample, the most popular is the Tukey supermodel (Tukey, 1960) based on the Gaussean low:

$$\mathcal{F} = \left\{ F : F(x) = (1-\varepsilon)\Phi(x) + \varepsilon\Phi(\frac{x-\theta}{k}), \quad 0 < \varepsilon < 1 \right\}.$$

Huber (Huber, 1964) considered more general model

$$\mathcal{F} = \left\{ F : F(x) = (1-\varepsilon)F_0(x) + \varepsilon H(x) \right\},$$

where F_0 is some given distribution (the ideal model) and $H(x)$ is an arbitrary continuous distribution (contamination). So ε is the expected percent of errors in the data.

LMS and LTS

The multiple regression is probably most used statistical procedure in the statistics. Consider the model

$$y_i = x_i^T \theta + \varepsilon_i,$$

where y_i is an observed response, x_i is a $(p \times 1)$–dimensional vector of explanatory variables and θ is a $(p \times 1)$ vector of unknown parameters. Classically ε_i, $i = 1, \ldots, n$ are assumed to be i.i.d. $N(0, \sigma^2)$, for some $\sigma^2 > 0$.

Denote by $r_i = y_i - x_i^T \theta$ the residuals of the model.

LMS and LTS

The multiple regression is probably most used statistical procedure in the statistics. Consider the model

$$y_i = x_i^T \theta + \varepsilon_i,$$

where y_i is an observed response, x_i is a $(p \times 1)$–dimensional vector of explanatory variables and θ is a $(p \times 1)$ vector of unknown parameters. Classically ε_i, $i = 1, \ldots, n$ are assumed to be i.i.d. $N(0, \sigma^2)$, for some $\sigma^2 > 0$.

Denote by $r_i = y_i - x_i^T \theta$ the residuals of the model.

The usual Least Squares Estimate (LSE) as proposed by Gauss:

$$LSE(r_1, \ldots, r_n) = \operatorname*{argmin}_{\theta} \sum_{i=1}^{n} r_i^2.$$

Suppose now that the sample is contaminated.

The usual Least Squares Estimate (LSE) as proposed by Gauss:

$$LSE(r_1, \ldots, r_n) = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} r_i^2.$$

Suppose now that the sample is contaminated.

The LMS (Least Median of Squares) and LTS (Least Trimmed Squares) estimators were proposed by Rousseeuw (Rousseeuw, 1984) as robust alternatives of the LSE

$$\mathsf{LMS}(r_1, \ldots, r_n) \;=\; \underset{\theta}{\operatorname{argmin}} \operatorname{med}\{r_i^2, i = 1, ..., n\},$$

The usual Least Squares Estimate (LSE) as proposed by Gauss:

$$LSE(r_1, \ldots, r_n) = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} r_i^2.$$

Suppose now that the sample is contaminated.

The LMS (Least Median of Squares) and LTS (Least Trimmed Squares) estimators were proposed by Rousseeuw (Rousseeuw, 1984) as robust alternatives of the LSE

$$\begin{aligned} \text{LMS}(r_1, \ldots, r_n) &= \underset{\theta}{\operatorname{argmin}} \operatorname{med}\{r_i^2, i = 1, ..., n\}, \\ \text{LTS}(k)(r_1, \ldots, r_n) &= \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{k} r_{\nu(i,\theta)}^2, k > n/2. \end{aligned}$$

The usual Least Squares Estimate (LSE) as proposed by Gauss:

$$LSE(r_1,\ldots,r_n) = \operatorname*{argmin}_{\theta} \sum_{i=1}^{n} r_i^2.$$

Suppose now that the sample is contaminated.

The LMS (Least Median of Squares) and LTS (Least Trimmed Squares) estimators were proposed by Rousseeuw (Rousseeuw, 1984) as robust alternatives of the LSE

$$\text{LMS}(r_1,\ldots,r_n) = \operatorname*{argmin}_{\theta} \operatorname{med}\{r_i^2, i=1,...,n\},$$

$$\text{LTS}(k)(r_1,\ldots,r_n) = \operatorname*{argmin}_{\theta} \sum_{i=1}^{k} r_{\nu(i,\theta)}^2, k > n/2.$$

Here $\nu(i,\theta)$ is a permutation of the indices, such that $r_{\nu(i,\theta)}^2 \leq r_{\nu(i+1,\theta)}^2$. Thus the idea was to minimize the sum of squares using smallest residuals only.

Robustified Maximum Likelihood

(Neykov and Neytchev, 1990) proposed to replace in these estimators (LMS and LTS) the squared residuals with -log likelihood´s of the individual observations and thus to obtain robustified likelihood.

Robustified Maximum Likelihood

(Neykov and Neytchev, 1990) proposed to replace in these estimators (LMS and LTS) the squared residuals with -log likelihood´s of the individual observations and thus to obtain robustified likelihood.

Let the observations $x_1, x_2, \ldots, x_n$ be generated by an arbitrary probability density function $\psi(x, \theta)$ with unknown vector parameter θ.

$$\mathsf{LME}(k) \; = \; \underset{\theta}{\operatorname{argmin}}\{-\log \psi(x_{\nu(k,\theta)}, \theta)\},$$

Robustified Maximum Likelihood

(Neykov and Neytchev, 1990) proposed to replace in these estimators (LMS and LTS) the squared residuals with - log likelihood´s of the individual observations and thus to obtain robustified likelihood.

Let the observations $x_1, x_2, \ldots, x_n$ be generated by an arbitrary probability density function $\psi(x, \theta)$ with unknown vector parameter θ.

$$\text{LME}(k) = \operatorname*{argmin}_{\theta} \{-\log \psi(x_{\nu(k,\theta)}, \theta)\}, \tag{1}$$

$$\text{LTE}(k) = \operatorname*{argmin}_{\theta} \sum_{i=1}^{k} \{-\log \psi(x_{\nu(i,\theta)}, \theta)\}. \tag{2}$$

Thus the idea was to maximize the likelihood over the best k observations (with largest likelihood).

2. Stochastic Approximation

The famous Robins-Monro procedure (Robins and S.Monro, 1951) when applied to the problem of minimizing the function $F(\theta)$ consists in the following. Let start with some $\theta = \theta_0$. Let now calculate the gradient $grad(F(\theta))$ at this point. It may be disturbed by some random variable with zero expectation.

2. Stochastic Approximation

The famous Robins-Monro procedure (Robins and S.Monro, 1951) when applied to the problem of minimizing the function $F(\theta)$ consists in the following. Let start with some $\theta = \theta_0$. Let now calculate the gradient $grad(F(\theta))$ at this point. It may be disturbed by some random variable with zero expectation.

$$\theta_{i+1} = \theta_i - \gamma_i * \frac{grad(F(\theta_i))}{||grad(F(\theta_i))||}.$$

2. Stochastic Approximation

The famous Robins-Monro procedure (Robins and S.Monro, 1951) when applied to the problem of minimizing the function $F(\theta)$ consists in the following. Let start with some $\theta = \theta_0$. Let now calculate the gradient $grad(F(\theta))$ at this point. It may be disturbed by some random variable with zero expectation.

$$\theta_{i+1} = \theta_i - \gamma_i * \frac{grad(F(\theta_i))}{||grad(F(\theta_i))||}.$$

The sequence $\{\gamma_i, i = 1, 2, ...\}$ is chosen to satisfy 2 relations:

$$\sum_{i=1}^{\infty} \gamma_i^2 < \infty,$$
$$\sum_{i=1}^{\infty} \gamma_i = \infty.$$

The Proposed Algorithm

- Step 0. Set maxn, set i=1, set δ.

The Proposed Algorithm

- Step 0. Set maxn, set i=1, set δ.
- Step 1. Chose at random 10 indexes among the numbers from 1 to n. Calculate these 10 functions. Sort their values.

The Proposed Algorithm

- Step 0. Set maxn, set i=1, set δ.
- Step 1. Chose at random 10 indexes among the numbers from 1 to n. Calculate these 10 functions. Sort their values.
- Step 2. Chose the value corresponding to the desired proportion ($j/10 = k/n$) and the function which produces that value (say f).

The Proposed Algorithm

- Step 0. Set maxn, set i=1, set δ.
- Step 1. Chose at random 10 indexes among the numbers from 1 to n. Calculate these 10 functions. Sort their values.
- Step 2. Chose the value corresponding to the desired proportion $(j/10 = k/n)$ and the function which produces that value (say f).
- Step 3. Calculate the normalized gradient $D(f)$ of f.

The Proposed Algorithm

- Step 0. Set maxn, set i=1, set δ.
- Step 1. Chose at random 10 indexes among the numbers from 1 to n. Calculate these 10 functions. Sort their values.
- Step 2. Chose the value corresponding to the desired proportion $(j/10 = k/n)$ and the function which produces that value (say f).
- Step 3. Calculate the normalized gradient $D(f)$ of f.
- Step 4. Set $B := B - D(f).\delta/i$. Set $i = i + 1$. If $i \leq$ maxn then Goto step 1.

The programs

```
function [theta] = soaml(x,theta0,FUN,pr,delta,iter)
[n,m]=size(x);
theta=theta0;
for k=1:iter
    gama=delta/k;            %   new gama
    J=round(ones(kkk,1)/2+rand(kkk,1)*n); %   10 random in (1:n) numbers
    eval(['[Y,X]=' FUN '(x(J,:),theta);']);% residuals, gradient
    [dum,list]=sort(Y);      %   sort 25 values
%=================LME or LTE ====================
    jj=list(pr,1);         %  jj=list(1:pr,1);
    s=X(jj,:)';            %  s=sum(X(jj,:))';
%===============================================
    w=sqrt(s'*s);
    theta=theta-s*(gama/w);
end
```

The programs

```
function [theta] = soaml(x,theta0,FUN,pr,delta,iter)
[n,m]=size(x);
theta=theta0;
for k=1:iter
    gama=delta/k;           %   new gama
    J=round(ones(kkk,1)/2+rand(kkk,1)*n); %   10 random in (1:n) numbers
    eval(['[Y,X]=' FUN '(x(J,:),theta);']);% residuals, gradient
    [dum,list]=sort(Y);      %   sort 25 values
%================LME or LTE ===================
    jj=list(pr,1);        %  jj=list(1:pr,1);
    s=X(jj,:)';           %  s=sum(X(jj,:))';
%===============================================
    w=sqrt(s'*s);
    theta=theta-s*(gama/w);
end
```

Here are some examples of usage

Location

```
function [Y,X]=gradmea(x,a)
[n,m]=size(x);
aa=a';
X=x-aa(ones(n,1),:);
Y= diag(X*X');
X=-2*X.*x;
```

The programs

```
function [theta] = soaml(x,theta0,FUN,pr,delta,iter)
[n,m]=size(x);
theta=theta0;
for k=1:iter
    gama=delta/k;            %   new gama
    J=round(ones(kkk,1)/2+rand(kkk,1)*n); %   10 random in (1:n) numbers
    eval(['[Y,X]=' FUN '(x(J,:),theta);']);% residuals, gradient
    [dum,list]=sort(Y);      %   sort 25 values
%==================LME or LTE ====================
    jj=list(pr,1);         %  jj=list(1:pr,1);
    s=X(jj,:)';            %  s=sum(X(jj,:))';
%================================================
    w=sqrt(s'*s);
    theta=theta-s*(gama/w);
end
```

Here are some examples of usage

Location

```
function [Y,X]=gradmea(x,a)
[n,m]=size(x);
aa=a';
X=x-aa(ones(n,1),:);
Y= diag(X*X');
X=-2*X.*x;
```

Regression

```
function [Y,X]=gradreg(x,a)
[n,m]=size(x);
xx=[ones(n,1),x(:,2:m)];
Y=x(:,1)-xx*a;
X= -2*(Y(:,ones(1,m)).*xx);
Y=Y.*Y;
```

The programs

```
function [theta] = soaml(x,theta0,FUN,pr,delta,iter)
[n,m]=size(x);
theta=theta0;
for k=1:iter
    gama=delta/k;             %   new gama
    J=round(ones(kkk,1)/2+rand(kkk,1)*n); %   10 random in (1:n) numbers
    eval(['[Y,X]=' FUN '(x(J,:),theta);']);% residuals, gradient
    [dum,list]=sort(Y);        %   sort 25 values
%=================LME or LTE ===================
    jj=list(pr,1);          %  jj=list(1:pr,1);
    s=X(jj,:)';             %  s=sum(X(jj,:))';
%===============================================
    w=sqrt(s'*s);
    theta=theta-s*(gama/w);
end
```

Here are some examples of usage

Location

```
function [Y,X]=gradmea(x,a)
[n,m]=size(x);
aa=a';
X=x-aa(ones(n,1),:);
Y= diag(X*X');
X=-2*X.*x;
```

Regression

```
function [Y,X]=gradreg(x,a)
[n,m]=size(x);
xx=[ones(n,1),x(:,2:m)];
Y=x(:,1)-xx*a;
X= -2*(Y(:,ones(1,m)).*xx);
Y=Y.*Y;
```

$N(\mu,\sigma)$ in R^1

```
function [Y,X]=gradnor(x,a)
[n,dum]=size(x);
mu=a(ones(n,1),1);
si=exp(a(2,1));
Y=(x-mu)/si;
X=[-Y/(2*si),(ones(n,1)-Y.*Y)];
Y=Y.*Y/2+ones(n,1)*a(2);
```

3. Regression

The first model was chosen to illustrate the robust properties of the used version of maximum likelihood.

3. Regression

The first model was chosen to illustrate the robust properties of the used version of maximum likelihood.
The response Y is generated by the following model:

$$y = 5 - 2 * x + e.$$

3. Regression

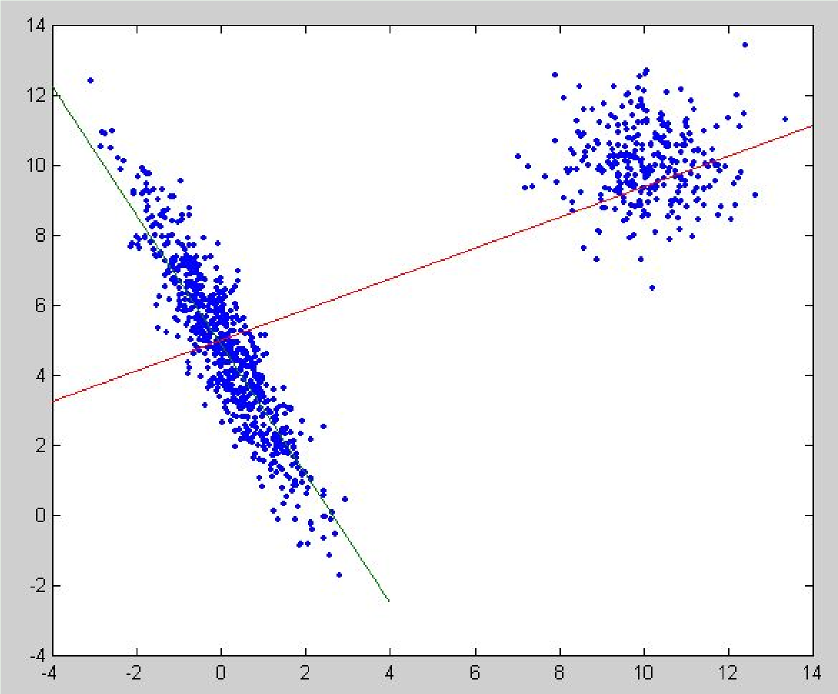
The first model was chosen to illustrate the robust properties of the used version of maximum likelihood.
The response Y is generated by the following model:

$$y = 5 - 2 * x + e.$$

Here e is a standard normal random variable. The sample consists of 1000 observations. It was corrupted by destroying 30% of the observations. A reasonable estimations is achieved after 150 iterations despite of the large number of outliers. The algorithm was used with fixed number of iterations equal to 150 and $\delta = 10$.

Bellow one solution is presented for the estimator $6/10$. For a comparison the unique least squares solution is also plotted in red.

Bellow one solution is presented for the estimator $6/10$. For a comparison the unique least squares solution is also plotted in red.



4. Multiple Regression

The Model

$$y = 2 - 2 * x_1 + 5 * x_2 - 5 * x_3 + x_4 + e.$$

4. Multiple Regression

The Model

$$y = 2 - 2 * x_1 + 5 * x_2 - 5 * x_3 + x_4 + e.$$

Simulation

The aim was to test the performance of different estimators of the same kind (LME) when the percent of contamination changes.

4. Multiple Regression

The Model

$$y = 2 - 2 * x_1 + 5 * x_2 - 5 * x_3 + x_4 + e.$$

Simulation

The aim was to test the performance of different estimators of the same kind (LME) when the percent of contamination changes.

In this case we each time generate totally new data. This was repeated 100 times in order to estimate the variance.

Simulation Results

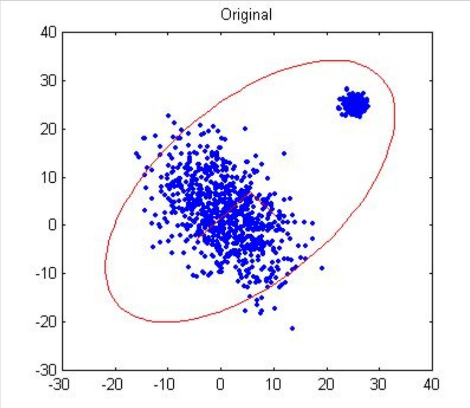
Cont.	Est.	$a_0 = 2$	$a_1 = -2$	$a_2 = 5$	$a_3 = -5$	LME
100	9/10	1.9235	-1.9421	4.9492	-4.8944	3.3777
		.1149	.1569	.1249	.1265	2.4862
	8/10	1.9644	-1.9821	4.9029	-4.9164	1.2839
		.0990	.0987	.1401	.1136	.1512
	7/10	1.9390	-2.0412	4.9467	-4.8380	1.1343
		.1596	.1834	.1959	.1705	.2168
	6/10	1.9823	-1.9756	4.7355	-4.7443	.9773
		.2313	.2328	.3107	.2601	.2017
200	8/10	1.9664	-2.0136	4.7889	-4.7541	5.9930
		.1534	.1828	.1808	.2410	3.7446
	7/10	1.9103	-1.9337	4.9010	-4.8629	1.3670
		.2338	.2833	.2781	.2783	.5853
	6/10	1.9484	-1.9812	4.8867	-4.9113	1.0957
		.1811	.2229	.2409	.1701	.2407
300	7/10	1.7643	-1.7374	4.4975	-4.5186	7.7961
		.4012	.3970	.6973	.7480	4.5630
	6/10	1.8873	-1.8956	4.8093	-4.7899	1.5153
		.3159	.2421	.5369	.4467	.8834
400	6/10	1.5886	-1.6556	4.2696	-4.1614	9.5648
		.5058	.4803	.7968	.9176	4.7157

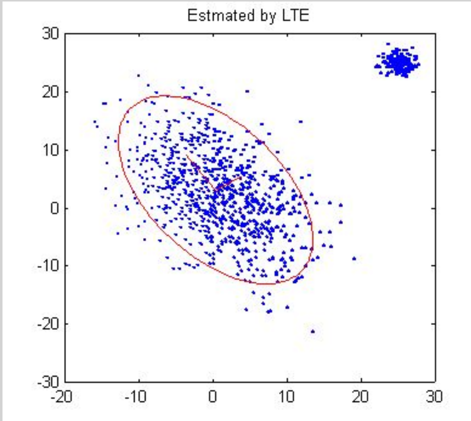
Simulation Results

Cont.	Est.	$a_0 = 2$	$a_1 = -2$	$a_2 = 5$	$a_3 = -5$	LME
100	9/10	1.9235	-1.9421	4.9492	-4.8944	3.3777
		.1149	.1569	.1249	.1265	2.4862
	8/10	1.9644	-1.9821	4.9029	-4.9164	1.2839
		.0990	.0987	.1401	.1136	.1512
	7/10	1.9390	-2.0412	4.9467	-4.8380	1.1343
		.1596	.1834	.1959	.1705	.2168
	6/10	1.9823	-1.9756	4.7355	-4.7443	.9773
		.2313	.2328	.3107	.2601	.2017
200	8/10	1.9664	-2.0136	4.7889	-4.7541	5.9930
		.1534	.1828	.1808	.2410	3.7446
	7/10	1.9103	-1.9337	4.9010	-4.8629	1.3670
		.2338	.2833	.2781	.2783	.5853
	6/10	1.9484	-1.9812	4.8867	-4.9113	1.0957
		.1811	.2229	.2409	.1701	.2407
300	7/10	1.7643	-1.7374	4.4975	-4.5186	7.7961
		.4012	.3970	.6973	.7480	4.5630
	6/10	1.8873	-1.8956	4.8093	-4.7899	1.5153
		.3159	.2421	.5369	.4467	.8834
400	6/10	1.5886	-1.6556	4.2696	-4.1614	9.5648
		.5058	.4803	.7968	.9176	4.7157

5. Estimation of mean and covariance

Before explaining difficulties let us present one not very successful example of two-dimensional estimate of mean and the covariance:

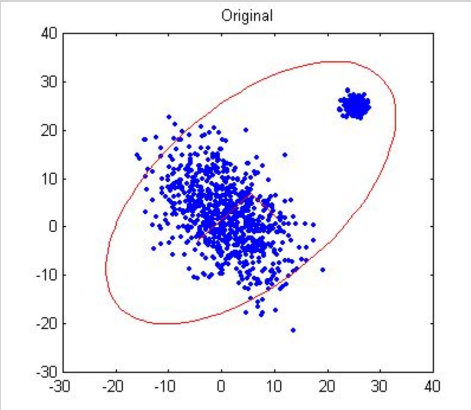


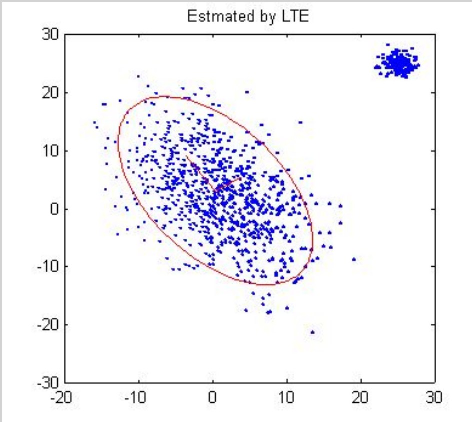


Original mean	0.6242	2.5444
Estimated mean	0.9204	3.0970
Original Cova	37.0107	-20.4700
	-20.4700	51.5504
Estimated Cova	35.1363	-10.3004
	-10.3004	43.6298

5. Estimation of mean and covariance

Before explaining difficulties let us present one not very successful example of two-dimensional estimate of mean and the covariance:





Original mean	0.6242	2.5444
Estimated mean	0.9204	3.0970
Original Cova	37.0107	-20.4700
	-20.4700	51.5504
Estimated Cova	35.1363	-10.3004
	-10.3004	43.6298

The multidimensional $N(\mu, \Sigma)$

Here the main problem consisted in calculation of the gradient of $Q = -\log L(x, m, \Sigma)$:

$$Q = \log \det(\Sigma)^{1/2} + (x - \mu)'\Sigma^{-1}(x - \mu).$$

The multidimensional $N(\mu, \Sigma)$

Here the main problem consisted in calculation of the gradient of $Q = -\log L(x, m, \Sigma)$:

$$Q = \log \det(\Sigma)^{1/2} + (x - \mu)'\Sigma^{-1}(x - \mu). \qquad (3)$$

Let denote $M = \Sigma^{-1}$. Then

$$\frac{dQ}{dM} = -M^{-1} + (x - \mu)(x - \mu)'.$$

The multidimensional $N(\mu, \Sigma)$

Here the main problem consisted in calculation of the gradient of $Q = -\log L(x, m, \Sigma)$:

$$Q = \log \det(\Sigma)^{1/2} + (x-\mu)'\Sigma^{-1}(x-\mu). \qquad (3)$$

Let denote $M = \Sigma^{-1}$. Then

$$\frac{dQ}{dM} = -M^{-1} + (x-\mu)(x-\mu)'. \qquad (4)$$

Let us replace $M = \exp(L)$ as in the univariate case and try to use the formal relation

$$\frac{dQ}{dL} = \frac{dQ}{dM} \bigotimes \frac{dM}{dL}.$$

Consider the standard definition of $\exp(L)$

$$M = \exp L = I + L + L^2/2 + L^3/3! + \dots$$

Consider the standard definition of $\exp(L)$

$$M = \exp L = I + L + L^2/2 + L^3/3! + \dots \qquad (5)$$

The question now is how to represent $\frac{dM}{dL}$.We propose the following approximation of this $(m \times m)^2$ tensor:

$$\frac{dM}{dL} = (I + L/18) \bigotimes (I + L/18)$$

Consider the standard definition of $\exp(L)$

$$M = \exp L = I + L + L^2/2 + L^3/3! + \dots \qquad (5)$$

The question now is how to represent $\frac{dM}{dL}$.We propose the following approximation of this $(m \times m)^2$ tensor:

$$\frac{dM}{dL} = (I + L/18) \bigotimes (I + L/18) \qquad (6)$$

Thus we come to the result:

$$\frac{dQ}{dL} = (I+L/18)'((x-\mu)(x-\mu)'-M^{-1})(I+L/18)$$

Consider the standard definition of $\exp(L)$

$$M = \exp L = I + L + L^2/2 + L^3/3! + \dots \qquad (5)$$

The question now is how to represent $\frac{dM}{dL}$.We propose the following approximation of this $(m \times m)^2$ tensor:

$$\frac{dM}{dL} = (I + L/18) \bigotimes (I + L/18) \qquad (6)$$

Thus we come to the result:

$$\frac{dQ}{dL} = (I+L/18)'((x-\mu)(x-\mu)' - M^{-1})(I+L/18) \quad (7)$$

Note that we are not sure how exact is this approximation.

The Simulation Results

Means

Original	18.0293	0.9973	-1.9745	3.0041	-6.0700	3.3209
Estimated	18.0253	1.0166	-2.0165	2.9931	-5.9780	3.3387
S.E.	0.1845	0.1340	0.1453	0.1536	0.1419	0.1077

The Simulation Results

Means

Original	18.0293	0.9973	-1.9745	3.0041	-6.0700	3.3209
Estimated	18.0253	1.0166	-2.0165	2.9931	-5.9780	3.3387
S.E.	0.1845	0.1340	0.1453	0.1536	0.1419	0.1077

Original covariance matrix

6.0332	0.4005	-0.6253	1.0875	-1.9673	1.0251
0.4005	1.0741	-0.0554	-0.0257	0.0529	0.0359
-0.6253	-0.0554	0.8846	-0.0332	0.0724	0.0294
1.0875	-0.0257	-0.0332	0.9997	-0.0110	0.0234
-1.9673	0.0529	0.0724	-0.0110	2.2645	0.0299
1.0251	0.0359	0.0294	0.0234	0.0299	0.4159

The Simulation Results

Means

Original	18.0293	0.9973	-1.9745	3.0041	-6.0700	3.3209
Estimated	18.0253	1.0166	-2.0165	2.9931	-5.9780	3.3387
S.E.	0.1845	0.1340	0.1453	0.1536	0.1419	0.1077

Original covariance matrix

6.0332	0.4005	-0.6253	1.0875	-1.9673	1.0251
0.4005	1.0741	-0.0554	-0.0257	0.0529	0.0359
-0.6253	-0.0554	0.8846	-0.0332	0.0724	0.0294
1.0875	-0.0257	-0.0332	0.9997	-0.0110	0.0234
-1.9673	0.0529	0.0724	-0.0110	2.2645	0.0299
1.0251	0.0359	0.0294	0.0234	0.0299	0.4159

Estimated covariance matrix

4.0793	0.2168	-0.3825	0.6135	-1.2302	0.6816
0.2168	0.6719	-0.0042	0.0088	-0.0132	0.0119
-0.3825	-0.0042	0.6703	-0.0101	0.0233	-0.0072
0.6135	0.0088	-0.0101	0.6832	-0.0281	0.0238
-1.2302	-0.0132	0.0233	-0.0281	1.4797	-0.0266
0.6816	0.0119	-0.0072	0.0238	-0.0266	0.3330

The Simulation Results

Means

Original	18.0293	0.9973	-1.9745	3.0041	-6.0700	3.3209
Estimated	18.0253	1.0166	-2.0165	2.9931	-5.9780	3.3387
S.E.	0.1845	0.1340	0.1453	0.1536	0.1419	0.1077

Original covariance matrix

6.0332	0.4005	-0.6253	1.0875	-1.9673	1.0251
0.4005	1.0741	-0.0554	-0.0257	0.0529	0.0359
-0.6253	-0.0554	0.8846	-0.0332	0.0724	0.0294
1.0875	-0.0257	-0.0332	0.9997	-0.0110	0.0234
-1.9673	0.0529	0.0724	-0.0110	2.2645	0.0299
1.0251	0.0359	0.0294	0.0234	0.0299	0.4159

Estimated covariance matrix

4.0793	0.2168	-0.3825	0.6135	-1.2302	0.6816
0.2168	0.6719	-0.0042	0.0088	-0.0132	0.0119
-0.3825	-0.0042	0.6703	-0.0101	0.0233	-0.0072
0.6135	0.0088	-0.0101	0.6832	-0.0281	0.0238
-1.2302	-0.0132	0.0233	-0.0281	1.4797	-0.0266
0.6816	0.0119	-0.0072	0.0238	-0.0266	0.3330

This bad result shows that the gradient calculated above is seriously biased.

References

HUBER, P. J. (1964). Robust estimation of a location parameter. *Ann. Math. Statist.*, 35:pp. 73 – 101.

NEYKOV, N. and NEYTCHEV, P. (1990). A robust alternative of the maximum likelihood estimators. In: *Short communications of COMPSTAT'90*, pp. 99 – 100. Dubrovnik, Yugoslavia.

ROBINS, H. and S.MONRO (1951). A stochastic approximation method. *Annals of Math. Stat*, 22:pp. 400–407.

ROUSSEEUW, P. J. and LEROY, A. (1987). *Robust Regression and Outlier Detection*. John Wiley and Sons, New York.

ROUSSEEUW, P. J. (1984). Least median of squares regression. *J. Amer. Statist. Assoc.*, 79(388):pp.

References

HUBER, P. J. (1964). Robust estimation of a location parameter. *Ann. Math. Statist.*, 35:pp. 73 – 101.

NEYKOV, N. and NEYTCHEV, P. (1990). A robust alternative of the maximum likelihood estimators. In: *Short communications of COMPSTAT'90*, pp. 99 – 100. Dubrovnik, Yugoslavia.

ROBINS, H. and S.MONRO (1951). A stochastic approximation method. *Annals of Math. Stat*, 22:pp. 400–407.

ROUSSEEUW, P. J. and LEROY, A. (1987). *Robust Regression and Outlier Detection*. John Wiley and Sons, New York.

ROUSSEEUW, P. J. (1984). Least median of squares regression. *J. Amer. Statist. Assoc.*, 79(388):pp.

871–880. ISSN 0162-1459.

TUKEY, J. W. (1960). A survey of sampling from contaminated distributions. In: OLKIN, I. (ed.), *Contributions to Probability and Statistics*, pp. 448 – 485. Stanford Univ. Press, Stanford.

VANDEV, D. (1992). Stochastic optimisation algorithm applied to least median type estimators. In: *8th Seminar on Stat.Data Analysis*, pp. 114–120. Varna.

871–880. ISSN 0162-1459.

Tukey, J. W. (1960). A survey of sampling from contaminated distributions. In: Olkin, I. (ed.), *Contributions to Probability and Statistics*, pp. 448 – 485. Stanford Univ. Press, Stanford.

Vandev, D. (1992). Stochastic optimisation algorithm applied to least median type estimators. In: *8th Seminar on Stat.Data Analysis*, pp. 114–120. Varna.