Section 2:
Optimisation for Robust Parameter Estimation

by Tat-Jun Chin

Outline

M-estimators

Least absolute deviation

Least maximum deviation

Least median and least k-th order

Robust norms

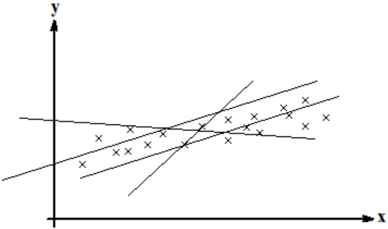
Recall the usage of robust norms:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} r_i(\boldsymbol{\theta})^2 \quad \Longrightarrow \quad \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \rho(r_i(\boldsymbol{\theta}))$$

To simplify the talk, lets focus on linear models:

$$r_i(\boldsymbol{\theta}) = \mathbf{x}_i^T\boldsymbol{\theta} - y_i$$

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \rho(\mathbf{x}_i^T\boldsymbol{\theta} - y_i)$$



M-estimators

"M" for "maximum likelihood-type" (Huber, 1981) — ML estimation is a special case of M-estimation.

Differentiating the objective function against $\boldsymbol{\theta}$ and setting to $\mathbf{0}$ yields a system of simultaneous equations:

$$\sum_{i=1}^{N} \psi(\mathbf{x}_i^T \boldsymbol{\theta} - y_i)\mathbf{x}_i = \mathbf{0},$$

where

$$\psi(t) = \rho'(t)$$

is called the **influence function**. The M-estimate is the solution of this system.

It is customary to use norms of the form

$$\psi(t) = t \cdot w(t).$$

where $w(t)$ is the **weight function**.

A short list of common ρ functions

type	$\rho(x)$	$\psi(x)$	$w(x)$
L_2	$x^2/2$	x	1
L_1	$\lvert x \rvert$	$\operatorname{sgn}(x)$	$\frac{1}{\lvert x \rvert}$
Huber $\begin{cases} \text{if } \lvert x \rvert \le k \\ \text{if } \lvert x \rvert \ge k \end{cases}$	$\begin{cases} x^2/2 \\ k(\lvert x \rvert - k/2) \end{cases}$	$\begin{cases} x \\ k \operatorname{sgn}(x) \end{cases}$	$\begin{cases} 1 \\ k/\lvert x \rvert \end{cases}$
Tukey $\begin{cases} \text{if } \lvert x \rvert \le c \\ \text{if } \lvert x \rvert > c \end{cases}$	$\begin{cases} \frac{c^2}{6}\left(1 - [1-(x/c)^2]^3\right) \\ (c^2/6) \end{cases}$	$\begin{cases} x[1-(x/c)^2]^2 \\ 0 \end{cases}$	$\begin{cases} [1-(x/c)^2]^2 \\ 0 \end{cases}$

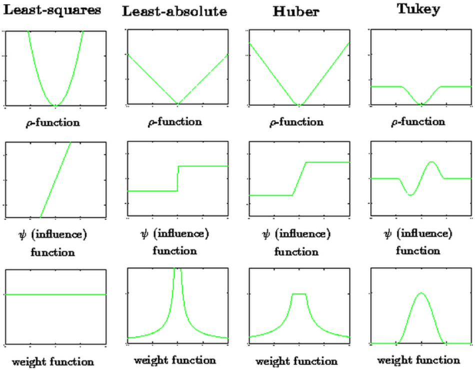


Figure adapted from [Z. Zhang, IVC 1997].

M-estimators

Define $w_i = w(\mathbf{x}_i^T \boldsymbol{\theta} - y_i)$ and rewrite

$$\sum_{i=1}^{N} (\mathbf{x}_i^T \boldsymbol{\theta} - y_i) w_i \mathbf{x}_i = \mathbf{0}$$

Rearranging we get

$$\sum_{i=1}^{N} \mathbf{x}_i w_i \mathbf{x}_i^T \boldsymbol{\theta} = \sum_{i=1}^{N} \mathbf{x}_i w_i y_i$$

Define $\mathbf{W} = \mathrm{diag}([w_1 \;\; w_2 \;\; \ldots \;\; w_N])$ and rewrite in matrix form

$$\mathbf{X}^T \underset{\uparrow}{\mathbf{W}} \mathbf{X} \underset{\uparrow}{\boldsymbol{\theta}} = \mathbf{X}^T \underset{\uparrow}{\mathbf{W}} \mathbf{y}$$

It is vital to see that $\mathbf{W}$ depends on θ.

IRLS

Use **iteratively reweighted least squares**:

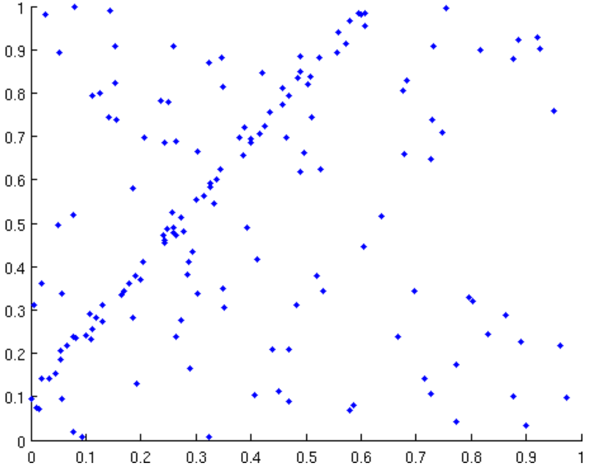
1. Initialise $\boldsymbol{\theta}^{(0)}$ and compute $\mathbf{W}^{(0)}$.
2. Revise $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta}^{(t+1)} = (\mathbf{X}^T\mathbf{W}^{(t)}\mathbf{X})^{-1}\mathbf{X}\mathbf{W}^{(t)}\mathbf{y}$$

3. Recompute $\mathbf{W}^{(t+1)}$ using $\boldsymbol{\theta}^{(t+1)}$.
4. Repeat from Step 2 until convergence.

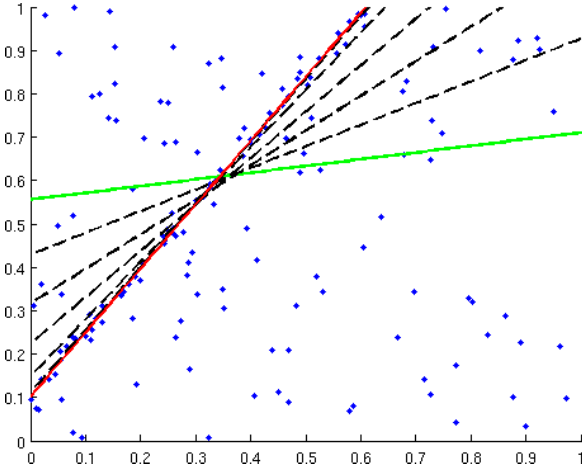
IRLS (cont.)

Data.



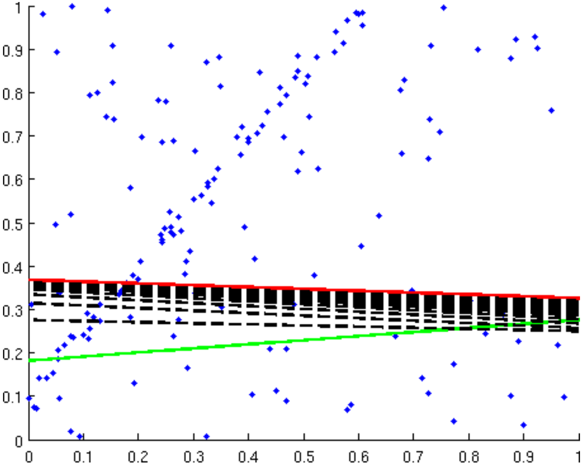
IRLS (cont.)

M-estimate with biweight function as ρ.



IRLS (cont.)

Non-convex ρ's do not guarantee unicity. Good initialition is crucial.



Outline

M-estimators

Least absolute deviation

Least maximum deviation

Least median and least k-th order

Least absolute deviation

Focussing again on linear models, the LAD estimate is

$$\boldsymbol{\theta}_{LAD} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} |y_i - \mathbf{x}_i^T \boldsymbol{\theta}|$$

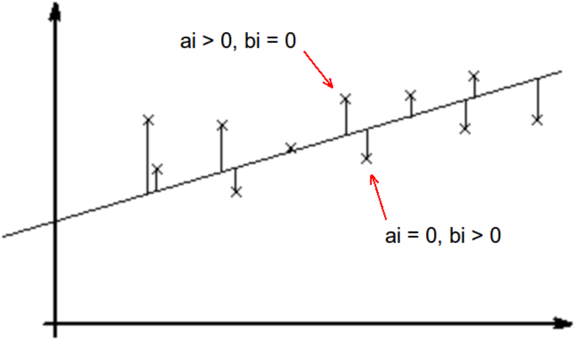
It is well known that this has an equivalent Linear Program

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{N} a_i + b_i \\ \text{s.t.} \quad & a_i \geq 0 \\ & b_i \geq 0 \\ & y_i - \mathbf{x}_i^T \boldsymbol{\theta} = a_i - b_i \end{aligned}$$

a_i and b_i are resp. the vertical deviations above and below the line.

Least absolute deviation (cont.)

By design, a_i and b_i cannot both be strictly positive.



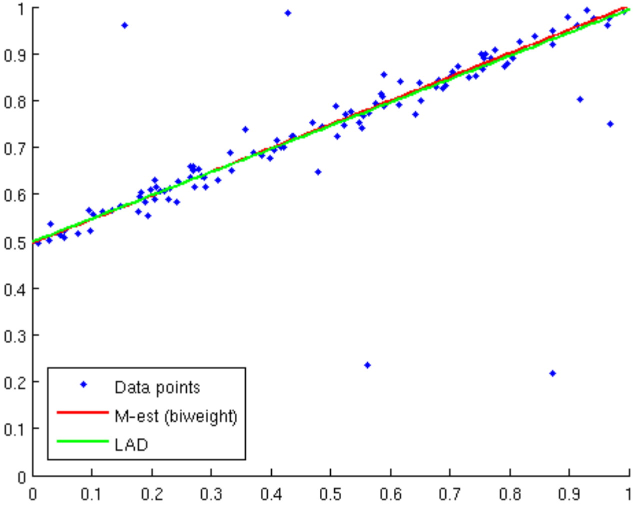
Least absolute deviation (cont.)

Can be converted into a simpler form.

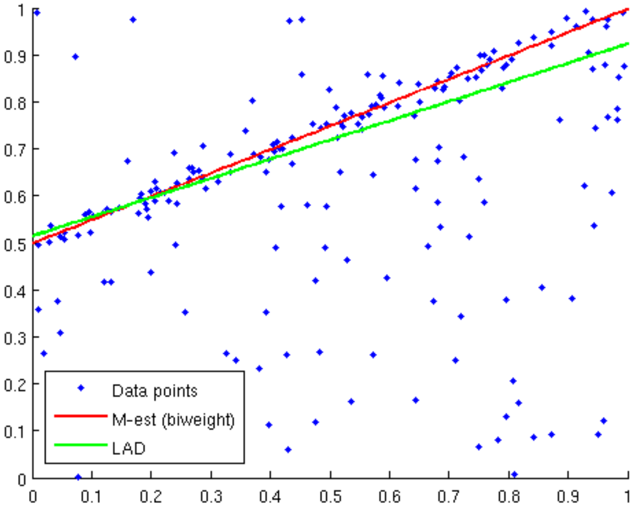
$$\min_{\boldsymbol{\theta}} \quad \sum_{i=1}^{N} s_i$$
$$\text{s.t.} \quad |y_i - \mathbf{x}_i^T \boldsymbol{\theta}| \leq s_i$$

```
function [ theta ] = leastabsdev(x,y)
   N = length(x);

   f = [ zeros(2,1) ; ones(N,1) ];
   A = [ -x -ones(N,1) -1*eye(N) ; x ones(N,1) -1*eye(N) ];
   b = [ -y ; y ];
   sol = linprog(f,A,b);

   theta = sol(1:2);
end
```

Results



Results (cont.)



Outline

M-estimators

Least absolute deviation

Least maximum deviation

Least median and least k-th order

Least maximum deviation

Minimise the maximum deviation:

$$\boldsymbol{\theta}_{LMD} = \arg\min_{\boldsymbol{\theta}} \left[\max_i |\mathbf{x}_i^T \boldsymbol{\theta} - y_i| \right]$$

$\boldsymbol{\theta}_{LMD}$ is also called the **minimax** or **Chebyshev** estimate.

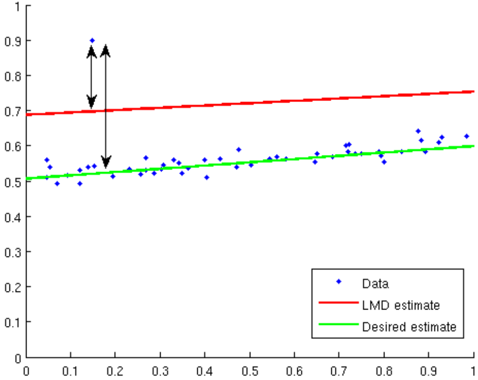
Stacking the residuals into a vector, we can rewrite

$$\boldsymbol{\theta}_{LMD} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{r}(\boldsymbol{\theta})\|_\infty, \qquad \mathbf{r}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{x}_1^T \boldsymbol{\theta} - y_1 \\ \vdots \\ \mathbf{x}_N^T \boldsymbol{\theta} - y_N \end{bmatrix}$$

i.e., minimise the **L-infinity norm** of the residuals.

Least maximum deviation (cont.)

The LMD estimator is **inherently non-robust** — the maximum deviation is primarily due to the outlier(s).



The relevance of LMD will be clear later.

Least maximum deviation (cont.)

The equivalent Linear Program is

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & s \\ \text{s.t.} \quad & |\mathbf{x}_i^T \boldsymbol{\theta} - y_i| \leq s \\ & s \geq 0 \end{aligned}$$

```
function [ theta ] = leastmaxdev(x,y)
   N = length(x);

   f = [ zeros(2,1) ; 1 ];
   A = [ x -ones(N,1) ; -x -ones(N,1) ];
   b = [ y ; -y ];
   sol = linprog(f,A,b);

   theta = sol(1:2);
end
```

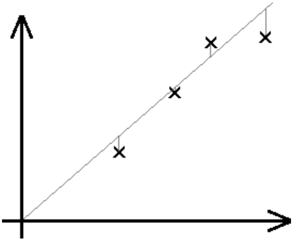
Characterising the LMD solution

We motivate with a simple example. Given points

x	2	4	5	6
y	1.2	2.1	2.6	3.1

we want to estimate the **non-affine line** $y = x\theta$ which solves

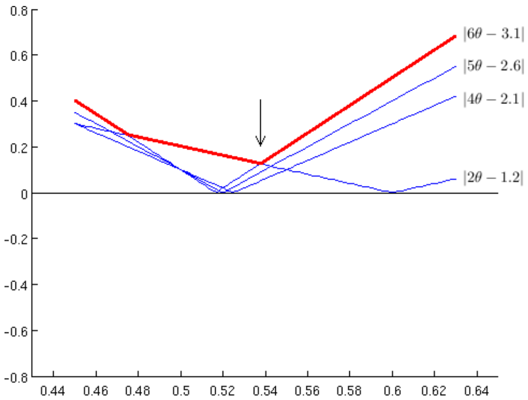
$$\min_{\theta} \max_{i} |x_i\theta - y_i|$$



Characterising the LMD solution (cont.)

We can graph the problem corresponding to the four points:

$$\min_{\theta} \max \{|2\theta - 1.2|, |4\theta - 2.1|, |5\theta - 2.6|, |6\theta - 3.1|\}$$

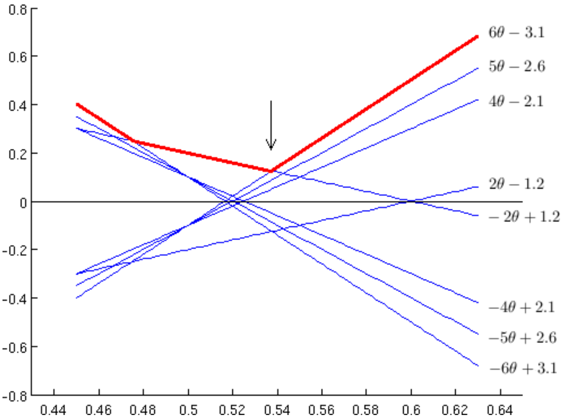


The objective function is **convex** but **non-differentiable**.

Characterising the LMD solution (cont.)

An equivalent problem is obtained by replacing each absolute residual by two residuals of differing signs.

$$\min_{\theta} \max\{2\theta - 1.2, 4\theta - 2.1, 5\theta - 2.6, 6\theta - 3.1,$$
$$-2\theta + 1.2, -4\theta + 2.1, -5\theta + 2.6, -6\theta + 3.1\}$$



Characterising the LMD solution (cont.)

Theorem: Every solution of the problem

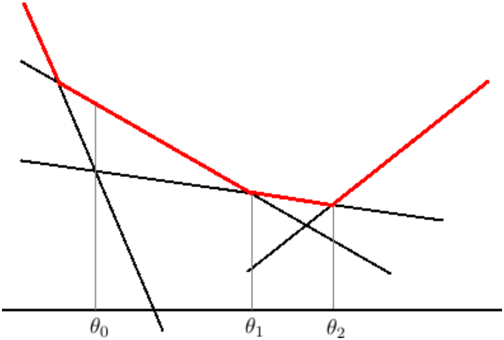
$$\min_{\boldsymbol{\theta}} \max_{i} |\mathbf{x}_i^T \boldsymbol{\theta} - y_i|$$

for N points $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$ with $\mathbf{x}_i \in \mathbb{R}^p$, is a solution of an appropriate subproblem of $p+1$ points

$$\min_{\boldsymbol{\theta}} \max_{i \in \mathcal{J}} |\mathbf{x}_i^T \boldsymbol{\theta} - y_i|,$$

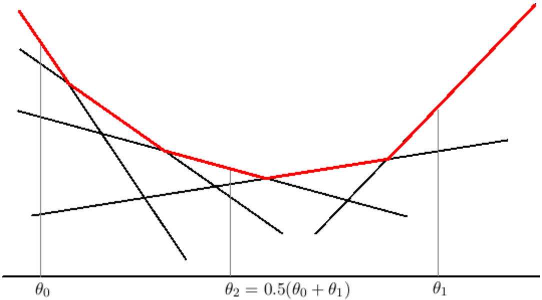
where $\mathcal{J} \subset \{1, 2, \dots, N\}$ and $|\mathcal{J}| = p + 1$.

LMD Algorithm 2 - vertex to vertex



Define $C(\theta) = \max_i x_i\theta - y_i$. Starting with any θ, define $M = \{j | x_j\theta - y_j = C(\theta)\}$. If M contains two elements k and l such that $x_k x_l < 0$, then θ is the solution. Else, select $m \in M$ for which $|x_m|$ is minimum. Then move θ along the right (resp. left) if $x_m < 0$ (resp. $x_m > 0$) until there is a n for which $x_m\theta - y_m = x_n\theta - y_n$. Stop if θ is optimal, else repeat.

LMD Algorithm 3 - bisection



Define $C(\theta) = \max_i x_i\theta - y_i$. Start with two estimates θ_0 and θ_1 on opposite sides of the solution. Let $x_i\theta_0 - y_i = C(\theta_0)$ and $x_j\theta_1 - y_j = C(\theta_j)$. If $x_i = 0$, then θ_0 is a solution. If $x_j = 0$, then θ_1 is a solution. Else, set $\theta_2 = 0.5(\theta_0 + \theta_1)$. Let $x_k\theta_2 = C(\theta_2)$. If $x_k < 0$, replace θ_0 by θ_2 and i by k. If $x_k > 0$, replace θ_1 by θ_2 and j by k. Repeat until desired level of accuracy.

Outline

M-estimators

Least absolute deviation

Least maximum deviation

Least median and least k-th order

Least median and least k-th order deviation

Least median **squares** (LMedS) solves for

$$\min_{\theta} \operatorname*{med}_{i} (x_i\theta - y_i)^2.$$

Robustness can also be achieved by minimising the median **absolute deviation**

$$\min_{\theta} \operatorname*{med}_{i} |x_i\theta - y_i|.$$

We can generalise to least k-th order **absolute deviation**

$$\min_{\theta} \{|x_i\theta - y_i|\}_{k:N}.$$

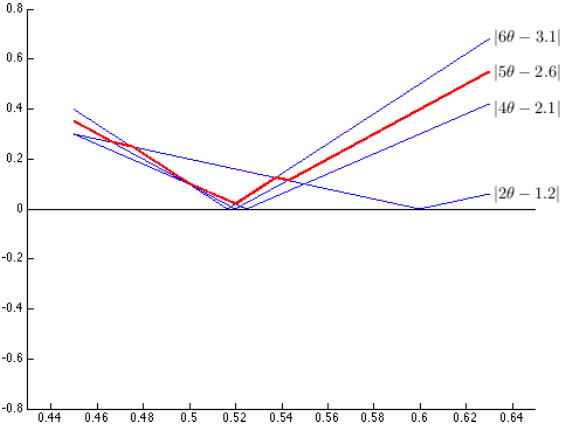
where $k : N$ indicates the k largest among N numbers.

Least median and least k-th order deviation (cont.)

For $k = N$, least k-th order becomes least maximum deviation.

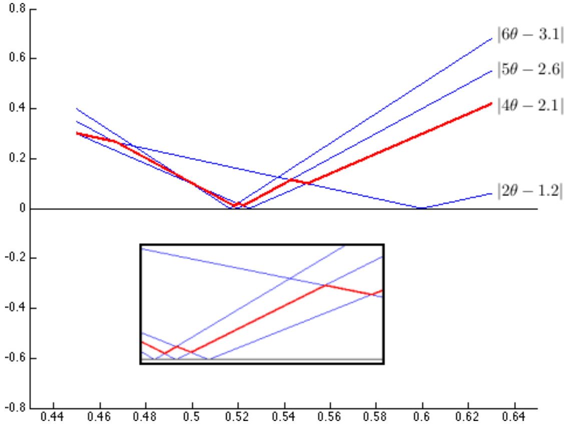
For $k < N$ the objective function is **non-convex**.

Example: $N = 4$ points, set $k = 3$.



Least median and least k-th order deviation (cont.)

Example: $N = 4$ points, set $k = 2$.



Characterising the LKO solution

Theorem: Every solution of the problem

$$\min_{\boldsymbol{\theta}} \{|\mathbf{x}_i^T \boldsymbol{\theta} - y_i|\}_{k:N}$$

for N points $\{\mathbf{x}_i, y_i\}_{i=1}^N$ with $\mathbf{x}_i \in \mathbb{R}^p$, is a solution of an appropriate subproblem of $p+1$ points

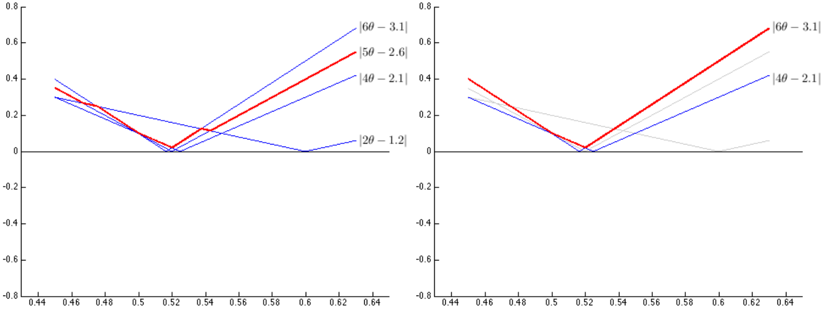
$$\min_{\boldsymbol{\theta}} \max_{i \in \mathcal{J}} |\mathbf{x}_i^T \boldsymbol{\theta} - y_i|,$$

where $\mathcal{J} \subset \{1, 2, \ldots, N\}$ and $|\mathcal{J}| = p + 1$.

Characterising the LKO solution (cont.)

ACCV 2012

Example: $N = 4$ points, $p = 1$, set $k = 3$. Use $\mathcal{J} = \{2, 4\}$ corresponding to residuals $|6\theta - 3.1|$ and $|4\theta - 2.1|$.



(a) LKO with $k = 3$.

(b) LMD with $\mathcal{J} = \{2, 4\}$.

Algorithm: Exhaustive sampling

Given $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^p$, and $k \le N$, do

1. Sample a $(p+1)$-tuple $\mathcal{J}$.
2. Solve $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \max_{i \in \mathcal{J}} |\mathbf{x}_i^T \boldsymbol{\theta} - y_i|$.
3. Compute residuals $r_j = |\mathbf{x}_j^T \boldsymbol{\theta}^* - y_j|$ for all $j = \{1, 2, \ldots, N\}$.
4. Seek k-th largest residual $r_{[k]}$.
5. If $r_{[k]}$ is smallest so far, record $\boldsymbol{\theta}^*$ and $r_{[k]}$.

until all $(p+1)$-tuples of $\{1, 2, \ldots, N\}$ have been sampled.

How many $(p+1)$-tuples need to be tested?

$$\binom{N}{p+1} = \frac{N!}{(p+1)!(N-p-1)!} = \frac{N(N-1)\ldots(N-p)}{(p+1)!}$$

which scales as $\mathcal{O}(N^{p+1})$ — doable, even for moderate N.

Solving LMD for $(p+1)$-tuples

Step 2 of the algorithm requires solving the least maximum deviation problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \max_{i \in \mathcal{J}} |\mathbf{x}_i^T \boldsymbol{\theta} - y_i|, \qquad |\mathcal{J}| = p + 1.$$

We can always solve this by linear programming (see MATLAB code for LMD), but this may be slow...

Solving LMD for $(p+1)$-tuples (cont.)

Fortunately analytic solutions also exist.

Without loss of generality, let $\mathcal{J} = \{1, 2, \ldots, p+1\}$. Let

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_{p+1}^T \end{bmatrix} \in \mathbb{R}^{(p+1)\times p}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_{p+1} \end{bmatrix} \in \mathbb{R}^{p+1}$$

First, we find the **least squares** fit for the $(p+1)$-tuple is

$$\boldsymbol{\theta}_{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

Let the LS residual of the i-th point in the $(p+1)$-tuple be

$$r_i = y_i - \mathbf{x}_i^T\boldsymbol{\theta}_{LS}.$$

Solving LMD for $(p+1)$-tuples (cont.)

The LMD criterion can be obtained from the LS residuals

$$\omega = \min_{\boldsymbol{\theta}} \max_{i \in \mathcal{J}} |\mathbf{x}_i^T \boldsymbol{\theta} - y_i|$$
$$= \begin{cases} 0 & \text{if } \sum_{i=1}^{p+1} r_i^2 = 0, \\ \frac{\sum_{i=1}^{p+1} r_i^2}{\sum_{i=1}^{p+1} |r_i|} & \text{otherwise.} \end{cases}$$

Defining the **sign** vector

$$\mathbf{s} = [\mathsf{sgn}(r_1)\ \mathsf{sgn}(r_2)\ \ \ldots\ \ \mathsf{sgn}(r_{p+1})]^T,$$

the LMD estimate is then

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}_{LS} - \omega(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{s}.$$