# Block-wise Minimization-Majorization algorithm for Huber's criterion: sparse learning and applications

**Esa Ollila** and Ammar Mian

Department of Signal Processing and Acoustics
Aalto University, Finland

**MLSP 2020**
Sept 21-24

Aalto University

# Menu

# Linear model

- *Outputs* (responses) $y_i \in \mathbb{R}$
- *Inputs* (predictors) $x_i^\top = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p$.
- Linear model of $N$ measurements:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} x_1^\top \\ \vdots \\ x_N^\top \end{pmatrix} \beta + \begin{pmatrix} e_1 \\ \vdots \\ e_N \end{pmatrix}$$
$$\mathbf{y} = \mathbf{X} \beta + \mathbf{e}$$

  where the error terms $e_i$ are i.i.d. with p.d.f. $f(e) = (1/\sigma) f_0(e/\sigma)$.

- *Goal*: to estimate robustly the unknown parameters
  - *regression coefficients* $\beta = (\beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^p$
  - *scale* parameter $\sigma > 0$

  given the data $(y_i, x_i^\top)$, $i = 1, \ldots, N$.

# Contributions

Huber's criterion [Hub81] for joint estimation of regression and scale:

$$L(\beta, \sigma) = N(\alpha\sigma) + \sum_{i=1}^{N} \rho_c\left(\frac{y_i - x_i^\top \beta}{\sigma}\right)\sigma,$$

where $\alpha > 0$ is a fixed scaling factor and $\rho_c$ is Huber's loss function.

1. Block-wise MM-algorithm for solving the optimum $(\hat{\beta}, \hat{\sigma})$ is derived rigorously.
2. Novel data-adaptive step sizes for regression and scale updates:

   $\Rightarrow$ improves convergence (observed empirically)
3. Applications of Huber's criterion are considered for:
   - Sparse signal recovery
   - Image denoising
   - Dictionary learning
4. Toolbox at: github.com/AmmarMian/huber_mm_framework

# Menu

# Robust ML approach

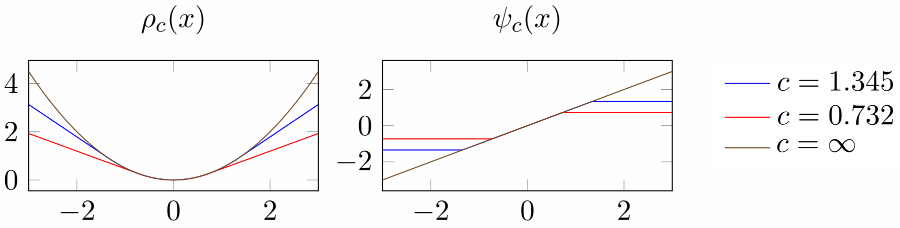
- Huber's unit scale ($\sigma = 1$) "least favorable distribution" (LFD) has p.d.f. $f_0(x) \propto \exp\{-\rho_c(x)\}$, where

$$\rho_c(x) = \frac{1}{2} \times \begin{cases} |x|^2, & \text{for } |x| \le c \\ 2c|x| - c^2, & \text{for } |x| > c, \end{cases}, \quad x \in \mathbb{R},$$

is called as Huber's loss function and $c$ is a user-defined *threshold*.

- The score function, $\psi_c = \rho_c'$ is a winsorizing function:

$$\psi_c(x) = \begin{cases} x, & \text{for } |x| \le c \\ c\,\text{sign}(x), & \text{for } |x| > c \end{cases},$$



# But direct ML approach fails

- The ML criterion function (assuming i.i.d. errors from LFD model)

$$L_{\mathsf{ML}}(\beta,\sigma) = -\sum_{i=1}^{N} \ln\left\{\frac{1}{\sigma} f_0\left(\frac{y_i - x_i^\top\beta}{\sigma}\right)\right\}$$
$$= N\ln\sigma + \sum_{i=1}^{N} \rho_c\left(\frac{y_i - x_i^\top\beta}{\sigma}\right)$$

  fails...
  - to be convex in $(\beta,\sigma)$
  - to provide robust estimates (bounded influence functions)
- Huber's modification

$$L(\beta,\sigma) = N(\alpha\sigma) + \sum_{i=1}^{N} \rho_c\left(\frac{y_i - x_i^\top\beta}{\sigma}\right)\sigma,$$

  is convex in $(\beta,\sigma)$ and provides robust estimates with bounded influence function.

# But direct ML approach fails

- The ML criterion function (assuming i.i.d. errors from LFD model)

$$L_{\mathsf{ML}}(\beta,\sigma) = -\sum_{i=1}^{N} \ln\left\{\frac{1}{\sigma} f_0\left(\frac{y_i - x_i^\top\beta}{\sigma}\right)\right\}$$
$$= N\ln\sigma + \sum_{i=1}^{N}\rho_c\left(\frac{y_i - x_i^\top\beta}{\sigma}\right)$$

  fails...
  - to be convex in $(\beta,\sigma)$
  - to provide robust estimates (bounded influence functions)
- Huber's modification

$$L(\beta,\sigma) = N(\alpha\sigma) + \sum_{i=1}^{N}\rho_c\left(\frac{y_i - x_i^\top\beta}{\sigma}\right)\sigma,$$

  is convex in $(\beta,\sigma)$ and provides robust estimates with bounded influence function.

# But direct ML approach fails

- The ML criterion function (assuming i.i.d. errors from LFD model)

$$\begin{aligned} L_{\mathsf{ML}}(\beta, \sigma) &= -\sum_{i=1}^{N} \ln \left\{ \frac{1}{\sigma} f_0\left( \frac{y_i - x_i^\top \beta}{\sigma} \right) \right\} \\ &= N \ln \sigma + \sum_{i=1}^{N} \rho_c\left( \frac{y_i - x_i^\top \beta}{\sigma} \right) \end{aligned}$$

  fails...
  - to be convex in $(\beta, \sigma)$
  - to provide robust estimates (bounded influence functions)

- Huber's modification

$$L(\beta, \sigma) = N(a\sigma) + \sum_{i=1}^{N} \rho_c\left( \frac{y_i - x_i^\top \beta}{\sigma} \right) \sigma,$$

  is convex in $(\beta, \sigma)$ and provides robust estimates with bounded influence function.

# But direct ML approach fails

- The ML criterion function (assuming i.i.d. errors from LFD model)

$$\begin{aligned} L_{\mathsf{ML}}(\beta, \sigma) &= -\sum_{i=1}^{N} \ln \left\{ \frac{1}{\sigma} f_0\left( \frac{y_i - x_i^\top \beta}{\sigma} \right) \right\} \\ &= N \ln \sigma + \sum_{i=1}^{N} \rho_c \left( \frac{y_i - x_i^\top \beta}{\sigma} \right) \end{aligned}$$

  fails...
  - to be convex in $(\beta, \sigma)$
  - to provide robust estimates (bounded influence functions)
- Huber's modification

$$L(\beta, \sigma) = N(\alpha\sigma) + \sum_{i=1}^{N} \rho_c \left( \frac{y_i - x_i^\top \beta}{\sigma} \right) \sigma,$$

  is convex in $(\beta, \sigma)$ and provides robust estimates with bounded influence function.

# Menu

# Blockwise Minimization-Majorization algorithm

$$\begin{aligned}\sigma^{(n+1)} &= \arg\min_{\sigma} g_2\big(\sigma|\beta^{(n)},\sigma^{(n)}\big)\\ \beta^{(n+1)} &= \arg\min_{\beta} g_1\big(\beta|\beta^{(n)},\sigma^{(n+1)}\big)\end{aligned}, \quad n = 0,1,\ldots$$

- $g_2$ is surrogate function for scale:

$$g_2(\sigma|\beta',\sigma') = a' + b'\frac{1}{\sigma} + N\alpha\sigma,$$

  s.t. $L(\beta',\sigma') = g_2(\sigma'|\beta',\sigma')$ and $\nabla_\sigma L(\beta',\sigma') = \nabla_\sigma g_2(\sigma'|\beta',\sigma')$.

- $g_1$ is a surrogate function for regression (and denote $r_i = y_i - x_i^\top\beta$):

$$g_1(\beta|\beta',\sigma') = N(\alpha\sigma') + \sum_{i=1}^{N}\left(a_i' + b_i'\frac{r_i}{\sigma'} + \frac{1}{2}\frac{r_i^2}{(\sigma')^2}\right)$$

  s.t. $L(\beta',\sigma') = g(\beta'|\beta',\sigma')$ and $\nabla_\beta L(\beta',\sigma') = \nabla_\beta g(\beta'|\beta',\sigma')$.

# Theorem 1

- $g_2(\sigma|\beta', \sigma') \geq L(\beta', \sigma)$ and the MM update of scale is

$$\sigma^{(n+1)} = \arg\min_{\sigma>0} g_2(\sigma|\beta^{(n)}, \sigma^{(n)}) = \sigma^{(n)}\tau \qquad ,$$

  where

$$\tau = \frac{1}{\sqrt{2N\alpha}} \left\| \psi_c\left( \frac{\mathbf{y} - \mathbf{X}\beta^{(n)}}{\sigma^{(n)}} \right) \right\|.$$

- $g_1(\beta|\beta', \sigma') \geq L(\beta, \sigma')$ and the MM update for regression is

$$\beta^{(n+1)} = \arg\min_{\beta\in\mathbb{R}^{p+1}} g_1(\beta|\beta^{(n)}, \sigma^{(n+1)}) = \beta^{(n)} + \delta \qquad ,$$

  where

$$\delta = \mathbf{X}^{+}\psi_c\left( \frac{\mathbf{y} - \mathbf{X}\beta^{(n)}}{\sigma^{(n+1)}} \right)\sigma^{(n+1)}.$$

- We introduce step-sizes $\lambda^{(n)}$ and $\mu^{(n)}$ to speed up the convergence.

# Theorem 1

- $g_2(\sigma|\beta', \sigma') \geq L(\beta', \sigma)$ and the MM update of scale is

$$\sigma^{(n+1)} = \arg\min_{\sigma > 0} g_2(\sigma|\beta^{(n)}, \sigma^{(n)}) = \sigma^{(n)} \tau^{\lambda^{(n+1)}},$$

where

$$\tau = \frac{1}{\sqrt{2N\alpha}} \left\| \psi_c \left( \frac{\mathbf{y} - \mathbf{X}\beta^{(n)}}{\sigma^{(n)}} \right) \right\|.$$

- $g_1(\beta|\beta', \sigma') \geq L(\beta, \sigma')$ and the MM update for regression is

$$\beta^{(n+1)} = \arg\min_{\beta \in \mathbb{R}^{p+1}} g_1(\beta|\beta^{(n)}, \sigma^{(n+1)}) = \beta^{(n)} + \delta \mu^{(n+1)},$$

where

$$\delta = \mathbf{X}^{+} \psi_c \left( \frac{\mathbf{y} - \mathbf{X}\beta^{(n)}}{\sigma^{(n+1)}} \right) \sigma^{(n+1)}.$$

- We introduce step-sizes $\lambda^{(n)}$ and $\mu^{(n)}$ to speed up the convergence.

# Step size computation

- To compute the step sizes we use line search.
- For regression, we minimize $L(\beta^{(n)} + \mu\delta, \sigma^{(n+1)})$ w.r.t. $\mu$:

$$\mu^{(n+1)} = \arg\min_{\mu} \sum_{i=1}^{N} \rho_c\left(\frac{r_i^{(n)} - \mu x_i^\top \delta}{\sigma^{(n+1)}}\right)$$

- For scale, we minimize $L(\beta^{(n)}, \sigma^{(n)}\tau^\lambda)$ w.r.t. $\lambda$:

$$\lambda^{(n+1)} = \arg\min_{\lambda} \; N\alpha\tau^\lambda + \sum_{i=1}^{N} \rho_c\left(\frac{y_i - x_i^\top \beta^{(n)}}{\sigma^{(n)}\tau^\lambda}\right)\tau^\lambda$$

- Instead of solving the optimization problems exactly, we use *closed-form approximations* of the solutions (*cf.* Algorithm 1).

# Menu

# Sparse learning

- Blockwise MM algorithm extends to *normalized iterative hard-thresholding (NIHT)* [BD10] algorithm used in sparse signal reconstruction. [DET06, DE11].
- $\beta$ is now assumed to be $K$-sparse:

$$\Gamma = \{i \in \{1, \dots, p\} \, : \, \beta_i \neq 0\} \qquad \text{with} \qquad \|\beta\|_0 = |\Gamma| \leq K.$$

- # predictors $\gg$ # of measurements ($p \gg N$).
- The main change in block MM algorithm is in the regression step:

$$\beta^{(n+1)} = H_K\left(\beta^{(n)} + \mu^{(n+1)} \mathbf{X}^\top \psi_c\left(\frac{\mathbf{y} - \mathbf{X}\beta^{(n)}}{\sigma^{(n+1)}}\right)\sigma^{(n+1)}\right),$$

  where $H_K$ denotes the hard-thresholding operator ($\psi_c = \rho_c'$).
- The algorithm is called **HUBNIHT** [OKK14] algorithm.

# Image denoising

- grayscale image is denoised in sliding windows (patches) of size $8 \times 8$.
- Each vectorized patch is modelled as

$$\mathbf{y} = \mathbf{u} + \mathbf{e},$$

  where $\mathbf{u}$ is the original noise-free image of size $N \times 1$
- $N = 64$ ( $\#$ of pixels in patches).
- $\mathbf{u}$ is assumed to have a sparse representation in an overcomplete dictionary $\mathbf{X}$, i.e., $\mathbf{u} = \mathbf{X}\beta$
- Reconstructed image patch $\hat{\mathbf{u}} = \mathbf{X}\hat{\beta}$ is solved using the HUBNIHT algorithm.
- We use threshold $c = 0.3529$ and $\mathbf{X}$ is the redundant 2D-DCT dictionary

# Denoised images

Noisy image

$K = 3, \mathrm{PSNR} = 24.50$

# Comparisons

Noisy image
PSNR $= 14.95$ dB

**HUBNIHT**, $K = 11$
PSNR $= 28.93$ dB

**Median filter**, $3 \times 3$
PSNR $= 26.27$ dB

**OMP**, $c = \frac{3}{2}$, $\lambda = \frac{1}{2}$
PSNR $= 22.22$ dB

**K-SVD**, $c = \frac{3}{2}$, $\lambda = \frac{1}{2}$
PSNR $= 21.58$ dB

**BM3D**
PSNR $= 24.17$ dB

# What’s cooking

A journal extension is currently being prepared . . . It includes

- More examples and applications
- Extended simulation studies and image denoising examples.
- Tuning of parameters (threshold $c$ and sparsity $K$) are discussed.
- Large extended discussion of dictionary learning applications for medical imaging.
- Journal extension will be sent to ArXiv
- Matlab and python functions are made publicly available with example scripts.

# References


Thomas Blumensath and Mike E Davies, *Normalized iterative hard thresholding: Guaranteed stability and performance*, IEEE Journal of selected topics in signal processing **4** (2010), no. 2, 298–309.

M. F. Duarte and Y. C. Eldar, *Structured compressed sensing: From theory to applications*, IEEE Trans. Signal Process. **59** (2011), no. 9, 4053–4085.

David L Donoho, Michael Elad, and Vladimir N Temlyakov, *Stable recovery of sparse overcomplete representations in the presence of noise*, IEEE Transactions on information theory **52** (2006), no. 1, 6–18.

P. J. Huber, *Robust statistics*, Wiley, New York, 1981.

Esa Ollila, Hyon-Jung Kim, and Visa Koivunen, *Robust iterative hard thresholding for compressed sensing*, 2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP) (Athens, Greece), May 21 – 23, 2014, pp. 226–229.