Huber Lasso: Robust Sparse Regression
IE505 Project II

Instructor: M.Ç. Pınar

Due Date: December 30, 2024 by 23:59, in groups of two if desired.

Introduction

The Huber Lasso problem is a modification of the standard Lasso regression that replaces the squared error loss with the robust *Huber loss*. This formulation is expected to improve performance in the presence of outliers. The objective function is given by:

$$\min_x \; F(x) = \sum_{i=1}^{m} h((Ax-b)_i) + \lambda \|x\|_1,$$

where:

- $A \in \mathbb{R}^{m \times n}$ is the design matrix,
- $b \in \mathbb{R}^m$ is the response vector,
- $h(z)$ is the *Huber loss function*, defined as:

$$h(z) = \begin{cases} \frac{z^2}{2\gamma}, & \text{if } |z| \le \gamma, \\ |z| - \frac{\gamma}{2}, & \text{if } |z| > \gamma, \end{cases}$$

- $\lambda > 0$ is the regularization parameter.

Gradient and Proximal Operators

The Huber Lasso objective consists of:

- A smooth part: $f(x) = \sum_{i=1}^{m} h((Ax-b)_i)$,
- A non-smooth part: $g(x) = \lambda \|x\|_1$.

Gradient of the Smooth Part

The gradient of $f(x)$ is:

$$\nabla f(x) = A^\top \phi(Ax - b),$$

where $\phi(z)$ is the derivative of the Huber loss:

$$\phi(z) = \begin{cases} \frac{z}{\gamma}, & \text{if } |z| \le \gamma, \\ \operatorname{sgn}(z), & \text{if } |z| > \gamma. \end{cases}$$

Proximal Operator of the Non-Smooth Part

The proximal operator for $g(x) = \lambda\|x\|_1$ is the *soft-thresholding operator*:

$$\text{prox}_{\alpha g}(v) = \text{sgn}(v) \cdot \max(|v| - \alpha\lambda, 0).$$

Proximal Gradient Method: HISTA

The proximal gradient method alternates between a gradient descent step for $f(x)$ and a proximal step for $g(x)$. The update rule is:

$$x^{k+1} = \text{prox}_{\alpha g}(x^k - \alpha\nabla f(x^k)),$$

where $\alpha > 0$ is the step size.

Accelerated FISTA-like Method: Fast-HISTA

The Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) accelerates convergence by incorporating momentum.

FISTA Updates

1. Momentum update:

$$y^{k+1} = x^k + \frac{t_k - 1}{t_{k+1}}(x^k - x^{k-1}),$$

where $t_k = \frac{1+\sqrt{1+4t_{k-1}^2}}{2}$.

2. Gradient step:

$$z^{k+1} = y^{k+1} - \alpha\nabla f(y^{k+1}).$$

3. Proximal step:

$$x^{k+1} = \text{prox}_{\alpha g}(z^{k+1}).$$

4. Update momentum parameter:

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}.$$

Test Cases

Below are test cases for Huber Lasso HISTA and Fast-HISTA

1. Simple Regression with Outliers

- Generate $A \in \mathbb{R}^{100\times 10}$ with $\mathcal{N}(0, 1)$ entries.
- Let $x_{\text{true}} \in \mathbb{R}^{10}$ have 3 nonzero entries.
- Compute $b = Ax_{\text{true}} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
- Add outliers to b: set 5 random entries of b to large values ($U(50, 100)$).

2. Regression with Heteroscedastic Noise

- Generate $A \in \mathbb{R}^{200\times 20}$ and x_{true}.
- Compute $b = Ax_{\text{true}} + \epsilon$, with $\epsilon_i \sim \mathcal{N}(0, 0.1^2)$ for most i, but $\epsilon_i \sim \mathcal{N}(0, 5^2)$ for a subset.

3. Sparse Regression with Outlier Features

- Generate $A \in \mathbb{R}^{100\times 10}$ with sparse columns.
- Add outliers to some columns of A.
- Compute $b = Ax_{\text{true}} + \epsilon$, with small noise.

Assignment

1. (50 points) Implement both algorithms in Julia (a. use a constant step size: e.g., $t = 1/L$, what is L for Huber function? b. A backtracking line search) and carry out the tests summarized in points 1 to 3 above. Give your observations in a short report using plots.

2. (25 points) Although the Huber function does not have continuous second derivatives everywhere, a Newton type method is still possible (check the web for references.) Can you devise and implement a Proximal Newton method for Huber Lasso (you might have to implement a clever line search as well) and do the tests 1-3 above? Report your observations.

3.(25 points) Download the paper "A fast iterative shrinkage thresholding algorithm for linear inverse problems", SIAM J. Imaging Sci. Vol. 2, No. 1, pp 183-202 (accessible from Bilkent). From Section 5 of the paper retrieve (from the web) the image data and repeat the deblurring experiments of Examples 1 and 2 with FISTA, HISTA and Fast-HISTA above. Make sure you find a method to blur the images in such a way that HISTA and Fast-HISTA return a higher quality image than FISTA. Give the blurred and deblurred images.