Pattern Information Processing: Robust Methods

Masashi Sugiyama
(Department of Computer Science)

Contact: W8E-505
sugi@cs.titech.ac.jp
http://sugiyama-www.cs.titech.ac.jp/~sugi/

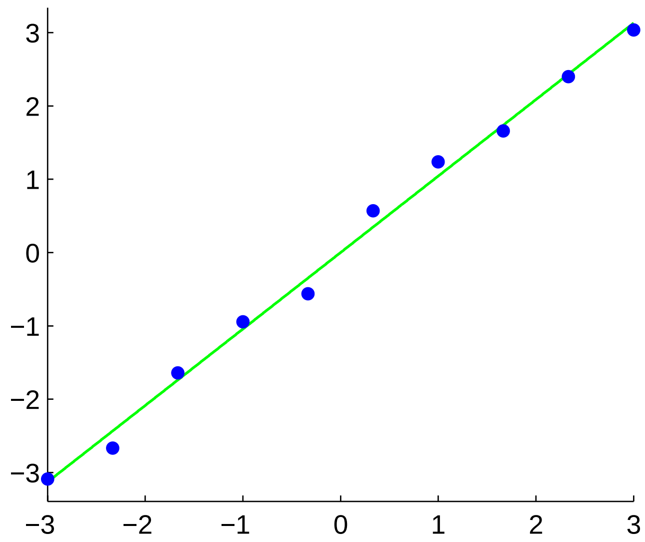
Outliers

- In practice, very large noise sometimes appears.
- Furthermore, irregular values can be observed by measurement trouble or by human error.
- Samples with such irregular values are called outliers.

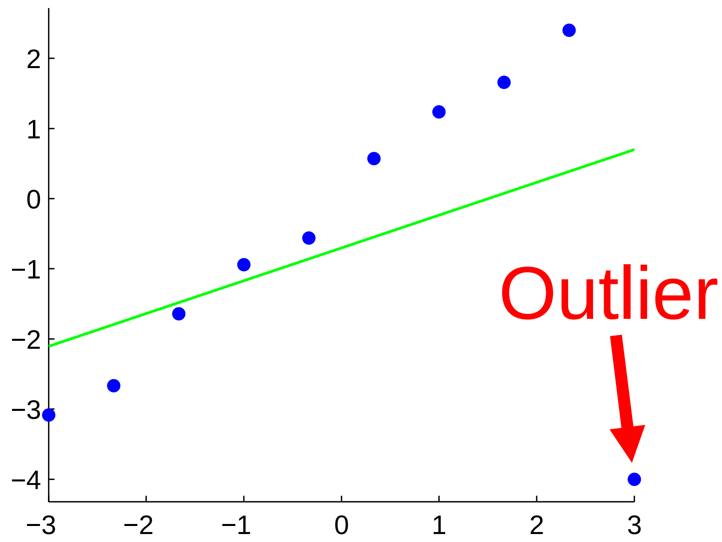
Outliers (cont.)

- LS criterion is sensitive to outliers.

$$\hat{f}(x) = \alpha_1 + \alpha_2 x$$



LS (without outlier)



LS (with outlier)

- Even a single outlier can corrupt the learning result badly!

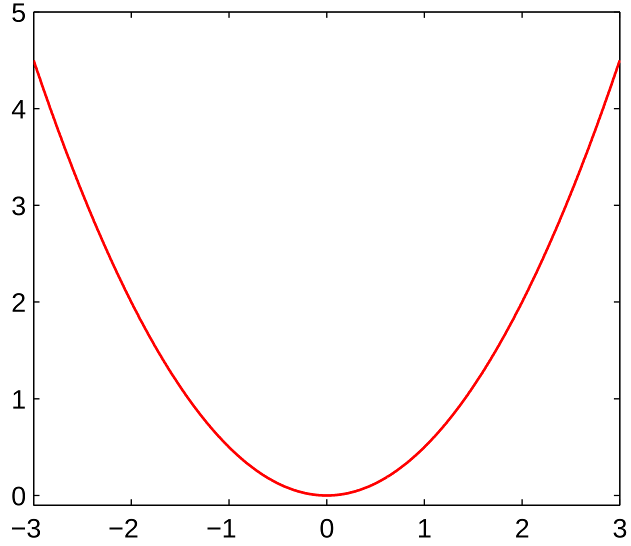
Today’s Plan

- Robust learning method
- How to obtain solusions
- Standard form of quadratic programs
- Robustness and sparseness

Quadratic Loss

$$J_{LS}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \left(\hat{f}(\boldsymbol{x}_i) - y_i \right)^2$$

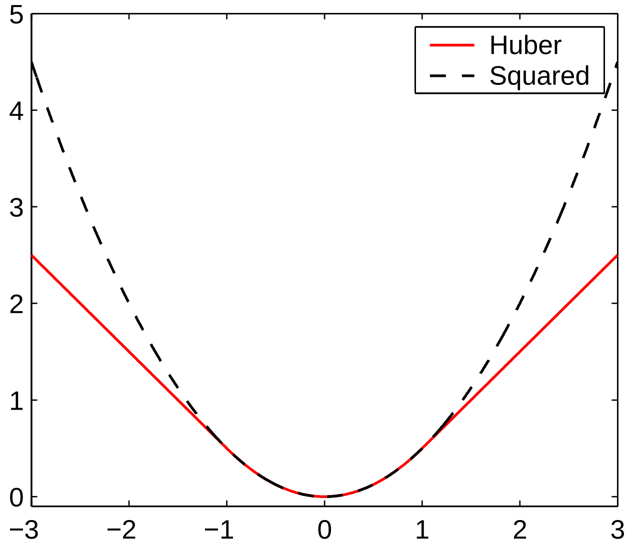
- In LS, goodness-of-fit is measured by the squared loss.
- Therefore, even a single outlier has quadratic power to “pull” the learned function
- The solution will be robust if the effect of outliers are deemphasized.



Huber's Robust Learning

$$\hat{\boldsymbol{\alpha}}_{Huber} = \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^b} \left[\sum_{i=1}^{n} \rho\left(\hat{f}(\boldsymbol{x}_i) - y_i \right) \right] \qquad t > 0$$

$$\rho(y) = \begin{cases} \frac{1}{2}y^2 & (|y| \leq t) \\ t|y| - \frac{1}{2}t^2 & (|y| > t) \end{cases}$$



- Squared-loss for non-outliers with small errors.
- Linear penalty for outliers with large errors.

P. J. Huber, Robust Statistics, Wiley, New York, 1981.

How to Obtain Solutions

- How to deal with Huber's loss?
- Use the following lemma:

Lemma

$$\rho(y) = \min_{v \in \mathbb{R}} g(v)$$

$$g(v) = \frac{1}{2}v^2 + t|y - v|$$

Proof of Lemma

- Here, we give a non-constructive proof.

See:

Mangasarian & Musicant, Robust linear and support vector regression, IEEE Trans. Pattern Analysis and Machine Intelligence, 22(9), 950-955,2000

$$g(v) = \begin{cases} \frac{1}{2}v^2 + ty - tv & (v \leq y) \\ \frac{1}{2}v^2 - ty + tv & (v > y) \end{cases}$$

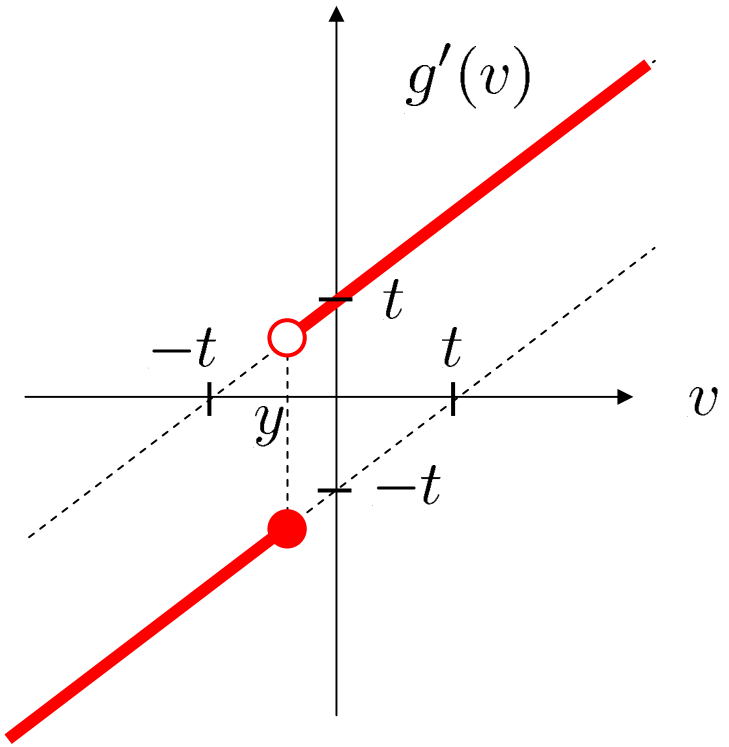
$$g'(v) = \begin{cases} v - t & (v \leq y) \\ v + t & (v > y) \end{cases}$$

Proof of Lemma (cont.)

- If $-t \le y \le t$, $g(v)$ is minimized at $v = y$.
  Then

$$\rho(y) = g(y) = \frac{1}{2}y^2$$

Note: $g(v)$ is continuous



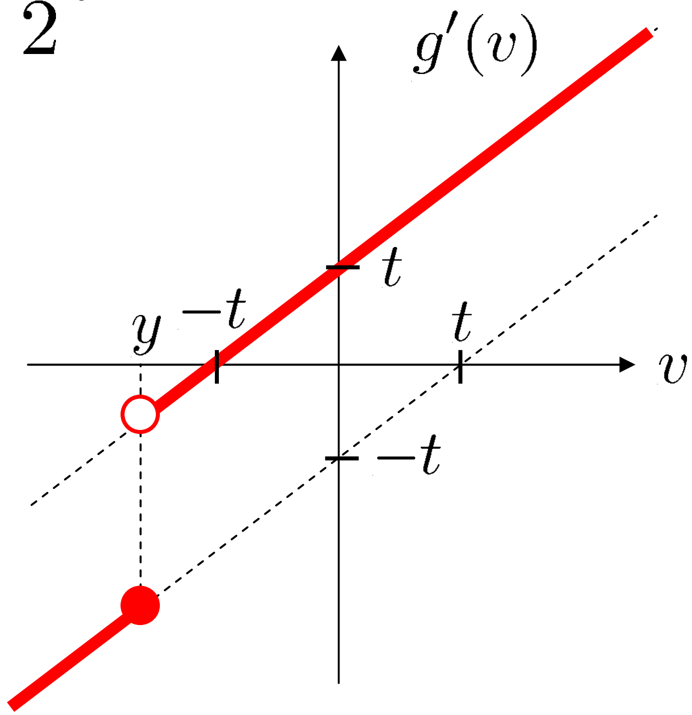
Proof of Lemma (cont.)

- If $y < -t$, $g(v)$ is minimized at $v = -t$. Then

$$\rho(y) = g(-t) = \frac{1}{2}t^2 + t|y+t|$$
$$= -ty - \frac{1}{2}t^2$$

Since $y < -t < 0$,

$$\rho(y) = t|y| - \frac{1}{2}t^2$$



Proof of Lemma (cont.)

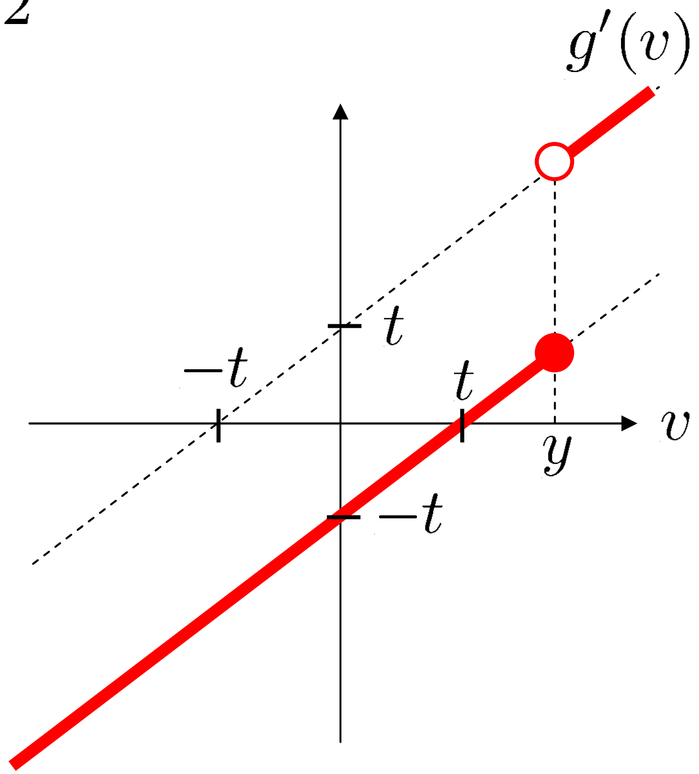
- If $y > t$, $g(v)$ is minimized at $v = t$. Then

$$\rho(y) = g(t) = \frac{1}{2}t^2 + t|y - t|$$

$$= ty - \frac{1}{2}t^2$$

Since $y > t > 0$,

$$\rho(y) = t|y| - \frac{1}{2}t^2$$



Q.E.D.

How to Obtain Solutions (cont.)

- Using

$$\rho(y) = \min_{v \in \mathbb{R}} \left[\frac{1}{2} v^2 + t|y - v| \right]$$

we have

$$\hat{\boldsymbol{\alpha}}_{Huber} = \mathop{\mathrm{argmin}}_{\boldsymbol{\alpha} \in \mathbb{R}^b, \boldsymbol{v} \in \mathbb{R}^n} \left[\frac{1}{2} \|\boldsymbol{v}\|^2 + t \|\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{y} - \boldsymbol{v}\|_1 \right]$$

$$\boldsymbol{X}_{i,j} = \varphi_j(\boldsymbol{x}_i)$$

$$\hat{\boldsymbol{\alpha}}_{Huber} \equiv \mathop{\mathrm{argmin}}_{\boldsymbol{\alpha} \in \mathbb{R}^b} \left[\sum_{i=1}^{n} \rho\left(\hat{f}(\boldsymbol{x}_i) - y_i \right) \right]$$

How to Obtain Solutions (cont.)

- Trick to avoid absolute value:

$$\|\boldsymbol{X}\boldsymbol{\alpha}-\boldsymbol{y}-\boldsymbol{v}\|_1=\min_{\boldsymbol{u}\in\mathbb{R}^n}\left[\sum_{i=1}^{n}u_i\right]$$

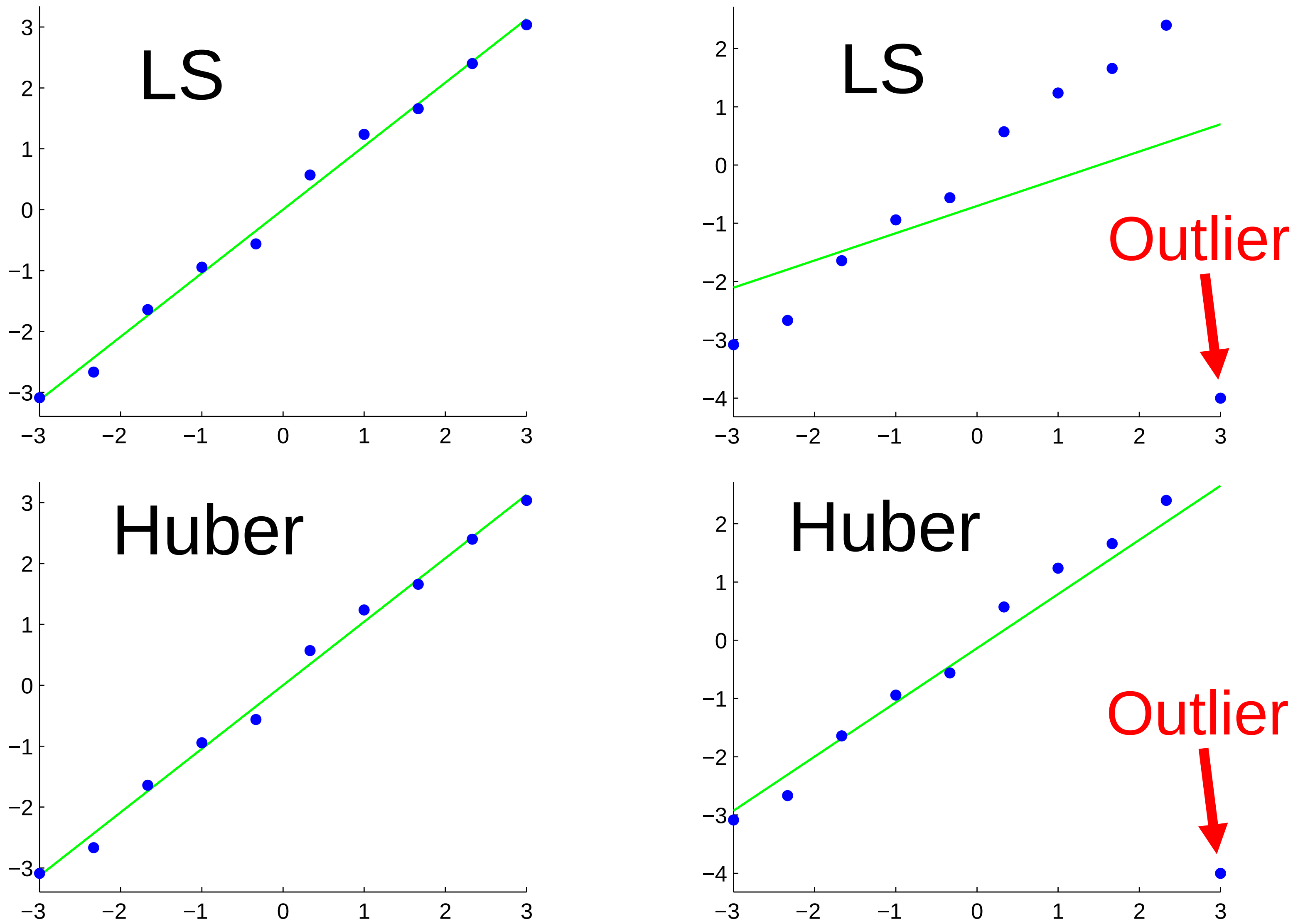
$$\text{subject to } -\boldsymbol{u}\leq\boldsymbol{X}\boldsymbol{\alpha}-\boldsymbol{y}-\boldsymbol{v}\leq\boldsymbol{u}$$

- $\hat{\boldsymbol{\alpha}}_{Huber}$ is given as the solution of

$$\mathop{\mathrm{argmin}}_{\boldsymbol{\alpha}\in\mathbb{R}^b,\boldsymbol{u},\boldsymbol{v}\in\mathbb{R}^n}\left[\frac{1}{2}\|\boldsymbol{v}\|^2+t\sum_{i=1}^{n}u_i\right]$$

$$\text{subject to } -\boldsymbol{u}\leq\boldsymbol{X}\boldsymbol{\alpha}-\boldsymbol{y}-\boldsymbol{v}\leq\boldsymbol{u}$$

Example of Huber's Method



Robust and Sparse

- Huber's method does not generally provide a sparse solution.
- Combining Huber's loss with ℓ_1 constraint.

$$\hat{\boldsymbol{\alpha}}_{SparseHuber} = \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^b} \left[\sum_{i=1}^{n} \rho\left(\hat{f}(\boldsymbol{x}_i) - y_i\right) \right]$$

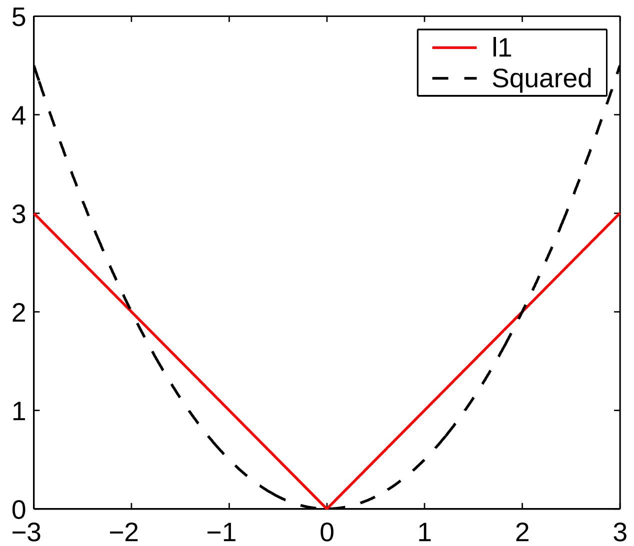
$$\text{subject to } \|\boldsymbol{\alpha}\|_1 \leq C$$

- Solving quadratic programming problem is computationally rather demanding.
- Is it possible to make it faster?

l1 Loss

- Quadratic term comes from Huber's loss.
- ℓ_1-loss is linear.

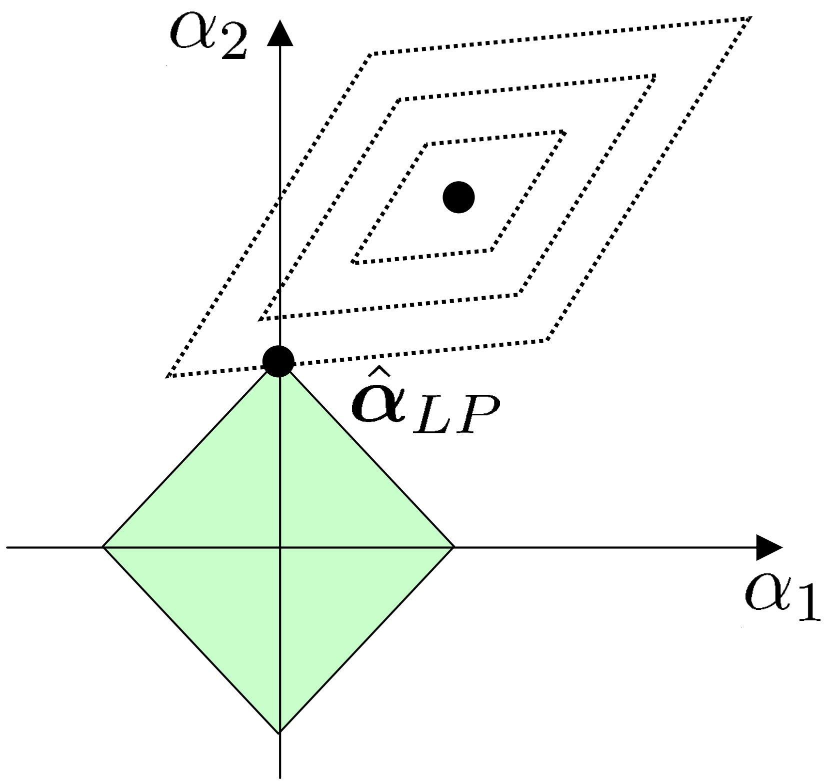
$$\sum_{i=1}^{n} \left| \hat{f}(\boldsymbol{x}_i) - y_i \right|$$



Linear Programming Learning

- Combine ℓ_1 loss with ℓ_1 regularizer:

$$\hat{\boldsymbol{\alpha}}_{LP} = \underset{\boldsymbol{\alpha} \in \mathbb{R}^b}{\mathrm{argmin}} \left[\sum_{i=1}^{n} \left| \hat{f}(\boldsymbol{x}_i) - y_i \right| + \lambda \sum_{i=1}^{b} |\alpha_i| \right]$$



How to Obtain Solutions

- Trick to avoid absolute value:

$$\|\boldsymbol{\alpha}\|_1 = \min_{\boldsymbol{u}\in\mathbb{R}^b} \left[\sum_{i=1}^{b} u_i\right]$$

$$\text{subject to } -\boldsymbol{u} \leq \boldsymbol{\alpha} \leq \boldsymbol{u},$$

- $\hat{\boldsymbol{\alpha}}_{LP}$ is given as the solution of

$$\mathop{\mathrm{argmin}}_{\boldsymbol{\alpha},\boldsymbol{u}\in\mathbb{R}^b,\boldsymbol{v}\in\mathbb{R}^n} \left[\sum_{i=1}^{n} v_i + \lambda \sum_{i=1}^{b} u_i\right]$$

$$\text{subject to } -\boldsymbol{v} \leq \boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{y} \leq \boldsymbol{v}$$

$$-\boldsymbol{u} \leq \boldsymbol{\alpha} \leq \boldsymbol{u}$$

Linearly Constrained Linear Programming Problem

- Standard optimization software can solve the following form of linearly constrained linear programming problems.

$$\min_{\boldsymbol{\beta}} \langle \boldsymbol{\beta}, \boldsymbol{q} \rangle \quad \text{subject to } \boldsymbol{V\beta} \leq \boldsymbol{v}$$
$$\boldsymbol{G\beta} = \boldsymbol{g}$$

Sparseness and Robustness

	Sparse-ness	Robust-ness	Optimi-zation
ℓ_1 constrained LS	Yes	No	Quadratic
Huber's method	No	Yes	Quadratic
ℓ_1 constrained Huber	Yes	Yes	Quadratic
Linear programming	Yes	Yes	Linear

Homework

1. Express the Huber learning problem in a standard form of quadratic programs.
2. Express the LP learning problem in a standard form of linear programs.

Homework (cont.)

2. For your own toy 1-dimensional data, perform simulations using
   - Linear/Gaussian kernel models
   - Huber/linear-programming learning

   and analyze the results, e.g., by changing
   - Target functions
   - Number of samples
   - Noise level
   - Width of Gaussian kernel
   - Robust/regularization parameter