#### Pattern Information Processing<sup>125</sup> Robust Method

Masashi Sugiyama (Department of Computer Science)

Contact: W8E-505 <u>sugi@cs.titech.ac.jp</u> 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.)

127

#### LS criterion is sensitive to outliers.



Even a single outlier can corrupt the learning result!

#### **Today's Plan**

128

# Robust learning method How to obtain solutions 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.



129

#### Huber's Robust Learning <sup>130</sup>

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

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



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

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

#### How to Obtain Solutions

131

How to deal with Huber's loss?

Use the following lemma:

Lemma  

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

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

See:

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

#### **Proof of Lemma**

We explicitly compute  $\min_{v \in \mathbb{R}} g(v)$  using g'(v).

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



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

132







## How to Obtain Solutions (cont.)<sup>36</sup>

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

#### we have

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

$$oldsymbol{X}_{i,j}=arphi_j(oldsymbol{x}_i)$$

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

#### How to Obtain Solutions (cont.)<sup>37</sup>

Trick to avoid absolute value:

$$egin{aligned} \|oldsymbol{X}oldsymbol{lpha}-oldsymbol{y}-oldsymbol{v}\|_1 &= \min_{oldsymbol{u}\in\mathbb{R}^n} \left[\sum_{i=1}^n u_i
ight] \end{aligned}$$

subject to  $-u \leq X\alpha - y - v \leq u$ 

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

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^{b}, \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{n}}{\operatorname{argmin}} \left[ \frac{1}{2} \|\boldsymbol{v}\|^{2} + t \sum_{i=1}^{n} u_{i} \right]$$
subject to  $-\boldsymbol{u} \leq \boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{y} - \boldsymbol{v} \leq \boldsymbol{u}$ 

#### **Standard Form (Huber)**

Let 
$$\beta = \begin{pmatrix} \alpha \\ u \\ v \end{pmatrix}$$
  
 $\Gamma_{\alpha} = (I_{b}, O_{b \times n}, O_{b \times n})$   
 $\Gamma_{u} = (O_{n \times b}, I_{n}, O_{n \times n})$   
 $\Gamma_{v} = (O_{n \times b}, O_{n \times n}, I_{n})$   
 $\alpha = \Gamma_{\alpha}\beta$   
 $u = \Gamma_{u}\beta$   
 $v = \Gamma_{v}\beta$   
 $\frac{1}{2} ||v||^{2} + t \sum_{i=1}^{n} u_{i} = \frac{1}{2} \langle \Gamma_{v}^{\top} \Gamma_{v} \beta, \beta \rangle + \langle \beta, t \Gamma_{u}^{\top} \mathbf{1}_{n} \rangle$   
 $-u \leq X\alpha - y - v \leq u$   
 $\left( \begin{array}{c} -u - X\alpha + v \\ X\alpha - v - u \end{array} \right) \leq \begin{pmatrix} -y \\ y \end{pmatrix}$   
 $\left( \begin{array}{c} -X\Gamma_{\alpha} - \Gamma_{u} + \Gamma_{v} \\ X\Gamma_{\alpha} - \Gamma_{u} - \Gamma_{v} \end{array} \right) \beta \leq \begin{pmatrix} -y \\ y \end{pmatrix}$ 

#### Example of Huber's Method<sup>139</sup>



#### **Robust and Sparse**

- Huber's method does not generally provide a sparse solution.
- **Combining Huber's loss with**  $\ell_1$  constraint.

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

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

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

#### **I1 Loss**

# Quadratic term comes from Huber's loss. *l*<sub>1</sub>-loss is linear.

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



Linear Programming Learning<sup>142</sup>

Combine  $\ell_1$  loss with  $\ell_1$  regularizer:

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



#### How to Obtain Solutions

#### Trick to avoid absolute value:

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

subject to  $-u < \alpha < u$ ,



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



143

Linearly Constrained Linear <sup>144</sup> Programming Problem

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

 $\min_{oldsymbol{eta}} \langle oldsymbol{eta}, oldsymbol{q} 
angle \ ext{ subject to } Voldsymbol{eta} \leq oldsymbol{v} \ Goldsymbol{eta} = oldsymbol{g}$ 

#### Standard Form (LP)

Let 
$$\beta = \begin{pmatrix} \alpha \\ u \\ v \end{pmatrix}$$
  
 $\Gamma_{\alpha} = (I_b, O_{b \times b}, O_{b \times n})$   
 $\Gamma_{u} = (O_{b \times b}, I_b, O_{b \times n})$   
 $\Gamma_{v} = (O_{n \times b}, O_{n \times b}, I_n)$   
 $\alpha = \Gamma_{\alpha}\beta$   
 $u = \Gamma_{u}\beta$   
 $v = \Gamma_{v}\beta$   
 $\sum_{i=1}^{n} v_i + \lambda \sum_{i=1}^{b} u_i = \langle \beta, \Gamma_v^{\top} \mathbf{1}_n + \lambda \Gamma_u^{\top} \mathbf{1}_b \rangle$   
 $-v \leq X\alpha - y \leq v$   
 $-u \leq \alpha \leq u$ 

# Sparseness and Robustness <sup>146</sup>

	Sparse- ness	Robust- ness	Optimi- zation
$\ell_1$ constrained LS	Yes	No	Quadratic
Huber's method	No	Yes	Quadratic
$\ell_1$ constrained Huber	Yes	Yes	Quadratic
Linear programming	Yes	Yes	Linear

#### Homework

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

Including outliers in the dataset would be essential for this homework.

#### Notification of Final Assignment

- 1. Apply supervised learning techniques to your data set and analyze it.
- 2. Write your opinion about this course

 Final report deadline: Aug 6<sup>th</sup> (Fri.)
 E-mail submission is also accepted! sugi@cs.titech.ac.jp

#### Mini-Workshop on Data Mining<sup>49</sup>

- On July 20<sup>th</sup> (final class), we have a mini-workshop on data mining, instead of regular lecture.
- Several students present their data mining results.
- Those who give a talk at the workshop will have very good grades!

### Mini-Workshop on Data Mining<sup>50</sup>

- Application (just to declare that you want to give a presentation) deadline: June 29<sup>th</sup>.
- Presentation: 10-15(?) minutes.
  - Specification of your dataset
  - Employed methods
  - Outcome
- OHP or projector may be used.
- Slides should be in English.
- Better to speak in English, but Japanese is also allowed.



- July 13<sup>th</sup>
- July 20<sup>th</sup>
  July 27<sup>th</sup>

#### Schedule

- : Neural Networks (regular lecture)
- : Preparation for workshop (no lecture)
- : Mini-workshop
- : Mini-workshop (if necessary)