电子科技大学
University of Electronic Science and Technology of China

A Brief View of Robust Semi-supervised Classification

Chen Huang

Data Mining Lab,
Big Data Research Center, UESTC
Email：huangchen.uestc@gmail.com

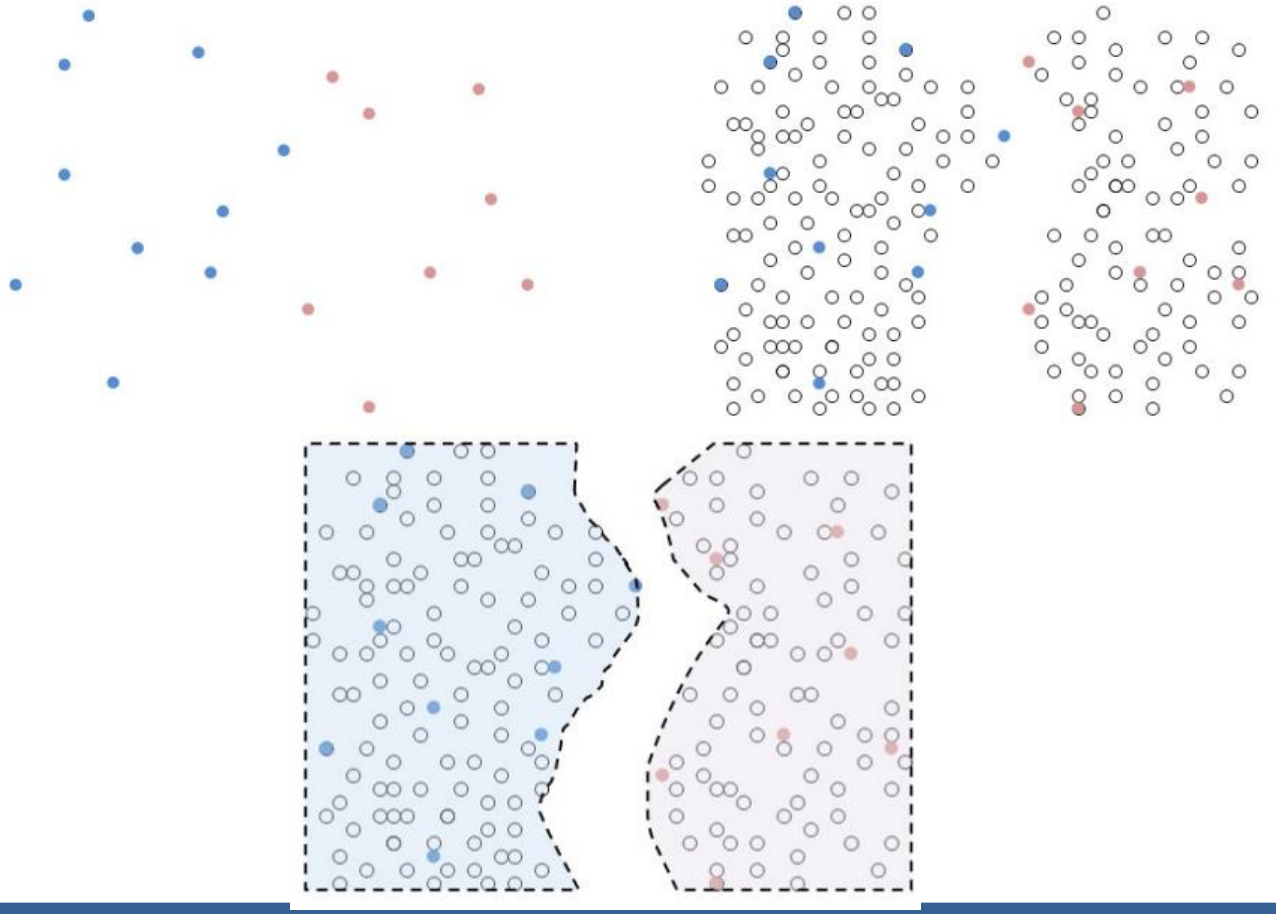
Table of Contents

Part One

Recap: Semi-supervised learning

What

- Learning from labeled data and unlabeled data to obtain a model with more generalization
- Transductive SSL / Inductive SSL



Assumption

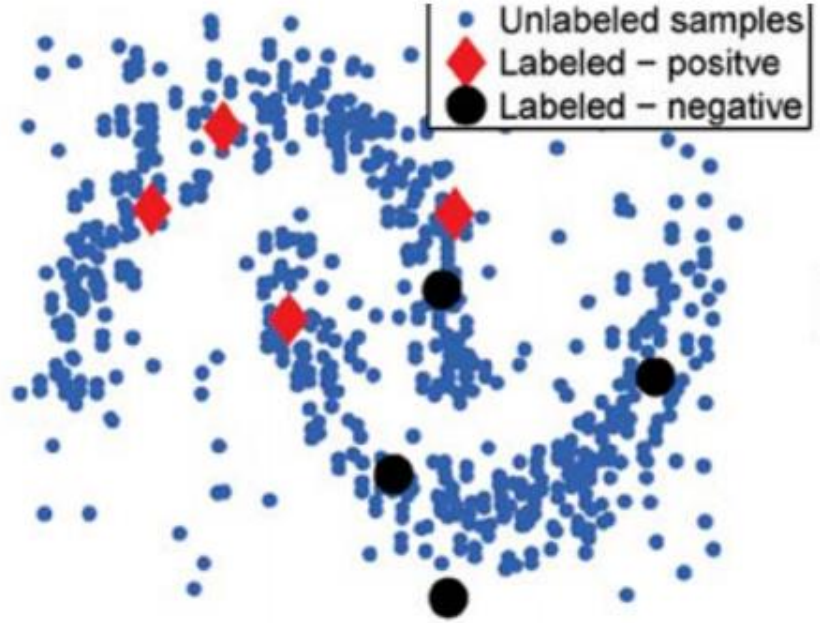
- Clustering Assumption
- Manifold Assumption

How

- Max margin model
- Co-training/Multi-view
- EM-Style
- Graph Based/ Label propagation
- Information Entropy
- Discriminative model (LDA Style)
- Linear neighbor propagation
- Negative matrix factorization
- ….

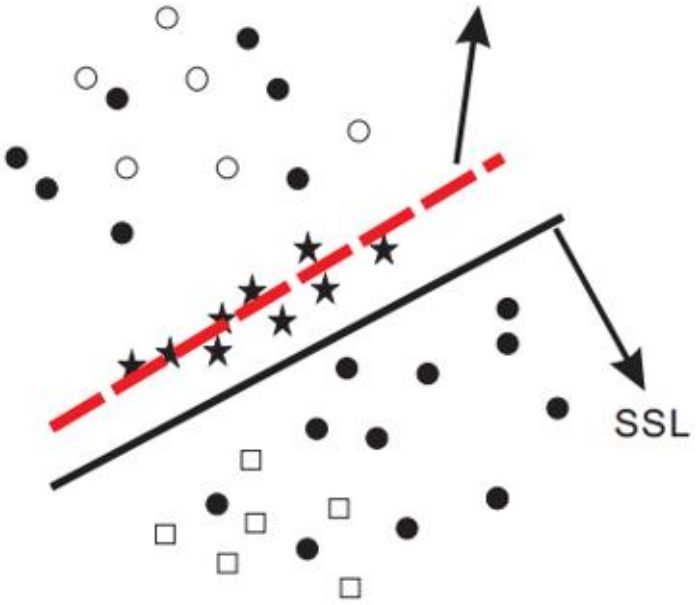
(X_L, Y_L) problems

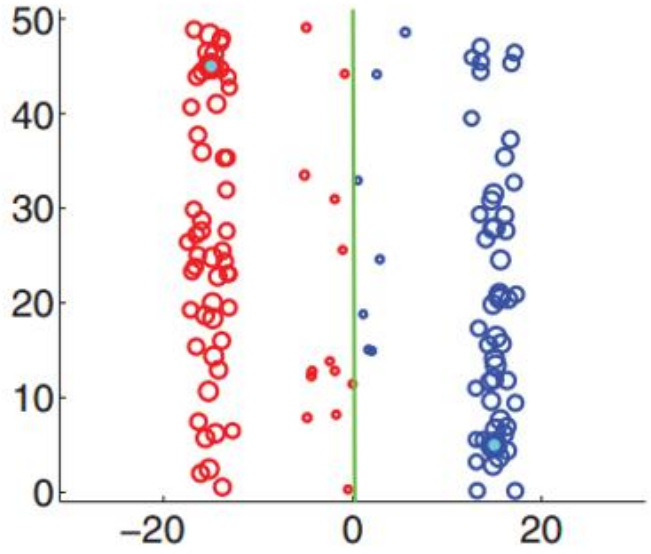
- X_U give no additional discriminative information (Y)
- What about label-noise?
- What about training data with incomplete information of Y



X_U problems

- No distribution information on X_L *or* X , how to identify that X_U are reliable ?
- Data structure V.S. class discrimination
- **How to integrate X_U with classifiers/target function**
  - Classification loss
  - Smooth penalty
  - *Regularization strength





Part Two

Unlabeled Data Help?

Categories

1. Multiple Classifiers Analysis
2. SSL and SL Trade-off
3. Clearing Max Margin
4. Adaptive Weighting

S3VM-us (Yu-Feng Li, AAAI_2011, arXiv_2010, PRICAI_2016)

- **Idea**: Select confident unlabeled data to use

Algorithm 3 S3VM-*us*

Input: y_{SVM}, y_{S3VM}, $\mathcal{D}$ and parameter ϵ

1: Let $\mathcal{S}$ be a set of the unlabeled data $\mathbf{x}$ such that $y_{SVM}(\mathbf{x}) \neq y_{S3VM}(\mathbf{x})$.
2: Perform hierarchical clustering, e.g., single linkage method (Jain and Dubes 1988).
3: For each unlabeled instance $\mathbf{x}_i \in \mathcal{S}$, calculate p_i and n_i, that is, the length of the paths from $\mathbf{x}_i$ to its nearest positive and negative labeled instances, respectively. Denote $t_i = (n_i - p_i)$.
4: Let $\mathcal{B}$ be the set of unlabeled instances $\mathbf{x}_i$ in $\mathcal{S}$ satisfying $|t_i| \geq \epsilon|l+u|$.
5: If $\sum_{\mathbf{x}_i \in \mathcal{B}} y_{S3VM}(\mathbf{x}_i)t_i \geq \sum_{\mathbf{x}_i \in \mathcal{B}} y_{SVM}(\mathbf{x}_i)t_i$, predict the unlabeled instances in $\mathcal{B}$ by S3VM and otherwise by SVM.
6: Predict the unlabeled data $\mathbf{x} \notin \mathcal{B}$ by SVM.

1. Multiple Classifiers Analysis

LEAD (Yu-Feng Li, IJCAI, 2016)

- **Idea**: when a certain graph owns a high quality, its predictive results on the unlabeled data may have a large margin separation

5NN Graph with Euclidean Distance		5NN Graph with Manhattan Distance	
Accuracy	Hinge Loss	Accuracy	Hinge Loss
91.6±3.1	0.529±0.110	92.6±2.7	0.370±0.106
60.9±6.2	0.341±0.109	62.4±6.9	0.276±0.109
57.7±4.9	0.632±0.139	56.4±4.5	0.671±0.145
52.3±3.3	0.964±0.006	50.3±0.0	0.994±0.009

- **Description**:
  - LEAD need T SSL classifiers and one SL classifier
  - Target to find a best label assignment that consistent with T Classifiers and labeled data

LEAD (Yu-Feng Li, IJCAI, 2016)

- **Description**:
  - LEAD need T SSL classifiers and one SL classifier
  - Target to find a best label assignment Z that consistent with T Classifiers and labeled data

$$\begin{aligned} \min_{\mathbf{z}} \quad & \sum_{e_{ij}\in\mathcal{E}} w_{ij}\|z_i - z_j\|^2 \\ \text{s.t.} \quad & z_i = y_i,\ i = 1,\ldots,l; \\ & z_j \in [-1,1],\ j = l+1,\ldots,l+u; \end{aligned}$$

  - Where ε is edge set
  - In fact, LEAD just apply S3VM on the predictions!

LEAD (Yu-Feng Li, IJCAI, 2016)

- In fact, LEAD just apply S3VM on the predictions!

$$\min_{\mathbf{w},\hat{\mathbf{y}}} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C_1 \sum_{i=1}^{l} \ell(y_i f(\mathbf{u}_i)) + C_2 \sum_{j=l+1}^{l+u} \ell(\hat{y}_j f(\mathbf{u}_j))$$

$$\text{s.t.} \quad \hat{y}_{l+j} \in \{+1, -1\}, \quad j = 1, \ldots, u;$$

$$\left| \frac{\sum_{j=l+1}^{l+u} \hat{y}_j}{u} - \frac{\sum_{i=1}^{l} y_i}{l} \right| \leq \beta \qquad (2$$

  - Where $\mathrm{u_i} = [\mathrm{z_i}^{(1)}, \ldots, z_i^{(T)}]$ is the prediction value of T classifiers for instance $\mathrm{x_i}$
  - However, LEAD use its own algorithm to solve (2), where it need additional SL information

LEAD (Yu-Feng Li, IJCAI, 2016)

- **Method**:
  - ✓ Perform GSSL on a set of graphs $\{G_t\}_t^T$
  - ✓ Generate new training data via $\{\mathrm{u_i}, y_i\}\ and\ \{u_i\}$, where $\mathrm{u_i} = [\mathrm{z_i}^{(1)}, \dots, z_i^{(T)}]$
  - ✓ Init $\hat{y} = sign\left(\frac{1}{T}\sum_{t=1}^{T} z^{(t)}\right), C_2 = 10^{-6}C_1$
  - ✓ Repeat until $C_2 \geq threshold$
    - ✓ Fix $\hat{y}$, update w by solving SVM problem
    - ✓ Fix w, update $\hat{y}$ by $\hat{y} = \begin{cases} 1 \ \ if\ r_j \leq \left(\frac{2\sum_{i=1}^{l} y_i}{l} - \beta\right)u \\ -1 \ \ if\ r_j \geq \left(\frac{2\sum_{i=1}^{l} y_i}{l} + \beta\right)u \\ sign\left(w'u_{l+j} + b\right)\ \ otherwise \end{cases}$
    - ✓ If $\widehat{y_{l+j}}\left(w'u_{l+j} + b\right) \leq 1$, then $\widehat{y_{l+j}} = y_{l+j}^{SVM}$

r is the rank of prediction on unlabeled in descending order

UMVP (Yu-Feng Li, AAAI, 2016)

- **Idea**: Integrating multiple SSL via maximizing performance gain w.r.t. performance criteria in the worst-case scenario.
- **Description**:
  - Using multiple SSL predictions, find a best

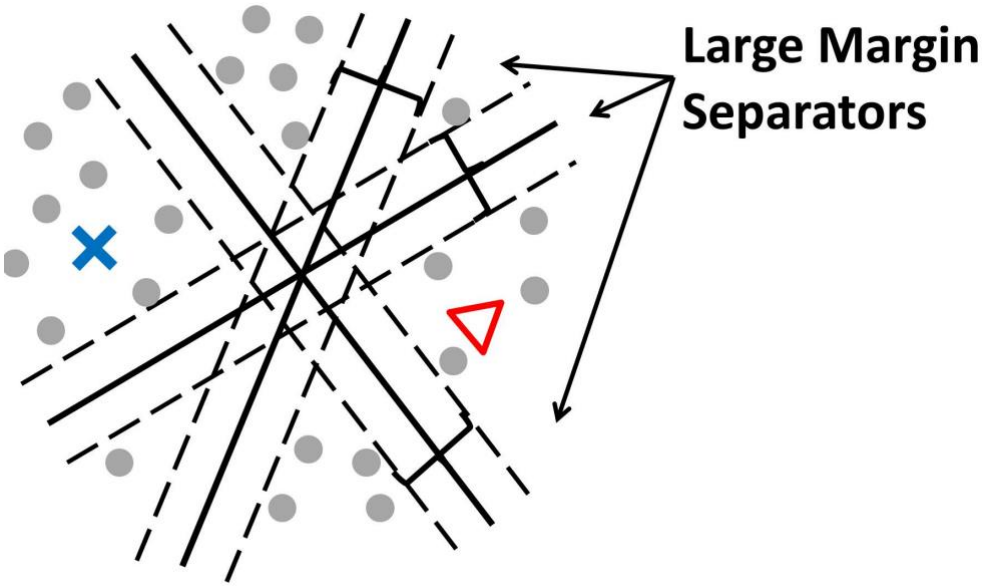
$$\max_{\hat{\mathbf{y}} \in \mathcal{Y}} \min_{\boldsymbol{\alpha} \in \mathcal{M}} \sum_{i=1}^{b} \alpha_i \left(perf(\hat{\mathbf{y}}, \mathbf{y}^i) - perf(\hat{\mathbf{y}}_0, \mathbf{y}^i) \right).$$

  - Where y_0 is SL
  - $pref$ is F1/AUC/Top-k Precision
  - a is the SSL weight, sum up to 1

- Experiments are performed on those criteria O_O" …

S4VM (Yu-Feng Li, ICML_2011, PAMI_2015)

- **Idea**: exploit the candidate low-density separators to reduce the risk of identifying a poor separator with unlabeled data.
- **Assumption**:
  - If there is a separator that reaches the ground truth, then S4VM never degeneration



S4VM (Yu-Feng Li, ICML_2011, PAMI_2015)

- **Method**:
  - Find T separators that satisfying S3VM condition and minimizing the penalty on diversity of separators

$$h(f,\hat{\mathbf{y}}) = \frac{\|f\|_{\mathcal{H}}}{2} + C_1 \sum_{i=1}^{l} \ell(y_i, f(\mathbf{x}_i)) + C_2 \sum_{j=1}^{u} \ell(\hat{y}_j, f(\hat{\mathbf{x}}_j)).$$

$$\min_{\{f_t, \hat{\mathbf{y}}_t \in \mathcal{B}\}_{t=1}^{T}} \sum_{t=1}^{T} h(f_t, \hat{\mathbf{y}}_t) + M\Omega(\{\hat{\mathbf{y}}_t\}_{t=1}^{T}),$$

$$\Omega(\{\hat{\mathbf{y}}_t\}_{t=1}^{T}) = \sum_{1 \le t \ne \tilde{t} \le T} \mathbf{I}\left(\frac{\hat{\mathbf{y}}_t' \hat{\mathbf{y}}_{\tilde{t}}}{u} \ge 1 - \epsilon\right)$$

  - Finally, classifier set to be
    - ✓ Combine by equal weights
    - ✓ Choose the one that minimizing target function

2. Trade off between SSL and SL

RsLapRLS (Haitao Gan, Expert Systems with Applications,2016)

- **Idea**: confidence & consistence tradeoff between SSL and SL
- **Definition** for unlabeled data:
  - Confidence → $\mathrm{cf}(\mathrm{x_j}) = |\hat{y}| - |\bar{y}|$
  - Consistence → $cs(x_j) = sign(\hat{y}.\bar{y})$
  - Risk → $s_j = \begin{cases} \exp\{-cf(x_j)\} & if\ cs(x_j) = 1\ and\ cf(x_j) \neq 0 \\ \exp\{|cf(x_j)|\} & if\ cs(x_j) = -1\ and\ cf(x_j) \neq 0 \\ \exp\{-cs(x_j)\} & if\ cf(x_j) = 0 \end{cases}$
- **Target**:

$$\mathcal{Q}(f) = \sum_{i=1}^{l} (f(x_i) - y_i)^2 + \gamma_A \|f\|_K^2 + \gamma_I f^T L f$$

$$+ \lambda \sum_{j=l+1}^{n} s_j \|f(x_j) - g(x_j)\|_2^2$$

2. Trade off between SSL and SL

SA-SSCCM (Yunyun Wang, IEEE trans,2013)

- **Idea**: tradeoff between SSL and SL by man
- **Target**:

$$\min_{f,v_k(x_j)} \; ||f||_{\mathcal{H}}^2 + \lambda_1 \sum_{i=1}^{n_l} ||f(x_i) - y_i||^2$$
$$+\lambda_2 \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} v_k(x_j)^2 ||f(x_j) - r_k||^2$$
$$+\left(\frac{1}{\lambda} - 1\right) \sum_{j=n_l+1}^{n} ||f(x_j) - g(x_j)||^2.$$

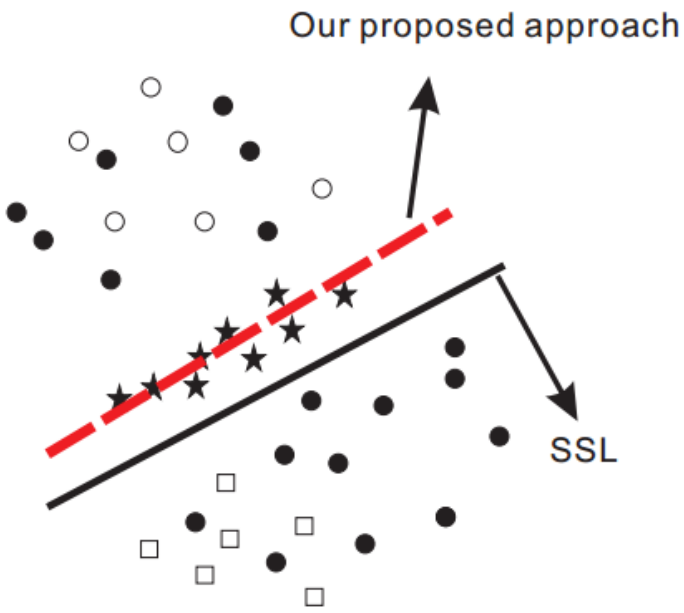
  - Where v is a unknown variable, indicating different loss for assigning x_j to class k

$$v_k(x_j) = \frac{1/||f(x_j) - r_k||^2}{\sum_{k=1}^{C} 1/||f(x_j) - r_k||^2}$$

3. To clear the margin

USSL (Zenglin Xu, ICDM, 2008)

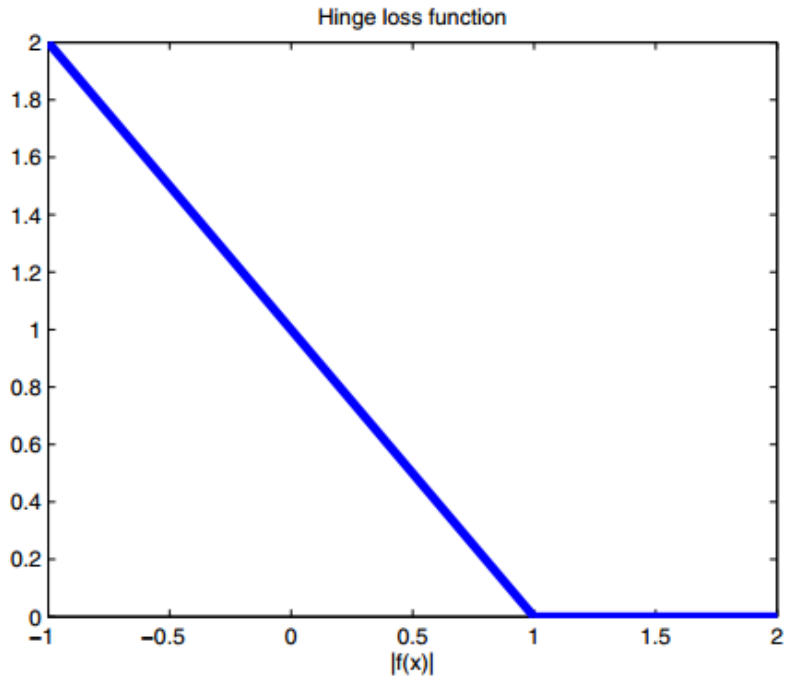
- **Motivation**: classification on universum data without any priori information
- **Idea**:
  - Not force to label all unlabeled data
  - Some of unlabeled data are useful
  - If x_j suffer too much classification loss
    Why not to un-classify it

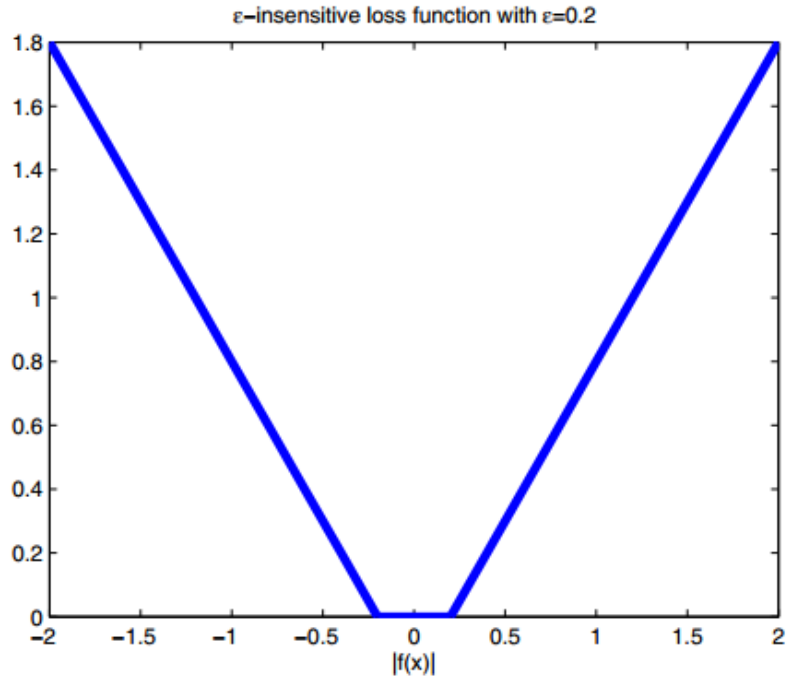


3. To clear the margin

USSL (Zenglin Xu, ICDM, 2008)

- **Method**: S3VM + ε **insensitive loss**





3. To clear the margin

USSL (Zenglin Xu, ICDM, 2008)

- **Method**: S3VM + ε insensitive loss

$$\min_{\mathbf{w},b,\xi,\eta,\mathbf{y}_{l+1:n}} \quad \frac{1}{2}||\mathbf{w}||^2 + C_L \sum_{i=1}^{l} \xi_i + C_U \sum_{j=l+1}^{n} \min(\eta_j, \xi_j)$$

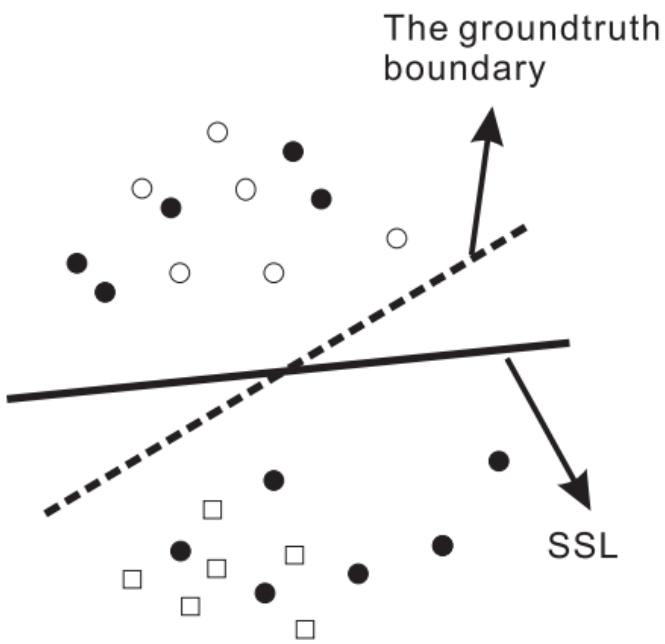
$$\begin{aligned}
\text{s.t.} \quad & y_i(\mathbf{w}_i \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, \ldots, l, && (1)\\
& y_j(\mathbf{w}_j \cdot \mathbf{x}_j + b) \geq 1 - \xi_j, && (2)\\
& |\mathbf{w}_j \cdot \mathbf{x}_j + b| \leq \varepsilon + \eta_j, && (3)\\
& \eta_j \geq 0, j = l+1, \ldots, n,\\
& \xi_k \geq 0, k = 1, \ldots, n,
\end{aligned}$$

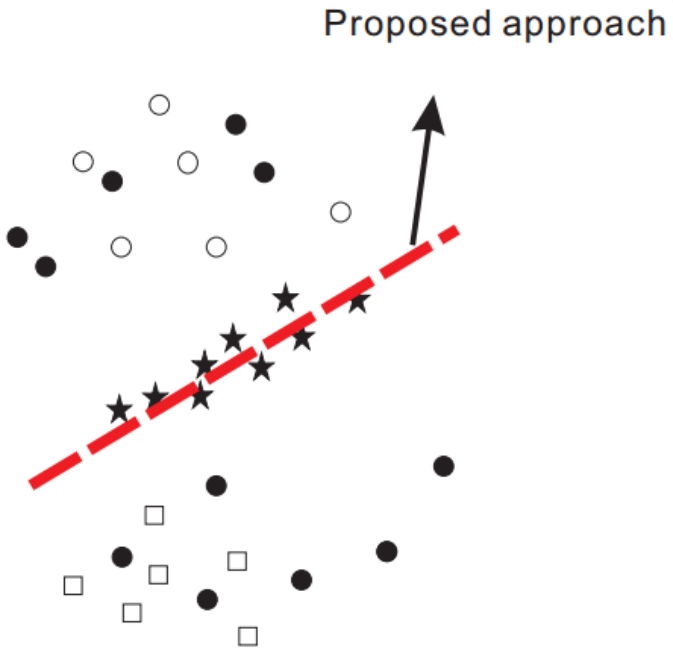
- “Useful” unlabeled data should far away from separating plane ;
- “Useless” unlabeled data should have small loss

3. To clear the margin

USSL (Zenglin Xu, ICDM, 2008)

- **Method**: S3VM + ε insensitive loss
- The "irrelevant" data can increase the performance when only a limited number of relevant unlabeled data is available





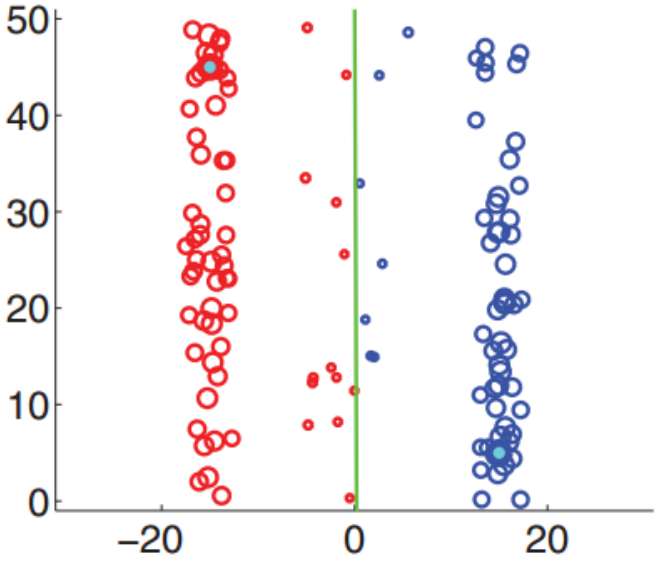
3. To clear the margin

ASL (De Wang, KDD, 2014)

- **Idea**: loss of assigning x_i to class k is different
- **Method**:

$$\min_{W,b,Y} \left\| X_l^T W + \mathbf{1}_{nl} b^T - Y_l \right\|_F^2 + \sum_{i=1}^{n} \sum_{k=1}^{c} y_{ik}^r \left\| x_i^T W + b^T - t_k^T \right\|_F^2$$

$$s.t. \ \ \forall i, y_{ik} \in [0,1], \sum_{k=1}^{c} y_{ik} = 1 \qquad ($$



4. Adaptive Weighting

TLAG (Yan-Ming Zhang, AAAI, 2010)

- **Idea**: adaptively construct graphs, updating the edge weight via current prediction results
- **Description**:
  - TLAG iteratively learns a new graph G' and build a classifier on G' instead of original G
- **Target**:

$$\min_{\mathbf{f}\in\mathbb{R}^n,\tilde{\mathbf{L}}\succeq 0} h(\mathbf{f},\tilde{\mathbf{L}}) = \mathbf{f}^T\tilde{\mathbf{L}}\mathbf{f} + (\mathbf{f}-\mathbf{y})^T\mathbf{C}(\mathbf{f}-\mathbf{y}) + \beta D(\tilde{\mathbf{L}},\mathbf{L}),$$

  - Where D is LogNet, used for measuring the dissimilarity between G and G'

$$D_{ld}(\mathbf{A},\mathbf{B}) = \mathrm{tr}(\mathbf{A}\mathbf{B}^{-1}) - \log\det(\mathbf{A}\mathbf{B}^{-1}) - n.$$

4. Adaptive Weighting

TLAG (Yan-Ming Zhang, AAAI, 2010)

- **Target**:

$$\min_{\mathbf{f}\in\mathbb{R}^n,\tilde{\mathbf{L}}\succeq 0} h(\mathbf{f},\tilde{\mathbf{L}}) = \mathbf{f}^T\tilde{\mathbf{L}}\mathbf{f} + (\mathbf{f}-\mathbf{y})^T\mathbf{C}(\mathbf{f}-\mathbf{y}) + \beta D(\tilde{\mathbf{L}},\mathbf{L}),$$

- Update G' by

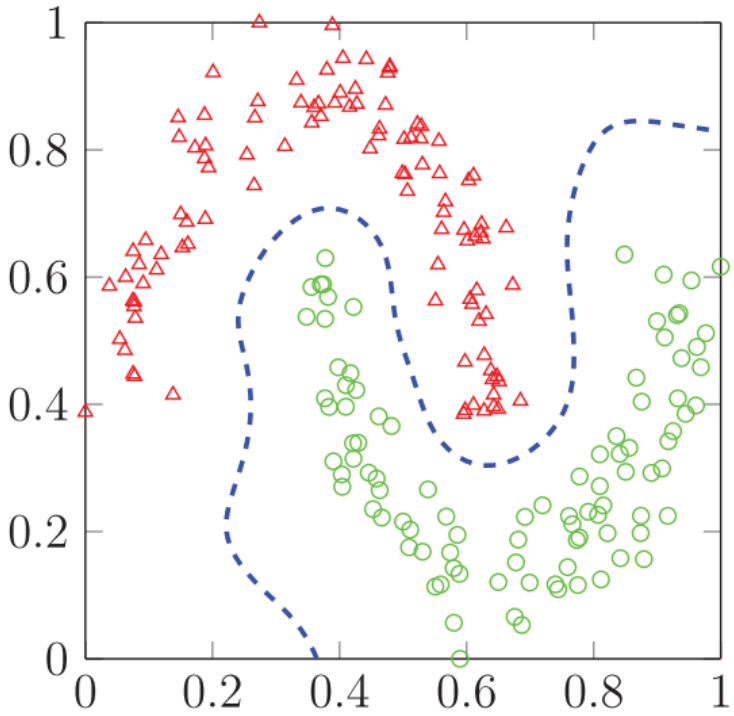
$$\tilde{\mathbf{L}} = \left(\frac{1}{\beta}\mathbf{f}\mathbf{f}^T + \mathbf{L}^{-1}\right)^{-1}.$$

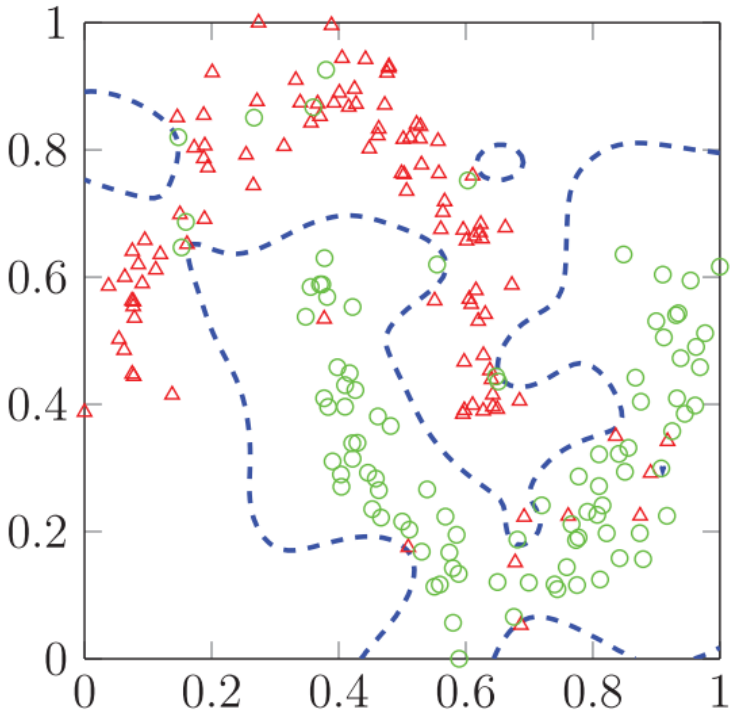
  - x_i and $\mathrm{x_j}$ get more similar if they share the same label

4. Adaptive Weighting

* RSVC (Yunlong Feng, Neural computation, 2016)

- **Notice**: It is not a SSL but SL algorithm, RSVC Indeed learns the instances weights.
- **Target**: label-noise robust SL

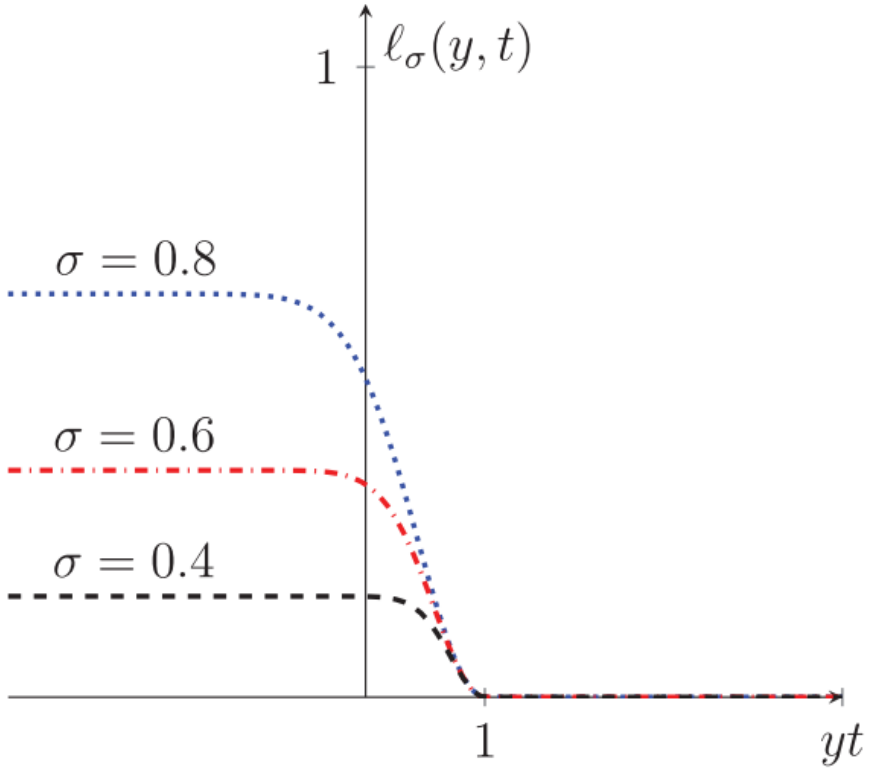




* RSVC (Yunlong Feng, Neural computation, 2016)

- **Description**:
  - Proposed a robust and smooth loss function family, example

$$\ell_\sigma(y,t) = \sigma^2(1-\exp(-(1-yt)_+^2/\sigma^2)),\ y \in \mathcal{Y}, t \in \mathbb{R},$$



4. Adaptive Weighting

* RSVC (Yunlong Feng, Neural computation, 2016)

- **Description**:
  - Proposed a robust and smooth loss function family
  - Convergence analysis & statistical properties are guaranteed
  - Sharing the same solution with re-weighted L2-SVM

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^m, b\in\mathbb{R}} \frac{1}{m}\sum_{i=1}^{m}\phi(y_i, \boldsymbol{\mathcal{K}}_i^\top\boldsymbol{\alpha}+b)+\lambda\boldsymbol{\alpha}^\top\boldsymbol{\mathcal{K}}\boldsymbol{\alpha}.$$

$$(\boldsymbol{\alpha}^{k+1}, b^{k+1}) = \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^m, b\in\mathbb{R}} \sum_{i=1}^{m}\omega_i^{k+1}(y_i-\boldsymbol{\mathcal{K}}_i^\top\boldsymbol{\alpha}-b)_+^2+\lambda\boldsymbol{\alpha}^\top\boldsymbol{\mathcal{K}}\boldsymbol{\alpha},$$

$$\omega_i^{k+1} = \exp(-(y_i-\boldsymbol{\mathcal{K}}_i^\top\boldsymbol{\alpha}^k-b^k)_+^2/\sigma^2),\ i=1,\ldots,m.$$

Part Three

Weights Learning for Unlabeled Data

番外篇

First-order Methods for Convex Problem

First-order Methods

- Gradient Descent
- Subgradient Methods
- Proximal Gradient Method

* **简单说说上面这些方面**
  - 为什么只说**这些**？因为别的我也不会…
  - 为什么只**简单**说说？因为深入的我也不会…

Gradient Descent

- 梯度下降(Gradient Descent)，无约束优化中的常用方法
- 假设函数f(x)是**可微的光滑凸函数**，满足$\mathrm{dom}(f) = R^n$，下式的优化目标形式成为无约束优化

$$\min f(x)$$

- 怎么解 → 求导~！→ $x^* = \arg\min_x f(x)$
- $\nabla f(x^*) = 0$不好解怎么办
  - 我们可以采用迭代的方式，找出一组x的序列$x^{(0)}, x^{(1)}, \ldots$.满足当$k \to \infty$时，$f(x^{(k)}) \to f(x^*)$

- **梯度下降**便是这样的迭代算法！

$$x^{(k)} = x^{(k-1)} - t_k \nabla f(x^{(k-1)})$$

➢迭代公式咋来的呢？

假设函数f是(二阶)可微的，那么用泰勒公式进行展开，其中$0 \le \theta \le 1$：

$$f(\mathrm{y}) \approx \mathrm{f(x)} + \nabla \mathrm{f(x)}^{\mathrm{T}}(y-x) + \frac{1}{2}(y-x)^{T}\nabla^{2} f(\theta(x-y)+y)(y-x)$$

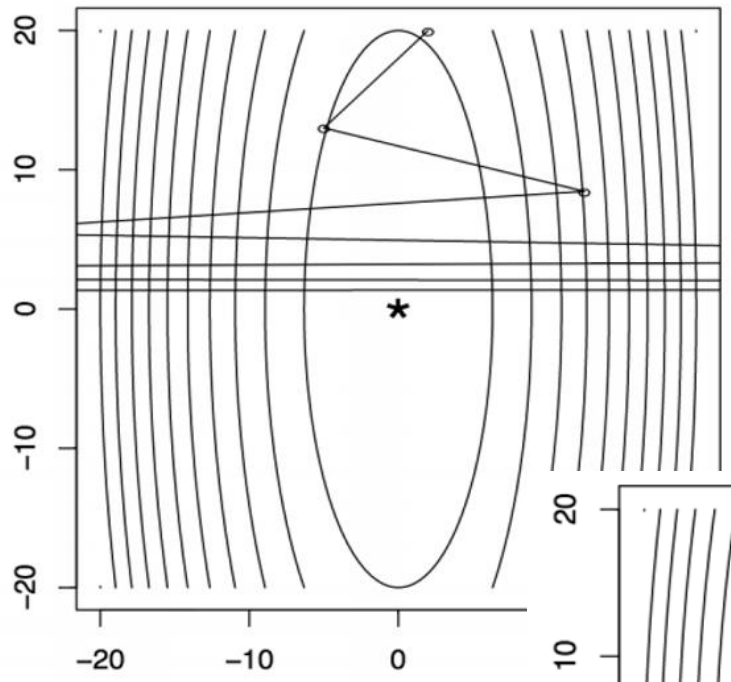
现在对$\nabla^{2} f(\theta(x-y)+y)$用$\frac{1}{\mathrm{t}} I$进行二阶近似，那么得到

$$\mathrm{f(y)} \approx \mathrm{f(x)} + \nabla \mathrm{f(x)}^{\mathrm{T}}(y-x) + \frac{1}{2\mathrm{t}}\|y-x\|_2^2 = g(y)$$

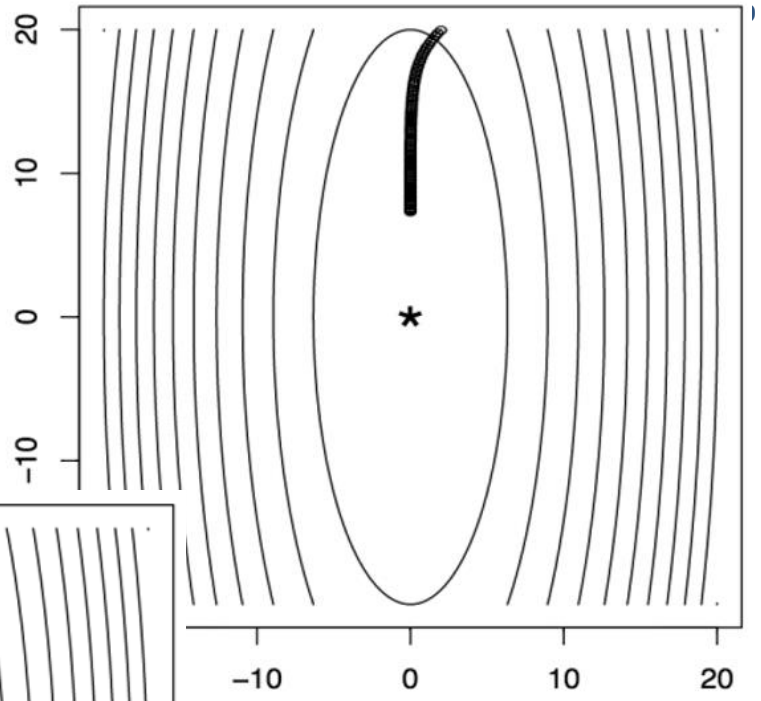
$$\nabla g(y) = 0$$

$$\nabla \mathrm{f(x)} + \frac{1}{\mathrm{t}}(y-x) = 0 \;\leftrightarrow\; y = x - t\nabla \mathrm{f(x)}$$

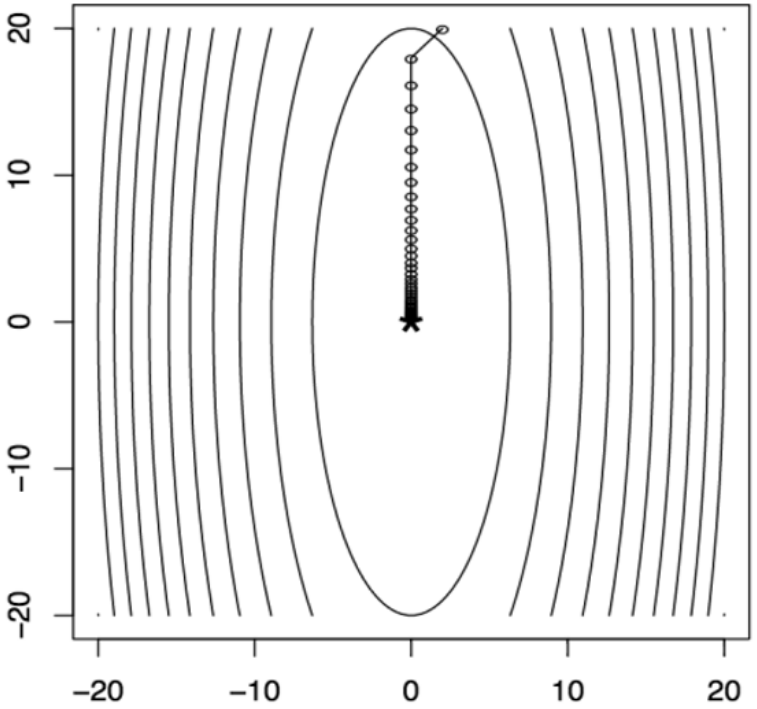
How to Choose Step



(a) t is too large (after 8



is too small (after 100 steps)



(c) t is good (after 40 steps)

Line Search

➢ **Exact Line Search (Not Practical)**

$$t = \arg\min_{s \geq 0} f(x - s\nabla f(x))$$

➢ **Backtracking Line Search**

✓ Given $0 < \beta < 1$ *and* $0 < \alpha \leq 0.5$

✓ At each iteration, start with t = 1,

if $\mathrm{f}\left(x - t\nabla f(x)\right) > f(x) - at\|\nabla f(x)\|_2^2$

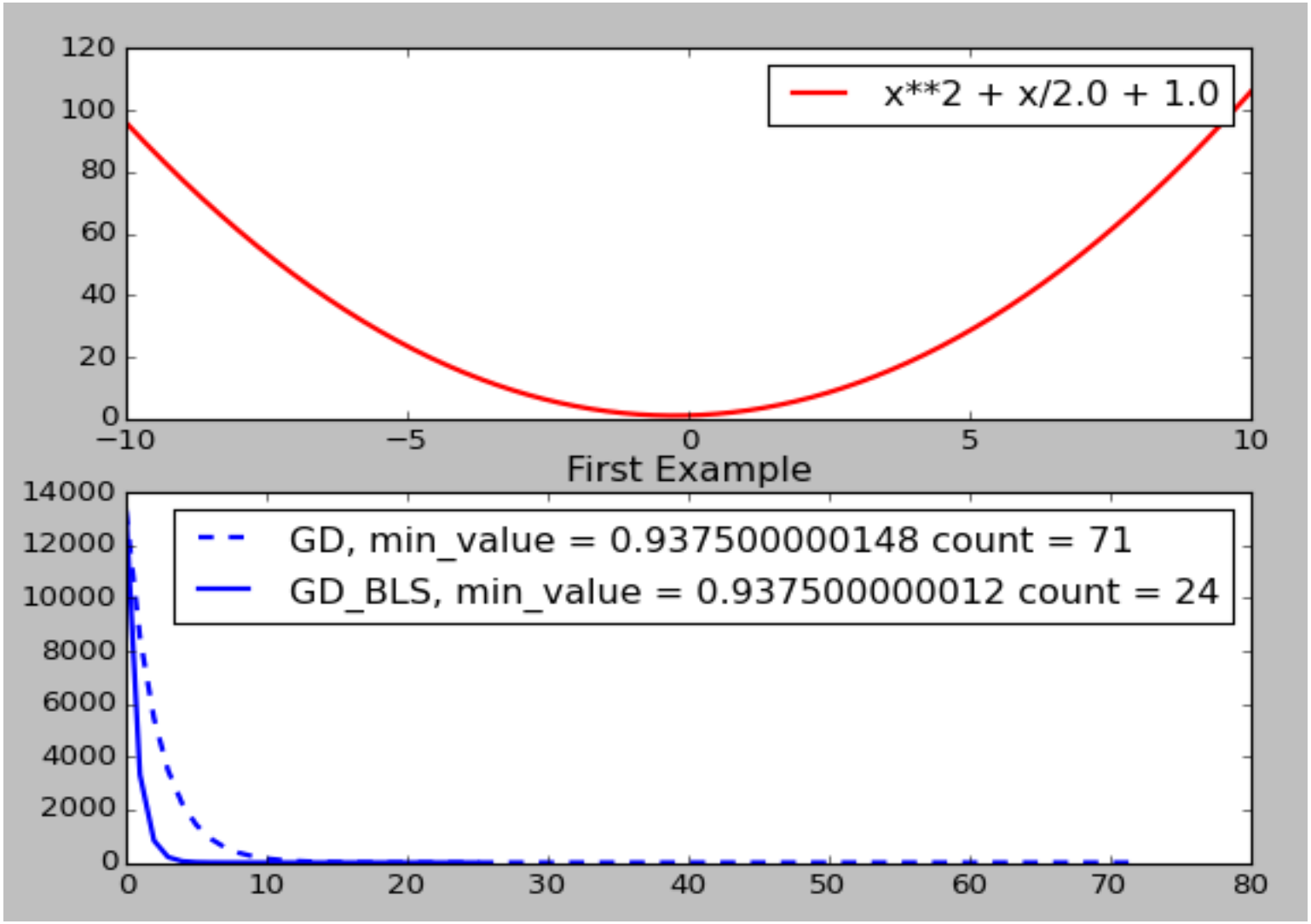
Shrink t = βt

else

update $x^{(k)} = x^{(k-1)} - t\,\nabla \mathrm{f}(x^{(k-1)})$

Backtracking Line Search

- Results



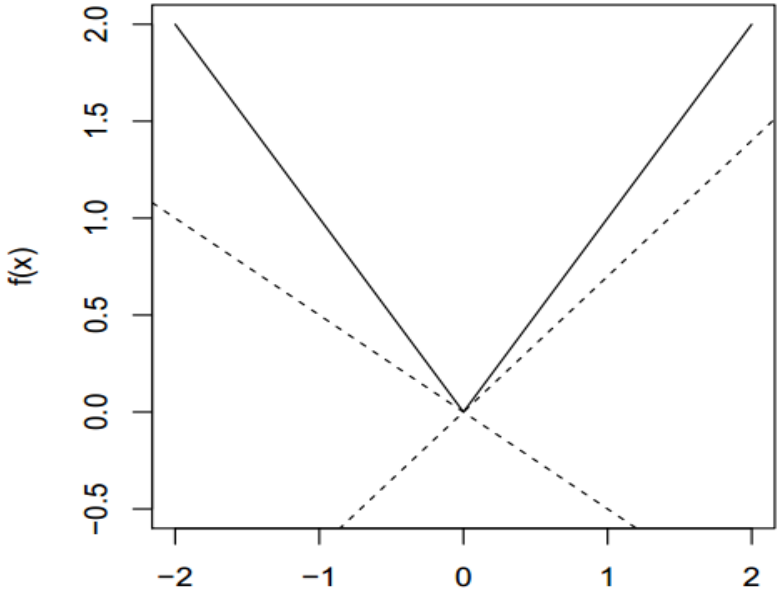
Subgradient Method

- 梯度下降虽然简单实用，但是遇到不可微的函数，那就捉襟见肘了。事实上在这种情况下，还有**次梯度(Subgradient)**的存在

凸函数f在点x处**subgradient**定义为，满足以下条件的**任意**$g \in R^n$

$$f(y) \geq f(x) + \nabla g^T \ (y - x) \qquad \forall y$$

记$\partial f(x)$表示在点x处，函数f的subgradient能取到的所有值的集合



Subgradient Method

- 梯度下降虽然简单实用，但是遇到不可微的函数，那就捉襟见肘了。事实上在这种情况下，还有**次梯度(Subgradient)**的存在

凸函数f在点x处**subgradient**定义为，满足以下条件的**任意**$g \in R^n$

$$f(y) \geq f(x) + \nabla g^T \ (y - x) \qquad \forall y$$

记$\partial f(x)$表示在点x处，函数f的subgradient能取到的所有值的集合

- 如果函数f在x点处可微，这点的subgradient是这点的gradient
- 对于凸函数，都存在subgradient，非凸函数，subgradient可能存在。
- 全局下估计，是gradient的概念升级版

Subgradient Method

- 举几个栗子

✧ 对于$\mathbf{f(x)} = |\mathbf{x}|$，$\partial \mathrm{f(x)} = \begin{cases} sign(x) & if\ x \neq 0 \\ any\ value\ in\ [-1,1] & otherwise \end{cases}$

✧ 对于$\mathbf{f(x)} = \|\boldsymbol{x}\|_{\mathbf{2}}$，$\partial \mathrm{f(x)} = \begin{cases} \frac{x}{\|x\|_2} & if\ x \neq 0 \\ any\ g\ such\ that\ \|g\|_2 \leq 1 & otherwise \end{cases}$

因为$x = 0$时，$\mathrm{f(y)} = \|y\|_2 \geq f(x) + g^T(y - x) = g^T y$

✧ 对于$\mathbf{f(x)} = \|\boldsymbol{x}\|_{\mathbf{1}}$，$\partial \mathrm{f(x)} = \begin{cases} sign(x) & if\ g_i \neq 0 \\ any\ value\ in\ [-1,1] & if\ g_i = 0 \end{cases}$

✧ 对于$\mathbf{L_p}\ \boldsymbol{Norm}\ \boldsymbol{f(x)} = [\![\boldsymbol{x}]\!]_{\boldsymbol{p}}$，$\partial \mathrm{f(x)} = [\![x]\!]_q$，满足$\frac{1}{\mathrm{p}} + \frac{1}{q} = 1$

Subgradient Optimality Condition

如果x^*是最优解，当且仅当在x^*处的subgradient的集合包含0

$$\mathrm{f}(\mathrm{x}^*) = \min_{\mathrm{x}} f(x) \leftrightarrow 0 \in \partial \mathrm{f}(\mathrm{x}^*)$$

$$\mathrm{f}(\mathrm{y}) \geq \mathrm{f}(\mathrm{x}^*) + 0^{\mathrm{T}}(y - x) \quad \forall y$$

- ✓ 这个条件，对于约束问题，同样适用

$$\min_{\mathrm{x}\in\mathrm{C}} f(x) \leftrightarrow \min_{x} f(x) + I_C(x)$$

- ✓ 要证明的话，会用到Normal cone的概念

$$0 \in \partial[\mathrm{f}(\mathrm{x}) + \mathrm{I}_{\mathrm{C}}(x)] \leftrightarrow 0 \in \partial \mathrm{f}(\mathrm{x}) + \mathrm{N}_{\mathrm{C}}(\mathrm{x}) \leftrightarrow -\partial \mathrm{f}(\mathrm{x}) \in \mathrm{N}_{\mathrm{C}}(\mathrm{x})$$
$$\rightarrow (\text{假设}\mathrm{f}(\mathrm{x})\text{可微}) \ -\nabla \mathrm{f}(\mathrm{x}) \in \mathrm{N}_{\mathrm{C}}(\mathrm{x}) \leftrightarrow -\nabla \mathrm{f}(\mathrm{x})^T x$$
$$\geq -\nabla \mathrm{f}(\mathrm{x})^T y \quad \forall \mathrm{y} \in C \leftrightarrow \nabla \mathrm{f}(\mathrm{x})^T (y - x) \geq 0 \quad \forall y \in C$$

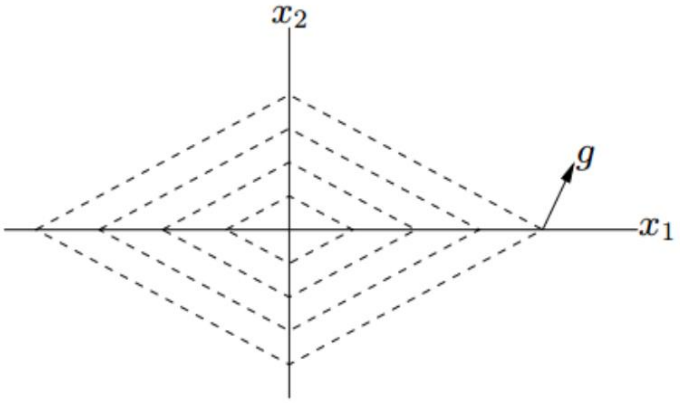
Subgradient Method

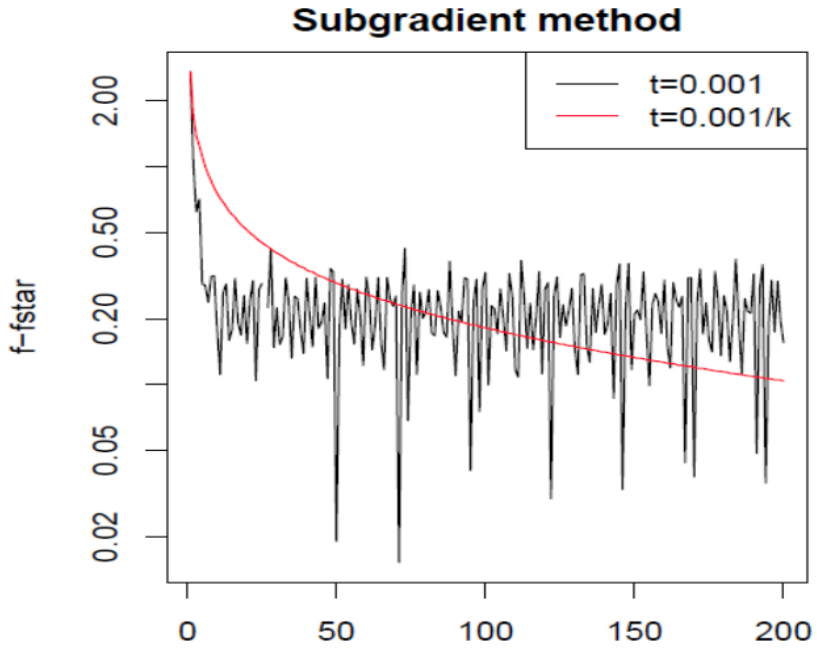
- **Update Rule**

$$x^{(k)} = x^{(k-1)} - t_k g^{k-1} \quad g^{k-1} \in \partial f(x^{(k-1)})$$

- BUT！它不能保证每步都下降哦，negative subgradient方向可能不是一个下降方向，如下（-1，-2）

example: $f(x_1, x_2) = |x_1| + 2|x_2|$





所以第k次迭代的最优值是

$$f\left(x_{best}^{(k)}\right) = \min_{0,1,\ldots,k} f(x^{(i)})$$

Step Choosing for Subgradient Method

- 没有像backtracking那样厉害的方法哎

- ✓ 固定步长法
- ✓ **Diminishing Step**：逐渐降低步长，但是步长的消散不能太快，避免多次迭代后，步长变成0，即要满足下面的条件。

$$\sum_{k=1}^{\infty} t_k = \infty \qquad \sum_{k=1}^{\infty} t_k^2 < \infty$$

那要怎么选呢？

$$t = 1/k$$

Subgradient这么厉害，赶紧尝试用一下！

Subgradient Method for Lasso

➢ 目标函数

$$\min_{\beta} f(x) = \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

$$\partial f(x) = \partial\left(\frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1\right) = -X^T(y - X\beta) + \lambda\partial\|\beta\|_1 = 0$$

$$\rightarrow \quad X^T(y - X\beta) = \lambda\boldsymbol{\partial}\|\boldsymbol{\beta}\|_{\mathbf{1}}$$

$$\partial\|\beta\|_1 = \begin{cases} 1 & if \ \ \beta_i > 0 \\ -1 & if \ \ \beta_i < 0 \\ [-1,1] & if \ \ \beta_i = 0 \end{cases}$$

$$\begin{cases} X_i^T(y - X\beta) = \lambda sign(\beta_i) & if \ \ \beta_i \neq 0 \\ \left|X_i^T(y - X\beta)\right| \leq \lambda & if \ \ \beta_i = 0 \end{cases}$$

Subgradient Method for Lasso

- 已知

$$\begin{cases} X_i^T(y - X\beta) = \lambda sign(\beta_i) & if \ \ \beta_i \neq 0 \\ \left|X_i^T(y - X\beta)\right| \leq \lambda & if \ \ \beta_i = 0 \end{cases}$$

- 现在我们把问题再简化一下，考虑X = I的情况，方便我们求解，此时，上面的式子简化为

$$\begin{cases} y_i - \beta_i = \lambda sign(\beta_i) & if \ \ \beta_i \neq 0 \\ |(y_i - \beta_i)| \leq \lambda & if \ \ \beta_i = 0 \end{cases}$$

这个形式在凸优化中称为
soft-thresholding operator

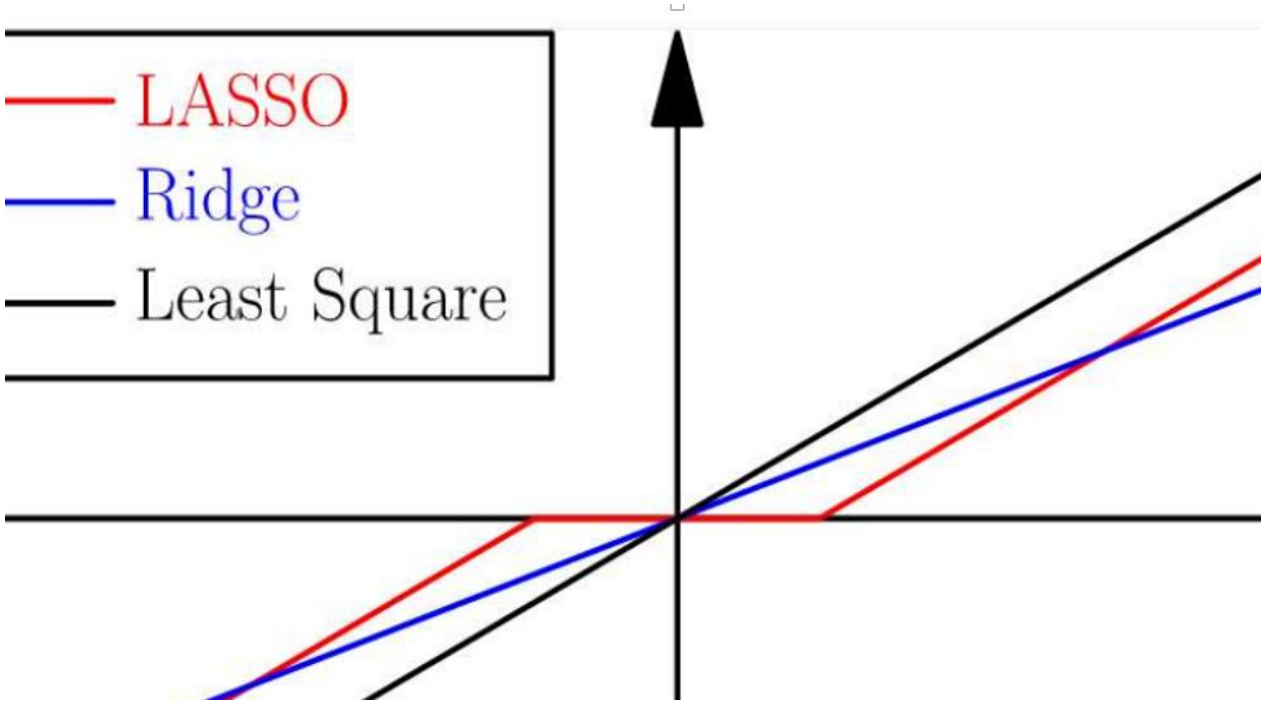
$$S_\lambda(\beta)_\mathrm{i} = \begin{cases} y_i - \lambda & if \ y_i > \lambda \\ 0 & if \ -\lambda \leq y_i \leq \lambda \\ y_i + \lambda & if \ y_i < -\lambda \end{cases}$$

Subgradient Method for Lasso

- 你还记得大明湖畔的Lasso吗？

$$S_\lambda(\beta)_\mathrm{i} = \begin{cases} y_i - \lambda & if\ y_i > \lambda \\ 0 & if\ -\lambda \le y_i \le \lambda \\ y_i + \lambda & if\ y_i < -\lambda \end{cases}$$

这不就稀疏解了嘛？



Proximal Gradient Descent

- 事实上，Lasso问题的经典算法是**ISTA**，而不是单纯的使用subgradient，**Why？？**

对于不可微的凸函数f，前面我们介绍的subgradient method，从收敛性的角度来说是不可观的。假设，我们想要误差精度$\varepsilon = 0.001$，即$f\left(\mathrm{x}^{(k)}\right) - \mathrm{f}^* \leq \varepsilon$，梯度下降要迭代O(1000)次，subgradient method却要用1百万迭代！那么需要如何**加速收敛**呢？

Tips (under certain conditions):

1. Gradient descent有$O(1/\varepsilon)$的收敛率
2. Subgradient Method有$O(1/\varepsilon^2)$的收敛率
3. Proximal gradient descent方法，保证$O(1/\varepsilon)$的收敛率！
4. Accelerated PGD有$O(1/\varepsilon^{1/2})$的收敛率

Proximal Gradient Descent

现在考虑函数f，可以分解成两个函数的和，其中g(z)是可微的凸函数，$\mathrm{dom}(g) = R^n$，h(z)是**形式简单**，不一定可微的凸函数

$$f(z) = g(z) + h(z)$$

既然g可微，那就泰勒展开他！

$$x^+ = argmin_z \; g(x) + \nabla g(x)^T(z-x) + \frac{1}{2t}\|z-x\|_2^2 + h(z)$$

$$x^+ = \boldsymbol{argmin_z} \; \frac{\left(t\nabla g(x)\right)^2}{2t} + \nabla g(x)^T(z-x) + \frac{1}{2t}\|z-x\|_2^2 + h(z)$$

$$= \frac{1}{2t}\|(z-x) + t\nabla g(x)\|_2^2 + h(z)$$

$$= \frac{1}{2t}\left\|z - (x - t\nabla g(x))\right\|_2^2 + h(z)$$

Proximal Gradient Descent

- 定义**proximal mapping**为关于h和t的函数，即

$$\mathrm{Prox}_{\mathrm{h,t}}(\mathrm{x}) = \mathrm{argmin}_{\mathrm{z}} \frac{1}{2t}\|z-x\|_2^2 + h(z)$$

$$\mathrm{x}^+ = \frac{1}{2t}\left\|z-\left(x-t\nabla g(x)\right)\right\|_2^2 + h(z) = \mathrm{Prox}_{\mathrm{h,t}}\left(x-t\nabla g(x)\right)$$

- 可以看出$\mathrm{Prox}_{\mathrm{h,t}}(\mathrm{x})$是与g(x)无关的，只有h(x)有关，所以，计算$\mathrm{Prox}_{\mathrm{h,t}}(\mathrm{x})$的复杂度很大程度上依赖h(x)。
- $\mathrm{Prox}_{\mathrm{h,t}}(x-t\nabla g(x))$从形式上看，这个式子的最优解z使得第一项逼近g的梯度下降值，并且也使得第二项变小

Proximal Gradient Descent

- 给定初始点$x^{(0)}$，第k次的迭代点为

$$x^{(k)} = \mathrm{Prox}_{h,t_k}\left(x^{(k-1)} - t_k \nabla g\left(x^{(k-1)}\right)\right)$$

- 或者可以写成，关于函数f的**generalized gradient** $G_t(x)$的形式

$$x^{(k)} = x^{(k-1)} - t_k G_{t_k}\left(x^{(k-1)}\right)$$

$$where\ G_t(x) = \frac{x - Prox_{h,t}\left(x - t\nabla g(x)\right)}{t}$$

- 是不是已经昏了？让大明湖畔的Lasso叫醒你！

Iterative Soft Thresolding Algorithm

- 目标函数

$$\min_{\beta} f(\beta) = \underbrace{\frac{1}{2}\|y - X\beta\|_2^2}_{g(\beta)} + \underbrace{\lambda\|\beta\|_1}_{h(\beta)}$$

- 贴心的把公式再写一遍！

$$\mathrm{Prox}_{h,t}(x) = \mathrm{argmin}_z \frac{1}{2t}\|z - x\|_2^2 + h(z)$$

proximal mapping的解就是Soft-thresholding operator $S_{\lambda t}(\beta)$

$$\mathrm{Prox}_{h,t}(\beta) = \mathrm{argmin}_z \frac{1}{2t}\|z - \beta\|_2^2 + \lambda\|\beta\|_1$$

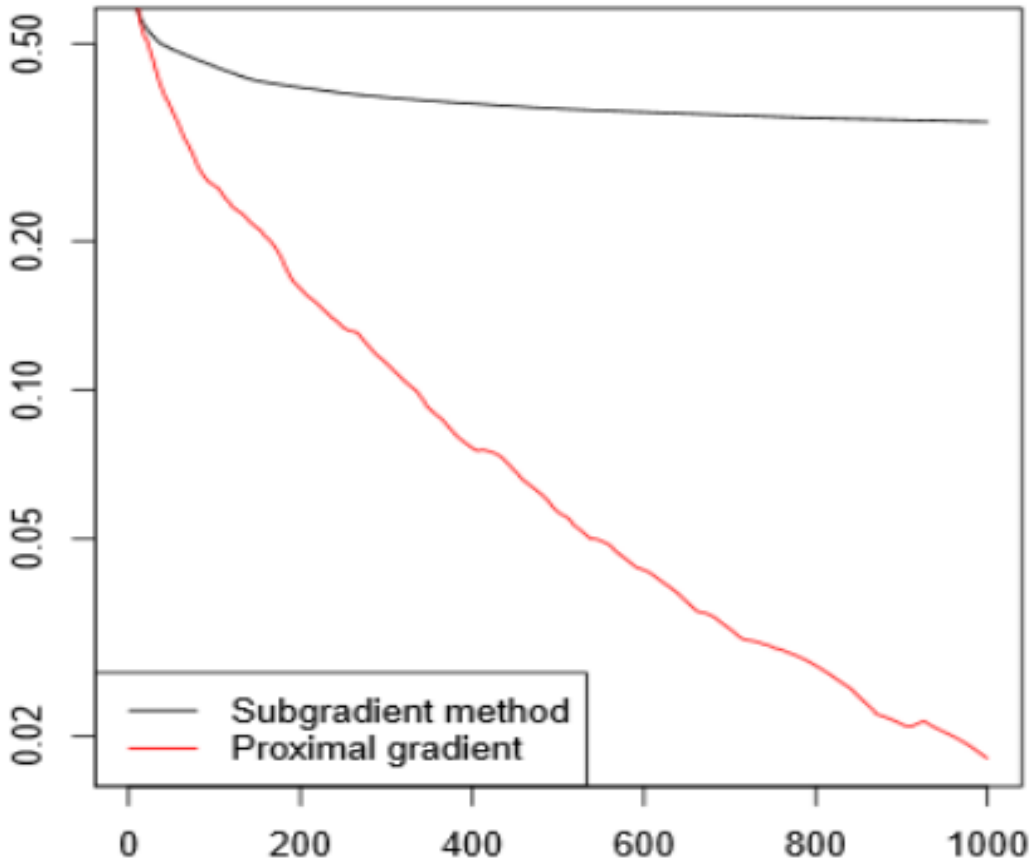
$$\leftrightarrow \mathrm{argmin}_z \frac{1}{2}\|z - \beta\|_2^2 + \lambda t\|\beta\|_1 = S_{\lambda t}(\beta)$$

不然，你以为这个算法的名字怎么来的？

Iterative Soft Thresolding Algorithm

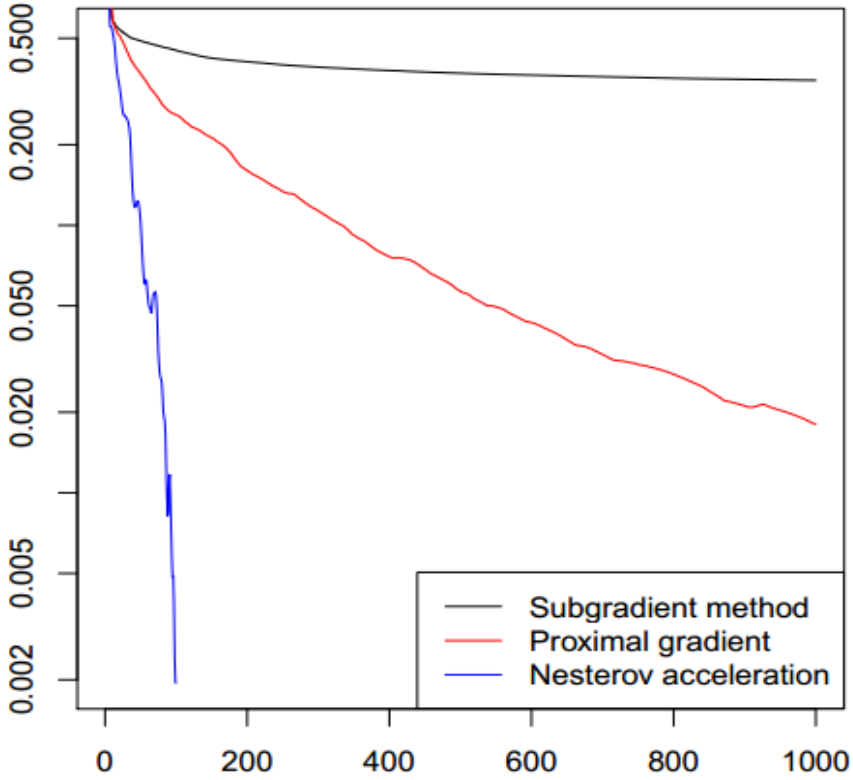
- 迭代公式

$$\beta^{+} = \mathrm{Prox}_{\mathrm{h,t}}\left(x - t\nabla g(\beta)\right) = S_{\lambda t}\left(\beta + tX^{T}(y - X\beta)\right)$$



Fast Iterative Soft Thresolding Algorithm

- 多了一个加速的步骤



$$\mathrm{v} = \mathrm{x}^{(k-1)} + \frac{k-2}{k+1}\left(\mathrm{x}^{(k-1)} - \mathrm{x}^{(k-2)}\right)$$

$$\mathrm{x}^{(k)} = Prox_{t_k}\left(v - t_k \nabla g(v)\right)$$

* Lasso In Dual View

- **Dual Problem**
- **Conjugate Function** (Thus, no need to calculate subgradient…)

$$\min_{\beta \in R^p} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

$y - u = X\beta$ ⟷ **stationarity condition**

$$\max_u \frac{1}{2}\|y\|_2^2 - \frac{1}{2}\|y - u\|_2^2$$
$$s.t.\ \|X^T u\|_\infty \leq \lambda$$

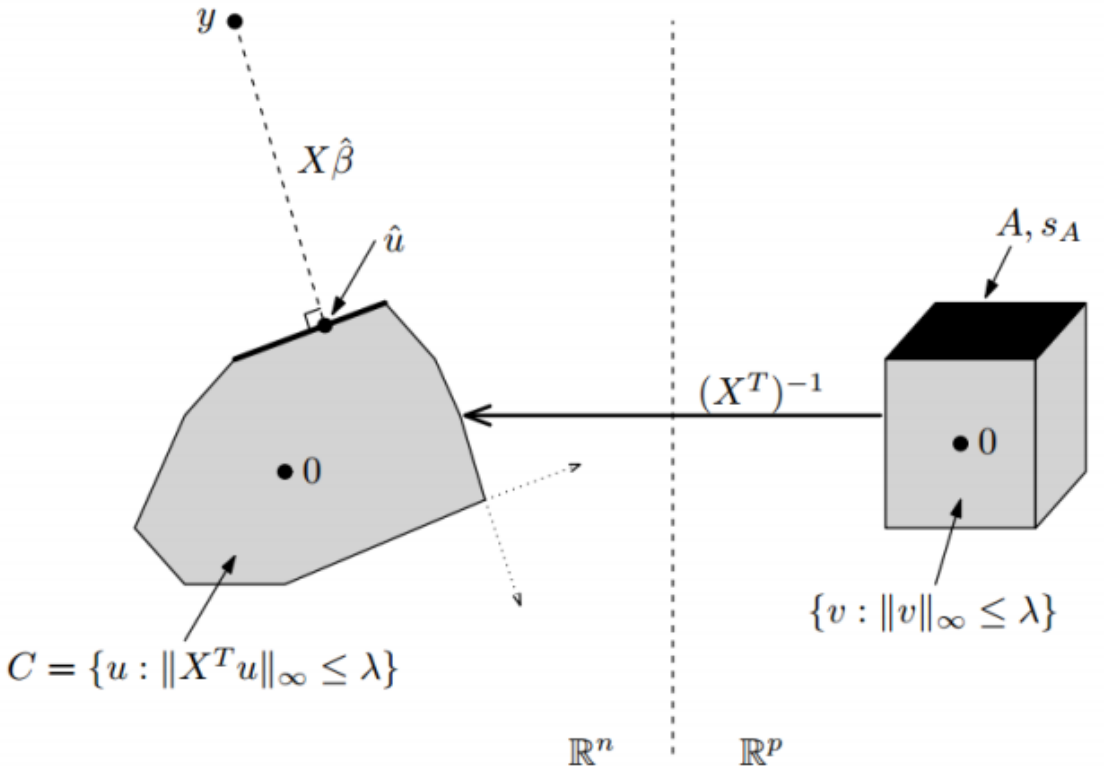
⟷ **The distance from y to a convex set/polyhedron**

* Lasso In Dual View

➢ **Dual Problem + Conjugate Function**

$$\min_{u \in C} \frac{1}{2}\|y-u\|_2^2 \quad C = \{u: \|X^T u\|_\infty \leq \lambda\}$$

Min-distance from y to a convex set/polyhedron



$$y - u = X\beta$$

Figure 13.2: Lasso dual problem

Next?

- Stochastic/Batch Subgradient Method
- Duality and KKT
- Second-order Methods
  - Newton Method
  - Barrier Method
  - Primal-dual Interior Point Method
  - Proximal Newton Method
- Coordinate Descent
- Non-convex Problem
- ……

然而我并不会。。。

Thanks

By HC