LEC3 DSA5103 $\rightarrow$

- Logistic Reg $\Rightarrow$ classification (binary)
- Regularization
  - LASSO $\rightarrow$ non-smooth optimization
  - Ridge
- PG $\rightarrow$ Proximal Gradient
- APG $\rightarrow$ accelerated

PG, APG $\rightarrow$ non-smooth optimization algo

---

1. Binary Classification $\rightarrow$ $\boxed{Y_i \in \{0,1\}}$

$\updownarrow$

Multi-class classification $\rightarrow$ $Y_i \in [K]$

$\rightarrow$ Linear Regression is not suitable for classification task $\rightarrow$ Many reasons

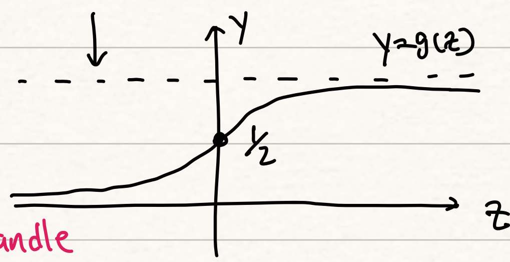
GLM Framework

$\rightarrow$ Logistic Regression works for classification

model: $\boxed{f(x) = \text{sigmoid}(\beta^T x)}$ $\Leftrightarrow$ $\text{logit}(p) = \beta^T x.$

$\boxed{\text{logit}(p) = \log\left(\frac{p}{1-p}\right)}$

sigmoid $g(z) = \dfrac{1}{1+\exp(-z)}$

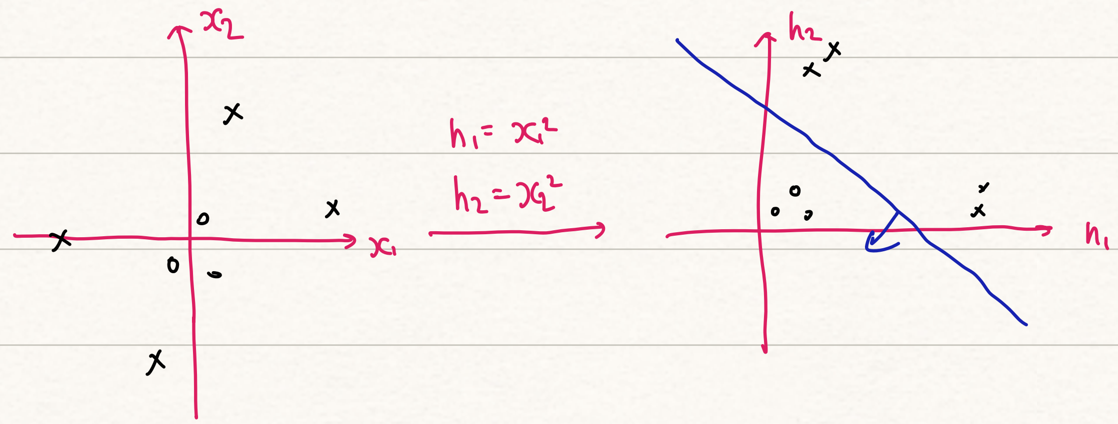



linear decision hyperplane (boundary)

since sigmoid(·) is monotone

can use Feature Map to handle non-linearly separable case

[e.g.]




[non-linearly separable in $(x_1, x_2)$-plane] [linearly separable in $(h_1, h_2)$-plane]

Minimize $\boxed{\text{Negative Log-likelihood}}$ ↳ cost function / Loss function for classification

$\Updownarrow$

2. $\boxed{\text{Loss function}}$ → Maximum Likelihood Estimator

$$\mathrm{Prob}(x_i \in C_1) \overset{\text{model}}{:=} \sigma(\beta^T x_i) := f(x_i)$$

$$\mathrm{Prob}(x_i \in C_0) \overset{\text{model}}{:=} 1 - \sigma(\beta^T x_i) := 1 - f(x_i)$$

$$\Rightarrow \mathrm{Prob}(y_i \mid x_i, \beta) = \left[\sigma(\beta^T x_i)\right]^{y_i} \left[1 - \sigma(\beta^T x_i)\right]^{1-y_i}$$

$$\Rightarrow \log\left[\mathrm{Prob}(y_i \mid x_i, \beta)\right] = y_i \log\left[f(x_i)\right] + (1-y_i)\log\left[1 - f(x_i)\right]$$

$$= -\,\mathrm{Cross\ Entropy}\left(\begin{bmatrix} y_i \\ 1-y_i \end{bmatrix}; \begin{bmatrix} f(x_i) \\ 1-f(x_i) \end{bmatrix}\right)$$

Recap:

$$D_{KL}(p \| q) := \mathbb{E}_{z \sim p}\left[\log\left(\frac{p(z)}{q(z)}\right)\right]$$

$$\propto -\,\mathrm{KL\text{-}Divergence}\left(\underbrace{\begin{bmatrix} y_i \\ 1-y_i \end{bmatrix}}_{\text{distribution 1}}; \underbrace{\begin{bmatrix} f(x_i) \\ 1-f(x_i) \end{bmatrix}}_{\text{distribution 2}}\right)$$

$$\downarrow$$

$$D_{KL}\left(\begin{bmatrix} y_i \\ 1-y_i \end{bmatrix} \,\Big\|\, \begin{bmatrix} f(x_i) \\ 1-f(x_i) \end{bmatrix}\right)$$

MLE Framework

$$\Rightarrow \hat{\beta} = \underset{\beta}{\arg\max} \prod_{i=1}^{n} \mathbb{P}(y_i \mid x_i, \beta)$$

$$= \underset{\beta}{\arg\max} \sum_{i=1}^{n} \log\left(\mathbb{P}(y_i \mid x_i, \beta)\right)$$

$$= \underset{\beta}{\arg\max} \sum_{i=1}^{n} \left[ y_i \log(f(x_i;\beta)) + (1-y_i) \log(1 - f(x_i;\beta)) \right]$$

$$= \underset{\beta}{\arg\min} - \sum_{i=1}^{n} \left[ y_i \log(f(x_i;\beta)) + (1-y_i) \log(1 - f(x_i;\beta)) \right]$$

$$:= \underset{\beta}{\arg\min} \begin{cases} \sum_{i=1}^{n} \text{Cross-Entropy}\left( \begin{bmatrix} y_i \\ 1-y_i \end{bmatrix}, \begin{bmatrix} f(x_i) \\ 1-f(x_i) \end{bmatrix} \right) \\ \sum_{i=1}^{n} D_{KL}\left( \begin{bmatrix} y_i \\ 1-y_i \end{bmatrix} \middle\| \begin{bmatrix} f(x_i) \\ 1-f(x_i) \end{bmatrix} \right) \end{cases}$$

## 3. Property of Logistic Regression Loss Function

$$L(\beta) = - \sum_{i=1}^{n} \left[ y_i \log(f(x_i)) + (1-y_i) \log(1 - f(x_i)) \right]$$

$$= - \sum_{i=1}^{n} \left[ y_i \log\left( \frac{f(x_i)}{1-f(x_i)} \right) + \log(1 - f(x_i)) \right]$$

$$= - \sum_{i=1}^{n} \left[ y_i \cdot \beta^T x_i - \log(1 + \exp(\beta^T x_i)) \right]$$

GLM Result

① Gradient Calculation

$$\nabla_\beta L(\beta) = - \sum_{i=1}^{n} \left[ y_i x_i - \frac{1}{1+\exp(\beta^T x_i)} \cdot \exp(\beta^T x_i) \cdot x_i \right]$$

Good Property From GLM

$$= \sum_{i=1}^{n} \left[ \frac{1}{1+\exp(-\beta^T x_i)} - y_i \right] \cdot x_i$$

$$= \sum_{i=1}^{n} ( f(x_i;\beta) - y_i ) \, x_i$$

$\boxed{\text{GLM}}$

- identity link function, Gaussian output → $\boxed{\text{Linear Regression}}$
- sigmoid (logit) link function, Bernolli output → $\boxed{\text{Logistic Regression}}$

4. Issue for Loss function

→ $\boxed{\text{minimum}}$ may not exist (although convex)

↓ receipe

$$\boxed{L(\beta) = \sum_{i=1}^{n} \left[ \log(1+\exp(-\beta^T x_i)) - y_i \beta^T x_i \right]}$$

$\boxed{\text{Regularization}}$ (also can help $\boxed{\text{over-fitting}}$)

Gradient Method works

a) Ridge Reg → $\ell_2$ penalty

b) LASSO
- Regularization → $\ell_1$ penalty
- Feature Selection

Gradient Method does not work

## 5. Optimization Algo for LASSO [Summary from my side]

Recap: for $\boxed{LR}$, LASSO Regularization is:

$$\hat{w}_{LASSO} = \underset{w}{\operatorname{argmin}} \underbrace{\|Xw - y\|_2^2}_{\substack{(convex)\\ \text{differentiable}}} + \underbrace{\lambda \|w\|_1}_{\text{non-differentiable}}$$

for $\boxed{\text{Logistic Regression}}$, LASSO Regularization is:

$$\hat{w}_{LASSO} = \underset{w}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{n}\left[\log(1+\exp(w^T x_i)) - y_i w^T x_i\right]}_{\text{differentiable (convex)}} + \underbrace{\lambda \|w\|_1}_{\text{non-differentiable}}$$

→ since objective = $\underbrace{\text{non-differentiable}}_{\text{penalty term}}$ + $\underbrace{\text{differentiable}}_{\text{original obj}}$

then Gradient-based method cannot work!

→ Introduce Non-smooth Optimization Algorithm

↓

$\boxed{\text{Proximal Gradient Method}}$ (PG)

## 6. Proximal Gradient Method (MA5243)

① Morean-Yosida Regularization

$$\phi_t(y) = f(y) + \boxed{\frac{1}{2t}\|y - x\|_2^2} \rightarrow \boxed{\text{quadratic convex term}}$$

$$\psi_{f,t}(x) := \min_{y} \phi_t(y)$$
$\begin{cases} \psi_f(\cdot) \Rightarrow \text{convex, differentiable} \\ \text{minimizer exists \& unique} \end{cases}$

② Proximal Mapping (minimizer)

$$P_{tf}(x) = \arg\min_{y} \left\{ t f(y) + \frac{1}{2} \|y - x\|_2^2 \right\}$$

$$= \arg\min_{y} \phi_t(y)$$

③
$$\begin{cases} \min_{x} f(x) = \min_{x} \varphi_{f,t}(x) \\ \arg\min_{x} f(x) = \arg\min_{x} \varphi_{f,t}(x) \end{cases}$$

④ How to calculate $\nabla_x \varphi_{f,t}(x)$ ?

Idea: $\boxed{\hat{x} = \arg\min_{x} f(x)}$ → convex but non-smooth

transform

$\boxed{\hat{x} = \arg\min_{x} \varphi_{f,t}(x)}$ → smooth

then we should know $\boxed{\nabla_x \varphi_{f,t}(x)}$

$$\rightarrow \nabla_x \varphi_{f,t}(x) = t^{-1}\left(x - P_{tf}(x)\right)$$

⑤ How to calculate $P_{tf}(x)$ ?

M-Y Decomposition $\quad x = P_f(x) + P_{f^*}(x)$

$f$ positive homogeneous $\Leftrightarrow f^*(x) = \delta_{\{x \in \partial f(0)\}}$

$f = \delta_C \Rightarrow P_f(x) = \Pi_C(x)$

$$:= \arg\min_{y \in C} \|y - x\|_2^2$$

$\Rightarrow$ can determine $f = \|\cdot\|_1$, $P_f(x) = ?$

a) $\partial f(0)$

$$\boxed{f^* = \|\cdot\|_\infty}$$

$x \in \partial f(0)$

$\Leftrightarrow \quad f(y) - f(0) \geq x^T y \quad \forall y$

$\Leftrightarrow \quad \|y\|_\# \geq \langle x, y \rangle \quad \forall y$

$\Leftrightarrow \quad \langle x, y \rangle \leq 1 \quad \forall \|y\|_\# \leq 1$

$\Leftrightarrow \quad \|x\|_* \leq 1$

b) $P_f(x) = x - P_{f^*}(x)$

$= x - \Pi_{B_*^2}(x)$

$B_*^1 := \{ x \in \mathbb{R}^n : \|x\|_* \leq 1 \}$

⑥ Proximal Gradient Algo Framework

$\rightarrow \min\limits_x f(x)$

$f$: non-smooth

$\rightarrow \min\limits_x \varphi_{t,f}(x)$

$\varphi_{f,t}$ : M-Y Regularization

$\rightarrow \quad x^{(k+1)} = x^{(k)} - t_k \nabla_x \varphi_{t_k,f}(x^{(k)})$

$\rightarrow \quad x^{(k+1)} = P_{t_k f}(x^{(k)})$

$\rightarrow \quad x^{(k+1)} \approx \operatorname*{argmin}\limits_y f(y) + \frac{1}{2t_k} \| y - x^{(k)} \|_2^2$

we hope we can ☆ approximately solve this part through ☆ closed-form solution

$$\boxed{\text{Real Application}}$$

$$\longrightarrow \min \; f(x) + g(x)$$

$$\begin{cases} f(\cdot) \longrightarrow \text{smooth} \\ g(\cdot) \longrightarrow \text{non-smooth but convex} \end{cases}$$

$$\longrightarrow x^{(k+1)} = \underset{y}{\operatorname{argmin}} \; \underbrace{f(y)}_{\text{bad part}} + \underbrace{g(y) + \frac{1}{2t_k} \| y - x^{(k)} \|_2^2}_{\text{good part}}$$

$\Big\downarrow$ linearize $f(\cdot)$

$$\approx \underset{y}{\operatorname{argmin}} \; f(x^{(k)}) + \nabla_x f(x^{(k)})^T (y - x^{(k)}) + g(y) + \frac{1}{2t_k} \| y - x^{(k)} \|_2^2$$

$$= \underset{y}{\operatorname{argmin}} \; \nabla_x f(x^{(k)})^T y + g(y) + \frac{1}{2t_k} \| y - x^{(k)} \|_2^2$$

$$= \underset{y}{\operatorname{argmin}} \; g(y) + \frac{1}{2t_k} \| y + t_k \nabla f(x^{(k)}) - x^{(k)} \|_2^2$$

$$= P_{t_k g(\cdot)} \left( t_k \nabla_x f(x^{(k)}) - x^{(k)} \right)$$

Note: here $P_{t_k g(\cdot)}(x)$ will have closed-form solution !

if $\boxed{g(\cdot) = \| \cdot \|_1}$ $\longrightarrow$ LASSO Regularization

then $P_{t_k g}(x) = x - P_{(t_k g)^*}(x)$

$$= x - \Pi_{B^*_{t_k}}(x)$$

$$B^*_{t_k} := \{ x \in \mathbb{R}^n : \| x \|_\infty \leq t_k \}$$

PS:

$$P_{t_k \|\cdot\|_1}(y^{(k)})$$

$$= \underset{u}{\operatorname{argmin}} \left\{ t_k \|u\|_1 + \frac{1}{2} \| u - y^{(k)} \|_2^2 \right\}$$

$$= \underset{u}{\operatorname{argmin}} \left\{ \left\| \frac{u}{t_k} \right\|_1 + \frac{1}{2} \left\| \frac{u}{t_k} - \frac{y^{(k)}}{t_k} \right\|_2^2 \right\}$$

$$\overset{t_k}{=} \underset{v}{\operatorname{argmin}} \left\{ \|v\|_1 + \frac{1}{2} \left\| v - \frac{y^{(k)}}{t_k} \right\|_2^2 \right\}$$

$$= P_{\|\cdot\|_1} \left( \frac{y^{(k)}}{t_k} \right) \cdot t_k$$

Another calculation:

$$P_{t_k \|\cdot\|_2}(x) = \underset{y}{\operatorname{argmin}} \; t_k \|y\|_1 + \frac{1}{2}\|y - x\|_2^2$$

$$= \underset{y}{\operatorname{argmin}} \; \frac{1}{t_k}\|y\|_2 + \frac{1}{2t_k^2}\|y - x\|_2^2$$

$$= \underset{y}{\operatorname{argmin}} \; \left\|\frac{y}{t_k}\right\|_1 + \frac{1}{2}\left\|\frac{y}{t_k} - \frac{x}{t_k}\right\|_2^2$$

$$\boxed{u = \frac{y}{t_k}} \qquad = t_k \underset{u}{\operatorname{argmin}} \; \|u\|_1 + \frac{1}{2}\left\|u - \frac{x}{t_k}\right\|_2^2$$

$$= t_k \cdot P_{\|\cdot\|_1}\left(\frac{x}{t_k}\right)$$