Linear classification Model:

Apply discrete function on linear regression model s.t. output is discrete. find a linear bound in space that best split the sample

$$g(f(x;w)) = \begin{cases} 1 & f(x;w) > 0 \\ 0 & f(x;w) < 0 \end{cases}$$

f: discriminant fn

g: activation/decision fn

$$f(x;w) = w^T x + b \qquad x, w \in \mathbb{R}^{p\times 1};\quad b \in \mathbb{R}$$

individual observation

learning criterion: loss function minimized

- Binary classification $y \in \{0,1\}$
  - one thought on loss function: 0-1 loss

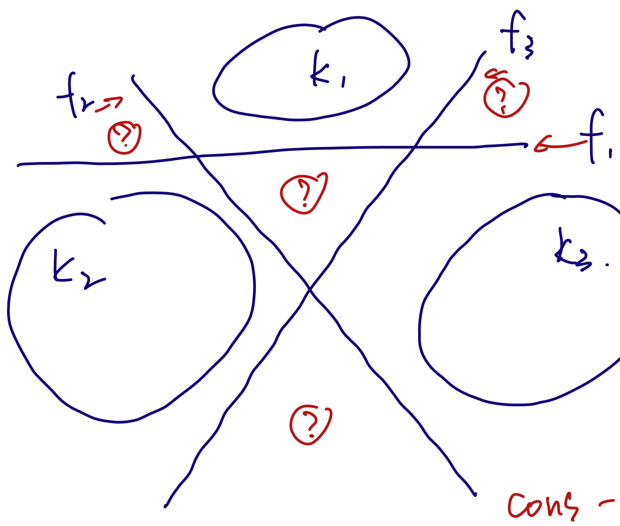
$$\mathcal{L}_{01} = \mathbb{1}(y \neq \hat{y}) = \mathbb{1}(y \neq g(f(x;w)))$$

prob is $\mathcal{L}_{01}$ is not diff. so cannot develope optimization prob on it; need smoother loss fn as criterion.

- Muti classification $y \in \{1, \dots c\}$

potential model:

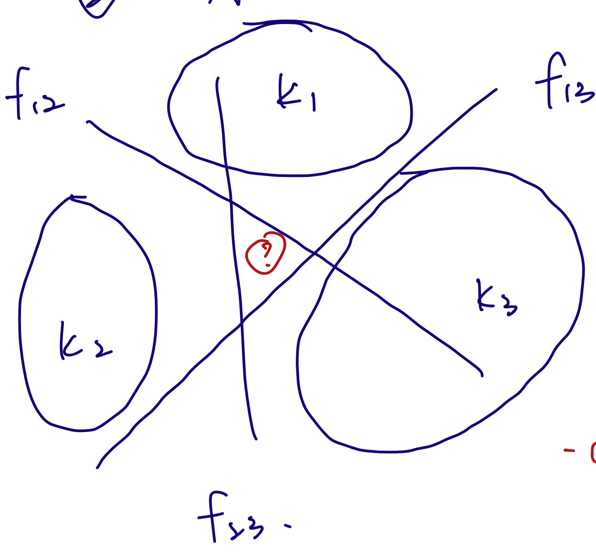
① 一对其余



- 3 discriminant f's
- each f_k decide whether $x^{(i)} \in C_k$ or not.
- then use majority voting to decide final output

cons - ?: indistinguishable area

② 一对一



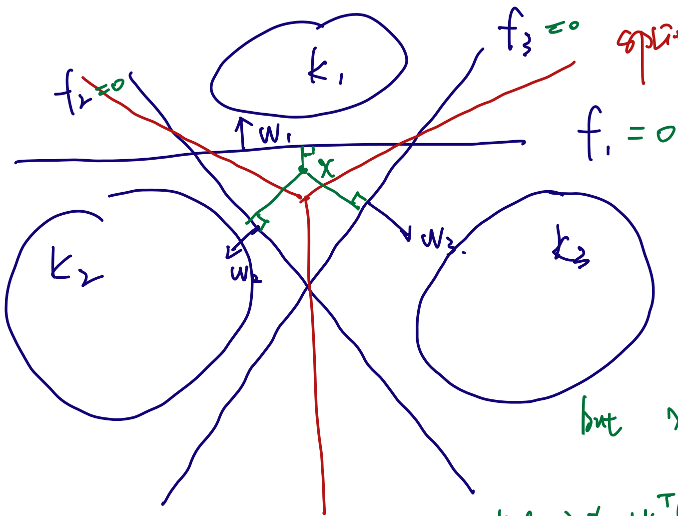
- 每两个 class 间建立一个分类器.
- total C_K^2 个 classifier.
- use majority voting

- cons: ① ?

② computationally costly.

③ argmax.

为对"一对其余"的一种改进.



splitted into C, no ?

$w_i^T x + b < 0 \quad \forall i$

(x and w_i forms obtuse $\forall i$)

but x is closest to f_1 boundary,

meaning $w_1^T x$ is the largest among $i \in [C]$

- need c f's
- decision fn: $\arg\max_c f_c(x;w) = 1$ for x in fig.

Cross-entropy: measures how similar the predicted dist is to the real dist; equiv to MLE; the smaller the $H(p,q)$ the closer

$$L(\theta) = \prod_{x \in X} P_\theta(X=x)^{n \cdot \Pr(X=x)}$$

$$\frac{1}{n} \cdot \log(L(\theta)) = \sum_{x \in X} \Pr(X=x) \cdot \log(P_\theta(X=x))$$

$$= -H(\Pr, P_\theta).$$

Intuition: if $\Pr$, P_θ really close, then larger $\Pr$ will have greater power than smaller, s.t. the $\log(L)$ is larger

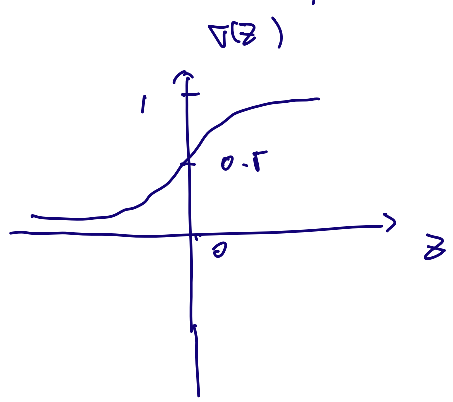
Logistic Regression.

As said 0-1 loss not diff. cannot do gradient descent.

So let g outputs the conditional prob. of the obs. being in class 1.

i.e. $\underbrace{P_\theta(y=1|x)}_{\text{模型预测} = \hat{y}} = \underbrace{g(f(x;w))}_{\in[0,1]}$

Sigmoid, logistic fn. $\quad \sigma(z) = \frac{1}{1+\exp(-z)}$



true probability for (x, y^*)

$Pr(y=1|x) = y^*$

$Pr(y=0|x) = 1-y^*$

Intuition: want to train w and b

s.t. $\hat{y} = P_\theta(y=1|x) = \sigma(w^T x + b) \sim y^* = Pr(y=1|x)$

use cross-entropy as loss function to measure how similar two prob. distributions are.

$$H(P_r, P_\theta) = -[y^* \log(\hat{y}) + (1-y^*)\log(1-\hat{y})]$$

$$\Rightarrow R(w;b) = -\frac{1}{n}\sum_{i=1}^{n}[y^{(i)}\log(\hat{y}^{(i)}) + (1-y^{(i)})\log(1-\hat{y}^{(i)})]$$

$$\operatorname{argmin} R(w;b) + \lambda\|w\|^2 \overset{\text{equiv}}{\Longleftrightarrow} \operatorname{argmax} \text{ log likelihood}$$

i.e. $\operatorname{argmax} \sum_{i=1}^{n} P(\hat{y} = y^{(i)} \mid x = x^{(i)})$

intuition: $y^* = 0$; want $1-\hat{y}$ large $\rightarrow$ $\hat{y}$ small

$y^* = 1$; want $\hat{y}$ large

Gradient:

$$\frac{\partial R}{\partial w} = -\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)}\frac{1}{\hat{y}^{(i)}}\frac{\partial \hat{y}^{(i)}}{\partial w} - (1-y^{(i)})\frac{1}{1-\hat{y}^{(i)}}\frac{\partial \hat{y}^{(i)}}{\partial w}\right)$$

$$\frac{\partial \hat{y}^{(i)}}{\partial w} = \frac{\partial}{\partial w}\sigma(w^T x^{(i)} + b) = \frac{1}{(1+e^{-z})^2}\cdot e^{-z}\cdot x$$

$$= \frac{1}{1+e^{-z}}\cdot\frac{e^{-z}}{1+e^{-z}}\cdot x$$

$$= \hat{y}^{(i)}\cdot(1-\hat{y}^{(i)})\cdot x^{(i)}$$

$$\frac{\partial R}{\partial w} = -\frac{1}{n}\sum_{i=1}^{n}\frac{y^{(i)}(1-\hat{y}^{(i)}) - \hat{y}^{(i)}(1-y^{(i)})}{\hat{y}^{(i)}(1-\hat{y}^{(i)})}\cdot\hat{y}^{(i)}\cdot(1-\hat{y}^{(i)})\cdot x^{(i)}$$

$$= -\frac{1}{n}\sum_{i=1}^{n}(\underbrace{y^{(i)} - \hat{y}^{(i)}}_{\in\mathbb{R}})\cdot\underbrace{x^{(i)}}_{P\times 1}$$

迭代 $W_{t+1} \leftarrow W_t + \alpha \cdot \frac{1}{n}\sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)}) \cdot x^{(i)}$

Softmax 模型.

$$f_c(x; W_c) = W_c^T x + b_c \in \mathbb{R}, \quad c \in \{1 \dots C\}.$$

Softmax 函数:

$$\text{Softmax}(x_k) = \frac{\exp\{x_k\}}{\underbrace{\sum_{i=1}^{C} \exp\{x_i\}}_{\text{all classes involved in}}}.$$

In this classification model:

$$\hat{y} = P_\theta(y=k|x) = \text{softmax}(W_k^T x) = \frac{\exp\{W_k^T x\}}{\sum_{i=1}^{C} \exp\{W_k^T x\}}$$

向量表示: $\hat{y} = \frac{\exp(W^T x)}{\mathbb{1}_C^T \exp(W^T x)}$

C个 classifier 的 coeff stacked together

$$W = \underbrace{\begin{bmatrix} | & | & & | \\ W_1 & W_2 & \cdots & W_C \\ | & | & & | \end{bmatrix}}_{P \times C}$$

eg. $\begin{bmatrix} 0.3 \\ 0.2 \\ 0.5 \end{bmatrix} \begin{array}{l} \leftarrow P_\theta(y=1|x) \\ \\ \leftarrow P_\theta(y=3|x) \end{array}$

Loss function (for one 样本) is same as logistic:

$$-\sum_{y=1}^{C} P_r(y|x) \log(P_\theta(y|x))$$

$$P_r(y|x) = \begin{cases} 1 & \text{if } y = y^* \\ 0 & \text{otherwise.} \end{cases}$$

intuition: find W that maximize.

$P_\theta(y=y^*|x)$

Criterion, cost fn: cross-entropy

$$R(W;b) = -\frac{1}{n}\sum_{i=1}^{n} (y^{(n)})^T \log(\hat{y}^{(n)})$$

$y^{(n)}$ is a one hot vector

$$y^{(n)} = \begin{bmatrix} \mathbb{1}_{y^*=1} \\ \vdots \\ \mathbb{1}_{y^*=C} \end{bmatrix}$$

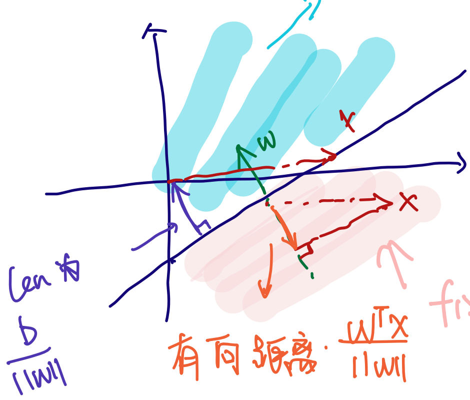
Gradient:

$$\underbrace{\frac{\partial R(W;b)}{\partial W}}_{p\times c} = -\frac{1}{n}\sum_{i=1}^{n} \overbrace{\underbrace{x^{(n)}}_{p\times 1} \underbrace{(y^{(n)} - \hat{y}^{(n)})}_{c\times 1}{}^T}^{p\times c}$$

$W : p\times c$

$f(x;w,b) > 0$ i.e. w和x 间角度更小 $w^T x$ 更大.

$f(x;w,b) = w^T x + b.$



$$\frac{b + w^T x}{||w||} = 0$$

$f(x;w,b) < 0$