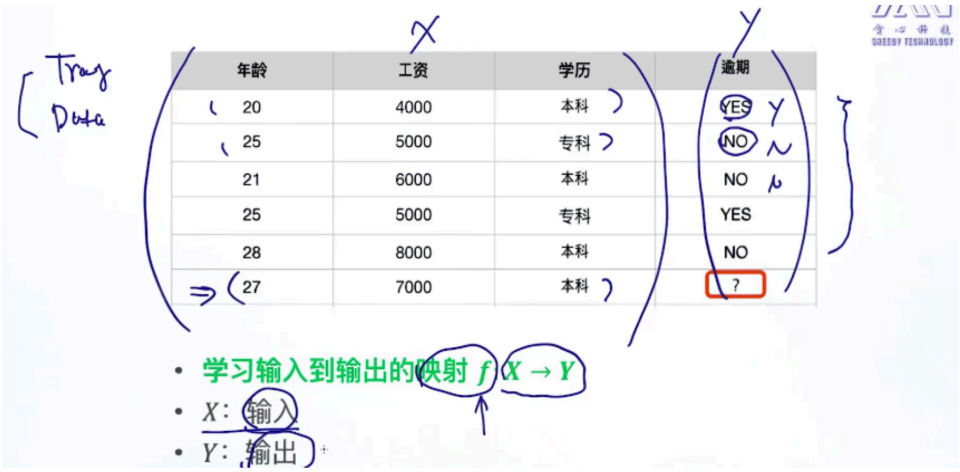
1. Logistic Regression

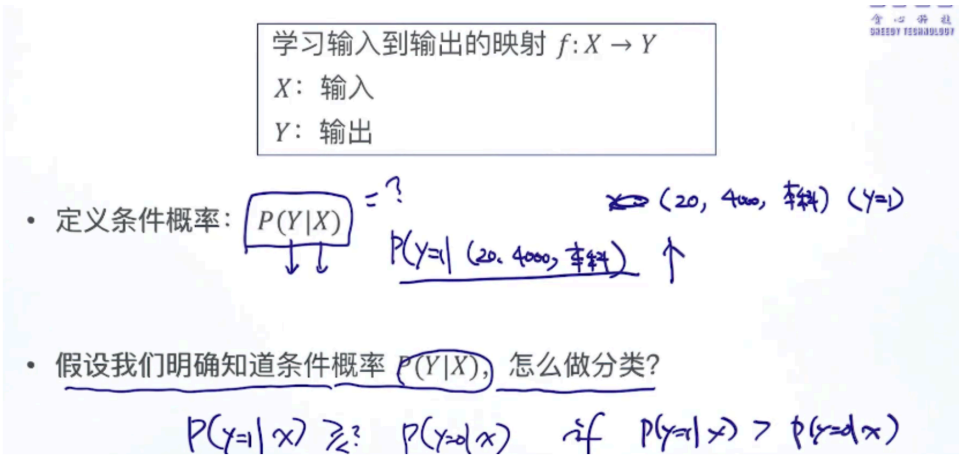
1.1 Classification tasks:

贷款违约、广告点击、商品推荐、情感分析、疾病诊断，etc

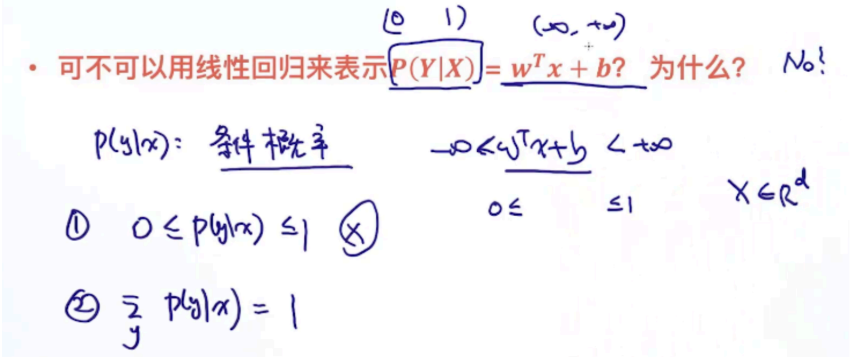
逻辑回归是一个很好的 baseline



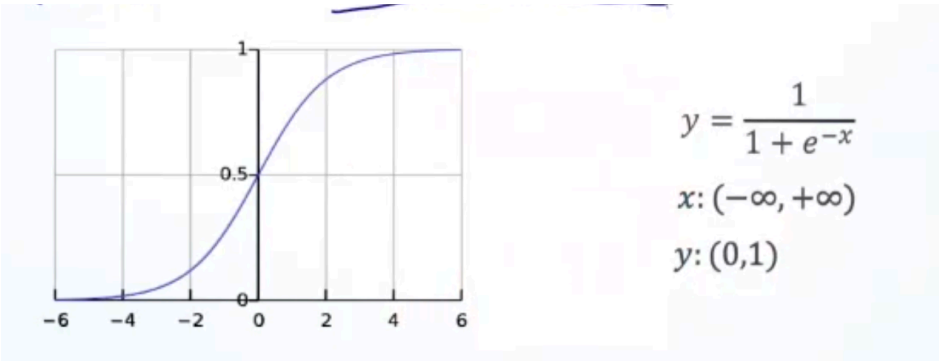
1.2 problem

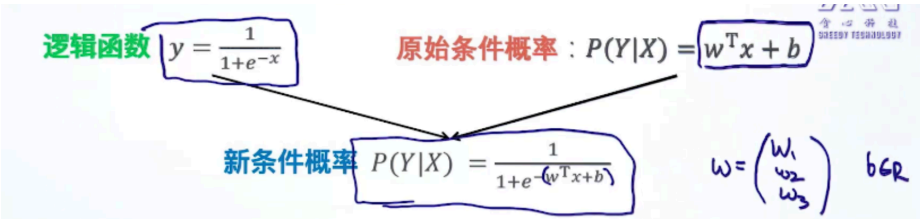


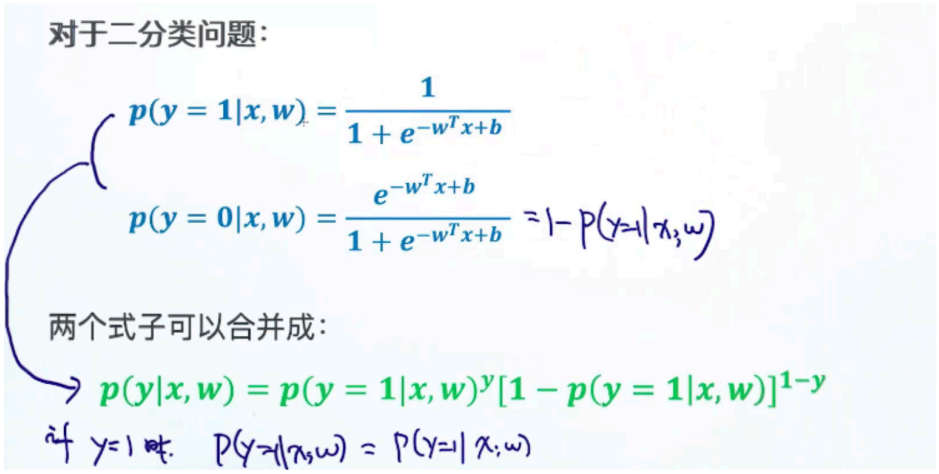
核心：怎么去设计条件概率



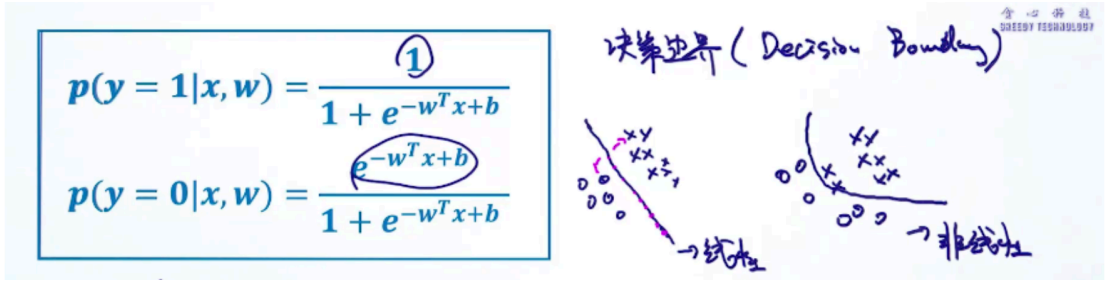
1.3 logistic function







1.4 linear classifier



1.5 objective function

假设我们拥有数据集 $D = \{(x_i, y_i)\}_{i=1}^{n} \quad x_i \in R^d, \quad y_i \in \{0, 1\}$

而且我们已经定义了：

$$p(y|x,w) = p(y=1|x,w)^y [1 - p(y=1|x,w)]^{1-y}$$

我们需要最大化目标函数：

$$\hat{w}_{MLE}, \hat{b}_{MLE} = argmax_w \prod_{i=1}^{n} p(y_i|x_i,w,b)$$

$$\hat{w}_{MLE}, \hat{b}_{MLE} = argmax_{w,b} \prod_{i=1}^{n} p(y_i|x_i,w,b) \quad ①$$

$$= \arg\max_{w,b} \log\left(\prod_{i=1}^{n} p(y_i|x_i,w,b)\right)$$

$$= \arg\max_{w,b} \sum_{i=1}^{n} \log p(y_i|x_i,w,b)$$

$\log(xyz) = \log x + \log y + \log z$

Objective Function

$$\arg\max_{w,b} \sum_{i=1}^{n} \log p(y_i|x_i,w,b)$$

$$argmin_{(w,b)} - \sum_{i=1}^{n} \log p(y_i|x_i,w,b)$$

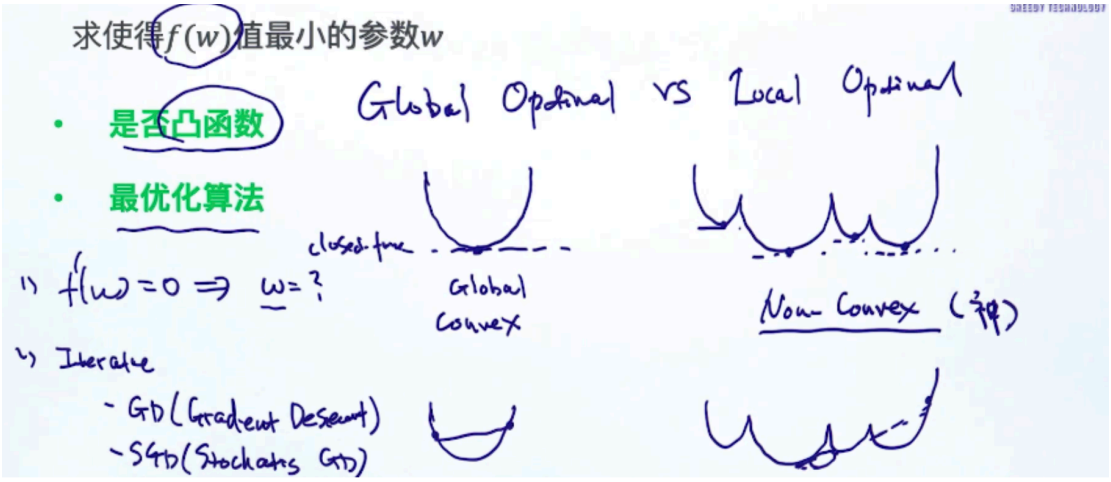
$$= \arg\min_{w,b} - \sum_{i=1}^{n} \log\left[p(y=1|x,w,b)^y \cdot [1 - p(y=1|x,w,b)]^{1-y}\right]$$

$$= \arg\min_{w,b} - \sum_{i=1}^{n} y \cdot \log p(y=1|x,w,b) + (1-y) \log[1 - p(y=1|x,w,b)]$$

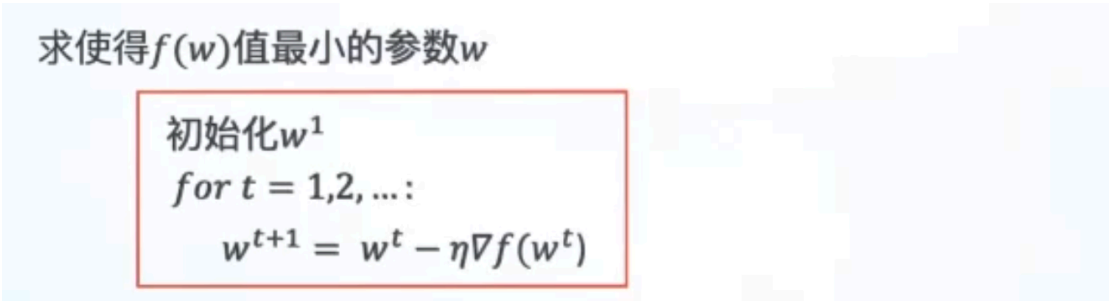
$\log(a^y b^x) = \log(a^y) + \log(b^x) = y \log a + x \log b$

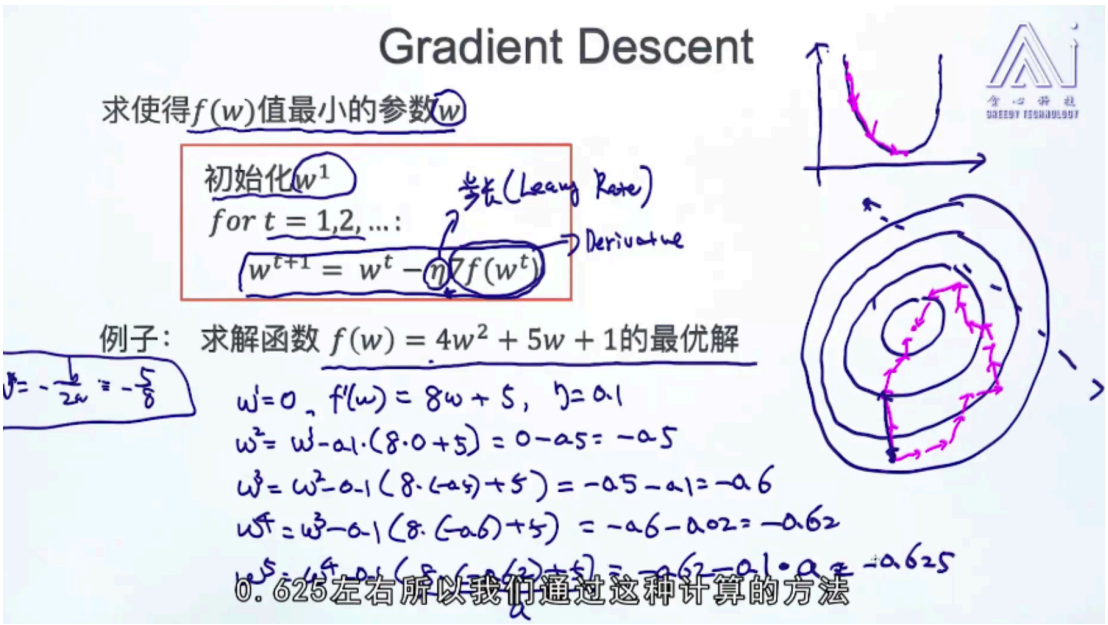
2. 梯度下降

2.1 求解函数的最小值



2.2 梯度下降





2.3 逻辑回归的梯度下降

$$p(y=1|x,w)=\frac{1}{1+e^{-w^Tx+b}}$$

$$argmin_{w,b}-\sum_{i=1}^{n} y_i \log p(y_i=1|x_i;w)+(1-y_i)\log(1-p(y_i=1|x_i;w))$$

$$= \underset{w,b}{\operatorname{argmin}} \underbrace{-\sum_{i=1}^{n} y_i \cdot \log \sigma(w^T x_i+b)+(1-y_i)\cdot \log[1-\sigma(w^T x_i+b)]}_{L(w,b)}$$

$$\frac{\partial L(w,b)}{\partial w} = -\sum_{i=1}^{n} y_i \cdot \frac{\sigma(w^T x_i+b)\cdot[1-\sigma(w^T x_i+b)]}{\sigma(w^T x_i+b)}\cdot x_i + (y_i-1)\cdot \frac{\sigma(w^T x_i+b)\cdot[1-\sigma(w^T x_i+b)]}{1-\sigma(w^T x_i+b)}\cdot x_i$$

$$= -\sum_{i=1}^{n} y_i(1-\sigma(w^T x_i+b))\cdot x_i + (y_i-1)\cdot \sigma(w^T x_i+b)\cdot x_i$$

$$= \sum_{i=1}^{n}[y_i - \sigma(w^T x_i+b)]\cdot x_i = \sum_{i=1}^{n}[\sigma(w^T x_i+b)-y_i]\cdot x_i$$

$$L(w,b) = -\sum_{i=1}^{n} y_i \log \sigma(w^T x_i+b)+(1-y_i)\log[1-\sigma(w^T x_i+b)]$$

$$\frac{\partial L(w,b)}{\partial b} = -\sum_{i=1}^{n} y_i \cdot \frac{\sigma(w^T x_i+b)[1-\sigma(w^T x_i+b)]}{\sigma(w^T x_i+b)} + (1-y_i)\cdot \frac{-\sigma(w^T x_i+b)[1-\sigma(w^T x_i+b)]}{1-\sigma(w^T x_i+b)}$$

$$= -\sum_{i=1}^{n} y_i\cdot(1-\sigma(w^T x_i+b)) + (1-y_i)\cdot(-\sigma(w^T x_i+b))$$

$$= \sum_{i=1}^{n}[\sigma(w^T x_i+b)-y_i]$$

初始化 w, b

for $t=1,2\cdots$

learning rate

$$w^{t+1} = w^t - \eta \cdot \sum_{i=1}^{n}[\sigma(w^T x_i+b)-y_i]\cdot x_i$$

$$b^{t+1} = b^t - \eta \cdot \sum_{i=1}^{n}[\sigma(w^T x_i+b)-y_i]$$

Gradient Descent

但是它的风险很大

Stochastic Gradient Descent for Logistic Regression

for $t=1,2\cdots$

$$w^{t+1} = w^t - \eta_t \sum_{i=1}^{n}[\sigma(w^T x_i+b)-y_i]\cdot x_i \quad O(n)$$

$$b^{t+1} = b^t - \eta_t \sum_{i=1}^{n}[\sigma(w^T x_i+b)-y_i] \quad O(n)$$

n: # of samples

① Gradient Descent

for $t=1,2\cdots$

mini-batch = sample($\{(x_i,y_i)\}_{i=1}^{n}$, m)

$$w^{t+1} = w^t - \eta_t \cdot \sum_{(x_i,y_i)\in \text{mini-batch}}[\sigma(w^T x_i+b)-y_i]\cdot x_i$$

$$b^{t+1} = b^t - \eta_t \sum_{(x_i,y_i)\in \text{mini-batch}}[\sigma(w^T x_i+b)-y_i]$$

① + ②

Mini-batch Gradient Descent

for $t=1,2\cdots T$

shuffle($\{(x_i,y_i)\}_{i=1}^{n}$)

for $i=1\cdots n$

$$w^{new} = w^{old} - \eta_t[\sigma(w^T x_i+b)-y_i]\cdot x_i$$

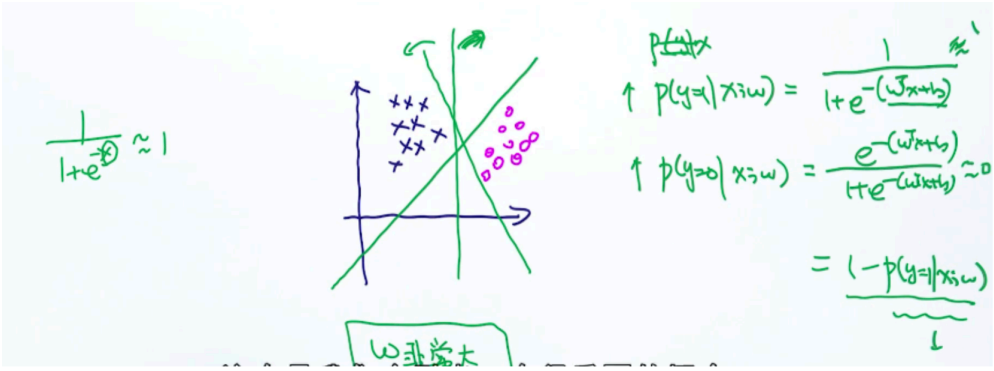
$$b^{new} = b^{old} - \eta_t[\sigma(w^T x_i+b)-y_i]$$

② Stochastic Gradient Descent

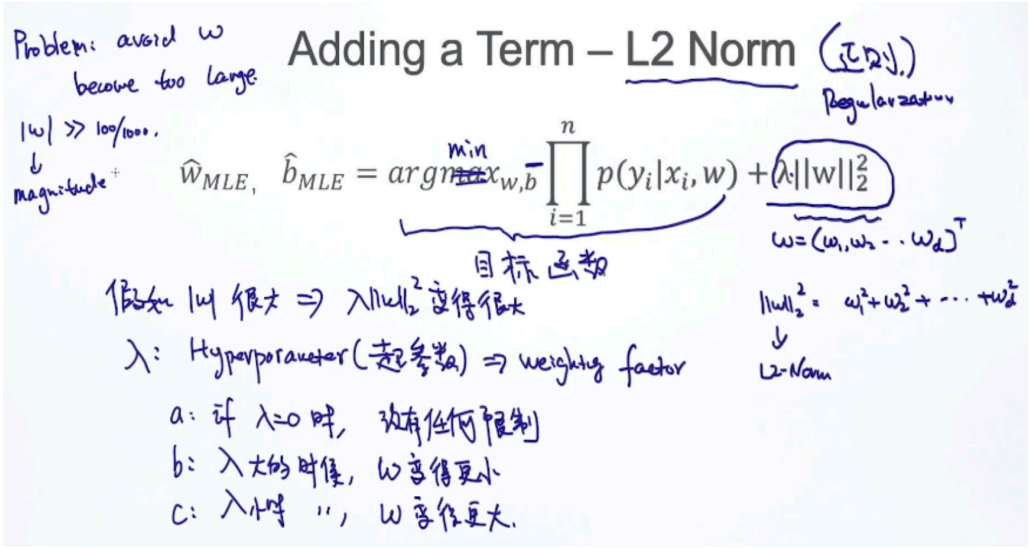
3. 正则化

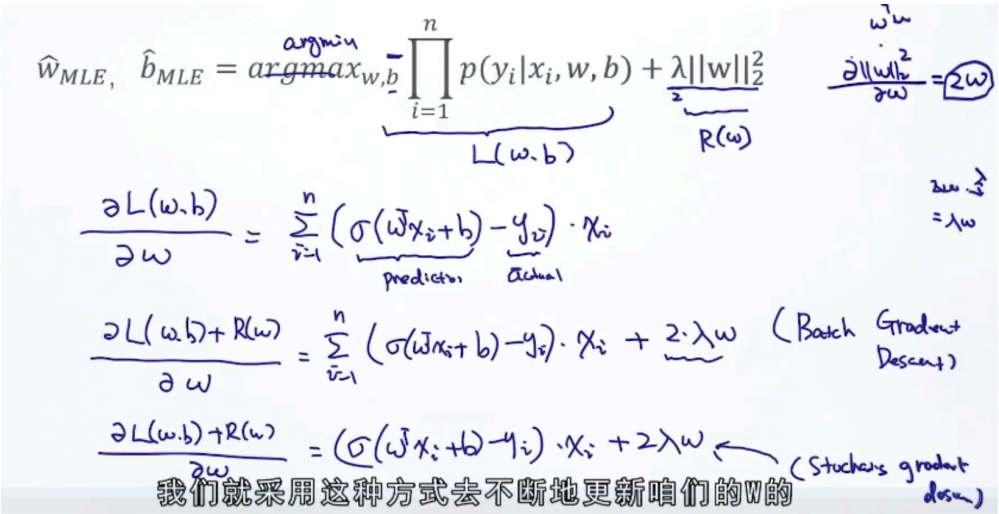
3.1 问题

如果数据线性可分，那么使用逻辑回归的时候，w 会趋于无穷大。（过拟合）



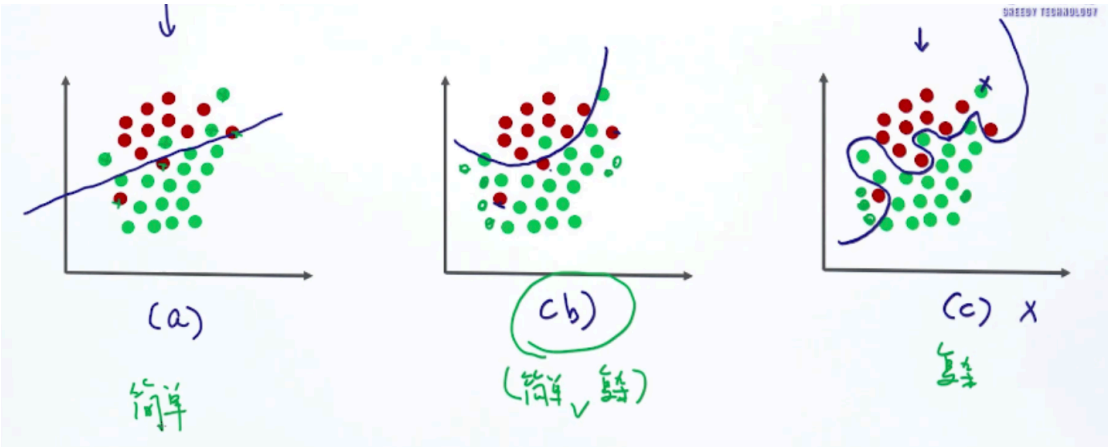
3.2 L2 正则化

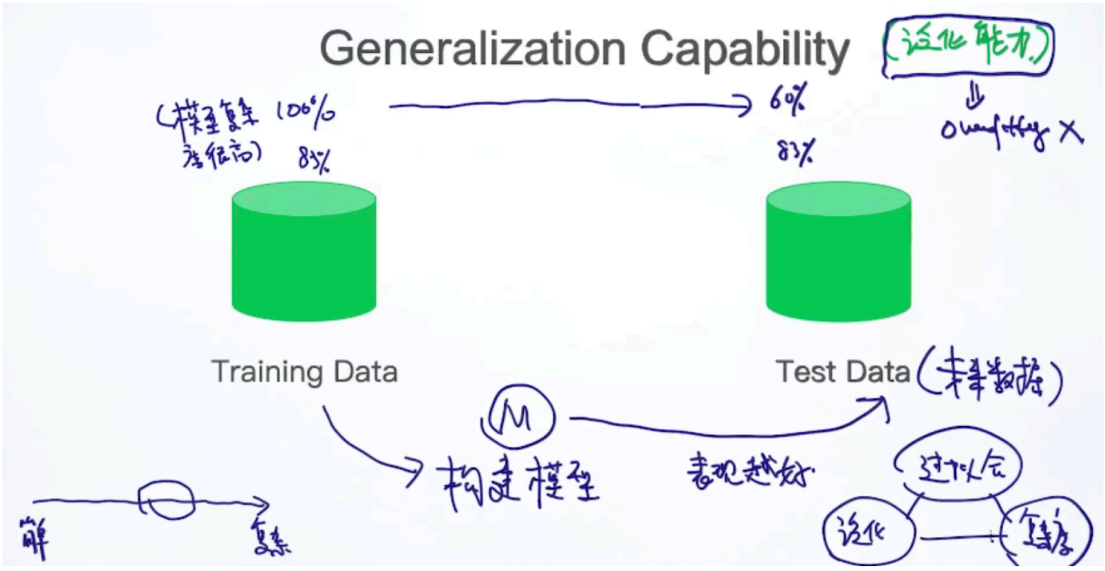


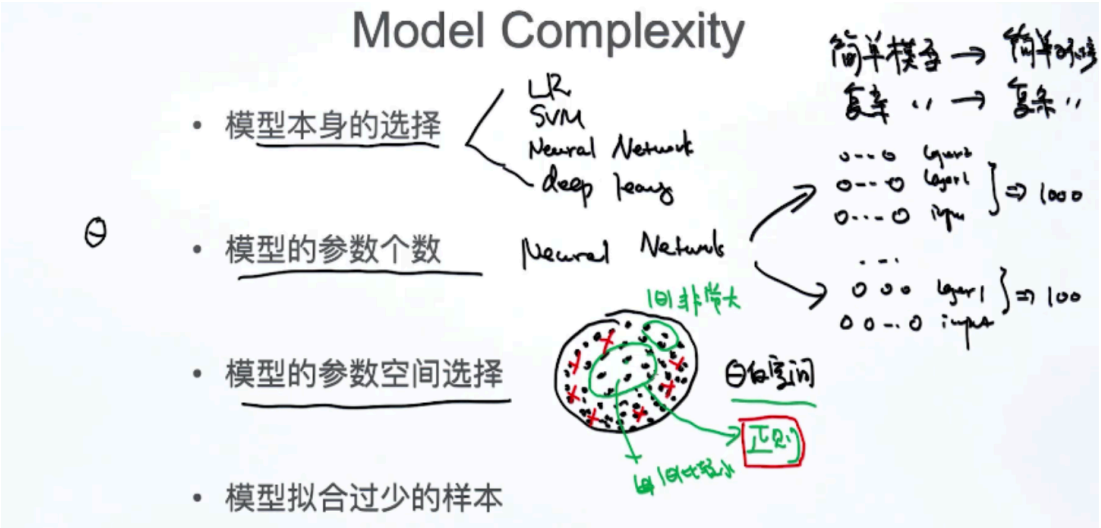


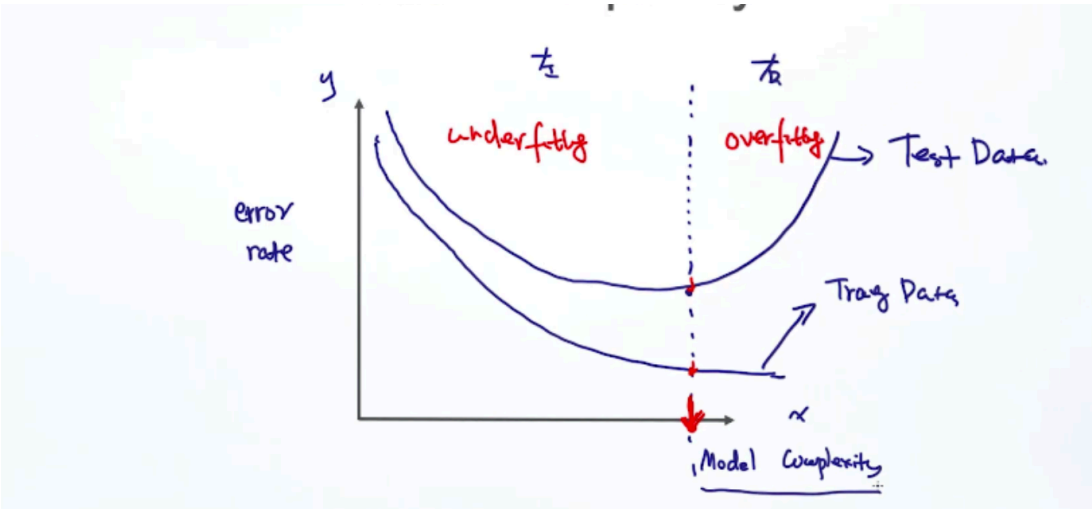
4. 过拟合

4.1 model complexity





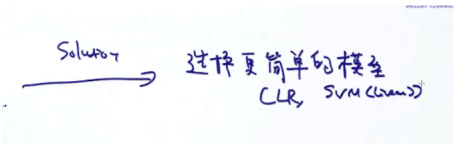




4.2 avoid overfitting

模型复杂度源于

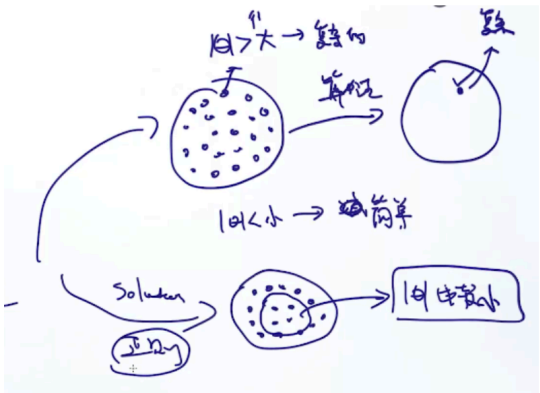
- **模型本身的选择**
- 模型的参数个数
- 模型的参数空间选择
- 模型拟合过少的样本



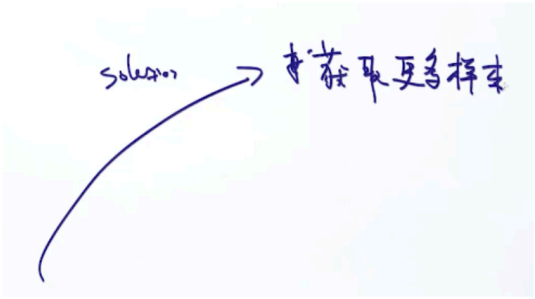
- 模型本身的选择
- **模型的参数个数**
- 模型的参数空间选择
- 模型拟合过少的样本

模型复杂度源于

- 模型本身的选择
- 模型的参数个数
- **模型的参数空间选择**
- 模型拟合过少的样本

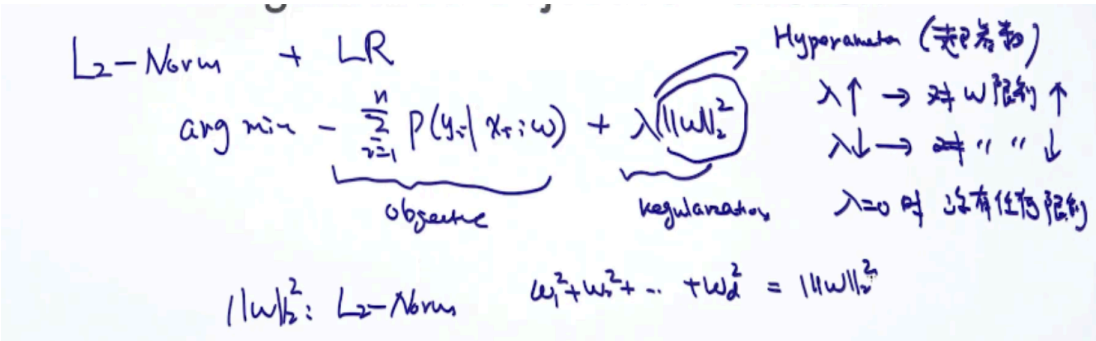


- 模型本身的选择
- 模型的参数个数
- 模型的参数空间选择
- **模型拟合过少的样本**

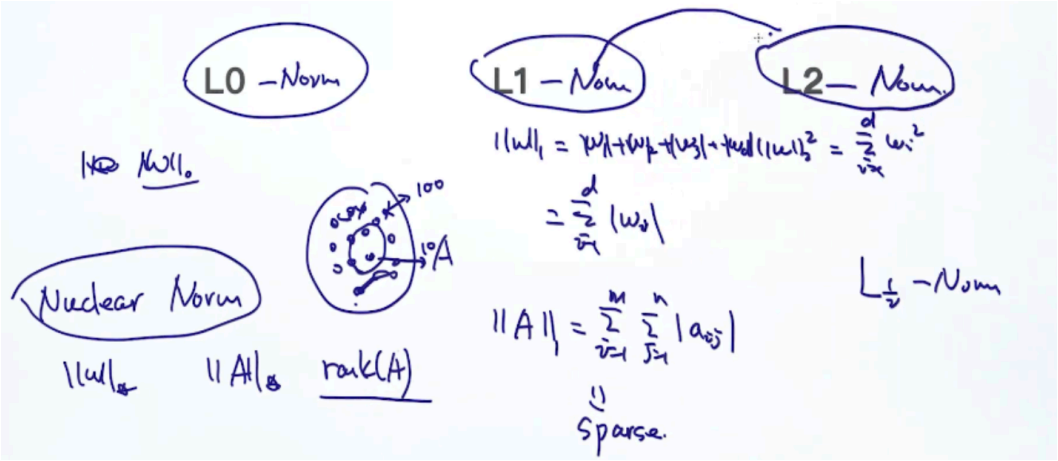


5. 正则化

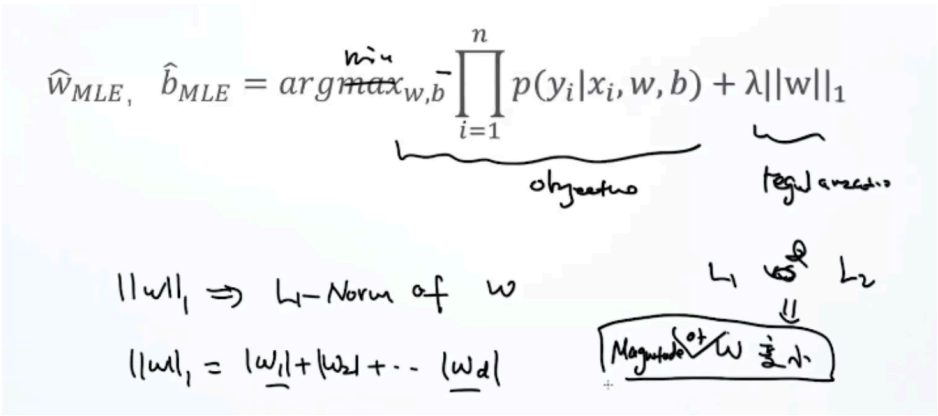
5.1 recap



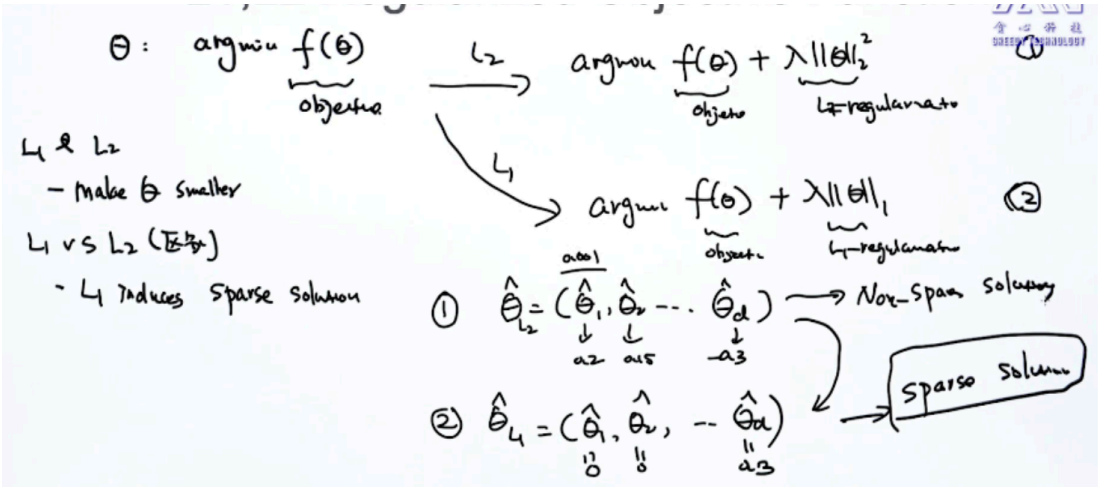
5.2 Regularization Terms



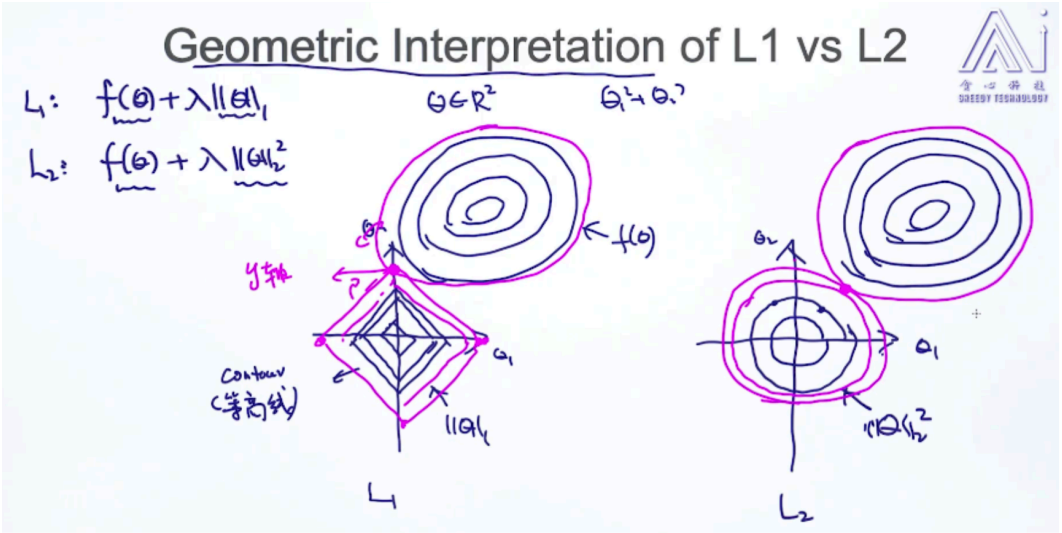
L1：会把很多参数变为 0.遇到比较稀疏的问题，可以选择用 L1
L1 和 L2 比较常见
第四种：nuclear norm
　　限制矩阵的 rank，把 rank 比较大的矩阵去掉
还有很多种范数。

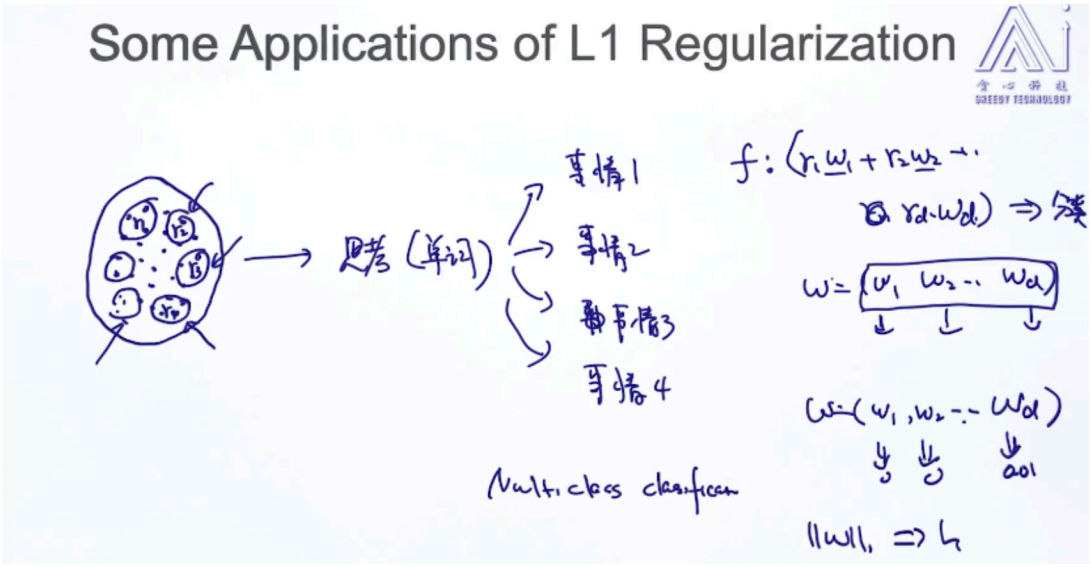


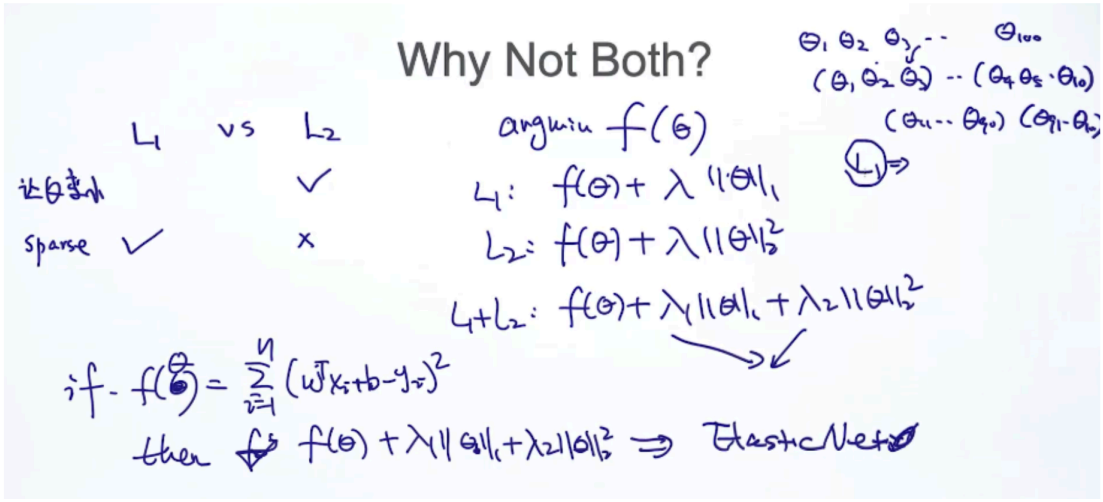
5.3 L1 vs L2



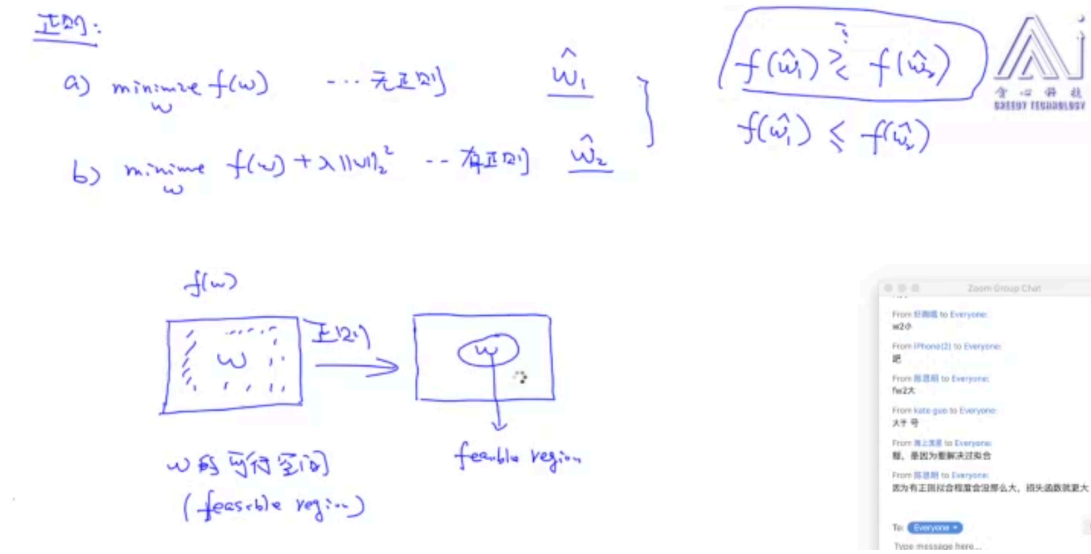
为什么 L1 稀疏，L2 不稀疏







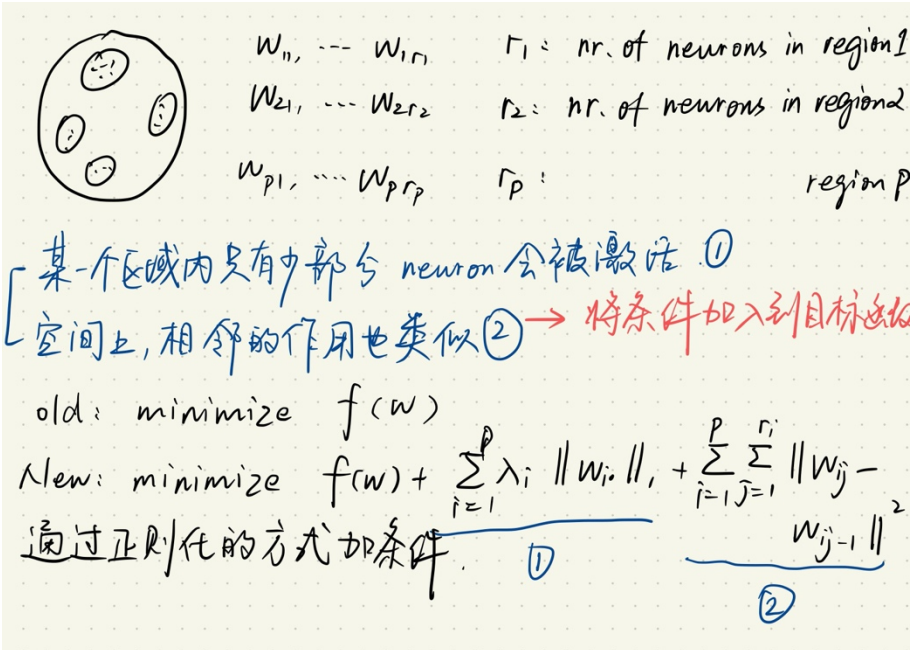
5.4 正则的作用



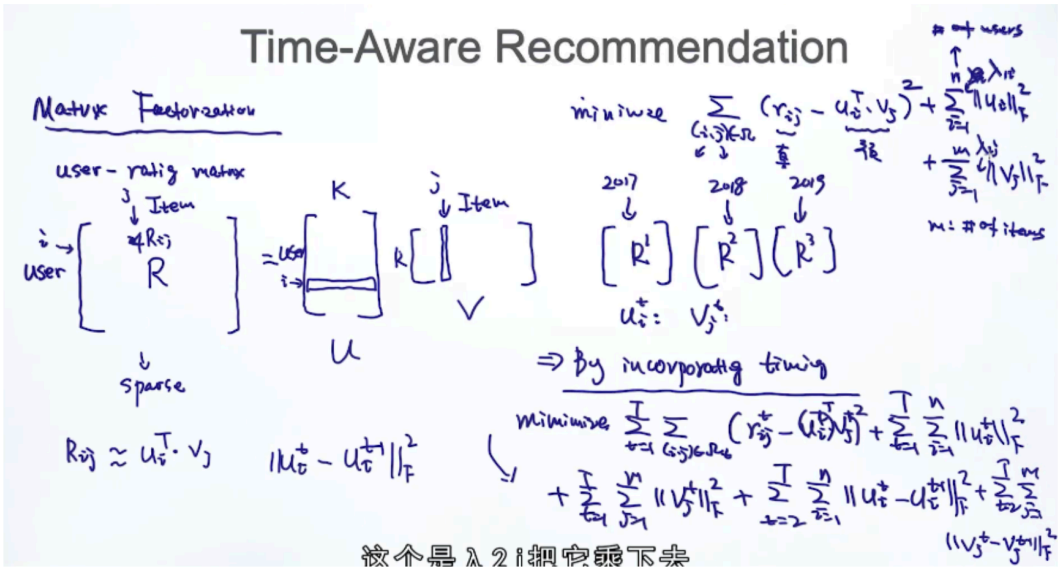
相当于对原来的空间做了过滤。

5.5 正则化的灵活运用

1) neuron science

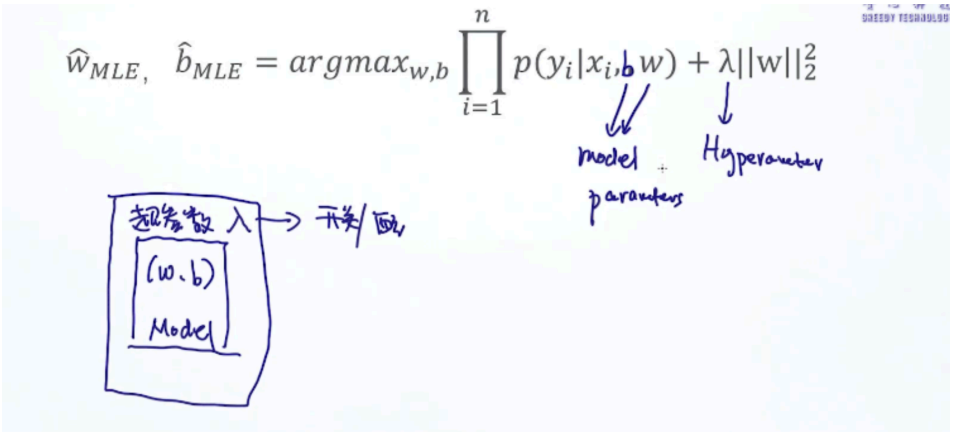


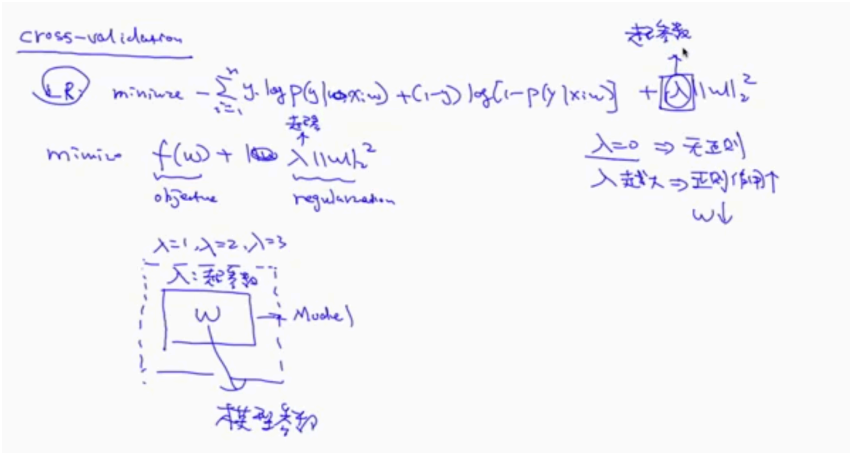
2) time-aware recommendation



6. cross validation 交叉验证

6.1 如何选择 lambda

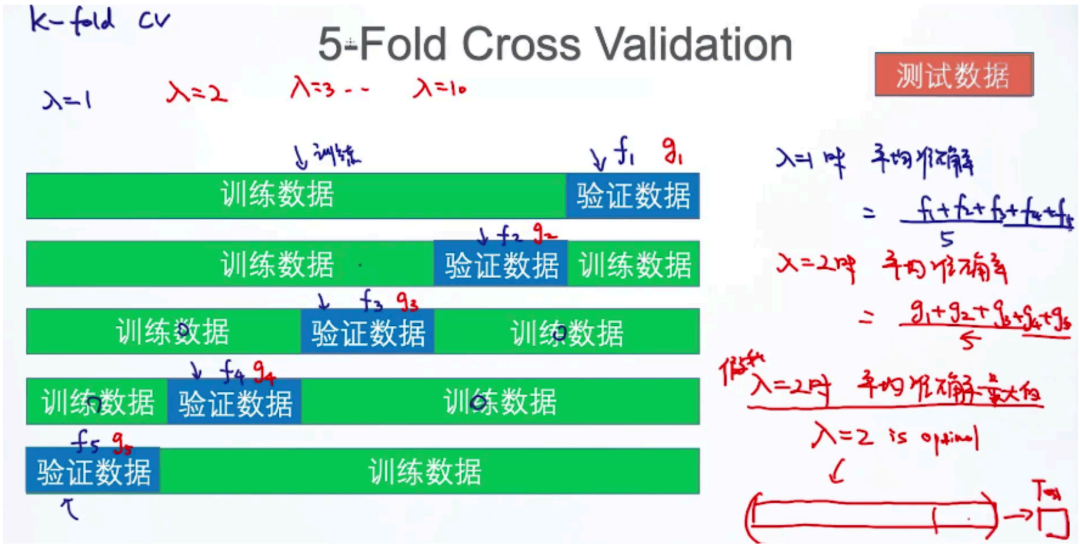




Intuition

把训练数据进一步分成训练数据（Training Data）和验证集（Validation Data）。 选择在验证数据里最好的超参数组合。

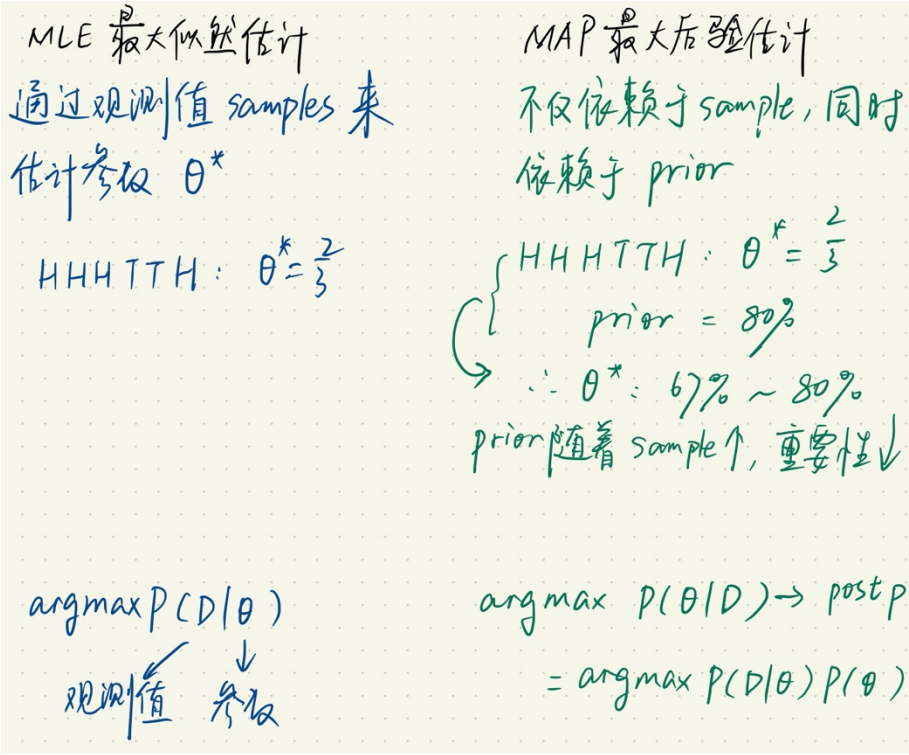
6.2 交叉验证



绝对不不能用测试数据来引导(guide)模型的训练!

7. MLE & MAP

7.1



7.2 from Gaussian prior to L2 regularization

$$P(D|w,b) = \prod_{i=1}^{n} P(y_i|x_i, w, b)$$

MLE: $\arg\max P(D|\theta)$

$= \arg\max \prod_{i=1}^{n} P(y_i|x_i; w)$

$= \arg\max \sum \log P(y_i|x_i; w)$

MAP: $\arg\max P(D|\theta) \cdot P(\theta)$

$= \arg\max \log P(D|\theta) + \log P(\theta)$

$= \arg\max \sum \log P(y_i|x_i; w) + \log P(\theta)$ $\quad \theta = \{w\}$

MLE ?

定义 $P(\theta) = P(w) \sim N(0, \sigma^2)$ (假设)

$P(w) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{w^2}{2\sigma^2}\right)$ → pdf

$\log P(\theta) = \log P(w)$

$= -\log(\sqrt{2\pi}\sigma) - \frac{w^2}{2\sigma^2}$

$= \arg\max \sum \log P(y_i|x_i; w) - \log(\sqrt{2\pi}\sigma) - \frac{w^2}{2\sigma^2}$

$= \arg\max \sum \log P(y_i|x_i; w) - \frac{1}{2\sigma^2}\|w\|_2^2$

L2 正则

$\lambda = \frac{1}{2\sigma^2}$

7.3 from Laplace prior to L1 regularization

$$P(D|w,b) = \prod p(y_i|x_i,w,b)$$

MLE: $\arg\max P(D|\theta)$

$$= \arg\max \sum_{i=1}^{n} \log p(y_i|x_i;w)$$

假设 $p(\theta) = p(w) \sim Laplace(\mu, b)$

$$p(w) \sim L(0,b)$$

$$= \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right)$$

MAP: $\arg\max \log p(D|\theta) + \log p(\theta)$

$$= \arg\max \sum \log p(y_i|x_i;w) + \log p(\theta)$$

$$\log p(\theta) = \log p(w)$$

$$= \log\left(\frac{1}{2b}\right) - \frac{|x|}{b}$$

$$= \arg\max \sum \log p(y_i|x_i;w) - \underbrace{\frac{|x|}{b}}$$

L1 regularization

$$\lambda = -\frac{1}{b}$$

7.4

Adding Prior is Equivalent to Regularization

Logistic Regression

任何模型

- Gaussian Prior ⇒ L_2 Reg
- Laplace Prior ⇒ L_1 Reg

7.5 MAP→MLE

当数据非常多的时候, MAP Solution ⟶ MLE Solution

MAP: $\arg\max \log p(D|\theta) + \log p(\theta)$

$$= \arg\max \log \underbrace{\sum_{i=1}^{n} \log p(y_i|x_i;w)}_{\text{MLE}} + \underbrace{\log p(\theta)}_{\text{prior}}$$

$n \to +\infty$ 时 MAP 解 = MLE 解

8. 参数搜索策略

$$L = \prod_{i=1}^{n} p(y_i|x_i, w, b) + \lambda_1||w||_1 + \lambda_2||w||_2^2$$

How to search for λ_1 and λ_2 ?

8.1 Grid Search

$$L = \prod_{i=1}^{n} p(y_i|x_i, w, b) + \lambda_1||w||_1 + \lambda_2||w||_2^2$$

Best (λ_1, λ_2) 组合

iter_λ_1 = [0.001, 0.01, 0.05, 0.1, 0.5, 1, 5, 10]

iter_λ_2 = [0.001, 0.01, 0.1, 1, 10, 100]

for λ_1 in iter_λ_1 :

  for λ_2 in iter_λ_2 :

    (λ_1, λ_2) ← score(手动调参)

    ↓

    交叉验证

    ↑

Advantage :

1) 完全可以并行

8.2 heuristic search

Heuristic Search ① λ_1, λ_2

$\lambda_1 \in (0.001, \quad 10)$

$\lambda_2 \in (0.01, \quad 17)$

for (λ_1, λ_2)

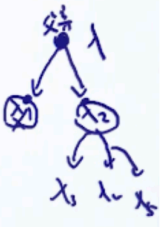
Grid Search

Candidate Set

随机搜索（Random Search） ①

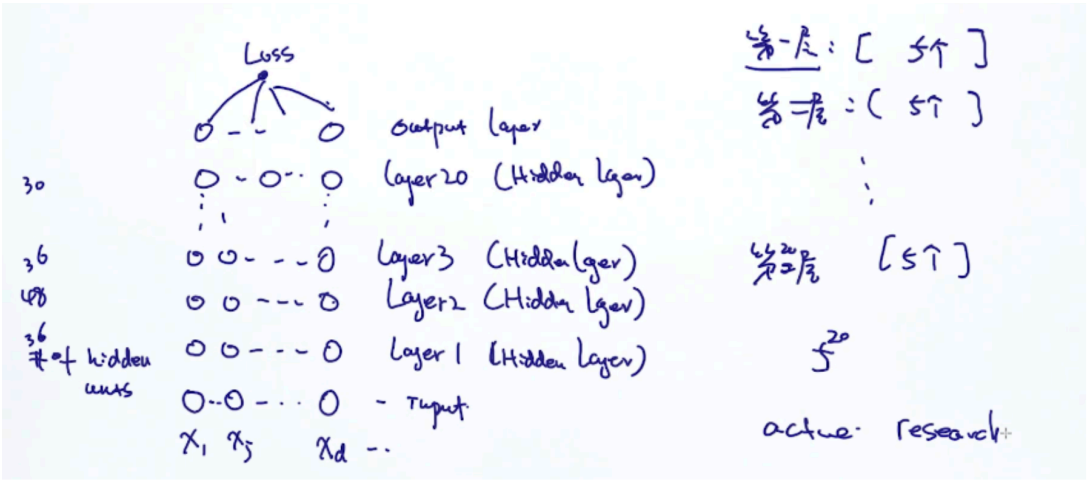
遗传算法（Genetic/Evolutionary Algorithm） ②

贝叶斯优化（Bayesian Optimization） ③



8.3 deep network

每一层都包含了一个超参数，神经元的个数



9. summary

- 好的模型拥有高的泛化能力
- 越复杂的模型越容易过拟合
- 添加正则项是防止过拟合的一种手段
- L1正则会带来系数特性
- 选择超参数时使用交叉验证
- 参数搜索过程最耗费资源