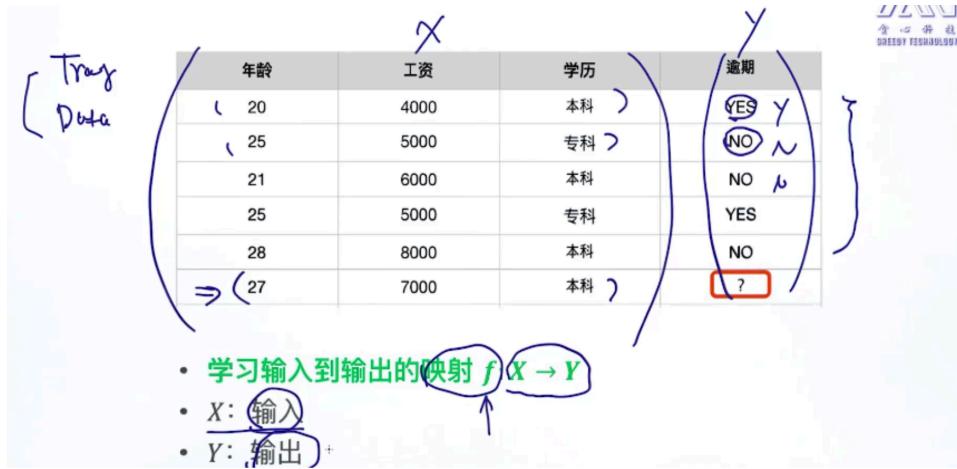


## 1. Logistic Regression

### 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

学习输入到输出的映射  $f: X \rightarrow Y$

$X$ : 输入

$Y$ : 输出

- 定义条件概率:  $P(Y|X) = ?$   $\Leftrightarrow (20, 4000, \text{本科}) (Y=)$

- 假设我们明确知道条件概率  $P(Y|X)$ , 怎么做分类?

$$P(Y_1|x) \geq ? \quad P(Y_0|x) \quad \text{if} \quad P(Y_1|x) > P(Y_0|x)$$

核心: 怎么去设计条件概率

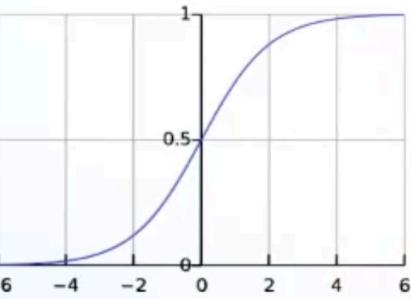
- 可不可以用线性回归来表示  $P(Y|X) = w^T x + b$ ? 为什么? No!

$$P(y|x) = \frac{1}{1 + e^{-w^T x - b}}$$

①  $0 \leq P(y|x) \leq 1$   $\otimes$

$$\textcircled{2} \quad \sum_y P(y|x) = 1$$

### 1.3 logistic function



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

$x: (-\infty, +\infty)$

$y: (0, 1)$

逻辑函数  $y = \frac{1}{1+e^{-x}}$

原始条件概率 :  $P(Y|X) = w^T x + b$

新条件概率  $P(Y|X) = \frac{1}{1+e^{-(w^T x + b)}}$

$$\omega = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}, b \in \mathbb{R}$$

对于二分类问题：

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

$$p(y=0|x, w) = \frac{e^{-w^T x + b}}{1 + e^{-w^T x + b}} = 1 - p(y=1|x, w)$$

两个式子可以合并成：

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

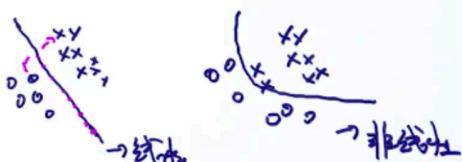
$$\text{if } y=1 \text{ 时. } P(Y=1|x, w) = p(y=1|x, w)$$

## 1.4 linear classifier

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

$$p(y=0|x, w) = \frac{e^{-w^T x + b}}{1 + e^{-w^T x + b}}$$

决策边界 (Decision Boundary)



## 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：

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

$$\begin{aligned} \hat{w}_{MLE}, \quad \hat{b}_{MLE} &= \operatorname{argmax}_{w,b} \prod_{i=1}^n p(y_i|x_i, w, b) \quad \textcircled{1} \\ &= \operatorname{argmax}_{w,b} \log \left( \prod_{i=1}^n p(y_i|x_i, w, b) \right) \\ &= \operatorname{argmax}_{w,b} \sum_{i=1}^n \log p(y_i|x_i, w, b) \end{aligned}$$

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

Objective Function

$$\begin{aligned} \operatorname{argmax}_{w,b} & \sum_{i=1}^n \log p(y_i|x_i, w, b) \\ \operatorname{argmin}_{w,b} & - \sum_{i=1}^n \log p(y_i|x_i, w, b) \\ &= \operatorname{argmin}_{w,b} - \sum_{i=1}^n \log \left[ p(y_i|x_i, w, b)^y \cdot [1 - p(y_i|x_i, w, b)]^{1-y} \right] \\ &= \operatorname{argmin}_{w,b} - \sum_{i=1}^n y \cdot \log p(y_i|x_i, w, b) + (1-y) \log [1 - p(y_i|x_i, w, b)] \end{aligned}$$

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

## 2. 梯度下降

### 2.1 求解函数的最小值

求使得  $f(w)$  值最小的参数  $w$

- 是否凸函数
- 最优化算法

1)  $f(w) = 0 \Rightarrow w = ?$

2) Iterative

- GD (Gradient Descent)
- SGD (Stochastic GD)

Global Optimal vs Local Optimal

closed form  
Global convex

Non-Convex (非凸)

### 2.2 梯度下降

求使得  $f(w)$  值最小的参数  $w$

初始化  $w^1$   
for  $t = 1, 2, \dots$ :  
 $w^{t+1} = w^t - \eta \nabla f(w^t)$

### Gradient Descent

求使得  $f(w)$  值最小的参数  $w$

初始化  $w^1$   
for  $t = 1, 2, \dots$ :  
 $w^{t+1} = w^t - \eta \nabla f(w^t)$

例子：求解函数  $f(w) = 4w^2 + 5w + 1$  的最优解

$\eta = -\frac{b}{2a} = -\frac{5}{8}$

$w^1 = 0, f(w) = 8w + 5, \eta = 0.1$   
 $w^2 = w^1 - \eta \cdot (8 \cdot 0 + 5) = 0 - 0.5 = -0.5$   
 $w^3 = w^2 - \eta \cdot (8 \cdot (-0.5) + 5) = -0.5 - 0.5 = -1.0$   
 $w^4 = w^3 - \eta \cdot (8 \cdot (-1.0) + 5) = -1.0 - 0.5 = -1.5$   
 $w^5 = w^4 - \eta \cdot (8 \cdot (-1.5) + 5) = -1.5 - 0.5 = -2.0$

0.625 左右所以我们通过这种计算的方法

## 2.3 逻辑回归的梯度下降

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

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

$$\begin{aligned}
 &= \operatorname{argmin}_{w,b} - \sum_{i=1}^n y_i \cdot \log \sigma(w^T x_i + b) + (1-y_i) \cdot \log [1 - \sigma(w^T x_i + b)] \\
 \boxed{\frac{\partial L(w,b)}{\partial w}} &= - \sum_{i=1}^n y_i \cdot \frac{\sigma(w^T x_i + b) \cdot [1 - \sigma(w^T x_i + b)]}{\sigma(w^T x_i + b)} \cdot x_i + (1-y_i) \cdot \frac{\sigma(w^T x_i + b) \cdot [1 - \sigma(w^T x_i + b)]}{1 - \sigma(w^T x_i + b)} \cdot x_i \\
 &= - \sum_{i=1}^n y_i \cdot (1 - \sigma(w^T x_i + b)) + (1-y_i) \cdot \sigma(w^T x_i + b) \cdot x_i \\
 &\Rightarrow \boxed{\sum_{i=1}^n [y_i - \sigma(w^T x_i + b)] \cdot x_i} = \boxed{\sum_{i=1}^n [\sigma(w^T x_i + b) - y_i] \cdot x_i}
 \end{aligned}$$

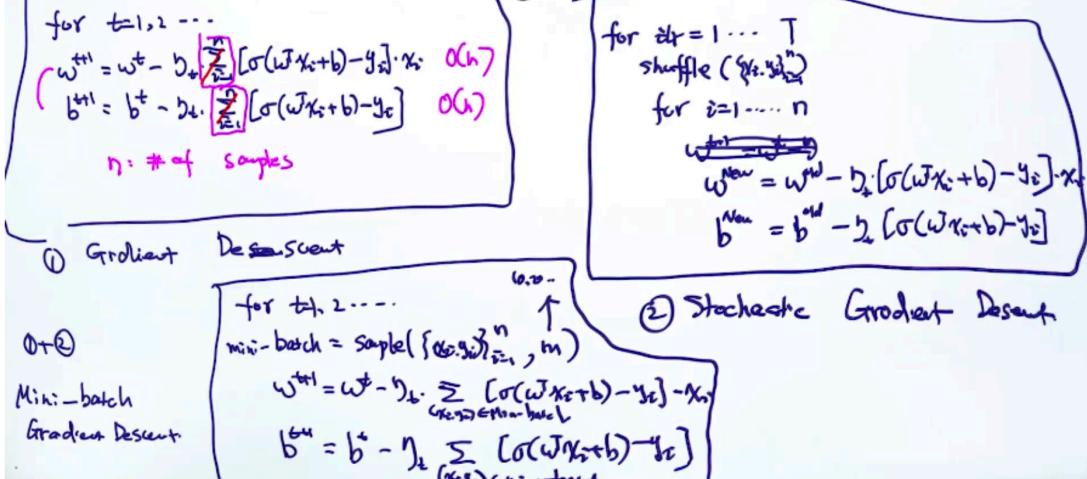
$$\begin{aligned}
 L(w,b) &= - \sum_{i=1}^n y_i \cdot \sigma(w^T x_i + b) + (1-y_i) \log [1 - \sigma(w^T x_i + b)] \\
 \boxed{\frac{\partial L(w,b)}{\partial b}} &= - \sum_{i=1}^n y_i \cdot \frac{\sigma(w^T x_i + b) \cdot [1 - \sigma(w^T x_i + b)]}{\sigma(w^T x_i + b)} + (1-y_i) \cdot \frac{-\sigma(w^T x_i + b) \cdot [1 - \sigma(w^T x_i + b)]}{1 - \sigma(w^T x_i + b)} \\
 &= - \sum_{i=1}^n y_i \cdot (1 - \sigma(w^T x_i + b)) + (1-y_i) \cdot (-\sigma(w^T x_i + b)) \\
 &\Rightarrow \boxed{\sum_{i=1}^n [\sigma(w^T x_i + b) - y_i]}
 \end{aligned}$$

初始化  $w^0, b^0$   
for  $t=1, 2, \dots$       learning rate  
 $w^{t+1} = w^t - \eta \cdot \sum_{i=1}^n [\sigma(w^T x_i + b^t) - y_i] \cdot x_i$   
 $b^{t+1} = b^t - \eta \cdot \sum_{i=1}^n [\sigma(w^T x_i + b^t) - y_i]$

Gradient Descent



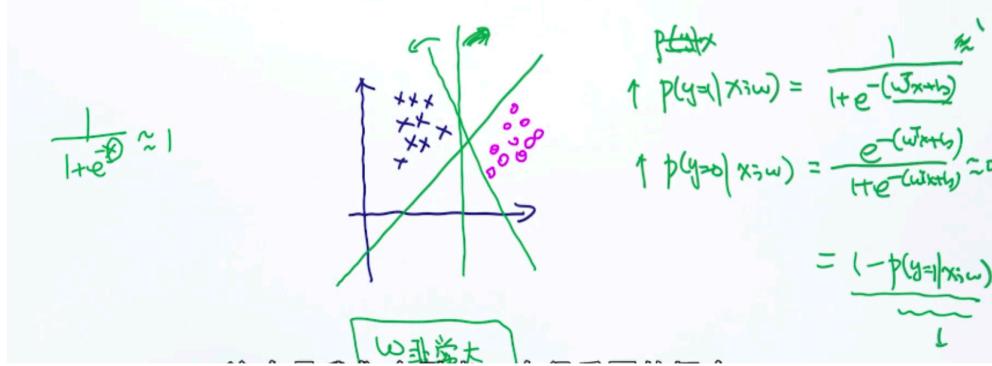
### Stochastic Gradient Descent for Logistic Regression



### 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

Problem: avoid  $w$  become too large

Adding a Term – L2 Norm (正则化)

$|w| \gg 100/1000,$   
 $\downarrow$  magnitude  
 $\hat{w}_{MLE}, \hat{b}_{MLE} = \underset{\substack{\min \\ w, b}}{\operatorname{argmax}} \prod_{i=1}^n p(y_i | x_i, w) + \lambda \cdot \|w\|_2^2$

目标函数  
 $\hat{w}_{MLE} \text{ 很大} \Rightarrow \|w\|_2^2 \text{ 变得很大}$

$\lambda$ : Hyperparameter (超参数)  $\Rightarrow$  weighting factor  
 a: if  $\lambda=0$  时, 没有任何限制  
 b: 入大的时候,  $w$  变得更小  
 c: 入小的 "",  $w$  变得更大.

$\|w\|_2^2 = w_1^2 + w_2^2 + \dots + w_d^2$   
 $\downarrow$   
 L2-Norm

$$\hat{w}_{MLE}, \hat{b}_{MLE} = \underset{\substack{\min \\ w, b}}{\operatorname{argmax}} \prod_{i=1}^n p(y_i | x_i, w, b) + \lambda \cdot \|w\|_2^2$$

$$L(w, b) \quad R(w)$$

$$\frac{\partial L(w, b)}{\partial w} = \sum_{i=1}^n (\underbrace{\sigma(w^T x_i + b)}_{\text{prediction}} - \underbrace{y_i}_{\text{actual}}) \cdot x_i$$

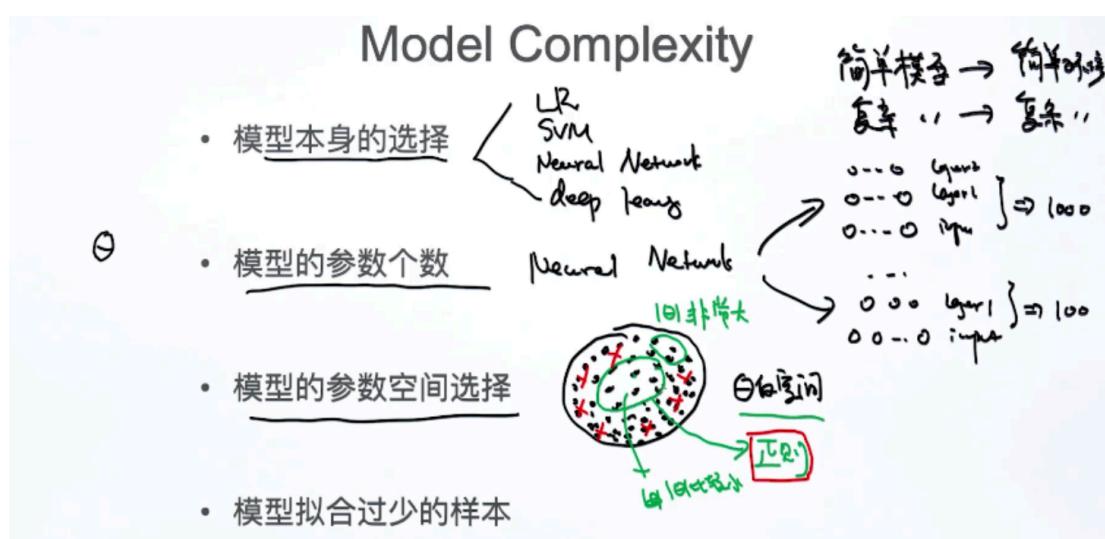
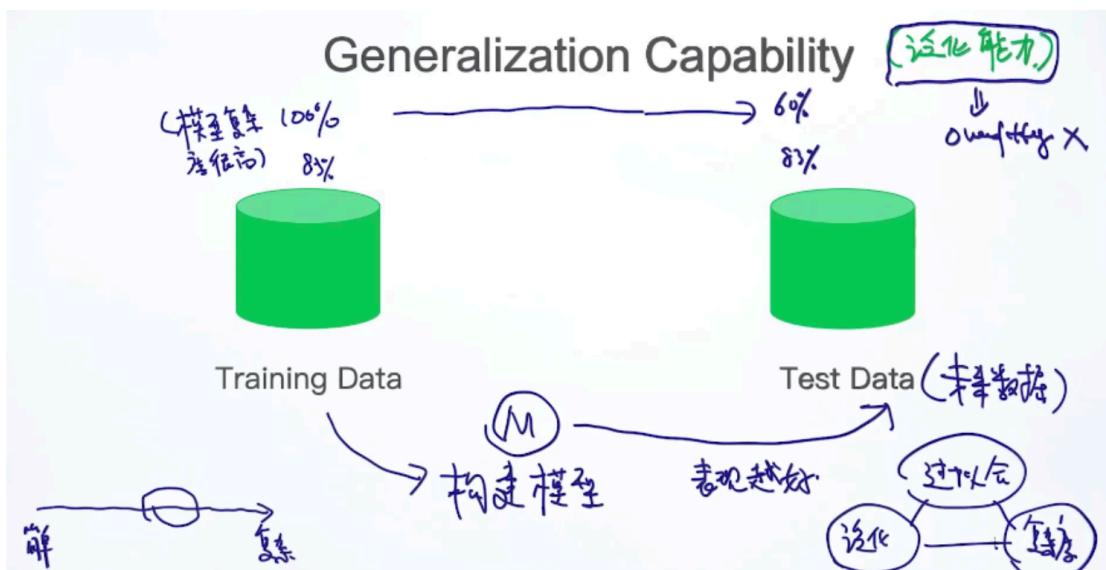
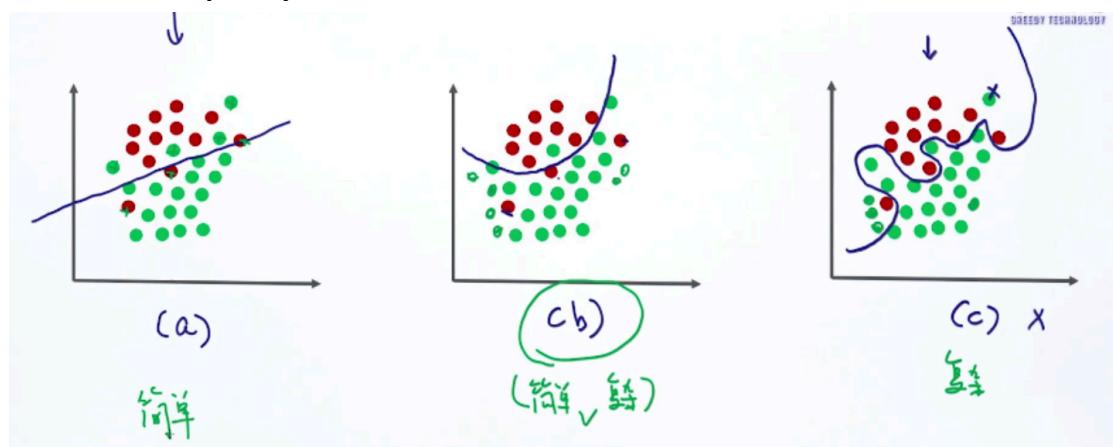
$$\frac{\partial L(w, b) + R(w)}{\partial w} = \sum_{i=1}^n (\sigma(w^T x_i + b) - y_i) \cdot x_i + 2 \cdot \lambda w \quad (\text{Batch Gradient Descent})$$

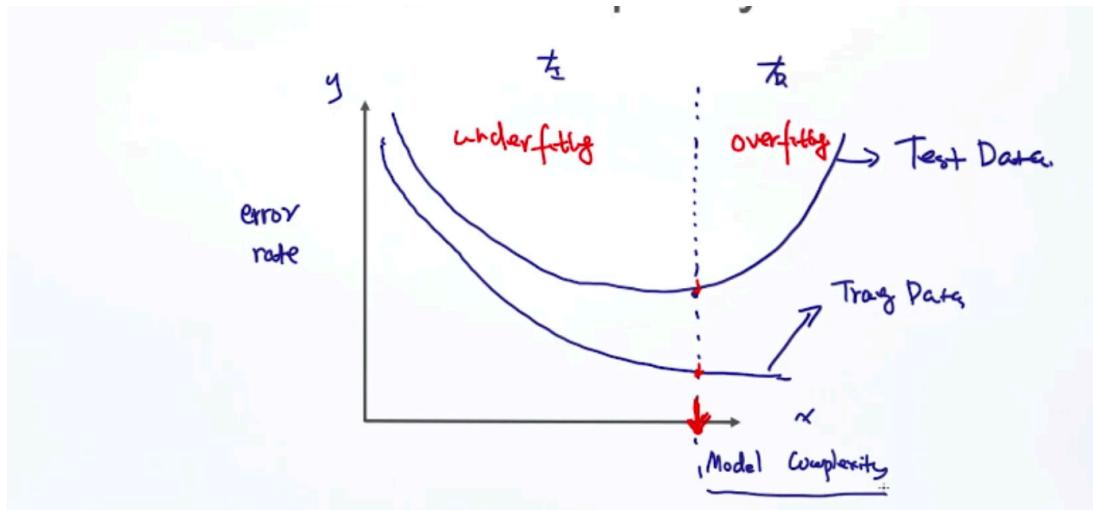
$$\frac{\partial L(w, b) + R(w)}{\partial w} = (\sigma(w^T x_i + b) - y_i) \cdot x_i + 2 \lambda w \quad (\text{Stochastic gradient descent})$$

我们就采用这种方式去不断地更新咱们的  $w$  的

## 4. 过拟合

## 4.1 model complexity





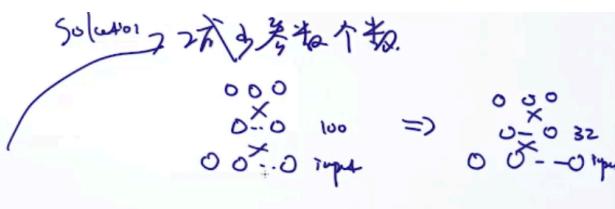
## 4.2 avoid overfitting

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

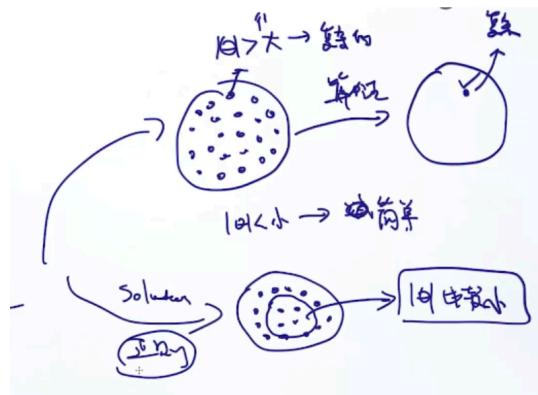
Solution → 选择更简单的模型  
(LR, SVM Class)

- 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

solver → 手动取更多样本

## 5. 正则化

### 5.1 recap

$$\text{L}_2\text{-Norm} \rightarrow \text{LR}$$

$$\arg \min - \underbrace{\sum_{i=1}^n P(y_i | x_i; w)}_{\text{objective}} + \lambda \underbrace{\|w\|_2^2}_{\text{regularization}}$$

Hyperparameter (超参数)

$\lambda \uparrow \rightarrow \|w\|_2 \uparrow$

$\lambda \downarrow \rightarrow \|w\|_2 \downarrow$

$\lambda = 0 \text{ 时 } \text{没有正则限制}$

$$\|w\|_2^2: \text{L}_2\text{-Norm} \quad w_1^2 + w_2^2 + \dots + w_d^2 = \|w\|_2^2$$

### 5.2 Regularization Terms

**L0-Norm**:  $\|w\|_0$

**Nuclear Norm**:  $\|A\|_*$ ,  $\|A\|_F$ ,  $\text{rank}(A)$

**L1-Norm**:  $\|w\|_1 = |w_1| + |w_2| + |w_3| + \dots + |w_d|$

$$\begin{aligned} &= \sqrt{\sum_{i=1}^d |w_i|^2} \\ &= \sqrt{\|w\|_2^2} \end{aligned}$$

**L2-Norm**:  $\|w\|_2$

**L<sub>F</sub>-Norm**:  $\|A\|_F = \sqrt{\sum_{i=1}^M \sum_{j=1}^N |a_{ij}|^2}$

**Sparse**

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。

$$\widehat{w}_{MLE}, \widehat{b}_{MLE} = \arg \max_{w,b} \underbrace{\prod_{i=1}^n p(y_i|x_i, w, b)}_{\text{objective}} + \lambda \underbrace{\|w\|_1}_{\text{regularization}}$$

$$\|w\|_1 \Rightarrow L_1\text{-Norm of } w$$

$$\|w\|_1 = |w_1| + |w_2| + \dots + |w_d|$$

$L_1 \leq L_2$

Magnitude of  $w$  较小

### 5.3 L1 vs L2

$$\Theta: \underset{\text{objective}}{\arg \min f(\theta)} \xrightarrow{L_2} \underset{\text{objective}}{\arg \min f(\theta)} + \lambda \underset{\text{regularization}}{\| \theta \|_2^2}$$

$L_1 \leq L_2$

- Make  $\theta$  smaller

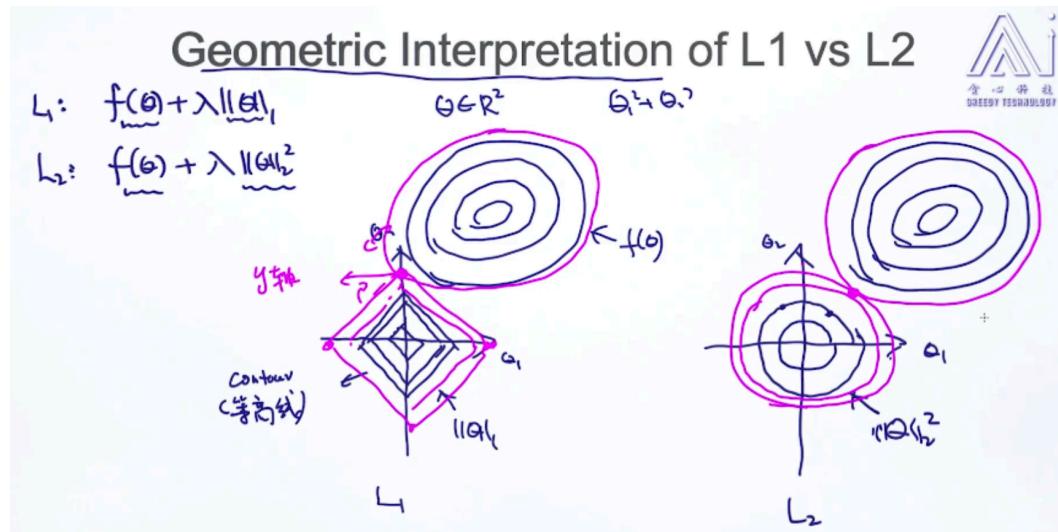
$L_1 \leq L_2$  (区别)

- $L_1$  induces Sparse Solution

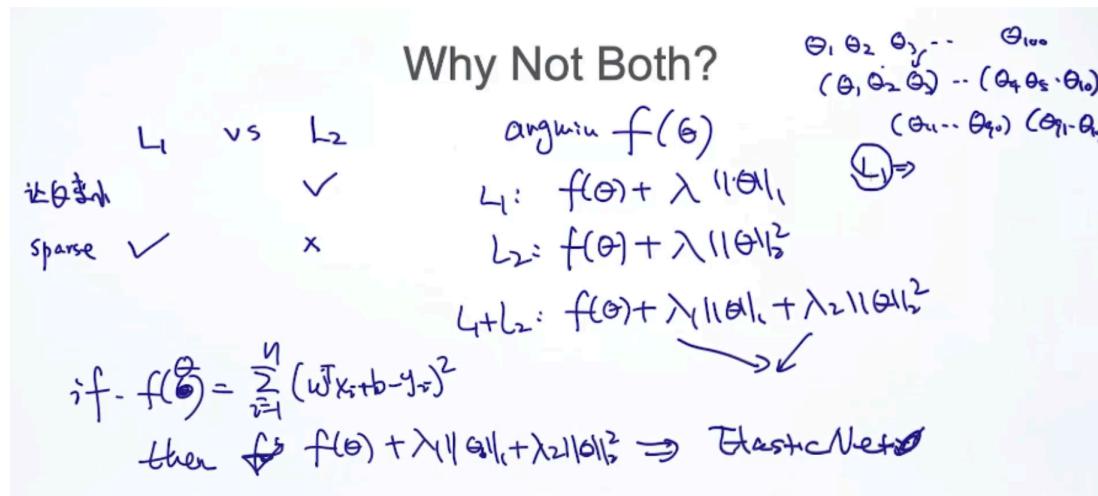
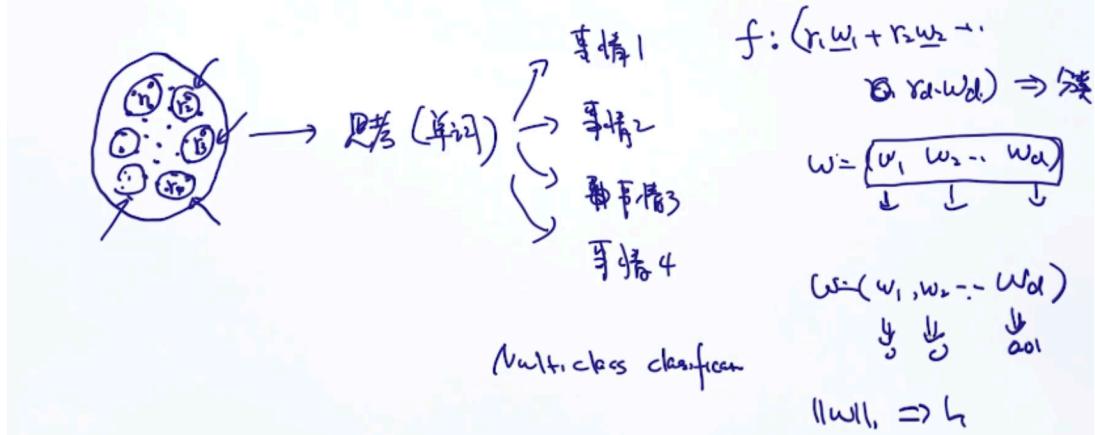
①  $\hat{\theta}_{L_2} = (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_d) \xrightarrow{\text{non-zero}} \text{Non-Sparse Solution}$

②  $\hat{\theta}_{L_1} = (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_d) \xrightarrow{\text{zero}} \text{Sparse Solution}$

为什么  $L_1$  稀疏,  $L_2$  不稀疏



# Some Applications of L1 Regularization



## 5.4 正则的作用

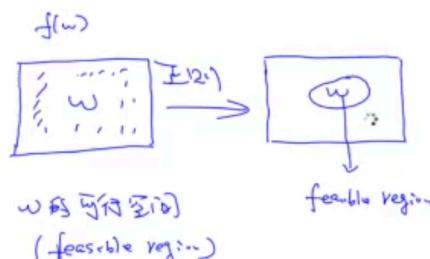
正则:

$$\begin{aligned} a) \min_w f(w) & \cdots \text{无正则} & \frac{\hat{w}_1}{\hat{w}_2} \end{aligned}$$

$$b) \min_w f(w) + \lambda \|w\|_2^2 \cdots \text{有正则}$$

$f(\hat{w}_1) \leq f(\hat{w}_2)$

$f(\hat{w}_1) \leq f(\hat{w}_3)$



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

Diagram showing neurons in three regions:

- Region 1:  $w_{11}, \dots, w_{1r_1}$ ,  $r_1$ : nr. of neurons in region 1
- Region 2:  $w_{21}, \dots, w_{2r_2}$ ,  $r_2$ : nr. of neurons in region 2
- Region P:  $w_{P1}, \dots, w_{Pr_p}$ ,  $r_p$ : nr. of neurons in region P

Handwritten notes:

- [某一个区域内只有部分 neuron 会被激活] ①
- [空间上, 相邻的作用也类似] ② → 将条件加入到目标函数

old: minimize  $f(w)$

New: minimize  $f(w) + \sum_{i=1}^P \lambda_i \|w_{i\cdot}\|_F + \sum_{i=1}^P \sum_{j=1}^{r_i} \|w_{ij} - w_{i,j-1}\|^2$

通过正则化的方式加条件 ① ②

### 2) time-aware recommendation

### Time-Aware Recommendation

Matrix Factorization

User rating matrix  $R$  (User  $\times$  Item) is approximated by  $U$  (User  $\times$  K) and  $V$  (K  $\times$  Item).

$$R \approx U V^T$$

$U$  is sparse.

$R_{ij} \approx u_i^T \cdot v_j$

$\|u_i^T - u_i^{\text{true}}\|_F^2$

minimize  $\sum_{(i,j) \in R} (r_{ij} - u_i^T \cdot v_j)^2 + \frac{\lambda_u}{2} \|u_i^T\|_F^2 + \frac{\lambda_v}{2} \|v_j\|_F^2$

$u_i^{\text{true}} = v_j^{\text{true}}$

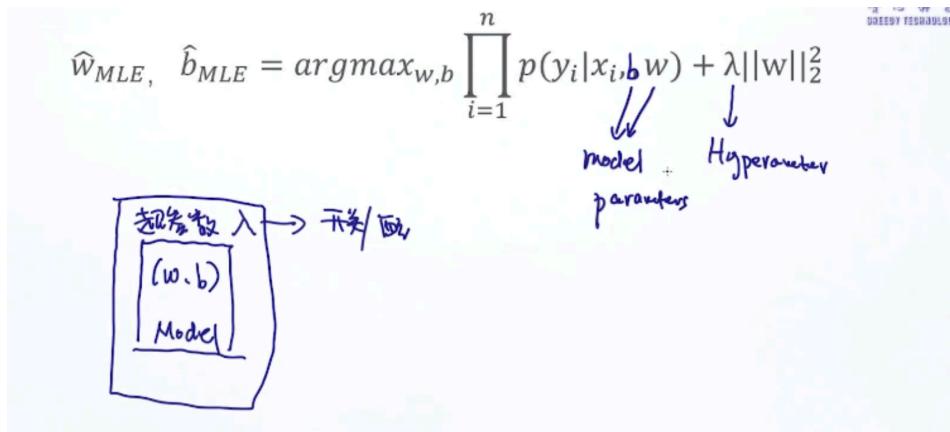
$\Rightarrow$  by incorporating timing

minimize  $\sum_{t=1}^T \sum_{(i,j) \in R_{it}} (r_{ij}^t - (U_i^t)^T V_j^t)^2 + \frac{\lambda_u}{2} \sum_{t=1}^T \|U_i^t\|_F^2 + \frac{\lambda_v}{2} \sum_{t=1}^T \|V_j^t\|_F^2 + \sum_{t=2}^T \sum_{i,j} \|U_i^t - U_i^{\text{true}}\|_F^2 + \sum_{t=2}^T \sum_{i,j} \|V_j^t - V_j^{\text{true}}\|_F^2$

这个是  $\lambda$  2 里面的内容

## 6. cross validation 交叉验证

### 6.1 如何选择 lambda



Cross-validation

$\text{LR}$  minimize  $-\sum_{i=1}^n y_i \log P(y_i|x_i; w) + (1-y_i) \log [1 - P(y_i|x_i; w)] + \lambda ||w||_2^2$

$\begin{aligned} \text{minimize } & f(w) + \lambda ||w||_2^2 \\ & \text{objective} \quad \text{regularization} \end{aligned}$

$\lambda = 0 \Rightarrow \text{无正则}$   
 $\lambda \text{越大} \Rightarrow \text{正则作用} \uparrow$   
 $w \downarrow$

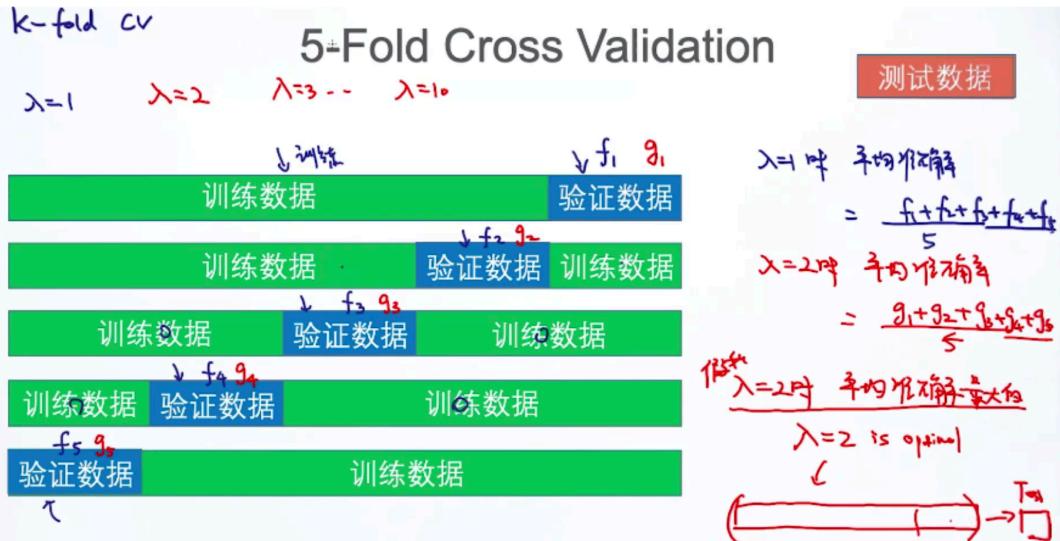
$\lambda = 1, \lambda = 2, \lambda = 3$   
 $\vdots$   
 入: 超参数  
 W → Model  
 模型参数

### 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



绝对不能用测试数据来引导(guide)模型的训练!

## 7. MLE & MAP

$$\text{MLE} \quad P(D|\theta)$$

$$\text{MAP} \quad P(\theta|D)^{\text{post}} \propto P(D|\theta) \cdot P(\theta)$$

$$\begin{aligned} \arg \max p(\theta|D) &= \arg \max P(D|\theta) P(\theta) \\ &= \underbrace{\arg \max \log P(D|\theta)}_{\text{MLE}} + \underbrace{\log P(\theta)}_{\text{regularization}} \end{aligned}$$

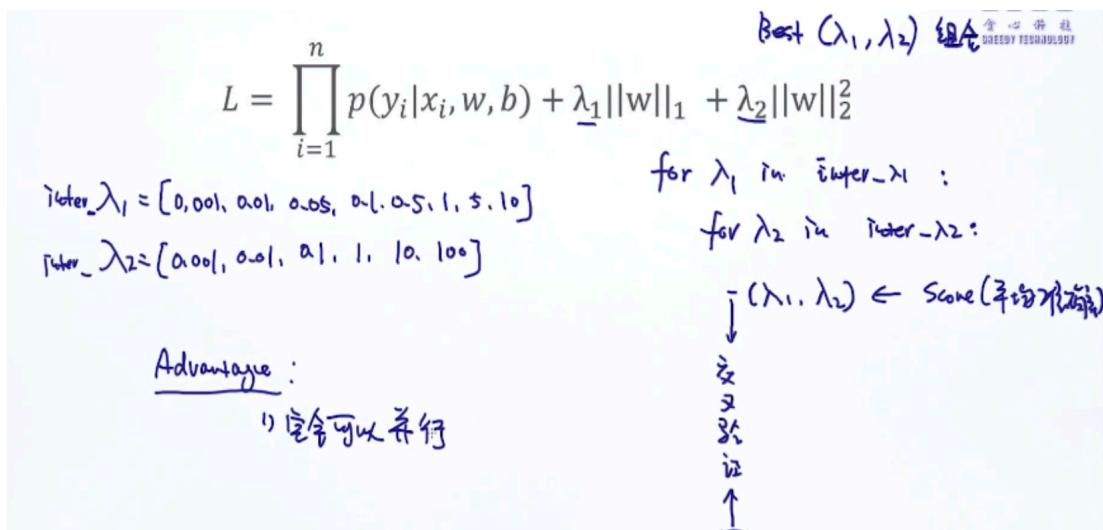
prior会随着样本数据的增加而变大，它重要性会变小

## 8. 参数搜索策略

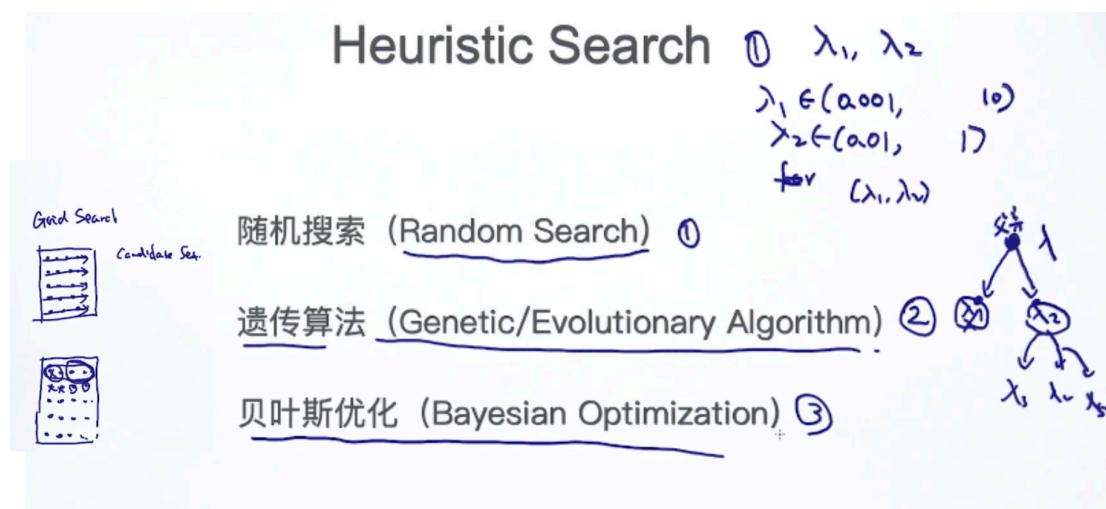
$$L = \prod_{i=1}^n p(y_i|x_i, w, b) + \lambda_1 ||w||_1 + \lambda_2 ||w||_2^2$$

How to search for  $\lambda_1$  and  $\lambda_2$ ?

### 8.1 Grid Search

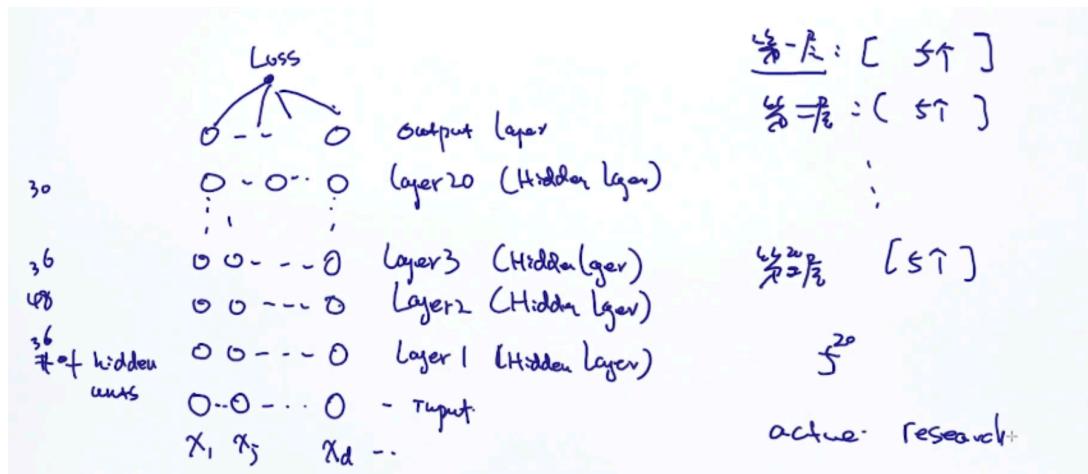


## 8.2 heuristic search



## 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