第2章

线性模型

Linear Models

赫 然

rhe@nlpr.ia.ac.cn

智能感知与计算研究中心(CRIPAC)

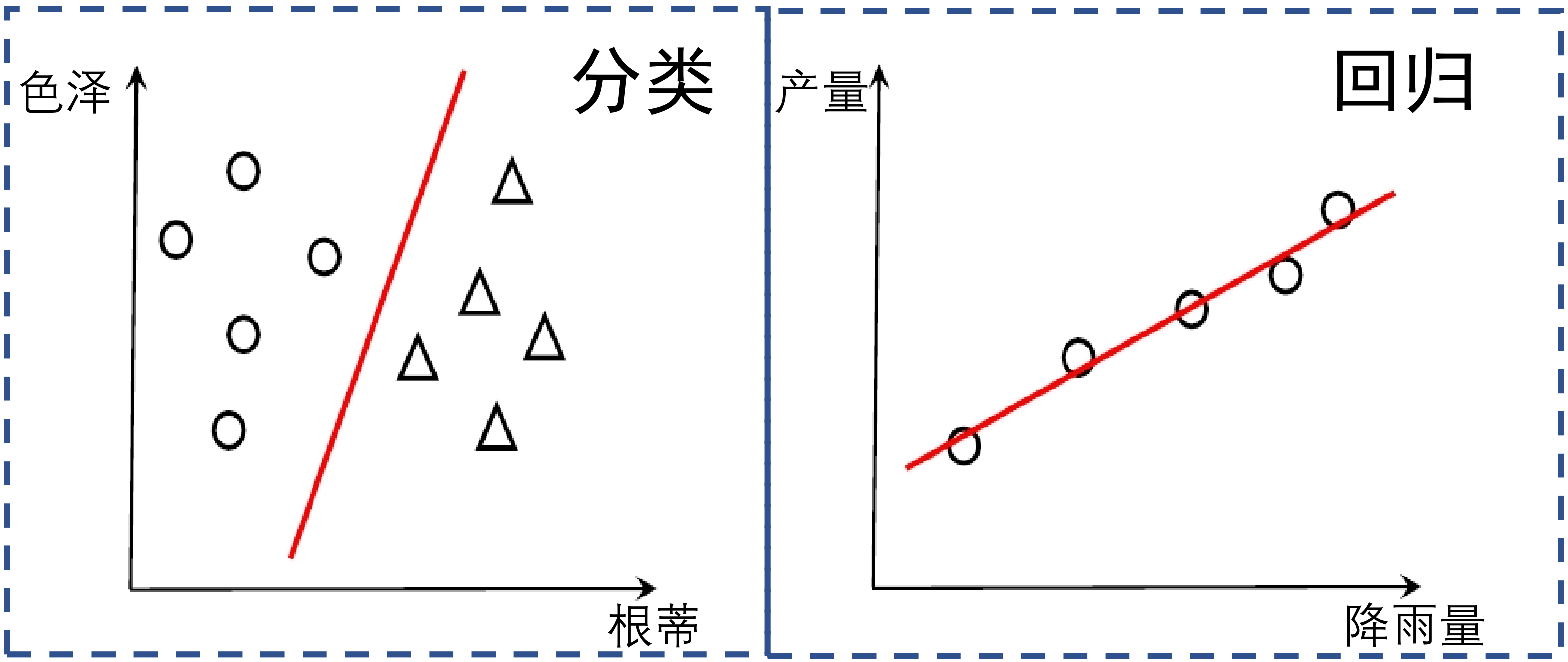
中科院自动化研究所　模式识别国家重点实验室

University of Chinese Academy of Sciences

内容提要

- 第二章 线性模型
  - 2.1 基本形式
  - 2.2 线性回归
  - 2.3 对数几率回归
  - 2.4 Softmax回归
  - 2.5 线性判别分析
  - 2.6 局部线性判别分析
  - 2.7 多分类学习
  - 2.8 类别不平衡问题

2.1 基本形式



- 给定由 d 个特征描述的示例（样本）$x = [x_1; x_2; \ldots x_d]$，线性模型的目标是学习如下关于特征的组合函数：

$$f(x) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b = w^T x + b$$

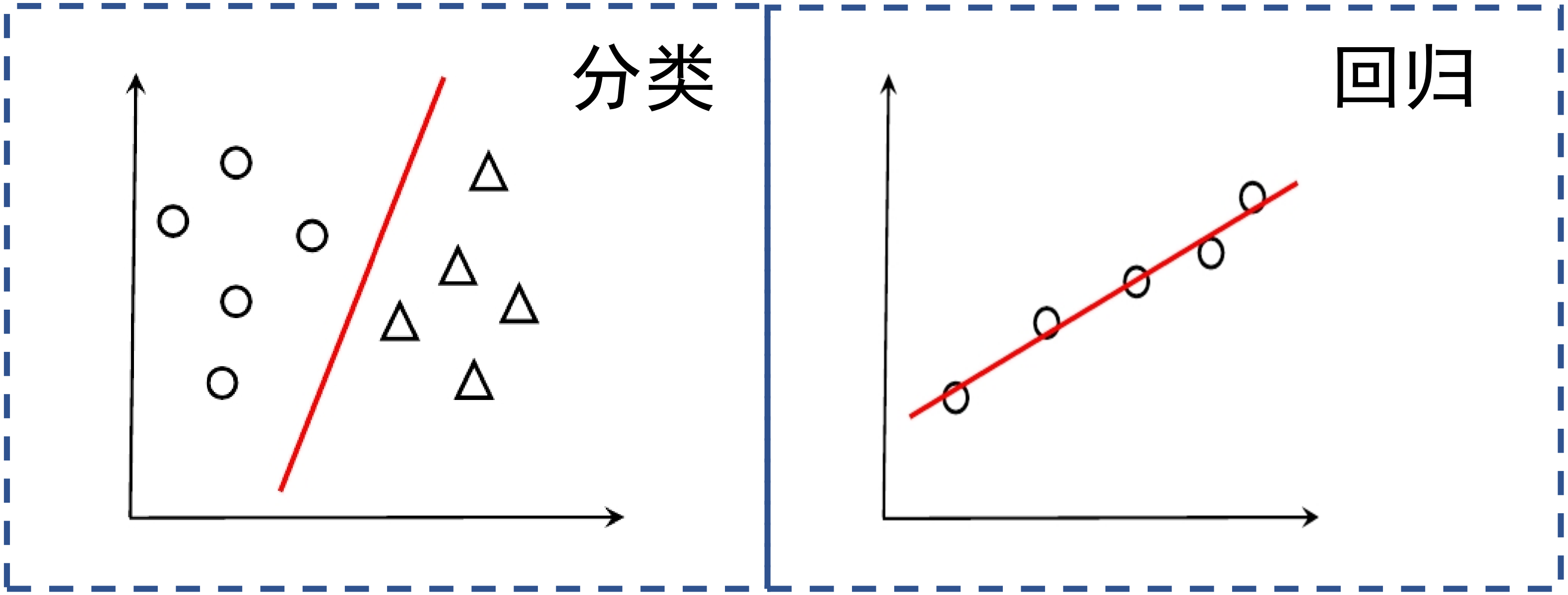
向量形式

2.1 基本形式

- 在线性模型中，w直观地表达了各属性在预测中的重要性，因此线性模型具有很好的可解释性。比如在西瓜问题中：

$$f_{好瓜}=0.2x_{色泽}+0.5x_{根蒂}+0.3x_{敲声}+1$$

- 线性模型简单，易于建模，但却蕴含着机器学习中的一些重要思想。许多非线性模型可在线性模型的基础上通过引入高维映射或者层级结构来得到。

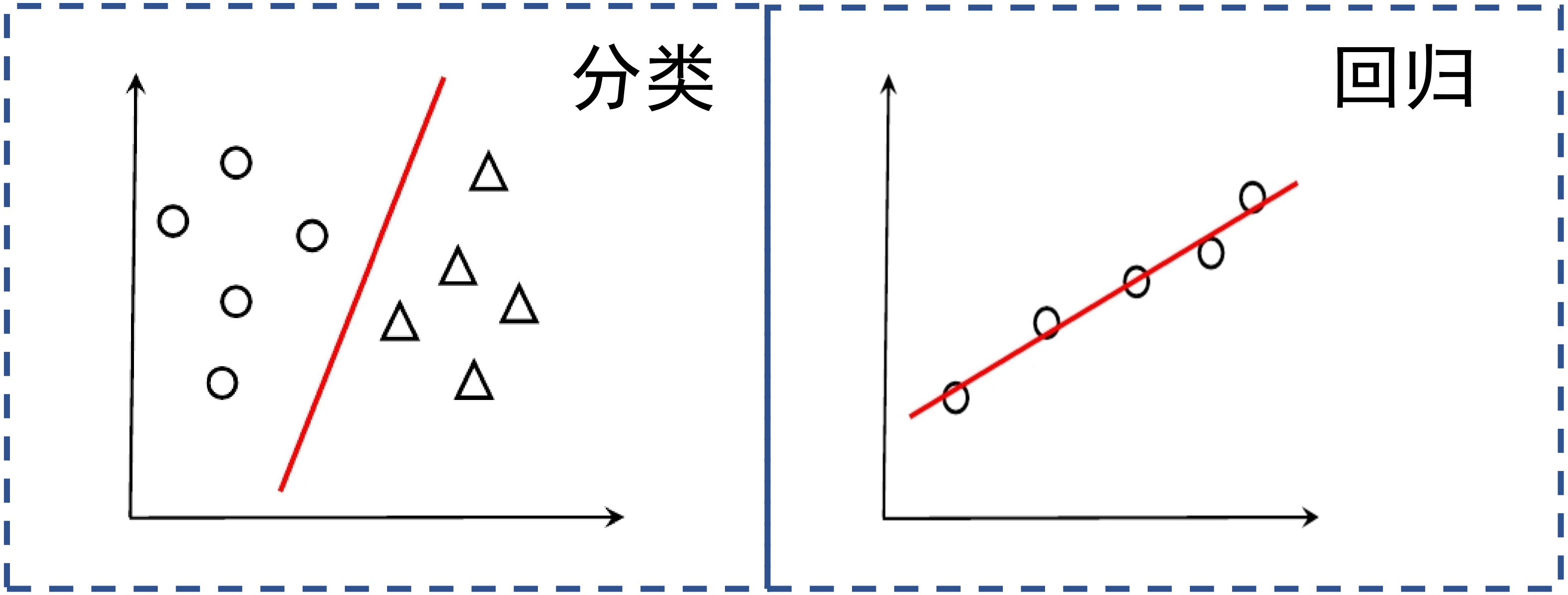


机器学习

[Mitchell, 1997] 对于某类任务T和性能度量P，如果一个计算机程序在T上以P衡量的性能随着经验E而自我完善，那么我们称这个计算机程序在从经验E中学习。

[周志华, “西瓜书”]通过计算的手段，利用经验（数据）来改善系统自身的性能

监督学习、无监督学习、半监督学习、强化学习



- 统计机器学习三要素：方法=模型+策略+算法
  - 模型：从假设空间中选取最优模型 $\mathcal{F} = \{f \mid Y = f(X)\}$
  - 策略：按照什么样的准则学习或选择最优模型
  - 算法：求解最优化问题找到全局最优解

2.2 线性回归(linear regression)

- 样本集 D = $\{(x_1,y_1),(x_2,y_2),\dots,(x_m,y_m)\}$，线性回归试图学得一个线性模型以尽可能地预测实值输出标记。

- 先考虑输入属性的数目只有1个的情况：

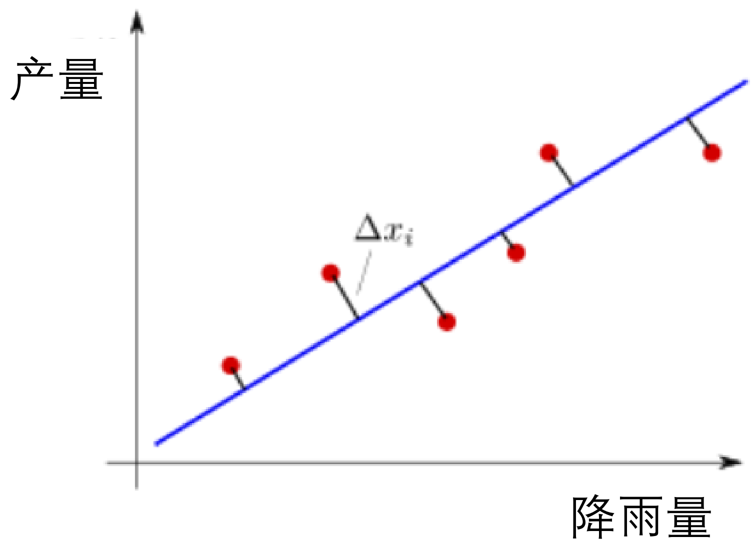
(忽略关于属性的下标，$D=\{(x_i,y_i)\}_{i=1}^m$)

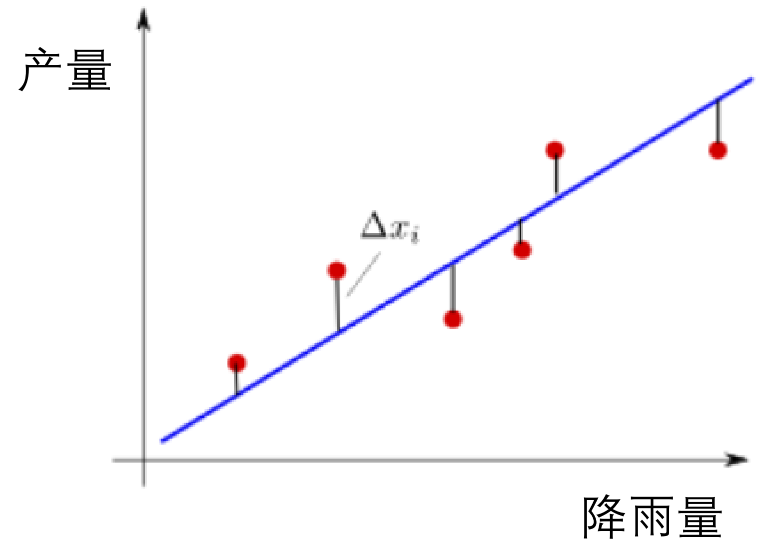
$$f(x_i)=wx_i+b, 使得\ f(x_i)\simeq y_i$$

如何求w和b呢？

2.2 线性回归

$f(x_i) = wx_i + b$, 使得 $f(x_i) \simeq y_i$





误差$= \sum_{i=1}^{6} (\Delta x_i)^2$

2.2 线性回归

$$f(x_i) = wx_i + b, \text{使得} \ f(x_i) \simeq y_i$$

- 令均方误差最小化，有

$$(w^*, b^*) = \underset{(w,b)}{\arg\min} \sum_{i=1}^{m} (y_i - wx_i - b)^2$$

均方误差有很好的几何意义，它对应着欧氏距离。在线性回归中，最小二乘法就是试图找到一条直线，使所有样本到直线上的欧氏距离之和最小。

2.2 线性回归

目标函数

- 对 $E_{(w,b)}=\sum_{i=1}^{m}(y_i-wx_i-b)^2$ 进行最小二乘估计，求w,b

1. 对w,b分别求导，得到：

$w^2x_i^2$　　$2w(y_i-b)x_i$

$$\frac{\partial E(w,b)}{\partial w}=2\left(w\sum_{i=1}^{m}x_i^2-\sum_{i=1}^{m}(y_i-b)x_i\right)\qquad (1)$$

$$\frac{\partial E(w,b)}{\partial b}=2\left(mb-\sum_{i=1}^{m}(y_i-wx_i)\right)\qquad (2)$$

2. 令(1)，(2)式为0，得到w和b的解：

$$w=\frac{\sum_{i=1}^{m}y_i(x_i-\bar{x})}{\sum_{i=1}^{m}x_i^2-\frac{1}{m}\left(\sum_{i=1}^{m}x_i\right)^2}\qquad b=\frac{1}{m}\sum_{i=1}^{m}(y_i-wx_i)$$

2.2 线性回归

- 多元线性回归(multivariable linear regression)
  - 一般的形式如数据集D，样本由d个属性描述($x_i \in \mathrm{R}^d$)，类似的，目标函数为：

$$E_{(w,b)} = \sum_{i=1}^{m}(y_i - w^{\boxed{T}} x_i - b)^2 = \sum_{i=1}^{m} |w^T x_i + b - y_i|^2$$

  - 将w和b吸收入向量形式：

$$X = \underbrace{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} & 1 \\ x_{21} & x_{22} & \cdots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{md} & 1 \end{pmatrix} \in R^{m \times (d+1)}}_{\text{(齐次坐标)}}, \underbrace{\hat{w} = \begin{pmatrix} w_1 \\ \vdots \\ w_d \\ b \end{pmatrix} \in R^{d+1}}_{\text{(参数)}}, \underbrace{y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \in R^m}_{\text{(回归目标)}}$$

2.2 线性回归

- **多元线性回归**

$$E(\hat{w})=(X\hat{w}-y)^T(X\hat{w}-y)=\hat{w}^T X^T X\hat{w}-2\hat{w}^T X^T y+y^T y$$

代价函数： $\hat{w}^* = \arg\min_{\hat{w}} E(\hat{w})$

- 求解—求偏导数：

$$\frac{\partial E(\hat{w})}{\partial \hat{w}} = \frac{\partial\left((X\hat{w}-y)^T(X\hat{w}-y)\right)}{\partial \hat{w}}$$

$$= \frac{\partial(\hat{w}^T X^T X\hat{w}-2\hat{w}^T X^T y+y^T y)}{\partial \hat{w}}$$

$$= 2X^T X\hat{w}-2X^T y$$

$$\Rightarrow \hat{w}^* = (X^T X)^{-1} X^T y$$

若$X^T X$不可逆怎么办？

2.2 线性回归

- **计算问题**
  - $X^T X$ 可能是不可逆的，此时，可以在$X^T X$的主对角线元素上加一个很小的正数λ，此时有：

$$\hat{w} = (X^T X + \lambda I)^{-1} X^T Y$$

  - 可以证明，“上式”即为如下结构风险最小化模型（正则化模型）的最优解：

$$\min_{\hat{w}} \| X\hat{w} - Y \|_F^2 + \lambda \| \hat{w} \|_F^2$$

2.2 线性回归

- **计算问题**

  结构风险最小化模型

$$\boxed{\min_{\hat{w}} \| X\hat{W} - Y \|_F^2 + \lambda \| \hat{W} \|_F^2}$$

  - 从贝叶斯决策的角度进行解释：

$$\max e^{-\left(\|X\hat{W}-Y\|_F^2+\lambda\|\hat{W}\|_F^2\right)} = e^{-\|X\hat{W}-Y\|_F^2} \times e^{-\lambda\|\hat{W}\|_F^2}$$

似然概率（误差分布）　　先验概率（参数分布）

2.2 线性回归

- **求解思路2**

$$\min_{w} \frac{1}{2}\|w\|_2^2 \;\; s.t. \;\; \mathrm{X}w = y$$

$$J(w) = \frac{1}{2} w^T w - \Lambda^T (Xw - y)$$

$$\partial J(w) \,/\, \partial w = w - X^T \Lambda = 0 \Rightarrow Xw - XX^T \Lambda = 0$$

$$\Rightarrow y - XX^T \Lambda = 0 \Rightarrow \Lambda = (XX^T)^{-1} y$$

$$w = X^T (XX^T)^{-1} y$$

2.2 线性回归

- **广义线性回归(generalized linear regression)**
  - 广义线性回归是为了克服线性回归模型的缺点出现的。
  - 线性回归虽然简单，但是输出无法确定。进行分类时效果也不理想。
  - 令预测值逼近y的衍生物，即**y**的函数？
  - 比如，输出标记是在对数尺度上的变化：

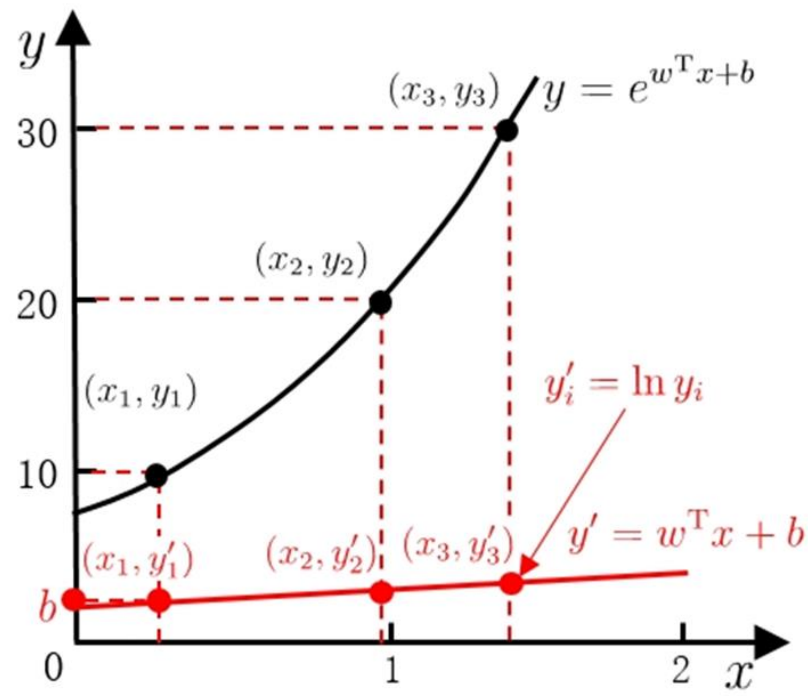
$$\ln y = w^T x + b$$

2.2 线性回归

- **广义线性回归**

$$\boxed{\ln y = w^T x + b}$$

  - 这就是对数线性回归。



  - 其形式上仍然是线性回归但其实质它是采用如下变换逼近y，因此是非线性的：

$$y = e^{w^T x + b}$$

中国科学院大学 University of Chinese Academy of Sciences

2.2 线性回归

- **广义线性回归**
  - 更一般地（广义线性回归），考虑单调可微函数$g(\cdot)$，将该函数作用于y(待回归的值)。

$$g(y) = (w^T x + b)$$

$$\Downarrow$$

$$y = g^{-1}(w^T x + b)$$

  - 这样得到的模型成为“广义线性模型”
  - $g(\cdot)$称为“联系函数”，对数线性回归是广义线性回归的特例

$$g(\cdot) = ln(\cdot) \Rightarrow \ln y = w^T x + b$$

2.3 对数几率回归（Logistic regression）

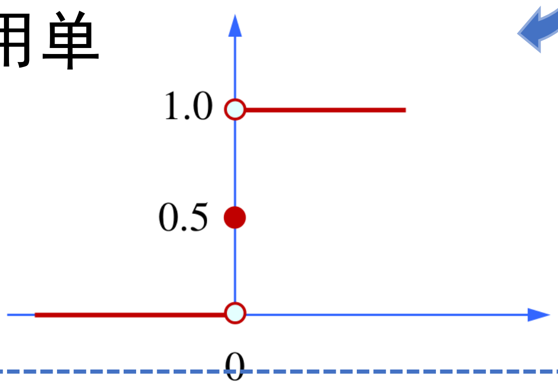
- 利用广义线性回归的思想，希望找到一个单调可微函数，利用该函数将分类任务的真实标记与线性回归模型的预测值联系起来。

线性回归模型产生的实值输出： $z = w^T x + b$

期望输出： $y \in (0,1)$

于是我们将实值z转换为0/1值。为此可采用单位阶跃函数

$$y = \begin{cases} 0, & z < 0; \\ 0.5, & z = 0; \\ 1, & z > 0, \end{cases}$$

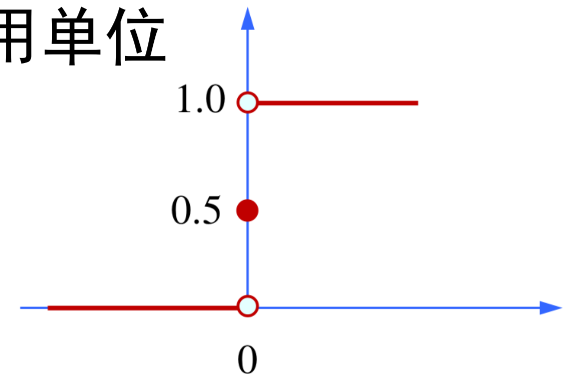


2.3 对数几率回归（Logistic regression）

- 利用广义线性回归的思想，希望找到一个单调可微函数，利用该函数将分类任务的真实标记与线性回归模型的预测值联系起来。

于是我们将实值z转换为0/1值。为此可采用单位阶跃函数：

$$y = \begin{cases} 0, & z < 0; \\ 0.5, & z = 0; \\ 1, & z > 0, \end{cases}$$



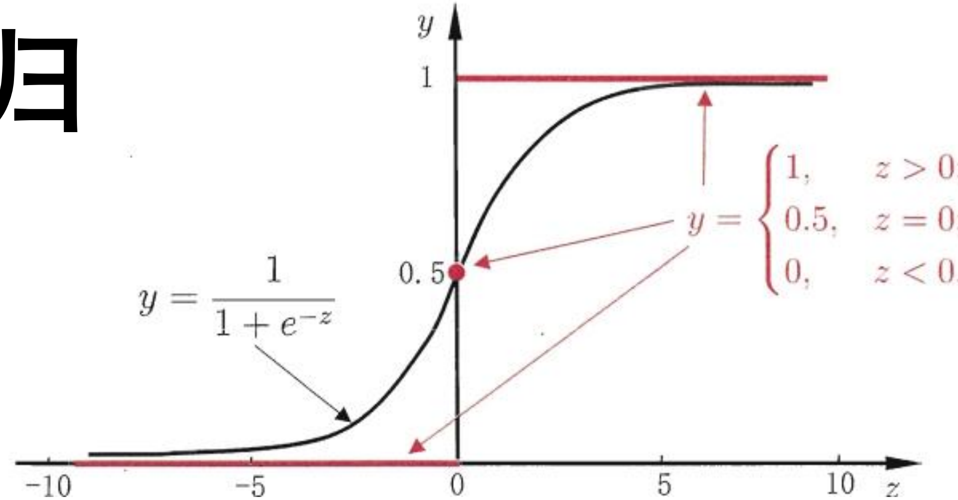
- 阶跃函数不连续，且反函数不存在。需要找到可近似单位阶跃函数的替换函数

对数几率函数！(logistic function)

2.3 对数几率回归

- 对数几率函数定义

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



– 以对率函数为联系函数：

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

联系函数

$$y=g^{-1}(w^T x+b)$$

即 $\ln \frac{y}{1-y} = w^T x + b$（本质是线性的）

几率(odds)：度量样本*x*作为正例的相对可能性。

- *y*视为样本*x*作为正例的可能性，*1-y*就是反例的可能性.

2.3 对数几率回归

- **优点**
  - 无需事先假设数据分布
  - 可得到“类别”的近似概率预测
  - 可直接应用现有数值优化算法求取最优解

- **分析**
  - 对数几率回归是一种二分类方法。
  - 对数几率回归（logistic regression），亦称logit regression），简称“对率回归”。
  - 输出样本分类结果的可能性（软标签）。

2.3 对数几率回归

- **求解思路**
  - 如果将y视为类后验概率估计$p(y=1|x)$，有：

$$\ln\frac{y}{1-y}=w^Tx+b \quad \Rightarrow \quad \ln\frac{p(y=1|x)}{p(y=0|x)}=w^Tx+b$$

  - 进一步，由变化可获得预测概率：

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

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

中国科学院大学 University of Chinese Academy of Sciences

2.3 对数几率回归

- **求解思路**
  - 通过“极大似然法”来估计w和b

1. 对率回归模型最大化“对数似然”：

$$l(w,b)=\sum_{i=1}^{m}\ln p(y_i|x_i;w,b) \quad (1)$$

2. 令 $\beta=(w;b),\hat{x}=(x;1)$ 则w^Tx+b可简写为$\beta^T\hat{x}$

3. 再令 $p_1=p(y=1|\hat{x};\beta)$,

$$p_0=p(y=0|\hat{x};\beta)=1-p_1(\hat{x};\beta)$$

4. 则(1)式中的似然项可写为

$$p(y_i|x_i;w,b)=y_ip_1(\hat{x}_i;\beta)+(1-y_i)p_0(\hat{x}_i;\beta) \quad (2)$$

2.3 对数几率回归

$w^T x + b$简写为$\beta^T \hat{x}$

– 通过"极大似然法"来估计w和b

$$l(w,b) = \sum_{i=1}^{m} \ln p(y_i | x_i; w, b) \quad (1)$$

$$p(y_i | x_i; w, b) = y_i p_1(\hat{x}_i; \beta) + (1 - y_i) p_0(\hat{x}_i; \beta) \quad (2)$$

(为了方便, 我们将$p_1(\hat{x}_i; \beta)$ 直接用p_1表示, $p_0(\hat{x}_i; \beta)$用p_0表示)

$$p_1 = \frac{e^{\beta^T \hat{x}_i}}{1 + e^{\beta^T \hat{x}_i}} \qquad p_0 = \frac{1}{1 + e^{\beta^T \hat{x}_i}}$$

5. 将(2)代入(1) 可得

$$l(\beta) = \sum_{i=1}^{m} \ln(y_i p_1 + (1 - y_i) p_0) \quad (3)$$

中国科学院大学 University of Chinese Academy of Sciences

2.3 对数几率回归

$w^T x + b$简写为$\beta^T \hat{x}$

- 通过“极大似然法”来估计w和b

$$l(\beta) = \sum_{i=1}^{m} \ln(y_i p_1 + (1 - y_i) p_0)$$

$$p_1 = \frac{e^{\beta^T \hat{x}_i}}{1 + e^{\beta^T \hat{x}_i}} \qquad p_0 = \frac{1}{1 + e^{\beta^T \hat{x}_i}}$$

6. 将p_1，p_0带入(3)式，得

$$l(\beta) = \sum_{i=1}^{m} \ln\left(\frac{y_i e^{\beta^T \hat{x}_i} + 1 - y_i}{1 + e^{\beta^T \hat{x}_i}}\right)$$

$$= \sum_{i=1}^{m} \left(\ln\left(y_i e^{\beta^T \hat{x}_i} + 1 - y_i\right) - \ln(1 + e^{\beta^T \hat{x}_i})\right)$$

2.3 对数几率回归

$w^Tx + b$简写为$\beta^T\hat{x}$

- 通过“极大似然法”来估计w和b

$$l(\beta) = \sum_{i=1}^{m} \left(\ln\left(y_i e^{\beta^T\hat{x}_i} + 1 - y_i\right) - \ln(1 + e^{\beta^T\hat{x}_i})\right)$$

7. 由于y_i=0 或1 则

$$l(\beta) = \begin{cases} \sum_{i=1}^{m} \left(-\ln(1 + e^{\beta^T\hat{x}_i})\right) & y_i = 0 \\ \sum_{i=1}^{m} \left(\beta^T\hat{x}_i - \ln(1 + e^{\beta^T\hat{x}_i})\right) & y_i = 1 \end{cases}$$

两式综合可得 $l(\beta) = \sum_{i=1}^{m} \left(y_i \beta^T\hat{x}_i - \ln(1 + e^{\beta^T\hat{x}_i})\right)$

2.3 对数几率回归

$w^T x + b$简写为$\beta^T \hat{x}$

- 通过“极大似然法”来估计w和b

$$l(\beta) = \sum_{i=1}^{m} \left(y_i \beta^T \hat{x}_i - \ln(1 + e^{\beta^T \hat{x}_i}) \right)$$

8.最大化似然函数等价于最小化似然函数的相反数

$$\max l(\beta) \Rightarrow \min(-l(\beta))$$

故 $l(\beta) = \sum_{i=1}^{m} (-y_i \beta^T \hat{x}_i + \ln(1 + e^{\beta^T x_i}))$

可采用梯度法或牛顿法进行求解。比如采用牛顿法

2.3 对数几率回归

- **求解思路**(牛顿法、梯度下降、随机梯度下降)
  - 用牛顿法求 $l(\beta)=\sum_{i=1}^{m}(-y_i\beta^T\hat{x}_i+\ln(1+e^{\beta^T x_i}))$ 最优解
  - 第t+1轮迭代解的更新公式为：

$$\boxed{\beta^{t+1}=\beta^{t}-\left(\frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T}\right)^{-1}\frac{\partial l(\beta)}{\partial\beta}}$$

  - 其中关于β的一阶，二阶导数分别为：

$$\frac{\partial l(\beta)}{\partial\beta}=-\sum_{i=1}^{m}\hat{x}_i(y_i-p_1(\hat{x}_i;\beta)),$$

$$\frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T}=\boxed{\sum_{i=1}^{m}\hat{x}_i\hat{x}_i^T p_1(\hat{x}_i;\beta)(1-p_1(\hat{x}_i;\beta))}$$

海赛矩阵

中国科学院大学 University of Chinese Academy of Sciences

• 小结

- Logistic 回归适用于二分类任务。
- 给定m个训练样本$(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$,其中 $x_i \in R^d, i = 1, \cdots, m$，为$d$维样本特征，$y_i \in \{0,1\}$ 其对应的类别标签。
- Logistic 回归采用的假设函数：

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

- 目标：训练模型参数(w,b), 最小化代价函数：
  - 采用最大似然估计：

$$l(w,b) = -\sum_{i=1}^{m} \left(y_i h(x_i \mid w,b) + (1 - y_i)\left(1 - h(x_i \mid w,b)\right) \right)$$

  - 也可采用交叉熵损失：

$$l(w,b) = -\sum_{i=1}^{m} \left(y_i \log\left(h(x_i \mid w,b)\right) + (1 - y_i) \log(1 - h(x_i \mid w,b)) \right)$$

中国科学院大学
University of Chinese Academy of Sciences

2.1 基本形式

2.2 线性回归

2.3 对数几率回归

2.4 Softmax回归

2.5 线性判别分析

2.6 局部线性判别分析

2.7 多分类学习

2.8 类别不平衡问题

2.4 Softmax回归

- **问题的背景**
  - Logistic回归利用后验概率最大化去计算权重w，但它不方便处理多类分类问题。
  - Softmax Regression是Logistic回归的推广，能以更加紧凑的方式来处理Logistic回归中所面临的多类分类问题。
  - 也就是说，Softmax适用于解决$y \in \{1, \cdots, k\}$分类的问题。

University of Chinese Academy of Sciences

2.4 Softmax回归

- **Softmax函数**
  - 设$z = [z_1, z_2, \cdots, z_c]^T$为一个c维空间中的一个向量，Softmax函数$\sigma$是一个[0, 1]上的一个$c$维映射函数：

$$[\sigma(z)_j] = \frac{e^{z_j}}{\sum_{i=1}^{c} e^{z_i}}, \qquad j = 1,2,\cdots,c$$

  - Softmax函数的输出可以用来描述关于类别的分布：

$$P(y = j \mid x) = \frac{e^{w_j^T x + b_j}}{\sum_{i=1}^{c} e^{w_i^T x + b_i}}$$

中国科学院大学 University of Chinese Academy of Sciences

2.4 Softmax回归

$$[\sigma(z)_j] = \frac{e^{z_j}}{\sum_{i=1}^{c} e^{z_i}}$$

- **Softmax函数**

C_1: w^1, b_1 $\quad z_1 = w^1 \cdot x + b_1$

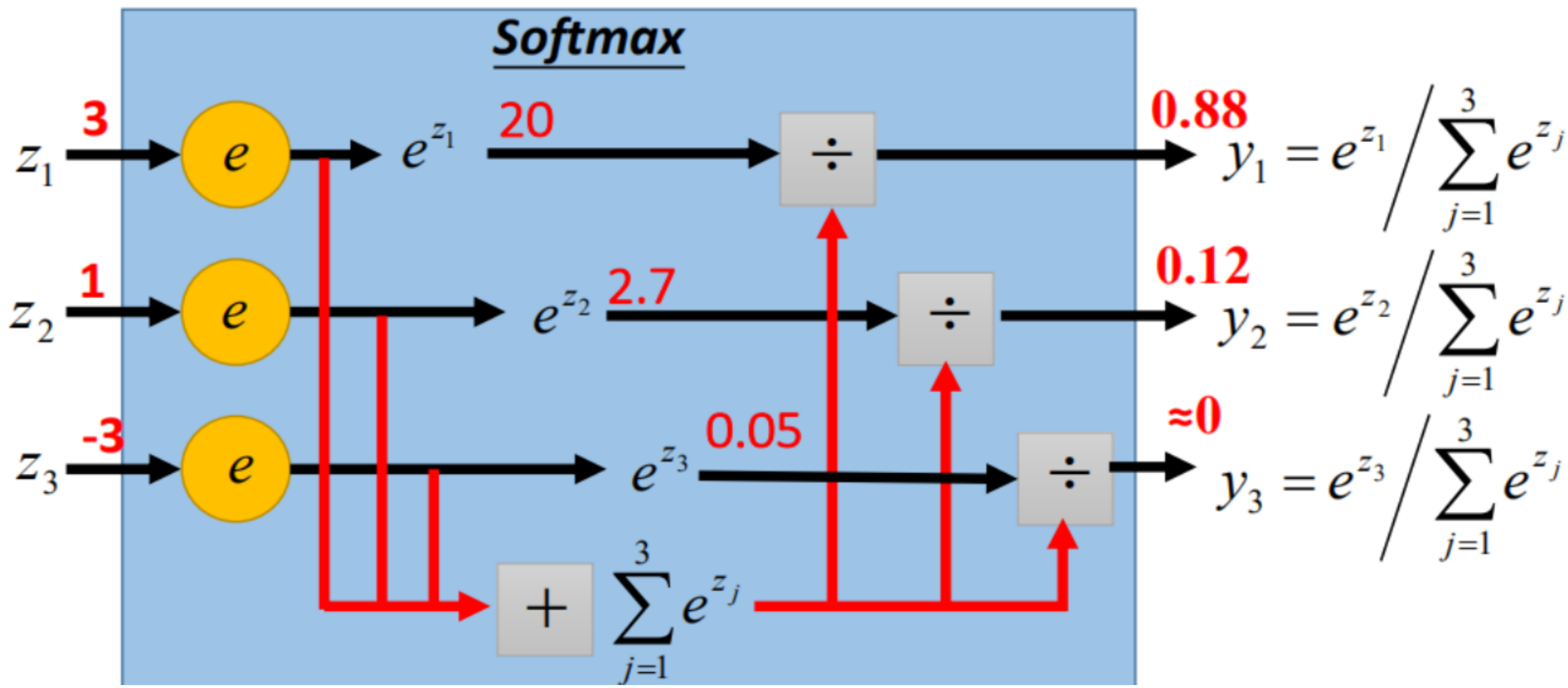
C_2: w^2, b_2 $\quad z_2 = w^2 \cdot x + b_2$

C_3: w^3, b_3 $\quad z_3 = w^3 \cdot x + b_3$

***Probability*:**

- $1 > y_i > 0$
- $\sum_i y_i = 1$

$$y_i = P(C_i \mid x)$$



图片来源:https://datawhalechina.github.io/leemlnotes/#/chapter11/chapter11?id=softmax

2.4 Softmax回归

- **Softmax函数**
  - 假定数据x是采用齐次坐标表示的，即x的维数是$d+1$维。齐次坐标则对应着平移量b。
  - 假设函数的具体形式（概率形式）：

$$h(x;W)=\begin{pmatrix} P(y=1\mid x;W) \\ P(y=2\mid x;W) \\ \vdots \\ P(y=c\mid x;W) \end{pmatrix} = \frac{1}{\sum_{i=1}^{c} e^{w_i^T x}} \begin{pmatrix} e^{w_1^T x} \\ e^{w_2^T x} \\ \vdots \\ e^{w_c^T x} \end{pmatrix} \in R^c$$

其中，$W=[w_1,w_2,\cdots,w_c]\in R^{(d+1)\times c}$

  - 假设函数会对每一个$i=1,\ldots,c$,给出$p(y=i\mid x;W)$概率的估计值

中国科学院大学 University of Chinese Academy of Sciences

2.4 Softmax回归

- **代价函数：从两类到多类**

采用交叉熵损失

两类

$$l(w,b) = -\sum_{i=1}^{m}\Big(y_i \log\big(h(x_i \mid w,b)\big) + (1-y_i)\log(1-h(x_i \mid w,b))\Big)$$

$$= -\sum_{i=1}^{m}\left[\sum_{j=0}^{1}\delta(y_i = j)\log P(y_i = j \mid x_i; w,b)\right]$$

$\delta(\cdot)$为指示函数

多类

$$l(W) = -\sum_{i=1}^{m}\left[\sum_{j=1}^{c}\delta(y_i = j)\log\frac{e^{w_j^T x_i}}{\sum_{k=1}^{c} e^{w_k^T x_i}}\right]$$

$\delta(\cdot)$为指示函数

$\delta(\cdot)$：如果参数为真，则等于1，反之则为0

2.4 Softmax回归

- **求解**

$$l(W) = -\sum_{i=1}^{m}\left[\sum_{j=1}^{c}\delta(y_i = j)\log\frac{e^{w_j^T x_i}}{\sum_{k=1}^{c}e^{w_k^T x_i}}\right]$$

$$\frac{\partial l(W)}{\partial w_j} = -\sum_{i=1}^{m}x_i\left(\delta(y_i = j) - \frac{e^{w_j^T x_i}}{\sum_{k=1}^{c}e^{w_k^T x_i}}\right) = -\left(\sum_{i=1}^{m}x_i\left(\delta(y_i = j) - P(y_i = j \mid x; W)\right)\right)$$

$$j = 1,2,\cdots,c$$

2.4 Softmax回归

- 求解

$$l(W) = -\sum_{i=1}^{m}\left[\sum_{j=1}^{c}\delta(y_i = j)\log\frac{e^{w_j^T x_i}}{\sum_{k=1}^{c} e^{w_k^T x_i}}\right]$$

$$\frac{\partial l(W)}{\partial w_j} = -\sum_{i=1}^{m} x_i\left(\delta(y_i = j) - \frac{e^{w_j^T x_i}}{\sum_{k=1}^{c} e^{w_k^T x_i}}\right) = -\left(\sum_{i=1}^{m} x_i\left(\delta(y_i = j) - P(y_i = j \mid x; W)\right)\right)$$

$$j = 1, 2, \cdots, c$$

$$\delta(y_i = j)\log e^{w_j^T x_i} = \delta(y_i = j) w_j^T x_i$$

中国科学院大学 University of Chinese Academy of Sciences

2.4 Softmax回归

- 求解

$$l(W) = -\sum_{i=1}^{m}\left[\sum_{j=1}^{c}\delta(y_i = j)\log\frac{e^{w_j^T x_i}}{\sum_{k=1}^{c} e^{w_k^T x_i}}\right]$$

$$\frac{\partial l(W)}{\partial w_j} = -\sum_{i=1}^{m} x_i\left(\delta(y_i = j) - \frac{e^{w_j^T x_i}}{\sum_{k=1}^{c} e^{w_k^T x_i}}\right) = -\left(\sum_{i=1}^{m} x_i\left(\delta(y_i = j) - P(y_i = j \mid x; W)\right)\right)$$

$$j = 1, 2, \cdots, c$$

$$\delta(y_i = j)\log\frac{1}{\sum_{k=1}^{c} e^{w_k^T x_i}} = -\delta(y_i = j)\log\sum_{k=1}^{c} e^{w_k^T x_i}$$

中国科学院大学
University of Chinese Academy of Sciences

2.4 Softmax回归

- **求解**

$$l(W) = -\sum_{i=1}^{m}\left[\sum_{j=1}^{c}\delta(y_i = j)\log\frac{e^{w_j^T x_i}}{\sum_{k=1}^{c} e^{w_k^T x_i}}\right]$$

$$\frac{\partial l(W)}{\partial w_j} = -\sum_{i=1}^{m} x_i\left(\delta(y_i = j) - \frac{e^{w_j^T x_i}}{\sum_{k=1}^{c} e^{w_k^T x_i}}\right) = -\left(\sum_{i=1}^{m} x_i\left(\delta(y_i = j) - P(y_i = j \mid x; W)\right)\right)$$

$$j = 1,2,\cdots,c$$

采用梯度下降法：$w_j^{t+1} = w_j^t - \eta\frac{\partial l(W)}{\partial w_j^t}, j = 1,2,\cdots,c$

2.4 Softmax回归

- **参数化的特点**

对任意$\theta \in R^{d+1}$:

$$P(y=j\,|\,x;W)=\frac{e^{(w_j-\theta)^T x_i}}{\sum_{k=1}^{c} e^{(w_k-\theta)^T x_i}}=\frac{e^{w_j^T x_i}e^{-\theta^T x_i}}{\sum_{k=1}^{c} e^{w_k^T x_i}e^{-\theta^T x_i}}=\frac{e^{w_j^T x_i}}{\sum_{k=1}^{c} e^{w_k^T x_i}}$$

– 这表明softmax回归中的参数是“冗余”的。即Softmax模型被过度参数化。“对于任意一个用于拟合数据的假设函数，可以求出多组参数值，这些参数得到的是结果完全相同的假设函数$h(x;W)$”。

2.4 Softmax回归

- **参数化的特点**
  - 因此，使$l(W)$最小化求得的参数并不是唯一的。
  - 海塞矩阵通常是奇异的/不可逆的，因此采用牛顿法优化会遇到数值计算的问题
  - 注意，当$\theta = w_1$时，总可以将w_1 替换为$w_1 - \theta$，并且这种变换不会影响假设函数
    - 因此，可以去掉参数w_1（或者其他任意一个），但不影响假设函数的表达能力
    - 可以令$w_1 = 0$，只优化剩余的$c-1$个参数。
    - 但是，实际中我们很少这样做！

2.4 Softmax回归

- **权重衰减:新的学习模型（结构风险最小化）**

$$l(W) = -\sum_{i=1}^{m}\left[\sum_{j=1}^{c}\delta(y_i = j)\log\frac{e^{w_j^T x_i}}{\sum_{k=1}^{c} e^{w_k^T x_i}}\right] + \lambda \| W \|_F^2$$

$$\frac{\partial l(w)}{\partial w_j} = -\left(\sum_{i=1}^{m} x_i \left(\delta(y_i = j) - P(y_i = j \mid x;W)\right)\right) + 2\lambda w_j$$

引入权重衰减项（正则项）后，代价函数就变成了严格的凸函数，可保证得到唯一的解。此时的海塞矩阵变为可逆矩阵，并且因为是凸函数，梯度下降法等算法可以保证收敛到全局最优解。

2.4 Softmax回归

- **Softmax regression VS 多个Logistic regression**
  - 将图像分到三个不同类别中。
    - 类型一：假设这三个类别分别是：室内场景、户外城区场景、户外荒野场景。
    - 类型二：假设这三个类别分别是：室内场景、黑白图像、包含人物的图像。

    - 在第一个例子中，三个类别是互斥的，因此更适于选择softmax回归分类器。
    - 在第二个例子中，建立三个独立的logistic回归分类器可能更加合适。

2.1 基本形式

2.2 线性回归

2.3 对数几率回归

2.4 Softmax回归

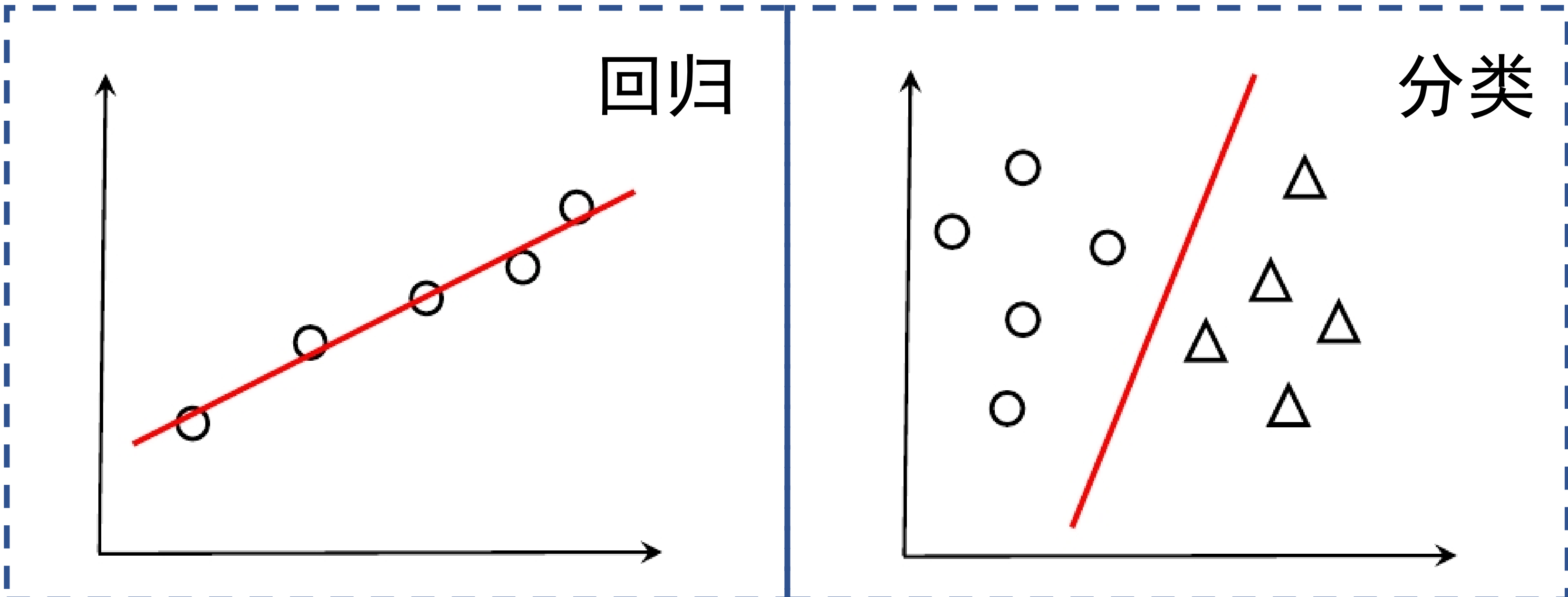
2.5 线性判别分析

2.6 局部线性判别分析

2.7 多分类学习

2.8 类别不平衡问题

2.5 线性判别分析(LDA)



广义线性模型通过“联系函数”

如何“直接”做分类？

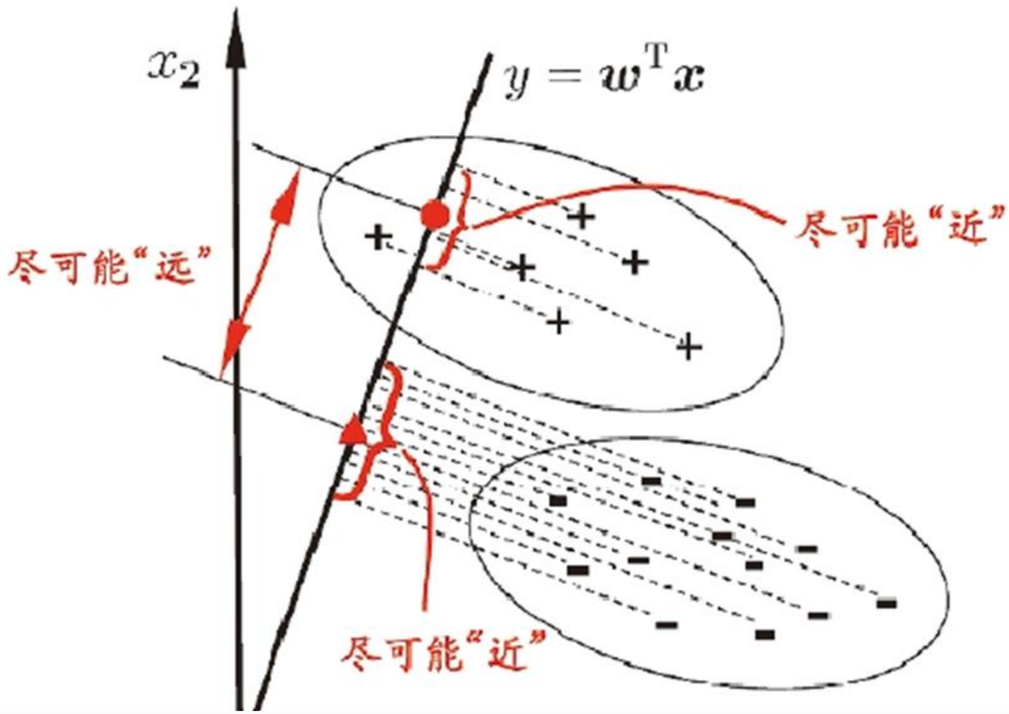
例如：对数几率回归

2.5 线性判别分析(LDA)

- **算法思想**
  - 线性判别分析（Linear Discriminant Analysis, LDA）是一种经典的线性学习方法。
  - LAD的思想较直观：对于二类分类问题，给定训练集，设法将样例投影到一条直线上，使得同类样例的投影点尽可能接近，不同类样例的投影点尽可能相互远离。
  - 在对新样本进行分类时，奖其投影到这条直线上，再根据投影点的位置来判断其类别。

2.5 线性判别分析(LDA)

- **算法思想**



– 由 … 因此
也 …

2.5 线性判别分析(LDA)

- **算法思想**
  - 给定数据集 $\{(x_i, y_i)\}_{i=1}^{m}$，其 i 类示例的集和--X_i，均值向量--μ_i,协方差矩阵 --$\sum_i$
  - 两类样本的中心在直线上的投影：$w^T\mu_0$ 和 $w^T\mu_1$
    两类样本的协方差：$w^T\sum_0 w$ 和 $w^T\sum_1 w$

同类样例的投影点尽可能接近：

让 $w^T\sum_0 w + w^T\sum_1 w$ 尽可能小

异类样例的投影点尽可能远离：

让 $\|w^T\mu_0 - w^T\mu_1\|^2$ 尽可能大

中国科学院大学 University of Chinese Academy of Sciences

2.5 线性判别分析(LDA)

- **算法思想**

同类样例的投影点尽可能接近 → $w^T\sum_0 w + w^T\sum_1 w$ 尽可能小

异类样例的投影点尽可能远离 → $\|w^T\mu_0\text{-}w^T\mu_1\|_2^2$ 尽可能大

最大化

目标函数：$$J = \frac{\|w^T\mu_0 - w^T\mu_1\|_2^2}{w^T\sum_0 w + w^T\sum_1 w}$$

$$= \frac{w^T(\mu_0-\mu_1)(\mu_0-\mu_1)^T w}{w^T(\sum_0+\sum_1)w}$$

两个类的中心尽可能远

两类的类内协方差尽可能小

2.5 线性判别分析(LDA)

- **LDA (Linear Discriminant Analysis)**
  - 类内散度矩阵(within-class scatter matrix)：

$$S_w = \Sigma_0 + \Sigma_1 = \sum_{x \in X_0} (x-\mu_0)(x-\mu_0)^T + \sum_{x \in X_1} (x-\mu_1)(x-\mu_1)^T$$

  - 类间散度矩阵(between-class scatter matrix)：

$$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$$

- 目标函数重写为： $J(w) = \dfrac{w^T S_b w}{w^T S_w w}$ （最大化广义Rayleigh商）

注意: $J(w)$ 的值与向量的长度无关，只与其方向有关。一般可令w为单位长度的向量。

2.5 线性判别分析(LDA)

$$J(w)=\frac{w^T S_b w}{w^T S_w w}$$

- **LDA算法**
  - 求解思路：(求使J最大的w)
  - 1. 令 $w^T S_w w=1$，最大化广义瑞利商等价形式为：

$$\min_{w} -w^T S_b w \quad s.t. \quad w^T S_w w=1$$

  - 2. 运用拉格朗日乘子法，得拉格朗日函数为：

$$L(w,\lambda)=-w^T S_b w+\lambda(w^T S_w w-1)$$

  - 3. 对w求偏导：$\dfrac{\partial L(w,\lambda)}{\partial \lambda}=-(S_b+S_b^T)w+\lambda(S_w+S_w^T)w$

2.5 线性判别分析(LDA)

- **LDA算法**
  - 求解思路:(求使J最大的w)

$$\boxed{\frac{\partial L(w,\lambda)}{\partial \lambda} = -(S_b + S_b^T)w + \lambda(S_w + S_w^T)w}$$

  - 4. 由于 $S_b = S_b^T, S_w = S_w^T$，

    所以: $$\frac{\partial L(w,\lambda)}{\partial w} = -2S_b w + 2\lambda S_w w$$

  - 5. 令上式等于0即可得:

$$S_b w = \lambda S_w w \Rightarrow S_w^{-1} S_b w = \lambda w$$

w为$S_w^{-1}S_b$的特征向量

中国科学院大学 University of Chinese Academy of Sciences

2.5 线性判别分析(LDA)

- **多类LDA算法**

  - 假定有N个类，且第i类示例数为m：

  - 全局散度矩阵：$S_t = S_b + S_w = \sum_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T, \mu = \frac{1}{m}\sum_{i=1}^{m} x_i$

  - 类内散度矩阵：$S_w = \sum_{i=1}^{N} S_{w_i}, S_{w_i} = \sum_{x \in X_i}(x - \mu_i)(x - \mu_i)^T, \mu_i = \frac{1}{n_i}\sum_{x \in X_i} x$

  - 类间散度矩阵：$S_b = S_t - S_w = \sum_{i=1}^{N} m_i(\mu_i - \mu)(\mu_i - \mu)^T$

多分类LDA有多种实现方法：采用S_b, S_w, S_t中的任意两个

2.5 线性判别分析(LDA)

$$
\begin{aligned}
S_B &= \frac{1}{2}\sum_{i=1}^{C}\sum_{j=1}^{C} p(i)p(j)\underline{(\mu_i-\mu_j)(\mu_i-\mu_j)}^T \\
&= \frac{1}{2}\sum_{i=1}^{C}\sum_{j=1}^{C} p(i)p(j)(\mu_i\mu_i^T-\mu_i\mu_j^T-\mu_j\mu_i^T+\mu_j\mu_j^T) \\
&= \frac{1}{2}\sum_{i=1}^{C} p(i)\mu_i\mu_i^T\sum_{j=1}^{C}p(j)-p(i)\mu_i\sum_{j=1}^{C}p(j)\mu_j^T-p(i)(\sum_{j=1}^{C}p(j)\mu_j)\mu_i^T-p(i)\sum_{j=1}^{C}p(j)\mu_j\mu_j^T \\
&= \frac{1}{2}\sum_{i=1}^{C} p(i)\mu_i\mu_i^T-p(i)\mu_i\mu^T-p(i)\mu\mu_i^T-p(i)\sum_{j=1}^{C}p(j)\mu_j\mu_j^T \\
&= \frac{1}{2}\sum_{i=1}^{C} p(i)\mu_i\mu_i^T-\mu\mu^T-\mu\mu^T-\sum_{j=1}^{C}p(j)\mu_j\mu_j^T \\
&= \sum_{i=1}^{C} p(i)(\mu_i\mu_i^T-\mu\mu^T) \\
&= \sum_{i=1}^{C} p(i)(\mu_i\mu_i^T-\mu\mu^T)+2\sum_{i=1}^{C}p(i)\mu\mu^T-\sum_{i=1}^{C}p(i)\mu_i\mu^T-\sum_{i=1}^{C}p(i)\mu\mu_i^T \\
&= \sum_{i=1}^{C} p(i)(\mu_i\mu_i^T-\mu_i\mu^T-\mu\mu_i^T+\mu\mu^T) \\
&= \sum_{i=1}^{C} \underline{p(i)(\mu_i-\mu)(\mu_i-\mu)}^T
\end{aligned}
$$

2.5 线性判别分析(LDA)

- **多类LDA算法**
  - 常见的优化目标为：

最大化投影后的距离

Problem 1：
(迹比值最大化)

$$\max_{w} \frac{tr(W^T S_b W)}{tr(W^T S_w W)} \Rightarrow S_b w = \lambda S_w w \quad s.t. \quad W^T W = I$$

  - Problem 1的求解较复杂，可以参考如下文献：

Shiming Xiang , Feiping Nie , Changshui Zhang. Learning a Mahalanobis distance metric for data clustering and classification . Pattern Recognition , 41(12) , Pages 3600 – 3612 , 2008

2.5 线性判别分析(LDA)

- **多类LDA算法** $tr(A)=\sum_t A_{tt} \rightarrow \sum_t w_t^T S_b w_t = tr(W^T S_b W)$
  - 常见的的优化目标为：

Problem 1：
（迹比值最大化）

$$\max_w \frac{tr(W^T S_b W)}{tr(W^T S_w W)} \Rightarrow S_b w = \lambda S_w w$$

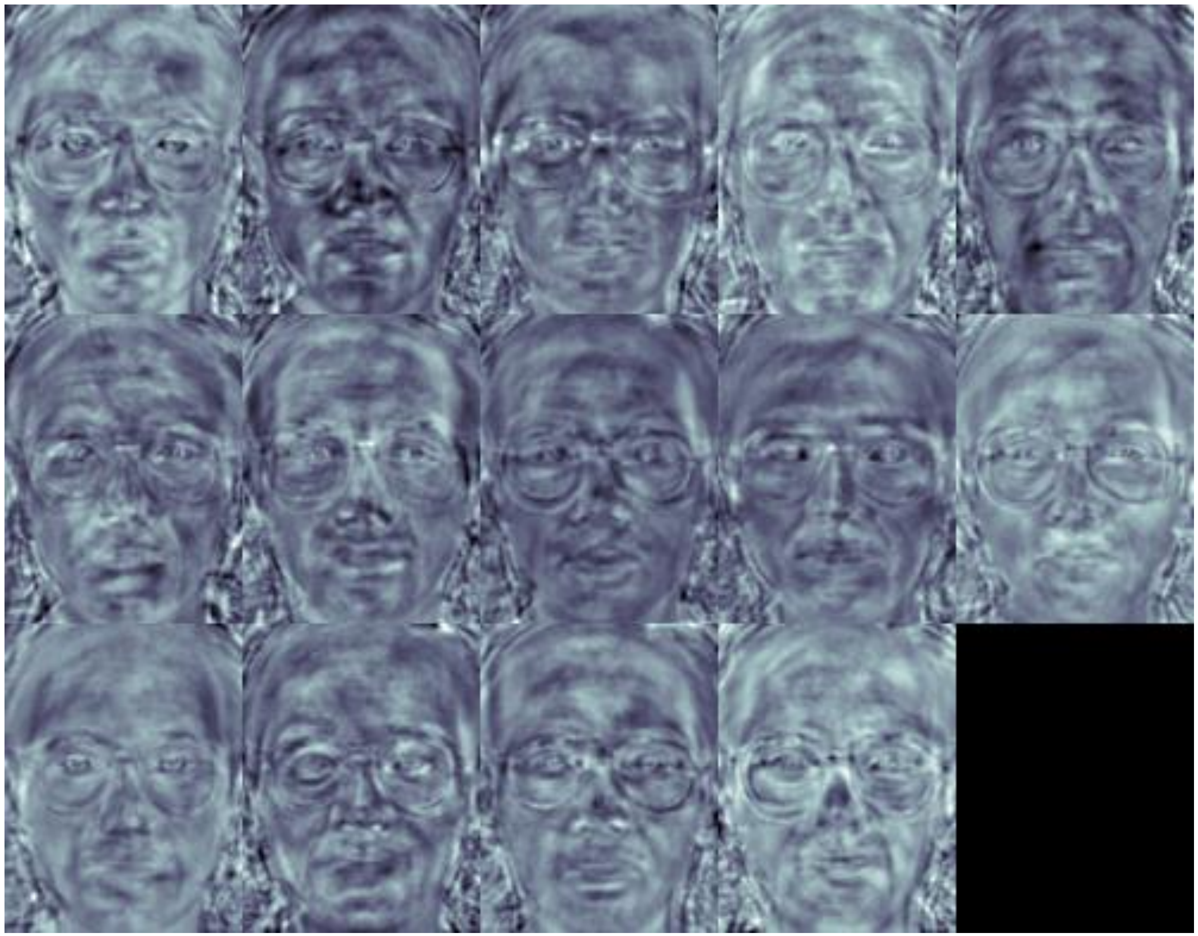
$$s.t. \quad W^T W = I$$

  - Problem 1的求解较复杂，可以参考如下文献：

Shiming Xiang , Feiping Nie , Changshui Zhang. Learning a Mahalanobis distance metric for data clustering and classification . Pattern Recognition , 41(12) , Pages 3600 – 3612 , 2008

2.5 线性判别分析(LDA)

- **特征向量的物理意义Fisher Faces**



https://blog.csdn.net/smartempire/article/details/23377385

2.1 基本形式

2.2 线性回归

2.3 对数几率回归

2.4 Softmax回归

2.5 线性判别分析

2.6 局部线性判别分析

2.7 多分类学习

2.8 类别不平衡问题

2.6 局部线性判别分析

- **The re-computation of S_w and S_b**

$$S_w = \sum_{i=1}^{c} \sum_{j:y_j=i} (x_j - \mu_i)(x_j - \mu_i)^T$$

$$= \frac{1}{2} \sum_{i,j} A_{ij}^{(w)} (x_i - x_j)(x_i - x_j)^T$$

$$S_b = \sum_{i=1}^{c} n_i (\mu_i - \mu)^T (\mu_i - \mu)$$

$$= \frac{1}{2} \sum_{i,j=1} A_{ij}^{(b)} (x_i - x_j)(x_i - x_j)^T$$

$$\mu_i = \frac{1}{n_i} \sum_{j:y_j=i} x_j$$

$$\mu = \frac{1}{n} \sum_{i=1}^{m} x_j$$

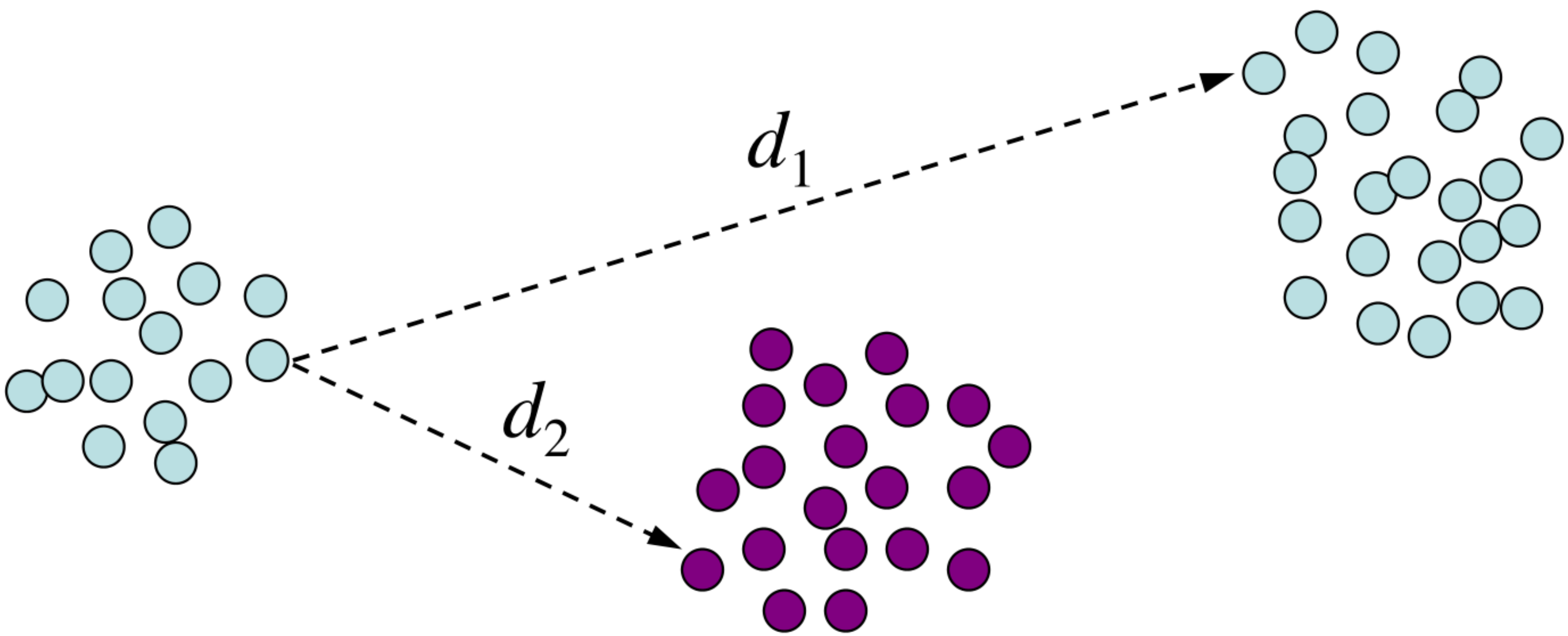
$$A_{ij}^{(w)} = \begin{cases} \frac{1}{n_k}, if \;\; y_i = y_j = k \\ 0, \qquad if \; y_i \neq y_j \end{cases}$$

$$A_{ij}^{(b)} = \begin{cases} \frac{1}{n} - \frac{1}{n_k}, if \;\; y_i = y_j = k \\ \frac{1}{n}, \qquad if \; y_i \neq y_j \end{cases}$$

下标y_i表示样本x_i的类别标签，即$y_i \in \{1,2,\ldots c\}$。另外，$k \in \{1,2,\ldots c\}$

2.6 局部线性判别分析

- **局限性**
  - In each class , the distribution of data is Gaussian

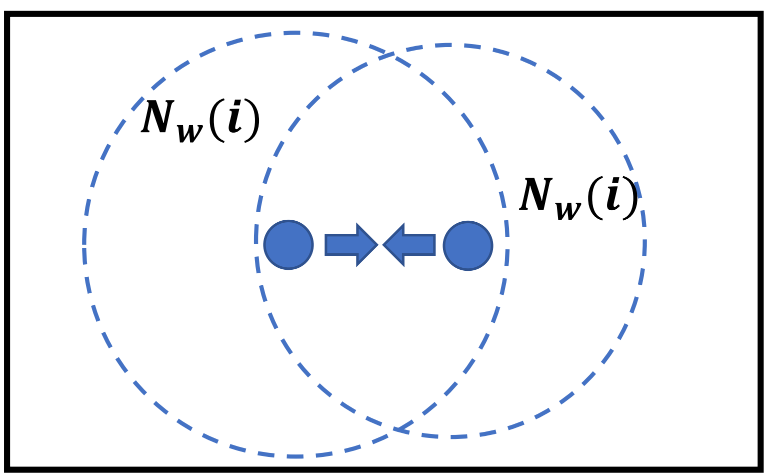


We hope $d_1 < d_2$. But difficult, or impossible!

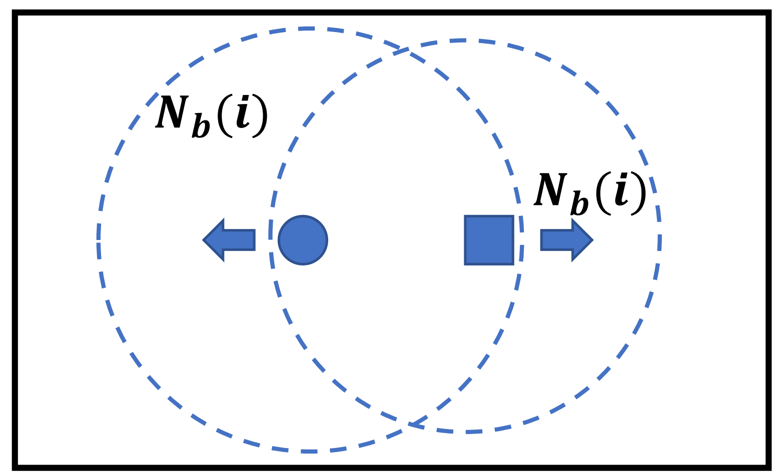
2.6 局部线性判别分析

- **Techniques of Local Analysis**
  - Neighborhood constraints（方法一）
  - Locally weighting（方法二）
    - Weighting for 1-NN
    - Local Fisher discriminant analysis

Motivation



Within class



Between class

2.6 局部线性判别分析

- **Modify S_w and S_b**
  - Neighborhood Constraints：

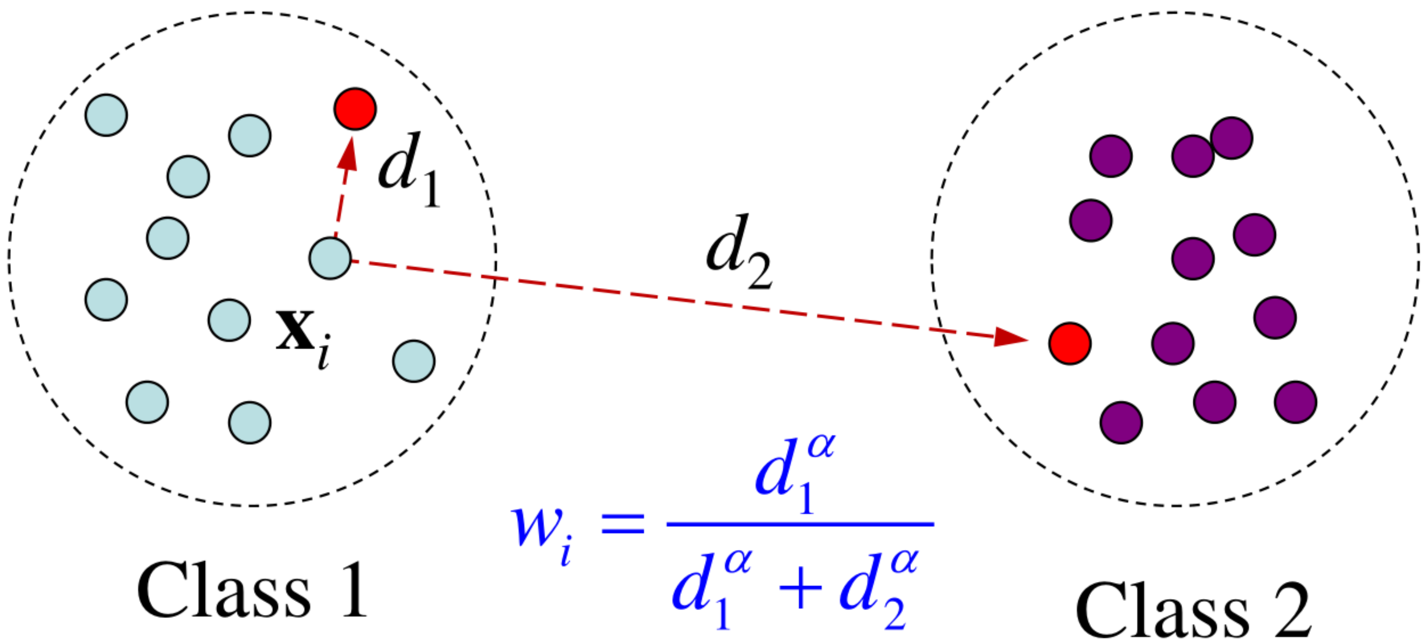
$$S_w = \sum_{\substack{y_i = y_j \\ x_i \in N(x_j), x_j \in N(x_i)}} (x_i - x_j)(x_i - x_j)^T$$

$$S_b = \sum_{\substack{y_i \neq y_j \\ x_i \in N(x_j), x_j \in N(x_i)}} (x_i - x_j)(x_i - x_j)^T$$

University of Chinese Academy of Sciences

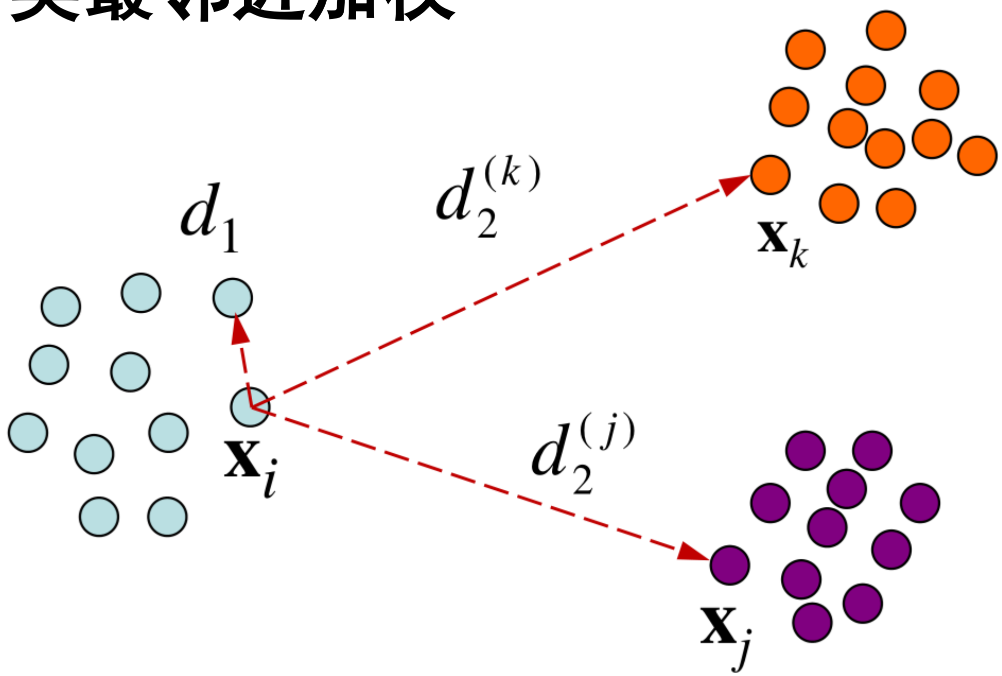
2.6 局部线性判别分析

- **Nearest Neighbor Discriminant Analysis, NNDA**
  - **最邻近加权**
    - A problem in neighborhood constraints is the selection of the number of nearest neighbors (k)



2.6 局部线性判别分析

- **Nearest Neighbor Discriminant Analysis, NNDA**
  - **多类最邻近加权**



$$w_i = \frac{d_1^{\alpha}}{d_1^{\alpha} + (\min\{d_2^{(j)}\})^{\alpha}}, \quad (0 < \alpha < 1)$$

2.6 局部线性判别分析

- **Nearest Neighbor Discriminant Analysis, NNDA**

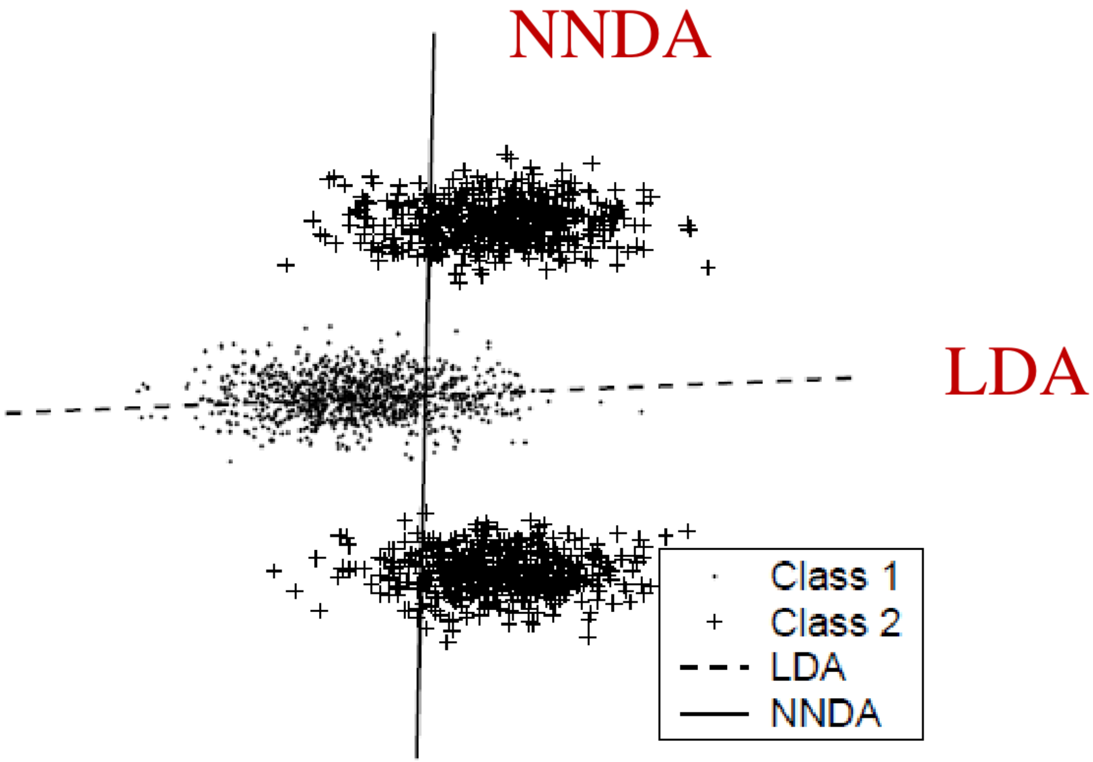
$$S_w = \sum_{\substack{y_i = y_j \\ x_j \in N_{1-nm}(x_i), i=1,2,\cdots,n}} w_i (x_i - x_j)(x_i - x_j)^T$$

$$S_b = \sum_{\substack{y_i \neq y_j \\ x_j \in N_{1-nm}(x_i), i=1,2,\cdots,n}} w_i (x_i - x_j)(x_i - x_j)^T$$

Xipeng Qiu, Lide Wu: Stepwise Nearest Neighbor Discriminant Analysis . IJCAI 2005: 829-834

2.6 局部线性判别分析

- **Nearest Neighbor Discriminant Analysis, NNDA**



NNDA finds the correct projection direction, but LDA failed !

2.6 局部线性判别分析

- **Local Fisher Discriminant Analysis, LFDA**
  - Motivation
    - LFDA does not impose far-apart data pairs of the same class to be close, by which local structure of the data tends to be preserved.
    - 邻域加权 (Locally Weighting)

Masashi Sugiyama , Local Fisher Discriminant Analysis for Supervised Dimensionality Reduction, ICML, 2006

University of Chinese Academy of Sciences

2.6 局部线性判别分析

- **Step1:Construct an affine matrix for n data points:**

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{pmatrix}$$

$$A_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_j \text{ is a neighbor of } \mathbf{x}_i \\ 0, & \text{otherwise} \end{cases}, \text{ or}$$

$$A_{ij} = \begin{cases} \exp(-\|\mathbf{x}_j - \mathbf{x}_i\|^2 / (2\sigma^2)), & \text{if } \mathbf{x}_j \text{ is a neighbor of } \mathbf{x}_i \\ 0, & \text{otherwise} \end{cases}$$

2.6 局部线性判别分析

- **Step2: Modify S_w and S_b :**

$$S_w = \frac{1}{2}\sum_{i,j} A_{ij}^{(w)}(x_i - x_j)(x_i - x_j)^T$$

$$S_b = \frac{1}{2}\sum_{i,j=1} A_{ij}^{(b)}(x_i - x_j)(x_i - x_j)^T$$

$$S_w = \frac{1}{2}\sum_{i,j} \overline{A}_{ij}^{(w)}(x_i - x_j)(x_i - x_j)^T$$

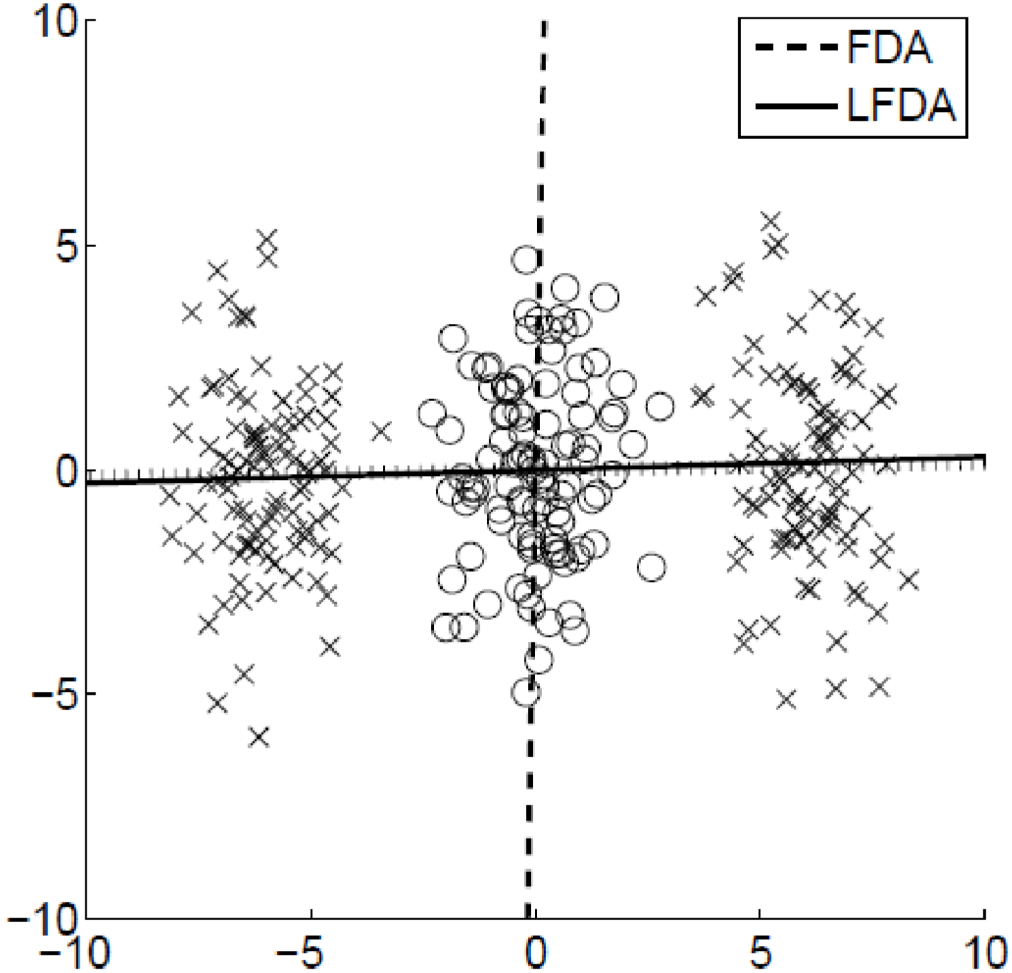
$$S_b = \frac{1}{2}\sum_{i,j=1} \overline{A}_{ij}^{(b)}(x_i - x_j)(x_i - x_j)^T$$

$$\overline{A}_{ij}^{(w)} = \begin{cases} A_{ij}/n_c\,, & if\ y_i = y_j = c \\ 0, & if\ y_i \neq y_j \end{cases}$$

$$\overline{A}_{ij}^{(b)} = \begin{cases} A_{ij}\left(\frac{1}{n} - \frac{1}{n_c}\right), & if\ y_i = y_j = c \\ \frac{1}{n}, & if\ y_i \neq y_j \end{cases}$$

2.6 局部线性判别分析

- **Demo**

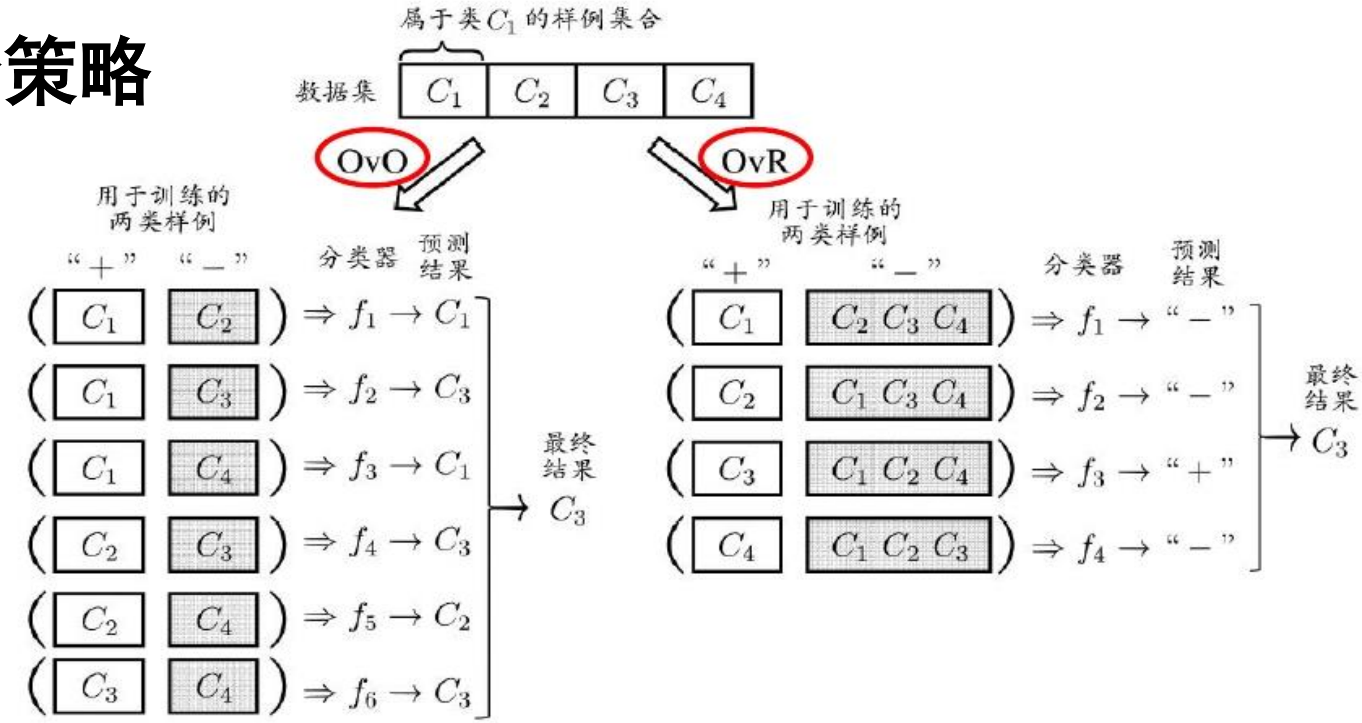


2.7 多分类学习

- **引言**
  - 现实应用中，通常需要处理多分类问题。一些二分类方法可以直接推广到多分类问题。有此则难以从形式上一次性得到推广。
  - 一个基本策略是：利用二分类学习器的组合来解决多分类问题。
  - 基本技术思路是“拆解法”：在分类器构造（训练阶段），即将多分类任务拆为多个二分类任务。在测试阶段，对这些分类器的结果进行集成以获得最终的分类结果。

2.7 多分类学习

- **拆分策略**



  - 一对一(0v0)：
    - N个类别两两配对，产生N(N-1)/2个分类器
  - 一对其余(0vR)：
    - 每次将一个类的样例作为正例，所有其他类的样例作为反例来训练N个分类器

2.7 多分类学习

- **拆分策略**
  - 多对多(MvM):
    - 每次将若干个类作为正例，若干个其它类作为反例。
    - 一种常见方法“纠错输出码”(Error Correcting Output Codes,ECOC
    - 类别划分通过“编码矩阵”(coding matrix)指定。编码矩阵有多种形式，常见的主要有二元码，三元码。

2.7 多分类学习

- **二元ECOC码示意图(四类五种划分)**

	f_1	f_2	f_3	f_4	f_5	海明距离	欧式距离
C_1	-1	+1	-1	+1	+1	3	$2\sqrt{3}$
C_2	+1	-1	-1	+1	-1	4	4
C_3	-1	+1	+1	-1	+1	1	2
C_4	-1	-1	+1	+1	-1	2	$2\sqrt{2}$
预测示例	-1	-1	+1	-1	+1		

计算海明距离的一种方法，就是对两个位串进行异或（xor）运算，并计算出异或运算结果中1的个数。

2.7 多分类学习

- **拆分策略**
  - 多对多(MvM)：
    - 每次将若干个类作为正例，若干个其它类作为反例。
    - 一种常见方法“纠错输出码”(Error Correcting Output Codes,ECOC

编码：对N个类别做M次划分，每次将一部分类别划分正类，一部分划为反类

产生M个训练集，可训练M个分类器；每类分类器对应一个长为M的编码(真实)

取距离最小的类为最终结果

解码：M个分类器对测试样本进行预测，这些预测标记组成一个编码

产生长度为M的预测结果编码

2.7 多分类学习

- **ECOC码为什么会纠错？**
  - ECOC编码对分类器错误有一定容忍和修正能力，编码越长、纠错能力越强
  - 对同等长度的编码，理论上来说，任意两个类别之间的编码距离越远，则纠错能力越强
  - 但并不是编码的理论性能越好，分类性能就越好，因为机器学习问题涉及很多因素。
  - 编码越长，则分类器越多，训练时间越长。

University of Chinese Academy of Sciences

2.8 类别不平衡问题

- **不同类别的样本比例相差很大；“小类”往往更重要**
  - 基本思路：

若$\frac{y}{1-y} > 1$ 则预测为正例. ➡ 若$\frac{y}{1-y} > \frac{m^+}{m^-}$ 则预测为正例.

  - 基本策略——再缩放：

$$\frac{y'}{1-y'} = \frac{y}{1-y} \times \frac{m^-}{m^+}$$

然而精确估计m^+/m^-通常很难！

常见类别不平衡学习方法：
- 过采样
- 欠采样
- 阈值移动

2.8 类别不平衡问题

- 过采样
  - 去除一些反例，来调整正反例比例（比如接近）
  - 缺点：丢失了一些训练样本
  - 利用集成学习机制，将反例划分为若干不同的子集合，训练、集成不同的学习器
- 欠采样
  - 增加一些正例，控制正反比例
  - 不能简单对正例进行重复采样，否则会过拟合
- 阈值移动
  - 用原始数据集，决策时，用缩放的阈值

参考文献及其他资料

- 周志华. 《机器学习》.清华大学出版社, 2015. 北京
- 李航.《统计学习方法》.清华大学出版社，2012年3月出版
- 阿斯顿・张，李沐《动手学深度学习》.人民邮电出版，社2019年6月出版
- 李宏毅《机器学习》课程网站 http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML20.html

致谢

- 感谢向世明老师的20版PPT作为原始材料
- 感谢徐雨婷与段俊贤对本PPT的制作与修改

Thank All of You!

(Questions?)

赫然

rhe@nlpr.ia.ac.cn

智能感知与计算研究中心(CRIPAC)

中科院自动化研究所· 模式识别国家重点实验室

附录： – 用牛顿法求最优解

Newton’s Method: **Solving a system of nonlinear equations** $F(x)=0$

- We **can not** solve arbitrary “*nonlinear equations*” very easily.
- We **can** solve “*linear equations*” using the techniques from linear algebra.
- We linearize the system of “*nonlinear equations*” with a first-order Taylor and solve the obtained “*linear system*”, and then iterate.

(0) *k=0,*

(1) *Linear approximation at point* x_k:

$$y = F(x_k) + \nabla F(x_k)^T (x - x_k)$$

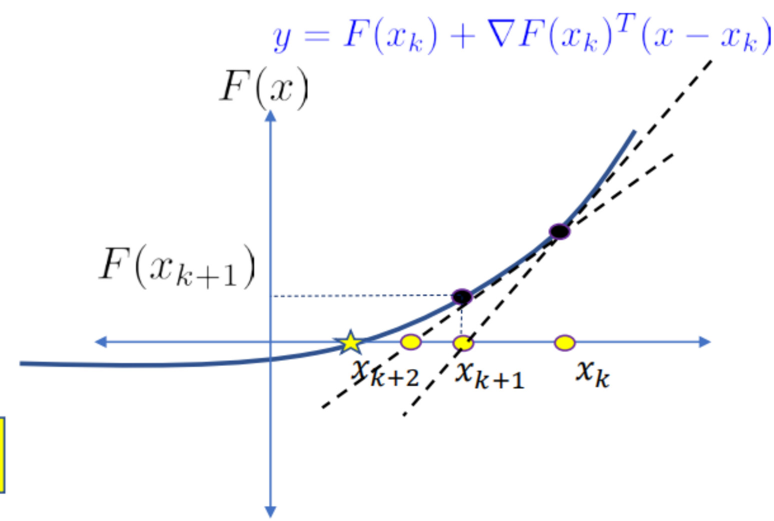
(2) *Solve linear system:*

$$y = 0 \longrightarrow F(x_k) + \nabla F(x_k)'(x - x_k) = 0$$

$$\Rightarrow (x - x_k) = -(\nabla F(x_k))^{-1} F(x_k)$$

$$\Rightarrow \boxed{x_{k+1} = x_k - (\nabla F(x_k))^{-1} F(x_k)}$$

(3) *Go to Step* (1)



二阶泰勒展开 $f(x) = f\left(x^{(k)}\right) + g_k^T \left(x - x^{(k)}\right) + \frac{1}{2}\left(x - x^{(k)}\right)^T H\left(x^{(k)}\right)\left(x - x^{(x)}\right)$

中国科学院大学
University of Chinese Academy of Sciences