电子科技大学
University of Electronic Science and Technology of China

Lecture 5 数据分析算法（II）

分类与聚类

电子科技大学
University of Electronic Science and Technology of China

教学目标

- 认识常用的数据分类与聚类的原理，并比较不同的分类与聚类方法之间的区别。

- 掌握贝叶斯公式在机器学习中的应用思路，明确分类器相关的基本概念，理解朴素贝叶斯分类器、AdaBoost分类器、支持向量机、K近邻、K-Means等算法，了解数据空间的转换与核方法、感知机、逻辑回归、深度学习、极大似然估计和期望最大化方法，了解分类器设计一般规则。

内容概述


数据分析算法
数据关系：TF-IDF，余弦相似, Apriori, PageRank
分类与聚类：Bayes, AdaBoost, SVM, KNN, K-Means, EM
决策：ID3、C4.5、CART

电子科技大学
University of Electronic Science and Technology of China

第5讲 分类与聚类

- 模式分类问题在大数据分析中得到广泛的应用
  - 它是数据的最基本处理方式之一
- 在很多业务场景中都需要用到数据分类
  - 对年收人人群的划分、事物的区分

- 监督学习：分类
- 非监督学习：聚类

问题的提出

水果类型	鲜红值（色度比）	直径（cm）	质量（g）
车厘子	0.81	1.02	8.85
车厘子	0.82	0.98	8.67
车厘子	0.78	0.99	8.75
车厘子	0.79	1.01	8.80
樱桃	0.56	0.85	7.32
樱桃	0.58	0.86	7.33
樱桃	0.59	0.83	7.29
樱桃	0.57	0.84	7.31
？？？	0.8	0.86	8

是车厘子还是樱桃？

6.1 朴素贝叶斯分类器

- 分类过程
  - 训练
  - 识别
- 朴素贝叶斯分类器是一种非常传统的分类方法
  - 具有深刻的统计数学基础
  - Naïve，假设样本的特征之间是彼此独立的

贝叶斯定理

贝叶斯公式：在事件B出现的前提下事件A出现的概率，等于事件A出现的前提下事件B发生的概率乘以事件A出现的概率再除以事件B出现的概率。

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

P (A I B)：事件B出现的前提下事件A出现的概率

P (B I A)：事件A出现的前提下事件B出现的概率

P (A)：　事件A出现的概率

P (B)：　事件B出现的概率

贝叶斯公式简单算例

某个医院早上收了六个门诊病人，诊断如下表：

症状	职业	疾病
打喷嚏	护士	感冒
打喷嚏	农夫	过敏
头痛	建筑工人	脑震荡
头痛	建筑工人	感冒
打喷嚏	教师	感冒
头痛	教师	脑震荡

现在又来了一个病人，是一个打喷嚏的建筑工人，请问他患上感冒的概率有多大？

贝叶斯公式简单算例

按照贝叶斯公式： $P(A|B) = P(B|A)\,P(A) / P(B)$

A: 感冒
B: 建筑工人 x 打喷嚏

P(感冒|打喷嚏x建筑工人) = P(打喷嚏x建筑工人|感冒) x P(感冒)
/ P(打喷嚏x建筑工人)

假定“打喷嚏”和“建筑工人”这两个特征是独立的，因此变成：

P(感冒|打喷嚏x建筑工人) = P(打喷嚏|感冒) x P(建筑工人|感冒) x P(感冒)
/ P(打喷嚏) x P(建筑工人)

贝叶斯公式简单算例

P(打喷嚏|感冒) = 2 / 3 = 0.66

P(建筑工人|感冒) = 1 / 3 = 0.33

P(感冒) = 3 / 6 = 0.50

P(打喷嚏) = 3 / 6 = 0.50

P(建筑工人) = 2 / 6 = 0.33

P(感冒|打喷嚏x建筑工人) = P(打喷嚏|感冒) x P(建筑工人|感冒) x P(感冒) / P(打喷嚏) x P(建筑工人)

= (0.66x0.33x0.5) / (0.5x0.33)

= 0.66

结论：这个打喷嚏的建筑工人有66%的可能得了感冒。

贝叶斯分类模型

- $P(A|B) = \frac{P(A)P(B|A)}{P(B)}$
- 给定特征$W = (w_1, \cdots, w_n)$，类别集合$C = \{c_1, \cdots, c_m\}$，有

$$P(C|W) = \frac{P(C)P(W|C)}{P(W)}$$

- 证据因子，$P(W) = \sum_{i=1}^{m}\left(P(c_i)\prod_{j=1}^{n}P(w_j|c_i)\right)$，常数
- $P(C)$表示选取某个类别的概率密度，分布未知时，假设$P(C)$服从一个分布
  - 例如贝努利模型$P(c) = \frac{|D_c|}{|D|}$，$|D|$表示样本总数，D_c表示属于c类的样本数

回到分类问题

假设训练集样本的特征满足高斯分布，计算得到下表

类别	鲜红值		直径		质量	
	均值	标准差	均值	标准差	均值	标准差
车厘子	0.8	0.018	1	0.0183	8.7675	0.077
樱桃	0.575	0.013	0.845	0.0129	7.3125	0.017

我们认为两种类别是等概率的(样本集中类别均衡)，也即

P(车厘子) = P(樱桃) = 0.5

概率密度函数为：

$$\frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

分类问题

水果类型	鲜红值（色度比）	直径（cm）	质量（g）
车厘子	0.81	1.02	8.85
车厘子	0.82	0.98	8.67
车厘子	0.78	0.99	8.75
车厘子	0.79	1.01	8.80
樱桃	0.56	0.85	7.32
樱桃	0.58	0.86	7.33
樱桃	0.59	0.83	7.29
樱桃	0.57	0.84	7.31
样　本	0.8	0.86	8

分类问题

样本属于车厘子的后验概率

$$\text{posterior}(\text{车厘子}) = \frac{P|(\text{车厘子})\,p(\text{鲜红值}|\text{车厘子})\,p(\text{直径}|\text{车厘子})\,p(\text{质量}|\text{车厘子})}{\text{evidence}}$$

样本属于樱桃的后验概率

$$\text{posterior}(\text{樱桃}) = \frac{P(\text{樱桃})\,p(\text{鲜红值}|\text{樱桃})\,p(\text{直径}|\text{樱桃})\,p(\text{质量}|\text{樱桃})}{\text{evidence}}$$

证据因子evidence 计算

$$\begin{aligned}\text{evidence} = {} & P(\text{车厘子}) \times p(\text{鲜红值}|\text{车厘子}) \times p(\text{直径}|\text{车厘子}) \times p(\text{质量}|\text{车厘子}) + \\ & P(\text{樱桃}) \times p(\text{鲜红值}|\text{樱桃}) \times p(\text{直径}|\text{樱桃}) \times p(\text{质量}|\text{樱桃})\end{aligned}$$

分类问题

$$P(\text{车厘子})=0.5,\ p(\text{鲜红值}|\text{车厘子})=\frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{(0.8-\mu)^2}{2\sigma^2}\right]=21.850968113$$

其中，μ=0.8，α=0.018257419，二者均为训练集样本的高斯分布参数

$p(\text{直径}|\text{车厘子})\approx 3.725748179\times 10^{-12}$

$p(\text{质量}|\text{车厘子})\approx 1.011897485\times 10^{-21}$

$\text{posteriornumerator}(\text{车厘子})=41.1898966\times 10^{-33}$

$p(\text{樱桃})=0.5$

$p(\text{鲜红值}|\text{樱桃})\approx 3.400214291\times 10^{-65}$

$p(\text{直径}|\text{樱桃})\approx 15.733918998,$

$p(\text{质量}|\text{樱桃})\approx 0$

$\text{posteriornumerator}(\text{樱桃})\approx 0。$

通过计算可以看出，车厘子的后验概率分子较大，由此可以估测这个样本属于车厘子的可能性稍大一点。

6.2 AdaBoost分类器

AdaBoost算法的思想是从训练数据中学习一系列弱分类器，然后将这些弱分类器组合成强分类器

- 基于测试过程中的错误反馈调节分类器的分类效果，中国哲学思想的“三个臭皮匠，顶个诸葛亮”原理！
- 算法中有两种权重，一种是数据权重，另一种是弱分类器权重。数据权重主要用于弱分类器寻找其分类误差最小的决策点，找到之后用这个最小误差计算出该弱分类器的权重，分类器权重越大说明该弱分类器在最终决策时拥有更大的发言权
- Adaboost 会对分错的数据加大权重，由于权重增大影响，那么下一轮弱分类器就有更大的把握将当前轮没有正确分类的数据分对，如果下一轮还是没有分对，那么这一点的数据权重将继续增加，这样一轮一轮的持续下去，后面的分类器将会更加注意这个数据的分类，这样将其分对的概率也就越高

6.2 AdaBoost分类器

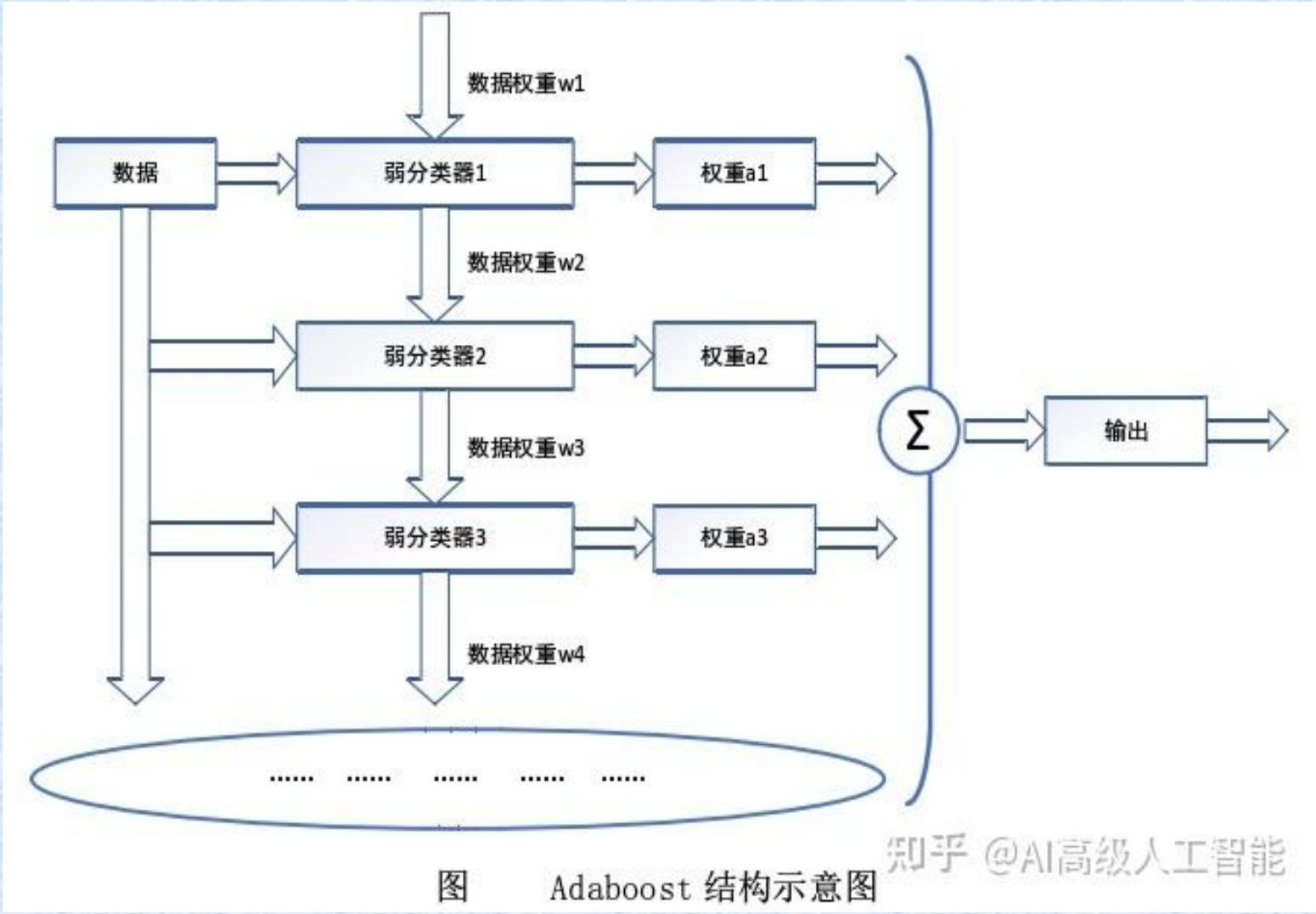


图　　Adaboost 结构示意图

6.2 AdaBoost分类器

分类样本集

序号	1	2	3	4	5	6	7	8	9	10
X	0	1	2	3	4	5	6	7	8	9
Y	1	1	1	-1	-1	-1	1	1	1	-1

初始权重 **D1=1/N =** { 0.1, 0.1, 0.1, 0.1, …, 0.1 }

计算错误率 $$\varepsilon_t = \frac{\sum_{i=1}^{N_t} I[h_t(x_i) \neq y_i] D_t}{N_t}$$

计算分类器权重 $$\alpha = \frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$$

计算下一轮数据权重 $$D(t+1) = \begin{cases} \frac{D_t(i)(1-\varepsilon_t)}{\varepsilon_t}, & h_t(x_i) \neq y_i \\ \frac{D_t(i)(\varepsilon_t)}{1-\varepsilon_t}, & h_t(x_i) = y_i \end{cases}$$

6.2 AdaBoost分类器

假设有三个基础分类器

$$f_1(x)=\begin{cases}1, x <= 2.5\\ -1, x > 2.5\end{cases}$$

$$f_2(x)=\begin{cases}-1, x <= 5.5\\ 1, x > 5.5\end{cases}$$

$$f_3(x)=\begin{cases}1, x <= 8.5\\ -1, x > 8.5\end{cases}$$

计算知道分类器 f1 的错误率为 0.3 (x 取值 6、7、8 时分类错误)；分类器 f2 的错误率为 0.4 (x 取值 0、1、2、9 时分类错误)；分类器 f3 的错误率为 0.3 (x 取值为 3、4、5时分类错误)。

这三个分类器中，f1、f3 分类器的错误率最低，因此我们选择 f1 或 f3 作为最优分类器，这里假设选 f1 分类器作为最优弱分类器，即第一轮训练得到：

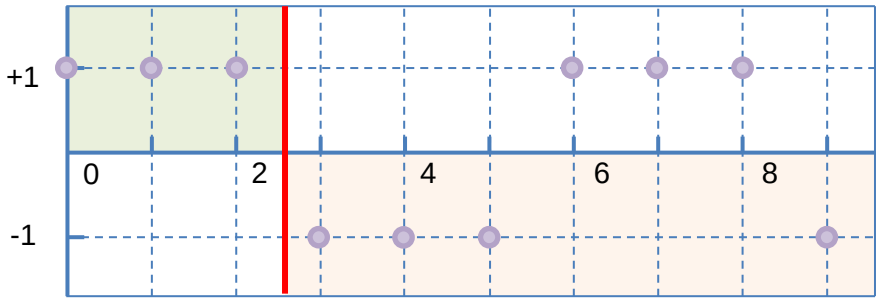
第一轮迭代

1、初始化样本权重: $D_1 = \{0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\}$

迭代

2、设定阈值: 2.5，设计弱分类器 $G_m(x): x \to \{-1, 1\}$

$$G_1(x) = \begin{cases} 1, x < 2.5 \\ -1, x > 2.5 \end{cases}$$



3、计算误差率：$\epsilon = 0.3$

4、计算弱分类器$G_m(x)$的权重，$a_1 = \frac{1}{2}\ln\frac{1-\epsilon_1}{\epsilon_1} \approx 0.42$

5、更新样本权重，

$$z_m = \sum_{j=1}^{m} a_j G_j(x)$$

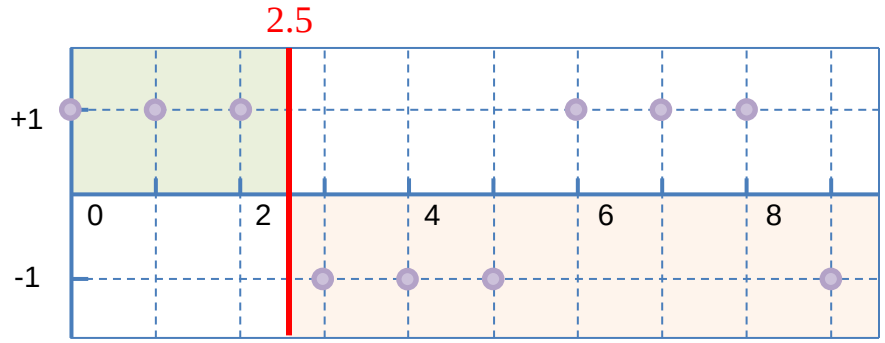
$$w_{m+1,i} = \frac{w_{m,i}}{z_m} e^{-a_m y_i G_m(x_i)}$$

$$D_{m+1} = (w_{m+1,1}, w_{m+1,2}, \cdots, w_{m+1,N})$$

最终形成分类器：$G(x) = \mathbf{sign}(\sum_{m=1}^{M} a_m G_m(x))$

第一轮迭代

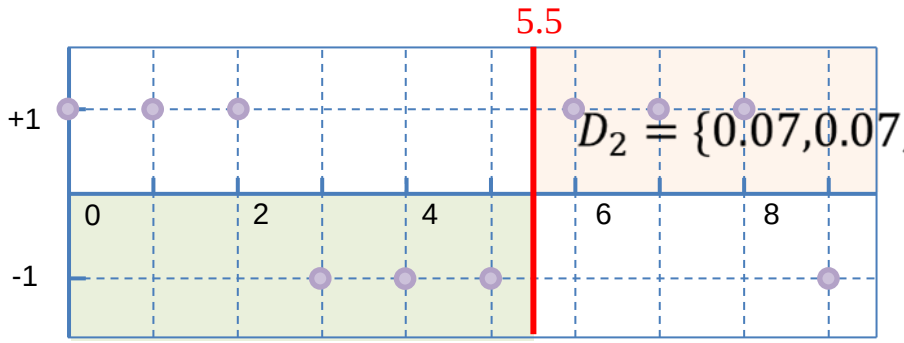
权重: $D_1 = \{0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\}$



$\epsilon_1 = 0.3$

$$G_1(x) = \begin{cases} 1, x < 2.5 \\ -1, x > 2.5 \end{cases}$$

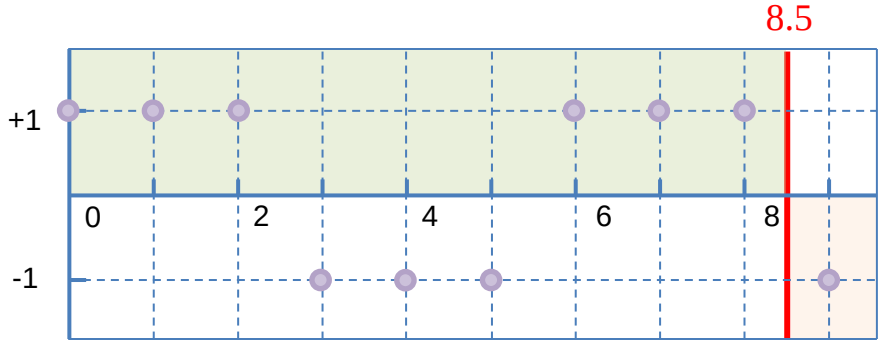
$$a_1 = \frac{1}{2}\ln\frac{1-\epsilon_1}{\epsilon_1} \approx 0.42$$



$D_2 = \{0.07, 0.07, 0.07, 0.07, 0.07, 0.07, 0.17, 0.17, 0.17, 0.07\}$

$\epsilon_1 = 0.4$

$$f_1(x) = a_1 \times G_1(x) = 0.42 G_1(x)$$



$\epsilon\_1 = 0.3$

第一轮迭代

对于m=1，在权值分布为**D1**（10个数据，每个数据的权值皆初始化为0.1）的训练数据上，经过计算可得：

阈值v取2.5时误差率为0.3（x < 2.5时取1，x > 2.5时取-1，**则6 7 8分错**，误差率为0.3），

阈值v取5.5时误差率最低为0.4（x < 5.5时取1，x > 5.5时取-1，则3 4 5 6 7 8皆分错，误差率0.6大于0.5，不可取。故令x > 5.5时取1，x < 5.5时取-1，则**0 1 2 9分错**，误差率为0.4）

故这时$G_2(x)$修正为： x > 5.5，y = 1; x < 5.5, y = -1

阈值v取8.5时误差率为0.3（x < 8.5时取1，x > 8.5时取-1，则**3 4 5分错**，误差率为0.3）。

所以**无论阈值v取2.5，还是8.5，总得分错3个样本**，故可任取其中任意一个如2.5，第一个弱分类器为：

第一轮最优弱分类器为 G1(x)：x < 2.5,则 y = 1; x > 2.5, 则 y = -1

第一轮迭代

计算最优弱分类器的权重

$\alpha = 0.3 * \ln((1 - 0.3) / 0.3) = 0.4236$

更新样本权重

x = 0, 1, 2, 3, 4, 5, 9时，y分类正确，则样本权重为：

0.1 * exp(-0. 4236) = 0.0715

x = 6, 7, 8时，y分类错误，则样本权重为：

0.1 * exp(0. 4236) = 0.1667

新样本权重总和为 0. 0715*7 + 0. 1667*3 = 1. 000

第一轮得到的强分类器：$G(x) = 0.4236*G_1(x)$

第二轮迭代

权重: $D_2 = \{0.07,0.07,0.07,0.07,0.07,0.07,0.17,0.17,0.17,0.07\}$

在第二轮训练中，我们继续统计三个分类器的准确率，可以得到分类器 f1 的错误率为 0.1666*3，也就是 x 取值为 6、7、8 时分类错误。分类器 f2 的错误率为 0.0715*4，即 x 取值为 0、1、2、9 时分类错误。分类器 f3 的错误率为 0.0715*3，即 x 取值 3、4、5 时分类错误。

在这 3 个分类器中，f3 分类器的错误率最低，因此我们选择 f3 作为第二轮训练的最优分类器，即：

$$G_2(x) = \begin{cases} 1, x <= 8.5 \\ -1, x > 8.5 \end{cases}$$

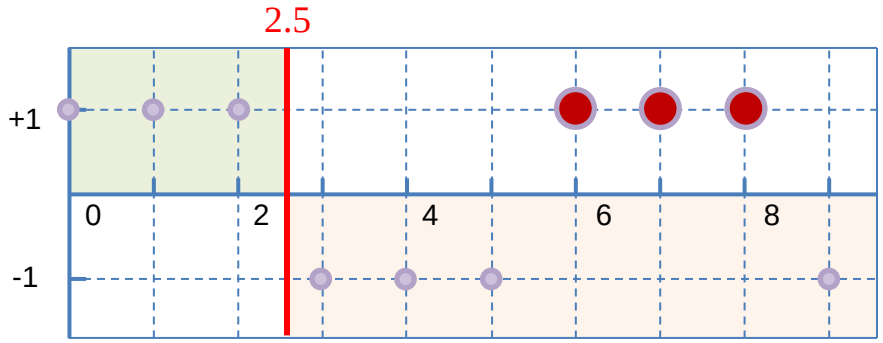
根据分类器权重公式得到：

$$\alpha_2 = \frac{1}{2}\log\frac{1-e_2}{e_2} = 0.6496$$

同样，我们对下一轮的样本更新求权重值，代入 $W_{k+1,i}$ 和 D_{k+1} 的公式，可以得到 D3=(0.0455,0.0455,0.0455,0.1667, 0.1667,0.01667,0.1060, 0.1060, 0.1060, 0.0455)。

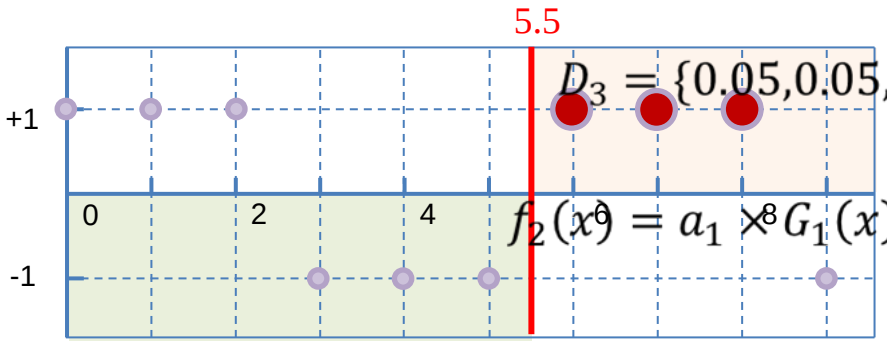
第二轮迭代

权重: $D_2 = \{0.07,0.07,0.07,0.07,0.07,0.07,0.17,0.17,0.17,0.07\}$



$\epsilon_2 = 0.51$

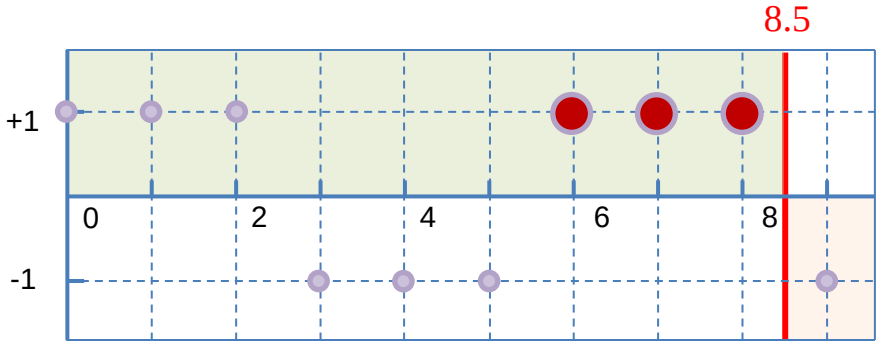
$$a_2 = \frac{1}{2}\ln\frac{1-\epsilon_2}{\epsilon_2} \approx 0.65$$



$D_3 = \{0.05,0.05,0.05,0.17,0.17,0.17,0.11,0.11,0.11,0.05\}$

$\epsilon_2 = 0.28$

$f_2(x) = a_1 \times G_1(x) + a_2 \times G_2(x) = 0.42G_1(x) + 0.65G_2(x)$



$\epsilon_2 = 0.21$

$$G_2(x) = \begin{cases} 1, x < 8.5 \\ -1, x > 8.5 \end{cases}$$

第三轮迭代

权重: $D_3 = \{0.05,0.05,0.05,0.17,0.17,0.17,0.11,0.11,0.11,0.05\}$

在第三轮训练中，我们继续统计三个分类器的准确率，可以得到分类器 f1 的错误率为 0.1060*3，也就是 x 取值 6、7、8 时分类错误。分类器 f2 的错误率为 0.0455*4，即 x 取值为 0、1、2、9 时分类错误。分类器 f3 的错误率为 0.1667*3，即 x 取值 3、4、5 时分类错误。

在这 3 个分类器中，f2 分类器的错误率最低，因此我们选择 f2 作为第三轮训练的最优分类器，即：

$$G_3(x) = \begin{cases} -1, x <= 5.5 \\ 1, x > 5.5 \end{cases}$$

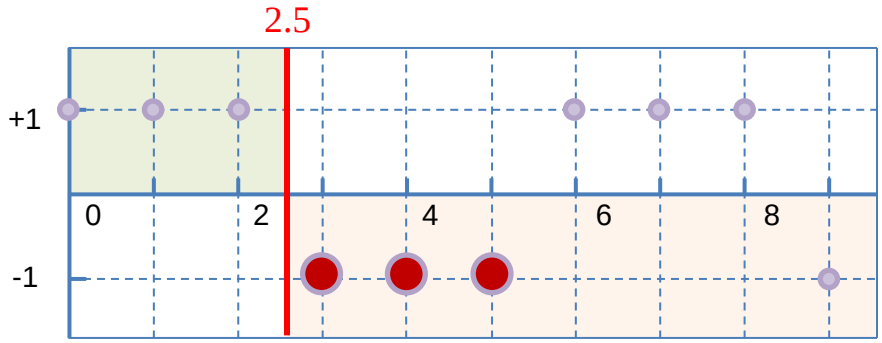
我们根据分类器权重公式得到：

$$\alpha_3 = \frac{1}{2}\log\frac{1-e_3}{e_3} = 0.7514$$

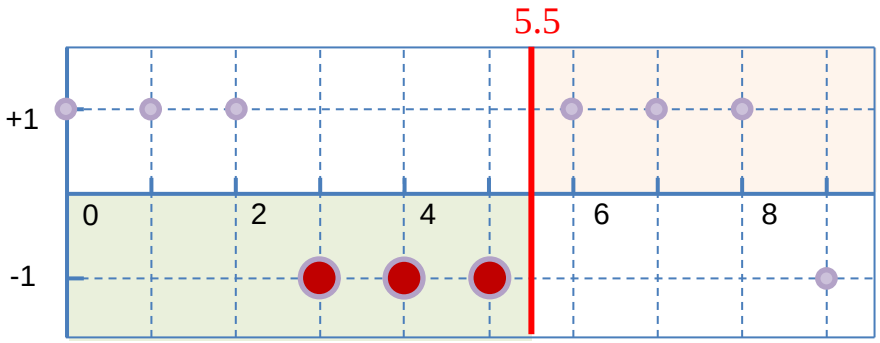
假设我们只进行 3 轮的训练，选择 3 个弱分类器，组合成一个强分类器，那么最终的强分类器 G(x) = 0.4236G1(x) + 0.6496G2(x)+0.7514G3(x)。

第三轮迭代

权重: $D_3 = \{0.05,0.05,0.05,0.17,0.17,0.17,0.11,0.11,0.11,0.05\}$



$\epsilon_3 = 0.33$

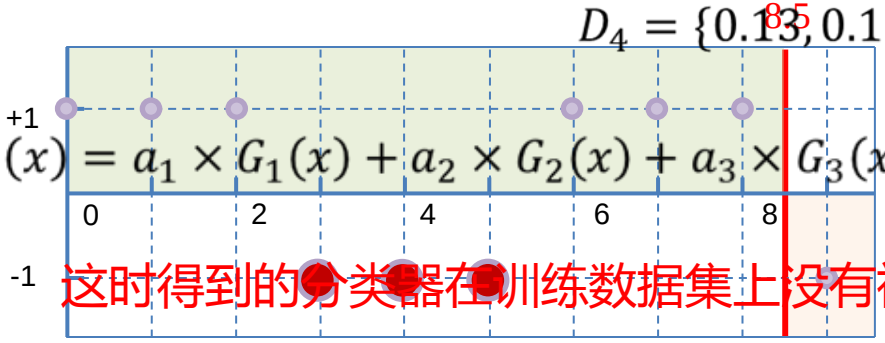


$\epsilon_3 = 0.2$

$$G_3(x) = \begin{cases} -1, x < 5.5 \\ 1, x > 5.5 \end{cases}$$

$$a_3 = \frac{1}{2}\ln\frac{1-\epsilon_3}{\epsilon_3} \approx 0.69$$

$D_4 = \{0.13, 0.13, 0.13, 0.1, 0.1, 0.1, 0.07, 0.07, 0.07, 0.13\}$

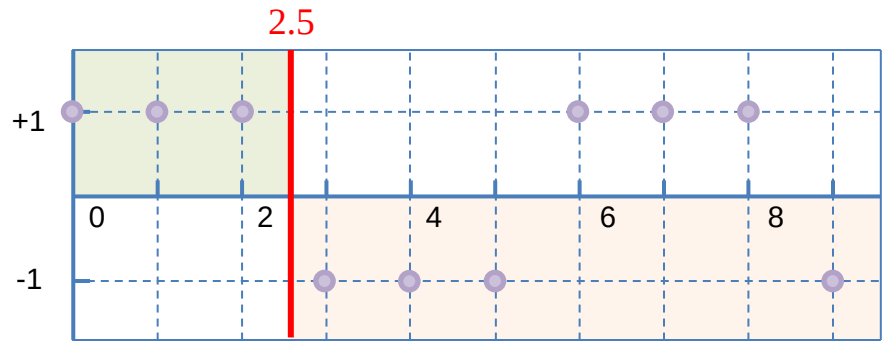


$$f_3(x) = a_1 \times G_1(x) + a_2 \times G_2(x) + a_3 \times G_3(x) = 0.42G_1(x) + 0.65G_2(x) + 0.69G_3(x)$$

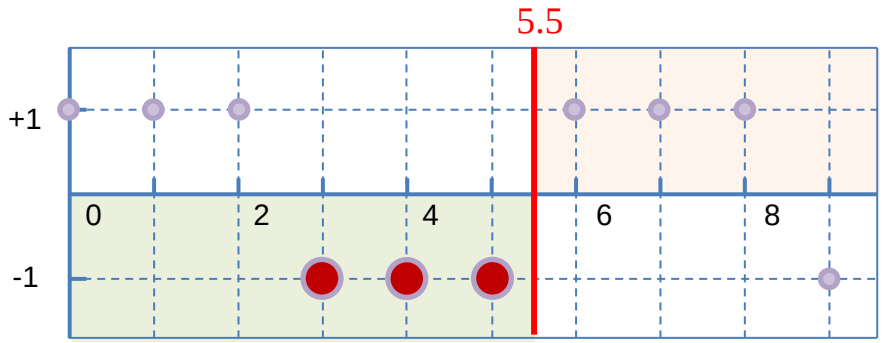
$\epsilon_3 = 0.51$

这时得到的分类器在训练数据集上没有被错分的点，整个训练过程结束。

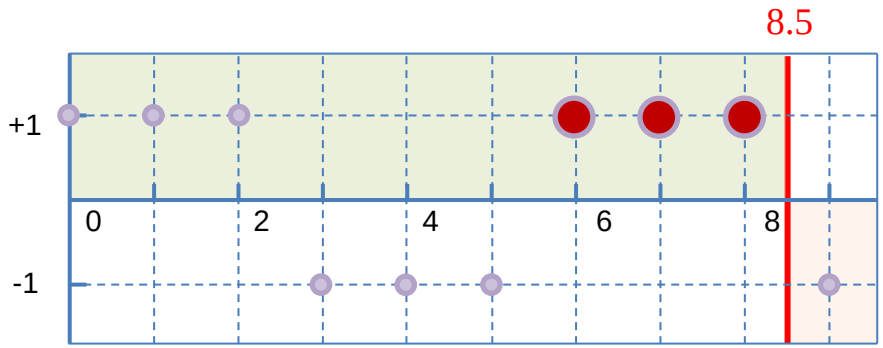
弱分类器的线性组合



$\epsilon_1 = 0.3$ $G_1(x) = \begin{cases} 1, x < 2.5 \\ -1, x > 2.5 \end{cases}$



$\epsilon_3 = 0.2$ $G_3(x) = \begin{cases} -1, x < 5.5 \\ 1, x > 5.5 \end{cases}$



$\epsilon_2 = 0.21$ $G_2(x) = \begin{cases} 1, x < 8.5 \\ -1, x > 8.5 \end{cases}$

$$f_3(x) = a_1 \times G_1(x) + a_2 \times G_2(x) + a_3 \times G_3(x) = 0.42G_1(x) + 0.65G_2(x) + 0.69G_3(x)$$

AdaBoost

序号	1	2	3	4	5	6	7	8	9	10
样本点 X	(1,5)	(2,2)	(3,1)	(4,6)	(6,8)	(6,5)	(7,9)	(8,7)	(9,8)	(10,2)
类别 Y	1	1	-1	-1	1	-1	1	1	-1	-1
权值分布 $D1$	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
权值分布 $D2$	1/14	1/14	1/14	1/14	1/6	1/14	1/6	1/6	1/14	1/14
$sign(f_1(x))$	1	1	-1	-1	-1	-1	-1	-1	-1	-1
权值分布 $D3$	1/22	1/22	1/6	1/6	7/66	1/6	7/66	7/66	1/22	1/22
$sign(f_2(x))$	1	1	1	1	1	1	1	1	-1	-1
权值分布 $D4$	1/6	1/6	11/114	11/114	7/114	11/114	7/114	7/114	1/6	1/38
$sign(f_3(x))$	1	1	-1	-1	1	-1	1	1	-1	-1

$$G(x) = \text{sign}\big(f_3(x)\big)$$
$$= \text{sign}\big(0.42G_1(x) + 0.65G_2(x) + 0.69G_3(x)\big)$$

AdaBoost

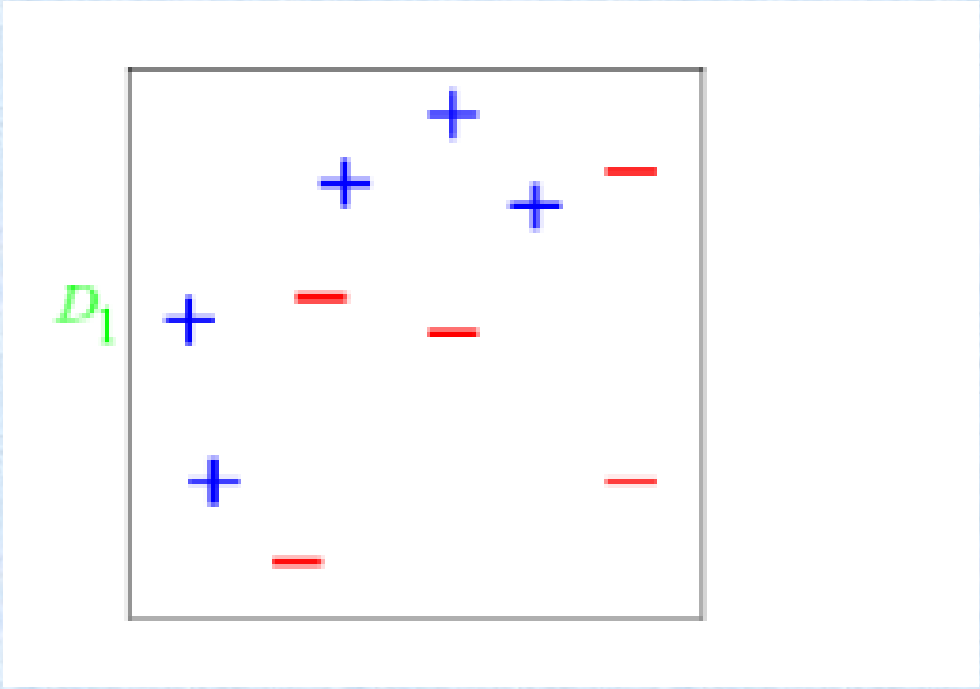
- 从上述三轮迭代可以看出，如果某个样本被错分，那么它们在下一轮迭代中的权值将被增大，从而被凸显出来；（凸显错分样本）
- 同时，分类正确的样本的权值在下一轮将被降低。（弱化正确分类的样本）
- 通过这样的方式，误差率ϵ不断降低。

$$G(x) = \text{sign}\big(f_3(x)\big)$$
$$= \text{sign}\big(0.42G_1(x) + 0.65G_2(x) + 0.69G_3(x)\big)$$

AdaBoost算例

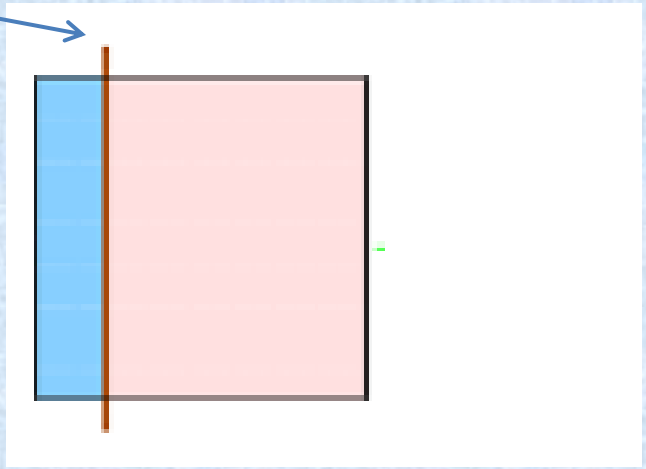
图中“+”和“-”表示两种类别

共10个样本，故每个样本权值为0.1



AdaBoost算例

第一次划分 $G_1(x)$



有3个点划分错误

得到误差：e1=（0.1+0.1+0.1) /1.0 = 0.30

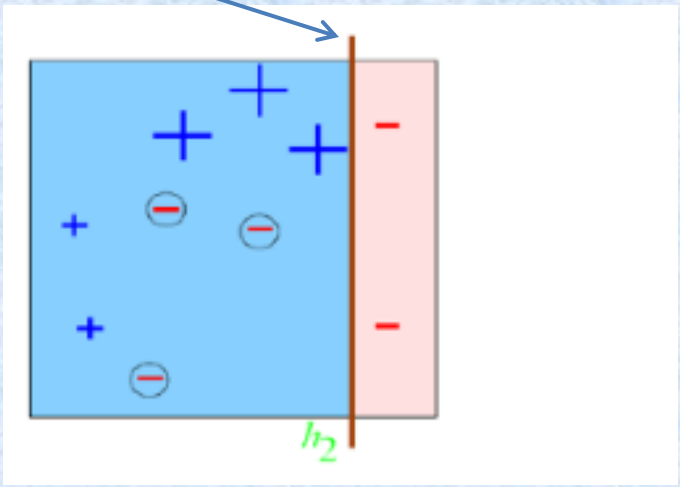
分类器权重：

$$\alpha_1=\frac{1}{2}\ln\left(\frac{1-\varepsilon_1}{\varepsilon_1}\right)=\frac{1}{2}\ln\left(\frac{1-0.3}{0.3}\right)=0.42$$

AdaBoost算例

根据算法，对于正确分类的7个点权值不变仍为0.1, 对于错分的3个点权值为：
D1=D0*(1-e1)/e1=0.1*(1-0.3)/0.3=0.2333

第二次划分 $G_2(x)$



第二次分类，有3个 "-" 分类错误，按照算法计算：

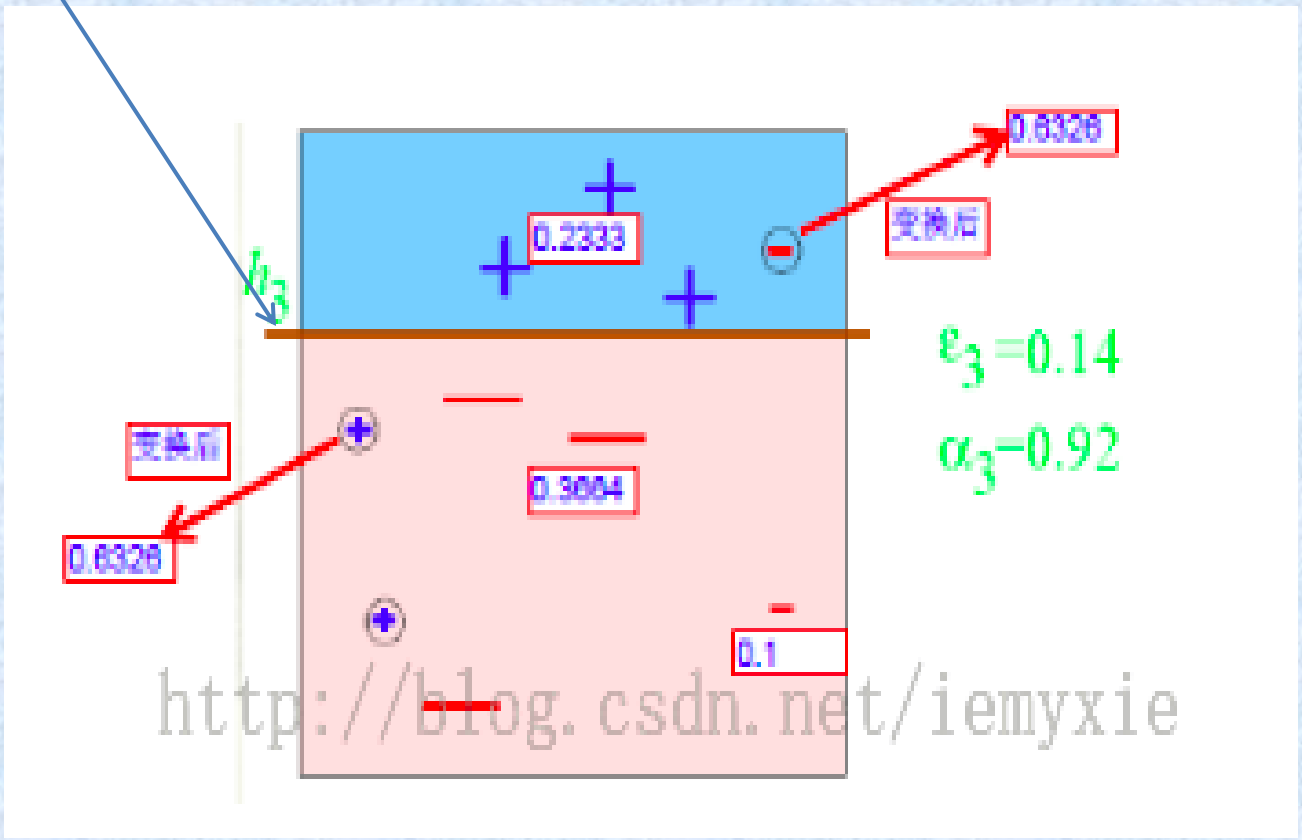
分类误差：e2=0.1*3/1.3990=0.2144

分类器权重：a2=0.6493

错分的3个点权值更新为为：D2=0.1*（1-0.2144）/0.2144=0.3664

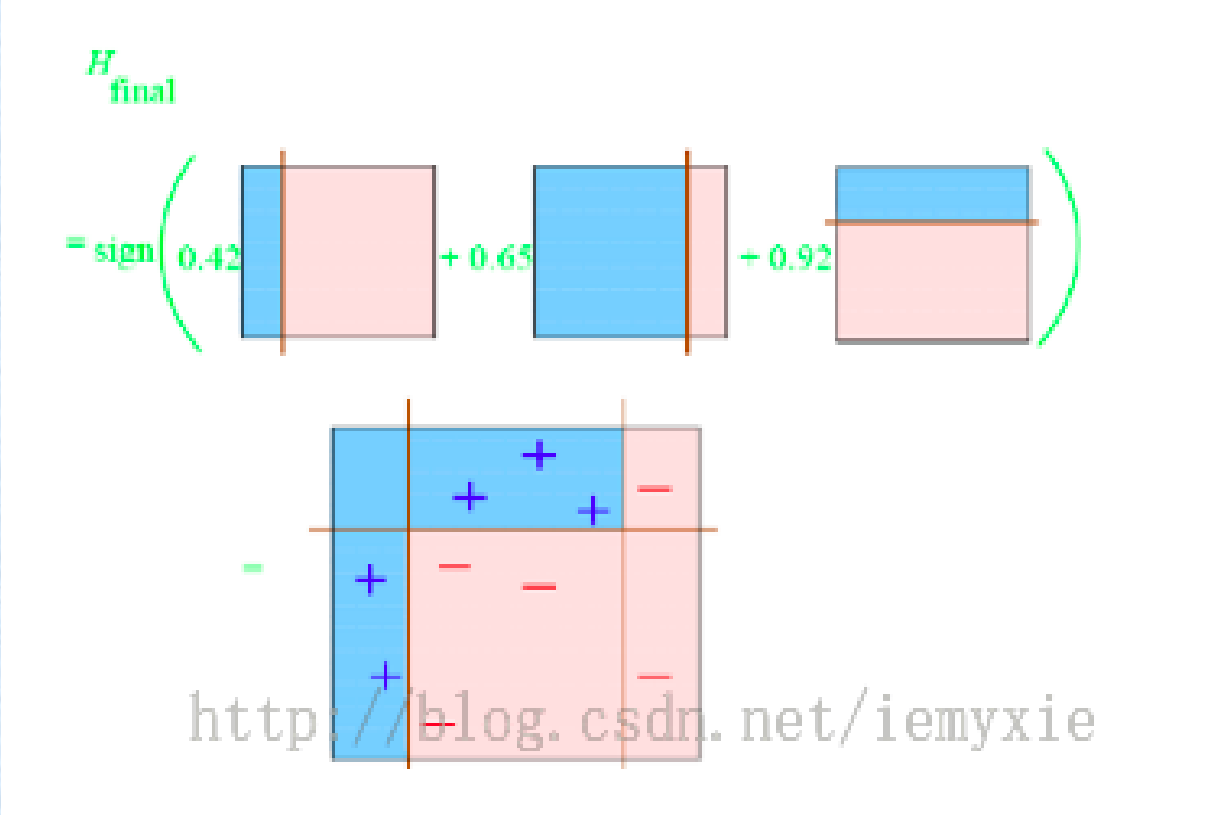
AdaBoost算例

第三次划分 $G_3(x)$



AdaBoost算例

最后将三次的分类器结合起来，得到如下的分类结果：



AdaBoost补充算例

训练当前迭代最优弱分类器

最优弱分类器是错误率最小的那个弱分类器。错误率的计算公式：

$$e_m = \sum_{i=1}^{N} w_{mi} I(G_m(x_i) \neq y_i)$$

其中m = 1,2,..,M，代表第m轮迭代。

i代表第i个样本。

w是样本权重。

I指示函数取值为1或0，

当I指示函数括号中的表达式为真时，I函数结果为1；当I函数括号中的表达式为假时，I函数结果为0。

取错误率最低的弱分类器为当前迭代的最优弱分类器。

注意，第一轮迭代计算时样本权重初始化为总样本数分之一。

AdaBoost补充算例

计算最优弱分类器的权重

弱分类器的权重计算公式（第m轮）如下：

$$\alpha_m = \frac{1}{2}\log\left(\frac{1-e_m}{e_m}\right)$$

可以看出，错误率越小，则α值越大，即该弱分类器的权重越高；

错误率越大，则α值越小，则该弱分类器的权重越小。

这样可以使分类精度高的弱分类器起到更大的作用，并削弱精度低的弱分类器的作用。

AdaBoost补充算例

根据错误率更新样本权重

样本权重的更新与当前样本权重和弱分类器的权重有关。样本权重更新公式如下：

$$w_{m+1,i} = \frac{w_{mi}}{z_m} \exp(-\alpha_m y_i G_m(x_i))$$

$$z_m = \sum_{i=1}^{N} w_{mi} \exp(-\alpha_m y_i G_m(x_i))$$

其中m = 1,2,..,M，代表第 m 轮迭代。

i代表第i个样本。

w 是样本权重。

α是弱分类器的权重。当样本被正确分类时，y 和 Gm 取值一致，则新样本权重变小；当样本被错误分类时，y 和 Gm 取值不一致，则新样本权重变大。

这样处理，可以使被错误分类的样本权重变大，从而在下一轮迭代中得到重视。

AdaBoost补充算例

表 1. 示例数据集

x	0	1	2	3	4	5
y	1	1	-1	-1	1	-1

样本初始权重值为 D_0 =（0.167，0.167，0.167，0.167，0.167，0.167）

表1数据集的切分点有5个：0.5，1.5，2.5，3.5，4.5

AdaBoost补充算例

若按0.5切分数据，得弱分类器x < 0.5,则 y = 1; x > 0.5, 则 y = -1
此时错误率为2 * 0.167 = 0.334

若按1.5切分数据，得弱分类器x < 1.5,则 y = 1; x > 1.5, 则 y = -1
此时错误率为1 * 0.167 = 0.167

若按2.5切分数据，得弱分类器x < 2.5,则 y = 1; x > 2.5, 则 y = -1
此时错误率为2 * 0.167 = 0.334

若按3.5切分数据，得弱分类器x < 3.5,则 y = 1; x > 3.5, 则 y = -1
此时错误率为3 * 0.167 = 0.501

若按4.5切分数据，得弱分类器x < 4.5,则 y = 1; x > 4.5, 则 y = -1
此时错误率为2 * 0.167 = 0.334

由于按1.5划分数据时错误率最小为0.167，则第一轮最优弱分类器为
G1(x)：x < 1.5,则 y = 1; x > 1.5, 则 y = -1

AdaBoost补充算例

计算最优弱分类器的权重

$\alpha = 0.5 * \ln((1 - 0.167) / 0.167) = 0.8047$

更新样本权重

x = 0, 1, 2, 3, 5时，y分类正确，则样本权重为：

0.167 * exp(-0.8047) = 0.075

x = 4时，y分类错误，则样本权重为：

0.167 * exp(0.8047) = 0.373

新样本权重总和为 0. 075*5 + 0. 373*1 = 0. 748

AdaBoost补充算例

规范化后

x = 0, 1, 2, 3, 5时，样本权重更新为：0.075 / 0.748 = 0.10

x = 4时, 样本权重更新为：0.373 / 0.748 = 0.50

综合上述，新的样本权重为 D_1 = (0.1, 0.1, 0.1, 0.1, 0.5, 0.1)

第一轮得到的强分类器：

$G(x) = 0.8047 * G_1(x)$

这里$G_1(x)$为x < 1.5,则 y = 1; x > 1.5, 则 y = -1

强分类器的错误率为1 / 6 = 0.167。

AdaBoost补充算例

第二轮迭代

若按0.5切分数据，得弱分类器x > 0.5,则 y = 1; x < 0.5, 则 y = -1 (修正)

此时错误率为 4 * 0.1 = 0.4　(错了x = 0, 2, 3, 5)

若按1.5切分数据，得弱分类器x < 1.5,则 y = 1; x > 1.5, 则 y = -1

此时错误率为 1 * 0.5 = 0.5　(错了x = 4)

若按2.5切分数据，得弱分类器x > 2.5,则 y = 1; x < 2.5, 则 y = -1 (修正)

此时错误率为 4 * 0.1 = 0.4　(错了x = 0, 1, 3, 5)

若按3.5切分数据，得弱分类器x > 3.5,则 y = 1; x < 3.5, 则 y = -1 (修正)

此时错误率为 3 * 0.1 = 0.3　(错了x = 0, 1, 5)

若按4.5切分数据，得弱分类器x < 4.5,则 y = 1; x > 4.5, 则 y = -1

此时错误率为 2 * 0.1 = 0.2　(错了x = 2, 3)

由于按4.5划分数据时错误率最小为0.2，则第二轮最优弱分类器为

$G_2(x)$: x < 4.5,则 y = 1; x > 4.5, 则 y = -1

AdaBoost补充算例

计算最优弱分类器的权重

$\alpha = 0.5 * \ln((1 - 0.2) / 0.2) = 0.6931$

更新样本权重

x = 0, 1, 5时，y分类正确，则样本权重为：

0.1 * exp(-0.6931) = 0.05

x = 4时，y分类正确，则样本权重为：

0.5 * exp(-0.6931) = 0.25

x = 2, 3时，y分类错误，则样本权重为：

0.1 * exp(0.6931) = 0.20

新样本权重总和为 0. 05*3 + 0. 25*1 + 0. 2*2 = 0. 800

AdaBoost补充算例

规范化后

x = 0, 1, 5时，样本权重更新为：0.05 / 0.8 = 0.0625

x = 4时, 样本权重更新为：0.25 / 0.8 = 0.3125

x = 2, 3时，样本权重更新为：0.20 / 0.8 = 0.2500

新的样本权重为 $D_2 = (0.0625,\ 0.0625,\ 0.25,\ 0.25,\ 0.3125,\ 0.0625)$

第二轮得到的强分类器：

$$G(x) = 0.8047 * G_1(x) + 0.6931G_2(x)$$

这里$G_1(x)$为 $x < 1.5$,则 $y = 1$; $x > 1.5$, 则 $y = -1$

$G_2(x)$为 $x < 4.5$,则 $y = 1$; $x > 4.5$, 则 $y = -1$

按G(x)分类会使x=4分类错误，则强分类器的错误率为1 / 6 = 0.167

AdaBoost补充算例

第三轮迭代

若按0.5切分数据，得弱分类器x < 0.5,则 y = 1; x > 0.5, 则 y = -1

此时错误率为 0.0625 + 0.3125= 0.3750 (错了x = 1, 4)

若按1.5切分数据，得弱分类器x < 1.5,则 y = 1; x > 1.5, 则 y = -1

此时错误率为 1 * 0.3125 = 0.3125 (错了x = 4)

若按2.5切分数据，得弱分类器x > 2.5,则 y = 1; x < 2.5, 则 y = -1 (修正)

此时错误率为 3 * 0.0625 + 0.25 = 0.4375 (错了x = 0, 1, 3, 5)

若按3.5切分数据，得弱分类器x > 3.5,则 y = 1; x < 3.5, 则 y = -1 (修正)

此时错误率为 3 * 0.0625 = 0.1875 (错了x = 0, 1, 5)

若按4.5切分数据，得弱分类器x < 4.5,则 y = 1; x > 4.5, 则 y = -1

此时错误率为 2 * 0.25 = 0.5000 (错了x = 2, 3)

由于按3.5划分数据时错误率最小为0.1875，则第三轮最优弱分类器为

$G_3(x)$: x > 3.5,则 y = 1; x < 3.5, 则 y = -1

AdaBoost补充算例

计算最优弱分类器的权重

$\alpha = 0.5 * \ln((1 - 0.1875) / 0.1875) = 0.7332$

更新样本权重

x = 2, 3时，y分类正确，则样本权重为：

0.25 * exp(-0.7332) = 0.12

x = 4时，y分类正确，则样本权重为：

0.3125 * exp(-0.7332) = 0.15

x = 0, 1, 5时，y分类错误，则样本权重为：

0.0625 * exp(0. 7332) = 0.13

新样本权重总和为 0. 12*2 + 0. 15*1 + 0. 13*3 = 0. 781

AdaBoost补充算例

规范化后

x = 2, 3时，样本权重更新为：0.12 / 0.781 = 0.154

x = 4时, 样本权重更新为：0.15 / 0.781 = 0.192

x = 0, 1, 5时，样本权重更新为：0.13 / 0.781 = 0.167

新的样本权重为 $D_3 = (0.167,\ 0.167,\ 0.154,\ 0.154,\ 0.192,\ 0.167)$

第三轮得到的强分类器：

$$G(x) = 0.8047 * G_1(x) + 0.6931G_2(x) + 0.7332G_3(x)$$

这里$G_1(x)$为$x < 1.5$,则 $y = 1$; $x > 1.5$, 则 $y = -1$

$G_2(x)$为$x < 4.5$,则 $y = 1$; $x > 4.5$, 则 $y = -1$

$G_3(x)$为$x > 3.5$,则 $y = 1$; $x < 3.5$, 则 $y = -1$ (修正)

按G(x)分类使所有数据分类正确，强分类器的错误率为0，迭代结束。

电子科技大学
University of Electronic Science and Technology of China

AdaBoost优点

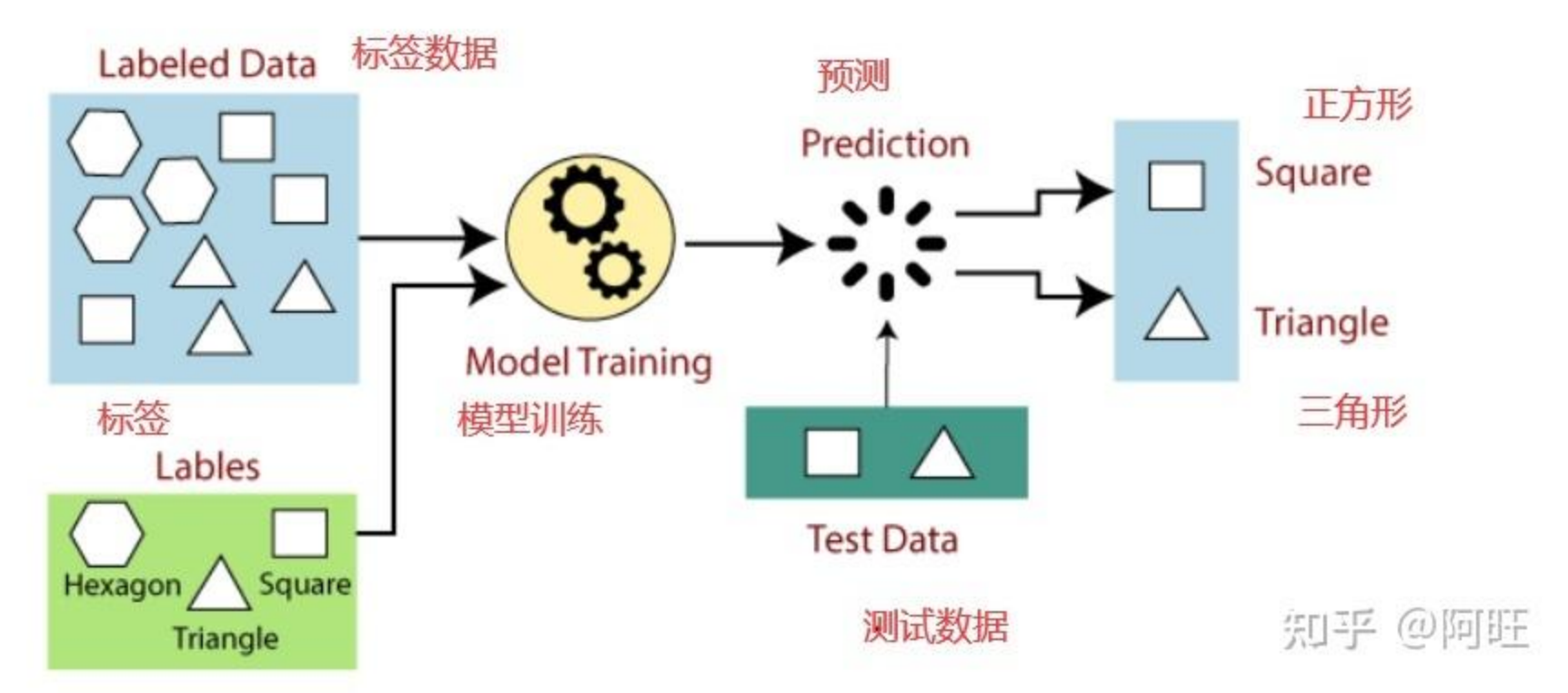
- 准确率得到大幅度提高。
- 分类速度快，且基本不用调参数。
- 过拟合的情况几乎不会出现。
- 在构建子分类器时有多种方法可以使用。
- 方法简单，容易理解和掌握且不用做特征分类。

6.3 支持向量机

支持向量机（Support Vector Machine, SVM）是一类按监督学习（supervised learning）方式对数据进行二元分类的广义线性分类器（generalized linear classifier）。

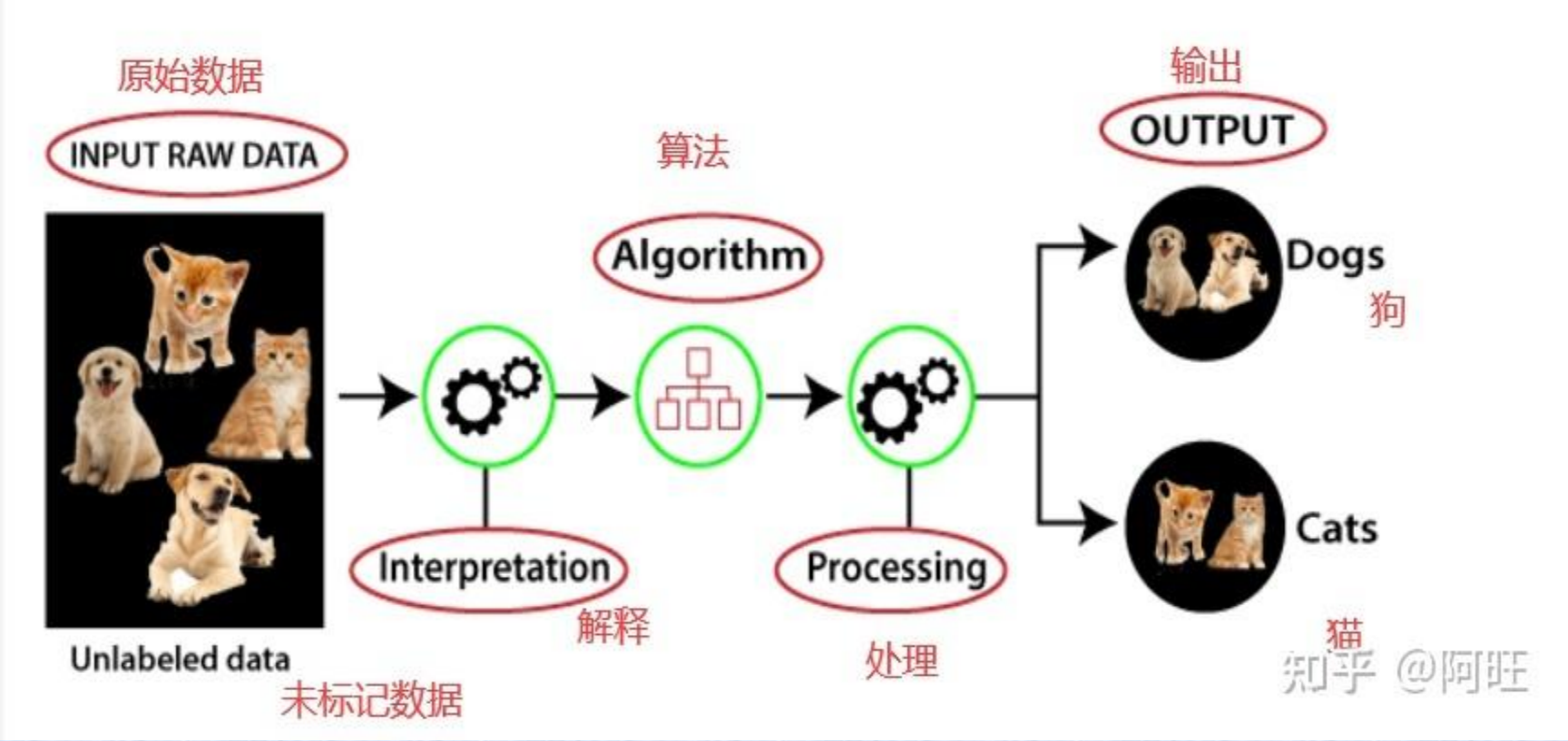
监督学习与无监督学习

监督学习是机器学习的类型，其中机器使用“标记好”（标签）的训练数据进行训练，并基于该数据，机器预测输出。标记的数据意味着一些输入数据已经用正确的输出标记。

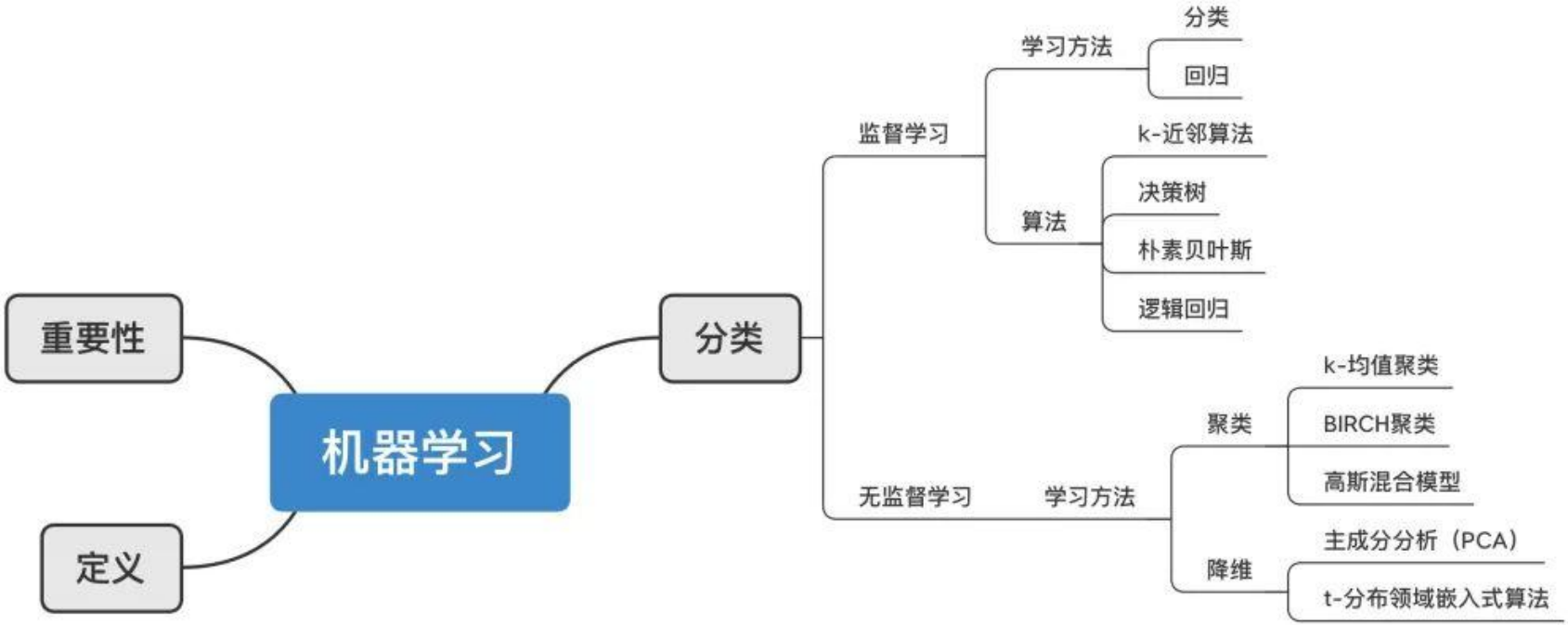


监督学习与无监督学习

无监督学习是机器学习的一种类型，模型使用未标记的数据集进行训练，允许在没有任何监督的情况下对数据进行操作，目标是找到数据集的底层结构，根据相似性对数据进行分组。



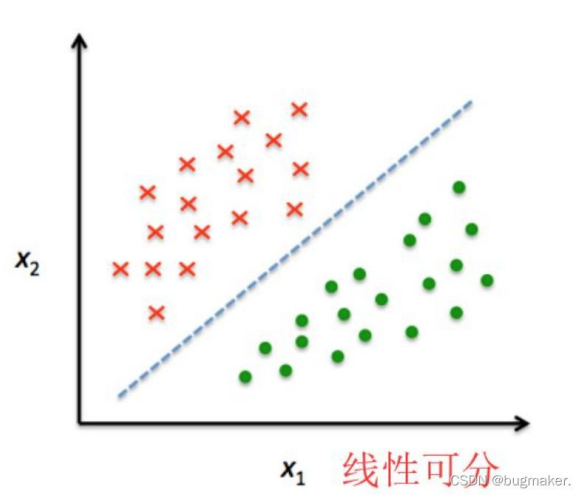
监督学习与无监督学习

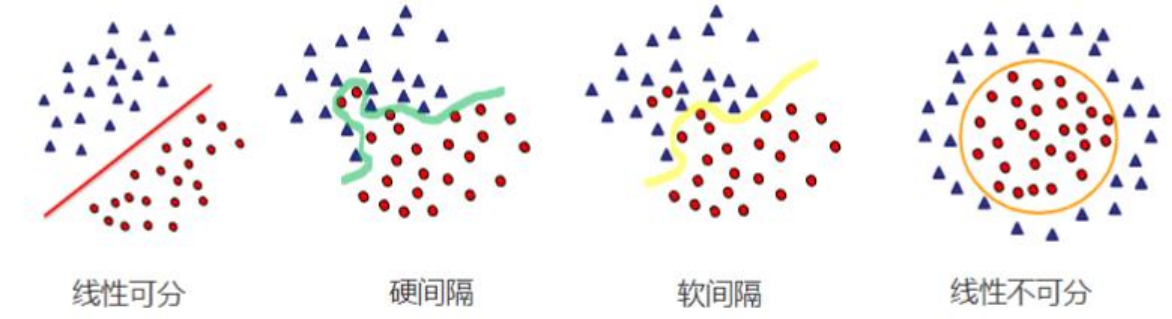


6.3 支持向量机

线性可分性：对于二维空间，如果我们可以找到一条线，将两个不同类别的样本划分开来，我们就说这个样本集是线性可分的。数学定义是：

D0 和 D1 是n维欧氏空间中的两个点集。如果存在n维向量w和实数 b，使得所有属于D0的点 x_i 都有 $wx_i + b > 0$，而对于所有属于 D1 的点 x_j 则有 $wx_j + b < 0$，则称 D0 和 D1 线性可分。





6.3 支持向量机

在三维空间中我们可以找一个面将不同类别的样本划分开来。超过三维的曲面则称为超平面，超平面的公式为：

$$w^T x + b = 0 \quad \text{其中 } w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}, \ b \text{为常数}$$

多维空间的线性可分性可以表述为：

对所有的 $\{(x_i, y_i)\}$ $\exists (w, b)$ 使得，对 $\forall_{i=1\sim N}$ 有

当 $y_i = 1$ 时，$w^T x_i + b \geq 0$

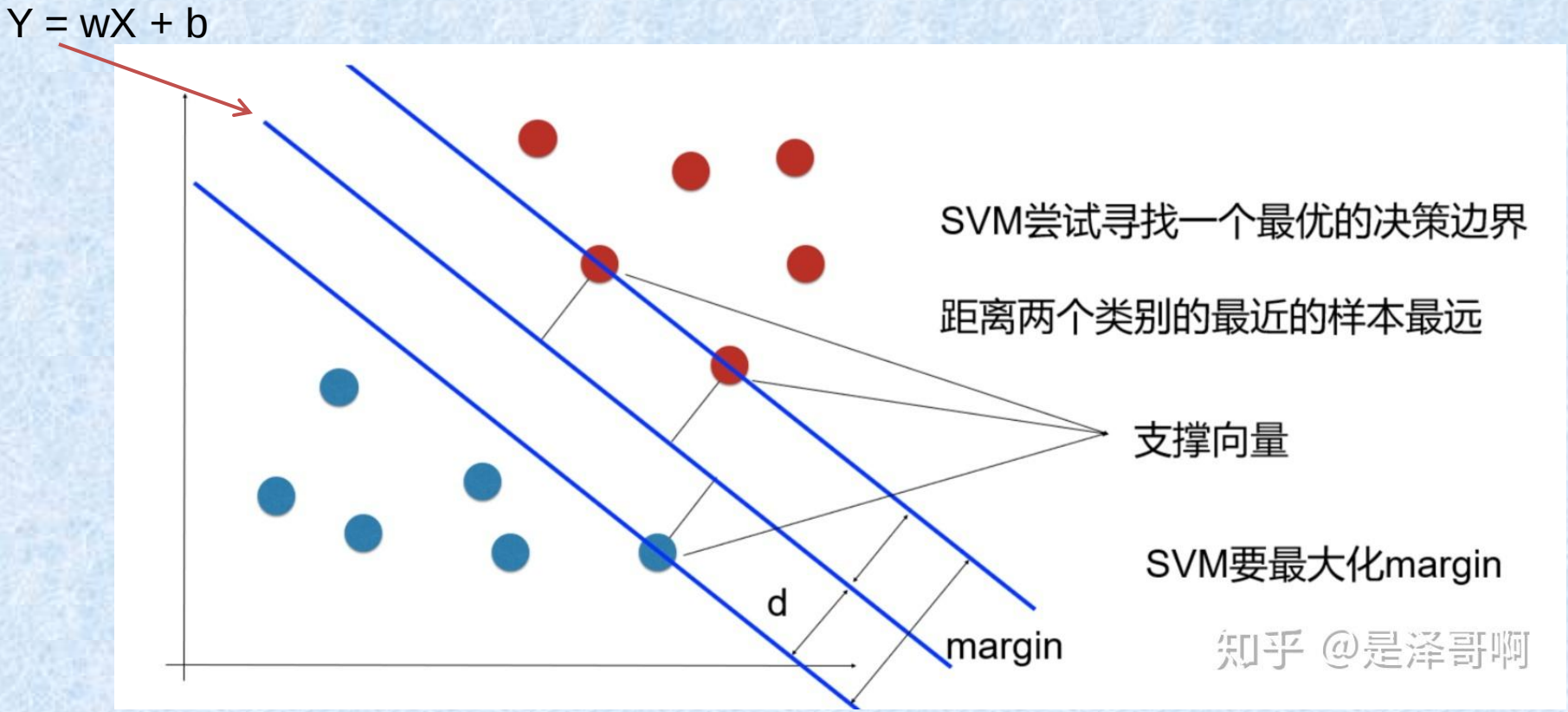
当 $y_i = -1$ 时，$w^T x_i + b < 0$

> 等价于 $y_i(w^T x_i + b) \geq 0$

6.3 支持向量机

最佳超平面：以最大间隔把两类样本分开的超平面

- 两类样本分别分割在该超平面的两侧
- 两侧距离超平面最近的样本点到超平面的距离最大化

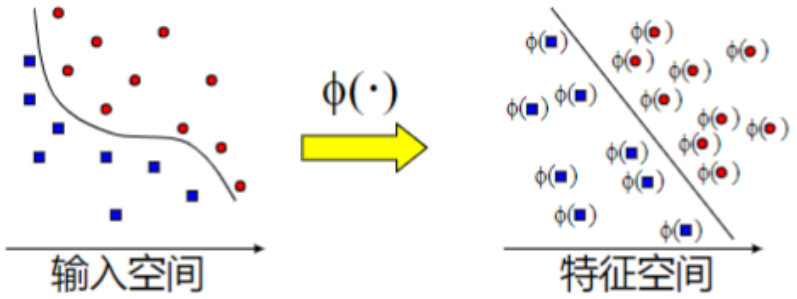


6.3 支持向量机

SVM算法：求解出能够划分训练数据集并且几何间隔最大的分离超平面

- 对于线性可分的数据集来说，分离超平面有无穷多个(即感知机)，但是几何间隔最大的分离超平面却是唯一的
- 软间隔松弛变量
- 核函数：非线性类型数据通常是二维平面不可分，需要通过一个函数将低维数据映射到高维空间，从而使得数据在高维空间能够区分，达到数据分类或回归的目的，实现这一目标的函数称为核函数。

用核函数来替换原来的内积。

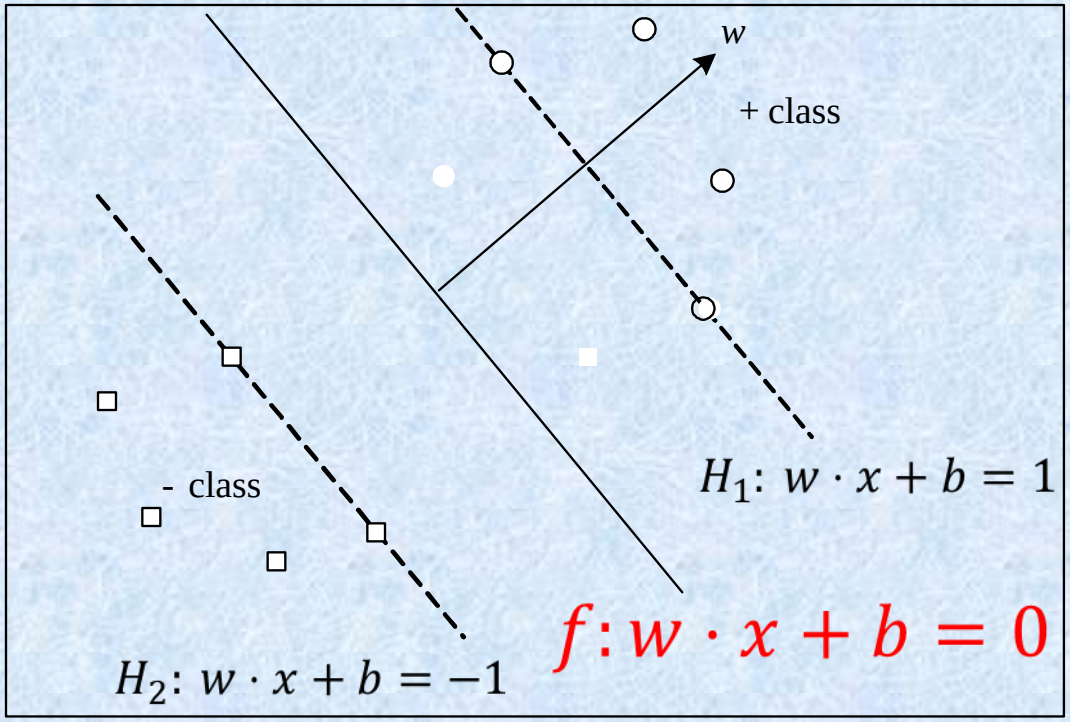


即通过一个非线性转换后的两个样本间的内积。具体地，$K(x,z)$是一个核函数，或正定核，意味着存在一个从输入空间到特征空间的映射，对于任意空间输入的x,z有:

$$K(x,z)=\phi(x)\cdot\phi(z)$$

6.3 支持向量机

- 适用于小样本、高维模式的识别
- 基于结构风险最小化：经验风险和置信区间的折衷

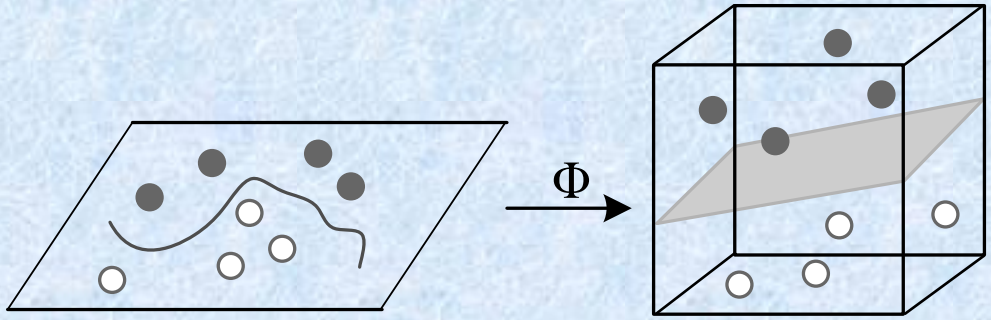


根据给定的训练集$S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$，$N$是样本点个数，$x_i$是$n$维空间的向量，$y_i = \{-1, 1\}$

$$g(x) = \text{sgn}(f(x)) = \text{sgn}(w \cdot x + b)$$

非线性SVM

- $x: \rightarrow \phi(x)$



- 引入核函数$K(x_1, x_2) = (\phi(x_1) \cdot \phi(x_2))$，形成决策面

$$y = \mathrm{sgn}(f(x)) = \mathrm{sgn}(w \cdot \phi(x) + b) = \mathrm{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i K(x, x_i) + b\right)$$

- Largrange乘子α和偏置b，通过二次规划求解。

多项式核: $K(x, y) = \left((x \cdot y) + c\right)^d$

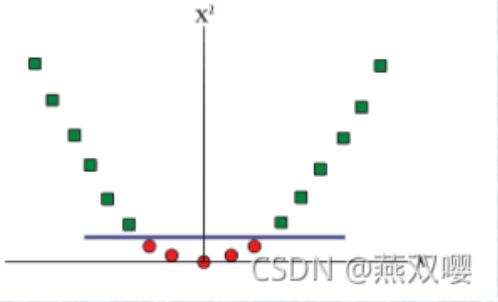
径向基核: $K(x, y) = \mathrm{e}^{\left(-\frac{||x-y||^2}{2\sigma^2}\right)}$

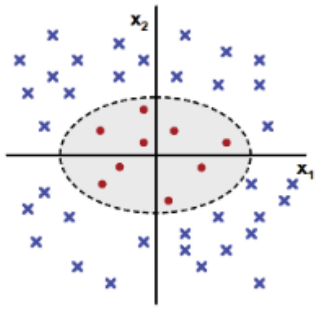
样条核: $K(x, y) = B_{2n+1}(x - y)$

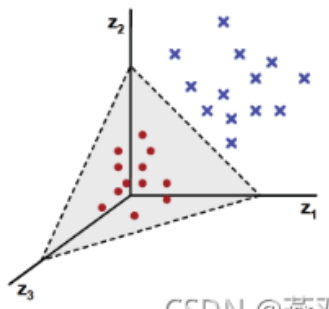
Sigmoid核: $K(x, y) = \tanh(\rho(x \cdot y) + b)$

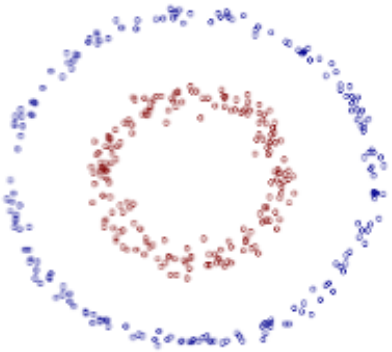
电子科技大学
University of Electronic Science and Technology of China

核函数

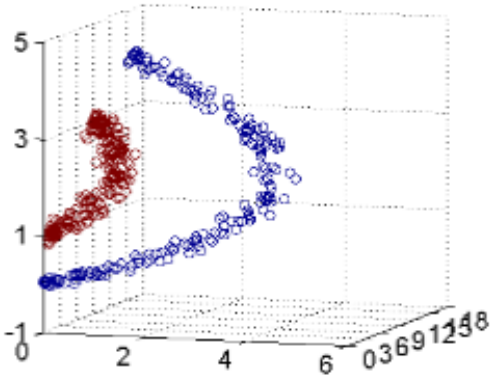








线性不可分



高维下线性可分

核函数

常用的核函数:

线性核函数: $K(X_i, X_j) = X_i \cdot X_j$ （就是最初公式里的内积）

多项式核函数: $K(X_i, X_j) = (\gamma X_i \cdot X_j + \gamma)^d, \gamma \geq 0$

高斯径向基核函数(RBF):

$$K(X_i, X_j) = \exp\left(-\frac{d(X_i, X_j)^2}{2\sigma^2}\right) = \exp(-\text{gamma} \cdot d(X_i, X_j)^2)$$

该核函数(RBF)是应用最广的一个，无论大样本还是小样本都有比较好的性能，而且其相对于多项式核函数参数要少，因此大多数情况下在不知道用什么核函数的时候，优先使用高斯核函数.

S型核函数: $K(X_i, X_j) = \tanh(\gamma x_i \cdot x_j + r)$

关于RBF函数参数gamma的选取:

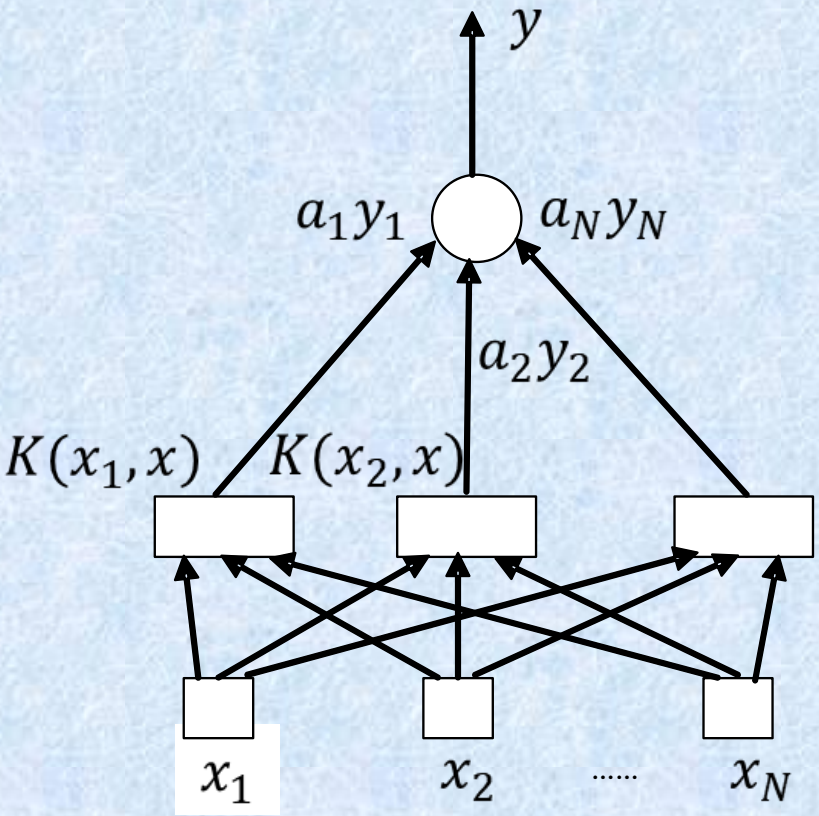
Gamma值选的过大，带高斯核的SVM可以模拟任何非线性数据。存在训练准确率很高，但是测试准确率不高的可能,即过拟合。 Gamma值选的过小,无法在测试集上得到特别高的准确率，也会影响测试集的准确率。

拓展：逻辑回归

- $h(x) = \frac{1}{1+e^{-f(x)}} = \frac{1}{1+e^{-w \cdot x - b}}$
- $h(x)$可以把$f(x)$的输出映射到[0,1]区间，当$h(x) > 0.5$时，判定y为正类，$h(x) < 0.5$时，判定y为负类。

拓展：感知机

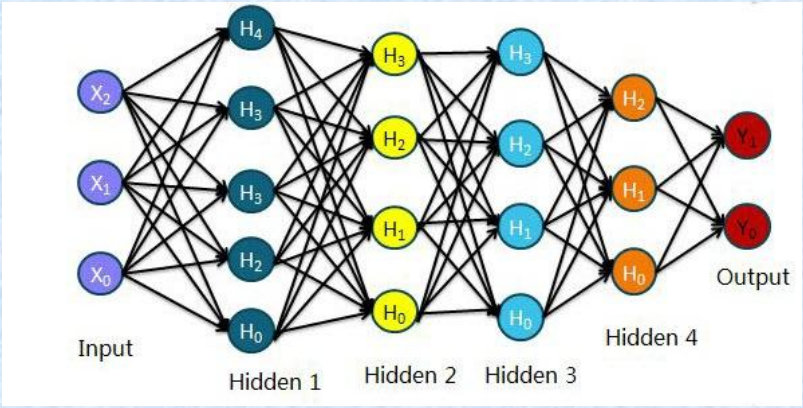
- 支持向量机可以视为一个两层感知机，中间层完成非线性混合，输出层输出样本的类别

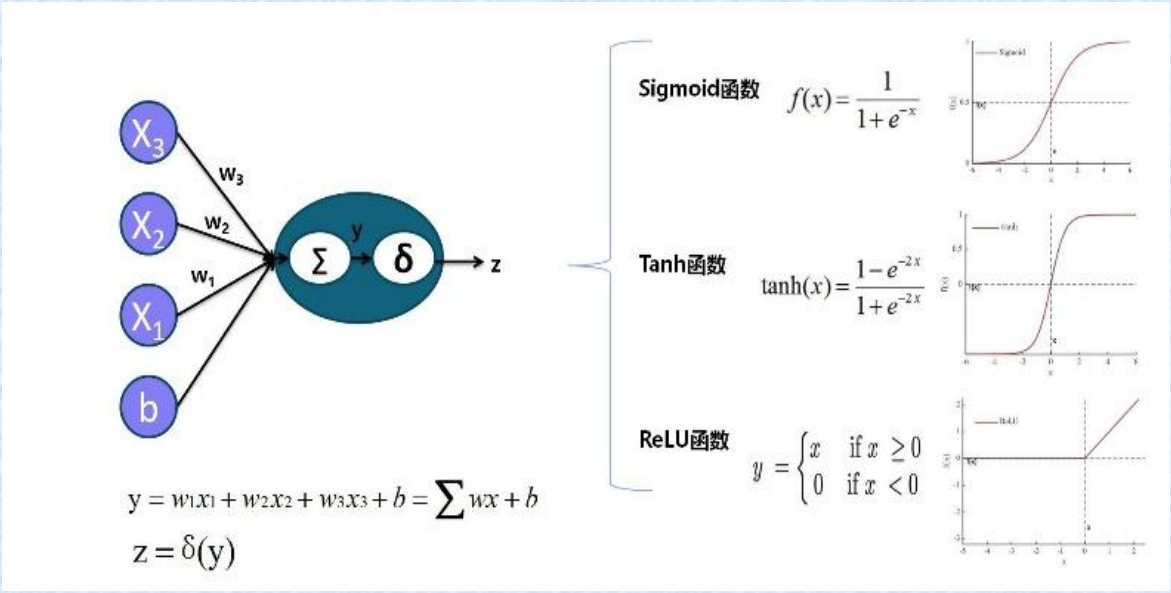


电子科技大学 University of Electronic Science and Technology of China

深度学习

- 深度学习，本质上是一个多层感知机
- 在中间层的节点处使用激活函数
- 增加节点数可以增加维度，即增加线性转换能力；增加层数也就增加激活函数的次数，即增加非线性转换次数

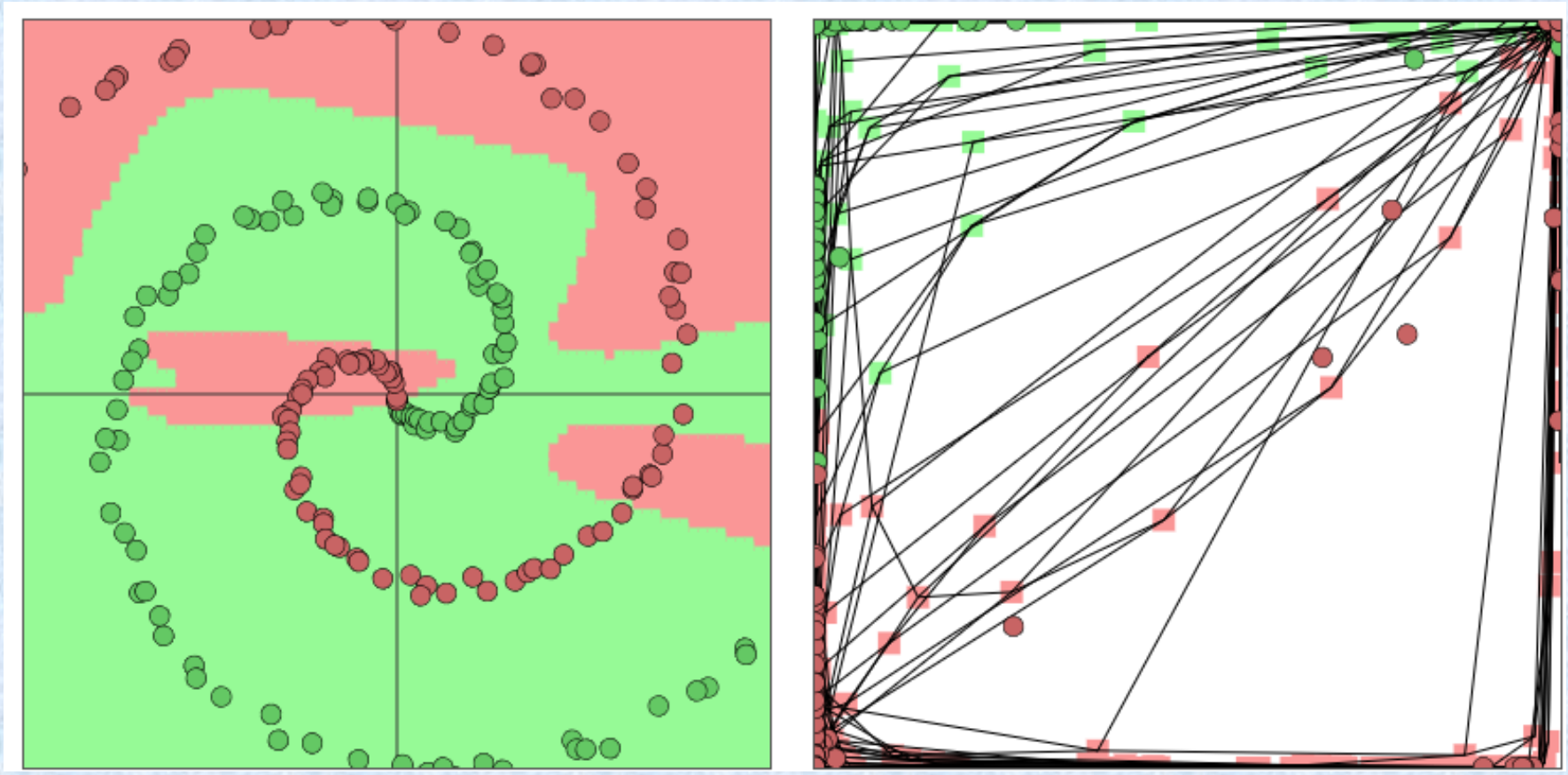




$$y = w_1x_1 + w_2x_2 + w_3x_3 + b = \sum wx + b$$

$$z = \delta(y)$$

空间的转换



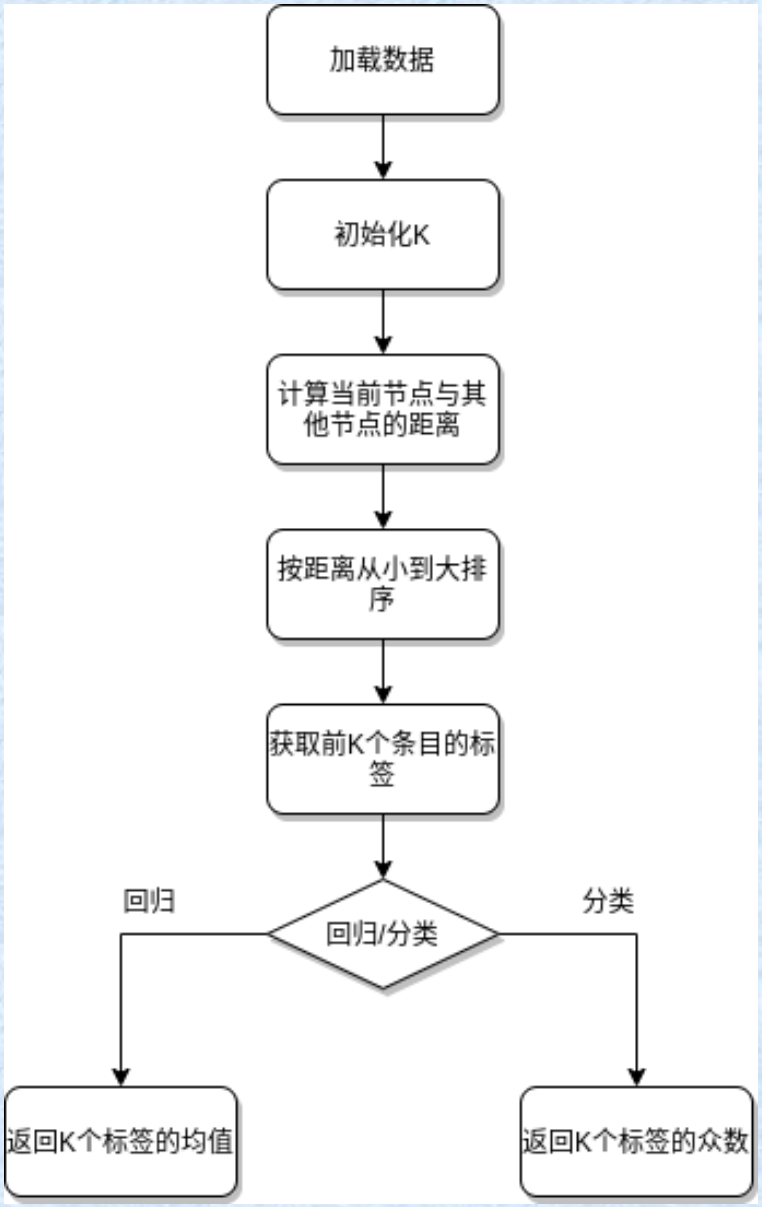
6.4 K邻近算法(KNN)

给定一个训练数据集，对新的输入实例，在训练数据集中找到与该实例最近邻的K个实例，这K个实例的多数属于某个类，就把该输入实例分为这个类。

- K邻近算法查找的是最邻近的K个样本点，是一种用于分类和回归的统计方法
- 通过以某个数据为中心，分析离其最近的K个邻居特征，获得该数据中心可能的特征。
  - 如果你最好的10个朋友中有9个都喜欢湘菜，那么很可能你也喜欢湘菜。

算法流程

- 计算待测样本与每个训练样本的距离。
- 计算并获得待测样本附近的K个样本。
- 根据分类决策规则（如多数表决）决定待测样本的类别。



示例：电影观众兴趣发现

- A、B、C、D、E、F六个观众，互不认识

电影名称	电影类型	观众
《寒战2》	动作	A、C、D
《变形金刚3》	科幻	A、B、C
《大鱼海棠》	动画	E、F
《独立日：卷土重来》	科幻	A、C、E
《惊天魔盗团2》	动作	B、D、F
《海底总动员2》	动画	D、E、B

- 电影院新上映《星球大战：第八部》，现在观众B、D、E已经表现出浓厚的兴趣购票观看，但是缺乏其他观众的观影兴趣，因此电影院试图通过历史数据了解其他用户，以便安排播放档期。

观众	电影
A	《寒战2》、《变形金刚3》、《独立日：卷土重来》
B	《变形金刚3》、《惊天魔盗团2》、《海底总动员2》
C	《寒战2》、《变形金刚3》、《独立日：卷土重来》
D	《寒战2》、《惊天魔盗团2》、《海底总动员2》
E	《大鱼海棠》、《独立日：卷土重来》、《海底总动员2》
F	《大鱼海棠》、《惊天魔盗团2》

- 计算观众的**余弦相似度**
  - 例如A和B，并集{寒战2、变形金刚3、独立日：卷土重来、惊天魔盗团2、海底总动员2}，那么A=(1, 1, 1, 0, 0)，B=(0, 1, 0, 1, 1)，A、B的余弦相似度为0.33

	观众A	观众B	观众C	观众C	观众E	观众F
观众A	1	0.33	1	0.33	0.33	0
观众B	0.33	1	0.33	0.66	0.33	0.41
观众C	1	0.33	1	0.33	0.33	0
观众D	0.33	0.66	0.33	1	0.33	0.41
观众E	0.33	0.33	0.33	0.33	1	0.41
观众F	0	0.41	0	0.41	0.41	1

- 约定两者的相似度不小于0.33时，视为相关用户

观众	相似观众
观众A	观众B（0.33）、观众D（0.33）、观众E（0.33）
观众B	观众D（0.66）、观众F（0.41）、观众A（0.33）、观众C（0.33）、观众E（0.33）
观众C	观众B（0.33）、观众D（0.33）、观众E（0.33）
观众D	观众B（0.66）、观众F（0.41）、观众A（0.33）、观众C（0.33）、观众E（0.33）
观众E	观众F（0.41）、观众A（0.33）、观众B（0.33）、观众C（0.33）、观众D（0.33）
观众F	观众B（0.41）、观众D（0.41）、观众E（0.41）

- B、D、E对《星球大战》是感兴趣的，只考察A、C、F

观众	K=3邻居	兴趣值
A	观众B（0.33）、观众D（0.33）、观众E（0.33）	$(0.33+0.33+0.33)\div 3=0.33$
C	观众B（0.33）、观众D（0.33）、观众E（0.33）	$(0.33+0.33+0.33)\div 3=0.33$
F	观众B（0.41）、观众D（0.41）、观众E（0.41）	$(0.41+0.41+0.41)\div 3=0.41$

- 观众F是比较有可能去的，观众A和观众C也有一定的可能性去。

距离度量

欧式距离： $d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

曼哈顿距离： $d_{12} = |x_1 - x_2| + |y_1 - y_2|$

切比雪夫距离： $d_{12} = \max(|x_1 - x_2|, |y_1 - y_2|)$

余弦距离： $\cos\theta = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2}\sqrt{x_2^2 + y_2^2}}$

Jaccard相似系数： $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$

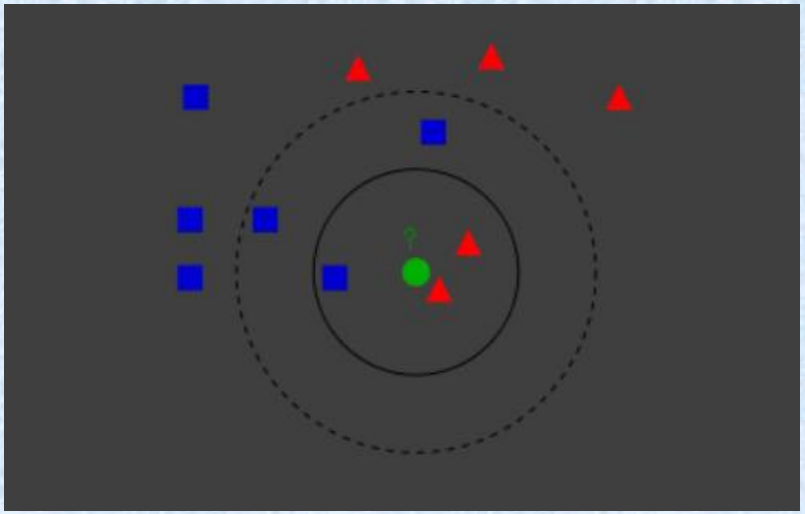
相关系数： $\rho_{XY} = \frac{Cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}} = \frac{E((X - EX)(Y - EY))}{\sqrt{D(X)}\sqrt{D(Y)}}$

K值选择

如果选择较小的k值，就相当于用较小的邻域中的训练实例进行预测，“学习”的近似误差(approximation error)会减小，但缺点是“学习”的估计误差(estimation error)会增大，预测结果会对近邻的实例点非常敏感。如果邻近的实例点恰巧是噪声，预测就会出错。换句话说，k值的减小就意味着整体模型变得复杂，容易发生过拟合.

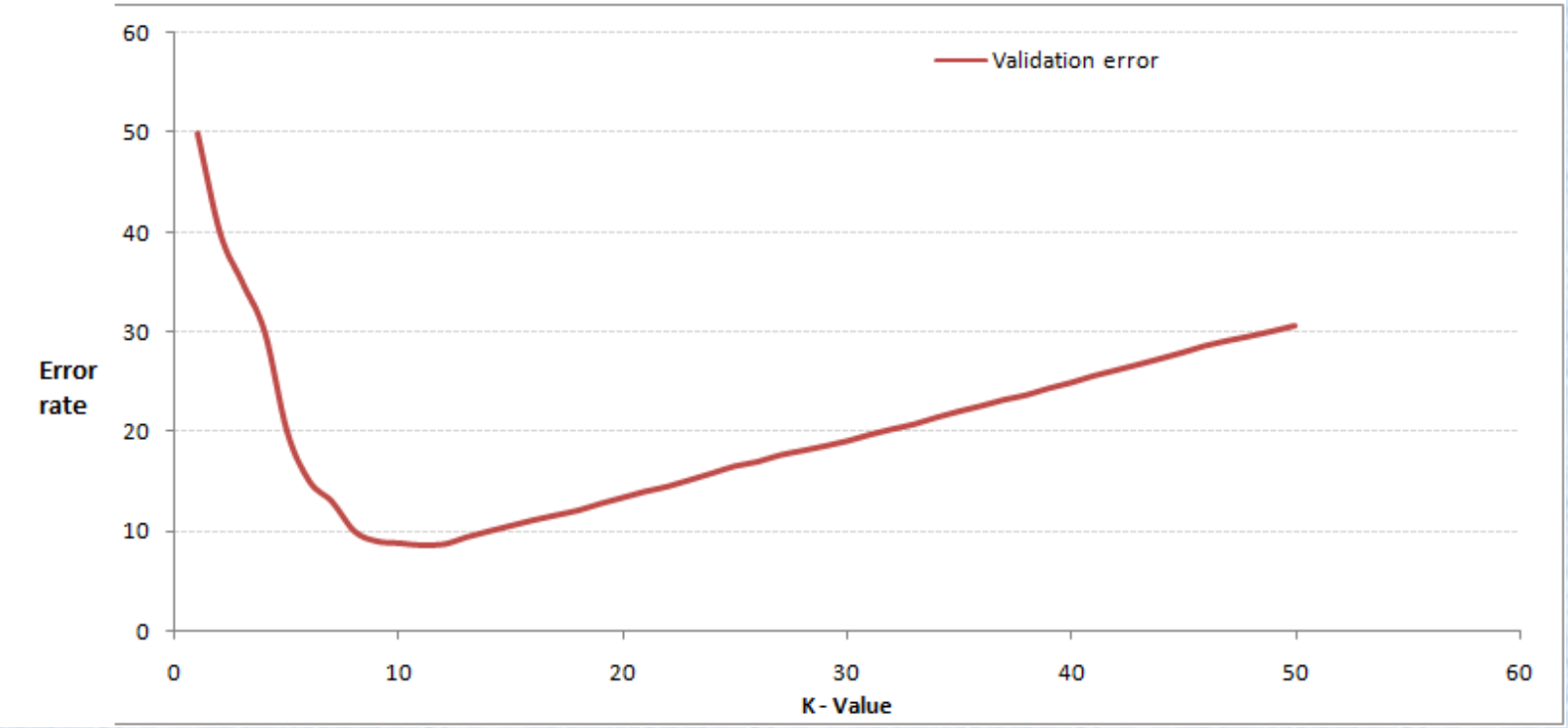
如果选择较大的k值，就相当于用较大邻域中的训练实例进行预测，其优点是可以减少学习的估计误差，但缺点是学习的近似误差会增大。这时与输入实例较远的(不相似的)训练实例也会对预测起作用，使预测发生错误，k 值的增大就意味着整体的模型变得简单。

右图，K=3 与 K=5 导致不同的输出结果



K值选择

在应用中，k值一般取一个较小的数值，通常采用交叉验证法来选取最优的k值。如下图所示，当增大k时，一般错误率会先降低，因为有周围更多的样本可以借鉴了，分类效果会变好。当k增大到一定程度使，分类模型变得越来越简单，错误率会更高。

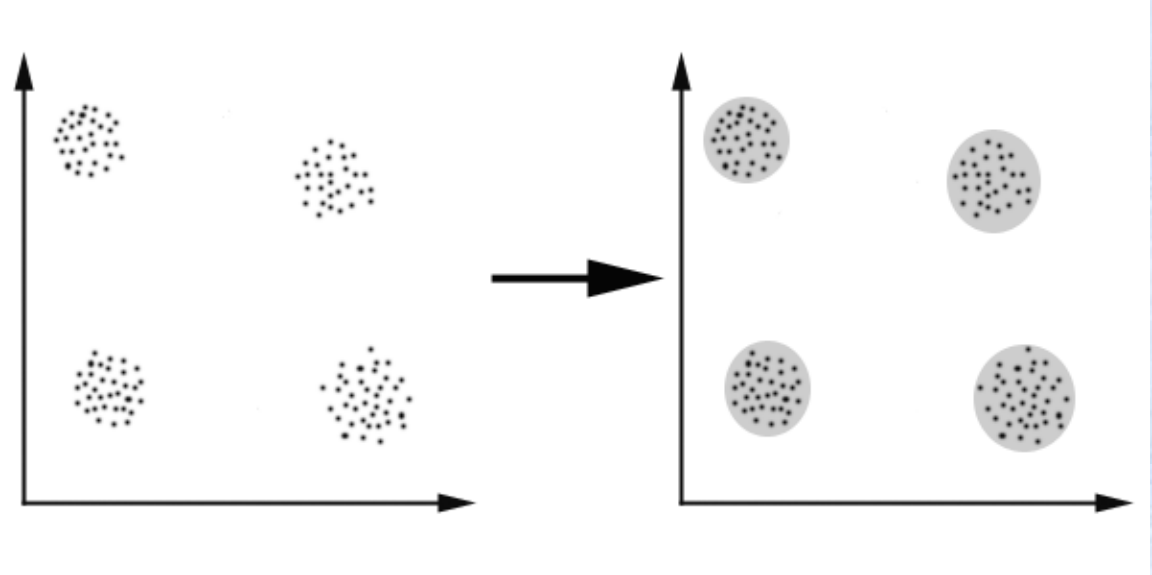


KNN的缺点

- 不平衡样本
- 计算量相对较大
- K 值的设定对算法的结果有较大的影响
- 解决途径：在实际应用过程中将类别典型的样本纳入样本库中。

6.5 K-Means聚类算法

- 聚类方式
  - 自上而下
  - 自下而上
- 核心思想：人以类聚，物以群分



K-Means原理

K-Means算法著名的牧师-村民模型

有K个牧师去郊区布道，一开始牧师们随意选了几个布道点，并且把这几个布道点的情况公告给了郊区所有的居民，于是每个居民到离自己家最近的布道点去听课。（随机选择K个中心点，第一次聚类）

听课之后，大家觉得布道点距离太远了，于是每个牧师统计了一下自己课上所有的居民的地址，搬到了所有地址的中心地带，并且在海报上更新了自己布道点的位置。（每个子类计算自己类的中心点（取距离平均值），更新发布K个新的中心点）

牧师每一次移动不可能离所有人都更近，有的人发现A牧师移动以后自己还不如去B牧师处听课更近，于是每个居民又去了离自己最近的布道点……（根据新的中心点重新聚类）

就这样，牧师每个礼拜更新自己的位置，居民根据自己的情况选择布道点，折腾几周后，最终稳定了下来。（多轮迭代，直到稳定）

K-Means算法流程

1. 从数据点集合中，随机选择K个点作为种子中心点
2. 对其余的数据点，依次判断它与K个中心点的距离，距离最近的表明它属于这个聚类
3. 重新计算，以新的类集合的平均值作为类的中心点
4. 如果新计算出来的中心点和原来的中心点之间的距离小于某一个设置的阈值(表示中心点位置变化不大，趋于稳定，或者说收敛)，我们可以认为聚类已经达到期望的结果，算法终止
5. 如果新质心和原质心距离变化大，则需要迭代2-4步骤

示例：新闻聚类

- “中国女足绝对主力伤别奥运 18+4名单将做调整”
- “雷军否认小米手机耍猴搞饥饿营销：绝对是误解”
- “中国两名南苏丹维和牺牲战士灵柩运抵乌干达”
- “惧怕赛卡！温网亚军拉奥尼奥宣布退出里约奥运”
- “手机市场陷入滞胀 部分中小品牌‘死’在上半年”
- “抗洪战士刘景泰失联7天 其母：战士们辛苦别搜了”
- “阿根廷男足奥运名单：马竞主帅之子 多名大将缺阵”
- “网购手机中现陌生人照片疑为翻新机 商家：进货渠道正规”
- “中国赴南苏丹维和步兵营为牺牲战士举行告别仪式”

分词

- “中国 女足 绝对 主力 伤别 奥运 18+4 名单 做 调整”
- “雷军 否认 小米 手机 耍猴 搞 饥饿 营销 绝对 误解”
- “中国 两名 南苏丹 维和 牺牲 战士 灵柩 运抵 乌干达”
- “惧怕 赛卡 温网 亚军 拉奥尼奥 宣布 退出 里约 奥运”
- “手机 市场 陷入 滞胀 部分 中小品牌 死 上半年”
- “抗洪 战士 刘景泰 失联 7天 其母 战士们 辛苦 别 搜”
- “阿根廷 男足 奥运 名单 马竞 主帅 之 子 多名 大将 缺阵”
- “网购 手机 中 现 陌生人 照片 疑为 翻新机 商家 进货 渠道 正规”
- “中国 赴 南苏丹 维和 步兵营 为 牺牲 战士 举行 告别 仪式”

- 第一步，设定$K=3$，即确定聚类个数。
  - 距离计算可以使用欧氏距离，但是对于新闻标题的距离，实质是句子的相似度，句子之间的相似度越高，则距离越小，因此句子的相似度可使用余弦相似性进行计算。

- 第二步，对中心点进行调整，并不断迭代计算。
  - 在迭代计算之前，需要假定初始状态下三个聚类的中心点位置，通常是随机句子中的三句，分别为聚类K_1, K_2, K_3的中心。
  - 将其他句子分别与三个初始类簇中心点计算相似度，例如，第m个句子S_m，分别计算其与K_1, K_2, K_3的中心句子相似度d_{m1}, d_{m2}, d_{m3}，若值d_{m1}最小，则说明在本次迭代中S_m属于聚类K_1。
  - 完成一次迭代之后，需要重新确定聚类K_1, K_2, K_3的中心句子。

第二步（续1）

- 确定一个聚类的中心句子的方式，是根据类中句子之间的相似度，设$\frac{\sum_{j=1}^{n} d_{ji}}{n}$表示第$K$个类簇中所有句子与类簇的第$i$个句子的平均相似度，$n$表示当前迭代过程中第$K$个类簇的句子数，取最小平均相似度的句子作为新的类簇中心句。

- 迭代，直至收敛。

聚类结果

聚 类	新闻标题
第一聚类簇	中国女足绝对主力伤别奥运 18+4名单将做调整 惧怕寨卡！温网亚军拉奥尼奥宣布退出里约奥运 阿根廷男足奥运名单：马竞主帅之子 多名大将缺阵
第二聚类簇	雷军否认小米手机耍猴搞饥饿营销：绝对是误解 手机市场陷入滞胀 部分中小品牌 ‘死’ 在上半年 网购手机中现陌生人照片疑为翻新机 商家：进货渠道正规
第三聚类簇	中国两名南苏丹维和牺牲战士灵柩运抵乌干达 抗洪战士刘景泰失联7天 其母：战士们辛苦别搜了 中国赴南苏丹维和步兵营为牺牲战士举行告别仪式

K-Means缺点

- 对异常值、摇摆值比较敏感，导致收敛变慢。
- 非常不适合分布均匀、数据界限不明晰的聚类。
- 初始中心点的选择对迭代次数影响较大。K-Means++算法，改进了初始点的选择。
- 需要提前确定聚类簇的值。

6.6 最大期望算法

- 最大期望算法（Expectation Maximization，EM），它是在概率模型中寻找参数最大似然估计或者最大后验估计的算法，其中概率模型依赖于无法观测的隐藏变量。
- 本节以一个抛硬币的实例来说明EM算法的原理和步骤。

示例

- 现在有两枚硬币1和2，随机抛掷后正面朝上概率分别为P_1和P_2，其真实值分别是0.4和0.5

硬币	结果	统计
1	正正反正反	3正-2反
2	反反正正反	2正-3反
1	正反反反反	1正-4反
2	正反反正正	3正-2反
1	反正正反反	2正-3反

- 上面是实验观测值，目标就是通过观测值推断P_1和P_2

示例

- 每次取的硬币已知：

$$P_1 = \frac{3+1+2}{15} = 0.4,\quad P_2 = \frac{2+3}{10} = 0.5$$

- 但每次取的硬币未知，怎么估算？

硬币	结果	统计
1	正正反正反	3正-2反
2	反反正正反	2正-3反
1	正反反反反	1正-4反
2	正反反正正	3正-2反
1	反正正反反	2正-3反

EM算法

- 加入隐含变量z，可以把它认为是一个5维的向量$z = (z_1, z_2, z_3, z_4, z_5)$，代表每次投掷时所使用的硬币。
  - 比如z_1就代表第一轮投掷时所使用的是硬币1还是2
- 必须先估计出z，然后才能进一步估计P_1和P_2。

- 假设$P_1 = 0.2$和$P_2 = 0.7$，可以计算

轮数	若是硬币1	若是硬币2	最有可能的硬币
1	0.00512	0.03087	硬币2
2	0.02048	0.01323	硬币1
3	0.08192	0.00567	硬币1
4	0.00512	0.03087	硬币2
5	0.02048	0.01323	硬币1

- 上表通过极大似然估计处一个估计序列$z = (2,1,1,2,1)$，这里估计出的是最有可能的z序列。
- 在这个序列下再按照极大似然估计新的

$$P_1 = \frac{2+1+2}{15} = 0.33\,,\quad P_2 = \frac{3+3}{10} = 0.6$$

更靠近真实值

- 如果不用最有可能的z序列，而是用z的分布，例如：

$$z_1 = \frac{0.00512}{0.00512 + 0.03087} = 0.14$$

轮数	z_i = 硬币1	z_i = 硬币2
1	0.14	0.86
2	0.61	0.39
3	0.94	0.06
4	0.14	0.86
5	0.61	0.39

- 估计出z的概率分布，称为E步

E步

- E步是根据参数的初始值或上一次迭代的模型参数来计算出的因变量（θ）的后验概率（条件概率），其实就是隐变量的期望值，来作为隐变量的当前估计值：

$$Q_i\left(z^{(i)}\right) = p\left(z^{(i)} \mid x^{(i)}; \theta\right)$$

- 其中$Q_i\left(z^{(i)}\right)$表示第i轮z的分布，$x^{(i)}$表示第i轮的观察量。

- $z_i = 1$时正反面分布

轮数	正面	反面
1	0.42	0.28
2	1.22	1.83
3	0.94	3.76
4	0.42	0.28
5	1.22	1.93
总计	4.22	7.98

- 估计P_1：$P_1 = \frac{4.22}{4.22+7.98} = 0.35$
- 新估计出的P_1要更加接近0.4，原因是使用了所有抛掷的数据，而不是部分的数据。

M步

- 根据E步中求出z的概率分布，依据最大似然概率法则去估计P_1和P_2，称为M步。
- M步就是最大化似然函数从而获得新的参数值$\theta = \{P_1, P_2\}$:

$$\theta = arg\max_{\theta} \sum_{i} \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p\left(x^{(i)}, z^{(i)}; \theta\right)}{Q_i(z^{(i)})}$$

- 然后用估计出的P_1和P_2再去估计z，迭代多次后P_1和P_2越来越接近真实值。

EM算法优缺点

- EM算法可以应用于聚类或参数估计，计算的结果稳定准确，数学证明该算法能收敛。
- EM算法对初始化数据敏感，计算较为复杂，收敛较慢，是局部最优的算法。