

A计划

AI人才专业力认证



京东大学
JD UNIVERSITY



JD AI Academy

A计划-中阶课程介绍及开课计划



开课形式	课程模块	课程名称	开课时间 休息日 (9:00-17:30)	课程简介	学习时间
线下面授	模块一	《机器学习》	12月8日 12月9日 12月16日	1、机器学习概述 2、分类模型评估 3、逻辑回归LR 4、支持向量机 5、KNN算法 6、决策树(ID3, C4.5) 7、贝叶斯算法 8、集成算法	3天面授
	模块二	《数据挖掘》	12月22日 12月23日	1、回归 2、聚类 3、数据挖掘算法	2天面授
	模块三	《深度学习》	1月12日 1月19日 1月20日	1、神经网络基础 2、卷积神经网络 3、循环神经网络 4、自然语言处理 5、深度学习前沿	3天面授
	项目答辩		待确认		



- 1.空杯心态；
- 2.积极参与；
- 3.手机静音；
- 4.不看电脑；
- 5.接打电话到室外。

贝叶斯算法

司博

2018-12-16

目录

CONTENTS

- 贝叶斯定理
- 朴素贝叶斯算法
- 朴素贝叶斯案例：文本分类

PART 01

01

贝叶斯定理

$$P(\omega_i|\mathbf{x}) = \frac{P(\mathbf{x}, \omega_i)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|\omega_i)P(\omega_i)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|\omega_i)P(\omega_i)}{\sum_{j=1}^n P(\mathbf{x}|\omega_j)P(\omega_j)}, i = 1, 2, \dots, n$$

癌症的例子

- ω_1 : 癌症; ω_2 : 正常
- $P(\omega_1) = 0.008$; $P(\omega_2) = 0.992$
- 实验测试输出: + vs. -
- $P(+|\omega_1) = 0.98$; $P(-|\omega_1) = 0.02$
- $P(+|\omega_2) = 0.03$; $P(-|\omega_2) = 0.97$
- 某人被检测出阳性
- 这真的是传说中的末日?

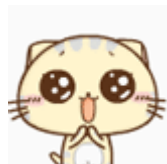


$$P(\omega_1 | +) \propto P(+ | \omega_1)P(\omega_1) = 0.98 \times 0.008 = 0.0078$$

$$P(\omega_2 | +) \propto P(+ | \omega_2)P(\omega_2) = 0.03 \times 0.992 = 0.0298$$

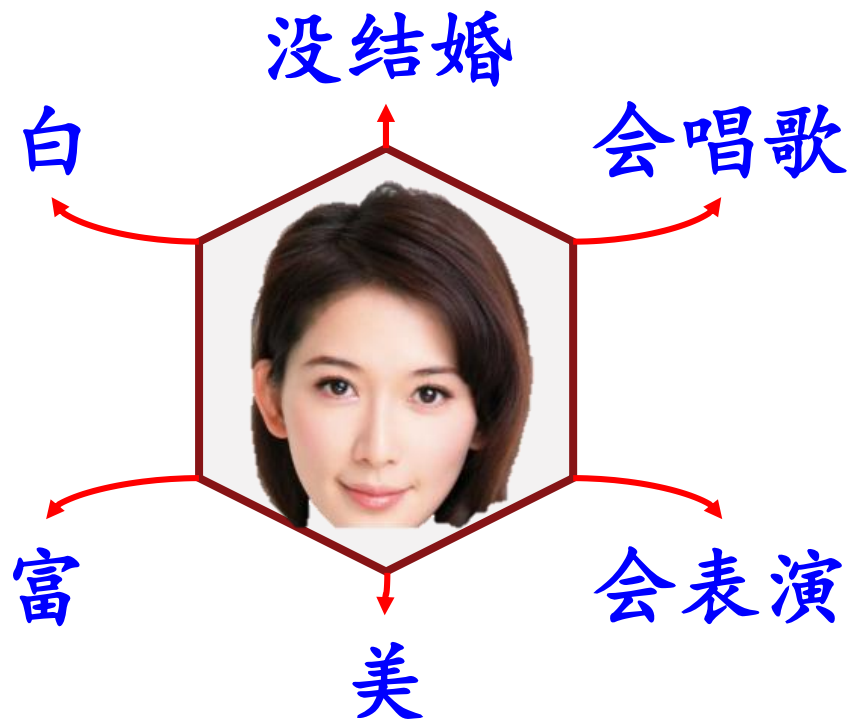
$$P(\omega_1 | +) < P(\omega_2 | +)$$

$$P(\omega_1 | +) = \frac{0.0078}{0.0078 + 0.0298} = 0.21 \gg P(\omega_1)$$



学了贝叶斯，从此变暖男！

- 我们的世界是多元的，样本的属性也是**多元**的。
- 通常情况下，这些多元的属性之间是**接近相互独立**的。



- 计算后验概率。



- 多元贝叶斯公式

$$P(\omega_i | \mathbf{x}_1, \dots, \mathbf{x}_n) = \frac{P(\mathbf{x}_1, \dots, \mathbf{x}_n | \omega_i) P(\omega_i)}{P(\mathbf{x}_1, \dots, \mathbf{x}_n)}$$

属性接近相互独立



属性接近相互独立



属性相互独立

- 多元贝叶斯公式

$$P(\omega_i | \mathbf{x}_1, \dots, \mathbf{x}_n) = \frac{P(\mathbf{x}_1, \dots, \mathbf{x}_n | \omega_i) P(\omega_i)}{P(\mathbf{x}_1, \dots, \mathbf{x}_n)}$$

- 独立性假设

$$P(\mathbf{x}_1, \dots, \mathbf{x}_n | \omega_i) = \prod_{j=1}^n P(\mathbf{x}_j | \omega_i)$$

- 如何后验概率最大化？

PART 02

02

朴素贝叶斯算法

由独立可得

$$\begin{aligned} P(\omega_i | \mathbf{x}_1, \dots, \mathbf{x}_n) &= \frac{P(\mathbf{x}_1, \dots, \mathbf{x}_n | \omega_i) P(\omega_i)}{P(\mathbf{x}_1, \dots, \mathbf{x}_n)} \\ &= \frac{\prod_{j=1}^n P(\mathbf{x}_j | \omega_i) P(\omega_i)}{P(\mathbf{x}_1, \dots, \mathbf{x}_n)} \end{aligned}$$

当样本确定时， $P(\mathbf{x}_1, \dots, \mathbf{x}_n)$ 确定，且大于0

最大化后验概率等价于

$$\omega_{NB} = \arg \max_{\omega_i \in \Lambda} P(\omega_i) \prod_{j=1}^n P(\mathbf{x}_j | \omega_i)$$

- 这就是朴素贝叶斯分类器，理论依据为**最大后验假设**。

- 某人打网球的数据

Day	Outlook	Temperature	PlayTennis
D1	Sunny	Hot	No
D2	Sunny	Hot	No
D3	Overcast	Hot	Yes
D4	Rain	Mild	Yes
D5	Rain	Cool	No
D6	Rain	Cool	No
D7	Overcast	Cool	Yes
D8	Sunny	mild	Yes
D9	Sunny	Cool	Yes
D10	Rain	mild	Yes

- 任务：估计在Outlook = Rain, Temperature = Hot情况下，此人是否打网球？

$$\omega_{NB} = \arg \max_{\omega_i \in \{yes, no\}} P(\omega_i) \prod_{j=1}^n P(x_j | \omega_i) =$$

$$\arg \max_{\omega_i \in \{yes, no\}} P(\omega_i) P(Outlook = Rain | \omega_i)$$

$$P(Temperature = Hot | \omega_i)$$

先验概率 (+2) 类条件概率 (+4)

为了计算 ω_{NB} , 需要计算 6 个概率

- 从10个训练例子中估计是否打网球的频率

$$P(\text{PlayTennis} = \text{yes}) = 6/10 = 0.6$$

$$P(\text{PlayTennis} = \text{no}) = 4/10 = 0.4$$

Day	Outlook	Temperature	PlayTennis
D1	Sunny	Hot	No
D2	Sunny	Hot	No
D3	Overcast	Hot	Yes
D4	Rain	Mild	Yes
D5	Rain	Cool	No
D6	Rain	Cool	No
D7	Overcast	Cool	Yes
D8	Sunny	mild	Yes
D9	Sunny	Cool	Yes
D10	Rain	mild	Yes

- 先估计 $\text{Outlook} = \text{Rain}$ 的类条件概率

$$P(\text{Outlook} = \text{Rain} | \text{PlayTennis} = \text{yes}) = 2/6 = 0.33$$

$$P(\text{Outlook} = \text{Rain} | \text{PlayTennis} = \text{no}) = 2/4 = 0.5$$

Day	Outlook	Temperature	PlayTennis
D1	Sunny	Hot	No
D2	Sunny	Hot	No
D3	Overcast	Hot	Yes
D4	Rain	Mild	Yes
D5	Rain	Cool	No
D6	Rain	Cool	No
D7	Overcast	Cool	Yes
D8	Sunny	mild	Yes
D9	Sunny	Cool	Yes
D10	Rain	mild	Yes

- 再估计 $Temperature = Hot$ 的类条件概率

$$P(Temperature = Hot | PlayTennis = yes) = 1/6 = 0.17$$

$$P(Temperature = Hot | PlayTennis = no) = 2/4 = 0.5$$

Day	Outlook	Temperature	PlayTennis
D1	Sunny	Hot	No
D2	Sunny	Hot	No
D3	Overcast	Hot	Yes
D4	Rain	Mild	Yes
D5	Rain	Cool	No
D6	Rain	Cool	No
D7	Overcast	Cool	Yes
D8	Sunny	mild	Yes
D9	Sunny	Cool	Yes
D10	Rain	mild	Yes

- 使用以上的概率， ω_{NB} 计算结果如下：

$$P(yes)P(Rain|yes)P(Hot|yes) = 0.6 \times 0.33 \times 0.17 = 0.033$$

$$P(no)P(Rain|no)P(Hot|no) = 0.4 \times 0.5 \times 0.5 = 0.1$$

- 使用朴素贝叶斯分类器，在Outlook = Rain, Temperature = Hot情况下，认为此人不玩网球。

- 此人不玩网球的概率是多少？
- 对上述计算所得的概率进行归一化，以确保概率之和为**1**，此时我们可计算此人不玩网球的概率为：

$$P(no|Rain, Hot) = \frac{0.1}{0.1 + 0.033} = 0.75$$

试用身高和体重对男女分类 你遇到了什么困难？



- 条件之间的绝对独立性往往很难成立。
- 但朴素贝叶斯在现实中往往表现出良好的性能，我们不需要一定要非常准确地估计概率，我们需要的是准确地进行决策。



原来是这样！

- 假设实际情况：

$$P(\text{Outlook} = \text{Rain} | \text{PlayTennis} = \text{no}) = 0.001$$

训练样本只有10个，此时很可能出现

$$P(\text{Outlook} = \text{Rain} | \text{PlayTennis} = \text{no}) = 0/10 = 0$$

当 $P(\text{Outlook} = \text{Rain} | \text{PlayTennis} = \text{no}) = 0/10 = 0$ 时，由于乘法规则，导致结果直接为0，也就说直接抹除了其他条件的影响。

- 怎么解决？

- 解决办法

$(\text{Outlook} = \text{Rain} | \text{PlayTennis} = \text{no})$

$$\frac{n_c + mp}{n + m}$$

$\text{PlayTennis} = \text{no}$

- m 和 p 都是特定的参数
- 比如：
 - ✓ p 是先验概率
 - ✓ m 样本总数

最是那第一次的朴素
让我的心花开了，就像心花睡醒了一样
回忆让我看清了你的脸庞
不是天真，而是简约
我如潮般思念
你会不会忽然地出现，在我的身后
回首，留恋……

PART 03

03

朴素贝叶斯案例：文本分类

感兴趣的？ 不感兴趣的？



政治？ 娱乐？ 体育？

- 文本类别：
 体育类、政治类、娱乐类.....

- 计算先验概率。如何算？

$$P(\omega_i) = \frac{\text{第}\omega_i\text{类的样本数}}{\text{样本总数}}$$

- ω_i 对应体育类、政治类、娱乐类.....

- 计算类条件概率

$word_j$ 在第 ω_i 类中出现的次数

$$P(word_j | \omega_i) = \frac{n_j + 1}{n + |Vocabulary|}$$

第 ω_i 类中的总字数 词汇量

为什么类条件概率的形式是这样？

- 决策：

$$\omega_{NB} = \arg \max_{\omega_i \in \Lambda} P(\omega_i) \prod_{j=1}^n P(word_j | \omega_i)$$

你能想到文本识别的其它案例吗？

用贝叶斯方法可以实现语音识别吗？ 怎么实现？要注意什么问题？

- 样本集 X 中的样例 x_j 具有多重属性时使用。
- 估计每一类的先验概率 $P(\omega_i)$ 。怎么估计？
- 估计类条件概率 $P(x_j|\omega_i)$ 。怎么估计？

- 对于新样本，采用如下的方式分类

$$\omega_{NB} = \arg \max_{\omega_i \in \Lambda} P(\omega_i) \prod_{j=1}^n P(x_j|\omega_i)$$

称之为最大后验假设。

- Tom Mitchell, Machine Learning (Chapter 6), McGraw-Hill
- Additional reading about Naïve Bayes Classifier
 - <http://www-2.cs.cmu.edu/~tom/NewChapters.html>
- Software for text classification using Naïve Bayes Classifier
 - <http://www-2.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html>

- 估计在Outlook = Sunny, Temperature = Cool情况下，此人是否打网球？

Day	Outlook	Temperature	PlayTennis
D1	Sunny	Hot	No
D2	Sunny	Hot	No
D3	Overcast	Hot	Yes
D4	Rain	Mild	Yes
D5	Rain	Cool	No
D6	Rain	Cool	No
D7	Overcast	Cool	Yes
D8	Sunny	mild	Yes
D9	Sunny	Cool	Yes
D10	Rain	mild	Yes

集成算法

目录

CONTENTS

- 集成算法思想
- Boosting算法代表：Adaboost
- Adaboost案例：人脸检测

PART 01

01

集成算法思想

怎样设计一个更好分类器？

- 提取好的特征
- 使用精度更好的分类算法
- 把样本变到一个新的空间
- 使用先验和上下文信息
- 使用更多的数据
-
- **将现有的方法联合起来**



组合起来味道好！

- 找到一个泛化能力非常强的分类器不容易。但找到一个弱分类器却比较容易，比如线性分类器，决策树等。
- **弱分类器是指比猜强的分类器**，比如二类分类，弱分类器的正确率应该大于50%。
- 我们是否可以将几个弱分类器组合起来变成一个强分类器？如何组合？
✓答：Bagging, Stacking, Boosting !



三个臭皮匠赛过一个诸葛亮

PART 02

02

Boosting算法代表：Adaboost

- 本讲以两类AdaBoost为例
- 设定 N 个训练样本 $X = \{x_1, \dots, x_N\}$ ，用 $f_m(x) \in \{-1, 1\} (m = 1, \dots, M)$ 表示 M 个弱分类器在样本 x 上的输出。
- AdaBoost算法的目的：构造这 M 个分类器并将其组合起来进行决策。

1. 初始化训练样本 $\{x_1, \dots, x_N\}$ 的权重 $w_i = \frac{1}{N}$, $i = 1, \dots, N$
2. 对 $m = 1$ 到 M , 重复以下过程
 - a) 利用 $\{w_i\}$ 加权后的训练样本构造 **“最好”** 的弱分类器 $f_m(x) \in \{-1, 1\}$ 。

注意：弱分类器的具体算法可以不同，比如采用线性分类器和决策树等，**那个好用那个。**

- b) 计算该 **“最好”** 的弱分类器分类错误率 $e_m = \sum_{i=1}^N w_i I(y_i \neq f_m(\mathbf{x}_i))$, 并令 $c_m = \ln(\frac{1-e_m}{e_m})$ 。 ($I(y_i \neq f_m(\mathbf{x}_i))$ 表示当 $y_i \neq f_m(\mathbf{x}_i)$ 时取1 , 否则取0)
- c) 令 $w_i = w_i \exp\{c_m I(y_i \neq f_m(\mathbf{x}_i))\}$, $i = 1, \dots, N$, 并归一化使 $\sum_{i=1}^N w_i = 1$ 。
3. 对于待分样本 \mathbf{x} , 分类器的输出为

$$\text{sgn} \left[\sum_{m=1}^M c_m f_m(\mathbf{x}) \right]$$

PART 03

03

Adaboost案例：人脸检测

- a) 利用 $\{w_i\}$ 加权后的训练样本构造 **“最好”** 的弱分类器 $f_m(x) \in \{-1, 1\}$ 。
- 一些分类器算法可以接受加权后的样本，但是很多分类器并不能，因此通常情况下，我们按照 $\{w_i\}$ 进行重采样。



- 重采样结果



a) 利用 $\{w_i\}$ 加权后的训练样本构造 **“最好”** 的弱分类器 $f_m(x) \in \{-1, 1\}$ 。



- “遍历”所有的弱分类器，找到 **“最好”** 的弱分类器，分类错误率最小的那个。比如

	$h_1(x)$	$h_2(x)$	$h_3(x)$	$h_t(x)$
错误率	0.46	0.36	0.16	0.43

- $h_3(x)$ 最好，将其作为当前的 $f_m(x)$ 。

b) 计算分类错误率 $e_m = \sum_{i=1}^N w_i I(y_i \neq f_m(x_i))$



- $e_m = \frac{1}{16} + \frac{1}{4} = \frac{5}{16}$
- 值得注意的是，错误率是在原样本上进行计算，而非重采样的样本上。
- $e_m < 0.5$ ：弱分类器果然比猜好！








- 计算 $c_m = \ln \left(\frac{1-e_m}{e_m} \right)$

由 $e_m = \frac{5}{16}$ 可得

$$c_m = \ln \frac{1 - \frac{5}{16}}{\frac{5}{16}} = \ln \frac{11}{5} \approx 0.8$$

- e_m 越小， c_m 越大，当前的分类器 $f_m(x_i)$ 也就越重要。
- 因而分类器的输出为 $\text{sgn}[\sum_{m=1}^M c_m f_m(x)]$

c) 重新计算权值 $w_i = w_i \exp\{c_m I(y_i \neq f_m(x_i))\}$

						
1/16	1/4	1/16	1/16	1/4	1/16	1/4
✓	✓	✓	✗	✗	✓	✓
⇓	⇓	⇓	⇓	⇓	⇓	⇓
1/16	1/4	1/16	$(1/16) \exp(0.8)$	$(1/4) \exp(0.8)$	1/16	1/4

- 分错的样本的权值增大了，也就是说这样的样本在下一次分类时需要得到更多的关注。

AdaBoost案例：step by step

- 权值归一化
归一化之前

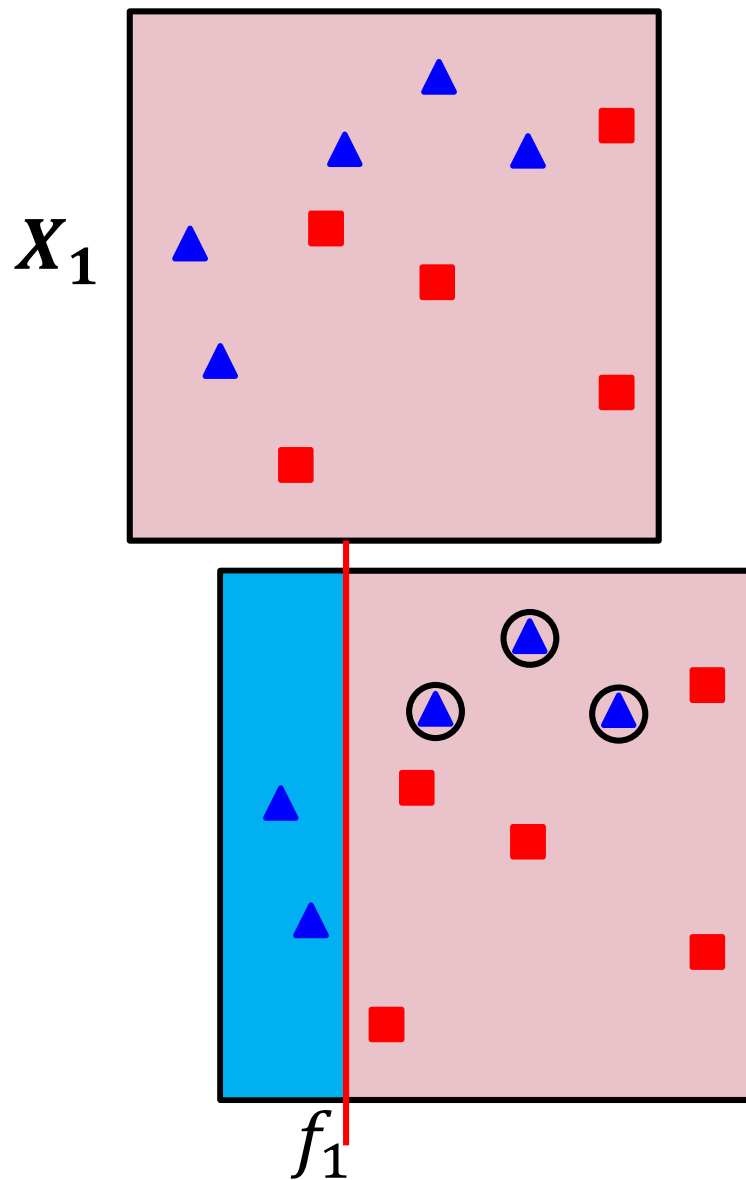


归一化之后



- 归一化之后按同样的步骤寻找下一个分类器。

AdaBoost例子



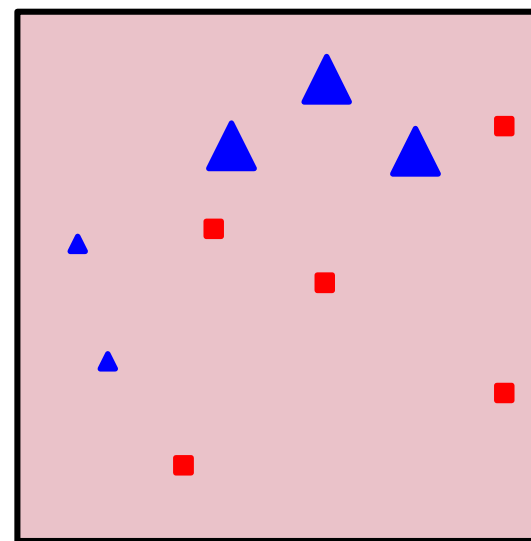
第一轮

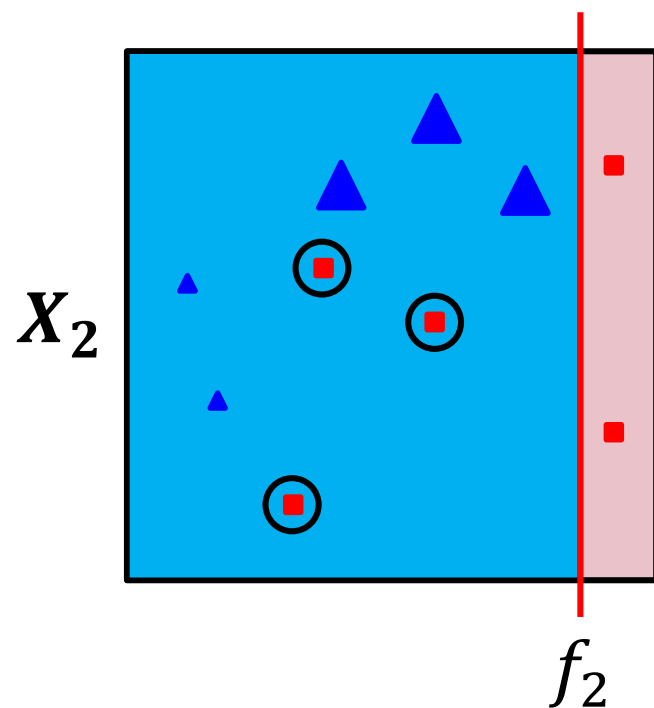
$$e_1 = 0.3$$

$$c_1 = 0.42$$



X_2



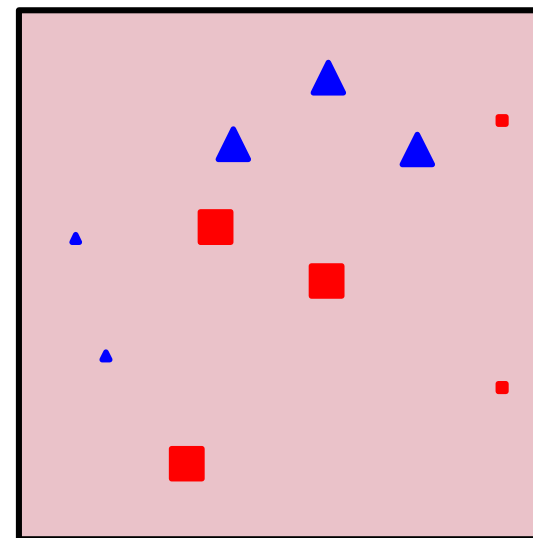


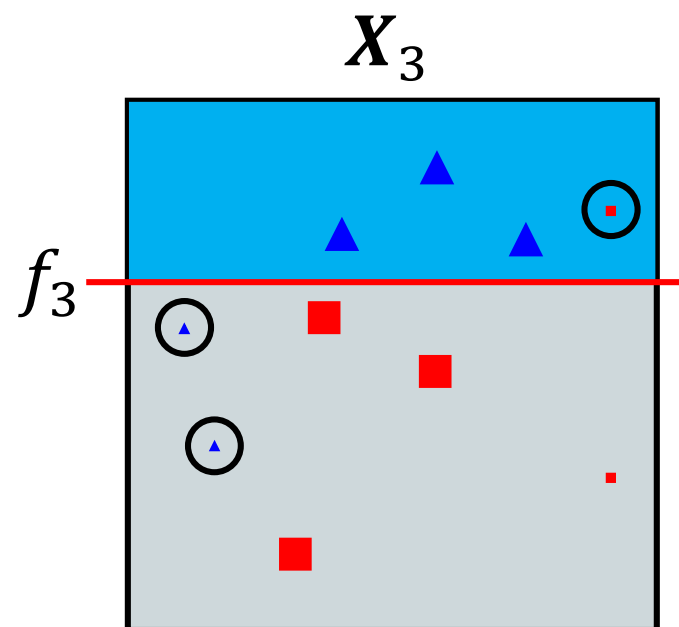
第二轮

$$e_2 = 0.21$$
$$c_2 = 0.65$$



X_3



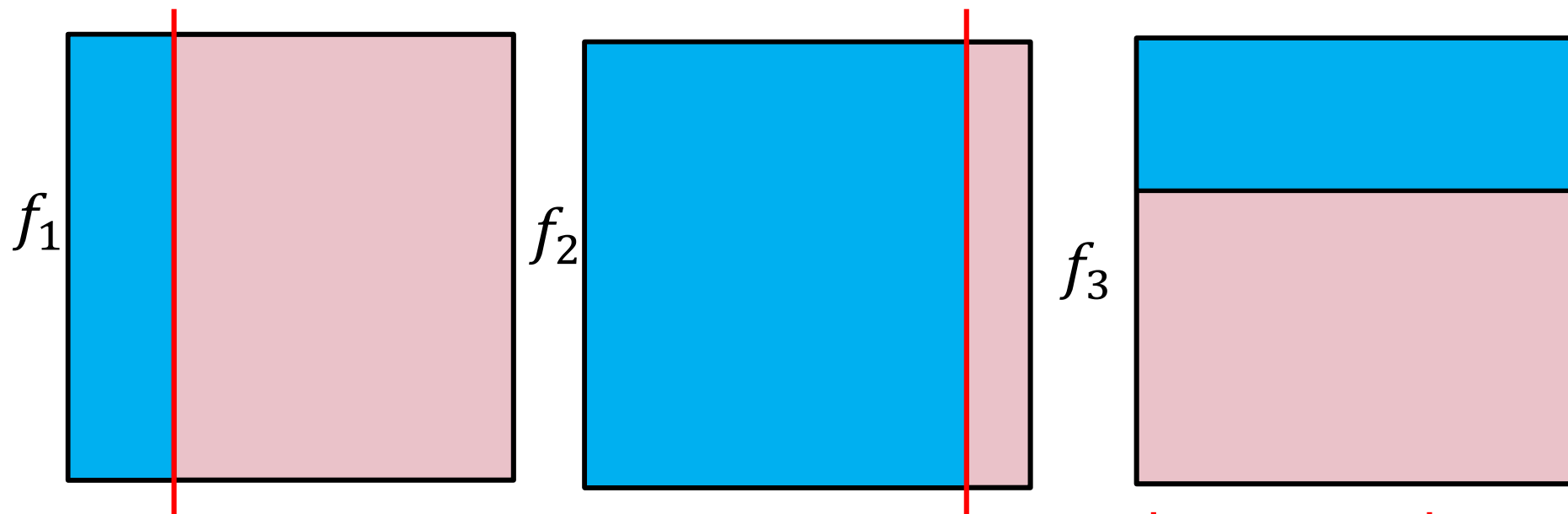


第三轮

$$e_3 = 0.14$$

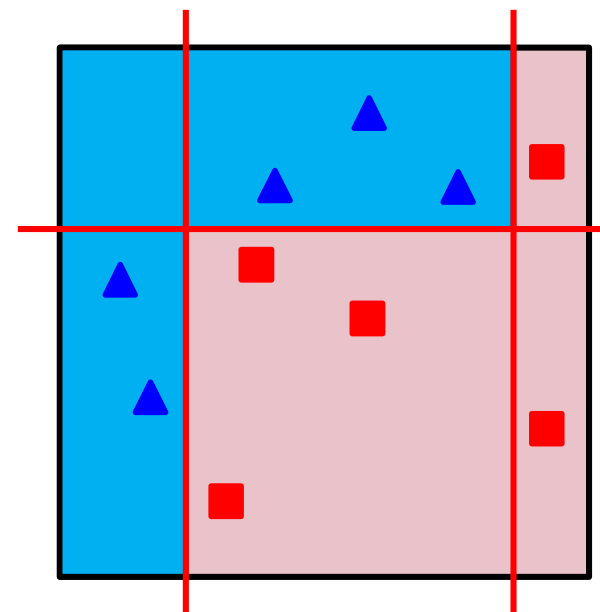
$$c_3 = 0.92$$

AdaBoost例子 (续)



$$c_1 = 0.42 \quad c_2 = 0.65 \quad c_3 = 0.92$$

$$f_{\text{final}} = \text{sgn}(0.42f_1 + 0.65f_2 + 0.92f_3)$$



小小大大大小
我不等你谁等你
大大小小小小大
我不等你我等谁
最不过是
你不等我我等你.....

试比较AdaBoost方法和我们讲过的投票分类方法（比如多个SVM投票分类）

- 找到一个泛化能力非常强的分类器不容易。但找到一个弱分类器却比较容易，比如SVM，决策树等。
- **弱分类器是指比猜强的分类器**，比如二类分类，弱分类器的正确率应该大于50%。
- 我们将几个弱分类器利用Boosting方法组合起来变成一个强分类器，Boosting方法的典型代表是AdaBoost算法。

- 1) 为什么要Boosting ?

助力京东基业长青
成就员工事业发展
赋能社会价值共创



京东大学公众号

Thanks !