机器学习

李成龙

安徽大学人工智能学院
“多模态认知计算”安徽省重点实验室
合肥综合性国家科学中心人工智能研究院

内容安排

- 什么是机器学习
- 机器如何学习
- 如何让机器学习的更好
- 为什么机器能学习

内容安排

- **机器如何学习**
  - 有监督学习
    - 感知机
    - 支持向量机
    - 朴素贝叶斯分类
    - 决策树
    - 集成学习（Bagging算法与随机森林、Boosting算法）
    - 线性回归
    - 逻辑回归
    - Softmax回归
    - 神经网络与深度学习
  - 无监督学习
    - 聚类
    - 主成分分析

本节目录

- Boosting集成策略
- Adaboost学习算法

本节目录

Boosting集成策略

- **基本思想**
  - 主要通过集成各个弱学习器的成功经验和失败教训实现对模型性的提升
  - 该方法使用迭代方式完成对各个弱学习器的训练构造，每次迭代对训练样本集的选择都与前面各轮的学习结果有关
  - 使用前面各轮学习结果更新当前各训练样本的权重，对前面被错误预测的赋予较大的权重，实现对当前训练样本集合数据分布的优化

Boosting集成策略

- **学习方法**
  - Boosting 集成学习通常使用两种方式调整训练样本集的数据分布
    - 仅调整样本数据的权重，而不改变当前训练样本集合
    - 改变当前训练样本集合，将被前面弱学习器错误预测的样本复制到关于当前弱学习器的训练样本集合中重新进行训练
  - 第一种方式的基本思想是提高当前训练样本集合中被错误预测样本的权重，降低已被正确预测样本的权重，使得后续对的弱学习器的训练构造更加重视那些被错误预测的样本

Boosting集成策略

- **例题：**现有均匀分配权重样本集训练得到的分类器C_1，其分类结果如表所示。试更新该训练样本集的权重并求出分类器C_1基于更新权重后样本集的分类错误率

编号	预测为+	预测为-	合计
实际为+	36	31	67
实际为-	9	24	33
合计	45	55	100

Boosting集成策略

编号	预测为+	预测为-	合计
实际为+	36	31	67
实际为-	9	24	33
合计	45	55	100

- 依题意可知，共有60个分类正确样本、共40个分类错误样本，分类错误率为ε =0. 4。错误分类样本权重更新因子$\alpha = 1/2\varepsilon = 1.25$，正确分类样本权重更新因子$\beta = 1/2(1-\varepsilon) = 5/6$，则权重更新后分类结果如表所示，此时错误率$\varepsilon' = 0.5$

编号	预测为+	预测为-	合计
实际为+	36	31	67
实际为-	9	24	33
合计	45	55	100

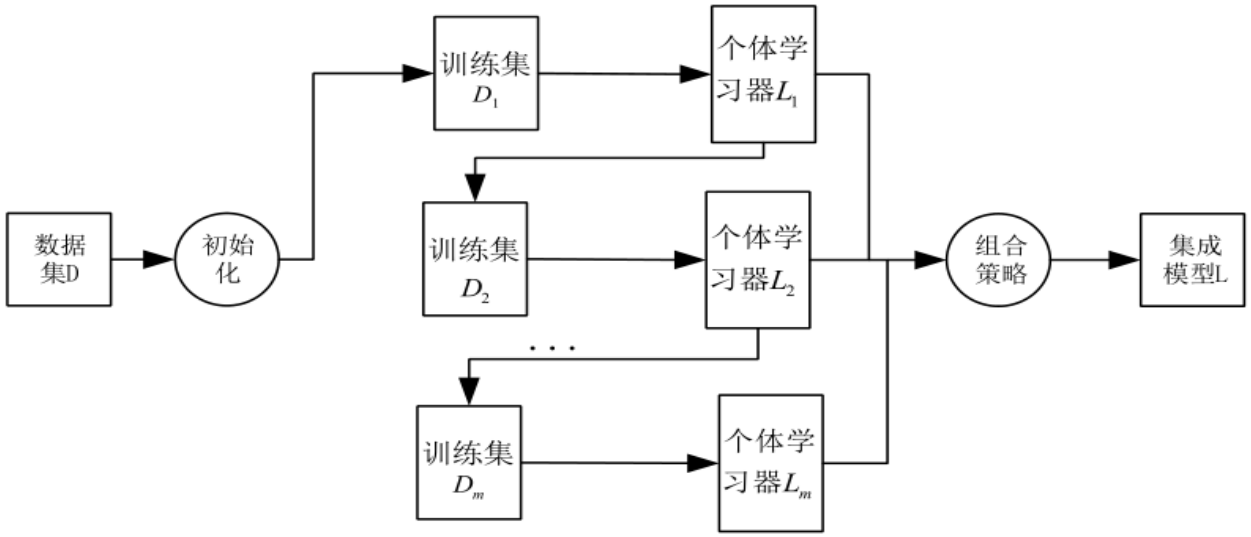
编号	预测为+	预测为-	合计
实际为+	30	39	69
实际为-	11	20	31
合计为	41	59	100

Boosting集成策略

- **学习方法**
  - Boosting 集成学习通常使用两种方式调整训练样本集的数据分布
    - 仅调整样本数据的权重，而不改变当前训练样本集合
    - 改变当前训练样本集合，将被前面弱学习器错误预测的样本复制到关于当前弱学习器的训练样本集合中重新进行训练
  - 第二种方式是复制被前面弱学习器错误预测样本到样本训练集当中重新进行训练
  - Schapire’s Boosting算法正是采用上述思想实现对训练样本集数据分布的调整。该算法是一种用于解决分类问题的 Boosting 集成学习算法，主要通过构建三个互补的弱分类器并由投票法将其集成为一个具有较强分类性能的强分类器

Boosting集成策略

- 学习方法
  - 与Bagging集成学习方法类似，Boosting集成学习通常采用加权平均或加权投票方法实现对多个弱学习器的集成



本节目录

- Boosting集成策略
- **Adaboost学习算法**

Adaboost学习算法

- 背景
  - 1990年，Schapire最先构造出一种多项式级的算法，即最初的Boost算法
  - 1993年，Drunker和Schapire第一次将神经网络作为弱学习器，应用Boosting算法解决OCR问题
  - 1995年，Freund和Schapire提出了Adaboost(Adaptive Boosting)算法，效率和原来Boosting算法一样，但是不需要任何关于弱学习器性能的先验知识，可以非常容易地应用到实际问题中

Adaboost学习算法

- **基本概念**
  - AdaBoost是一种具有自适应性质的Boosting集成学习算法
  - 自适应性主要表现在自动提升被错误预测样本的权重，自动减少被正确预测样本的权重，使得弱学习器训练过程能够根据模型预测性能自动进行调整

AdaBoost

Adaptive Boosting

Adaboost学习算法

- 基本概念

$$h_1(x) \in \{-1, +1\}$$
$$h_2(x) \in \{-1, +1\}$$
$$\vdots$$
$$h_T(x) \in \{-1, +1\}$$

$$H_T(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

弱学习器　　强学习器

比随机猜测性能好一些

Adaboost学习算法

- **算法过程**
  - 现以二分类任务为例介绍该算法的具体过程
    - 对于训练样本集$D = \{(X_1, y_1), (X_2, y_2), \cdots, (X_n, y_n)\}$，其中$y_i \epsilon \{-1, +1\}$，由 AdaBoost 集成学习算法构造集成模型的基本步骤如下

Adaboost学习算法

- **算法过程**

（1）令$i=1$并设定弱学习器的数目m。使用均匀分布初始化训练样本集的权重分布，令n维向量w^i表示第i次需更新的样本权重，则有：$w^1=(w_{i1},w_{i2},\cdots,w_{in})^T=\left(\frac{1}{n},\frac{1}{n},\cdots,\frac{1}{n}\right)^T$

（2）使用权重分布为w^i的训练样本集D_i学习得到第i个弱学习器f_i

（3）计算f_i在训练样本集D_i上的分类错误率e_i ：

$$e_i=\sum_{k=1}^{n}w_{ik}I(f_i(X_k)\neq y_k)$$

（4）确定弱学习器f_i的组合权重α_i。由于弱学习器f_i的权重取值应与其分类性能相关，对于分类错误率e_i越小的f_i，则其权重α_i应该越大，故有$\alpha_i=\frac{1}{2}\ln\frac{1-e_i}{e_i}$

（5）依据弱学习器f_i对训练样本集D_i的分类错误率e_i更新样本权重，更新公式为$w_{i+1,j}=\frac{w_{ij}\exp(-\alpha_i y_k L_i(X_k))}{Z_i}$，其中$Z_i=\sum_{k=1}^{n}w_{ij}\exp(-\alpha_i y_k f_i(X_k))$为归一化因子，保证更新后权重向量为概率分布

（6）若$i<m$，则令$i=i+1$并返回步骤（2），否则执行步骤（7）

（7）对于m个弱分类器$f_1,f_2,\cdots,f_m$，分别将每个f_i按权重α_i进行组合：$G=sign(\sum_{i=1}^{m}\alpha_i f_i(X))$，得到并输出所求集成模型$G$，算法结束

Adaboost学习算法

- **算法过程**
  - 现以二分类任务为例介绍该算法的具体过程
    - 算法关键要点是如何更新样本权重，即步骤（5）中的权重更新公式。则可将该公式改写为如下形式

$$w_{i+1,j} = \begin{cases} \frac{w_{ij}}{Z_i}\exp(-\alpha_i), f_i(X_k) = y_k \\ \frac{w_{ij}}{Z_i}\exp(\alpha_i), f_i(X_k) \neq y_k \end{cases}$$

    - 即当某个样本被前一个弱学习器错误预测时，该样本的权重会被放大$e_i/(1-e_i)$倍以便在后续弱学习器构造过程得到应有的重视

Adaboost学习算法

- **例题：**试以所示数据集为训练样本，使用AdaBoost集成学习算法构建集成模型

编号	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

- 对m=1，取在训练数据集的初始样本权值$w_1 = (w_{11}, w_{12}, \cdots, w_{110})^T$，其中$w_{1i} = 0.1, i = 1,2,3, \cdots, 10$
- 现通过数据集训练第一个弱学习器f_1并依据f_1更新样本权值分布，由于当阈值$v = 2.5$时，分类错误率e_1最小，故可得到：$f_1(X) = \begin{cases} 1, X < 2.5 \\ -1, X \geq 2.5 \end{cases}$
- 错误率$e_1 = P(f_1(X_i) \neq y_i) = 0.3$。进一步计算$f_1(X)$的集成系数$\alpha_1$：$\alpha_1 = \frac{1}{2}\ln\frac{1-e_1}{e_1} = 0.4236$
- 根据权重更新公式和弱学习器$f_1(X)$分类结果更新权重，得到权重向量$w_2 = (0.0715,0.0715,0.0715,0.0715,0.0715,0.0715,0.1666,0.1666,0.1666,0.0715)^T$
- $G_1(X) = \text{sgn}[0.4236 f_1(X)]$
- 弱基本分类器$G_1(X)$在更新的数据集上有3个误分类点

Adaboost学习算法

- **例题：**试以所示数据集为训练样本，使用AdaBoost集成学习算法构建集成模型

编号	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

- 对m=2，在分布权值w_2上，由于当阈值$v = 8.5$时，分类错误率e_2最小，故可得到：$f_2(X) = \begin{cases} 1, X < 8.5 \\ -1, X \geq 8.5 \end{cases}$
- 错误率$e_2 = P(f_2(X_i) \neq y_i) = 0.2143$。进一步计算$f_2(X)$的集成系数$\alpha_2$：$\alpha_2 = \frac{1}{2}\ln\frac{1-e_2}{e_2} = 0.6496$
- 根据权重更新公式和弱学习器$f_2(X)$分类结果更新权重，得到权重向量$w_3 = (0.0455,0.0455,0.0455,0.1667,0.1667,0.1667,0.1060,0.1060,0.1060,0.0455)^T$
- $G_2(X) = \text{sgn}[0.4236f_1(X) + 0.6496f_2(X)]$
- 分类器$G_2(X)$有三个误分类点

Adaboost学习算法

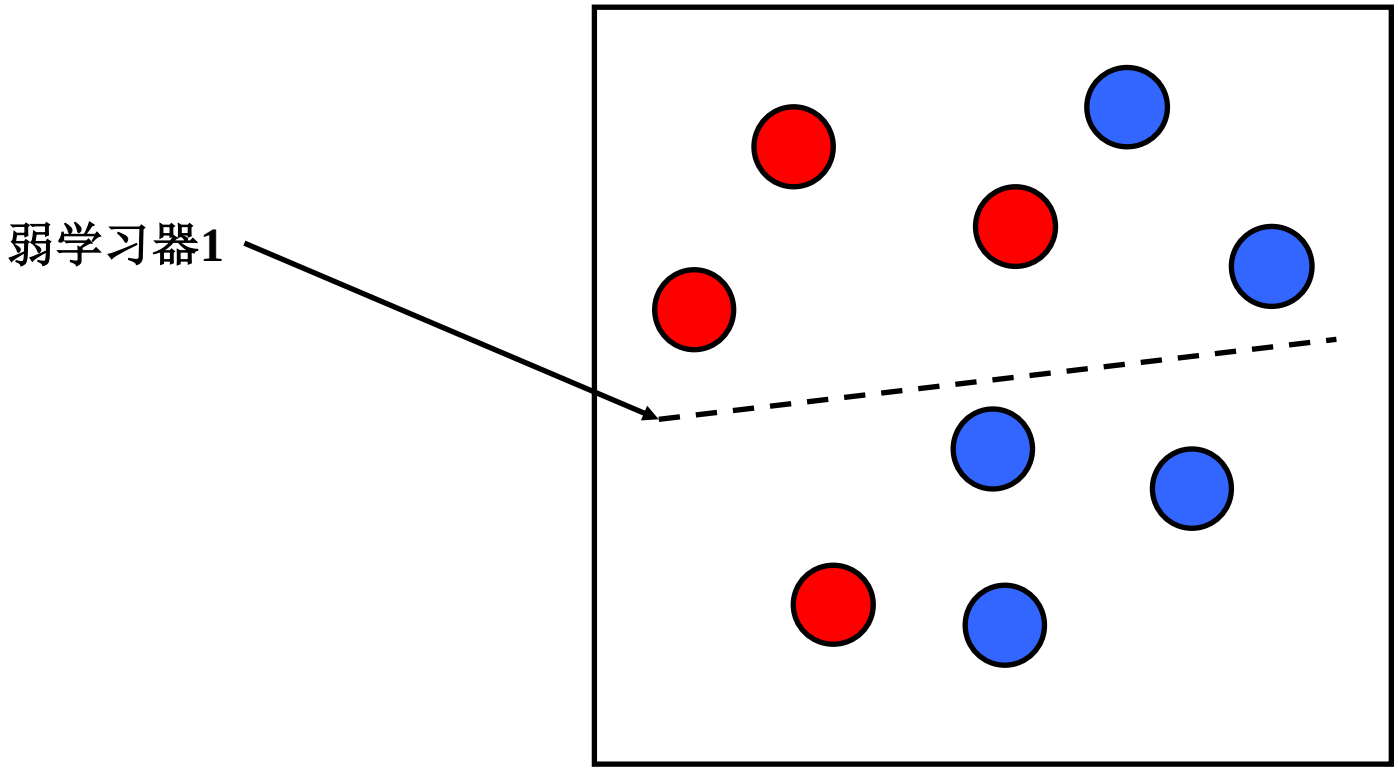
- **例题：**试以所示数据集为训练样本，使用AdaBoost集成学习算法构建集成模型

编号	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

- 对m=3，在分布权值w_3上，由于当阈值$v = 5.5$时，分类错误率e_3最小，故可得到：$f_3(X) = \begin{cases} 1, X < 5.5 \\ -1, X \geq 5.5 \end{cases}$
- 错误率$e_3 = P(f_3(X_i) \neq y_i) = 0.1820$。进一步计算$f_3(X)$的集成系数$\alpha_3$：$\alpha_3 = \frac{1}{2}\ln\frac{1-e_3}{e_3} = 0.7514$ 根据权重更新公式和弱学习器 $f_3(X)$ 分类结果更新权重，得到权重向量 $w_4 =$ $(0.125,0.125,0.125,0.102,0.102,0.102,0.065,0.065,0.065,0.125)^T$
- $G_3(X) = \text{sgn}[0.4236f_1(X) + 0.6496f_2(X) + 0.7514f_3(X)]$
- 分类器$G_3(X)$的误分类点为0

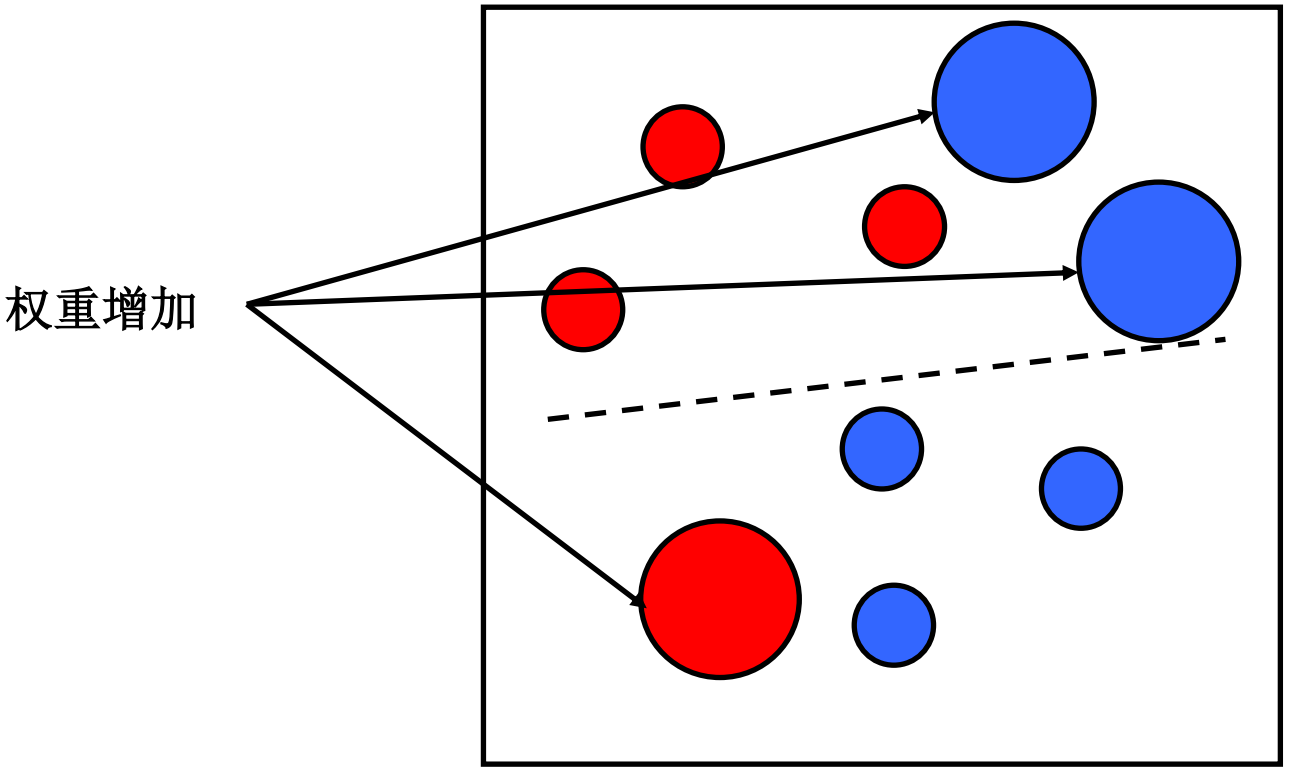
Adaboost学习算法

- Boosting过程示例



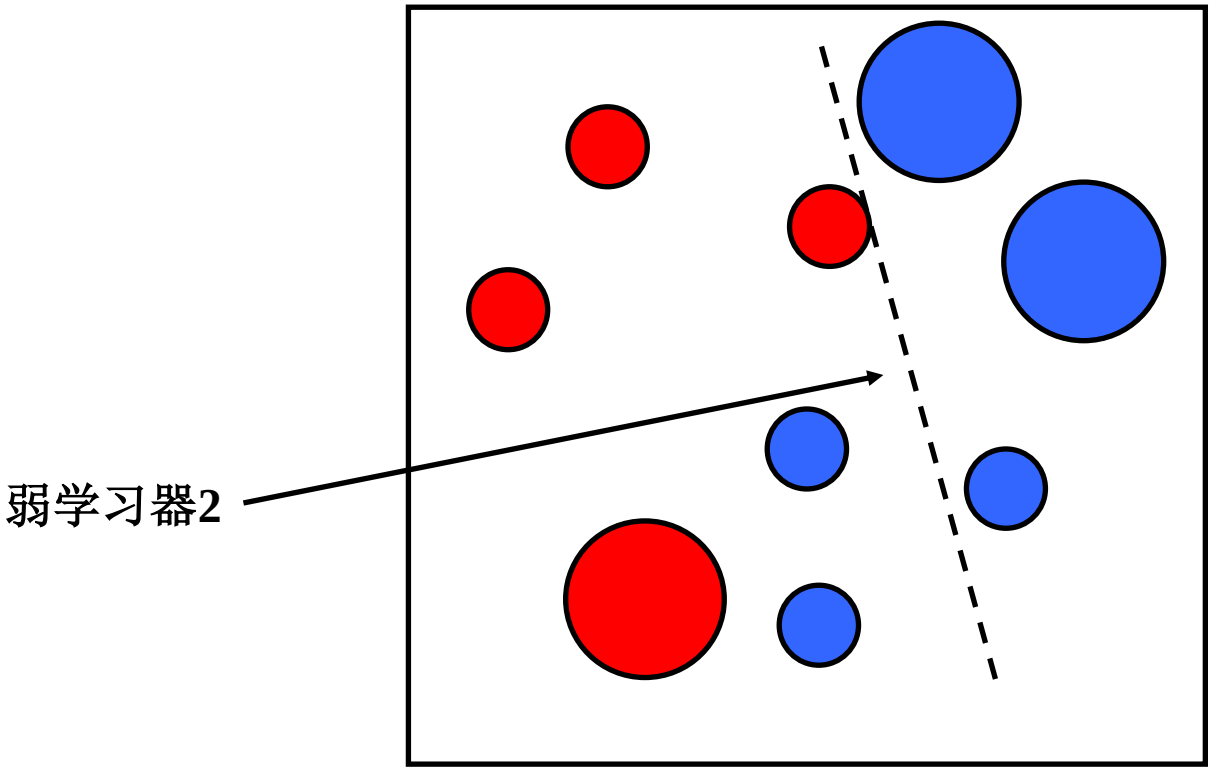
Adaboost学习算法

- Boosting过程示例



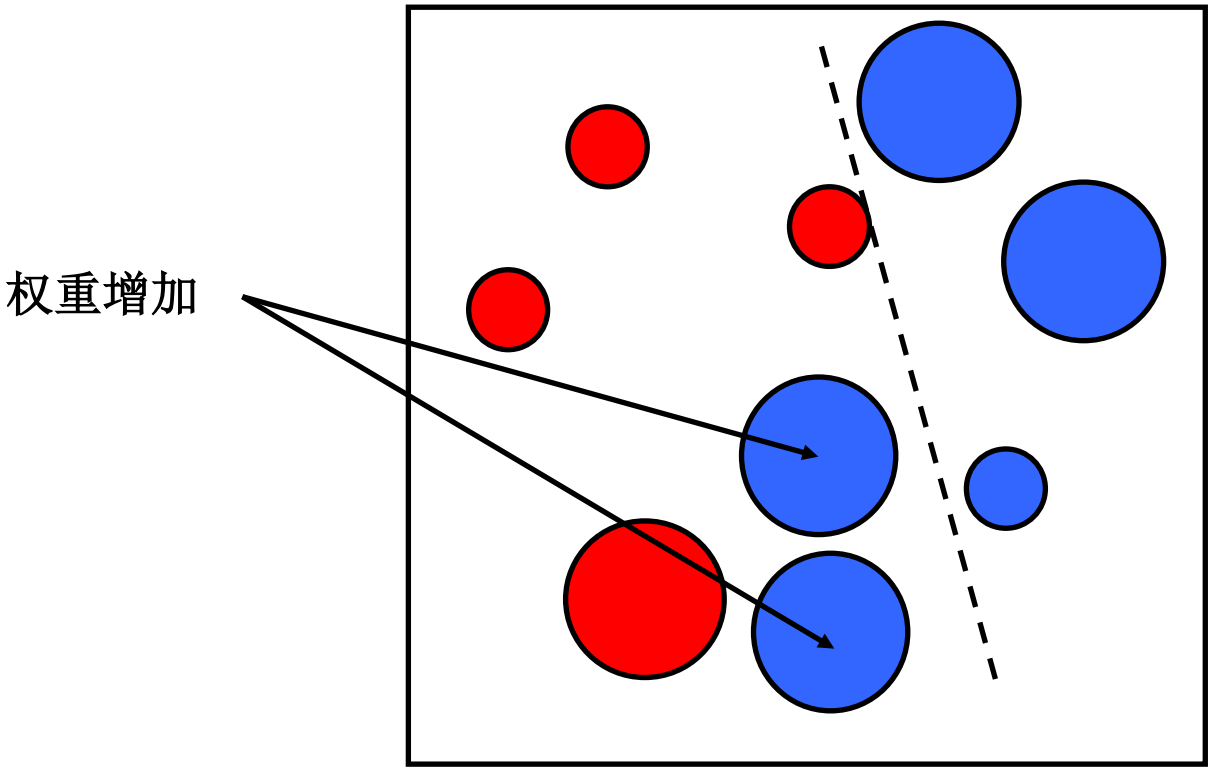
Adaboost学习算法

- Boosting过程示例



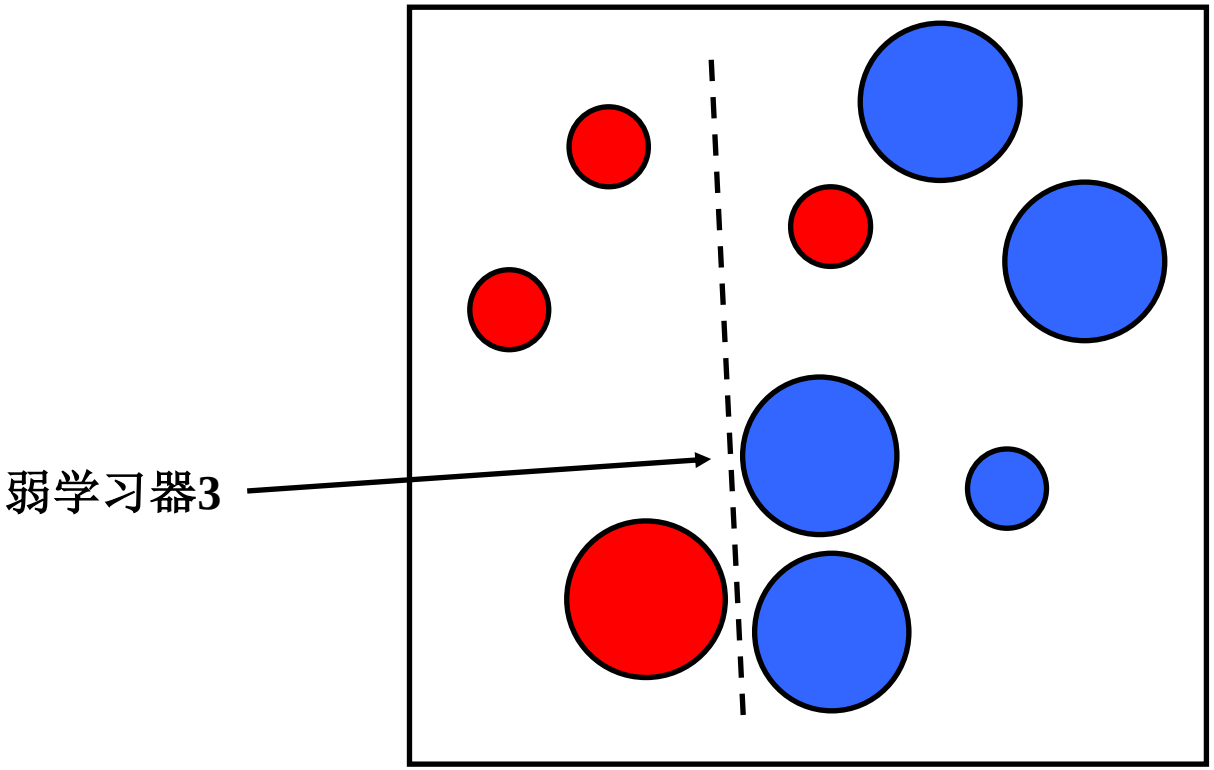
Adaboost学习算法

- Boosting过程示例



Adaboost学习算法

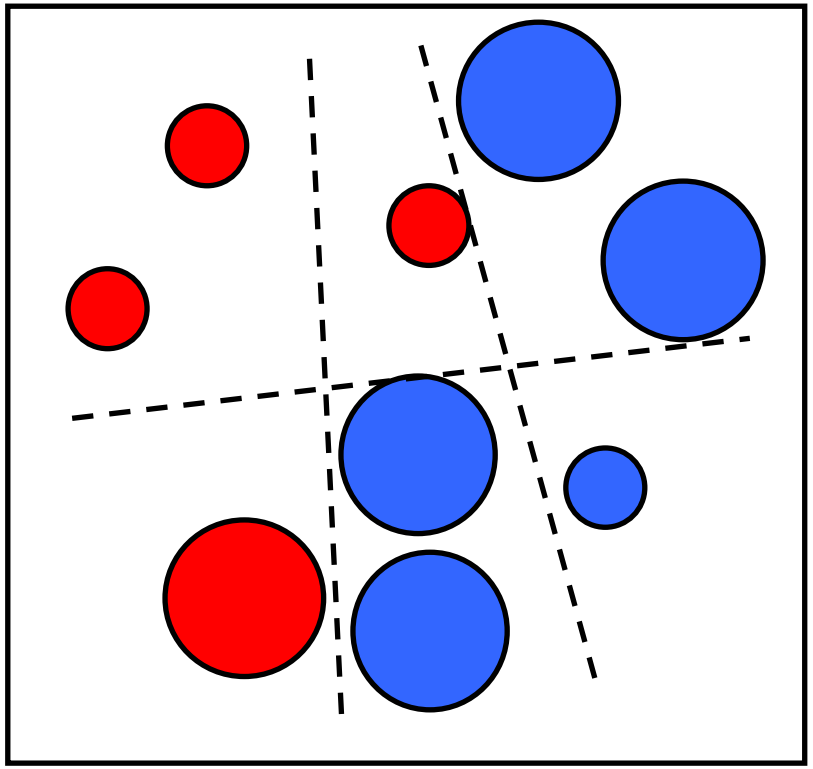
- Boosting过程示例



Adaboost学习算法

- Boosting过程示例

最终的学习器为
弱学习器的组合



Adaboost学习算法

- **误差分析**
  - 算法能在学习过程中不断减少训练误差

Adaboost的误差上界：$$Error = \frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leq \frac{1}{N}\sum_{i} exp(-y_i f(x_i)) = \prod_{m} Z_m$$

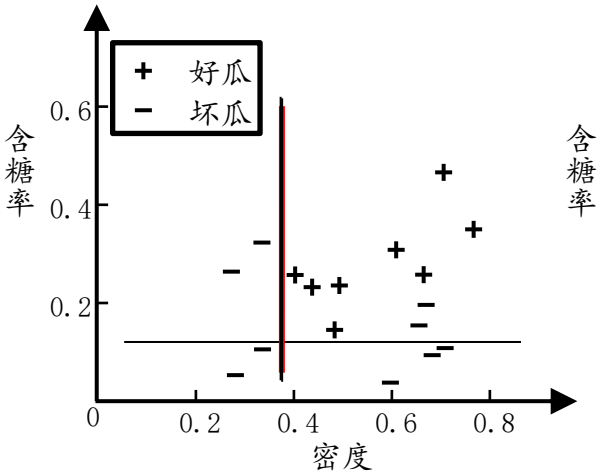
  - 算法的训练误差以指数速率下降的

Adaboost的误差上界：$$Error = \frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leq \prod_{m=1}^{M} Z_m \leq exp\left(-2\sum_{m=1}^{M}\gamma_m^2\right) \leq exp(-2M\gamma^2)$$

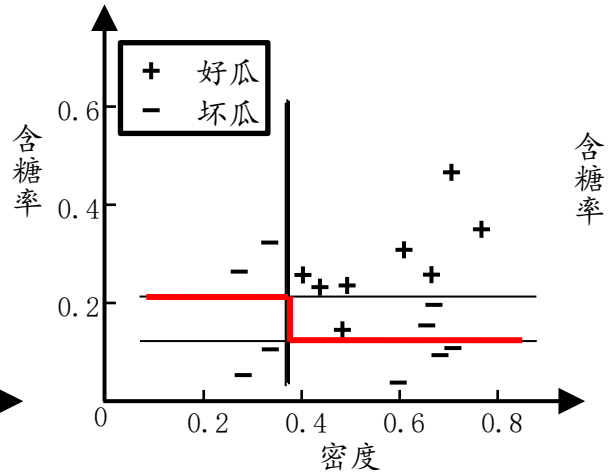
  - 与一些早期的提升方法不同，AdaBoost具有适应性，即它能适应弱分类器各自的训练误差率，这也是它的名称的由来

Adaboost学习算法

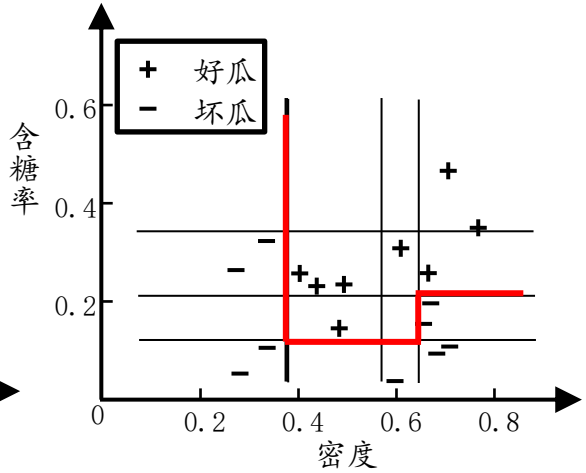
- **误差分析**
  - 从偏差-方差分解的角度看，Boosting主要关注降低偏差，因此可以基于泛化性能较弱的学习器构建很强的集成，如决策树桩



(a) 3个基学习器



(b) 5个基学习器



(c) 11个基学习器

Adaboost学习算法

- **算法解释**
  - 从前向分步算法的角度来解释分类器权重、分类误差以样本权重的更新公式的由来
    - 分类器权重公式

$$\alpha_m = \frac{1}{2} ln \frac{1 - e_m}{e_m}$$

    - 分类误差公式

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

    - 样本权重的更新公式

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} e^{-\alpha_m y_i G_m x_i}, i = 1, 2, \ldots N$$

Adaboost学习算法

- **算法解释**
  - Adaboost算法其实可以认为是模型为加法模型、损失函数为指数函数、学习算法为前向分布算法的二分类学习方法
    - 加法模型

基函数的参数

$$f(x)=\sum_{m=1}^{M}\beta b(x;\gamma_m)$$

基函数

基函数的系数

$$f(x)=\sum_{m=1}^{M}\alpha_m G_m(x)$$

    - 指数损失函数

$$L(y,f(x))=\sum_{i=1}^{N}exp(-yf(x))$$

    - 优化目标：经验风险极小化问题

$$\min_{\beta_m\gamma_m}\sum_{i=1}^{N}L(y_i,\sum_{m=1}^{m}\beta_m b(x_i;\gamma_m))$$

复杂的优化问题

Adaboost学习算法

- **算法解释**
  - 前向分布算法
    - 求解这一优化问题的思路是： 从前往后每一步只学习一个基函数及其系数，逐步逼近优化目标函数
    - 每步只需优化如下损失函数

$$\min_{\beta_m \gamma_m} \sum_{i=1}^{N} L(y_i, \sum_{m=1}^{m} \beta_m b(x_i; \gamma_m))$$

$$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma))$$

    - 假设经过m-1轮迭代前向分步算法已经得到：

$$f_{m-1}(x) = f_{m-2}(x) + \alpha_{m-1} G_{m-1}(x) = \alpha_1 G_1(x) + \ldots + \alpha_{m-1} G_{m-1}(x)$$

    - 在第m轮迭代得到 α_m ， $G_m(x)$ 和 $f_m(x)$ ：

$$f_m(x) = f_{m-1}(x) + \alpha_m G_m(x)$$

Adaboost学习算法

- **算法解释**
  - 前向分布算法
    - 目标是使前向分布算法得到的α_m，$G_m(x)$和$f_m(x)$在训练集上的指数损失最小，可以表示为：

$$(\alpha_m, G_m(x)) = \arg\min_{\alpha,G} \sum_{i=1}^{N} exp(-y_i(f_{m-1}(x) + \alpha_m G_m(x_i))$$

$$= \arg\min_{\alpha,G} \sum_{i=1}^{N} \overline{w}_{mi} exp[-y_i \alpha G(x_i)]$$

其中$\overline{w}_{mi} = exp[-y_i f_{m-1}(x_i)]$和优化变量无关

- 对于任意的α>0，使上式子最小的$G(x)$可以由下式子得到：

$$G_m^*(x) = \arg\min_{G} \sum_{i=1}^{N} \overline{w}_{mi} I(y_i \neq G(x_i))$$

G_m^*其实就是Adaboost算法的基分类器，基本思想就是让使加权训练数据分类误差率最小的分类器

Adaboost学习算法

- **算法解释**
  - 前向分布算法
    - 通过推导，可以得到：

$$\sum_{i=1}^{N} \overline{w}_{mi} exp(-y_i \alpha G(x_i))$$

$$= \sum_{y_i = G_m(x_i)} \overline{w}_{mi} e^{-\alpha} + \sum_{y_i \neq G_m(x_i)} \overline{w}_{mi} e^{\alpha}$$

$$= (e^{\alpha} - e^{-\alpha}) \sum_{i=1}^{N} \overline{w}_{mi} I(y_i \neq G_(x_i)) + e^{-\alpha} \sum_{i=1}^{N} \overline{w}_{mi}$$

    - 对 α求导并令导数为0，可得到：

$$\alpha_m^* = \frac{1}{2} ln \frac{1 - e_m}{e_m}$$

其中e^m是分类误差

$$e_m = \frac{\sum_{i=1}^{N} \overline{w}_{mi} I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} \overline{w}_{mi}} = \sum_{i=1}^{N} \overline{w}_{mi} I(y_i \neq G_m(x_i))$$

Adaboost学习算法

- **算法解释**
  - 前向分布算法
    - 对于样本权重的更新而言，由于

$$f_m(x) = f_{m-1}(x) + \alpha_m G_m(x)$$

$$\overline{w}_{mi} = exp[-y_i f_{m-1}(x_i)]$$

所以可得：

$$f_{m-1}(x_i) = -y_i ln\overline{w}_{mi}$$

$$f_m(x_i) = -y_i ln\overline{w}_{m+1,i}$$

$$\overline{w}_{m+1,i} = \overline{w}_{m,i} exp[-y_i \alpha_m G_m(x)]$$

    - 标注为红色的公式即为从前向分步的角度推导出来的分类器权重、分类误差以及样本权重更新公式

本节目录

- **Boosting集成策略**
  - 基本思想
  - 学习方法
- **Adaboost学习算法**
  - 基本概念
  - 算法过程
  - 误差分析
  - 算法解释

思考题

- 与Bagging集成策略相比，Boosting集成策略有哪些优点和缺陷

练习题

- 某公司招聘职员考查身体、业务能力、发展潜力这3项。身体分为合格1、不合格0两级，业务能力和发展潜力分为上1、中2、下3三级。分类为合格1、不合格0两类。已知数据如下表所示，以决策树桩为基学习器，试用Adaboost算法学习一个强分类器。

	1	2	3	4	5	6	7	8	9	10
身体	0	0	1	1	1	0	1	1	1	0
业务能力	1	3	2	1	2	1	1	1	3	2
发展潜力	3	1	2	3	3	2	2	1	1	1
分类	-1	-1	-1	-1	-1	-1	-1	1	1	-1