

# 统计学习方法之提升算法

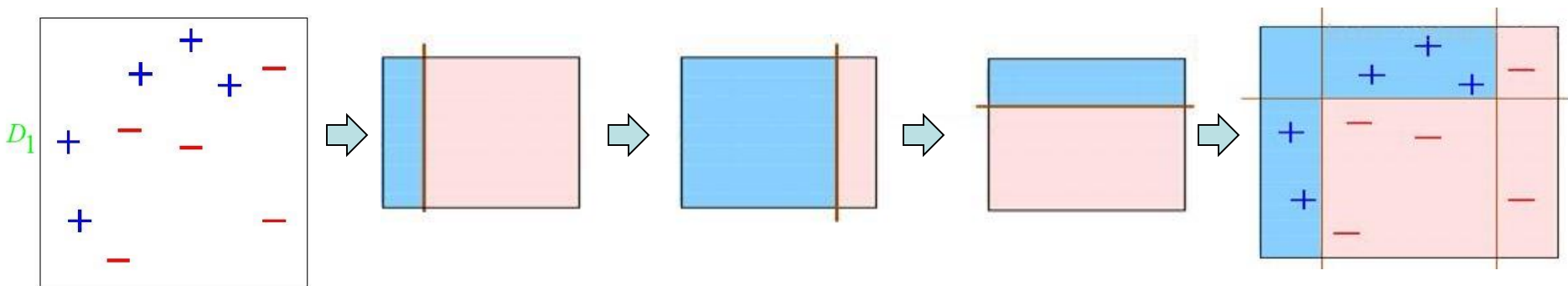
# 提升算法

## ◆算法起源

历史上，Kearns和Valiant首先提出了**强可学习**和**弱可学习**的概念，即在**PAC**（概率近似正确）学习的框架中，如果存在一个多项式的学习算法能够学习它，并且正确率很高，那么就称这个概念是**强可学习**；如果学习的正确率仅比随机猜测略好，则称这个概念是**弱可学习**的。

## ◆基本思想

- 多轮训练，多个分类器
- 每轮训练增加错误分类样本的权值，降低正确分类样本的权值
- 降低错误率高的分类器的权值，增加正确率高的分类器的权值



# AdaBoost算法

## ◆算法过程

- 输入：训练数据集:弱学习算法

- 输出：最终分类器

- 过程：

- (1) 初始化训练数据的权值分布  $D_1 = (w_{11}, \dots, w_{1i}, \dots, w_{1N}), w_{1i} = \frac{1}{N}, i = 1, 2, \dots, N$

- (2) 用户指定训练的迭代次数  $m$

- (a) 使用具有权值分布的训练数据集学习，得到基本分类器

- (b) 计算 $G(x)$ 在训练数据集上的分类误差率  $e_m = P(G_m(x_i) \neq y_i) = \sum_{i=1}^N w_{mi} I(G_m(x_i) \neq y_i)$

- (c) 计算 $G(x)$ 在训练集上的分类误差率  $\alpha_m = \frac{1}{2} \log \frac{1-e_m}{e_m}$

- (d) 更新训练数据集的权值分布

- (3) 构建基本分类器的线性组合，得到最终分类器

## 【注意】

- $G(x)$ 的系数之所以是这个式子，是由前向分布算法的推导得出的

- $Z_m$ 规范化因子的作用是使 $D_{m+1}$ 成为一个概率分布

## ◆算法实例

### 【注意】

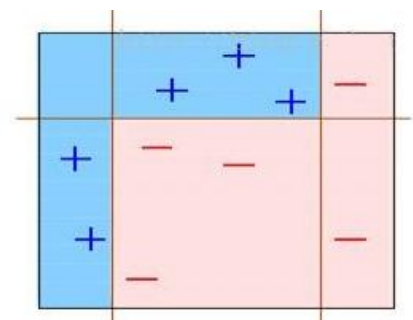
#### ●阈值的选取

$V = \min + \lambda * T$  (T为迭代次数, 从1开始)  $\rightarrow$  计算误差率  $\rightarrow$  交换正负  
 $\rightarrow$  再次计算误差率  $\rightarrow$  当误差率比上一个小的时候替换  $\rightarrow$   
T+1.直到  $V \geq \max$  时

## ◆训练误差分析

【注】二类分类问题的训练误差界

(1) 由泰勒展开式推出不等式



## ◆算法优缺点

- AdaBoost是一种有很高精度的分类器
- 可以使用各种方法构建子分类器
- 简单, 不用担心过拟合

(1) 是弱分类器非常简单, 即使很多融合也不易过拟合, 但如果弱分类器太强, 则易过拟合;

(2) 是分类错误率上界随着训练增加而稳定下降

# Bagging算法

在训练的每一轮中，均从原始样本集 $S$ 中有放回地随机抽取训练样本集，这样一个初始样本在某轮训练中可能出现多次或根本不出现，最终的分类规则为简单多数投票法。

区分AdaBoost算法，Bagging算法的训练集的选取是**随机**的，各轮**训练集之间相互独立**，每个分量分类器**并行生成**，没有**权重**，

# 前向分步算法

## ◆基本思想

从前向后，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式【注意针对的是加法模型】

## ◆算法过程

- 输入：训练数据集，损失函数 $L$ ，基本函数集

- 输出：加法模型 $f(x)$

- 过程：

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

- (2) 更新  $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$

- (3) 得到加法模型

## ◆前向分步算法和AdaBoost

# 提升树

## ◆算法过程

- 输入：训练数据集
- 输出：提升树 $f(x)$
- 过程：
  - (1) 初始化 $f(x)=0$
  - (2) 对 $m=1,2,3\dots,M$ 
    - (a) 计算残差
    - (b) 拟合残差，学习一个回归树 $T$
    - (c) 更新 $f_m(x)=f_{m-1}(x)+T$
  - (3) 得到回归问题提升树

# 课后习题

## ◆习题1

详情见代码附件

## ◆习题2

比较支持向量机、AdaBoost、逻辑斯谛回归模型的学习策略与算法

### •支持向量机

(1) 学习策略：最大化样本点到分离超平面的最小距离

(2) 算法：线性支持向量机 (P112)、非线性支持向量机 (P124)

### •逻辑斯谛回归模型

(1) 学习策略：将分离超平面作为sigmoid函数的自变量进行输入，获得样本点被分为正反例的条件概率，极大似然估计样本的后验概率分布。

(2) 算法： P79

### •感知机模型

(1) 学习策略：利用分离超平面对数据分割，找出所有错误的分类点，使错误分类点到超平面距离之和最小化。

(2) 算法： P33 (对偶形式)

### •AdaBoost

(1) 学习策略：训练弱分类器，更新权重，直到误差率足够小

(2) 算法： P138