第17章：不平衡数据问题及组合模型

《Python数据科学：技术详解与商业实践》

讲师：Ben

自我介绍

- 天善商业智能和大数据社区　　讲师 –Ben
- 天善社区 ID - Ben_Chang
- https://www.hellobi.com – 学习过程中有任何相关的问题都可以提到技术社区数据挖掘版块。

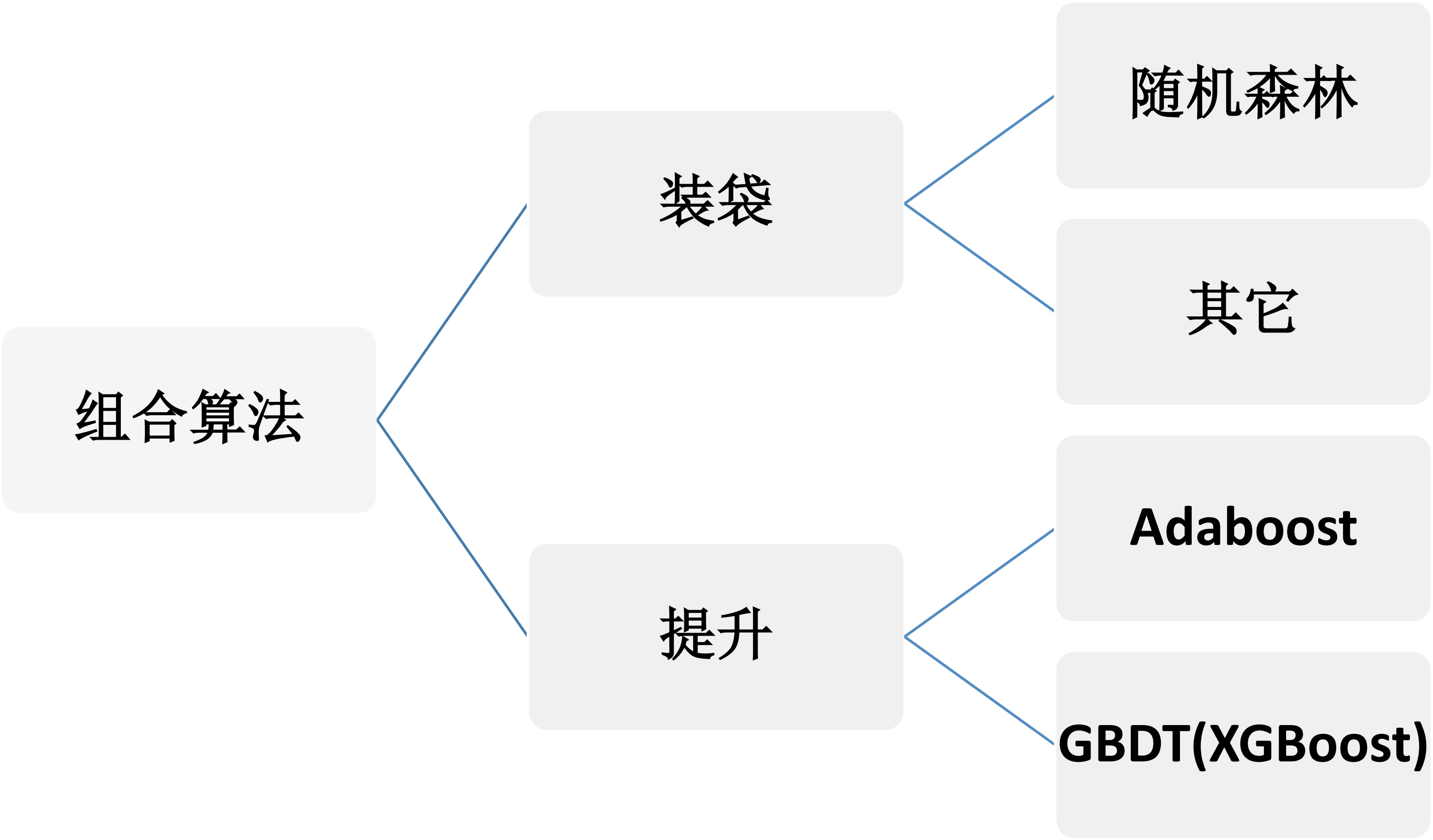
主要内容

- 集成学习概述
  - 装袋法（Bagging）
  - 提升（boosting）
- 随机森林
- Adaboost算法
- 提升树、GBDT和XGBoost
- 偏差（Bias）-方差（Variance）权衡与集成方法

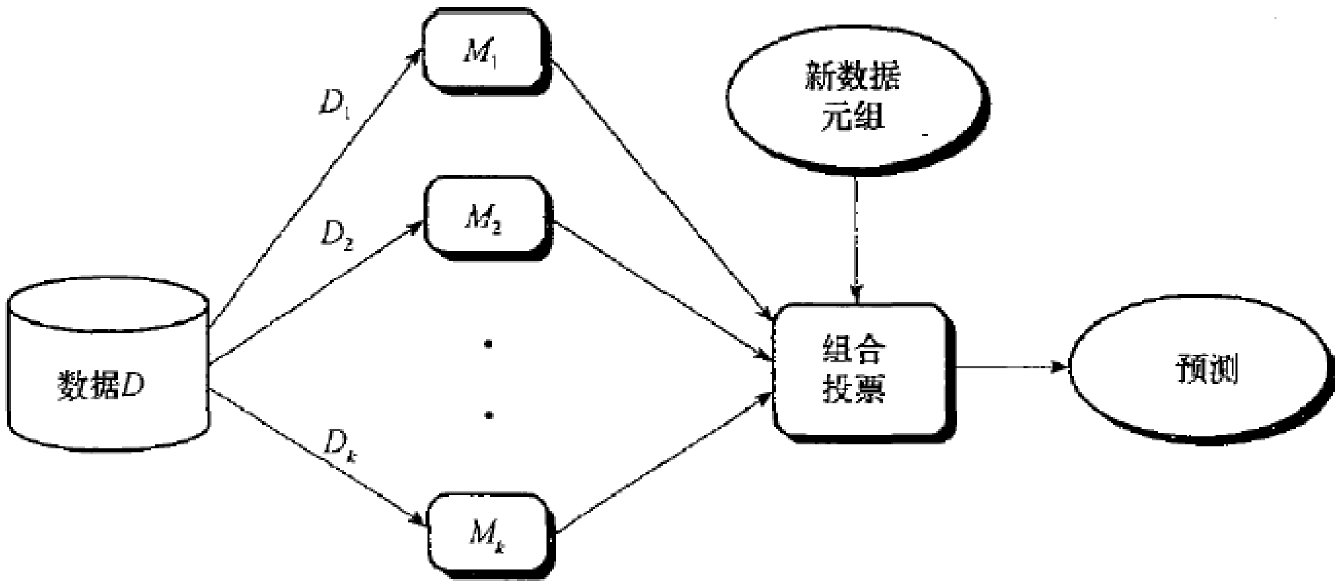
1 集成学习概述

提升分类器准确率的组合方法

- 组合方法包括：装袋（bagging）和提升（boosting）



装袋(Bagging)

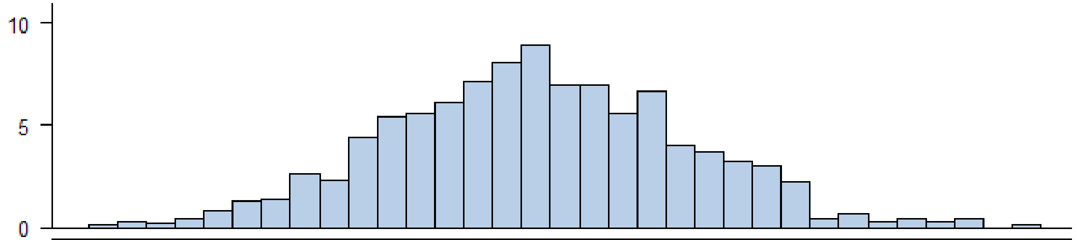


- ✓ 从样本集中重采样(有重复的)选出n个样本
- ✓ 在所有属性上，对这n个样本建立分类器(ID3、C4.5、CART、SVM、Logistic回归等)
- ✓ 重复以上两步m次，即获得了m个分类器
- ✓ 将数据放在这m个分类器上，最后根据这m个分类器的投票结果，决定数据属于哪一类

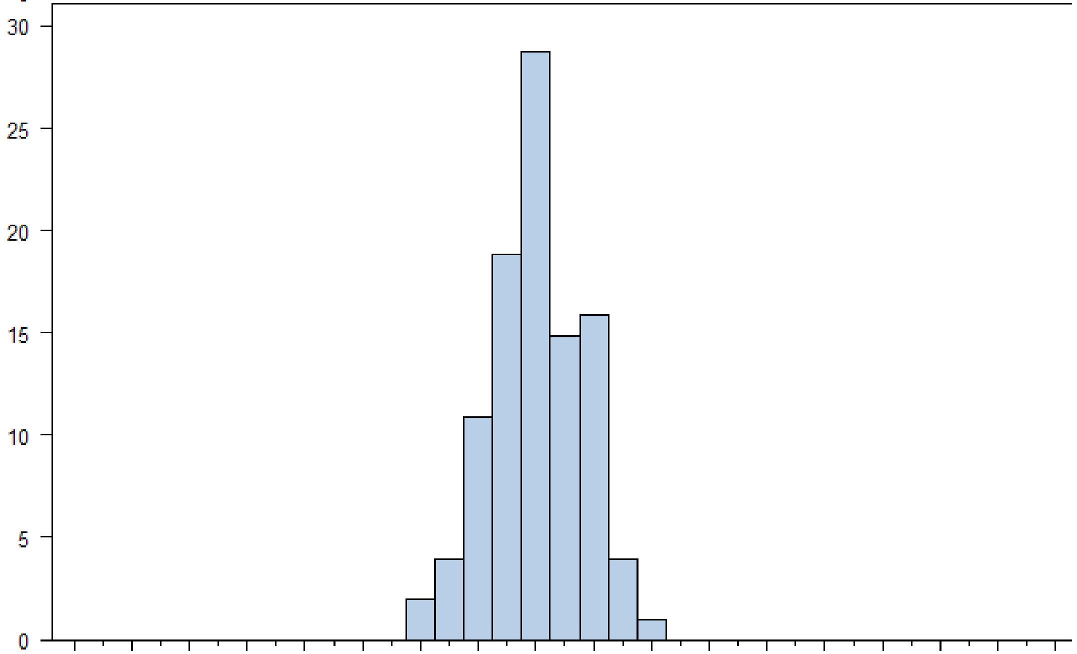
袋装算法的优势

- 准确率明显高于组合中任何单个的分类器
- 对于较大的噪音，表现不至于很差，并且具有鲁棒性
- 不容易过度拟合

随机变量分布



随机变量均值分布



天善智能 TIANSHAN SOFT

袋装的效果

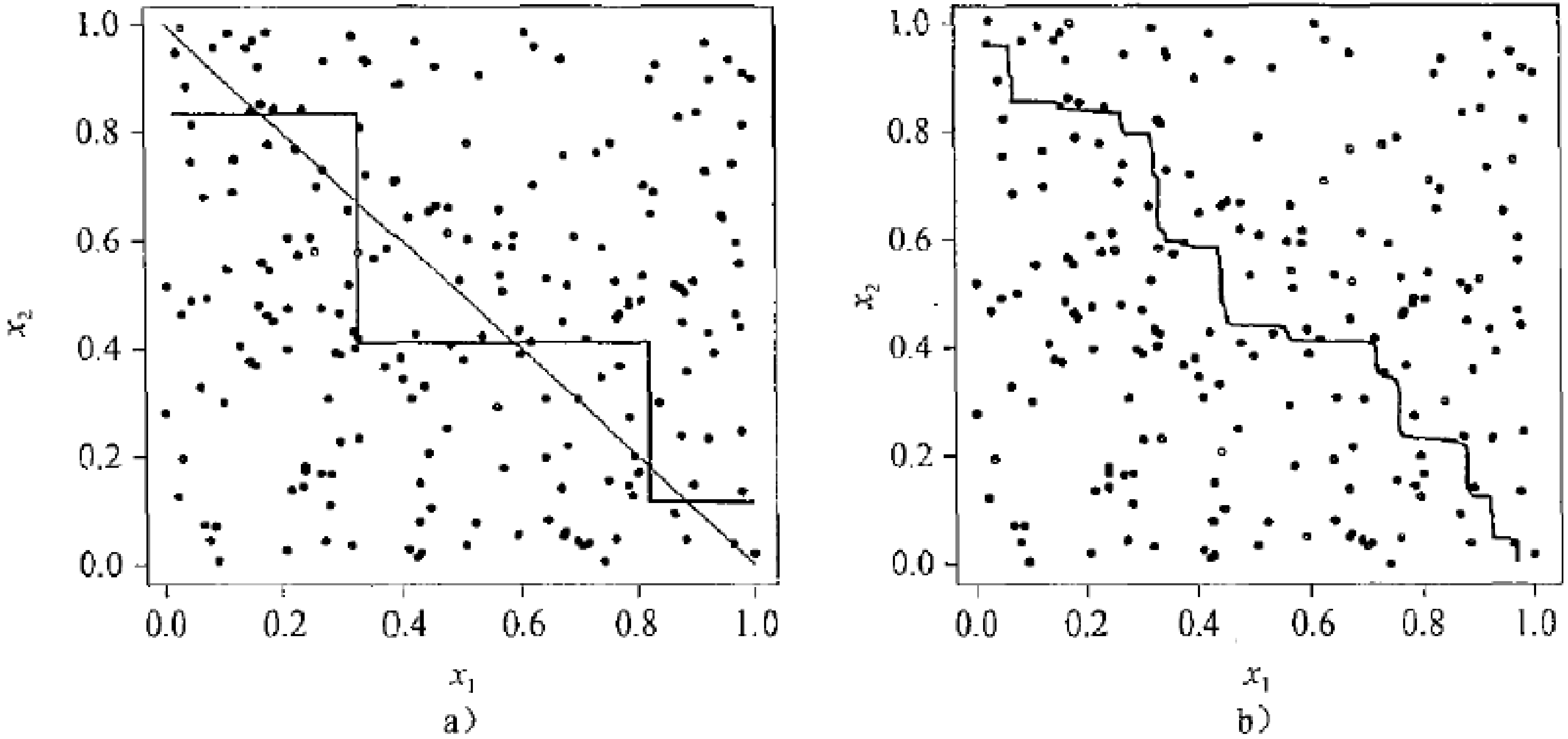
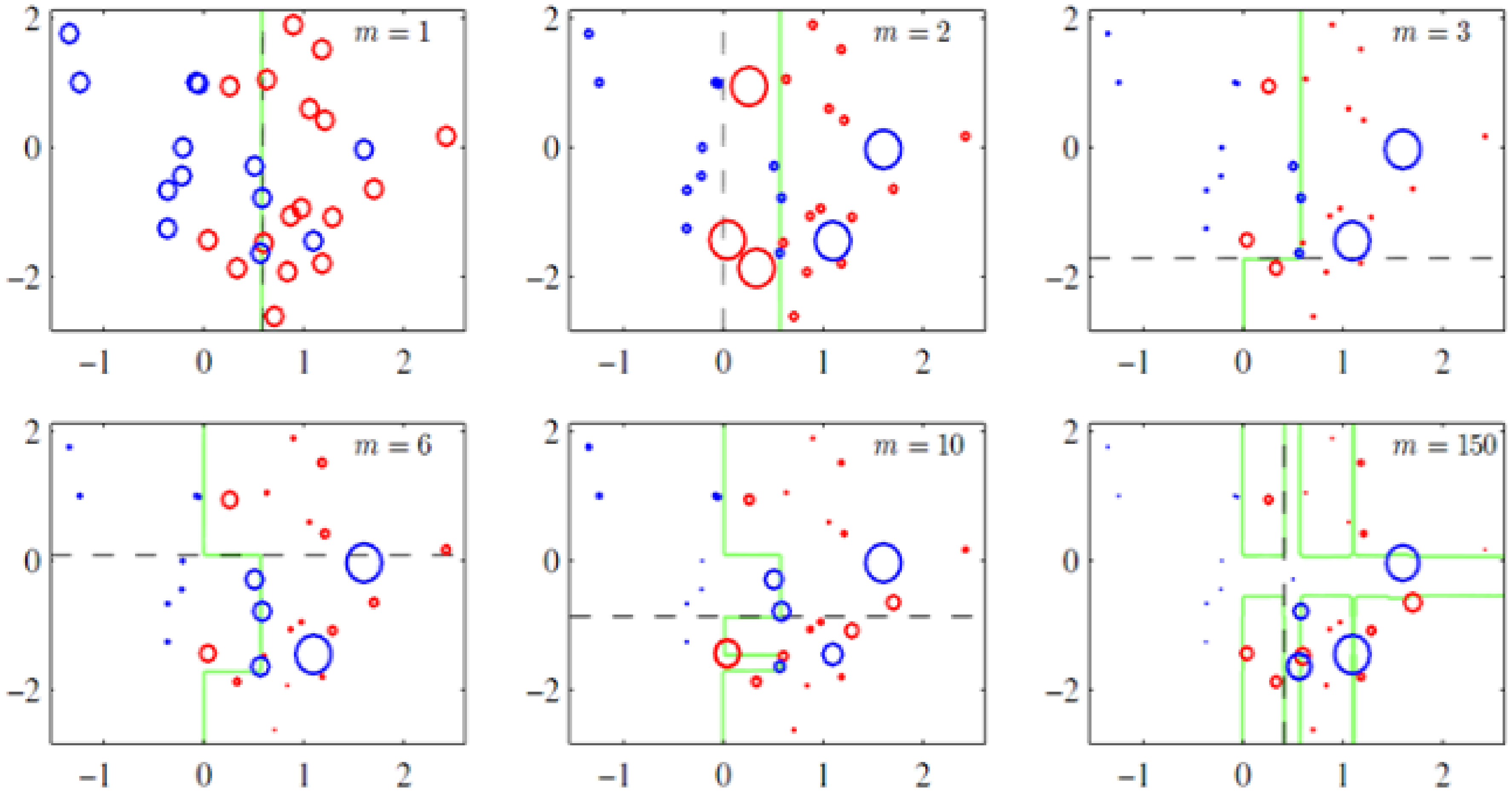


图 8.22　一个线性可分问题（即实际的决策边界是一条直线）的决策边界：a）单棵决策树；b）决策树的组合分类器。决策树努力近似线性边界。组合分类器更接近于真实的边界。取自 Seni 和 Elder[SE10]

天善智能 TIANSHAN SOFT

提升（boosting）算法思想

Boosting的过程，绿色的线表示目前取得的模型（模型是由前m次得到的模型合并得到的），虚线表示当前这次模型。每次分类的时候，会更关注分错的数据，上图中，红色和蓝色的点就是数据，点越大表示权重越高。



天善智能
TIANSHAN SOFT

提升算法的优缺点

- 可以获得比bagging更快的收敛速度
- 容易过度拟合

2 随机森林

随机森林（Random Forest）算法

- 是用装袋法在行列上进行随机抽样
- 由很多决策树分类器组合而成（因而称为“森林”）
- 单个的决策树分类器用随机方法构成。首先，学习集是从原训练集中通过有放回抽样得到的自助样本。其次，参与构建该决策树的变量也是随机抽出，参与变量数通常大大小于可用变量数。
- 单个决策树在产生学习集和确定参与变量后，使用CART算法计算，不剪枝
- 最后分类结果取决于各个决策树分类器简单多数选举

随机森林算法优点

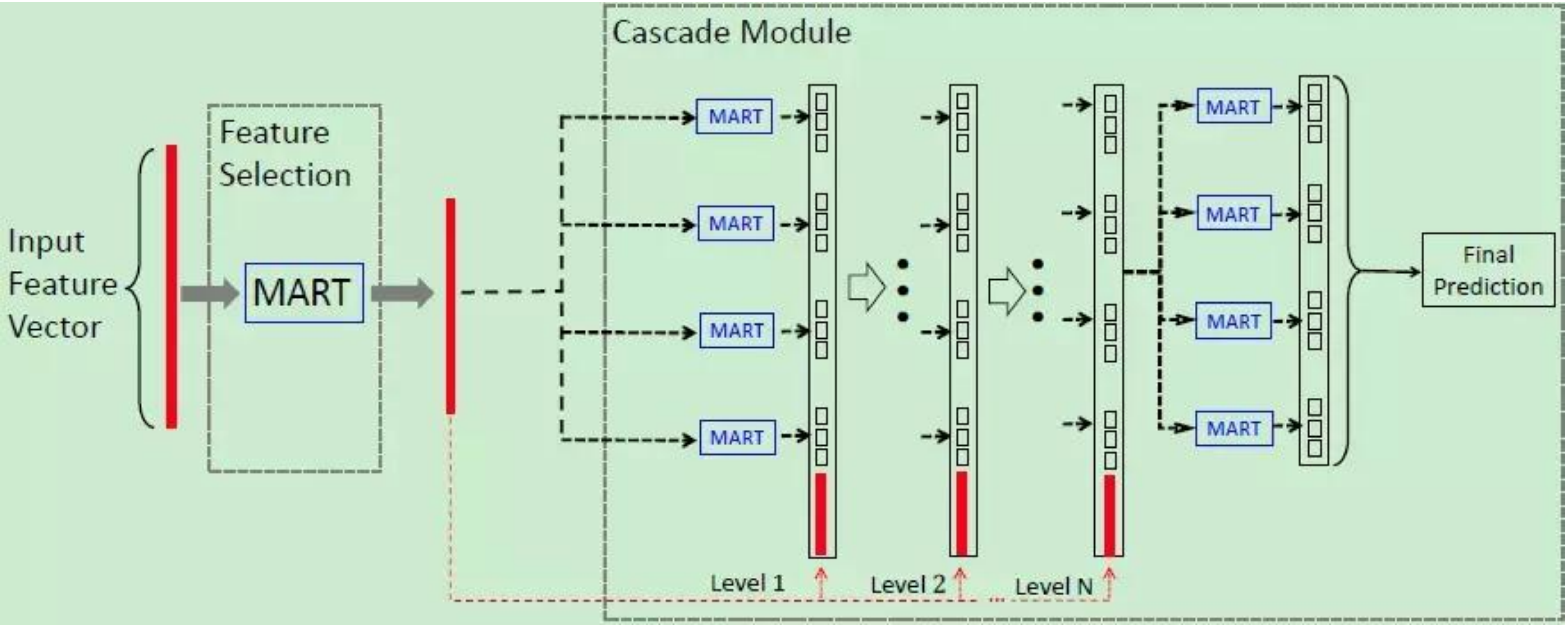
- 准确率可以和神经网络媲美，比逻辑回归高
- 对错误和离群点更加鲁棒性
- 决策树容易过度拟合的问题会随着森林规模而削弱
- 在大数据情况下速度快，性能好

https://mp.weixin.qq.com/s/fq8Y57i0RjI7smj32Zd1fA

天善智能 TIANSHAN SOFT

深度随机森林

- 周志华团队和蚂蚁金服合作：用分布式深度森林算法检测套现欺诈



https://mp.weixin.qq.com/s/fq8Y57i0RjI7smj32Zd1fA

3 Adaboost算法

Adaboost算法

对于二分类问题，给定训练样本$\{(x_i, y_i)|i = 1, 2, \cdots, N\}$，$y_i \in \{-1, +1\}$。AdaBoost训练若干棵分类树$G_m(x)$，每棵分类树都是个弱分类器，这些弱分类器线性加权组合在一起构成一个强分类器$f(x) = \sum_{m=1}^{M} \alpha_m G_m(x)$。

算法步骤：

1.初始时每个样本赋予相同的权重$w_{1i} = \frac{1}{N}$

2.进行M轮迭代，第m轮迭代产生一个弱分类器$G_m(x)$

(1)采用使分类错误率最低的方法找到最佳的切分点，形成弱分类器$G_m(x) : x \to \{-1, +1\}$

(2)计算$G_m(x)$在所有样本上的分类错误率

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

Adaboost算法

即分类错误率e_m是$G_m(x)$误分样本的权重之和。

(3)计算$G_m(x)$的权重α_m。

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

$e_m < \frac{1}{2}$时$\alpha_m > 0$，且e_m越小α_m 越大，即分类错误类越小的弱分类器在最终的分类器$f(x)$中权重越大。

(4)更新每一个样本的权重。在下一轮迭代中

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

Z是归一化因子，确保所有样本的权重之和为1。样本i被错误分类时$y_i G_m(x_i) < 0$，此时$w_{m+1,i} > w_{mi}$，即被错误分类的样本在下次迭代时得到更高的权重。

3.最终的分类器$G(x) = sign(f(x)) = sign\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right)$

举例

- 给定下列训练样本，试用AdaBoost算法学习一个强分类器。

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

***本例摘自:李航，统计学习方法，清华大学出版社，2012**

解

- 初始化训练数据的权值分布

- $W_{1i} = 0.1$

$$D_1 = \left(w_{11}, w_{12} \Lambda\ w_{1i} \Lambda\ , w_{1N}\right),\ w_{1i} = \frac{1}{N},\ i = 1,2,\Lambda\ , N$$

m=1

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

- 对于m=1
- 在权值分布为D1的训练数据上，阈值v取2.5时误差率最低，故基本分类器为：

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

天善智能 TIANSHAN SOFT

m=1

- G1(x)在训练数据集上的误差率e1=P(G1(xi)≠yi) = 0.3
- 计算G1的系数：

$$\alpha_1 = \frac{1}{2}\log\frac{1-e_1}{e_1} = 0.4236$$

- $f_1(x)=0.4236*G_1(x)$
- 分类器$sign(f_1(x))$在训练数据集上有3个误分类点。

m=1

- 更新训练数据的权值分布：

$$D_{m+1} = \left(w_{m+1,1}, w_{m+1,2} \Lambda \ w_{m+1,i} \Lambda \ , w_{m+1,N}\right),$$

$$w_{m+i} = \frac{w_{mi}}{Z_m} \exp\left(-\alpha_m y_i G_m\left(x_i\right)\right), \ i = 1,2,\Lambda \ , N$$

- D_2=(0.0715, 0.0715, 0.0715, 0.0715, 0.0715, 0.0715, 0.1666, 0.1666, 0.1666, 0.0715)
  - 计算D_2，是为下一个基本分类器使用
- $f_1(x)=0.4236*G_1(x)$
- 分类器sign($f_1(x)$)在训练数据集上有3个误分类点。

m=2

X	0	1	2	3	4	5	6	7	8	9
Y	1	1	1	−1	−1	−1	1	1	1	−1
w	0.0715	0.0715	0.0715	0.0715	0.0715	0.0715	0.1666	0.1666	0.1666	0.0715

- 对于m=2
- 在权值分布为D2的训练数据上，阈值v取8.5时误差率最低，故基本分类器为：

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

m=2

- G2(x)在训练数据集上的误差率e2=P(G2(xi)≠yi) = 0.2143(0.0715*3)
- 计算G2的系数：

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

m=2

- 更新训练数据的权值分布：

$$D_{m+1} = \left(w_{m+1,1}, w_{m+1,2} \Lambda\ w_{m+1,i} \Lambda\ , w_{m+1,N}\right),$$

$$w_{m+i} = \frac{w_{mi}}{Z_m} \exp\left(-\alpha_m y_i G_m\left(x_i\right)\right),\ i = 1,2,\Lambda\ , N$$

- D3=(0.0455, 0.0455, 0.0455, 0.1667, 0.1667, 0.01667, 0.1060, 0.1060, 0.1060, 0.0455)
- f2(x)=0.4236G1(x) + 0.6496G2(x)
- 分类器sign(f2(x))在训练数据集上有3个误分类点。

m=3

X	0	1	2	3	4	5	6	7	8	9
Y	1	1	1	-1	-1	-1	1	1	1	-1
w	0.0455	0.0455	0.0455	0.1667	0.1667	0.1667	0.1060	0.1060	0.1060	0.0455

- 对于m=3
- 在权值分布为D3的训练数据上，阈值v取5.5时误差率最低，故基本分类器为：

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

天善智能 TIANSHAN SOFT

m=3

- G3(x)在训练数据集上的误差率e3=P(G3(xi)≠yi) = 0.1820(0.0455*4)
- 计算G3的系数：

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

m=3

- 更新训练数据的权值分布：

$$D_{m+1}=\left(w_{m+1,1},w_{m+1,2}\Lambda\ w_{m+1,i}\Lambda\ ,w_{m+1,N}\right),$$

$$w_{m+i}=\frac{w_{mi}}{Z_m}\exp\left(-\alpha_m y_i G_m\left(x_i\right)\right),\ \ i=1,2,\Lambda\ ,N$$

- D4=(0.125, 0.125, 0.125, 0.102, 0.102, 0.102, 0.065, 0.065, 0.065, 0.125)
- f3(x)=0.4236G1(x) + 0.6496G2(x)+0.7514G3(x)
- 分类器sign(f3(x))在训练数据集上有0个误分类点。

4 提升树与梯度提升树(GBDT)

提升树

以Y为连续变量为例，演示提升树

$$f_M(x)=\sum_{m=1}^{M}T(x;\Theta_m)$$

其中，$T(x;\Theta_m)$表示决策树；Θ_m为决策树的参数；M为树的个数.

提升树得迭代算法：

$$f_m(x)=f_{m-1}(x)+T(x;\Theta_m)$$

$$\hat{\Theta}_m=\arg\min_{\Theta_m}\sum_{i=1}^{N}L(y_i,f_{m-1}(x_i)+T(x_i;\Theta_m))$$

提升树

表 8.2 训练数据表

x_i	1	2	3	4	5	6	7	8	9	10
y_i	5.56	5.70	5.91	6.40	6.80	7.05	8.90	8.70	9.00	9.05

解 按照算法 8.3，第 1 步求 $f_1(x)$ 即回归树 $T_1(x)$.

首先通过以下优化问题：

$$\min_s \left[\min_{c_1} \sum_{x_i \in R_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2 \right]$$

求解训练数据的切分点 s：

$$R_1 = \{x \mid x \leqslant s\}, \quad R_2 = \{x \mid x > s\}$$

容易求得在 R_1，R_2 内部使平方损失误差达到最小值的 c_1，c_2 为

$$c_1 = \frac{1}{N_1} \sum_{x_i \in R_1} y_i, \quad c_2 = \frac{1}{N_2} \sum_{x_i \in R_2} y_i$$

这里 N_1，N_2 是 R_1，R_2 的样本点数.

***本例摘自:李航，统计学习方法，清华大学出版社，2012**

天善智能 TIANSHAN SOFT

提升树

求训练数据的切分点．根据所给数据，考虑如下切分点：

$$1.5,\ 2.5,\ 3.5,\ 4.5,\ 5.5,\ 6.5,\ 7.5,\ 8.5,\ 9.5$$

对各切分点，不难求出相应的 R_1，R_2，c_1，c_2 及

$$m(s)=\min_{c_1}\sum_{x_i\in R_1}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2}(y_i-c_2)^2$$

例如，当 $s=1.5$ 时，$R_1=\{1\}$，$R_2=\{2,3,\cdots,10\}$，$c_1=5.56$，$c_2=7.50$，

$$m(s)=\min_{c_1}\sum_{x_i\in R_1}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2}(y_i-c_2)^2=0+15.72=15.72$$

***本例摘自:李航，统计学习方法，清华大学出版社，2012**

提升树

表 8.3 计算数据表

s	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
$m(s)$	15.72	12.07	8.36	5.78	3.91	1.93	8.01	11.73	15.74

由表 8.3 可知，当$s=6.5$时$m(s)$达到最小值，此时$R_1=\{1,2,\cdots,6\}$，$R_2=\{7,8,9,10\}$，$c_1=6.24$， $c_2=8.91$，所以回归树$T_1(x)$为

$$T_1(x)=\begin{cases}6.24, & x<6.5\\ 8.91, & x\geqslant 6.5\end{cases}$$

$$f_1(x)=T_1(x)$$

用$f_1(x)$拟合训练数据的残差见表 8.4，表中$r_{2i}=y_i-f_1(x_i)$， $i=1,2,\cdots,10$.

表 8.4 残差表

x_i	1	2	3	4	5	6	7	8	9	10
r_{2i}	−0.68	−0.54	−0.33	0.16	0.56	0.81	−0.01	−0.21	0.09	0.14

*本例摘自:李航，统计学习方法，清华大学出版社，**2012**

天善智能 TIANSHAN SOFT

提升树

用 $f_1(x)$ 拟合训练数据的平方损失误差：

$$L(y, f_1(x)) = \sum_{i=1}^{10} (y_i - f_1(x_i))^2 = 1.93$$

第 2 步求 $T_2(x)$．方法与求 $T_1(x)$ 一样，只是拟合的数据是表 8.4 的残差．可以得到：

$$T_2(x) = \begin{cases} -0.52, & x < 3.5 \\ 0.22, & x \geqslant 3.5 \end{cases}$$

$$f_2(x) = f_1(x) + T_2(x) = \begin{cases} 5.72, & x < 3.5 \\ 6.46, & 3.5 \leqslant x < 6.5 \\ 9.13, & x \geqslant 6.5 \end{cases}$$

***本例摘自:李航，统计学习方法，清华大学出版社，2012**

提升树

用 $f_2(x)$ 拟合训练数据的平方损失误差是

$$L(y, f_2(x)) = \sum_{i=1}^{10}(y_i - f_2(x_i))^2 = 0.79$$

继续求得

$$T_3(x) = \begin{cases} 0.15, & x < 6.5 \\ -0.22, & x \geqslant 6.5 \end{cases} \qquad L(y, f_3(x)) = 0.47\,,$$

$$T_4(x) = \begin{cases} -0.16, & x < 4.5 \\ 0.11, & x \geqslant 4.5 \end{cases} \qquad L(y, f_4(x)) = 0.30\,,$$

$$T_5(x) = \begin{cases} 0.07, & x < 6.5 \\ -0.11, & x \geqslant 6.5 \end{cases} \qquad L(y, f_5(x)) = 0.23\,,$$

$$T_6(x) = \begin{cases} -0.15, & x < 2.5 \\ 0.04, & x \geqslant 2.5 \end{cases}$$

***本例摘自:李航，统计学习方法，清华大学出版社，2012**

提升树

$$f_6(x) = f_5(x) + T_6(x) = T_1(x) + \cdots + T_5(x) + T_6(x)$$

$$= \begin{cases} 5.63, & x < 2.5 \\ 5.82, & 2.5 \leqslant x < 3.5 \\ 6.56, & 3.5 \leqslant x < 4.5 \\ 6.83, & 4.5 \leqslant x < 6.5 \\ 8.95, & x \geqslant 6.5 \end{cases}$$

用 $f_6(x)$ 拟合训练数据的平方损失误差是

$$L(y, f_6(x)) = \sum_{i=1}^{10} (y_i - f_6(x_i))^2 = 0.17$$

假设此时已满足误差要求，那么 $f(x) = f_6(x)$ 即为所求提升树.

***本例摘自:李航，统计学习方法，清华大学出版社，2012**

GBDT （Bradient Boosting Decision Tree）

BoostedTree是把boost思想用在树的生长上面，即子节点的预测值$y^{(t)}$是在父节点的预测值$y^{(t-1)}$的基础上前进一小步。如果把boost思想用在森林的扩展上面就是GBDT，即第t轮迭代的预测值依赖于第t-1轮的预测结果：

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)$$

其中$f(x)$表示一棵决策树，第t轮迭代生成一棵树f_t，第t轮迭对样本x_i的预测值等于第t-1轮迭代对x_i的预测值加上f_t对x_i的预测值。核心问题是：f_t取多少才能使$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$尽可能地接近真实值$y_i$呢？这变为一个优化问题:

$$\underset{f_t(x_i)}{arg\ min}\ \ loss(y_i, \hat{y}_i^{(t-1)} + f_t(x_i))$$

GBDT

二阶泰勒展开式$f(x+\Delta x) \simeq f(x) + f'(x)\Delta x + \frac{1}{2}f''(x)\Delta x^2$

用$loss(y_i, \hat{y}_i^{(t-1)} + f_t(x_i))$类比上式的$f(x+\Delta x)$，$loss(y_i, \hat{y}_i^{(t-1)})$类比于$f(x)$，$\hat{y}_i^{(t-1)}$类比于x，$f_t(x_i)$类比于$\Delta x$，$y_i$是已知常量。

令$g_i = \frac{\partial loss(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}, h_i = \frac{\partial^2 loss(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)2}}$得

$$loss(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) = loss(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)$$

二次函数在

$$f_t(x_i) = -\frac{g_i}{h_i} \tag{2}$$

处取得极小值，这与牛顿法的结论是一样的，GBDT中的Gradient就是从此而来。

$f_t(x_i)$确定之后就可以采用上文讲到过的回归树或Boosted回归树来拟合$f_t(x_i)$。

GBDT

损失函数$loss$有多种取法，当取平方误差时

$$loss(y_i, \hat{y}_i^{(t-1)}) = (\hat{y}_i^{(t-1)} - y_i)^2$$

此时$g_i = 2(\hat{y}_i^{(t-1)} - y_i), h_i = 2$

从而$f_t(x_i) = -\frac{g_i}{h_i} = y_i - \hat{y}_i^{(t-1)}$，所以很多地方把GBDT叫做基于残差的训练方法就是基于平方误差推导出来的。

例如$y_i = 10$，第1次迭代采用普通的回归树得到$y_i^{(1)} = f_1(x_i) = 8$。第2次迭代时我们构造一棵回归树使得$f_2(x_i)$尽可能逼近于$10 - 8 = 2$，假设训练得到$f_2(x_i) = 3$，那第2次迭代结束后我们对y_i的预测值就是$\hat{y}_i = y_i^{(2)} = y_i^{(1)} + f_2(x_i) = 11$。第3次迭代构造回归树时使得$f_3(x_i)$尽可能逼近于$10 - 11 = -1$。依此类推。

GBDT

例如$y_i = 10$，λ取0.1。第1次迭代采用普通的回归树得到$f_1(x_i) = 8, y_i^{(1)} = 0 + 0.1 * f_1(x_i) = 0.8$。第2次迭代时我们构造一棵回归树使得$f_2(x_i)$尽可能逼近于$10 - 0.8 = 9.2$，假设训练得到$f_2(x_i) = 7$，那第2次迭代结束后我们对$y_i$ 的预测值就是$\hat{y}_i = y_i^{(2)} = y_i^{(1)} + 0.1 * f_2(x_i) = 0.8 + 0.1 * 7 = 1.5$。第3次迭代构造回归树时使得$f_3(x_i)$ 尽可能逼近于$10 - 1.5 = 8.5$。依此类推。

GBDT

算法	试验20次R2平均值	参数设置	训练耗时/秒
RegressionTree	0.939656	J=400,epsilon=0.001	0.059
BoostedTree	0.948220	J=400,epsilon=0.001,shrinkage=0.25,bagging=0.5	0.083
GBDT	0.956198	J=6,M=40,epsilon=0.001,shrinkage=0.1	2.202

Table 2: 各种回归树效果对比

$$R2 = \frac{Var(y) - MES(y, \hat{y})}{Var(y)} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}$$

训练数据

200条数据取自：

math.pow(x-50,2)+300*np.random.rand()

测试数据

200条数据取自：

math.pow(x-50,2)

GBDT

对于二分类问题，$y \in \{0,1\}$，设$P_+ = p(y=1), P_- = p(y=0)$，则伯努力试验的似然函数为$P_+^y P_-^{1-y}$，转化为对数似然$ylnP_+ + (1-y)lnP_-$，极大化对数似然等价于

$$min \quad -ylnP_+ - (1-y)lnP_- \tag{3}$$

而$-ylnP_+ - (1-y)lnP_-$ 正好就是交叉熵。

GBDT是回归树，它的输出是任意实数$f(x) \in [-\infty, +\infty]$，如果把$f(x)$传给一个logistic函数就可以实现二分类。

$$P_+ = \frac{1}{1+e^{-f(x)}}, P_- = \frac{1}{1+e^{f(x)}} \tag{4}$$

把(4)式代入(3)式得

GBDT

$$min\ \ loss = yln(1+e^{-f(x)}) + (1-y)ln(1+e^{f}(x)) \tag{5}$$

$$\frac{\partial loss}{\partial f} = \frac{-y}{1+e^{-f(x)}}e^{-f(x)} + \frac{1-y}{1+e^{f(x)}}e^{f(x)}$$

$$= \frac{-y}{1+e^{f(x)}} + \frac{1-y}{1+e^{-f(x)}} = -yP_{-} + (1-y)P_{+} = P_{+} - y$$

$$\frac{\partial^2 loss}{\partial f^2} = \frac{\partial P_{+} - y}{\partial f} = \frac{\partial P_{+}}{\partial f}$$

$$= \frac{e^{-f(x)}}{(1+e^{-f(x)})^2} = \frac{1}{(1+e^{f(x)})(1+e^{-f(x)})} = P_{+}P_{-}$$

由(2)式得$f_t(x) = -\frac{P_{+}-y}{P_{+}P_{-}} = \frac{\frac{y}{P_{+}}-1}{P_{-}}$，这就是第t轮迭代回归树要拟合的值。

天善智能 TIANSHAN SOFT

XGBoost

在GBDT的基础上自剪枝的决策树

GBDT每次新建的树并没有力图构建一个最合理的新树。Xgboost则希望新加入的树是最优的。
其每次建树是在最优化以下的目标函数：

$$Obj(\Theta) = \underbrace{L(\Theta)}_{\text{误差函数：我们的模型有多拟合数据}} + \underbrace{\Omega(\Theta)}_{\text{正则化项：惩罚复杂模型}}$$

连续被变量时的损失函数为：

$$L(\theta) = \sum_i (y_i - \hat{y}_i)^2$$

二分类被变量时的损失函数为：

$$L(\theta) = \sum_i [y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i)\ln(1 + e^{\hat{y}_i})]$$

正则项

$$\Omega(f_t) = \gamma \underbrace{T}_{\text{叶子的个数}} + \frac{1}{2}\lambda \underbrace{\sum_{j=1}^{T} w_j^2}_{\text{w的L2模平方}}$$

通过加法模型的泰勒展开：

$$Obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T$$

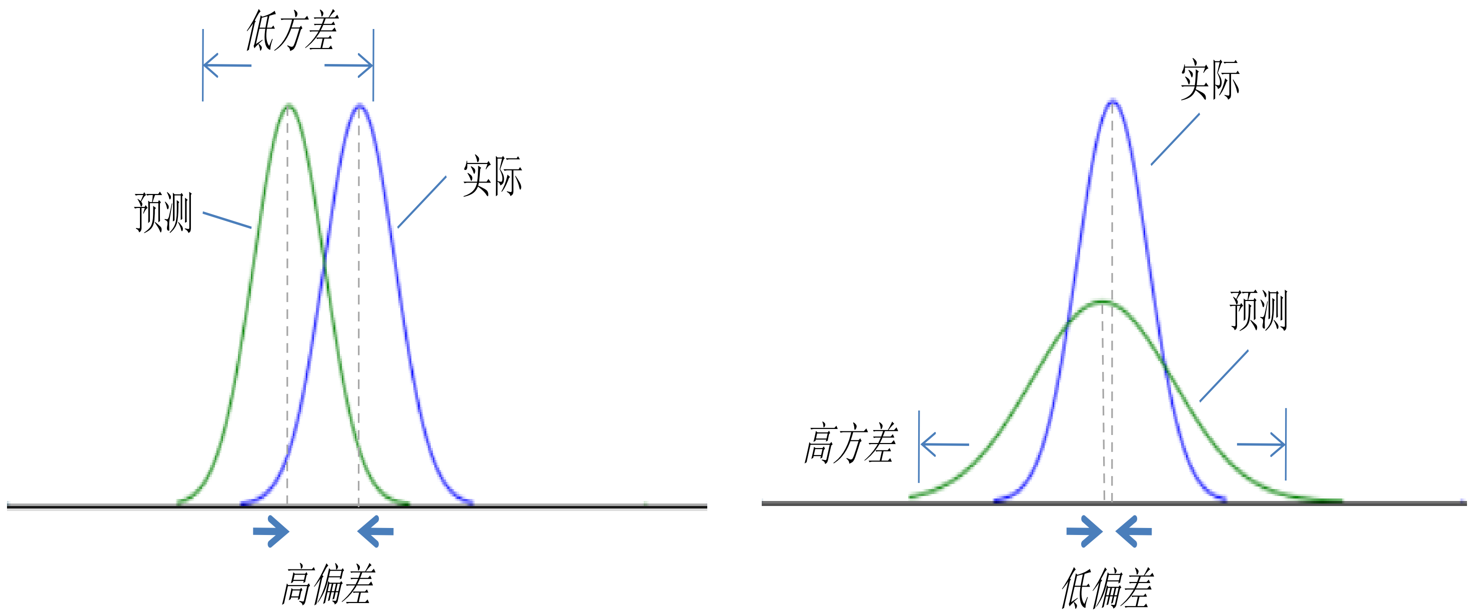
等价于CART的剪枝策略：

$$Gain = \frac{1}{2}\left[\underbrace{\frac{G_L^2}{H_L+\lambda}}_{\text{左子树分数}} + \underbrace{\frac{G_R^2}{H_R+\lambda}}_{\text{右子树分数}} - \underbrace{\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}}_{\text{不分割我们可以拿到的分数}}\right] - \underbrace{\gamma}_{\text{加入新叶子节点引入的复杂度代价}}$$

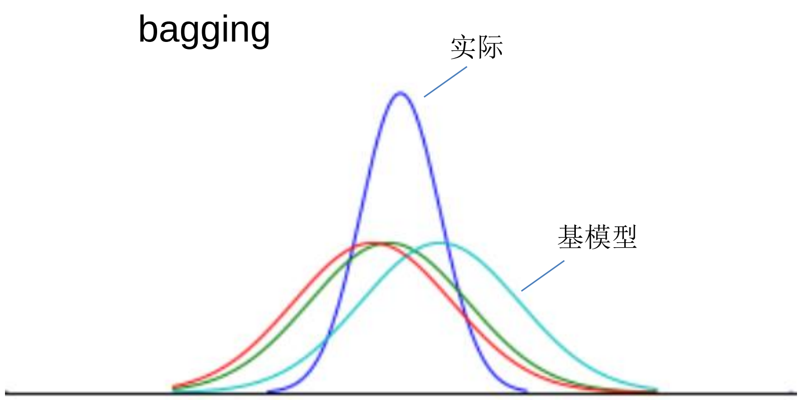
详细算法介绍参见:http://xgboost.readthedocs.io/en/latest/model.html

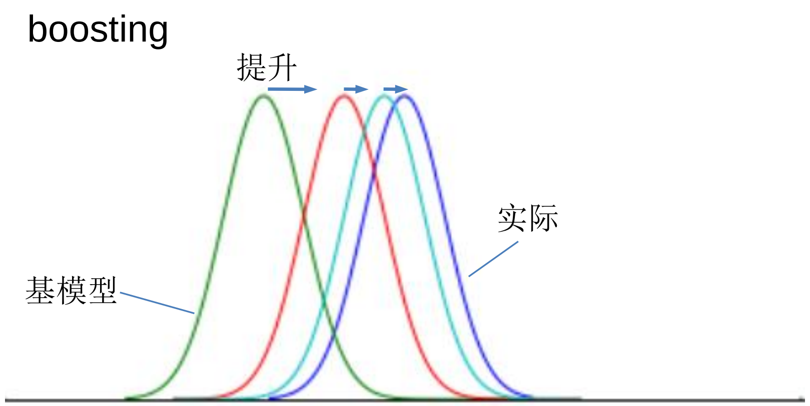
5 偏差（Bias）-方差（Variance）权衡与集成方法

偏差-方差权衡



bagging与boosting的直观理解





更多商业智能BI和大数据精品视频尽在 www.hellobi.com

BI、商业智能
数据挖掘 大数据
数据分析师
R语言 Python
机器学习
深度学习
人工智能
Hive Hadoop
Tableau
BIEE ETL
数据科学家
PowerBI

天善智能
TIANSHAN SOFT