集成学习

Bagging (bootstrapp aggregating的缩写) ⇒降低方差.

1) Random Forest

行随机：使用bootstrap方法有放回的采样N个包含N个样本的数据集.

列随机：从所有属性中选择m个属性进行比较，选择最好的.

每棵树尽可能的生长，没有剪枝过程.

特点：

1° 有极好的准确率.

2° 能有效处理高维特征，不需要降维.

3° 对缺失值不敏感.

Random Forest 错误率相关因素. → 如何选择m个属性，RF中的性一个参数.

1° 森林中任意2个树之间的相关性，相关性↑，错误率↑.

2° 每棵树的分类能力，错误率↓.

选择m ⇒ 袋外错误率 (oob error)

随机森林不需要交叉验证或额外的测试集，因为内部估计.

bootstrap采样方法使得对每棵树，约有0.368数据没有用于训练，它们称为该棵树的oob样本.

1) 对于每个样本，计算其作为oob样本的树对它的分类情况.

2) 使用多数投票作为预测结果

3) 用误分样本个数/总样本个数作为模型的oob error.

↳ oob error是RF的泛化无偏估计，适用于需要大量计算的比较及验证.

Boosting ⇒ 降低偏差

~~1) AdaBoost~~. 思想：先从容易的样例学到弱分类器，然后通过反复学习得到一个强分类器.

- 加法模型：强分类器由若干弱分类器的权值组合而得. $F_M = \sum_{m=1}^{M} \alpha_m h(x)$
- 前向分步算法：T-1轮迭代产生的分类器由上一轮迭代得到. $F_{m+1} = F_m + \alpha_m h(x)$

1) AdaBoost.

1° 给每个训练样本赋予其权重，初始时训练样本权重相同.

2° 每次训练完毕后，对分类错误的样本赋予较大的权重，让模型更关注该样本.

3° 根据分类器的错误率赋予权重，采用加权多数表决的方法.

给定数据集：$T=\{(x_1,y_1),\cdots,(x_N,y_N)\}$, $y \in \{-1,+1\}$,

(1) 初始化权值分布.

$$w_1 = (w_{1,1}, w_{1,2}, \cdots, w_{1,N}), \quad w_{1,i} = \frac{1}{N}.$$

(2) 对于 m 次迭代.

(a) 得到分类器 $G_m(x)$.

(b) 计算 $G_m(x)$ 的误差：$e_m = \sum_{i=1}^{N} w_{m,i} I(G_m(x_i) \neq y_i)$

⇒ 分类器分错则其权重变小.

(c) 计算 $G_m(x)$ 在强分类器中的权重：$\alpha_m = \frac{1}{2}\log\frac{1-e_m}{e_m} = \frac{1}{2}\log\left(\frac{1}{e_m}-1\right)$

(d) 更新样本的权值分布：

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

规范化因子.

$\begin{cases} w_{m,i}\cdot e^{-\alpha_m} \\ w_{m,i}\cdot e^{+\alpha_m} \end{cases}$ ⇒ 分对了 $-\alpha_m$ 小于0，$e^{-\alpha_m}$ 小于1，$w_{m,i}\cdot e^{-\alpha_m}$ 变小，

⇒ 则分对的样本在下一次迭代中权重变小.

(3) 得到强分类器

$$F(x) = \mathrm{sign}\left(\sum_{m=1}^{M}\alpha_m G_m(x)\right)$$

Adaboost 训练误差上界：

$$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i)\neq y_i) \leq \frac{1}{N}\sum_{i=1}^{N} \exp(-y_i f(x_i)) = \prod_m Z_m$$

证明：

1) 若 $G(x_i)\neq y_i$，则 $y_i f(x_i) < 0$，有 $\exp(-y_i f(x_i)) \geq 1$

$\therefore \frac{1}{N}\sum_{i=1}^{N} I(G(x_i)\neq y_i) \leq \frac{1}{N}\sum_{i=1}^{N} \exp(-y_i f(x_i))$ 得证.

2) $\because w_{m+1,i} = \frac{w_m}{Z_m}\exp(-\alpha_m y_i G_m(x_i))$

$\therefore Z_m w_{m+1,i} = w_m \exp(-\alpha_m y_i G_m(x_i))$

$\therefore \frac{1}{N}\sum_{i=1}^{N} \exp(-y_i f(x_i))$

$= \frac{1}{N}\sum_{i=1}^{N} \exp\left(-\sum_{m=1}^{M} \alpha_m y_i G_m(x_i)\right)$

$= \sum_{i=1}^{N} w_{1,i} \prod_{m=1}^{M} \exp(-\alpha_m y_i G_m(x_i))$ $\Rightarrow w_{1,i} = \frac{1}{N}$.

$= \sum_{i=1}^{N} w_{1,i} \exp(-\alpha_1 y_i G_1(x_i)) \prod_{i=2}^{M} \exp(-\alpha_m y_i G_m(x_i))$

$= \sum_{i=1}^{N} Z_1 w_{2,i} \prod_{i=2}^{M} \exp(-\alpha_m y_i G_m(x_i))$ $\Rightarrow Z_m w_{m+1,i} = w_{m,i}\exp(-\alpha_m y_i G_m(x_i))$

$= Z_1 \sum_{i=1}^{N} w_{2,i} \prod_{i=2}^{M} \exp(-\alpha_m y_i G_m(x_i))$

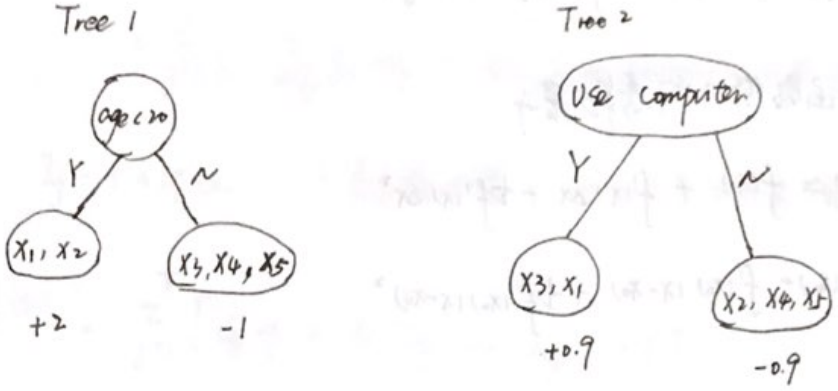
...

$= Z_1 Z_2 \cdots Z_{m-1} \sum_{i=1}^{N} w_{m+1,i} \exp(-\alpha_m y_i G_m(x_i))$

$= \prod_{m=1}^{M} Z_m$

2) XGBOOST

1° Decision Tree Ensembles



$\Rightarrow f(x_1) = 2 + 0.9 = 2.9 \qquad f(x_5) = -1 - 0.9 = -1.9$

模型: $\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$, $f_k \in F$

目标函数: $obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$

2° Tree Boosting

目标函数: $obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i)$

↳ 需要学习(t)棵树, 但是~~无法一次~~一次性学习所有树结构是很难的

↳ 加法模型

$$\hat{y}_i^{(0)} = 0$$
$$\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i)$$
$$\hat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i)$$
$$\vdots$$
$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)$$

$$\therefore obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i)$$

$$= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant$$

求解过程 ⇒ 对损失函数做二阶泰勒展开:

$$\begin{cases} f(x+\Delta x) \simeq f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2 \\ f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2} f''(x_0)(x-x_0)^2 \end{cases}$$

$$\therefore obj^{(t)} = \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + constant$$

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$$

$$h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$$

$$\Rightarrow obj^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

3° Model Complexity

假设定义 $f_t(x) = w_{q(x)},\ w \in R^T,\ q: R^d \to \{1, 2, \dots, T\}$.

→ 样本属于的一个叶子节点

↓ T维的向量, 表示每个叶子节点的得分.

则模型复杂度定义为: $\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$

~~实际使用时~~

4° Structure Score

$$obj^{(t)} \approx \sum_{i=1}^{n}\left[g_i w_{q(x_i)} + \frac{1}{2}h_i w_{q(x_i)}^2\right] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2$$

$$= \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)w_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)w_j^2\right] + \gamma T$$

$I_j = \{i \mid q(x_i) = j\}$ 表示属于第 j 个叶子节点的样本下标.

$$\Rightarrow obj^{(t)} = \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}(H_j + \lambda)w_j^2\right] + \gamma T$$

求导得到:

$$w_j^* = -\frac{G_j}{H_j + \lambda}$$

⇒ 实际使用时会乘以学习率防止过拟合.

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

5° 属性选择

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma$$

分节点划分后的方差比增加一个节点的代价(γ)高.

代码用例与说明：

假设有5个样本，2个特征，$y \in \{0,1\}$。使用CART作为基学习器，树的最大深度为3，棵树为5，学习率为0.1，正则化参数 $\lambda=1$，$\gamma=0$。损失函数使用logloss。

1. 损失函数

$$L(y_i, \hat{y}_i) = y_i \ln(1+e^{-\hat{y}_i}) + (1-y_i)\ln(1+e^{\hat{y}_i})$$

$$\therefore \frac{\partial L(y_i,\hat{y}_i)}{\partial \hat{y}_i} = y_i \frac{-e^{-\hat{y}_i}}{1+e^{-\hat{y}_i}} + (1-y_i)\frac{e^{\hat{y}_i}}{1+e^{\hat{y}_i}}$$

$$= y_i \frac{-1}{1+e^{\hat{y}_i}} + (1-y_i)\frac{1}{1+e^{-\hat{y}_i}}$$

$$= y_i\left(\frac{1}{1+e^{-\hat{y}_i}} - 1\right) + (1-y_i)\frac{1}{1+e^{-\hat{y}_i}}$$

$$= y_i(y_{i,pred} - 1) + (1-y_i)y_{i,pred}$$

$$= y_{i,pred} - y_i, \Rightarrow g_i$$

$$\therefore \frac{\partial^2 L(y_i,\hat{y}_i)}{\partial^2 \hat{y}_i} = y_{i,pred}(1-y_{i,pred}) \Rightarrow h_i$$

2. 计算每一个样本的 g_i 和 h_i。

1° 每个样本之间相互独立，列并行 ⇒ XGBOOST快。

2° 每次迭代不需要重新计算 g_i 和 h_i

*第一次迭代时，$y_{i,pred}$ 初始化为0.5。⇒可以是(0,1)中任意一个

3. 遍历每一个特征，对特征值排序，然后依次从左往右移动划分点计算 Gain 值。

找到 Gain 最大的划分。

时间复杂度：$m \cdot n\log n$。

不断重复以上步骤，直至到达最大深度为。

4. 对于叶子节点，其对应的分数值为 $w^* = -\frac{G_j}{H_j + \lambda}$

实际上，叶子节点的取值还需要乘以一个学习率 $\Rightarrow w^* \cdot$ 学习率。

这是为了防止过拟合。

5. 接着构造第二棵树，它的样本对应的初始 $y_{i,pred}$ 由下式给出：

$$[redacted] = \hat{y}_i^{(1)} = f_0(x_i) + f_1(x_i) = 0 + w_{q(x_i)}$$

$$y_{i,pred} = \frac{1}{1+e^{-\hat{y}_i^{(1)}}}$$

$$0.5 = \frac{1}{1+e^{-f_0(x_i)}} \Rightarrow f_0(x_i) = 0$$