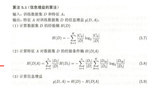
Ensemble model

prerequisite

decision tree

特征选择

- 熵
  - $H(p) = -\sum_{i=1}^{n} p_i \log p_i$
  - 熵越大，随机变量的不确定性就越大
- 信息增益
  - 定义：$g(D,A) = H(D) - H(D|A)$
  - 如何根据信息增益选择特征？ —— 对训练集D，计算其每个特征A的信息增益，并比较他们的大小，选择信息增益最大的特征

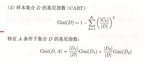


K是分类问题的类别(如0/1)，C_k是属于该类别的个数，A是特征，n个特征下有很多类别(如年龄特征下就有年轻中年老年)，D_i就是每个类别的数量,D_ik就是该i类别的0/1个数。

缺点：会偏向选择取值较多的特征

- 信息增益比

- 基尼系数



决策树生成

- ID3(基于信息增益最大)
- C4.5(基于信息增益比最大)
- CART(基于基尼系数最小)

决策树剪枝

- 预剪枝
- 后剪枝

集成模型主要思想 —— 人多力量大

bagging

思想

- 所有基础模型都一致对待（民主）
- 特性：经过bagging得到的结果方差更小

流程

从原始样本集中抽取训练集。每轮从原始样本集中使用Bootstraping的方法抽取n个训练样本（在训练集中，有些样本可能被多次抽取到，而有些样本可能一次都没有被抽中），共进行k轮抽取，得到k个训练集。（k个训练集之间是相互独立的）

每次使用一个训练集得到一个模型，k个训练集共得到k个模型。（注：这里并没有具体的分类算法或回归方法，我们可以根据具体问题采用不同的分类或回归方法，如决策树、感知器等）

对分类问题：将上步得到的k个模型采用投票的方式得到分类结果；对回归问题，计算上述模型的均值作为最后的结果。（所有模型的重要性相同）

算法 —— random forest

随机森林是由很多决策树构成的，不同决策树之间没有关联。

当我们进行分类任务时，新的输入样本进入，就让森林中的每一棵决策树分别进行判断和分类，每个决策树会得到一个自己的分类结果，决策树的分类结果中哪一个分类最多，那么随机森林就会把这个结果当做最终的结果。

boosting

思想

- 挑选出精英模型，精英获得更多权值
- 特征：经过boosting得到的结果bias更小

流程

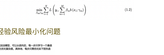
- 利用初始训练样本集训练得到一个基学习器
- 提高被基学习器误分的样本的权重，使得那些被错误分类的样本在下一轮训练中可以得到更大的关注，利用调整后的样本训练得到下一个基学习器
- 重复上述步骤，直至得到 M个学习器
- 对于分类问题，采用有权重的投票方式；对于回归问题，采用加权平均得到预测值。

数学表达

Boosting方法使用加法模型和前向分步算法

- 加法模型

- 学习加法模型



- 前向分步算法
  - 用于求解加法模型的经验风险最小化问题

  - 从前往后，一次只学习一个基函数及系数

算法

- Adaboost
- Gradient boosting

Boosting Decision Tree(BDT)

- 以决策树为基函数的boosting方法
- 模型
  - 对决策树的新定义

  - 按照boosting的定义可得出

  - 当使用平方误差损失函数时，通过拟合残差r来学习回归树

- 算法

GBDT

- Gradient Boosting
  - 思想

根据当前模型损失函数的付梯度信息来训练新加入的弱分类器，然后将训练好的弱分类器以累加的形式结合到现有模型中。

  - 算法

- GBDT

GBDT使用损失函数的负梯度在当前模型的值作为回归问题提升决策树算法中残差的近似值

- 算法

- 例子

XGBoost

- 数学推导
  - 对决策树的新定义

  - 目标函数（加上正则项）

  - 对第t轮目标函数进行二阶泰勒展开

  - 将目标函数表示为按叶节点累加的形式

  - 得到每个叶节点的最优分数从而得到最优化目标函数值

  - 用分裂后的损失减少量评估待分裂节点

- 解释与理解 —— 二阶泰勒展开

lightGBM

CatBoost

stacking

Question

1. bagging和boosting有什么主要区别？

- 样本选择
  - Bagging：训练集是在原始集中有放回选取的，从原始集中选出的各轮训练集之间是独立的。
  - Boosting：每一轮的训练集不变，只是训练集中每个样例在分类器中的权重发生变化。而权值是根据上一轮的分类结果进行调整。
- 样例权重
  - Bagging：使用均匀取样，每个样例的权重相等
  - Boosting：根据错误率不断调整样例的权值，错误率越大则权重越大。
- 预测函数
  - Bagging：所有预测函数的权重相等。
  - Boosting：每个弱分类器都有相应的权重，对于分类误差小的分类器会有更大的权重。
- 并行计算
  - Bagging：各个预测函数可以并行生成
  - Boosting：各个预测函数只能顺序生成，因为后一个模型参数需要前一轮模型的结果。

2. bias 和variance有什么区别