算法核心原理

- 决策树的核心思想：相似的输入产生相似的输出。

年龄	学历	经历	性别	==>	薪资
1	1	1	1	==>	6000（低）
2	1	3	1	==>	10000（中）
3	3	4	1	==>	50000（高）
1	3	2	1	==>	15000（中）
...	...	...	...	==>	...
1	3	2	2	==>	?

CART分类树

CART分类树算法对每个特征进行二分，寻找分割点时使用基尼系数来表达数据集的不纯度，基尼系数越小，不纯度越低，数据集划分的效果越好。

知识讲解

CART分类树划分子表的过程：

针对每个特征，基于基尼系数计算最优分割值。在计算出来的各个特征的每个分割值对数据集D的基尼系数中，选择基尼系数最小的特征A和对应的分割值a。根据这个最优特征和最优分割值，把数据集划分成两部分D_1和D_2，同时建立当前节点的左右节点，左节点的数据集D为D_1，右节点的数据集D为D_2。对左右的子节点递归调用这个过程，生成决策树。

而CART分类树也基于基尼系数来决定子表划分所选特征的次序。

基尼系数

对于样本D，个数为|D|，假设K个类别，第k个类别的数量为$|C_k|$，则样本D的基尼系数表达式：

知识讲解

$$\text{Gini}(D) = 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{|D|} \right)^2$$

有100个样本(D_1)包含A与B两个类别，数量分别为40与60，Gini（D_1）= ？

$$1 - \left(\left(\frac{40}{100} \right)^2 + \left(\frac{60}{100} \right)^2 \right) = 1 - (0.16 + 0.36) = 0.48$$

有100个样本(D_2)包含A与B两个类别，数量分别为10与90，Gini（D_2）= ？

$$1 - \left(\left(\frac{10}{100} \right)^2 + \left(\frac{90}{100} \right)^2 \right) = 1 - (0.01 + 0.81) = 0.18$$

基尼系数（续1）

对于样本D，个数为|D|，根据特征A的某个值a，把D分成$|D_1|$和$|D_2|$，则在特征A的条件下，样本D的基尼系数表达式为：

$$\text{Gini}(D,\text{A}) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

决策树的生成过程

算法输入训练集D，基尼系数的阈值，样本个数阈值。输出决策树T。

(1)对于当前节点的数据集为D，如果样本个数小于阈值，则返回决策子树，当前节点停止递归。

(2)计算样本集D的基尼系数，如果基尼系数小于阈值，则返回决策树子树，当前节点停止递归。

(3)计算当前节点现有的各个特征的各个特征值对数据集D的基尼系数。

(4)在计算出来的各个特征的各个特征值对数据集D的基尼系数中，选择基尼系数最小的特征A和对应的特征值a。根据这个最优特征和最优特征值，把数据集划分成两部分D_1和D_2，同时建立当前节点的左右节点，做节点的数据集D为D_1，右节点的数据集D为D_2。

(5)对左右的子节点递归的调用1-4步，生成决策树。

预测过程：对生成的决策树做预测的时候，假如测试集里的样本A落到了某个叶子节点，而节点里有多个训练样本。则对于A的类别预测采用的是这个叶子节点里概率最大的类别。

决策树分类实现

- 决策树分类器模型相关API：

```
import sklearn.tree as st

# 决策树分类器
model = st.DecisionTreeClassifier(
        max_depth=6, min_samples_split=3, random_state=7)
model.fit(train_x, train_y)
```

知识讲解

集合模型分类实现

- 集合模型提供的常用分类器：

```
import sklearn.ensemble as se
```

- model = se.RandomForestClassifier(...) 随机森林分类器
- model = se.AdaBoostClassifier(...) AdaBoost分类器
- model = se.GridientBoostingClassifier(...) GBDT分类器

知识讲解