集成模型之随机森林(二)

1 随机森林原理

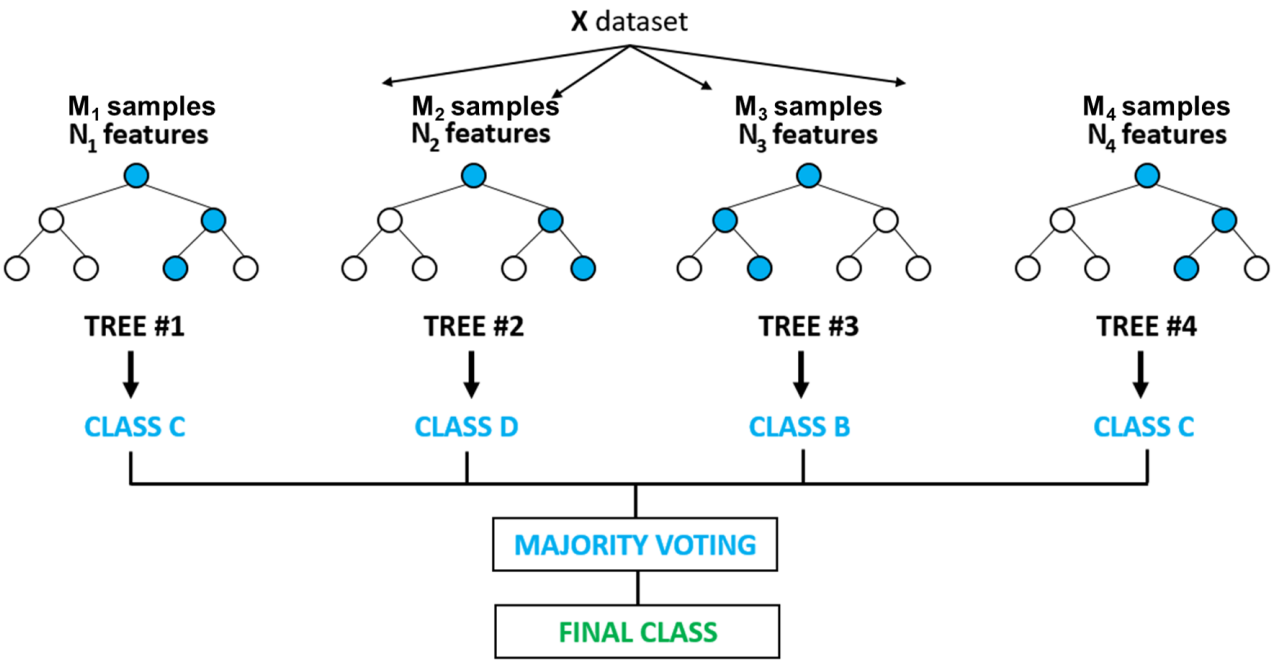
随机森林（Random Forest，以下简称RF）是bagging的一个特化进阶版。

- 所谓的特化是因为随机森林使用了CART决策树作为基学习器。
- 所谓的进阶是随机森林在bagging的样本随机采样基础上，又加上了特征字段的随机选择。这样进一步增强了模型的泛化能力。

Q1: 为什么叫做随机森林？

随机森林包含两个关键词，一个是“随机”，一个就是“森林”。随机森林的基学习器是决策树，一个决策树称为“树”，那多个决策树聚集在一起便是森林了，这体现了随机森林算法的集成思想。 随机森林的"随机"有两层含义，即样本抽样的随机性和特征抽样的随机性。

随机森林的原理如下图所示：



输入：

- 样本集 $D=\{(x_1,y_1),(x_2,y_2),...(x_m,y_m)\}$，
- 弱分类器迭代次数T。

算法流程：

1. 对于第 $t=1,2...,T$棵树而言:
    - 随机有放回地从训练集中的抽取 m 个训练样本（bootstrap sample），作为该树的训练集 D_t；
    - 从 D_t中随机选择一部分特征子集，训练第 t 个决策树模型；
    - **每棵树没有剪枝过程**，生长到指定的树的深度。

2. 如果是分类算法预测，则T个弱学习器投出最多票数的类别或者类别之一为最终类别。如果是回归算法，T个弱学习器得到的回归结果进行算术平均得到的值为最终的模型输出。

输出：

最终的强分类器 $f(x)$

Q2: 为什么要进行随机抽样？

如果不进行随机抽样，每棵树的训练集都一样，那么最终训练出的树分类结果也是完全一样的，依然没有解决决策树过拟合问题。随机抽样是为了保证不同决策树之间的多样性，从而提高模型的泛化能力。使得随机森林不容易陷入过拟合，并且具有较好的抗噪能力（比如：对缺省值不敏感）。

Q3: 为什么要有放回地抽样？

而随机森林最后分类取决于多棵树（弱分类器）的投票表决，这种表决应该是"求同"。如果是无放回的抽样，那么每棵树的训练样本都是不同的，都是没有交集的，这样每棵树都是"有偏的"，从而影响最终的投票结果。为了保证最终结果的可靠性，同时又要保证模型的泛化能力，需要每一颗树既要“求同“ 又要 ”存异”。

综上，随机森林的性能与两个因素有关：

- 森林中每棵树的分类能力：每棵树的分类能力越强，整个森林的错误率越低。
- 森林中任意两棵树的相关性：相关性越大，泛化能力越弱；

减小特征选择个数n，树的相关性和分类能力也会相应的降低；增大n，两者也会随之增大。所以选择最优的n非常重要。

2. 随机森林的特点

下面总结一下 RF 的特点：

- 在当前所有算法中，具有极好的准确率；
- 能够在大数据集上有效地运行；
- 能够处理具有高维特征的输入样本，而且不需要降维；
- 能够评估各个特征的重要性；
- 在生成过程中，能够获取到内部生成误差的一种无偏估计；
- 对于缺省值问题也能够获得很好得结果；
- ...

Q4: 上面提到随机森林可以评估特征重要度，其原理是怎样的呢？

- 对每一颗决策树，选择相应的袋外数据（OOB）计算袋外数据误差，记为 errOOB1；
- 随机对袋外数据OOB所有样本的特征加入噪声干扰(随机的改变样本在该特征的值)，再次计算袋外数据误差，记为 errOOB2；
- 特征的重要度= $\sum(err_{OOB2} - err_{OOB1})/N$，N 表示随机森林中决策树的个数。

当改变样本在该特征的值，若袋外数据准确率大幅度下降，则该特征对于样本的预测结果有很大影响，说明特征的重要度比较高。

3. 随机森林变种

RF有很多变种算法，不光可以用于分类回归，还可以用于特征转换，异常点检测等。

1. Extra Trees

extra trees（ET）是RF的一个变种, 与 RF 的区别有：

1. RF采用的是随机采样bootstrap来选择采样集作为每个决策树的训练集，而extra trees采用原始训练集。
2. 在选定了划分特征后，RF的决策树会基于信息增益，基尼系数，均方差之类的原则，选择一个最优的特征值划分点，这和传统的决策树相同。但是extra trees会随机的选择一个特征值来划分决策树。

ET采用原始训练集计算量一般会大于RF。ET随机选择了特征值的划分点位，而不是最优点位，模型的方差相对于RF进一步减少，但是偏倚相对于RF进一步增大。在某些时候，ET的泛化能力比RF更好。

2. Totally Random Trees Embedding

Totally Random Trees Embedding (TRTE) 是一种非监督学习的数据转化方法。和支持向量机相似，它将低维的数据集映射到高维，从而更好的运用分类回归模型。但是TRTE采用了另外一种映射方法。

TRTE类似于RF，建立T个决策树拟合数据。当决策树建立完毕以后，数据集里的每个数据在T个决策树中叶子节点的位置也定下来了。比如我们有3颗决策树，每个决策树有5个叶子节点，某个数据特征 x 划分到第一个决策树的第2个叶子节点，第二个决策树的第3个叶子节点，第三个决策树的第5个叶子节点。则 x 映射后的特征编码为(0,1,0,0,0, 0,0,1,0,0, 0,0,0,0,1) 有15维的高维特征。这里特征维度之间加上空格是为了强调三颗决策树各自的子编码。映射到高维特征后，就可以使用监督学习的各种分类回归算法了。

3. Isolation Forest

Isolation Forest（以下简称IForest）是一种异常点检测的方法。

对于在T个决策树的样本集，IForest也会对训练集进行随机采样，且采样个数要远远小于训练集个数，且最大决策树深度max_depth也比较小。为什么呢？因为我们的目的是异常点检测，只需要部分的样本我们一般就可以将异常点区别出来了。

对于每一个决策树的建立，IForest采用随机选择一个划分特征，对划分特征随机选择一个划分阈值。

对于异常点的判断，则是将测试样本点 x 拟合到T颗决策树。计算在每颗决策树上该样本的叶子节点的深度 $h_t(x)$ 。从而可以计算出平均高度 $h(x)$。此时我们用下面的公式计算样本点 x 的异常概率:

$$s(x,m) = 2^{-\frac{h(x)}{c(m)}}$$

其中，m为样本个数。$c(m)$的表达式为：

$$c(m) = 2\ln(m-1) + \xi - 2\frac{m-1}{m},\ \xi\text{为欧拉常数}$$

$s(x,m)$ 的取值范围是[0,1]，取值越接近于1，则是异常点的概率也越大。

4. 随机森林参数解读

首先看一下sklearn中的随机森林都包含哪些参数：

```
RandomForestClassifier(n_estimators=10, criterion='gini', max_depth=None, min_samples_split=2,
                       min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto',
                       max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None,
                         bootstrap=True, oob_score=False, n_jobs=1, random_state=None, verbose=0,
                         warm_start=False, class_weight=None)
```

下面对几个重要的参数进行分析：

1. n_estimators： RF中子树的数量 较多的子树可以让模型有更好的性能，但同时会降低算法的速度。

在保证计算速度的同时，应选择尽可能高的值。

2. max_depth：决策树最大深度

默认决策树在建立子树的时候不会限制子树的深度。

一般数据少或者特征少的时候取默认值。如果样本量多，特征也多的情况下，需要根据数据量和特征量进行设置。常用的取值范围为10-100之间。

3. min_samples_split：内部节点再划分所需最小样本数

这个值限制了子树继续划分的条件，如果某节点的样本数少于min_samples_split，则不会继续再尝试选择最优特征来进行划分。

默认是 2，如果样本量不大，取默认值即可。如果样本量非常大，则增大这个值。

4. min_samples_leaf：叶子节点最少样本数

如果某叶子节点数目小于样本数，则会和兄弟节点一起被剪枝。可以输入最少的样本数的整数，或者最少样本数占样本总数的百分比。

默认是1，如果样本量不大，取默认值即可。如果样本量非常大，则增大这个值。

5. max_features： RF允许单个决策树使用特征的最大数量。

取值为以下几种 (下式中N为样本总特征数) ：

- auto或sqrt：每颗子树可以利用总特征数的平方根个$\sqrt{N}$。 例如，如果变量（特征）的总数是100，所以每颗子树只能取其中的10个。
- “log2”：意味着划分时最多考虑 $log_2 N$个特征 。
- 整数：代表考虑的特征绝对数。 例如 5，就代表选取5个特征。
- 浮点数 ：代表考虑特征总数的百分比。例如0.8表示每个随机森林的子树可以利用特征数为80%*N。

增加max_features一般能提高模型的性能，因为在每个节点上，我们有更多的选择可以考虑。 然而，会降低了单个树的泛化能力，同时降低算法的速度。 因此，你需要适当的平衡和选择最佳max_features。

6. class_weight : 设置不同类别的样本权重

默认所以的样本权重均为1，当样本类别分布比较均衡时，可以取默认值。当类别非均衡问题比较严重时，需要对该值进行设置，可以设置为"balanced"，也可以设置字典{class_label: weight} 来自定义每个类别的样本权重。

7. min_weight_fraction_leaf：叶子节点最小的样本权重和

这个值限制了叶子节点所有样本权重和的最小值，如果小于这个值，则会和兄弟节点一起被剪枝。

默认不考虑权重。如果我们缺失值较多，或者样本类别非均衡问题比较严重，就需要考虑这个值。

8. oob_score：是否采用袋外数据验证模型的泛化能力

默认不采用袋外数据验证。个人建议设置为True。

关于袋外数据 集成模型概述(一) 已经介绍过。实际上，oob可以作为随机森林交叉验证的验证集对模型的性能进行评估而无需额外的单独设置验证集。

总结一下：

以上1-5个参数是调优经常用到的参数，遇到样本类别不均衡问题显著时，需要调整6-7。

参考文献：

Machine Learning & Algorithm] 随机森林（Random Forest）

Random Forests

scikit-learn随机森林调参小结