**随机森林算法学习**

最近在做kaggle的时候，发现随机森林这个算法在分类问题上效果十分的好，大多数情况下效果远要比svm，log回归，knn等算法效果好。因此想琢磨琢磨这个算法的原理。

要学随机森林，首先先简单介绍一下集成学习方法和决策树算法。下文仅对该两种方法做简单介绍（具体学习推荐看统计学习方法的第5章和第8章）。

**Bagging和Boosting的概念与区别**

该部分主要学习自：<http://www.cnblogs.com/liuwu265/p/4690486.html>

随机森林属于集成学习（Ensemble Learning）中的bagging算法。在集成学习中，主要分为bagging算法和boosting算法。我们先看看这两种方法的特点和区别。

**Bagging（套袋法）**

bagging的算法过程如下：

1. 从原始样本集中使用Bootstraping方法随机抽取n个训练样本，共进行k轮抽取，得到k个训练集。（k个训练集之间相互独立，元素可以有重复）
2. 对于k个训练集，我们训练k个模型（这k个模型可以根据具体问题而定，比如决策树，knn等）
3. 对于分类问题：由投票表决产生分类结果；对于回归问题：由k个模型预测结果的均值作为最后预测结果。（所有模型的重要性相同）

**Boosting（提升法）**

boosting的算法过程如下：

1. 对于训练集中的每个样本建立权值wi，表示对每个样本的关注度。当某个样本被误分类的概率很高时，需要加大对该样本的权值。
2. 进行迭代的过程中，每一步迭代都是一个弱分类器。我们需要用某种策略将其组合，作为最终模型。（例如AdaBoost给每个弱分类器一个权值，将其线性组合最为最终分类器。误差越小的弱分类器，权值越大）

**Bagging，Boosting的主要区别**

1. 样本选择上：Bagging采用的是Bootstrap随机有放回抽样；而Boosting每一轮的训练集是不变的，改变的只是每一个样本的权重。
2. 样本权重：Bagging使用的是均匀取样，每个样本权重相等；Boosting根据错误率调整样本权重，错误率越大的样本权重越大。
3. 预测函数：Bagging所有的预测函数的权重相等；Boosting中误差越小的预测函数其权重越大。
4. 并行计算：Bagging各个预测函数可以并行生成；Boosting各个预测函数必须按顺序迭代生成。

下面是将决策树与这些算法框架进行结合所得到的新的算法：

1）Bagging + 决策树 = 随机森林

2）AdaBoost + 决策树 = 提升树

3）Gradient Boosting + 决策树 = GBDT

**决策树**

常用的决策树算法有ID3，C4.5，CART三种。3种算法的模型构建思想都十分类似，只是采用了不同的指标。决策树模型的构建过程大致如下：

**ID3，C4.5决策树的生成**

输入：训练集D，特征集A，阈值eps 输出：决策树T

1. 若D中所有样本属于同一类Ck，则T为单节点树，将类Ck作为该结点的类标记，返回T
2. 若A为空集，即没有特征作为划分依据，则T为单节点树，并将D中实例数最大的类Ck作为该结点的类标记，返回T
3. 否则，计算A中各特征对D的信息增益(ID3)/信息增益比(C4.5)，选择信息增益最大的特征Ag
4. 若Ag的信息增益（比）小于阈值eps，则置T为单节点树，并将D中实例数最大的类Ck作为该结点的类标记，返回T
5. 否则，依照特征Ag将D划分为若干非空子集Di，将Di中实例数最大的类作为标记，构建子节点，由结点及其子节点构成树T，返回T
6. 对第i个子节点，以Di为训练集，以A-{Ag}为特征集，递归地调用1~5，得到子树Ti，返回Ti

**CART决策树的生成**

这里只简单介绍下CART与ID3和C4.5的区别。

1. CART树是二叉树，而ID3和C4.5可以是多叉树
2. CART在生成子树时，是选择一个特征一个取值作为切分点，生成两个子树
3. 选择特征和切分点的依据是基尼指数，选择基尼指数最小的特征及切分点生成子树

**决策树的剪枝**

决策树的剪枝主要是为了预防过拟合，过程就不详细介绍了。

主要思路是从叶节点向上回溯，尝试对某个节点进行剪枝，比较剪枝前后的决策树的损失函数值。最后我们通过动态规划（树形dp，acmer应该懂）就可以得到全局最优的剪枝方案。

**随机森林（Random Forests）**

随机森林是一种重要的基于Bagging的集成学习方法，可以用来做分类、回归等问题。

随机森林有许多优点：

* 具有极高的准确率
* 随机性的引入，使得随机森林不容易过拟合
* 随机性的引入，使得随机森林有很好的抗噪声能力
* 能处理很高维度的数据，并且不用做特征选择
* 既能处理离散型数据，也能处理连续型数据，数据集无需规范化
* 训练速度快，可以得到变量重要性排序
* 容易实现并行化

随机森林的缺点：

* 当随机森林中的决策树个数很多时，训练时需要的空间和时间会较大
* 随机森林模型还有许多不好解释的地方，有点算个黑盒模型

与上面介绍的Bagging过程相似，随机森林的构建过程大致如下：

1. 从原始训练集中使用Bootstraping方法随机有放回采样选出m个样本，共进行n\_tree次采样，生成n\_tree个训练集
2. 对于n\_tree个训练集，我们分别训练n\_tree个决策树模型
3. 对于单个决策树模型，假设训练样本特征的个数为n，那么每次分裂时根据信息增益/信息增益比/基尼指数选择最好的特征进行分裂
4. 每棵树都一直这样分裂下去，直到该节点的所有训练样例都属于同一类。在决策树的分裂过程中不需要剪枝
5. 将生成的多棵决策树组成随机森林。对于分类问题，按多棵树分类器投票决定最终分类结果；对于回归问题，由多棵树预测值的均值决定最终预测结果