## 集成学习

### bagging：“投票选举”——基于数据随机重抽样分类器构造的方法；

#### bagging代表——随机森林算法；

#### 随机森林算法：

####

### boosting：“再学习”——基于所有分类器的加权求和方法；

#### boosting代表——adaboosting；

### bagging和boosting的区别:

#### 1. Bagging与boosting是一种很类似的技术，所使用的多个分类器的类型都是一致的。

#### 2. Bagging是由不同的分类器（1. 数据随机化；2.特征随机化），经过训练，综合得出的出现最多分类结果；

#### 2. Boosting是通过调整已有分类器那些错分的那些数据来获取新的分类器，得出目前最优的结果；

#### 3. Bagging中分类de 器的权重是相等的；而boosting中的分类器加权求和，所以权重并不相等，每个权重代表的是其分类器在上一轮迭代中的成功度。

## 样例非均衡现象

在分类器训练时，正例数目和反例数目不相等（相差很大）。或者发生在正负例分类错误的成本不同的时候。

###

## 能否使用弱分类器和多个实例来构建一个强分类器？

Adaptive Boosting——自适应boosting

### AdaBoost开发流程：

1. 搜集数据——可以使用任意方法；
2. 准备数据——依赖所使用的弱分类器类型。针对机器学习实战第7章-基于adaboost的分类一节，使用的是单层决策树，该分类器可以处理任何数据类型；
   1. 传统决策树：
      1. 它的层数是一般是由数据的特征数目决定的，每一层的的构建难度主要取决于如何找到这样一个特征属性，使得使用该属性划分当前样本集具有最大信息增益；
      2. 按照上述规则递归构建决策子树，直至某一层子树对应的样本集全部为同一种标签或者样本集里只剩下一种特征属性；
   2. 单层决策树：顾名思义，该分类器只有一层分类节点，该分类节点主要包括划分属性，阈值，判断条件；找到该分类节点的策略就是遍历样本数据的属性，在他们的有限离散值里面找到具有最低分类误差率的阈值；
3. 分析数据：可以使用任意方法；
4. 训练算法：
   1. AdaBoost的大部分时间都用在训练上，分类器将多次在同一数据集上训练弱分类器， 只是每一轮数据的权重不一样:
      1. ；
      2. ；
   2. 每个弱分类器都对应一个单层决策树——：划分属性， 阈值， 判断条件（大于或小于阈值）；
   3. 对每一个弱分类器，通过计算它在样本集上的分类误差率，就可以得到它在最终分类器中的权重：
      1. ;
      2. ；
   4. 注意Adaboost虽然是分迭代训练，但是并不是会一致迭代下去；每一轮迭代训练后，Adaboost对已经训练出来的弱分类器进行加权叠加：
      1. ；
      2. 然后计算f(x)在训练集上的分类误差率，如果为0或者达到指定阈值，就终止迭代；
5. 测试算法：计算分类的错误率；
6. 使用算法：同SVM一样，Adaboost预测两个类别中的一个；如果想把它应用到多类别的场景，则可以固定一个类别，然后将其他类别归类为第二种类别，这样就可以得到多个二值类分类器；

### Adaboost优缺点：

1. 泛化（由具体的、个别的扩大为一般的）错误率低，易编码，可以应用在大部分分类器上，无需参数调节；
2. 缺点：对离群点敏感；
3. 使用数据类型：数值型和标称型数据；

## 无偏估计：

1.定义：就是某个公式对采样后的样本进行统计得到某个指标，这个指标会随着样本的不同而浮动；如果多次采样后，对该指标求期望，如果最终结果与样本变量无关，就说用该指标估计没有系统上的偏差，产生的误差是随机因素造成的；

2.特别地，对样本方差，如果按照：

2.1 计算样本方差，对该方差求期望，最终的结果里会引入样本变量，所以使用该公式对样本进行估计会引入样本偏差；

2.2 为消除样本偏差，需要把样本方差公式调整为：，对该方差公式计算期望，可得到；——此时称该方差公式为样本的无偏估计。

## 局部加权线性回归：