姓名： 李坚松
学号： 201618013229011

# Bagging 和 AdaBoost

本次作业主要利用 MATLAB 语言实现了 Bagging 和 AdaBoost 算法，数据集是 Stanford CS229 Machine Learning 课程作业的样例数据集和 UIC 的垃圾邮件数据集。下面简介一下四种典型的重采样方法的主要思想。

## 1. 重采样方法简介

Bootstrap、Bagging、Boosting、Adaboost 是机器学习中比较常见的几种重采样方法。其中，Bootstrap 重采样方法主要用于统计量的估计，Bagging、Boosting、Adaboost 则主要用于多个子分类器的组合。

### 1.1 Bootstrap

Bootstrap 主要用于统计量的估计，其主要思想是对原始数据集进行有放回地抽样，得到多个训练集。用这多个训练集对模型统计量进行估计，统计量的估计值定义为训练集上估计量的平均。

### 1.2 Bagging

Bagging 方法的主要思想是从原始数据集中随机选择样本点组成一个新的训练集，选择过程独立重复多次，得到多个训练集。对每个训练集进行训练，得到一个子分类器，最终分类器的分类结果由这些子分类器投票决定。

### 1.3 Boosting

Boosting 依次训练 k 个子分类器，最终分类结果由这些子分类器投票决定。首先从大小为 n 的原始数据集中随机选择 n1 个样本点作为训练集训练出第一个分类器，记为 C1。然后构造第二个分类器 C2 的训练集 D2，要求：D2 中一半样本能被 C1 正确分类，而另一半样本被 C1 错分。接着构造第三个分类器 C3 的训练集 D3，要求：C1、C2 对 D3 中样本点的分类结果不同。剩余的子分类器按照类似的思路进行训练。Boosting 构造新训练集的主要原则是使用最富信息的样本。

### 1.4 AdaBoost

AdaBoost 是 Boosting 的一种改进。它为每一个样本赋予一个权值，AdaBoost 希望在下一轮训练时被上一个子分类器的正确分类的样本权重和被错误分类的样本权重相等，从而下

一个子分类器和前一个子分类器有较大的差别。

## 1.5 Bagging 和 AdaBoost 的区别

从上面的介绍，可以总结出 AdaBoost 和 Bagging 算法的主要区别如下：

1. Bagging 方式主要是训练出多个弱分类器，虽然单个弱分类器的效果不理想，但是多个多分类器集成在一起，邮票可以产生更准确的分类结果。Bagging 要求分类器的学习算法不稳定，也就是当数据发生小变化时，训练的分类器会产生很大的不同，依次来增加分类器的多样性，使得分类系统更加稳定，泛化能力更强。Bagging 算法的训练集往往是从原始数据集中有放回地抽样得到的，每个分类器是相互独立的，并列的，而且最后的分类器要等权重地投票。
2. 而在 AdaBoost 算法中，分类器是依次训练的，分错的样本点在接下来的训练中会更加地被侧重，每个分类器的训练都是建立在之前分类器的表现的基础上的，最后各个分类器加权投票。相比之下，AdaBoost 算法训练的分类器比 Bagging 更加精致一些，更加有针对性一些，但这样也会导致过拟合的问题。

# 2. CART 决策树

本次作业中分类算法采用的是决策树分类算法，准确地说是 CART 分类算法。决策树算法的基本思想是利用带有分类信息的训练数据集训练出一棵决策树。利用该决策树对给定的实例进行分类时，从根节点开始，对实例的某一个特征进行测试，CART 算法对特征进行测试的标准是基尼指数(Gini Index)，根据测试结果，将实例分配到具体的子节点中；这时，每一个子节点对应着该特征的一个取值。如此递归地对实例进行测试并分配，直到达到叶节点为止。最后，将该实例分配到叶节点所在的类中。

CART 决策树生成算法如下：

---

输入：训练数据集 D，停止计算的条件

输出：CART 决策树

---

根据训练数据集，从根节点开始，递归地对每个节点进行以下操作，构建二叉决策树：

(1)设节点的训练数据集为 D，计算现有特征对该数据集的基尼指数。此时，对每个特征 A，对其可能取的每个值 a，根据样本点对 A=a 的测试为“是”或者“否”将 D 分割成 D1 和 D2 两部分，计算 A=a 时的基尼指数。

(2)在所有可能的特征 A 以及它们所有可能的切分点 a 中，选择基尼指数最小的的特征及其对应的切分点作为最优特征与最优切分点。依最优特征以及最优切分点，从现节点生成两个子节点，将训练数据集特征分配到两个子节点中。

(3)对两个子节点递归地调用(1)、(2)，直到满足停止条件。

(4)生成 CART 决策树。

---

注：算法停止计算的条件是节点中的样本个数小于预定的阈值，或样本集的基尼指数小于预定的阈值或者没有更多特征。

# 3. 实验过程

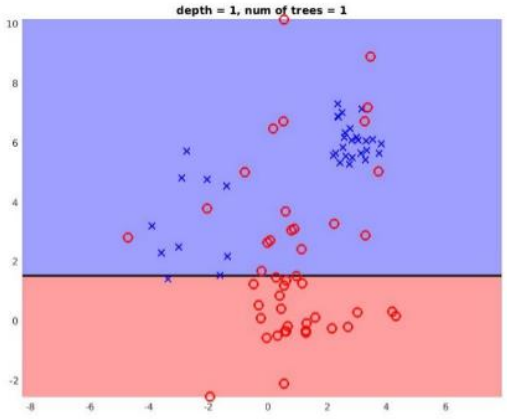
此次实验以 Bagging 和 AdaBoost 算法为例，对 Stanford CS229 机器学习课程提供的样例数据集是进行分类。分类算法采用 CART 决策树算法。class2d.ascii 该文件主要由 3 列构成(X,Y,C)，其中 X 和 Y 是特征，C 是分类信息，分为两类，类标号是-1 和 1。为了比较 Bagging 和 AdaBoost，此次实验中，对原始数据集进行采样训练，分别得到 1,10,100,1000 棵决策树分类器。为了避免决策树分类算法本身对实验效果的影响，实验中构建了不同深度的决策树，决策树深度的取值分别为 1,2,3。通过对 class2d.ascii 样例数据集进行实验可以对 Bagging 和 AdaBoost 算法的结果有一个直观的认识。

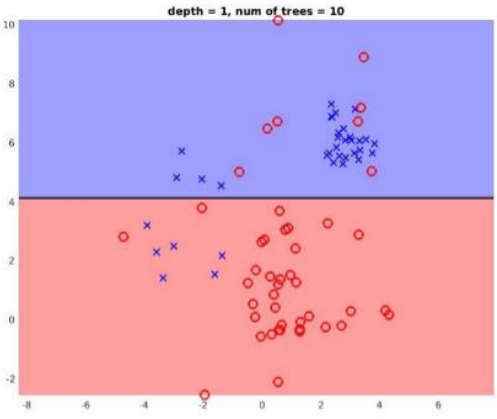
但是，为了进一步探讨和评估 Bagging 和 AdaBoost 算法，利用 UCI(加州大学欧文分校)提供的大规模数据集[3]进行实验。该数据集主要是关于垃圾邮件分类的，共有 58 个列，前 57 个列是数据特征，第 58 列是分类信息，取值为 1 或者-1,1 表示正常邮件，-1 表示垃圾邮件。该数据集的具体信息见参考文献[3]中的链接信息。实验过程中，数据分为两部分，一部分是训练数据集，主要用于分类器的训练；另一部分是测试数据集，主要用于测试 Bagging 算法和 AdaBoost 算法的准确率。训练过程，对训练数据集进行采样训练，分别得到 1,10,100,1000 棵决策树分类器，决策树深度的取值分别为 1,2,3。训练过程中也统计了算法的准确率，该准确率是所有决策树准确率的均值。为了测试二者的准确率，利用测试数据集测试 Bagging 算法和 AdaBoost 算法训练出来的模型的准确率。

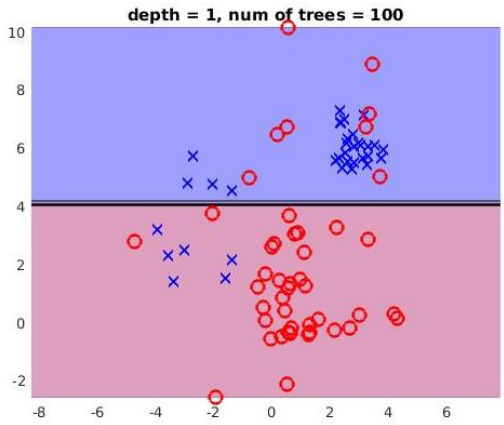
# 4. 结果分析

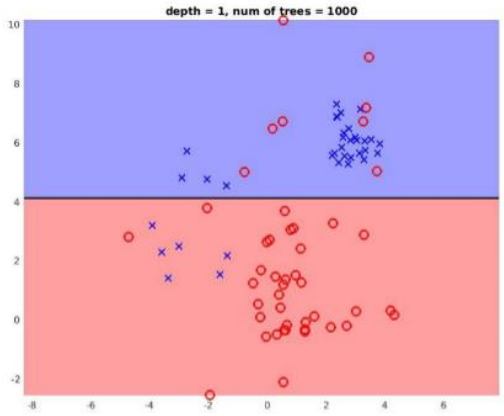
## 4.1 Bagging

对于 Bagging 算法，当 CART 决策树深度为 1，决策树分类器的数量分别为 1,10,100,1000 时的分类效果如下面四幅图所示：

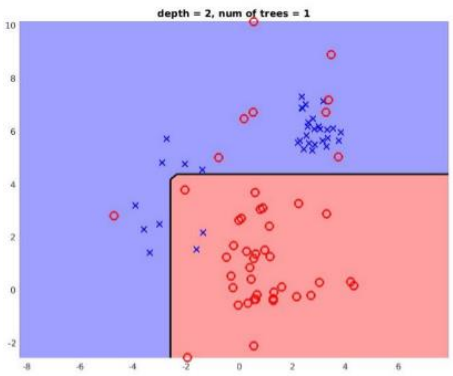


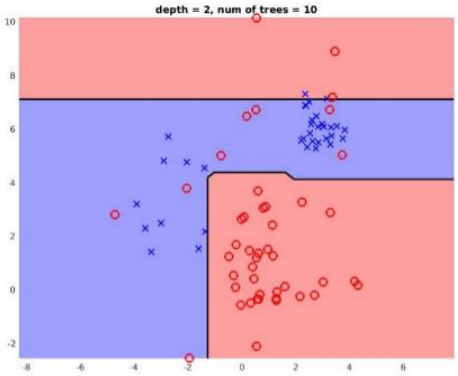



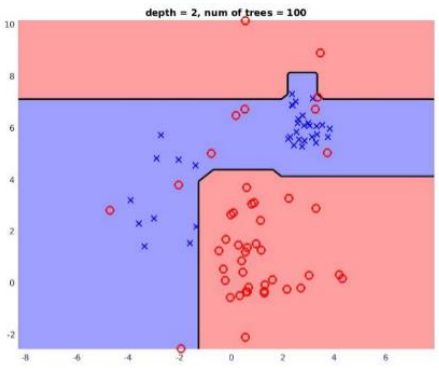


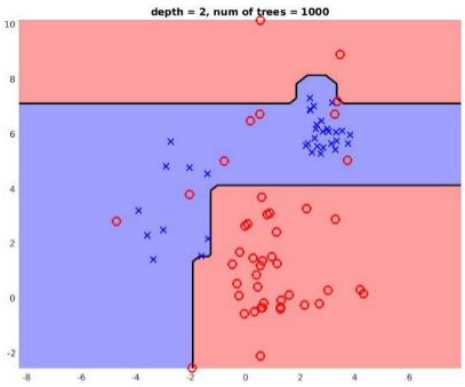




对于 Bagging 算法，当 CART 决策树深度为 2，决策树分类器的数量分别为 1,10,100,1000 时的分类效果如下面四幅图所示：

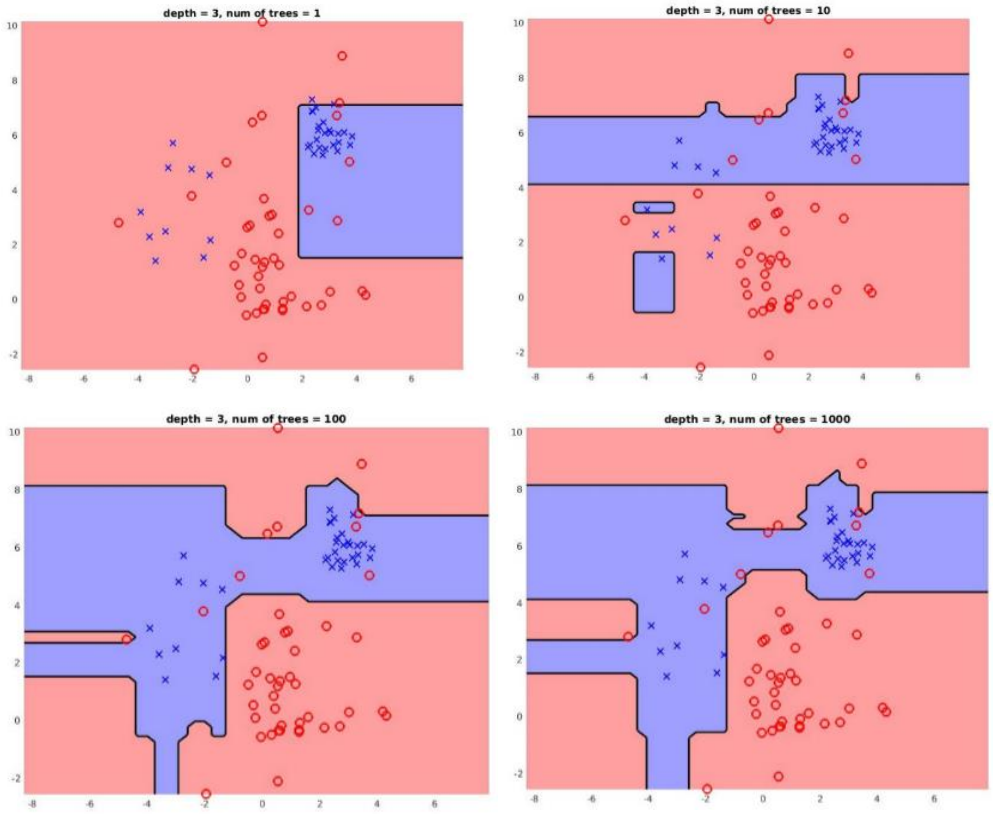


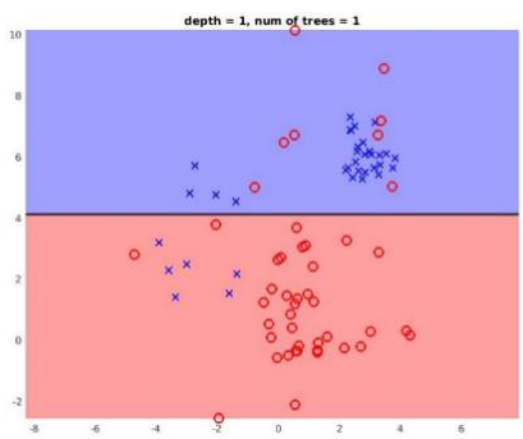



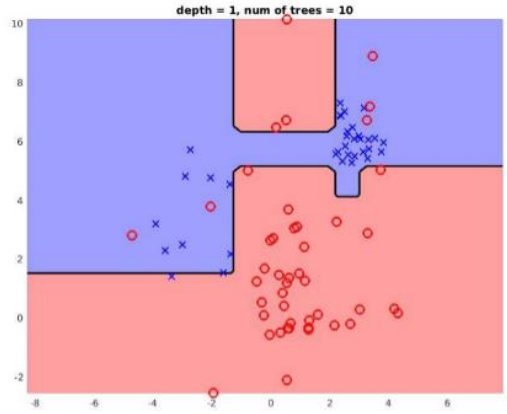

对于 Bagging 算法，当 CART 决策树深度为 3，决策树分类器的数量分别为 1,10,100,1000 时的分类效果如下面四幅图所示：

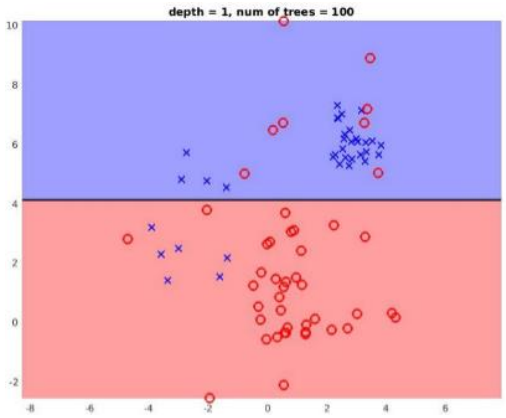

depth = 3, num of trees = 1
depth = 3, num of trees = 10
depth = 3, num of trees = 100
depth = 3, num of trees = 1000

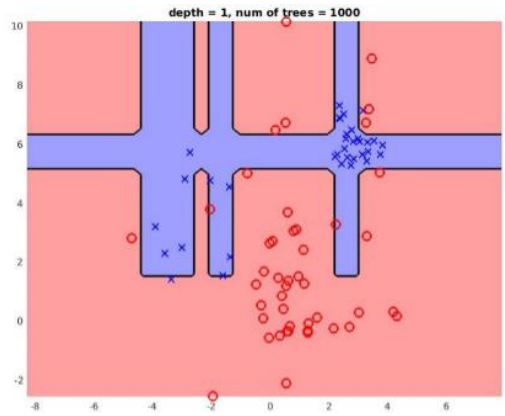
## 4.2 AdaBoost

对于 AdaBoost 算法，当 CART 决策树深度为 1，决策树分类器的数量分别为 1,10,100,1000 时的分类效果如下面四幅图所示：

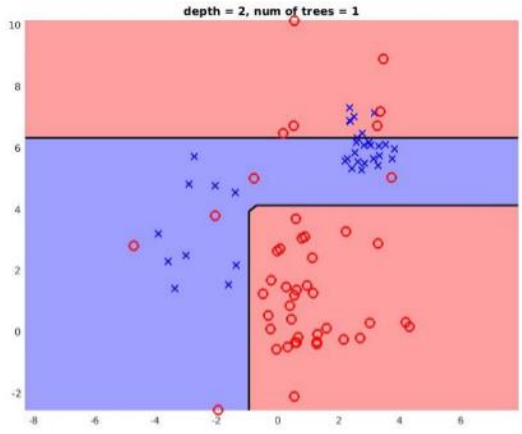


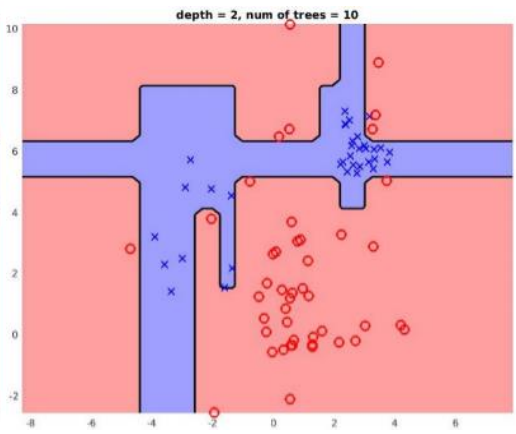



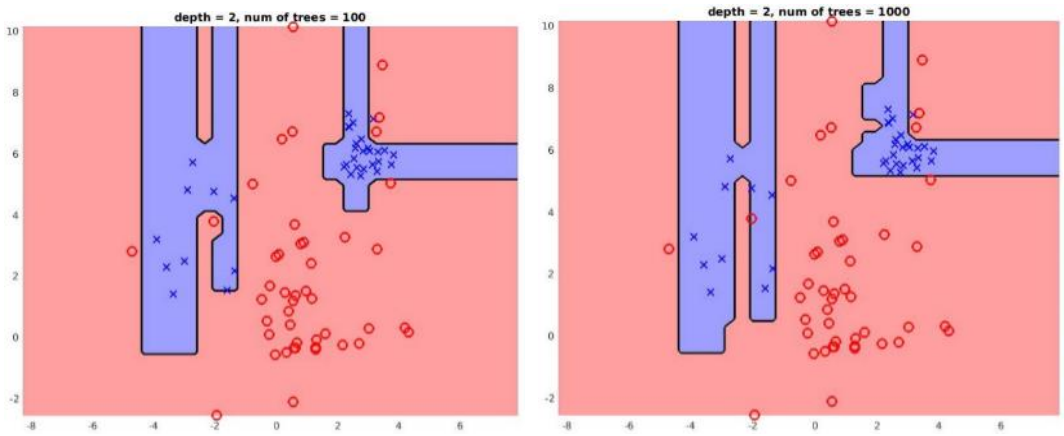



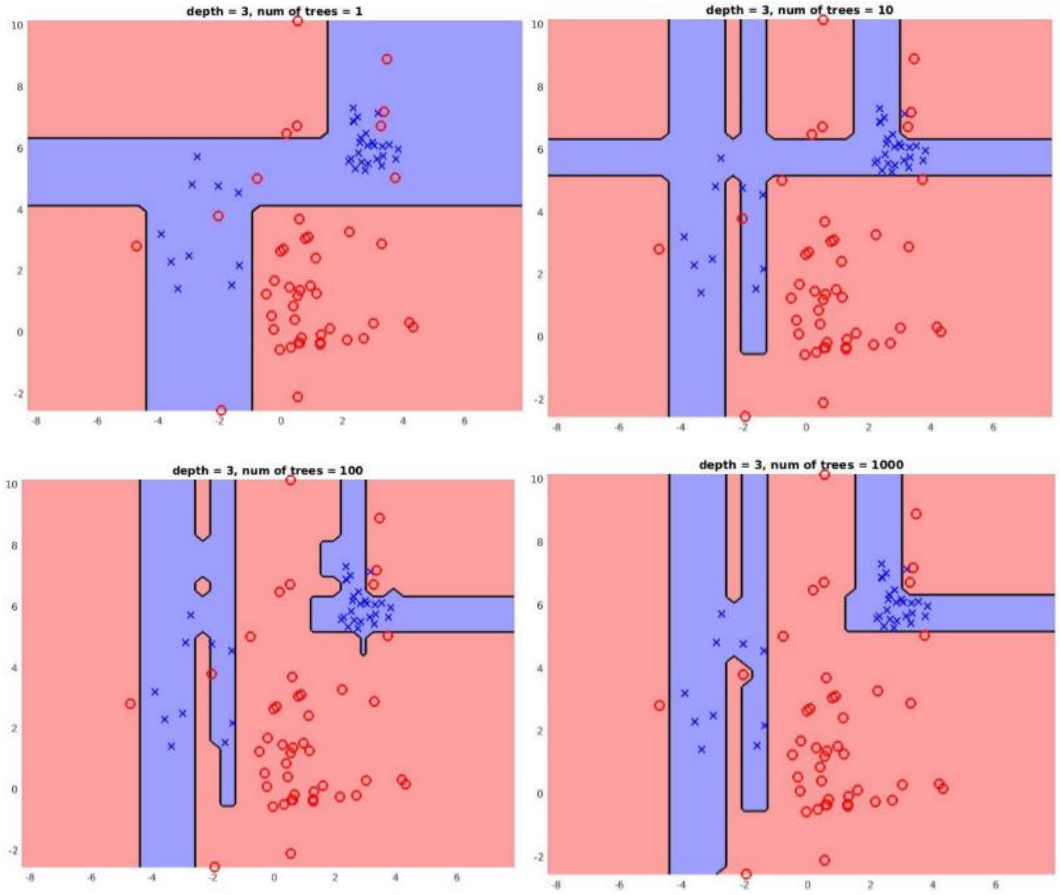

对于 AdaBoost 算法，当 CART 决策树深度为 2，决策树分类器的数量分别为 1,10,100,1000 时的分类效果如下面四幅图所示：



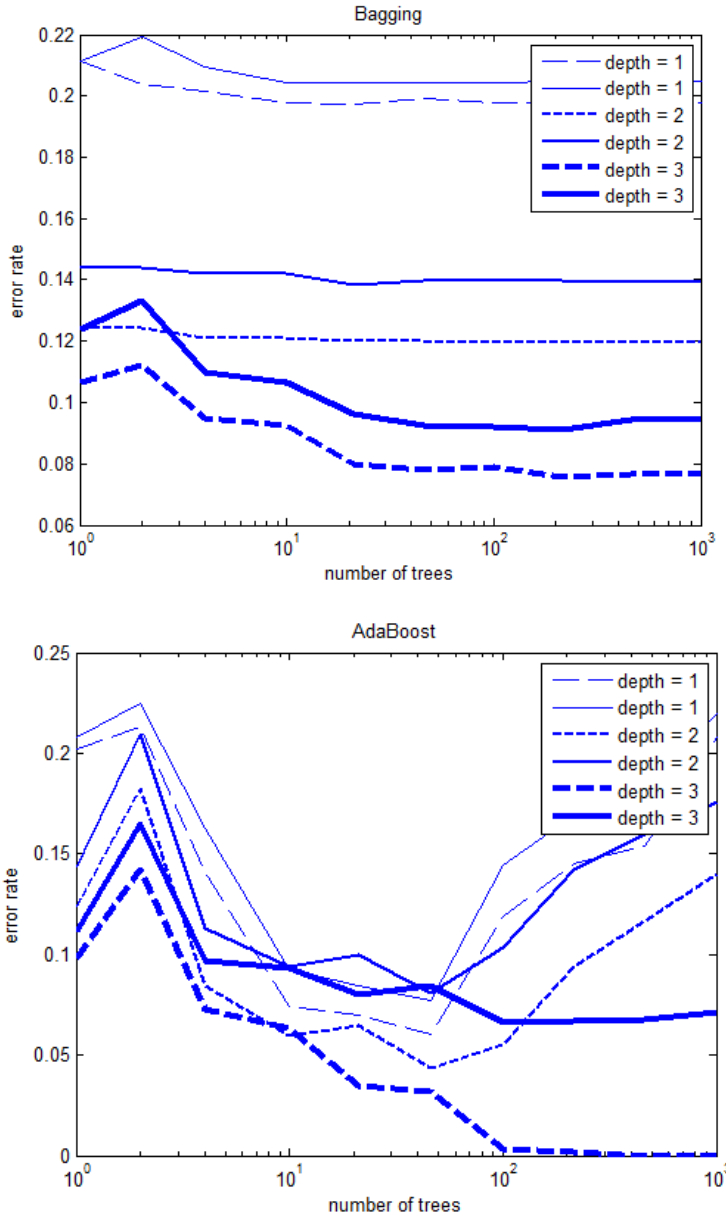






对于 AdaBoost 算法，当 CART 决策树深度为 3，决策树分类器的数量分别为 1,10,100,1000 时的分类效果如下面四幅图所示：



## 4.3 结果比对




注：上图中虚线表示训练集的错误率，实线表示测试集的错误率。

## 4.4 结论分析

从 4.3 的两幅结果图可以很清晰地得出以下结论：

**1. 横向来看**

- 随着决策树分类器数目的增加，Bagging 算法准确率在逐渐增加，但是当决策树的数目多于 100 的时候，Bagging 算法的准确率逐渐趋于稳定。这个很好理解，因为决策树数目的增加，意味着随机抽样的次数也在增加，不同的训练集训练出来的决策树采用投票算法，准确率提高是正常的。但是，当决策树数目大于 100 的时候，抽样达到饱和状态，即使决策树的数目再多，但是能预测的结果也趋于稳定，准确率不再增加。

- 随着决策树分类器数目的增加，AdaBoost 算法准确率的变化很不稳定。当决策树的数目增加到 50 的时候，AdaBoost 算法的准确率达到局部最高，但是当决策树再增加的时候，算法的准确率反而下降了，之所以出现这种现象，可以认为是过拟合的问题。因为 AdaBoost 的分类器是依次训练的，分错的点在接下来的训练会更加被侧重，也就是说每个分类器的训练都是建立在之前分类器的表现基础上的，这就很容易出现过拟合的现象。

**2. 纵向来看**

- 随着 CART 决策树深度的增加，Bagging 和 AdaBoost 算法的准确率都在逐渐升高。Bagging 算法比较稳定，对着决策树深度的增加，其准确率也在很平稳地增加，几乎没有什么抖动现象。而 AdaBoost 算法则不然，其抖动现象比较严重。这是因为，Bagging 算法在训练的时候，每个分类器是相互独立的，最后得到的分类器是等权重进行投票。而 AdaBoost 算法则是依次训练，也就说每个分类器都是建立在之前分类器表现的基础上的，之前分类器表现的好坏会直接影响到后续的分类的表现，而且最后的投票是加权投票，所以 AdaBoost 算法的抖动现象很严重。
- 比较 Bagging 算法和 AdaBoost 算法的准确率，可以很明显地看出 AdaBoost 算法的准确率要高于 Bagging。这是因为，Bagging 算法的训练集采用的是从原始数据集进行有放回的抽样得到的，训练出的是弱分类器，而 AdaBoost 算法的分类器是依次训练的，一个比一个强，每次选择的是最富信息的样本，即含有信息价值最高的样本，每个分类器都是建立在之前分类器的表现的基础上，也就是说被分错的样本点在接下来的训练中会被重点关注，最后得到的分类器也是加权的分类器，所以它的准确率比 Bagging 算法要高。

# 5. 参考文献


[1] 李航.统计学习方法[M].北京:清华大学出版社,2012.

[2] Stanford.CS229 Machine Learning[EB/OL]. http://cs229.stanford.edu,2017.

[3] UCI.Machine Learning Repository[EB/OL].

http://archieve.ics.uci.edu/ml/datasets/Spambase,2017.