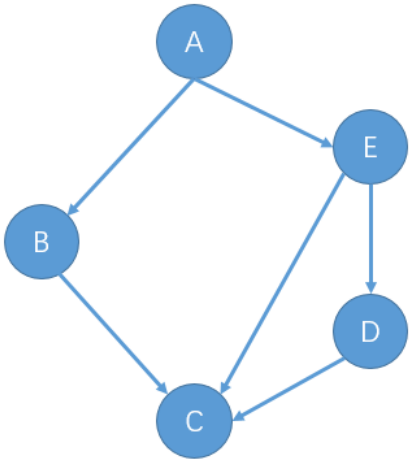
机器学习导论
作业五

151250189，翟道京，151250189@smail.nju.edu.cn

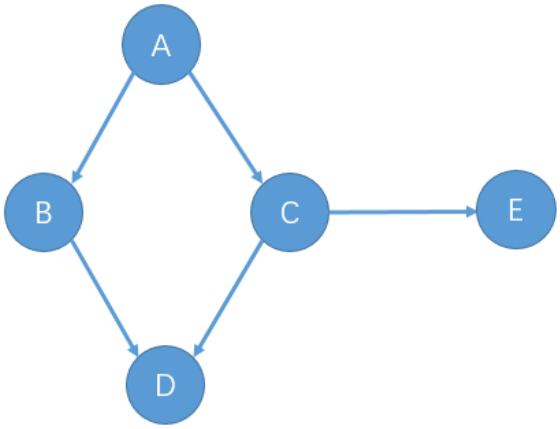
2018 年 6 月 19 日

1 [30pts] Conditional Independence in Bayesian Network

(1) **[5pts]** 请给出图中贝叶斯网结构的联合概率分布的分解表达式。



(2) **[5pts]** 请给出下图中按照道德化方法可以找到的所有条件独立的组合 (即哪些变量关于哪些变量或者变量集条件独立)，独立也算做条件独立的一种特例。



(3) **[10pts]** 在这里，首先我们将给出关于“阻塞”的概念，然后我们根据“阻塞”的概念给出条件独立的充要条件。（大家也可以参考这个网站）

> **定义 1** (阻塞). 设 X, Y, Z 分别是一个有向无环图 G 里互没有交集的结点集，Z 阻塞 X 中的一结点到 Y 中一结点的通路 P(关于“通路”，在这里只要连通就算一条通路，对路中每条边的方向无任何要求)，当且仅当满足以下条件之一：
>
> 1. P 中存在顺序结构 $x \to z \to y$ 或同父结构 $x \leftarrow z \to y$，结点 z 包含在集合 Z 中；
> 2. P 中存在 V 型结构 $x \to z \leftarrow y$，结点 z 及其孩子结点不包含在集合 Z 中。

> **定理 1** (条件独立). 设 X, Y, Z 分别是一个有向无环图 G 里互没有交集的结点集，如果集合 Z 阻塞 X 到 Y 的任何一条道路，则 X 和 Y 在给定 Z 时条件独立，即 $X \perp\!\!\!\perp Y | Z$。

请根据定理1，判断第一问中有哪些条件独立的组合（独立也算条件独立的一种特例），只考虑 X 和 Y 是单变量即可。

(4) **[10pts]** 由以上两问我们可知，道德化方法中的“除去集合 z 后，x 和 y 分属两个连通分支”并不构成条件独立性的充要条件。如果对道德化方法稍加修改，在连接 V 型结构父结点前，我们只保留图中 X, Y, Z 及他们的非孩子结点，之后的步骤则相同。请问你认为用修改后的方法可以保证得到全部的正确的条件独立集合吗？如果可以，请说明理由；如果不能，请给出反例。

Proof.

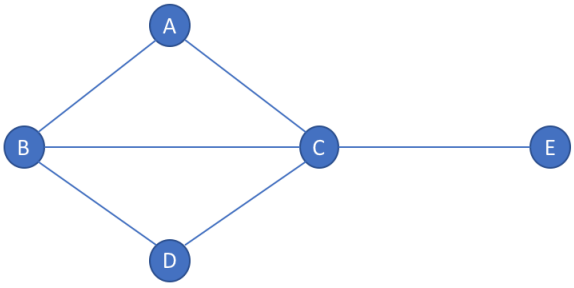
(1) 对贝叶斯网络

$$\Pr(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{d} \Pr(x_i | \pi_i)$$

因此对本网络有

$$\Pr(A, B, C, D, E) = \Pr(A) \Pr(B|A) \Pr(C|B, E, D) \Pr(D|E) \Pr(E|A)$$

(2) 该网络的道德图如下图所示



根据道德图，可以找到以下所有的条件独立的关系 *(这里我们仅选取了单变量进行分析条件独立关系)*

$A \perp D|\{B,C\}, A \perp D|\{B,C,E\}.$

$A \perp E|C, A \perp E|\{B,C\}, A \perp E|\{C,D\}, A \perp E|\{B,C,D\}.$

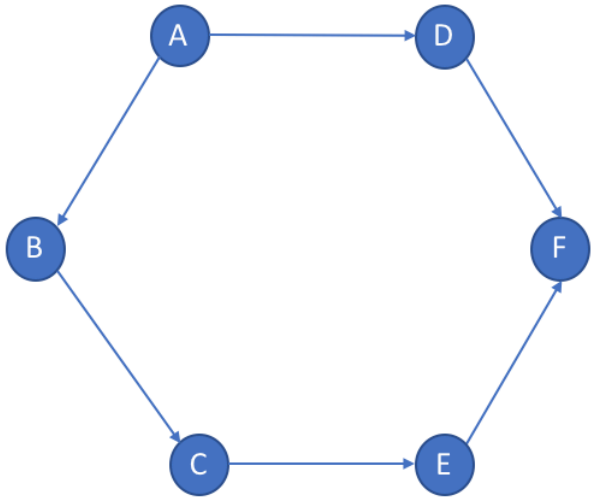
$B \perp E|C, B \perp E|\{A,C\}, B \perp E|\{C,D\}, B \perp E|\{A,C,D\}.$

$D \perp E|C, D \perp E|\{A,C\}, D \perp E|\{C,B\}, D \perp E|\{A,C,B\}$

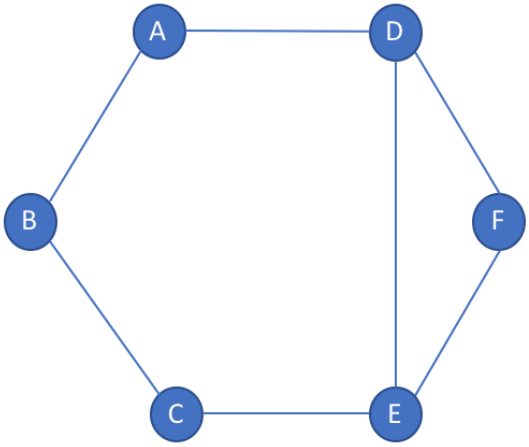
(3) 根据定理*1*，如果我们仅考虑 X, Y 单变量的情况，可以判断出的条件独立的组合为

- 根据顺序结构 $A \to B \to C, A \to E \to C$，得到 $A \perp C|\{B,E\}.$
- 根据顺序结构 $A \to E \to D$，得到 $A \perp D|\{E\}$
- 根据同父结构 $B \leftarrow A \to E$，可以得到 $B \perp D|\{A,E\}.$
- 根据 V 型结构 $B \to A \leftarrow E$，可以得到 $B \perp E|A$

(4) 考虑下列结构



如果根据定理一，可以得到 $A \perp C|B$；如果根据题目中的道德化方法，可以绘制道德图：



根据道德图，显然不能得到 $A \perp C|B$。因此，题目中的结论不正确，不能保证得到全部正确的条件独立集合。

2 [20pts] Naive Bayes Classifier

通过对课本的学习，我们了解了采用“属性条件独立性假设”的朴素贝叶斯分类器。现在我们有如下表所示的一个数据集：

表 1: 数据集

编号	x_1	x_2	x_3	x_4	y
样本 1	1	1	1	0	1
样本 2	1	1	0	0	0
样本 3	0	0	1	1	0
样本 4	1	0	1	1	1
样本 5	0	0	1	1	1

(1) **[10pts]** 试计算：$\Pr\{y=1|\mathbf{x}=(1,1,0,1)\}$ 与 $\Pr\{y=0|\mathbf{x}=(1,1,0,1)\}$ 的值；

(2) **[10pts]** 使用“拉普拉斯修正”之后，再重新计算上一问中的值。

Solution.

(1) 根据贝叶斯定理，得到

$$\Pr\{y=1|\mathbf{x}=(1,1,0,1)\}=\frac{\Pr\{\mathbf{x}=(1,1,0,1)|y=1\}\Pr\{y=1\}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}$$

$$\Pr\{y=0|\mathbf{x}=(1,1,0,1)\}=\frac{\Pr\{\mathbf{x}=(1,1,0,1)|y=0\}\Pr\{y=0\}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}$$

在朴素贝叶斯分类器中，根据“属性条件独立性假设”，得到

$$\begin{aligned}\Pr\{\mathbf{x}=(1,1,0,1)|y=1\}&=\Pr\{x_1=1|y=1\}\Pr\{x_2=1|y=1\}\Pr\{x_3=0|y=1\}\Pr\{x_4=1|y=1\}\\&=\frac{2}{3}\times\frac{1}{3}\times 0\times\frac{2}{3}=0\end{aligned}$$

因此有

$$\Pr\{y=1|\mathbf{x}=(1,1,0,1)\}=\frac{\Pr\{\mathbf{x}=(1,1,0,1)|y=1\}\Pr\{y=1\}}{\Pr\{\mathbf{x}=(1,1,0,1)\}}=0$$

同时

$$\Pr\{y=0|\mathbf{x}=(1,1,0,1)\}=1-\Pr\{y=1|\mathbf{x}=(1,1,0,1)\}=1$$

(2) 通过“拉普拉斯修正”后，

$$\hat{\Pr}\{c|\mathbf{x}\}=\frac{\hat{\Pr}\{c\}\hat{\Pr}\{\mathbf{x}|c\}}{\Pr\{\mathbf{x}\}}=\frac{\hat{\Pr}\{c\}}{\Pr\{\mathbf{x}\}}\prod_{i=1}^{d}\hat{\Pr}\{x_i|c\}$$

其中有

$$\hat{\Pr}\{c\}=\frac{|D_c|+1}{|D|+N}$$

$$\hat{\Pr}\{x_i|c\} = \frac{|D_{c,x_i}| + 1}{|D_c| + N_i}$$

因此

$$\hat{\Pr}\{y = 1\} = \frac{4}{7}$$

$$\hat{\Pr}\{y = 0\} = \frac{3}{7}$$

同时有

$$\hat{\Pr}\{\mathbf{x} = (1, 1, 0, 1)|y = 1\} = \frac{3}{5} \times \frac{2}{5} \times \frac{1}{5} \times \frac{3}{5} = \frac{18}{625}$$

$$\hat{\Pr}\{\mathbf{x} = (1, 1, 0, 1)|y = 0\} = \frac{2}{4} \times \frac{2}{4} \times \frac{2}{4} \times \frac{2}{4} = \frac{1}{16}$$

因此有

$$\hat{\Pr}\{y = 1|\mathbf{x} = (1, 1, 0, 1)\} = \frac{\frac{4}{7} \times \frac{18}{625}}{\Pr\{\mathbf{x} = (1, 1, 0, 1)\}}$$

$$\hat{\Pr}\{y = 1|\mathbf{x} = (1, 1, 0, 1)\} = \frac{\frac{3}{7} \times \frac{1}{16}}{\Pr\{\mathbf{x} = (1, 1, 0, 1)\}}$$

同时

$$\hat{\Pr}\{y = 1|\mathbf{x} = (1, 1, 0, 1)\} + \hat{\Pr}\{y = 0|\mathbf{x} = (1, 1, 0, 1)\} = 1$$

得到

$$\hat{\Pr}\{y = 1|\mathbf{x} = (1, 1, 0, 1)\} = 0.3806$$

$$\hat{\Pr}\{y = 1|\mathbf{x} = (1, 1, 0, 1)\} = 0.6194$$

3 [50pts] Ensemble Methods in Practice

由于出色的性能和良好的鲁棒性，集成学习方法 (Ensemble methods) 成为了极受欢迎的机器学习方法，在各大机器学习比赛中也经常出现集成学习的身影。在本次实验中我们将结合两种经典的集成学习思想：Boosting 和 Bagging，对集成学习方法进行实践。

本次实验选取 UCI 数据集 Adult，此数据集为一个二分类数据集，具体信息可参照链接，为了方便大家使用数据集，已经提前对数据集稍作处理，并划分为训练集和测试集，大家可通过此链接进行下载。

由于 Adult 是一个类别不平衡数据集，本次实验选用 AUC 作为评价分类器性能的评价指标，AUC 指标的计算可调用sklearn 算法包。

(1) **[5pts]** 本次实验要求使用 Python 3 或者 Matlab 编写，要求代码分布于两个文件中，BoostMain.py、RandomForestMain.py (Python) 或 BoostMain.m、RandomForestMain.m (Matlab)，调用这两个文件就能完成一次所实现分类器的训练和测试；

(2) **[35pts]** 本次实验要求编程实现如下功能：

- **[10pts]** 结合教材 8.2 节中图 8.3 所示的算法伪代码实现 AdaBoost 算法，基分类器选用决策树，基分类器可调用 sklearn 中决策树的实现；
- **[10pts]** 结合教材 8.3.2 节所述，实现随机森林算法，基分类器仍可调用 sklearn 中决策树的实现，当然也可以自行手动实现，在实验报告中请给出随机森林的算法伪代码；
- **[10pts]** 结合 AdaBoost 和随机森林的实现，调查基学习器数量对分类器训练效果的影响 (参数调查)，具体操作如下：分别对 AdaBoost 和随机森林，给定基分类器数目，在训练数据集上用 5 折交叉验证得到验证 AUC 评价。在实验报告中用折线图的形式报告实验结果，折线图横轴为基分类器数目，纵轴为 AUC 指标，图中有两条线分别对应 AdaBoost 和随机森林，基分类器数目选取范围请自行决定；
- **[5pts]** 根据参数调查结果，对 AdaBoost 和随机森林选取最好的基分类器数目，在训练数据集上进行训练，在实验报告中报告在测试集上的 AUC 指标；

(3) **[10pts]** 在实验报告中，除了报告上述要求报告的内容外还需要展现实验过程，实验报告需要有层次和条理性，能让读者仅通过实验报告便能了解实验的目的，过程和结果。

实验报告.

3.1 实验目的

1. 实现集成学习典型算法：AdaBoost 与 RandomForest 算法。
2. 通过实验结果评估，掌握 AdaBoost 与 RandomForest 算法的原理。
3. 体会参数设置在集成学习问题中的作用。

3.2 实验内容

3.2.1 AdaBoost 算法

RandomForest 伪代码

随机森林的算法伪代码为

算法 1 AdaBoost Algorithm

输入: 训练集 $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_m, y_m)\}$

基学习算法 $\mathcal{L}$ (Decision Tree)

训练轮数 T.

function AdaBoost($\mathbf{D}, \mathcal{L}, T$)
  $\mathbf{D_1}(\mathbf{x}) = 1/m.$
  for $t = 1, 2, 3, \cdots, T$ **do**
    $h_t = \mathcal{L}(\mathbf{D}, \mathbf{D}_t)$
    $\epsilon_t = P(h_t(\mathbf{x}) \neq f(\mathbf{x}))$
    if $\epsilon > 0.5$ **then** Break
    end if
    $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$;
    $\mathbf{D}_{t+1} = \frac{\mathbf{D}_t exp(-\alpha_t f(\mathbf{x}) h_t(\mathbf{x}))}{Z_t}$
  end for
  return $H(\mathbf{x}) = sign(\sum_{i=1}^{T} \alpha_t h_t(\mathbf{x}))$
end function

RandomForest 代码描述

在本次实验中，使用 Python 3 实现了 AdaBoost 算法，具体代码见 BoostMain.py 程序。其中，我们调用了 sklearn 软件包中的 DecisionTreeClassifier 和 roc_auc_score。在函数 adaboost_clf 中，我们实现了上述算法。

3.2.2 RandomForest 算法

RandomForest 算法伪代码

随机森林的算法伪代码为

算法 2 Random Forest Algorithm

输入: 训练集 $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_m, y_m)\}$

特征 $\mathbf{F}$

训练轮数 T.

function RandomForest($\mathbf{D}, \mathbf{F}, T$)
    $\mathbf{H} \leftarrow 0$
    for $i \in 1, 2, \cdots, T$ **do**
        $\mathbf{D}^{(i)} \leftarrow$ A bootstrap sample from $\mathbf{D}$
        $h_i =$ RandomizedTreeLearn($\mathbf{D}, \mathbf{D}^{(i)}$)
        $\mathbf{H} \leftarrow \mathbf{H} \cup h_i$
    end for
    return H
end function

function RandomizedTreeLearn($\mathbf{D}, \mathbf{F}$)
    for Each node **do**
        $f \leftarrow$ very small subset of $\mathbf{F}$
        Split on best feature in f
        return The learned tree
    end for
end function

RandomForest 代码描述

在本次实验中，使用 Python 3 实现了 RandomForest 算法，具体代码见 RandomForest-Main.py 程序。其中，我们调用了 sklearn 软件包中的 DecisionTreeClassifier 和 roc_auc_score。在类 RandomForest 中，我们的函数 random_fit 实现了上述伪代码的功能。

3.3 实验结果

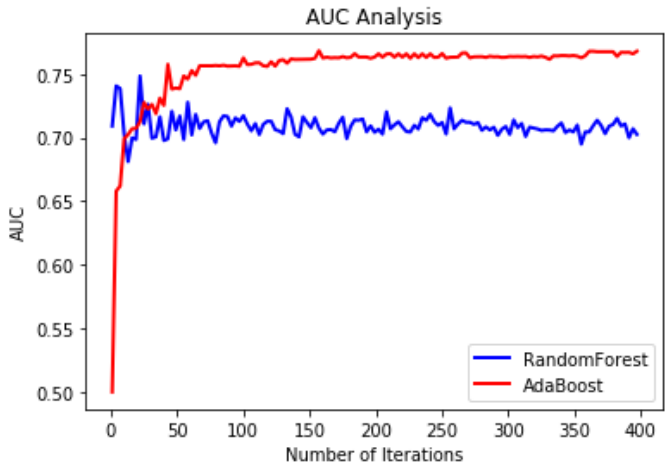
3.3.1 AdaBoost 与 RandomForest 实验结果对比

我们对基本参数下的 AdaBoost 和 RandomForest 实验结果进行对比，其中参数设置如下所示

表 2: My caption

Algorith	Adaboost	RandomForest
Parameter	max_depth=1 random_state=1	selected_feature=5 max_depth=1 random_state=1

绘制出两者的 AUC 对比图像，如下图所示



我们可以观测出，

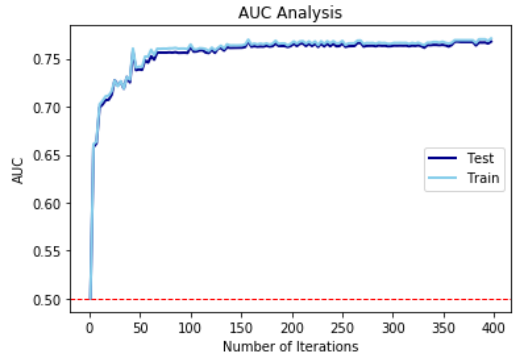
1. 在基分类器数目较少时，RandomForest 算法性能更好；随着基分类器数目增加，AdaBoost 算法渐入上风。

2. 针对过拟合问题，AdaBoost 算法更不容易出现过拟合现象。

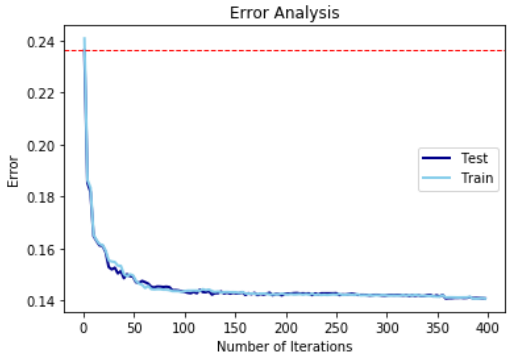
3.3.2 不同深度基分类器条件下 AdaBoost 算法实现情况

本次实验中，我们对 AdaBoost 算法进行了重点研究。

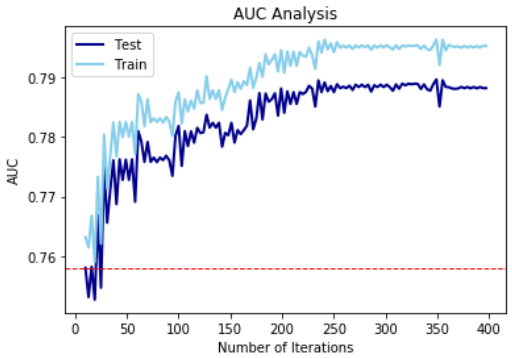
我们在控制树的最大深度为 1 和 3 的两种情况下，在基分类器数目范围为 0-400 的条件下在测试集中进行了训练，并对训练集进行了测试，绘制两种情况下误差和 AUC 随基分类器变化的图像如下图所示。



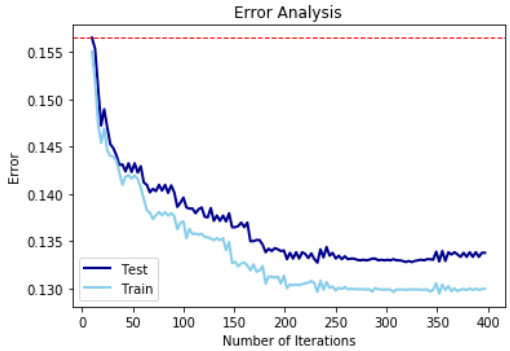
(a) AUC vs. Iteration, max_depth=1, random_state=1



(b) Error vs. Iteration, max_depth=1, random_state=1



(c) AUC.vs. Iteration, max_depth=3, random_state=1



(d) Error vs. Iteration, max_depth=3, random_state=1

图 1: AdaBoost implement result

我们可以从上述图像中，得到两点结论

1. 相比于之前的单一学习器，AdaBoost 学习器明显不容易产生过拟合现象。
2. 一定程度上提高集学习器的复杂程度 (在本例中体现为深度)，可以提高最终集成分类器的性能。但是从另一方面，训练误差的不稳定性加剧。

在思考与总结部分，我们对上述结论给出一定的解释。

3.3.3 RandomForest 算法实验结果

在 RandomForest 算法中，我们设定每次随机选取、待寻找最优特征的特征数为 5，在基分类器数目范围为 0-400 的条件下在测试集中进行了训练，并对训练集进行了测试，绘制 AUC 随基分类器变化的图像如下图所示。

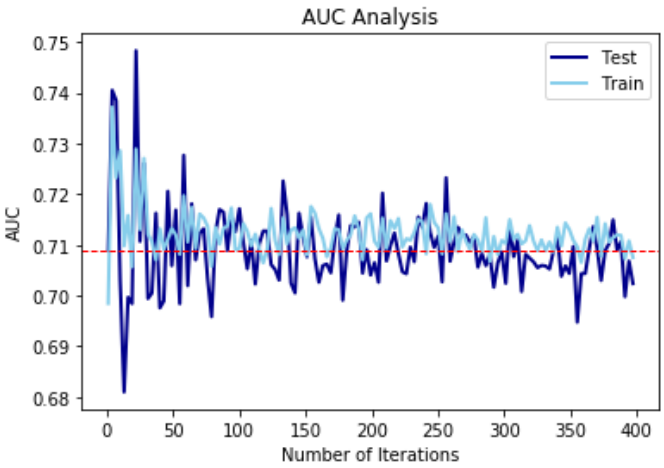


图 2: RandomForest: AUC vs. Iteration, Selected feature = 5

我们发现，随着基分类器数目增加，AUC 性能并没有大幅提升，但其随机性逐渐减少，数据趋于稳定。RandomForest 算法也有较好的防止过拟合效果。

3.3.4 测试集实验效果

我们分别把在训练集上最优的分类器应用于测试集中，得到的 AUC 指标如下表所示

表 3: My caption

Algorithm	Adaboost	Adaboost	RandomForest
Parameter	max_depth=1 random_state=1	max_depth=3 random_state=1	select_feature=1 max_depth=1 random_state=1
Tree Number	398	347	22
AUC	0.7682	0.7883	0.7482

相比较而言，我们的 AdaBoost 算法实验效果更好，但为实现如此效果，需要较长的训练时间。

3.4 思考与总结

3.4.1 AdaBoost 算法中的过拟合现象

本次实验中我们发现，集成学习算法过拟合现象不如单一学习器严重，通过结合周老师在最后一节课关于”Boosting Margin” 所做的报告，查询相关资料[1]，我得到了一些解释。

首先，对 bagging 来说，从偏差和方差的的角度，bagging 能够减少方差，从而防止过拟合。

[1]The Boosting Margin, or Why Boosting Doesn’t Overfit

但对 AdaBoost 来说，作为一种线性模型，而且采用了指数损失函数，通过不断增加基本分类器来达到损失函数最小化的效果。每增加一级损失函数，损失函数都会减小，即相应偏差越来越小，方差越来越大。从这个角度来说，AdaBoost 模型比构成他的基础分类器来说更容易过拟合。

但从另一方面，AdaBoost 模型对损失函数的逼近，不是无限的，到了一定程度受噪声的影响，损失函数减小很缓慢。当学习器在训练器上的误差接近 0 时，继续增加基础学习器，在训练集上仍然会减少误差，这是由于 $BoostingMargin$ 造成。这两方面对过拟合问题的影响强度大小是有基础分类器的复杂程度和数据分布的复杂程度决定的。

也就是说，AdaBoost 模型比起组成的基础分类器更容易过拟合。但当基础分类器的选择非常简单时，模型在实际应用中难以发生过拟合。在易于分割的数据集合中，AdaBoost 的迭代还有增加 Margin 的效果。但是在基础分类器复杂时，模型在复杂的数据分布上非常容易发生过拟合。