机器学习导论
习题六

171840708, 张逸凯, zykhelloha@gmail.com

2020 年 6 月 11 日

学术诚信

本课程非常重视学术诚信规范，助教老师和助教同学将不遗余力地维护作业中的学术诚信规范的建立。希望所有选课学生能够对此予以重视。[1]

(1) 允许同学之间的相互讨论，但是**署你名字的工作必须由你完成**，不允许直接照搬任何已有的材料，必须独立完成作业的书写过程;

(2) 在完成作业过程中，对他人工作（出版物、互联网资料）中文本的直接照搬（包括原文的直接复制粘贴及语句的简单修改等）都将视为剽窃，剽窃者成绩将被取消。**对于完成作业中有关键作用的公开资料，应予以明显引用**；

(3) 如果发现作业之间高度相似将被判定为互相抄袭行为，**抄袭和被抄袭双方的成绩都将被取消**。因此请主动防止自己的作业被他人抄袭。

作业提交注意事项

(1) 请在LaTeX模板中**第一页填写个人的姓名、学号、邮箱信息**；

(2) 本次作业需提交该pdf文件、 问题3可直接运行的源码(BoostMain.py, RandomForestMain.py， 不需要提交数据集)， 将以上三个文件压缩成zip文件后上传。zip文件格式为**学号.zip**，例如170000001.zip；pdf文件格式为**学号_姓名.pdf**，例如170000001_张三.pdf。

(3) 未按照要求提交作业，或提交作业格式不正确，将会**被扣除部分作业分数**；

(4) 本次作业提交截止时间为**6月11日23:59:59。本次作业不允许缓交，截止时间后不接收作业，本次作业记零分**。

[1] 参考尹一通老师高级算法课程中对学术诚信的说明。

1 [25pts] Bayesian Network

贝叶斯网(Bayesian Network)是一种经典的概率图模型，请学习书本7.5节内容回答下面的问题：

(1) **[5pts]** 请画出下面的联合概率分布的分解式对应的贝叶斯网结构：

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

(2) **[5pts]** 请写出图1中贝叶斯网结构的联合概率分布的分解表达式。

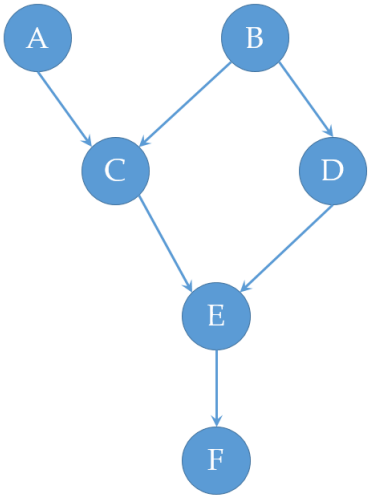


图 1: 题目1-(2)有向图

(3) **[15pts]** 基于第(2)问中的图1, 请判断表格1中的论断是否正确。首先需要作出对应的道德图，并将下面的表格填完整。

表 1: 判断表格中的论断是否正确

序号	关系	True/False	序号	关系	True/False
1	$A \perp\!\!\!\perp B$	True	7	$F \perp B \mid C$	False
2	$A \perp B \mid C$	False	8	$F \perp B \mid C, D$	True
3	$C \perp\!\!\!\perp D$	False	9	$F \perp B \mid E$	True
4	$C \perp D \mid E$	False	10	$A \perp\!\!\!\perp F$	False
5	$C \perp D \mid B, F$	False	11	$A \perp F \mid C$	False
6	$F \perp\!\!\!\perp B$	False	12	$A \perp F \mid D$	False

Solution. (1)

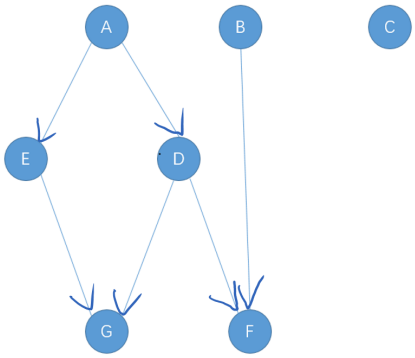


图 2: 贝叶斯网结构

对不起因为PPT画图的箭头太不清楚了, 所以用手画的箭头.

(2)

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

(3)

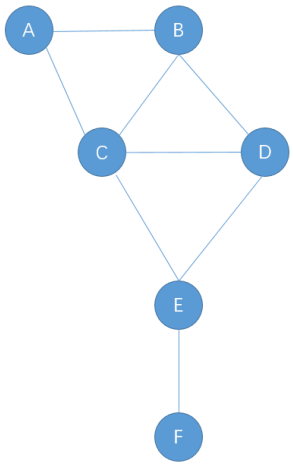


图 3: 道德图

表格请见上面. 谢谢.

2 [35+10pts] Theoretical Analysis of k-means Algorithm

给定样本集$\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, k-means聚类算法希望获得簇划分$\mathcal{C} = \{C_1, C_2, \cdots, C_k\}$

使得最小化欧式距离

$$J\left(\gamma,\mu_1,\ldots,\mu_k\right)=\sum_{i=1}^{n}\sum_{j=1}^{k}\gamma_{ij}\left\|\mathbf{x}_i-\mu_j\right\|^2 \tag{1}$$

其中$\mu_1,\ldots,\mu_k$为k个簇的中心(means), $\gamma\in\mathbb{R}^{n\times k}$为指示矩阵(indicator matrix)定义如下：若$\mathbf{x}_i$属于第j个簇, 则$\gamma_{ij}=1$, 否则为0，则最经典的k-means聚类算法流程如算法1中所示

Algorithm 1 $k-$means Algorithm

1: Initialize $\mu_1,\ldots,\mu_k$;

2: **repeat**

3: **Step 1:** Decide the class memberships of $\{\mathbf{x}_i\}_{i=1}^n$ by assigning each of them to its nearest cluster center.

$$\gamma_{ij}=\begin{cases}1, & \left\|\mathbf{x}_i-\mu_j\right\|^2\le\left\|\mathbf{x}_i-\mu_{j'}\right\|^2,\forall j'\\ 0, & \text{otherwise}\end{cases} \tag{2}$$

4: **Step 2:** For each $j\in\{1,\cdots,k\}$, recompute μ_j using the updated γ to be the center of mass of all points in C_j :

$$\mu_j=\frac{\sum_{i=1}^{n}\gamma_{ij}\mathbf{x}_i}{\sum_{i=1}^{n}\gamma_{ij}} \tag{3}$$

5: **until** the objective function J no longer changes;

(1) **[5pts]** 试证明, 在算法1中, Step 1和Step 2都会使目标函数J的值降低.

(2) **[5pts]** 试证明, 算法1会在有限步内停止。

(3) **[10pts]** 试证明, 目标函数J的最小值是关于k的非增函数, 其中k是聚类簇的数目。

(4) **[15pts]** 记$\hat{\mathbf{x}}$为n个样本的中心点, 定义如下变量,

$$\begin{aligned}T(X)&=\textstyle\sum_{i=1}^{n}\left\|\mathbf{x}_i-\hat{\mathbf{x}}\right\|^2/n\\ W_j(X)&=\textstyle\sum_{i=1}^{n}\gamma_{ij}\left\|\mathbf{x}_i-\mu_j\right\|^2/\sum_{i=1}^{n}\gamma_{ij}\\ B(X)&=\textstyle\sum_{j=1}^{k}\frac{\sum_{i=1}^{n}\gamma_{ij}}{n}\left\|\mu_j-\hat{\mathbf{x}}\right\|^2\end{aligned}$$

试探究以上三个变量之间有什么样的等式关系？基于此请证明, k-means聚类算法可以认为是在最小化$W_j(X)$的加权平均, 同时最大化$B(X)$.

(5) **[Bonus 10pts]**在公式1中, 我们使用ℓ_2-范数来度量距离(即欧式距离), 下面我们考虑使用ℓ_1-范数来度量距离

$$J'\left(\gamma,\mu_1,\ldots,\mu_k\right)=\sum_{i=1}^{n}\sum_{j=1}^{k}\gamma_{ij}\left\|\mathbf{x}_i-\mu_j\right\|_1 \tag{4}$$

- 请仿效算法1,给出新的算法(命名为$k-$means-ℓ_1算法)以优化公式4中的目标函数J'.
- 当样本集中存在少量异常点(outliers)时, 上述的$k-$means-ℓ_2和$k-$means-ℓ_1算法, 我们应该采用哪种算法？即哪个算法具有更好的鲁棒性？请说明理由。

Solution. (1)

- 证明 Step 1 使目标函数J的值降低:

  易见因为x被分配给了最近的中心, 显然比之前更远的中心欧氏距离更小.

  形式化证明: 不妨设i_0 (x_{i_0})原先属于第j_0簇, 即原先$\gamma_{i_0 j_0} = 1$, 更新后属于第j_0', 即更新后先$\gamma_{i_0 j_0'} = 1$. 我们对于x_{i_0}有:

  $$\sum_{j=1}^{k} \gamma_{i_0 j} \left\|\mathbf{x}_i - \mu_j\right\|^2 = \gamma_{i_0 j_0} \left\|\mathbf{x}_i - \mu_{j_0}\right\|^2 = \left\|\mathbf{x}_i - \mu_{j_0}\right\|^2 \geq \left\|\mathbf{x}_i - \mu_{j_0'}\right\|^2 = \gamma_{i_0 j_0'} \left\|\mathbf{x}_i - \mu_{j_0'}\right\|^2$$

  同理可以推广至$\forall\ i$.

- 证明 Step 2 使目标函数J的值降低:

  先证明一个引理:

  $$Lemma\ 1 : \mathbb{E}\|X - \xi\|^2 = \mathbb{E}\|X - \mu_X\|^2 + \|\xi - \mu_X\|^2, \text{ 其中}\xi\text{是原来的中心}$$

  证明如下:

  $$\begin{aligned}
  &\mathbb{E}\|X - \mu_X\|^2 + \|\xi - \mu_X\|^2 \\
  &= \mathbb{E}\left[\|X\|^2 + \|\mu_X\|^2 - 2X \cdot \mu_X\right] + \left[\|\xi\|^2 + \|\mu_X\|^2 - 2\xi \cdot \mu_X\right] \\
  &= \mathbb{E}\|X\|^2 + \|\mu_X\|^2 - 2\mu_X \cdot \mu_X + \|\xi\|^2 + \|\mu_X\|^2 - 2\xi \cdot \mu_X \\
  &= \mathbb{E}\|X - \xi\|^2
  \end{aligned}$$

  不妨设j_0簇更新后的中心为$\mu_{j_0'}$, 由上述引理, 我们有:

  $$\sum_{i=1}^{n} \gamma_{i j_0} \left\|\mathbf{x}_i - \mu_{j_0}\right\|^2 \geq \sum_{i=1}^{n} \gamma_{i j_0'} \left\|\mathbf{x}_i - \mu_{j_0'}\right\|^2$$

  同理可以推广至$\forall\ j$.

(2)

以上证明了J是减小的, 下面证明有限步内停止(收敛性): 我们知道最多只有 $\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$ (第二类Stirling数)种方法将n个数据划分成k类, 这是有限的. 注意到每次迭代, 有且仅有以下两种影响:

- 上述J值在两次迭代之间不变, 则接下来也不会变, 收敛.
- 由上(1)的证明我们知道, 如果两次迭代后J值不同, 则迭代后J减小. 又划分数有限, 所以收敛.

综上所述, 收敛性得证, 算法在有限步内停止.

(3)

问题等价于证明, k增大时, 目标函数J的最小值非增.

对聚类簇数目k使用数学归纳法证明:

- 初始步: $k=1$时:

$$J=\sum_{i=1}^{n}\left\|\mathbf{x}_i-\mu_1\right\|^2/n$$

$k=2$时, 在J达到最小值时, 不妨将$\mathbf{x}_i$分为两个簇: $\mathcal{D}_1,\mathcal{D}_2$, 新的中心记为$\mu'$:

$$J'=\left(\sum_{x_i\in\mathcal{D}_1}^{n}\left\|\mathbf{x}_i-\mu_1'\right\|^2+\sum_{x_i\in\mathcal{D}_2}^{n}\left\|\mathbf{x}_i-\mu_2'\right\|^2\right)/n\leq\left(\sum_{x_i\in\mathcal{D}_1}^{n}\left\|\mathbf{x}_i-\mu_1\right\|^2+\sum_{x_i\in\mathcal{D}_2}^{n}\left\|\mathbf{x}_i-\mu_1\right\|^2\right)/n=J$$

所以归纳法初始步得证, 即$k=1\to 2$时J非增.

- 证明$k\to k+1,\ \forall k\geq 2$ 时成立:

$\Rightarrow$ k个簇时$\mu_i,\ \gamma_{ij}$已收敛, 则当$k+1$时, (1)题中的Step 1和Step 2会变为可执行的.

$\left\|\mathbf{x}_i-\mu_{k+1}\right\|^2>\left\|\mathbf{x}_i-\mu_{j'}\right\|^2,\forall j'$是不可能发生的, 因为$\mu_{k+1}$总是某些$\mathbf{x}_i$的中心, 如果像上面这样则Step 2也无法更新, 任何$\mathbf{x}_i$不会属于μ_{k+1}.

所以$k=k+1$时, 新增了一个中心μ_{k+1}, 那么必存在一个点会更新到它的簇内, 以此类推, 加下来Step 2也会被执行. 得证.

(4)

类似地, 我们可以定义 $B_j(X)$:

$$B_j(X)=\frac{\sum_{i=1}^{n}\gamma_{ij}}{n}\left\|\mu_j-\hat{\mathbf{x}}\right\|^2$$

不妨取定第j_0簇(接下来再推广到所有簇), 令$\sum_{i=1}^{n}\gamma_{ij_0}=k_{j_0}$, 属于第$j_0$簇的元素为$\mathbf{x}^{(j_0)}$, 我们有:

$$B_j(X)+\frac{k_{j_0}}{n}W_j(X)=\frac{k_{j_0}}{n}\left\|\mu_{j_0}-\hat{\mathbf{x}}\right\|^2+\left(\sum_{i=1}^{k_{j_0}}\left\|\mathbf{x}_i^{(j_0)}-\mu_{j_0}\right\|^2/k_{j_0}\right)\cdot\frac{k_{j_0}}{n}$$

如上是一个比较好的分析方向, 结合所有簇我们有:

$$\sum_{j=1}^{k}\left(\left(\sum_{i=1}^{n}\gamma_{ij}\left\|\mathbf{x}_i-\mu_j\right\|^2/\sum_{i=1}^{n}\gamma_{ij}\right)\cdot\frac{\sum_{i=1}^{n}\gamma_{ij}}{n}\right)+\sum_{j=1}^{k}\frac{\sum_{i=1}^{n}\gamma_{ij}}{n}\left\|\mu_j-\hat{\mathbf{x}}\right\|^2$$

$$=\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\gamma_{ij}\left(x_i^2+2\mu_j^2-2x_i\mu_j-2\mu_j\hat{\mathbf{x}}+\hat{\mathbf{x}}^2\right)$$

又 $T(X)=\sum_{i=1}^{n}\left\|x_i-\hat{\mathbf{x}}\right\|^2/n=\sum_{i=1}^{n}\sum_{j=1}^{k}\gamma_{ij}\left\|x_i-\hat{\mathbf{x}}\right\|^2/n$

上式 $=T(X)+\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\gamma_{ij}\left(2x_i\hat{\mathbf{x}}+2\mu_j^2-2x_i\mu_j-2\mu_j\hat{\mathbf{x}}\right)$

所以: $T(X)+\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\gamma_{ij}\left(2x_i\hat{\mathbf{x}}+2\mu_j^2-2x_i\mu_j-2\mu_j\hat{\mathbf{x}}\right)=\sum_{j=1}^{k}\left(W_j(X)\cdot\frac{\sum_{i=1}^{n}\gamma_{ij}}{n}\right)+B(X)$

(这里感觉题目是有一丢丢问题的)

上式是成立的, 在最小化$W_j(X)$的加权平均时, 我们注意到等式另一边会在一个范围内变化, 所以此时最大化 $B(X)$.

(5)

$$J'(\gamma, \mu_1, \ldots, \mu_k) = \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij} \|\mathbf{x}_i - \mu_j\|_1$$

Algorithm 2 $k-$means-ℓ_1

1: Initialize $\mu_1, \ldots, \mu_k$;

2: **repeat**

3: **Step 1:** Decide the class memberships of $\{\mathbf{x}_i\}_{i=1}^n$ by assigning each of them to its nearest cluster center. 使用ℓ_1范数:

$$\gamma_{ij} = \begin{cases} 1, & \|\mathbf{x}_i - \mu_j\|_1 \leq \|\mathbf{x}_i - \mu_{j'}\|_1, \forall j' \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

4: **Step 2:** For each $j \in \{1, \cdots, k\}$, recompute μ_j using the updated γ to be the center of mass of all points in C_j, 选取中位数, $\mathbf{x}_i$是多维的分别处理即可:

$$\mu_j = middle\ number\ of\ \{\gamma_{ij}\ \mathbf{x}_i\}\ \forall i \tag{6}$$

5: **until** the objective function J no longer changes;

当存在少量异常点时, 我们选择$k-$means-ℓ_1.

因为离群点出现后中位数可能变化仍然不是很大. 这对中心的影响就小, 对聚类结果的影响就小. 所以$k-$means-ℓ_1更加鲁棒.

3 [40pts] Coding: Ensemble Methods

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，数据集文件夹为adult_dataset。

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，可调用sklearn算法包对AUC指标进行计算。

(1) 本次实验要求使用Python3编写，要求代码分布于两个文件中，BoostMain.py, RandomForestMain.py ，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，也可以手动实现，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，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) **[5pts]** 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.

- **实验目的:**

  如题, 实现Adaboost和随机森林, 研究AUC和基分类器数目的关系, 在测试集上输出AUC.

- **设计思路:**

  扎实学习课本知识, 就容易转化成代码.

  – Adaboost:

  最关键的是课本算法过程, 特别是图8.3:

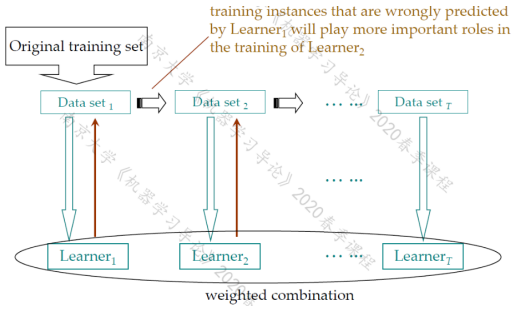
输入: 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
基学习算法 $\mathfrak{L}$;
训练轮数 T.

过程:

1: $\mathcal{D}_1(\boldsymbol{x}) = 1/m$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: $h_t = \mathfrak{L}(D, \mathcal{D}_t)$;
4: $\epsilon_t = P_{\boldsymbol{x} \sim \mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$;
5: **if** $\epsilon_t > 0.5$ **then break**
6: $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$;
7: $\mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases}$
$= \frac{\mathcal{D}_t(\boldsymbol{x}) \exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}$
8: **end for**

输出: $H(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \right)$

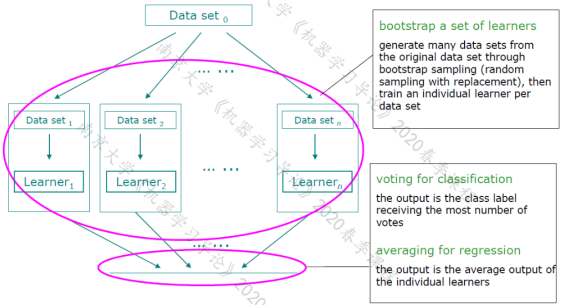
Boost类集成学习:



先训练出一个基学习器, **做错的样本在之后会受到更多的关注**, 基于调整后的样本分布来训练下一个基学习器, 一个基学习器的权重就是它错误率的函数(函数形式由loss求偏导可证), 最后结果是这些基学习器的线性组合.

– 随机森林:

Bagging类集成学习:



其他不难, 注意基决策树的每个结点, 都是先从该结点的属性集合中随机选择一个包含k个属性的子集, 然后从这个子集里选择一个最优属性进行划分. 推荐$k = log_2 d$.

– 从偏置-方差分解的角度:

Boosting 关注的是降低偏置.

Bagging (随机森林)关注的是降低方差.

- **代码实现:**

都打好注释了, 这里不赘述, 以Adaboost为例, 课本图8.3非常重要:

```
25      # 样本权值分布初始化, 均匀分布:
26      # 书上图8.3第一行:
27      self.sample_weights[0] = np.ones(shape=n) / n
28
29      for t in tqdm.trange(iters):
30          # 学习基分类器:
31          # 图8.3第三行:
32          cur_weights = self.sample_weights[t]
33          stump = DecisionTreeClassifier(max_depth=3)
34          # 对于具有权值分布的样本进行学习.
35          stump.fit(X, y, sample_weight=cur_weights)
36
37          # 计算误差:
38          # 图8.3第四行:
39          stump_pred = stump.predict(X)
40          err = cur_weights[(stump_pred != y)].sum()
41
42          # 计算基分类器权值:
43          # 图8.3第六行, 这里的证明看8.11以上.
44          stump_weight = np.log((1 - err) / err) / 2
```

使用五折交叉验证:

```
75      kf = KFold(n_splits=5)
76      kf.get_n_splits(X_train)
77
78      for train_index, test_index in kf.split(X_train):
79          X_cur_train, X_cur_test = X_train[train_index], X_train[test_index]
80          y_cur_train, y_cur_test = y_train[train_index], y_train[test_index]
```

- **实验结果**

  AUC-基分类器数目结果图如下:

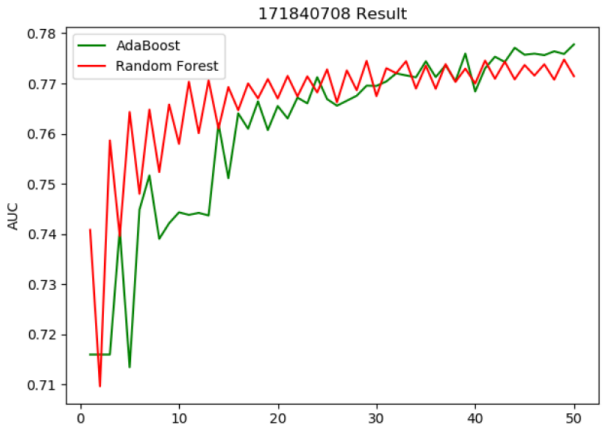


图 4: AUC-基分类器数目图

Adaboost在测试集上的**AUC指标**为: **0.7762**

随机森林在测试集上的**AUC指标**为: **0.7739**