山东大学

SHANDONG UNIVERSITY

山东大学机器学习课程

实验报告

——实验八：以 adaboost 为例的集成学习的设计与实现

姓名：刘梦源

学院：计算机科学与技术学院

班级：计算机 14.4

学号：201400301007

一、实验目的：

（1）学习集成学习的思想
（2）学习 adaboost 的算法原理
（3）根据已给数据集，编写代码完成 adaboost 分类器
（4）体会 adaboost 集成学习的优势

二、实验环境：

（1）硬件环境：
英特尔® 酷睿™ i7-7500U 处理器
512 GB PCIe® NVMe™ M.2 SSD
8 GB LPDDR3-1866 SDRAM
（2）软件环境：
Windows10 家庭版 64 位操作系统
Matlab R2016a

三、实验内容

（1）Adaboost 的原理

Adaboost 算法是经过调整的 Boosting 算法，其能够对弱学习得到的弱分类器的错误进行适应性(Adaptive)调整。上述算法中迭代了 T 次的主循环，每一次循环根据当前的权重分布对样本 x 定一个分布 P，然后对这个分布下的样本使用弱学习算法得到一个弱分类器，对于这个算法定义的弱学习算法，对所有的样本都有错误率，而这个错误率的上限并不需要事先知道，实际上。每一次迭代，都要对权重进行更新。更新的规则是：减小弱分类器分类效果较好的数据的概率，增大弱分类器分类效果较差的数据的概率。最终的分类器是个弱分类器的加权平均。

（2）Adaboost 的算法

一.样本

Given: m examples $(x_1, y_1), \ldots, (x_m, y_m)$

where $x_i \in X, y_i \in Y = \{-1, +1\}$

x_i表示X中第i个元素，

y_i表示与x_i对应元素的属性值，+1表示x_i属于某个分类，

-1表示x_i不属于某个分类

二.初始化训练样本x_i的权重D(i) :i=1,.....，m;

(1).若正负样本数目一致，$D_1(i) = 1/m$

(2).若正负样本数目m_+，m_-则正样本$D_1(i) = 1/m_+$，

负样本$D_1(i) = 1/m_-$

三. 训练弱分类器

For t=1, ……, T

1. Train learner h_t with min error $\varepsilon_t = \Pr_{i\sim D_t}[h_t(x_i) \neq y_i]$

若划分正确，则不计入误差，若所有元素都被正确划分，则误差为0

若划分错误，则计入误差

2. If $\varepsilon_t \geq 0.5$, then stop

3. Compute the hypothesis weight $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$

The weight **Ada**pts. The bigger ε_t becomes the smaller α_t becomes.

4. $D_{t+1}(i) = D_t(i)\exp\left(\alpha_t * 1_{(h_t(i)=y_i)}\right) = \begin{cases} D_t(i), & 若 y_i = h_t(x_i) \\ D_t(i)\frac{1-\varepsilon_t}{\varepsilon_t}, & 若 y_i \neq h_t(x_i) \end{cases}$;

5.最后得到的强分类器: $H(x) = \mathrm{sign}\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right)$

笼统来看，有以下两个方面需要考虑：

1. 使用加权后选取的训练数据代替随机选取的训练样本，这样将训练的焦点集中在比较难分的训练数据样本上；

2. 将弱分类器联合起来，使用加权的投票机制代替平均投票机制。让分类效果好的弱分类器具有较大的权重，而分类效果差的分类器具有较小的权重。

(3)本次实验的设计

“半圆对拱形”的数据集过去做过，而本次实验意在探究集成学习的优势，所以我们摒弃了之前可以解决这类线性不可分的 SVM 和 BP 神经网络，因为这些分类方法单个模型就可以很好的解决这种问题，无需集成学习。

相应的，线性不可分的单层感知机是无法很好的分割两类样本，所以，这就是很好的集成学习对比工具，不妨用单层感知机的线性分类器充当我们的弱分类器。

所以我们训练了 5 个单层感知机，还是采取普适的梯度下降法训练每个弱分类器，不同的是，本次实验还需要考虑权重的概念，也就是说 lost 的准则函数还需要乘上数据的权重，体现到代码，也就是

```
lost=lost+0.5*(re-1)^2*dd(i,1);
```

其中，dd 是储存数据权重的向量，具体调整权重的计算公式在上边的算法中已经给出，不必赘述。

另外，需要强调的问题是，不同的弱分类器应该是串行训练的关系，而万万不可以设计成并行的，如果设计成并行训练的方式，就变成了我们的另一种集成学习方法，而失去了 adaboost 的核心思想。

四、实验结果

图 1 是我用梯度下降训练的 5 个单层感知机线性分类器，它们存在着不同程度的线性不可分程度。

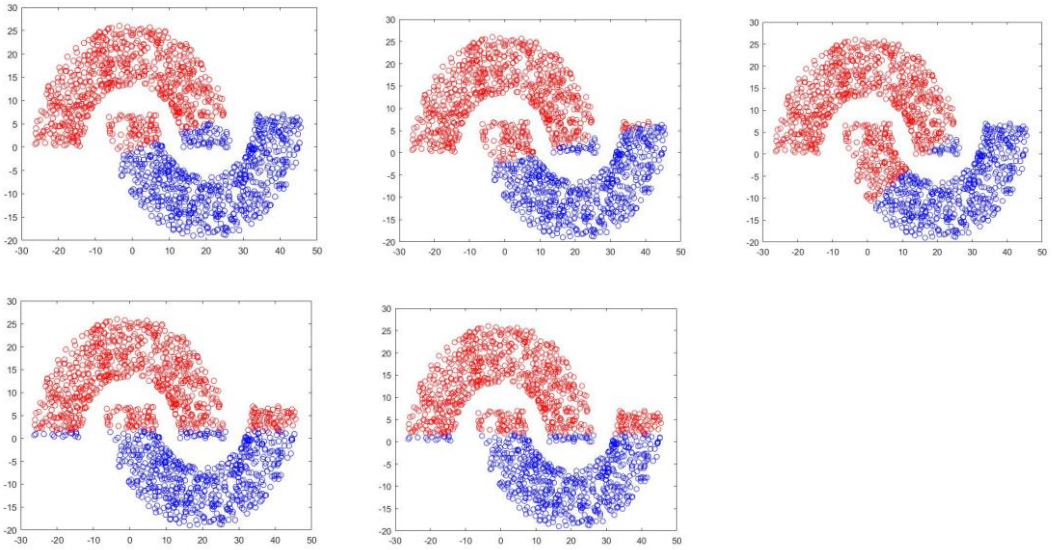

图 1 五个弱分类器分类情况

图 2 是最终的强分类器分类情况：

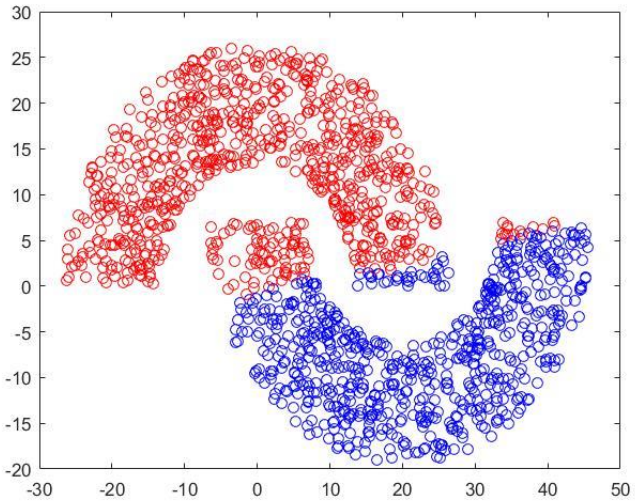

图 2 强分类器分类情况

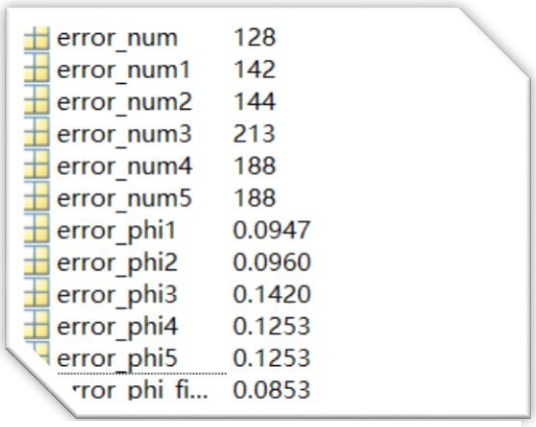

图 3 matlab 数据截图

左图是 matlab 的数据截图，128 个（一共测试 1500 个）的 error_num 来自强分类器，其余来自弱分类器；0.0853 的错误率来自强分类器，其余来自弱分类器；

比较统计图作图如下：

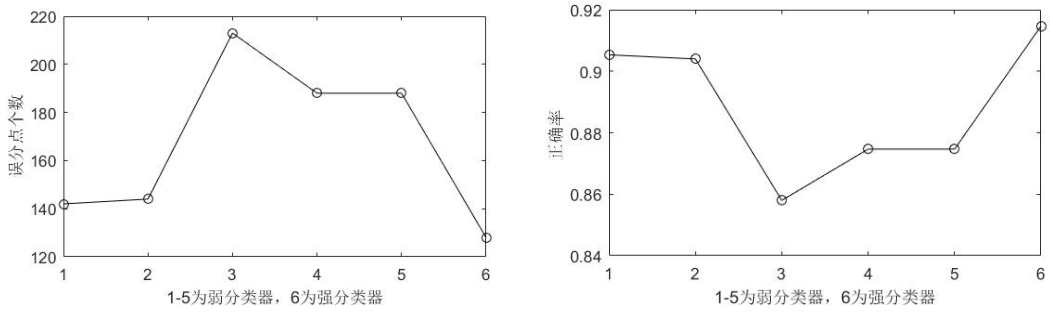

图 4 统计数据折线图

可以看出，强分类器的 128 个五分点是最少的，0.915 的正确率是最高的，可以看出，adaboost 集成学习的分类效果相比单个线性分类器有明显提升。

实验效果已经比较理性，但还可以有所突破，只不过，我们只使用了 5 个弱分类器，试想使用 15 个、25 个…最终可以实现彻底的线性不可分，并且不用担心 overfitting 问题。这在周志华教授的文章里有过详细证明。

五、总结

最后，我们可以总结下 adaboost 算法的一些实际可以使用的场景：

1）用于二分类或多分类的应用场景

2）用于做分类任务的 baseline，无脑化，简单，不会 overfitting，不用调分类器

3）用于特征选择（feature selection）

4）Boosting 框架用于对 badcase 的修正

只需要增加新的分类器，不需要变动原有分类器由于 adaboost 算法是一种实现简单，应用也很简单的算法。Adaboost 算法通过组合弱分类器而得到强分类器，同时具有分类错误率上界随着训练增加而稳定下降，不会过拟合等的性质，应该说是一种很适合于在各种分类场景下应用的算法。

至此，实验比较的达到了预期目标。