1. 主要工作

本次作业对数据挖掘的常用算法 adaboost 算法进行了实现，并使用部分数据对算法的性能进行了测试，同时还与其他常见的分类算法做了简单的性能对比。本次实现当中算法的弱分类器基于 decision stump。

2. 算法概述

Adaboost 实际上是集成学习方法当中一个很经典的算法。集成学习主要包括 bagging 和 boosting 两种。所谓的 bagging 是说我们不断的将数据集进行采样，然后训练不同的分类器，最后采用投票的方式对所有分类器的算法进行综合，得到最后的结果。Bagging 思想产生的代表算法是随机森林。而 boosting 是通过对样本进行赋值，不断提高分类错误样本的比重，来进行训练，并且根据错误率给予不同分类器一定的权重，最后综合各个分类器得出最后的结果。Boosting 思想代表算法是 Adaboost 以及 GBDT.

Adaboost 算法的几个核心步骤在于（1）使用弱分类器。一般来说，选择的弱分类器需要好于随机猜测，也就是误差要小于 0.5，不过也有文献说如果错误率大于 0.5，可以将分类器权值设置成负数，这样也对最终效果有一定提升。另外通常来说，如果单个分类器太强，会影响最终分类的效果。（2）对分类器赋予权重。实际上，我们通过分类器的错误率来定义分类器的重要性大小，一般定义分类器器权重为: $a_m=\log((1-e_m)/e_m)$。至于为什么这么定义，可以参考后面给出的文章 1 当中的推导。（3）对错误样本提高权值，提高权值的大小和分类器的权重相关。使用指数增加和指数衰减的方式来修改样本权值。

本次实验选择 decision stump 也就是所谓的决策树桩作为弱分类器，这个分类器每次选择一个特征对不同类进行划分，最优划分阈值考虑遍历特征能够取到的所有值，然后选择使分类错误最小的那个值。

Adaboost 算法通常来说具有速度快，准确率高，不容易过拟合，而且基本不需要调整什么参数。一般来说，分类器错误率由 bias 和 variance 组成，弱分类器通常 bias 比较大，但是 variance 相对比较小，如果将这些弱分类器进行组合，一定程度上可以减小 bias，于是整体来看，分类器的 bias 和 variance 都会比较小，从而得到相对比较好的结果。又由于 adaboost 当中的权值选择和错误率相关，所以很少有需要调整的参数，这点非常重要。

3. 算法主要流程

算法步骤描述如下：

Algorithm 1 Adaboost .

Input:

training dataset$(X,Y)=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}, y_i\in\{-1,+1\}$;

iterators: M;

Output:

1: initialize each sample weight: $D_{1,i}=\frac{1}{N}, i=1,2,\cdots,N$, and $f_0(x)=0$

2: **for** $m=1$ to M **do**

3: training a base learner: $G_m(x)$

4: calculate error ratio: $e_m=\sum_{i=1}^{N} D_{mi} I(G_m(x_i)\neq y_i)$

5: if $e_m>0.5$ then break

6: $\alpha_m=\frac{1}{2}ln(\frac{1-e_m}{e_m})$

7: update sample weight: $D_{m+1,i}=\frac{D_{m,i}}{Z_m}\cdot exp\{-\alpha_m y_i G_m(x_i)\}$,
$Z_m=\sum_{i=1}^{N} D_{m,i}\cdot exp\{-\alpha y_i G_m(x_i)\}$

8: $f_m(x)=f_{m-1}(x)+\alpha_m G_m(x)$

9: **end for**

10: **return** $G(x)=sign(f_M(x))=sign(\sum_{m=1}^{M}\alpha_m G_m(x))$.

其中涉及到的一些关键的参数，分类器的权值更新参数以及样本的权值更新参数的推导参考文章 1。其余不赘述。

4. 实验结果

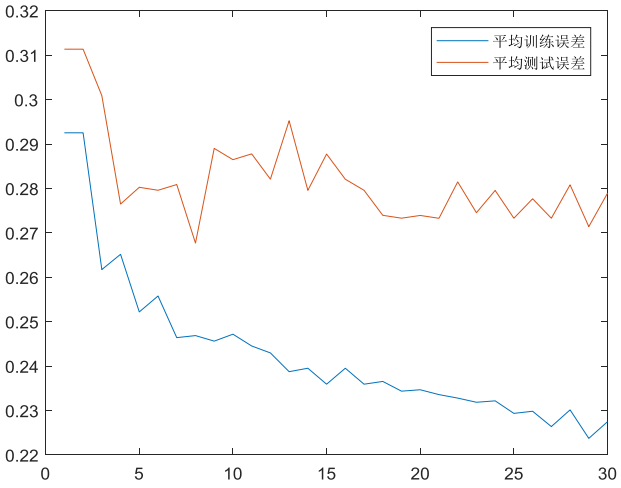
（1） 实验数据集

数据集 1 选择 http://archive.ics.uci.edu/ml/index.php 网站当中提供的 wine_quality 数据对实验进行测试。原始网站对酒的质量的分类用十二个等级来评价，我们对数据进行了预处理，将这 12 个等级人为的分成两类，并给定标签（-1,+1）。除了标签之外，数据还包含十一个特征，分别是：fixed acidity，volatile acidity, citric acid, residual, sugar, chlorides, free sulfur dioxide, total sulfur dioxide，density, pH, sulphates, alcohol，这些特征都是连续值，且没有缺失。经过预处理之后，实验数据集一共有样本 1599 个，其中正样本 855 个，负样本 744 个。为了后续处理数据方便，在实验当中对数据集中每一个特征都做了（0,1）归一化。

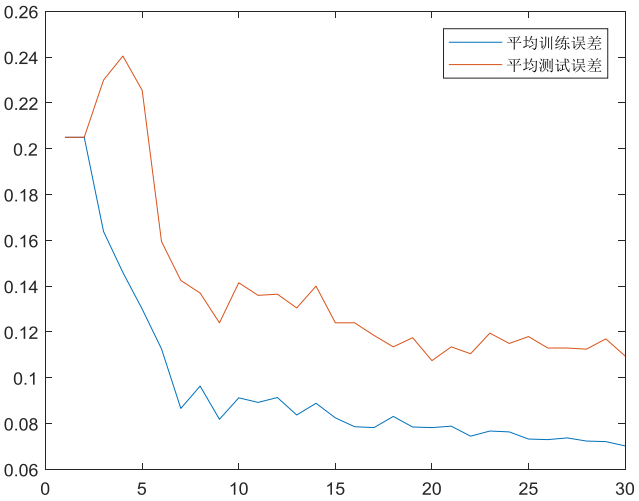
数据集 2 选择相同网站提供的分娩心电图描记法数据集，我们从原始数据特征当中选择了 21 个特征，包括胎心率（FHR）等相关特征，这些特征也都是连续值，没有缺失。经过预处理之后，实验数据集当中共有样本 2000 个，其中正样本 1027 个，负样本 973 个。实验当中对各个特征进行了（0,1）归一化

（2） 实验结果

a) 5 折交叉验证，最多 30 个分类器时，两个数据集误差变化情况如下：



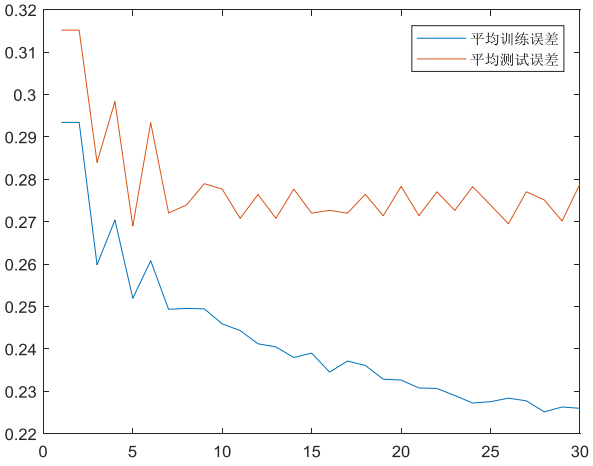
数据集 2：



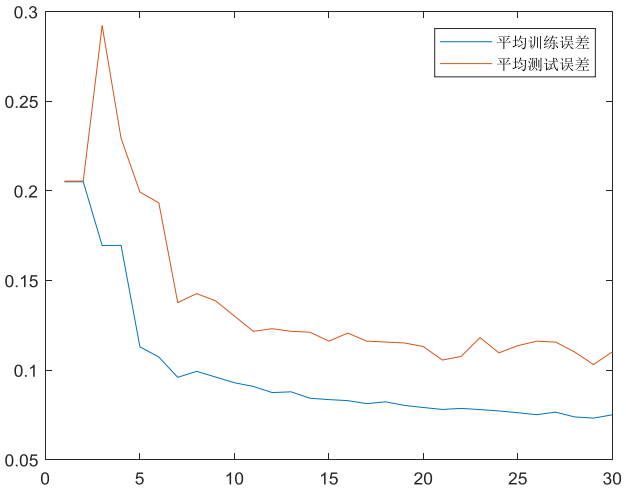
显然本次实验的数据集 1 上在 6 个分类器以后在再加分类器，虽然训练误差还在下降，但测试误差已经不再下降，此参数下在此数据集上的平均测试误差约为 28%左右。

数据集 2 上 10 个分类器，性能已经基本比较稳定，后续增加分类器作用不大，在此数据集上平均误差为 12%左右。

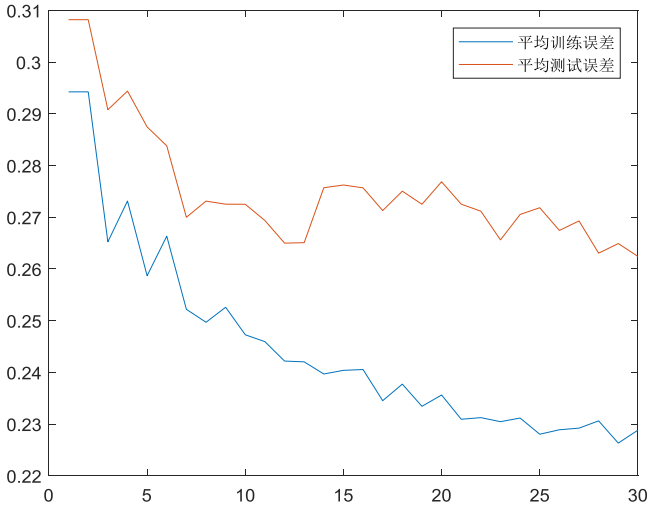
b) 7 折交叉验证，最多 30 个分类器，误差变化情况



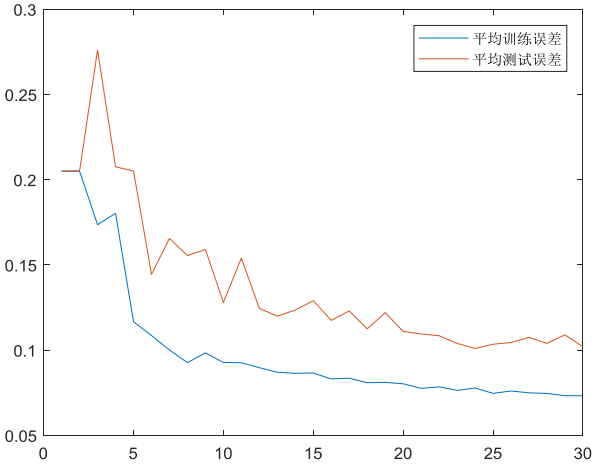
数据集 2：



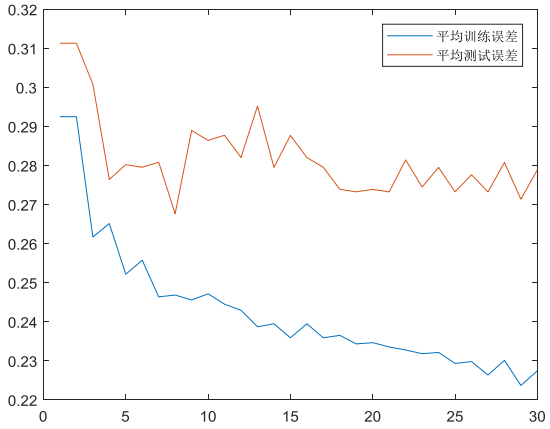
c) 9 折交叉验证，最多 30 个分类器，误差变化情况



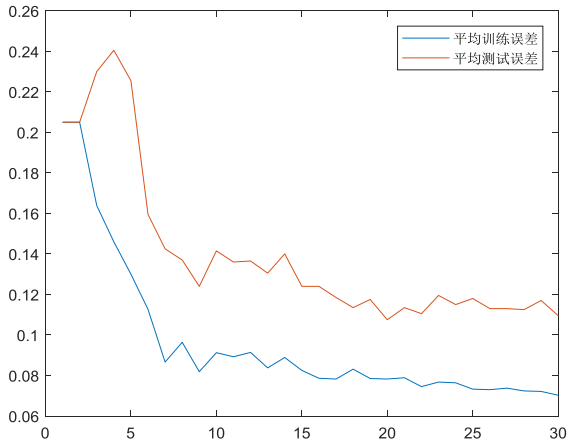
数据集 2：



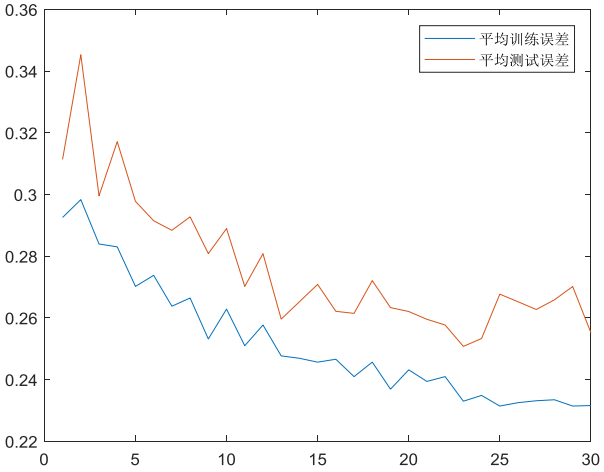
d) 去掉算法更新样本权值时候的 Zm 步骤，也就是去掉归一化步骤，5 次交叉验证，最多 30 个分类器，实验误差变化情况：



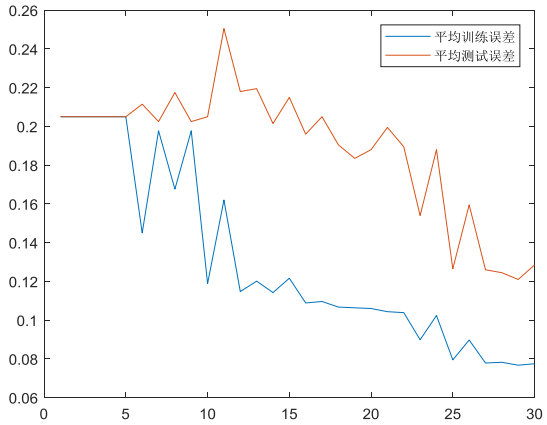
数据集 2：



e) 直接将分类器权值设置成常数 0.1，5 折交叉验证，最多 30 个迭代器



数据集 2：



f) 其他分类器分类结果

使用 SVM 算法：

数据集 1：

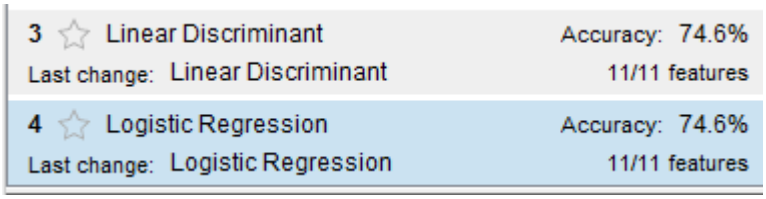
1.1 ☆ SVM Last change: Linear SVM	Accuracy: 73.9% 11/11 features
1.2 ☆ SVM Last change: Quadratic SVM	Accuracy: 74.0% 11/11 features
1.3 ☆ SVM Last change: Cubic SVM	Accuracy: 75.7% 11/11 features
1.4 ☆ SVM Last change: Fine Gaussian SVM	Accuracy: 76.1% 11/11 features
1.5 ☆ SVM Last change: Medium Gaussian SVM	Accuracy: **76.7%** 11/11 features
1.6 ☆ SVM Last change: Coarse Gaussian SVM	Accuracy: 74.2% 11/11 features

数据集 2：

3.1 ☆ SVM Last change: Linear SVM	Accuracy: 88.8% 21/21 features
3.2 ☆ SVM Last change: Quadratic SVM	Accuracy: 93.1% 21/21 features
3.3 ☆ SVM Last change: Cubic SVM	Accuracy: 93.8% 21/21 features
3.4 ☆ SVM Last change: Fine Gaussian SVM	Accuracy: 87.5% 21/21 features
3.5 ☆ SVM Last change: Medium Gaussian SVM	Accuracy: 91.6% 21/21 features
3.6 ☆ SVM Last change: Coarse Gaussian SVM	Accuracy: 87.7% 21/21 features

使用 logistic regression 和 linear discriminant analysis:

数据集 1:



数据集 2：

5. 结果分析

从结果上来看，对本次作业实现的 Adaboost 算法，不同的交叉验证对结果有相对比较小的影响，而且从实验结果来看一般 Adaboost 算法需要级联的分类器不需要很多，通常对分类而言 10 个左右效果应该就比较稳定了。从实验结果来看，使用固定的分类器权值在不同的数据集上效果不稳定，整体来看不如使用错误率来对分类器权重进行评价效果好。另外，从实验来看，对权值进行归一化本身对实验结果没有影响，因为从理论上来说，归一化相当于对所有的权值都乘以一个数，这个数级联起来还是乘积，对数值的大小排序关系没有影响，可能概率上的意义更大一些。

和一些常见的分类算法对比来看，本次实现的模型和常见模型的线性模型性能相似，相比于非线性的模型，比如高斯核和多项式核的 SVM 相比，性能稍微差一点。不过基于 decision stump 的 Adaboost 相比这些非线性模型没有太多参数需要调整，且训练速度相对较快。

6. 参考资料

a） http://www.inf.fu-berlin.de/inst/ag-ki/adaboost4.pdf