**华中科技大学**

**计算机科学与技术学院**

**《机器学习》结课报告**



专 业： 计算机科学与技术

班 级： 计卓2001

学 号： U202015290

姓 名： 朱子城

成 绩：

指导教师： 张腾

**完成日期： 2022年 7 月 5 日**

目录

[1. 实验要求 2](#_Toc106106989)

[2. 算法设计与实现 2](#_Toc106106990)

[2.1 数据预处理 2](#_Toc106106991)

[2.2 对数几率回归实现 2](#_Toc106106992)

[2.3 决策树桩实现 3](#_Toc106106993)

[2.4 Adaboost算法实现 4](#_Toc106106994)

[3. 实验环境与平台 5](#_Toc106106995)

[4. 结果与分析 5](#_Toc106106996)

[5. 个人体会 6](#_Toc106106997)

Adaboost算法实现

# 实验要求

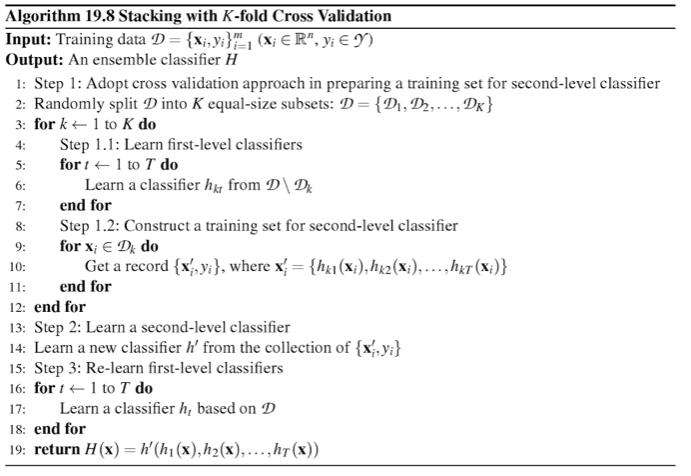
本次实验要求分别实现以对数几率回归和决策树桩为基分类器的 AdaBoost 算法。实现了对数几率回归时调整学习率的固定值，并能够基于给出的数据完成基分类器和AdaBoost模型的训练和十折交叉验证。

# 2. 算法设计与实现

## 2.1 数据预处理

观察实验数据可得，本次实验所给数据不存在缺失数据，因此无需对数据进行缺失值填充。

首先从指定的数据csv函数中读取训练用数据矩阵和标签值向量。由于需要进行十折交叉验证，因此在kfold\_split函数中将样本随机分成十份，随机取一份作为验证集，九份作为训练集。伪代码如图所示：



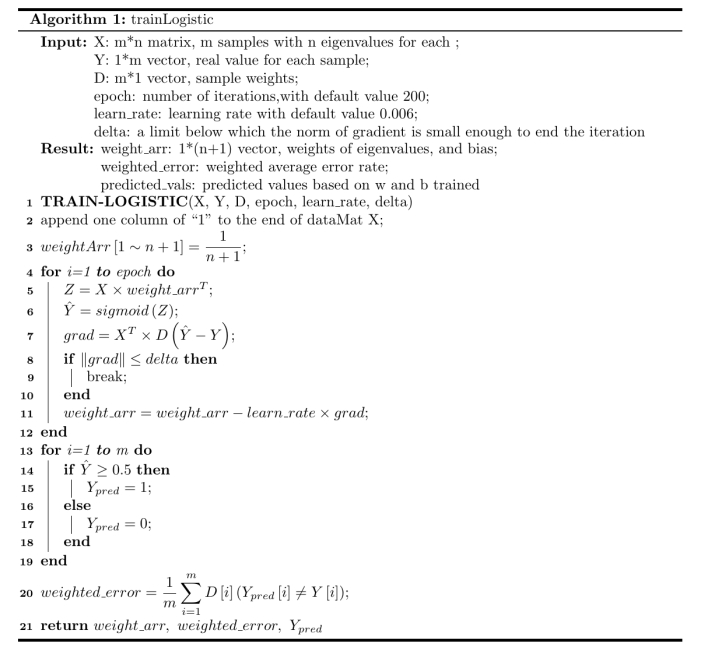
第二步是对特征值内部进行归一化。在决策树桩分类时，由于只要保证大小关系不变，不同特征值的数量级差距不影响分类结果，因此无需进行归一化。而在对数几率回归实现中，求出了每个特征值的平均值和标准差，并对特征值进行了零均值规范化。

## 2.2 对数几率回归实现

为了解决一个二分型问题，需要这样的函数模型：一个分段函数，不连续，在取中间值时函数变化陡峭，参数在取+∞或-∞时，函数趋近与1和0。而对数几率函数可以很好的拟合这些特性，其标准形式为



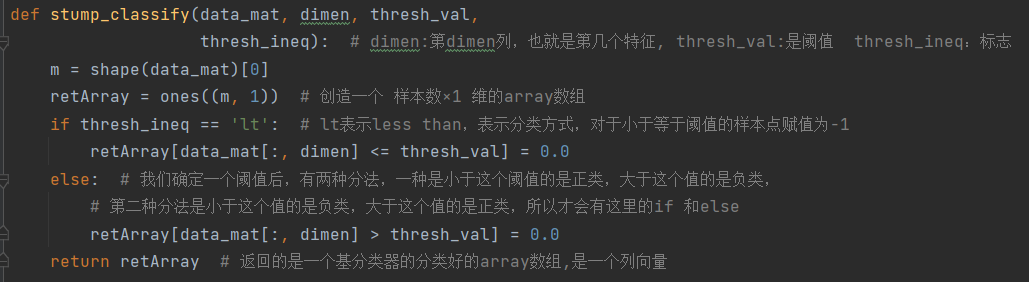
该函数实际上是一种概率分布函数，它表示的是随机事件发生的概率,有着单调递增，连续以及值域在 [0,1][0,1] 等良好特性,伪代码如图所示：



## 2.3 决策树桩实现

单层决策树（decision stump），也称决策树桩，它是一种简单的决策树，通过给定的阈值进行分类。从实际意义上来看，决策树桩根据一个属性的单个判断(但是实际上待判断的物体具有多个属性)就确定最终的分类结果。这种特性比较适合做集成学习中的弱学习器，因为其至少比随机的效果好一些，又计算较为容易。

如图是决策树桩的分类方法。参数表中有阈值thresh\_val、操作标志thresh\_ineq、当前正在分类的项是data\_mat[:, index]。当flag==’lt’，那么小于阈值的是正类，大于阈值的是负类，flag！=’lt’则相反。



## 2.4 Adaboost算法实现

Adaboost是一种迭代算法，其核心思想是针对同一个训练集训练不同的分类器(弱分类器)，然后把这些弱分类器集合起来，构成一个更强的最终分类器（强分类器）。Adaboost 算法系统具有较高的检测速率，且不易出现过适应现象。但是该算法在实现过程中为取得更高的检测精度则需要较大的训练样本集，在每次迭代过程中，训练一个弱分类器则对应该样本集中的每一个样本，每个样本具有很多特征，因此从庞大的特征中训练得到最优弱分类器的计算量增大。典型的 Adaboost 算法采用的搜索机制是回溯法，虽然在训练弱分类器时每一次都是由贪心算法来获得局部最佳弱分类器，但是却不能确保选择出来加权后的是整体最佳。在选择具有最小误差的弱分类器之后，对每个样本的权值进行更新，增大错误分类的样本对应的权值，相对地减小被正确分类的样本权重。且执行效果依赖于弱分类器的选择，搜索时间随之增加，故训练过程使得整个系统的所用时间非常大，也因此限制了该算法的广泛应用。另一方面，在算法实现过程中，从检测率和对正样本的误识率两个方面向预期值逐渐逼近来构造级联分类器，迭代训练生成大量的弱分类器后才能实现这一构造过程。由此推出循环逼近的训练分类器需要消耗更多的时间。

该算法其实是一个简单的弱分类算法提升过程，这个过程通过不断的训练，可以提高对数据的分类能力。整个过程如下所示：

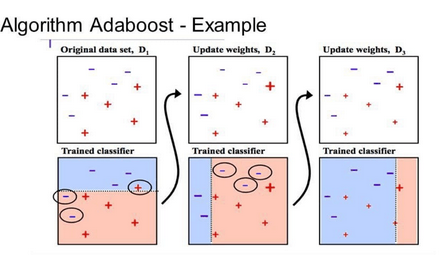
1. 先通过对N个训练样本的学习得到第一个弱分类器；

2. 将分错的样本和其他的新数据一起构成一个新的N个的训练样本，通过对这个样本的学习得到第二个弱分类器 ；

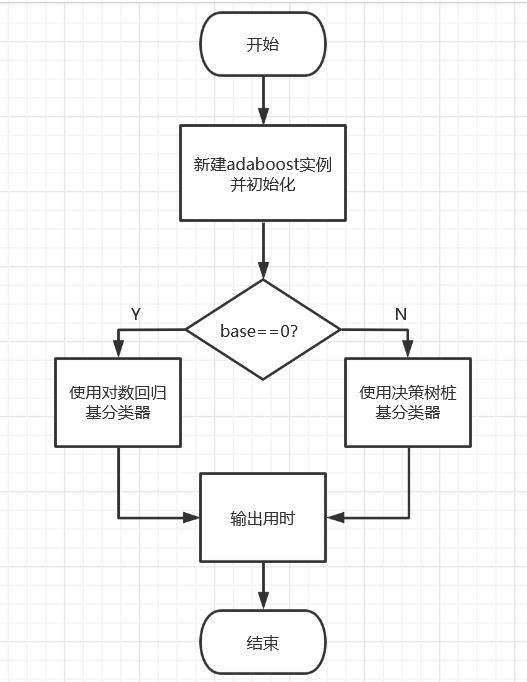
3. 将1和2都分错了的样本加上其他的新样本构成另一个新的N个的训练样本，通过对这个样本的学习得到第三个弱分类器；

4. 最终经过提升的强分类器。即某个数据被分为哪一类要由各分类器权值决定。

由Adaboost算法的描述过程可知，该算法在实现过程中根据训练集的大小初始化样本权值，使其满足均匀分布，在后续操作中通过公式来改变和规范化算法迭代后样本的权值。样本被错误分类导致权值增大，反之权值相应减小，这表示被错分的训练样本集包括一个更高的权重。这就会使在下轮时训练样本集更注重于难以识别的样本，针对被错分样本的进一步学习来得到下一个弱分类器，直到样本被正确分类。在达到规定的迭代次数或者预期的误差率时，则强分类器构建完成。如图所示：



本次实验封装了一个adaboost类，代码流程图如图所示：



# 3. 实验环境与平台

Python版本：3.9.2

集成开发环境：PyCharm Community Edition 2021.3

操作系统：Windows 10

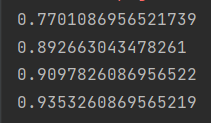
处理器：Intel(R)Core(TM)i7-10750H CPU @ 2.60GHz 2.59 GHz

RAM:16.0GB

# 4. 结果与分析

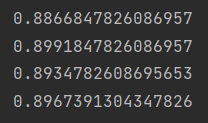
运行后能够成功将预测结果输入对应csv文件，且经过evaluate.py测试，结果符合正确率要求。

以对数几率回归作为基分类器时，预测结果的准确率如下图：



对率函数为基分类器实现的adaboost中，基分类器数目对结果的影响很小，而学习率对结果影响稍大。

以决策树桩作为基分类器时，预测结果的准确率如下图：



可见随着基分类器数量从1到5、10正确率提升幅度较大，10到100提升幅度较小，可见10个基分类器时已经基本可以满足预测要求，100个基分类器则是更进一步。

# 5. 个人体会

通过机器学习这门课程的学习，我学到了很多非常经典的机器学习算法，如KNN、SVM、Logistic、朴素贝叶斯等。很多算法的实现过程并不难，却能取得非常好的效果，体现了算法的巧妙。

此次实验综合性很强，使我更加理解决策树桩logistic回归和Adaboost算法的思想，理解了从数据处理到基分类器训练再到优化组合的基本流程，并将它们运用于实践中。初接到这个任务，我一头雾水，因为我还没有理清adaboost的原理，只是在课内学习了一些皮毛。于是我上网查阅大量资料，学习别人的代码，明白了adaboost的大致原理和实现方法。并一步一步在老师和同学的帮助下完成了整个实验。

此前的我从未接触过机器学习，这门课程作为我的启蒙课，通过老师讲解和项目的练习，不仅让我掌握了机器学习的基本知识，还让我对它产生了浓厚的兴趣。很感谢老师的教学和指导，使我收获颇丰。