**中山大学移动信息工程学院本科生实验报告**

**（2015年秋季学期）**

课程名称：Artificial Intelligence 任课教师：饶洋辉

|  |  |  |  |
| --- | --- | --- | --- |
| 年级 | 13级 | 专业（方向） | 移动信息工程 |
| 学号 | 13354223 | 姓名 | 刘晓松 |

**1&2 kNN & NB 使用lab2实验数据**

**1.阐述你的思路。**

**（1）kNN算法**

①首先我需要读取文本文件和每个训练文本的情感值。由于每个文本都有一些固定的属性，所以我需要一个数据结构将他们聚合起来，以文本为单位。

②得到不同的单词（这个在读文本的时候可以直接判断）以及计算每个文本的向量，并将他们归一化。

③选用stopwords，用来筛选掉一些停用词，可以有效防止一些无关紧要的介词，助词等的影响。

④计算每个测试文本与每个训练文本之间的距离（可以是欧式距离，也可以是city-block距离或者余弦距离）

⑤根据kNN算法，需要手动输入k。然后我们对于每个测试文本需要找到与之距离最小的k个训练文本（余弦距离需要找最大的k个文本）。

⑥ 将这k个文本距离进行取倒数（余弦不需要取倒数），然后将这些权重归一化。最后将这些权重乘以他们的情感值即为测试文本情感值。

⑦输出情感值到指定文本。

**（2）NB算法**

①②③与（1）相同

④在②中进行对**训练文本**向量归一化的时候，假设每个单词都事先出现一遍。这样就解决了乘法因子为零的问题。

⑤ NB算法：对于每个测试文本，将其每个不为0的维度对应的训练文本进行相乘。在将每个训练文本得到的值进行累加。

⑥对每个测试文本的六种情感进行归一化。

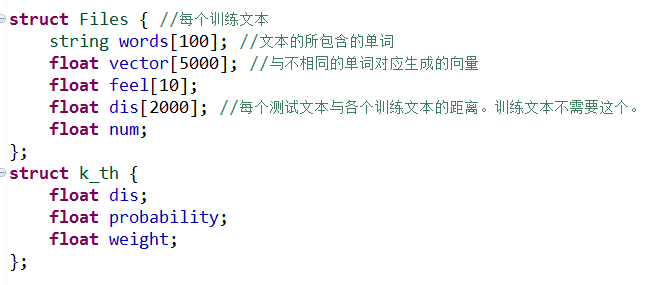
⑦输出得到的上述情感值到指定的文本

**2.截图你的关键代码（注明使用的是什么语言）**

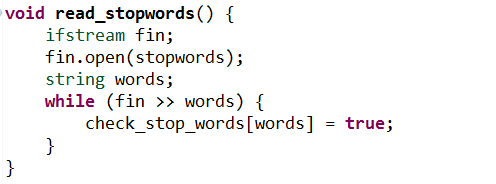
【说明】使用c++语言；

(1)kNN

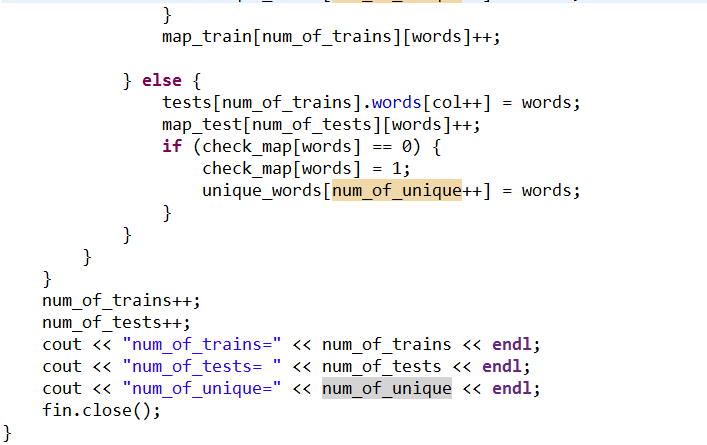
①关于文本的数据结构。k\_th为距离最近的k个训练文本的一些参数。



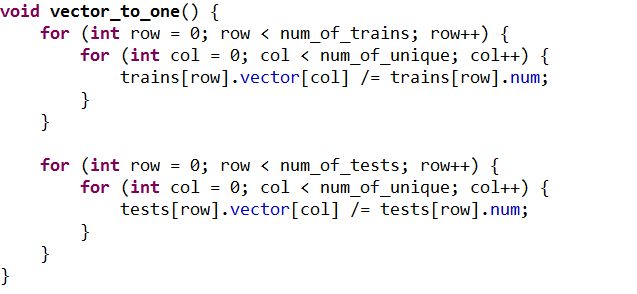
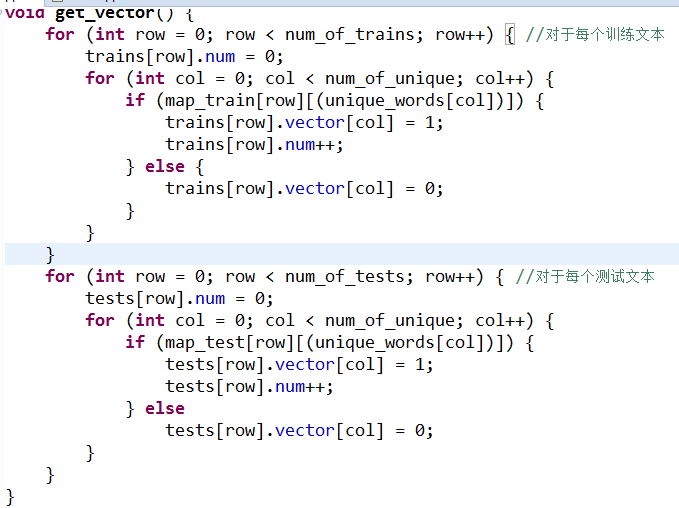
②读取stopwords以及读取数据集的词并筛掉stopwords。均用map来实现。Stopwords中以每个单词为关键字，返回的是bool类型。而读取数据集是同样是用单词做关键字，返回int类型，有利于计数，来记录每个单词在文本中出现的顺序。





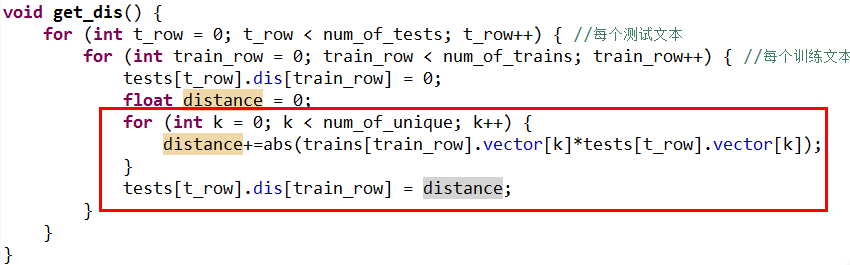


**③计算向量。以及向量的归一化**

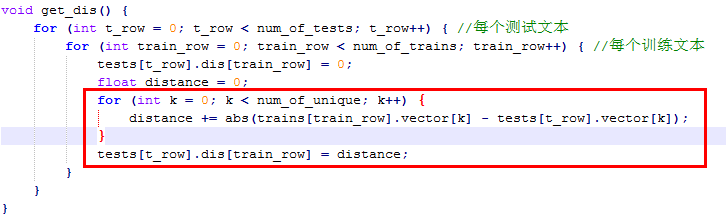


**④计算距离。最终我选用的是余弦距离，这个跑出来的效果最理想。**

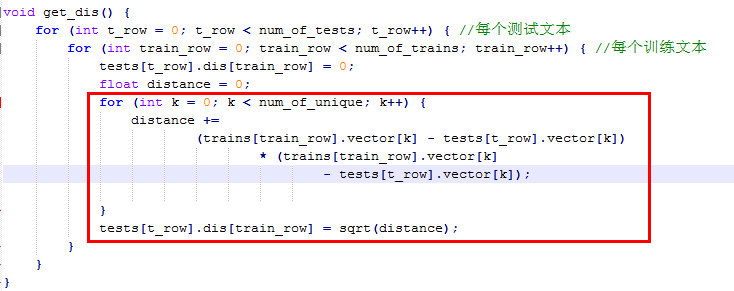
**A 余弦距离**



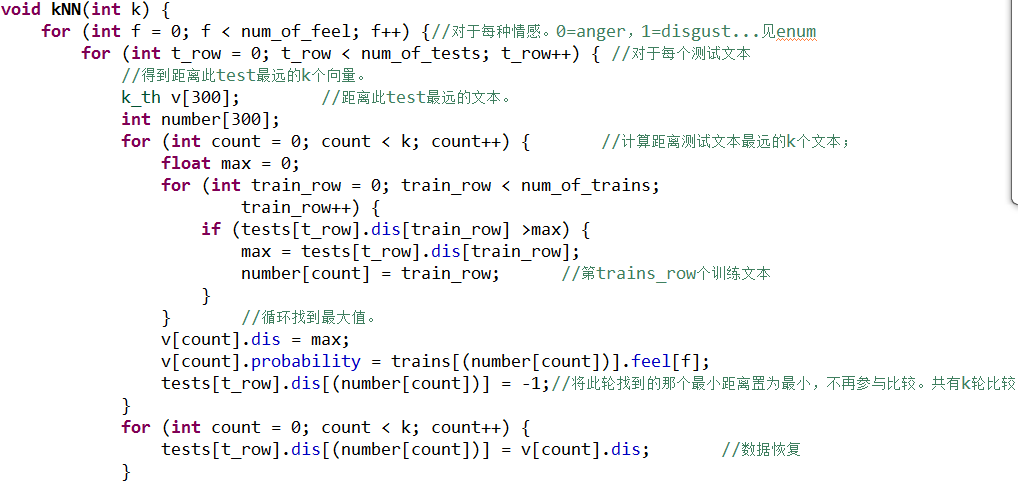
**B 曼哈顿距离**

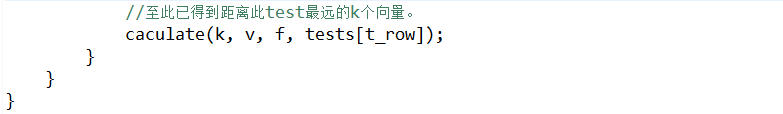


**C 欧氏距离**

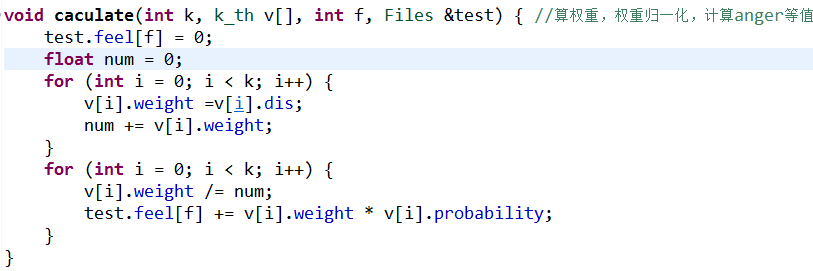


**④kNN。**这里计算k个最大值是采用了类似于冒泡的方法，循环k次，每次都找到最大值并且将其置为最小值不再参与比较，最后再将其数据恢复。这种方法避免了直接全排序，在k值比较小的情况下大大降低了时间复杂度。——以余弦距离来进行说明

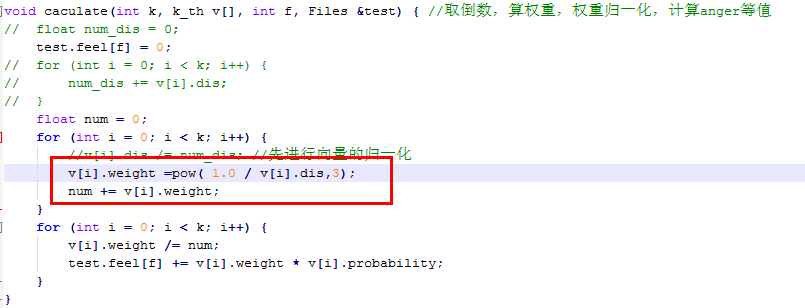




Calculate函数：用来计算权重和文本预测的情感值

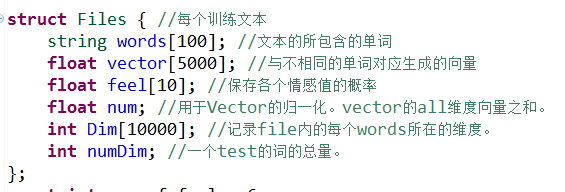


**Calculate函数2：**这个是实现欧式距离或者街区距离的关于权重的计算，需要用距离的倒数的三次方来进一步稀疏画得到的距离，有利于使得最接近的文本产生更好的权重

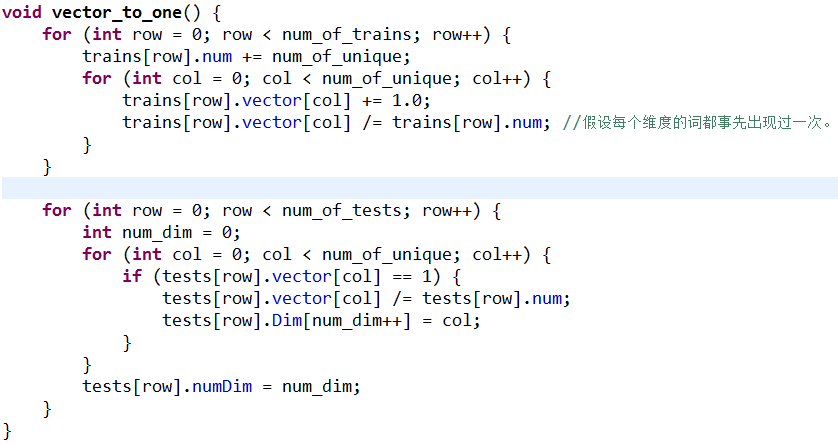


**(2)NB**

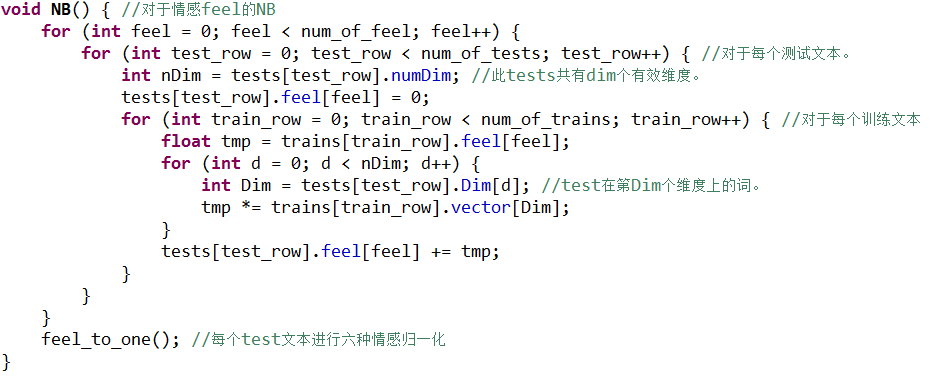
①NB算法的数据结构，他们的作用均写在声明后面

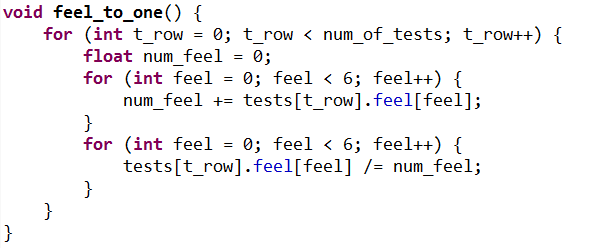


②关于文本的读取和kNN相同，不过,在vector进行归一化的时候，先要设定每个词都预先出现过一次



**③NB关键代码实现，以及6种情感的归一化**





**二. 实验结果**

**(1)kNN**

**A 余弦距离**

|  |  |  |
| --- | --- | --- |
| **①anger** | **②disgust** | **③fear** |
|  |  |  |
| **④joy** | **⑤sad** | **⑥surprise** |
|  |  |  |
| **平均值** | **0.300（k=10）** | |

**B city-clock距离**

|  |  |  |
| --- | --- | --- |
| **①anger** | **②disgust** | **③fear** |
|  |  |  |
| **④joy** | **⑤sad** | **⑥surprise** |
|  |  |  |
| **平均值** | **0.280（k=10）** | |

**(2)NB**

|  |  |  |
| --- | --- | --- |
| **①anger** | **②disgust** | **③fear** |
|  |  |  |
| **④joy** | **⑤sad** | **⑥surprise** |
|  |  |  |
| **平均值** | **0.280** | |

**【实验分析】**

①kNN算法中，我共前后选用了city-block距离，欧式距离，余弦距离三种，欧氏距离由于数据比较密集，不太好处理，所以实现的效果也就在0.20左右，街区距离的效果比较好一点，尤其是添加了筛掉停用词这一步，6种情感相关度的平均值可以达到0.28，后来我尝试用余弦距离来做，从理论上来讲余弦距离解决了前两种距离求差的一些误差，能将与文本完全不相关的训练文本剔去。

②在实现选取k个距离最大值作为权重的时候，我采取的方式是冒泡形式，只冒到第k个值就停止，避免了直接排序，减低了复杂度。

③ NB算法，除了上述方式我还尝试了使用一个很小的值代替0的情况（比如0.01），但是实现的效果均在0.25左右，并不能很好的实现。后来我又用1代替0，即忽略掉那个单词，这个的效果更差一些，大概平均值只有0.18的样子，所以最终还是选择了效果最好的一组。

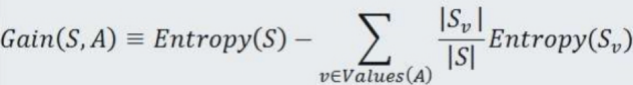
④ 上次的实验所交的材料中，实现的效果不是很理想，当时是选用街区距离进行计算的，后来加了停用词进行筛选之后，效果得到了显著提升。

**3.决策树—ID3**

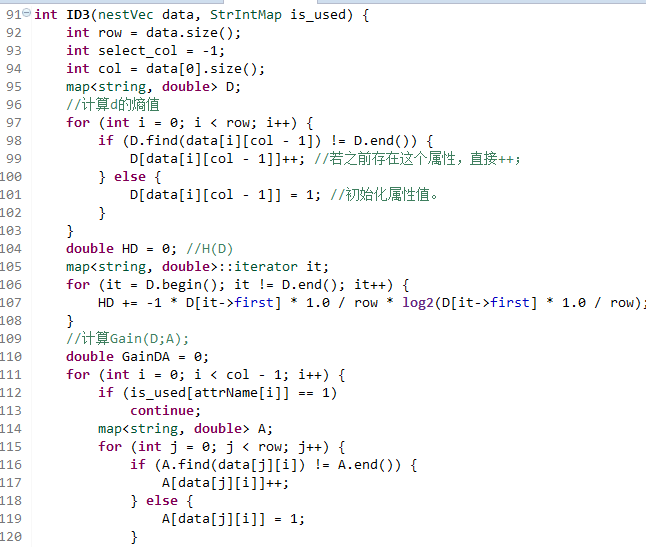
**【算法解析】**

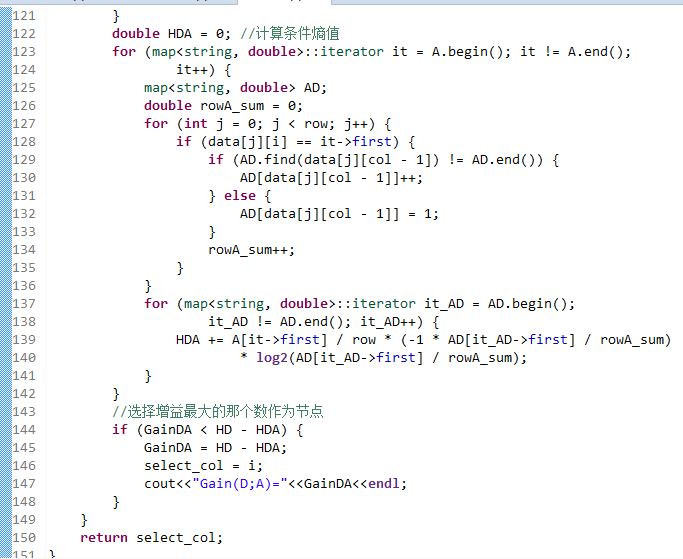
ID3算法的核心思想就是以信息增益度量属性选择，选择分裂后信息增益最大的属性进行分裂。

**增益是由信息熵进行计算的，具体公式如下：**

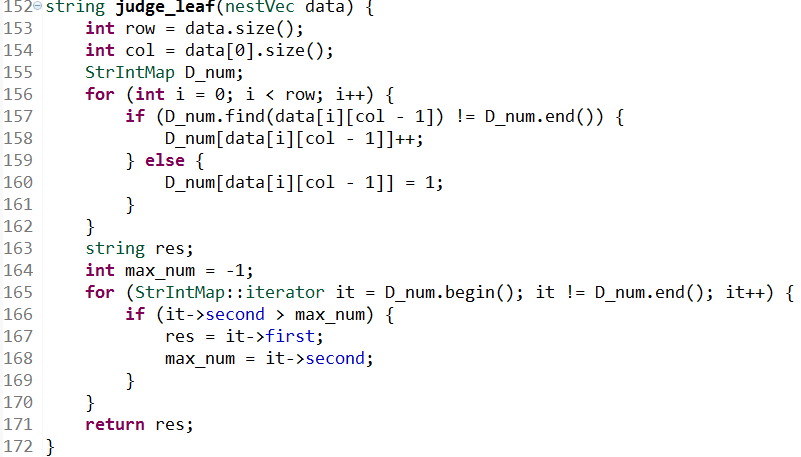


1. **计算熵值和增益，并选择最大增益的节点**

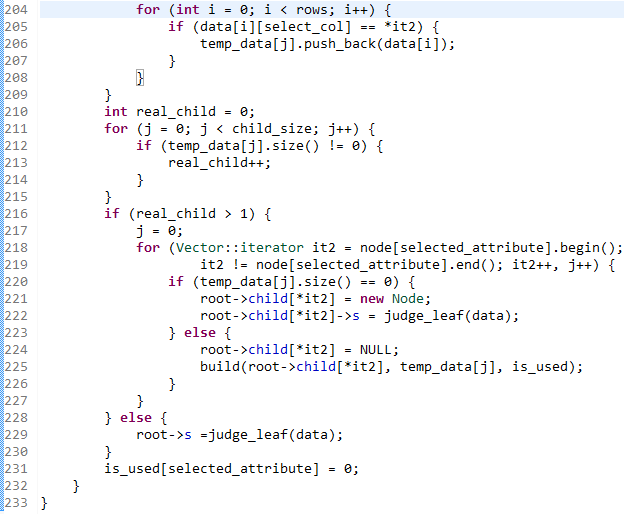
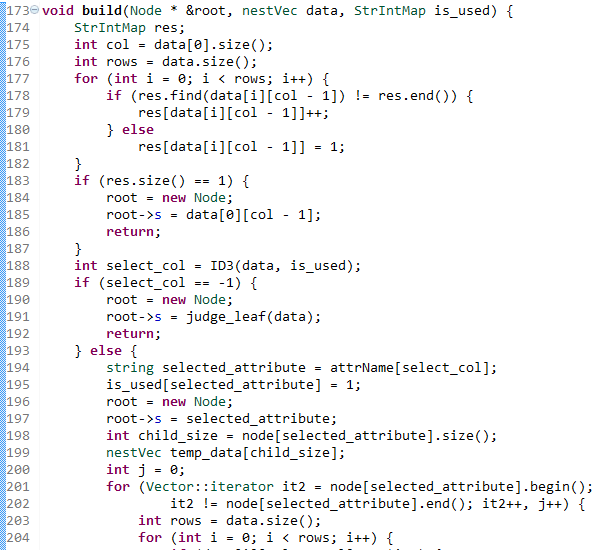




**(2)判断是否是叶子节点，即判断此树是否能够进行所有训练集的正确分类**



**(3)建树**

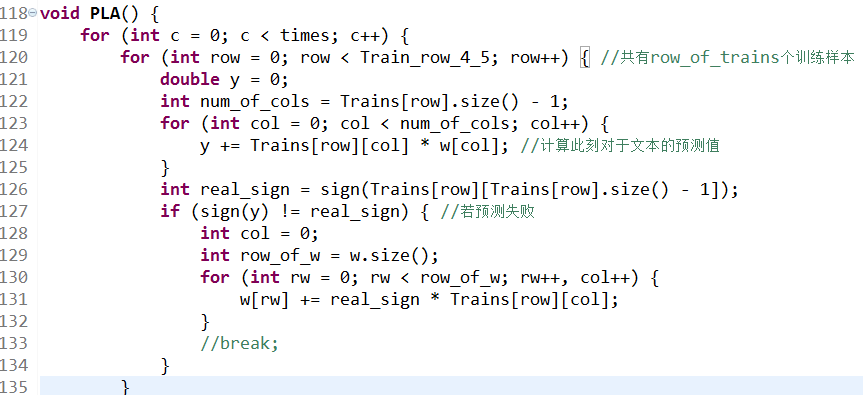


**【算法分析】**最终得到的结果如下

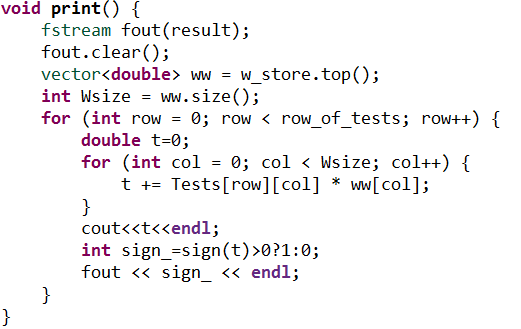


1. **PLA**

**（1）主体实现部分**



**(2)输出部分**



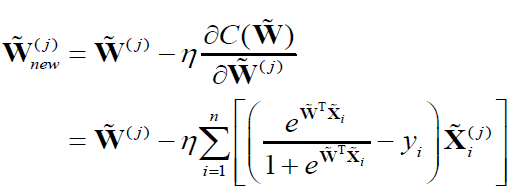
**【最终结果】**0.58



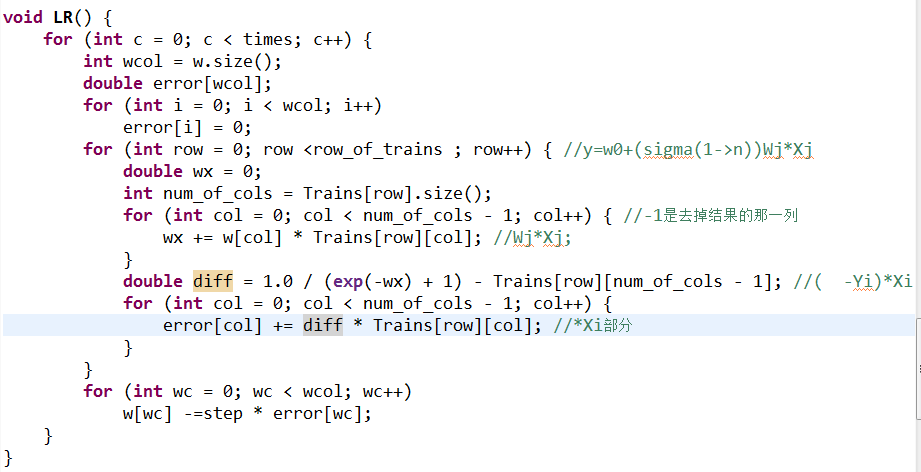
**5.LR**

**【算法解析】**

Logic Regression，逻辑回归算法



这本次实验中我先用一些常规的做法来实现。有实现进行向量列归一化，LR实现部分如下：



结果如下：



**二 实验结果分析+实验感想**

最终LR实现的效果最好是0.59。本来还想进行一些扩展，比如实现交叉验证，或者数据处理，但是实现交叉验证之后对于结果没有什么影响。因此放弃此优化方法。