目录

[一 Project内容 1](#_Toc471204196)

[二 贝叶斯分类器理论介绍 1](#_Toc471204197)

[三 贝叶斯分类器训练的MapReduce算法设计 2](#_Toc471204198)

[四 源代码清单 3](#_Toc471204199)

[五 数据集说明 16](#_Toc471204200)

[六 程序运行说明 16](#_Toc471204201)

[七 实验结果分析 21](#_Toc471204202)

# 一 Project内容

1：用MapReduce算法实现贝叶斯分类器的训练过程，并输出训练模型；

2：用输出的模型对测试集文档进行分类测试。测试过程可基于单机Java程序，也可以是MapReduce程序。输出每个测试文档的分类结果；

3：利用测试文档的真实类别，计算分类模型的Precision，Recall和F1值。

# 二 贝叶斯分类器理论介绍

问题：给定一个类标签集合C = {c1,c2, …, cj}以及一个文档d，给文档d分配一个最合适的类别标签

基本思想：对于类标签集合C中的每个类标签ci (i = 1, …, j), 计算条件概率p (ci |d)，使条件概率p (ci |d)最大的类别作为文档d最终的类别。

1、计算条件概率p (ci |d )

p (ci|d): 后验概率或条件概率(posterior)

p (ci): 先验概率(prior)

p (d|ci): 似然概率(likelihood)

p (d): 证据(evidence)

当观察到evidence p (d) 时，后验概率p (ci|d)取决于似然概率p (d|ci )和先验概率p (ci)。因为当evidence p (d) 已知时，p (d)成为常量，Bayes公式变成



2、朴素贝叶斯方法

对于类标签集合C中的每个类标签ci (i = 1, …, j), 计算条件概率p (ci |d)，使条件概率p (ci |d)最大的类别作为文档d最终的类别。



其中cd为文档d被赋予的类型， cd=使得条件概率p (ci |d)最大的类型。根据Bayes公式， cd=使得p (d | ci) p (ci)值最大的类型。

3、先验概率

假设类别标签集合C = {c1,c2, …, cj}

假设训练集D包含N个文档，其中每个文档都被标上了类别标签

首先估计先验概率p (ci) (i=1, …, j)



4、似然概率

假设类别标签集合C = {c1,c2, …, cj}

假设训练集D包含N个文档，其中每个文档都被标上了类别标签

还需要估计似然概率p (d | ci) (i=1, …, j)

为了估计p (d | ci) ，需要一个假设：Term独立性假设

文档中每个term的出现是彼此独立的

基于这个假设，似然概率p (d | ci) 的估计方法如下：

假设文档d包含nd个term：t1, t2, …, tnd

根据Term的独立性假设，有

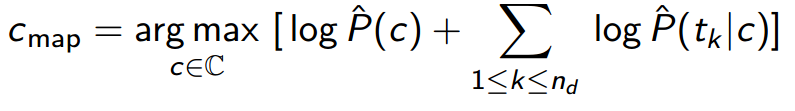


因此，估计p (d | ci) 就需要估计p (tk|ci)

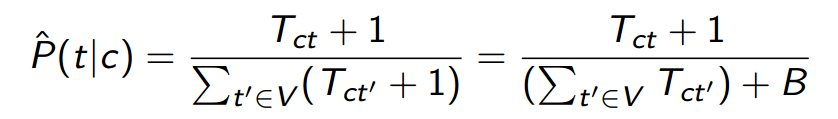


5、取对数和加一平滑

当单词表比较大时，每个单词出现的概率数值很小，在计算过程中会损失精度，导致最终结果的不准确，因此需要进行取对数处理，将很小的数值转化成的负数，同时计算结果大小也保持了一致性。



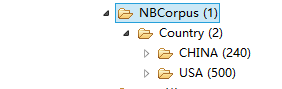
加一平滑，当某个文档中某些单词的出现次数为0，因此出现的概率也为0，这样该文档中所有单词概率的乘积也为0了，结果显示不出该文档先验概率的大小，因此需要加1处理，保证该单词出现概率很小的同时，也不会影响到最终结果的准确性。



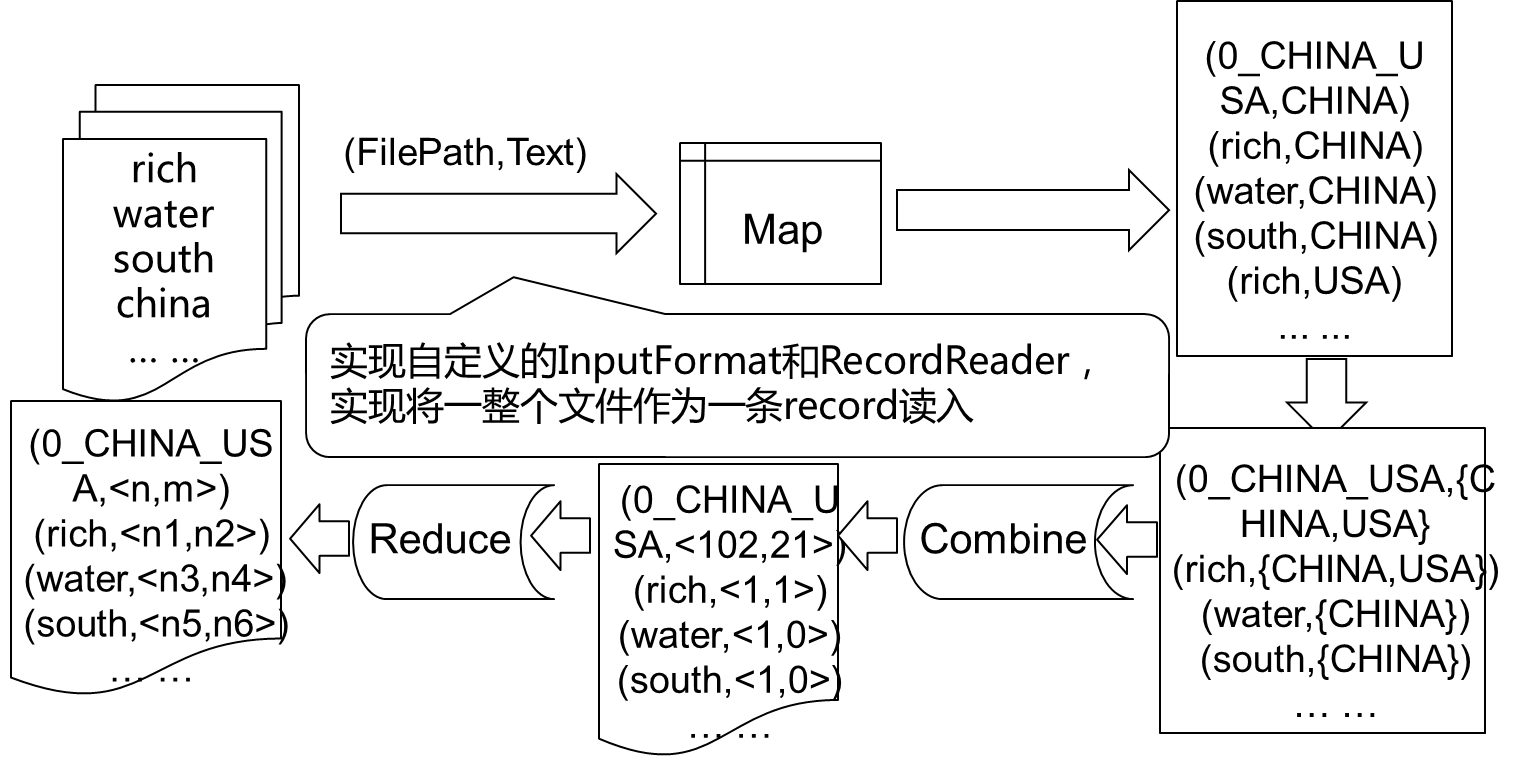
# 三 贝叶斯分类器训练的MapReduce算法设计

解释说明所设计的算法的执行过程，要求给出MapReduce的Data Flow示意图，并说明作业输入文件格式是是什么，Map的Key是什么含义，Value是什么含义，Reduce的输出Key，Value是什么含义

1、作业输入文件目录：



2、训练阶段的相关说明和DataFlow示意图



3、相关说明

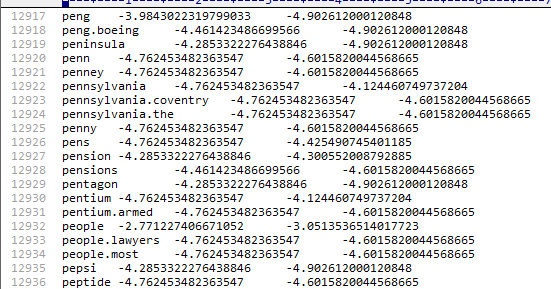
Map输入：重写了CombineFileInputFormat和CombineFileRecordReader，实现将小文件打包放入同一个Split中，使用同一个Map进行处理。Map输入的一对key和value分别对应一个小文件的文件路径和文件内容。

Map输出：从key中提取出文件所属类别（CHINA和USA），同时将文件text进行分词，对于每一个词都输出<单词，所属类别名>。此外，为了方便使用一个MapReduce实现文档的统计和单词的统计，在map的开始处，首先输出< 0\_CHINA\_USA,CHINA/USA>，0\_CHINA\_USA，作为专门统计文档数量的关键词。

Reduce的输入：<单词，类别名组成的列表>

Reduce的输出：<单词，{CHINA中出现的次数，USA中出现的次数}>，reduce分别统计value列表中出现的CHINA和USA次数，作为key分别在CHINA和USA中出现的次数，除此之外，遇到0\_CHINA\_USA时，统计的CHINA和USA次数，作为训练文档中CHINA文档的个数和USA文档的个数。Reduce的输出结果写入hdfs的文档中保存。

然后利用单机JAVA程序，依据朴素贝叶斯算法的基本原理，对结果文档逐行进行计算，分别将单词的频数替换成单词的概率，方便测试过程中依据训练模型的结果对文件类别进行判别，文档处理结果如下图：



# 四 源代码清单

要求对Map，Reduce方法的关键代码进行解释；源代码关键地方要有注释

说明:

包括5个java文件，分别为：Bayes.java，ProcessData.java，Test.java，CombineSmallfileRecordReader.java和CombineSmallfileInputFormat.java。其中Bayes.java为训练阶段的MapReduce，ProcessData.java为对训练阶段MapReduce输出数据进行相关处理的单机JAVA程序，Test.java为测试阶段的MapReduce程序，CombineSmallfileInputFormat.java和CombineSmallfileRecordReader.java分别为重写的CombineSmallFile，实现将小文件打包，减少Split的个数，提高系统运行效率。

# 五 数据集说明

用了多少个文档，多少个类别，每个类别的文档个数?

训练阶段输入文件共740个英文单词文件，其中CHINA类240个文件，USA类500个文件。

测试阶段共测试了250个文档类别。

# 六 程序运行说明

要求说明运行时有多少个Map任务，多少个Reduce任务，并说明为什么会有这么多Map任务；要求给出程序运行时基于Web页面的作业监控截图和程序运行截图

1. 训练阶段MapReduce

训练阶段共2个map任务，1个Reduce任务。由于在MapReduce的main方法中规定了conf.setInt("mapred.min.split.size", 1)和 conf.setLong("mapred.max.split.size", 512000)，而输入文档的总大小为800多KB，一个split的大小为500KB，所以分为2个split，对应两个MAP进行处理。

1. 测试阶段MapReduce

测试阶段共3个map任务，1个Reduce任务。由于在MapReduce的main方法中规定了conf.setInt("mapred.min.split.size", 1)和 conf.setLong("mapred.max.split.size", 102400)，而输入文档的总大小为300多KB，一个split的大小为100KB，所以分为3个split，对应三个MAP进行处理。

# 七 实验结果分析

计算分类结果的Precision，Recall和F1值。

表1 CHINA类别邻接表 表2 USA类别邻接表

|  |  |  |
| --- | --- | --- |
| 真实类别  分类结果 | YES | NO |
| YES | 15(TP) | 3(FP) |
| NO | 0(FN) | 232(TN) |

|  |  |  |
| --- | --- | --- |
| 真实类别  分类结果 | YES | NO |
| YES | 232(TP) | 0(FP) |
| NO | 3(FN) | 15(TN) |

表3 合并邻接表

|  |  |  |
| --- | --- | --- |
| 真实类别  分类结果 | YES | NO |
| YES | 247(TP) | 3(FP) |
| NO | 3(FN) | 247(TN) |

Precision=TP/(TP+FP) =247/(247+3)=0.988

Recall=TP/(TP+FN) =247/(247+3)=0.988

F1=2×Precision×Recall/( Precision +Recall)=0.988