

《MapReduce海量数据处理》

课程 实验开题报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **研究题目** | **：** | YouTube热门视频记录分析 | | | | | |
| **小组成员** | **：** | 林喜鹏 李朝阳 饶璐 唐玉婷 | | | | | |
| **学号** | **：** | MG1833046 MF1833035 MF1833059 MF1833070 | | | | | |
| **小组组长** | **：** | 林喜鹏 | | | | | |
| **联系信息** | **：** | linxp@smail.nju.edu.cn | | | | | |
| **时间** | **：** | 2018 | 年 | 12 | 月 | 19 | 日 |

1. 研究问题概述和背景

YouTube（世界著名的视频分享网站）维护着其平台上不同国家和地区每天的热门视频列表记录[1]。为了确定年度最热门的视频，YouTube使用了多个评价指标，其中包括浏览量、衡量用户互动的评论数和点赞数以及反对数等。 也就是说，最热门的视频其实并不一定是全年中观看次数最多的视频！最终的热门视频列表包括是音乐视频，名人访谈和真人秀电视节目以及YouTube上众所周知的恶搞视频。

我们希望通过分析YouTube热门视频记录来得出一些有用的结论，例如，流行的视频是什么类型的，不同国家地区的人都偏好什么样的视频，对视频相关描述进行情感分析等等；

1. 拟解决的问题
2. 根据视频标签(tag)统计不同国家和地区以及全网的热门视频的词云（WordCloud），得到不同国家地区以及全网在流行什么；
3. 统计不同国家地区以及全网总的浏览量、点赞数、反对数和评论数，进行对比，可以得到在哪个国家地区的视频更容易被浏览、点赞、反对或者评论；
4. 对视频的相关描述进行情感分析，了解视频表达的基本态度；
5. 统计热门视频主要的指标（包括**浏览量**和点赞数、反对数和评论数）之间的关联性，从而综合探究出一个热门视频具有的什么特征才更容易被浏览；
6. 主要技术难点
7. 数据量非常大，总记录大致有40万条，且每条记录的数据量也比较大，传统算法运行时间过长，需要使用并行化的策略进行分析；
8. 在对标签进行统计的时候，实际上就是对标签进行WordCount计数，此处非常适合使用MapReduce进行并行化；
9. 在对主要的几个评价指标进行统计的时候，实际上也就是对这些指标值进行累加求和，使用MapReduce可以很简单地实现并行化，并且避免传统算法下可能导致的算术溢出问题；
10. 在对视频的相关描述进行情感分析的时候，需要使用自然语言处理的情感分析算法，由于Java进行自然语言处理并不方便，我们可能需要使用 Jython [2] 调用python相关算法库；
11. 统计指标之间的相关性显然也不适合使用传统算法，同时由于浏览量是我们比较关心的指标，我们希望得到其他指标关于浏览量的一个更为准确的数学模型；
12. 基本设计思路和可行性分析
13. 使用MapReduce实现WordCount算法对标签进行计数，统计不同国家和地区以及全网的热门标签，然后对结果进行处理并绘制对应WordCloud词云图；

WordCount算法在MapReduce中已经非常成熟，中间结果处理主要是去除诸如只出现一次的标签，防止词云图过大；

1. 对主要的几个评价指标进行统计的时候，实际上也就是对这些指标值进行累加求和，使用MapReduce可以很简单地实现并行化；

这里主要是考虑在统计全网的指标值的时候，需要考虑可能出现的算术溢出问题，保证结果的正确性；

1. 在对视频描述进行分析的时候，首先可能需要使用TextBlob[3] 或者自己实现类似的情感分析算法，得到视频描述的情感倾向，然后对其进行统计的过程就和前面进

行标签计数类似；

1. 在分析浏览量关于其他指标的关系的时候，我们希望得到一个浏览量关于点赞数、反对数和评论数的数学模型；

初步采用简单的线性回归（由于MapReduce没有类似于TensorFlow等深度学习工具自动求梯度的接口，所以需要自己手工求梯度并实现，这也限制了模型结构不能过于复杂，否则难以求梯度表达式）；梯度更新可以采用经典的随机梯度下降，我的思路如下：

步骤1：为了保持原始数据的分布，需要打乱原始数据，思路是在Mapper端输出随机值作为key，输出当前记录作为value，在Reducer端直接遍历每个key的所有values，直接输出value以及NullWritable.get即可；

步骤2：使用线性回归采用随机梯度下降的方法来更新参数，每个Mapper都会使用同样的初始化参数，并且在每个Mapper中使用自己的数据来更新参数；然后在每个Mapper的cleanup函数中直接输出参数值即可，不需要用到Reducer;

步骤3： 合并参数值，步骤2得到的是局部数据的参数值，需要计算全局的参数值；可以采用各个参数值的全局误差作为参数来进行加权。所以，在Mapper的setup方法中会读取步骤2中的多个输出参数值，在map函数中针对各个原始数据求其误差，输出到reducer的数据为参数值和其误差的键值对；在Reducer端，直接针对每个键（也就是参数值）把各个误差加起来，在cleanup函数中采用加权来合并参数值；

1. 课题分工

林喜鹏：负责小组分工、撰写报告、数据预处理及拟解决问题4）

李朝阳：负责实现拟解决问题3）

唐玉婷：负责实现拟解决问题2）

饶璐：负责实现拟解决问题1）

1. 参考文献
2. YouTube热门视频数据集及其简介：

<https://www.kaggle.com/datasnaek/youtube-new>

1. Jython: <http://www.jython.org/index.html>
2. TextBlob: <https://textblob.readthedocs.io/en/dev/>