计算机科学与技术学院 大数据管理与分析 课程实报告

|  |
| --- |
| 实验目的：  随着Hadoop与Spark产生的影响越来越深，各种基于Hadoop与Spark平台的数据分析系统也随之出现。本次实验要求各位同学利用之前实验以及所学知识实现一个基于Hadoop、Spark或其他大数据平台的数据分析系统，理解其中的实现细节以及各种算法的原理。 |
| 实验软件和硬件环境：  软件:  1.系统:Ubuntu 18  2.编程环境:Java  硬件:  Hp Probook 430 G2 |
| 实验原理和方法：  1.数据集简单描述。数据集主要包含以下文件:    Rating打开如下:    第一列表示用数字标识的用户唯一标识。第二列表示电影的用数字的唯一标识。第三列代表用户对电影的评分，第四列代表用户评分的时间，代码中没有用到，其实评分时间越近确实一定程度上更能表示用户近段时间喜欢的电影。  Movies中截图如下:    就是电影的数字标识和对应的电影名字以及电影的种类。  另外的几个文件基本没有用到，就不一一叙述。  其中分数是最小值0，最大值5，步长0.5的小数。  2.概述。总共实现三个小算法，最主要的算法只有一个。  第一个小算法是根据用户选择的十部自己喜欢的电影进行推荐。本来准备根据用户选择十部电影之后，用频繁项集的办法进行电影推荐，因为刚学完这个算法，想练练手，但是后来发觉这里用频繁项集不是很合适，因为频繁项集本身是针对目标未知。具体说就是这里是已经知道用户喜欢具体哪十部电影，而频繁项集更好的是在于各个项目之间没有约定好目标，只是单纯想找到所有项目中尽量项目公共个数最多的集合。 这里具体采用的办法就是用Mapreduce函数将用户聚集起来，然后判断用户聚集起来的函数当中，哪个本地用户和当前推荐用户中喜欢的电影公共个数最多的，喜欢的电影目前在代码里面是用 >= 3.5分来评判的。  第二个就是主要的算法了。首先说明一下，代码中涉及到的推荐算法都是基于用户协同的算法，直接针对电影本身的推荐算法暂时只想到针对标签直接推荐，这个比较简单就没有实现。另外没有针对电影本身进行推荐是因为确实不同电影之间不好进行推荐，不靠谱。 第二个主要的算法主要思想是让用户对十部电影进行打分，然后对用户未打分电影进行预测评分，然后对用户未评分电影进行比分从高到低进行排序，然后选取前十部分数最高的电影进行评分。  预测评分也分成几部分。首先是定义推荐用户和本地用户之间的距离。这里的距离是欧式距离。首先找到本地用户中也对推荐用户选中电影打过分的，然后利用分数差计算的平方除以总共公共电影的数目作为推荐用户和其中一个用户之间距离的定义，如果某个本地用户没有对推荐用户打分电影中的任何一部进行评分，则这两个用户之间的距离为默认值，也就是最大值。最大值的计算就是用分数相差最大的原则算出来的距离:作为初始值。  然后是对每步电影真正进行预测评分阶段。评分的方式是加权评分，用距离的倒数作为权值，因为如果两个用户之间的距离越大，则该用户的评分应该越不能作为这个用户的评分，即权值就需要小一点。最后将所有打分电影进行从高到低进行排序，然后排序的结果选取从高到底的方式进行推荐。  第三个小算法其实是针对第二种推荐算法的一种检测。因为第二个主要的算法算出来的结果并不知道好坏，而且自己也没看过这些电影，推荐的结果就不知道到底好不好，所以不管怎么也还是得需要一个用来大致检测算法可靠性的指标或者过程。 主要思想就是，之前是针对新来用户进行了推荐，现在用本地用户，比如本地用户有100部打分电影，我选取其中的前10条进行输入，预测评分完毕之后计算这100部电影之间打分的误差，误差的计算方式就是用预测评分减去实际评分，取绝对值，然后除以总数。算出来的数字基本可以代表打分的误差范围。 |
| 实验步骤：（不要求罗列完整源代码）  1.实验需要写界面，用的是Eclipse的SWT工具。主界面如下所示:  C:\Users\50264\Desktop\exp\pics\Screenshot from 2019-06-12 14-32-12.png  直接显示的就是有三种算法，然后需要对哪种算法进行展示就点击哪种算法。点击推荐算法1之后显示界面:  C:\Users\50264\Desktop\exp\pics\Screenshot from 2019-06-12 14-32-20.png  上面的8个按钮上，每一个都有一个对应的电影名称，这个电影名称的显示是通过随机出几部电影来显示的。    通过设置里面的current\_moviename(String 类型)来进行设置当前可以显示在Button上面的电影名字，之后在按钮初始化的时候将电影名字显示出来。    每个按钮基本结构都是这样，按钮显示的是current\_moviename，然后按钮被点击之后将对应的movieid记录下来，放到一个数组里面，以供之后的Mapreduce程序执行。  2.点击完毕之后进行推荐，推荐部分的核心代码如下:      Map部分就是直接将用户id作为key聚集，然后values中进行计数，用Vector对象存储当前公共喜欢电影最多的用户id，如果有更多公共喜欢电影的用户，就将当前的Vector清空，然后将这个新的用户id放入Vector中，如果是和当前最大个数一样，就将当前的用户id放入Vector中。  3.推荐界面。后面几个算法的推荐界面基本都是这样。  C:\Users\50264\Desktop\exp\pics\Screenshot from 2019-06-12 14-32-46.png  主要就是在一个空窗口中放置一个可以显示推荐电影名称的textviewer，然后最下方有一个Button用于返回主菜单，然后继续用别的推荐算法进行推荐。  4.第二个主要的推荐算法。第二个主要的推荐算法直接点开界面如下:  C:\Users\50264\Desktop\exp\pics\Screenshot from 2019-06-12 14-32-55.png  这个界面基本是和第一个小算法的界面一样，但是因为是打分，所以点击对应按钮之后有一个打分的界面:  C:\Users\50264\Desktop\exp\pics\Screenshot from 2019-06-12 14-33-03.png  对自己选定的十部电影进行评分完毕之后就可以进行后续的推荐算法。最后推荐完毕之后:  C:\Users\50264\Desktop\exp\pics\Screenshot from 2019-06-12 14-34-02.png  5.主要推荐代码分析。在主窗口和第一个小算法基本一样，但是和第一个小算法的不同之处在于，这个算法还需要记录分数，多一个数据结构。    在Map的Setup阶段初始化数组，其中定义用户之间距离的数组设置初始值为25，最上面一个判断shell\_num是哪个然后进行赋初始值，是为了减少mapreduce函数的冗余代码的编写。在图形界面对应的代码中，通过上述的变量可以判断出到底是哪个窗口跳转到该窗口，然后赋予不同的初始值。      Map阶段主要就是将用户id设置成key，然后再将values设置成对应的电影id和对对应电影的评分。然后Reduce阶段就负责进行计算欧式距离，找到本地用户和推荐用户公共评分的电影，然后进行评分。  之后一个Mapreduce算法就是对电影进行预测评分。      这个里面就是采用用距离的倒数进行加权评分，并且必须要评价电影大于等于三部，这个可以进行调整。  然后最后还需要调用一个sort的mapreduce算法，将最后的结果进行从高分到低分排序，主要代码如下:    这个部分的代码主要是用来将默认的排序算法修改成从高分到低分进行排序。  6.最后一个测评算法。该算法的基本思想在方法中已经叙述了，只要直接打开的界面；  C:\Users\50264\Desktop\exp\pics\Screenshot from 2019-06-12 14-34-36.png  选取特定的一个人之后进行打分。和第二个算算法中不用的部分在于：      Setup部分主要是将一些数值进行初始化，目的在于可以重复运行。 Map传入的是两个部分，一个是前面的预测评分，还有一个输入就是最原始的输入，这两个输入通过一行用逗号分开之后的元素个数不同区分开来，然后将元素相同的合并。 Reduce之中就直接算出两个的差值，最后显示出来。  最后显示：  C:\Users\50264\Desktop\exp\pics\Screenshot from 2019-06-12 14-34-51.png  C:\Users\50264\Desktop\exp\pics\Screenshot from 2019-06-12 14-35-07.png  有的数据误差会比较大，有的数据误差会比较小，不是很稳定。 |
| 结论分析与体会：  1.改进部分。感觉还可以改进的部分就是第一个算法换别的思路，或者是可以加一下什么算法来检测一下最后的结果。 并且可以不仅仅只针对初来的用户，还可以通过引入数据库来管理用户信息，这样基于历史可以产生更多的推荐算法。  2.界面问题。本次实验直接使用的是SWT的工具来实现的界面，界面不是很客观，如果可以改进的话，还可以对系统的界面进行美化，或者是换一种界面实现的工具。  3.实验虽然Mapredcue本身写的内容比较简单，但是写多了之后对Mapreduce的整个流程还是更加熟悉。 |
| 就实验过程中遇到和出现的问题，你是如何解决和处理的，自拟1－3道问答题：  1.Mapredcue存在一些冗余代码，怎么实现。由于一些参数的不同而导致了Mapredcue方法需要微调，这时候可以巧妙利用Setup方法，设置窗口参数，通过判断是哪个窗口跳转过来的，临时设置参数。 |