CS6350: BIG DATA ANALYTICS and MANAGEMENT

Spring 2015

HW #4

Related to: Spark, Data Analytics and Recommendation System

Due: April 22, 2015

This homework consists of two parts. The first part focuses on K-means clustering (data analytics) and the second one focuses on recommendation systems.

Q1.

Implement the k-means algorithm from the scratch using SCALA and spark. Please use this attached dataset in file **Q1_testkmean.txt** as input.

Your number of cluster K should be 3.

Your Scala code will produce output in the following ways:

- Print each point and the corresponding cluster it belongs to.
- Print the final centroids

Q2. Read the following link for co-occurrence based recommendation implementing in mahout.

https://mahout.apache.org/users/recommender/intro-cooccurrence-spark.html

Currently Mahout switches from MapReduce to Apache Spark. It has an interactive shell (will show in the class, lecture contains how to install it). Using that, apply item-based collaborative filtering using mahout's ***spark-***

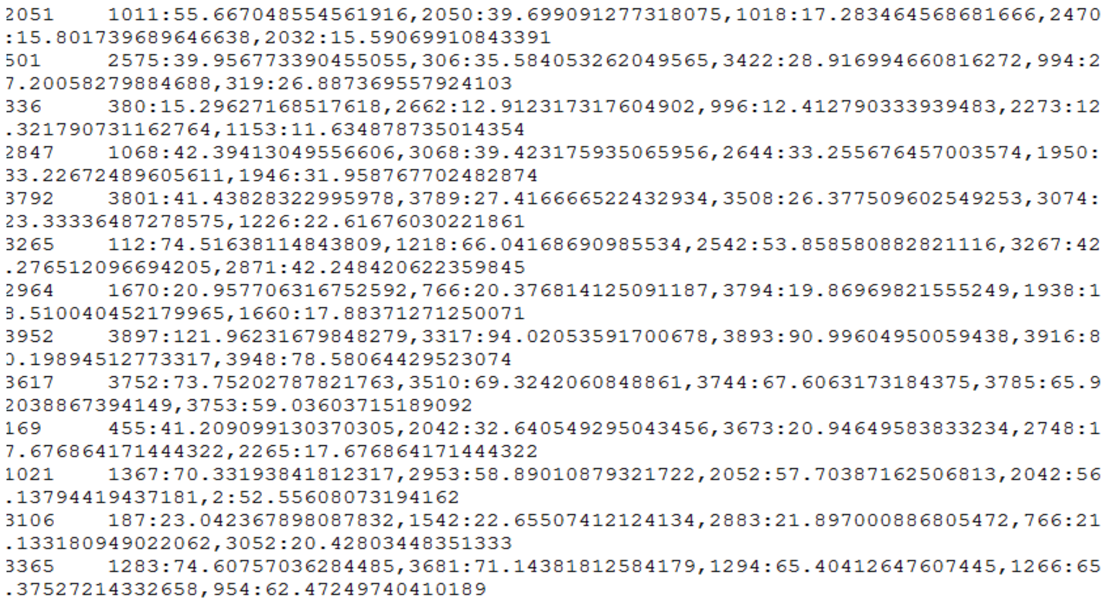
itemsimilarity. ***spark-itemsimilarity*** can be used to create "***other people also liked these things***" type recommendations.

You can find the dataset in ***elearning***. Copy the data into your hadoop cluster and use it as input data. You can use the put or ***copyFromLocal*** HDFS shell command to copy those files into your HDFS directory. There are 3 data files: ***movies.dat***, ***ratings.dat***, ***users.dat.*** Please read the "***README_Important***" file to know about the data organization and to know about the Attribute of the data. All are very well explained in that ***README_Important*** file.

"A user rates some movies with rating 3. Our task is to recommend some movies to him that has the similar ratings from other users."

Steps to follow:

Read the above link carefully and construct the item-similarity matrix of each movie having rating 3 (use ratings.dat). The output should be like this:



```
2051    1011:55.667048554561916,2050:39.699091277318075,1018:17.283464568681666,2470
:15.801739689646638,2032:15.59069910843391
501     2575:39.956773390455055,306:35.584053262049565,3422:28.916994660816272,994:2
7.20058279884688,319:26.887369557924103
336     380:15.29627168517618,2662:12.912317317604902,996:12.412790333939483,2273:12
.321790731162764,1153:11.634878735014354
2847    1068:42.39413049556606,3068:39.423175935065956,2644:33.255676457003574,1950:
33.22672489605611,1946:31.958767702482874
3792    3801:41.43828322995978,3789:27.416666522432934,3508:26.377509602549253,3074:
23.33336487278575,1226:22.61676030221861
3265    112:74.51638114843809,1218:66.04168690985534,2542:53.858580882821116,3267:42
.276512096694205,2871:42.248420622359845
2964    1670:20.957706316752592,766:20.376814125091187,3794:19.86969821555249,1938:1
8.510040452179965,1660:17.88371271250071
3952    3897:121.96231679848279,3317:94.02053591700678,3893:90.99604950059438,3916:8
0.19894512773317,3948:78.58064429523074
3617    3752:73.75202787821763,3510:69.3242060848861,3744:67.6063173184375,3785:65.9
2038867394149,3753:59.03603715189092
169     455:41.209099130370305,2042:32.640549295043456,3673:20.94649583833234,2748:1
7.676864171444322,2265:17.676864171444322
1021    1367:70.33193841812317,2953:58.89010879321722,2052:57.70387162506813,2042:56
.13794419437181,2:52.55608073194162
3106    187:23.042367898087832,1542:22.65507412124134,2883:21.897000886805472,766:21
.133180949022062,3052:20.42803448351333
3365    1283:74.60757036284485,3681:71.14381812584179,1294:65.40412647607445,1266:65
.37527214332658,954:62.47249740410189
```

In the above matrix, the first integer is the movie id (The movie for which we recommend), then the rest of the text contains the list of the recommended movies id with their value (movie id: value)

1. Save the above file to HDFS. Now, Run Apache spark interactive shell. From the shell, take the user id as input (you can fix the id, e.g., val userID = 20). Now find all the movies that he rates with rating 3.

2. Load/read the above file (item-similarity file) and find the movies that match with the user’s rated movies with the key of the item-similarity file.
   For example, suppose a user has id 20 and he rates movies 120 and 855 as 3.

   Write the code to extract the movie ids from item-similarity matrix file that are present in the row for 955 and 123 movies and generate the matrix like following:

   ```
   120     898,951,910,905,1269
   855     3265,1218,1089,3224,247
   ```

3. Now replace the movie Id with movieid:movie_name.
   For example,
   120:<Movie_Name>

   898:<Movie_Name>,951:<Movie_Name>,910:<Movie_Name>,
   905:<Movie_Name>, 1269:<Movie_Name>

   855:<Movie_Name>
   3265:<Movie_Name>,1218:<Movie_Name>,1089:<Movie_Name>,
   3224:<Movie_Name>, 247:<Movie_Name>

   You can apply join if it is necessary. (Use movies.dat and ratings.dat)
   Note: In, 120:<Movie_Name>
   <Movie_Name> should be replaced with movie id 120. Display without angle brackets.

Submission:

You have to upload your submission via e-learning before due date. Please upload the following to eLearning:

1. A scripting file like, Q2_1.txt that shows the building of spark-itemsimilarity and another scripting file Q2_2.txt shows the scala/java program (contains codes for step 1 - 3).
   If you use java/scala, then submit all source files.