SON Implementation using Apriori Algorithm

Implemented SON algorithm using two phase Map Reduce Technique.

Map Phase 1

a) Joined (user ID,movie ID) from ratings.dat and (userID,Gender) from users.dat in order to get one user's gender/movie ID. Used sparks' functionalities to filter by gender and created baskets
b) Case 1 baskets contain a list of lists – All the movie ID's rated by each unique male user
c) Case2 baskets contain a list of lists – All the female users who rated one unique movie.
d) Ran Apriori Algorithm on these baskets using map partition in order to obtain the frequent singletons/doubles/triples in each partition. Apriori algorithm receives a chunk of the entire list of baskets to process and find frequent sets in each partition.
e) The support threshold is also equally partitioned using ratios. The formula used is (Number of baskets in each partition/Total number of baskets)*Given support threshold
f) Output is (each frequent item,1)

Reduce Phase 1

a) Produces candidate sets, which basically does reduceByKey for those frequent items that appear one or more times

Map Phase 2

a) Map phase two uses broadcast variables in order to increase speed for phase 2 of SON
b) We check if each value in the candidate set is present in the entire list of baskets i.e In program I check if the basket is a super set of each frequent item, if yes we increment count. I return (C,v) where C is candidate set and v is count value

Reduce Phase 2

a) If the count value is less than the given support threshold value then we filter the candidate sets
b) We only account for those candidate sets whose value is greater than the support threshold. They are the frequent sets of the given case = output

Commands Used to Run Program:

Input file names: ratings.dat users.dat
Source Code file name: Vishnupriya_Ravibalan_SON.py
Output file name: Vishnupriya_Ravibalan_SON.case1_1200.txt,
Vishnupriya_Ravibalan_SON.case1_1300.txt,
Vishnupriya_Ravibalan_SON.case2_500.txt,
Vishnupriya_Ravibalan_SON.case2_600.txt

How to run?

1. Move the input files and source code file inside the spark-1.6.1-bin-hadoop2.4 folder in your machine
2. In terminal enter the same spark-1.6.1-bin-hadoop2.4 directory and run the following command.

Command to enter directory -->

```
cd spark-1.6.1-bin-hadoop2.4
```

Command used to run source code -->

./bin/spark-submit Vishnupriya_Ravibalan_SON.py 1 ratings.dat users.dat 1200 Vishnupriya_Ravibalan_SON.case1_1200.txt

./bin/spark-submit Vishnupriya_Ravibalan_SON.py 1 ratings.dat users.dat 1300 Vishnupriya_Ravibalan_SON.case1_1300.txt

./bin/spark-submit Vishnupriya_Ravibalan_SON.py 2 ratings.dat users.dat 500 Vishnupriya_Ravibalan_SON.case2_500.txt

./bin/spark-submit Vishnupriya_Ravibalan_SON.py 2 ratings.dat users.dat 600 Vishnupriya_Ravibalan_SON.case2_600.txt