INF553
Bufan Zeng
2/25/18

Description

The *Bufan_Zeng_SON.jar* file is a SON algorithm to find the frequent items in the provided csv files. The main class is *Bufan_Zeng_SON*.

The algorithm firstly read in the csv file to a RDD and then use groupByKey method to get all the baskets. The keys and baskets are decided by the *case number* where if the number is 1, the baskets contain the productID; if the number is 2, the baskets contain the reviewerID. Then use the mapPartitions function to divide the baskets into segments. For each segment, run the apriori function to find the candidate frequent item sets with the support equals to the support divided by the number of partitions. Then using reduceByKey method to remove the duplicated item sets between different partitions.

After the first MapReduce, the algorithm gets a list of candidate frequent sets. Then run through each segment again and using a HashMap to store the count of those candidate frequent sets. The last reduce procedure sums the count from all the segments and filter out the real frequent items. The last step is to write the frequent items to a text file named result.txt.

Command to run the algorithm (under spark/bin directory):
./spark-submit --class Bufan_Zeng_SON Bufan_Zeng_SON.jar [case number] [path/to/input file] [support]

For example, command to run small1.csv with case 1 and support 4:
./spark-submit --class Bufan_Zeng_SON Bufan_Zeng_SON.jar 1 small.csv 4

Can also try to declare the memory and cores to run the algorithm, which may have better computing power:
./spark-submit --class Bufan_Zeng_SON --executor-memory 1g --driver-memory 2g --executor-cores 2g Bufan_Zeng_Son.jar [case number] [path/to/input file] [support]

The runtimes of the files are in the following table:

File Name	Case Number	Support	Runtime (sec)
beauty.csv	1	50	239
beauty.csv	2	40	53
books.csv	1	1200	338
books.csv	2	1500	59