1. How to implement algorithm

(1) In the first map of SON algorithm, I use Apriori algorithm. In Apriori algorithm, I use mapPartition to slice data to chunks and then for each chunk, I generate all possible combinations of this chunk, and calculate the frequent combinations in this chunk depend on customize support of this chunk. In the end, we will reduce all map's results.

(2) In second map of SON algorithm, I split data file to several chunks, and then I take result of first map-reduce to each chunk, to count the appearance of each possible candidates. If the candidate (one of first map-reduce result) is larger than support, it will be considered as frequent set.

2. How to run my code

```
./bin/spark-submit --class task Jiayue_Shi_SON.jar 1 small2.csv 3

./bin/spark-submit --class task Jiayue_Shi_SON.jar 2 small2.csv 5

./bin/spark-submit --class task Jiayue_Shi_SON.jar 1 beauty.csv 50

./bin/spark-submit --class task Jiayue_Shi_SON.jar 2 beauty.csv 40

./bin/spark-submit --class task Jiayue_Shi_SON.jar 1 books.csv 1200

./bin/spark-submit --class task Jiayue_Shi_SON.jar 2 books.csv 1500
```

3. Performance Table

File Name	Case Number	Support	Runtime (sec)
beauty.csv	1	50	656
beauty.csv	2	40	170
books.csv	1	1200	1400
books.csv	2	1500	100.6

4. Versions

Spark version: 2.2.1
Scala version: 2.11
Hadoop version: 2.7