CS 5683: Big Data Analytics

Project-1: Distributed Market Basket Analysis

Total Points: 50; Due date: Sept. 15, 2020

Association rules are frequently used for Market Basket Analysis by retailers to understand the purchase behavior of their customers. This information can then be used for many different purposes such as cross-selling and up-selling of products, sales promotions, loyalty programs, store design, discount plans, and many other.

Application in product recommendations: This action or practice of selling additional products or services to existing customers is called *cross-selling*. Giving product recommendation is one of the examples of cross-selling that are frequently used by online retailers. One simple method to give product recommendations is to recommend products that are frequently browsed together by the customers.

Suppose we want to recommend new products to the customers based on the products they have already browsed online. Write *A-Priori* algorithm using a Spark program to find products which are frequently browsed together. Fix the support to *s = 85* (i.e. itemsets need to occur at least 85 times to be considered frequent) and find itemsets of size 2, 3, and 4. With frequent itemsets, generate association rules that has only one items on the right hand-side of the rule (i.e) {X} → {Y} for frequent 2-itemset, {X,Y} → Z, {X,Z} → {Y}, and {Y, Z} → {X} for frequent 3-itemset and so on and so forth. Generate rules with confidence threshold *c = 90%.* Sort all association rules by decreasing order of confidence.

Use the online browsing behavior dataset from *browsing.txt* given with this instruction. Each line in the data represents a browsing session of a customer. On each line, each string of 8 characters represents the ID of an item browsed during that session. The items are separated by spaces.

Tips for your Spark program:

1. One Spark job to find both frequent itemsets and association rules
2. All computations should be done only with RDDs. If you collect() results and do computations, you are not utilizing Spark at all
3. Make use of the broadcast() wisely. Broadcast is very essential for the A-Priori algorithm
4. Except for frequent item mining (first pass of A-Priori), other frequent itemset mining should be in a recursive function or a ‘for’ loop. DO NOT WRITE 3 functions to generate frequent 2-itemsets, frequent 3-itemsets, and frequent 4-itemsets
5. Your Spark program should be memory efficient. Follow the A-Priori steps as it is. There are other memory inefficient programs available in web
6. Make a wise decision to choose whether to persist and write the intermediate results (frequent itemsets, for example)

Expected Output Format:

The Spark program should generate only one output file – association rules. If you write any intermediate outputs, your program should delete them before its termination.

ID1,ID5 → ID2; Confidece=98.45%

ID234,ID712,ID1 → ID193; Confidence=94.2%

(NOTE: ID*** should be replaced with item names from the input)

Grading Rubric:

Read data and generate frequent items: 10 points

Recursive/For-loop logic: 5 points

Candidate itemset generation: 5 points

Frequent itemset generation: 5 points

Handling intermediate results: 5 points

Association rules generation: 10 points

Efficient programming: 5 points

Program documentation and output: 5 points

What to submit in Canvas?

Submit only your complete python notebook (.ipynb) file. We will use Google Colab to test and grade your work. Type your name on the first cell of the notebook. Add sufficient documentions in your program – the reader should understand what your logic tries to achieve from the documentation.

Simple reminder: You can discuss the logic and implementation details with your friends. But, you should write your own program and documentation. *We will not grade your work if we suspect cheating.* Give references wherever necessary, if you are using a logic from any internet sources.