Advanced Data Engineering: Assignment 3

NGUYEN T. Hoang - SID: 15M54097

Fall 2015, W831 Tue. Period 5-6

Due date: 2015/11/10

Problem

In this assignment, we assume that the cardinality of the large 1-itemset is N for the Apriori Algorithm.

Question 1

How many combinations are there as the candidate 2-itemset?

Answer: The candidate 2-itemset is generated from the large 1-itemset with cardinality of size N. Therefore, the size of candidate 2-itemset is the combination of 2 elements out of N elements in large 1-itemset.

$$|Candidate\ 2itemset| = \binom{N}{2} = \frac{N \times (N-1)}{2}$$

Question 2

How many times should the fact table be scanned to derive the large 2-itemset?

Answer: Suppose we already have the candidate of size $\frac{N \times (N-1)}{2}$. At this step, the Apriori Algorithm will scan the fact table k time to count support of each candidate. The number of times that the fact table is scanned to derive the large 2-itemset from the candidate 2-itemset is:

$$\#Fact\ table\ scan = 2 \times \frac{N \times (N-1)}{2}$$

Question 3

Discuss the effect of minimum support value for the cost to derive association rules.

Answer: In the Apriori Algorithm, minimum support plays a role as a prunning parameter to reduce the size of large K-itemset from the candidate K-itemset. Furthermore, minimum support and confidence tell us about the value of the newly discovered association rules. In my answer, I will consider different size of minimum support relative to the size of a dataset in general and also present some experimental data with a toy dataset named *retail.dat* [1].

In our Apriori Algorithm, the Candidate Generation process uses brute-force method, which consider every large k-itemset as a potential candidate for k+1-itemset. Therefore, the Candidate Prunning process is extremely expensive due to the large size of data generated, especially the

candidate 2-itemset since its size is $\binom{N}{2}$. Suppose $O(k)$ is the computational time for each candidate, the time required for each step is $O(k \times \binom{N}{k})$, with N is the total number of large k-item. Besides, when there is a large amount of candidate items, the storage space also becomes a major problem.

Small minimum support value will increase the amount of frequent item found in the dataset, which increase value of N for the next pass of the Apriori Algorithm. As mentioned above, the increase of frequent item number will lead to increase in computional cost and storage space of candidate itemset as N tends to be large. Although there are many association rules will be generated for a small minimum support value, but the most of these rules will not have valuable information. With a dataset of 88,163 transactions, 16470 items [1] and minSupport = 0.1%, my computer with 2.53 GHz Intel Dual 2 Core Processor takes more than 3 hours to derive 2785 frequent 2-itemsets.

Large minimum support value will drastically decrease the amount of frequent item found in the dataset. As a consequence, the computational cost and space cost is also drastically reduced. With the same dataset mentioned above and minSupport = 50%, my computer takes 11.596 seconds to get 1 frequent itemsets of size 1 and 0 frequent itemsets of size 2. The similar result is obtained with the minSupport range of 20% and above. Here, we can see that although the computational cost is reduced, but we cannot find any more than size 3 frequent itemsets.

Choosing the appropriate minimum support value is essential in data mining. As the computational cost drastically increase in the case of small minimum supoprt, decrease in the case of large minimum support, we have to consider trade-off between amount of association rules obtained and the quality of the association rules. In the case of the toy dataset I have choosen, I think the appropriate minimum support value is 0.025 (2.5%). With this value, the computational cost is still low (15.704 seconds) and we have 14 frequent 1-itemsets, 18 frequent 2-itemsets, 6 frequent 3-itemsets, and 0 frequent 4-itemsets.

References

[1] Tom Brijs and Gilbert Swinnen and Koen Vanhoof and Geert Wets, *Using Association Rules for Product Assortment Decisions: A Case Study*, Knowledge Discovery and Data Mining, 254-260, 1999.