CS412 Assignment-3

Yuetong Liu
Department of Statistics

November 4, 2016

1 Frequent Pattern Mining

In this section, frequent patterns are extracted from 5 topic files. I used Apriori algorithm to mine the frequent patterns, finding frequent itemsets by confined candidate generation.

1.1 Implementation

Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994[1]. Apriori employs an iterative approach known as a levelwise search, where the result of k-itemsets are used to explore $(k+1)$-itemsets. First, the set of frequent 1-itemsets is found by scanning the topic files to accumulate the count for each term, and collecting those terms that satisfy minimum support. The result set is denoted by L_1. Next, L_1 is used to find L_2, the set of frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent k-itemsets can be found. The finding of each L_k requires one full scan of the files.

Because the Apriori property, which is all nonempty subsets of a frequent itemset must also be frequent. We can improve the efficiency when generating the candidates for k-itemset L_k. In order to use this anti-monotonicity property to reduce search space, two steps are applied, consisting **join** and **prune** actions. For instance, if we are going to use L_{k-1} to find L_k for $k \geq 2$, these following two procedures are applied:

1. **Join step.** To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} with itself. This set of candidates is denoted C_k. To get C_k, I traversed $k-1$-itemsets L_{k-1}, in which each pattern has the length $k-1$. Let notate $l_i[j]$ refers to the jth item in l_i. First, I sorted L_{k-1} by their term id, such that $l_i[1] < l_i[2] < \cdots < l_i[k-1]$. Then I traversed L_{k-1}. If two members share the first $k-2$ items, they will be joined. That is to say, members l_1, l_2 are joined if $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge (l_1[3] = l_2[3]) \wedge \cdots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$. The resulting itemset formed by joining l_1 and l_2 is $\{l_1[1], l_1[2], \cdots l_1[k-2], l_1[k-1], l_2[k-1]\}$.

2. **Prune step.** To reduce the size of candidates, the Apriori property is used as follows. Any $k-1$-itemset that is not frequent cannot be a subset of a frequent k-itemset. Hence, if any $k-1$-subset of a candidate k-itemset is not in L_{k-1}, then the candidate

cannot be frequent either and so it can be removed from C_k. Then I traversed the candidates from join step, to check if there is any $k-1$-subset of a candidate is not in L_{k-1}.

1.2 Question to ponder A

As for the min_sup, I chose it by considering the length of topic files and the volume of frequent patterns. The length of 5 topic files are 10520, 9931, 9362, 10671, 9644 respectively. First I tried min_sup as 0.01, which means the frequency of patterns must satisfy 105, 99, 93, 106, 96 in corresponding files. It turns out the numbers of frequent patterns in each files are 84, 77, 75, 70, 59. The results seem pretty reasonable for further analysis and explanation. But if we raise the min_sup to 0.02, the volume of corresponding frequent patterns will become 30, 30, 23, 24, 19. That's a big gap between results of 0.01 and of 0.02. Less than 30 patterns might be insufficient for topic files with more than 9000 lines.

2 Maximal and Closed Patterns Mining

2.1 Implementation

Based on the output of frequent patterns mining, maximal and closed patterns are mined in this section. A max-pattern X is a frequent pattern that if there exists no frequent that is a superset $Y \supset X$. A pattern X is closed if X is frequent, and there exists no super-pattern $Y \supset X$, with the same support as X.

To get the maximal patterns, I used the output of patterns in 5 topics. According to the definition, iterated through all the patterns in each topic, and output it to the maximal patterns if it is not subset of any frequent patterns in this topic. Likewise, I found all the closed patterns as it is defined. Traversed through all the patterns, if there exists no super frequent pattern shared the same support, then output it.

2.2 Question to ponder B

After mapping the ids back to terms, I make a conjecture about the topics and corresponding domains:

- topic-0: Information Retrieval(IR)
- topic-1: Data Mining(DM)
- topic-2: Database(DB)
- topic-3: Machine Learning(ML)
- topic-4: Theory(TH)

This conclusion is drawn based on the observations with frequent patterns, maximal patterns and closed patterns in all topics. Like, the third frequent maximal pattern in topic-0 is "information retrieval", which means these two words have pretty large chance appeared in paper titles together. Besides, there is no another two words can show together at this frequency in this topic. Thus, we may conclude that this topic is about Information Retrieval(IR). Topic-1 has most frequent 1-itemset "mining", "data", "pattern", "tree", "rule" that basic elements in Data Mining(DM) domain. Also, obviously, "query", "database", "distributed", "system", "optimization" in topic-2 are under the domain of Database(DB).

2.3 Question to ponder C

Compare the result of frequent patterns, maximal patterns and closed patterns, I found some interesting points:

- The number of maximal patterns are closer to the number of frequent patterns in each topic. This is reasonable considering there are many 1-itemset pattern and few 2-itemset and even no 3 or larger itemsets in all 5 topic files. For example, there are 69 1-itemsets in topic-1 but only 8 2-itemsets. In reality, there won't be too many words in paper titles, thus there won't be many frequent long patterns.
- The closed patterns are totally identical to the frequent patterns. Likewise, since there are not many frequent 2-itemset patterns, there is not much chance that there are supersets shared the same support. This also can be easily explained by reality meaning, that is the terms in specific area .

3 Re-rank by Purity of Patterns

Purity[2] is introduced as one of the ranking measures for phrases. A phrase is pure in topic t if it is only frequent in documents about topic t and not frequent in documents about other topics. For example, 'query processing' is a more pure phrase than 'query' in the Database topic. The purity is measured by comparing the probability of seeing a phrase in the topic-t collection $D(t)$ and the probability of seeing it in any other topic-t' collection $(t' = 1, \cdots, k, t'! = t)$.

According to the definition, purity is measured by this formula:

$$purity(p)_t = log\frac{f_t(p)}{\mid D_t \mid} - log \max_{t'}\frac{f_t(p) + f_{t'}(p)}{\mid D_{t,t'} \mid}$$

When implemented, I looked into every frequent patterns in 5 topics and computed their purity. With respect to each frequency, using brute force to iterate topic files and count their support. Then looked through all other 4 topics and chose the maximum second term.

In the output files, frequent patterns are sorted from high to low first by **Purity** then by **Support**. Because purity indicates how much this term is exclusive under certain topic, support shows how frequent this term is, thus the terms come head of output are the terms

more exclusive and frequent under this topic.

The purity ranges from $(-2.7, 0.7)$. According to the definition, the second term can be regarded as the the maximum of average relative support among two topics. Hence, the purity is the base-2 logarithm of ratio between relative support in this topic and maximum of average relative support among two topics. If purity is greater than 0, which means the ratio is greater than 1, this term is more exclusive in this topic than in the maximum union topics.

Looking into the result after mapping back to phrases, we can verify the purity of terms in each topic. For instance, in topic-1 Data Mining(DM), the top purity terms are "rule association", "mining association", "mining frequent", with purity more than 0.70. Obviously, these are the terms only frequent in this Data Mining topic, but not other domains. Likewise, terms "query", "join", "query optimization" are exclusively top-purity terms in Database domain, with purity above 0.7. On the other hand, terms "using", "time", "system", "approach" have low purity less than -1.2, which indicates they can be frequent in many topics, not just in the Database domain.

4 Source File Organization

- Step 2: Preprocessing
  - **s2_2Preprocessing.py** create title.txt
- Step 3: Partitioning
  - **s3_2Reorganizing.py** create topic-0.txt – topic-4.txt
- Step 4: Mining Frequent Patterns for Each Topic
  - **s4MiningFreqPatterns.py** create pattern-0.txt – pattern-4.txt
- Step 5: Mining Maximal/Closed Patterns
  - **s5MaxClosedPatterns.py** create max and closed pattern files
- Step 6: Re-rank by Purity of Patterns
  - **s6Purity.py** create purity-0.txt – purity-4.txt

References

[1] R. Agrawal and R. Srikant. "Fast algorithms for mining association rules". In: *Very Large Data Bases (VLDB'94). Proc. 1994 Int. Conf.* Santiago, Chile, Sept. 1994, 487–499.

[2] N. Desai J. Guo M. Danilevsky C. Wang and J. Han. "KERT: Automatic Extraction and Ranking of Topical Keyphrases from Content-Representative Document Titles". In: (Jun 2013).