CS 412 Data Mining Assignment 3 report

Explanation for step4~6

Step 4:

- I read data from topic files and count the number of lines (number of titles)
- I combination terms from 1 item set to k items set (which k is the number of words in this line) line by line and store them in a dictionary as key.
- Meanwhile, when I add sets into dictionary, I count how many times them show up and divide the number by the number of lines (from step 1). Then add them into dictionary as key.
- For now, I have the whole patterns as keys and their supports as values in dictionary. Then I add values (support number) into an array if it is greater than min support.
- After that, I sort this array and remove duplicates.
- I write the keys and values of that dictionary into output file according to the order of values in array.

Step 5:
Max:

- I read data from pattern files and convert them into set.
- Then, I add these sets into a list.
- I test each set whether it is a subset of other elements in that list. If it is a subset, I remove it from list.
- Finally, I write elements of the list into output file by the order they occur from pattern files since pattern files is a sorted file.

Closed:

- I read data from pattern files and convert them into set.
- Then, I add these sets into a list.
- I test each set whether it is a subset of other elements in that list and whether it has the same support as their superset. If it is a subset and has same support, I remove it from list.
- Finally, I write elements of the list into output file by the order they occur from pattern files since pattern files is a sorted file.

Step 6:

- I read data from pattern file, counting the number of lines and store data into a dictionary which key is pattern and value is their support times the number of lines. So, we have f(t,p) and D(t).
- I go through all topic files except the one with same order number as current pattern file to test whether current pattern shows up in other topic.
- If it occurs in other topic, I count the support of this pattern and union two topic files. So, we have f(t',p) and D(t,t').
- After that, I add results of f(t,p) + f(t',p)) / | D(t,t') | of all matched files into a list. Then I use max function to get their maximum.
- For now, we get all elements we need to generate purity. I calculate purity of each patterns and add them into a dictionary which key is pattern and value is purity.

- Finally, I write purity and pattern into output file according to the order of their combination purity and support. I combine purity and support by purity*0.1+support*0.9. I think purity is a helper value as TF-IDF but it is useful for the pattern only shows up in one file. And I test it many times and find that 0.1 is an appropriate weight for purity.

Answers of ponder

ponder A: How you choose min_sup for this task? Note that we prefer min_sup to be the consistent percentage (e.g. 0.05 / 5%) w.r.t. number of lines in topic files to cope with various-length topic files.

I choose 0.01 as min support for this task since it would not generate too much patterns (very low min support would generate too many frequent pattern) and it also can generate enough data to find max and closed pattern.

ponder B: Can you figure out which topic corresponds to which domain based on patterns you mine?

Topic -0: Data Mining
Topic -1: Machine Learning
Topic -2: Theory
Topic -3: Database
Topic -4: Information Retrieval

ponder C: Compare the result of frequent patterns, maximal patterns and closed patterns, is the result satisfying? Write down your analysis.

Yes, the result is satisfying. Frequent patterns show the appropriate domain for each topic. And maximal patterns and closed patterns reduce the size of patterns. Eg. The term – data, shows up in multi files. By using maximal patterns and closed patterns, we can easily find which type of data in that topic.

List source file

Step 1-2: assignment3.py
Step 3: step3.py
Step 4: step4.py
Step 5: step5.py
Step 6: step6.py