Frequent Pattern Mining Programming

Prakyath Jaladhar
jaladha2@illinois.edu

This is a report for the frequent pattern mining assignment of CS412. In this report I will be explaining the algorithm used to mine frequent patterns in the given topic files. Then I will be answering the "Questions to ponder", followed by the explaining how patterns are re-ranked by purity of patterns given in the KERT paper. The bonus part is also implemented and the list of source files used is also explained.

STEP 1: Implementation of the Frequent Pattern Mining Algorithm

In this step I have implemented the **Apriori algorithm** to mine frequent patterns. Each line in the "topic" file can be considered as a transaction or a tuple and each word in a transaction is an item. For this assignment I choose a minimum support of 1% (0.01 – this is explained in the questions to ponder part).

- To run the algorithm place topic-i.txt (i = 0, …, 4) in "data-assign3" folder and run the following command: $ python Apriori.py
- First part of the algorithm involves generating frequent 1-itemsets. Each topic file topic-t (t = 0, ..., 4).txt is scanned to derive the sets of frequent items(1-itemsets) and their support counts. Then the 1-itemsets are sorted according to their respective support count in descending order.
- Then for every frequent 1-itemset, the Apriori algorithm is implemented which follows two steps
  - Joining of to form k-itemsets and checking if the sets have infrequent itemsets using the Apriori condition.
  - The second step is pruning; where each k-itemset is scanned with the database to get the support count and removed if it is less than the minimum support chosen.
- The last step in this is to match the numbers in the frequent items mined to the keys in the vocab.txt file and to print the output in the pattern-i.txt file. (i = 0, …, 4).
  - *The format is: [relative support] [term1] [term 2] … [term n]*
- The mapping to the vocab.txt is stored in a dictionary with the number as assigned to the word as the "key" and the word assigned to it as the "value".

STEP 2: Mining Maximal and Closed Patterns

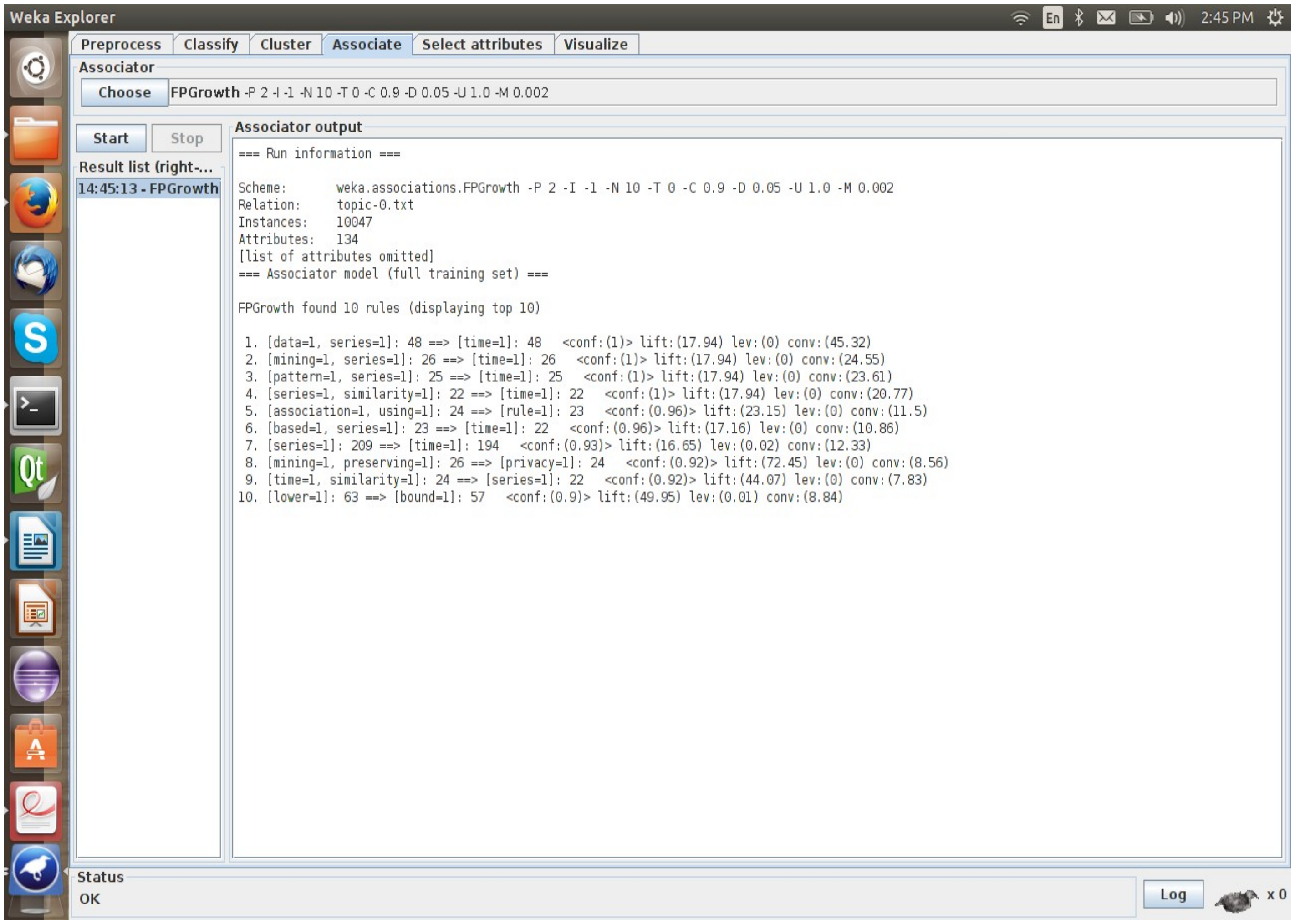
For the step 2, I use the output of the step 1 to generate the maximal and closed patterns. I follow the following steps as explained by the professor in the class:

1. Closed patterns: I considered an itemset to be a closed pattern if it is frequent and there exists no super-pattern, with the same support as the itemset.
   The output of this step is stored in the files closed-i.txt (i = 0, …, 4) in the "closed" folder.
   The format is: [relative support] [term1] [term 2] … [term n]
2. Maximal patterns: I considered an itemset to be a Maximal pattern if it is frequent and there exists no frequent super-pattern.
   The output of this step is stored in the files max-i.txt (i = 0, …, 4) in the "max" folder.
   The format is: [relative support] [term1] [term 2] … [term n]

STEP 3: Association rule mining by WEKA

After following the steps given in the assignment statement to generate the association rules by using WEKA. The following are the outputs generated for each of the topic files:

1. topic-0.arff - lowerBoundMinSupport = 0.002 and minMetric = 0.9



FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.002

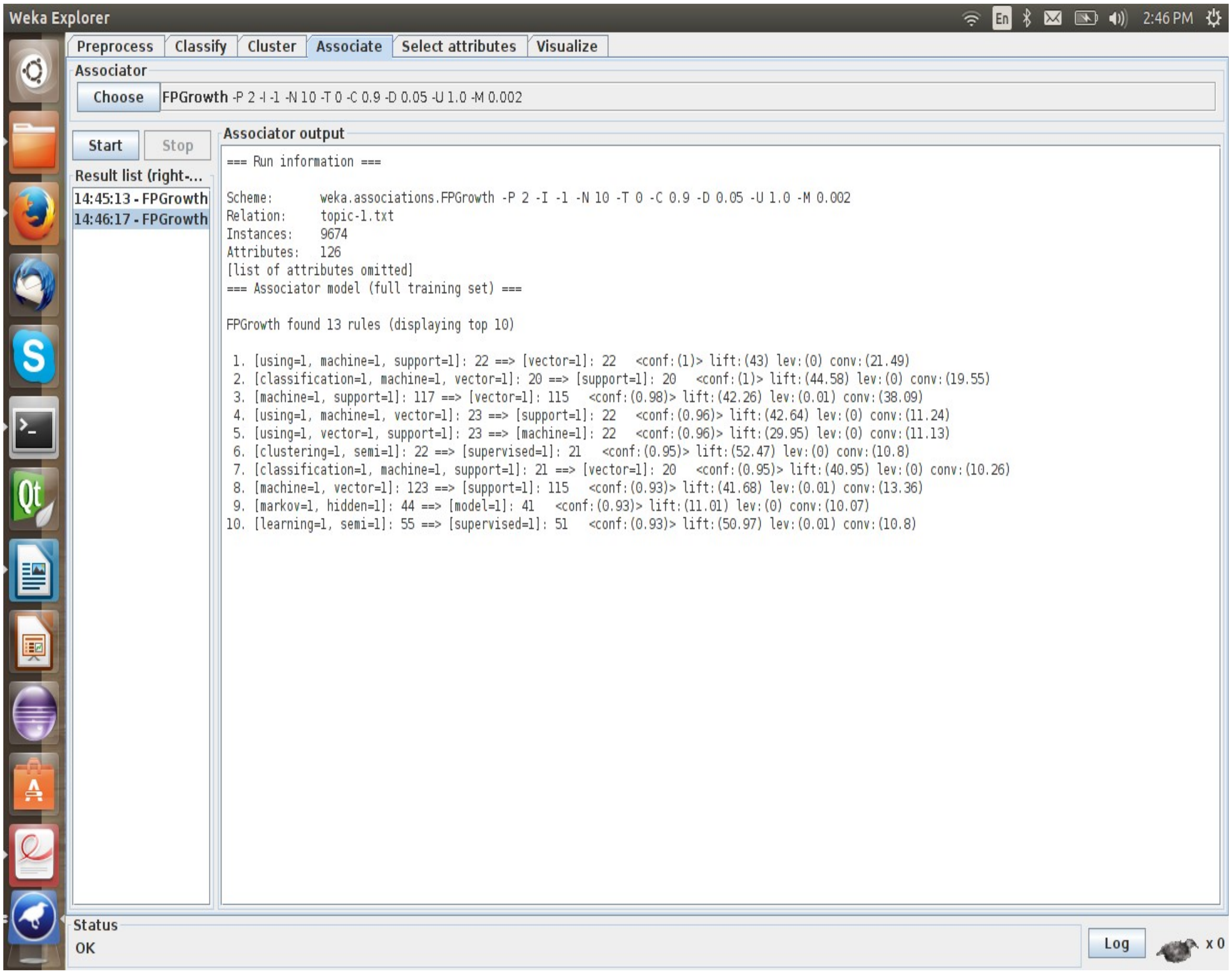
```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.002
Relation:     topic-0.txt
Instances:    10047
Attributes:   134
[list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 10 rules (displaying top 10)

 1. [data=1, series=1]: 48 ==> [time=1]: 48   <conf:(1)> lift:(17.94) lev:(0) conv:(45.32)
 2. [mining=1, series=1]: 26 ==> [time=1]: 26   <conf:(1)> lift:(17.94) lev:(0) conv:(24.55)
 3. [pattern=1, series=1]: 25 ==> [time=1]: 25   <conf:(1)> lift:(17.94) lev:(0) conv:(23.61)
 4. [series=1, similarity=1]: 22 ==> [time=1]: 22   <conf:(1)> lift:(17.94) lev:(0) conv:(20.77)
 5. [association=1, using=1]: 24 ==> [rule=1]: 23   <conf:(0.96)> lift:(23.15) lev:(0) conv:(11.5)
 6. [based=1, series=1]: 23 ==> [time=1]: 22   <conf:(0.96)> lift:(17.16) lev:(0) conv:(10.86)
 7. [series=1]: 209 ==> [time=1]: 194   <conf:(0.93)> lift:(16.65) lev:(0.02) conv:(12.33)
 8. [mining=1, preserving=1]: 26 ==> [privacy=1]: 24   <conf:(0.92)> lift:(72.45) lev:(0) conv:(8.56)
 9. [time=1, similarity=1]: 24 ==> [series=1]: 22   <conf:(0.92)> lift:(44.07) lev:(0) conv:(7.83)
10. [lower=1]: 63 ==> [bound=1]: 57   <conf:(0.9)> lift:(49.95) lev:(0.01) conv:(8.84)
```

2. topic-1.arff - lowerBoundMinSupport = 0.002 and minMetric = 0.9



FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.002

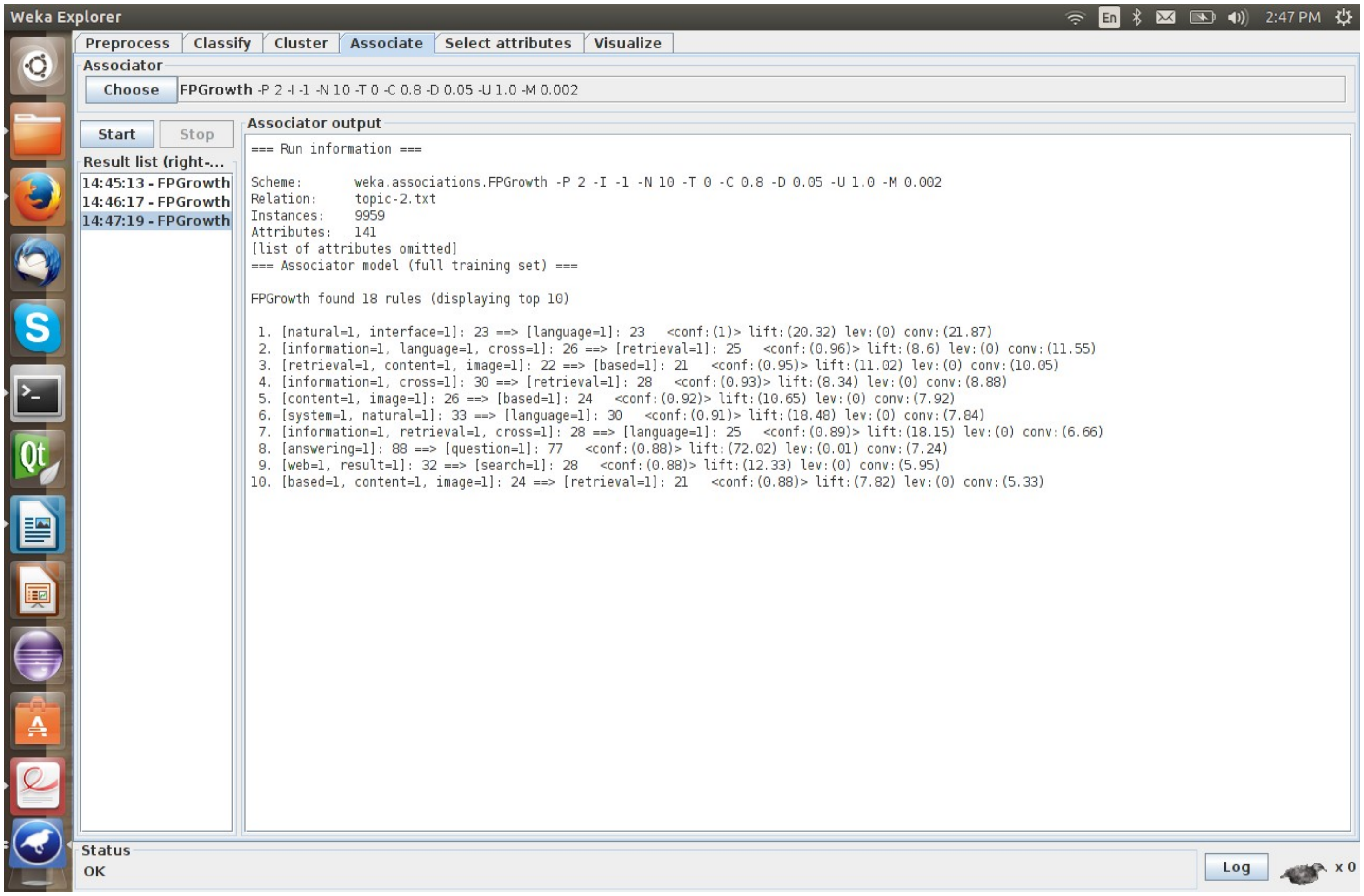
```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.002
Relation:     topic-1.txt
Instances:    9674
Attributes:   126
[list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 13 rules (displaying top 10)

 1. [using=1, machine=1, support=1]: 22 ==> [vector=1]: 22   <conf:(1)> lift:(43) lev:(0) conv:(21.49)
 2. [classification=1, machine=1, vector=1]: 20 ==> [support=1]: 20   <conf:(1)> lift:(44.58) lev:(0) conv:(19.55)
 3. [machine=1, support=1]: 117 ==> [vector=1]: 115   <conf:(0.98)> lift:(42.26) lev:(0.01) conv:(38.09)
 4. [using=1, machine=1, vector=1]: 23 ==> [support=1]: 22   <conf:(0.96)> lift:(42.64) lev:(0) conv:(11.24)
 5. [using=1, vector=1, support=1]: 23 ==> [machine=1]: 22   <conf:(0.96)> lift:(29.95) lev:(0) conv:(11.13)
 6. [clustering=1, semi=1]: 22 ==> [supervised=1]: 21   <conf:(0.95)> lift:(52.47) lev:(0) conv:(10.8)
 7. [classification=1, machine=1, support=1]: 21 ==> [vector=1]: 20   <conf:(0.95)> lift:(40.95) lev:(0) conv:(10.26)
 8. [machine=1, vector=1]: 123 ==> [support=1]: 115   <conf:(0.93)> lift:(41.68) lev:(0.01) conv:(13.36)
 9. [markov=1, hidden=1]: 44 ==> [model=1]: 41   <conf:(0.93)> lift:(11.01) lev:(0) conv:(10.07)
10. [learning=1, semi=1]: 55 ==> [supervised=1]: 51   <conf:(0.93)> lift:(50.97) lev:(0.01) conv:(10.8)
```

3. topic-2.arff - lowerBoundMinSupport = 0.002 and minMetric = 0.8



FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.002

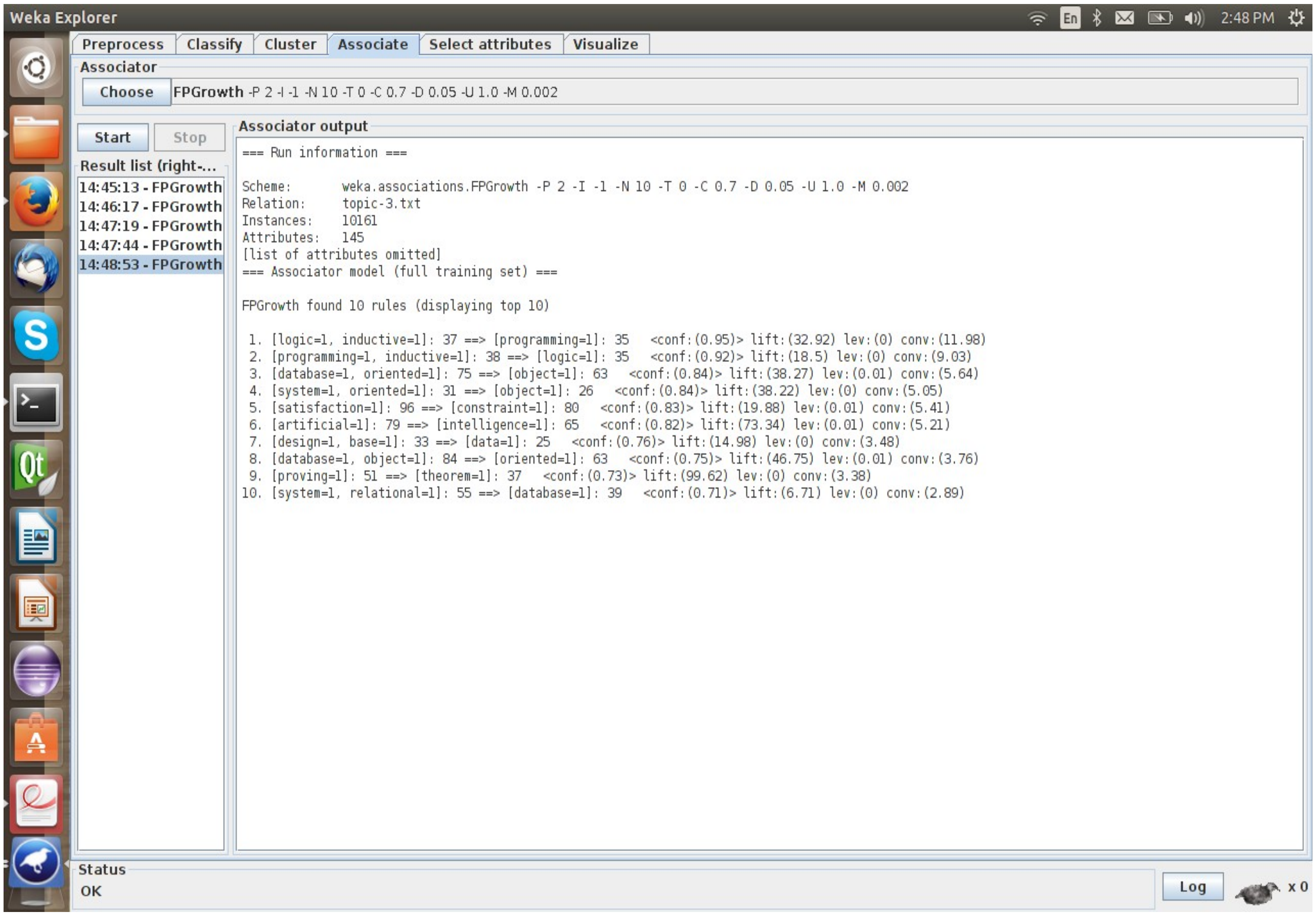
```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.002
Relation:     topic-2.txt
Instances:    9959
Attributes:   141
[list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 18 rules (displaying top 10)

 1. [natural=1, interface=1]: 23 ==> [language=1]: 23   <conf:(1)> lift:(20.32) lev:(0) conv:(21.87)
 2. [information=1, language=1, cross=1]: 26 ==> [retrieval=1]: 25   <conf:(0.96)> lift:(8.6) lev:(0) conv:(11.55)
 3. [retrieval=1, content=1, image=1]: 22 ==> [based=1]: 21   <conf:(0.95)> lift:(11.02) lev:(0) conv:(10.05)
 4. [information=1, cross=1]: 30 ==> [retrieval=1]: 28   <conf:(0.93)> lift:(8.34) lev:(0) conv:(8.88)
 5. [content=1, image=1]: 26 ==> [based=1]: 24   <conf:(0.92)> lift:(10.65) lev:(0) conv:(7.92)
 6. [system=1, natural=1]: 33 ==> [language=1]: 30   <conf:(0.91)> lift:(18.48) lev:(0) conv:(7.84)
 7. [information=1, retrieval=1, cross=1]: 28 ==> [language=1]: 25   <conf:(0.89)> lift:(18.15) lev:(0) conv:(6.66)
 8. [answering=1]: 88 ==> [question=1]: 77   <conf:(0.88)> lift:(72.02) lev:(0.01) conv:(7.24)
 9. [web=1, result=1]: 32 ==> [search=1]: 28   <conf:(0.88)> lift:(12.33) lev:(0) conv:(5.95)
10. [based=1, content=1, image=1]: 24 ==> [retrieval=1]: 21   <conf:(0.88)> lift:(7.82) lev:(0) conv:(5.33)
```

4. topic-3.arff - lowerBoundMinSupport = 0.002 and minMetric = 0.8



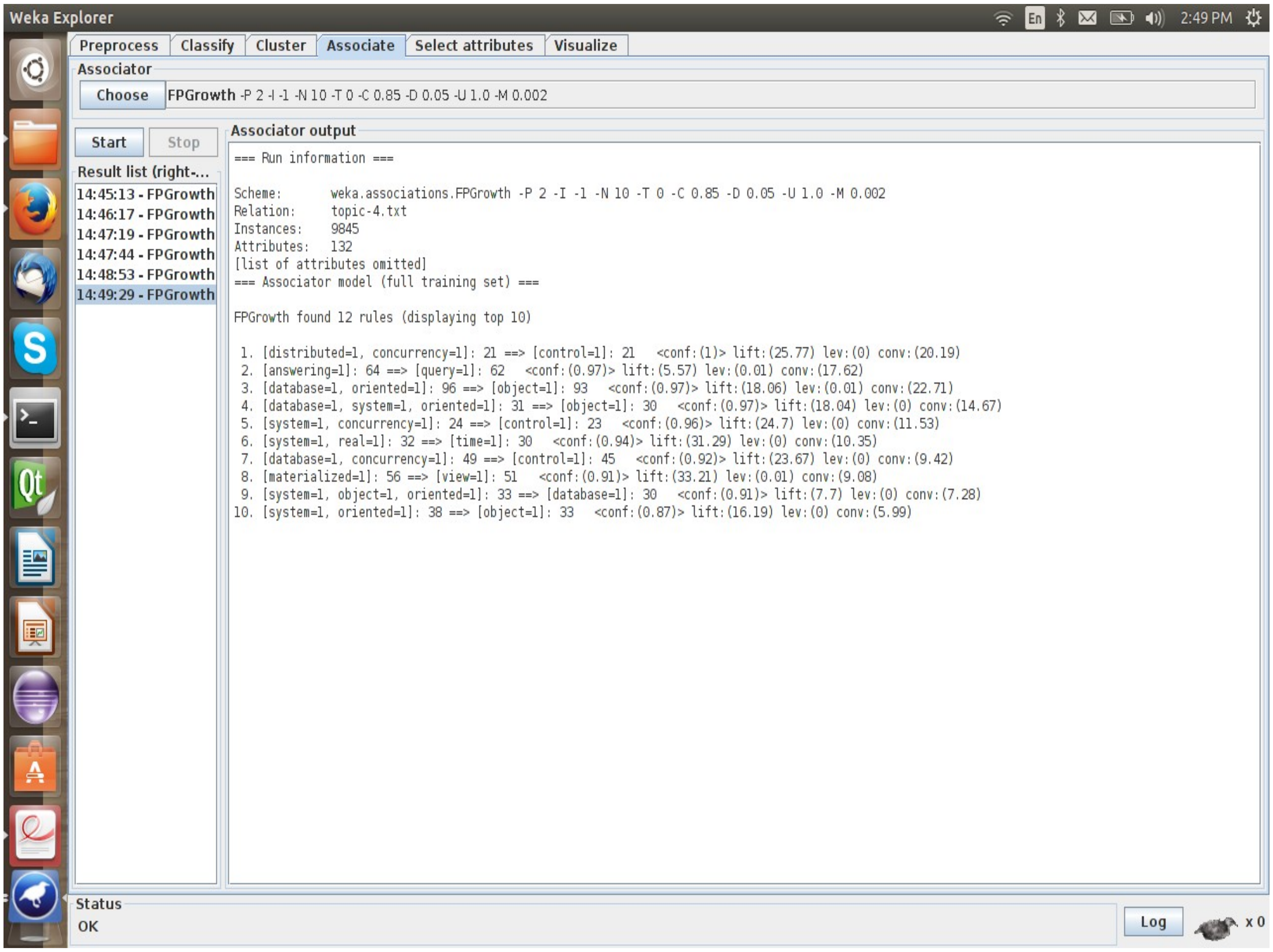
```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.002
Relation:     topic-3.txt
Instances:    10161
Attributes:   145
[list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 10 rules (displaying top 10)

 1. [logic=1, inductive=1]: 37 ==> [programming=1]: 35   <conf:(0.95)> lift:(32.92) lev:(0) conv:(11.98)
 2. [programming=1, inductive=1]: 38 ==> [logic=1]: 35   <conf:(0.92)> lift:(18.5) lev:(0) conv:(9.03)
 3. [database=1, oriented=1]: 75 ==> [object=1]: 63   <conf:(0.84)> lift:(38.27) lev:(0.01) conv:(5.64)
 4. [system=1, oriented=1]: 31 ==> [object=1]: 26   <conf:(0.84)> lift:(38.22) lev:(0) conv:(5.05)
 5. [satisfaction=1]: 96 ==> [constraint=1]: 80   <conf:(0.83)> lift:(19.88) lev:(0.01) conv:(5.41)
 6. [artificial=1]: 79 ==> [intelligence=1]: 65   <conf:(0.82)> lift:(73.34) lev:(0.01) conv:(5.21)
 7. [design=1, base=1]: 33 ==> [data=1]: 25   <conf:(0.76)> lift:(14.98) lev:(0) conv:(3.48)
 8. [database=1, object=1]: 84 ==> [oriented=1]: 63   <conf:(0.75)> lift:(46.75) lev:(0.01) conv:(3.76)
 9. [proving=1]: 51 ==> [theorem=1]: 37   <conf:(0.73)> lift:(99.62) lev:(0) conv:(3.38)
10. [system=1, relational=1]: 55 ==> [database=1]: 39   <conf:(0.71)> lift:(6.71) lev:(0) conv:(2.89)
```

5. topic-4.arff - lowerBoundMinSupport = 0.002 and minMetric = 0.85



FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.85 -D 0.05 -U 1.0 -M 0.002

```
14:45:13 - FPGrowth
14:46:17 - FPGrowth
14:47:19 - FPGrowth
14:47:44 - FPGrowth
14:48:53 - FPGrowth
14:49:29 - FPGrowth
```

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.85 -D 0.05 -U 1.0 -M 0.002
Relation:     topic-4.txt
Instances:    9845
Attributes:   132
[list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 12 rules (displaying top 10)

 1. [distributed=1, concurrency=1]: 21 ==> [control=1]: 21   <conf:(1)> lift:(25.77) lev:(0) conv:(20.19)
 2. [answering=1]: 64 ==> [query=1]: 62   <conf:(0.97)> lift:(5.57) lev:(0.01) conv:(17.62)
 3. [database=1, oriented=1]: 96 ==> [object=1]: 93   <conf:(0.97)> lift:(18.06) lev:(0.01) conv:(22.71)
 4. [database=1, system=1, oriented=1]: 31 ==> [object=1]: 30   <conf:(0.97)> lift:(18.04) lev:(0) conv:(14.67)
 5. [system=1, concurrency=1]: 24 ==> [control=1]: 23   <conf:(0.96)> lift:(24.7) lev:(0) conv:(11.53)
 6. [system=1, real=1]: 32 ==> [time=1]: 30   <conf:(0.94)> lift:(31.29) lev:(0) conv:(10.35)
 7. [database=1, concurrency=1]: 49 ==> [control=1]: 45   <conf:(0.92)> lift:(23.67) lev:(0) conv:(9.42)
 8. [materialized=1]: 56 ==> [view=1]: 51   <conf:(0.91)> lift:(33.21) lev:(0.01) conv:(9.08)
 9. [system=1, object=1, oriented=1]: 33 ==> [database=1]: 30   <conf:(0.91)> lift:(7.7) lev:(0) conv:(7.28)
10. [system=1, oriented=1]: 38 ==> [object=1]: 33   <conf:(0.87)> lift:(16.19) lev:(0) conv:(5.99)
```

STEP 4: Re-ranking by purity of the patterns

According to the assignment instructions and the KERT paper the purity of an item is given by the formula,

$$\boldsymbol{\pi \text{ or pur } (p) = \log (f_t(p) / | D_t |) - \log \max ((f_t(p) + f_{t'}(p))/ | D_{t,t'} |)}$$

Thus we first get the purity values for each pattern and then re-rank the patterns using a measure combined purity and support.

Here $f_t(p)$ denotes support count of pattern p in topic t and we can directly get it from the result of step 2. D_t is the collection of documents where there is at least one word assigned to topic t, which is actually the number of lines in topic-t (t = 0, ..., 4).txt and thus we can get it from the assignment instructions. $D_{t,t'}$ is the union of D_t and $D_{t'}$, which means the collection of documents where there is at least one word assigned to topic t or topic t' .

I have used a dictionary to store all the frequent patterns generated and to calculate purity each pattern is scanned in its own pattern file to get the $f_t(p)$ and in other pattern-i files to get $f_{t'}(p)$ values and then by taking the max value of it and calculating the frequency of all frequent patterns. The output of the step is recorded in the file purity-i.txt (i = 0, …, 4). In order to rank the patterns I have multiplied each patterns' purity with its relative support and have sorted the patterns from high to low by the measure.

*The format is: [purity * relative support] [term1] [term 2] … [term n]*

or

[measure] [term1] [term 2] … [term n]

Note: I have also recorded the output without combining the purity and the support of each patterns it is recorded in "purity_measure" folder with files named purity-measure-i.txt (i = 0, …, 4). The patterns are sorted in from high to low with respect to their purity in these files.

<u>QUESTIONS TO PONDER</u>:

1. How do you choose min_sup for this task? Explain how you choose the min_sup in your report.

For the assignment, I set relative min sup to 1% (0.01). Since the number of documents varies in different topic files, I use relative minimum support instead of an absolute minimum support. By experiment, many meaningful phrases could be discovered in each topic by setting min sup to 1%.

A reasonable minimum support is supposed to generate enough output patterns so that many potential good phrases for each topic can be retrieved. But, the min sup cannot be too low that it takes almost all the itemsets as frequent patterns.

2. Can you figure out which topic corresponds to which domain based on patterns you mine? Write your observations in the report.

We can figure out which topic corresponds to which domain by analyzing the frequent patterns, closed patterns and max patterns mined for each topic.

- "rule mining association", "distance" are found in max patterns of Topic-0, this terms are related to the Data Mining domain
- Pattern-1 has "learning", "classification", "clustering" and series of other correlated terms in machine learning area as its top ranked frequent patterns and thus corresponds to Machine Learning domain.
- "information retrieval", "analysis", "web" and many other terms are related to the Information Retrieval domain.
- Pattern-3 has many words such as "database", "system database", "database relational" which correspond to Database Management domain.
- Finally the pattern-4 file has "xml", "processing" and similar terms which are related to the Theory domain.

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

I believe the result is satisfying. By mining for frequent patterns, the patterns generated are very large in number and some of which overlap with the patterns causing the pattern to appear multiple times. For example, in the output file pattern-0.txt the patterns "association", "data", "rule association" and "rule mining association", but only "rule mining association" is mined as the max pattern in the file max-0.txt.

4. What is the quality of the phrases that satisfies both min_sup and min_conf? Please compare it to the results of Step1 and put down your observations.

The phrases which satisfy both the min_sup and min_conf are the ones which are present in the frequent itemsets generated by the Apriori algorithm. As we can see the output using the WEKA which is shown above, the association rules generated are of the items which satisfy both minimum support and the minimum confidence chosen. These provide a meaningful insight into what items are related in the given topic files. For example, in pattern-0.txt we can see that "association" → "rule" this is also generated by WEKA.

<u>SOURCE FILE STRUCTURE:</u>

There is only one source file used in this assignment and I used python to code,

1. Apriori.py - STEP 1, STEP 2, STEP 4 and STEP 5

The functions corresponding to the steps are:

STEP 1:	apriori_generation, find_frequent, candidate_generation, add_frequency
STEP 2:	findSuperPattern, convertFS, rev_freq_pattern and writePattern
STEP 4:	pure_items
STEP 5:	phraseness_coverage

<u>BONUS:</u>

To implement the ranking criteria I referred the KERT paper, I have implemented coverage and phraseness.

Phraseness: A group of words should be combined together as phrase if they co-occur significantly more often than the expected chance co-occurrence frequency, given that each term in the phrase occurs independently. The formula to calculate the phraseness is,

$$\boldsymbol{\pi\ phr(p) = \log(f_t(p) / |D_t|) - \sum \log((f_t(w)) / |D_t|)}$$

While calculating the phraseness coverage of each phrase is calculated,

$$\boldsymbol{cov(p) = f_t(p) / |D_t|}$$

The output is stored in phraseness-i.txt and coverage-i.txt (i = 0, …, 4).