CS412: Introduction to Data Mining **Fall 2015**

Assignment 3

Li Miao *October 22, 2015*

General Instruction

- Errata: After the assignment is released, any further corrections of errors or clarifications will be posted at the Errata page at Piazza. Please watch it.
- Feel free to talk to other members of the class when doing the homework. We are more concerned about whether you learn how to solve the problem than whether you solve it entirely on your own. You should, however, write down the solution yourself.
- Please use Piazza first if you have questions about the homework. Also feel free to send us e-mails and come to office hours.
- For each question, you will **NOT** get full credit if you only give a final result. Necessary calculation steps and reasoning are required.
- For each question, you should show the necessary calculation steps and reasoning—not only final results. Keep the solution brief and clear.
- For a good balance of cognitive activities, we label each question with an activity type:
  - **L1 (Knowledge)** Definitions, propositions, basic concepts.
  - **L2 (Practice)** Repeating and practicing algorithms/procedures.
  - **L3 (Application)** Critical thinking to apply, analyze, and assess.

Assignment Submission

- Please submit your work before the due time. **We do NOT accept late homework!**
- Please submit your answers electronically via Compass (`http://compass2g.illinois.edu`). Contact TAs if you have technical difficulties in submitting the assignment.
- Please **type** your answers in an **Answer Document**, and submit it in PDF. **Handwritten answers or hand-drawn pictures are not acceptable**.
- This assignment consists of four written assignments and one large Machine Problem (MP). Your answers to all questions (including MP) should be included in one Answer Document, named as `NetId_assign3_answer.pdf`.
- The four written questions do not require programming at all. The last part is the first MP (not a mini one) we have in this course. It consists of several programming assignments, which usually take more time than written assignments, so please start early.
- Find detailed submission guidelines for codes and results in the requirements of MP.

Question 1 (12 points)

Based on the tiny database of 5 transactions in Table 1, use the Apriori algorithm to find the frequent patterns with relative $min\_sup = 0.6$.

Purpose

- Get a better understanding as well as hands-on experience of the Apriori algorithm.

Requirements

- For this question, you are required to simulate the basic Apriori algorithm and write down all intermediate as well as final results. No programming is needed.
- Use the abbreviations we give you (C1, F1...) to denote which list you are writing about. You may use a table to contain all lists, or just write them one by one.
- For each itemset you write down in the Fi lists, put its corresponding absolute support after it, with a colon between them, such as $F1 = \{m : 4, \ldots\}$.
- Do not forget to write down the actions (pruning or self-joining) you take to generate F1 and C2.

Trans.	Items
1	b,d,f,g,l
2	f,g,h,l,m,n
3	b,f,h,k,m
4	a,f,h,j,m
5	d,f,g,j,m

Table 1: A tiny transaction database

a. $(2', L1)$ We do db-scanning to get rid of non-frequent 1-itemsets

C1	
Itemset	Sup
{a}	1
{b}	2
{d}	2
{f}	5
{g}	3
{h}	3
{j}	2
{k}	1
{l}	2
{m}	4
{n}	1

L1	
Itemset	Sup
{f}	5
{g}	3
{h}	3
{m}	4

b. $(2', L1)$ We do self-joining to generate all candidate 2-itemsets

C2	
Itemset	Sup
{f,g}	3
{f,h}	3
{f,m}	4
{g,h}	1
{g,m}	2
{h,m}	3

c. $(2', L1)$

L2	
Itemset	Sup
{f,g}	3
{f,h}	3
{f,m}	4
{h,m}	3

d. $(2', L1)$ We need pruning.

C3	
Itemset	Sup
{f,h,m}	3

e. $(2', L2)$ There is no frequent 4-itemset. Because we only have one frequent 3-itemset.

f. $(2', L3)$ I think the self-joining involves the heaviest computation. We need to generate so many joint candidates. When *min_sup* is really low, we need to consider more itemsets. Therefore it would be extremely bad. M^N where M is number of distinct items and N is the max length of transactions

Question 2 (13 points)

Based on the same database Question 1, use the Frequent Pattern Growth algorithm with relative $min\_sup = 0.4$ to find the frequent patterns.

Purpose

- Get a better understanding as well as hands-on experience of the FP-Growth algorithm.

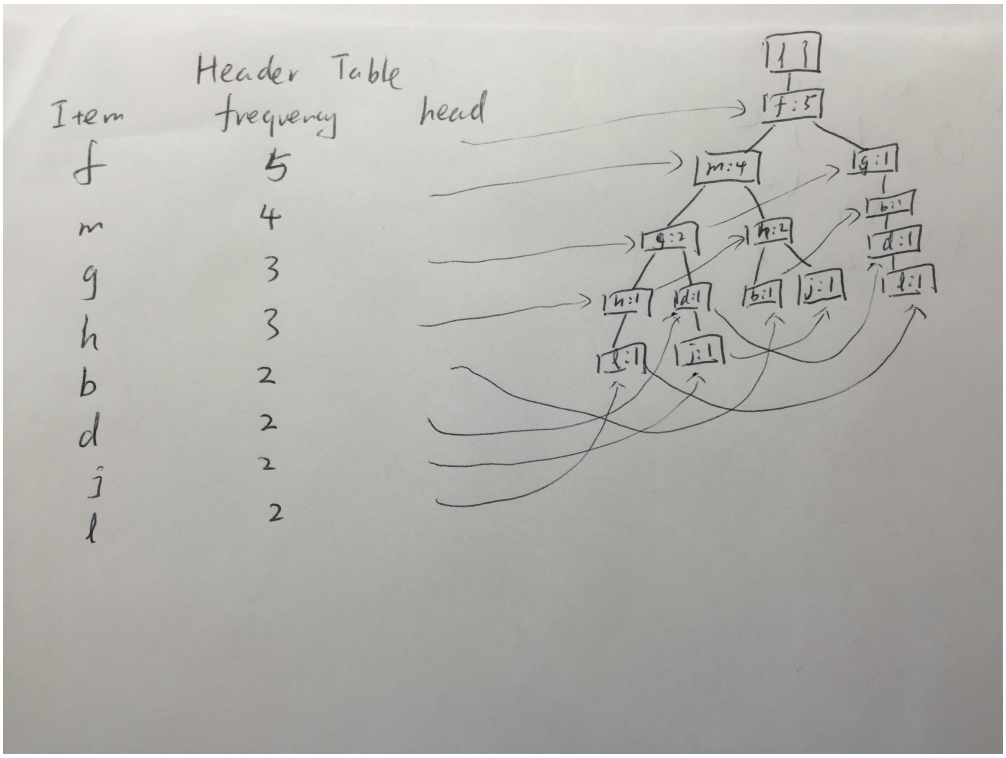
Requirement

- For this question, you are required to simulate the basic FP-Growth algorithm. No programming is needed.
- You are required to generate some tables and figures. You can use any software to do that, but you can not draw with hand and then scan or take photos.
- For sub-question a, generate a table to present the results.
- For sub-question b, put the Header Table and FP-tree side by side, preferably with Header Table on the left (just like those in the slides). Use a colon to separate an item and its corresponding count in the FP-tree.
- For sub-question c, for each of the items, write down its Conditional Pattern Base followed by the frequent patterns computed based on it. In order to generate the correct frequent patterns, please first generate the Conditional FP-trees. But you do not need to show the Conditional FP-trees in the Answer Document.

a. $(2', L2)$

TID	Items	Ordered frequent items
1	b,d,f,g,l	{f,g,b,d,l}
2	f,g,h,l,m,n	{f,m,g,h,l}
3	b,f,h,k,m	{f,m,h,b}
4	a,f,h,j,m	{f,m,h,j}
5	d,f,g,j,m	{f,m,g,d,j}

b. $(5', L2)$



c. $(4', L2)$

Conditional Pattern Bases		
Item	Cond.pattern base	Frequent patterns
m	f:4	m,mf
h	fmg:1 fm:2	h,hf,hm,hfm
b	fmh:1 fg:1	b, bf
j	fmgd:1 fmh:1	j, jf ,jm,jmf

d. $(2', L3)$ We hope to use FP tree to compress our database. When we order items according to the frequency, we can make the tree smaller. So it will be more efficient and save the computation cost.

Question 3 (10 points)

Based on the frequent patterns computed in Question 1, find closed patterns, maximal patterns and association rules.

Purpose

- Get a better understanding of closed patterns, maximal patterns and association rules.

Requirement

- For this question, you will be doing some counting. No programming is needed.
- For sub-question c, show the relative support and confidence of each association rule you find.

a. $(3', L1)$ Closed: f, fg ,fm, fhm

b. $(3', L1)$ Maximal: fg, fhm

c. $(2', L2)$

- $g \rightarrow f$ (0.6,1)
- $h \rightarrow f$ (0.6,1)

d. $(2', L3)$ Max implies closed, but not versa vice. So if we only hope to get the general idea of frequent patterns, like what items would like to show up together, we may choose Max. But if we hope to get more details, we would choose closed patterns.

Question 4 (15 points)

This is a set of **true** or **false** questions. Please answer the following questions.

Purpose

- Have a better understanding of some basic concepts about frequent pattern mining.

Requirement

- For each sub-question, choose **true (T)** or **false (F)** and provide a brief explanation of your choice. You will not get credit without explanation.

a. $(3', L3)$ T. Max means support(super(x)) $<$ *mim_support* <=support(x) ,and Closed means support(super(x)) <support(x). It is obvious that max implies closed, but not versa vice.

b. $(3', L2)$ F. When K =1, this statement is not true. Because(K-1) = 0. And to remove infrequent items, we also use pruning, not just scanning the database.

c. $(3', L2)$ F. For FP-Growth, in order to mine all frequent patterns, we have to recursively generate conditional frequent bases and conditional FP-trees until the FP-tree generated has single path.

d. $(3', L1)$ F. CLOSET is based on FP-Growth while MaxMiner is based on Apriori.

e. $(3', L3)$ F. Support and confidence are not good to indicate correlations. But we cannot say Lift is always better. We can see the obvious example on textbook. The Lift is low but there is a clear association relationship for D2. We can use support and confidence to find interesting association rules.

Table 6.9 Comparison of Six Pattern Evaluation Measures Using Contingency Tables for a Variety of Data Sets

Data Set	mc	$\overline{m}c$	$m\overline{c}$	$\overline{m}\overline{c}$	χ^2	*lift*	*all_conf.*	*max_conf.*	*Kulc.*	*cosine*
D_1	10,000	1000	1000	100,000	90557	9.26	0.91	0.91	0.91	0.91
D_2	10,000	1000	1000	100	0	1	0.91	0.91	0.91	0.91
D_3	100	1000	1000	100,000	670	8.44	0.09	0.09	0.09	0.09
D_4	1000	1000	1000	100,000	24740	25.75	0.5	0.5	0.5	0.5
D_5	1000	100	10,000	100,000	8173	9.18	0.09	0.91	0.5	0.29
D_6	1000	10	100,000	100,000	965	1.97	0.01	0.99	0.5	0.10

Machine Problem (MP, 50 points)

Computing frequent patterns by hand is so tedious - this is where computers come into use! In this MP, given preprocessed data of the paper titles collected from computer science conferences from 5 domains, you are required to 1) implement a frequent pattern mining

algorithm to mine frequent patterns from each of the 5 domains, so as to find 'meaningful' patterns for each domain; 2) mine closed/maximal patterns based on the frequent patterns you find, so as to understand the different definitions and applications of closed/maximal patterns; 3) find association rules using Weka. You can find `data.zip` from the course website.

Description of Data Preprocessing

- We use paper titles collected from conferences in computer science of 5 domains: Data Mining (DM), Machine Learning (ML), Database (DB), Information Retrieval (IR) and Theory (TH). The raw data is named as `paper_raw.txt`. Each line contains two columns, the `PaperID` and the `Title` of a paper, separated by a tab. Recall the example in class. You can consider each line in the file as one transaction. Each term in the title is then equivalent to an item in a transaction. We provide this file to give you a basic idea about what the task is and what data you are using. You will not work on this file directly.

- For this assignment, we have pre-processed the raw data by removing stop words, converting the words to lower cases, and lemmatization. The results are in `paper.txt`. In this file, each line is a list of terms. Terms are separated by one space. Again, this file is for you to understand the task, and you will not work on this file directly.

- To make computation easier, we generate a vocabulary from `paper.txt`, and name it as `vocab.txt`. Each line in this file has two columns: the first column is the term index and the second column is a unique term extracted from `paper.txt`; columns are separated by Tab. Each term in `paper.txt` appears exactly once in `vocab.txt`. With this vocabulary, we can always map between each term and its corresponding unique indexing number. You do not need to know how this vocabulary is generated, but you do need this file to show mined patterns as required in Step 1 and Step 2 of the MP.

- Recall we have papers from 5 domains. We want you to mine frequent patterns for each of the 5 domains so as to find 'meaningful' patterns of each domain. However, the terms of 5 domains are mixed together in `paper_raw.txt` and `paper.txt`. In order to separate them, we apply LDA with 5 topics to assign one topic to each term. Then we re-organize the terms and create one file `topic-`i`.txt` for each topic i, where $i = 0, 1, 2, 3, 4$. Each line in file `topic-`i`.txt` corresponds to one paper title in the dataset. File `topic-`i`.txt` only contains paper titles with at least one term assigned with topic i by LDA, and within each paper title in `topic-`i`.txt`, terms assigned with topics other than topic i are removed. Note that we do not know which `topic-`i`.txt` corresponds to which of the 5 domains now, and we will use frequent pattern mining to figure it out. These `topic-`i`.txt` are the files you are going to work on to mine frequent patterns.

Step 1: $(25', L3)$ **Mining frequent patterns for each topic.**

Question to ponder A: How do you choose min_sup for this task? Explain your criteria; any reasonable choice will be good.

Answers: I choose the *min_sup* to be 1% of the dataset. It is a reasonable assumption because we will not have too many items to combine or miss some interesting frequent patterns when we set the *min_sup* to be 1% of the dataset.

Step 2: $(20', L3)$ **Mining closed/maximal patterns.**

Question to ponder B: Can you figure out which topic corresponds to which domain based on patterns you mine? Write down your observations.

Answers: The topic-0 file corresponds to Data Mining. The topic-1 file corresponds to Machine Learning. The topic-2 file corresponds to Information Retrieval . The topic-3 file corresponds to Database. The topic-4 file corresponds toTheory.

Question to ponder C: Compare the results of frequent patterns, closed patterns and maximal patterns, is there any difference? If so, what kind of patterns give more satisfying results? Write down your analysis.

Answers: Yes. There are some difference among the results from frequent patterns, closed patterns and maximal patterns. The results from closed patterns and frequent patterns are similar. For maximal patterns, the result files include fewer items than closed patterns or frequent patterns. I think the the frequent patterns give more satisfying results because it capture most key words for the specific domain and exclude some redundant information. And we can easily to figure which topic corresponds to which domain by frequent patterns.

Step 3: $(5', L2)$ **Mining association rules by Weka.**

In this step, you will use Weka to mine association rules. In class, we have demonstrated how to do this. You can download the slides *Weka-Associate* from the course schedule page or watch the online lecture video to review the process. To make things easier, we also provide detailed procedures here.

Figure 1: For topic-0,$min\_sup = 0.01$, $min\_conf = 0.3$

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.3 -D 0.05 -U 1.0 -M 0.01
Relation:     topic-0.txt
Instances:    10047
Attributes:   134
[list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 17 rules (displaying top 10)

 1. [series=1]: 209 ==> [time=1]: 194   <conf:(0.93)> lift:(16.65) lev:(0.02) conv:(12.33)
 2. [mining=1, rule=1]: 159 ==> [association=1]: 123   <conf:(0.77)> lift:(23.13) lev:(0.01) conv:(4.15)
 3. [mining=1, association=1]: 159 ==> [rule=1]: 123   <conf:(0.77)> lift:(18.68) lev:(0.01) conv:(4.12)
 4. [association=1]: 336 ==> [rule=1]: 233   <conf:(0.69)> lift:(16.75) lev:(0.02) conv:(3.1)
 5. [stream=1]: 211 ==> [data=1]: 141   <conf:(0.67)> lift:(5.21) lev:(0.01) conv:(2.59)
 6. [rule=1]: 416 ==> [association=1]: 233   <conf:(0.56)> lift:(16.75) lev:(0.02) conv:(2.19)
 7. [frequent=1]: 227 ==> [mining=1]: 127   <conf:(0.56)> lift:(4.83) lev:(0.01) conv:(1.99)
 8. [rule=1, association=1]: 233 ==> [mining=1]: 123   <conf:(0.53)> lift:(4.56) lev:(0.01) conv:(1.86)
 9. [association=1]: 336 ==> [mining=1]: 159   <conf:(0.47)> lift:(4.09) lev:(0.01) conv:(1.67)
10. [pattern=1]: 528 ==> [mining=1]: 203   <conf:(0.38)> lift:(3.32) lev:(0.01) conv:(1.43)
```

Figure 2: For topic-1,$min\_sup = 0.01$, $min\_conf = 0.5$

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.01
Relation:     topic-1.txt
Instances:    9674
Attributes:   126
[list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 14 rules (displaying top 10)

 1. [machine=1, support=1]: 117 ==> [vector=1]: 115   <conf:(0.98)> lift:(42.26) lev:(0.01) conv:(38.09)
 2. [machine=1, vector=1]: 123 ==> [support=1]: 115   <conf:(0.93)> lift:(41.68) lev:(0.01) conv:(13.36)
 3. [neighbor=1]: 137 ==> [nearest=1]: 121   <conf:(0.88)> lift:(60.6) lev:(0.01) conv:(7.94)
 4. [nearest=1]: 141 ==> [neighbor=1]: 121   <conf:(0.86)> lift:(60.6) lev:(0.01) conv:(6.62)
 5. [neural=1]: 128 ==> [network=1]: 101   <conf:(0.79)> lift:(16.49) lev:(0.01) conv:(4.35)
 6. [vector=1, support=1]: 146 ==> [machine=1]: 115   <conf:(0.79)> lift:(24.66) lev:(0.01) conv:(4.42)
 7. [support=1]: 217 ==> [vector=1]: 146   <conf:(0.67)> lift:(28.93) lev:(0.01) conv:(2.94)
 8. [vector=1]: 225 ==> [support=1]: 146   <conf:(0.65)> lift:(28.93) lev:(0.01) conv:(2.75)
 9. [semi=1]: 165 ==> [supervised=1]: 105   <conf:(0.64)> lift:(34.98) lev:(0.01) conv:(2.66)
10. [supervised=1]: 176 ==> [semi=1]: 105   <conf:(0.6)> lift:(34.98) lev:(0.01) conv:(2.4)
```

Figure 3: For topic-2,$min\_sup = 0.01$, $min\_conf = 0.2$

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.2 -D 0.05 -U 1.0 -M 0.01
Relation:     topic-2.txt
Instances:    9959
Attributes:   141
[list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 11 rules (displaying top 10)

 1. [page=1]: 131 ==> [web=1]: 107   <conf:(0.82)> lift:(6.63) lev:(0.01) conv:(4.59)
 2. [natural=1]: 228 ==> [language=1]: 170   <conf:(0.75)> lift:(15.15) lev:(0.02) conv:(3.67)
 3. [engine=1]: 181 ==> [search=1]: 122   <conf:(0.67)> lift:(9.49) lev:(0.01) conv:(2.8)
 4. [service=1]: 267 ==> [web=1]: 117   <conf:(0.44)> lift:(3.56) lev:(0.01) conv:(1.55)
 5. [retrieval=1]: 1114 ==> [information=1]: 475   <conf:(0.43)> lift:(3.51) lev:(0.03) conv:(1.53)
 6. [information=1]: 1211 ==> [retrieval=1]: 475   <conf:(0.39)> lift:(3.51) lev:(0.03) conv:(1.46)
 7. [language=1]: 490 ==> [natural=1]: 170   <conf:(0.35)> lift:(15.15) lev:(0.02) conv:(1.49)
 8. [semantic=1]: 414 ==> [web=1]: 104   <conf:(0.25)> lift:(2.04) lev:(0.01) conv:(1.17)
 9. [document=1]: 564 ==> [retrieval=1]: 141   <conf:(0.25)> lift:(2.23) lev:(0.01) conv:(1.18)
10. [search=1]: 707 ==> [web=1]: 173   <conf:(0.24)> lift:(1.99) lev:(0.01) conv:(1.16)
```

Figure 4: For topic-3,$min\_sup = 0.01$, $min\_conf = 0.2$

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.2 -D 0.05 -U 1.0 -M 0.01
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. [oriented=1]: 163 ==> [object=1]: 102   <conf:(0.63)> lift:(28.51) lev:(0.01) conv:(2.57)
 2. [reinforcement=1]: 224 ==> [learning=1]: 117   <conf:(0.52)> lift:(9.51) lev:(0.01) conv:(1.96)
 3. [discovery=1]: 202 ==> [knowledge=1]: 104   <conf:(0.51)> lift:(7.04) lev:(0.01) conv:(1.89)
 4. [object=1]: 223 ==> [oriented=1]: 102   <conf:(0.46)> lift:(28.51) lev:(0.01) conv:(1.8)
 5. [base=1]: 337 ==> [data=1]: 138   <conf:(0.41)> lift:(8.1) lev:(0.01) conv:(1.6)
 6. [relational=1]: 375 ==> [database=1]: 129   <conf:(0.34)> lift:(3.25) lev:(0.01) conv:(1.36)
 7. [base=1]: 337 ==> [knowledge=1]: 111   <conf:(0.33)> lift:(4.5) lev:(0.01) conv:(1.38)
 8. [data=1]: 514 ==> [base=1]: 138   <conf:(0.27)> lift:(8.1) lev:(0.01) conv:(1.32)
 9. [system=1]: 928 ==> [database=1]: 195   <conf:(0.21)> lift:(1.99) lev:(0.01) conv:(1.13)
10. [learning=1]: 558 ==> [reinforcement=1]: 117   <conf:(0.21)> lift:(9.51) lev:(0.01) conv:(1.23)
```

Figure 5: For topic-4,$min\_sup = 0.01$, $min\_conf = 0.28$

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.28 -D 0.05 -U 1.0 -M 0.01
Relation:     topic-4.txt
Instances:    9845
Attributes:   132
[list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 10 rules (displaying top 10)

 1. [concurrency=1]: 133 ==> [control=1]: 107   <conf:(0.8)> lift:(20.73) lev:(0.01) conv:(4.73)
 2. [oriented=1]: 183 ==> [object=1]: 141   <conf:(0.77)> lift:(14.37) lev:(0.01) conv:(4.03)
 3. [real=1]: 260 ==> [time=1]: 151   <conf:(0.58)> lift:(19.38) lev:(0.01) conv:(2.29)
 4. [stream=1]: 212 ==> [data=1]: 112   <conf:(0.53)> lift:(5) lev:(0.01) conv:(1.88)
 5. [time=1]: 295 ==> [real=1]: 151   <conf:(0.51)> lift:(19.38) lev:(0.01) conv:(1.98)
 6. [processing=1]: 637 ==> [query=1]: 313   <conf:(0.49)> lift:(2.82) lev:(0.02) conv:(1.62)
 7. [optimization=1]: 380 ==> [query=1]: 173   <conf:(0.46)> lift:(2.62) lev:(0.01) conv:(1.51)
 8. [system=1]: 767 ==> [database=1]: 276   <conf:(0.36)> lift:(3.05) lev:(0.02) conv:(1.37)
 9. [object=1]: 528 ==> [database=1]: 154   <conf:(0.29)> lift:(2.47) lev:(0.01) conv:(1.24)
10. [control=1]: 382 ==> [concurrency=1]: 107   <conf:(0.28)> lift:(20.73) lev:(0.01) conv:(1.37)
```

Question to ponder D: What are the differences between phrases which satisfy only the *min_sup* criterion and phrases (association rules) which satisfy both *min_sup* and *min_conf* criteria? Compare the results of Step 1 and Step 3 and write down your observations.

Answers: If we hope the phrases to satisfy both *min_sup* and *min_conf*, we hope to find some association rules and it is the stricter requirement. But when we only need the *min_sup* to be satisfied, we just hope to find some interesting frequent patterns. And we do not know exactly the association or the causality between items.