CS8803 BDS / CS4365

# Homework Assignment 3

## (Programming Category)

Student Name:_____Enmao Diao_____
Student Session: cs8803 or CS4365 (circle one)

In this third programming assignment, you are given one type of programming problem to gain hand-on experience with Frequent Pattern Mining (FRM). There are three problems and you only need to choose one.

1. The first problem can be solved by hand, though you are encouraged to try to write a program to solve it.

2. The second problem requires you to run two open source implementation of the FRM algorithms and compare their performance.

3. The third problem asks you to implement an algorithm for FRM with an option to include some optimization technique(s).

Feel free to choose any of your favorite programming language: Java, C, Perl, Python. This assignment is designed to help you understand and use Data Mining and Machine Learning packages in Mahout/R/DEKA and also to encourage you to design some optimizations for the baseline algorithms.

**Post Date**: Monday of Week 8 (Oct. 5)
**Due Date**: midnight on Friday of Week 10 (Oct. 16 with no penalty grace period until midnight on Saturday of Oct. 17)

### Problem 1. Learning Association Rule Mining by Example

Consider the following transaction database:

| Transaction ID | Items |
|---|---|
| T1 | A, B, C, D |
| T2 | A, B, C, E |
| T3 | A, B, E, F, H |
| T4 | A, C, H |

Suppose that minimum support is set to 50% and minimum confidence to 60%.

You are asked to answer the following four questions. You can answer all five subquestions by hand. You are encouraged to write a baseline program to answer the first four questions though it is optional. You can also find some source code at http://fimi.ua.ac.be/src/ and WEKA/R/Mahout website.

a) List all frequent itemsets together with their support.

b) Which of the itemsets from a) are closed? Which of the itemsets from a) are maximal?

c) For all frequent itemsets of maximal length, list all corresponding association rules satisfying the requirements on (minimum support and) minimum confidence together with their confidence.

d) Using the lift of an association rule defined as follows to compute the lift for the association rules from c).

```
    lift = confidence / support(head)
```

e) Why are only those association rules interesting that have a lift (significantly) larger than 1.0? [Hint: the following equations are true]

```
lift = confidence / support(head)
=confidence / ( support(body) * support(head) / support(body) )
= confidence / expected_confidence
```

a) List all frequent itemsets together with their support.

I write java code FPM.java to generate maximal itemsets.
Since minimum support is 50%, itemsets with support below 2 is closed
The output is

```
----Start FPM----
----1st scan----
----Initial DataBase----
{4=[A, C, H], 3=[A, B, E, F, H], 2=[A, B, C, E], 1=[A, B, C, D]}
----C1----
{[A]=4, [H]=2, [F]=1, [E]=2, [D]=1, [C]=3, [B]=3}
----L1----
{[A]=4, [H]=2, [E]=2, [C]=3, [B]=3}
----2nd scan----
----Candidate----
[[A, H], [A, E], [E, H], [A, C], [C, H], [C, E], [A, B], [B, H], [B, E],
[B, C]]
----C2----
{[A, H]=2, [C, E]=1, [B, E]=2, [A, E]=2, [B, C]=2, [A, C]=3, [A, B]=3, [E,
H]=1, [C, H]=1, [B, H]=1}
----L2----
{[A, H]=2, [B, E]=2, [A, E]=2, [B, C]=2, [A, C]=3, [A, B]=3}
----3th scan----
----Candidate----
[[A, E, H], [A, B, E], [B, C, E], [A, C, H], [A, C, E], [A, B, C], [A, B,
H]]
----C3----
{[A, B, E]=2, [A, B, C]=2}
----L3----
{[A, B, E]=2, [A, B, C]=2}
----4th scan----
----Candidate----
[[A, B, C, E]]
----C4----
{}
----L4----
{}
```

b) Which of the itemsets from a) are closed? Which of the itemsets from a) are maximal?

Closed itemsets from 1st scan: [F], [D] because their support is below 2.
Closed itemsets from 2nd scan: [C, E], [E, H], [C, H], [B, H] because their support is below 2.
Closed itemsets from 3rd scan:
[A, E, H]: L2 not contains [E, H]
[B, C, E]: L2 not contains [C, E]
[A, C, H]: L2 not contains [C, H]
[A, C, E]: L2 not contains [C, E]
[A, B, H]: L2 not contains [B, H]
Closed itemsets from 4th scan:
[A, B, C, E]: L3 not contains [B, C, E], [A, C, E]

Maximal itemsets: [A, B, E]=2, [A, B, C]=2

c) For all frequent itemsets of maximal length, list all corresponding association rules satisfying the requirements on (minimum support and) minimum confidence together with their confidence.

For [A, B, E], support = 50%, 3^2-2=6 association rules, minimum confidence = 60%
[A]->[B, E] confidence = support([A, B, E]) / support([A]) = 50%/100% = 50% not satisfied
[B]->[A, E] confidence = support([A, B, E]) / support([B]) = 50%/75% = 67% satisfied
[E]->[A, B] confidence = support([A, B, E]) / support([E]) = 50%/50% = 100% satisfied
[A, B]->[E] confidence = support([A, B, E]) / support([A, B]) = 50%/75% = 67% satisfied
[A, E]->[B] confidence = support([A, B, E]) / support([A, E]) = 50%/50% = 100% satisfied
[B, E]->[A] confidence = support([A, B, E]) / support([B, E]) = 50%/50% = 100% satisfied

For [A, B, C], support = 50%, 3^2-2=6 association rules, minimum confidence = 60%
[A]->[B, C] confidence = support([A, B, C]) / support([A]) = 50%/100% = 50% not satisfied
[B]->[A, C] confidence = support([A, B, C]) / support([B]) = 50%/75% = 67% satisfied
[C]->[A, B] confidence = support([A, B, C]) / support([C]) = 50%/75% = 67% satisfied
[A, B]->[C] confidence = support([A, B, C]) / support([A, B]) = 50%/75% = 67% satisfied
[A, C]->[B] confidence = support([A, B, C]) / support([A, C]) = 50%/75% = 67% satisfied
[B, C]->[A] confidence = support([A, B, C]) / support([B, C]) = 50%/50% = 100% satisfied

d) Using the lift of an association rule defined as follows to compute the lift for the association rules from c).

Lift = confidence / support(head)

I check Wikipedia and I found that

Confidence = support(all) / support(body)

Lift = support(all)/(support(body)*support(head))

For [A, B, E],

[B]->[A, E] confidence = 67%,

Lift = support([A, B, E]) / (support([B])*support([A, E])) = 50%/(75%*50%)=1.33

[E]->[A, B] confidence = 100%

Lift = support([A, B, E]) / (support([E])*support([A, B])) = 50%/(50%*75%) = 1.33

[A, B]->[E] confidence = 67%

Lift = support([A, B, E]) / (support([A, B])*support([E])) = 50%/(75%*50%) = 1.33

[A, E]->[B] confidence = 100%

Lift = support([A, B, E]) / (support([A, E])*support([B])) = 50%/(50%*75%) = 1.33

[B, E]->[A] confidence = 100%

Lift = support([A, B, E]) / (support([B, E])*support([A]))= 50%/(50%*100%) = 1

For [A, B, C]

[B]->[A, C] confidence = 67%

Lift = support([A, B, C]) / (support([B])*support([A, C])) = 50%/(75%*75%) = 0.89

[C]->[A, B] confidence = 67%

Lift = support([A, B, C]) / (support([C])*support([A, B])) = 50%/(75%*75%) = 0.89

[A, B]->[C] confidence = 67%

Lift = support([A, B, C]) / (support([A, B])*support([C])) = 50%/(75%*75%)= 0.89

[A, C]->[B] confidence = 67%

Lift = support([A, B, C]) / (support([A, C])*support([B])) = 50%/(75%*75%) = 0.89

[B, C]->[A] confidence = 100%

Lift = support([A, B, C]) / (support([B, C]) *support([A])) = 50%/(50%*100%) = 1

e) Why are only those association rules interesting that have a lift (significantly) larger than 1.0? [Hint: the following equations are true]
Lift = confidence / support(head)
= confidence / ( support(body) * support(head) / support(body) )
= confidence / expected_confidence

To have a large lift, expected confidence should be small which means support(head) should be small, while increasing or decreasing support(body) will not asymptotically increase lift. For example [B]->[A, E], small support(head) means that support([A,E]) is small. It means also that [A, E] appears not very often in the database, and therefore, a large proportion of the correlation of [A, E] is [B]. Also notice that the lift of [B]->[A, E] and [A, E]->[B] is the same because the formula involves both head and body. Lift is more like a dependency and correlation measurement between head and body. If we know [B], we are more likely to predict [A,E], vice versa.
Sometimes we may have a high confidence for a rule. However, it may have a low lift value because support(head) is large which means head has a lot other correlations in the database. Also, if rules with high confidence but low lift incorporate some elements from head to body, the new rules may have a higher lift.

**Thought**
I know I could do this problem by hand but I thought it would be a great practice for me to write some code. It reminds me a lot of about data structure like HashSet and HashTable in java. At first, I try to do all homework by coding. Later when I try to find the association rules from maximal itemsets, I find out that it is kind of tedious to find the subsets and link them together, and I also have to find a good data structure to represent it. Then I just decide to do it by hand. I can possibly make a Hashtable with key be the body and values be a List<Object> that contains head, confidence, and lift. The difficult part is to build the association rules by finding the subsets of maximal itemsets and to keep track of the support of itemsets in each scan.