**CS412: Introduction to Data Mining** **Summer 2016**

# Assignment 3

*Due: 07/07/2016 11:59pm*

## 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 in doing the homework. We are more concerned that you learn how to solve the problem than that you demonstrate that you solved it entirely on your own. You should, however, write down the solution yourself.
- Try to keep the solution brief and clear.
- 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 out a final result. Necessary calculation steps and reasoning are required.
- 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!**
- We will be using Compass for collecting the homework assignments. Please submit your answers via Compass (`http://compass2g.illinois.edu`). Please do NOT hand in a hard copy of your write-up. Contact the TAs if you are having technical difficulties in submitting the assignment.
- The homework **MUST** be submitted in **pdf** format. **Scanned handwritten and hand-drawn pictures** inside your documents **are not acceptable**. Answers to the written part and mini-MP should be included in one `.pdf` file.
- Please **DO NOT** zip the PDF file so that graders can access your PDF directly on Compass. You can compress other files into a single zip file. In summary, you need to submit one PDF file, named as `hw3.netid.pdf`, and one `.zip` file, named as `hw3.netid.zip`.

- If scripts are used to solve problems, you are required to submit the source code, and use the file names to identify the corresponding questions or sub-questions. For instance, `question1.netid.py` refers to the python source code for Question 1; and `question1a.netid.py` refers to the python source code for sub-question 1(a); replace `netid` with your netid. You can submit separate files for sub-questions or a single file for the entire question.

# Question 1 (12 points)

Based on the transactions in Table 1, answer the following questions.

**Purpose**

- Covering the basics concepts of frequent pattern mining.

| Transaction Number | Items |
|---|---|
| 1 | Beer, Cheese, Diaper |
| 2 | Beer, Diaper |
| 3 | Beer, Tylenol |
| 4 | Diaper, Tylenol |
| 5 | Cheese |
| 6 | Beer, Cheese, Diaper, Tylenol |

Table 1: Transactions records

a. (2, L1) What is the support count of {Beer}, {Diaper}, and {Beer, Tylenol}?

b. (2, L1) What is confidence({Beer} $\Rightarrow$ {Diaper})? confidence({Diaper} $\Rightarrow$ {Beer} )?

c. (2, L1) Is {Beer, Cheese, Diaper, Tylenol} closed? ($min\_sup = 1$) Please explain your answer.

d. (2, L1) Is {Beer, Cheese, Diaper} closed? ($min\_sup = 1$) Please explain your answer.

e. (2, L1) Is {Beer, Cheese, Diaper} a max pattern? ($min\_sup = 1$) Please explain your answer.

f. (2, L3) If an itemset is a max pattern does that mean it is also a closed pattern? If yes, explain why. If not, provide a counter example.

**Solution**

a. **4, 4, 2**

b. **3/4, 3/4**

c. **Yes. The items has no supersets.**

d. **Yes. None of its supersets have the same support count**.

e. **No. Its superset is also frequent.**

f. **Yes. For any max-pattern** $support(super(A))$**. Since A is frequent,** $suppport(A) \geq min\_sup$**. So** $support(super(A)) < min\_sup <\leq support(A)$**. So** $support(super(A)) < support(A)$**. So A is a closed pattern.**

# Question 2 (24 points)

Based on Table 2, use the Apriori algorithm to find the frequent patterns with $min\_sup = 3$.
**Purpose**

- Get a better understanding of the Apriori algorithm.

**Requirements**

- Provide all the intermediate steps and explain how you calculated the final values. No programming is needed. Please keep it brief and clear.
- Please use abbreviations ($C1, L1...$) in your work to specify which list you are writing about.
- You can use B, C, D, E, M, and T in place of Beer, Cheese, Diaper, Eggs, Milk, and Tylenol, respectively.

| Transaction Number | Items |
|---|---|
| 1 | Beer, Cheese, Diaper, Eggs, Milk |
| 2 | Beer, Diaper, Eggs |
| 3 | Beer, Milk, Tylenol |
| 4 | Beer, Diaper, Milk, Tylenol |
| 5 | Cheese, Eggs |
| 6 | Beer, Cheese, Diaper, Milk, Tylenol |

Table 2: Transactions records

a. (6, L2) List all candidate 1-itemsets (C1). List all frequent 1-itemsets (L1). $min\_sup = 3$.

b. (8, L2) List all candidate 2-itemsets (C2). Please include the number of itemsets in C2. List all frequent 2-itemsets (L2). $min\_sup = 3$.

c. (6, L2) List all candidate 3-itemsets (C3). List all frequent 2-itemsets (L3). $min\_sup = 3$.

d. (4, L1) Is there at least one frequent 4-itemset? $min\_sup = 3$.

**Solution**

a. $C1 = L1 = \{B = 5, C = 3, D = 4, E = 3, M = 4, T = 4\}$

b. $C2 = \{BC = 2, BD = 4, BE = 2, BM = 4, BT = 3, CD = 2, CE = 2, CM = 2, CT = 1, DE = 2, DM = 3, DT = 2, EM = 1, ET = 0, MT = 3\}$, $L2 = \{BD = 4, BM = 4, BT = 3, DM = 3, MT = 3\}$ C2 count is 15.

c. $C3 = \{BDM = 3, BMT = 3\}$, $L2 = \{BDM = 3, BMT = 3\}$

d. No, count of BDMT is 2.

# Question 3 (26 points)

Based on Table 3, use the FP Growth algorithm with $min\_sup = 3$ to find frequent patterns.
**Purpose**

- Get a better understanding of FP-Growth algorithm

**Requirements**

- You are required to generate tables and figures for this problem. You can use any software to do that. **For only sub-question b**, you can draw the Header Table and FP-Tree by hand and scan or take a picture to include in the final submission.
- Please make sure everything is **clearly** visible in the pdf file - we will not hesitate to give a 0 if the image is not legible.
- For sub-question b, position the Header Table and FP-tree side-by-side, with the Header Table on the left.
- You can use A, B, C, D, M, T, and Y in place of Apple, Beer, Cheese, Diaper, Milk, Tylenol, and Yogurt, respectively.

| Transaction Number | Items |
|---|---|
| 1 | Apple, Beer, Cheese, Diaper |
| 2 | Beer, Diaper |
| 3 | Apple, Beer, Tylenol |
| 4 | Diaper, Milk |
| 5 | Apple, Cheese |
| 6 | Apple, Beer, Diaper, Tylenol, Milk |
| 7 | Beer, Cheese, Yogurt |

Table 3: Transactions records

a. (2, L2) Generate an ordered list of frequent items based on the raw transaction database. To break ties, use the alphabetical order. $min\_sup = 3$

b. (5, L2) Generate a Header Table and FP-tree based on the frequent item list. Link nodes to the corresponding positions in the Header Table. $min\_sup = 3$

c. (5, L2) Generate Conditional Pattern Bases and Conditional FP-trees for items Apple and Cheese based on the FP-tree, and list the frequent patterns computed based on each of the Conditional FP-trees. (You only need to show the Conditional Pattern Bases and the frequent patterns.) $min\_sup = 3$

d. (5, L3) Why do we order the items in each transaction by their frequency before constructing the FP-tree? Hints: Think about the purpose of FP-tree and how this order will affects its structure.

e. (5, L3) Is FP Growth algorithm faster than the Apriori algorithm? Please justify your answer.

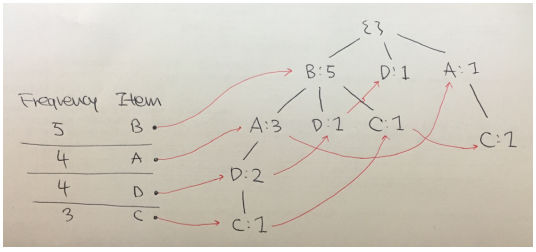
f. (4, L3) Is constructing a FP-tree with a transaction dataset a lossless compression technique? Please justify your answer.

**Solution**

a. $B = 5$, $A = 4$, $D = 4$, $C = 3$

| Transaction Number | Items | Ordered frequent items |
|---|---|---|
| 1 | Apple, Beer, Cheese, Diaper | B, A, D, C |
| 2 | Beer, Diaper | B, D |
| 3 | Apple, Beer, Tylenol | B, A |
| 4 | Diaper, Milk | D |
| 5 | Apple, Cheese | A, C |
| 6 | Apple, Beer, Diaper, Tylenol, Milk | B, A, D |
| 7 | Beer, Cheese, Yogurt | B, C |

Table 4: Transactions records




b.

| Item | Cond. Pattern Base | Frequent Patterns |
|---|---|---|
| Apple | B:3 | A, AB |
| Cheese | BAD: 1, B: 1, A: 1 | C |

Table 5: Transactions records

c.

d. We want to use FP-tree as a compression of the original transaction database. Ordering the items by their frequency will lead to more shared nodes and therefore smaller FP-trees.

e. Yes. No candidate generation, compact data structure, no repeated db scans.

f. Yes. No information is lost (for frequent patterns) when the tree is constructed. Redundancy is reduced through creation of the tree. We will also accept answers that say No since the infrequent patterns are lost. However if the answer does not have a valid explanation you will lose points.

# Question 4 (13 points)

A database with 150 transactions has its FP-tree shown in Figure 1. Let the **relative** $min\_support = 0.5$.

**Purpose**

- Get a better understanding of FP-trees.

**Requirements**

- Provide all the intermediate steps and explain how you calculated the final values. Please keep it brief and clear.

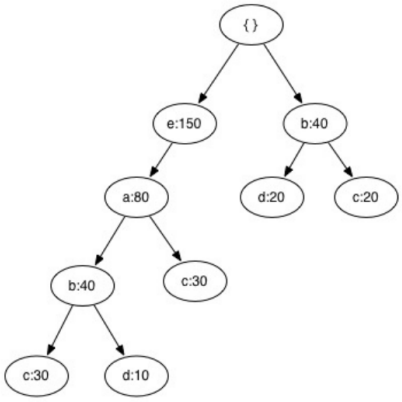



Figure 1: FP tree of a transaction DB

a. (6, L2) Show d's conditional (i.e., projected) database. **Relative** $min\_support = 0.5$

b. (7, L3) Present all frequent 3-itemsets and 2-itemsets. **Relative** $min\_support = 0.5$

**Solution**

a. eab: 10, b: 20

b. ae:80