Big Data Analytics: HW#2

By J. H. Wang

Oct. 24, 2023

Homework #2: Frequent Pattern Mining

- Chap.6:
  - 6.3
  - 6.6
  - 6.14
- Programming exercises:
  - *6.7(a)(b)
  - *6.15
- * denotes the two programming exercises where each team must implement at least one among the two exercises
- Due: 2 weeks (Nov. 7, 2023)

Exercises for Chap.6

- 6.3: (**30pt**) The Apriori algorithm makes use of *prior knowledge* of subset support properties.
- (a) (5pt) Prove that all nonempty subsets of a frequent itemset must also be frequent.
- (b) (5pt) Prove that the support of any nonempty subset *s*' of itemset *s* must be at least as great as the support of *s*.
- (c) (10pt) Given frequent itemset *l* and subset *s* of *l*, prove that the confidence of the rule "*s*' => (*l* - *s*')" cannot be more than the confidence of "*s*=>(*l* –*s*)", where *s*' is a subset of *s*.
- (d) (10pt) A *partitioning* variation of Apriori subdivides the transactions of a database *D* into *n* nonoverlapping partitions. Prove that any itemset that is frequent in *D* must be frequent in at least one partition of *D*.

- 6.6: (**30 pt**) A database has six transactions. Let min_sup=60% and min_conf=80%.

TID	Items bought
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y}
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I, E}

- (a) (15pt) Find all frequent itemsets using Apriori and FP-growth, respectively. Compare the efficiency of the two mining processes.
- (b) (15pt) List all the *strong* association rules (with support *s* and confidence *c*) matching the following metarule, where *X* is a variable representing customers and $item_i$ denotes variables representing items (e.g., *"A," "B,"*):
  $\forall X \in$ transactions, $buys(X,item_1) \cap buys(X,item_2) \Rightarrow buys(X,item_3)$ [*s, c*]

- 6.14: (**40pt**) The following contingency table summarizes supermarket transaction data, where hot dogs refers to the transactions containing hot dogs, !(hot dogs) refers to the transactions that do not contain hot dogs, hamburgers refers to the transactions containing hamburgers, !(hamburgers) refers to the transactions that do not contain hamburgers.

	hot dogs	!(hot dogs)	total
hamburgers	2000	500	2500
!(hamburgers)	1000	1500	2500
Total	3000	2000	5000

- (a) (10pt) suppose that the association rule “hot dogs => hamburgers” is mined. Given a minimum support threshold of 25% and a minimum confidence threshold of 50%, is this association rule strong?
- (b) (10pt) Based on the given data, is the purchase of hot dogs independent of the purchase of hamburgers? If not, what kind of correlation relationship exists between the two?
- (c) (20pt) Compare the use of the all_confidence, max_confidence, Kulczynski, and cosine measures with lift and correlation on the given data.

Programming Projects

- *6.7: (**50pt**) Using a programming language that you are familiar with, such as C++ or Java, implement the following frequent itemset mining algorithms introduced in this chapter:

(a) (20pt) Apriori,

(b) (30pt) FP-Growth.

Compare the performance of each algorithm with various kinds of large data sets.

- *6.15: (**50pt**) The DBLP data set (*www.informatik.uni-trier.de/~ley/db/*) consists of over one million entries of research papers published in computer science conferences and journals. Among these entries, there are a good number of authors that have coauthor relationships.
  (a) (30pt) Propose a method to efficiently mine a set of coauthor relationships that are closely correlated (e.g. often coauthoring papers together).
  (b) (20pt) Based on the mining results and the pattern evaluation measures discussed in this chapter, discuss which measure may convincingly uncover close collaboration patterns better than others.

Note on Programming Exercises

- Implementation can be done as a team (<= four persons per team)
  - Please remember to register your team members to the TA *as soon as possible*
- Each team has to complete at least one programming exercise
  - If you complete more programming projects, you will get extra bonus
- You can use any programming language to implement
  - You must provide detailed documentation on the necessary packages or libraries you used, where to download, how to install and configure

Homework Submission

- For hand-written exercises, please hand in your homework in class (paper version)
  - Remember to write your student ID

- For programming projects, please submit a compressed file containing your source codes, sample input and generated output, and documentation on how to compile, install, or configure the environment.

Thanks for Your Attention!