CMPT459 Spring 2020
Data Mining
Martin Ester
TAs: Ruijia Mao and Ruchita Rozario

Solution of Assignment 4

Solution posted and discussed in class: April 9, 2020

Assignment 4.1

Consider the following transaction database:

TransID	Items
ID1	1, 2, 3, 4
ID2	1, 3, 4, 6, 8
ID3	1, 3, 4, 5, 6, 8, 9
ID4	2, 4, 6, 8, 9
ID5	3, 4, 7, 8, 9

Suppose that minimum support is set to 60% and minimum confidence to 75%.

a) List all frequent itemsets together with their support.

1 60% 3 80% 4 100% 6 60% 8 80% 9 60%

1,3 60% 1,4 60% 3,4 80% 3,8 60% 4,6 60% 4,8 80% 4,9 60%
6,8 60% 8,9 60%

1,3,4 60% 3,4,8 60% 4,6,8 60% 4,8,9 60%

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

Closed: 4 3,4 4,8 1,3,4 3,4,8 4,6,8 4,8,9

Maximal: 1,3,4 3,4,8 4,6,8 4,8,9

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 itemset 1,3,4

1 → 3,4	100%
3 → 1,4	75%
1,3→4	100%
1,4→3	100%
3,4→1	75%

For itemset 3,4,8

3 → 4,8	75%
8 → 3,4	75%
3,4→8	75%
3,8→4	100%
4,8→3	75%

For itemset 4,6,8

6 → 4,8	100%
8 → 4,6	75%
4,6→8	100%
4,8→6	75%
6,8→4	100%

For itemset 4,8,9

8 → 4,9	75%
9 → 4,8	100%
4,8→9	75%
4,9→8	100%
8,9→4	100%

Assignment 4.2

Mining frequent itemsets can be expensive. In a large, dynamic database of transactions, we can store the set of frequent itemsets and incrementally update that set upon arrival of a set of new transactions. Let *DB* denote the last state of our database and ΔDB a set of new transactions. The task is to incrementally determine the set of frequent itemsets in $DB \cup \Delta DB$ with respect to min-supp, without re-applying the Apriori-algorithm to the whole updated database $DB \cup \Delta DB$. More specifically, given the sets *DB* and ΔDB and the set of all frequent itemsets in *DB* together with their support, return the set of all frequent itemsets in $DB \cup \Delta DB$, without re-applying the Apriori-algorithm to the whole updated database $DB \cup \Delta DB$.

We can use a non-incremental implementation of the Apriori-algorithm

Apriori (S: set of transactions, min-supp: float)

that returns all itemsets that are frequent in S together with their support. Note that min-supp is a relative frequency threshold.

a) Prove the following property: If an itemset is not frequent in *DB* and not frequent in ΔDB, then it cannot be frequent in $DB \cup \Delta DB$.

Proof:

Let δ denote the (relative) minimum support threshold, and let *A* denote some itemset.

If *A* is not frequent in *DB*, then $\frac{|\{t \in DB \mid A \subseteq t\}|}{|DB|} < \delta$, i.e. $|\{t \in DB \mid A \subseteq t\}| < \delta |DB|$ (1).

If *A* is not frequent in ΔDB, then

$\frac{|\{t \in \Delta DB \mid A \subseteq t\}|}{|\Delta DB|} < \delta$, *i.e.* i.e. $|\{t \in \Delta DB \mid A \subseteq t\}| < \delta |\Delta DB|$ (2).

Using (1), (2), and the disjointness of DB and ΔDB we obtain

$$|\{t \in DB \cup \Delta DB \mid A \subseteq t\}| < \delta |DB| + \delta |\Delta DB| = \delta(|DB| + |\Delta DB|) \quad (3)$$

Since DB and ΔDB are disjoint, $|DB \cup \Delta DB| = |DB| + |\Delta DB|$ (4).

Using (3) and (4), we conclude

$$\frac{|\{t \in DB \cup \Delta DB \mid A \subseteq t\}|}{|DB \cup \Delta DB|} < \frac{\delta(|DB| + |\Delta DB|)}{|DB \cup \Delta DB|} = \frac{\delta(|DB| + |\Delta DB|)}{|DB| + |\Delta DB|} = \delta,$$

i.e. A is infrequent in $DB \cup \Delta DB$.

b) Based on the property that we have proven in a), as a first step, the incremental Apriori algorithm applies the non-incremental Apriori algorithm to ΔDB to determine the frequent itemsets in ΔDB and their support. After having performed this first step, for which itemsets do you need to count the support in DB, and for which itemsets do you need to count the support in ΔDB? Explain why the support counting is necessary.

We need to count the support in DB of all itemsets that are frequent in ΔDB but not in DB. The reason is that the support of these itemsets in DB was not returned by the Apriori algorithm, because they were infrequent in DB, but they may become frequent in $DB \cup \Delta DB$.

We need to count the support in ΔDB of all itemsets that are frequent in DB but not in ΔDB. The reason is that the support in ΔDB of these itemsets was not returned by the Apriori-algorithm, because they were infrequent in ΔDB, but these itemsets may become frequent in $DB \cup \Delta DB$.

Assignment 4.3

Consider the following bag of 1-dimensional data points {1, 2, 4, 8, 10, 12, 14, 14, 14, 14, 16}.

a) Apply the knn-distance-based algorithm for outlier detection, using k = 2. Report the outlier scores (2-nn distances) for all points.

Points	1	2	4	8	10	12	14	14	14	14	16
Outlier score	3	2	3	4	2	2	0	0	0	0	2

What points have the highest outlier score? 8

b) Apply the LOF algorithm for outlier detection with k = 2. For all points p, report the average reachability distance $AR_k(p)$ to its neighboring points, and the LOF value LOF(p). Use the following definitions to deal with the case of $AR_k = 0$:

$$\forall x, x > 0 : \frac{x}{0} = \infty, \text{ and } \frac{0}{0} = 1.$$

Points	1	2	4	8	10	12	14	14	14	14	16
AR_k	2.5	3	2.5	3.3	3	2	0	0	0	0	2
LOF values	.92	1.2	.92	1.36	1.20	∞	1	1	1	1	∞

What points have the highest outlier score? 12 and 16