Algorithms for Data Science
Exercise 1

Johanna Luz, Jan Polster, Richard Palme, Fouad Alkhoury

2023-November-08

QR-Code this week

Task 1: Number of Association Rules

Task: Prove that the # of assoc. rules is $3^d - 2^{d+1} + 1$, where $d = |I|$

Def: Association rule: $X \to Y$, where $X, Y \subseteq I$ are disjoint and non-empty

$$A = \{(X, Y) \subseteq I \times I \mid X, Y \text{ disjoint}\} \qquad \text{size} = 3^d$$

$$B = \{(X, Y) \subseteq I \times I \mid X = \emptyset\} \qquad \text{size} = 2^d$$

$$C = \{(X, Y) \subseteq I \times I \mid Y = \emptyset\} \qquad \text{size} = 2^d$$

$$\{\text{assoc. rules}\} = A \setminus (B \cup C)$$

We have counted the set $(\emptyset \to \emptyset)$ twice, so we need to add 1!

$$\implies \# \text{ of assoc. rules } = |A| - (|B| + |C|) + 1$$

$$|A| - (|B| + |C|) + 1 = 3^d - 2^d - 2^d + 1$$

Task 2: The Apriori Algorithm

TID	transaction
1	M, O, N, K, E, Y
2	D, O, N, K, E, Y
3	M, A, K, E
4	M, U, C, K, Y
5	C, O, K, E

freq. threshold: $t = 3$

Fix some order:
$M < O < N < K < E < Y < D < A < U < C$

$\mathcal{C}_1 = \{M, O, N, K, E, Y, D, A, U, C\}$

$\mathcal{F}_1 = \{M, O, K, E, Y\}$

$\mathcal{C}_2 = \{MO, MK, ME, MY, OK, OE, OY, KE, KY, EY\}$

$\mathcal{F}_2 = \{MK, OK, OE, KE, KY\}$

$\mathcal{C}_3 = \{OKE\}$

$\mathcal{F}_3 = \{OKE\}$

$\mathcal{C}_4 = \emptyset$

Note: $KEY \notin \mathcal{C}_3$ (Why?)

Task 3: Correctness of Apriori

Correct: Sound and complete
Sound: All returned answers are correct
Complete: All correct answers are returned

Soundness: Can we print an unfrequent item?
No: Before printing any itemset, we always check whether it is frequent.

Completeness: Assume there are frequent itemsets, that are not generated by our algorithm. Let Z be inclusion minimal in these sets.
W.l.o.g. $|Z| = k + 1 \geq 2$. If Z is just a single element set, it would have been added in the first step.
By minimality of Z, all k-subsets of Z are in $\mathcal{F}_k$.
Let $p \neq q$ be the two largest items in Z.
$\implies X := Z \setminus \{p\}$ and $Y := Z \setminus \{q\}$ are in $\mathcal{F}_k$.
$\implies Z$ is added to $\mathcal{C}_{k+1}$, then evaluated for frequency and added to $\mathcal{F}_{k+1}$

Task 3: Irredundancy of Apriori

Irredundancy: Assume that Z is generated more than once.
W.l.o.g. Z is a minimal itemset with this property.
W.l.o.g. $|Z| \geq 2$, as all single item sets are simply added ince in the first step, no possibility of multiple instances.
Let $p \neq q$ be the two largest items in Z.
By minimality of Z, both $X := Z \setminus \{p\}$ and $Y := Z \setminus \{q\}$ are generated exactly once.
$\Longrightarrow$ Z is added to $\mathcal{C}_{k+1}$ exactly once

An itemset is only added if we merge the two itemsets that start the same but end in two different biggest elements. We only merge these once. As the itemset cannto be created by any other means, we don't generate any redundancies.

Task 4: Complexity of Apriori

Task: Prove that Apriori runs in incremental polyn. time.

i) Before printing $\mathcal{F}_1$, we check the freq. of each item in I: $\mathcal{O}(|D||I|)$
ii) Between printing $\mathcal{F}_k$ and $\mathcal{F}_{k+1}$ we

1. use CandidateGeneration: $\mathcal{F}_k \to \mathcal{C}_{k+1}$
2. check for frequency

iii) After printing the last $\mathcal{F}_k$ we check once more for frequency

Task 4: Complexity of Apriori

CandidateGeneration

1: $\mathcal{C}_{k+1} = \emptyset$
2: **for all** $X, Y \in \mathcal{F}_k$ such that they differ only in their last elements
3: make a $(k+1)$-element set Z by concatenating the common $(k-1)$-prefix with the two differing elements according to the order
4: **if** all k-subsets of Z are in $\mathcal{F}_k$ **then** add Z to C_{k+1}
5: **return** $\mathcal{C}_{k+1}$

1.: for loop does $\mathcal{O}(|\mathcal{F}_k|^2)$ iterations,
line 3 is in $\mathcal{O}(|I|)$,as both X and Y are at most that big,
line 4 is done in $\mathcal{O}(|\mathcal{F}_k||I|^2)$.
Candidate Generation is done in polynomial time in size(D) and the sizes of the prior Frequent sets.
2.: Checking the frequency of each itemset in $\mathcal{C}_{k+1}$: $\mathcal{O}(|\mathcal{F}_k|^2|D||I|)$, which is also polynomial in size(D) and sizes of prior frequent sets.
$\implies$ Apriori runs in incremental poly. time!

Task 4: Complexity of Apriori

(i):

Before printing $\mathcal{F}_1$, we check the freq. of each item in I: $\mathcal{O}(|D||I|)$

(ii):

The number of iterations of "For all $X, Y \in \mathcal{F}_k$" is $\mathcal{O}(|\mathcal{F}_k|^2)$

Inside the loop: Checking whether all k-subsets of Z are in $\mathcal{F}_k$: $\mathcal{O}(|\mathcal{F}_k||I|^2)$

$\Longrightarrow$ CandidateGeneration has a runtime of $\mathcal{O}(|\mathcal{F}_k|^3|I|^2)$

Checking the frequency of each itemset in $\mathcal{C}_{k+1}$: $\mathcal{O}(|\mathcal{F}_k|^2|D||I|)$

$\Longrightarrow$ Runtime between printing $\mathcal{F}_k$ and $\mathcal{F}_{k+1}$: $\mathcal{O}(|\mathcal{F}_k|^2|I|^2(|\mathcal{F}_k| + |D|))$

(iii):

After printing the last set $\mathcal{F}_K$, we call CandidateGeneration: $\mathcal{O}(|\mathcal{F}_K|^3|I|^2)$

Check for frequency: $\mathcal{O}(|\mathcal{F}_K|^2|D||I|)$

Task 5: Rule Generation

Confidence: $c(X \to Y) = \frac{|D[X \cup Y]|}{|D[X]|}$

$min\_conf = 0.8$

$$\mathcal{F}_1 = \{M_3, O_3, K_5, E_4, Y_3\}$$
$$\mathcal{F}_2 = \{MK_3, OK_3, OE_3, KE_4, KY_3\}$$
$$\mathcal{F}_3 = \{OKE_3\}$$

$F_2 = MK$:
$c(M \to K) = 1 \implies$ **print** $M \to K$
$c(K \to M) = 0.6 < 0.8$
$F_2 = OK$:
$c(O \to K) = 1 \implies$ **print** $O \to K$
$c(K \to O) = 0.6 < 0.8$
$F_2 = OE$:
$c(O \to E) = 1 \implies$ **print** $O \to E$
$c(E \to O) = 0.75 < 0.8$
$F_2 = KE$:
$c(K \to E) = 0.8 \implies$ **print** $K \to E$
$c(E \to K) = 1 \implies$ **print** $E \to K$
$F_2 = KY$:
$c(K \to Y) = 0.6 < 0.8$
$c(Y \to K) = 1 \implies$ **print** $Y \to K$

Task 5: Rule Generation

$F_3 = OKE$:

$\mathcal{H}_1 = \emptyset$

$c(KE \to O) = 0.75 < 0.8$

$c(OE \to K) = 1 \quad \Longrightarrow \quad$ **print** $OE \to K$

$\mathcal{H}_1 = \{K\}$

$c(OK \to E) = 1 \quad \Longrightarrow \quad$ **print** $OK \to E$

$\mathcal{H}_1 = \{K, E\}$

GenerateRules($F_3, \mathcal{H}_1$):

- $\mathcal{H}_2 = \{KE\}$
- $c(O \to KE) = 1 \quad \Longrightarrow \quad$ **print** $O \to KE$
- GenerateRules($F_3, \mathcal{H}_2$):
  - Stop, since conclusions cannot have length 3