TD2: Frequent Itemsets Mining

Pablo Mollá Chárlez & Junjie Chen

January 28, 2025

Contents

1 Exercise 1

Let D_1 be a transactional database represented in the horizontal format H_{D_1} as follows:

Trans.	**Items**					
t_1		B	C	D		
t_2	A	B	C		E	
t_3	A	B	C	D		F
t_4				D	E	
t_5	A	B				
t_6	A		C		E	F
t_7	A	B			E	F
t_8				D		F
t_9			C		E	
t_{10}	A	B				F

Figure 1: Transactional Database D_1

- **Question 1:** Provide the vertical representation VD1 and the matrix representation M_{D_1} of D_1.

  The vertical representation V_{D_1} is as follows:

$$V_{D_1} = \begin{pmatrix} A & B & C & D & E & F \\ t_2 & t_1 & t_1 & t_1 & t_2 & t_3 \\ t_3 & t_2 & t_2 & t_3 & t_4 & t_6 \\ t_5 & t_3 & t_3 & t_4 & t_6 & t_7 \\ t_6 & t_5 & t_6 & t_8 & t_7 & t_8 \\ t_7 & t_7 & t_9 & 0 & t_9 & t_{10} \\ t_{10} & t_{10} & 0 & 0 & 0 & 0 \end{pmatrix}$$

  The matrix representation M_{D_1} is as follows:

$$M_{D_1} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- **Question 2:** Calculate the support, absolute frequency, and relative frequency of the following:

$$L = \{ACD,\ CE,\ BCE,\ ABCE,\ E,\ D,\ BC,\ F,\ CDF,\ EF\}$$

  The support or cover in this case is equivalent to the absolute frequency (Absolute_Frequency(Itemset) = cover(Itemset)) of the itemsets, and the relative frequency is given by:

$$\text{Relative\_Frequency(Itemset)} = \frac{\text{cover(Itemset)}}{\text{|D|}} \text{ where cover(Itemset)} = \text{support(Itemset)}.$$

From the exercise we know that $D = \{t_1, t_2, \dots, t_{10}\}$, therefore $|D| = 10$. Then, we just need to determine the cover of each itemset in L.

$$\text{cover(ACD)} = |\{t_3\}| = 1 \implies \text{Relative\_Frequency(ACD)} = \frac{\text{cover(ACD)}}{|\text{D}|} = \frac{1}{10}$$

$$\text{cover(CE)} = |\{t_2, t_6, t_9\}| = 3 \implies \text{Relative\_Frequency(CE)} = \frac{\text{cover(CE)}}{|\text{D}|} = \frac{3}{10}$$

$$\text{cover(BCE)} = |\{t_2\}| = 1 \implies \text{Relative\_Frequency(BCE)} = \frac{\text{cover(BCE)}}{|\text{D}|} = \frac{1}{10}$$

$$\text{cover(ABCE)} = |\{t_2\}| = 1 \implies \text{Relative\_Frequency(ABCE)} = \frac{\text{cover(ABCE)}}{|\text{D}|} = \frac{1}{10}$$

$$\text{cover(E)} = |\{t_2, t_4, t_6, t_7, t_9\}| = 5 \implies \text{Relative\_Frequency(E)} = \frac{\text{cover(E)}}{|\text{D}|} = \frac{1}{2}$$

$$\text{cover(D)} = |\{t_1, t_3, t_4, t_8\}| = 4 \implies \text{Relative\_Frequency(D)} = \frac{\text{cover(D)}}{|\text{D}|} = \frac{2}{5}$$

$$\text{cover(BC)} = |\{t_1, t_2, t_3\}| = 3 \implies \text{Relative\_Frequency(BC)} = \frac{\text{cover(BC)}}{|\text{D}|} = \frac{3}{10}$$

$$\text{cover(F)} = |\{t_3, t_6, t_7, t_8, t_10\}| = 5 \implies \text{Relative\_Frequency(F)} = \frac{\text{cover(F)}}{|\text{D}|} = \frac{1}{2}$$

$$\text{cover(CDF)} = |\{t_3\}| = 1 \implies \text{Relative\_Frequency(CDF)} = \frac{\text{cover(CDF)}}{|\text{D}|} = \frac{1}{10}$$

$$\text{cover(EF)} = |\{t_6, t_7\}| = 2 \implies \text{Relative\_Frequency(EF)} = \frac{\text{cover(EF)}}{|\text{D}|} = \frac{1}{5}$$

- **Question 3:** Identify the frequent itemsets with minimum support values $\alpha \in \{5, 6, 7, 8, 9, 10\}$.

  Let's determine the cover for each individual itemset and then proceed combining the rest of them that satisfy the fact that their support value $\alpha \in \{5, 6, 7, 8, 9, 10\}$.

  - $\text{cover(A)} = |\{t_2, t_3, t_5, t_6, t_7, t_{10}\}| = 6$ ✓
  - $\text{cover(B)} = |\{t_1, t_2, t_3, t_5, t_7, t_{10}\}| = 6$ ✓
  - $\text{cover(C)} = |\{t_1, t_2, t_3, t_6, t_9\}| = 5$ ✓
  - $\text{cover(D)} = |\{t_1, t_3, t_4, t_8\}| = 4$ ✗
  - $\text{cover(E)} = |\{t_2, t_4, t_6, t_7, t_9\}| = 5$ ✓
  - $\text{cover(F)} = |\{t_3, t_6, t_7, t_8, t_{10}\}| = 5$ ✓

  Now, let's analyze the possible pairs of itemsets in lexical order:

  - $\text{cover(AB)} = |\{t_2, t_3, t_5, t_7, t_{10}\}| = 5$ ✓
  - $\text{cover(AC)} = |\{t_2, t_3, t_6\}| = 3$ ✗
  - $\text{cover(AE)} = |\{t_2, t_6, t_7\}| = 3$ ✗
  - $\text{cover(AF)} = |\{t_3, t_6, t_7, t_{10}\}| = 4$ ✗

  - $\text{cover(BC)} = |\{t_1, t_2, t_3\}| = 3$ ✗
  - $\text{cover(BE)} = |\{t_2, t_7\}| = 2$ ✗
  - $\text{cover(BF)} = |\{t_3, t_7, t_{10}\}| = 3$ ✗

  - $\text{cover(CE)} = |\{t_2, t_6, t_9\}| = 3$ ✗
  - $\text{cover(CF)} = |\{t_3, t_6\}| = 2$ ✗

  - $\text{cover(EF)} = |\{t_6, t_7\}| = 2$ ✗

- **Question 4:** Provide an example of two comparable itemsets and two non-comparable itemsets.

  Let's start by defining what are **comparable itemsets and non-comparable itemsets**:

  - Comparable itemsets are two itemsets X, Y that satisfy: $\boxed{X \subseteq Y \vee Y \subseteq X}$
    For instance, the itemsets A, B and AB, satisfy that $A \subseteq AB$ and $B \subseteq AB$, therefore they are comparable.
  - Non-Comparable itemsets are two itemsets X, Y that satisfy: $\boxed{X \nsubseteq Y \wedge Y \nsubseteq X}$
    For example, the itemsets A and BC are not comparable because $A \nsubseteq BC \wedge BC \nsubseteq A$.

2 Exercise 2

- **Question 1:** Write a proof for the anti-monotone property of frequent itemsets.

  Let's first define the mentioned property before proving it. The anti-monotone property states that if an itemset is frequent, then all of its subsets must also be frequent. However, if an itemset is not frequent, then all of its supersets must also be not frequent.

  Proof

  Let's assume that an itemset X is frequent, meaning that its support is greater than or equal to a given minimum support threshold α and a a subset $Y \subseteq X$ (a subset of itemset X). Since the support of a set is the number of transactions that contain the set, every transaction that contains Y must also contain X (because $Y \subseteq X$). Therefore, the support of Y cannot be greater than the support of X, and it follows that:

  $$\text{Support}(Y) \geq \text{Support}(X)$$

  which implies that Y must also be frequent.

  On the other hand, if X is not frequent, then there exists no transaction that contains X with sufficient support. Hence, any superset of X, say $Z \supseteq X$, will also have even fewer transactions supporting it than X, making it not frequent. Thus, the anti-monotonic property is proved: if an itemset is frequent, all of its subsets are frequent; if an itemset is not frequent, all of its supersets are also not frequent.

- **Question 2:** Write a proof for the Apriori property.

  As we previously, let's first start defining the Apriori property. The Apriori property is a direct consequence of the anti-monotone property. It states that if an itemset is frequent, then all of its subsets must also be frequent. In other words, frequent itemsets are closed under subsets.

  Proof

  Let's assume that X is an itemset of size k and that X is frequent, meaning that the support of X is greater than or equal to the minimum support threshold α. According to the anti-monotone property, all subsets of X that have fewer than k items must also be frequent. Therefore, the subsets of X of size $k-1$ must also be frequent.

  Now, if any subset of X of size $k-1$ is not frequent, we can conclude that the itemset X, which includes this subset, cannot be frequent either. This implies that the algorithm can prune such itemsets from the search space, significantly reducing the computational complexity. Thus, the Apriori property allows us to prune non-frequent itemsets efficiently. Specifically, if an itemset is frequent, all of its subsets must also be frequent; if any subset of an itemset is not frequent, the itemset itself cannot be frequent.

  In conclusion, the Apriori property ensures that we can build frequent itemsets efficiently by leveraging the anti-monotone property and pruning non-frequent subsets early in the process.

3 Exercise 3

Let D_2 be a transactional database as follows:

Trans.	Items				
t_1	A		C	D	
t_2		B	C		E
t_3	A	B	C		E
t_4		B			E
t_5	A	B	C		E
t_6		B	C		E

Figure 2: Transactional Database D_2

- **Question 1:** Run the Apriori algorithm on D_2 with a minimum support $\alpha = 3$, without using the canonical operator κ.

  The Apriori algorithm involves generating itemsets of increasing size and filtering out those that do not meet the minimum support threshold. Let's follow step by step the algorithm implementation:

  - **Step 1: Initialize the set of candidates** $(1 - itemsets)$

    First, we need to find the $1 - itemsets$ (i.e., individual items) and their frequencies (support/cover). From the transactions:

    - A: Appears in $t_1, t_3, t_5 \longrightarrow$ Support $= 3$ ✓
    - B: Appears in $t_2, t_3, t_4, t_5, t_6 \longrightarrow$ Support $= 5$ ✓
    - C: Appears in $t_1, t_2, t_3, t_5, t_6 \longrightarrow$ Support $= 5$ ✓
    - D: Appears in $t_1 \longrightarrow$ Support $= 1$ ✗
    - E: Appears in $t_2, t_3, t_4, t_5, t_6 \longrightarrow$ Support $= 5$ ✓

  - **Step 2: Prune the** $1 - itemsets$

    The minimum support is 3, so we prune itemsets that have support less than 3:

    - Keep A, B, C, E (since their supports are ≥ 3)
    - Discard D (since its support is 1)

  - **Step 3: Generate** $2 - itemsets$ **and count their supports**

    We generate all possible 2-itemsets from the frequent 1-itemsets A, B, C, E:

    - AB: Appears in $t_3, t_5 \longrightarrow$ Support $= 2$ ✗
    - AC: Appears in $t_1, t_3, t_5 \longrightarrow$ Support $= 3$ ✓
    - AE: Appears in $t_3, t_5 \longrightarrow$ Support $= 2$ ✗
    - BC: Appears in $t_2, t_3, t_5, t_6 \longrightarrow$ Support $= 4$ ✓
    - BE: Appears in $t_2, t_3, t_4, t_5, t_6 \longrightarrow$ Support $= 5$ ✓
    - CE: Appears in $t_2, t_3, t_5, t_6 \longrightarrow$ Support $= 4$ ✓

  - **Step 4: Prune the** $2 - itemsets$

    We prune $2 - itemsets$ with support less than 3, therefore we keep $(A, C), (B, C), (B, E), (C, E)$.

  - **Step 5: Generate 3-itemsets and count their supports**

    We generate all possible $3 - itemsets$ from the frequent $2 - itemsets$:

    - ABC: Appears in $t_3, t_5 \longrightarrow$ Support $= 2$ ✗

    - ABE: Appears in $t_3, t_5 \longrightarrow$ Support $= 2$ ✗
    - ACE: Appears in $t_3, t_5 \longrightarrow$ Support $= 2$ ✗
    - BCE: Appears in $t_2, t_3, t_5, t_6 \longrightarrow$ Support $= 4$ ✓

  - **Step 6: Prune the 3-itemsets**

    Once again, we prune $3 - itemsets$ with support less than 3 and therefore we keep (B, C, E).

  - **Step 7: Stop**

    There are no frequent itemsets of size 4 or greater. Thus, the frequent itemsets are:

    - **1-itemsets:** $\{A, B, C, E\}$
    - **2-itemsets:** $\{AC, BC, BE, CE\}$
    - **3-itemset:** $\{BCE\}$

- **Question 2:** Run the Apriori algorithm on $D2$ with a minimum support $\alpha = 3$, using the child operator based on a lexicographical order **lex**.

  The key difference in this case is that we will generate itemsets based on lexicographical order, rather than considering all combinations.

  - **Step 1: Initialize the set of candidates** $(1 - itemsets)$

    The list of items sorted lexicographically: A, B, C, D, E. From the transactions:

    - A: Appears in $t_1, t_3, t_5 \longrightarrow$ Support $= 3$ ✓
    - B: Appears in $t_2, t_3, t_4, t_5, t_6 \longrightarrow$ Support $= 5$ ✓
    - C: Appears in $t_1, t_2, t_3, t_5, t_6 \longrightarrow$ Support $= 5$ ✓
    - D: Appears in $t_1 \longrightarrow$ Support $= 1$ ✗
    - E: Appears in $t_2, t_3, t_4, t_5, t_6 \longrightarrow$ Support $= 5$ ✓

  - **Step 2: Prune the** $1 - itemsets$

    The minimum support is 3, so we prune itemsets that have support less than 3:

    - Keep A, B, C, E
    - Discard D

  - **Step 3: Generate** $2 - itemsets$ **in lexicographical order**

    The $2 - itemsets$, sorted lexicographically, are:

    - AB: Appears in $t_3, t_5 \longrightarrow$ Support $= 2$ ✗
    - AC: Appears in $t_1, t_3, t_5 \longrightarrow$ Support $= 3$ ✓
    - AE: Appears in $t_3, t_5 \longrightarrow$ Support $= 2$ ✗
    - BC: Appears in $t_2, t_3, t_5, t_6 \longrightarrow$ Support $= 4$ ✓
    - BE: Appears in $t_2, t_3, t_4, t_5, t_6 \longrightarrow$ Support $= 5$ ✓
    - CE: Appears in $t_2, t_3, t_5, t_6 \longrightarrow$ Support $= 4$ ✓

  - **Step 4: Prune the** $2 - itemsets$

    We, prune $2 - itemsets$ with support less than 3, threfore we keep $\{AC, BC, BE, CE\}$.

  - **Step 5: Generate** $3 - itemsets$ **in lexicographical order**

    The $3 - itemsets$, sorted lexicographically, are:

    - ABC: Appears in $t_3, t_5 \longrightarrow$ Support $= 2$ ✗
    - ABE: Appears in $t_3, t_5 \longrightarrow$ Support $= 2$ ✗
    - ACE: Appears in $t_3, t_5 \longrightarrow$ Support $= 2$ ✗
    - BCE: Appears in $t_2, t_3, t_5, t_6 \longrightarrow$ Support $= 4$ ✓

  - **Step 6: Prune the** $3-itemsets$

    We, prune $3-itemsets$ with support less than 3, therefore we keep (B, C, E).

  - **Step 7: Stop**

    There are no frequent itemsets of size 4 or greater. Thus, the frequent itemsets are:

    * **1-itemsets:** $\{A, B, C, E\}$
    * **2-itemsets:** $\{AC, BC, BE, CE\}$
    * **3-itemset:** $\{BCE\}$

  Conclusion: In both cases (without using the canonical operator and using the lexicographical operator), the frequent itemsets found are the same.

- **Question 3:** Implement the Apriori algorithm in Java with and without the **child+lex** operator. Compare the performance of the two versions on the datasets provided in **.\DataSets\\.**

- **Question 4:** Propose an algorithm with a bottom-up exploration approach to extract the set of frequent itemsets. Implement it and compare its performance with the Apriori algorithm.

- **Question 5:** Revise the Apriori algorithm to extract only frequent itemsets with a size greater than a specified value **size**. Implement this modified version.

  For questions from 3 to 5, the corresponding answers are in the University GitHub Repository or Personal GitHub Repository, therefore we will include the explanation of the different folders and code implementations. This project implements the Apriori algorithm and a bottom-up approach for frequent itemset mining. The project is organized into several directories and files that handle different aspects of the project.

  - **Database:** Variables are created here to store the transactions. The transactions are represented using Sets of Strings, where each set contains items present in a transaction.
  - **Folders and Files Overview:**
    * **test/java/AprioriTest:** This is a unit test class designed to test the core functionalities of the Apriori algorithm. It ensures that the standard and optimized Apriori algorithms are running correctly by verifying their outputs.
    * **main/Apriori.java:** Contains the core implementation of the Apriori algorithm. It includes two versions of the Apriori algorithm:
      · The standard version of Apriori.
      · The optimized version using child, order, and lex operators.

      This file is used for **Question 1** and **Question 2** to compute frequent itemsets. Also includes the comparePerformance function, which compares the runtime performance of the standard and optimized algorithms for **Question 3**.
    * **main/BottomUp.java:**
      · Implements the Bottom-Up algorithm, encapsulating the logic for this alternative frequent itemset mining approach.
      · Also includes a comparePerformance function to compare the performance of the Bottom-Up algorithm with the Apriori algorithm for Question 4.
    * **main/AprioriQ5.java:**
      · A modified version of the Apriori algorithm specifically designed for Question 5.
      · The file includes a key enhancement: the filterCandidates function, which filters frequent itemsets to keep only those with a size greater than a predefined threshold (minSize). This allows for the extraction of larger frequent itemsets.

  This structure is designed to modularize the implementation of the Apriori algorithm and related approaches, making it easier to test, compare, and extend the functionality for different questions and tasks.

4 Exercise 4

- **Question 1:** Let the set of maximal itemsets M_α be as follows:

$$M_\alpha = \{ABC^3,\ DE^2,\ EF^5\}$$

Provide the list of frequent itemsets.

We are given the set of maximal itemsets M_α, and we know by definition that a maximal itemset is a frequent itemset which has no frequent supersets, and by the Apriori property, we also know that any subset of a maximal itemset is also frequent, therefore from M_α, we can deduce the frequent itemsets:

  - **From ABC^3:** All subsets of ABC with support ≥ 3 are frequent: ABC, AB, AC, BC, A, B, C.
  - **From DE^2:** Since the support threshold for DE is 2, we only include subsets of DE with support ≥ 2: DE, D, E.
  - **From EF^5:** All subsets of EF with support ≥ 5 are frequent: EF, E, F.

Consequently, the final list of frequent itemsets is:

$$\{ABC, AB, AC, BC, A, B, C, DE, D, E, EF, F\}.$$

- **Question 2:** Let the set of closed itemsets C_α be as follows:

$$C_\alpha = \{ABC^3, ABE^5, DE^2, EF^5\}$$

Provide the list of frequent itemsets.

We are given the set of closed itemsets C_α, and we know by definition that a closed itemset is a frequen itemset for which no proper superset has the same support, and applying once again the Apriori property, we know that all subsets of a closed itemset are also frequent. Using C_α, we can determine all frequent itemsets:

  - **From ABC^3:** All subsets of ABC with support ≥ 3 are frequent: ABC, AB, AC, BC, A, B, C.
  - **From ABE^5:** All subsets of ABE with support ≥ 5 are frequent: ABE, AB, AE, BE, A, B, E.
  - **From DE^2:** All subsets of DE with support ≥ 2 are frequent: DE, D, E.
  - **From EF^5:** All subsets of EF with support ≥ 5 are frequent: EF, E, F.

Consequently, the final list of frequent itemsets is:

$$\{ABC, AB, AC, BC, A, B, C, ABE, AE, BE, DE, D, E, EF, F\}.$$

- **Question 3:** Consider now the transactional database D_2. Determine the sets of maximal and closed frequent itemsets with a minimum support $\alpha = 3$.

Trans.	Items				
t_1	A		C	D	
t_2		B	C		E
t_3	A	B	C		E
t_4		B			E
t_5	A	B	C		E
t_6		B	C		E

Figure 3: Transactional Database D_2

Let's recall that the dataset D_2 above and then carefully determine the frequent itemsets based on the provided transactions:

- **Step 1: Frequency of Single Items** We calculate the frequency (support) of each single item:
  - A: Appears in $t_1, t_3, t_5 \longrightarrow$ Support = 3. ✓
  - B: Appears in $t_2, t_3, t_4, t_5, t_6 \longrightarrow$ Support = 5. ✓
  - C: Appears in $t_1, t_2, t_3, t_5, t_6 \longrightarrow$ Support = 5. ✓
  - D: Appears in $t_1 \longrightarrow$ Support = 1. ✗
  - E: Appears in $t_2, t_3, t_4, t_5, t_6 \longrightarrow$ Support = 5. ✓
- **Step 2: Frequency of Itemsets (Size 2)** Next, calculate the frequency of itemsets of size 2:
  - AB: Appears in $t_3, t_5 \longrightarrow$ Support = 2. ✗
  - AC: Appears in $t_1, t_3, t_5 \longrightarrow$ Support = 3. ✓
  - BC: Appears in $t_2, t_3, t_5, t_6 \longrightarrow$ Support = 4. ✓
  - BE: Appears in $t_2, t_3, t_4, t_5, t_6 \longrightarrow$ Support = 5. ✓
  - CE: Appears in $t_2, t_3, t_5, t_6 \longrightarrow$ Support = 4. ✓
- **Step 3: Frequency of Itemsets (Size 3)** Now calculate the frequency of itemsets of size 3:
  - ABC: Appears in $t_3, t_5 \longrightarrow$ Support = 2. ✗
  - BCE: Appears in $t_2, t_3, t_5, t_6 \longrightarrow$ Support = 4. ✓
  - ACE: Appears in $t_3, t_5 \longrightarrow$ Support = 2. ✗
  - ABE: Appears in $t_3, t_5 \longrightarrow$ Support = 2. ✗
- **Step 4: Frequency of Itemsets (Size 4)** Finally, calculate the frequency of itemsets of size 4:
  - $ABCE$: Appears in $t_3, t_5 \longrightarrow$ Support = 2. ✗
- **Step 5: Identify Frequent Itemsets** Using the minimum support threshold $\alpha = 3$, only itemsets with support ≥ 3 are considered frequent:
  - **Single items:** $\{A, B, C, E\}$ (supports 3, 5, 5, 5).
  - **Itemsets of size 2:** $\{AC, BC, BE, CE\}$ (supports 3, 4, 5, 4).
  - **Itemsets of size 3:** $\{BCE\}$ (support 4).
- **Step 6: Maximal Frequent Itemsets** A maximal frequent itemset is a frequent itemset that has no frequent supersets. Remember that in order to find closed or maximal itemsets we just need to study frequent itemsets considering:

$$\text{Frequent Itemsets} \supseteq \text{Closed Itemsets} \supseteq \text{Maximal Itemsets}$$

  - **Maximal Itemsets:** Clearly we can't consider maximal itemsets the original single itemsets found ($\{A, B, C, E\}$) as when adding another itemsets they still are frequent as shown in the frequent itemsets of size 2.

    Now, in terms of itemsets of size 2, we can add to the itemsets BC, BE and CE the corresponding itemsets E, C and B composing in the 3 cases the itemset BCE which is frequent, therefore none of them are maximal itemsets, except AC, which if you add B (S=2), D (S=1) and E (S=2). The only maximal itemset of size 2 is then AC.

    Finally, the only left frequent itemset of size 3 is BCE which is maximal because there are no frequent itemsets of size 4. Then, the maximal itemsets are:

$$\{AC, BCE\}$$

  - **Closed Itemsets:** A closed itemset is a frequent itemset that no superset of itself has the same support.

    Let's start with the single frequent itemsets. The itemset A can't be a closed itemset as adding C maintains the same support (S=3), the same happens with B (respectively with E) as adding E

(respectively B) shares the same support (S=5). However, with the itemset C, no added itemset creates a superset sharing the same support than C, then C is a closed itemset.

In terms of itemsets of size 2, we know that maximal itemsets are closed itemsets, therefore we know for sure that AC is a closed itemset. In regards to BC, by adding E we obtain the itemset BCE with the same support (S=4), then it can't be a closed itemset. For BE, adding the itemset C creates an itemset with support 5, different from the original, therefore BE is a closed itemset. Finally, CE isn't a closed itemset as adding B, gives the itemset BCE with the same support 4.

As previously mentioned, BCE is a maximal itemset of size 3 and therefore a closed itemset. The closed itemsets are:

$$\{C, AC, BE, BCE\}$$

5 Exercise 5

- **Question 3:** Run the LCM algorithm on D_1 with a minimum support threshold $\alpha = 3$.

Let's proceed with the steps of the algorithm. To apply the LCM (Linear time Closed itemset Miner) algorithm on the given dataset D_1 with min support = 3, we will mine the closed frequent itemsets step by step:

1. **Frequent Items (Single Items):** We calculate the support of each single itemset:
   - A: Appears in $t_2, t_3, t_5, t_6, t_7, t_{10} \longrightarrow$ Support = 6. ✓
   - B: Appears in $t_1, t_2, t_3, t_5, t_7, t_{10} \longrightarrow$ Support = 6. ✓
   - C: Appears in $t_1, t_2, t_3, t_6, t_9 \longrightarrow$ Support = 5. ✓
   - D: Appears in $t_1, t_3, t_4, t_8 \longrightarrow$ Support = 4. ✓
   - E: Appears in $t_2, t_4, t_6, t_7, t_9 \longrightarrow$ Support = 5. ✓
   - F: Appears in $t_3, t_6, t_7, t_8, t_{10} \longrightarrow$ Support = 5. ✓

   Then, the single frequent itemsets are:

   $$\{A, B, C, D, E, F\}.$$

   Prune those which are not closed itemsets, in this case all are closed itemsets.

2. For the frequent itemsets of size 2:
   - AB: Appears in $t_2, t_3, t_5, t_7, t_{10} \longrightarrow$ Support = 5. ✓
   - AC: Appears in $t_2, t_3, t_6 \longrightarrow$ Support = 3. ✓
   - AD: Appears in $t_3 \longrightarrow$ Support = 1. ✗
   - AE: Appears in $t_2, t_6, t_7 \longrightarrow$ Support = 3. ✓
   - AF: Appears in $t_3, t_6, t_7, t_{10} \longrightarrow$ Support = 4. ✓
   - BC: Appears in $t_1, t_2, t_3 \longrightarrow$ Support = 3. ✓
   - BD: Appears in $t_1, t_3 \longrightarrow$ Support = 2. ✗
   - BE: Appears in $t_2, t_7 \longrightarrow$ Support = 2. ✗
   - BF: Appears in $t_3, t_7, t_{10} \longrightarrow$ Support = 3. ✓
   - CD: Appears in $t_1, t_3 \longrightarrow$ Support = 2. ✗
   - CE: Appears in $t_2, t_6, t_9 \longrightarrow$ Support = 3. ✓
   - CF: Appears in $t_3, t_6 \longrightarrow$ Support = 2. ✗
   - DE: Appears in $t_4 \longrightarrow$ Support = 1. ✗
   - DF: Appears in $t_3, t_8 \longrightarrow$ Support = 2. ✗
   - EF: Appears in $t_6, t_7 \longrightarrow$ Support = 2. ✗

Then, the frequent itemsets of size 2 are: $\{AB, AC, AE, AF, BC, BF, CE\}$.

The closed itemsets of size 2 are: $\{AB, AC, AE, AF, BC, CE\}$.

3. We now calculate the frequent itemsets of size 3, considering only combinations of frequent size-2 itemsets:
   - ABC: Appears in $t_2, t_3 \longrightarrow$ Support = 2. ✗
   - ABF: Appears in $t_3, t_7, t_{10} \longrightarrow$ Support = 3. ✓
   - ACF: Appears in $t_3, t_6 \longrightarrow$ Support = 2. ✗
   - ACE: Appears in $t_2, t_6 \longrightarrow$ Support = 2. ✗
   - AEF: Appears in $t_6, t_7 \longrightarrow$ Support = 2. ✗
   - BCF: Appears in $t_3 \longrightarrow$ Support = 1. ✗
   - BCE: Appears in $t_2 \longrightarrow$ Support = 1. ✗
   - ABE: Appears in $t_2, t_7 \longrightarrow$ Support = 2. ✗
   - BEF: Appears in $t_7 \longrightarrow$ Support = 1. ✗
   - FEC: Appears in $t_6 \longrightarrow$ Support = 1. ✗

   Then, the frequent itemsets of size 3 is $\{ABF\}$, and is the only closed itemset of size 3. No more frequent itemsets with size greater than 3 can be found.
4. **Results:** The frequent closed itemsets are:

$$\{A, B, C, D, E, F, AB, AC, AE, AF, BC, CE, ABF\}.$$

- **Question 4:** Implement the LCM algorithm on the datasets provided in **.\DataSets\\.**

  Both answers are provided in the following Github Link.

6 Exercise 6

Consider the following query and its two interpretations:

$$Q : \text{frequent}(P) \ \wedge \ \text{closed}(P) \ \wedge \ \text{maxSize}_{ub}(P)$$

Interpretations:

1. Mine all frequent closed itemsets that additionally have a size less than or equal to **ub**.
2. Mine all frequent itemsets of size less than or equal to **ub** that additionally have the property of being closed.

- **Question 1:** Provide the set of solutions for **Q** under both interpretations on the dataset D_1 with a minimum support threshold $\theta = 3$.

  The first thing to determine is the set of frequent itemsets of different sizes and the closed itemsets.
  - **Frequent Itemsets:**

    We calculate the support of each single itemset:
    - A: Appears in $t_2, t_3, t_5, t_6, t_7, t_{10} \longrightarrow$ Support = 6. ✓
    - B: Appears in $t_1, t_2, t_3, t_5, t_7, t_{10} \longrightarrow$ Support = 6. ✓
    - C: Appears in $t_1, t_2, t_3, t_6, t_9 \longrightarrow$ Support = 5. ✓
    - D: Appears in $t_1, t_3, t_4, t_8 \longrightarrow$ Support = 4. ✓
    - E: Appears in $t_2, t_4, t_6, t_7, t_9 \longrightarrow$ Support = 5. ✓
    - F: Appears in $t_3, t_6, t_7, t_8, t_{10} \longrightarrow$ Support = 5. ✓

Trans.	Items					
t_1		B	C	D		
t_2	A	B	C		E	
t_3	A	B	C	D		F
t_4				D	E	
t_5	A	B				
t_6	A		C		E	F
t_7	A	B			E	F
t_8				D		F
t_9			C		E	
t_{10}	A	B				F

Figure 4: Transactional Database D_1

Then, the single frequent itemsets are:

$$\{A, B, C, D, E, F\}.$$

For the frequent itemsets of size 2:

- AB: Appears in $t_2, t_3, t_5, t_7, t_{10} \longrightarrow$ Support $= 5$. ✓
- AC: Appears in $t_2, t_3, t_6 \longrightarrow$ Support $= 3$. ✓
- AD: Appears in $t_3 \longrightarrow$ Support $= 1$. ✗
- AE: Appears in $t_2, t_6, t_7 \longrightarrow$ Support $= 3$. ✓
- AF: Appears in $t_3, t_6, t_7, t_{10} \longrightarrow$ Support $= 4$. ✓
- BC: Appears in $t_1, t_2, t_3 \longrightarrow$ Support $= 3$. ✓
- BD: Appears in $t_1, t_3 \longrightarrow$ Support $= 2$. ✗
- BE: Appears in $t_2, t_7 \longrightarrow$ Support $= 2$. ✗
- BF: Appears in $t_3, t_7, t_{10} \longrightarrow$ Support $= 3$. ✓
- CD: Appears in $t_1, t_3 \longrightarrow$ Support $= 2$. ✗
- CE: Appears in $t_2, t_6, t_9 \longrightarrow$ Support $= 3$. ✓
- CF: Appears in $t_3, t_6 \longrightarrow$ Support $= 2$. ✗
- DE: Appears in $t_4 \longrightarrow$ Support $= 1$. ✗
- DF: Appears in $t_3, t_8 \longrightarrow$ Support $= 2$. ✗
- EF: Appears in $t_6, t_7 \longrightarrow$ Support $= 2$. ✗

Then, the frequent itemsets of size 2 are:

$$\{AB, AC, AE, AF, BC, BF, CE\}.$$

We now calculate the frequent itemsets of size 3, considering only combinations of frequent size-2 itemsets:

- ABC: Appears in $t_2, t_3 \longrightarrow$ Support $= 2$. ✗
- ABF: Appears in $t_3, t_7, t_{10} \longrightarrow$ Support $= 3$. ✓
- ACF: Appears in $t_3, t_6 \longrightarrow$ Support $= 2$. ✗
- ACE: Appears in $t_2, t_6 \longrightarrow$ Support $= 2$. ✗
- AEF: Appears in $t_6, t_7 \longrightarrow$ Support $= 2$. ✗

    - ∗ BCF: Appears in $t_3 \longrightarrow$ Support = 1. ✗
    - ∗ BCE: Appears in $t_2 \longrightarrow$ Support = 1. ✗
    - ∗ ABE: Appears in $t_2, t_7 \longrightarrow$ Support = 2. ✗
    - ∗ BEF: Appears in $t_7 \longrightarrow$ Support = 1. ✗
    - ∗ FEC: Appears in $t_6 \longrightarrow$ Support = 1. ✗

  Then, the frequent itemsets of size 3 is: $\{ABF\}$. No more frequent itemsets with size greater than 3 can be found. The frequent itemsets found are:

    - ∗ **Single items:** $\{A, B, C, D, E, F\}$ (supports 6, 6, 5, 4, 5, 5).
    - ∗ **Itemsets of size 2:** $\{AB, AC, AE, AF, BC, BF, CE\}$ (supports 5, 3, 3, 4, 3, 3, 3).
    - ∗ **Itemsets of size 3:** $\{ABF\}$ (support 3).

- – **Closed and Maximal Itemsets:**
    - ∗ Closed Itemsets: A closed itemset has no superset with the same support. Based on the frequent itemsets, we deduce that:
        - · **Single Closed-Itemsets:** $\{A, B, C, D, E, F\}$
        - · **Closed-Itemsets of size 2:** $\{AB, AC, AE, AF, BC, CE\}$
        - · **Closed-Itemsets of size 3:** $\{ABF\}$
    - ∗ Maximal Itemsets: A maximal itemset has no frequent supersets. Based on the frequent itemsets:
        - · **Single Maximal Itemsets:** $\{D\}$
        - · **Maximal Itemsets of size 2:** $\{AC, AE, BC, CE\}$
        - · **Maximal Itemsets of size 3:** $\{ABF\}$

  Now, we can interpret the query with its interpretations to extract the set of solutions **Q**:

    - ∗ **Interpretation 1:** Mine frequent closed itemsets of size ≤ **ub**.

      If **ub** = 1: $\{A, B, C, D, E, F\}$

      If **ub** = 2: $\{AB, AC, AE, AF, BC, CE, A, B, C, D, E, F\}$

      If **ub** = 3: $\{ABF, AB, AC, AE, AF, BC, CE, A, B, C, D, E, F\}$

    - ∗ **Interpretation 2:** Mine frequent itemsets of size ≤ **ub** that are closed. The frequent itemsets found are:
        - · **Single items:** $\{A, B, C, D, E, F\}$.
        - · **Itemsets of size 2:** $\{AB, AC, AE, AF, BC, BF, CE\}$.
        - · **Itemsets of size 3:** $\{ABF\}$.

      Therefore, the frequent itemsets of size ≤ **ub** are:

      If **ub** = 1:

$$\underbrace{\{A, B, C, D, E, F\}}_{\text{Frequent Itemsets of size } \leq 1} \underbrace{\Longrightarrow}_{\text{Closed}} \underbrace{\{A, B, C, D, E, F\}}_{\text{Frequent Itemsets of size } \leq 1 \text{ that are closed}}$$

      If **ub** = 2:

$$\underbrace{\{AB, AC, AE, AF, BC, BF, CE, A, B, C, D, E, F\}}_{\text{Frequent Itemsets of size } \leq 2}$$

$$\underbrace{\Longrightarrow}_{\text{Closed}} \underbrace{\{AB, AC, AE, AF, BC, CE, A, B, C, D, E, F\}}_{\text{Frequent Itemsets of size } \leq 2 \text{ that are closed}}$$

      If **ub** = 3:

$$\underbrace{\{ABF, AB, AC, AE, AF, BC, BF, CE, A, B, C, D, E, F\}}_{\text{Frequent Itemsets of size } \leq 3}$$

$$\underbrace{\Longrightarrow}_{\text{Closed}} \underbrace{\{ABF, AB, AC, AE, AF, BC, CE, A, B, C, D, E, F\}}_{\text{Frequent Itemsets of size } \leq 2 \text{ that are closed}}$$

- **Question 2:** What is the correct semantic of this query? Explain your reasoning.

  The correct semantic is **Interpretation 1**:

  > "Mine all frequent closed itemsets that additionally have a size less than or equal to **ub**."

  This is because the query prioritizes the closed property, meaning itemsets must first be frequent and closed before applying the size constraint. The size limit acts as a final filter, ensuring only valid closed itemsets of size ≤ **ub** are included. This aligns with how frequent closed itemsets are typically mined for efficiency and correctness.