Quantitative Association Rules

Pietro Sala

Data Mining 24/25 - Exercise 1

QAR: definitions

We consider a quantitative dataset D on the schema $A_1, \dots, A_n, C$, where each A_i is a natural number and C is a positive natural number. In this context, we define two key concepts: itemsets and the satisfaction relation.

We define the set of all intervals over natural numbers as follows:

$$\mathbb{IN} = \{[b, e] \mid b, e \in \mathbb{N}, b \leq e\}$$

For two intervals $[b_1, e_1], [b_2, e_2] \in \mathbb{IN}$, we say that $[b_1, e_1]$ is contained in $[b_2, e_2]$, denoted as $[b_1, e_1] \subseteq [b_2, e_2]$, if and only if:

$$[b_1, e_1] \subseteq [b_2, e_2] \iff b_2 \leq b_1 \leq e_1 \leq e_2$$

An itemset I over the dataset D is a function that maps each attribute to an interval over the natural numbers. Formally, we express this as:

$$I : \{A_1, \dots, A_n\} \to \mathbb{IN}$$

where $\{A_1, \dots, A_n\}$ is the set of attributes in D but C.

Over the itemsets, we define a containment relation $\sqsubseteq$ as follows. For two itemsets I and I', we say that I is contained in I' (denoted as $I \sqsubseteq I'$) if and only if for each attribute A_i, the interval $I(A_i)$ is contained into interval in $I'(A_i)$. Formally:

$$\begin{gathered} I \sqsubseteq I' \\ \Updownarrow \\ \forall A_i \in \{A_1, \dots, A_n\} : I(A_i) \subseteq I'(A_i) \end{gathered}$$

For a point $p \in \mathbb{N}$ and an interval $[b, e] \in \mathbb{IN}$, we say that p belongs to $[b, e]$, denoted as $p \in [b, e]$, if and only if:

$$p \in [b, e] \iff b \leq p \leq e$$

Given an itemset I and a tuple t in D, we define a satisfaction relation to determine whether t satisfies I. We say that t satisfies I (denoted as $t \models I$) if and only if, for each attribute A_i, the value of $t[A_i]$ falls within the interval $I(A_i)$. Formally:

$$\begin{gathered} t \models I \\ \Updownarrow \\ \forall A_i \in \{A_1, \dots, A_n\} : t[A_i] \in I[A_i] \end{gathered}$$

where $t[A_i]$ denotes the value of attribute A_i in tuple t. This definition effectively checks each attribute A_i in the tuple, looking if $I(A_i)$ contains the value $t[A_i]$. If this holds for all attributes, then t is considered to satisfy I. It's important to note that while $t[A_i]$ is a single natural number, $I(A_i)$ is an interval.

Now, let us consider a specific instance $\boldsymbol{d}$ of the dataset D. For each attribute A_i, we denote by max_i the maximum value of that attribute in $\boldsymbol{d}$. Formally, we can express this as:

$$max_i = \max\{t[A_i] \mid t \in \boldsymbol{d}\}$$

where $t[A_i]$ represents the value of attribute A_i in tuple t. This definition of max_i provides us with the upper bound of values for each attribute within the given instance $\boldsymbol{d}$ of our dataset.

Given an itemset I, a dataset instance $\boldsymbol{d}$, and a real number $\varepsilon \in [0, 1]$, we say that I is ε-supported in $\boldsymbol{d}$ if and only if the sum of C values for tuples in $\boldsymbol{d}$ that satisfy I, divided by the total sum of C values in $\boldsymbol{d}$, is greater than ε. Formally:

$$\begin{gathered} I \text{ is } \varepsilon\text{-supported in } \boldsymbol{d} \\ \Updownarrow \\ \frac{\sum_{t \in \boldsymbol{d} : t \models I} t[C]}{\sum_{t \in \boldsymbol{d}} t[C]} \geq \varepsilon \end{gathered}$$

This concept of ε-support provides a threshold for considering an itemset as sufficiently represented in a dataset instance. It allows us to focus on itemsets that occur frequently enough to be of interest,

Algorithm 1 Apriori Algorithm for Quantitative Itemsets

```
Require: Dataset $\boldsymbol{d}$, support threshold $\varepsilon$
Ensure: Relation $\mathcal{R}(itemset, support)$ of all $\varepsilon$-supported itemsets and their support values
Initialize empty relation $\mathcal{R}(I, support)$ with key $I$
$SW_0 \leftarrow \{I_0\}$                    ▷ Set of supported witnesses of level 0
$k \leftarrow 1$
while $SW_{k-1} \neq \emptyset$ do
    $W_k \leftarrow \{I : \forall I'(I \sqsubseteq I' \wedge \Delta(I') = \Delta(I) - 1 \implies I' \in SW_{k-1})\}$
    $SW_k \leftarrow \{I : I \in W_k \wedge I$ is $\varepsilon$-supported in $\boldsymbol{d}\}$
    for all $I \in SW_k$ do
        Insert $(I, support(I))$ into $\mathcal{R}$
    end for
    $k \leftarrow k + 1$
end while
return $\mathcal{R}$
```

with the threshold ε determining the minimum required level of support.

Given two intervals $[b, e]$ and $[b', e']$ such that $[b, e] \sqsubseteq [b', e']$, we define their shrink difference, denoted by $\delta([b, e], [b', e'])$, as:

$$\delta([b, e], [b', e']) = (b - b') + (e' - e)$$

This shrink difference quantifies the total amount by which the larger interval $[b', e']$ needs to be "shrunk" from both ends to obtain the smaller interval $[b, e]$. It provides a measure of how much the intervals differ in size and position.

We define the bottom itemset, denoted by I_0, as the itemset that maps each attribute A_i to the interval $[0, max_i]$, formally, $I_0(A_i) = [0, max_i]$, for each $1 \leq i \leq n$.

Clearly, for any itemset I defined over the same dataset instance, we have $I \sqsubseteq I_0$. This property allows us to define the shrinking of an itemset I, denoted by $\Delta(I)$, as the sum of the shrink differences between the intervals in I and the corresponding intervals in I_0 for all attributes. Formally:

$$\Delta(I) = \sum_{i=1}^{n} \delta(I(A_i), [0, max_i])$$

where $\delta([b, e], [0, max_i])$ is the shrink difference as defined earlier. This shrinking $\Delta(I)$ quantifies how much more specific the itemset I is compared to the bottom itemset I_0, considering all attributes and all intervals in I.

Lemma 1. *Let I be an itemset that is ε-supported in a dataset instance $\boldsymbol{d}$. Then, all itemsets I' such that $I \sqsubseteq I'$ and $\Delta(I') = \Delta(I) - 1$ are also ε-supported in $\boldsymbol{d}$.*

We now present the Apriori (Algorithm 1) algorithm adapted for quantitative itemsets, which efficiently finds all ε-supported itemsets in a given dataset.

Where:

- $\boldsymbol{d}$ is the input dataset
- ε is the support threshold
- I_0 is the bottom itemset as defined earlier
- SW_k is the set of ε-supported witnesses at level k
- W_k is the set of candidate itemsets at level k
- $\Delta(I)$ is the shrinking of itemset I as defined earlier
- $support(I)$ is calculated as $\frac{\sum_{t \in \boldsymbol{d}: t \models I} t[C]}{\sum_{t \in \boldsymbol{d}} t[C]}$

This algorithm generates and tests itemsets level by level, utilizing the property established in our previous lemma. It starts with the bottom itemset I_0 and progressively generates more specific itemsets, pruning those that cannot be ε-supported based on the support of their generalizations. The algorithm terminates when no new ε-supported itemsets are found at a given level.

Key features of this algorithm:

1. It leverages the monotonicity of support with respect to itemset containment.
2. It uses the shrinking measure Δ to systematically explore the space of itemsets.

3. It efficiently prunes the search space by only considering itemsets whose all immediate generalizations are supported.

4. It returns a complete set of all ε-supported itemsets along with their support values.

Exercise 1

Implementation and Analysis of Apriori Algorithm for Quantitative Itemsets. This exercise will guide you through the process of implementing, testing, and applying the Apriori algorithm for quantitative itemsets. You will also perform post-processing on the results to extract and rank association rules. Assignment:

1.a) **Algorithm Implementation and Testing**

- Implement the Apriori algorithm for quantitative itemsets in your programming language of choice.
- Create a set of test cases to verify the correctness of your implementation.
- Ensure your implementation can handle various input sizes and support thresholds.

1.b) **Application to Air Quality Dataset**

- Obtain the air quality dataset.
- Implement an encoder/decoder for the dataset:
  - For each attribute, map its values to integers between 0 and n, where $v_0 < \ldots < v_n$ are the distinct values for the attribute.
  - Apply this encoding to the original table to obtain a new table that respects the schema $A_1, \ldots, A_m, C$.
- Apply your Apriori algorithm implementation to this encoded dataset.
- Choose an appropriate support threshold ε based on the characteristics of your dataset.

1.c) **Extracting Association Rules**

- From the output of the Apriori algorithm, extract all association rules with a given confidence threshold γ, where $0 \leq \gamma \leq 1$.
- Implement a function to calculate the confidence of a rule given the supports of the itemsets involved.
- **Important:** Ensure that the final set of rules is presented in the decoded version, using the original attribute values from the air quality dataset.

1.d) **Ranking the Rules**

- Rank the obtained association rules (in their decoded form) using the following criteria separatedly:
  (a) rank by P-values in ascending order (lower P-values indicate higher statistical significance).
  (b) rank by Lift in descending order.

CPO for QAR

Let us introduce the concept of single tests over attributes. Given an attribute A_i, we define:

- A lower test is a pair $[\![A_i \geq n]\!]$ where $k \in \mathbb{N}$
- An upper test is a pair $[\![A_i \leq n]\!]$ where $k \in \mathbb{N}$

Given an instance of a quantitative dataset $\boldsymbol{d}$ on the schema $A_1, \ldots, A_n, C$, we denote by $\mathbb{T}$ any set of tests over $\boldsymbol{d}$. We define $\mathbb{T}_{\not{i}}$ as the set of all possible tests on $\boldsymbol{d}$ excluding those on attribute A_i. Formally:

$$\mathbb{T}_{\not{i}} = \{[\![A_j \bowtie n]\!] \mid j \neq i, \bowtie \in \{\leq, \geq\}, k \in \mathbb{N}\}$$

where $\bowtie$ represents either $\leq$ or $\geq$.

Given a set of tests $T \subseteq \mathbb{T}_{\not{i}}$, we say that it entails an interval $[b, e]$ on the attribute A_j, written as $T \rightarrow A_j[b, e]$, if and only if one of the following conditions holds:

1. $b = 0$ there exists $[\![A_j \leq e]\!] \in T$ and for all lower tests $[\![A_j \geq k]\!] \in T$, we have $k > e$

2. $b \neq 0$ and there exists a lower test $[\![A_j \geq b]\!] \in T$, and e is the minimum value $e > b$ such that $[\![A_j \leq e]\!] \in T$

3. $b \neq 0$ and there exists a lower test $[\![A_j \geq b]\!] \in T$, and for all upper tests $[\![A_j \leq k]\!] \in T$, we have $b > k$, and $e = max_j$

$$J(X \to Y) = p(X) \cdot \left(p(Y|X) \log_2 \left(\frac{p(Y|X)}{p(Y)} \right) + (1 - p(Y|X)) \log_2 \left(\frac{1 - p(Y|X)}{1 - p(Y)} \right) \right)$$

$$J(AI \to A_i[b,e]) = \mathit{support}(AI, \boldsymbol{d}) \cdot \left(\begin{array}{c} \frac{\mathit{support}(AI \cup \{A_i[b,e]\}, \boldsymbol{d})}{\mathit{support}(AI, \boldsymbol{d})} \log_2 \left(\frac{\mathit{support}(AI \cup \{A_i[b,e]\}, \boldsymbol{d})}{\mathit{support}(AI, \boldsymbol{d}) \cdot \mathit{support}(\{A_i[b,e]\}, \boldsymbol{d})} \right) \\ + \\ \left(1 - \frac{\mathit{support}(AI \cup \{A_i[b,e]\}, \boldsymbol{d})}{\mathit{support}(AI, \boldsymbol{d})}\right) \log_2 \left(\frac{1 - \frac{\mathit{support}(AI \cup \{A_i[b,e]\}, \boldsymbol{d})}{\mathit{support}(AI, \boldsymbol{d})}}{1 - \mathit{support}(\{A_i[b,e]\}, \boldsymbol{d})} \right) \end{array} \right)$$

Figure 1: Top: the J-measure (Smyth and Goodman) combines coverage and accuracy in a single metric. Here, $p(X)$ represents the probability of the antecedent, $p(Y)$ the probability of the consequent, and $p(Y|X)$ the conditional probability of Y given X. Bottom: specialized form for labelled interval rules.

From now on, we refer to the notation $A_j[b, e]$ as a labelled interval.
We denote by $\mathbb{AI}$ the set of all labelled intervals on dataset $\boldsymbol{d}$. Formally:

$$\mathbb{AI} = \{A_j[b, e] \mid 1 \leq j \leq n, [b, e] \in \mathbb{IN}\}$$

Given a set of labelled intervals $AI \subseteq \mathbb{AI}$, we say that a tuple t satisfies AI (written as $t \models AI$) if and only if:

$$\begin{array}{rl} & t \models AI \\ & \Updownarrow \\ \forall i \in \{1, \ldots, n\} : & (\nexists A_i[b, e] \in AI) \vee \\ & (\exists A_i[b, e] \in AI : b \leq t[A_i] \leq e) \end{array}$$

Given a set of labelled intervals $AI \subseteq \mathbb{AI}$, its support in dataset instance $\boldsymbol{d}$ is defined as:

$$\mathit{support}(AI, \boldsymbol{d}) = \frac{\sum\limits_{t \in \boldsymbol{d} : t \models AI} t[C]}{\sum\limits_{t \in \boldsymbol{d}} t[C]}$$

The J-measure (J) is an information theoretic measure that combines precision and recall for rules.
As shown in Figure 1, the J-measure provides a way to evaluate the interestingness of rules by combining both their coverage and accuracy in a single metric. In particular, in Figure 1, the top equation shows the standard form for general rules, while the bottom equation shows its adaptation to a labelled interval rule $AI \to A_i[b, e]$ where probabilities are expressed in terms of support in the dataset $\boldsymbol{d}$.
Given a labelled interval $A_i[b, e]$, we associate to it a coalition payoff function:

$$\mathit{cpo}_{A_i[b,e]} : 2^{\mathbb{T}_{\not{i}}} \to \mathbb{R}^+$$

defined as:

$$\mathit{cpo}_{A_i[b,e]}(T) = J(AI \to A_i[b, e])$$

where AI is the set of labelled intervals entailed by T.

Would you like me to add more details about the relationship between the test set T and the entailed intervals AI, or continue with additional properties?

Exercise 2

2.a) **Algorithm Implementation**

- Implement a function to compute the approximate Shapley value for each test in $\mathbb{T}_{\not{i}}$ using $\mathit{cpo}_{A_i[b,e]}$ as the coalition payoff function
- The implementation should:
  - Take as input: a quantitative dataset, a labelled interval $A_i[b, e]$, and the number of samples
  - Use sampling to approximate the Shapley values (Algorithm 2)
  - Return the Shapley value for each test in $\mathbb{T}_{\not{i}}$

2.b) **Application to Air Quality Dataset**

- Choose a meaningful labelled interval $A_i[b, e]$ from the air quality dataset
- Run your implementation with different sample sizes to evaluate convergence

$$\phi_{test} = \sum_{S \subseteq \mathbb{T}_i \setminus \{test\}} \frac{|S|!(|\mathbb{T}_i| - |S| - 1)!}{|\mathbb{T}_i|!} (cpo_{A_i[b,e]}(S \cup \{test\}) - cpo_{A_i[b,e]}(S)) \tag{1}$$

Figure 2: The Shapley value (non -approximated) computation for a test *test*, where $cpo_{A_i[b,e]}$ is the coalition payoff function based on the J-measure of the entailed labelled interval rule.

Algorithm 2 Estimate Shapley (CPO, X, Y, n_sampler)

```
Pick randomly $X_1, \ldots, X_{n\_samples}$ distinct subsets of $X$-$\{j\}$
Initialize $N, D \leftarrow 0, 0$
for $i = 1, \ldots, n\_samples$ do
    $D_i \leftarrow |X_i|!(|X| - |X_i| - 1)!$
    $N \leftarrow N + D_i \cdot (cpo(X_i \cup \{Y\}) - cpo(X_i))$
    $N \leftarrow N + D_i \cdot (cpo(X/X_i) - cpo((X/X_i) \cup \{Y\}))$
    $D \leftarrow D + 2 \cdot D_i$
end for
return $\frac{N}{D}$
```

- Analyze the distribution of Shapley values across the tests

Note: Recall that the (non-approximated) Shapley value for a test is shown in Figure 2.