Declarative Itemset Mining

Course Notes 4

Nadjib Lazaar
Paris-Saclay University
lazaar@lisn.fr

Contents

1 Introduction

Data mining is the process of extracting meaningful and relevant information from large datasets and transforming it into useful knowledge. Among its many subfields, Frequent Pattern Mining is one of the most studied and widely applied. Originally introduced by Agrawal et al. [Agrawal et al., 1993], this field focuses on discovering recurring patterns in data, such as frequent itemsets [Agrawal et al., 1994, Han et al., 2000], closed itemsets [Pasquier et al., 1999, Zaki and Hsiao, 2002], association rules [Agrawal et al., 1994], and sequential patterns [Agrawal and Srikant, 1995]. These patterns play a crucial role in various applications, including market basket analysis, bioinformatics, and software testing.

Despite its success, traditional pattern mining faces a major challenge: scalability and relevance. The sheer number of extracted patterns often surpasses the size of the dataset itself, making it difficult for users to identify the most meaningful ones [Bringmann and Zimmermann, 2007]. To address this, researchers have shifted from purely efficiency-driven methods to more user-centric approaches that allow incorporating domain-specific constraints. However, integrating user-defined constraints into specialized mining algorithms remains a bottleneck. Existing techniques include preprocessing the dataset, filtering results after mining, or modifying mining algorithms to embed constraints directly [Wojciechowski and Zakrzewicz, 2002]. Each of these approaches has limitations, especially when user constraints evolve dynamically.

A promising alternative is Declarative Data Mining (DIM), which leverages constraint programming (CP) and propositional satisfiability (SAT) to model and solve data mining tasks in a more flexible way [De Raedt et al., 2008, Kemmar et al., 2015, Schaus et al., 2017, Boudane et al., 2017]. Unlike traditional approaches, CP-based methods allow users to define mining tasks using constraints without altering the underlying algorithm. While CP-based techniques may not yet outperform specialized algorithms in terms of runtime for standard queries, they excel in scenarios where users seek a small, relevant set of patterns rather than exhaustive enumeration. Moreover, CP offers compact and expressive formulations, making it easier to adapt mining tasks to new constraints without re-engineering the entire process.

In this lecture, we will explore how declarative approaches can revolutionize pattern mining by offering flexible, constraint-driven solutions that enhance user control over the mining process. We will discuss fundamental concepts, key contributions, and future directions for DIM, highlighting its potential in bridging the gap between data mining and constraint programming.

2 Preliminaries

In addition to the standard notations introduced in previous notes on Constraint Programming (CP), we introduce additional ones specific to Declarative Itemset Mining (DIM).

A data mining task is defined under a set of parameters that characterize a specific configuration Ω. For example, when mining association rules in a dataset $\mathcal{D}$, a user may specify a minimum frequency α, a minimum confidence β, a minimum size lb for the rule's head, and a maximum size ub for its body. The configuration space in this case is:

$$\Omega = \langle \mathcal{D}, \alpha, \beta, lb, ub \rangle$$

In this context, we define a *size-independent constraint network* as a *CP model*, which will be the central concept discussed in this section.

CP solvers typically use backtracking search to explore the space of partial assignments. At each step, *constraint propagation* algorithms (propagators) enforce local consistency to prune infeasible solutions. A constraint c_S is said to be *domain consistent (DC)* if, for every variable $X_i \in S$ and every value $v \in D(X_i)$, there exists a valid assignment satisfying c_S such that $X_i = v$.

One of the key strengths of CP lies in *global constraints*, which capture relationships involving multiple variables. These constraints allow the solver to leverage structural properties of the problem, improving efficiency. Examples include:

- AllDifferent: Ensures that all variables in a set take distinct values.
- Regular: Enforces that a sequence of variables conforms to a given finite-state automaton.
- Among: Counts how many variables in a set take values from a specified subset (see [Rossi et al., 2006]).

Global constraints generally require specialized propagators to ensure efficient pruning, as generic propagation methods often become intractable due to exponential complexity.

Example 1 *Consider a constraint network:*

- *Variables:* $X = \{x_1, x_2, x_3\}$
- *Domains:* $D(x_1) = \{0, 2\}$*,* $D(x_2) = \{0, 2, 4\}$*,* $D(x_3) = \{1, 2, 3, 4\}$
- *Constraints:*
  - $c_1 : x_1 \geq x_2$
  - $c_2 : x_1 + x_2 = x_3$

Enforcing domain consistency (DC) on c_1 *removes the value* 4 *from* $D(x_2)$*, while enforcing DC on* c_2 *removes* 1 *and* 3 *from* $D(x_3)$*. The constraint network thus admits two solutions:*

$$(x_1 = 2, x_2 = 0, x_3 = 2) \quad \textit{and} \quad (x_1 = 2, x_2 = 2, x_3 = 4)$$

$\diamondsuit$

3 Constraint Programming for Itemset Mining

The first CP model for itemset mining was introduced by *Tias Guns* during his PhD thesis, in collaboration with *Luc De Raedt* and *Siegfried Nijssen* [De Raedt et al., 2008, Guns et al., 2011]. Their model, called **CP4IM**, provides a declarative encoding of fundamental constraints such as frequency, maximality, and closedness.

The CP4IM model relies on two vectors of Boolean variables:

- **Decision variables:** $X_1, X_2, \ldots, X_n$, where $X_i = 1$ if and only if item i is included in the extracted itemset.

- **Auxiliary variables:** $T_1, T_2, \ldots, T_m$, where $T_t = 1$ if and only if the extracted itemset appears in transaction t of the dataset $\mathcal{D}$.

Encoding Constraints in CP4IM The *cover* of an itemset is enforced using constraints of arity $(n + 1)$:

$$\forall t \in \mathcal{D} : (T_t = 1) \Leftrightarrow \sum_{a_i \in \mathcal{I}} X_i(1 - cover(t, a_i)) = 0 \tag{1}$$

The *minimum frequency* constraint, given a threshold α, is encoded as:

$$\sum_{t \in \mathcal{D}} T_t \geq \alpha \tag{2}$$

The *closedness* constraint is expressed as:

$$\forall a_i \in \mathcal{I} : (X_i = 1) \Leftrightarrow \sum_{t \in \mathcal{D}} T_t(1 - cover(t, a_i)) = 0 \tag{3}$$

Similarly, the *maximality* constraint (ensuring no superset of the extracted itemset is frequent) is encoded as:

$$\forall a_i \in \mathcal{I} : (X_i = 1) \Leftrightarrow \sum_{t \in \mathcal{D}} T_t \ cover(t, a_i) \geq \alpha \tag{4}$$

3.1 SAT-Based Approaches to Itemset Mining

Given that CP4IM primarily relies on Boolean variables, using SAT (Boolean Satisfiability) for itemset mining appears natural. However, encoding integer arithmetic constraints in SAT can be costly, making direct translation nontrivial.

SAT-based methods have been successfully applied to data mining tasks, particularly through the work of *Lakhdar Sais* and *Said Jabbour* at CRIL, Lens. Their research introduced efficient SAT-based models for:

- **Closed frequent itemset mining** [Boudane et al., 2018b]

- **Maximal frequent itemset mining** [Jabbour et al., 2018]
- **Top-k frequent itemsets** [Jabbour et al., 2013]
- **Association rule mining** [Boudane et al., 2016, Boudane et al., 2017]

SAT-based approaches significantly outperform CP4IM and narrow the gap with specialized algorithms by employing the following strategies:

- *On-the-fly constraint learning:* Blocking clauses dynamically enforce maximality constraints, reducing model size and enhancing scalability [Jabbour et al., 2018].
- *Parallel SAT-based itemset mining:* Incremental decomposition [Jabbour et al., 2015] and parallel SAT solvers [Dlala et al., 2018] improve efficiency.
- *Efficient encoding of pseudo-Boolean constraints:* Conditional cardinality encodings allow handling integer arithmetic efficiently [Izza et al., 2020, Boudane et al., 2018a].

4 Global Constraints in Itemset Mining

The rationale behind proposing global constraints for data mining is twofold. First, it provides a high level of expression by compactly formulating mining tasks. Second, it provides CP solvers with a high level of propagation. Data Mining is a broad field with many ideas, original algorithmic concepts, and dedicated data structures that can be drawn on to design effective propagators.

With the ambition to show the power of the declarative model using CP in Itemset Mining, the first objective was to equip CP solvers with a DM toolbox composed of global constraints capturing basic Itemset mining tasks like closedness.

4.1 closedPattern

The closedPattern global constraint was introduced for mining closed frequent itemsets [Lazaar et al., 2016]. This constraint combines the coverage, frequency, and closedness conditions, as presented in equations (1), (2), and (3). closedPattern operates directly on the n decision variables encoding the extracted itemsets, eliminating the need for reified constraints or auxiliary variables.

For the rest of the section, the following notations are used:

- $\mathcal{P}_X = \{a_i \in \mathcal{I} \mid D(X_i) = \{1\}\}$.
- $\mathcal{N}_X = \{a_i \in \mathcal{I} \mid D(X_i) = \{0\}\}$.
- $\mathcal{U}_X = \{a_i \in \mathcal{I} \mid D(X_i) = \{0, 1\}\}$.

Note that for a complete assignment on X, we have $\mathcal{P}_X \cup \mathcal{N}_X = \mathcal{I}$ and $\mathcal{U}_X = \emptyset$.

Definition 1 (CLOSEDPATTERN global constraint) *Let $X_1, \ldots, X_n$ be binary item variables, and let $\mathcal{D}$ be a dataset with minimum frequency α. Given a complete assignment σ on $X_1, \ldots, X_n$, CLOSEDPATTERN $_{\mathcal{D},\alpha}(\sigma)$ holds if and only if $\mathtt{freq}(\mathcal{P}_X) \geq \alpha$ and $\mathcal{P}_X$ is closed.*

In addition to being expressive and compact, CLOSEDPATTERN provides CP solvers with a clearer view of the problem's structure, thanks to its optimal propagator. An algorithm that implements three propagation rules has been proposed:

- **Rule-1:** This rule prunes the 0 value from item variables X_i that are closure extensions of the partial instantiation.
- **Rule-2:** This rule prunes the value 1 from X_i if its inclusion would decrease the frequency of the itemset below the threshold α.
- **Rule-3:** This rule prunes the value 1 from items that are always present in the dataset when items absent in the partial assignment are also excluded.

It has been proven in **Theorem 1** [Lazaar et al., 2016] that applying the three propagation rules is sufficient to enforce domain consistency on the CLOSEDPATTERN constraint in time complexity $O(n^2 \times m)$, with space complexity $O(n \times m)$. This means that enumerating the total number of closed frequent itemsets is backtrack-free with the implementation of these rules (**Proposition 3** [Lazaar et al., 2016]).

When compared with the CP4IM model using reified constraints, a speed-up factor ranging between 10 and 200 is observed in terms of CPU time when using CLOSEDPATTERN, assuming no out-of-memory errors with CP4IM. Uno et al. [Uno et al., 2004] introduced LCM, the fastest closed frequent itemset mining algorithm, which is 15 times faster than a declarative model using CLOSEDPATTERN. This remains true when comparing specialized methods to declarative models for simple DM tasks such as enumerating closed frequent itemsets. However, the advantage of declarative models becomes clear when user-specific constraints are involved. For instance, one may request k distinct closed frequent itemsets of a given size. In this case, a declarative CP model consists of a CLOSEDPATTERN and two additional user-specific constraints. On the other hand, using a specialized method like LCM requires expensive post-processing to handle these user-specific constraints, or would necessitate the development of a new algorithm to incorporate them.

The introduction of CLOSEDPATTERN in 2016 raised awareness in the AI community about the need for more global constraints, helping to equip CP technology with a toolbox to simplify the expression of mining tasks and make CP solvers more effective in Data Mining (DM). A year later, an improved version of CLOSEDPATTERN was proposed, focusing on better data structures and the computation of the cover of an itemset, presented in the CoverSize global constraint [Schaus et al., 2017]. Recent research also suggests revising closedPattern to account for diversity distances between extracted itemsets [Hien et al., 2020].

4.2 Generator

In a similar context, the Generator global constraint was introduced in [Belaid et al., 2019a], ensuring that the extracted itemsets have no proper subsets with the same cover.

Definition 2 (Generator global constraint) *Let $X_1, \ldots, X_n$ be binary item variables, and let $\mathcal{D}$ be a dataset. Given a complete assignment σ on $X_1, \ldots, X_n$, Generator $_{\mathcal{D}}(\sigma)$ holds if and only if $\mathcal{P}_X$ is a generator.*

A complete polynomial algorithm was proposed to achieve domain consistency on the Generator constraint, with time complexity $O(n^2 \times m)$ (**Theorems 2 and 3** [Belaid et al., 2019a]). The propagator leverages a monotonic property stating that all supersets of a non-generator are also non-generators [Szathmary et al., 2009].

4.3 FreqSub and InfreqSup

In note 4, it was shown that a maximal itemset is (a) a *frequent itemset* and (b) *all its proper supersets are infrequent*. It was also observed that a minimal itemset is (c) *an infrequent itemset* and (d) *all its proper subsets are frequent*. The frequent/infrequent constraints ((a) and (c)) have been addressed in CP using the CP4IM model [Guns et al., 2011], the CoverSize constraint [Schaus et al., 2017], and the Frequent constraint presented in [Belaid et al., 2019b], which includes a propagator implementing **Rule-2** of closedPattern. For constraints (b) and (d), two new global constraints were proposed to further enhance the frequent/infrequent constraints, enabling the mining of maximal and minimal itemsets [Belaid et al., 2019b]:

Definition 3 (FreqSub global constraint) *Let $X_1, \ldots, X_n$ be binary item variables, and let $\mathcal{D}$ be a dataset with a minimum frequency α. Given a complete assignment σ on $X_1, \ldots, X_n$, FreqSub $_{\mathcal{D},\alpha}(\sigma)$ holds if and only if $\forall Q \subset \mathcal{P}_X$, $\texttt{freq}(Q) \geq \alpha$.*

Definition 4 (InfreqSup global constraint) *Let $X_1, \ldots, X_n$ be binary item variables, and let $\mathcal{D}$ be a dataset with a minimum frequency α. Given a complete assignment σ on $X_1, \ldots, X_n$, InfreqSup $_{\mathcal{D},\alpha}(\sigma)$ holds if and only if $\forall Q \supset \mathcal{P}_X$, $\texttt{freq}(Q) < \alpha$.*

The decomposition of mining maximal/minimal itemsets into the constraints $(a)+(b)$ and $(c)+(d)$ allows for duality between the propagators of $((a),(c))$ and $((b),(d))$. Polynomial algorithms have been developed to enforce domain consistency on both the FreqSub and InfreqSup constraints (**Theorem 2 and 3** [Belaid et al., 2019b] for FreqSub and **Theorem 4 and 5** [Belaid et al., 2019b] for InfreqSup). These results further demonstrate the power of CP to add expressive new constraints while maintaining efficiency.

4.4 FreqRare

In itemset mining, setting the minimum frequency threshold α presents a significant challenge. A high value for α may result in missing important correlations between frequent and rare items. Conversely, a low value for α can lead to the generation of a large number of meaningless itemsets. For example, in a sales transactions dataset, the item $\{necklace\}$ occurs rarely, while the item $\{flowers\}$ is frequent. A high threshold for α would fail to identify the interesting itemset $\{necklace, flowers\}$, while a low threshold would result in the itemset being lost among many irrelevant ones.

To address the *rare item problem* [Liu et al., 1999], several approaches have been proposed for mining frequent itemsets with multiple minimum frequency thresholds [Liu et al., 1999, Tseng and Lin, 2001, Kiran and Re, 2009, Kiran and Reddy, 2011, Gan et al., 2017]. The problem of mining frequent itemsets with a set $\mathcal{S}$ of multiple Minimum Item Support (MIS) values was introduced in [Liu et al., 1999]. In this scenario, each item $i \in \mathcal{I}$ has its own minimum frequency threshold $\alpha_i \in \mathcal{S}$, and an itemset P is considered frequent if:

$$freq(P) \geq \min_{i \in P, \alpha_i \in \mathcal{S}} \alpha_i \tag{5}$$

For the example of the itemset $\{necklace, flowers\}$, we might set $\alpha_{necklace} = 10$ and $\alpha_{flowers} = 10,000$.

Example 2 *The dataset D shown in Figure 1 contains 4 items and 5 transactions. Based on its multiple MIS values, the following conclusions hold:*

- *AB is frequent since $freq(AB) = 3$ and $\alpha_B = 3$;*
- *ABC is infrequent since $freq(ABC) = 2$ and $\alpha_B = \alpha_C = 3$;*
- *$ABCD$ is frequent since $freq(ABCD) = 1$ and $\alpha_D = 1$.*

Figure 1: Transaction dataset example D with its corresponding MIS set $\mathcal{S}$.

D:

trans.	Items			
t_1	A	B		D
t_2	A		C	D
t_3	A	B	C	D
t_4		B	C	
t_5	A	B	C	

$\mathcal{S}$:

A	B	C	D
4	3	3	1

♢

It is important to note that the frequency with MIS in equation 5 is not a monotonic function, as demonstrated in Example 2, where AB (frequent) $\subset ABC$ (infrequent) $\subset ABCD$ (frequent).

Recently, the FreqRare global constraint was introduced to mine frequent itemsets with multiple MIS [Belaid and Lazaar, 2021].

Definition 5 (FreqRare global constraint) *Let $X_1, \dots, X_n$ be binary item variables, and let $\mathcal{D}$ be a dataset with the corresponding set of minimum item supports (MIS) $\mathcal{S}$. Given a complete assignment σ on $X_1, \dots, X_n$, FreqRare $_{\mathcal{D},\mathcal{S}}(\sigma)$ holds if and only if:*

$$freq(\mathcal{P}_X) \geq \min_{i \in \mathcal{P}_X, s_i \in \mathcal{S}} s_i.$$

This constraint is presented with a polynomial propagator (**Proposition 2** [Belaid and Lazaar, 2021]). Moreover, **Proposition 1** [Belaid and Lazaar, 2021] demonstrates that enumerating frequent itemsets with multiple MIS using the proposed propagator, while employing increasing minimum item support as a variable ordering heuristic, is *backtrack-free.*

The `CFPGrowth`++ algorithm, proposed in [Kiran and Reddy, 2011], is highly efficient in enumerating all frequent itemsets with multiple MIS. It performs between 2 and 18 times faster than a CP model with the FreqRare constraint. For example, `CFPGrowth`++ can enumerate over 100 million itemsets in under a minute, while the optimized CP implementation using FreqRare requires 2 minutes. In such cases, the CP solver becomes primarily focused on enumerating itemsets.

For user-specific constraints, such as limiting the distance between MIS pairs (which can easily be expressed as constraints within a CP model), the FreqRare constraint can be particularly useful. Additional constraints on forbidden or mandatory items, or on the cardinality of itemsets, can also be incorporated. In this context, the use of FreqRare in a CP model is from 4 to 43 times faster than `CFPGrowth`++ with postprocessing. For instance, using FreqRare with extra constraints on a given dataset can yield the 14 desired solutions in less than a minute, whereas `CFPGrowth`++ requires over 6 minutes to enumerate more than 15 million itemsets and filter out those violating user-specific constraints. This efficiency arises from the enhanced propagation process that helps to enumerate the few valid solutions in the large search space.

4.5 Confident

The *confidence* measure is widely recognized as a fundamental metric in association rule mining, as it quantifies how often a rule holds true in the dataset. In [Belaid et al., 2019a], a global constraint, denoted Confident, was introduced to ensure the minimum confidence of extracted rules.

Definition 6 (Confident global constraint) *Let $X_1, \dots, X_n$ and $Y_1, \dots, Y_n$ be binary item variables, and let x and y represent two vectors of Boolean variables. Let $\mathcal{D}$ denote a dataset and β be a minimum confidence threshold. Given a complete assignment σ_X on $X_1, \dots, X_n$ and σ_Y on $Y_1, \dots, Y_n$, the constraint* Confident $_{\mathcal{D},\beta}(\sigma_X, \sigma_Y)$ *holds if and only if:*

$$conf(\mathcal{P}_X \rightarrow \mathcal{P}_Y) \geq \beta.$$

It has been shown that enforcing domain consistency on the Confident constraint is NP-hard (**Theorem 1** in [Belaid et al., 2019a]). Therefore, a non-complete propagator was proposed for Confident. The algorithm is based on a propagation rule, which states that, for a rule $r : X \rightarrow Y$, increasing the number of conclusions in the head Y of the rule can only reduce the confidence of the rule. This propagation can be triggered only if the body X of the rule is fully instantiated. Ensuring this property can be achieved in polynomial time.

5 CP Models in Itemset Mining

With the extensive set of constraints available, we can now demonstrate the declarative power of constraint programming (CP) through models that address various itemset mining tasks.

5.1 A Generic CP Model for Itemset Mining

One of the key advantages of using constraint programming (CP) for itemset mining is the ability to incorporate user-specific constraints directly into the model. This allows for the generation of only *interesting* itemsets without incurring additional implementation costs. While it is straightforward to impose constraints on the types of itemsets to extract, users may also be interested in mining specific parts of the dataset (such as certain items or transactions).

For example, a user may want to extract itemsets that involve appliances, but only in transactions that exceed 100 EUR in total. In traditional approaches, the dataset would be preprocessed with a specialized algorithm to create a sub-dataset containing only appliances (as items) and transactions exceeding 100 EUR. This approach becomes more complex if the user is interested in itemsets that are frequent in specific time windows (e.g., one hour) or based on price differences (e.g., 10 EUR between items). Preprocessing can generate a large number of sub-datasets, leading to inefficiencies.

This section introduces a generic CP model, ItemSet, which is expressive enough to handle both user-specific constraints on the itemsets to mine (*what to mine*) and constraints on the items and transactions in the dataset (*where to mine*) [Bessiere et al., 2018].

5.1.1 Vocabulary

To mine itemsets under various constraints, we need at least four Boolean vectors of variables: X, T, H, and V. These variables encode the following:

- $X = \langle X_1, \ldots, X_n \rangle$: This vector represents the itemset we are looking for. Each Boolean variable X_i indicates whether item i is part of the extracted itemset.
- $T = \langle T_1, \ldots, T_m \rangle$: This vector represents the transactions that are covered by the extracted itemset.

- $H = \langle H_1, \dots, H_n \rangle$: This vector represents the items in the sub-dataset where the mining will occur. If $H_i = 0$, item i is ignored.
- $V = \langle V_1, \dots, V_m \rangle$: This vector represents the transactions in the sub-dataset where the mining will occur. If $V_j = 0$, transaction j is ignored.

The pair $\langle H, V \rangle$ defines the sub-dataset used to extract the itemset. The CP solver explores different sub-datasets, backtracking from one sub-dataset and branching onto another. The pair $\langle X, T \rangle$ represents the itemset we are mining and its corresponding cover set in terms of transactions.

5.1.2 Constraints

The generic CP model consists of three sets of constraints:

$$\textsc{ItemSet}(\langle X, T \rangle, \langle H, V \rangle) = \begin{cases} \textsc{DataSet}(\langle H, V \rangle) \\ \textsc{Channeling}(\langle X, T \rangle, \langle H, V \rangle) \\ \textsc{Mining}(\langle X, T \rangle, \langle H, V \rangle) \end{cases}$$

1. $\textsc{DataSet}(\langle H, V \rangle)$: This set of constraints expresses user-specific constraints on items (i.e., H) and/or transactions (i.e., V). It defines the sub-datasets that will be used for mining.

2. $\textsc{Channeling}(\langle X, T \rangle, \langle H, V \rangle)$: This set of channeling constraints defines the relationship between the two sets of variables, $\langle X, T \rangle$ and $\langle H, V \rangle$:

   $$H_i = 0 \Rightarrow X_i = 0$$
   $$V_j = 0 \Rightarrow T_j = 0$$

   These constraints ensure that if an item (or transaction) is not part of the mining process, it will not be part of the extracted itemset (or the cover set).

3. $\textsc{Mining}(\langle X, T \rangle, \langle H, V \rangle)$: This set of constraints expresses itemset constraints such as frequency, closedness, size, and more sophisticated user-specific constraints.

Suppose items are categorized into k categories (e.g., *food, electronics, cleaning*, etc.), denoted as $\mathcal{I} = \mathcal{I}_1 \cup \ldots \cup \mathcal{I}_k$. Similarly, transactions are categorized into v categories based on customer attributes (e.g., age or gender), denoted as $\mathcal{T} = \mathcal{T}_1 \cup \ldots \cup \mathcal{T}_v$. The user can request closed frequent itemsets involving items from at least lb_I categories and at most ub_I categories, as well as itemsets present in at least lb_T and at most ub_T categories of transactions.

This query can be easily expressed in the ItemSet model, where the Mining part is reduced to a closedPattern constraint. The Channeling part remains unchanged, and the DataSet part is expressed as follows:

$$\textsc{DataSet}(H,V) = \begin{cases} lb_I \leq \sum_{j=1}^{k} \min_{a_i \in \mathcal{I}_j} H_i = \sum_{j=1}^{k} \max_{a_i \in \mathcal{I}_j} H_i \leq ub_I \\ lb_T \leq \sum_{j=1}^{v} \min_{i \in \mathcal{T}_j} V_i = \sum_{j=1}^{v} \max_{i \in \mathcal{T}_j} V_i \leq ub_T \end{cases}$$

For each category, the first constraint ensures that either all items in that category are activated, or none are. The number of categories with activated items is between lb_I and ub_I, and the same applies to transactions in the second constraint.

This example illustrates how constraints on both the itemsets to mine and the dataset to mine from can be expressed. While methods like LCM can be used to enumerate closed frequent itemsets, satisfying user-specific constraints on items and transactions would typically require preprocessing (e.g., PP-LCM). The ItemSet model, however, can efficiently handle these constraints and significantly outperforms PP-LCM, especially as the number of sub-datasets grows exponentially. Preprocessing becomes infeasible in such cases.

5.2 CP model for Mining Borders

Given a minimum frequency threshold, a borderline is drawn between frequent and infrequent itemsets. Extracting the itemsets on the borders (the maximals on the positive border and the minimals on the negative border) is sufficient to decide whether a given itemset is a frequent or infrequent one. A duality exists between maximal and minimal itemsets. The problem of inferring minimals from maximals and vice-versa is known as the dualization problem [Mannila and Toivonen, 1997, Boros et al., 2003, Nourine and Petit, 2012].

An interesting property in mining borders is that:

"An itemset, that has only frequent subsets and only infrequent supersets, is either maximal or minimal."

The *"only frequent subsets"* is captured by FreqSub constraint and the *"only infrequent supersets"* by InfreqSup. Posting the two global constraints will have the two borders (the minimals + the maximals) as solutions. Thus, any subset (resp. superset) of a solution of (FreqSub +InfreqSup) is a frequent itemset (resp. infrequent itemset). If we want to have only the positive/negative border and ensure that a maximal is a frequent itemset and a minimal is an infrequent one, we need just to ensure that the solutions are frequent itemsets for the positive border, or that solutions are infrequent itemsets for the negative border.

In [Belaid et al., 2019b], a CP model Borders$_\alpha$ was presented, able to switch from mining minimals to maximals according to a direction α. Borders$_\alpha$ acts on a vector X of n Boolean variables, where X_i represents the presence of item i in the extracted itemset.

$$\textsc{Borders}_\alpha(X) = \begin{cases} \textsc{FrequentSubs}(X) \\ \textsc{InfrequentSupers}(X) \\ \alpha \iff \textsc{Frequent}(X) \end{cases}$$

The FreqSub holds if the extracted itemset has only frequent subsets, and InfreqSup holds if the extracted itemset has only infrequent supersets. The third constraint in the $\textsc{Borders}_\alpha$ model is reified. The Boolean parameter α reifies the constraint Frequent. Frequent holds if and only if the extracted itemset is frequent. Thus, the solutions of $\textsc{Borders}_\alpha$ correspond to the set of maximals if $\alpha = true$, the set of minimals otherwise (**Theorem 1** – [Belaid et al., 2019b]).

The faster-specialized algorithms to enumerate the border or the other are `FPGrowth` (for maximals) [Grahne and Zhu, 2003] and $\texttt{Walky} - \texttt{G}$ (for minimals) [Szathmary et al., 2012]. $\textsc{Borders}_\alpha$ is very competitive, and even faster in some instances, in addition, it can mine the positive border as well as the negative one just by flipping a parameter.

5.3 CP Model for Mining Association Rules

For mining association rules, two vectors X, Y of n Boolean variables are needed. One for the body of the rule and the second one for the head. A third vector of auxiliary variables Z can be introduced to ease the expression of some constraints. Here, X_i, Y_i, and Z_i represent the presence of item i in the body, in the head, and the rule as a whole (body or head).

In [Belaid et al., 2019a], the Rule model was presented, which involves five types of constraints:

$$\textsc{Rule}(X, Y, Z) = \begin{cases} \forall i \in \mathcal{I} : \ \neg X_i \vee \neg Y_i & (1) \\ \bigvee_{i \in \mathcal{I}} Y_i & (2) \\ \forall i \in \mathcal{I} : \ Z_i \iff X_i \vee Y_i & (3) \\ \textsc{Confident}(X, Y) & (4) \\ \textsc{Frequent}(Z) & (5) \end{cases}$$

The role of each type of constraint is the following:

(1) ensures that a given item cannot be both in the body and in the head of a rule;

(2) ensures that the head of a rule is not empty;

(3) is a channelling constraint ensuring that $Z = X \cup Y$;

(4) is a global constraint that ensures that the extracted rule is confident;

(5) is a global constraint that ensures that the extracted rule is frequent.

The Rule model can be extended to only return some representative rules with MNRs (definition **??**). An operational characterization of MNRs can be exploited in the modelling task [Bastide et al., 2000]:

> *"An association rule $r : X \to Y$ is an MNR if and only if (i) r is a valid rule, (ii) the body X is a generator and (iii) the rule as a whole $X \cup Y$ is closed."*

The Rule model can be extended to a model able to extract MNRs by adding two constraints:

$$\text{MNR}\textsc{ule}(X, Y, Z) = \begin{cases} \textsc{Rule}(X, Y, Z) & (1) \\ \textsc{Generator}(X) & (2) \\ \textsc{ClosedPattern}(Z) & (3) \end{cases}$$

In the model MNRule,

(1) ensures that the extracted rule is valid;

(2) ensures that the body is a generator;

(3) ensures that the body + the head form a closed itemset.

Eclat-Z is one of the fastest association rules mining algorithms [Szathmary et al., 2007]. Eclat-Z acts in three steps. First, it uses the vertical algorithm Eclat for extracting all frequent itemsets. Second, it filters out the set of frequent itemsets to get the closed and generator itemsets. Then, it associates generator itemsets with their closed itemsets. In experimental evaluations, Eclat-Z performs very well compared to Rule and MNRule, which are also competitive compared to Eclat-Z and sometimes faster. However, in most cases where millions of rules are extracted, Eclat-Z performs very well. It is important to note that CP returns solutions on the fly, while Eclat-Z needs to build a complex data structure before enumerating the rules. For instance, with a timeout of one hour on a large dataset, the CP model can enumerate more than 7 billion rules, while Eclat-Z reaches the timeout without any extracted rule. On constrained rules (mandatory items, forbidden items, cardinality constraints on the body/head, etc.), the gap becomes large between a declarative model and a specialized algorithm with a post-processing step. For instance, the CP model can extract the 73 rules of the problem in less than a second, whereas with Eclat-Z, it takes more than 6 minutes, starting by extracting more than 10 million rules and then spending most of the time filtering out the rules violating user-specific constraints.

6 Constrained Itemset Mining

Bonchi and Lucchese [Bonchi and Lucchese, 2004] shed light on the ambiguous semantics around mining itemsets by joining closedness and user-specific constraints. The main idea is that constraints can interfere with closedness (or maximality) when they are not monotone. Likewise, constraints can interfere with generator/minimality when they are not anti-monotone. Consider the example of a user query $\mathcal{Q}$ to show the confusion behind:

$\mathcal{Q}$: *"Mining closed itemsets with some forbidden items."*

Here, two operational interpretations of the query $\mathcal{Q}$ arise:

- Φ_1: Mine closed itemsets and then, keep only the ones not including the forbidden items.

- Φ_2: Mine all itemsets not including the forbidden items and then, return the closed ones.

The two interpretations will return different solutions for the same user query. This is explained by the fact that being maximal/closed or minimal/generator is a property true or false in the context of a set of itemsets, not a property inherent to the itemset itself.

References

[Agrawal et al., 1993] Agrawal, R., Imielinski, T., and Swami, A. N. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, May 26-28, 1993.*, pages 207–216.

[Agrawal and Srikant, 1995] Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of ICDE'95, Taipei, Taiwan*, pages 3–14. IEEE.

[Agrawal et al., 1994] Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. In *Proceedings of VLDB'94, Santigo de Chile, Chile*, volume 1215, pages 487–499.

[Bastide et al., 2000] Bastide, Y., Pasquier, N., Taouil, R., Stumme, G., and Lakhal, L. (2000). Mining minimal non-redundant association rules using frequent closed itemsets. In [Bastide et al., 2000], pages 972–986.

[Belaid et al., 2019a] Belaid, M., Bessiere, C., and Lazaar, N. (2019a). Constraint programming for association rules. In *Proceedings of SDM'19, Calgary, Alberta, Canada*. SIAM.

[Belaid et al., 2019b] Belaid, M., Bessiere, C., and Lazaar, N. (2019b). Constraint programming for mining borders of frequent itemsets. In Kraus, S., editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 1064–1070. ijcai.org.

[Belaid and Lazaar, 2021] Belaid, M. and Lazaar, N. (2021). Constraint programming for itemset mining with multiple minimum supports. In *33rd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2021, Washington, DC, USA, November 1-3, 2021*, pages 598–603. IEEE.

[Bessiere et al., 2018] Bessiere, C., Lazaar, N., and Maamar, M. (2018). User's constraints in itemset mining. In Hooker, J. N., editor, *Principles and Practice of Constraint Programming - 24th International Conference, CP 2018, Lille, France, August 27-31, 2018, Proceedings*, volume 11008 of *Lecture Notes in Computer Science*, pages 537–553. Springer.

[Bonchi and Lucchese, 2004] Bonchi, F. and Lucchese, C. (2004). On closed constrained frequent pattern mining. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK*, pages 35–42.

[Boros et al., 2003] Boros, E., Gurvich, V., Khachiyan, L., and Makino, K. (2003). On maximal frequent and minimal infrequent sets in binary matrices. *Annals of Mathematics and Artificial Intelligence*, 39(3):211–221.

[Boudane et al., 2018a] Boudane, A., Jabbour, S., Raddaoui, B., and Sais, L. (2018a). Efficient sat-based encodings of conditional cardinality constraints. In Barthe, G., Sutcliffe, G., and Veanes, M., editors, *LPAR-22. 22nd International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Awassa, Ethiopia, 16-21 November 2018*, volume 57 of *EPiC Series in Computing*, pages 181–195. EasyChair.

[Boudane et al., 2016] Boudane, A., Jabbour, S., Sais, L., and Salhi, Y. (2016). A sat-based approach for mining association rules. In Kambhampati, S., editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2472–2478. IJCAI/AAAI Press.

[Boudane et al., 2017] Boudane, A., Jabbour, S., Sais, L., and Salhi, Y. (2017). Enumerating non-redundant association rules using satisfiability. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 824–836. Springer.

[Boudane et al., 2018b] Boudane, A., Jabbour, S., Sais, L., and Salhi, Y. (2018b). Sat-based data mining. *Int. J. Artif. Intell. Tools*, 27(1):1840002:1–1840002:24.

[Bringmann and Zimmermann, 2007] Bringmann, B. and Zimmermann, A. (2007). The chosen few: On identifying valuable patterns. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*, pages 63–72. IEEE Computer Society.

[De Raedt et al., 2008] De Raedt, L., Guns, T., and Nijssen, S. (2008). Constraint programming for itemset mining. In *Proceedings of KDD'08, Las Vegas, Nevada, USA*, pages 204–212. ACM.

[Dlala et al., 2018] Dlala, I. O., Jabbour, S., Raddaoui, B., and Sais, L. (2018). A parallel sat-based framework for closed frequent itemsets mining. In *International Conference on Principles and Practice of Constraint Programming*, pages 570–587. Springer.

[Gan et al., 2017] Gan, W., Lin, J. C.-W., Fournier-Viger, P., Chao, H.-C., and Zhan, J. (2017). Mining of frequent patterns with multiple minimum supports. *Engineering Applications of Artificial Intelligence*, 60:83–96.

[Grahne and Zhu, 2003] Grahne, G. and Zhu, J. (2003). Efficiently using prefix-trees in mining frequent itemsets. In *Proceedings of FIMI'03, Melbourne, Florida, USA*, volume 90.

[Guns et al., 2011] Guns, T., Nijssen, S., and De Raedt, L. (2011). Itemset mining: A constraint programming perspective. *Artificial Intelligence*, 175(12):1951–1983.

[Han et al., 2000] Han, J., Pei, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.*, pages 1–12.

[Hien et al., 2020] Hien, A., Loudni, S., Aribi, N., Lebbah, Y., Laghzaoui, M. E. A., Ouali, A., and Zimmermann, A. (2020). A relaxation-based approach for mining diverse closed patterns. In Hutter, F., Kersting, K., Lijffijt, J., and Valera, I., editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part I*, volume 12457 of *Lecture Notes in Computer Science*, pages 36–54. Springer.

[Izza et al., 2020] Izza, Y., Jabbour, S., Raddaoui, B., and Boudane, A. (2020). On the enumeration of association rules: A decomposition-based approach. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1265–1271. ijcai.org.

[Jabbour et al., 2018] Jabbour, S., Mana, F. E., Dlala, I. O., Raddaoui, B., and Sais, L. (2018). On maximal frequent itemsets mining with constraints. In *International Conference on Principles and Practice of Constraint Programming*, pages 554–569. Springer.

[Jabbour et al., 2013] Jabbour, S., Sais, L., and Salhi, Y. (2013). The top-k frequent closed itemset mining using top-k sat problem. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 403–418. Springer.

[Jabbour et al., 2015] Jabbour, S., Sais, L., and Salhi, Y. (2015). Decomposition based SAT encodings for itemset mining problems. In Cao, T. H., Lim, E., Zhou, Z., Ho, T. B., Cheung, D. W., and Motoda, H., editors, *Advances in Knowledge Discovery and Data Mining - 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam, May 19-22, 2015, Proceedings, Part II*, volume 9078 of *Lecture Notes in Computer Science*, pages 662–674. Springer.

[Kemmar et al., 2015] Kemmar, A., Loudni, S., Lebbah, Y., Boizumault, P., and Charnois, T. (2015). PREFIX-PROJECTION global constraint for sequential pattern mining. In *CP 2015*, volume 9255 of *LNCS*, pages 226–243. Springer.

[Kiran and Re, 2009] Kiran, R. U. and Re, P. K. (2009). An improved multiple minimum support based approach to mine rare association rules. In *2009 IEEE Symposium on Computational Intelligence and Data Mining*, pages 340–347. IEEE.

[Kiran and Reddy, 2011] Kiran, R. U. and Reddy, P. K. (2011). Novel techniques to reduce search space in multiple minimum supports-based frequent pattern mining algorithms. In *Proceedings of the 14th international conference on extending database technology*, pages 11–20. ACM.

[Kryszkiewicz, 1998] Kryszkiewicz, M. (1998). Representative association rules. In [Kryszkiewicz, 1998], pages 198–209.

[Lazaar et al., 2016] Lazaar, N., Lebbah, Y., Loudni, S., Maamar, M., Lemière, V., Bessiere, C., and Boizumault, P. (2016). A global constraint for closed frequent pattern mining. In *Proceedings of CP'16, Toulouse, France*, pages 333–349. Springer.

[Liu et al., 1999] Liu, B., Hsu, W., and Ma, Y. (1999). Mining association rules with multiple minimum supports. In *Proceedings of KDD'99, San Diego, California, USA*, pages 337–341. ACM.

[Mannila and Toivonen, 1997] Mannila, H. and Toivonen, H. (1997). Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258.

[Nourine and Petit, 2012] Nourine, L. and Petit, J. (2012). Extending set-based dualization: Application to pattern mining. In *Proceedings of ECAI'12, Montpellier, France*, pages 630–635. IOS Press.

[Pasquier et al., 1999] Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999). Efficient mining of association rules using closed itemset lattices. *Inf. Syst.*, 24(1):25–46.

[Rossi et al., 2006] Rossi, F., van Beek, P., and Walsh, T. (2006). *Handbook of Constraint Programming.* Volume 2 of [Rossi et al., 2006].

[Schaus et al., 2017] Schaus, P., A., J., and Guns, T. (2017). Coversize: A global constraint for frequency-based itemset mining. In *Proceedings of CP'17, Melbourne, Australia*, pages 529–546. Springer.

[Szathmary et al., 2007] Szathmary, L., Valtchev, P., Napoli, A., and Godin, R. (2007). An efficient hybrid algorithm for mining frequent closures and generators. In [Szathmary et al., 2007], pages 47–58.

[Szathmary et al., 2009] Szathmary, L., Valtchev, P., Napoli, A., and Godin, R. (2009). Efficient vertical mining of frequent closures and generators. In *International Symposium on Intelligent Data Analysis*, pages 393–404. Springer.

[Szathmary et al., 2012] Szathmary, L., Valtchev, P., Napoli, A., and Godin, R. (2012). Efficient vertical mining of minimal rare itemsets. In *Proceedings of CLA'12, Malaga, Spain*, pages 269–280. Citeseer.

[Tseng and Lin, 2001] Tseng, M.-C. and Lin, W.-Y. (2001). Mining generalized association rules with multiple minimum supports. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 11–20. Springer.

[Uno et al., 2004] Uno, T., Asai, T., Uchida, Y., and Arimura, H. (2004). An efficient algorithm for enumerating closed patterns in transaction databases. In *DS 2004*, pages 16–31.

[Wojciechowski and Zakrzewicz, 2002] Wojciechowski, M. and Zakrzewicz, M. (2002). Dataset filtering techniques in constraint-based frequent pattern mining. In [Wojciechowski and Zakrzewicz, 2002], pages 77–91.

[Zaki and Hsiao, 2002] Zaki, M. J. and Hsiao, C. (2002). CHARM: an efficient algorithm for closed itemset mining. In *Proceedings of the Second SIAM International Conference on Data Mining, Arlington, VA, USA, April 11-13, 2002*, pages 457–473. SIAM.