Constraint Programming for Diversity Pattern Set Mining

Samir LOUDNI

Join work with A. Hien, M. Douad, A. Zimmermann, N. Aribi, A. Ouali, Y. Lebbah

GT CAVIAR

08/04/2025

Data mining

… is "the use of sophisticated data analysis tools to **discover** unknown, **valid patterns and relationships** in large datasets"

Data mining:

- Core of KDD
- Search for regularities from transaction databases
- Pattern domain : item-sets, sequences, graphs, etc.
- examples including pattern mining, clustering, association rules, etc.

	g_1	g_2	g_3	g_4
s_1	X			X
s_2	X	X	X	
s_3		X		X
s_4	X	X	X	
s_5	X	X		X

frequent pattern : $g_1 g_2$

association rule : $g_1 g_2 \rightarrow g_3$

Problem Definition

Given the objects in $\mathcal{O}$ described with the Boolean attributes in $\mathcal{I}$, listing every itemset having a frequency above a given threshold $\theta \in \mathbb{N}$.

Input:

$$\begin{array}{c|cccc} & a_1 & a_2 & \dots & a_n \\ \hline o_1 & d_{1,1} & d_{1,2} & \dots & d_{1,n} \\ o_2 & d_{2,1} & d_{2,2} & \dots & d_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ o_m & d_{m,1} & d_{m,2} & \dots & d_{m,n} \end{array}$$

and a minimal frequency $\theta \in \mathbb{N}$.

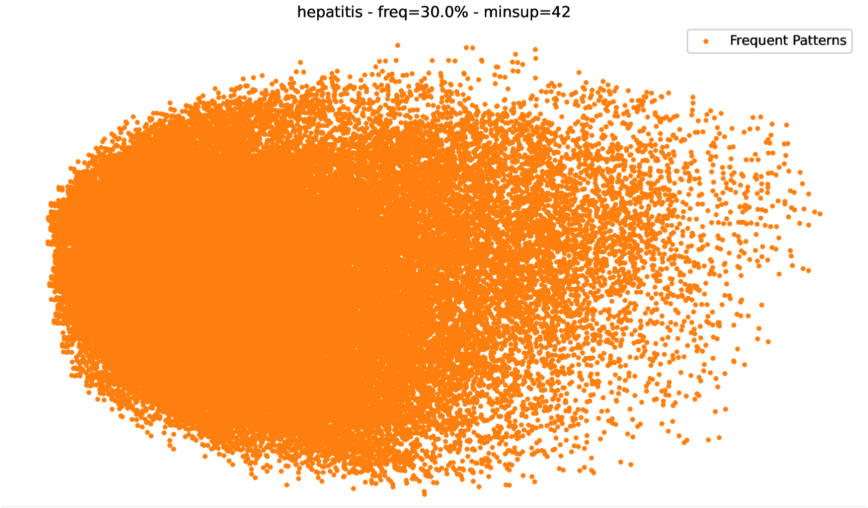
where $d_{i,j} \in \{\text{true,false}\}$

Problem Definition

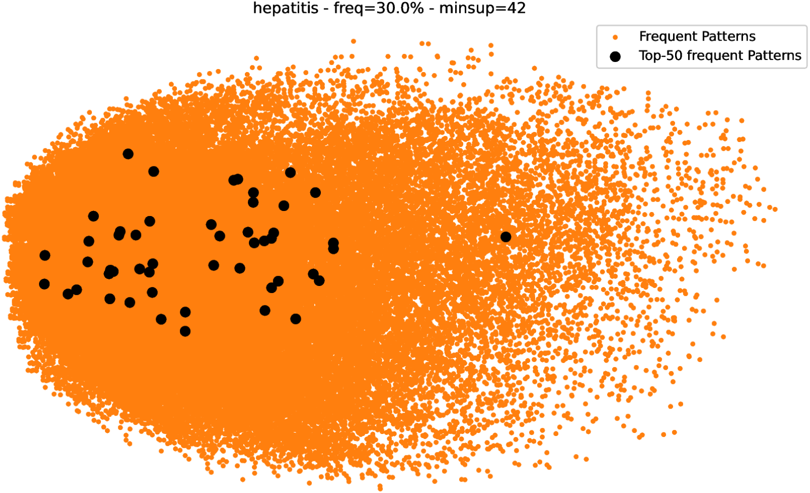
Given the objects in $\mathcal{O}$ described with the Boolean attributes in $\mathcal{I}$, listing every itemset having a frequency above a given threshold $\theta \in \mathbb{N}$.

Output: every $X \subseteq \mathcal{I}$ such that there are at least θ objects having all attributes in X.

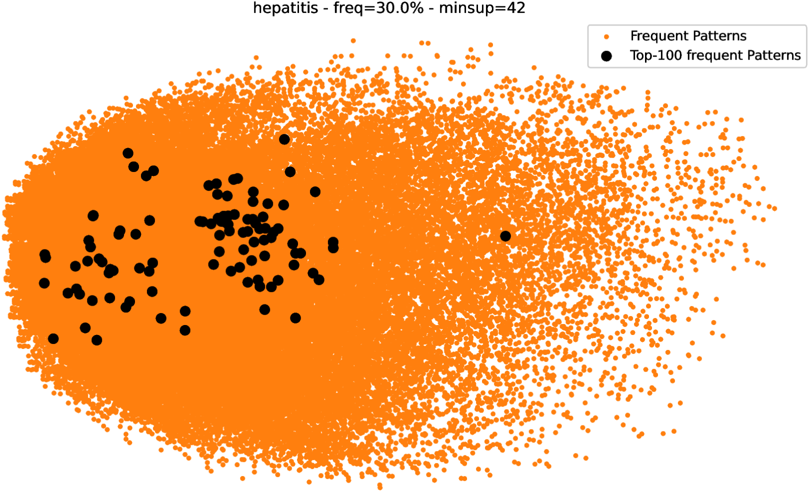
Pattern flooding: Mining frequent itemsets



Pattern flooding: Mining top-k frequent itemsets



Pattern flooding: Mining top-k frequent itemsets

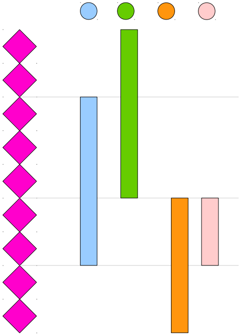


Redundancy

- Too many patterns, unmanageable and diversity not necessary assured

- Find a set of patterns that is:
  - small
  - non-redundant

- Several approaches for mining non-redundant patterns :
  - Mikis (Knobbe et al., ECML-PKDD'06)
  - Piker (Bringmann et al., KIS 2009)
  - Flexics (Dzyuba et al., DMKD 2017),
  - CFTP (Boley et al., KDD 2012)



➠ Exploiting CP for mining diverse set of patterns

Constraint programming

A generic framework for solving combinatorial problems

- A declarative description of the problem by a triplet (X,D,C) where
  - $X = \{x_1, \ldots, x_n\}$ is finite set of variables
  - $D = \{D_1, \ldots, D_n\}$ is finite set of domains (a.k.a possible values) of variables
  - $C = \{c_1, \ldots, c_e\}$ is a set of constraints restricting the values of variables x_i

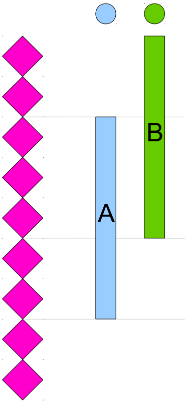
- **Resolution** = Enumeration + Filtering

solution $\equiv$ assignments on X satisfying all constraints of C

Measuring redundancy

Exploiting similarity measure to compare pairs of patterns :

- $|A \cap B| / \min(|A|, |B|)$ (Overlap)
- $|A \cup B| - |A \cap B|$ (Hamming distance)
- $|A \cap B| / |A|.|B|$ (Cosine similarity)
- $|A \cap B| / |A \cup B|$ (Jaccard similarity)



Definition (Hebrard et al., AAAI 2005)

Let $k \geq 2$ be an integer, and $\mathcal{S}$ be a set of solutions.

- The $\textsc{MaxDiverseKSet}(k)$ problem is the problem of finding a subset $\widetilde{S}$ of $\mathcal{S}$ of size k such that for all $S \subseteq \mathcal{S}$, $|S| = k$,

$$\min_{\substack{s_1, s_2 \in \widetilde{S} \\ s_1 \neq s_2}} J_{\mathcal{V}}(s_1, s_2) \leq \min_{\substack{s_1, s_2 \in S \\ s_1 \neq s_2}} J_{\mathcal{V}}(s_1, s_2).$$

- Given a set S of previously found solutions (a history), the $\textsc{MostDistant}(S)$ problem is the problem of finding a solution $\widetilde{s} \in \mathcal{S}$ such that for all $s \in \mathcal{S}$,

$$\min_{s' \in S} J_{\mathcal{V}}(\widetilde{s}, s') \leq \min_{s' \in S} J_{\mathcal{V}}(s, s').$$

- $\textsc{MaxDiverseKSet}(k)$ can be seen as finding a set of k solutions minimizing the pairwise Jaccard values.

- $\textsc{MostDistant}$ aims at finding the most distant solution from the history of solution ; can be seen as a greedy algorithm solving $\textsc{MaxDiverseKSet}$ problem.

First approximation of the MostDistant problem (Hien et al., ECML-PKDD 2020 & Constraints 2024)

Definition (Maximum Diversity Constraint)

Let $\mathcal{H}$ be a history of patterns, j_{max} a bound on the maximum allowed Jaccard, and P an itemset. The maximum diversity constraint ensures that P is diverse w.r.t. $\mathcal{H}$ and J_{max}.

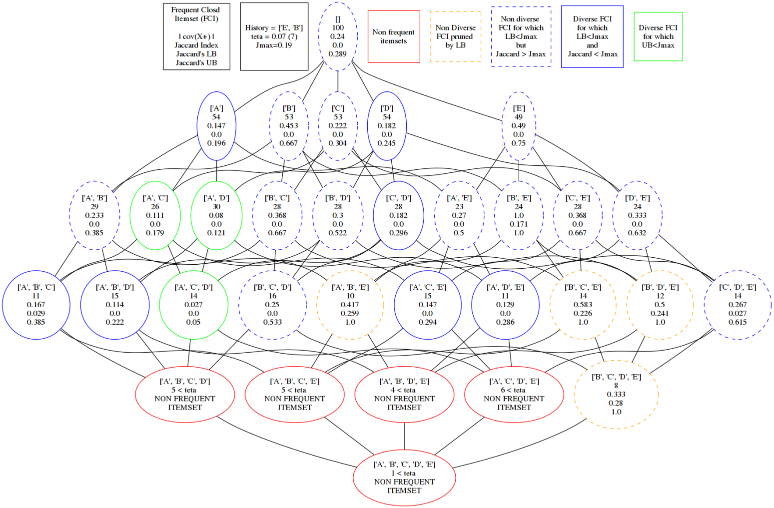
$$\texttt{div}(P, \mathcal{H}, j_{max}) \Leftrightarrow \forall H \in \mathcal{H}, J_{\mathcal{V}}(P, H) \leq J_{max}$$

Idea: Push the Jaccard constraint during pattern discovery to prune non-diverse patterns.

Task : Given a history $\mathcal{H}$ of k pairwise diverse patterns (initially empty), the task is to mine new patterns P such that $\texttt{div}(P, \mathcal{H}, j_{max})$ is satisfied.

Anti-monotonicity of the maximum Jaccard constraint

The anti-monotonicity does not hold for the Maximum Jaccard constraint



- For $\mathcal{H} = \{BE\}$ and $J_{max} = 0.19$, $Jac(AE, \mathcal{H}) = 0.27 \geq J_{max}$ whereas $Jac(ACE, \mathcal{H}) = 0.147 \leq J_{max}$.

Relaxations of the Jaccard constraint

(i) **A lower bound** LB_J, which allows to prune non-diverse patterns

$$LB_J(H,P) = \begin{cases} \frac{\theta - |\mathcal{V}_H^{pr}(P)|}{|\mathcal{V}_{\mathcal{D}}(H)| + |\mathcal{V}_H^{pr}(P)|} & \text{if } (\mathcal{V}_H^{pr}(P) < \theta) \\ 0 & \text{else} \end{cases}$$

(ii) **An upper bound** UB_J to find patterns ensuring diversity

$$UB_J(H,P) = \frac{|\mathcal{V}_{\mathcal{D}}(H) \cap \mathcal{V}_{\mathcal{D}}(P)|}{|\mathcal{V}_H^{pr}(P)| + \max\{\theta, |\mathcal{V}_{\mathcal{D}}(H)| \cap |\mathcal{V}_{\mathcal{D}}(P)|\}}$$

Jaccard lower and upper bounds

- **Monotonicity of LB_J:** Let $H \in \mathcal{H}$ be an itemset. For any two patterns $Q \subseteq P$, the relationship $LB_J(P, H) \geq LB_J(Q, H)$ holds.

  ➟ If $LB_J(P, H) > J_{max} \Rightarrow J_{\mathcal{V}}(P, H) > J_{max} \Rightarrow P$ **is not diverse**
  (any super-itemset $P' \supseteq P$ **will not be diverse**)

- **Anti-monotonicity of UB_J:** Let $H \in \mathcal{H}$ be an itemset. For any two patterns $P \subseteq Q$, the relationship $UB_J(P, H) \geq UB_J(Q, H)$ holds.

  ➟ If $UB_J(P, H) \leq J_{max} \Rightarrow J_{\mathcal{V}}(P, H) \leq J_{max} \Rightarrow P$ **is diverse**
  (any super-itemset $P' \supseteq P$ **will also be diverse**)

- A new global constraint $\textsc{ClosedDiversity}_{\mathcal{D},\theta}(X, \mathcal{H}, J_{max})$ for mining pairwise diverse patterns that exploits the two previous relaxations.

$\textsc{ClosedDiversity}_{\mathcal{D},\theta}(X, \mathcal{H}, J_{max})$ (A. Hien et al., Constraints 2024)

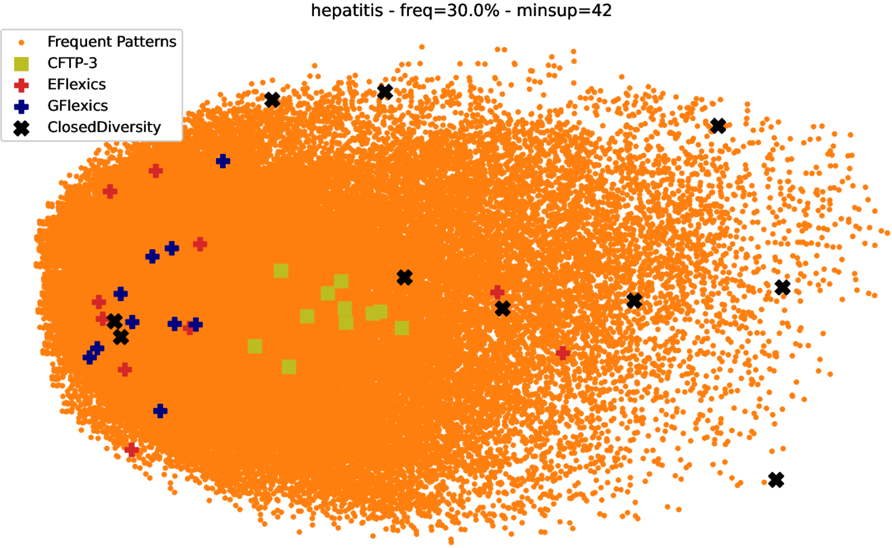
- Use a vector X of Boolean variables $(X_1, \ldots, X_{|\mathcal{I}|})$ for representing item sets
- X^+ will denote the set of present items,
- $\textsc{ClosedDiversity}_{\mathcal{D},\theta}(X, \mathcal{H}, J_{max})$ holds if and only if :
  (1) $freq(X^+) \geq \theta$ and X^+ is closed ➠ $\textsc{ClosedPatterns}$
  (2) X^+ is diverse, $\forall\, H \in \mathcal{H}, LB_J(X^+, H) \leq J_{max}$.

Two filtering rules : Let X_{Div} be the set of items filtered by (Rule #1)

1. remove 1 from $dom(X_i)$ if $\exists\, H \in \mathcal{H}$ s.t. $LB_J(X^+ \cup \{i\}, H) > J_{max}$
2. remove 1 from $dom(X_i)$ if $\exists\, k \in X_{Div}$ s.t. $cover(X^+ \cup \{i\}) \subseteq cover(X^+ \cup \{k\})$, then $LB_J(X^+ \cup \{i\}, H) > LB_J(X^+ \cup \{k\}, H) > J_{max}$

Time complexity: $O(n \times (n \times m))$

Comparative Exploration of Extracted Pattern Landscapes



False positives

ClosedDiversity ensures that $\forall H \in \mathcal{H},\ LB_J(P,H) \leq J_{max}$

$$LB_J(P,H) \leq J_{max} : \begin{cases} J_{\mathcal{V}}(P,H) \leq J_{max} \\ J_{\mathcal{V}}(P,H) > J_{max} \end{cases}$$

False positives

$\textsc{ClosedDiversity}$ ensures that $\forall H \in \mathcal{H},\ LB_J(P,H) \leq J_{max}$

$$LB_J(P,H) \leq J_{max} : \begin{cases} J_{\mathcal{V}}(P,H) \leq J_{max} \\ J_{\mathcal{V}}(P,H) > J_{max} \Rightarrow \textit{false positive} \end{cases}$$

False positives

ClosedDiversity ensures that $\forall H \in \mathcal{H},\ LB_J(P,H) \leq J_{max}$

$$LB_J(P,H) \leq J_{max} : \begin{cases} J_{\mathcal{V}}(P,H) \leq J_{max} \\ J_{\mathcal{V}}(P,H) > J_{max} \Rightarrow \textit{false positive} \end{cases}$$

Proposition : adding the Jaccard test before any update of $\mathcal{H}$
➠ ClosedDiv+Jaccard

Experiments

- UCI datasets
- Comparisons:
  1. ClosedP
  2. ClosedP+Jaccard
  3. ClosedDiv
  4. ClosedDiv+Jaccard
  5. FullCP
  6. Picker
  7. PatternsTeam
  8. Flexics (EFlexics et GFlexics)
  9. Gibbs
  10. CFTP
- Implementation: `Choco solver, timeout = 24 hours`

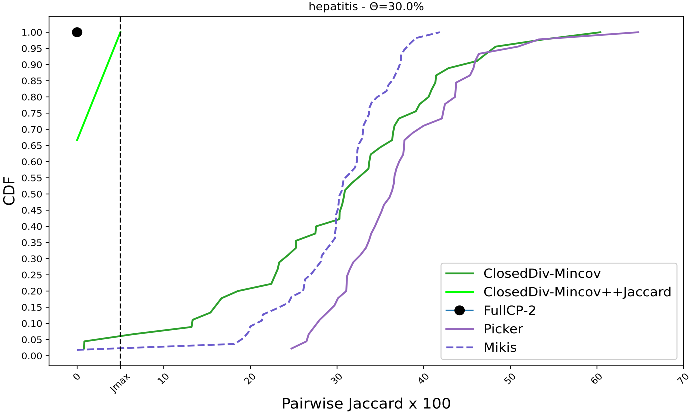
Quality of diversification

Let a set of patterns $\mathcal{H} = \{H_1, \ldots, H_k\}$

$$CDF_{\mathcal{H}}(\tau) = \#\{(i,j) | J_{\mathcal{V}}(H_i, H_j) \leq \tau, 1 \leq i < j \leq k\} \cdot \frac{2}{k(k-1)}$$

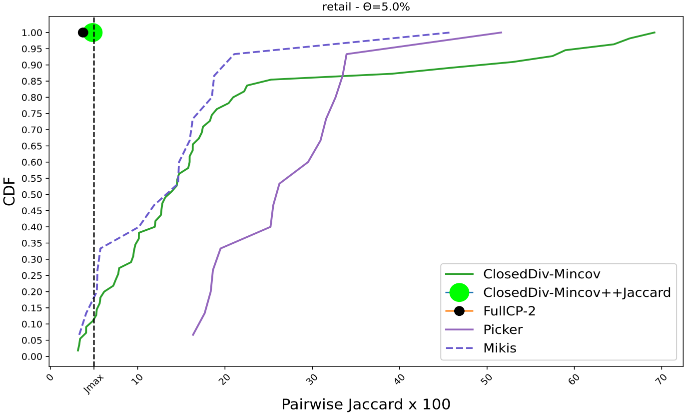
➟ Cumulative distribution function on the Jaccard indices of each pair of patterns

Quality of diversification : heuristic approaches



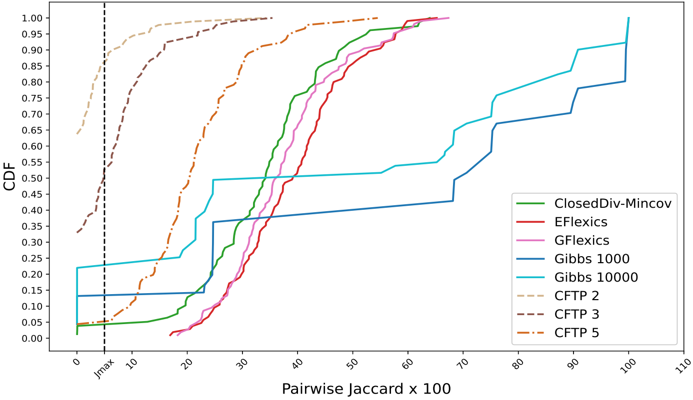
(a) HEPATITIS ($\theta = 30\%$)

Quality of diversification : heuristic approaches



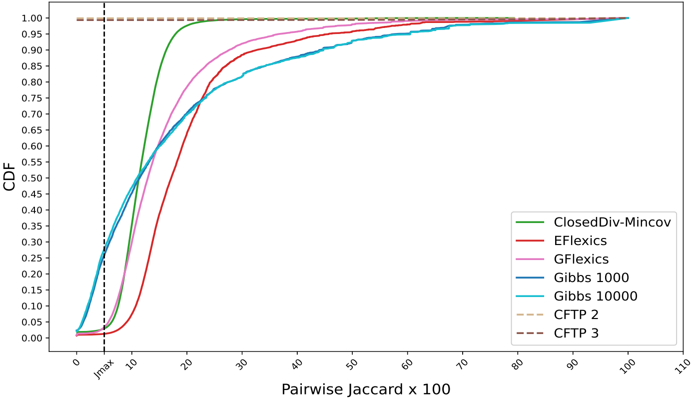
(b) RETAIL ($\theta = 5\%$)

Quality of diversification : sampling approaches



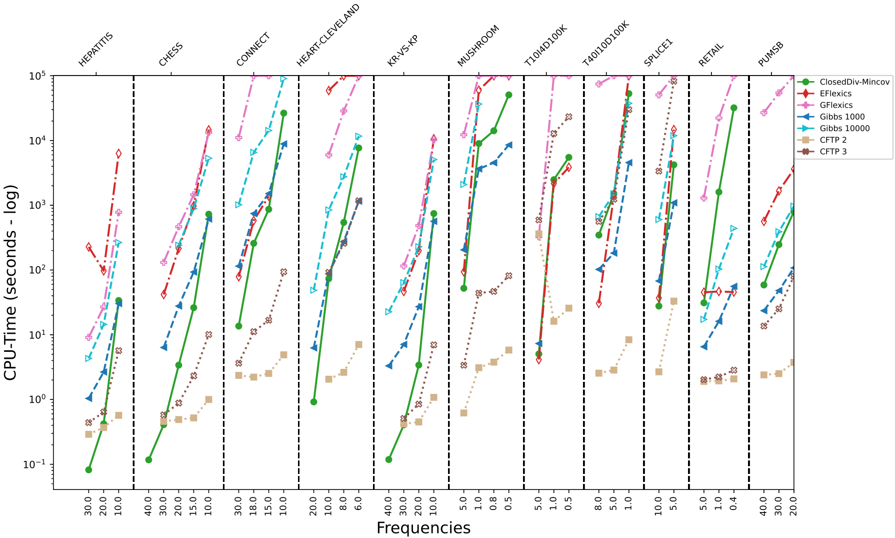
(c) HEPATITIS ($\theta = 10\%$)

Quality of diversification : sampling approaches

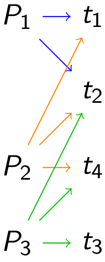


(d) SPLICE1 ($\theta = 10\%$)

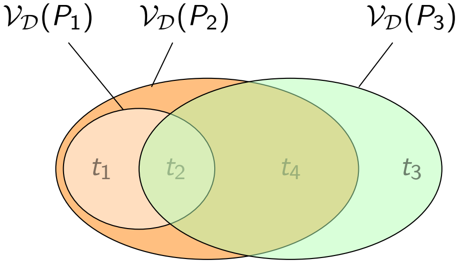
CPU times : sampling approaches



Jaccard index vs. Overlap



(e)

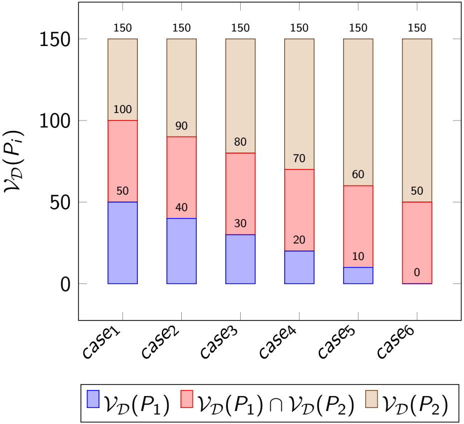


(f)

Pairwise comparison	**Jaccard Index**	**Overlap Coefficient**
$P1$ vs. $P2$	$\frac{\lvert P1 \cap P2 \rvert}{\lvert P1 \cup P2 \rvert} = \frac{2}{3} \approx 0.66$	$\frac{\lvert P1 \cap P2 \rvert}{\min(\lvert P1 \rvert, \lvert P2 \rvert)} = \frac{2}{2} = 1.0$
$P1$ vs. $P3$	$\frac{\lvert P1 \cap P3 \rvert}{\lvert P1 \cup P3 \rvert} = \frac{1}{4} = 0.25$	$\frac{\lvert P1 \cap P3 \rvert}{\min(\lvert P1 \rvert, \lvert P3 \rvert)} = \frac{1}{2} = 0.5$

Table: Comparison of Similarity Measures

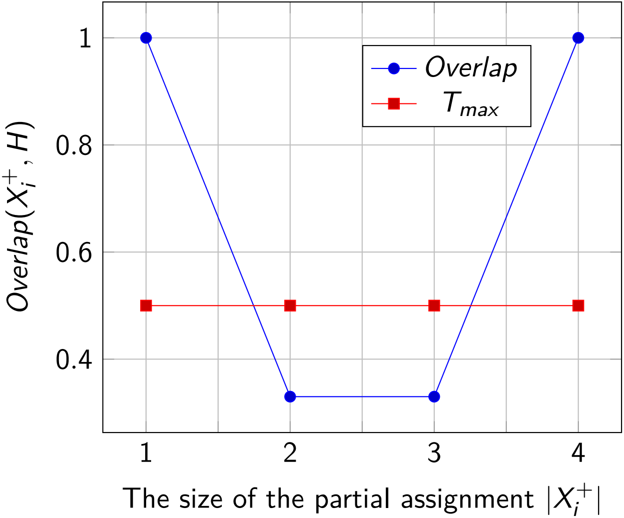
Sensitivity analysis of the Overlap coefficient



case#	$\lvert\mathcal{V}_{\mathcal{D}}(P_1)\rvert$	$\lvert\mathcal{V}_{\mathcal{D}}(P_2)\rvert$	**(J)**	**(O)**
$case_1$	100	100	0.333	0.500
$case_2$	110	90	0.333	0.556
$case_3$	120	80	0.333	0.625
$case_4$	130	70	0.333	0.714
$case_5$	140	60	0.333	0.833
$case_6$	150	50	0.333	1.0

➠ By considering the proportion of the smaller set contained within the larger set, the Overlap coefficient offers a more informative and nuanced measure of set similarity.

Anti-monotonicity of the Overlap Coefficient



- For $\mathcal{H} = \{AE\}$, $X^+ = \{A\} \subseteq \{A, C\} \subseteq \{A, C, D\} \subseteq \{A, C, D, E\}$, $Overlap(A, \mathcal{H}) = 1 \geq T_{max}$ whereas $Overlap(AC, \mathcal{H}) = .33 \leq T_{max}$.

Relaxations of the overlap coefficient (M. Douad et al., APIN 2025)

(i) **A lower bound** which allows to prune non-diverse patterns

$$LB_{OC}(P, H) = \begin{cases} \frac{\theta - |\mathcal{V}_H^{pr}(P)|}{\min(|\mathcal{V}_\mathcal{D}(P)|, |\mathcal{V}_\mathcal{D}(H)|)} & \text{if } (\mathcal{V}_H^{pr}(P) < \theta) \\ 0 & \text{else} \end{cases}$$

(ii) **An upper bound** to find patterns ensuring diversity

$$UB_{OC}(P, H) = \min\left(\frac{|\mathcal{V}_\mathcal{D}(P) \cap \mathcal{V}_\mathcal{D}(H)|}{\theta}, 1\right)$$

Overlap lower and upper bounds

- Monotonicity of LB_{OC}: Let $H \in \mathcal{H}$ be an itemset. For any two patterns $Q \subseteq P$, the relationship $LB_{OC}(P,\ H) \leq LB_{OC}(Q,\ H)$ holds.

  ➟ If $LB_{OC}(P, H) > J_{max} \Rightarrow J_{\mathcal{V}}(P, H) > J_{max} \Rightarrow P$ **is not diverse** (any super-itemset $P' \supseteq P$ **will not be diverse**)

- Anti-monotonicity of UB_{OC}: Let $H \in \mathcal{H}$ be an itemset. For any two patterns $P \subseteq Q$, the relationship $UB_{OC}(P,\ H) \geq UB_{OC}(Q,\ H)$ holds.

  ➟ If $UB_{OC}(P, H) \leq J_{max} \Rightarrow J_{\mathcal{V}}(P, H) \leq J_{max} \Rightarrow P$ **is diverse** (any super-itemset $P' \supseteq P$ **will also be diverse**)

- A new global constraint $\mathtt{OverlapDiv}_{\mathcal{D},\theta}(X, \mathcal{H}, T_{max})$ for mining pairwise diverse patterns that exploits the two previous relaxations.

$\texttt{OverlapDiv}_{\mathcal{D},\theta}(X, \mathcal{H}, T_{max})$ (A. Douad et al., APIN 2025)

- Use a vector X of Boolean variables $(X_1, \ldots, X_{|\mathcal{I}|})$ for representing item sets
- X^+ will denote the set of present items,
- $\texttt{OverlapDiv}_{\mathcal{D},\theta}(X, \mathcal{H}, T_{max})$ holds if and only if :

  X^+ is diverse; $\forall\, H \in \mathcal{H}, LB_{OC}(X^+, H) \leq T_{max}$

Two filtering rules : Let X_{Div} be the set of items filtered by (Rule #1)

$$1 \notin Dom(X_i) \text{ iff } \begin{cases} \exists\ \mathrm{H} \in \mathcal{H} \text{ s.t. } \mathrm{LB_{OC}}(\mathrm{X}^+ \cup \{\mathrm{i}\}, \mathrm{H}) > \mathrm{T_{max}} & \textbf{(R1)} \\ \exists \mathrm{k} \in \mathrm{X_{Div}} \text{ s.t. } \mathcal{V}_{\mathcal{D}}(\mathrm{X}^+ \cup \{\mathrm{i}\}) \subseteq \mathcal{V}_{\mathcal{D}}(\mathrm{X}^+ \cup \{\mathrm{k}\}) & \textbf{(R2)} \end{cases}$$

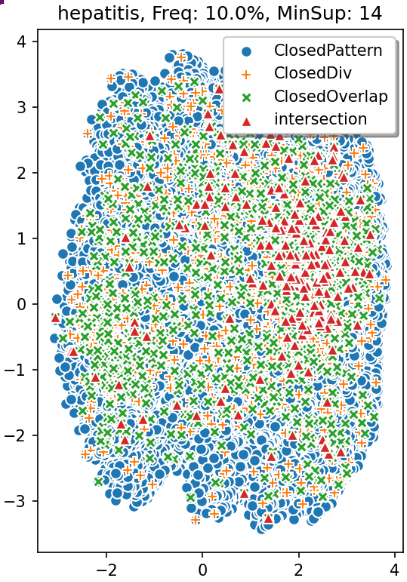
Time complexity: $O(n \times (n \times m))$

A CP model for mining diverse closed and frequent itemsets

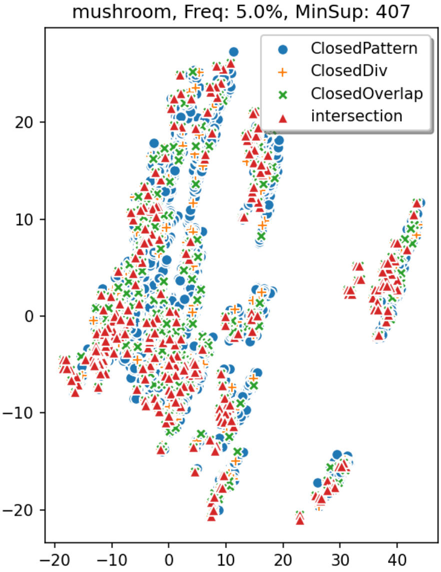
$$\texttt{ClosedOverlap}_{\mathcal{D},\theta}(X, c, T_{max}, \mathcal{H}) \equiv \begin{cases} \texttt{OverlapDiv}_{\mathcal{D},\theta}(X, T_{max}, \mathcal{H}) & (1) \\ \texttt{CoverSize}_{\mathcal{D},\theta}(X, c) \wedge c \geq \theta & (2) \\ \texttt{CoverClosure}_{\mathcal{D}}(X) & (3) \end{cases}$$

- **`OverlapDiv`:** Enforces diversity by controlling the overlap between the current itemset and the previously mined itemsets in $\mathcal{H}$.
- **`CoverSize` (schaus et al., 2017):** Ensures frequency by requiring that the cover size of the itemset satisfies: $\text{Freq}_{\mathcal{D}}(X) \geq \theta$.
- **`CoverClosure`:** Guarantees closure by ensuring that the itemset is maximal w.r.t. the frequency measure, i.e., $Clos_{freq}(X) = X$.

PCA of Extracted Pattern Landscapes



(g)



(h)

Optimization model or mining k-diverse frequent itemsets

Both `ClosedOverlap` and ClosedDiversity are sensitive to order in which patterns are explored ➠ can significantly influence the output

Formulate the initial problem as an optimization one: ➠ select the best pattern set S^* of size k maximizing some quality measure ϕ over all subsets of $\mathcal{P}$

Joint entropy measure as a quality measure:

Definition (Joint entropy of an itemset)

Let $P = \{i_1, \ldots, i_n\}$ be an itemset and $c = (c_1, \ldots, c_n) \in \{0,1\}^n$ a tuple of binary values. The joint entropy of P is given as

$$\mathbf{H}(P) = -\sum_{\mathbf{c_i} \in \{0,1\}^n} p(i_1 = c_1, , \ldots, i_n = c_n) \log_2 p(i_1 = c_1, , \ldots, i_n = c_n) \tag{1}$$

Join entropy-based formulation for minimizing k-pattern set redundancy

Each pattern $H_i \in \mathcal{H}$ is represented as a **binary vector**, within a $|\mathcal{D}| \times |\mathcal{H}|$ matrix, indicating the transactions where H_i is present.

$$\mathcal{H} = \{ABCD, AE, BE\}$$

$\{A, B, C, D\}$	$\{A, E\}$	$\{B, E\}$
0	0	1
0	0	0
0	0	0
1	1	1
1	0	0
0	0	0
0	1	0
0	1	0

Join entropy-based formulation for minimizing k-pattern set redundancy

The Entropy Computation exploits the binary vectors $c \in \{0,1\}^k$ derived from the binary matrix:

$$H(\mathcal{H}) = -\sum_{c \in \{0,1\}^k} \frac{freq(H_1 = c_1, \ldots, H_k = c_k)}{|\mathcal{D}|} log_2(\frac{freq(H_1 = c_1, \ldots, H_k = c_k)}{|\mathcal{D}|})$$

- Diversity reflects the variability or balance in the presence of patterns across transactions:

Join entropy-based formulation for minimizing k-pattern set redundancy

The Entropy Computation exploits the binary vectors $c \in \{0,1\}^k$ derived from the binary matrix:

$$H(\mathcal{H}) = -\sum_{c\in\{0,1\}^k} \frac{freq(H_1 = c_1, \ldots, H_k = c_k)}{|\mathcal{D}|} log_2(\frac{freq(H_1 = c_1, \ldots, H_k = c_k)}{|\mathcal{D}|})$$

- Diversity reflects the variability or balance in the presence of patterns across transactions:
  - Pattern $\{A, B, C, D\}$: Appears in 2 transactions (t_4,t_5)
  - Pattern $\{A, E\}$: Appears in 3 transactions (t_4,t_7,t_8)
  - Pattern $\{B, E\}$: Appears in 2 transactions (t_1,t_4)

Join entropy-based formulation for minimizing k-pattern set redundancy

The Entropy Computation exploits the binary vectors $c \in \{0,1\}^k$ derived from the binary matrix:

$$H(\mathcal{H}) = -\sum_{c \in \{0,1\}^k} \frac{freq(H_1 = c_1, \ldots, H_k = c_k)}{|\mathcal{D}|} log_2\left(\frac{freq(H_1 = c_1, \ldots, H_k = c_k)}{|\mathcal{D}|}\right)$$

- Diversity reflects the variability or balance in the presence of patterns across transactions:
  - If all patterns appear in all transactions, diversity is **low** because there is no variability.
  - If each pattern appears in a unique set of transactions, diversity is **high**.

Join entropy-based formulation for minimizing k-pattern set redundancy

The Entropy Computation exploits the binary vectors $c \in \{0,1\}^k$ derived from the binary matrix:

$$H(\mathcal{H}) = -\sum_{c \in \{0,1\}^k} \frac{freq(H_1 = c_1, \ldots, H_k = c_k)}{|\mathcal{D}|} log_2(\frac{freq(H_1 = c_1, \ldots, H_k = c_k)}{|\mathcal{D}|})$$

- Unique bit codes in this matrix are: $\{001, 000, 111, 100, 010\}$
- Occurrences: 001: 1, 000: 3, 111: 1, 100: 1, 010: 2
- $P(001) = \frac{1}{8},\ P(000) = \frac{3}{8},\ P(111) = \frac{1}{8},\ P(100) = \frac{1}{8},\ P(010) = \frac{2}{8} = \frac{1}{4}$
- The joint entropy H is:

$$H(\mathcal{H}) = -\left(\frac{1}{8}\log_2\frac{1}{8} + \frac{3}{8}\log_2\frac{3}{8} + \frac{1}{8}\log_2\frac{1}{8} + \frac{1}{8}\log_2\frac{1}{8} + \frac{2}{8}\log_2\frac{2}{8}\right)$$
$$= \mathbf{2.1556\ bits}$$

Join entropy-based formulation for minimizing k-pattern set redundancy

- Note that the joint entropy is always between 0 and k.
- It is maximal when patterns are fully uncorrelated and uniformly distributed.
- For 3 binary variables, there are $2^3 = 8$ possible combinations.
- If each occurs exactly once, then:

$$
\begin{aligned}
H(P_1, P_2, P_3) &= -\left(\sum_{i=1}^{8} \frac{1}{8} \log_2 \frac{1}{8}\right) \\
&= -\left(8 \times \frac{1}{8} \times (-3)\right) = \mathbf{3\,bits}
\end{aligned}
$$

Join entropy-based formulation for minimizing *k*-pattern set redundancy

A Greedy Optimization Approach: Iteratively integrates patterns with the highest entropy contribution.

{A B C D}	{A E}	{B E}
0	0	1
0	0	0
0	0	0
1	1	1
1	0	0
0	0	0
0	1	0
0	1	0

Joint Entropy = 2.1556

→

{E}	{A E}	{B E}
1	0	1
1	0	0
0	0	0
1	1	1
0	0	0
0	0	0
1	1	0
1	1	0

Joint Entropy = 2.1556

→

{A B C D}	**{E}**	{B E}
0	1	1
0	1	0
0	0	0
1	1	1
1	0	0
0	0	0
0	1	0
0	1	0

Joint Entropy = 2.1556

→

{A B C D}	{A E}	**{E}**
0	0	1
0	0	1
0	0	0
1	1	1
1	0	0
0	0	0
0	1	1
0	1	1

Joint Entropy = **2.25**

Figure: Maximizing Joint Entropy: finding the best k-pattern set through candidate itemset replacement

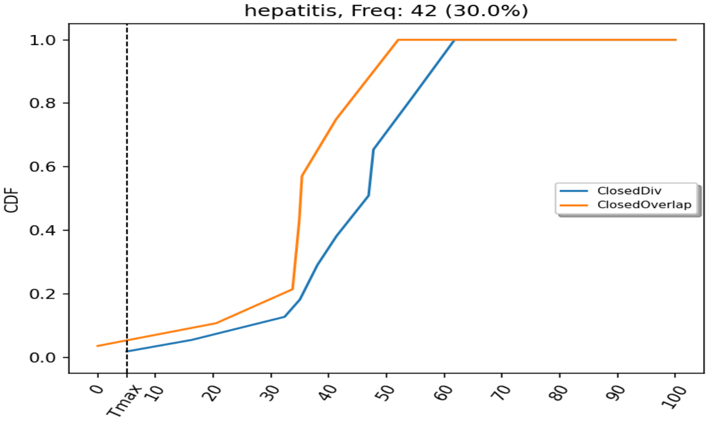
CPU times: ClosedDiversity vs. ClosedOverlap

Dataset $\mathcal{T} \times \mathcal{I}$	θ%	ClosedDiversity		ClosedOverlap	
		CPU-Time (*sec*)	# Patterns	CPU-Time (*sec*)	# Patterns
Chess	20	1.57	64	**0.35**	**56**
75 x 3196	15	12.36	237	**1.00**	**224**
49.33%	10	408.19	1621	**15.62**	**1597**
Kr-vs-kp	30	0.24	13	**0.11**	**10**
73 x 3196	20	1.41	63	**0.35**	**57**
49.32 %	10	368.50	1608	**15.30**	**1570**
Connect	30	6.92	18	**0.59**	**11**
129 x 67557	18	134.91	140	**5.76**	**104**
33.33%	15	492.53	296	**17.35**	**251**
Heart-cleveland	10	40.05	1469	**2.26**	**1394**
95 x 296	8	313.23	**4760**	**12.98**	4814
47.37%	6	4630.88	**20489**	**281.19**	22050
Splice1	10	11.30	412	**2.04**	**374**
287 x 3190	5	2316.70	7919	**112.46**	**5763**
20.91%	2	TO	NA	**112.46**	**5763**
Mushroom	5	23.65	**547**	**1.96**	551
112 x 8124	1	5172.93	9934	**202.43**	**9323**
18.75%	0.5	27656.11	23930	**1073.44**	**21974**

CPU times: ClosedDiversity vs. ClosedOverlap

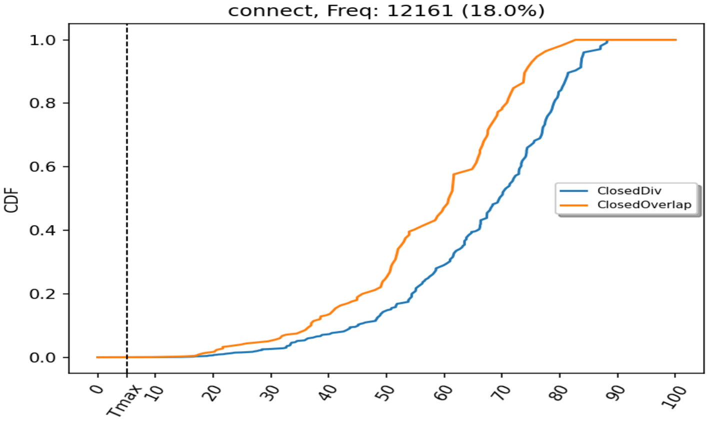
Dataset $\mathcal{T} \times \mathcal{I}$	θ%	ClosedDiversity CPU-Time (*sec*)	ClosedDiversity # Patterns	ClosedOverlap CPU-Time (*sec*)	ClosedOverlap # Patterns
T40I10D100K	8	196.77	124	**44.14**	**114**
942 x 100000	5	880.13	283	**200.19**	**271**
4.20%	1	31213.08	7216	**3523.54**	**5924**
Pumsb	40	31.33	3	**2.67**	3
2113 x 49046	30	126.25	13	**10.41**	**11**
3.50%	20	431.91	**38**	**35.04**	39
T10I4D100K	5	2.26	10	**0.62**	10
870 x 100000	1	1455.39	359	**305.70**	**351**
1.16%	0.5	2910.10	606	**645.35**	**595**
Retail	5	15.94	11	**4.49**	11
16470 x 88162	1	864.30	104	**206.57**	**103**
0.06%	0.4	12489.13	514	**3500.13**	**503**

Diversity : ClosedDiversity vs. ClosedOverlap



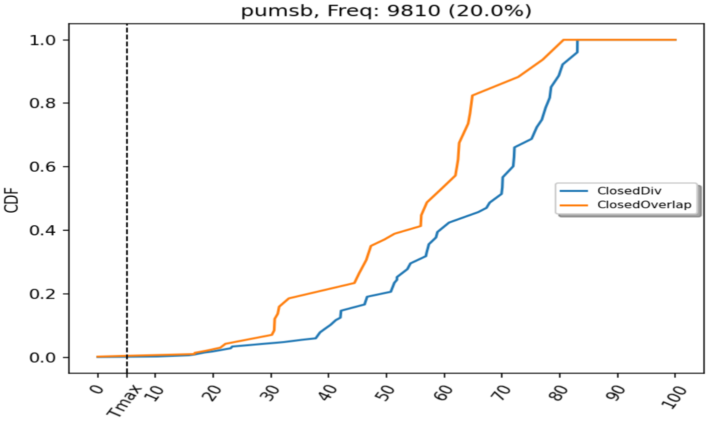
(a) Hepatitis ($\theta = 30\%$)

Diversity : CLOSEDDIVERSITY vs. ClosedOverlap



(b) CONNECT ($\theta = 18\%$)

Diversity : ClosedDiversity vs. ClosedOverlap



(c) Pumsb ($\theta = 20\%$)

Search strategies for MostDistant problem (M. Vavrille et al., 2023)

ClosedDiversity and `ClosedOverlap` global constraints work at the propagation level of the solver

Idea: Define dedicated new search strategies to orient the search towards diverse patterns ➠ no modification of the model nor the solver's inner structure

Task : Given a history $\mathcal{H}$ of solutions (initially empty), the task is to mine new patterns P ensuring a minimum diversity according to a some History score

OrientedDeterministic strategy

A vector X of Boolean variables $(X_1, \ldots, X_{|\mathcal{I}|})$ for representing a pattern P

Definition (History score)

Given a pattern P being constructed, an item i whose associated variables X_i is uninstantiated, and a history $\mathcal{H}$ of solutions, we define the history score as

$$\max_{H \in \mathcal{H}} J_{\mathcal{V}}(P \cup \{i\}, H)$$

The history score ensures a minimum diversity between $P \cup \{i\}$ and solutions of $\mathcal{H}$

Brunching heuristic : select the next uninstantiated variable $X_{i_{min}}$ that minimizes the score (a low Jaccard means a good diversity):

$$i_{min} \leftarrow \underset{\substack{1 \leq i \leq |\mathcal{I}| \\ |\mathcal{D}(X_i)|=2}}{\arg\min} \max_{H \in \mathcal{H}} J_{\mathcal{V}}(P \cup \{i\}, H)$$

➠ The search may be stuck in a local optimum, i.e. solutions will be searched repeatedly in the same space.

OrientedRandom Strategy

Select the next uninstantiated variable according to some weight distribution

Definition (variable weight)

Let W be an array of size $|\mathcal{I}|$. Given a pattern P being constructed, an item i whose associated variables X_i is uninstantiated, and a history $\mathcal{H}$ of solutions, we the define the weight of X_i as

$$W[i] = \frac{1}{\max\limits_{H \in \mathcal{H}} J_{\mathcal{V}}(P \cup \{i\}, H) + \epsilon}$$

Instantiated variables are given a weight of 0

To bias the random distribution towards small values, we have to invert the computed history score

Brunching heuristic : the next uninstantiated variable is randomly chosen with respect to the weights in W

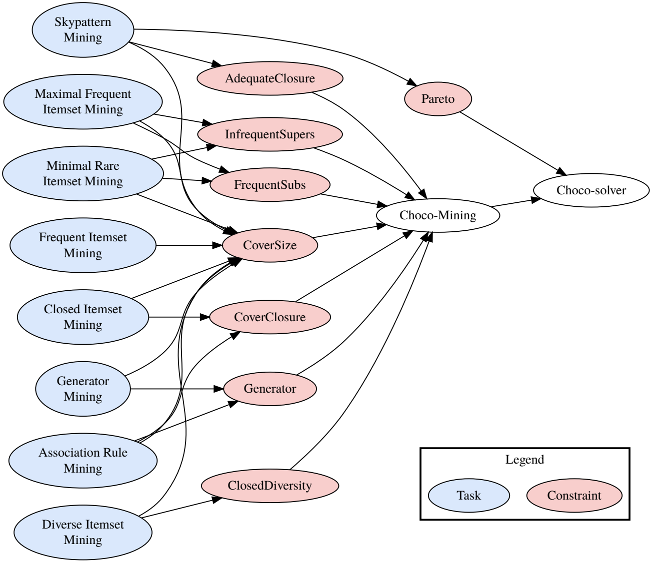
Choco-Mining: A Java Library For Itemset Mining (Vernerey et al. JOSS 2023)

- Une nouvelle bibliothèque Choco-Mining[1] qui regroupe un ensemble de contraintes pour la fouille de motifs.

- Elle permet à un utilisateur intéressé de facilement réutiliser ces contraintes dans ses propres projets.

[1] https://gitlab.com/chaver/choco-mining

Choco-Mining: A Java Library For Itemset Mining (Vernerey et al. JOSS 2023)



Summary

A new generic solution for mining diverse patterns

- Exploiting relaxations and search strategies
- Other diversity measures (entropy) ?

Exploiting Jaccard index within search strategies show improved diversity compared to ClosedDiversity