Reject By One Algorithm

Richard EMILION (MAPMO Lab., Orléans University, France)
Gérard LÉVY (University Paris Dauphine, France)

DATA ANALYSIS AND MODELING LAB
Palacky University, Olomouc, Czech Republic

european social fund in the czech republic
EUROPEAN UNION
MINISTRY OF EDUCATION, YOUTH AND SPORTS
OP Education for Competitiveness
INVESTMENTS IN EDUCATION DEVELOPMENT

Outline

- **I - Introduction**
- **II - Galois connection properties**
- **III - Expectation of RCL size**
- **IV - Sampling**

I - Introduction

I. Introduction

- Binary Context $\mathcal{C} = (\mathcal{O}, \mathcal{A}, \mathcal{D})$, $\mathcal{L}$ its concept lattice.
- Time consuming tasks : listing $\mathcal{L}$, frequent itemsets, closed frequent itemsets, associative rules
- Most of present algorithms first provide duplicate itemsets and then they eliminate duplicates: waste time
- Our aim: Proposing an efficient algorithm that does not provide duplicate closed frequent itemsets.
- Close by one idea is quite a natural idea that will be used
- Our new idea: Reject by One
- A parallel version can be derived easily

I. Introduction

- Binary Context $\mathcal{C} = (\mathcal{O}, \mathcal{A}, \mathcal{D})$, $\mathcal{L}$ its concept lattice.
- Time consuming tasks : listing $\mathcal{L}$, frequent itemsets, closed frequent itemsets, associative rules
- Most of present algorithms first provide duplicate itemsets and then they eliminate duplicates: waste time
- Our aim: Proposing an efficient algorithm that does not provide duplicate closed frequent itemsets.
- Close by one idea is quite a natural idea that will be used
- Our new idea: Reject by One
- A parallel version can be derived easily

• Binary Context $\mathcal{C} = (\mathcal{O}, \mathcal{A}, \mathcal{D})$, $\mathcal{L}$ its concept lattice.

• Time consuming tasks : listing $\mathcal{L}$, frequent itemsets, closed frequent itemsets, associative rules

• Most of present algorithms first provide duplicate itemsets and then they eliminate duplicates: waste time

• Our aim: Proposing an efficient algorithm that does not provide duplicate closed frequent itemsets.

• Close by one idea is quite a natural idea that will be used

• Our new idea: Reject by One

• A parallel version can be derived easily

I. Introduction

• Binary Context $\mathcal{C} = (\mathcal{O}, \mathcal{A}, \mathcal{D})$, $\mathcal{L}$ its concept lattice.

• Time consuming tasks : listing $\mathcal{L}$, frequent itemsets, closed frequent itemsets, associative rules

• Most of present algorithms first provide duplicate itemsets and then they eliminate duplicates: waste time

• Our aim: Proposing an efficient algorithm that does not provide duplicate closed frequent itemsets.

• Close by one idea is quite a natural idea that will be used

• Our new idea: Reject by One

• A parallel version can be derived easily

I. Introduction

• Binary Context $\mathcal{C} = (\mathcal{O}, \mathcal{A}, \mathcal{D})$, $\mathcal{L}$ its concept lattice.

• Time consuming tasks : listing $\mathcal{L}$, frequent itemsets, closed frequent itemsets, associative rules

• Most of present algorithms first provide duplicate itemsets and then they eliminate duplicates: waste time

• Our aim: Proposing an efficient algorithm that does not provide duplicate closed frequent itemsets.

• Close by one idea is quite a natural idea that will be used

• Our new idea: Reject by One

• A parallel version can be derived easily

I. Introduction

- Binary Context $\mathcal{C} = (\mathcal{O}, \mathcal{A}, \mathcal{D})$, $\mathcal{L}$ its concept lattice.
- Time consuming tasks : listing $\mathcal{L}$, frequent itemsets, closed frequent itemsets, associative rules
- Most of present algorithms first provide duplicate itemsets and then they eliminate duplicates: waste time
- Our aim: Proposing an efficient algorithm that does not provide duplicate closed frequent itemsets.
- Close by one idea is quite a natural idea that will be used
- Our new idea: Reject by One
- A parallel version can be derived easily

I. Introduction

- Binary Context $\mathcal{C} = (\mathcal{O}, \mathcal{A}, \mathcal{D})$, $\mathcal{L}$ its concept lattice.
- Time consuming tasks : listing $\mathcal{L}$, frequent itemsets, closed frequent itemsets, associative rules
- Most of present algorithms first provide duplicate itemsets and then they eliminate duplicates: waste time
- Our aim: Proposing an efficient algorithm that does not provide duplicate closed frequent itemsets.
- Close by one idea is quite a natural idea that will be used
- Our new idea: Reject by One
- A parallel version can be derived easily

I. Introduction

• Binary Context $\mathcal{C} = (\mathcal{O}, \mathcal{A}, \mathcal{D})$, $\mathcal{L}$ its concept lattice.

• Time consuming tasks : listing $\mathcal{L}$, frequent itemsets, closed frequent itemsets, associative rules

• Most of present algorithms first provide duplicate itemsets and then they eliminate duplicates: waste time

• Our aim: Proposing an efficient algorithm that does not provide duplicate closed frequent itemsets.

• Close by one idea is quite a natural idea that will be used

• Our new idea: Reject by One

• A parallel version can be derived easily

II - Galois connection

II.1 Intent Extent Closure Mappings

$\mathcal{E} = 2^{\mathcal{O}}, \;\; \mathcal{F} = 2^{\mathcal{A}}$

Intent decreasing mapping $f : \mathcal{E} \longrightarrow \mathcal{F}$ defined as follows:

$$\forall O \in \mathcal{E} \;\; f(O) = \{a \in \mathcal{A} : o \, \mathcal{R} \, a \,, \forall \, o \in O\} \text{ if } O \neq \emptyset, f(\emptyset) = \mathcal{A}. \tag{1}$$

Extent decreasing mapping $g : \mathcal{F} \longrightarrow \mathcal{E}$ defined as follows:

$$\forall A \in \mathcal{F} \;\; g(A) = \{o \in \mathcal{O} : o \, \mathcal{R} \, a \,, \forall a \; \in A\} \text{ if } A \neq \emptyset, g(\emptyset) = \mathcal{O}. \tag{2}$$

Closure mappings

$$h = g \circ f : \mathcal{E} \longrightarrow \mathcal{E} \text{ and } k = f \circ g : \mathcal{F} \longrightarrow \mathcal{F} \text{ are extensive} \tag{3}$$

$$h \circ h = h, \;\; k \circ k = k \tag{4}$$

Notations: $\wedge$ stand for $\cap$ and $\leq$ stands for $\subseteq$

Observe that

$$f(\{o\}) = \{a \in \mathcal{A} : o\,\mathcal{R}\,a\} \quad \text{yields} \quad f(O) = \wedge_{o \in O} f(\{o\}) \tag{5}$$

$$f(O_1 \cup O_2) = f(O_1) \wedge f(O_2), \ \forall\, O_1,\ O_2 \in \mathcal{E} \tag{6}$$

and

$$g(\{a\}) = \{o \in \mathcal{O} : o\,\mathcal{R}\,a\} \ \text{yields} \ f(O) = \{a \in \mathcal{A} : O \leq g(\{a\})\} \tag{7}$$

Similarly

$$g(A) = \wedge_{a \in A} g(\{a\}) \ \ g(A) = \{o \in \mathcal{O} : A \leq f(\{o\})\} \tag{8}$$

$$g(A \cup A_2) = g(A) \wedge g(A_2), \ \forall\, A,\ A_2 \in \mathcal{F} \tag{9}$$

$$h(O) = g(f(O)) = \{o \in \mathcal{O} : f(O) \leq f(\{o\})\}. \tag{10}$$

$$k(A) = f(g(A)) = \{a \in \mathcal{A} : g(A) \leq g(\{a\})\}. \tag{11}$$

III - Attracted Set

III.1 Closing by one and truncation operator

Any closed set containing itemset A and item $j \notin A$ contains $k(A \cup \{j\})$ and

$$k(A \cup \{j\}) = \{a \in \mathcal{A} : g(A \cup \{j\}) \leq g(\{a\})\} \quad (12)$$
$$= \{a \in \mathcal{A} : g(A) \wedge g(\{j\}) \leq g(\{a\})\} \quad (13)$$
$$= A \cup \{a \in \mathcal{A} \setminus A : g(A) \wedge g(\{j\}) \leq g(\{a\})\} \quad (14)$$

As $g(A) \wedge g(\{j\}) \leq g(A)$, this is equivalent to

$$k(A \cup \{j\}) = A \cup \{a \in \mathcal{A} \setminus A : g(A) \wedge g(\{j\}) \leq g(A) \wedge g(\{a\})\} \quad (15)$$
$$= A \cup \{a \in \mathcal{A} \setminus A : T_A(j) \leq T_A(a)\} \quad (16)$$

where
Definition Let $A \in \mathcal{F}$ be an itemset

$$T_A : \mathcal{A} \longrightarrow \mathcal{E} \text{ par } T_A(a) = g(A) \wedge g(\{a\}).$$

Note that $T_A(a) = g(A) \iff \in k(A)$.

III.2 Attracted set

Proposition 1 Any closed set containing itemset A and item $j \notin A$ contains the attracted set $Atr(A, j)$ defined as

$$Atr(A, j) = \{a \in \mathcal{A} \setminus A : T_A(j) \leq T_A(a)\} \tag{17}$$

and contains $k(A \cup \{j\})$ with

$$k(A \cup \{j\}) = A \cup Atr(A, j) \tag{18}$$

IV - Rejected Set

IV.1 Rejected set

We have seen that is easy to describe a closed itemset containing A and item j.
What happen if a closed itemset contains A but does not contain item j ?
Let $a, j \in \mathcal{A} \setminus A$, by definition $a \in Atr(A, j) \iff T_A(j) \leq T_A(a)$
Let $a, j \in \mathcal{A} \setminus A$, by definition $j \in Atr(A, a) \iff T_A(a) \leq T_A(j)$
Let $a, j \in \mathcal{A} \setminus A$.

$$j \in Atr(A, a) \iff T_A(j) \leq T_A(j) \iff a \in \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Key observation

$$j \notin Atr(A, a) \iff a \notin \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Definition The Rejected set $Rej(A, j)$ is defined as

$$Rej(A, j) = \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\} \tag{19}$$

Proposition 2
$j \in Atr(A, a) \iff a \in Rej(A, j)$ and $j \notin Atr(A, a) \iff a \notin Rej(A, j)$

IV.1 Rejected set

We have seen that is easy to describe a closed itemset containing A and item j.
What happen if a closed itemset contains A but does not contain item j ?
Let $a, j \in \mathcal{A} \setminus A$, by definition $a \in Atr(A, j) \Longleftrightarrow T_A(j) \leq T_A(a)$
Let $a, j \in \mathcal{A} \setminus A$, by definition $j \in Atr(A, a) \Longleftrightarrow T_A(a) \leq T_A(j)$
Let $a, j \in \mathcal{A} \setminus A$.

$$j \in Atr(A, a) \Longleftrightarrow T_A(j) \leq T_A(j) \Longleftrightarrow a \in \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Key observation

$$j \notin Atr(A, a) \Longleftrightarrow a \notin \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Definition The Rejected set $Rej(A, j)$ is defined as

$$Rej(A, j) = \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\} \quad (19)$$

Proposition 2
$j \in Atr(A, a) \Longleftrightarrow a \in Rej(A, j)$ and $j \notin Atr(A, a) \Longleftrightarrow a \notin Rej(A, j)$

IV.1 Rejected set

We have seen that is easy to describe a closed itemset containing A and item j.
What happen if a closed itemset contains A but does not contain item j ?
Let $a, j \in \mathcal{A} \setminus A$, by definition $a \in Atr(A, j) \iff T_A(j) \leq T_A(a)$
Let $a, j \in \mathcal{A} \setminus A$, by definition $j \in Atr(A, a) \iff T_A(a) \leq T_A(j)$
Let $a, j \in \mathcal{A} \setminus A$.

$$j \in Atr(A, a) \iff T_A(j) \leq T_A(j) \iff a \in \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Key observation

$$j \notin Atr(A, a) \iff a \notin \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Definition The Rejected set $Rej(A, j)$ is defined as

$$Rej(A, j) = \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\} \tag{19}$$

Proposition 2

$j \in Atr(A, a) \iff a \in Rej(A, j)$ and $j \notin Atr(A, a) \iff a \notin Rej(A, j)$

IV.1 Rejected set

We have seen that is easy to describe a closed itemset containing A and item j.
What happen if a closed itemset contains A but does not contain item j ?
Let $a, j \in \mathcal{A} \setminus A$, by definition $a \in Atr(A, j) \iff T_A(j) \leq T_A(a)$
Let $a, j \in \mathcal{A} \setminus A$, by definition $j \in Atr(A, a) \iff T_A(a) \leq T_A(j)$
Let $a, j \in \mathcal{A} \setminus A$.

$$j \in Atr(A, a) \iff T_A(j) \leq T_A(j) \iff a \in \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Key observation

$$j \notin Atr(A, a) \iff a \notin \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Definition The Rejected set $Rej(A, j)$ is defined as

$$Rej(A, j) = \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\} \tag{19}$$

Proposition 2
$j \in Atr(A, a) \iff a \in Rej(A, j)$ and $j \notin Atr(A, a) \iff a \notin Rej(A, j)$

IV.1 Rejected set

We have seen that is easy to describe a closed itemset containing A and item j.
What happen if a closed itemset contains A but does not contain item j ?
Let $a, j \in \mathcal{A} \setminus A$, by definition $a \in Atr(A, j) \iff T_A(j) \leq T_A(a)$
Let $a, j \in \mathcal{A} \setminus A$, by definition $j \in Atr(A, a) \iff T_A(a) \leq T_A(j)$
Let $a, j \in \mathcal{A} \setminus A$.

$$j \in Atr(A, a) \iff T_A(j) \leq T_A(j) \iff a \in \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Key observation

$$j \notin Atr(A, a) \iff a \notin \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Definition The Rejected set $Rej(A, j)$ is defined as

$$Rej(A, j) = \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\} \tag{19}$$

Proposition 2

$j \in Atr(A, a) \iff a \in Rej(A, j)$ and $j \notin Atr(A, a) \iff a \notin Rej(A, j)$

We have seen that is easy to describe a closed itemset containing A and item j.
What happen if a closed itemset contains A but does not contain item j ?
Let $a, j \in \mathcal{A} \setminus A$, by definition $a \in Atr(A, j) \iff T_A(j) \leq T_A(a)$
Let $a, j \in \mathcal{A} \setminus A$, by definition $j \in Atr(A, a) \iff T_A(a) \leq T_A(j)$
Let $a, j \in \mathcal{A} \setminus A$.

$$j \in Atr(A, a) \iff T_A(j) \leq T_A(j) \iff a \in \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Key observation

$$j \notin Atr(A, a) \iff a \notin \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\}$$

Definition The Rejected set $Rej(A, j)$ is defined as

$$Rej(A, j) = \{a \in \mathcal{A} \setminus A : T_A(a) \leq T_A(j)\} \tag{19}$$

Proposition 2
$j \in Atr(A, a) \iff a \in Rej(A, j)$ and $j \notin Atr(A, a) \iff a \notin Rej(A, j)$

Main Proposition 3
Let $A \subseteq F$ with F closed and let $j \not inF$ then $F \cap Rej(A, j) = \emptyset$

In other words if F rejects j it also rejects all elements of $Rej(A, j)$

Proof
If $A = F$ we have nothing to prove since by definition $Rej(A, j) \subseteq \mathcal{A} \setminus A = \mathcal{A} \setminus F$
Otherwise let $a \in F \setminus A$. Since F is closed, we have $Atr(A, a) \subseteq F$.
As $j \notin F$ we also have $j \notin Atr(A, a)$.
By Proposition 2 this is equivalent to $a \notin Rej(A, j)$
So, no element of $F \setminus A$ can be in $Rej(A, j)$: $(F \setminus A) \cap Rej(A, j) = \emptyset$
Since by definition we also have $A \cap \in Rej(A, j) = \emptyset$, we get $F \cap Rej(A, j) = \emptyset$

Main Proposition 3

Let $A \subseteq F$ with F closed and let $j \not{i}nF$ then $F \cap Rej(A, j) = \emptyset$

In other words if F rejects j it also rejects all elements of $Rej(A, j)$

Proof

If $A = F$ we have nothing to prove since by definition $Rej(A, j) \subseteq \mathcal{A} \setminus A = \mathcal{A} \setminus F$

Otherwise let $a \in F \setminus A$. Since F is closed, we have $Atr(A, a) \subseteq F$.

As $j \notin F$ we also have $j \notin Atr(A, a)$.

By Proposition 2 this is equivalent to $a \notin Rej(A, j)$

So, no element of $F \setminus A$ can be in $Rej(A, j)$: $(F \setminus A) \cap Rej(A, j) = \emptyset$

Since by definition we also have $A \cap \in Rej(A, j) = \emptyset$, we get $F \cap Rej(A, j) = \emptyset$

Main Proposition 3
Let $A \subseteq F$ with F closed and let $j \not inF$ then $F \cap Rej(A, j) = \emptyset$

In other words if F rejects j it also rejects all elements of $Rej(A, j)$

Proof
If $A = F$ we have nothing to prove since by definition $Rej(A, j) \subseteq \mathcal{A} \setminus A = \mathcal{A} \setminus F$
Otherwise let $a \in F \setminus A$. Since F is closed, we have $Atr(A, a) \subseteq F$.
As $j \notin F$ we also have $j \notin Atr(A, a)$.
By Proposition 2 this is equivalent to $a \notin Rej(A, j)$
So, no element of $F \setminus A$ can be in $Rej(A, j)$: $(F \setminus A) \cap Rej(A, j) = \emptyset$
Since by definition we also have $A \cap \in Rej(A, j) = \emptyset$, we get $F \cap Rej(A, j) = \emptyset$

IV.3 Maximality of Rejected Set

Proposition 4
Let $C \subseteq \mathcal{A} \setminus A$ be such that any closed itemset which contains A and does not contain j is disjoint from C. Then we have necessarily $C \subseteq Rej(A, j)$

IV - Closed and Reject by One Algorithm

IV.1 ALgorithm (Recursive version) 16 / 13

$L = \{k(\emptyset)\}$
$\beta = \alpha|\mathcal{O}|$
Procedure $FF(A, J)$
Begin
. **If** $|g(A)| < \beta$ **or** $J = \emptyset$ **exit**
. Choose any $j \in J$
. Compute $B = Atr(A, j)$ et $C = Rej(A, j)$
. **If** $B \subseteq J$ **and** $|g(A) \wedge g(B)| \geq \beta$
. **Begin**
. Add $A \cup B$ to list L
. $FF(A \cup B, J \setminus B)$
. **End**
. $FF(A, J \setminus C)$
End