The Top-k Frequent Closed Itemset Mining Using Top-k SAT Problem

Saïd Jabbour and Lakhdar Sais and Yakoub Salhi

CRIL CNRS - Univ. Artois

ECMLPKDD

Praha

26 septembre 2013

Context and objectives

Context

- Declarative data mining
- Top-k patterns mining

Goal

- General logic based framework

Contributions

- Top-k SAT problem
- Application to itemsets mining problem

Outline

1. Boolean Satisfiability (SAT)
2. Top-k SAT problem
3. Application in Data Mining
4. Experimental evaluation
5. Conclusion & Perspectives

Boolean Satisfiability Problem (SAT)

A formula Φ in conjunctive normal form (CNF) : set of clauses

$$\Phi = \underbrace{(a \vee b \vee c)}_{clause} \wedge (\neg a \vee \neg b \vee \neg c) \wedge (\neg a \vee b) \wedge (\neg b \vee c)$$

- **SAT Problem** : Φ is satisfiable ?
  - **Yes** : Return the model $\mathcal{M}$ (e.g. $\mathcal{M} = \{\neg a, b, c\}$)
  - **No** : Proof of unsatisfiability

Boolean Satisfiability Problem (SAT)

- NP-Complete Problem [Cook 71]
- Used to prove the NP-Completeness of other problems
- Spectacular progress $\rightarrow$ Modern SAT solvers
  - application instances with millions of variables and clauses
- Many applications
  - Formal Verification
  - Planning
  - Bioinformatics
  - Cryptography
  - ...
- Around SAT
  - Max-SAT, (Weighted) Partial Max-SAT, QBF, ...

- $\bar{l}$ the complementary literal of l. If $l = p$ then $\bar{l}$ is $\neg p$ and if $l = \neg p$ then $\bar{l}$ is p.
- For a set of literals L, $\bar{L}$ is defined as $\{\bar{l} \mid l \in L\}$.
- $\overline{\mathcal{M}}$ denotes the clause $\bigvee_{p \in Var(\Phi)} s(p)$, where $s(p) = p$ if $\mathcal{M}(p) = 0$, $\neg p$ otherwise.
- $\mathcal{M}(\Phi)$ the set of clauses satisfied by $\mathcal{M}$.
- Let $X \subseteq Var(\Phi)$. $\mathcal{M}(X)$ = $\mathcal{M} \cap X$ = $\{p \in X | \mathcal{M}(p) = 1\}$.
- $\mathcal{M}_{|X}$ denotes the restriction of $\mathcal{M}$ to X.

Let Φ be a propositional formula and Λ_Φ the set of models of Φ.

Definition (Preference relation)

A preference relation $\succeq$ over Λ_Φ is a reflexive and transitive binary relation (a preorder).

$\mathcal{M} \succeq \mathcal{M}'$ means that $\mathcal{M}$ is at least as preferred as $\mathcal{M}'$.

$$P(\Phi, \mathcal{M}, \succeq) = \{\mathcal{M}' \in \Lambda_\Phi \mid \mathcal{M}' \succ \mathcal{M}\}$$

where $\mathcal{M}' \succ \mathcal{M}$ means that $\mathcal{M}' \succeq \mathcal{M}$ holds but $\mathcal{M} \succeq \mathcal{M}'$ does not.

$\approx_X$ an equivalence relation over $P(\Phi, \mathcal{M}, \succeq)$:

$$\mathcal{M}' \approx_X \mathcal{M}'' \text{ iff } \mathcal{M}' \cap X = \mathcal{M}'' \cap X$$

$[P(\Phi, \mathcal{M}, \succeq)]^X$ is a partition of $P(\Phi, \mathcal{M}, \succeq)$ w.r.t $\approx_X$.

Definition (Top-k Model)

$\mathcal{M}$ is a Top-k model w.r.t. $\succeq$ and X iff $|[P(\Phi, \mathcal{M}, \succeq)]^X| \leq k - 1$.

Definition (Top-k SAT problem)

Compute the set $\mathcal{L}$ of Top-k models of Φ w.r.t $\succeq$ and X satisfying :

1. for all $\mathcal{M}$ Top-k model, there exists $\mathcal{M}' \in \mathcal{L}$ s.t. $\mathcal{M} \approx_X \mathcal{M}'$;
2. for all $\mathcal{M}$ and $\mathcal{M}'$ in $\mathcal{L}$, if $\mathcal{M} \neq \mathcal{M}'$ then $\mathcal{M} \not\approx_X \mathcal{M}'$.

Definition (δ-preference)

$\succeq$ is a δ-preference relation, if there exists a polytime function $f_{\succeq}$ from Boolean interpretations to the set of CNF formulae such that, for all $\mathcal{M}$ model of Φ and for all $\mathcal{M}'$ Boolean interpretation, $\mathcal{M}'$ is a model of $\Phi \wedge f_{\succeq}(\mathcal{M})$ iff $\mathcal{M}'$ is a model of Φ and $\mathcal{M} \not\succ \mathcal{M}'$.

- add $f_{\succeq}(\mathcal{M})$ together with $\overline{\mathcal{M}}$ to Φ allows to find models $\mathcal{M}$ that are at least as preferred as $\mathcal{M}$.
- $\rightarrow$ introduce a lower bound during the enumeration process.

Definition ($\succeq_{\Phi_s}$ preference relation)

Let $\Phi = \Phi_h \wedge \Phi_s$ be a partial MAX-SAT instance such that Φ_h is the hard part and Φ_s the soft part.

$\succeq_{\Phi_s}$ is a preference relation : $\mathcal{M} \succeq_{\Phi_s} \mathcal{M}'$ if and only if $|\mathcal{M}(\Phi_s)| \geq |\mathcal{M}'(\Phi_s)|$.

$\succeq_{\Phi_s}$ is a δ-preference relation. $f_{\succeq_{\Phi_s}}$ can be defined as :

$$f_{\succeq_{\Phi_s}}(\mathcal{M}) = (\bigwedge_{C \in \Phi_s} p_C \leftrightarrow C) \wedge \sum_{C \in \Phi_s} p_C \geq |\mathcal{M}(\Phi_s)|$$

where p_C for $C \in \Phi_s$ are fresh propositional variables.

- The Top-1 models of Φ_h with respect to $\succeq_{\Phi_s}$ and $Var(\Phi)$ correspond to the set of all solutions of Φ in Partial Max-SAT.
- $\rightarrow$ the Top-k SAT problem can be seen as a generalization of Partial MAX-SAT.

Definition (X-minimal Model)

$\mathcal{M}$ is said to be smaller than $\mathcal{M}'$ w.r.t X, written $\mathcal{M} \preceq_X \mathcal{M}'$, if $\mathcal{M} \cap X \subseteq \mathcal{M}' \cap X$.

$\mathcal{M} \preceq_X \mathcal{M}'$ means that $\mathcal{M}$ is at least as preferred as $\mathcal{M}'$.

- $\preceq_X$ is a δ-preference relation :

$$f_{\preceq_X}(\mathcal{M}) = (\bigvee_{p \in \mathcal{M} \cap X} \overline{p}) \vee \bigwedge_{p' \in X \setminus \mathcal{M}} \overline{p'}$$

Algorithm 1: Top-k SAT

Input: a CNF formula Φ, a preorder relation $\succeq$, an integer $k \geq 1$, and a set X of Boolean variables
Output: A set of Top-k models $\mathcal{L}$

```
Φ' ← Φ;
ℒ ← ∅;                                         /* Set of all Top-k models */
while (solve(Φ')) do                           /* ℳ is a model of Φ' */
    if (∃ℳ' ∈ ℒ.ℳ ≈_X ℳ' & ℳ ≻ ℳ') then
        replace(ℳ, ℳ', ℒ);
    else if (∀ℳ' ∈ ℒ.ℳ ≉_X ℳ' & |preferred(ℳ, ℒ)| < k) then
        S ← min_top(k, ℒ);
        add(ℳ, ℒ);
        remove(k, ℒ);
        S ← min_top(k, ℒ) \ S;
        Φ' ← Φ' ∧ ⋀_{ℳ'∈S} f_⪰(ℳ');
    else
        Φ' ← Φ' ∧ f_⪰(ℳ)
    Φ' ← Φ' ∧ ¬ℳ̄;
return ℒ;
```

Algorithm 2: Top-k SAT for total preference relations

Input: a CNF formula Φ, a total preorder relation $\succeq$, an integer $k \geq 1$,and a set X of Boolean variables
Output: the set of all Top-k models $\mathcal{L}$

```
Φ' ← Φ; L ← ∅;                                         /* Set of all Top-k models */
for (i ← 0 to k − 1) do
    if (solve(Φ')) then
        add(M, L);                                      /* M is a model of Φ' */
        Φ' ← Φ' ∧ ¬M|X;
    else
        return L;

Φ' ← Φ ∧ ⋀_{M∈L} ¬M ∧ ⋀_{M'∈min(L)} f_⪰(M');
while (solve(Φ')) do                                    /* M is a model of Φ' */
    if (∃M' ∈ L. M ≈_X M'  &  M ≻ M') then
        replace(M, M', L);
    else if (∀M' ∈ L. M ≉_X M') then
        S ← min(L);
        add(M, L);
        remove(k, L);
        S ← min(L) \ S;
        Φ' ← Φ' ∧ ⋀_{M'∈S} f_⪰(M');
    else
        Φ' ← Φ' ∧ f_⪰(M)
    Φ' ← Φ' ∧ ¬M;
return L;
```

Given $\mathcal{D} = \{(0, t_0), \ldots, (n-1, t_{n-1})\}$ a transaction database over a set of items $\mathcal{I}$ and k and *min* positives integers.

Frequent Itemset Mining problem

Compute

$$\mathcal{FIM}(\mathcal{D}, \lambda) = \{I \subseteq \mathcal{I} \mid \mathcal{S}(I, \mathcal{D}) \geq \lambda\}$$

Closed Itemset : I an itemset ($I \subseteq \mathcal{I}$) such that $\mathcal{S}(I, \mathcal{D}) \geq 1$. I is closed if for all itemset J such that $I \subset J$, $\mathcal{S}(J, \mathcal{D}) < \mathcal{S}(I, \mathcal{D})$.

Top-k frequent closed itemsets $\mathcal{FCIM}^k_{min}$ mining problem

Compute all closed itemsets of length at least *min* such that, for each one, there exist no more than $k-1$ closed itemsets of length at least *min* with supports greater than its support.

- Associate to each item $a \in \mathcal{I}$ a boolean variable p_a.
  - Such boolean variables encode the candidate itemset $I \subseteq \mathcal{I}$, i.e., $p_a = true$ **iff** $a \in I$.
- $\forall\ i \in \{0, \ldots, n-1\}$, associate to the *i-th* transaction a Boolean variable b_i.

A constraint to consider only the itemsets of length at least *min* :

$$\sum_{a \in \mathcal{I}} p_a \geq min \tag{1}$$

A constraint to capture all the transactions where the candidate itemset does not appear :

$$\bigwedge_{i=0}^{n-1} (b_i \leftrightarrow \bigvee_{a \in \mathcal{I} \setminus t_i} p_a) \tag{2}$$

A constraint to force the candidate itemset to be **closed** :

$$\bigwedge_{a \in \mathcal{I}} (\bigwedge_{i=0}^{n-1} \overline{b_i} \rightarrow a \in t_i) \rightarrow p_a \tag{3}$$

SAT-based Encoding for $\mathcal{FCIM}_{min}^{k}$

Proposition

Computing the Top-k closed itemsets of length at least min corresponds to computing the Top-k models of (1), (2) and (3) with respect to $\succeq_B$ *and* $X = \{p_a | a \in \mathcal{I}\}$, *where*

- $B = \{b_0, \ldots, b_{n-1}\}$ *and* $\succeq_B$:
    - $\mathcal{M} \succeq_B \mathcal{M}'$ *if and only if* $|\mathcal{M}(B)| \leq |\mathcal{M}'(B)|$.

This preorder relation is a δ-preference relation :

- $f_{\succeq_B}(\mathcal{M}) = (\sum_{i=0}^{n-1} \overline{b_i} > |\mathcal{M}(B)|)$

1. **Mining Top-k closed itemsets of length at most max**

   $\rightarrow$ Add to (2) and (3) the following constraint :

$$\sum_{a \in \mathcal{I}} p_a \leq max \qquad (4)$$

2. **Mining Top-k closed itemsets of supports at least λ (minimal support threshold).**

   Add to (2) and (3) the constraint :

$$\sum_{i=0}^{n} \overline{b_i} \geq \lambda \qquad (5)$$

We use the δ-preference relation $\succeq_B$ defined previously.

3 **Mining Top-k maximal itemsets of supports at least λ**
The encoding of this problem consists of (2) and (5).

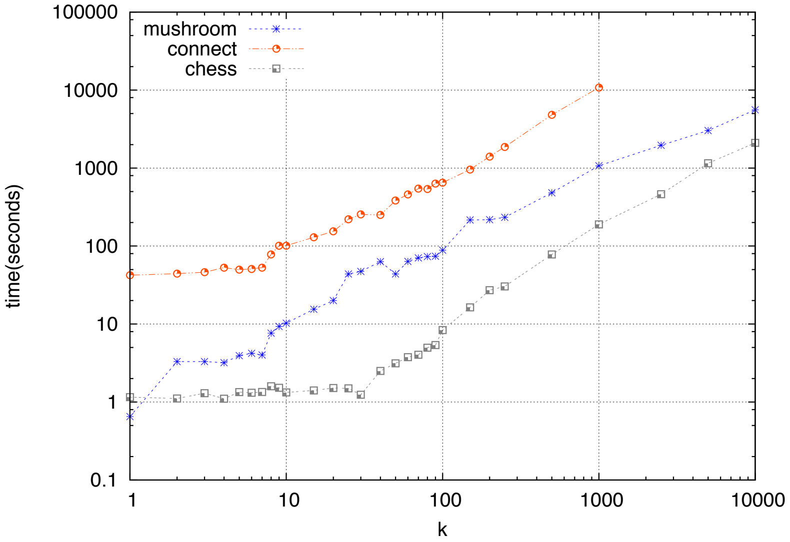
Preference relation $\succeq_\mathcal{I}$: $\mathcal{M} \succeq_\mathcal{I} \mathcal{M}'$ iff $|\mathcal{M}(\mathcal{I})| \geq |\mathcal{M}'(\mathcal{I})|$.

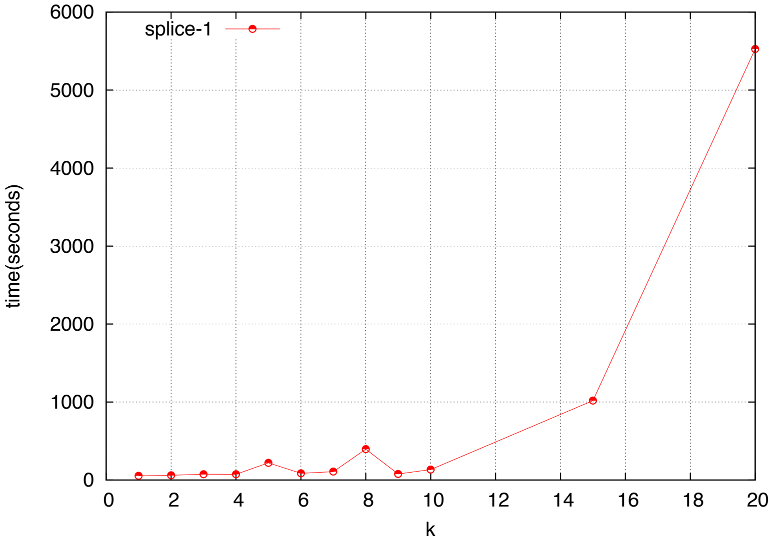
$\succeq_\mathcal{I}$ is a δ-preference :

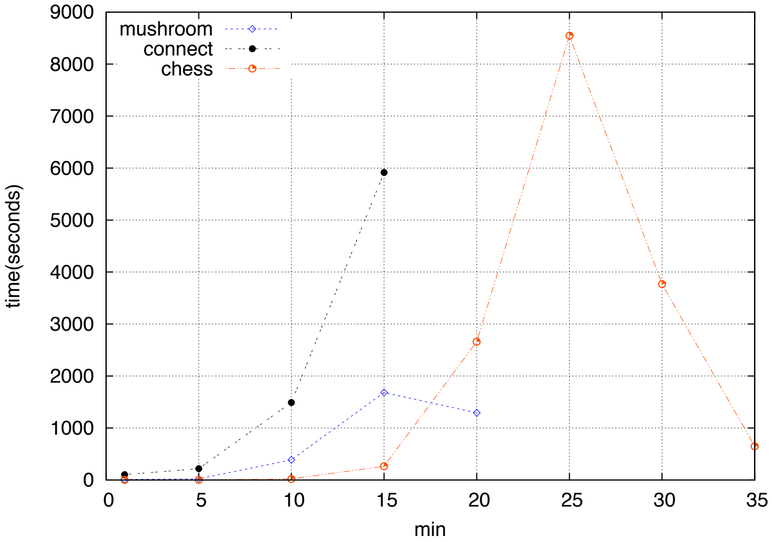
$$f_{\succeq_\mathcal{I}}(\mathcal{M}) = \sum_{a \in \mathcal{I}} p_a \geq |\mathcal{M}(\mathcal{I})|$$

- Algorithm 1 (Top-k) implemented on the top of the state-of-the-art SAT solver MiniSAT 2.2
- Sorting networks encoding to translate the cardinality Constraints [Een etal 06]
- A variety of datasets taken from the `FIMI` repository and `CP4IM`
- All the experiments were done on Intel Xeon quad-core machines with 32GB of RAM running at 2.66 Ghz.
- A timeout of 4 hours of CPU time used for each instance

instance	#trans	#items	$dens(\%)$	#vars	#clauses
zoo-1	101	36	44	173	2196
Hepatitis	137	68	50	273	4934
Lymph	148	68	40	284	6355
audiology	216	148	45	508	17575
Heart-cleveland	296	95	47	486	15289
Primary-tumor	336	31	48	398	5777
Vote	435	48	33	531	14454
Soybean	650	50	32	730	22153
Australian-credit	653	125	41	901	48573
Anneal	812	93	45	990	39157
Tic-tac-toe	958	27	33	1012	18259
german-credit	1000	112	34	1220	73223
Kr-vs-kp	3196	73	49	3342	121597
Hypothyroid	3247	88	49	3419	143043
chess	3196	75	49	3346	124797
splice-1	3190	287	21	3764	727897
mushroom	8124	119	18	8348	747635
connect	67558	129	33	67815	5877720


mushroom
connect
chess
time(seconds)
100000
10000
1000
100
10
1
0.1
1
10
100
1000
10000
k


splice-1
time(seconds)
k
6000
5000
4000
3000
2000
1000
0
0
2
4
6
8
10
12
14
16
18
20


mushroom
connect
chess
time(seconds)
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
35
min

Conclusion

- New problem Top-k SAT
- Application in data mining (Top-k itemsets mining)

Futures works

- Other data mining problems

Questions ?