An On-Line Approximation Algorithm for Mining Frequent Closed Itemsets Based on Incremental Intersection

Koji Iwanuma
University of Yamanashi
4-4-11 Takeda, Kofu-shi
Yamanashi, Japan
iwamura@yamanashi.ac.jp

Yoshitaka Yamamoto
University of Yamanashi
4-4-11 Takeda, Kofu-shi
Yamanashi, Japan
yyamamoto@yamanashi.ac.jp

Shoshi Fukuda
University of Yamanashi
4-4-11 Takeda, Kofu-shi
Yamanashi, Japan

ABSTRACT
We propose a new on-line $\epsilon$-approximation algorithm for mining closed itemsets from a transactional data stream, which is also based on the incremental/cumulative intersection principle. The proposed algorithm, called LC-CloStream, is constructed by integrating CloStream algorithm and Lossy Counting algorithm. We investigate some behaviors of the LC-CloStream algorithm. Firstly we show the incompleteness and the semi-completeness for mining all frequent closed itemsets. Next, we give the completeness of $\epsilon$-approximation for extracting frequent itemsets.

Keywords
On-line algorithm, approximation, closed itemset, intersection, completeness

1. INTRODUCTION
Intersecting transactions in a data set is an alternative characterization of closed itemsets [1, 3, 4], which naturally leads to an incremental/cumulative computation of closed itemsets in a transaction data stream. CloStream [6] is an exact-computing on-line mining algorithm, which is a direct implementation of the incremental intersecting approach. Such an incremental intersection approach, however, has great difficulties, in practice, for quitting or breaking intersections in early stages, because it is difficult to predict in advance that current intersection operations never produce any frequent closed itemsets[1].

In this paper, we propose a new on-line $\epsilon$-approximation algorithm for mining closed itemsets from a stream, which is also based on the incremental/cumulative intersection principle. The proposed algorithm, called LC-CloStream, is constructed by integrating CloStream [6] algorithm and Lossy Counting algorithm [2]. LC-CloStream succeeded in overcoming the above difficulties using $\epsilon$-approximation [2, 5].

We study fundamental properties of LC-CloStream algorithm. Firstly we show the incompleteness and the semi-completeness for mining all frequent closed itemsets in a stream. Next, we give the completeness of $\epsilon$-approximation for extracting frequent itemsets from a transaction streams.

2. PRELIMINARIES
Let $I = \{e_1, e_2, \ldots, e_m\}$ be a set of items. A non-empty subset $A$ of $I$ is called an itemset (or transaction). A transaction stream of length $N$ is a sequence of $N$ transactions $(A_1, A_2, \ldots, A_N)$. In this paper, we denote items as $a$, $b$, $c$, $\ldots$, and itemsets as $A$, $B$, $C$, $\ldots$. We also abbreviate an itemset $\{e_1, e_2, \ldots, e_m\}$ as $e_1 e_2 \cdots e_m$, for simplicity.

Let $S$ be a stream $(A_1,\ldots,A_N)$ and $B$ be an itemset. We define a multisets $K(B, t)$ at time $t$ ($1 \leq t \leq N$) as $K(B, t) = \{A_j \in S | B \subseteq A_j, 1 \leq j \leq t\}$. The frequency of $B$ at time $t$, denoted as $\text{sup}(B, t)$, is $|K(B, t)|$. Given a minimal frequency threshold $\sigma$ ($0 < \sigma < 1$), $B$ is frequent at time $t$ in $S$ if $\text{sup}(B, t) \geq \sigma \cdot t$. An itemset $B$ is closed at time $t$ in $S$ if there is no itemset $C$ such that $B \neq C$ and $B \subseteq C$ and $\text{sup}(B, t) = \text{sup}(C, t)$.

The following recursive relation makes it possible to incrementally compute closed itemsets in a stream $S$. Let $\text{CIS}(S)$ be a set of all closed itemsets in $S$ and $\circ$ be a well-known concatenation operator of two sequences.

**Proposition 1** ([1, 3]). Let $S$ be a stream $(A_1,\ldots,A_N)$. We have:

$$\text{CIS}((A_1)) = \{A_1\}$$

$$\text{CIS}(S_k) \circ (A_{k+1}) = \text{CIS}(S_k) \cup \{A_{k+1}\} \cup \{B \mid C \in \text{CIS}(S_k) : B = C \cap A_{k+1}\}$$

where $S_k$ is the $k$ element prefix of $S$, i.e., $(A_1, \ldots, A_k)$.

CloStream [6] is an on-line exact counting algorithm for mining closed itemsets in a stream, which uses the above recursive relation in a straightforward way, and thus cannot avoid a combinatorial explosion problem caused by $\text{CIS}(S)$.

3. LC-CLOSTREAM
The LC-CloStream algorithm maintains an internal frequency table $TS$. Formally, $TS$ is a set of tuples $(B, f(B), \delta(B))$, where $B$ is an itemset, $f(B)$ is the number of occurrences of $B$ after the time $t_B$ when $B$ was last stored in $TS$, and $\delta(B)$ is the maximal error count at time $t_B$. We write the frequency table $TS$ at time $t$ as $TS(t)$, and similarly for $f(B, t)$ and $\delta(B, t)$. Let $SP(B, t)$ denote the set of supersets of $B$ belonging to the frequency table $TS(t)$, that is, $SP(B, t) = \{C \in TS(t) \mid B \subseteq C\}$. We define $\text{maxSP}(B, t)$ as follows:

$$\text{maxSP}(B, t) = \arg\max_{C \in SP(B, t)} (f(C, t) + \delta(C, t))$$

The former part of LC-CloStream algorithm, i.e., in lines 5 to 18, performs the incremental intersection and the latter...
Algorithm 1 LC-CloStream algorithm

Input: a stream $S = \langle A_1, A_2, \ldots, A_N \rangle$
  a relative minimal frequency threshold $\sigma (0 < \sigma < 1)$,
  a maximal permissible error ratio $e (0 < e < \sigma)$.
Output: a family $\mathcal{FCS}$ of frequent closed item sets in $S$

1: $t \leftarrow 1$
2: Initialize the frequency table $T_S$.
3: while $t \leq N$ do
4:   Read $A_t$.
5:   for each $B \in T_S$ do
6:      $C \leftarrow B \cap A_t$
7:      if $C \neq \emptyset$ then
8:         $D \leftarrow \max \text{SP}(C)$
9:         if $C \neq T_S$ then
10:            register $C$ as a new entry
11:       else
12:          $T_S \leftarrow T_S \cup \{ (C, f(D) + 1, \delta(D)) \}$
13:       end if
14:   end if
15:   end for
16:   if $A_t \in T_S$ then
17:      register $A_t$ as a new entry
18:   end if
19:   $T_S \leftarrow T_S \cup \{ (A_t, 1, e \cdot (t-1)) \}$
20: end while
21: return $\mathcal{FCS}(N) = \{ B \in T_S \mid f(B) + \delta(B) \geq \sigma \cdot N \}$

Theorem 1 (Semi-completeness for closed itemsets). Let $S$ be a stream of length $N$ and $B$ be a frequent closed itemset in $S$. If $B$ is NOT $e$-extendable, then $B \in \mathcal{FCS}(N)$.

Definition 2. Let $S$ be a stream of length $N$, $\sigma$ be a minimal frequency threshold and $\mathcal{FCS}(N)$ be a output produced from $S$ by LC-CloStream algorithm. Then we define $\mathcal{RS}(S)$ as follows:

$$\mathcal{RS}(S) = \mathcal{FCS}(N) \cup \{ C \mid \exists B \in \mathcal{FCS}(N) : C \subset B, C \neq \emptyset \}$$

Theorem 2 (Completeness for itemsets). Let $S$ be a stream of length $N$ and $B$ be a frequent itemset in $S$. Then $B \in \mathcal{RS}(S)$.

Definition 3. Let $S$ be a stream of length $N$ and $e$ be a maximal error ratio. For any itemset $B$ at time $t$ ($1 \leq t \leq N$), we define $F(B, t)$ and $\Delta(B, t)$ as follows:

1. if $SP(B, t) = \emptyset$, then $F(B, t) = 0$, $\Delta(B, t) = e \cdot t$
2. if $SP(B, t) \neq \emptyset$, then
   $$F(B, t) = f(\max \text{SP}(B, t), t), \quad \Delta(B, t) = \delta(\max \text{SP}(B, t), t)$$

We call $F(B, t) + \Delta(B, t)$ the estimated frequency of $B$ at time $t$.

Notice the estimated frequency $F(B, t) + \Delta(B, t)$ is defined based on $TS(t)$ of time $t$, while the counting frequency $f(B, t) + \delta(B, t)$ depends just on $TS(t-1)$ of the previous time $t-1$.

Theorem 3 ($\epsilon$-Approximation of Frequency). Let $S$ be a stream of length $N$ and $\epsilon$ be a maximal error ratio. For any itemset $B$, we have

$$F(B, N) \leq sup(B, N) \leq F(B, N) + \epsilon \cdot N$$

4. CONCLUSIONS

LC-CloStream can avoid a part of combinational explosion problems in a bursty transactional data stream [5]. In the future, we will study an efficient implementation using a sophisticated data structure, and also have a plan to investigate a more advanced framework where the frequency table has a fixed constant size [5].

5. ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 25730133 and 25330256, and also supported by JST PRESTO (Sakigake).

6. REFERENCES


705