SDSC3001 Assignment 3

2024.11.28

Q1

1. (**25 points**) Design a sampling algorithm to maintain k uniform samples from a stream of elements $x_1, x_2, x_3, \ldots$. Prove the correctness of your algorithm, that is, your algorithm can guarantee that at any time point $t \geq k$ (the time we have received $x_1, x_2, \ldots, x_t$) the probability that x_i ($i \leq t$) is kept as a sample is $\frac{k}{t}$.

Solution:

Assume that after $t-1$ samples, we obtain $S_{old} = \{s_1, s_2, ..., s_k\}$. And $Pr(x_i \in S_{old}) = \frac{k}{t-1}$ for all $i \leq t-1$. When x_t comes, with probability $\frac{k}{t}$, we add X_t to S_{old}. And we delete an s_j in S_{old} with probability $\frac{1}{k}$. Then we get $S_{new} = \{s_1, s_2, ..., s_{j-1}, s_{j+1}, ..., s_k\} \cup \{x_t\}$. We have:

$$Pr(x_t \in S_{new}) = \frac{k}{t}$$

For $i \leq t-1$, we have:

$$Pr(x_i \in S_{new}) = Pr(x_i \in S_{old})\underbrace{(1 - \frac{k}{t})}_{\text{Does not add } x_t} + Pr(x_i \in S_{old}) \cdot \underbrace{\frac{k}{t}}_{\text{add } x_t} \cdot \underbrace{(1 - \frac{1}{k})}_{\text{Does not delete } x_i}$$

$$= \frac{k}{t}$$

Q2

2. **(75 points)** Download the file "trans.txt" and implement a streaming algorithm for mining the top-k most frequent patterns. In the data file "trans.txt", every line is a transaction represented by a set of item ids and the largest transaction contains 15 items.

a) **(15 points)** Prove that to mine top-k most frequent patterns, we do not need to consider patterns of size greater than $m = \lceil \log_2(k+1) \rceil$.

b) **(60 points)** Apply the idea of the Misra–Gries Algorithm to mine approximate frequent patterns by scanning each transaction only once. Specifically, implement your algorithm as follows.

(1). Maintain at most C counters. Each counter is a (key, value) pair where "key" represents a specific pattern and "value" indicates the corresponding (approximate) support of the pattern.

(2). When reading a transaction, enumerate all its subsets of size at most m. Suppose for the i-th transaction we have L_i such valid subsets and clearly, $L_i = \sum_{j=1}^{\min(l_i, m)} \binom{l_i}{j}$ where l_i is the size of the i-th transaction. Transform the i-th transaction to a stream of L_i subsets (the order could be arbitaray) and use the Misra–Gries Algorithm to count each subset's number of appearances (support).

Q2

a) **(15 points)** Prove that to mine top-k most frequent patterns, we do not need to consider patterns of size greater than $m = \lceil \log_2(k+1) \rceil$.

Solution:

If $S = \{x_1, ..., x_m\}$ is a frequent pattern. Then for all $S' \subseteq S$ is a frequent pattern. The number of subsets of S is:

$$2^m - 1 = 2^{\lceil \log_2(k+1) \rceil} - 1 \geq 2^{\log_2(k+1)} - 1 = k + 1 - 1 = k$$

Note: If S' is a subset of S, then support(S') $\geq$ support(S).

Q2

Misra–Gries Algorithm

- Maintain k counters that are initialized as 0
- All counters of value 0 are considered as "available"
- Upon receiving a_t, check if there is a counter for a_t
  - If there is one, increase the counter by 1
  - If no and there is at least one counter available, use an available counter to count a_t
  - If no and no available counters, decrease each counter by 1 (decrement)

A Running Example

- $m = 8,\ k = 4$

Data Stream	1	2	3	2	6	7	8	2	2	1	3	3	1	1	3
Key$_1$	1	1	1	1	1	1	8	8	8	8	8	8	8	8	8
Value$_1$	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
Key$_2$		2	2	2	2	2	2	2	2	2	2	2	2	2	2
Value$_2$		1	1	2	2	1	1	2	3	3	3	3	3	3	3
Key$_3$			3	3	3	3				1	1	1	1	1	1
Value$_3$			1	1	1	0				1	1	1	2	3	3
Key$_4$					6	6					3	3	3	3	3
Value$_4$					1	0					1	2	2	2	3

Q2

b.1) **(8 points)** Suppose in total we have M transactions. Let $L = \sum_{i=1}^{M} L_i$. Suppose f_S is the real support of a pattern S and $\hat{f}_S$ is the approximate support maintained by your Misra–Gries Algorithm. Prove that for any pattern S, we have that $f_S \geq \hat{f}_S \geq f_S - \frac{L}{C+1}$.

Solution:

$\hat{f}_S \leq f_S$ is trivial. We can prove that at time t, we have at most $\frac{t}{C+1}$ decrements. Let V = sum of the counters. When we decrease: $V = V - C$ and when we increase: $V = V + 1$. Denote a as the number of increments, and b as the number of decrements. Then:

$$a + b = t$$

$$a - Cb \geq 0 \quad (V \geq 0 \quad always \quad holds)$$

We have $t - b - Cb \geq 0 \Rightarrow b \leq \frac{t}{C+1}$.

- ▶ Any element appearing more than $\frac{t}{k+1}$ has a counter at time t (**Heavy-Hitters**)

  Lec 7, Page 13. C is equivalent to k in Question 2 here.

- Case1: At any time t, if a pattern(key) S does **not** exist in the counter (so the count $\hat{f}_S$ is 0). Then the real support f_S is less than $\frac{t}{C+1}$. So $\hat{f}_S \geq f_S - \frac{t}{C+1}$.
- Case2: At any time t, if a pattern(key) S exists in the counter, the total number of decrements applied to the count $\hat{f}_S$ is less than $\frac{t}{C+1}$. So we also have $\hat{f}_S \geq f_S - \frac{t}{C+1}$.
- We have L streams of data (patterns), so let t=L, which leads to the proven inequality.

Q2

b.2) **(7 points)** Suppose S^k is the real k-th most frequent pattern. Let $\hat{f}^k$ be the k-th largest (approximate) support obtained by your Misra–Gries Algorithm. Prove that

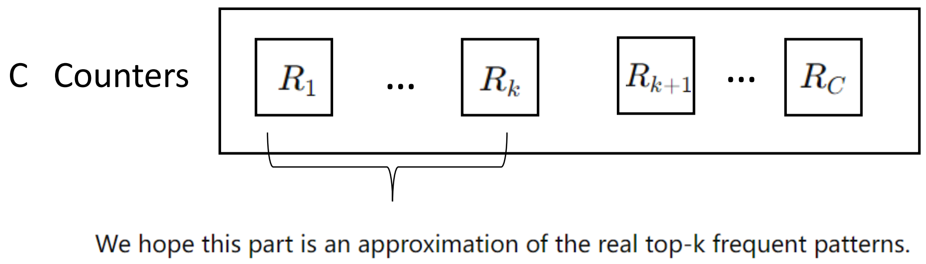
$$f_{S^k} \geq \hat{f}^k \geq f_{S^k} - \frac{L}{C+1} \ .$$

Solution:

$S_1, S_2, \ldots, S_k$ are the real k most frequent patterns. $S_{k+1}, S_{k+2}, \ldots$ are other patterns.

$R_1, R_2, \ldots, R_k$ are the k most frequent patterns approximated using the counter. $R_{k+1}, R_{k+2}, \ldots, R_C$ are other patterns in the counter.

The approximated support of R_i is $\hat{f}^i$.



Q2

b.2) **(7 points)** Suppose S^k is the real k-th most frequent pattern. Let $\hat{f}^k$ be the k-th largest (approximate) support obtained by your Misra–Gries Algorithm. Prove that

$f_{S^k} \geq \hat{f}^k \geq f_{S^k} - \frac{L}{C+1}$.

Solution:

Denote the key of $\hat{f}^i$ as $R_i, (i = 1, 2, ..., k)$.

$$\hat{f}^k \leq \hat{f}^i \leq f_{R_i} \Rightarrow \hat{f}^k \leq f_{S^k}$$

Let y_k be the k^{th} largest value among $\{x_1, x_2, \ldots, x_n\}$. For a real number t, if there exist k different $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$, such that, for all $j \in [k]$,

$$x_{i_j} \geq t$$

Then we have $y_k \geq t$.

k^th largest value

Q2

b.2) **(7 points)** Suppose S^k is the real k-th most frequent pattern. Let $\hat{f}^k$ be the k-th largest (approximate) support obtained by your Misra–Gries Algorithm. Prove that

$f_{S^k} \geq \hat{f}^k \geq f_{S^k} - \frac{L}{C+1}$.

Solution:

1) When $R_k = S^j, j \leq k$, we have:

$$\hat{f}^k \geq f_{R_k} - \frac{L}{C+1} = f_{S^j} - \frac{L}{C+1} \geq f_{S^k} - \frac{L}{C+1}$$

2) When $R_k = S^m, m \geq k+1$, we have:

$$|\{S^1, S^2, ..., S^k\} - \{R_1, R_2, ..., R_k\}| > 0$$
$$\Leftrightarrow \exists S^j, j \leq k, S^j \notin \{R_1, R_2, ..., R_k\}$$

2.a) When $S^j = R_\tau, k+1 \leq \tau \leq C$, we have:

$$\hat{f}^k \geq \hat{f}^\tau \geq f_{R_\tau} - \frac{L}{C+1} = f_{S^j} - \frac{L}{C+1} \geq f_{S^k} - \frac{L}{C+1}$$

2.b) When $S^j \neq R_\tau$, for $\tau = 1, 2, ..., C$, then each S^j is associated with a decrement. We have:

$$f_{S^j} \leq \frac{L}{C+1}$$
$$\Rightarrow \hat{f}^k \geq 0 \geq f_{S^j} - \frac{L}{C+1} \geq f_{S^k} - \frac{L}{C+1}$$

Note: In 2.a), S^j exists in the counter but is not included in the set of the top k approximate supports.
In 2.b), S^j does not exist in the counter.

we have that $f_S \geq \hat{f}_S \geq f_S - \frac{L}{C+1}$.

Q2

b.3) **(15 points)** Since we only have the approximate supports of patterns obtained by our Misra–Gries Algorithm, we can only use such approximate supports to return approximate top-k patterns. We hope to collect all the true top-k patterns by returning a collection of patterns $A = \{S \mid \hat{f}_S \geq t\}$ where t is a threshold for us to filter out non-frequent patterns. Prove that if we set $t = \hat{f}^k - \frac{L}{C+1}$, we can guarantee that

(1) The returned pattern collection A has 100% recall. This means that if for a pattern S, $f_S \geq f_{S^k}$, then $S \in A$; (6 points)

(2) The minimum support of patterns in A, denoted by $minSup(A) = \min_{S \in A} f_S$, is at least $f_{S^k} - \frac{2L}{C+1}$. That is, $minSup(A) \geq f_{S^k} - \frac{2L}{C+1}$. (9 points)

Q2

Solution:

(1) The returned pattern collection A has 100% recall. This means that if for a pattern S, $f_S \geq f_{S^k}$, then $S \in A$; (6 points)

Denote the key of $\hat{f}^i$ as R_i. If $S = R_i$, $i \in \{1, 2, ..., C\}$ and $f_S \geq f_{S^k}$, then:

$$\begin{aligned}\hat{f}^i \geq f_{R_i} - \frac{L}{C+1} &= f_S - \frac{L}{C+1} \\ &\geq f_{S^k} - \frac{L}{C+1} \\ &\geq \hat{f}^k - \frac{L}{C+1} = t\end{aligned}$$

So, $S \in A$.

we have that $f_S \geq \hat{f}_S \geq f_S - \frac{L}{C+1}$.

Note: We should prove that the approximate supports of the real top-k frequent sets are greater than the threshold, so that they can be recalled.

Q2

Solution:

(2) The minimum support of patterns in A, denoted by $minSup(A) = \min_{S \in A} f_S$, is at least $f_{S^k} - \frac{2L}{C+1}$. That is, $minSup(A) \geq f_{S^k} - \frac{2L}{C+1}$. (9 points)

If $S \in A$, then:

$$\begin{aligned} f_S \geq \hat{f}_S \geq t = \hat{f}^k - \frac{L}{C+1} \\ \geq f_{S^k} - \frac{L}{C+1} - \frac{L}{C+1} \\ = f_{S^k} - \frac{2L}{C+1} \end{aligned}$$

we have that $f_S \geq \hat{f}_S \geq f_S - \frac{L}{C+1}$.

Q2

b.4) **(30 points)** Set $k = 500$. Run your Misra–Gries Algorithm on the "trans.txt" dataset and report the values of L and $minSup(A)$ when setting $C = 500000, 750000, 1000000$. To compute $minSup(A)$, you can refer to the file "patterns_Apriori.txt" containing all the frequent patterns of support at least 21. Each line of "patterns_Apriori.txt" is in the form $id_1, id_2, \ldots, id_l : sup$, where $id_1, id_2, \ldots, id_l$ denotes a pattern $\{id_1, id_2, \ldots, id_l\}$ and sup is the support of this pattern. (**Hint**: the file "patterns_Apriori.txt" contains enough information. If your algorithm returns some pattern that is not in the "patterns_Apriori.txt" file, probably your algorithm is not implemented correctly.)

C =500000 minSup(A)=1037
C =750000 minSup(A)=1077
C =1000000 minSup(A)=1098