CMPUT 675: Algorithms for Streaming and Big Data **Fall 2019**

Lecture 16 (Oct 30, 2019): Selection

Lecturer: Mohammad R. Salavatipour *Scribe: Aditya Jayaprakash*

The topic in todays lecture will be about the process of selection. Suppose we are given a stream $a_1, \ldots, a_n$ where elements a_i are from a domain D with a total order. Our goal is to find the $\approx \frac{n}{2}$ element in the stream i.e., finding the median. In the offline setting, there is an $O(n)$ deterministic algorithm to find the median. For simplicity, we will assume that all the elements are distinct.

Definition 1 *Given a stream or set S over a domain D, we can define the rank of an element $x \in D$ with respect to s to be*

$$rank(x, S) = |\{y \in S : y \leq x\}|$$

$rank(x, S)$ is the number of elements in S that are less than or equal to x. An element with rank $\frac{|S|}{2}$ would be the median element.

Munro and Patterson were the first to initiate the study of what we today call streaming algorithms in their paper [MP80] where they studied selection and sorting problems in a stream fashion. In their model, the only operation that is allowed on the underlying elements is a pairwise comparison. They also considered multi-pass algorithms where we could go through the stream multiple times. They proved the following lower bound,

Theorem 1 (MP80) *Any p-pass comparison-based algorithm to solve the selection problem on a stream of n elements requires $\Omega(n^{1/p})$ space.*

They also presented an algorithm would require $\tilde{O}(n^{1/p})$ space for a multipass algorithm for p-passes.

Our goal is to show given space s and a stream of n elements, one one pass, we can reduce the problem to an instance for selection over $O\left(\frac{n \log^2 n}{s}\right)$ items. Suppose we choose $s = n^{1/p} \log^{1-2/p} n$, then after i passes, it reduces the instance to one having $n^{1-i/p} \log^{2i/p} n$ items. Hence, after p passes, we get $O(s)$ items.

16.1 i-sample

Suppose we wish to find the element with rank k in a stream. During each pass through the stream, we wish to filter the elements while still preserving some information. At the start of the i^{th} pass, we have filters a_i, b_i such that

$$a_i \leq \text{rank-k element } \leq b_i$$

Initially, we set $a_1 = -\infty$ and $b_1 = \infty$. After the each pass, we wish to shrink the gap between a_i and b_i and eventually arrive at finding the element with rank k. We will denote m_i to be the number of elements between a_i and b_i.

Definition 2 *An i-sample of a set of size $2^i \cdot s$ is a sorted sequence of length s defined recursively as follows where $A \circ B$ is stream A followed by stream B:*

- *For $i = 0$, the 0-sample(A) = sort(A)*

- *For $i > 0$, the i-sample($A \circ B$) = sort(evens(($i-1$)-sample(A) $\bigcup$ evens(($i-1$)-sample(B)))*

For $i = 0$ the entire set S is sorted. For $i+1$, take $2^{i+1} \cdot s$ elements, divide it into two different halves and take an i-sample of each half, thin each one by taking even index items and merge the two thinned samples into a sorted one of size s.

16.2 Selection

In this selection, we will introduce the more general version of selection. We are given a parameter $0 < \phi \leq 1$ and we would like to return an element of rank ϕn.

The problem of ϵ-approximate quantile involves finding an element whose rank is $(\phi \pm \epsilon)n$. In this case, we want to find an element that is approximately close to the rank.

Suppose we select elements of rank $\frac{in}{k}$ where $k = \frac{1}{\epsilon}$. We can think of a quantile summary Q as a set of elements $\{q_1, \ldots, q_l\}$ along with an interval $[\mathbf{rmin}_Q(q_i), \mathbf{rmax}_Q(q_i)]$ for each value of q_i where $\mathbf{rmin}_Q(q_i)$ is a lower bound for the rank of q_i in S and $\mathbf{rmax}_Q(q_i)$ is an upper bound for the rank of q_i in S.

Suppose $q_1 \leq \ldots \leq q_l$ where q_1 is the minimum element in S and q_l is the maximum element in S and $\max_i (\mathbf{rmax}_Q(q_i) - \mathbf{rmin}_Q(q_i)) \leq 2\epsilon|S|$, Q can be used to give an ϵ-quantile summary. We will state this as a lemma,

Lemma 1 *Suppose Q is a quantile summary for S such that $\max_{1 \leq i \leq l} (\boldsymbol{rmax}_Q(q_i) - \boldsymbol{rmin}_Q(q_i)) \leq 2\epsilon|S|$, then Q is an ϵ-approximate quantile summary.*

Proof. This proof is from [GK16]. Let $r = \lceil \phi|S| \rceil$. We will identify an index i such that $r - \epsilon|S| \leq \mathbf{rmin}_Q(q_i)$ and $\mathbf{rmax}_Q(q_i) \leq r + \epsilon|S|$. Clearly, such a value q_i approximates the ϕ-quantile to within the claimed error bounds. We will now argue that such an index i must always exist.

Let $e = \max_i (\mathbf{rmax}_Q(q_{i+1}) - \mathbf{rmax}_Q(q_i)/2$. Consider the case $r \geq |S| - e$. We have $\mathbf{rmin}_Q(q_l) \geq (1-\epsilon)|S|$, and therefore $i = l$ has the desired propertt. We now focus on the case $r < |S| - e$, and start by choosing the smallest index j such that $\mathbf{rmax}_Q(q_j) > r + e$. If $j = 1$, then j is the desired index since $r + e < \mathbf{rmax}_Q(q_1) \leq \epsilon|S|$. Otherwise, $j \geq 2$, and it follows that $r - e \leq \mathbf{rmin}_Q(q_{j-1})$. If $r - e > \mathbf{rmin}_Q(q_{j-1})$, then $\mathbf{rmax}_Q(q_j) - \mathbf{rmin}_Q(q_j - 1) > 2e$ which is a contradiction since $e = \max_i (\mathbf{rmax}_Q(q_{i+1}) - \mathbf{rmax}_Q(q_i)/2$. By our choice of j, we have $\mathbf{rmax}_Q(q_{j-1}) \leq r + e$. Thus $i = j - 1$ is an index i with the above described property. ■

We know that we can take the union of two quantile summaries is also a quantile summary. Suppose we are given a quantile $Q' = \{x_1, \ldots, x_a\}$ and $Q'' = \{y_1, \ldots, y_b\}$ are two quantile summaries for sets S' and S''. We want to combine Q' and Q'' to be one quantile summary for $S' \cup S''$. We can view $S = S' \cup S''$ as a multiset. We would like to keep the approximation of the resulting summary similar to those of Q' and Q''.

Suppose we combine Q' and Q'' i.e., let $Q = Q' \cup Q'' = \{z_1, \ldots, z_{a+b}\}$. We will sort the union of summaries and define new estimates. Choose some $z_i \in Q$ and suppose $z_i = x_r$ for $1 \leq r \leq a$ and let y_s be the

largest element in Q'' not larger than x_r.

$$\mathbf{rmin}_Q(z_i) = \begin{cases} \mathbf{rmin}_{Q'}(x_r), & \text{if no such } y_s \\ \mathbf{rmin}_{Q'}(x_r) + \mathbf{rmin}_{Q''}(y_s), & \text{otherwise} \end{cases}$$

$$\mathbf{rmax}_Q(z_i) = \begin{cases} \mathbf{rmax}_{Q'}(x_r) + \mathbf{rmax}_{Q''}(y_s), & \text{if } y_t \text{ is undefined} \\ \mathbf{rmax}_{Q'}(x_r) + \mathbf{rmax}_{Q''}(y_t) - 1, & \text{otherwise} \end{cases}$$

We can state this as a lemma,

Lemma 2 *Let Q' be an ϵ'-approximate quantile summary for S' and Q'' be an ϵ''-approximate quantile summary for S'', then $Q = Q' \cup Q''$ i.e., combining Q' and Q'' produces an $\bar{\epsilon}$-approximate quantile summary for $S = S' \cup S''$ where $\bar{\epsilon} = \frac{\epsilon' n' + \epsilon'' n''}{n' + n''} \leq \max\{\epsilon', \epsilon''\}$ where $n' = |S'|$ and $n'' = |S''|$.*

Proof. This proof is from [GK16].Let n' and n'' denote the number of observations covered by Q' and Q''. Consider any two consecutive elements z_i, z_{i+1} in Q. From the last lemma we proved, it is sufficient to show that $\mathbf{rmax}_Q(z_{i+1}) - \mathbf{rmin}_Q(z_i) \leq 2\bar{\epsilon}(n' + n'')$. We will analyze two cases. First, z_i, z_{i+1} are both from a single summary, say elements x_r, x_{r+1} in Q'. Let y_s be the largest element in Q'' that is smaller than x_r and let y_t be the smallest element in Q'' that is larger than x_{r+1}. Observe that if y_s and y_t are both defined, then they must have consecutive elements in Q''.

$$\begin{aligned} \mathbf{rmax}_Q(z_{i+1}) - \mathbf{rmin}_Q(z_i) &\leq (\mathbf{rmax}_{Q'}(x_{r+1}) + \mathbf{rmax}_{Q''}(y_t) - 1) - (\mathbf{rmin}_{Q'}(x_r) + \mathbf{rmin}_{Q''}(y_s)) \\ &\leq (\mathbf{rmax}_{Q'}(x_{r+1}) + \mathbf{rmin}_{Q'}(x_r)) + (\mathbf{rmax}_{Q''}(y_t) - \mathbf{rmin}_{Q''}(y_s) - 1) \\ &\leq 2(2n'\epsilon' + n''\epsilon'') = 2\bar{\epsilon}(n' + n'') \end{aligned}$$

Otherwise, if only y_s is defined, then it must be the largest element in Q''; or if only y_t is defined, it must be the smallest element in Q''. We can use the same analysis for these cases.

Next, we consider the case when z_i and z_{i+1} come from different summaries, say, z_i corresponds to x_r in Q' and z_{i+1} corresponds to y_t in Q''. Then observe that x_r is the largest element smaller than y_r in Q' and y_t is the smallest element larger than x_r in Q''. Moreover, x_{r+1} is the smallest element in Q' that is larger than y_t, and y_{t-1} is the largest element in Q'' that is smaller than x_r. Using these observations, we get

$$\begin{aligned} \mathbf{rmax}_Q(z_{i+1}) - \mathbf{rmin}_Q(z_i) &\leq (\mathbf{rmax}_{Q''}(y_t) + \mathbf{rmax}_{Q'}(x_{r+1}) - 1) - (\mathbf{rmin}_{Q'}(x_r) + \mathbf{rmin}_{Q''}(y_{t-1})) \\ &\leq (\mathbf{rmax}_{Q''}(y_t) - \mathbf{rmin}Q''(y_{t-1})) - (\mathbf{rmax}_{Q'}(x_{r+1}) + \mathbf{rmin}_{Q'}(x_r) - 1) \\ &\leq 2(2n'\epsilon' + n''\epsilon'') = 2\bar{\epsilon}(n' + n'') \end{aligned}$$

■

We will now discuss the PRUNE operation which takes two parameters, an ϵ-approximate quantile summary Q' and parameter B and produces a new quantile summary Q' of size $B + 1$ with accuracy $\left(\epsilon' + \frac{1}{2B}\right)$. We can generate the output Q by by querying the element of Q' with rank $1, \frac{|S|}{B}, \frac{2|S|}{B}, \ldots, |S|$. For each $q_i \in Q$, we define

$$\mathbf{rmin}_Q(q_i) = \mathbf{rmin}_{Q'}(q_i) \text{ and } \mathbf{rmax}_Q(q_i) = \mathbf{rmax}_{Q'}(q_i)$$

For any consecutive pairs $q_i, q_{i+1} \in Q$, we have that

$$\mathbf{rmax}_Q(q_{i+1}) - \mathbf{rmin}_Q(q_i) \leq \frac{i|S|}{B} + \epsilon'|S| - \left(\frac{(i-1)|S|}{B} - \epsilon'|S|\right) \leq \frac{|S|}{B} + 2\epsilon'|S| = \left(2\epsilon' + \frac{1}{B}\right)|S|$$

16.3 An $O\left(\frac{1}{\epsilon}\log^2(\epsilon n)\right)$ space algorithm

We will use ideas from a paper by Manku, Rajagopalan and Lindsey [MRL98] which was inspired by the Munro-Paterson algorithm. Their algorithm was deterministic and used $O\left(\frac{\log^2(\epsilon n)}{\epsilon}\right)$ space. There are two new operations we will introduce, NEW and COLLAPSE.

Suppose we have l summaries of size k when a buffer summarizes k' observations, then the weight of the buffer is $\left\lceil \frac{k'}{k} \right\rceil$. The NEW operation will build a buffer with k new elements from the input. We will need to reuse the buffer when we encounter new elements in the stream. We will combine the buffer and/or PRUNE it. The COLLAPSE operation makes it into a single suffer of the same size k. From the last section, we know PRUNE produces some error.

Let b be the total number of buffers and let k be the size of the buffers. For simplicity, we will also assume $\frac{n}{k}$ is a power of 2. Consider a full binary tree with $\frac{n}{k}$ leaves, each corresponding to k elements in a stream. If we assign one buffer to k elements, we have a 0-error quantile summary for them.

We could assign a buffer of size k to an internal node to maintain an approximate quantile summary for the elements of the stream in the subtree. That node would in turn use the buffers of it children and PRUNE it to get a buffer. Suppose we have a node v with children u and w. The buffer of v can be produced by combining the sum of u and w and PRUNE it back to size k to obtain v at an additional error of $\frac{1}{2k}$. We build the buffer at each node in a bottom-up fashion. Our output would be the quantile summary at the root of size k. The total error of the algorithm can be evaluated by looking at the error at the root of the tree.

The height of the tree would be $h = O\left(\log \frac{n}{k}\right)$ and we need $O(h)$ buffers. The quantile summaries at the leaves have zero error since we store all the elements in the buffer, but at each level, the error increases by $\frac{1}{2k}$. Hence, after h levels, the error at the root is $O\left(\frac{h}{2k}\right)$. Suppose we want to obtain an ϵ-approximate quantile summary, then we require $\frac{h}{2k} \leq \epsilon$. We would need to choose $k \geq \frac{\log(2\epsilon n)}{2\epsilon}$ in order to obtain an ϵ-approximate quantile summary.

The total space usage is $O(hk)$ and since $h = \log \frac{n}{k}$, we have the total space complexity to be $O\left(\frac{1}{\epsilon}\log^2(\epsilon n)\right)$. [GK16] is a very useful and detailed survey on this topic.

References

MP80 J Ian Munro and Mike S Paterson,. Selection and sorting with limited storage. Theoretical Computer Science, 12(3):315323, 1980.

MRL98 Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G Lindsay. Approximate medians andother quantiles in one pass and with limited memory. ACM SIGMOD Record, volume 27, pages 426435. ACM, 1998.

GK16 Michael Greenwalk and Sanjeev Khanna Quantiles and equidepth histograms over streams `https://www.cis.upenn.edu/~sanjeev/papers/quantiles-chapter.pdf` In Data Stream Management: Processing High-Speed Data Streams, ed. M. Garofalakis, J. Gehrke, and R. Rastogi, Springer, 2016