Data mining: homework 2

Edo liberty

The setup is as follows. We have a universe of $N$ items $A = \{a_1, \ldots, a_N\}$ and $m$ subsets $S_i \subseteq A$, $i \in \{1, \ldots, m\}$. We assume that given a set $S_i$ we can iterate over its elements one by one. The exercise will deal with approximating the size of different unions of these sets.

1. In case you wanted to give an $\epsilon$ approximation, w.p. $1 - \delta$, to the size of $S_1 \cup S_2$. How would your ability to approximate the zero’th frequency moment in streams help you with that? (We assume here that $O(|S_1| + |S_2|)$ running time is acceptable, and that $\epsilon$ and $\delta$ are both constants)

2. Assuming it is only possible to compute the second frequency moment of streams, one can still give an $\epsilon$ approximation, w.p. $1 - \delta$, to the size of $S_1 \cup S_2$. How?

3. Assume now that you are tasked with designing an algorithm. Your algorithm is allowed to preprocess the sets $S_i$ in any amount of time and produce any data structure. It should then be able to take as input a set of indexed $I \subset \{1, \ldots, m\}$ and produce an $\epsilon$ approximation of $|\bigcup_{i \in I} S_i|$ with probability at least $1 - \delta$. The aim is to create an algorithm which runs in time $O(\sum_{i \in I} |S_i|)$, i.e., the solution from question 1 is not the answer. It is assumed that for all $i$, $|S_i| \in o(1)$.

   - describe the preprocessing stage and its resulting data structure. (before $I$ is given)
   - describe the estimation process. (after $I$ is given)
   - prove your algorithm’s correctness.
   - give the space usage of your data structures.
   - give the runtime complexity your estimation process.

Solutions

1. Consider concatenating the two sets $S_1$ and $S_2$ into a stream

   $$A = [S_1(1), S_1(2), \ldots, S_1(|S_1|), S_2(1), \ldots, S_2(|S_2|)]$$

   where the order of the elements in $S_1$ and $S_2$ is arbitrary. It is quite immediate to see that the number of distinct elements in the stream, $f_0(A)$,
is exactly $|S_1 \cup S_2|$. More explicitly, the items in $S_1 \cap S_2$ appear twice in the stream $A$ and all others appear once. Therefore, $f_0(A) = |A| - |S_1 \cap S_2| = |S_1| + |S_2| - |S_1 \cap S_2| = |S_1 \cup S_2|$. Given our ability to approximate $f_0(A)$ frequency moments using $O(1/\varepsilon^2 \delta)$ space and $O(|A|/\varepsilon^2 \delta)$ operations we conclude that a running time of $O(|S_1 \cup S_2|)$ is sufficient. We used that $\varepsilon$ and $\delta$ are constant and that $2|S_1 \cup S_2| \geq |S_1| + |S_2|$.

2. Since each item in $S_1 \cap S_2$ appears twice in the stream $A$ and all others appear once we have the following expression for $f_2(A)$.

$$f_2(A) = |S_1 \setminus S_2| + |S_2 \setminus S_1| + 4|S_2 \cap S_1| = |A| + 2|S_2 \cap S_1|$$

Moreover, $|S_1 \setminus S_2| + |S_2 \setminus S_1| = |S_2 \cup S_1| - |S_2 \cap S_1|$. Also since $f_0(A) = |A| - |S_2 \cap S_1|$ we have that $f_0(A) = (3|A| - f_2(A))/2$. Thus, since we know $|A|$ exactly, if we could approximate $f_2(A)$ we could also approximate $f_0(A)$. Note that to insure the approximation factor is still constant we must have that $\varepsilon f_2(A)/2 \leq \varepsilon f_0(A)$ which is indeed true.

3. • We first choose $s \geq 8/\varepsilon^2 \delta$ hash functions $h_i : a \to [0, 1]$ uniformly. For each set $S_i$ of the $m$ sets we compute for each hash function $h_j$ its minimal value over the elements of $S_i$. Storing these concludes the preprocessing step which requires $O(s \sum_{i=1}^m |S_i|)$ hash evaluations and $O(sm)$ storage. Note that here we assume that the number of elements in the universe $n$ is such that $\log(n)$ is small enough to be treated as a constant. Otherwise, the hash functions must contain $\Omega(\log(n))$ bits which would give an $O(s \log(n) \sum_{i=1}^m |S_i|)$ running time and $O(sm \log(n))$ storage.

• Once $I$ is received, we compute the $s$ minimal values over the sets $S_i$ s.t. $i \in I$ for each hash function. This is done simply by taking the minimal values from the ones already computed in the preprocessing step. Denoting by $x_j$ this minimal value (for hash function $h_j$) we return \[ \frac{1}{s} \sum_{i=1}^s x_i. \]

• The proof is identical to a proof given in the class (and the class notes) so I will only repeat it here. The main statement is that the reciprocal to the mean of $s \geq 8/\varepsilon^2 \delta$ minimal hash value over a set of $n'$ objects is an $\varepsilon$ approximation to $n'$ with probability at least $1 - \delta$. The algorithm clearly computes these minimal values for the set $\cup_{i \in I} S_i$ which completes the proof.

• The amount of space is as stated before $O(sm) = O(8m/\varepsilon^2 \delta)$ or $O(8m \log(n)/\varepsilon^2 \delta)$ depending on the computational model.

• Given that all $sm$ minimal hash values are given in an array with $O(1)$ access time, the amount of time to compute the approximated size of $\cup_{i \in I} S_i$ is $O(s|I|)$. 

2