CMPSCI590D: Algorithms for Data Science

Homework 3

Instructor: Barna Saha — Posted: Nov 10th, Due: Nov 28th

Do not look up materials on the Web. You can consult the reference books mentioned on the course website, and also the class slides for solving the homework problems. You may work in a group of size at most 3. *Every person is allowed to talk to at most 2 others. All communications are bidirected: A talks with B, automatically implies B talks with A. Therefore, those who are working in a group of* 3 *are not allowed to talk with anyone else outside the group. Mention any collaboration clearly in the submission. Submit one homework solution per group. No late homework will be accepted.*

For programming assignments, submit your code with a detailed readme file that contains instruction for running it. Also include any test dataset that you have used and results obtained to show correctness of your implementation.

Total Point: 200, *Extra Point:* 25

1. Suppose we have a universe U of elements. The similarity of two sets $A, B \subseteq U$ is defined by the Jackard Similarity $s(A, B) = \frac{|A \cap B|}{|A \cup B|}$. Fix k random orderings $\sigma_1, \sigma_2, ..., \sigma_k$ of the elements of U, and let the sketch of a set $S \subseteq U$ be the vector of size k whose ith component is the element of S that comes first in the ordering σ_i.

   (i) Given the sketches of two sets A and B, how can we estimate $s(A, B)$? How large must we make k to be confident that our estimate is fairly accurate? In other words, derive a high-probability bound for the error in the estimate in terms of k. That is, given an $\epsilon > 0$ and δ, derive the minimum value of k such that the returned estimate is within $[(1-\epsilon)s(A, B), (1+\epsilon)s(A, B)]$ with probability $\geq (1-\delta)$. We are only interested to return a good estimate when $s(A, B) \geq \epsilon$.

   [40]

   (ii) We have 1 million documents $\mathcal{D}$. For each document $D_i \in \mathcal{D}$, $i = 1, 2, .., 10^6$, we first construct a set of shingles T_i as discussed in the class. Given any query document D_j, we are interested in returning some document $D_i \in \mathcal{D}$ such that $s(T_i, T_q) \geq 0.8$. Give a scheme such that we can preprocess all the documents in $\tilde{O}(|\mathcal{D}|^{\frac{3}{2}})$ time (assuming processing each document requires $O(1)$ time), such that we use on expectation $\tilde{O}(\sqrt{|\mathcal{D}|})$ time for querying. If there is no D_i with $s(T_i, T_q) \geq 0.8$, then it is ok for the algorithm to return none. Otherwise, the algorithm returns a document within similarity at least $c * 0.8$ for some $c < 1$ of the queried document with probability at least $1 - \frac{1}{e^2}$. What is the maximum possible value of c in your scheme?

   [40]

2. Consider the following algorithm for finding frequent item:

   Maintain a list of items being counted. Initially the list is empty. For each item, if it is the same as some item on the list, increment its counter by one. If it differs from all the items on the list, then if there are less than k items on the list, add the item to the list with its counter *set to one. If there are already k items on the list decrement each of the current counters by one. Delete an element from the list if its count becomes zero.*

Show that if the total stream size is m, then any item that has frequency $> \frac{m}{k+1}$ times occur in the list.

[20]

3. This problem is about estimating basic statistics on graphs in small space. We are given a connected graph $G = (V, E)$ which is undirected and unweighted. Let $|V| = N$ and $|E| = M$. For every vertex $v \in V(G)$, let $N(v)$ denote the neighbors of v in G. We also use the notation $deg_G(v)$ to denote the degree of v in G. Hence $deg_G(v) = |N(v)|$. the Consider the following sampling procedure.

   - Set $p = \frac{60}{\sqrt{M}}$.
   - Sample each vertex $v \in V$ independently with probability p.
   - Let V_p be the set of vertices sampled in the previous step.
   - For every $u \in V_p$ store its entire list of neighbors, that is $N(u)$

   Therefore, the storage requirement is $|V_p| + \sum_{u \in V_p} |N(u)|$. Let $G_p = (V, E_p)$ denote the sampled graph, where the edge $(u, v) \in E_p$ if $u \in V_p$ and $v \in N(u)$ or if $v \in V_p$ and $u \in N(v)$.

   (a) What is the expected storage requirement of the above algorithm? [20]

   (b) Give an upper and lower bound on the expected storage requirement that you have computed as a function of M? [5]

   (c) Define $deg_{G_p}(u) = |\{v \in V \mid (u, v) \in E_p\}|$, that is the degree of u in G_p. Let us defineD_p as follows,

   $$D_p = \frac{1}{|V|} \sum_{u \in V_p} deg_{G_p}(u).$$

   Define

   $$\hat{D} = \frac{1}{p} D_p.$$

   i. Compute $E[\hat{D}]$. [20]

   ii. Compute $Var[\hat{D}]$ [30]

   iii. Suppose D denotes the actual average degree of $G = (V, E)$, that is $D = \frac{1}{|V|} \sum_{v \in V} deg_G(v)$, and assume $\max_{v \in G} deg_G(v) \leq \sqrt{|E|}$. What can you say about the following?

   $$Prob(|D - \hat{D}| \geq \frac{D}{2})$$

   Can you apply Chernoff Bound here? [30]

   (d) Now suppose, we are interested in counting the number of **distinct** pairs of vertices u, v such that there is some vertex w, with $(u, w) \in E(G)$ and $(w, v) \in E(G)$, that is the distance between u and v is at most two. Note that if there are multiple such w's, that is there are multiple ways to reach from u to v via paths of length 2, then we only want to count the pair u, v once. Let T denote the number of such distinct pairs of vertices $\{u, v\}$ in G that are reachable via paths of length 2.

   i. Obtain an estimate $\hat{T}$ of number of distinct pairs of vertices that are reachable via paths of length 2 in G from G_p such that $E[\hat{T}] = T$. Justify your answer. [20]