Homework 3
Due 2015-Nov-2 by 11:59pm

Prof. Amit Chakrabarti
Computer Science Department
Dartmouth College

General Instructions: Please write concisely, but rigorously, and show your calculations explicitly, as we do in class. Each problem is worth 7 points. You can find a detailed description of my grading principles under the 7-point scale on the course website.

Please submit your homework electronically (by email). For hand-written solutions, please scan and email. The big scanner/copier in the Sudikoff front office can be useful for this.

Honor Principle: You are allowed to discuss the problems and exchange solution ideas with your classmates. But when you write up any solutions for submission, you must work alone. You may refer to any textbook you like, including online ones. However, you may not refer to published or online solutions to the specific problems on the homework. *If in doubt, ask the professor for clarification!*

Coresets and Clustering

9. Consider the minimum enclosing ball (MEB) problem, which we discussed in class. Fix a collection of t nonzero vectors $u_1, \ldots, u_t \in \mathbb{R}^d$ with the property that

$$\forall\, z \in \mathbb{R}^d \setminus \{0\}\ \exists\, i \in [t]:\ \operatorname{ang}(z, u_i) \le \theta\,,$$
$$\text{where } \operatorname{ang}(x,y) := \arccos \frac{\langle x, y\rangle}{\|x\|_2 \|y\|_2}\,.$$

   For a finite set of points $P \subseteq \mathbb{R}^d$, we constructed a coreset Q (for the MEB problem) as follows.

$$Q \;:=\; \bigcup_{i=1}^{t} \left\{ \arg\max_{x \in P} \langle x, u_i\rangle, \arg\min_{x \in P} \langle x, u_i\rangle \right\}.$$

   Prove, formally (without appealing to intuitive reasoning by picture), that this construction gives us a $(1+\delta)$-coreset for P, for the setting $\theta = \Theta(\sqrt{\delta})$. Also prove that this coreset has the disjoint union property: i.e., if Q_1, Q_2 are α-coresets for disjoint sets P_1, P_2 respectively, then $Q_1 \cup Q_2$ is an α-coreset for $P_1 \cup P_2$.

10. *(Ignore this one until the Oct 27 class.)* A summarization cost function Δ is said to be *metric* if it satisfies the following condition, for all streams σ, π and summaries $S \subseteq \sigma, T \subseteq \sigma[S] \circ \pi$:

$$\Delta(\sigma[S] \circ \pi, T) - \Delta(\sigma, S) \;\le\; \Delta(\sigma \circ \pi, T) \;\le\; \Delta(\sigma[S] \circ \pi, T) + \Delta(\sigma, S)\,. \tag{1}$$

   Here, $\sigma[S]$ is the stream obtained by replacing each token of σ with its best representative from S.

   Suppose that our streams consist of points in some metric space (M, d), and our cost function is the k-center cost function, i.e.,

$$\Delta(\sigma, S) \;=\; \begin{cases} \infty\,, & \text{if } S \not\subseteq \sigma \text{ or } |S| > k \\ \max_{x \in \sigma} \min_{y \in S} d(x,y)\,, & \text{otherwise.} \end{cases}$$

   Give a rigorous proof that this particular function Δ is metric. (Write out the steps of reasoning explicitly and point out exactly which steps use the properties that define a metric space.) Note that in the case of k-center, we might as well assume $\sigma[S] = S$. This corresponds to the version of Eq. (1) given in class.

Space requirements for triangle counting

These problems involve graph streams. Recall that such a stream specifies an input graph G, with vertex set $V(G) = [n]$ and edge set $E(G)$ of size m. Each token is a pair $\{u, v\} \in E(G)$, and the tokens are all distinct; we are assuming that each edge is seen exactly once in the stream. We are interested in estimating T_3, where

$$T_i \;=\; \left| \left\{ \{u,v,w\} \in \binom{V}{3} :\ |E(G) \cap \{\{u,v\},\{v,w\},\{u,w\}\}| = i \right\} \right|.$$

In both these problems, we are promised that $T_3 \ge t$, for some given value $t > 0$.

11. The sampling-based algorithm for estimating T_3 is based on the following basic estimator: pick an edge $\{u, v\}$ uniformly at random from the stream; pick a vertex w uniformly at random from $V(G) \setminus \{u, v\}$; output $m(n-2)$ if edges $\{u, w\}$ and $\{v, w\}$ occur after $\{u, v\}$ in the stream, and 0 otherwise.

    Prove that the output of this algorithm has expectation exactly T_3. By running some number, p, of independent copies of this algorithm in parallel and averaging the outputs, we would like to obtain an $(\varepsilon, \frac{1}{3})$-approximation to T_3. By using appropriate probabilistic analysis (as in the AMS repeat-count algorithm), show that $p = O(\varepsilon^{-2}mn/t)$ copies suffice.

12. The sketch-based triangle counting algorithm uses the following idea. We process a virtual stream of triples of vertices derived from the given stream of edges, where an actual token $\{u, v\}$ gives rise to $n-2$ virtual tokens

$$\{u, v, w_1\}, \{u, v, w_2\}, \ldots, \{u, v, w_{n-2}\}, \qquad \text{where } \{w_1, w_2, \ldots, w_{n-2}\} = [n] \setminus \{u, v\}.$$

    We then compute F_0, F_1 and F_2 for this stream. Prove that $F_k = T_1 + 2^k T_2 + 3^k T_3$.

    To count triangles in the graph, we can write three such equations, for $k \in \{0, 1, 2\}$, and then solve for T_3. Work out an exact formula for T_3 in terms of n, m, F_0 and F_2. Based on your formula, work out exactly what guarantees you need on your estimates of F_0 and F_2 so that the formula gives you a $(1 \pm \varepsilon)$ approximation to T_3. Based on these required guarantees, work out an upper bound on the total space needed by the algorithm to give an $(\varepsilon, \frac{1}{3})$-approximation to T_3. The space may depend on t, as in the previous problem.

Distance estimation, generalized

13. Recall that the distance estimation problem asks us to process a streamed graph G so that, given $x, y \in V(G)$, we can return an t-approximation of $d_G(x, y)$, i.e., an estimate $\hat{d}(x, y)$ with the property

$$d_G(x, y) \;\le\; \hat{d}(x, y) \;\le\; t \cdot d_G(x, y).$$

    Here t is a fixed integer known beforehand. In class, we solved this using space $\widetilde{O}(n^{1+2/t})$, by computing a subgraph H of G that happened to be a t-spanner. Now suppose that the input graph is edge-weighted, with weights being integers in $[W]$. Each token in the input stream is of the form (u, v, w_{uv}), specifying an edge (u, v), and its weight $w_{uv} \in [W]$. Distances in G are defined using weighted shortest paths, i.e.,

$$d_{G,w}(x, y) \;:=\; \min\left\{\sum_{e \in \pi} w_e : \pi \text{ is a path from } x \text{ to } y\right\}$$

    Give an algorithm that processes G using space $\widetilde{O}(n^{1+2/t} \log W)$ so that, given $x, y \in V(G)$, we can then return a $(2t)$-approximation of $d_{G,w}(x, y)$. Give careful proofs of the quality and space guarantees of your algorithm.

    Hint: Partition the edges into $\lceil \log W \rceil$ disjoint *classes,* where class i consists of all edges e with $2^{i-1} \le w_e < 2^i$, and compute multiple t-spanners.