CS550 **Problem Set 4**
Spring 2022 **Due:** Apr 29, 2022, 11:59 p.m.

This homework has a total of 30 points, it will be rescaled to 10 points eventually.

Submission instructions: These questions require thought but do not require long answers. Please be as concise as possible. You should submit your answers as a writeup in PDF format, for those questions that require coding, write your code for a question in a single source code file, and name the file as the question number (e.g., question_1.java or question_1.py). Finally, put your PDF answer file and all the code files in a folder named as your Name and NetID (i.e., Firstname-Lastname-NetID.pdf), compress the folder as a zip file (e.g., Firstname-Lastname-NetID.zip), and submit the zip file via Canvas. For the answer writeup PDF file, we have provided both a word template and a latex template for you, after you finished the writing, save the file as a PDF file, and submit both the original file (word or latex) and the PDF file.

Late Policy: The homework is due on 4/29 (Friday) at 11:59pm. We will release the solutions of the homework on Canvas on 5/2 (Monday) 11:59pm. If your homework is submitted to Canvas before 4/29 11:59pm, there will no late penalty. If you submit to Canvas after 4/29 11:59pm and before 5/2 11:59pm, your score will be penalized by 0.9^k, where k is the number of days of late submission. For example, if you submitted on 5/2, and your original score is 80, then your final score will be $80 \times 0.9^3 = 58.32$ for 3 days of late submission. If you submit to Canvas after 5/2 11:59pm, then you will earn no score for the homework.

Clustering Data Streams [30pt]

Introduction. In this problem, we study an approach for clustering massive data streams. We will study a framework for turning an approximate clustering algorithm into one that can work on data streams, i.e., one which needs a small amount of memory and a small number of (actually, just one) passes over the data. As the instance of the clustering problem, we will focus on the k-means problem.

Definitions: Before going into further details, we need some definitions:

- The function $d : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^+$ denotes the Euclidian distance:

$$d(x, y) = \|x - y\|_2$$

- For any $x \in \mathbb{R}^p$ and $T \subset \mathbb{R}^p$, we define:

$$d(x, T) = \min_{z \in T}\{d(x, z)\}$$

- Having subsets $S, T \subset \mathbb{R}^p$, and a weight function $w : S \to \mathbb{R}^+$, we define:

$$\text{cost}_w(S, T) = \sum_{x \in S} w(x) d(x, T)^2$$

- Finally, if for all $x \in S$ we have $w(x) = 1$, we simply denote $\mathrm{cost}_w(S, T)$ by $\mathrm{cost}(S, T)$.

Reminder: k-means clustering. The k-means clustering problem is as follows: given a subset $S \subset \mathbb{R}^p$, and an integer k, find the set T (with $|T| = k$), which minimizes $\mathrm{cost}(S, T)$. If a weight function w is also given, the k-means objective would be to minimize $\mathrm{cost}_w(S, T)$, and we call the problem the weighted k-means problem.

Strategy for clustering data streams. We assume we have an algorithm ALG which is an α-approximate weighted k-means clustering algorithm (for some $\alpha > 1$). In other words, given any $S \subset \mathbb{R}^p$, $k \in \mathbb{N}$, and a weight function w, ALG returns a set $T \subset \mathbb{R}^p$, $|T| = k$, such that:

$$\mathrm{cost}_w(S, T) \leq \alpha \min_{|T'|=k} \mathrm{cost}_w(S, T')$$

We will see how we can use ALG as a building block to make an algorithm for the k-means problem on data streams.

The basic idea here is that of divide and conquer: if S is a huge set that does not fit into main memory, we can read a portion of it that does fit into memory, solve the problem on this subset (i.e., do a clustering on this subset), record the result (i.e., the cluster centers and some corresponding weights, as we will see), and then read a next portion of S which is again small enough to fit into memory, solve the problem on this part, record the result, etc. At the end, we will have to combine the results of the partial problems to construct a solution for the main big problem (i.e., clustering S).

To formalize this idea, we consider the following algorithm, which we denote as ALGSTR:

- Partition S into ℓ parts $S_1, ..., S_\ell$.

- For each $i = 1$ to ℓ, run ALG on S_i to get a set of k centers $T_i = \{t_{i1}, t_{i2}, ..., t_{ik}\}$, and assume $\{S_{i1}, S_{i2}, ..., S_{ik}\}$ is the corresponding clustering of S_i (i.e., $S_{ij} = \{x \in S_i | d(x, t_{ij}) < d(x, t_{ij'}) \forall j' \neq j, 1 \leq j' \leq k\}$).

- Let $\hat{S} = \cup_{i=1}^{\ell} T_i$, and define weights $w(t_{ij}) = |S_{ij}|$.

- Run ALG of $\hat{S}$ with weights w, to get k centers T.

- Return T.

Now, we analyze this algorithm. Assuming $T^* = \{t_1^*, ..., t_k^*\}$ to be the optimal k-means solution for S (that is, $T^* = \mathrm{argmin}_{|T'|=k}\{\mathrm{cost}(S, T')\}$), we would like to compare $\mathrm{cost}(S, T)$ (where T is returned by ALGSTR) with $\mathrm{cost}(S, T^*)$.

A small fact might be useful in the analysis below: for any $(a, b) \in R_+$, we have:

$$(a + b)^2 \leq 2a^2 + 2b^2$$

1. First, we show that the cost of the final clustering can be bounded in terms of the total cost of the intermediate clusterings:

Task: Prove that:

$$\mathrm{cost}(S,T) \le 2\cdot \mathrm{cost}_w(\hat{S},T) + 2\sum_{i=1}^{\ell} \mathrm{cost}(S_i,T_i)$$

2. So, to bound the cost of the final clustering, we can bound the terms on the right hand side of the inequality in part (1). Intuitively speaking, we expect the second term to be small compared to $\mathrm{cost}(S,T^*)$, because T^* only uses k centers to represent the data set (S), while the T_i's, in total, use $k\ell$ centers to represent the same data set (and $k\ell$ is potentially much bigger than k). We show this formally:

   Task: Prove that:

   $$\sum_{i=1}^{\ell} \mathrm{cost}(S_i,T_i) \le \alpha\cdot \mathrm{cost}(S,T^*)$$

3. Prove that ALGSTR is a $(4\alpha^2+6\alpha)$-approximation algorithm for the k-means problem.

   Task: Prove that:

   $$\mathrm{cost}(S,T) \le (4\alpha^2+6\alpha)\cdot \mathrm{cost}(S,T^*)$$

   Hint: You might want to first prove two useful facts, which help bound the first term on the right hand side of the inequality in part (1):

   $$\mathrm{cost}_w(\hat{S},T) \le \alpha\cdot \mathrm{cost}_w(\hat{S},T^*)$$

   $$\mathrm{cost}_w(\hat{S},T^*) \le 2\sum_{i=1}^{\ell} \mathrm{cost}(S_i,T_i) + 2\cdot \mathrm{cost}(S,T^*)$$

Additional notes: We have shown above that ALGSTR is a $(4\alpha^2+6\alpha)$-approximation algorithm for the k-means problem. Clearly, $4\alpha^2+6\alpha > \alpha$, so ALGSTR has a somewhat worse approximation guarantee than ALG (with which we started). However, ALGSTR is better suited for the streaming application, as not only it takes just one pass over the data, but also it needs a much smaller amount of memory.

Assuming that ALG needs $\Theta(n)$ memory to work on an input set S of size n (note that just representing S in memory will need $\Omega(n)$ space), if we partitioning S into $\sqrt{n/k}$ equal parts, ALGSTR can work with only $O(\sqrt{nk})$ memory. (Like in the rest of the problem, k represents the number of clusters per partition.)

Note that for typical values of n and k, assuming $k \ll n$, we have $\sqrt{nk} \ll n$. For instance, with $n = 10^6$, and $k = 100$, we have $\sqrt{nk} = 10^4$, which is 100 times smaller than n.

What to Submit

(a) [10pt] Proof that $\mathrm{cost}(S,T) \le 2\cdot \mathrm{cost}_w(\hat{S},T) + 2\sum_{i=1}^{\ell} \mathrm{cost}(S_i,T_i)$.
(b) [10pt] Proof that $\sum_{i=1}^{\ell} \mathrm{cost}(S_i,T_i) \le \alpha\cdot \mathrm{cost}(S,T^*)$.
(c) [10pt] Proof that $\mathrm{cost}(S,T) \le (4\alpha^2+6\alpha)\cdot \mathrm{cost}(S,T^*)$.