Lecture 6:

Summary of Data via Sampling

Course: Algorithms for Big Data

Instructor: Hossein Jowhari

Department of Computer Science and Statistics
Faculty of Mathematics
K. N. Toosi University of Technology

Spring 2021

Outline

- Finding an approximate median in sublinear time
- k-median clustering in sublinear time

Approximate median

Input: A large set of elements $A = \{a_1, \ldots, a_n\}$. We assume D has a total ordering.

Rank of an element: $rank(x) = |\{y \in A \mid y \leq x\}|$

Median: $\mathrm{med}(A) = x$ where $rank(x) = \lceil \frac{n}{2} \rceil$.

Approximate Median: An ϵ-approximate median of A is an $y \in A$ where

$$\lceil \frac{n}{2} \rceil - \epsilon n \leq rank(y) \leq \lceil \frac{n}{2} \rceil + \epsilon n$$

$$Sorted(A) = b_1, b_2, \ldots, \underbrace{b_{\lceil \frac{n}{2} \rceil - \epsilon n}, \ldots, \overbrace{b_{\lceil \frac{n}{2} \rceil}}^{\text{median}}, \ldots, b_{\lceil \frac{n}{2} \rceil + \epsilon n}}_{\epsilon-\text{approximate medians}}, \ldots, b_{n-1}, b_n$$

Finding an approximate median via sampling

Algorithm: Sample s elements from A (with replacement) and return the median of the sample set.

Lemma: If $s \geq \frac{7}{\epsilon^2} \ln(\frac{2}{\delta})$, the algorithm returns an ϵ-approximate median with probability at least $1 - \delta$.

Proof: Partition A into 3 groups:

$$A_L = \{x \in A : rank(x) < \lceil \frac{n}{2} \rceil - \epsilon n\}$$

$$A_M = \{x \in A : \lceil \frac{n}{2} \rceil - \epsilon n \leq rank(x) \leq \lceil \frac{n}{2} \rceil + \epsilon n\}$$

$$A_H = \{x \in A : rank(x) > \lceil \frac{n}{2} \rceil + \epsilon n\}$$

Observation: If less than $\frac{s}{2}$ elements from both A_L and A_H are present in the sample set then the median of the sample is an ϵ-approximate median.

Proof: The argument is similar to what we discussed in Lecture 4 (see page 6).

Let $X_i = 1$ if the i-th sample is from A_L, otherwise $X_i = 0$. $X = \sum_{i=1}^{s} X_i$.

$$E[X] \le (\frac{1}{2} - \epsilon)s$$

Assume $\epsilon \le 0.1$. By Chernoff bound,

$$Pr\big(X \ge \frac{s}{2}\big) \le Pr\big(X \ge (1+\epsilon)E[X]\big) \le e^{-\frac{\epsilon^2}{3}(\frac{1}{2}-\epsilon)s} \le \frac{\delta}{2}$$

By similar argument, if we set $s \geq 7\epsilon^{-2}\ln(\frac{2}{\delta})$ (assuming $\epsilon \leq 0.1$) the probability that the number of elements from A_H in the sample set is at least $\frac{s}{2}$ is bounded by $\delta/2$.

By union bound, number of elements from both A_L and A_H in the sample set is less than $\frac{s}{2}$ with probability at least $1-\delta$.

Therefore with probability $1-\delta$, the output of the algorithm is an ϵ-approximate median of A.

Sample complexity: $O(\frac{1}{\epsilon^2}\ln(\frac{1}{\delta}))$

Homework: Generalize this result to the problem of finding an element with (approximate) rank t.

k-median clustering

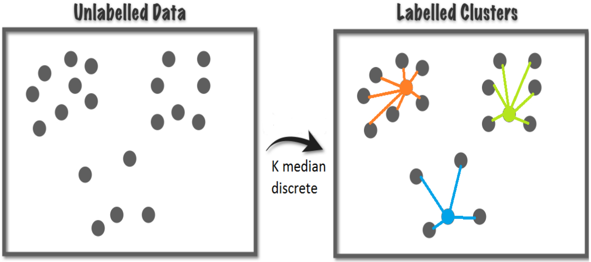
k-median clustering problem: Given a metric (X, d) where X is a finite set of data points and d is a distance defined over X, in the (discrete) k-median problem, the goal is to select k center points $c_1, \ldots, c_k$ from X, so that the sum of distances to the closest center is minimized.

$$X = \{x_1, \ldots, x_n\}$$

$$\min_{c_1,\ldots,c_k \subseteq X} \sum_{i=1}^{n} \min_{j=1,\ldots,k} \{d(x_i, c_j)\}$$

Note: In a metric space, the distance is a symmetric function and the triangle inequality holds.

Note: If $|X| = n$, the metric (X, d) can be represented by a symmetric n by n matrix.

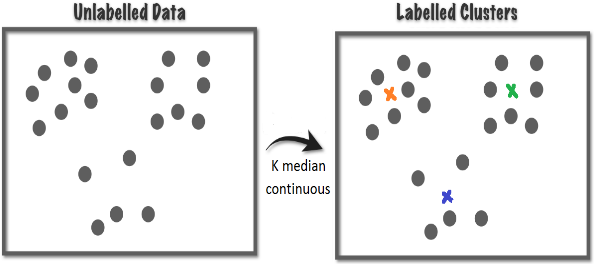


Note: The problem is equivalent to the problem of minimizing the average distance to the closest center.

$$\min_{c_1,\ldots,c_k \subseteq X} \frac{1}{n}\sum_{i=1}^{n} \min_{j=1,\ldots,k} \{d(x_i, c_j)\}$$

Continuous k-median problem

In the continuous version, the finite set of points X lie in a continuous space (for example $X \subset \mathbb{R}^d$ with the Euclidean distance.) Here we are allowed to choose the k centers from the entire space, not just from the given points X.



Note: Both discrete and continuous versions of k-median clustering are NP-hard problems. It means, assuming $NP \neq P$, there is no polynomial time algorithm for finding an optimal k-median clustering.

Some algorithmic facts

- Trivially, there is a $O(kn^{k+1})$ time algorithm for finding an optimal k-median clustering (discrete version). why?

  There are $\binom{n}{k} = O(n^k)$ ways for selecting the centers.

- The problem is NP-hard even for points in $\mathbb{R}^2$.

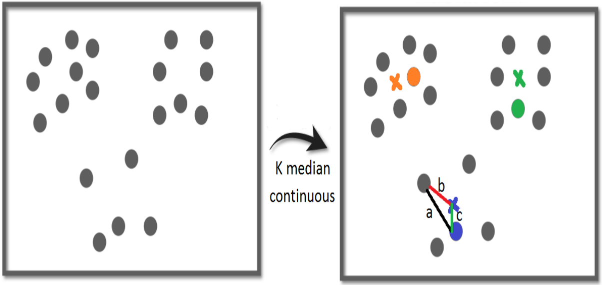
- There is a polynomial time approximation algorithm for k-median clustering that returns a solution with cost at most $\alpha = 2.611$ times the optimal cost.

- There is $O(n \log n \log k)$ time constant factor approximation algorithm for k-median clustering when the points lie in $\mathbb{R}^d$ with constant d.

Lemma: An optimal solution for the discrete version is a 2-factor approximation solution for the continuous version.

Proof: Use triangle inequality.

Replace each optimal continuous center with its closest point in X. See the figure below.

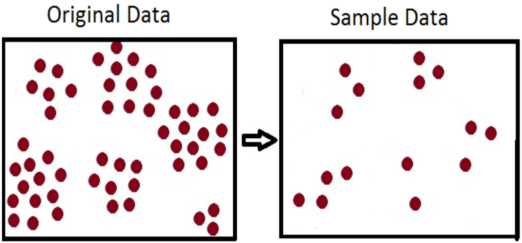


$$a \leq b + c, \quad c \leq b$$

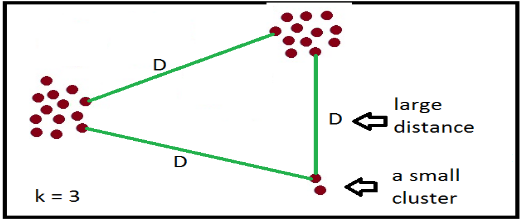
$$\Rightarrow \quad a \leq 2b$$

Corollary: Any α-factor approximation algorithm for the discrete version is a 2α-factor approximation algorithm for the continuous version.

Sublinear time clustering via sampling



Is the sample a good representative of the whole data?



In general, we need to see the whole data to get a good approximation.

If we make certain assumptions about the data, we may hope that a small sample is a good representative of the whole.

Some algorithmic results in this direction:

- There is a $\tilde{O}(\frac{D^2}{\epsilon^2} k \ln(\frac{n}{\delta}))$ time randomized algorithm that returns a solution with cost at most $O(OPT) + \epsilon n$ with probability $1 - \delta$. Here D is the diameter of the points. Mishra, Oblinger, Pitt, 2001.

- There is a $O(\frac{k^3}{\epsilon^2} \log^3 k)$ time randomized algorithm that returns a solution with cost $O(OPT)$ under the assumption that every optimal cluster is of size at least $\Omega(\frac{n\epsilon}{k})$. Meyerson et al.2004

- There is a $O(\frac{D}{\epsilon^2} k \ln(\frac{1}{\delta}))$ time randomized algorithm that returns a solution with cost at most $O(OPT) + \epsilon n$ with probability $1 - \delta$. Czumaj, Sohler.

Mishra, Oblinger, Pitt (MOP)'s Algorithm

Assumption: Suppose there is a deterministic α-factor approximation algorithm A for the k-median clustering problem that runs in $T(n, k, \alpha)$ time.

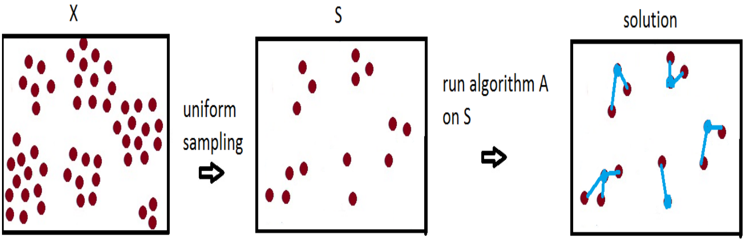
MOP's Idea:

- Fix $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$.
- Pick a sample S of size at least $\frac{(\alpha D)^2}{\epsilon^2} k \ln(\frac{n}{\delta})$ from the points X. Here D is the diameter of the input points.
- Run algorithm A on the sample S and return the solution.

Claim: With probability at least $1 - \delta$, we have

$$cost(MOP^{AVG}) \leq 2\alpha\, cost(OPT^{AVG}) + \epsilon$$

MOP's algorithm



Main Tool:

(Haussler/Pollard) *Let F be a finite set of functions on X with $0 \leq f(x) \leq M$ for all $f \in F$ and $x \in X$. Let $S = x_1, \ldots, x_m$ be a sequence of m examples drawn independently and identically from X and let $\epsilon > 0$. $Pr(\exists f \in F : |E_X(f) - E_S(f)| \geq \epsilon) \leq \delta$ when $m \geq \frac{M^2}{2\epsilon^2}(\ln |F| + \ln \frac{2}{\delta})$.*

More details for the next lecture.