A note on 1-median selection in metric spaces

Ching-Lueh Chang
Department of Computer Science and Engineering
Yuan Ze University, Taoyuan, Taiwan
Innovation Center for Big Data and Digital Convergence
Yuan Ze University, Taoyuan, Taiwan
clchang@saturn.yzu.edu.tw

Abstract

Consider the problem of finding a point x in a metric space of size $n \in \{k^h \mid k \in \mathbb{Z}^+\}$ to minimize the average distance from x to other points, where $h \in \mathbb{Z}^+ \setminus \{1\}$. We show that this problem has a deterministic, nonadaptive, $O(hn^{1+1/h})$-time, $O(n^{1+1/h})$-query and $(2h)$-approximation algorithm by modifying Chang's [2] proof of a similar result.

1 Introduction

A metric space (M, d) is a nonempty set M endowed with a function $d\colon M \times M \to [0, \infty)$ such that $d(x, x) = 0$, $d(x, y) = d(y, x)$, $d(x, y) + d(y, z) \geq d(x, z)$ and $d(x, u) > 0$ for all x, y, $z \in M$ and $u \in M \setminus \{x\}$ [7]. Given an n-point metric space (M, d), the METRIC 1-MEDIAN problem asks for $\operatorname{argmin}_{x \in M} \sum_{y \in M} d(x, y)$, breaking ties arbitrarily. An algorithm for METRIC 1-MEDIAN is nonadaptive if its queries (i.e., the pairs $(x, y) \in M \times M$ such that $d(x, y)$ is asked for) depend on n but not on d.

Indyk [4, 5] shows that METRIC 1-MEDIAN has a Monte-Carlo $O(n/\epsilon^2)$-time $(1 + \epsilon)$-approximation algorithm with a success probability of $\Omega(1)$, where $\epsilon > 0$. Other results on METRIC 1-MEDIAN and on the more general problem of metric k-median selection abound, especially for Euclidean spaces. See, e.g., [3, 5, 6].

We will focus on deterministic $o(n^2)$-time algorithms for METRIC 1-MEDIAN; such algorithms read an $o(1)$ fraction of all distances. In this respect, Guha et al. [3, Secs. 3.1–3.2] give a deterministic $O(n^{1+\epsilon})$-time $O(n^{\epsilon})$-space $2^{O(1/\epsilon)}$-approximation algorithm that reads the distances in only one pass. This paper modifies Chang's [2] technique to show that for all $h \in \mathbb{Z}^+ \setminus \{1\}$, METRIC 1-MEDIAN with $n \in \{k^h \mid k \in \mathbb{Z}^+\}$ has a deterministic, nonadaptive, $O(hn^{1+1/h})$-time, $O(n^{1+1/h})$-query and $(2h)$-approximation algorithm. Previously, a series of works by Chang [1, 2] and Wu [8] establish the same result with the larger query complexity of $O(hn^{1+1/h})$ but without the restriction that $n \in \{k^h \mid k \in \mathbb{Z}^+\}$.[1]

2 Main result

Let (S^h, d) be a metric space and $n \stackrel{\text{def.}}{=} |S|^h$, where S is a finite set and $h \in \mathbb{Z}^+ \setminus \{1\}$. For u_1, u_2, …, u_h, v_1, v_2, …, $v_h \in S$,

$$\begin{aligned} & \tilde{d}\left((u_1, u_2, \ldots, u_h), (v_1, v_2, \ldots, v_h)\right) \\ \stackrel{\text{def.}}{=} & \sum_{i=0}^{h-1} d\left((u_{i+1}, u_{i+2}, \ldots, u_h, v_1, v_2, \ldots, v_i),\right. \\ & \left.(u_{i+2}, u_{i+3}, \ldots, u_h, v_1, v_2, \ldots, v_{i+1})\right); \quad (1) \end{aligned}$$

hence

$$\begin{aligned} & \tilde{d}\left((u_1, u_2, \ldots, u_h), (v_1, v_2, \ldots, v_h)\right) \\ \geq & \; d\left((u_1, u_2, \ldots, u_h), (v_1, v_2, \ldots, v_h)\right) \quad (2) \end{aligned}$$

by the triangle inequality for d. Note that a sequence with a starting index greater than the ending index is empty by convention. So for example,

$$(u_{i+2}, u_{i+3}, \ldots, u_h, v_1, v_2, \ldots, v_{i+1}) = (v_1, v_2, \ldots, v_h)$$

when $i = h - 1$.

The following lemma shows that a 1-median with respect to $\tilde{d}$ is a $(2h)$-approximate 1-median with respect to d.

Lemma 1. *Let*

$$\begin{aligned} \hat{\mathbf{u}} \stackrel{\text{def.}}{=} & \operatorname*{argmin}_{(u_1, u_2, \ldots, u_h) \in S^h} \sum_{v_1, v_2, \ldots, v_h \in S} \\ & \tilde{d}\left((u_1, u_2, \ldots, u_h), (v_1, v_2, \ldots, v_h)\right), \quad (3) \end{aligned}$$

[1] The $O(h)$ factor in the time and query complexities is omitted in [8] because h is independent of n.

breaking ties arbitrarily. Then for all $\mathbf{x} \in S^h$,

$$\begin{aligned} & \sum_{v_1,v_2,\ldots,v_h \in S} d\left(\hat{\mathbf{u}}, (v_1, v_2, \ldots, v_h)\right) \\ \leq \; & 2h \cdot \sum_{v_1,v_2,\ldots,v_h \in S} d\left(\mathbf{x}, (v_1, v_2, \ldots, v_h)\right). \end{aligned}$$

Proof. Let $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, …, $\boldsymbol{q}_h$, $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, …, $\boldsymbol{r}_h$ be independent and uniformly random elements of S. All expectations in the proof will be taken over these random variables.

By equation (3),

$$\begin{aligned} & \mathrm{E}\left[\tilde{d}\left(\hat{\mathbf{u}}, (\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_h)\right)\right] \\ = \; & \min_{u_1,u_2,\ldots,u_h \in S} \\ & \mathrm{E}\left[\tilde{d}\left((u_1, u_2, \ldots, u_h), (\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_h)\right)\right]. \end{aligned} \tag{4}$$

Now,

$$\begin{aligned} & \frac{1}{n} \cdot \sum_{v_1,v_2,\ldots,v_h \in S} d\left(\hat{\mathbf{u}}, (v_1, v_2, \ldots, v_h)\right) \\ = \; & \mathrm{E}\left[d\left(\hat{\mathbf{u}}, (\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_h)\right)\right] \\ \overset{(2)}{\leq} \; & \mathrm{E}\left[\tilde{d}\left(\hat{\mathbf{u}}, (\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_h)\right)\right] \\ \overset{(4)}{\leq} \; & \mathrm{E}\left[\tilde{d}\left((\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_h), (\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_h)\right)\right] \\ \overset{(1)}{=} \; & \mathrm{E}\left[\sum_{i=0}^{h-1} d\left((\boldsymbol{q}_{i+1}, \boldsymbol{q}_{i+2}, \ldots, \boldsymbol{q}_h, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_i),\right.\right. \\ & \left.\left.(\boldsymbol{q}_{i+2}, \boldsymbol{q}_{i+3}, \ldots, \boldsymbol{q}_h, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{i+1})\right)\right] \\ \leq \; & \mathrm{E}\left[\sum_{i=0}^{h-1} d\left(\mathbf{x}, (\boldsymbol{q}_{i+1}, \boldsymbol{q}_{i+2}, \ldots, \boldsymbol{q}_h, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_i)\right)\right. \\ & \left. + d\left(\mathbf{x}, (\boldsymbol{q}_{i+2}, \boldsymbol{q}_{i+3}, \ldots, \boldsymbol{q}_h, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{i+1})\right)\right] \\ = \; & \sum_{i=0}^{h-1} \mathrm{E}\left[d\left(\mathbf{x}, (\boldsymbol{q}_{i+1}, \boldsymbol{q}_{i+2}, \ldots, \boldsymbol{q}_h, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_i)\right)\right] \\ + \; & \sum_{i=0}^{h-1} \mathrm{E}\left[d\left(\mathbf{x}, (\boldsymbol{q}_{i+2}, \boldsymbol{q}_{i+3}, \ldots, \boldsymbol{q}_h, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{i+1})\right)\right] \\ = \; & \frac{2h}{n} \cdot \sum_{v_1,v_2,\ldots,v_h \in S} d\left(\mathbf{x}, (v_1, v_2, \ldots, v_h)\right), \end{aligned}$$

where the last equality follows from the uniform distribution of $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, …, $\boldsymbol{q}_h$, $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, …, $\boldsymbol{r}_h$ and their independence. □

By equation (1), we may compute $\tilde{d}(\mathbf{u}, \mathbf{v})$ for all $\mathbf{u}$, $\mathbf{v} \in S^h$ with only $|S|^{h+1} = n^{1+1/h}$ distinct queries to d. So Lemma 1 alone gives a deterministic $O(n^{1+1/h})$-query $(2h)$-approximation algorithm for METRIC 1-MEDIAN. But the time complexity would be $O(hn^2)$, which we now proceed to improve to $O(hn^{1+1/h})$.

For u_1, u_2, …, $u_h \in S$ and $k \in \{0, 1, \ldots, h\}$,

$$\begin{aligned} & f\left((u_1, u_2, \ldots, u_h), k\right) \\ \overset{\text{def.}}{=} \; & \sum_{v_1,v_2,\ldots,v_k \in S} \sum_{i=0}^{k-1} d\left((u_{i+1}, u_{i+2}, \ldots, u_h, v_1, v_2, \ldots, v_i),\right. \\ & \left.(u_{i+2}, u_{i+3}, \ldots, u_h, v_1, v_2, \ldots, v_{i+1})\right). \end{aligned} \tag{5}$$

As empty sums vanish,

$$f(\cdot, 0) \equiv 0. \tag{6}$$

Clearly,

$$\begin{aligned} & f\left((u_1, u_2, \ldots, u_h), h\right) \\ = \; & \sum_{v_1,v_2,\ldots,v_h \in S} \sum_{i=0}^{h-1} \\ & d\left((u_{i+1}, u_{i+2}, \ldots, u_h, v_1, v_2, \ldots, v_i),\right. \\ & \left.(u_{i+2}, u_{i+3}, \ldots, u_h, v_1, v_2, \ldots, v_{i+1})\right) \\ \overset{(1)}{=} \; & \sum_{v_1,v_2,\ldots,v_h \in S} \\ & \tilde{d}\left((u_1, u_2, \ldots, u_h), (v_1, v_2, \ldots, v_h)\right). \end{aligned} \tag{7} \tag{8}$$

The following lemma shows how to compute $f(\cdot, k+1)$ in the increasing order of $k \in \{0, 1, \ldots, h-1\}$ by standard dynamic programming.

Lemma 2. *For all* u_1, u_2, …, $u_h \in S$ *and* $k \in \{0, 1, \ldots, h-1\}$,

$$\begin{aligned} & f\left((u_1, u_2, \ldots, u_h), k+1\right) \\ = \; & |S|^k \cdot \sum_{v_1 \in S} d\left((u_1, u_2, \ldots, u_h), (u_2, u_3, \ldots, u_h, v_1)\right) \\ + \; & \sum_{v_1 \in S} f\left((u_2, u_3, \ldots, u_h, v_1), k\right). \end{aligned}$$

Proof. By equation (5),

$$\begin{aligned} & f\left((u_1, u_2, \ldots, u_h), k+1\right) \\ = \; & \sum_{v_1,v_2,\ldots,v_{k+1} \in S} d\left((u_1, u_2, \ldots, u_h), (u_2, u_3, \ldots, u_h, v_1)\right) \\ + \; & \sum_{v_1,v_2,\ldots,v_{k+1} \in S} \sum_{i=1}^{k} d\left((u_{i+1}, u_{i+2}, \ldots, u_h, v_1, v_2, \ldots, v_i),\right. \\ & \left.(u_{i+2}, u_{i+3}, \ldots, u_h, v_1, v_2, \ldots, v_{i+1})\right). \end{aligned} \tag{9}$$

We have

$$\begin{aligned} & \sum_{v_1,v_2,\ldots,v_{k+1} \in S} d\left((u_1, u_2, \ldots, u_h), (u_2, u_3, \ldots, u_h, v_1)\right) \\ = \; & |S|^k \cdot \sum_{v_1 \in S} d\left((u_1, u_2, \ldots, u_h), (u_2, u_3, \ldots, u_h, v_1)\right) \end{aligned}$$

because the common summand of both sides is independent of v_2, v_3, …, v_{k+1}. By equation (5) (with u_1, u_2, …, u_{h-1}, u_h, v_1, v_2, …, v_k replaced by u_2, u_3, …, u_h, v_1, v_2, …, v_{k+1}, respectively,

1: **for** u_1, u_2, …, $u_h \in S$ **do**
2: $f[(u_1, u_2, \ldots, u_h)][0] \leftarrow 0$;
3: **end for**
4: **for** $k = 0$ up to $h-1$ **do**
5: **for** u_1, u_2, …, $u_h \in S$ **do**
6: $f[(u_1, u_2, \ldots, u_h)][k+1] \leftarrow |S|^k \cdot \sum_{v \in S} d((u_1, u_2, \ldots, u_h), (u_2, u_3, \ldots, u_h, v))$;
7: $f[(u_1, u_2, \ldots, u_h)][k+1] \leftarrow f[(u_1, u_2, \ldots, u_h)][k+1] + \sum_{v \in S} f[(u_2, u_3, \ldots, u_h, v)][k]$;
8: **end for**
9: **end for**
10: Output $\operatorname{argmin}_{(u_1,u_2,\ldots,u_h) \in S^h} f[(u_1, u_2, \ldots, u_h)][h]$, breaking ties arbitrarily;

Figure 1: Algorithm FIND-MEDIAN.

and after adjusting the indices),

$$\begin{aligned} & f\left((u_2, u_3, \ldots, u_h, v_1), k\right) \\ = & \sum_{v_2, v_3, \ldots, v_{k+1} \in S} \sum_{i=1}^{k} \\ & d\left((u_{i+1}, u_{i+2}, \ldots, u_h, v_1, v_2, \ldots, v_i), \right. \\ & \left. (u_{i+2}, u_{i+3}, \ldots, u_h, v_1, v_2, \ldots, v_{i+1})\right). \end{aligned} \tag{10}$$

for each $v_1 \in S$. Equations (9)–(10) complete the proof. □

Without loss of generality, assume $S = \{0, 1, \ldots, n^{1/h} - 1\}$. Then every tuple in S^h can be accessed as an $O(\log n)$-bit word in $O(1)$ time under the unit-cost RAM model (which is standard in the analysis of algorithms). So, once we have computed $f(\cdot, k)$, we can compute $f(\mathbf{u}, k+1)$ in $O(|S|)$ time for each $\mathbf{u} \in S$ by Lemma 2.

We now arrive at our main theorem.

Theorem 3. *Let $h \in \mathbb{Z}^+ \setminus \{1\}$. Then* METRIC 1-MEDIAN *has a deterministic, nonadaptive, $O(hn^{1+1/h})$-time, $O(n^{1+1/h})$-query and $(2h)$-approximation algorithm for a metric space of size a perfect hth power.*

Proof. By equation (6), lines 1–3 of FIND-MEDIAN in Fig. 1 compute $f(\cdot, 0)$. By Lemma 2, lines 4–9 compute $f(\cdot, k+1)$ in the increasing order of $k \in \{0, 1, \ldots, h-1\}$. By equations (7)–(8), line 10 outputs a 1-median with respect to $\tilde{d}$; hence Lemma 1 gives the approximation ratio of $2h$.

It is easy to see that FIND-MEDIAN runs in time $O(hn|S|) = O(hn^{1+1/h})$ deterministically and nonadaptively. Because every query (to d) of FIND-MEDIAN is for

$$d\left((u_1, u_2, \ldots, u_h), (u_2, u_3, \ldots, u_h, v)\right)$$

for some u_1, u_2, …, u_h, $v \in S$, the query complexity is at most $|S|^{h+1} = O(n^{1+1/h})$. □

Let us briefly describe the major technical difference between this and the related papers. By equation (1), computing $\tilde{d}(\mathbf{u}, \mathbf{v})$ for all $\mathbf{u}$, $\mathbf{v} \in S^h$ requires only $|S|^{h+1}$ (distinct) queries to d. However, the same does not hold for Chang's [2] version of $\tilde{d}$, forbidding him to lower the query complexity from $O(hn^{1+1/h})$ to $O(n^{1+1/h})$. Wu [8] uses the famous "median of medians" technique, which is entirely different from that of Chang, and gives a deterministic, adaptive, $O(hn^{1+1/h})$-time, $O(hn^{1+1/h})$-query and $(2h)$-approximation algorithm for all $h \in \mathbb{Z}^+ \setminus \{1\}$.

Acknowledgments

The author is supported in part by the Ministry of Science and Technology of Taiwan under grant 103-2221-E-155-026-MY2.

References

[1] C.-L. Chang. Deterministic sublinear-time approximations for metric 1-median selection. *Information Processing Letters*, 113(8):288–292, 2013.

[2] C.-L. Chang. A deterministic sublinear-time nonadaptive algorithm for metric 1-median selection. Technical Report arXiv: 1502.06764, 2015.

[3] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):515–528, 2003.

[4] P. Indyk. Sublinear time algorithms for metric space problems. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 428–434, 1999.

[5] P. Indyk. *High-dimensional computational geometry*. PhD thesis, Stanford University, 2000.

[6] A. Kumar, Y. Sabharwal, and S. Sen. Linear-time approximation schemes for clustering problems in any dimensions. *Journal of the ACM*, 57(2):5, 2010.

[7] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 3rd edition, 1976.

[8] B.-Y. Wu. On approximating metric 1-median in sublinear time. *Information Processing Letters*, 114(4):163–166, 2014.