Median using 1.5n + o(n) comparisons

Indian Institute of Technology Kanpur
Randomized Algorithms

Group Name	#Group#
Instructor	Surender Baswana
Course	CS648A
Submitted on	31-03-2024

Introduction

It has been proven that every deterministic algorithm for finding the exact median of n numbers must perform at least $2n$ comparisons in the worst case. However, there is a randomized Las Vegas algorithm that computes the exact median of n numbers while performing only $1.5n+o(n)$ comparisons on expectation. This report contains the description and analysis of such an algorithm and discusses how we arrived at it.

Intuition

To determine the median of a set of elements, it is crucial to identify a continuous range of elements containing the median and establish the rank of at least one element within that range. Specifically, let the given set of n elements be $a_1, a_2, a_3, ..., a_n$ such that $a_1 \leq a_2 \leq a_3 \leq ... \leq a_n$. We need to find $a_l, a_{l+1}, ..., a_{r-1}, a_r$ such that $1 \leq l \leq \frac{n}{2} \leq r \leq n$, and determine the rank of at least one a_i where $l \leq i \leq r$. Once we have the range of elements and the rank of one element, we can find the median by sorting the range of elements and using the element with the known rank.

Idea

Let the number of elements in the range be m. Sorting m elements requires $O(m \log m)$ comparisons. Therefore, m must be much smaller than n. To obtain the desired range of elements, we can remove elements that are not contenders for the median in one or more steps. To remove elements, in each step we can randomly and uniformly select an element from the set of elements. Finding the rank of this element partitions the given set into sets: the set containing the median and the set not containing the median. We can discard the set not containing the median. However, if the randomly selected element is far away from the median, the number of elements discarded will be very less. Therefore, an element that is closer to the median is desired. Obviously, we will stop if the selected element is itself the median.

Half Approximate Median

An element whose rank in the given set of n elements falls within the range $[\frac{n}{4}, \frac{3n}{4}]$ is considered a Half Approximate Median. It can be found as follows: randomly and uniformly pick k elements from the given set of n elements with replacement. The median of these elements is a Half Approximate Median with a probability of at least $1 - 2n^{-2}$ for $k = 10 \log n$. The number of comparisons is $O(k \log k) = o(n)$.

Using Half Approximate Median

Instead of randomly and uniformly selecting an element in each step, we can find a Half Approximate Median of the set of elements and use it to discard elements that are not contenders for the median of the original set of elements.

Analysis

Let the probability of median being present among k elements picked and the probability of median getting reported as Half Approximate Median in the i^{th} round be p_i and q_i respectively. Let the number of elements present and the number of comparisons (ignoring $o(n)$) in the i^{th} round be n_i and c_i respectively. Let total number of comparisons be c.

Round 1

$$p_1 = \frac{k}{n} \implies q_1 \leq \frac{k}{n}$$
$$c_1 = n_1 = n$$

Round 2

$$c_2 = n_2 \geq \frac{n}{2}$$
$$p_2 \leq \frac{2k}{n} \implies q_2 \leq \frac{2k}{n}$$

Round 3

$$c_3 = n_3 \geq \frac{n}{4}$$

probability of entering Round 3 is $\geq 1 - \frac{3k}{n} \implies P(c \geq 1.75n) \geq 1 - \frac{3k}{n}$

$$k = 10\log n \implies k \ll n \implies E[c] \geq 1.75n$$

Therefore, this method is not efficient enough.

Insight and New Idea

Based on the above algorithm, the insight gained is that we can find the ranks of no more than 2 elements. Therefore, we can find the median by identifying two elements a_l and a_r such that $1 \leq l \leq \frac{n}{2} \leq r \leq n$ and $r - l = o(n)$. Then we can find all elements a_i such that $l \leq i \leq r$ by comparing them with a_l and a_r. Finally, we can determine the median by sorting these elements.

Towards designing efficient Algorithm

Let the given set be $p_1, p_2, p_3, ..., p_n$ is a permutation of $a_1, a_2, a_3, ..., a_n$. Sample a set S of distinct integers in the range $[1, n]$ as follows: pick each integer i in the range $[1, n]$ randomly, uniformly and independently with probability $\frac{k}{n}$. Let $S' = \{p_i | i \in S\}$. Sort S'. The two elements a_l and a_r are identified by choosing the $(\frac{k}{2} - \alpha)^{th}$ element and $(\frac{k}{2} + \alpha)^{th}$ element in S' as a_l and a_r. The values of k and α are to be fixed later on.

Analysis

Let the size of set S' be h. Then,

$$P(|h - k| \geq \frac{k}{10}) \leq e^{-\frac{k}{150}}$$

The size of set S' lies in the range $[\frac{9k}{10}, \frac{11k}{10}]$ with a probability of at least $1 - e^{-\frac{k}{150}}$. Sorting S' will take $O(k \log k)$ comparisons. The median doesn't lie in the range $[l, r]$ when $l > \frac{n}{2}$ or $r < \frac{n}{2}$. Let the number of elements a_i in S' such that $i < \frac{n}{2}$ be z.

$$l > \frac{n}{2} \iff z < \frac{k}{2} - \alpha$$
$$P(z < \frac{k}{2} - \alpha) = P(z < E[z] - \alpha) \leq e^{-\frac{2\alpha^2}{3k}}$$
$$r < \frac{n}{2} \iff z \geq \frac{k}{2} + \alpha$$

$$P(z \geq \frac{k}{2} + \alpha) = P(z \geq E[z] + \alpha) \leq e^{-\frac{2\alpha^2}{3k}}$$

Therefore,

$$P(\frac{n}{2} \in [l, r]) \geq 1 - 2e^{-\frac{2\alpha^2}{3k}}$$

The median lies in the range $[l, r]$ with a probability of at least $1 - 2e^{-\frac{2\alpha^2}{3k}}$.
Let b be an integer in the range $[1, \frac{n}{2})$ and d be the number of elements a_i in S' such that $1 \leq i \leq \frac{n}{2} - b$. Let $E[d] = \frac{k}{2} - f$ where f is an integer in the range $[0, \frac{k}{2}]$.

$$l < \frac{n}{2} - b \iff d > \frac{k}{2} - \alpha$$

$$P(d > \frac{k}{2} - \alpha) = P(d > \frac{k}{2} - f + f - \alpha) = P(d > E[d] + f - \alpha)$$

$$P(d > E[d] + f - \alpha) \leq e^{-\frac{(f-\alpha)^2}{3(\frac{k}{2} - f)})} < e^{-\frac{(f-\alpha)^2}{\frac{3k}{2}}}$$

For $f = 2\alpha$,

$$P(d > E[d] + f - \alpha) < e^{-\frac{2\alpha^2}{3k}}$$

Let g be the number of elements a_i in S' such that $\frac{n}{2} - b < i \leq \frac{n}{2}$.

$$z = d + g$$
$$E[z] = E[d] + E[g]$$
$$\frac{k}{2} = \frac{k}{2} - f + E[g]$$
$$E[g] = f = 2\alpha$$
$$\frac{bk}{n} = 2\alpha$$
$$b = \frac{2n\alpha}{k}$$

Therefore,

$$P(l < \frac{n}{2} - \frac{2n\alpha}{k}) < e^{-\frac{2\alpha^2}{3k}}$$

Similarly, for r using symmetry,

$$P(r > \frac{n}{2} + \frac{2n\alpha}{k}) < e^{-\frac{2\alpha^2}{3k}}$$

Therefore,

$$P(r - l \leq \frac{4n\alpha}{k}) \geq 1 - 2e^{\frac{2\alpha^2}{3k}}$$

Compare each element a_i with a_l: if $a_i < a_l$, discard it; otherwise, compare a_i with a_r and insert a_i into set Q if $a_i \leq a_r$.
The number of comparisons for a_i such that $i < l$ is 1.
The number of comparisons for a_i such that $i \geq l$ is 2.
The number of comparisons required for all elements is at most $1.5n + \frac{2n\alpha}{k}$.
Sorting Q require $O(\frac{4n\alpha}{k} \log \frac{4n\alpha}{k})$ comparisons.
Therefore, the total number of comparisons is $O(k \log k) + 1.5n + \frac{2n\alpha}{k} + O(\frac{4n\alpha}{k} \log \frac{4n\alpha}{k})$.
To minimize the comparisons required to sort S' and Q,

$O(k) = O(\frac{4n\alpha}{k})$ and $O(\alpha) = O(\sqrt{k \log n}) \implies k = \frac{1}{5}(n^2 \log n)^{\frac{1}{3}} \implies \alpha = \frac{1}{5}\sqrt{k \log n}$

Constants are taken appropriately to minimize the number of comparisons.
Since,

$$P((\frac{n}{2} \notin [l, r]) \bigcup (r - l > \frac{4n\alpha}{k})) \leq 4e^{-\frac{2\alpha^2}{3k}}$$

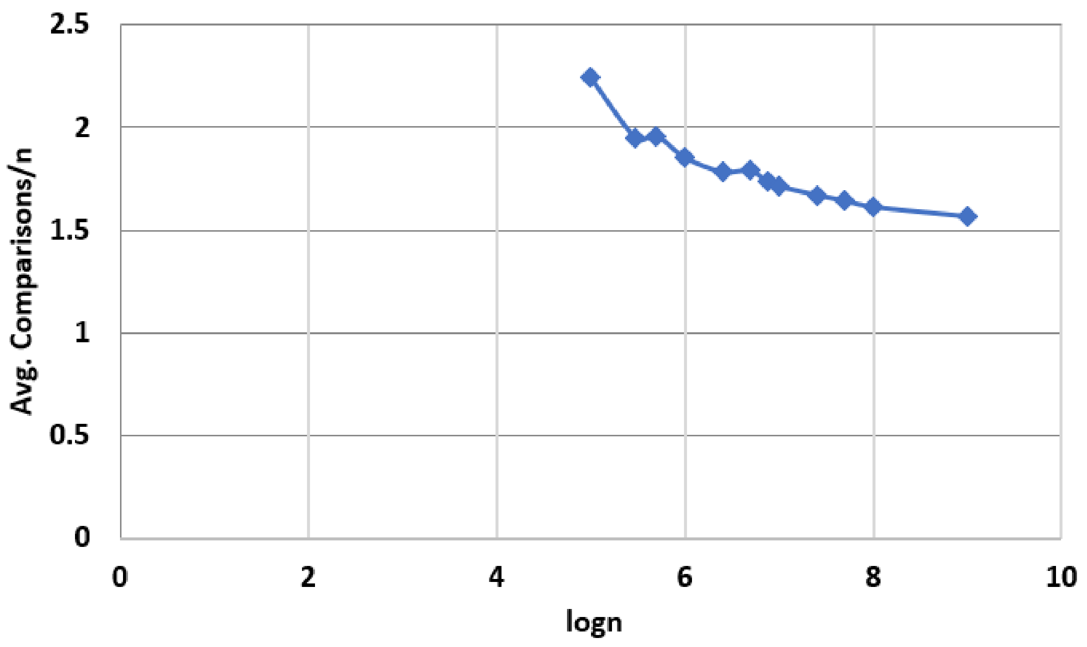
$k = \frac{1}{5}(n^2 \log n)^{\frac{1}{3}}$ and $\alpha = \frac{1}{5}\sqrt{k \log n} \implies \frac{2\alpha^2}{3k} = O(\log n)$

Therefore, error probability is inverse polynomial in n.

Monte Carlo to Las Vegas

While $h \notin [\frac{9k}{10}, \frac{11k}{10}]$ or $l > \frac{n}{2}$ or $r < \frac{n}{2}$, start from sampling S again.

Variation of Average comparisons with n


Avg. Comparisons/n
logn