CS 388R: Randomized Algorithms, Fall 2023 10/4/2023

Lecture 13: Sampling and Median Finding

Prof. Eric Price *Scribe: Gary Wang, Trung Nguyen*

NOTE: THESE NOTES HAVE NOT BEEN EDITED OR CHECKED FOR CORRECTNESS

1 Overview

We consider the idea of random sampling and show an application for median finding.

2 Sampling Example

Goal: Estimate π by inscribing a circle inside a square of side length 2 and sampling random points.

$Pr[\text{in circle}] = \frac{\pi}{4}$, $O(\frac{1}{\epsilon^2}\log\frac{2}{\delta})$ samples needed to estimate to ϵ precision with $1-\delta$ confidence.

Can be extended to any polytope/polyhedron, but if the object is small with respect to the bounding box/cube, we need to take a number of samples until we have seen at least some number of points inside the object. The number of samples needed remains linear with respect to the object's area/volume.

3 Median Finding

Goal: Given x_1 to x_n array of n unsorted real numbers, return the median number.
More general problem: return the r^{th} smallest element.
Some algorithms that can be used:

3.1 Quicksort

We sort the list and return return the median. The runtime is $O(n \log n)$.

3.2 Quickselect

We use quickselect with one recursive call on the same side as the median. The expected runtime is $O(n)$, and is $O(n\frac{\log n}{\log\log n})$ whp.

Proof. We show that Quickselect is $O(n)$ expected, $O(n\frac{\log n}{\log\log n})$ whp.
Expected:

Similar to quicksort analysis,the pivot can shave off $\frac{1}{4}$ of the elements with $\frac{1}{2}$ probability. Therefore it takes $O(1)$ time for an array to go from size n to size $\frac{3}{4}n$. From a geometric sum with common ratio $\frac{3}{4}$, the expected runtime is indeed $O(n)$.
With High Probability:
We note that in this problem, the probability of all of the first k choices for a pivot lie before $\frac{n}{k}$ is at least $1/k^k$.
If this case happens, then k pivots has reduced our array to size $n(1-1/k)^k \approx n/e$.
Therefore, there's a $1/k^k$ chance of taking $\Omega(kn)$ time and thus a $1/n$ chance of taking $\Omega(\frac{n \log n}{\log \log n})$ time.
As such, we cannot show that Quickselect is $O(n)$ with high probability but rather $\Omega(\frac{n \log n}{\log \log n})$ with high probability, which is not much better than sorting the array.

$\square$

3.3 Median-of-Medians

There exists a determinisic algorithm that is $O(n)$ worst case. Ref CLRS.
Overall method:
Split array elements into groups of 5 and take median of each group, take recursive median of the medians use that as pivot.
$T(n) = O(n) + T(\frac{n}{5}) + T(\frac{7}{10}n) \implies O(n)$ Master Thm.
Today: Show $1.5n + o(n)$

4 Median By Sampling

Let S be the subset of X obtained by sampling each element in X independently with probability p. Using Chernoff bound, $|S| = \Theta(np)$ w.h.p. in n.
What is the rank of median of X in S?
Denote the rank of median of X in S by $rank_S(med(X))$ Let Z_i be the indicator of the event that element i of S is at most median of X.
So $rank_S(med(X)) = \sum_{i=1}^{|S|} Z_i$.
We have $\mathbb{P}[Z_i = 1] = 0.5$, so $\mathbb{E}[rank_S(med(X))] = |S|/2$.
Applying additive Chernoff bound, we have

$$\mathbb{P}[rank_S(med(X)) > \frac{|S|}{2} + t] \leq e^{-2t^2/|S|}$$

$$\mathbb{P}[rank_S(med(X)) < \frac{|S|}{2} - t] \leq e^{-2t^2/|S|}$$

Choose $t = \sqrt{|S| \ln n}$, so $\frac{|S|}{2} - t \leq rank_S(med(X)) \leq \frac{|S|}{2} + t$ w.p at least $1 - O(1/n^2)$.
Let two elements whose ranks in S are $\frac{|S|}{2} - t$, $\frac{|S|}{2} + t$ be s_{lr}, s_{hr} respectively. With at most $2n$ time and expected $1.5n$ time, we partition X into three subsets: X_l: less than s_{lr}, X_h: more than s_{hr}, and X_b: between s_{lr} and s_{hr} (For each element in X, we randomly choose which of s_{lr}, s_{hr} to compare first).
For any rank-αn element in X, its rank in S is $\alpha|S| \pm \sqrt{|S| \ln n}$ w.h.p. So choose α such that $\alpha|S| + \sqrt{|S| \ln n} = |S|/2 - \sqrt{|S| \ln n} \Rightarrow \alpha = .5 - 2\sqrt{\frac{\ln n}{|S|}}$, then the $(\alpha n) - th$-ranked element in X

is in X_l w.h.p. Similarly, for $\alpha' = .5 + 2\sqrt{\frac{\ln n}{|S|}}$, the $(\alpha' n) - th$-ranked element in X is in X_l w.h.p. So X_b has at most $\frac{4n\sqrt{\ln n}}{\sqrt{|S|}}$ elements w.h.p. Choose p is constant, so $|X_b| = \Theta(\sqrt{\ln n})$ whp. We can figure out the median by sorting X_b since we know the size of X_l and X_h.

References