Algorithms M2 IF
Even More Randomized Algorithms

Michael Lampis

Fall 2019

The power of sampling

Finding the Median

Consider the following basic problem:

- We are given an array A of n integers.
- We are asked to calculate the **Median** of A.
- Reminder: the median is the element of the array that is larger than at most $\frac{n}{2}$ elements, and smaller than at most $\frac{n}{2}$ elements.

Easy algorithms?

Finding the Median

Consider the following basic problem:

- We are given an array A of n integers.
- We are asked to calculate the **Median** of A.
- Reminder: the median is the element of the array that is larger than at most $\frac{n}{2}$ elements, and smaller than at most $\frac{n}{2}$ elements.

Easy algorithms?

- For each $i \in \{1, \ldots, n\}$ check if $A[i]$ is a median element.
  - Complexity: $O(n^2)$. :-(

Finding the Median

Consider the following basic problem:

- We are given an array A of n integers.
- We are asked to calculate the **Median** of A.
- Reminder: the median is the element of the array that is larger than at most $\frac{n}{2}$ elements, and smaller than at most $\frac{n}{2}$ elements.

Easy algorithms?

- For each $i \in \{1, \ldots, n\}$ check if $A[i]$ is a median element.
  - Complexity: $O(n^2)$. :-(
- Sort the array, return $A[n/2]$
  - Complexity: $O(n \log n)$.

Anything better than $O(n \log n)$?

A sampling-based algorithm for Median

- Select a random subset R of A with size $n^{3/4}$.
  - Repeat $n^{3/4}$ times: select a random element of A.
  - How to produce such an element? Flip $\log n$ fair coins, result gives index in binary.
  - Note: same element could be selected twice!
- Sort R
  - Takes time $O(n^{3/4} \log n) = o(n)$.
- Find elements d, u in R
  - d is $\sqrt{n}$ positions before the middle of R
  - u is $\sqrt{n}$ positions after the middle of R

A sampling-based algorithm for Median

- Select a random subset R of A with size $n^{3/4}$.
  - Repeat $n^{3/4}$ times: select a random element of A.
  - How to produce such an element? Flip $\log n$ fair coins, result gives index in binary.
  - Note: same element could be selected twice!
- Sort R
  - Takes time $O(n^{3/4} \log n) = o(n)$.
- Find elements d, u in R
  - d is $\sqrt{n}$ positions before the middle of R
  - u is $\sqrt{n}$ positions after the middle of R

Main intuition of the algorithm:

- We hope that if we sort A, the true median would be between d and u.

Sampling for the median continued

- Reminder: A has size n, R is a random subset of size $n^{3/4}$, $d, u \in R$
- Let
  - A_1 be the elements of A which are $< d$
  - A_2 be the elements of A which are between d and u
  - A_3 be the elements of A which are $> u$
- These sets can be found in $O(n)$ time.
- We FAIL if
  - $|A_1| > n/2$ or $|A_3| > n/2$
  - $|A_2| > 4n^{3/4}$

Sampling for the median continued

- Reminder: A has size n, R is a random subset of size $n^{3/4}$, $d, u \in R$
- Let
  - A_1 be the elements of A which are $< d$
  - A_2 be the elements of A which are between d and u
  - A_3 be the elements of A which are $> u$
- These sets can be found in $O(n)$ time.
- We FAIL if
  - $|A_1| > n/2$ or $|A_3| > n/2$
  - $|A_2| > 4n^{3/4}$
- In first case, A_2 does **not** contain the median
- In the second, A_2 is too large

Median continued

- Reminder: if we haven't failed so far A_2 has size at most $4n^{3/4}$ AND contains the median
- Also, we know $|A_1|$, which is $< n/2$.
- To finish things off, sort A_2
  - Time $O(n^{3/4} \log n)$, which is $o(n)$.
- Let i be such that $i + |A_1| = n/2$.
- Return $A_2[i]$.

Median continued

- Reminder: if we haven't failed so far A_2 has size at most $4n^{3/4}$ AND contains the median
- Also, we know $|A_1|$, which is $< n/2$.
- To finish things off, sort A_2
  - Time $O(n^{3/4} \log n)$, which is $o(n)$.
- Let i be such that $i + |A_1| = n/2$.
- Return $A_2[i]$.
- If we've come this far
  - The algorithm ran in time $O(n)$.
  - We output the correct median.
  - Need to calculate probability of reaching this point.

Probability of failure

The algorithm fails if any of the following happen:

- A_1 is too big $(> n/2)$
- A_3 is too big $(> n/2)$
- A_2 is too big $(> 4n^{3/4})$

Probability of failure

The algorithm fails if any of the following happen:

- A_1 is too big $(> n/2)$
- A_3 is too big $(> n/2)$
- A_2 is too big $(> 4n^{3/4})$

- The first two are symmetric.
- A_1 too big $\leftrightarrow d > m$, where m is the median.

Probability of failure

The algorithm fails if any of the following happen:

- A_1 is too big $(> n/2)$
- A_3 is too big $(> n/2)$
- A_2 is too big $(> 4n^{3/4})$

- The first two are symmetric.
- A_1 too big $\leftrightarrow d > m$, where m is the median.
- $d > m \leftrightarrow$ there exist $\frac{|R|}{2} + \sqrt{n}$ elements of R which are $> m$

Probability of failure

The algorithm fails if any of the following happen:

- A_1 is too big $(> n/2)$
- A_3 is too big $(> n/2)$
- A_2 is too big $(> 4n^{3/4})$

- The first two are symmetric.
- A_1 too big $\leftrightarrow d > m$, where m is the median.
- $d > m \leftrightarrow$ there exist $\frac{|R|}{2} + \sqrt{n}$ elements of R which are $> m$
- Each element of R is $> m$ with probability at most $1/2$
- If S_1 is number of elements of R which are $> m$ then $E[S_1] \leq \frac{|R|}{2}$.

Probability of failure

The algorithm fails if any of the following happen:

- A_1 is too big $(> n/2)$
- A_3 is too big $(> n/2)$
- A_2 is too big $(> 4n^{3/4})$

- The first two are symmetric.
- A_1 too big $\leftrightarrow d > m$, where m is the median.
- $d > m \leftrightarrow$ there exist $\frac{|R|}{2} + \sqrt{n}$ elements of R which are $> m$
- Each element of R is $> m$ with probability at most $1/2$
- If S_1 is number of elements of R which are $> m$ then $E[S_1] \leq \frac{|R|}{2}$.
- What is $Pr[S_1 > \frac{|R|}{2} + \sqrt{n}]$?
  - $Pr[S_1 > \frac{|R|}{2} + \sqrt{n}] \leq Pr[|S_1 - E[S_1]| > \sqrt{n}]$
  - Should use Chebyshev!

Probability of failure

WE are looking at the following failure possibility:

- A_1 is too big $(> n/2)$
- We argued it suffices to bound $Pr[|S_1 - E[S_1]| > \sqrt{n}]$
- S_1 is number of elements of R which are $> m$

Probability of failure

WE are looking at the following failure possibility:

- A_1 is too big $(> n/2)$
- We argued it suffices to bound $Pr[|S_1 - E[S_1]| > \sqrt{n}]$
- S_1 is number of elements of R which are $> m$

- Let X_i be the event that the i-th sample (element of R) is $> m$.
- $S_1 = \sum_{i=1}^{n^{3/4}} X_i$ and the X_i are independent.
- $Var[S_1] = n^{3/4} Var[X_i] \leq \frac{n^{3/4}}{4}$.

Probability of failure

WE are looking at the following failure possibility:

- A_1 is too big $(> n/2)$
- We argued it suffices to bound $Pr[|S_1 - E[S_1]| > \sqrt{n}]$
- S_1 is number of elements of R which are $> m$

- Let X_i be the event that the i-th sample (element of R) is $> m$.
- $S_1 = \sum_{i=1}^{n^{3/4}} X_i$ and the X_i are independent.
- $Var[S_1] = n^{3/4} Var[X_i] \leq \frac{n^{3/4}}{4}$.

- Using Chebyshev's inequality we get

$$Pr[|S_1 - E[S_1]| > \sqrt{n}] < \frac{1}{4n^{1/4}} = o(1)$$

- Second type of failure is handled in the same way

Last type of failure

- A_2 has size $> 4n^{3/4}$.
- Reminder: A_2 has the elements of A which are between d and u.

Last type of failure

- A_2 has size $> 4n^{3/4}$.
- Reminder: A_2 has the elements of A which are between d and u.
- Either A_2 has $> 2n^{3/4}$ elements which are $> m$
- or it has $> 2n^{3/4}$ elements which are $< m$
- Cases are symmetric, so we handle the first.

Last type of failure

- A_2 has size $> 4n^{3/4}$.
- Reminder: A_2 has the elements of A which are between d and u.
- Either A_2 has $> 2n^{3/4}$ elements which are $> m$
- or it has $> 2n^{3/4}$ elements which are $< m$
- Cases are symmetric, so we handle the first.
- If A_2 has $> 2n^{3/4}$ elements $> m$,
  - then u in the sorted array A is in position at least $n/2 + 2n^{3/4}$
  - But R has $\frac{|R|}{2} - \sqrt{n}$ elements bigger than u
  - So R has $\frac{|R|}{2} - \sqrt{n}$ among the last $n/2 - 2n^{3/4}$ elements of A

Last type of failure

- A_2 has size $> 4n^{3/4}$.
- Reminder: A_2 has the elements of A which are between d and u.
- Either A_2 has $> 2n^{3/4}$ elements which are $> m$
- or it has $> 2n^{3/4}$ elements which are $< m$
- Cases are symmetric, so we handle the first.
- If A_2 has $> 2n^{3/4}$ elements $> m$,
  - then u in the sorted array A is in position at least $n/2 + 2n^{3/4}$
  - But R has $\frac{|R|}{2} - \sqrt{n}$ elements bigger than u
  - So R has $\frac{|R|}{2} - \sqrt{n}$ among the last $n/2 - 2n^{3/4}$ elements of A
  - So what?

Last type of failure

- A_2 has size $> 4n^{3/4}$.
- Reminder: A_2 has the elements of A which are between d and u.
- Either A_2 has $> 2n^{3/4}$ elements which are $> m$
- or it has $> 2n^{3/4}$ elements which are $< m$
- Cases are symmetric, so we handle the first.
- If A_2 has $> 2n^{3/4}$ elements $> m$,
  - then u in the sorted array A is in position at least $n/2 + 2n^{3/4}$
  - But R has $\frac{|R|}{2} - \sqrt{n}$ elements bigger than u
  - So R has $\frac{|R|}{2} - \sqrt{n}$ among the last $n/2 - 2n^{3/4}$ elements of A
  - So what?
- The expected number of elements of R from the $n/2 - 2n^{3/4}$ largest elements of A is

$$E[Z] = n^{3/4}\left(\frac{1}{2} - \frac{2}{n^{1/4}}\right) = \frac{|R|}{2} - 2\sqrt{n}$$

Last type of failure

- Reminder: we consider the case $|A_2| > 4n^{3/4}$
- In particular, A_2 has $> 2n^{3/4}$ elements $> m$
- If A' is the set of $n/2 - 2n^{3/4}$ largest elements of A we argued that if Z is the expected number of such elements in R we have
  - $E[Z] = \frac{|R|}{2} - 2\sqrt{n}$
  - $Z > \frac{|R|}{2} - \sqrt{n}$

Last type of failure

- Reminder: we consider the case $|A_2| > 4n^{3/4}$
- In particular, A_2 has $> 2n^{3/4}$ elements $> m$
- If A' is the set of $n/2 - 2n^{3/4}$ largest elements of A we argued that if Z is the expected number of such elements in R we have
  - $E[Z] = \frac{|R|}{2} - 2\sqrt{n}$
  - $Z > \frac{|R|}{2} - \sqrt{n}$

- So if we have a problem then $Z - E[Z] > \sqrt{n}$.
- Let's show this is unlikely
- Recall $Z = \sum_{i=1}^{n^{3/4}} X_i$, and X_i is 1 with probability $\frac{1}{2} - \frac{2}{n^{1/4}}$
- $Var[Z] = n^{3/4}\left(\frac{1}{2} - \frac{2}{n^{1/4}}\right)\left(\frac{1}{2} + \frac{2}{n^{1/4}}\right) < n^{3/4}$
- Chebyshev: $Pr[Z - E[Z] > \sqrt{n}] < \frac{1}{n^{1/4}} = o(n)$

Putting everything together

- Sampled a **sub-linear** number of elements, still enough to get a good representation of A
- Three types of failure. Probability of at least one $\leq$ sum of their probabilities
  - Union bound!
- Used variance to show that each failure type has $o(1)$ probability.
- Here, all events were independent
  - Pair-wise independent would have been enough!

Approximate Sampling

Approximate Median

- We saw an $O(n)$ algorithm to find the median.
- Clearly impossible to do better. (Why??)
- Consider an approximate version of the problem:
- A has n elements
  - A_1 is the set of 45% smallest elements
  - A_3 is the set of 45% largest elements
  - A_2 is the rest (10% in the middle)
- Question: Output any item of A_2
- Such an item is "close" to a median
- Problem can be adjusted (decreasing size of A_2)
- Obviously, previous algorithm works.
- Objective to do $o(n)$
- In fact, possible to do $O(1)$!!!
  - More precisely, running time depends on acceptable size of A_2 and failure probability.

A simple algorithm for approximate median

- Form a random subset R of size s by sampling A
- Sort R
- Output the median of R

A simple algorithm for approximate median

- Form a random subset R of size s by sampling A
- Sort R
- Output the median of R

- Algorithm runs in $O(s \log s)$ (provided we have random access to A)
- Question is success probability (as a function of s)

A simple algorithm for approximate median

- Form a random subset R of size s by sampling A
- Sort R
- Output the median of R

- Algorithm runs in $O(s \log s)$ (provided we have random access to A)
- Question is success probability (as a function of s)
- Let A'_1, A'_2, A'_3 be the elements of R which come from A_1, A_2, A_3 respectively.

$$\begin{aligned} E[A'_1] &= s\frac{|A_1|}{n} = 0.45s \\ E[A'_2] &= s\frac{|A_2|}{n} = 0.10s \\ E[A'_3] &= s\frac{|A_3|}{n} = 0.45s \end{aligned}$$

We want to be close to these values.

Chebyshev again

- $Var[A'_1] = s\frac{|A_1|}{n}(1 - \frac{|A_1|}{n}) \leq s/4$
- $Var[A'_2] = s\frac{|A_2|}{n}(1 - \frac{|A_2|}{n}) \leq s/4$
- $Var[A'_3] = s\frac{|A_3|}{n}(1 - \frac{|A_3|}{n}) \leq s/4$

Chebyshev again

- $Var[A'_1] = s\frac{|A_1|}{n}(1 - \frac{|A_1|}{n}) \leq s/4$
- $Var[A'_2] = s\frac{|A_2|}{n}(1 - \frac{|A_2|}{n}) \leq s/4$
- $Var[A'_3] = s\frac{|A_3|}{n}(1 - \frac{|A_3|}{n}) \leq s/4$

$$
\begin{aligned}
Pr[|A'_1 - E[A'_1]| > s^{3/4}] &< \frac{1}{\sqrt{s}} \\
Pr[|A'_2 - E[A'_2]| > s^{3/4}] &< \frac{1}{\sqrt{s}} \\
Pr[|A'_3 - E[A'_3]| > s^{3/4}] &< \frac{1}{\sqrt{s}}
\end{aligned}
$$

Chebyshev again

- $Var[A'_1] = s\frac{|A_1|}{n}(1 - \frac{|A_1|}{n}) \leq s/4$
- $Var[A'_2] = s\frac{|A_2|}{n}(1 - \frac{|A_2|}{n}) \leq s/4$
- $Var[A'_3] = s\frac{|A_3|}{n}(1 - \frac{|A_3|}{n}) \leq s/4$

$$
\begin{aligned}
Pr[|A'_1 - E[A'_1]| > s^{3/4}] &< \frac{1}{\sqrt{s}} \\
Pr[|A'_2 - E[A'_2]| > s^{3/4}] &< \frac{1}{\sqrt{s}} \\
Pr[|A'_3 - E[A'_3]| > s^{3/4}] &< \frac{1}{\sqrt{s}}
\end{aligned}
$$

- All we need to do is to increase s to a sufficiently large constant (independent of n!) and we get that the algorithm is correct with high probability.