CS 224: Advanced Algorithms Spring 2017

Lecture 4 — February 2nd, 2017

Prof. Jelani Nelson *Scribe: Rohil Prasad*

1 Overview

In the last lecture we covered topics in hashing, including load balancing, k-wise independence, and a couple of approaches to the dynamic dictionary problem.

In this lecture we will cover linear probing analysis, approximate membership (bloom filters), and cuckoo hashing and bloomier filters.

2 Notes about the Chernoff bound

Recall the Chernoff bound:

Theorem 1. *For independent random variables $X_1, \ldots, X_n \in [0,1]$ with $E[\sum X_i] = \mu$, the following bound holds:*

$$\mathbb{P}(\sum X_i = \mu > \delta\mu) < (\frac{e^\delta}{(1+\delta)^{1+\delta}})^\mu$$

There are two **regimes of importance** in the Chernoff bound.

The first is when $\delta << 1$. We rewrite the bound as

$$(e^{\frac{e^\delta}{(1+\delta)\log(1+\delta)}})^\mu$$

By Taylor expansion, $\log(1+\delta) = \delta - \delta^2/2 + O(\delta^3)$. Substituting this into the expression above, we obtain a bound

$$e^{-(\frac{\delta^2}{2}+O(\delta^3))\mu}$$

The second regime is when $\delta >> 1$. Then we can calculate the bound to be approximately $\delta^{-\delta\mu}$.

3 Linear probing analysis

Recall the setup of linear probing. We set a hash function $h : [u] \to [m]$, where u is the size of our universe. However, if we have some $x \in [u]$ such that $h(x)$ is already filled, then we walk to the right in $[m]$ until we find a spot that is not filled.

Claim 2. *If $m \geq 2n$, then*

$$\mathbb{E}[\textit{time per operation}] = O(1)$$

over hash functions h drawn from a 5-wise independent family ([1]).

Definition 3. *For I an interval in the array, define*

$$L(I) = |\{x \in S \,:\, h(x) \in I\}|$$

*We say I is **full** if $L(I) \geq |I|$.*

Lemma 4. *If we spend k units of time querying/inserting a key x, then there are at least k full intervals of different lengths containing $h(x)$.*

Proof. Since it takes time k to insert a key x, then we have k elements to the right of $h(x)$ in $[m]$ are occupied.

Then we walk back to the highest empty spot before $h(x)$. In total, we have several elements to the left and right of $h(x)$. By construction, we know every element in this big interval hashed at the leftmost spot in the interval or afterwards.

Since this interval has length $\geq k$, we can produce our desired k full intervals by taking the leftmost parts of the intervals to be the leftmost part of the big interval and the rightmost parts of the intervals to be $h(x), h(x)+1, \ldots, h(x)+k-1$. □

Now let

$$E_k = \begin{cases} 1, \text{ if some length } k \text{ full interval contains } h(x). \\ 0, \text{otherwise.} \end{cases}$$

Then the time per operation is bounded above by $\sum_{k=1}^{n} E_k$. Therefore, we can calculate

$$\begin{aligned} \mathbb{E}(\text{time per op.}) &\leq \sum_{k=1}^{n} \mathbb{P}(\text{some length } k \text{ interval containing } h(x) \text{ is full}) \\ &\leq \sum_{k=1}^{n} k \cdot \mathbb{P}(\text{a particular length } k \text{ interval containing } h(x) \text{ is full}) \end{aligned}$$

Now we can calculate this probability. Fix an interval I of length k. Define random variables

$$X_i = \begin{cases} 1, \text{ if } h(i) \in I. \\ 0, \text{ otherwise.} \end{cases}$$

Then $L(I) = \sum_{i=1}^{n} X_i$ so

$$\mathbb{E}\, L(I) = \sum_{i=1}^{n} \mathbb{E}\, X_i = n \cdot \frac{k}{m} < \frac{k}{2}$$

Finally, by Chernoff we can deduce

$$\mathbb{P}(I \text{ is full}) < \mathbb{P}(L(I) - \mathbb{E}\, L(I) \geq \frac{k}{2}) = e^{-\Omega(k)}$$

and plug that into our expression above to get

$$\mathbb{E}(\text{time per op.}) \leq \sum_{k} k \cdot e^{-\Omega(k)} = O(1)$$

Now we will show that 7-wise independence will suffice, rather than the independence required by Chernoff. In particular, this will change our $e^{-\Omega(k)}$ to $O(k^{-3})$.

3.1 7-wise analysis

We will attack this by examining the moments of our variable $L(I)$. By Markov's inequality, we have

$$\mathbb{P}(|\sum X_i - \mathbb{E}\sum X_i|^p > \lambda^p) < \lambda^{-p} \cdot \mathbb{E}(|\sum X_i - \mathbb{E}\sum X_i|^p)$$

To analyze this further, we need a tiny bit of math.

Definition 5. *If Z is a random variable and $p \geq 1$, then*

$$||Z||_p = (\mathbb{E}\,|Z|^p)^{\frac{1}{p}}$$

.

Theorem 6. *(Minkowski) The function $||\cdot||_p$ is a norm.*

Theorem 7. *(Jensen) If f is convex, then $f(\mathbb{E}\,Z) \leq \mathbb{E}\,f(Z)$.*

In particular, we will analyze the sixth moment (set $p = 6$ in the expression above). Instead of brute forcing, we will use a computational trick called **symmetrization**.

Pretend that the X_i's are fully independent. Let $\sigma_1, \ldots, \sigma_n$ be independent random variables taking on the values ± 1 with equal probability.

Also consider random variables $Y_1, \ldots, Y_n$ that are distributed as $X_1, \ldots, X_n$ but are independent of them. Then we can derive:

$$\begin{aligned}
||\sum_i X_i - \mathop{\mathbb{E}}_{Y}\sum_i Y_i||_6 &= (\mathop{\mathbb{E}}_{X}(|\mathop{\mathbb{E}}_{Y}[\sum X_i - \sum Y_i]|^6))^{1/6} \\
&\leq ||\sum_i (X_i - Y_i)||_6 \quad \text{(Jensen)} \\
&= ||\sum_i \sigma_i(X_i - Y_i)||_6 \\
&\leq 2 \cdot ||\sum_i \sigma_i X_i||_6 \quad \text{(Triangle inequality)}
\end{aligned}$$

We expand out

$$\mathbb{E}[(\sum_{i=1}^{n} \sigma_i X_i)^6] = \sum_{i_1,\ldots,i_6} \mathbb{E}[X_{i_1} X_{i_2} \ldots X_{i_6}] \cdot \mathbb{E}[\sigma_{i_1} \ldots \sigma_{i_6}]$$

Note that $\mathbb{E}[\sigma_{i_1} \ldots \sigma_{i_6}]$ is equal to 0 if it contains any odd powers (say, $\sigma_1^3\sigma_2^2\sigma_3$) and 1 if it contains all even powers.

Each X_i has $\mathbb{E}[X_i] = q = \frac{k}{m}$, which gives us an upper bound of $O(n^3 \cdot q^3) = O(k^3 \cdot (n^3/m^3)) = O(k^3)$ on the above expression (use casework on the index set $i_1, i_2, \ldots, i_6$).

To tie it all together, by Markov on the sixth moment we have

$$\mathbb{P}(L(I) - \mathbb{E}\, L(I) > \frac{k}{2}) \leq (\frac{k}{2})^{-6} \cdot \mathbb{E}[|L(I) - \mathbb{E}\, L(I)|^6] = O(k^{-6}) \cdot O(k^3) = O(k^{-3})$$

3.2 5-wise analysis

This is the real deal.

The improved analysis here comes from impoving on the union bound we used earlier to derive the inequality.

Then we only need to get an $O(k^{-2})$ bound on the probability that an interval containing $h(x)$ is full, which will only require a fourth-moment analysis, which will only require 5-wise independense.

We construct a perfect binary tree on the set of our hashes. The leaf nodes all "cover" one hash, while the nodes i levels above the leaves cover 2^i hashes.

There are k intervals of length k containing $h(x)$. Pick one of them and call it I. Then $|I| \in [2^h, 2^{h+1})$ (h is just $\log k$, not the hash function).

This implies that there are between 8 and 17 nodes at level $h - 3$ of our tree that cover I.

However, all length k intervals containing $h(x)$ are themselves in a length $(2k-1)$-interval. Therefore, the union of all the nodes spanning all these k intervals has size in $[16, 33)$, which is $O(1)$.

We can actually then just union bound over these $O(1)$ nodes instead of the k original bounds.

This follows from observing that if some length k full interval contains $h(x)$, then we will be able to find at least one of these $O(1)$ intervals are full.

Then a fourth-moment analysis gives us

$$\mathbb{E}(\text{time per op}) \leq \sum_{k=1}^{n} O(1) \cdot O(\frac{1}{k^2}) = O(1)$$

4 Approximate membership

This is a **data-structural problem**, so it can be either static or dynamic.

We want to maintain a subset $S \subseteq [u]$ subject to the following operations:

- `query`(x): Is $x \in S$? (If $x \in S$, say yes. If $x \notin S$, $\mathbb{P}(\text{yes}) < \varepsilon$.)
- `insert`(x): Add x to S.

4.1 Bloom filters

One data structure we can use to attack the above problem is the **Bloom filter**. A Bloom filter consists of a $m = 2n$-bit array and a two-wise independent hash $h : [u] \to [m]$.

The `insert` operation is done by hashing x.

To do `query`, we check if $x \in S$. Set Y_j to be 1 if $h(j) = h(x)$ and 0 otherwise and say yes if some Y_j is equal to 1. Then

$$\begin{aligned}\mathbb{P}(\text{yes}) &= \mathbb{P}(\text{some } Y_j = 1) \\ &\leq \sum_{j \in S} \mathbb{P}(h(j) = h(x))\end{aligned}$$

Therefore, if $x \in S$ the probability above is 1, while if $x \notin S$ the probability above is by two-wise independence $\frac{n}{m}$ which is less than $1/2$.

To improve on this, pick $\varepsilon > 0$. Then set $t = \log \varepsilon^{-1}$ and use a famly of t hash functions and an array of size $m \times t$.

Insertion is done in the same way. Querying is done in the same way as in the one case, except now we query one hash function after another. This gives us a failure probability of $2^{-\log \varepsilon^{-1}} = \varepsilon$, takes time $\log \varepsilon^{-1}$ and space $2n \log \varepsilon^{-1}$.

References

[1] Anna Pagh, Rasmus Pagh, and Milan Ružić. Linear Probing with 5-wise independence. *SIAM Review* 53.3 (2011):547-558, 2011.