Lecture 16

1 Load balancing, Cuckoo hashing

Consider the problem of placing m balls in n bins with the objective of minimizing the maximum load in a bin. The abstract setting of balls and bins models several concrete problems like distributing jobs over machines and hashing items to avoid collisions.

1.1 Randomized load balancing

The simplest randomized strategy for load balancing is to assign each ball to a randomly chosen bin. For simplicity, we analyze the case where the number of balls and bins is equal to n. At several points in the analysis we will use the following approximation for binomial coefficients,

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \frac{n^k}{k!} \leq \left(\frac{ne}{k}\right)^k \tag{1}$$

Let X_i denote the number of balls in the i-th bin at the end of the process. There are $\binom{n}{k}$ sets of k elements, and the probability that all the k elements in a set are assigned to bin i is $(1/n)^k$. Applying the union bound we can bound the probability that bin i contains at least k balls,

$$
\begin{aligned}
\Pr[X_i \geq k] &\leq \binom{n}{k}\left(\frac{1}{n}\right)^k \\
&\leq \frac{n^k}{k!}\left(\frac{1}{n}\right)^k = \frac{1}{k!}
\end{aligned} \tag{2}
$$

If k is chosen so that $\Pr[X_i \geq k] \leq 1/n^2$, then applying the union bound we have $\Pr[\exists i \text{ s.t. } X_i \geq k] \leq 1/n$. The probability that the maximum load in the bins exceeds k is at most $\frac{1}{n}$.

A suitable value of k satisfies $2\log n \leq \log k! \approx k \log k - k$, clearly choosing $k = O(\log n)$ suffices, in fact we could have chosen $k = O(\log n/\log\log n)$. The bounds is stated for $k = O(\log n)$ for simplicity,

$$\Pr[\forall i \ \ X_i \leq c\log n] \geq 1 - \frac{1}{\text{poly}(n)} \tag{3}$$

The probability of some bin having load more than $O(\log n)$ is $1/\text{poly}(n)$ for the simple randomized load balancing strategy.

1.2 Power of two choices:

The two choice strategy chooses two bins uniformly at random for every ball and places the ball into the bin with the smaller load. The maximum load for the two choice strategy is $O(\log \log n)$ with probability $1 - \frac{1}{\text{poly}(n)}$, an exponential improvement over the simple randomized strategy.

We analyze the two choice strategy with $n/8$ balls and n bins, the constant 8 is chosen because 8 is the smallest integer greater than e^2. The process is modeled by a graph with bins as vertices, and edges between each pair of bins that was chosen by a ball during the process. All vertices start with a count of zero, a uniformly random edge is chosen in each step and the end point with the lower value of count gets incremented. The maximum load for the two choice strategy is equal to the maximum count over vertex in the graph at the end of the process.

An example to gain intuition for the process is a tree of depth k where the nodes at depth i are connected to $k - i$ nodes at depth $i + 1$. The number of nodes in the tree is $2k!$. Suppose the process introduces the edges of the tree in a bottom up manner, with edges at higher depth being added first. In the worst case, the two choice process can assign $k - i$ balls to a node at depth i. The root of the tree is assigned k balls, the example shows that if we have a connected component of size $2k!$ it is possible to have a maximum load of k.

The analysis of the two choice strategy depends on two observations: (i) The size of the largest connected component is $O(\log n)$ with high probability. (ii) For a connected component of size k the maximum count can be at most $O(\log k)$, combining with part (i) we have the $O(\log \log n)$ bound on the maximum load.

CLAIM 1
The size of the largest connected component in a random graph with n vertices and $n/8$ edges is $O(\log n)$ with probability at least $1 - \frac{1}{poly(n)}$.

PROOF: Suppose the size of the largest connected component C in G is at least k. The induced subgraph for some subset of C of size k has at least $k - 1$ edges incident on it. By the union bound, the probability that there is a connected component of size k can be bounded by,

$$\Pr[|C| \geq k] \leq \binom{n}{k}\binom{n/8}{k-1}\left(\frac{k}{n}\right)^{2(k-1)} \tag{4}$$

The first term chooses the a subset of k vertices, the second chooses the $k - 1$ incident edges while the third term is the probability that both the end points of the edges lie in the subset. It is convenient to treat $k - 1$ as k as a difference of 1 does not change the asymptotic calculation.

$$
\begin{aligned}
\Pr[|C| \geq k] &\leq \binom{n}{k}\binom{n/8}{k}\left(\frac{k}{n}\right)^{2k} \\
&\leq \left(\frac{ne}{k}\right)^k \left(\frac{ne}{8k}\right)^k \left(\frac{k}{n}\right)^{2k} = \left(\frac{e^2}{8}\right)^k \leq (0.93)^k \tag{5}
\end{aligned}
$$

Choosing $k = O(\log n)$ the probability of the size of the largest connected component being larger than k is at most $\frac{1}{\text{poly}(n)}$. $\square$

Bounds on the induced degree: The induced degree of $S \subset V$ is the number of edges with both the end points in S. We show that for all $S \subset V$ the induced degree is S is at most $8|S|$ with probability $1 - 1/n^2$. The probability that some subset with $|S| = k$ has induced degree more than $8k$ can be bounded by,

$$\Pr[|S| = k, deg(S) \geq 8k] \leq \binom{n}{k}\binom{n/8}{4k}\left(\frac{k}{n}\right)^{8k} \leq \left(\frac{e^{1.25}}{32}\right)^{4k}\left(\frac{k}{n}\right)^{3k} \leq \left(\frac{k}{n}\right)^{3k} \quad (6)$$

The first term counts the number of ways of choosing a subset S of size k, if the induced degree is more than 8 then at least $4k$ edges must be incident on S hence the second term, while the third term is the probability that both the end points of the $4k$ edges lie in S. The inequality follows using the approximation (1) for the binomial coefficients.

The union bound is used to show that with probability $1 - \frac{1}{n^2}$ the induced degree for all $S \subset V$ is at most $8|S|$. A subset with induced degree more than $8|S|$ can have size at most $n/32$ as there are $n/8$ edges.

$$\Pr[\exists S \text{ s.t. } deg(S) \geq 8k] \leq \sum_{k \in [n/32]} \binom{n}{k}\left(\frac{k}{n}\right)^{3k} = O\left(\frac{1}{n^2}\right) \quad (7)$$

The maximum asymptotic contribution to the above sum comes from the term corresponding to $k = 1$.

Conditioning: The rest of the argument is conditioned on the largest connected component in G having size $\Theta(\log n)$ and the induced degree of all the subsets of G being at most 8, both these events happen with probability at least $1 - \frac{1}{n^2}$ from claim 1 and the preceding argument.

An iterative decomposition of G: The analysis of maximum count depends on an iterative decomposition of G, let A_1 be the set of vertices in G that have degree at most 16, in general define A_i to be the set of vertices that have degree at most 16 in the induced subgraph with vertex set $G \setminus \cup_{j \in [i-1]} A_j$.

The average degree of every induced subgraph of G is at most 8, at most half the nodes can have more than twice the average degree, so each round of the iterative decomposition reduces the number of nodes in a connected component by at least half. The number of nodes in a connected component is $O(\log n)$ so the total number of iterations is $O(\log \log n)$.

The following claim completes the analysis of the two choice strategy:

CLAIM 2
The maximum count for a vertex is at most $16t + 1$ where t is the number of the iteration in which it was removed from the graph.

PROOF: The proof is by induction on the number of iterations, the base case is true as for vertices in A_1 have degree at most 16 and the count for a vertex does not exceed the degree. The count for a vertex $v \in A_i$ does not increase by more than 16 when the edges in A_i are removed, so it suffices to show that the count for v before iteration i begins is at most $16(t-1) + 1$.

If v has count more than $16(t-1) + 1$ consider the edge (v, w) was used when the count of v was last incremented. The count of w must be at least $16(t-1) + 1$ otherwise the count

of w would have been incremented instead of v. The vertex $w \in \cup_{j \in [i-1]} A_j$ and has a count more than $16(t-1)+1$ contradicting the inductive hypothesis. $\square$

Remark: The proofs would go through if instead of choosing bins uniformly at random, each ball chose $h(x), h(y)$ where h is drawn from a family of $O(\log n)$ universal hash functions, as we need independence over $k = O(\log n)$ distinct points. A hash function family is t universal if the knowledge of $h(x)$ at $t-1$ distinct points reveals nothing about the function, an example of a t wise independent hash function family is $p(x) \mod p \mod B$ where $p(x)$ is a polynomial of degree t.

1.3 Cuckoo Hashing

The two choice strategy can be applied to hashing, an element x is hashed to the bucket out of $h_1(x), h_2(x)$ having a smaller load, where h_1 and h_2 are independent hash functions. The length of the hash lists will be $O(\log \log n)$ if $O(n)$ elements are hashed. Is it possible to have $O(1)$ insertion and lookup times?

The cuckoo hashing strategy is the following: Insert x into the positions $h_1(x)$ or $h_2(x)$ if either of these positions are empty. If both the positions are occupied select $y = h_i(x)$ uniformly at random, replace y by x and run the insertion algorithm with input y. If the depth of the recursion exceeds a certain threshold without completing the insertion, declare failure and rebuild the table using a different pair of hash functions.

The cuckoo hash algorithm fails if there is a cycle in the corresponding random graph. For example consider the case of five elements where element i can hash to slot i or $i+1$ mod 5, and a new element that hashes to slots 1 and 4 needs to be inserted, the cuckoo hash algorithm runs into an infinite loop and the table needs to be rebuilt.

The probability of the existence of cycles can be bounded in a manner similar to the bound on the size of the connected component (5) in the analysis of the two choice strategy. For a cycle of length l we need to select $l+1$ out of $m = n/8$ edges, l out of n vertices and each of the $2(l+1)$ end points must be in a set of size l,

$$\Pr[C_l] \leq \binom{m}{l+1} \binom{n}{l} \left(\frac{l}{n}\right)^{2(l+1)} \leq \left(\frac{e^2}{8}\right)^l \tag{8}$$