CS265/CME309: Randomized Algorithms and Probabilistic Analysis
Lecture #6: Balls in Bins and Power-of-Two-Choices

Gregory Valiant*, updated by Mary Wootters

September 29, 2020

1 Introduction

Many phenomenon can be modeled via the process of tossing n balls, uniformly at random, into one of m different "bins", and then examining certain properties of the resulting allocation of balls to bins. For example, the *coupon collector* problem can be reformulated in this framework as the following question: *what is the minimum value of n, as a function of m, such that we expect there to be zero empty bins after tossing the n balls?* Similarly, nearly all questions regarding the construction or analysis of hashing schemes (including Bloom filters that you may have seen in other courses) can trivially be restated as a balls-in-bins problem.

Example 1. *In case you haven't formally seen it before, the "birthday paradox" is the question of how small n can be, as a function of the number of bins, m, such that we expect a pair of balls to end up in the same bin. The probability that the first two balls do not collide is $\frac{m-1}{m}$, similarly the probability that there are no collisions within the first three balls is $\frac{(m-1)(m-2)}{m^2}$, and in general the probability that none of the n balls collide is*

$$\frac{(m-1)(m-2)\dots(m-n+1)}{m^{n-1}}.$$

Hence if $m = 365$, and birthdays are randomly distributed over the days of the year, then working through the math we get that as long as we have $n \geq 23$ people, there is at least a 0.5 probability that a pair of people share a birthday.

In these lecture notes, we will mainly focus on analyzing the *maximum bin load* (i.e. the maximum number of balls that end up in any single bin, and will see the surprising "power-of-two-choices" load balancing approach, which significantly reduces the expected maximum bin load by giving each ball the power to choose a bin between two random options.

Proposition 2. *Consider tossing n balls into n bins. There is a constant c such that with high probability, the maximum load will be at most $c \log n / \log \log n$, for sufficiently large n.*

Proof. The high-level approach will be to show that for any $k > 3\log n/\log\log n$,

$$\Pr[\text{bin 1 has load exactly } k] = o(1/n^2),$$

and hence we can union bound of the n bins, and the $< n$ values of $k \in \{3\log n/\log\log n, 1 + 3\log n/\log\log n, \ldots, n\}$.

$$\Pr[\text{bin 1 has load exactly } k] \leq \binom{n}{k}(1/n)^k(1-1/n)^{n-k} \leq \frac{n^k}{k!}(1/n)^k \leq \frac{1}{k!}.$$

Using the fact that $k! \geq (k/e)^k$, if $k = c\frac{\log n}{\log\log n}$, for $c > e$, then

$$1/k! \leq \left(\frac{\log n}{\log\log n}\right)^{-c\log n/\log\log n} \leq e^{-c(\log\log n - \log\log\log n)\log n/\log\log n} = e^{-c\log n + c\log n\frac{\log\log\log n}{\log\log n}}.$$

Since the second term in the exponent is $o(\log n)$, this entire expression is at least $n^{-c+o(1)}$. As the load can be at most n, by doing a union bound over all $k \geq c\frac{\log n}{\log\log n}$ we get,

$$\Pr\left[\text{bin 1 has load} > k\right] \leq n^{-c+1+o(1)}.$$

Taking any $c > 3$ yields that even after a union bound over the n bins and $< n$ values of k between $c\frac{\log n}{\log\log n}$ and n, the probability that the maximum load is at least $c\frac{\log n}{\log\log n}$ is $o(1)$, as desired. □

Before proving that the guarantees of Proposition 2 are fairly tight, in the sense that we *do* expect the maximum load to be $\approx \log n/\log\log n$, we will take a brief detour and discuss the Poisson distribution.

2 The Poisson Distribution

Many of you have encountered the Poisson distribution before in other classes. The Poisson distribution parameterized by a non-negative value λ, is typically denoted $Poi(\lambda)$. Below are some of the crucial properties that are worth remembering:

1. For $X \leftarrow Poi(\lambda)$ and any integer $k \geq 0$, $\Pr[X = k] = \frac{e^{-\lambda}\lambda^k}{k!}$.
2. An alternate definition of the Poisson distribution is the limit, as $n \to \infty$, of the binomial distribution corresponding to n independent tosses of a coin that lands heads with probability λ/n.
3. Both of the above two definitions yield that for $X \leftarrow Poi(\lambda)$, $\mathbf{E}[X] = \mathbf{Var}[X] = \lambda$.
4. For independent random variables X and Y with $X \leftarrow Poi(\lambda_1)$ and $Y \leftarrow Poi(\lambda_2)$, the sum $X + Y$ is distributed according to $Poi(\lambda_1 + \lambda_2)$.

Poisson distributions satisfy quite strong tail bounds, which can be proved via their moment-generating function.

Fact 3. *Poisson distributions satisfy strong tail bounds: Letting $X \leftarrow Poi(\lambda)$, for any $c > 0$,*

$$\Pr[|X - \lambda| \geq c] \leq 2e^{-\frac{c^2}{2(c+\lambda)}}.$$

One of the most useful properties of Poisson distributions is summarized in the following theorem.

Theorem 1. *Suppose we draw $k \leftarrow Poi(n)$, and then toss k balls uniformly at random into m bins, then the number of balls in bin 1, bin 2, etc, are all independent, distributed according to $Poi(n/m)$.*

Proof. For clarity, we give the proof in the case when there are just $m = 2$ bins, though the proof of the more general statement is essentially identical. Letting X_1, X_2 denote the number of balls in each bin, we have that, for any integers i, j,

$$\Pr[X_1 = i, X_2 = j] = \Pr[k = i + j] \Pr[Binomial(i + j, 1/2) = i],$$

since this event can only occur if the total number of balls that were tossed is $k = i + j$, in which case the probability that bin 1 gets i of them is given by the binomial distribution, since the balls are tossed independently into one of the two bins. Simplifying, we get

$$\begin{aligned} \Pr[k = i + j] \Pr[Binomial(i + j, 1/2) = i] &= \frac{e^{-n} n^{i+j}}{(i+j)!} \frac{(i+j)!}{j!i!} \frac{1}{2^{i+j}} = \frac{e^{-n/2}(n/2)^i}{i!} \cdot \frac{e^{-n/2}(n/2)^j}{j!} \\ &= \Pr[Poi(n/2) = i] \cdot \Pr[Poi(n/2) = j], \end{aligned}$$

where $\Pr[Poi(\lambda) = i]$ denotes the probability that a Poisson random variable with expectation λ is equal to i. We now use this expression to calculate the probability that $X_1 = i$. Note that by the law of total probability,

$$\begin{aligned} \Pr[X_1 = i] &= \sum_{j=0}^{\infty} \Pr[X_1 = i, X_2 = j] = \sum_{j=0}^{\infty} \Pr[Poi(n/2) = i] \cdot \Pr[Poi(n/2) = j] \\ &= \Pr[Poi(n/2) = i] \cdot \sum_{j=0}^{\infty} \Pr[Poi(n/2) = j] = \Pr[Poi(n/2) = i]. \end{aligned}$$

Similarly, $\Pr[X_2 = j] = \Pr[Poi(n/2) = j]$, and hence the number of balls in each bin is distributed according to $Poi(n/2)$. We will now complete the proof by showing that the number of balls in the two bins is also independent. To do this, we calculate the conditional probability and show that it is equal to the marginal probability,

$$\Pr[X_1 = i | X_2 = j] = \frac{\Pr[X_1 = i, X_2 = j]}{\Pr[X_2 = j]} = \frac{\Pr[Poi(n/2) = i] \cdot \Pr[Poi(n/2) = j]}{\Pr[Poi(n/2) = j]} = \Pr[Poi(n/2) = i].$$

□

What does the above theorem really mean? If you flip exactly 100 coins, the number of heads and tails will be completely determined by each other (e.g. $heads = 100 - tails$). The above theorem says that if you pick $n \leftarrow Poi(100)$, then flip n coins, the number of heads and tails are *independent*, both distributed according to $Poi(50)$. The nice thing is that this holds even though $Poi(100)$ is very

tightly distributed about 100 (since the expectation is 100, and the standard deviation is 10, and the probability it deviates from its expectation by more than c standard deviations is inverse exponential in c, as given in Fact 3.)

The power of Theorem 1 is that if we toss a Poisson number of balls, we end up with independent bin loads. And, analyzing independent random variables is usually pretty easy. (We can also apply tools like Chernoff bounds to analyze properties of these independent random variables, and we would not be able to apply Chernoff bounds to sums of *dependent* random variables). Additionally, since the Poisson distribution is concentrated about its expectation, we can relate this "Poissonized" setting back to the setting where we toss a deterministic number of balls. We now give two examples of this general of technique.

Proposition 4. *(Coupon Collector) Assuming we get a uniformly random one of n distinct coupons each day, letting X denote the number of days until we see at least one of each coupon, we have that for any (possibly negative) constant c,*

$$\lim_{n\to\infty} \Pr[X \geq n\log n + cn] = 1 - e^{-e^{-c}}.$$

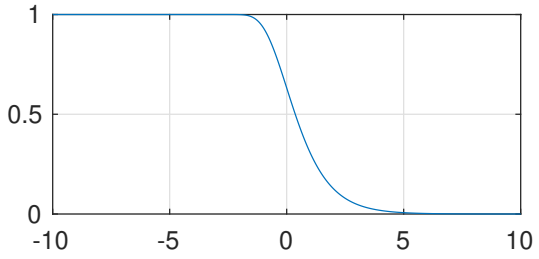

Figure 1: Plot of $1 - e^{-e^{-c}}$ as a function of c, illustrating that the time until all coupons are collected is sharply concentrated about $n \log n$.

Proof. Suppose the number of coupons we will consider is $k \leftarrow Poi(n\log n + cn)$, and note that the number of coupons of each type will be independent, distributed according to $Poi(c + \log n)$. Since $\Pr[Poi(\lambda) = 0] = e^{-\lambda}$, the probability we see at least one of each type of coupon is $(1 - e^{-(c+\log n)})^n = (1 - e^{-c}/n)^n$, which tends towards $e^{-e^{-c}}$ in the limit as $n \to \infty$ using the fact that $(1-x)^{1/x} \to e^{-1}$ as $x \to 0$.

To relate this to the non-Poissonized setting, we take the following 2 steps: 1) We argue that the probability of being done after N coupons can change by only by a *subconstant* amount if we give or take an extra $n^{0.9}$ random coupons, and 2) Since Poisson random variables are tightly concentrated about their expectation, $Poi(n\log n + cn)$ deviates from its expectation by $n^{0.9}$ with subconstant probability.

For the first part, note that $\Pr[X > n\log n + cn + n^{0.9}] \leq \Pr[X > n\log n + cn] \leq \Pr[X > n\log n + cn - n^{0.9}]$. How different can $\Pr[X > n\log n + cn + n^{0.9}]$ and $\Pr[X > n\log n + cn - n^{0.9}]$ actually be? The difference in probability is at most the probability that, after getting $n\log n + cn - n^{0.9}$ coupons, there is at least one coupon type we haven't seen, which we then receive in our next set of $2n^{0.9}$ coupons. However, even if there was just a single type of coupon that we haven't seen, the probability that this is in the next batch of $2n^{0.9}$ coupons is at most $2n^{0.9}/n = o(1)$. Hence

$\Pr[X > n\log n + cn + n^{0.9}] = \Pr[X > n\log n + cn - n^{0.9}] + o(1)$. To conclude, from the tail bounds of Fact 3,

$$\Pr[|k - (n\log n + cn)| \geq n^{0.9}] \leq 2e^{-n^{1.8}/O(n\log n)} = o(1).$$

[We could have also just used Chebyshev's inequality to get this $o(1)$ bounds, based on the variance of the Poisson distribution.] □

The proof of the following proposition, showing that Proposition 2 is tight, is a second example of this Poissonization technique.

Proposition 5. *Consider tossing n balls into n bins. With high probability, the maximum load will be at least $c\log n/\log\log n$.*

Proof. First, consider drawing $k \leftarrow Poi(n/2)$, and then tossing k balls. By Theorem 1, each bin load is independently drawn from $Poi(1/2)$. Hence, the probability that at least one of the n bins has load at least b is at least the probability that at least one has load exactly b, which is

$$1 - (1 - \frac{e^{-1/2}(1/2)^b}{b!})^n \geq 1 - e^{-n\frac{e^{-1/2}(1/2)^b}{b!}}.$$

Analyzing the exponent, by Stirling's approximation, $\log(b!) = b\log b - b + O(\log b)$ and hence for $b = c\log n/\log\log n$, this term is $c\log n + o(\log n)$. Thus for $c < 1$ the dominant term in the exponent of the probability we are analyzing is $-n/n^c$, which tends to $-\infty$. Hence for any constant $c < 1$, with probability $1 - o(1)$, there will be a bin with load b in this Poissonized setting.

We now relate this Poissonized setting to the setting where exactly n balls are tossed. Because the number of balls in each bin increases monotonically with the number of balls tossed (i.e. by tossing additional balls, you can never decrease the number of balls in a bin), all we need to argue is that with high probability, $k \leq n$. By Chebyshev's inequality,

$$\Pr[k \geq n] = \Pr[|k - \mathbf{E}[k]| \geq n/2] \leq \frac{\mathbf{Var}[k]}{(n/2)^2} = \frac{n/2}{n^2/4} = o(1).$$

(We could have also used Fact 3 for an even tighter, inverse exponential tail bound here.) □

Note: At this point we have finished the material that goes with the before-class videos. The notes below are for reference after class.

3 Power of Two Choices

Propositions 2 and 5 show that when n balls are tossed into n bins uniformly at random, with high probability the maximum bin load is $\theta(\frac{\log n}{\log\log n})$. Given that the average bin load is 1, one might wonder whether there is an "easy" way to end up with an asymptotically smaller maximum bin load without having the balls coordinated (i.e. ball i goes to bin i), or without the balls needing to look at a bunch of bins (i.e. each ball looks through all the bins and goes into the first empty bin it finds).

As a bit of practical motivation/context for this question, the balls-in-bins random process is a natural model of hashing, and there are number of practical allocation tasks that can be considered

via this model. For example, consider allocating tasks to processors, or choosing a server in a network. At least in some of these settings, querying the objects in question to find a "free" one might be fairly wasteful in terms of communication, or the time it takes to process these queries. Hence a very simple randomized scheme would be ideal. Also, in many of these settings, it does make sense to have the number of requests in a given time period (i.e. the number of balls, n) approximately equal to the number of resources (i.e. the number of bins, m). For example, with a machine at full utilization, it doesnt make sense to have a bunch of extra processors that aren't doing anything, and if we have far fewer processors than jobs requested, we will end up with a long backlog.

The *power of two choices* is the surprising fact that, if each ball chooses 2 uniformly random bins, and then goes into the less full bin (breaking ties arbitrarily), then the maximum bin load decreases from $\theta(\frac{\log n}{\log \log n})$ to $\theta(\log \log n)$. This influential insight was discovered by Michael Mitzenmacher [1] around 2000, and has had a significant impact on both practice, and theory.

Theorem 2. *Suppose we allocate n balls to n bins as follows: the balls are allocated one at a time, and for each ball, two bins are selected uniformly at random, with the ball "choosing" the least full out of these two options, breaking ties* in any way. *With high probability, the maximum bin load will be at most* $\log_2 \log n + O(1)$.

Before formally proving this theorem, we first sketch the proof. Let B_i denote the number of bins that have at least i balls after all n balls are tossed. Trivially, $B_2 \leq \frac{n}{2}$, since otherwise there would be more than n balls. We will now use a bound on B_i to give a bound on B_{i+1}. Given that $B_2 \leq n/2$ after all the balls are tossed, at any intermediate step of the algorithm, there will certainly be at most $n/2$ bins with at least 2 balls. Let us bound B_3 by bounding the number of balls that will be the 3rd (or higher) ball to join a bin. For a ball to be the 3rd (or 4th, etc) ball in a bin, it must be the case that *both* of its options had at least 2 balls. Hence, the probability that a ball becomes the 3rd ball in a bin is at most $(B_2/n)^2$, where the exponent of 2 is there because both bin choices must be bad. Plugging in our bound of $B_2 \leq n/2$, we get that each ball will become the 3rd ball in a bin with probability at most $(1/2)^2 = 1/4$. For the time being, lets ignore the fact that the events that different balls become a "3rd ball in a bin" are dependent. If they were independent, then, by a Chernoff bound, we would expect that $B_3 \leq n/4 + o(n)$ with high probability.

For the purpose of this sketch, lets ignore this $o(n)$, and continue the argument. Assuming that, at the end, at most $B_3 \leq n/4$ buckets have 3 or more balls, for a ball to end up as the 4th ball in a bucket, it must have both its options having 3 or more balls, and the probability of this will be $(1/4)^2$, yielding that $\mathbf{E}[B_4] \approx \frac{n}{2^4}$, and we might expect $B_4 \leq n/2^4 + o(n)$ with high probability. In general, ignoring the little-o term, this reasoning argues that if $B_i \leq cn$, then we expect $B_{i+1} \leq c^2 n$. At what point does this end? Well suppose $B_i \ll \sqrt{n}$, then or each ball, the probability it is the $i+1$st ball in a bin will be at most $1/\sqrt{n}^2 < 1/n$, and by a union bound, $B_{i+1} = 0$ with some reasonable probability.

Putting this all together, we have $B_2 \approx \frac{1}{2}n$, $B_3 \approx \frac{1}{2^2}n$, $B_4 \approx \frac{1}{2^{2^2}}n$, and in general, $B_i \approx \frac{1}{2^{2^{i-2}}}n$. This expression will be 1 (certainly less than $\sqrt{n}$) when $2^{2^{i-2}} \approx n$, which is precisely when $i = 2 + \log \log n$.

To end up with a formal proof, we need to modify the above sketch in two ways: first, we obviously need to do some bookkeeping and keep track of the $o(n)$ terms which we ignored. Second, and more fundamentally, we need to be a little careful in how we use a Chernoff bound to get a high-probability bound on B_{i+1} in terms of a bound on B_i. After we analyze the randomness of

the choices of each ball to derive a bound on B_i, we cannot simply pretend that the balls have new randomness in our analysis of B_{i+1}. All of the B_i's are dependent on each other.

To help us handle this conditioning and still get a Chernoff-style bound, the following intuitive lemma will be helpful. In the following lemma, you can think of Z_i as the set of bin loads after the first i balls have been tossed.

Lemma 6. *Let $X_1, \ldots, X_n$ be a set of 0/1 random variables, and let $Z_1, \ldots, Z_n$ be a set of random variables such that X_i depends on $Z_1, \ldots, Z_i$. Then if $\Pr[X_i = 1 | Z_1, \ldots, Z_{i-1}] \leq p$ for all i, then for any c,*

$$\Pr[\sum_{i=1}^{n} X_i \geq c] \leq \Pr[Binomial(n, p) \geq c].$$

Proof. (Sketch) Considering each X_i sequentially, it holds that the probability of $X_i = 1$ is less than if X_i were an independent flip of a p-biased coin. The proof then follows by induction. □

Using this lemma, it is possible to make the intuition sketched above go through. We omit the formal proof in these notes—if you are curious, you can check out the proof in [1].

3.1 Beyond Two Choices

Given that choosing between two options is almost *exponentially* better than only having one choice—$\log n / \log \log n \approx \log n$ is much worse than $\log \log n$—you might wonder what happens if you get three, or more choices? More choices helps, but not much—it just changes the exponent in the base of the logarithm:

Theorem 3. *Suppose we allocate n balls to n bins as follows: the balls are allocated one at a time, and for each ball, d bins are selected uniformly at random, with the ball "choosing" the least full out of these d options, breaking ties* in any way. *With high probability, the maximum bin load will be at most* $\log_d \log n + O(1) = \frac{\log \log n}{\log d} + O(1)$.

The proof of this theorem is analogous to the the case when $d = 2$, and it is worth sanity-checking this for yourself by performing the proof sketch in this more general case.

It is also worth thinking about how tie-breaking rules might improve the maximum load. Perhaps surprisingly, if you tie-break by putting the ball in the bucket with lower index, this does improve the leading coefficient! (You will see a problem about this on the weekly problem set :)

References

[1] Michael Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1094–1104, 2001.