Tail Bound Examples

1 Balls & Bins

Imagine we throw m balls i.i.d. uniformly into n bins. Let X_i be the number of balls in bin i. Clearly $\mathbb{E}[X_i] = m/n$ by linearity of expectation.

One interesting statistic is the maximum load $\mathbb{E}[\max_i X_i]$. Note that it does **not** equal $\max_i \mathbb{E}[X_i] = m/n$. In particular for $m = n$, it is very unlikely to have exactly one ball in each bin.

We need to find a z such that $\Pr[X_i > z]$ is small for all i – so small that we can apply a union bound to bound

$$\bigcup_i [X_i > z].$$

To do so, we reverse engineer a small z for which this works. Say we want $\Pr[\exists i : X_i > z] < 1/n$. To apply a union bound, we want

$$\forall i : \Pr[X_i > z] < \frac{1}{n^2}.$$

Let X_{ij} be the indicator random variable for whether ball j goes to bin i. We know $X_i = \sum_{j=1}^{m} X_{ij}$, and the X_{ij} are independent Bernoulli, for fixed i. (For fixed j, they are of course not independent). So, we can apply Chernoff bounds:

$$\Pr[X_i > (1+\delta)\mathbb{E}[X_i]] < \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{\mathbb{E}[X_i]}.$$

Substituting,

$$\Pr[X_i > (1+\delta)\frac{m}{n}] < \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{\frac{m}{n}}.$$

How large do we have to make δ such that the right-hand-side is less than $1/n^2$? This will give us $z = (1+\delta)\cdot m/n$.

Two example ranges:

1. $m = 8n\log n$. Then $m/n = 8\log n$, so

$$\left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{\frac{m}{n}} = \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{8\log n}.$$

   Choose $\delta = 1$. Then

$$\frac{e^\delta}{(1+\delta)^{(1+\delta)}} = \frac{e}{4}.$$

   So,

$$\begin{aligned}\left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{8\log n} &= \left(\frac{e}{4}\right)^{8\log n} \\ &= \left(\left(\frac{e}{4}\right)^8\right)^{\log n} \\ &< \left(\frac{1}{e^2}\right)^{\log n} \\ &= \frac{1}{n^2}.\end{aligned}$$

2. $m = n$. Then $m/n = 1$, so
$$\left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{\frac{m}{n}} = \frac{e^\delta}{(1+\delta)^{(1+\delta)}}.$$

We have
$$\begin{aligned}\frac{e^\delta}{(1+\delta)^{(1+\delta)}} &= \frac{e^{1+\delta}}{(1+\delta)^{(1+\delta)}}\\ &< \frac{e}{n^2}.\end{aligned}$$

More generally, we want to solve
$$\frac{e^x}{x^x} = \frac{1}{y}$$

Here $y = n^2/e$.

Lemma 1. *Let x solve $x^x/e^x = y$. Then $x = \Theta(\log y / \log\log y)$.*

Proof. Take logs to obtain
$$\begin{aligned}\log y &= x\log x - x\\ &= x(\log x - 1).\end{aligned}$$

Dropping the -1 and taking logs again,
$$\log\log y = \log x + \log\log x.$$

For large enough y,
$$\begin{aligned}2\log x &\geq \log x + \log\log x\\ &\geq \log\log y\\ &\geq \log x.\end{aligned}$$

Thus,
$$\begin{aligned}\log y &\leq x\log x\\ &\leq x\log\log y.\end{aligned}$$

So,
$$x \geq \frac{\log y}{\log\log y}.$$

In the other direction,
$$\begin{aligned}x &= \frac{\log y}{\log x - 1}\\ &\leq \frac{\log y}{\frac{1}{2}\log\log y - 1}\\ &\leq \frac{4\log y}{\log\log y}.\end{aligned}$$

Therefore $x = \Theta(\log y / \log\log y)$. □

Conclusion. In the range $m \geq 8n\log n$, $\delta = 1$ is enough, so with high probability no bin contains more than $16\log n$ balls. Generally, when μ is large enough, it is unlikely that any bin exceeds its expectation by more than a constant factor. In the range $m = n$, we can still guarantee that the load of each bin is at most $\Theta(\log n / \log\log n)$ with high probability. The most common cases in applying Chernoff bounds are $\mathbb{E}[X_i] = \Omega(\log n)$ and $\mathbb{E}[X_i] = \Theta(1)$.

2 Set Balancing

Given a matrix $\mathbf{A} \in \{0,1\}^{n\times n}$, our goal is to pick a vector $\boldsymbol{b} \in \{-1,1\}^n$ to minimize the maximum absolute entry in $\mathbf{A}\boldsymbol{b}$, *i.e.*, minimize $\max_i |(\mathbf{A}\boldsymbol{b})_i|$. This is a basic problem in a subfield called discrepancy theory.

The motivation is that each row represents a skill, and each column represents a player. $\boldsymbol{b}$ says which team player i belongs to. We want the teams to be as even as possible for all n skills.

Example. Let

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Then

$$\boldsymbol{b} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \implies \begin{bmatrix} -2 \\ -2 \\ -2 \end{bmatrix} \implies 2.$$

$$\boldsymbol{b} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \implies \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} \implies 2.$$

$$\boldsymbol{b} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} \implies \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \implies 2.$$

For each i, let $b_i = \pm 1$ i.i.d uniformly. We are interested in X_j, the differential in team strengths for skill j. Our goal is to bound $\max_j |X_j|$. We have

$$X_j = \sum_i b_i a_{ji}.$$

And

$$\begin{aligned} \mathbb{E}[X_j] &= \sum_i \mathbb{E}[b_i] a_{ji} \\ &= 0 \end{aligned}$$

To apply Chernoff bounds, we would need to shift values to make $\mu > 0$. Instead we'll just use Hoeffding bounds. Because $\mu = 0$,

$$\begin{aligned} \Pr[|X_j - \mathbb{E}[X_j]| > \Delta] &= \Pr[|X_j| > \Delta] \\ &< 2\exp\Big(-\frac{2\Delta^2}{\sum_j 2^2}\Big) \\ &= 2\exp\big(-\frac{2\Delta^2}{4n}\big) \end{aligned}$$

We'd like this to be $\leq 2/n^2$, so we reverse engineer Δ. That is,

$$\exp\big(-\frac{\Delta^2}{2n}\big) \leq \frac{1}{n^2}.$$

Thus

$$\frac{\Delta^2}{2n} \geq \log(n^2) = 2\log n.$$

So

$$\Delta = \sqrt{4n\log n} = 2\sqrt{n\log n}.$$

This gives $|X_j| \leq 2\sqrt{n\log n}$ with probability at least $1 - 2/n^2$, so by a union bound we have that all $|X_j| \leq 2\sqrt{n\log n}$ with probability $1 - 2/n$.

3 Connecting Terminals to Minimize Congestion (Raghavan-Thompson)

Given a graph G (directed or undirected) and k terminal pairs (s_i, t_i). The goal is to find a (possibly directed) path P_i from s_i to t_i for each i. The **congestion** of edge e is the number of paths P_i that include edge e. We'd like to minimize the maximum congestion over all edges.

One special case would ask if there is a possible assignment with congestion 1. This is called the edge-disjoint paths problem (EDP). It is $\mathcal{NP}$-hard for directed graphs, even for $k = 2$. It is $\mathcal{NP}$-hard for undirected graphs when k is part of the input. For any constant k, there is a polynomial time algorithm for the edge-disjoint paths. The dependence on k is horrendous! This uses the graph minors framework of Robertson and Seymour.

For approximating congestion:

Theorem 2. *[Chuzhoy/Naur]. For directed graphs, max-congestion is $\mathcal{NP}$-hard to approximate to within $\mathcal{O}(\log \log n)$.*

Theorem 3. *[Andrews/Zhang]. For undirected graphs, max-congestion is $\mathcal{NP}$-hard to approximate to within $\mathcal{O}((\log \log n)^{1-\epsilon})$ for any $\epsilon > 0$.*

The distinction from the easy EDP problem is that in the easy version, any s_i can be connected to any t_j, but here each s_i must be connected to a specific t_j. In the easy version, we had a multi-flow problem where one unit of flow entered at each of the sources, one unit of flow exited at each of the sinks, and each edge had capacity one. It was essential that there was an optimal integral flow so we could do a path decomposition.

Our approach is to set up a flow problem via an LP and obtain a fractional flow. Somehow we will need to round this flow intelligently.

Our flow problem has one unit of flow entering at s_i which must be routed to t_i. Let $f_i(e)$ be the flow through edge e originating from s_i and c be the overall congestion. We have the LP:

$$
\begin{aligned}
\min \quad & c \\
\text{s.t.} \quad & \sum_{e \text{ out of } s_i} f_i(e) = 1 && \forall i \\
& \sum_{e \text{ into } v} f_i(e) = \sum_{e \text{ out of } v} f_i(e) && \forall i \text{ and } \forall v \notin \{s_i, t_i\} \\
& f_i(e) \geq 0 && \forall i \text{ and } \forall e \\
& c \geq \sum_i f_i(e) && \forall e
\end{aligned}
$$

For an integrality gap, imagine we have two nodes s_i and t_i with m edges. Then the LP splits flow evenly for a congestion of $1/n$, while the actual congestion is 1. Then the integrality gap is m – pretty bad!

But, the actual lower bound on the congestion is 1, so if we encode that as a constraint, we will get rid of the bad integrality gap.

$$
\begin{aligned}
\min \quad & c \\
\text{s.t.} \quad & \sum_{e \text{ out of } s_i} f_i(e) = 1 && \forall i \\
& \sum_{e \text{ into } v} f_i(e) = \sum_{e \text{ out of } v} f_i(e) && \forall i \text{ and } \forall v \notin \{s_i, t_i\} \\
& f_i(e) \geq 0 && \forall i \text{ and } \forall e \\
& c \geq 1 \\
& c \geq \sum_i f_i(e) && \forall e
\end{aligned}
$$