CS265/CME309: Randomized Algorithms and Probabilistic Analysis

Lecture #5: Moment-Generating Functions, Chernoff Bounds, and Randomized Routing on the Hypercube

Gregory Valiant[*], updated by Mary Wootters

October 9, 2023

1 Introduction

Continuing the theme of tail bounds, in today's class we will begin by proving some Chernoff bounds. Chernoff bounds, in their most simple form, apply to sums of independent random variables, and roughly state that the probability that the sum deviates from its expectation by more than c standard deviations, decreases inverse *exponentially* with c^2. To see the intuition for why we should expect this, consider the Central Limit Theorem:

Theorem 1. *Let $X_1, \ldots, X_n$ be independent, identically distributed random variables with $\mathbf{E}[X_i] = \mu$ and $\mathbf{Var}[X_i] = \sigma^2$. Then, as $n \to \infty$, the distribution of $\frac{\frac{1}{n}\sum_{i=1}^n (X_i - \mu)}{\sigma}$ converges to the standard Gaussian, $N(0,1)$.*

The above central limit theorem implies that, for sufficiently large n, we would expect that $\Pr\left[|\sum_{i=1}^n X_i - n\mu| \geq c\sqrt{\mathbf{Var}[\sum X_i]}\right] \approx \Pr[|Z| \geq c] \leq e^{-c^2/2}$, where Z denotes a Gaussian random variable with mean 0 and variance 1, and the $e^{-x^2/2}$ comes from the probability density function of a Gaussian.

How can we make this sort of statement rigorous? The above analysis only holds in the limit, as n approaches infinity, and we would like to make statements about the probability of deviations from the expectation in concrete settings with finite n. Additionally, we might hope to make similar statements in settings where the distribution we care about is *not* that similar to a Gaussian. Chernoff bounds will provide a relatively easy approach to addressing both of these hopes.

2 Moment-Generating Functions

Chernoff bounds proceed by analyzing the *moment-generating function* of a random variable:

[*]©2019, Gregory Valiant. Not to be sold, published, or distributed without the authors' consent.

Definition 1. *The moment-generating function of a random variable, X, is a function $M_X : \mathbb{R} \to \mathbb{R}$, defined by $M_X(t) = \mathbf{E}[e^{tX}]$.*

These are referred to as moment-generating functions, because the derivatives of M_X, evaluated at 0, give the moments of X (provided M_X exists in a neighborhood of 0). To see this, note that one can switch the order of $\mathbf{E}$ and differentiation (provided the function in question is well-behaved), and so the kth derivative of $M_X(t)$, evaluated at $t = 0$, is given by $\frac{d^k M_X(t)}{dt^k}(0) = \mathbf{E}[X^k e^{0 \cdot X}] = \mathbf{E}[X^k]$.

The following fact asserts that the moment-generating function actually characterizes the distribution of the variable in question, in the same sense that the Taylor expansion of a function at a point uniquely defines the function, provided the function is "well-behaved" (i.e. is "analytic").

Fact 2. *Given random variables, X and Y, if there is some $\delta > 0$ such that $M_X(t) = M_Y(t)$ for all $t \in (-\delta, \delta)$, then the distribution of X is the same as the distribution of Y.*

Example 3. *Let $Z \leftarrow N(\mu, \sigma^2)$ denote a random variable distributed according to the Gaussian of mean μ and variance σ^2. Using the fact that the probability density function at value x is $\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, we have that*

$$M_X(t) = \mathbf{E}[e^{tX}] = \int_{-\infty}^{\infty} e^{tx} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx.$$

"Completing the square" in the exponent yields that the above is equal to:

$$\begin{aligned}
\int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2 - 2\mu x + \mu^2 - 2\sigma^2 t x}{2\sigma^2}} dx &= \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-(\mu+\sigma^2 t))^2 + \mu^2 - (\mu+\sigma^2 t)^2}{2\sigma^2}} dx \\
&= e^{\frac{(\mu+\sigma^2 t)^2 - \mu^2}{2\sigma^2}} \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-(\mu+\sigma^2 t))^2}{2\sigma^2}} dx \\
&= e^{\frac{(\mu+\sigma^2 t)^2 - \mu^2}{2\sigma^2}} = e^{\mu t + \frac{1}{2}\sigma^2 t^2}.
\end{aligned}$$

In the second from last line, we leveraged the fact that the integral is integrating a probability density function of a Gaussian, and hence the integral will evaluate to the total amount of probability, namely, 1.

Why is the above useful? Well, suppose we want to know the distribution of the sum of two independent Gaussians, $X \leftarrow N(\mu_1, \sigma_1^2)$ and $Y \leftarrow N(\mu_2, \sigma_2^2)$. By linearity of expectation, we know $\mathbf{E}[X + Y] = \mu_1 + \mu_2$, and $\mathbf{Var}[X + Y] = \mathbf{Var}[X] + \mathbf{Var}[Y] = \sigma_1^2 + \sigma_2^2$. But what is the distribution of $X + Y$? Well, lets consider the moment-generating function

$$M_{X+Y}(t) = \mathbf{E}[e^{t(X+Y)}] = \mathbf{E}[e^{tX} e^{tY}] = \mathbf{E}[e^{tX}]\mathbf{E}[e^{tY}],$$

where we leveraged the assumption that X and Y are independent in this last equality. Plugging in the moment-generating function for the Gaussian that we derived above, we have

$$M_{X+Y}(t) = \mathbf{E}[e^{tX}]\mathbf{E}[e^{tY}] = e^{\mu_1 t + \frac{1}{2}\sigma_1^2 t^2} \cdot e^{\mu_2 t + \frac{1}{2}\sigma_2^2 t^2} = e^{(\mu_1+\mu_2)t + \frac{1}{2}(\sigma_1^2 + \sigma_2^2)t^2}.$$

This expression is exactly the moment generating function for a Gaussian with mean $\mu_1 + \mu_2$ and variance $\sigma_1^2 + \sigma_2^2$, hence by Fact 2, the sum of independent Gaussians must be Gaussian!

Not all moment generating functions are as annoying to compute as in Example 3. In particular, because summations in the exponent turn into multiplications, the moment generating function of a sum of independent random variables is the product of the moment generating functions. The following example illustrates one important case we will use in our derivation of Chernoff bounds:

Example 4. *Let X_i denote a 0/1 valued random variable that is 1 with probability p_i, and assume all the X_i's are independent.*

$$M_{X_i}(t) = \mathbf{E}[e^{tX_i}] = p_i e^{t\cdot 1} + (1-p_i)e^{t\cdot 0} = p_i e^t + (1-p_i) = 1 + p_i(e^t - 1).$$

Hence if $X = \sum_i X_i$, then

$$M_X(t) = \mathbf{E}[e^{t\sum_i X_i}] = \mathbf{E}[\prod_i e^{tX_i}] = \prod_i M_{X_i}(t) = \prod_i \left(1 + p_i(e^t - 1)\right).$$

3 Chernoff Bounds

As mentioned in the introduction, Chernoff bounds give inverse exponential tail bounds to "nice" random variables, such as those given by sums of independent random variables. One reason why such tail bounds are so useful from an algorithmic standpoint, is that it will allow us to apply a union bound over *exponentially* many such tail events. (We will explore one such example in Section 4.)

There are many different variants of Chernoff bounds, though they are all proved via the same sort of approach: applying Markov's inequality to the moment generating function of the random variable in question. The approach to deriving a Chernoff bound proceeds from realizing that, for any positive number t, $a \geq b$ if and only if $ta \geq tb$ which is true if and only if $e^{ta} \geq e^{tb}$ since the exponential function is monotonically increasing. Similarly, if $t < 0$, then $a \leq b$ is equivalent to $ta \geq tb$. Hence, we have the following two statements, where the last inequality follows from apply Markov's inequality to the random variable e^{tX}:

- For any $t > 0$, $\Pr[X \geq c] = \Pr[e^{tX} \geq e^{tc}] \leq \frac{\mathbf{E}[e^{tX}]}{e^{tc}}$.
- For any $t < 0$, $\Pr[X \leq c] = \Pr[e^{tX} \geq e^{tc}] \leq \frac{\mathbf{E}[e^{tX}]}{e^{tc}}$.

The top statement corresponds to bounding the probability that X is too large, and the bottom statement corresponds to bounding the probability that X is too small. The beauty of these statement is that the top statement holds *for any* $t > 0$, and the bottom holds *for any* $t < 0$. Hence, to obtain the best bounds possible, we will analyze the right hand side, and plug in the value of t that minimizes these probabilities. Hence the only difference between proving upper tail bounds and lower tail bounds, is that in one case we are restricted to positive t's, and the other case we are restricted to $t < 0$.

3.1 Sums of Independent 0/1 Random Variables

We now instantiate the above general approach for the special case where X is the sum of independent 0/1 random variables. Specifically, let $X = \sum_{i=1}^{n} X_i$, where X_i is an independent 0/1 valued random variable, with $\Pr[X_i = 1] = p_i$. Let $\mu = \mathbf{E}[X] = \sum_i p_i$.

Theorem 2. *Let $X = \sum_{i=1}^{n} X_i$, where X_i is an independent 0/1 valued random variable, with $\Pr[X_i = 1] = p_i$.*

- *For any $\delta > 0$, $\Pr[X \geq (1+\delta)\mathbf{E}[X]] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mathbf{E}[X]}$.*
- *For any $\delta \in (0,1]$, $\Pr[X \leq (1-\delta)\mathbf{E}[X]] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mathbf{E}[X]}$.*

Proof. For notational convenience, let μ denote $\mathbf{E}[X]$.From Example 4,

$$\mathbf{E}[e^{tX}] = \prod_i \left(1 + p_i(e^t - 1)\right) \leq \prod_i e^{p_i(e^t-1)} = e^{\mu(e^t-1)},$$

where, in the last inequality, we used the fact that for any $x > 0$, $1 + x < e^x$.

Plugging this into the statements from the previous section, we have that for any $t > 0$,

$$\Pr[X \geq c] = \Pr[e^{tX} \geq e^{tc}] \leq \frac{\mathbf{E}[e^{tX}]}{e^{tc}} \leq \frac{e^{\mu(e^t-1)}}{e^{tc}}.$$

Plugging in $c = (1+\delta)\mu$, this expression becomes $\frac{e^{\mu(e^t-1)}}{e^{t(1+\delta)\mu}} = e^{\mu\left((e^t-1)-t(1+\delta)\right)}$. Now we just need to plug in a value of t that minimizes this expression, which is easy since this is minimized by minimizing the exponent. Plugging in $t = \log(1+\delta)$ yields that $e^t = 1+\delta$, and this expression becomes $e^{\mu(\delta - \log(1+\delta)(1+\delta))} = \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}$, as desired.

The proof of the lower bound is very similar: we have that for any $t < 0$,

$$\Pr[X \leq c] = \Pr[e^{tX} \geq e^{tc}] \leq \frac{\mathbf{E}[e^{tX}]}{e^{tc}} \leq \frac{e^{\mu(e^t-1)}}{e^{tc}}.$$

Plugging in $c = (1-\delta)\mu$, this expression becomes $\frac{e^{\mu(e^t-1)}}{e^{t(1-\delta)\mu}} = e^{\mu\left((e^t-1)-t(1-\delta)\right)}$. Now we just need to plug in a (negative) value of t that minimizes this expression. Plugging in $t = \log(1-\delta)$ (which is negative) yields that this expression becomes $e^{\mu(-\delta - \log(1-\delta)(1-\delta))} = \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu}$ as claimed. □

The bounds given by Theorem 2 might be a bit tricky to parse, and the following easy corollaries might be handier to apply.

Corollary 5. *Let $X = \sum_{i=1}^{n} X_i$, where X_i is an independent 0/1 valued random variable, with $\Pr[X_i = 1] = p_i$, and $\mu = \sum p_i$, the following bounds hold:*

- *For any $\delta \in (0,1]$, $\Pr[X \geq (1+\delta)\mu] \leq e^{-\frac{\mu\delta^2}{3}}$.*
- *For any $\delta \in (0,1]$, $\Pr[X \leq (1-\delta)\mu] \leq e^{-\frac{\mu\delta^2}{2}}$.*
- *For any $c \geq 6$, $\Pr[X \geq c\mu] \leq 2^{-c\mu}$.*

Proof. For the first statement, note that $\frac{e^{\delta}}{(1+\delta)^{1+\delta}} = e^{\delta - (1+\delta)\log(1+\delta)} \leq e^{-\delta^2/3}$, which can be verified by showing that the exponent $\delta - (1+\delta)\log(1+\delta) \leq -\delta^2/3$ by analyzing the taylor expansion of $\log(1+\delta)$. The proofs of the other two statements are similar—it's a good exercise to work them out! □

The following corollary is also handy when all you have is a bound on the expectation in question. This corollary basically says that if you are trying to bound the upper tail, you can just plug in your upper bound on the expectation in place of the expectation, and analogously for lower tail bounds.

Corollary 6.

For any $\delta > 0$, and any $c \geq \mathbf{E}[X]$, $\Pr[X \geq (1+\delta)c] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^c$.

For any $\delta \in (0,1]$, and any $c \leq \mathbf{E}[X]$, $\Pr[X \leq (1-\delta)c] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^c$.

Proof. To prove the first statement, note that you can define independent 0/1 random variables $Y_1, \ldots, Y_m$ such that $\mathbf{E}[\sum_j Y_j] = c - \mathbf{E}[\sum_i X_i]$, and hence if you define the random variable $Z = \sum_i X_i + \sum_j Y_j$, then it is *always* true that $X \leq Z$. Additionally, since $\mathbf{E}[Z] = c$, Theorem 2 applies to Z, and hence we have:

$$\Pr[X > (1+\delta)c] \leq \Pr[Z > (1+\delta)c] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^c.$$

The proof of the second statement is similar. One can define independent 0/1 random variables $Y_1, \ldots, Y_n$ such that if we set $Z = \sum_i X_i Y_i$, then $X \geq Z$, and $\mathbf{E}[Z] = c$. [We can do this by setting $\Pr[Y_i = 1] = c/\mathbf{E}[X]$ for all i. Since Z is a sum of independent 0/1 random variables, our Chernoff bound of Theorem 2 applies, yielding:

$$\Pr[X \leq (1-\delta)c] \leq \Pr[Z \leq (1-\delta)c] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^c.$$

$\square$

Remark 7. *It is possible to prove a variety of other Chernoff bounds, by making different choices for how to simplify the moment-generating function $\mathbf{E}[e^{tX}] = \prod_i (p_i e^t + (1-p_i))$. Rather than expressing this as $\prod_i (1 + p_i(e^t - 1))$ and bounding this by $\prod_i e^{p_i(e^t-1)}$ as we did above, we could apply the arithmetic-mean geometric-mean ("AMGM") inequality, to conclude that*

$$\prod_i \left(p_i e^t + (1-p_i)\right) \leq \left(pe^t + (1-p)\right)^n,$$

where $p = \frac{1}{n}\sum_i p_i$. If we had taken this route, and then optimized for t, we would have ended up with a different set of perfectly reasonable Chernoff bounds, which would be even better than those given in Theorem 2 in the case that the AMGM inequality is tight—namely if all the p_i's are identical, or very similar.

There are also other Chernoff-like bounds that can be useful. For reference, here are a few of them.

Hoeffding's inequality allows you to apply the Chernoff bound to any bounded random variables, not just $0/1$-valued:

Theorem 3 (Hoeffding's Inequality)**.** *Suppose that $X_1, \ldots, X_n$ are independent random variables with $X_i \in [a_i, b_i]$ almost surely for all i. Then for any $t > 0$,*

$$\Pr\left[\left|\sum_i (X_i - \mathbb{E}X_i)\right| \geq t\right] \leq \exp\left(\frac{-2t^2}{\sum_i (a_i - b_i)^2}\right).$$

Bernstein's inequality can give an even better bound if the X_i additionally have small variance:

Theorem 4 (Bernstein's Inequality). *Suppose that $X_1, \dots, X_n$ are independent mean-zero random variables with $|X_i| \leq M$ almost surely for all i. Then for any $t > 0$,*

$$\Pr\left[\left|\sum_i X_i\right| \geq t\right] \leq \exp\left(\frac{-t^2/2}{\sum_i \mathbb{E}[X_i^2] + Mt/3}\right).$$

Note: At this point we are done with the material covered in the mini-lectures to be watched before class. In Section 4 below, there are notes for an application to randomized routing that we will discuss in class. These notes are here for reference after class.

4 Randomized Routing on the Hypercube

We will now investigate an extremely simple randomized routing scheme. The stylized setting we will consider is as follows: There are $N = 2^n$ "nodes", indexed 1 through N, and each of which corresponds to one of the vertices of the n-dimensional hypercube. Every node has a packet that must be routed to a distinct other node. We can think of this as the problem of routing a permutation, π, of the N nodes, where node i wishes to send a packet to node $\pi(i)$. We assume that each node has direct connections to n other nodes, corresponding to the adjacent vertices in the hypercube (i.e. i, j have a direct edge if i and j differ in exactly 1 bit in their binary representations). A valid routing scheme is any way for getting all the packets to their destinations, such that the following rules are upheld:

1. Time proceeds in discrete steps, and at each time step, at most one packet can traverse each edge (in each direction).
2. Packets can queue at nodes. For simplicity, we will assume that nodes queue in a FIFO (first-in-first-out) fashion.

Given these rules, the goal will be to route every packet to its destination in such a way that the total number of timesteps is minimized. Note that since the diameter of the network is n, there are permutations that trivially require n steps to route, even if there is no delay/congestion. How close can we get to $O(n)$ time routing?

Beyond minimizing the total routing time, ideally, the routing protocol will be *oblivious*, in the sense that the route that each packet takes depends only on the source and destination of the packet (and does not depend on the other packets).

Definition 8. *An* oblivious *routing scheme is one where the route that the ith packet takes from node i to node $\pi(i)$ depends only on i and $\pi(i)$.*

The following theorem shows that there is no deterministic oblivious scheme that works well for all routings, π. The proof is not too difficult, though we omit it (feel free to check out the original reference).

Theorem 5 ([1]). *For any deterministic, oblivious routing scheme on the hypercube, there exists a set of source/destinations, π that require $\geq \frac{2^{n/2}}{n}$ steps to successfully route.*

We now describe an extremely simple randomized routing. In one sentence, each packet chooses a node uniformly at random, and routes the packet to that intermediate node, and then routes from that node to the destination. The randomization in the choice of the intermediate node ensures that, with high probability, there is relatively little congestion. The specifics for how the packet gets from the source to the intermediate node, and then from the intermediate node to the destination, does not matter too much. For concreteness, we will specify that these two phases occur by "bit-fixing".

Definition 9. *The* bit-fixing *path from a node i to node j is a sequence of nodes, starting with i and ending with j, where each successive node has a binary representation that differs from the previous node in exactly one location, where that location is the leftmost bit on which the current node differs from j. An example will clarify: suppose $i = 001010$ and $j = 101001$, the bit-fixing sequence from i to j is*

$$i = 001010 \to 101010 \to 101000 \to 101001 = j.$$

Algorithm 10. RANDOMIZED ROUTING ON THE HYPERCUBE

```
Given permutation $\pi : \{1, \dots, N\} \to \{1, \dots, N\}$, for $N = 2^n$:
  1. Each node, $i$, chooses an ''intermediate node'' $\delta_i$ uniformly at random
     from $\{1, \dots, N\}$.
  2. Phase 1: each packet is routed from $i$ to $\delta_i$ via ''bit-fixing'', and
     if a packet reaches $\delta_i$ before a total of $3n$ timesteps, the packet
     waits at the node $\delta_i$.
  3. Phase 2: After $3n$ timesteps, if the ith packet is at $\delta_i$, then it is
     routed from $\delta_i$ to $\pi(i)$ via ''bit-fixing''. If the ith packet is not at
     $\delta_i$ after $4n$ timesteps, then send it to $\pi(i)$ from wherever it is via the
     ''bit-fixing'' route.
```

Theorem 6. *With probability at least $1 - 2^{-3n+1}$, the above scheme will terminate after at most $6n$ timesteps.*

The high-level structure of the proof will be to focus on a single packet, i, and prove that the probability that it does not reach $\pi(i)$ by the claimed time is much less than 2^{-n}, and hence we can union bound over the $N = 2^n$ different packets. To get the inverse exponential probability that the ith packet fails to reach its destination in time, we will use a Chernoff bound. The analysis will first argue that Phase 1 successfully completes within the specified time of $4n$, and then essentially the same argument will show that, given that Phase 1 is successful, Phase 2 will also be successful with high probability within an additional $4n$ timesteps.

We will conduct the analysis from the perspective of the packet routing from i to δ_i. Let $D(i)$ denote the "delay" of the ith packet, namely the number of timesteps that the packet spends waiting in a queue (as opposed to traversing an edge). The following lemma asserts that $D(i)$ is bounded by the number of packets whose bit-fixing paths intersect packet i's path from i to δ_i.

Lemma 11. *Letting $P_i = (e_1, e_2, \dots, e_k)$ denote the bit-fixing path from i to δ_i, we claim that*

$$D(i) \leq |\{j : P_j \cap P_i \neq \emptyset\}| .$$

Proof. Although the lemma statement makes intuitive sense (if there are only 10 people that are using our edges, we should expect to be delayed at most 10 timesteps), we need to rigorously assign a unique intersecting packet to "blame" for each unit of delay that i experiences. To do this, we first observe that every two paths P_i and P_j that intersect, must intersect in a single contiguous segment. This is true because both paths are via bit-fixing orders, hence once the paths diverge, then can not merge later on. One way to do this is to imagine that, at the time when packet i incurs delay ℓ, i gives a certificate c_ℓ to whichever packet got to use the edge that caused i's delay. Once a packet, j, has a certificate, it carries it around, and will transfer the certificate to packet j' if j experiences a delay along path P_i caused by packet j' using the edge that j was waiting for. First, trivially, there is at most one certificate in existence for each unit of delay that i experiences. Now, we need to argue that, at the end of the protocol, no packet can end up with more than one certificate. To see this, we claim that, at any time step, the nodes in any one queue can in total either have 0 or 1 certificates. This is true because once a certificate c_ℓ is assigned, at every subsequent timestep it will either leave path P_i (never to return), or it will advance along path P_i by one node. In particular, the certificate will never "wait" at a node, since if the packet carrying the certificate is forced to wait, it will transfer that certificate to the packet using its desired edge. Hence, since no two packets originate at the same place in P_i at the same time, and they travel along P_i in the same (bit-fixing) order, they can never "collide" at a subsequent time, and hence no packet can get more than one certificate. □

We now finish the proof of Theorem 6.

Proof of Theorem 6. From Lemma 11, the number of time units that packet i spends waiting during Phase 1 is bounded by the number of other paths that intersect path P_i from i to δ_i. To bound this, fix path $P_i = (e_1, \ldots, e_k)$, and consider the expected number of paths P_j that intersect it. For each $j \neq i$, let X_j denote the 0/1 random variable representing the event that P_j intersects P_i. These random variables are not identically distributed, but they *are* independent, conditional on P_i, since once δ_i is fixed, X_j depends only on the randomness of δ_j. To bound $\mathbf{E}[\sum_j X_j]$, consider the expected number of paths that will use a given edge, e, in the hypercube. By symmetry, this expected number will be equal, for all edges, and hence it must be bounded by

$$\frac{\text{expected total sum of lengths of all paths}}{\text{number edges in hypercube}} = \frac{(n/2)2^n}{2^n \cdot n} = 1/2,$$

where the numerator follows from the expected distance between j and δ_j is $n/2$ since in expectation half the bits will differ, and there are 2^n paths, and the denominator is because each of the 2^n nodes has out degree n (again, we are imagining a directed edge between every pair of adjacent nodes).

Since P_i has at most n edges, and each of these edges, in expectation, intersects $1/2$ paths, $\mathbf{E}[\sum_j X_j] \leq n \cdot \frac{1}{2}$. Since $\sum_j X_j$ is the sum of independent 0/1 random variables, Chernoff bounds apply, and we can conclude by Corollary 6 that

$$\Pr[\sum_j X_j > 3n] \leq \Pr[\sum_j X_j > (1+5)n/2] \leq \left(\frac{e^5}{(1+5)^{1+5}}\right)^{n/2} \leq (1/2^8)^{n/2} \leq 2^{-4n}.$$

A union bound over the 2^n nodes yields that the probability that Phase 1 of algorithm does not successfully terminate after $3n$ timesteps is at most 2^{-3n}. The analysis that Phase 2 of the algorithm will also successfully resolve after at most this number of steps is identical, yielding the desired theorem. □

Remark 12. *You might wonder what would have happened if the algorithm routes packet i to δ_i, and then immediately routes from δ_i to $\pi(i)$, without waiting for the initial phase to finish. Intuitively, this sort of scheme seems quite reasonable, and my guess would be that it also would successfully terminate with high probability after $O(n)$ time steps, though the analysis is certainly more difficult.*

4.1 Beyond the Hypercube

Stepping back, you might wonder why routing a permutation on the hypercube is a question worth thinking about. Whether you are thinking about designing a set of network routers (e.g. for the stanford network), or designing connections between processors on a parallel computer, this settings is not too much of a caricature. For example, in the case of a parallel computer, after each operation, perhaps each processor might want to send its information to a different processor. Obviously if everyone wants to send their information to a single processor, there will be a bottleneck, and we should probably re-think whatever algorithm we are designing for the parallel computer. So, suppose we do have an algorithm which, after every local computation, allows processors to transmit a packet to another processor, subject to the condition that no processor is planning to receive more than one (for example) packets. How can we connect up the processors to allow for this sort of communication, and how efficiently can this communication be routed? From the previous section, we get the following punchline: If we have $N = 2^n$ processors, we then we can get away with only $\log N$ wires leaving each processor (e.g. a degree $\log N$ network), and the communication can be accomplished via a very simple randomized scheme that takes time only $O(\log N)$.

Can we hope to do any better? Is it conceivable that we could do this routing with degree $O(\log n)$ in much less than $O(\log N)$ time? No–any graph with degree at most $\log N$ will have two nodes with distance nearly $\log N$, and hence we shouldnt hope for this. What about improving on the number of wires? Can we hope for a graph with *constant* degree, that allows for routing in time $O(\log N)$? Yes!! There is a family of graphs, known as *butterfly networks*, which have constant degree, logarithmic diameter, and support a randomized routing procedure similar to the randomized routing of the hypercube that we saw above.

References

[1] C. Kaklamanis, D. Krizanc, and T. Tsantilas. Tight bounds for oblivious routing in the hypercube. In *Proceedings of the Second Annual ACM Symposium on Parallel Algorithms and Architectures*, SPAA '90, pages 31–36, New York, NY, USA, 1990. ACM.