Chernoff-Hoeffding's Bound

- One of the strongest (and perhaps most useful) tail bounds for "independent" var's.

Chernoff-Hoeffding: Assume $X_1, \dots X_n$ are independent random variables take values in $[0,1]$. Then for $X = \sum_{i=1}^{n} X_i$, $\mu = E[X]$:

1) for any $0 < \delta$: $\Pr[X \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$

2) for any $0 < \delta \leq 1$: $\Pr[X \geq (1+\delta)\mu] \leq e^{-\mu\delta^2/3}$

3) for any $r \geq 6\mu$: $\Pr[X \geq r] \leq 2^{-r}$.

Proof: For now suppose X_i Bernouli random variable $X_i \in \{0,1\}$ with $E[X_i] = \mu_i$

For all $t > 0$:

$$\Pr[X \geq (1+\delta)\mu] = \Pr[tX \geq t(1+\delta)\mu]$$
$$= \Pr[e^{tX} \geq e^{t(1+\delta)\mu}].$$

Using Markov's:

$$\Pr[e^{tX} \geq e^{t(1+\delta)\mu}] \leq \frac{E[e^{tX}]}{e^{t(1+\delta)\mu}} = \frac{E[\prod_i e^{tX_i}]}{e^{t(1+\delta)\mu}}$$
$$= \frac{\prod_i E[e^{tX_i}]}{e^{t(1+\delta)\mu}}$$

Note: $e^{tX_i} = \begin{cases} e^t & \mu_i \\ \dots \end{cases}$ and $1 + x \leq e^x$

Note: $e^{tX_i} = \begin{cases} e^{t} & \mu_i \\ 1 & 1-\mu_i \end{cases}$ and $1+x \leq e^{x}$

So $E[e^{tX_i}] = e^t\mu_i + (1-\mu_i) = 1+\mu_i(e^t-1) \leq e^{\mu_i(e^t-1)}$

$$\Pr[X \geq (1+\delta)\mu] \leq \frac{\prod_i [e^{\mu_i(e^t-1)}]}{e^{t(1+\delta)\mu}}$$

$$= \frac{e^{(e^t-1)\sum_i \mu_i}}{e^{t(1+\delta)\mu}} = \frac{e^{(e^t-1)\mu}}{e^{t(1+\delta)\mu}}$$

- let $t = \ln(1+\delta)$: $\Pr[X \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$
- Assumption on $X_i \sim$ Bernoulli $\in \{0,1\}$ can be removed.

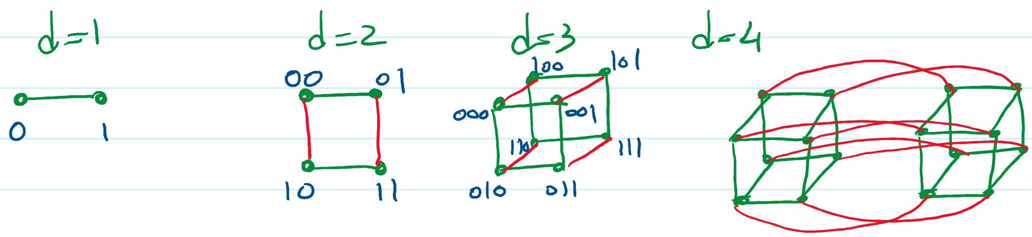
Corollary: For $0 < \delta < 1$:

$$\Pr[|X-\mu| \geq \delta\mu] \leq 2e^{-\mu\delta^2/3}.$$

- So with high probability $X \sim (1 \pm \delta)\mu$

Example: oblivious Routing on Hypercubes

d-dimensional Hypercube: $N = 2^d$ nodes



Definition: A hypercube or d_cube has 2^d nodes with d-bit binary strings as labels for nodes.

Two nodes are adjacent iff their labels differ in one bit (So it has degree d).

- Each vertex has a packet (packet i at node i) with destination $d(i)$. We assume $d(i)$'s are all distinct (So they form a permutation of nodes)
- The system is synchronous: works in rounds
- In each round some vertices may send packets to their neighbours.
- in each time step each node can send at most one packet to each of its neighbours
- Packets can buffer at nodes; send one by one over any edge they need to go
- Use FIFO for buffers
- Goal: Find a routing scheme for all the packets using minimum number of rounds
- Oblivious: decision for each packet is made locally and independent of other packets.
- Decision for each packet i at each node only depends on $d(i)$ (destination)

Bit fixing algorithm:

At each node $\sigma(i)$, given a packet with destination $d(i)$ find the left-most-bit they differ & send it to the neighbour that agrees with $d(i)$ on that bit.

e.g. $1011000101 \to 1111000101 \to 1111010101$

$\to 1111010001 \to 1111010011 \to 1111010010$

- without presence of other packets: $\leq d$ rounds.

Theorem: For any deterministic algorithm there are instances of packet routing that need $\Omega\left(\sqrt{\frac{N}{d}}\right)$ $\left(\text{i.e. } \Omega\left(\sqrt{\frac{2^d}{d}}\right)\right)$ steps.

- Our goal: $O(d)$ rounds.

Two phase (Randomized) algorithm

- Phase 1: Route packets to a random intermediate destination each (not a permutation!) in $4d$ rounds
- Phase 2: Route packets from intermediate locations to their destination in $4d$ rounds

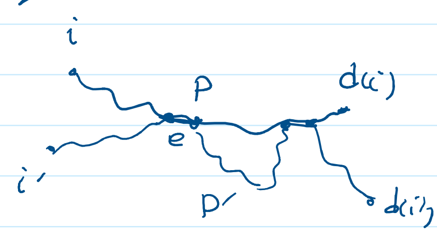
- let's consider phase 1 for now.

Definition: For each edge e, let $T(e)$ be the # of routes that use e.

- By symmetry $E[T(e)]$ is the same for all edges.
- $E[\text{length of each route}] = \frac{d}{2}$
- Since total number of routes is $N \longrightarrow$

$E[\text{Sum of lengths of routes}] = \frac{Nd}{2}$

$\xrightarrow{dN \text{ (directed) edges}} E[T(e)] = 1/2$



Fact 1: For any two routes P and P' that intersect (share edge e), once they separate they cannot intersect again.

- The proof is left as an (easy) exercise.

- let v_i be a fixed packet with route

$\rho_i = e_1, e_2, \ldots, e_k$

- S : set of packets that intersect v_i.

Lemma: The delay for v_i is $\leq |S|$.

Proof: We charge each delay for v_i to some packet in S.

- Def: a packet waiting at time t at e_j is having lag $t-j$
- total delay for v_i = the lag when crosses e_k
- Observation: when lag of v_i goes from l to $l+1$: Some packet from S using e_j instead of v_i and will leave P_i at this time.
- So some packet from S must reach lag l. Since some packet from S is to use the same edge as v_i at the same time.

- Say t' is the last time step any packet in S, has lag l
- Say $v \in S$ is ready to travel $e_{j'}$ at t': $t'-j'=l$
- Say $w \in S$ crosses $e_{j'}$ at time t' (w maybe v)
- For any packet waiting to cross $e_{j'}$: lag increased by 1.
- Observe: w will no longer use P_i; o.w. Some packet from S will use $e_{j'+1}$ at $t'+1$. s.
- We can charge increase in lag of v_i from l to $l+1$ to w.

- $H_{ij} = \begin{cases} 1 & \text{packet } v_i \text{ \& } v_j \text{ share an edge} \\ 0 & \text{o.w.} \end{cases}$

- D_i : delay for packet v_i ; $D_i = |S|$

$$E[D_i] = E\left[\sum_{j=1}^{N} H_{ij}\right] = \sum_{j=1}^{N} E[H_{ij}]$$

$$\leq \sum_{i=1}^{k} E[T(e_i)]$$

$$\leq \frac{k}{2} \leq \frac{d}{2}$$

- Since H_{ij}'s are independent, we can use Chernoff:

$$\Pr[D_i \geq 3d] < 2^{-3d}$$

Lemma: With probability at least $1-2^{-2d}$ each packet reaches their intermediate destination in $4d$ steps.

Proof: $N = 2^d$ packets ; using union bound d the probability of any delayed $> 3d$ steps is $\leq 2^d \cdot 2^{-3d} = 2^{-2d}$. ∎

- So with probability $\geq 1-2^{-2d}$ phase 1 is successful in $4d$ rounds.

- Using the same argument for phase 2.

- with probability $> 1 - \frac{1}{N}$ all packets are delivered in $8d$ steps.