Lecture Notes CS:5360 Randomized Algorithms
Lectures 14 and 15: Oct 9 and Oct 11, 2018
Scribe: Andrew Marmaduke

1 Proof of (Upper Tail) Chernoff Bounds

Theorem 1. *Let $X_1, X_2, \ldots, X_n$ be independent $0-1$ random variables with $Pr(X_i = 1) = p_i$. Let $X = \sum_{i=1}^{n} X_i$. Let μ denote $E[X]$. Then for any $\delta > 0$,*

$$Pr(X \geq (1+\delta)\mu) \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu$$

Proof: For arbitrary $t > 0$, $Pr(X \geq a) = Pr(e^{tx} \geq e^{ta})$. By Markov's Inequality, $Pr(e^{tx} \geq e^{ta}) \leq \dfrac{E[e^{tx}]}{e^{ta}}$. To get a good upper bound on the $Pr(X \geq a)$ we want to find a $t > 0$ that minimizes $\dfrac{E[e^{tx}]}{e^{ta}}$. In general, bounds of the form $Pr(X \geq a) \leq \min_{t>0} \dfrac{E[e^{tx}]}{e^{ta}}$ are called Chernoff Bounds. Now let us simplify $\dfrac{E[e^{tx}]}{e^{ta}}$. First, $e^{tx} = e^{t\sum_{i=1}^{n} X_i} = \prod_{i=1}^{n} e^{tX_i}$. Next, $E[e^{tx}] = E[\prod_{i=1}^{n} e^{tX_i}] = \prod_{i=1}^{n} E[e^{tX_i}]$. Note that we can pull the product out of the expectation because the $X_i's$ are independent. Additionally, it is sufficient to show that this final equality is actually $\leq$ and the proof will still go through. Now, with some algebra we have, $E[e^{tX_i}] = 1 + p_i(e^t - 1)$. Using the fact that $1 + x \leq e^x$ for all x we have, $1 + p_i(e^t - 1) \leq e^{p_i(e^t-1)}$. Thus, $E[e^{tx}] \leq \prod_{i=1}^{n} e^{p_i(e^t-1)} = e^{\sum_{i=1}^{n} p_i(e^t-1)} = e^{(e^t-1)\mu}$. This gives us, $Pr(X \geq a) \leq \frac{e^{(e^t-1)\mu}}{e^{ta}}$. We now plug in $a = (1+\delta)\mu$ to obtain $Pr(X \geq (1+\delta)\mu) \leq \dfrac{e^{(e^t-1)\mu}}{e^{t(1+\delta)\mu}} = \left(\dfrac{e^{e^t-1}}{e^{t(1+\delta)}}\right)^\mu$. Recall that we can pick any $t > 0$, but we want to minimize this expression. In order to do so, we pick $t = -\ln(1+\delta)$. Therefore, $\Pr(X \geq (1+\delta)\mu) \leq \left(\dfrac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu$. □

As an aside, it is useful to think about how the proof structure can apply to relaxed random variables, like $X_i's$ that take on the values 1 or -1, or $X_i's$ such that $0 \leq X_i \leq c_i$.

Now we will use the more general form (or the (a) form) proved above to derive the (b) and (c) forms. First, the (b) form:

Corollary 2. *For $0 < \delta \leq 1$, $Pr(X \geq (1+\delta)\mu) \leq e^{-\mu\delta^2/3}$*

Proof: We will show that for $0 < \delta \leq 1$, $\left(\dfrac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu \leq e^{-\mu\delta^2/3}$. In order to show this, we need

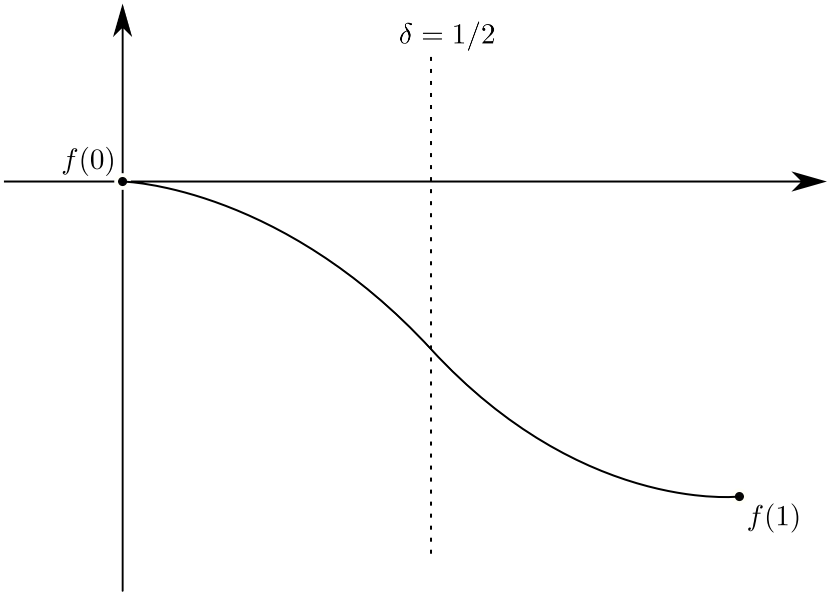

Figure 1: Sketch of how the function $f(\delta)$ behaves on the interval $[0, 1]$.

to show:

$$\begin{aligned}
\frac{e^\delta}{(1+\delta)^{1+\delta}} &\leq -\delta^2/3 \\
\delta - (1+\delta)\ln(1+\delta) &\leq -\delta^2/3 \\
\delta - (1+\delta)\ln(1+\delta) + \delta^2/3 &\leq 0
\end{aligned}$$

Let $f(\delta) = \delta - (1+\delta)\ln(1+\delta) + \delta^2/3$. We will show $f(\delta) \leq 0$ for $0 < \delta \leq 1$. Note that $f(0) = 0$, and $f(1) = 4/3 - 2\ln(2) < 0$. Next, $f'(\delta) = -\ln(1+\delta) + 2\delta/3$ with $f'(0) = 0$ and $f'(1) = 2/3 - \ln(2) < 0$. Finally, $f''(\delta) = \frac{-1}{1+\delta} + 2/3$ with $f''(0) = -1/3$ and $f''(1) > 0$. Thus, as δ increases the second derivative, $f''(\delta)$, increases with $f''(1/2) = 0$. Therefore, the function $f(\delta) \leq 0$, for $0 < \delta \leq 1$. □

Corollary 3. *For $R \geq 6\mu$, $Pr(X \geq R) \leq 2^{-R}$*

Proof: Let $R = (1+\delta)\mu$, so $\frac{R}{\mu} = 1 + \delta \geq 6$. Thus, $\delta \geq 5$. Note that $\left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu \leq$

$\left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu e^\mu$, because $\mu \geq 0$. Now we have:

$$\begin{aligned}
&\left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu e^\mu \\
&= \left(\frac{e}{1+\delta}\right)^{\mu(1+\delta)} \\
&\leq \left(\frac{e}{6}\right)^{\mu(1+\delta)} \\
&\leq (1/2)^{\mu(1+\delta)} \\
&= 2^{-R}
\end{aligned}$$

Therefore, $Pr(X \geq R) \leq 2^{-R}$ as desired. □

Note that these bounds on δ are not tight. Form (b) applies over a larger range and form (c) does not need to be as large as stated.

2 Applications

2.1 Randomized Routing

General Setting. We have a network of processors (or machines or nodes). Each node has some number of packets it wants to send to possibly different destinations. The goal is to route these packets in parallel so that time taken by the slowest packet is minimized. In particular, congestion and bandwidth, are the principle problems. To make the algorithm and analysis simpler we assume a synchronous system. We assume that nodes are executing a routing protocol over a directed network with an implicit queuing policy.

Typical step of routing protocol at node v.

1. receive packets sent along its incoming edges at previous step,
2. v performs some computations,
3. v sends packets along outgoing edges.

Note that only one packet can be sent along a directed edge at each step.

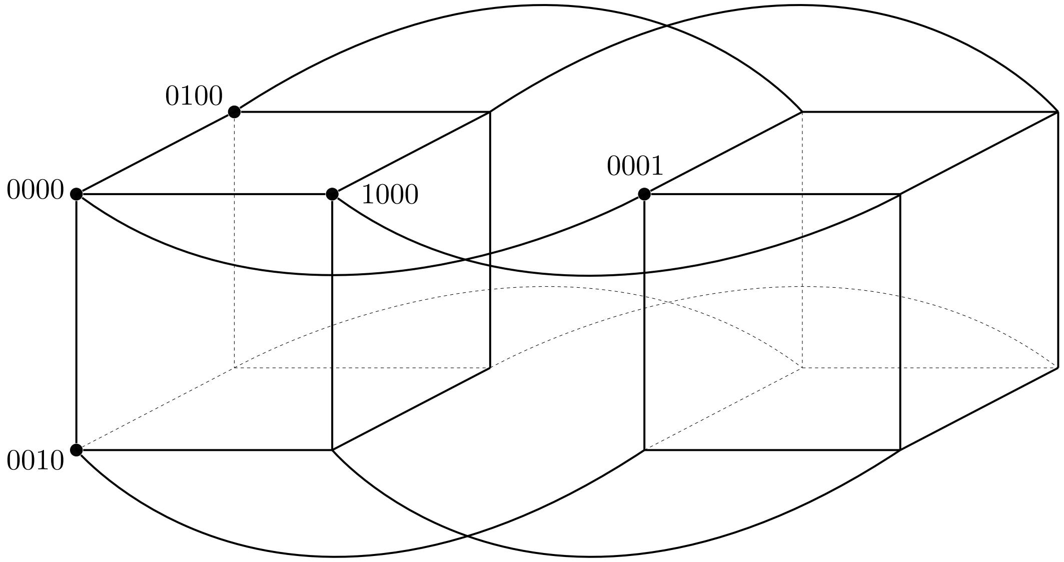

Figure 2: 4-dimensional hypercube with each vertex labeled with four bits.

Permutation Routing on a Hypercube.

Definition 4. *An n-dimensional hypercube contains $N = 2^n$ nodes, each with a distinct label from $\{0,1\}^n$ (or n-bit binary strings). Two nodes u and v are connected by an edge if label(u) and label(v) differ in exactly one position.*

Example. 5-Dimensional Hypercube Neighborhood

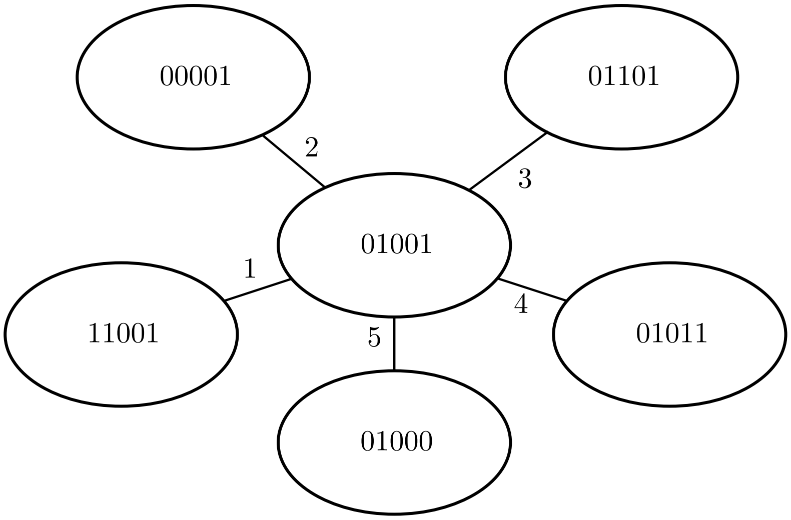

Figure 3: 5-dimensional hypercube neighborhood around vertex 01001.

Observation 5.

- *Each node in an n-dimensional hypercube has n neighbors*
- *The number of edges is $2^n \times \frac{n}{2}$*
- *The diameter of the network is n because all shortest paths have length $\leq n$.*

Definition 6. *Permutation routing is the routing problem in which each node holds a packet and the destinations of different packets are different.*

Example. 2-dimensional hypercube

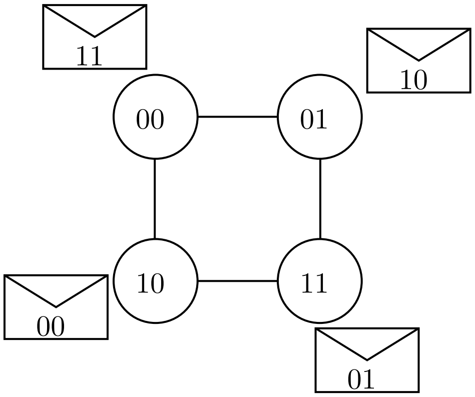

Figure 4: 2-dimensional hypercube with example packet routing.

Our goal is to solve the permutation routing problem on a hypercube.

Bit-fixing Algorithm. Suppose we have a packet at node $(a_1, a_2, \ldots, a_n)$ whose destination is $(b_1, b_2, \ldots, b_n)$.

Algorithm 1: Bit Fix Step(Dimension n, Packet Source a, Packet Destination b):

```
for $i \leftarrow 1$ to $n$ do
    if $a_i \neq b_i$ then
        send packet along edge in dimension $i$
    end
end
```

Example. Node 001011 holds packet with destination 111101.

001011, source
⇒ 101011, bit 1 fixed
⇒ 111011, bit 2 fixed
⇒ 111111, bit 4 fixed
⇒ 111101, bit 5 fixed, destination

We are using a FIFO queuing policy with arbitrary tie-breaking.

Observation 7. *The algorithm uses a shortest path for every packet.*

Lemma 8. *The bit-fixing algorithm requires* $\Omega(\sqrt{N}) = \Omega(2^{n/2})$ *steps.*

We will show an example permutation routing configuration to demonstrate this. The trouble is ultimately from node congestion. Let n be even. For each $\bar{a} = (a_1, a_2, \ldots, a_{n/2}, a_{n/2+1}, \ldots, a_n)$ let us use $\bar{a}_L$ to denote $(a_1, a_2, \ldots, a_{n/2})$ and $\bar{a}_R = (a_{n/2+1}, a_{n/2+2}, \ldots, a_n)$. For a packet at node $\bar{a} = \bar{a}_L \cdot \bar{a}_R$ make its destination $\bar{a}_R \cdot \bar{a}_L$. For example, a packet at 011001 will be sent to 001011.

source		intermediate node		destination
$\bar{a}_L \cdot \bar{a}_R$	$\rightsquigarrow$	$\bar{a}_R \cdot \bar{a}_R$	$\rightsquigarrow$	$\bar{a}_R \cdot \bar{a}_L$
$\bar{b}_L \cdot \bar{a}_R$	$\rightsquigarrow$	$\bar{a}_R \cdot \bar{a}_R$	$\rightsquigarrow$	$\bar{a}_R \cdot \bar{b}_L$

There are $2^{n/2}$ packets that reach node $\bar{a}_R \cdot \bar{a}_R$. Since each node has n edges, it takes $\Omega(2^{n/2}/n)$ steps for $\bar{a}_R \cdot \bar{a}_R$ to process all of these packets. This example can be polished to remove the division by n. This is the best, as far as deterministic protocols, that we know of.

Much Faster Randomized Protocol.

Phase I. Every packet is sent (using bit-fixing) to a randomly selected destination.

Phase II. Every packet is sent (using bit-fixing) from the intermediate random destination to its final destination.

Theorem 9. *With probability* $\geq 1 - \frac{1}{N}$ *all packets reach their destinations in* $O(n)$ *steps.*

Proof: Let us assume that Phase II is executed after Phase I is completed. Let M be a packet (or message). Let $T_1(M)$ denote the number of steps it takes for M to reach its Phase I destination. Let $X_1(e)$ denote the number of packets in Phase I that traverse edge e. Then, if M traverses path P in Phase I, $T_1(M) \leq \sum_{e \in P} X_1(e)$. The intuition behind this is that, worst case, M must wait at every edge for every packet that traverses that edge. Note that X_1 are not $0-1$ mutually independent random variables. We will bound the right hand side of this inequality. Let $T_1(P)$ denote $\sum_{e \in P} X_1(e)$.

Definition 10. *Let* $P = (v_0, v_1, \ldots, v_{m-1}, v_m)$ *be an arbitrary path of length* m*. Let the bit in position* j *be fixed along edge* (v_{i-1}, v_i)*. A packet* M *is said to be active for* v_{i-1} *if it reaches* v_{i-1} *before bit* j *is fixed.*

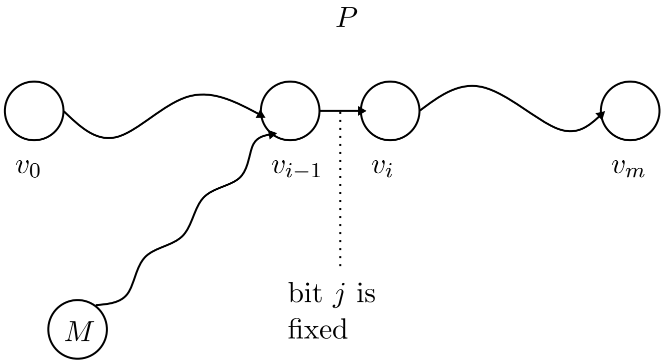

Figure 5: Example path with pair v_{i-1} and v_i for which M may be active.

Example. Suppose $v_{i-1} = 0000$ and $v_i = 0010$ so bit 3 is fixed. Also, that m is at source 1110 with destination 0001 (note that we're still in Phase I). If M's jth bit is fixed prior to v_{i-1}, then M will not travel to v_i. In other words, M would not be active for v_{i-1}.

Proof to be continued .... □