CSE 203A: Randomized Algorithms

October 13, 2017

Chernoff Bounds

For \( x = x_1 + x_2 + ... + x_n \),
when \( x_i \)'s are independent and \( x_i \in [0,1] \)

Let \( \mu = \mathbb{E}[x] \)
for \( \delta > 0 \),

\[
Pr((x > (1 + \delta)\mu) \leq \left( \frac{e^\delta}{(1 + \delta)^{1+\delta}} \right)^\mu \\

\text{for } 0 < \delta < 1,
\]

\[
Pr((x < (1 - \delta)\mu) \leq \left( \frac{e^{-\delta}}{(1 - \delta)^{1-\delta}} \right)^\mu
\]

Applications of Chernoff Bound:

Routing problem:

\( G = \) hypercube graph
\( V = \{0,1\}^n \)

- Edges are between vertices that differ in one coordinate
- Message in each vertex \( s \) going to vertex \( d(s) \) for some permutation \( d \)
- Each edge can only send 1 message per round

If all vertices send messages to the same vertex, there could be too much congestion. We need a protocol to send messages from start to the destination in not too many rounds. Here, we have a simple algorithm for doing so.

Randomized Routing:

For each \( s \) route, pick a random vertex \( u \), then route the message to \( u \) before going to \( d(s) \); \( s \rightarrow u \rightarrow d(s) \).

For each of the path, each routing is performed by naive bit fixing. Check the bits from the first bit to last bit, if the starting bit is different from the destination, we fix that bit by moving along the edge. Otherwise, the bit stay as it is and we’ll move on to check the next bits.
Analysis of Randomized Routing

First, let’s start with a baby problem for this:

The only way for this algorithm to fail is when there’s an edge that has to transmit too many messages. We start with edges \( b_1 b_2 ... b_n \).

\[
\begin{align*}
  b_1 b_2 ... b_n &\xrightarrow{\text{flip the } k^{th} \text{ bit}} b_1 b_2 ... \bar{b}_k ... b_n
\end{align*}
\]

Let’s count the number of message routed across the edge \( e \) in the half routing \((s \rightarrow u)\),

**When does \((s \rightarrow u)\) use \( e \)?**

by our algorithm, naive bit fixing, every bit after the \( k^{th} \) (not bit-fixed) bit needs to agree with \( s \). Similarly, every bit up to the \( k^{th} \) bit (bit-fixed) needs to agree with \( u \).

\[
\begin{align*}
  s_k &= b_k, s_{k+1} = b_{k+1}, ..., s_n = b_n \\
  b_1 &= u_1, b_2 = u_2, ..., \bar{b}_k = u_k
\end{align*}
\]

To compute \( E[x] \) we need to count the number of \( s, u \) pairs that agree with the above conditions.

Since we have to fix all but the first \( k-1 \) fixed bits, there are \( 2^{k-1} \) possible \( s \)'s that could possibly use that edge.

And for \( u \), the first \( k \) bits need to be specified. We have a probability of \( 2^{-k} \) that \( u \) agrees.

Probability of starting at \( s \) and end up using the edge \( e \) are independent for different \( s \)'s.

Let \( x = \text{number of routes using the edge } e = \text{sum of independent indicator random variables} \)

\[
E[x] = 2^{k-1} \times 2^{-k} = 1/2
\]

We can use Chebyshev bound to show that \( x \) is not too large. However, we have to bound this over \( n \times 2^n \) vertices, thus, we need a better tail bound. In this case, we’ll need to use Chernoff Bound.

\[
Pr(x > m\mu) \leq \left( \frac{e^m}{m^m} \right)^{1/2}
\]

when \( m = n \) assuming \( n \) is sufficiently large, we can show that,

\[
Pr(x > n) \leq 2^{-5n}
\]

Using the Union Bound over \( n \times 2^n \) edges,

\[
Pr(\text{any edge used in } > n \text{ routes}) \leq n \times 2^n \times 2^{-5n} = n \times 2^{-4n}
\]

We can say that, with high probability, no edge is in \( > n \) routes in the 1st half \((s \rightarrow u)\). Similarly, with high probability, no edge is in \( > n \) routes in the 2nd half \((u \rightarrow d(s))\). Combine together, there’s a high probability that no edge is used in \( > 2 \) routes.

From the claim above, we need to somehow connect to our problem – to show that the message arrived in not too many rounds.

**Lemma:**

\# of rounds to deliver a given message \( \leq \# \) of edges in its path + \# of messages that share an edge on its path
Proof:
To analyze this, let’s consider this claim

**Claim:**

The value of,

\[
\text{# of edges left to follow} + \text{# of other messages that still overlap} - \text{# of distinct vertices used by overlapping messages}
\]

decreases by at least 1 per round

Let’s consider the situation where the message \(m\) is at one of the vertices, and we are attempting to deliver \(m\). There could be some other messages that is currently sharing the same path and causes \(m\) to stuck. Also, there could be another message that joins the path in the middle and follows the path for a while before leaving the path. By our naive bit fixing algorithm, once a message leaves the path, it never comes back.

For every round,

- Message \(m\) advances or
- Message \(m\) waits on the queue, another message on the same vertex advances, creating a new vertex along that path that is not used
- There could be some \(v\) on the path that becomes unoccupied during the rounds. However, when this happens, somewhere near the end of the path becomes occupied or we advance one message at the end of the chain

Now, we can use the Chernoff bound to compute the probability of the message not being delivered in a timely manner.

For the 1st half of our routing,

\[
Pr(s \rightarrow u \text{ not delivered after } 10n \text{ rounds }) \leq Pr(s \rightarrow u \text{ has } > 9n \text{ conflicting messages})
\]

Fix \(u\), count the conflicting messages. For each other starting vertex \(s'\), the chance \(s'\) conflicts with \(s\) is independent.

The number of conflicting messages is then equal to a sum of independent indicator variables.

\[
E[\text{the number of conflicting messages}] \leq \sum_{\text{edges in path}} E[\text{# of messages using } e] \leq \sum 1/2 = n/2
\]

Note that \(E[\text{# of messages using } e]\) is slightly less than 1/2 because we are no longer counting \(s\). We can show that we can increase \(x\) by a constant to the make such value becomes exactly 1/2.

\[
Pr(x > 18(n/2)) \leq \left(\frac{18^{18}}{e^{18}}\right)^{-n/2} \leq (2^{2*18})^{-n/2} \leq 2^{-18n}
\]

\[
Pr(\text{any message takes } > 10n \text{ round in its 1st half}) \leq 2^n * 2^{-18n} = 2^{-17n}
\]

We can say that, with high probability, all messages are delivered within 20n rounds.
Generalizations/Extensions of Chernoff Bounds:

There are a couple ways to generalize this.

The bounds are based on \( \mu, \sigma \). We need a small support or at least most of the elements have small support. The problem is, when one of the random variables has a decent probability of being very far from its mean, the sum would have a probability of being very far away from its mean because this random variable could get this extraordinary large value. Fortunately, there’s a way to generalize this, so we can get a better bound when the variance is small. This type of bound is called Bernstein’s Inequality.

Since \( \Sigma \) independent random variables is a strong assumption. Another way to generalize is, If we can assume Martingale instead, we still have a Chernoff-like bound by using Azuma’s Inequality.

*Martingale: \( \mathbb{E}[x_k|x_1, x_2, ..., x_{k-1}] = x_{k-1} \)