CMPUT 675: Approximation Algorithms **Fall 2014**

Lecture 12 (Sep 29): Chernoff Bounds and MINIMIZING CONGESTION

Lecturer: Zachary Friggstad *Scribe: Bradley Hauer*

12.1 Chernoff Bounds

Theorem 1 *Let $X_1, \ldots, X_n$ be independent $\{0,1\}$ random variables $X_1, \ldots, X_n$. Let $X = \sum_{i=1}^n X_i$. Then for any $\delta > 0$ and any $U \geq \mathrm{E}[X]$:*

$$\Pr[X \geq (1+\delta) \cdot U] < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^U.$$

If $0 < \delta \leq 1$:

$$\Pr[X \geq (1+\delta) \cdot U] < e^{-U\delta^2/3}.$$

Proof. Say $\Pr[X_i = 1] = p_i$ for each $1 \leq i \leq n$.

Now to prove the first statement. Consider any value $t > 0$ (we will set it to a particular value later):

$$\Pr[X > (1+\delta)U] = \Pr[e^{t \cdot X} \geq e^{t \cdot (1+\delta) \cdot U}] \leq \frac{\mathrm{E}[e^{t \cdot X}]}{e^{t \cdot (1+\delta) \cdot U}}$$

This holds because $\Pr[X \geq a] \leq \frac{\mathrm{E}[X]}{a}$, which is equivalent to Markov's inequality.

Working with the numerator, we have

$$\mathrm{E}[e^{t \cdot X}] = E\left[\prod_{i=1}^n e^{t \cdot X_i}\right] = \prod_{i=1}^n \mathrm{E}[e^{t \cdot X_i}]$$

since the X_i are independent. Continuing with the argument,

$$\prod_{i=1}^n \mathrm{E}[e^{t \cdot X_i}] = \prod_{i=1}^n ((1-p_i) + p_i \cdot e^t) = \prod_{i=1}^n (1 + p_i(e^t - 1)) \leq \prod_{i=1}^n e^{p_i(e^t-1)}$$

since $1 + x \leq e^x$ for $x \geq 0$. Further,

$$\prod_{i=1}^n e^{P_i(e^t-1)} = e^{(e^t-1)\sum_i p_i} = e^{(e^t-1)\mathrm{E}[X]} \leq e^{(e^t-1) \cdot U}$$

So far, we have show

$$\Pr[X \geq (1+\delta)U] \leq \frac{e^{(e^t-1) \cdot U}}{e^{t \cdot (1+\delta) \cdot U}}.$$

Setting $t := \ln(1+\delta) > 0$ to minimize this expression, we see it is bounded by $\left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^U$.

When $\delta \leq 1$, we show

$$\left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^U \leq e^{-U \cdot \delta^2/3}$$

Note that it holds for $\delta = 0$ and $\delta = 1$. Take logarithms of both sides; note that the left-hand side is concave in $\delta \in (0, 1)$ and the right-hand side is linear. Therefore, the inequality must hold over all $\delta \in [0, 1]$. ■

Other Chernoff-style bounds, such as those appearing in the text, are proven with a similar strategy: apply Markov's inequality to an exponential function in $\sum_i X_i$ and use independence.

12.2 Minimizing Congestion

In the MINIMIZING CONGESTION problem, we are given a directed graph $G = (V, E)$, and $(s_1, t_1), \ldots, (s_k, t_k)$ pairs of nodes. For each i from 1 to k, we must select a path P_i from s_i to t_i. The goal is to minimize the *congestion* of the paths found, where the congestion of a set of paths is the maximum number of times any single edge appears in a path, i.e. minimize $\max_{e \in E}(\# i \text{ s.t. } e \in P_i)$

This problem is NP-Hard in general even for $k = 2$, as the problem of determining if there are edge-disjoint paths P_1, P_2 connecting the respective pairs is NP-complete [FHW80].

In the special case where all s_i are identical, we can solve the problem efficiently as it is equivalent to maximum flow. That is, add an auxiliary sink node $\bar{t}$ to the graph and connect each t_i to $\bar{t}$ with a new capacity 1 edge. Finally, find the smallest integer c such that if we set the remaining edge capacities to c then the graph supports k units of flow from the common s_i node to t. Since the maximum flow in a flow network with integer capacities can be taken to be integral, this corresponds to a collection of k paths connecting s_i to each of the sink nodes $\{t_1, \ldots, t_k\}$.

We now want to formulate MINIMIZING CONGESTION as a linear program. Let $\mathcal{P}_i$ be the set of all $s_i - t_i$ paths. Note that the $\mathcal{P}_i$ may be exponential in the size of the input. For each i from 1 to k, and each path $P \in \mathcal{P}_i$, let x_P^i be a variable indicating pair i uses path P. An LP formulation is as follows:

$$
\begin{array}{lrcll}
\text{minimize :} & W & & & \\
\text{subject to :} & \displaystyle\sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i \text{ s.t. } e \in P} x_p^i & \leq & W & \text{for each edge } e \in E \\
& \displaystyle\sum_{P \in \mathcal{P}_i} x_P^i & = & 1 & \text{for each } 1 \leq i \leq k \\
& W & \geq & 1 & \\
& \mathbf{x} & \geq & 0 &
\end{array}
$$

The first set of constraints ensures that W is not less than the congestion of the solution. The second set ensures that exactly one path between each $s_i - t_i$ path is chosen. The constraint $W \geq 1$ is necessary, as otherwise the integrality gap can be as bad as n even when $k = 1$: consider the following instance.

$$V = \{v_1, \ldots, v_n\}, \quad E = \{(v_1, v_i), (v_i, v_n) : 2 \leq i \leq n - 1\}, \quad (s_1, t_1) = (v_1, v_n)$$

Certainly $OPT = 1$ but the LP can get away by selecting each of the paths $\{(v_1, v_i), (v_i, v_n))$ to the extent of $1/(n - 2)$ and setting $W = 1/(n - 2)$.

This formulation may have size exponential in the size of the input as there can be exponentially many $s_i - t_i$ paths. However, there is an equivalent LP with polynomial size (in the sense that there is a natural correspondence between their LP solutions). This will be an assignment question.

We can use this LP to create an approximation algorithm for MINIMIZING CONGESTION. Note that the constraints $\mathbf{x} \geq 0$ and $\sum_{P \in \mathcal{P}_i} x_P^i = 1$ suggest a natural probability distribution over the paths $P \in \mathcal{P}_i$ for each i. The algorithm simply samples from this distribution for each i to get the required paths.

Algorithm 1 Randomized Rounding for MINIMIZING CONGESTION

Solve the LP, and let $(\mathbf{x}^*, W^*)$ be an optimal solution.

Independently, for each i from 1 to k, randomly sample one path from $\mathcal{P}_i$ from the distribution given by $\Pr[P \in \mathcal{P}_i \text{ selected}] = x_p^{*i}$.

end

Consider the following $\{0,1\}$ random variables. For each $e \in E, 1 \leq i \leq k$ let Y_e^i indicate the event that the path chosen for pair i uses edge e. For each $1 \leq i \leq k$ and each $P \in \mathcal{P}_i$, let Z_P^i be the random variable that is 1 if path P was chosen to connect pair i. For $e \in E$, let Y_e denote the congestion of e.

Thus, we have $Y_e = \sum_{i=1}^k Y_e^i$ for each $e \in E$ and we also have $Y_e^i = \sum_{P \in \mathcal{P}_i} Z_P^i$ for each $1 \leq i \leq k$ and each $e \in E$. Finally, note that the maximum congestion on any edge is $\max_{e \in E} Y_e$.

Lemma 1 *For any edge $e \in E$, $\Pr[Y_e > 18 \cdot \ln(n) \cdot W^*] \leq \frac{1}{n^3}$*

Proof. Note that:

$$\begin{aligned}
\mathrm{E}[Y_e] &= E\left[\sum_i \sum_{P \in \mathcal{P}_i \text{ s.t. } e \in P} Z_p^i\right] \\
&= \sum_i \sum_{P \in \mathcal{P}_i \text{ s.t. } e \in P} \mathrm{E}[Z_p^i] \\
&= \sum_i \sum_{P \in \mathcal{P}_i \text{ s.t. } e \in P} x_p^{*i} \\
&\leq W^*
\end{aligned}$$

We now use a Chernoff bound. Note that for this edge e, the variables $Y_e^1, \ldots, Y_e^k$ are $\{0,1\}$ random variables. Furthermore, they are independent because we samples the path for each pair independently. Set $U = 9 \cdot \ln(n) \cdot W^*$, and $\delta = 1$.

$$\begin{aligned}
\Pr[Y_e \geq (1+\delta) \cdot U] &\leq e^{-U \cdot \delta^2 / 3} \\
&= e^{-3 \cdot \ln(n) \cdot W^*} && \text{(Definitions of } U \text{ and } \delta.) \\
&= n^{-3 \cdot W^*} \\
&\leq 1/n^3 && \text{(Since } W^* \geq 1.)
\end{aligned}$$

■

Now we can prove the main result for this algorithm. When we say "with high probability", we mean with probability that approaches 1 as the size of the instance grows.

Theorem 2 *With high probability, the congestion of this solution is $\leq 18 \cdot \ln(n) \cdot W^*$.*

Proof. Continuing from the lemma, by the union bound, we have:

$$\begin{aligned}
\Pr[Y_e > 18 \cdot \ln(n) \cdot W^* \text{ for some } e \in E] &\leq \sum_{e \in E} \Pr[Y_e > 18 \cdot \ln(n) \cdot W^*] \\
&\leq |E|/n^3 \\
&\leq 1/n && (|E| \leq n^2)
\end{aligned}$$

That is, the maximum congestion is at most $18 \cdot \ln n \cdot W^*$ with probability at least $1 - 1/n$. ■

12.2.1 A Tighter Bound

We used the simpler form of the Chernoff bound for simplicity. In fact, we can show that the maximum congestion is in fact $O(\log n / \log\log n)$ with high probability by using the sharper form of the Chernoff bound: $\Pr[X \geq (1+\delta) \cdot U] < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^U$.

To see this, set $U = W^*$ and $\delta = 5\frac{\log n}{\log\log n} - 1$. For large enough n, working through the calculations shows the probability that $Y_e > 5\frac{\log n}{\log\log n}W^*$ is small enough so that taking a union bound over all edges shows the maximum congestion is at most $5\frac{\log n}{\log\log n}W^*$ with high probability.

This rounding algorithm is due to Raghavan and Thompson [RT87] who were the first to introduce the idea of randomized rounding of linear programs. Interestingly, this algorithm is essentially the best possible. Unless NP $\subseteq$ ZPTIME($n^{O(\log\log n)}$ there is no $o(\log n / \log\log n)$-approximation for MINIMIZING CONGESTION in directed graphs [C+07]. In undirected graphs, the best lower bound is currently $\Omega(\log\log n / \log\log\log n)$ [AZ05].

This is a stronger assumption than P $\neq$ NP and asserts there is no *randomized* algorithms that can decide, say, SAT in *expected* running time $n^{O(\log\log n)}$. That is, such algorithms never return an incorrect answer, but the running time is a random variable that is *quasi-polynomial* in expectation. This is stronger than saying P $\neq$ NP, but it is still an open problem and many would find it surprising if SAT could be decided with such an algorithm.

References

AZ05 M. Andrews, L. Zhang, Hardness of the undirected congestion minimization problem, In Proceedings of *ACM Symposioum on Theory of Computing*, 2005.

C+07 J. Chuzhoy, V. Guruswami, S. Khanna, and K. Talwar, Hardness of routing with congestion in directed graphs, In Proceedings of *ACM Symposium on Theory of Computing*, 2007.

FHW80 S. Fortune, J. E. Hopcroft, and J. Wyllie, The directed subgraph homeomorphism problem, Theoretical Computer Science, 10(2):111-121, 1980.

RT87 P. Raghavan and C. D. Thompson, Randomized rounding: a technique for provably good algorithms and algorithmic proofs, *Combinatorica*, 7:365–374, 1987.