**CSE 525: Randomized Algorithms** **Spring 2023**

# Lecture 4: Strong Concentration Bounds

*Lecturer: Shayan Oveis Gharan* *10/13/2022*

**Disclaimer**: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

We have seen how knowledge of the variance of a random variable $X$ can be used to control deviation of $X$ from its mean. This is the heart of the second moment method. But often we can control even higher moments, and this allows us to obtain much stronger concentration properties. A prototypical example is when $X_1, X_2, \ldots, X_n$ is a family of independent (but not necessarily identically distributed) $\{0, 1\}$ random variables and $X = X_1 + X_2 + \cdots + X_n$. Let $p_i = \mathbb{E}[X_i]$ and define $\mu = \mathbb{E}[X] = \sum_{i=1}^{n} p_1 + p_2 + \cdots + p_n$. In that case, we have the following multiplicative form of the "Chernoff bound".

**Theorem 6.1** (Multiplicative Chernoff bound)**.** . *For every $\delta \geq 0$, it holds that*

$$\mathbb{P}[X \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}.$$

*and*

$$\mathbb{P}[X < (1-\delta\mu] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu}$$

*Consequently,*

$$\mathbb{P}[X \geq (1+\delta)\mu] \leq e^{-\delta^2\mu/(2+\delta}, \mathbb{P}[X \leq (1-\delta)\mu] \leq e^{-\delta^2\mu/2}$$

*Proof.* Let $t$ be a parameter that we choose later.

$$\mathbb{P}[X \geq (1+\delta)\mu] = \mathbb{P}\left[e^{tX} \geq e^{t(1+\delta)\mu}\right] \underset{\text{Markov's Inequality}}{\leq} \frac{\mathbb{E}\left[e^{tX}\right]}{e^{t(1+\delta)\mu}}. \tag{6.1}$$

The first inequality uses that the exponential function is a monotone function.

Now, we can write

$$\mathbb{E}\left[e^{tX}\right] = \mathbb{E}\left[e^{t\sum_i X_i}\right] = \mathbb{E}\left[\prod_{i=1}^{n} e^{tX_i}\right] \underset{\text{independence}}{=} \prod_{i=1}^{n} \mathbb{E}\left[e^{tX_i}\right].$$

Now, observe that

$$\mathbb{E}\left[e^{tX}\right] = p_i e^t + (1-p_i) = 1 + p_i(e^t - 1) \underset{1+x\leq e^x}{\leq} e^{p_i(1-e^t)}$$

Plugging this back we obtain

$$\mathbb{E}\left[e^{tX}\right] \leq \prod_{i=1}^{n} e^{p_i(e^t-1)} = e^{\mu(1-e^t)}$$

Putting back in (6.1), we obtain

$$\mathbb{P}[X \geq (1+\delta)\mu] \leq \frac{e^{\mu(e^t-1)}}{e^{t(1+\delta)\mu}} = e^{\mu(e^t-1-(1+\delta)t)} \underset{\text{set } t=\ln(1+\delta)}{=} \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}$$

The other case can be proven similarly. □

## 6.1 Congestion Minimization Problem

A classical technique in the field of approximation algorithms is to write down a linear programming relaxation of a combinatorial problem. The linear program (LP) is then solved in polynomial time, and one rounds the fractional solution to an integral solution that is, hopefully, not too much worse than the optimal solution.

A classical example goes back to Raghavan and Thompson [**RT87**]. Let $G = (V, A)$ be a directed network, and suppose that we are given a sequence of terminal pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$ where $\{s_i\}, \{t_i\} \subseteq V$. The goal is to choose, for every $i$, a directed $s_i$-$t_i$ path $P_i$ in $G$ so as to minimize the maximum congestion of an arc $e \in A$:

$$OPT = \min\{\max_{e \in A} \#\{i : e \in P_i\}\}$$

This problem is NP-hard. Our goal will be to design an approximation algorithm that outputs a solution so that the congestion of every edge is at most $\alpha \cdot OPT$, for $\alpha$ as small as possible. The number $\alpha$ is called the approximation factor of our algorithm. We will see that for this problem we will be able to achieve $\alpha = O(\frac{\log n}{\log \log n})$.

We start by writing a linear programming relaxation for this problem. Let $\mathcal{P}_i$ be the set of (directed) paths from $s_i$ to $t_i$ and let $\mathcal{P} = \cup_i \mathcal{P}_i$. For every path $P$, we have a variable $x_P$ to denote the amount of flow that we route along $P$.

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} y_P = 1 && \forall 1 \leq i \leq k \\ & \sum_{P \in \mathcal{P} : e \in P} y_P \leq t && \forall e \in A \\ & y_P \geq 0 && \forall P \end{aligned} \tag{6.2}$$

A few observations are in order:

- $OPT(LP) \leq OPT$. This is simply because the optimum solution is a feasible solution in the above program. Note that the optimum solution satisfies all of the above constraints with the additional property that $y_P \in \{0, 1\}$ for all paths $P$.
- Although the above program has exponentially many variable one for every directed path connecting $s_i \to t_i$ (for all $i$), its optimum solution can be computed in polynomial time. To do that we need two observations:
  i) We can write a linear program to find a flow of value 1 from $s_i$ to $t_i$. We have a variable $f_e^{(i)}$ to denote the flow of every edge.

  $$\begin{aligned} & \sum_{s_i \to e} f_e^{(i)} = 1 \\ & \sum_{e \to v} f_e^{(i)} = \sum_{v \to e} f_e^{(i)} && \forall v \neq s_i, t_i \\ & f_e^{(i)} \geq 0 && \forall e. \end{aligned}$$

  Having that, the congestion of $e$ due to the flow routed between the $i$-th pair is $f_e^{(i)}$; so the total congestion of $e$ is $\sum_i f_e^{(i)}$.

  ii) A (fractional) flow (of value 1) from $s_i$ to $t_i$ can be decomposed into a distribution of paths from $s_i$ to $t_i$. To see that, given the solution $\{f_e^{(i)}\}_{e \in A}$, greedily find a path $P$ from $s_i$ to $t_i$ on edges with positive flow; let $y_P$ be $\min_{e\in P} f_e^{(i)}$. Then, subtract $y_P$ from the flow of all edges along $P$. We will obtain a new flow of value $1 - y_P$ from $s_i$ to $t_i$. So we repeat this procedure until we get the 0 flow.

## 6.2 Independent Rounding

Given a solution $y$ to (6.2), we want to round it to an integral solution. Namely, we want to choose exactly one path from each $\mathcal{P}_i$ such that the union of the chosen paths have small congestion, at most $\alpha OPT$.

We follow the independent rounding method. Recall that, by feasiblity of $y$, for every $1 \leq i \leq k$, we know that $\sum_{P\in\mathcal{P}_i} y_P = 1$. So, we can think of $\{y_P\}_{P\in\mathcal{P}_i}$ as providing a probability distribution over $s_i$-$t_i$ paths. For every $i$, independently, we choose one of the paths $P \in \mathcal{P}_i$ with probability $y_P$. This procedure, by definition, gives a feasible set of directed paths from $s_i$ to $t_i$ for all $i$. So, it remains to bound the maximum congestion. We prove the following theorem.

**Theorem 6.2.** *With probability at least $1-1/n$ the above algorithm produces a integral set of paths connecting all terminals with maximum congest at most*

$$C\frac{\log n}{\log\log n}OPT$$

Let $Y_P$ be the indicator random variable that $P$ is chosen. For an edge $e$, let $X_e$ be the random variable that is the congestion of edge $e$. So,

$$X_e = \sum_{P\in\mathcal{P}:e\in P} Y_P.$$

By linearity of expectations,

$$\mathbb{E}\left[X_e\right] = \sum_{P\in\mathcal{P}:e\in P} y_P \leq OPT(LP).$$

So, the expectations are small. We just need to use a Chernoff bound/union bound argument. Unfortunately, the random variables $Y_P$ are not independent. So, we need to use a slightly different random variables that are truly independent.

The idea is to note that we always have exactly one path from $s_i$ to $t_i$. So, let $Y_{e,i}$ be the indicator random variable that the unique path from $s_i$ to $t_i$ uses edge $e$. So, $X_e = \sum_i Y_{e,i}$. It follows that $Y_{e,i}$'s are independent. Let $\beta = (1+\delta)\frac{OPT}{OPT(LP)}$, and note that $\beta > 1$ since $OPT \geq OPT(LP)$. So, by Chernoff bound,

$$\mathbb{P}\left[X_e \geq (1+\delta)OPT\right] = \mathbb{P}\left[X_e \geq \beta\cdot OPT(LP)\right] \leq \left(\frac{e^\beta}{\beta^\beta}\right)^{OPT(LP)} \leq \left(\frac{e^{1+\delta}}{(1+\delta)^{1+\delta}}\right)^{OPT} \leq \left(\frac{e^{1+\delta}}{(1+\delta)^{1+\delta}}\right)$$

where in the last inequality we simply use that $OPT \geq 1$. Now, to get the strong concentration bound we need to choose $\delta$ large enough such that the RHS is at most $n^{-3}$. It turns out that for that purpose it is enough to let $1+\delta = C\frac{\log n}{\log\log n}$ for a large enough constant $C > 1$.

Since, $G$ has at most $n^2$ edges, $|A| \leq n^2$, by union bound

$$\mathbb{P}\left[\exists e : X_e \geq (1+\delta)OPT\right] \leq n^2\cdot n^{-3} \leq \frac{1}{n}.$$

So, the algorithm succeeds with probability at least $1 - 1/n$.

## 6.3 Future Works and Open Problems

Chuzhoy, Guruswami, Khanna and Talwar showed that min-congestion problem is NP-hard to approximate within any factor better than $\frac{\log n}{\log \log n}$ when the underlying graph is directed. Note that the same algorithm that we discussed here works if the underlying graph is undirected (we can just put two copies of every edge one in every direction). However, for undirected, the result of Raghavan-Thompson is still the best known approximation factor. The best hardness result is $\log \log n$ by Andrews-Zhang. It is a fundamental open problem in the field of network routing to beat the Raghavan-Thompson's classical algorithm.