Approximation Algorithm

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 09

Lecture 42

Lecture 42 : Chernoff Bound (Contd.)

Welcome. So, in the last class we have stated and proved Chernoff bounds, we have stated two variations, but remember that there are various other forms of Chernoff bound, but they convey the same message that it is if the sum of n independent random variables and they are all bounded they are not very they do not take unbounded values their expectations are bounded and so on. Or in the basic form they take only finite values then it is quite likely very likely that the some random variable $X = \sum_{i=1}^{n} X_i$ its value is within expectation of X in multiplicative term with very high probability close to 1 probability. So, let us continue Chernoff bound and see some more of its useful forms. So, Chernoff bound So, let us recall we have shown this theorem let $X_1, \ldots, X_n$ be n independent random variable not necessarily identically distributed.

independent $\{0, a_i\}$ random variables. That means, they take value 0 or a_i these are the 2 possible values a_i s are greater than 0 and less than equal to 1 and suppose we have 2 numbers L lower bound on expectation of X and U upper bound of expectation of x, where $X = \sum_{i=1}^{n} X_i$ greater than 0 probability that X takes value greater than equal to $(1+\delta)U$ is less than equal to $\left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{U}$.

and probability that X takes value less than equal to $(1-\delta)L$ this is less than equal to $\left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{L}$ ok. is the Chernov bound in full generality in terms of this bounds upper bounds. And then we showed two useful to work with bounds for these terms which are this is less than equal to $e^{-U\delta^2/3}$, this holds for δ in between 0 and 1 both inclusive. And for the lower bound we have $e^{-L\frac{\delta^2}{2}}$ this holds for δ greater than equal to 0 and less than 1 ok. In particular this whole inequality holds for δ greater than 0 and less than 1.

So, we can remove this also. So, probability that X takes value than equal to $(1+\delta)U$ is less than equal to $e^{-U\delta^2/3}$ for δ greater than 0 and less than equal to 1, but the on the lower side probability that X is less than equal to $(1-\delta)L$ this is less than equal to $e^{-L\frac{\delta^2}{2}}$. This holds for δ greater than 0, but δ should be strictly less than 1. not for equal to 1, but often we need this inequality for equal to 1. So, let us prove what we can get for δ equal to 1 this lower bound.

So, for that let us prove this lemma again let $X_1,\ldots,X_n$ be n independent $\{0,a_i\}$ random variables where a i are greater than 0 and less than equal to 1 ok, then for $X=\sum_{i=1}^{n} X_i$ ok. If L is less than equal to expectation of X, then probability that X equal to 0, You see here if $\delta=1$ then probability that X is less than equal to 0, but X cannot take negative value. So, this probability that probability X is less than equal to $(1-\delta)L$ for δ equal to 0 is same as the probability that X equal to 0 this is less than e^{-L}. So, if you put δ equal to 1 here.

So, this is even a stronger inequality we get we need $e^{-\frac{L}{2}}$ we are proving something stronger which is e^{-L} ok. So, let us prove it. So, what is probability again let us assume probability that X_i equal to a_i is equal to p_i and this is not equal to 0 for all $i\in[n]$. Because if expectation of X is 0 then probability of probability if expectation of X equal to 0 then probability of X equal to 0 is 1. So, we need an assumption that each X_i takes value a_i with probability non with non 0 probability $X_i$takes a_i with non-zero probability for every $i\in[n]$ suppose that probability is p_i.

So, now, let us compute what is probability that X equal to 0 the only way X could be 0 is each X_i is 0. So, this is probability that X_i equal to 0 for all $i\in[n]$. Now here again we use independence of $X_1,\ldots,X_n$ to write this probability as product $i=1,\ldots,n$ probability X_i equal to 0 and what is probability X_i equal to 0 this is product $i=1,\ldots,n$. $1-p_i$. Now, applying a m g m inequality that geometric mean of n positive numbers is less than equal to arithmetic mean of n positive numbers.

This is less than equal to $\left(\frac{1}{n}\sum_{i=1}^{n}(1-p_i)\right)^n$. So, this is $\left(\frac{1}{n}\sum_{i=1}^{n}(1-p_i)\right)^n$. Now, because each a_i is less than equal to 1, you can pretend that p_i is $1\times p_i$. So, this is less than equal to because I have a negative sign here $1-\left((\frac{1}{n})\sum_{i=1}^{n} a_i p_i\right)^n$ ok. And so, what is $\sum a_i p_i$ this is nothing, but expectation of X.

So, this is $\left(1-\frac{E[X]}{n}\right)^n$. Now, use the fact that $1+x \leq e^x$ for all x. So, this is less than equal to e^{-x} is $\frac{E[X]}{n}$. this is so because $1+x \leq e^x$ for all real number x, this is $e^{-E[X]}$. but L is a lower bound of expectation of x.

So, this is e^{-L} which concludes the proof. So, now, we observe that e^{-L} is even smaller than $e^{-L\frac{\delta^2}{2}}$ for δ equal to 1, what we can write? We can write this inequality even for δ equal to 1. which is often useful corollary. Let $X_1, \ldots, X_n$ be n independent $\{0, a_i\}$ random variable, X_i takes value a_i with probability with nonzero probability for every $i \in [n]$, then for $X = \sum_{i=1}^{n} X_i$ δ greater than 0 less than equal to 1 ok and U is an upper bound of expectation of X. and L is a lower bound, then we have probability that X takes value greater than equal to $(1+\delta)U$ is less than equal to $e^{-U\delta^2/3}$ and probability that x takes value less than equal to $(1-\delta)L$ is less than equal to $e^{-L\frac{\delta^2}{2}}$.

So, this concludes our review of Chernoff bound. Let us again go back to algorithm design approximation algorithms by randomized rounding and see how this bounds can be used effectively. So, for that we consider the problem of integer multi commodity flow. integer multi commodity flows. So, what is the input? input is an undirected graph $G = (V, E)$ ok and k source destination pairs $\{s_1, t_1\}, \{s_2, t_2\}, \ldots, \{s_k, t_k\}$ arr k source destination pairs.

And, we want to send one unit of commodity from s_i to t_i along one path we need to send one commodity from s_i to t_i along an s_i to t_i path P_i ok. So, pictorially here is a graph G. and we have various sources $\{s_1, t_1\}, \{s_2, t_2\}, \ldots, \{s_k, t_k\}$ there is not be distinct s_1 may be same as t_{10} and so on they can be different they can be same s_i could be equal to t_j or s_i could be equal to s_j for i and $j \in [n]$.

And I need to find various paths they can share vertices they can share edges sorry this is s_1 to t_1 then I have to find s_2 to t_2 maybe this is the path and then maybe s_3 to t_3 is another path and so on ok. What is the goal? I define the congestion or the load of an edge is the number of paths that are using that edges number of paths among $P_1, \ldots, P_k$. Load of an edge E is the number of paths among $P_1, \ldots, P_k$ that use the edge E. ok and the congestion of the network is the maximum load of any path congestion of the network is the maximum load of any edge. So, the goal of the problem is to route these commodities to minimize the congestion of the network.

So, route k commodities to minimize the congestion of the network. So, this problem has various applications for example, in chip design. So, in chip design we need to connect various components of the chip using wires and there could be pathways and the bandwidth which is the bandwidth of an edge is the number of wires that passes through that edge. and the congestion of the network then corresponds to the bandwidth required to connect this source and destinations these various pairs of components. So, this is one applications and it has various other applications, it has it is known to be NP complete problem and we will see an approximation algorithm using randomized rounding technique.

So, for that here is a ILP formulation for that problem is we have a variable c for the congestion which you want to minimize. So, we have a variable c denoting congestion of the network and for every edge E and commodity $i \in [k]$, $c_{e,i}$ is 1 if edge e is used to send commodity i from s_i to t_i ok, otherwise it is 0. So, what we want to minimize? We want to minimize the congestion. So, minimize C subject to this flow variables. So, flow variables should satisfy there is no capacity constraint.

So, it only needs to satisfy conservation of flows. So, at every vertex v in so, for every commodity $i \in [k]$ the flow should be conserved for the i-th commodity at all vertices except s_i and t_i. Then for all vertex $v \in V \setminus \{s_i, t_i\}$ So, here is a vertex v total incoming flow for i-th commodity will should be same as the total outgoing flow. So, $\sum_{(u,v)\in E} c_{(u,v),i}$ if (u, v, i) should be same as total outgoing flows $\sum_{(v,w)\in E} c_{(v,w),i}$ this is the flow conservation and at source and destination. So, for all at s_i the outgoing flow should be 1 and at t_i incoming flow should be 1.

So, for all $(s_i, v) \in E$ $f_{s_i,v,i}$ this sum should be 1 which is the total outgoing flow for ith commodity at s_i should be same as the total incoming flow for i-th commodity. So, these are the things and if for every age and every commodity this should be either 0 or 1, this is for every age e and for every commodity $i \in [k]$. So, this is the exact formulation integer linear programming formulation, we relax this integrality requirement f_{e_i} to lay to lie in between 0 and 1. So, greater than 0 and less than equal to 1 and take it as a homework that I can get rid of these inequalities without affecting the optimum. So, I get rid of these inequalities.

So, I only keep f_{e_i} is greater than equal to 1 for every edge E and for every commodity i. So, this is the relaxed LP. In the next class we will see how we can use this relaxed LP to design a beautiful randomized rounding based algorithm for integer multi-commodity flow. It is an approximation algorithm with small approximation ratio ok. Let us stop here. Thank you.