Lecture 4: Concentration Inequalities

Nicolás Rivera John Sylvester Luca Zanetti Thomas Sauerwald

Lent 1920

UNIVERSITY OF CAMBRIDGE

The useful Chernoff Bounds

Remember from last class.

Nicer Chernoff Bounds

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum p_i$. Then,

- For all $t > 0$,

$$\mathbf{P}[X \geq \mathbf{E}[X] + t] \leq e^{-2t^2/n}$$

$$\mathbf{P}[X \leq \mathbf{E}[X] - t] \leq e^{-2t^2/n}$$

- For $0 < \delta < 1$,

$$\mathbf{P}[X \geq (1+\delta)\mathbf{E}[X]] \leq \exp\left(-\frac{\delta^2 \mathbf{E}[X]}{3}\right)$$

$$\mathbf{P}[X \leq (1-\delta)\mathbf{E}[X]] \leq \exp\left(-\frac{\delta^2 \mathbf{E}[X]}{2}\right)$$

Outline

Randomised QuickSort

Extension of Chernoff Bounds

Examples

Applications: QuickSort

Quick sort is a sorting algorithm that works as following.

Algorithm: QuickSort

Input: Array of different number A.
Output: array A sorted in increasing order

- Pick an element from the array, the so-called pivot .
- If $|A| = 0$ or $|A| = 1$; return A.
- Else
  - Generate two subarrays A_1 and A_2:
    A_1 contains the elements that are smaller than the pivot ;
    A_2 contains the elements that are greater than the pivot ;
  - Recursively sort A_1 and A_2.

Applications: QuickSort

Quick sort is a sorting algorithm that works as following.

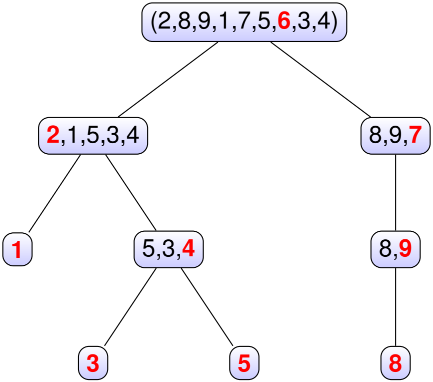
Algorithm: QuickSort

Input: Array of different number A.
Output: array A sorted in increasing order

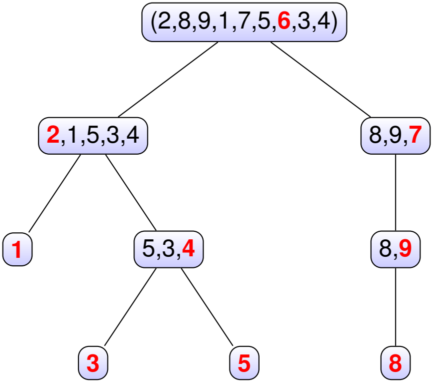
- Pick an element from the array, the so-called pivot .
- If $|A| = 0$ or $|A| = 1$; return A.
- Else
  - Generate two subarrays A_1 and A_2:
    A_1 contains the elements that are smaller than the pivot ;
    A_2 contains the elements that are greater than the pivot ;
  - Recursively sort A_1 and A_2.

E.g. Let $A = (2, 8, 9, 1, 7, 5, 6, 3, 4)$, choose 6 as pivot, then we get $A_1 = (2, 1, 5, 3, 4)$ and $A_2 = (8, 9, 7)$.
It is well-known that the worst-case complexity (number of comparisons) of quick sort is $O(n^2)$. This happens when pivots are pretty bad, generating one large array and one small array.

Applications: QuickSort



Applications: QuickSort



Note that the number of comparison performed in quick sort is equivalent to the sum of the height of all nodes in the tree. In this case

$$0 + 1 + 1 + 2 + 2 + 2 + 3 + 3 + 3 = 17.$$

Applications: QuickSort

How to pick a good pivot ? we don't, **just pick one at random.**

Applications: QuickSort

How to pick a good pivot ? we don't, **just pick one at random.**

This should be your standard answer in this course

Applications: QuickSort

How to pick a good pivot ? we don't, **just pick one at random.**

This should be your standard answer in this course

Let's analyse quicksort with random pivots.

Applications: QuickSort

How to pick a good pivot ? we don't, **just pick one at random.**

This should be your standard answer in this course

Let's analyse quicksort with random pivots.

1. Consider n different number, wlog, $\{1, \ldots, n\}$

Applications: QuickSort

How to pick a good pivot ? we don't, **just pick one at random.**

> This should be your standard answer in this course

Let's analyse quicksort with random pivots.

1. Consider n different number, wlog, $\{1, \ldots, n\}$
2. let H_i be the last level where i appears in the tree. Then the number of comparison is $H = \sum_{i=1}^{n} H_i$

Applications: QuickSort

How to pick a good pivot ? we don't, **just pick one at random.**

This should be your standard answer in this course

Let's analyse quicksort with random pivots.

1. Consider n different number, wlog, $\{1, \ldots, n\}$
2. let H_i be the last level where i appears in the tree. Then the number of comparison is $H = \sum_{i=1}^{n} H_i$
3. we will prove that exists $C > 0$ such that

$$\mathbf{P}[\forall i, H_i \leq C \log n] \geq 1 - 1/n$$

Applications: QuickSort

How to pick a good pivot ? we don't, **just pick one at random.**

> This should be your standard answer in this course

Let's analyse quicksort with random pivots.

1. Consider n different number, wlog, $\{1, \ldots, n\}$
2. let H_i be the last level where i appears in the tree. Then the number of comparison is $H = \sum_{i=1}^{n} H_i$
3. we will prove that exists $C > 0$ such that

$$\mathbf{P}[\forall i, H_i \leq C \log n] \geq 1 - 1/n$$

4. actually, we will prove something equivalent but easier: we will prove that all leaves of the tree are at distance at most $C \log n$ from the root with probability at least $1 - 1/n$.

Applications: QuickSort

How to pick a good pivot ? we don't, **just pick one at random.**

This should be your standard answer in this course

Let's analyse quicksort with random pivots.

1. Consider n different number, wlog, $\{1, \ldots, n\}$
2. let H_i be the last level where i appears in the tree. Then the number of comparison is $H = \sum_{i=1}^{n} H_i$
3. we will prove that exists $C > 0$ such that

$$\mathbf{P}[\forall i, H_i \leq C \log n] \geq 1 - 1/n$$

4. actually, we will prove something equivalent but easier: we will prove that all leaves of the tree are at distance at most $C \log n$ from the root with probability at least $1 - 1/n$.
5. then $H = \sum_{i=1}^{n} H_i \leq Cn \log n$, with probability at least $1 - 1/n$.

Applications: QuickSort

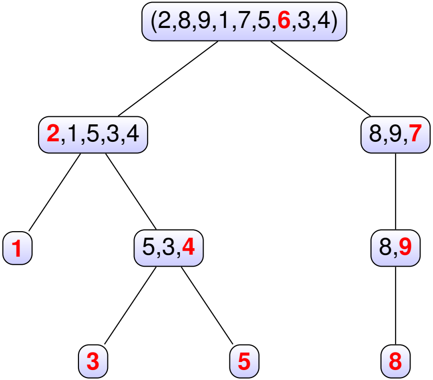
- Let P be a path from the root to a leaf. A node in P is called **good** if the corresponding pivot partition the array into two subarrays each of size at least $1/3$ of the previous one, the node is **bad** otherwise.

Applications: QuickSort

- Let P be a path from the root to a leaf. A node in P is called **good** if the corresponding pivot partition the array into two subarrays each of size at least $1/3$ of the previous one, the node is **bad** otherwise.
- Denote by s_t the size of the array at level t in P.

Applications: QuickSort

- Let P be a path from the root to a leaf. A node in P is called **good** if the corresponding pivot partition the array into two subarrays each of size at least $1/3$ of the previous one, the node is **bad** otherwise.
- Denote by s_t the size of the array at level t in P.



E.g. Path: $(2, 8, 9, 1, 7, 5, \mathbf{6}, 3, 4) \to (\mathbf{2}, 1, 5, 3, 4) \to (5, 3, \mathbf{4}) \to (\mathbf{5})$
The vertices are: good, bad, good
$s_0 = 9, s_1 = 5, s_2 = 3, s_3 = 1.$

Applications: QuickSort

- Let P be a path from the root to a leaf. A node in P is called **good** if the corresponding pivot partition the array into two subarrays each of size at least $1/3$ of the previous one, the node is **bad** otherwise.
- Denote by s_t the size of the array at level t in P.

Applications: QuickSort

- Let P be a path from the root to a leaf. A node in P is called **good** if the corresponding pivot partition the array into two subarrays each of size at least $1/3$ of the previous one, the node is **bad** otherwise.
- Denote by s_t the size of the array at level t in P.
- After a good vertex we have that $s_t \leq (2/3)s_{t-1}$.

Applications: QuickSort

- Let P be a path from the root to a leaf. A node in P is called **good** if the corresponding pivot partition the array into two subarrays each of size at least $1/3$ of the previous one, the node is **bad** otherwise.
- Denote by s_t the size of the array at level t in P.
- After a good vertex we have that $s_t \leq (2/3)s_{t-1}$.
- Therefore, there are at most $T = \frac{\log n}{\log(3/2)} \leq 2 \log n$ good nodes in a path P,

Applications: QuickSort

- Let P be a path from the root to a leaf. A node in P is called **good** if the corresponding pivot partition the array into two subarrays each of size at least $1/3$ of the previous one, the node is **bad** otherwise.
- Denote by s_t the size of the array at level t in P.
- After a good vertex we have that $s_t \leq (2/3)s_{t-1}$.
- Therefore, there are at most $T = \frac{\log n}{\log(3/2)} \leq 2 \log n$ good nodes in a path P,
- Set $C = 21$ and suppose that $|P| > C \log n$.

Applications: QuickSort

- Let P be a path from the root to a leaf. A node in P is called **good** if the corresponding pivot partition the array into two subarrays each of size at least $1/3$ of the previous one, the node is **bad** otherwise.
- Denote by s_t the size of the array at level t in P.
- After a good vertex we have that $s_t \leq (2/3)s_{t-1}$.
- Therefore, there are at most $T = \frac{\log n}{\log(3/2)} \leq 2 \log n$ good nodes in a path P,
- Set $C = 21$ and suppose that $|P| > C \log n$.
- this implies that the number of bad vertices in the first $21 \log n$ nodes is more than $19 \log n$.

- Consider the first $\lfloor 21 \log n \rfloor$ vertices of P. Denote by $X_i = 1$ if the node at height i of P is bad, and $X_i = 0$ if it is good. Let $X = \sum_{i=1}^{\lfloor 21 \log n \rfloor} X_i$.

- Consider the first $\lfloor 21 \log n \rfloor$ vertices of P. Denote by $X_i = 1$ if the node at height i of P is bad, and $X_i = 0$ if it is good. Let $X = \sum_{i=1}^{\lfloor 21 \log n \rfloor} X_i$.
- Note that the X_i's are independent and $\mathbf{P}[\, X_i = 1\,] = 2/3$, and $\mathbf{E}[\, X\,] = (2/3)21 \log n = 14 \log n$. Then, by the (nicer) Chernoff Bounds

- Consider the first $\lfloor 21 \log n \rfloor$ vertices of P. Denote by $X_i = 1$ if the node at height i of P is bad, and $X_i = 0$ if it is good. Let $X = \sum_{i=1}^{\lfloor 21 \log n \rfloor} X_i$.
- Note that the X_i's are independent and $\mathbf{P}[\, X_i = 1\,] = 2/3$, and $\mathbf{E}[\, X\,] = (2/3)21 \log n = 14 \log n$. Then, by the (nicer) Chernoff Bounds

$$\mathbf{P}[\, X > \mathbf{E}[\, X\,] + t\,] \leq e^{-2t^2/n}$$

- Consider the first $\lfloor 21 \log n \rfloor$ vertices of P. Denote by $X_i = 1$ if the node at height i of P is bad, and $X_i = 0$ if it is good. Let $X = \sum_{i=1}^{\lfloor 21 \log n \rfloor} X_i$.
- Note that the X_i's are independent and $\mathbf{P}[\, X_i = 1 \,] = 2/3$, and $\mathbf{E}[\, X \,] = (2/3)21 \log n = 14 \log n$. Then, by the (nicer) Chernoff Bounds

$$\mathbf{P}[\, X > \mathbf{E}[\, X \,] + t \,] \leq e^{-2t^2/n}$$

$$\begin{aligned}\mathbf{P}[\, X > 19 \log n \,] = \mathbf{P}[\, X > \mathbf{E}[\, X \,] + 5 \log n \,] &\leq e^{-2(5 \log n)^2/(21 \log n)} \\ &= e^{-(50/21) \log n} \leq 1/n^2.\end{aligned}$$

- Consider the first $\lfloor 21 \log n \rfloor$ vertices of P. Denote by $X_i = 1$ if the node at height i of P is bad, and $X_i = 0$ if it is good. Let $X = \sum_{i=1}^{\lfloor 21 \log n \rfloor} X_i$.
- Note that the X_i's are independent and $\mathbf{P}[\, X_i = 1\,] = 2/3$, and $\mathbf{E}[\, X\,] = (2/3)21 \log n = 14 \log n$. Then, by the (nicer) Chernoff Bounds

$$\mathbf{P}[\, X > \mathbf{E}[\, X\,] + t\,] \leq e^{-2t^2/n}$$

$$\begin{aligned} \mathbf{P}[\, X > 19 \log n\,] = \mathbf{P}[\, X > \mathbf{E}[\, X\,] + 5 \log n\,] &\leq e^{-2(5 \log n)^2/(21 \log n)} \\ &= e^{-(50/21) \log n} \leq 1/n^2. \end{aligned}$$

- Hence, we conclude the path has more than $21 \log n$ nodes with probability at most n^{-2}. There are at most n leaves, then by union bound , the probability that at least one path has more than $21 \log n$ nodes is n^{-1}

Applications: QuickSort

Remarks

- It is know that no sorting algorithm based on comparison takes less than $\Omega(n \log n)$

Applications: QuickSort

Remarks

- It is know that no sorting algorithm based on comparison takes less than $\Omega(n \log n)$
- The constant C can be improved a little bit, but in any case we will obtain that our randomised version of QuickSort that whp compares $O(n \log n)$ pairs

Applications: QuickSort

Remarks

- It is know that no sorting algorithm based on comparison takes less than $\Omega(n \log n)$
- The constant C can be improved a little bit, but in any case we will obtain that our randomised version of QuickSort that whp compares $O(n \log n)$ pairs
- It is possible to deterministically choose the best pivot that divide the array into two subarrays of the same size.

Applications: QuickSort

Remarks

- It is know that no sorting algorithm based on comparison takes less than $\Omega(n \log n)$
- The constant C can be improved a little bit, but in any case we will obtain that our randomised version of QuickSort that whp compares $O(n \log n)$ pairs
- It is possible to deterministically choose the best pivot that divide the array into two subarrays of the same size.
- The later requires to compute the median of the array in linear time, which is not easy to do

Applications: QuickSort

Remarks

- It is know that no sorting algorithm based on comparison takes less than $\Omega(n \log n)$
- The constant C can be improved a little bit, but in any case we will obtain that our randomised version of QuickSort that whp compares $O(n \log n)$ pairs
- It is possible to deterministically choose the best pivot that divide the array into two subarrays of the same size.
- The later requires to compute the median of the array in linear time, which is not easy to do
- Randomised solution for QuickSort is much easier to implement.

Outline

Randomised QuickSort

Extension of Chernoff Bounds

Examples

Chernoff Bound: Extension to other Random Variables

- Most of the time we will use Chernoff Bounds for sum of independent Bernoulli random variables

Chernoff Bound: Extension to other Random Variables

- Most of the time we will use Chernoff Bounds for sum of independent Bernoulli random variables
- but not always

Chernoff Bound: Extension to other Random Variables

- Most of the time we will use Chernoff Bounds for sum of independent Bernoulli random variables
- but not always
- it does not hurt to know how to derive similar bounds for other random variables

Chernoff Bound: Extension to other Random Variables

- Most of the time we will use Chernoff Bounds for sum of independent Bernoulli random variables
- but not always
- it does not hurt to know how to derive similar bounds for other random variables

Chernoff Bound: Extension to other Random Variables

- Most of the time we will use Chernoff Bounds for sum of independent Bernoulli random variables
- but not always
- it does not hurt to know how to derive similar bounds for other random variables

Remember the key steps:

Chernoff Bound recipe

1. Let $\lambda > 0$, then

$$\mathbf{P}[\, X \geq (1+\delta)\mu \,] \leq e^{-\lambda(1+\delta)\mu} \mathbf{E}\left[\, e^{\lambda X} \,\right]$$

2. Compute an upper bound for $\mathbf{E}\left[\, e^{\lambda X} \,\right]$
3. Optimise the value of $\lambda > 0$.

Exercise:

- Let X be a Poisson random variable of mean μ. Prove that

$$\mathbf{E}\left[e^{\lambda X} \right] = e^{\mu(e^{\lambda}-1)}$$

and deduce that for $t \geq \mu$

$$\mathbf{P}[\, X \geq t \,] \leq e^{-\mu} \left(\frac{e\lambda}{t} \right)^{t} \quad \text{and} \quad \mathbf{P}[\, X \geq (1+\delta)\mu \,] \leq e^{-\delta^2 \mu},$$

and the corresponding lower tails.

Exercise:

- Let X be a Poisson random variable of mean μ. Prove that

$$\mathbf{E}\left[e^{\lambda X} \right] = e^{\mu(e^{\lambda}-1)}$$

and deduce that for $t \geq \mu$

$$\mathbf{P}[X \geq t] \leq e^{-\mu}\left(\frac{e\lambda}{t}\right)^{t} \quad \text{and} \quad \mathbf{P}[X \geq (1+\delta)\mu] \leq e^{-\delta^2\mu},$$

and the corresponding lower tails.

- Let X be a Normal random variable of mean μ and variance σ^2. Prove that

$$\mathbf{E}\left[e^{\lambda X} \right] = e^{\mu\lambda+\sigma^2\lambda^2/2},$$

and deduce that for $t > \mu$

$$\mathbf{P}[X \geq t] \leq e^{-(t-\mu)^2/2}.$$

Hoeffding's Extension

- Beside sums of independent Bernoulli Random variables, sums of independent and bounded random variables is very important in applications.

Hoeffding's Extension

- Beside sums of independent Bernoulli Random variables, sums of independent and bounded random variables is very important in applications.
- Unfortunately the distribution of the X_i will be unknown or very hard to compute, thus it will be very hard to compute the moment-generating function of X_i.

Hoeffding's Extension

- Beside sums of independent Bernoulli Random variables, sums of independent and bounded random variables is very important in applications.
- Unfortunately the distribution of the X_i will be unknown or very hard to compute, thus it will be very hard to compute the moment-generating function of X_i.
- Hoeffding's Lemma helps us here

Hoeffding's Extension

- Beside sums of independent Bernoulli Random variables, sums of independent and bounded random variables is very important in applications.
- Unfortunately the distribution of the X_i will be unknown or very hard to compute, thus it will be very hard to compute the moment-generating function of X_i.
- Hoeffding's Lemma helps us here

Hoeffding's Extension Lemma

Let X be a random variable with mean 0 such that $a \leq X \leq b$, then for all $\lambda \in \mathbb{R}$.

$$\mathbf{E}\left[e^{\lambda X}\right] \leq \exp\left(\frac{(b-a)^2\lambda^2}{8}\right)$$

Hoeffding's Extension

- Beside sums of independent Bernoulli Random variables, sums of independent and bounded random variables is very important in applications.
- Unfortunately the distribution of the X_i will be unknown or very hard to compute, thus it will be very hard to compute the moment-generating function of X_i.
- Hoeffding's Lemma helps us here

You can always consider $X' = X - \mathbf{E}[X]$

Hoeffding's Extension Lemma

Let X be a random variable with mean 0 such that $a \leq X \leq b$, then for all $\lambda \in \mathbb{R}$.

$$\mathbf{E}\left[e^{\lambda X}\right] \leq \exp\left(\frac{(b-a)^2\lambda^2}{8}\right)$$

Hoeffding's Extension

- Beside sums of independent Bernoulli Random variables, sums of independent and bounded random variables is very important in applications.
- Unfortunately the distribution of the X_i will be unknown or very hard to compute, thus it will be very hard to compute the moment-generating function of X_i.
- Hoeffding's Lemma helps us here

You can always consider $X' = X - \mathbf{E}[X]$

Hoeffding's Extension Lemma

Let X be a random variable with mean 0 such that $a \leq X \leq b$, then for all $\lambda \in \mathbb{R}$.

$$\mathbf{E}\left[e^{\lambda X}\right] \leq \exp\left(\frac{(b-a)^2\lambda^2}{8}\right)$$

We will not study the proof of this Lemma

Chernoff-Hoeffding Bounds

Chernoff-Hoeffding's Bounds

Let $X_1, \ldots, X_n$ be independent random variable with mean μ_i such that $a_i \leq X_i \leq b_i$. Let $X = X_1 + \ldots + X_n$, and let $\mu = \mathbf{E}[\,X\,] = \sum_{i=1}^n \mu_i$. Then for any $t > 0$

$$\mathbf{P}[\,X \geq \mu + t\,] \leq \exp\left[\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right]$$

and

$$\mathbf{P}[\,X \leq \mu - t\,] \leq \exp\left[\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right]$$

Chernoff-Hoeffding Bounds

Chernoff-Hoeffding's Bounds

Let $X_1, \ldots, X_n$ be independent random variable with mean μ_i such that $a_i \leq X_i \leq b_i$. Let $X = X_1 + \ldots + X_n$, and let $\mu = \mathbf{E}[\,X\,] = \sum_{i=1}^n \mu_i$. Then for any $t > 0$

$$\mathbf{P}[\,X \geq \mu + t\,] \leq \exp\left[\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right]$$

and

$$\mathbf{P}[\,X \leq \mu - t\,] \leq \exp\left[\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right]$$

Proof:

- Let $X_i' = X_i - \mu_i$ and $X' = X_1' + \ldots, X_n'$, then $\mathbf{P}[\,X \geq \mu + t\,] = \mathbf{P}[\,X' \geq t\,]$

Chernoff-Hoeffding Bounds

Chernoff-Hoeffding's Bounds

Let $X_1, \ldots, X_n$ be independent random variable with mean μ_i such that $a_i \leq X_i \leq b_i$. Let $X = X_1 + \ldots + X_n$, and let $\mu = \mathbf{E}[\,X\,] = \sum_{i=1}^{n} \mu_i$. Then for any $t > 0$

$$\mathbf{P}[\,X \geq \mu + t\,] \leq \exp\left[\frac{-2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right]$$

and

$$\mathbf{P}[\,X \leq \mu - t\,] \leq \exp\left[\frac{-2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right]$$

Proof:

- Let $X_i' = X_i - \mu_i$ and $X' = X_1' + \ldots, X_n'$, then $\mathbf{P}[\,X \geq \mu + t\,] = \mathbf{P}[\,X' \geq t\,]$
- $\mathbf{P}[\,X' \geq t\,] \leq e^{-\lambda t} \prod_{i=1}^{n} \mathbf{E}\left[\,e^{\lambda X_i'}\,\right] \leq \exp\left[-\lambda t + \frac{\lambda^2}{8} \sum_{i=1}^{n}(b_i - a_i)^2\right]$

Chernoff-Hoeffding Bounds

Chernoff-Hoeffding's Bounds

Let $X_1, \ldots, X_n$ be independent random variable with mean μ_i such that $a_i \leq X_i \leq b_i$. Let $X = X_1 + \ldots + X_n$, and let $\mu = \mathbf{E}[\, X \,] = \sum_{i=1}^n \mu_i$. Then for any $t > 0$

$$\mathbf{P}[\, X \geq \mu + t \,] \leq \exp\left[\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right]$$

and

$$\mathbf{P}[\, X \leq \mu - t \,] \leq \exp\left[\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right]$$

Proof:

- Let $X_i' = X_i - \mu_i$ and $X' = X_1' + \ldots, X_n'$, then $\mathbf{P}[\, X \geq \mu + t \,] = \mathbf{P}[\, X' \geq t \,]$
- $\mathbf{P}[\, X' \geq t \,] \leq e^{-\lambda t} \prod_{i=1}^n \mathbf{E}\left[\, e^{\lambda X_i'} \,\right] \leq \exp\left[-\lambda t + \frac{\lambda^2}{8} \sum_{i=1}^n (b_i - a_i)^2\right]$
- Choose $\lambda = \frac{4t}{\sum_{i=1}^n (b_i - a_i)^2}$ to get the result.

Chernoff-Hoeffding Bounds

Chernoff-Hoeffding's Bounds

Let $X_1, \ldots, X_n$ be independent random variable with mean μ_i such that $a_i \leq X_i \leq b_i$. Let $X = X_1 + \ldots + X_n$, and let $\mu = \mathbf{E}[\, X \,] = \sum_{i=1}^n \mu_i$. Then for any $t > 0$

$$\mathbf{P}[\, X \geq \mu + t \,] \leq \exp\left[\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2} \right]$$

and

$$\mathbf{P}[\, X \leq \mu - t \,] \leq \exp\left[\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2} \right]$$

Proof:

- Let $X_i' = X_i - \mu_i$ and $X' = X_1' + \ldots, X_n'$, then $\mathbf{P}[\, X \geq \mu + t \,] = \mathbf{P}[\, X' \geq t \,]$
- $\mathbf{P}[\, X' \geq t \,] \leq e^{-\lambda t} \prod_{i=1}^n \mathbf{E}\left[\, e^{\lambda X_i'} \,\right] \leq \exp\left[-\lambda t + \frac{\lambda^2}{8} \sum_{i=1}^n (b_i - a_i)^2 \right]$
- Choose $\lambda = \frac{4t}{\sum_{i=1}^n (b_i - a_i)^2}$ to get the result.

This is not magic! you just need to optimise on λ

Chernoff-Bounds: Final Remarks

- There are several version of Chernoff-style Bounds that work for sum of independent random variables.
- The proof of all of them usually follows the same **recipe**
- Some bounds include more information about the random variables, e.g. the variance
- the limit is the amount of information we have about the random variables and our ability to manipulate/bound quantities.

Beyond sum of independent variables

Can we prove concentration of other type of random variables? Yes.. but

Beyond sum of independent variables

Can we prove concentration of other type of random variables? Yes.. but

- There is no general tool to prove concentration beyond the basic **recipe**

Beyond sum of independent variables

Can we prove concentration of other type of random variables? Yes.. but

- There is no general tool to prove concentration beyond the basic **recipe**
- but in general it is very hard to compute moment generating functions

Beyond sum of independent variables

Can we prove concentration of other type of random variables? Yes.. but

- There is no general tool to prove concentration beyond the basic **recipe**
- but in general it is very hard to compute moment generating functions
- It is worth trying to transform the problem into the setting of sum of independent random variable
- There is one more very important bound

Method of Bounded Differences

Suppose, we have random variables $X_1, \dots, X_n$. We want to study the random variable

$$f(X_1, \dots, X_n)$$

Method of Bounded Differences

Suppose, we have random variables $X_1, \ldots, X_n$. We want to study the random variable

$$f(X_1, \ldots, X_n)$$

Some examples:

1. $X = X_1 + \ldots + X_n$

Method of Bounded Differences

Suppose, we have random variables $X_1, \ldots, X_n$. We want to study the random variable

$$f(X_1, \ldots, X_n)$$

Some examples:

1. $X = X_1 + \ldots + X_n$
2. In balls into bins, X_i indicate where ball i is allocated, and $f(X_1, \ldots, X_m)$ is the number of empty bins

Method of Bounded Differences

Suppose, we have random variables $X_1, \ldots, X_n$. We want to study the random variable

$$f(X_1, \ldots, X_n)$$

Some examples:

1. $X = X_1 + \ldots + X_n$
2. In balls into bins, X_i indicate where ball i is allocated, and $f(X_1, \ldots, X_m)$ is the number of empty bins
3. X_i indicates if the i-th edge belongs to a graph G, and $f(X_1, \ldots, X_m)$ represent the number of connected components of G

Method of Bounded Differences

Suppose, we have random variables $X_1, \ldots, X_n$. We want to study the random variable

$$f(X_1, \ldots, X_n)$$

Some examples:

1. $X = X_1 + \ldots + X_n$
2. In balls into bins, X_i indicate where ball i is allocated, and $f(X_1, \ldots, X_m)$ is the number of empty bins
3. X_i indicates if the i-th edge belongs to a graph G, and $f(X_1, \ldots, X_m)$ represent the number of connected components of G

We can simply prove concentration of X around it means by the so-called Method of Bounded Differences

Method of Bounded Differences

A function f is called Liptchitz of parameter $\mathbf{c} = (c_1, \ldots, c_n)$ if for all i

$$|f(x_1, x_2, \ldots, x_{i-1}, \boldsymbol{x_i}, x_{i+1}, \ldots, x_n) - f(x_1, x_2, \ldots, x_{i-1}, \boldsymbol{y_i}, x_{i+1}, \ldots, x_n)| \leq c_i$$

where x_i and y_i are in the domain of the i-th coordinate

Method of Bounded Differences

A function f is called Liptchitz of parameter $\mathbf{c} = (c_1, \ldots, c_n)$ if for all i

$$|f(x_1, x_2, \ldots, x_{i-1}, \mathbf{x_i}, x_{i+1}, \ldots, x_n) - f(x_1, x_2, \ldots, x_{i-1}, \mathbf{y_i}, x_{i+1}, \ldots, x_n)| \leq c_i$$

where x_i and y_i are in the domain of the i-th coordinate

McDiarmid's inequality

Let $X_1, \ldots, X_n$ be independent random variables. Let f be Liptchitz of parameter $\mathbf{c} = (c_1, \ldots, c_n)$. Let $X = f(X_1, \ldots, X_n)$. Then

$$\mathbf{P}[\, X - \mathbf{E}[\, X \,] \geq t \,] \leq \exp\left(-\frac{2t^2}{\sum c_i^2}\right)$$

and

$$\mathbf{P}[\, X - \mathbf{E}[\, X \,] \leq -t \,] \leq \exp\left(-\frac{2t^2}{\sum c_i^2}\right)$$

We will not study the Proof of McDiarmid's Inequality

Outline

Randomised QuickSort

Extension of Chernoff Bounds

Examples

Examples: Balls into Bins

- Consider m balls assigned uniformly at random into n bins.
- Enumerate the balls from 1 to m. Ball i is assigned to a random bin X_i.
- Let Z be the number of empty bins (after assigning the balls)
- $Z = f(X_1, \ldots, X_m)$ and f is Liptchitz with $\mathbf{c} = (1, \ldots, 1)$ (because if we move one ball to another bin, the number of empty bins changes at most in 1)
- By the McDiarmid's inequality

$$\mathbf{P}[\,|F - \mathbf{E}[\,F\,]\,| > t\,] \leq 2e^{-2t^2/m}$$

Example: Bin Packing

Consider the Bin Packing problem

1. We are given n items of sizes in the unit interval $[0, 1]$

Example: Bin Packing

Consider the Bin Packing problem

1. We are given n items of sizes in the unit interval $[0, 1]$
2. We want to pack those items into the fewest number of unit-capacity bins as possible

Example: Bin Packing

Consider the Bin Packing problem

1. We are given n items of sizes in the unit interval $[0, 1]$
2. We want to pack those items into the fewest number of unit-capacity bins as possible
3. Suppose that the item sizes X_i are independent random variables in the interval $[0, 1]$
4. let $B = B(X_1, \ldots, X_n)$ the optimal number of bins that suffice to pack the items

Example: Bin Packing

Consider the Bin Packing problem

1. We are given n items of sizes in the unit interval $[0, 1]$
2. We want to pack those items into the fewest number of unit-capacity bins as possible
3. Suppose that the item sizes X_i are independent random variables in the interval $[0, 1]$
4. let $B = B(X_1, \ldots, X_n)$ the optimal number of bins that suffice to pack the items
5. The Lipschitz conditions holds with $\boldsymbol{c} = (1, \ldots, 1)$, **Why?**

Example: Bin Packing

Consider the Bin Packing problem

1. We are given n items of sizes in the unit interval $[0, 1]$
2. We want to pack those items into the fewest number of unit-capacity bins as possible
3. Suppose that the item sizes X_i are independent random variables in the interval $[0, 1]$
4. let $B = B(X_1, \dots, X_n)$ the optimal number of bins that suffice to pack the items
5. The Lipschitz conditions holds with $\boldsymbol{c} = (1, \dots, 1)$, **Why?**
6. Therefore

$$\mathbf{P}[\, B - \mathbf{E}[\, B\,] \geq t\,]\,, \mathbf{P}[\, B - \mathbf{E}[\, B\,] \leq -t\,] \leq e^{-2t^2/n}.$$

A random distance problem

Consider an n by n square grid $\{0, 1, \ldots, n\}^2$, where each point is connected to each of its (at most) four neighbours (N, S, E, W). Within each inner square of the grid, we draw a diagonal from SW to NE with probability p.

A random distance problem

Consider an n by n square grid $\{0, 1, \ldots, n\}^2$, where each point is connected to each of its (at most) four neighbours (N, S, E, W). Within each inner square of the grid, we draw a diagonal from *SW* to *NE* with probability p.

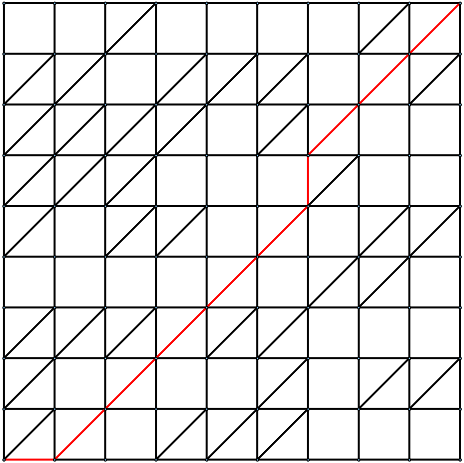
We say that $(0, 0)$ is on the bottom left corner and (n, n) in the top right corner.

A random distance problem

Consider an n by n square grid $\{0, 1, \ldots, n\}^2$, where each point is connected to each of its (at most) four neighbours (N, S, E, W). Within each inner square of the grid, we draw a diagonal from SW to NE with probability p.

We say that $(0, 0)$ is on the bottom left corner and (n, n) in the top right corner.

Can we prove concentration of the shortest path from $(0, 0)$ to (n, n)?



A random distance problem

Can we prove concentration of the shortest path from $(0, 0)$ to (n, n)?

A random distance problem

Can we prove concentration of the shortest path from $(0, 0)$ to (n, n)?
Yes! Let Z be the total length of the shortest path.Two options

A random distance problem

Can we prove concentration of the shortest path from $(0, 0)$ to (n, n)?
Yes! Let Z be the total length of the shortest path.Two options

1. Define $X_{ij} = 1$ if there is a diagonal in square ij, otherwise $X_{ij} = 0$. Then $Z = f(X_{11}, \ldots, X_{nn})$ satisfies the Lipschitz conditions with $\boldsymbol{c} = (2 - \sqrt{2})(1, \ldots, 1)$,

A random distance problem

Can we prove concentration of the shortest path from $(0, 0)$ to (n, n)?
Yes! Let Z be the total length of the shortest path.Two options

1. Define $X_{ij} = 1$ if there is a diagonal in square ij, otherwise $X_{ij} = 0$. Then $Z = f(X_{11}, \ldots, X_{nn})$ satisfies the Lipschitz conditions with $\boldsymbol{c} = (2 - \sqrt{2})(1, \ldots, 1)$, **Why?** .

A random distance problem

Can we prove concentration of the shortest path from $(0, 0)$ to (n, n)?
Yes! Let Z be the total length of the shortest path.Two options

1. Define $X_{ij} = 1$ if there is a diagonal in square ij, otherwise $X_{ij} = 0$. Then $Z = f(X_{11}, \ldots, X_{nn})$ satisfies the Lipschitz conditions with $\boldsymbol{c} = (2 - \sqrt{2})(1, \ldots, 1)$, **Why?** .
   Then
   $$\mathbf{P}[\,|Z - \mathbf{E}[\,Z\,]\,| \geq t\,] \leq 2\exp\left[\frac{-t^2}{(2-\sqrt{2})^2 n^2}\right]$$
2. Enumerate the columns of squares from 1 to n. Let $Y_i = (X_{1i}, \ldots, X_{ni})$. Then $Z = g(Y_1, \ldots, Y_n)$. g satisfies the Lipschitz conditions with $\boldsymbol{c} = (2 - \sqrt{2})(1, \ldots, 1)$.

A random distance problem

Can we prove concentration of the shortest path from $(0,0)$ to (n,n)?
Yes! Let Z be the total length of the shortest path.Two options

1. Define $X_{ij} = 1$ if there is a diagonal in square ij, otherwise $X_{ij} = 0$. Then $Z = f(X_{11}, \ldots, X_{nn})$ satisfies the Lipschitz conditions with $\boldsymbol{c} = (2 - \sqrt{2})(1, \ldots, 1)$, **Why?** .
   Then
$$\mathbf{P}[\,|Z - \mathbf{E}[\,Z\,]\,| \geq t\,] \leq 2\exp\left[\frac{-t^2}{(2-\sqrt{2})^2 n^2}\right]$$
2. Enumerate the columns of squares from 1 to n. Let $Y_i = (X_{1i}, \ldots, X_{ni})$. Then $Z = g(Y_1, \ldots, Y_n)$. g satisfies the Lipschitz conditions with $\boldsymbol{c} = (2 - \sqrt{2})(1, \ldots, 1)$. **Why?**

A random distance problem

Can we prove concentration of the shortest path from $(0,0)$ to (n,n)?
Yes! Let Z be the total length of the shortest path.Two options

1. Define $X_{ij} = 1$ if there is a diagonal in square ij, otherwise $X_{ij} = 0$. Then $Z = f(X_{11}, \ldots, X_{nn})$ satisfies the Lipschitz conditions with $\boldsymbol{c} = (2 - \sqrt{2})(1, \ldots, 1)$, **Why?** .
   Then
$$\mathbf{P}[\,|Z - \mathbf{E}[\,Z\,]\,| \geq t\,] \leq 2\exp\left[\frac{-t^2}{(2-\sqrt{2})^2 n^2}\right]$$

2. Enumerate the columns of squares from 1 to n. Let $Y_i = (X_{1i}, \ldots, X_{ni})$. Then $Z = g(Y_1, \ldots, Y_n)$. g satisfies the Lipschitz conditions with $\boldsymbol{c} = (2 - \sqrt{2})(1, \ldots, 1)$. **Why?**
   Then
$$\mathbf{P}[\,|Z - \mathbf{E}[\,Z\,]\,| \geq t\,] \leq 2\exp\left[\frac{-t^2}{(2-\sqrt{2})^2 n}\right]$$

A random distance problem

Can we prove concentration of the shortest path from $(0,0)$ to (n,n)?
Yes! Let Z be the total length of the shortest path.Two options

1. Define $X_{ij} = 1$ if there is a diagonal in square ij, otherwise $X_{ij} = 0$. Then $Z = f(X_{11}, \dots, X_{nn})$ satisfies the Lipschitz conditions with $\boldsymbol{c} = (2 - \sqrt{2})(1, \dots, 1)$, **Why?** .
   Then
   $$\mathbf{P}[\,|Z - \mathbf{E}[\,Z\,]\,| \geq t\,] \leq 2\exp\left[\frac{-t^2}{(2-\sqrt{2})^2 n^2}\right]$$
2. Enumerate the columns of squares from 1 to n. Let $Y_i = (X_{1i}, \dots, X_{ni})$. Then $Z = g(Y_1, \dots, Y_n)$. g satisfies the Lipschitz conditions with $\boldsymbol{c} = (2 - \sqrt{2})(1, \dots, 1)$. **Why?**
   Then
   $$\mathbf{P}[\,|Z - \mathbf{E}[\,Z\,]\,| \geq t\,] \leq 2\exp\left[\frac{-t^2}{(2-\sqrt{2})^2 n}\right]$$

   Note the second bound is way more useful than the first one.

Example: Clique Number in Random Graphs

Example: Clique Number in Random Graphs

1. Consider a random graph $G = G_{n,p}$ on n vertices where each possible edge appears with probability p independent of each other.

Example: Clique Number in Random Graphs

1. Consider a random graph $G = G_{n,p}$ on n vertices where each possible edge appears with probability p independent of each other.
2. Denote by K the clique number of G defined as the size of the largest complete subgraph of G.

Example: Clique Number in Random Graphs

1. Consider a random graph $G = G_{n,p}$ on n vertices where each possible edge appears with probability p independent of each other.
2. Denote by K the clique number of G defined as the size of the largest complete subgraph of G.
3. K is a function of the number of edges of the graph, i.e. $K = K(X_1, \ldots, X_{\binom{n}{2}})$ where X_i represent if the i-th possible edge is in the graph or not.

Example: Clique Number in Random Graphs

1. Consider a random graph $G = G_{n,p}$ on n vertices where each possible edge appears with probability p independent of each other.
2. Denote by K the clique number of G defined as the size of the largest complete subgraph of G.
3. K is a function of the number of edges of the graph, i.e. $K = K(X_1, \ldots, X_{\binom{n}{2}})$ where X_i represent if the i-th possible edge is in the graph or not.
4. Lipschitz conditions holds with $\boldsymbol{c} = (1, \ldots, 1)$. **Why?**

Example: Clique Number in Random Graphs

1. Consider a random graph $G = G_{n,p}$ on n vertices where each possible edge appears with probability p independent of each other.
2. Denote by K the clique number of G defined as the size of the largest complete subgraph of G.
3. K is a function of the number of edges of the graph, i.e. $K = K(X_1, \ldots, X_{\binom{n}{2}})$ where X_i represent if the i-th possible edge is in the graph or not.
4. Lipschitz conditions holds with $\boldsymbol{c} = (1, \ldots, 1)$. **Why?**
5. Therefore, for $t > 0$

$$\mathbf{P}[\, K - \mathbf{E}[\, K\,] \geq t\,]\,, \mathbf{P}[\, K - \mathbf{E}[\, K\,] \leq t\,] \leq e^{-2t^2/\binom{n}{2}}.$$

Example: Clique Number in Random Graphs

1. Consider a random graph $G = G_{n,p}$ on n vertices where each possible edge appears with probability p independent of each other.

Example: Clique Number in Random Graphs

1. Consider a random graph $G = G_{n,p}$ on n vertices where each possible edge appears with probability p independent of each other.
2. Denote by K the clique number of G defined as the size of the largest complete subgraph of G.

Example: Clique Number in Random Graphs

1. Consider a random graph $G = G_{n,p}$ on n vertices where each possible edge appears with probability p independent of each other.
2. Denote by K the clique number of G defined as the size of the largest complete subgraph of G.
3. Enumerate the vertices from 1 to n
4. Let $X_{i,j} = 1$ if there is a edge between vertices i and j, otherwise $X_{i,j} = 0$
5. Let $Y_i = (X_{i,1}, X_{i,2}, \dots, X_{i,i-1})$
6. K is a function of the Y_i.
7. Lipschitz conditions holds with $\boldsymbol{c} = (1, \dots, 1)$. **Why?**

Example: Clique Number in Random Graphs

1. Consider a random graph $G = G_{n,p}$ on n vertices where each possible edge appears with probability p independent of each other.
2. Denote by K the clique number of G defined as the size of the largest complete subgraph of G.
3. Enumerate the vertices from 1 to n
4. Let $X_{i,j} = 1$ if there is a edge between vertices i and j, otherwise $X_{i,j} = 0$
5. Let $Y_i = (X_{i,1}, X_{i,2}, \ldots, X_{i,i-1})$
6. K is a function of the Y_i.
7. Lipschitz conditions holds with $\boldsymbol{c} = (1, \ldots, 1)$. **Why?**
8. Therefore, for $t > 0$

$$\mathbf{P}[\, K - \mathbf{E}[\, K\,] > t\,]\,, \mathbf{P}[\, K - \mathbf{E}[\, K\,] < t\,] \leq e^{-2t^2/n}.$$

Observe this bound is better than the previous one

MaxCut on Random Graphs

We analyse the Max-Cut problems on Random Graphs, i.e. instead of assuming worst case input, we assume a random input.

MaxCut on Random Graphs

We analyse the Max-Cut problems on Random Graphs, i.e. instead of assuming worst case input, we assume a random input.

1. Consider a random graph $G_{n,1/2}$ on vertices $[n] = \{1, \ldots, n\}$ where each possible edge appears with probability $1/2$

MaxCut on Random Graphs

We analyse the Max-Cut problems on Random Graphs, i.e. instead of assuming worst case input, we assume a random input.

1. Consider a random graph $G_{n,1/2}$ on vertices $[n] = \{1, \ldots, n\}$ where each possible edge appears with probability $1/2$
2. Let $S \subseteq [n]$. Denote by $E(S : S^c)$ be the set of edges between S and its complement (i.e. the size of the cut given by S).

MaxCut on Random Graphs

We analyse the Max-Cut problems on Random Graphs, i.e. instead of assuming worst case input, we assume a random input.

1. Consider a random graph $G_{n,1/2}$ on vertices $[n] = \{1, \dots, n\}$ where each possible edge appears with probability $1/2$
2. Let $S \subseteq [n]$. Denote by $E(S : S^c)$ be the set of edges between S and its complement (i.e. the size of the cut given by S).
3. $\mathbf{E}[\,|E(S : S^c)|\,] = \frac{|S|(n-|S|)}{2} \leq n^2/8$

MaxCut on Random Graphs

We analyse the Max-Cut problems on Random Graphs, i.e. instead of assuming worst case input, we assume a random input.

1. Consider a random graph $G_{n,1/2}$ on vertices $[n] = \{1, \ldots, n\}$ where each possible edge appears with probability $1/2$
2. Let $S \subseteq [n]$. Denote by $E(S : S^c)$ be the set of edges between S and its complement (i.e. the size of the cut given by S).
3. $\mathbf{E}[\,|E(S : S^c)|\,] = \frac{|S|(n-|S|)}{2} \leq n^2/8$
4. Note that $C_S = |E(S : S^c)|$ depends on the possible $|S|(n - |S|)$ edges between S and S^c

MaxCut on Random Graphs

We analyse the Max-Cut problems on Random Graphs, i.e. instead of assuming worst case input, we assume a random input.

1. Consider a random graph $G_{n,1/2}$ on vertices $[n] = \{1, \ldots, n\}$ where each possible edge appears with probability $1/2$
2. Let $S \subseteq [n]$. Denote by $E(S : S^c)$ be the set of edges between S and its complement (i.e. the size of the cut given by S).
3. $\mathbf{E}[\,|E(S : S^c)|\,] = \frac{|S|(n-|S|)}{2} \leq n^2/8$
4. Note that $C_S = |E(S : S^c)|$ depends on the possible $|S|(n - |S|)$ edges between S and S^c
5. $C_S = C_S(X_1, \ldots, X_m)$ where $m = |S|(n - |S|)$, where X_i indicates if the i-th edge appears in the cut or not

MaxCut on Random Graphs

We analyse the Max-Cut problems on Random Graphs, i.e. instead of assuming worst case input, we assume a random input.

1. Consider a random graph $G_{n,1/2}$ on vertices $[n] = \{1, \ldots, n\}$ where each possible edge appears with probability $1/2$
2. Let $S \subseteq [n]$. Denote by $E(S : S^c)$ be the set of edges between S and its complement (i.e. the size of the cut given by S).
3. $\mathbf{E}[\,|E(S : S^c)|\,] = \frac{|S|(n-|S|)}{2} \leq n^2/8$
4. Note that $C_S = |E(S : S^c)|$ depends on the possible $|S|(n - |S|)$ edges between S and S^c
5. $C_S = C_S(X_1, \ldots, X_m)$ where $m = |S|(n - |S|)$, where X_i indicates if the i-th edge appears in the cut or not
6. C_S is Lipschitz with $\boldsymbol{c} = (1, \ldots, 1)$

MaxCut on Random Graphs

We analyse the Max-Cut problems on Random Graphs, i.e. instead of assuming worst case input, we assume a random input.

1. Consider a random graph $G_{n,1/2}$ on vertices $[n] = \{1, \ldots, n\}$ where each possible edge appears with probability $1/2$
2. Let $S \subseteq [n]$. Denote by $E(S : S^c)$ be the set of edges between S and its complement (i.e. the size of the cut given by S).
3. $\mathbf{E}[\,|E(S : S^c)|\,] = \frac{|S|(n-|S|)}{2} \leq n^2/8$
4. Note that $C_S = |E(S : S^c)|$ depends on the possible $|S|(n - |S|)$ edges between S and S^c
5. $C_S = C_S(X_1, \ldots, X_m)$ where $m = |S|(n - |S|)$, where X_i indicates if the i-th edge appears in the cut or not
6. C_S is Lipschitz with $\boldsymbol{c} = (1, \ldots, 1)$
7. Therefore, for $\delta > 0$,

$$\mathbf{P}[\, C_S - \mathbf{E}[\, C_S \,] \geq \delta \mathbf{E}[\, C_S \,] \,] \leq \exp\left(- \frac{2\delta^2 \mathbf{E}[\, C_S \,]^2}{|S|(n - |S|)} \right)$$

8. **Exercise:** Deduce that for any $S \subseteq [n]$,

$$\mathbf{P}\left[C_S \geq \frac{n^2}{8} + \delta \frac{n^2}{4} \right] \leq e^{-\Omega(\delta^2 n^2)}$$

9. By the union bound, we have that

$$\mathbf{P}\left[\exists S : C_S \geq \frac{n^2}{8} + \delta \frac{n^2}{4} \right] \leq 2^n e^{-\Omega(\delta^2 n^2)} = 2^n e^{-\Omega(c^2 n)}$$

10. Recall that $\delta = c/\sqrt{n}$, now we pick c to be large enough, such that $2^n e^{-\Omega(c^2 n)} = 2^{-n}$
11. The main result is:

Theorem

There is a constant c, such that w.h.p. the Max Cut in $G_{n,1/2}$ is at most $n^2/8 + cn^{3/2}$