# Lecture 5: Concentration Inequalities

John Sylvester Nicolás Rivera Luca Zanetti Thomas Sauerwald

Lent 2019

UNIVERSITY OF CAMBRIDGE

# Outline

# Concentration Inequalities

- Concentration refers to the phenomena where random variables are very close to their mean.
- This is very useful in randomised algorithms as it ensures an **almost** deterministic behaviour
- It gives us the best of two worlds:
  1. **Randomised Algorithms:** Easy to Design and Implement
  2. **Deterministic Algorithms:** They do what they claim to do

Recall the Markov and Chebyshev inequalities from the first lecture

**Markov Inequality**

If $X$ is a non-negative random variable and $a > 0$, then

$$\mathbf{P}[\, X \geq a \,] \leq \mathbf{E}[\, X \,] \,/ a.$$

**Chevyshev Inequality**

If $X$ is a random variable and $a > 0$, then

$$\mathbf{P}[\, |X - \mathbf{E}[\, X \,]| \geq a \,] \leq \mathbf{Var}[\, X \,] \,/ a^2.$$

Let $f : \mathbb{R} \to [0, \infty)$ and increasing, then $Y = f(X) \geq 0$, and thus

$$\mathbf{P}[\, X \geq a \,] \leq \mathbf{P}[\, f(X) \geq f(a) \,] \leq \mathbf{E}[\, f(X) \,] \,/ f(a).$$

Similarly, if $g : \mathbb{R} \to [0, \infty)$ and decreasing, then $Y = g(X) \geq 0$, and thus

$$\mathbf{P}[\, X \leq a \,] \leq \mathbf{P}[\, g(X) \geq g(a) \,] \leq \mathbf{E}[\, g(X) \,] \,/ g(a).$$

By choosing an appropriate function we can obtain inequalities that are much sharper than the Markov and Chevyshev Inequality.

# Example: coin flip

Consider $n$ fair coins and let $X$ be the total number of head. In an experiment we expect to see around $n/2$ heads. Can we justify that? Let $\delta > 0$

- Markov inequality :

$$\mathbf{P}[\,X \geq (1+\delta)(n/2)\,] \leq \frac{\mathbf{E}[\,X\,]}{(1+\delta)(n/2)} = \frac{n/2}{(1+\delta)(n/2)} = \frac{1}{1+\delta}$$

Not good! Independent of $n$

- Chebychev inequality :

$$\begin{aligned} \mathbf{P}[\,(X - n/2) \geq \delta(n/2)\,] &\leq \mathbf{P}\left[\,(X - n/2)^2 \geq (\delta(n/2))^2\,\right] \\ &\leq \frac{4\mathbf{Var}\,[\,X\,]}{\delta^2 n^2} = \frac{1}{\delta^2 n} \end{aligned}$$

Better! Linear in $n$

Markov and Chebychev use the first and second moment of the random variable. Can we keep going?

- Yes.

We can consider first, second, third and more moments! that is the basic idea behind the Chernoff Bounds

# Outline

# Chernoff Bounds

**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 any $\delta > 0$ it holds that

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

and for $t > \mu$ it holds that

$$\mathbf{P}[\,X \geq t\,] \leq e^{-\mu}\left(\frac{e\mu}{t}\right)^{t},$$

## Example: Coin Flip

Consider $n$ fair coins and let $X$ be the total number of head. Then

$$\mathbf{P}[\, X \geq (1+\delta)(n/2)\,] \leq \left[ \frac{e^{\delta}}{(1+\delta)^{(1+\delta)}} \right]^{n/2}$$

Node that the above expression equals 1 only for $\delta = 0$, and then it gives a value strictly less than 1 (**Ex: check this!**). Note as well the inequality is exponential in $n$, (for fixed $\delta$) i.e. much better than Chebychev inequality.

# Example: Coin Flip

Consider 100 independent coin flips . We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

- Markov's inequality : $X = \sum_{i=1}^{100} X_i$, $X_i \in \{0,1\}$ and $\mathbf{E}[X] = 100 \cdot \frac{1}{2} = 50$.

$$\mathbf{P}[X \geq 3/2 \cdot \mathbf{E}[X]] \leq 2/3 = 0.666.$$

- Chebyshev's inequality : $\mathbf{Var}[X] = \sum_{i=1}^{100} \mathbf{Var}[X_i] = 100 \cdot (1/2)^2 = 25$.

$$\mathbf{P}[|X - \mu| \geq t] \leq \frac{\mathbf{Var}[X]}{t^2},$$

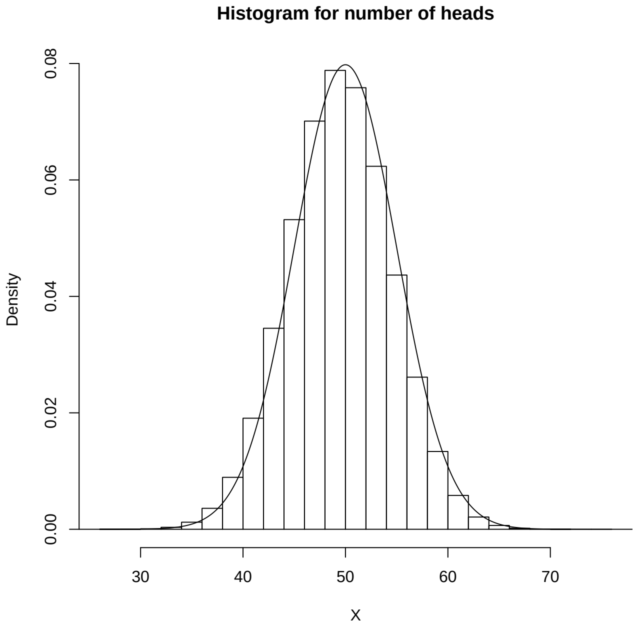
  and plugging in $t = 25$ gives an upper bound of $25/25^2 = 1/25 = 0.04$, much better than what we obtained by Markov's inequality.

- The Chernoff bound (first) with $\delta = 1/2$ gives:

$$\mathbf{P}[X \geq 3/2 \cdot \mathbf{E}[X]] \leq \left( \frac{e^{1/2}}{(3/2)^{3/2}} \right)^{50} = 0.004472$$

- the exact probability is $0.00000028 \ldots$

Chernoff bound yields a more accurate result but needs independence!


Histogram for number of heads
Density
0.00
0.02
0.04
0.06
0.08
30
40
50
60
70
X

# Outline

Concentration Inequalities

Chernoff Bounds

Balls into Bins

Proof of Chernoff Bounds

Randomised QuickSort

# Example: Balls into Bins

Recall the balls into bins process: Assign $m$ balls uniformly and independently to $n$ bins.

**Balls into Bins question**

How large is the maximum load?

- Focus in one bin. Let $X_i$ the indicator variable that indicates if ball $i$ is assigned to this bin. The total balls in the bin is given by $X = \sum_i X_i$. Note that $p_i = \mathbf{P}[\, X_i = 1 \,] = 1/n$.
- Suppose that $m = 2n \log n$, then $\mu = \mathbf{E}[\, X \,] = 2 \log n$

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

- By the Chernoff Bound,

$$\mathbf{P}[\, X \geq 6 \log n \,] \leq e^{-2 \log n} \left( \frac{2e \log n}{6 \log n} \right)^{6n \log n} \leq e^{-2 \log n} = n^{-2}$$

# Example: Balls into Bins

- Let $\mathcal{E}_j$ be the event that bin $j$ receives more than $6 \log n$ balls.
- We are interested in the probability that at least one bin receives more than $6 \log n$ balls
- This is the event $\cup_{j=1}^{n} \mathcal{E}_j$
- By the Union Bound, $\mathbf{P}\left[\,\cup_{j=1}^{n} \mathcal{E}_j\,\right] \leq \sum_{j=1}^{n} \mathbf{P}[\,\mathcal{E}_j\,] \leq n \times n^{-2} = n^{-1}$
- Therefore *whp*, no bin receives more than $6 \log n$ balls.
- Note that the max loaded bin receives at least $2 \log n$. So our bound is pretty sharp.

whp stands for *with high probability* :
An event $\mathcal{E}$ (that implicitly depends on an input parameter $n$) occurs whp if
$\mathbf{P}[\,\mathcal{E}\,] \to 1$ as $n \to \infty$.
This is a very standard notation in randomised algorithms
but it may very from author to author. Be careful!

# Example: Balls into Bins

Consider now the case $m = n$, i.e. same number of balls and bins.
Using the Chernoff Bounds

$$\mathbf{P}[\,X > t\,] \leq e^{-1}\left(\frac{e}{t}\right)^{t} \leq \left(\frac{e}{t}\right)^{t}$$

$$\boxed{\mathbf{P}[\,X \geq t\,] \;\leq\; e^{-\mu}(e\mu/t)^{t}}$$

By setting $t = 4\log n/\log\log n$, we obtain that $\mathbf{P}[\,X > t\,] \leq n^{-2}$. Indeed

$$\left(\frac{e\log\log n}{4\log n}\right)^{4\log n/\log\log n} = \exp\left(\frac{4\log n}{\log\log n}\cdot\log\left(\frac{e\log\log n}{4\log n}\right)\right) \qquad (1)$$

The term inside the exponential is

$$\left(\frac{4\log n}{\log\log n}\cdot\left(\log(4/e) + \log\log\log n - \log\log n\right)\right) \leq \left(\frac{4\log n}{\log\log n}\left(-\frac{1}{2}\log\log n\right)\right)$$

obtaining that $\mathbf{P}[\,X > t\,] \leq n^{-4/2} = n^{-2}$.

This inequality only works for large enough $n$

# Example: Balls into Bins

We just proved that

$$\mathbf{P}[\, X > 4 \log n / \log \log n \,] \leq n^{-2},$$

thus by the union bound , no bin receives more than $O(\log n / \log \log n)$ balls with probability at least $1 - 1/n$.

- You will see in your Exercise class how to prove that whp at least one bin receives at least $c \log n / \log \log n$ balls, for some $c$. You will need to use the Probabilistic Method mentioned in the first lecture

## Example: Balls into Bins

### Conclusions

- If the number of balls is $2 \log n$ times $n$ (the number of bins), then to distribute balls at random is a good algorithm
  - This is because the worst case load is whp. $6 \log n$, while the expected number of balls is $2 \log n$
- For the case $n = m$, the algorithm is not good, because the maximum load is whp. $\Theta(\log n / \log \log n)$, while the expected number of balls per bin is 1.
- For the case $n = m$, we can improve the balls into bin process by sampling two bins in each step, then assigning the ball into the bin with lesser load. This gives a maximum load $\Theta(\log \log n)$ with probability at least $1 - 1/n$.

This is called the power of two choices :
It is a standard technique to improve the
performance of randomised algorithms.

# Outline

# Chernoff Bound: Proof

**Chernoff Bound**

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 any $\delta > 0$ it holds that

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

Proof:

1. For $\lambda > 0$,

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

2. $\mathbf{E}[\, e^{\lambda X} \,] = \mathbf{E}\left[\, e^{\lambda \sum X_i} \,\right] \underset{\text{indep}}{=} \prod_{i=1}^{n} \mathbf{E}[\, e^{\lambda X_i} \,]$

3. $$\mathbf{E}\left[\, e^{\lambda X_i} \,\right] = e^{\lambda} p_i + (1 - p_i) = 1 + p_i(e^{\lambda} - 1) \underset{\substack{1+x \leq e^{x} \\ \text{for } x > 0}}{\leq} e^{p_i(e^{\lambda} - 1)}$$

# Chernoff Bound: Proof

1. For $\lambda > 0$,

$$\mathbf{P}[\, X \geq (1+\delta)\mu \,] \underset{e^{\lambda x} \text{ is incr}}{=} \mathbf{P}\left[\, e^{\lambda X} \geq e^{\lambda(1+\delta)\mu} \,\right] \underset{\text{Markov}}{\leq} e^{-\lambda(1+\delta)\mu} \mathbf{E}\left[\, e^{\lambda X} \,\right]$$

2. $\mathbf{E}[\, e^{\lambda X} \,] = \mathbf{E}\left[\, e^{\lambda \sum X_i} \,\right] \underset{\text{indep}}{=} \prod_{i=1}^{n} \mathbf{E}[\, e^{\lambda X_i} \,]$

3. $$\mathbf{E}\left[\, e^{\lambda X_i} \,\right] = e^{\lambda} p_i + (1 - p_i) = 1 + p_i(e^{\lambda} - 1) \underset{\substack{1+x \leq e^x \\ \text{for } x>0}}{\leq} e^{p_i(e^{\lambda}-1)}$$

4. Putting all together

$$\mathbf{P}[\, X \geq (1+\delta)\mu \,] \leq e^{-\lambda(1+\delta)\mu} \prod_{i=1}^{n} e^{p_i(e^{\lambda}-1)} = e^{-\lambda(1+\delta)\mu} e^{\mu(e^{\lambda}-1)}$$

5. Choose $\lambda = \log(1+\delta) > 0$ to get the result.

# Chernoff Bound: The recipe

The proof of the Chernoff bound is based in three key steps. These are

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 $\mathbf{E}[\, e^{\lambda X} \,]$ **(This is the hard one)**
3. Optimise the value of $\lambda > 0$.

The function $\lambda \to \mathbf{E}[\, e^{\lambda X} \,]$ is called the **moment-generating function** of $X$ and it is very important to obtain sharp concentration inequalities.

**Exercise:** prove that $\mathbf{P}[\, X \geq t \,] \leq e^{-\mu} \left(\frac{e\mu}{t}\right)^t$,

# Chernoff Bounds: Lower Tails

We can also use Chernoff Bounds to shows that a random variable is **not too small** compared to its mean.

**Chernoff Bounds: Lower Tails**

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 any $\delta > 0$ it holds that

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

and for any $t < \mu$

$$\mathbf{P}[\,X \leq t\,] \leq e^{-\mu}\left(\frac{e\mu}{t}\right)^{t}.$$

**Exercise: Prove it.**

Hint: multiply both sides by $-1$ and repeat the proof of the Chernoff Bound

## The useful Chernoff Bounds

Our current form of the Chernoff Bound is rather atrocious. We can derive a slightly weaker but more readable version.

**Nicer Chernoff Bounds**

Suppose $X_1, \dots, X_n$ 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 $t > 0$,

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

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

- For $0 < \delta < 1$,

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

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

**Exercise: Prove it**

# Outline

Concentration Inequalities

Chernoff Bounds

Balls into Bins

Proof of Chernoff Bounds

Randomised QuickSort

# Applications: QuickSort

Quick sort is a sorting algorithm that works as following.

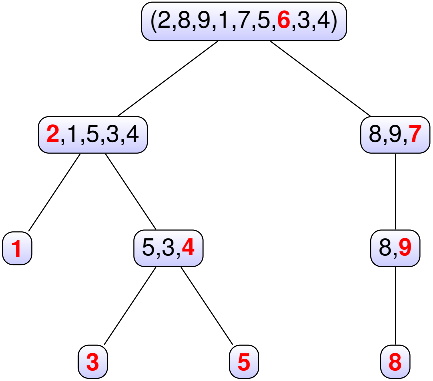
**Algorithm: QuickSort**

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




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.**

This should be your standard answer in this course

Let's analyse quicksort with random pivots.

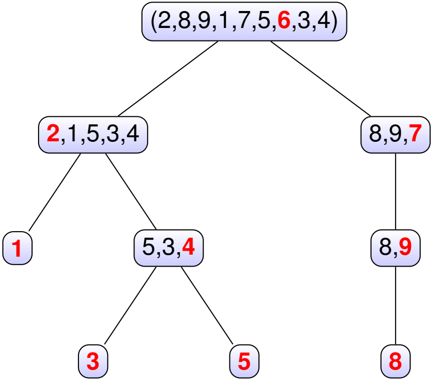
1. Consider $n$ different number, wlog, $\{1, \ldots, n\}$
2. let $H_i$ be the last level where $i$ appears in the tree. Then the number of comparison is $H = \sum_{i=1}^{n} H_i$
3. we will prove that exists $C > 0$ such that

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

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.
- 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 $P$.
- 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 $X = \sum_{i=1}^{\lfloor 21 \log n \rfloor} X_i$.
- 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

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

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

- 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 $\Omega(n \log n)$
- 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.