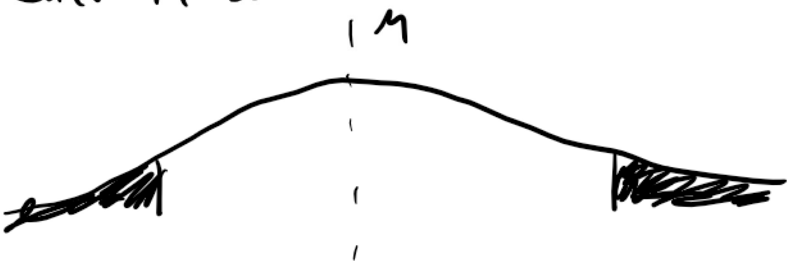
Tutorial VII

02 April 2020 14:03

Concentration bounds

- → Chernoff bounds
  - ↳ exercise
- → Algorithm to estimate π
- → Intro to routing on hypergraphs

Chernoff bounds



$$X = \sum_{i=1}^{n} X_i$$

X_i ≃ Poisson r.v. (taking values 0 and 1 only) with $P_r(X_i = 1) = p$

$$E(X) = n \cdot p = \mu$$

X_i are iid (identically independently distributed)

$$P_r\left(X \le (1-\delta)\cdot\mu\right) \le e^{-\frac{\mu\delta^2}{2}}$$

$$P_r\left(X \ge (1+\delta)\mu\right) \le e^{-\frac{\mu\delta^2}{3}}$$

$0 \le \delta \le 1$

$$P_r\left(|X-\mu| \ge \delta\mu\right) \le 2 \cdot e^{-\frac{\mu\delta^2}{3}}$$

Polling problem

Two presidential candidates A and B.

We want to estimate the number of people who will vote for A. Let's assume everyone will vote.

Let's define answer of i^{th} person with r.v. X_i

Such that $X_i = 1$ votes for A

$X_i = 0$ votes for B

$Pr(X_i = 1) = p$ (the percentage of people voting for A = the number we want to estimate.)

After asking n people, the estimate $X = \frac{X_1 + X_2 + \ldots + X_n}{n}$

$$Pr(|X - p| \leq (0.1) \cdot p) > 90\%$$

$$Pr\left(|X - p| \geq \frac{p}{10}\right) \leq 10\%$$

$X - \mu$

by ch.b. $\leq 2 \cdot e^{-\frac{p \cdot (\frac{1}{10})^2}{3}}$

we would like to estimate how large n should be to give good estimates

this does not contain n!

$$E(X) = \sum_{i=1}^{n} E\left(\frac{X_i}{n}\right) = \frac{1}{n} \sum_{i=1}^{n} E(X_i) = \frac{1}{n} n \cdot p = p = \mu$$

$E(X_i) = p$

$$Pr\left(|nX - np| \geq \frac{np}{10}\right) \leq \frac{1}{10}$$

μ'

$E(nX) = n.p = \mu'$

$$< 2 \cdot e^{-\frac{\mu' \delta^2}{3}}$$

$-\frac{\mu' \delta^2}{} \quad . \quad 1$

$$2 \cdot e^{-\frac{n' \sigma^2}{3}} \le \frac{1}{10}$$

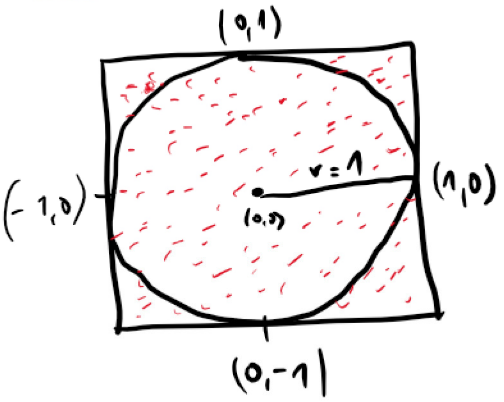
$$e^{-\frac{n.p}{3.100}} \le \frac{1}{20} \qquad /\ln$$

$$-\frac{n.p}{300} \le \ln\left(\frac{1}{20}\right)$$

$$n \ge -\frac{300}{p} \cdot \ln\left(\frac{1}{20}\right)$$

$$n \ge 900/p$$

Algorithm to estimate π ≈ 3.1415



$z_i = 1$ if i^{th} point is inside the circle

$z_i = 0$ if i^{th} point is outside the circle

$$Pr(z_i = 1) = \frac{V(\text{circle})}{V(\text{square})} = \frac{\pi r^2}{(2r)^2} = \frac{\pi}{4}$$

After n trials we have $Z = \sum_i z_n$

$$E(Z) = \sum_{i=1}^{n} E(z_i) = n \cdot \frac{\pi}{4}$$

$z' = \frac{4.z}{n}$ is an estimate of π. $E(z') = \pi$

$$Pr(|z' - \pi| \le \varepsilon.\pi)$$

$$Pr(|z' - \pi| \ge \varepsilon \cdot \pi) \le 2e^{-\frac{\pi.\varepsilon^2}{3}} \qquad /.n/4$$

$Pr(|Z-\Pi| \geq \varepsilon \cdot \Pi) \leq e^{-} \quad \ldots$

$$Pr\left(\left|\tfrac{n}{4}Z' - \tfrac{n}{4}\Pi\right| > \tfrac{n}{4}\varepsilon \cdot \Pi\right) \leq \qquad E(nZ') = \frac{n}{4}E(Z') = E(Z) = \frac{n\Pi}{4} = (\mu)$$

$$\leq e^{\frac{-n\Pi\varepsilon^2}{3 \cdot 4}} = e^{\frac{-n\Pi\varepsilon^2}{12}} = \frac{1}{e^{\frac{n\Pi\varepsilon^2}{12}}} = \frac{1}{a^{\frac{1}{12}}} \qquad e^{n\Pi\varepsilon^2} = a$$

Can we get better (less samples) using a cube and a sphere?

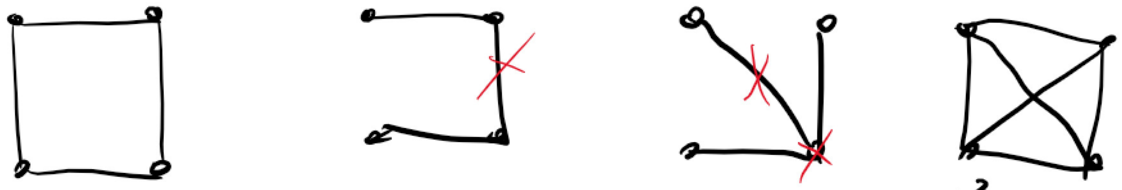
$$Pr(Z_i = 1) = \frac{V(\text{sphere})}{V(\text{cube})} = \frac{\frac{4}{3} \cdot \Pi \cdot r^3}{(2r)^3} = \frac{\Pi}{6}$$

$$Pr((nZ' - n\Pi) \leq e^{\frac{-n\Pi\varepsilon^2}{18}} = \frac{1}{e^{\frac{n\Pi\varepsilon^2}{18}}} = \frac{1}{a^{\frac{1}{18}}}$$

$$a^{\frac{1}{18}} < a^{1/16} \qquad \frac{1}{a^{1/18}} > \frac{1}{a^{1/16}}$$

Routing in Hypercubes

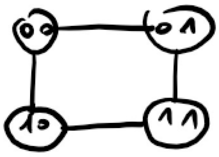
Hypercube is a network architecture

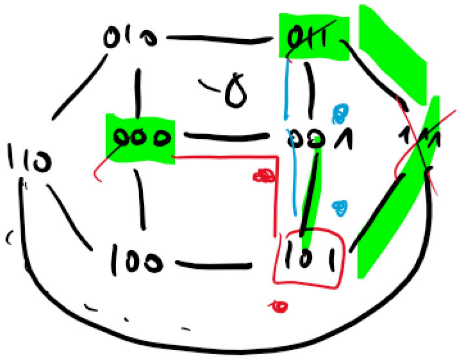


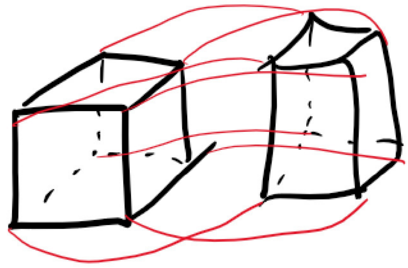
$\frac{N^2}{2}$

$N = 2^d$ nodes labeled by binary strings (of length d)

two nodes are connected iff they differ in a single bit







Total number of edges is $\frac{N.d}{2}$

Packet $(i, X, d(i))$

- i → Source node
- X → data
- $d(i)$ → destination

Assumption: 1 packet in a link per timestep

Deterministic routing left to right bit fixing

1100 → 1000 → 1010

Experiment (to investigate routing algorithms)

Each node gets a packet with a destination.
How long will it take to deliver all of the packets?

The worst case

$x_1, \dots, x_d \rightarrow x_d, \dots, x_1$

i = 1100 $d(i)$ = 0011
j = 0100 $d(j)$ = 0010
k = 1000 $d(k)$ = 0001

L->R

i: 1100 -> 0100 -> 0000 -> 0010 -> 0011

j: 0100 -> 0000 -> 0010

k: 1000 -> 0000 -> 0001

$x_1 \dots x_{\frac{n}{2}}$ 00000

00000 $x_{\frac{n}{2}} \dots x_n$

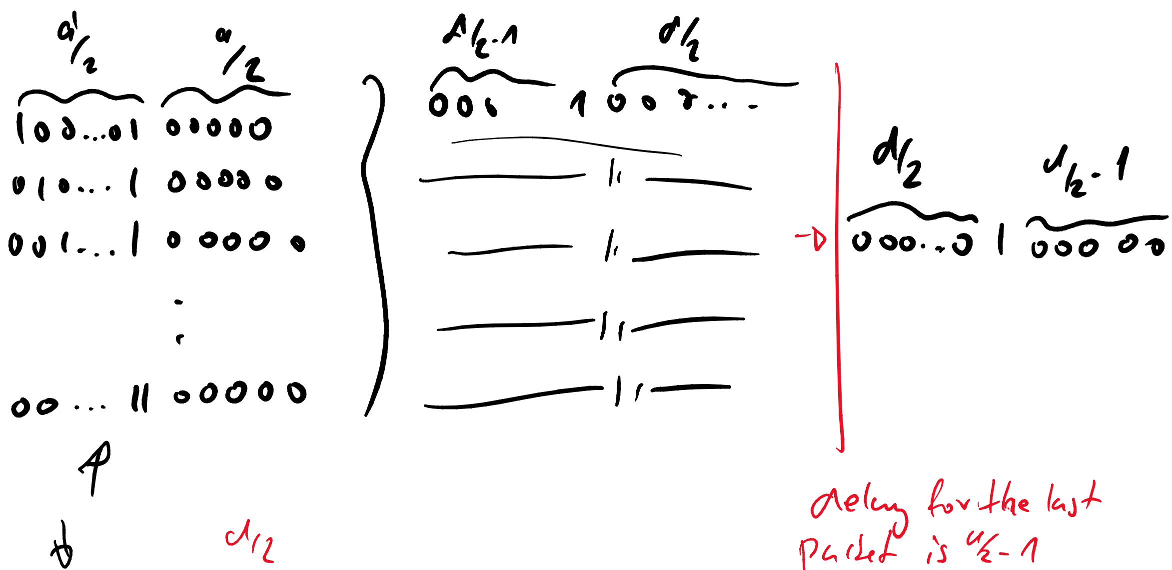
i: 101 000 -> 000 101

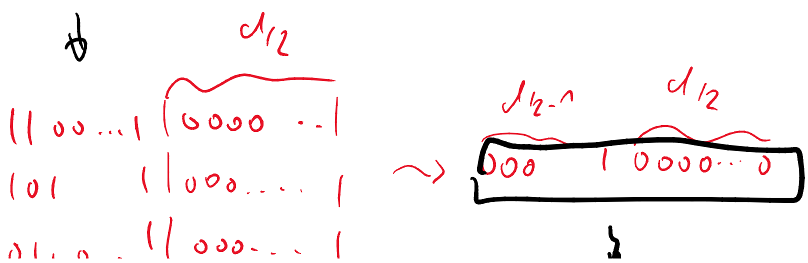
j: 011 000 -> 000 110

101000 -> 001000 | -> 000 000 -> 000 100 -> 000101

011000 -> 001000 | -> 000 000 -> 000 100 -> 000110

delay





101 11000.... 1 ~> 000 1 0000... 0

010.. 11 000... 1

$\underbrace{\quad}_{2^{d/2}}$

d out lines which can be used at the same time

$$\Omega\left(\frac{2^d}{d}\right)$$

Random algorithm:

$\sigma(i)$ → chosen at random

$i \to \sigma(i)$ (L → R)

$\sigma(i) \to d(i)$

w.p. at least $1 - 2^{-5d}$ every packet gets delivered in time

$14d$.