Lecture 22/04/16

Friday, April 22, 2016 2:53 PM

① Concentration Inequalities
- Markov
- Chernoff
- Chebyshev
- Bernstein

highly relevant

Matrix Chernoff (eigenvalues, etc). ← occasionally useful

② Hashing
- Universal Hash
- → LSH

③ Gaussian RV
- Spherical symmetries
- Concentration of length

{ eg LSH & JL }

④ JL ← data-oblivious dimensionality reduction

SVD ← data-aware

(singular values, vectors, low-rank matrix approximations)

values, vectors → proves row rank = column rank

~~Best~~ Big Data Algorithms

⑤ Introduced Linear Programming
- BFS
- Extreme Point
- Corner Point

all are same → motivated Simplex Algorithm

- Duality

Techniques

- Max Flow = Min Cut, Min Cost Flow, Matching
- Facility location
- Compressed Sensing {RIP + gaussians}
- Machine Scheduling

⑥ Convex Programming
- What is convexity?
- Gradient descent
- $\boxed{\text{MW algo}}$ (online learning) ← ML

Finance, Forecasting, Prediction, Ads

Online Algorithms ("Under the hood")
- resource allocation ← Potential Functions, Online LP Solving
- Paging, — online "stochastic" optimization
- Set cover, Circuit routing
- Secretary Problem ← online, semi-adversarial

Expander Graphs
- Definition
- Examples of expansion, non-expanders ← random graphs, (e.g)
- AKS sorting networks ("derandomization")

→ Definition
→ Examples of expansion, non-expanders → random graphs, (e.g)
→ AKS sorting networks ("derandomization")
→ ~~Sp~~ Graph Sparsifiers

Big Data, Analytics, Forecasting, System Building Algos, Routing, Pricing
JL → SVD, MW
Hashing, Resource Alloc, etc.
LPs

Solutions to selected HW problems :-

Mid-sem polling :-
Accuracy, Confidence

{ p is the prob. that a random person votes congress. }

$X_1 = 1$ if 1^{st} sampled person votes Congress

$X_2 = 1$ if 2^{nd} " "

:

$X_m = 1$ if m^{th} sampled " "

$E[X_1] = p$, $E[X_2] = p$, ... $E[X_m] = p$.

$S = \sum X_i$

$E(S) = mp$

$\boxed{\text{Predict our estimate as } \frac{S}{m}}$

Q1) how accurate is prediction? (ε)

Q2) How confident are we? ($1-\delta$)

Q3) When are we $1-\delta$ confident that accuracy is within ε?

$\boxed{\text{When is } \Pr\left(\left|\frac{S}{m} - p\right| \geq \varepsilon\right) \leq \delta}$

$E[\frac{S}{m}]$ $\Rightarrow$ $\Pr(|S - pm| \geq \varepsilon m) \overset{?}{\leq} \delta$

Chernoff / Bernstein

tail prob = $\exp\left(\frac{-\varepsilon^2 m^2}{\text{Variance}(S) + m}\right)$

$\text{Variance}(S) \leq mp \leq m$

$$\text{tail prob} = \exp\left(\frac{-\varepsilon^2 m^{\cancel{2}}}{2\cancel{m}}\right) = \exp(-\varepsilon^2 m)$$

Want $\exp(-\varepsilon^2 m) \le \delta$

$$\boxed{m \geq \frac{1}{\varepsilon^2} \ln\left(\frac{1}{\delta}\right)}$$

Confidence is easy to achieve (inside ln)

Beautiful fact :-

$\boxed{m \text{ is independent of } N!}$

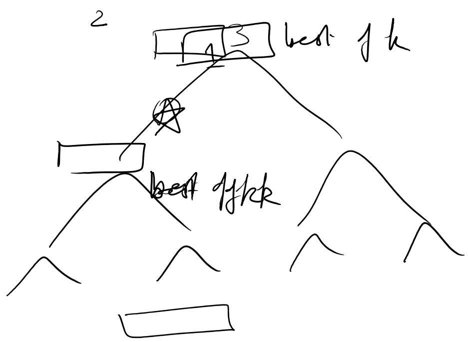
$\varepsilon = 0.01$, $\varepsilon = 0.001 \rightarrow m \geq 10^6 \ln \frac{1}{\delta}$

$m \geq 10000 \ln\left(\frac{1}{\delta}\right)$

Ex 2 (HW ①) ← Designing a tournament

There are n teams. (Unknown implicit ranking)

team i vs team j : better team wins wp $\frac{1}{2}+\delta$ (≈ 0.6)



best team = w.p. wins the tournament

$= \left(\frac{1}{2}\left(1+2\delta\right)\right)^{\log n} = 0.6^{\log_2 n}$

$=$ quite small ($\log$ tiny) !! :(

Pr(best team does not win first series)

= when he/she loses $> \frac{k}{2}$ matches

E[Matches best team wins in $\}$ = $k\left(\frac{1}{2}+\delta\right)$

$\Rightarrow$ E[Matches best team loses] $= k\left(\frac{1}{2}-\delta\right)$

Won't happen too often !

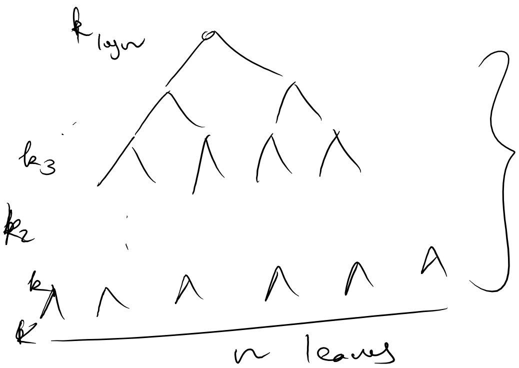
Chernoff :- $\Pr(\text{bad event}) = \exp\left(\frac{-k^{\cancel{2}}\delta^2}{\cancel{k}}\right)$

$\therefore$ set $k = \frac{1}{\delta^2}\log\left(\frac{\log n}{\varepsilon}\right)$

$\Pr(\text{best team loses } 1^{st} \text{ series}) \le \frac{\varepsilon}{\log n}$

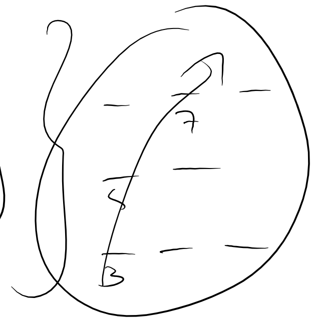
Can UNION bound over all the ... to overall pr $\le \varepsilon$.

Can UNION bound over $\log n$ different series, to overall pr $\leq \varepsilon$.



Overall # matches

$= O\left(\frac{1}{\delta^2} n \log\log \frac{n}{\varepsilon}\right)$



better tournament

to have non-equal # matches in each level. $\leq \varepsilon$

Overall $\Pr(\text{failure}) \leq \left(\exp(-k_1\delta^2) + \exp(-k_2\delta^2) + \dots + \exp(-k_{\log n}\delta^2)\right)$

Overall # games $= \left(k_1 \cdot n + k_2 \cdot \frac{n}{2} + k_3 \cdot \frac{n}{4} + \dots + k_{\log n} \cdot 1\right)$

$k_1 \leq k_2 < k_3 \dots < k_{\log n}$ $\quad \| \quad \Theta(n)$

JL exercise

given n vectors $v_1, v_2, \dots, v_n \in \mathbb{R}^D$ D is very high.

Q) Can I sample a small # co-ordinates to preserve the lengths of resulting vectors?

$= \left(\frac{\log n}{\varepsilon^2}\right)$ dimensions, say.

A: No!! ☹

$v_1 = (1, 0, 0, \dots, 0)$

$v_2 = (0, 1, 0, 0, \dots 0)$ $\quad D = n$

$v_3 = (0, 0, 1, \dots)$

$\vdots$

$v_n = (0, 0, 0, \dots 1)$

even if I sample $d = \frac{\log n}{\varepsilon^2}$ co-ordinates,

almost all of these vectors collapse to $\boxed{0}$.

Main Issue :- Variance / co-ordinate values vary a lot!

Main Issue :- Variance / co-ordinate values vary a lot !

Fix :- Make sure all co-ordinates are bounded in a nice way.

Rotate each vector "randomly"

$$\begin{bmatrix} — g_1 — \\ — g_2 G — \\ \vdots \\ — g_n — \end{bmatrix} \begin{bmatrix} | \\ v_1 \\ | \end{bmatrix} = \text{random rotation of every vector.}$$

$\equiv$ choosing coordinate axis as random rotation

What was originally $(1, 0, 0, 0 \ldots)$

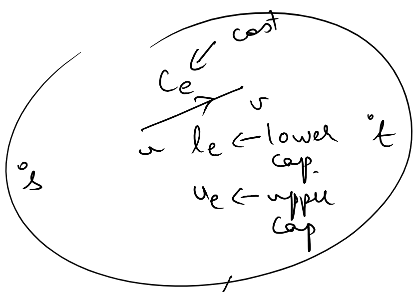
now looks like $(u_1, u_2, \ldots u_n)$

where all u_i's are $\left[-\frac{6}{\sqrt{n}}, \frac{6}{\sqrt{n}}\right]$

SAMPLING WORKS NOW !

behave like a uniformly random rotation of co-ordinate space.

Flows, Hamilton Paths, & the ambulance problem.



G directed

If all l_e's $\in \mathbb{Z}_{\geq 0}$,

u_e's $\in \mathbb{Z}_{\geq 0}$

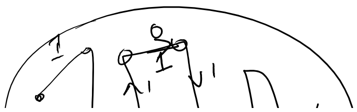
Want a min-cost flow satisfying these bounds

FORMULATED as LINEAR PROGRAM.

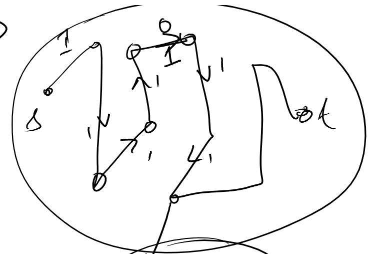
→ Specifically for flow problem, for OPT LP sol^n, the flow values on edges are integral when l_e & $u_e \in \mathbb{Z}_{\geq 0}$

Can Hamilton Path be solved using the integrality of flow?

→



Is there a path from $s \longrightarrow t$ which



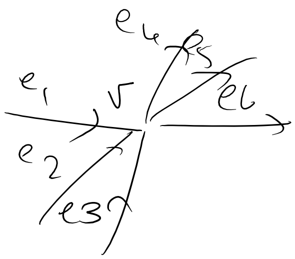
Is there a path from $s \to t$ which visits every other vertex exactly once?

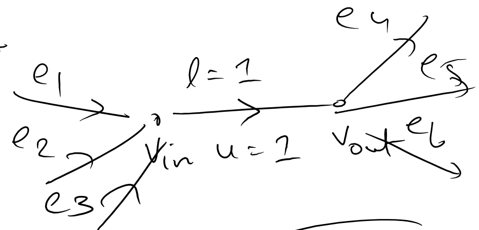
$l_v = 1$
$u_v = 1$

there are now upper & lower capacities on vertices also!

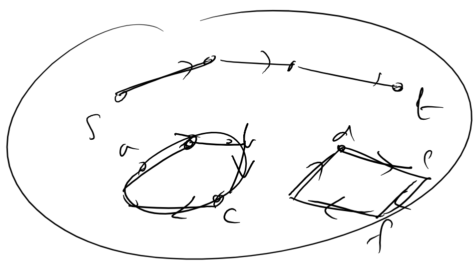
Undirected $\to$ directed is easy

Vertex capacities are easy to handle.





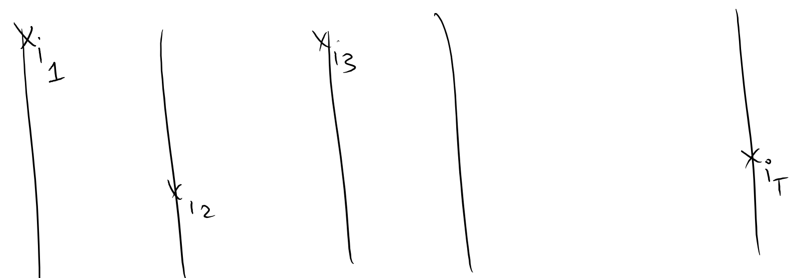
does this solve Hamilton Path? $\Rightarrow$ NP = P



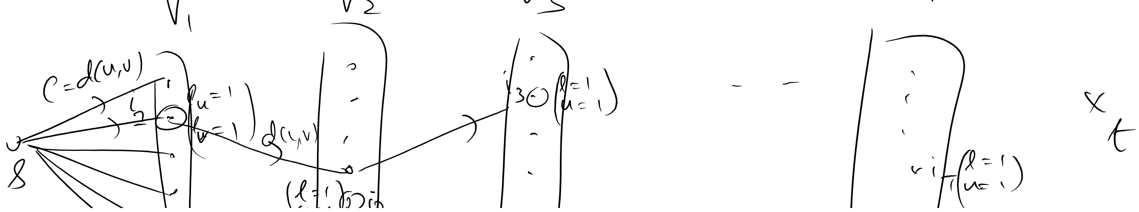
flow can output soln of this kind

AMBULANCE QUESTION

i_0
i_k ambulances



$\to$ Cast it as a flow problem.



Find a ~~k~~ min-cost k flow from s to t satisfying all lower & upper capacity constraints, that will give optimal migration policy