2/8/23 HS1090

classes: M, Tu, W, Th (opt-)

Eval

Quiz 1,2 - 15% × 2 = 30%

Endsem - 40%

Spoken assignment - 10% (task, record video) [4-5 mins.]

Group task - 20% (in English, groups of 3-4)
- some aspect of Germany, culture, lit., philosophy, history
- banned topics: Hitler, cars, beer, football
↳ presented to class after quiz 2

Kaiser - Czar - Caesar - King

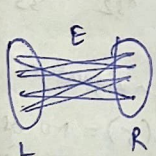
Liechtenstein
2 syllables

* Oder-Neisse line

2/8/23 CS6170

Perfect matching in bipartite graphs

$G(L, R, E)$



A matching $M \subseteq E$ s.t. $\forall u \in L \cup R$, $\exists$ at most one $e \in M$ that is incident on u

A perfect matching is a bijection ($|L| = |R|$)

Q: Given $G(L, R, E)$, check if G has a perfect matching.

There exist polynomial-time deterministic algos., but we're interested in looking at a rand. one

Q': Given $G(L, R, E)$, compute a p.m. if one exists.

We can consider some edge & check if it is present in any p.m. by deleting it & recursing on the smaller graph ⟹ property called self-reducibility

We form a soln to this by solving a diff prob.

Polynomial Identity Testing (PIT)

I/P: Program C that computes an n-variate degree d poly.

$$p(x_1, \ldots, x_n)$$

$$C(a_1, \ldots, a_n) = p(a_1, \ldots, a_n)$$

Q: check if $p \equiv 0$.

i.e., $\forall\, \alpha_1, \ldots, \alpha_n$, $p(\alpha_1, \ldots, \alpha_n) = 0$

No. of coefficients $= \binom{n+d-1}{n} \approx n^d$

How are these probs. related?

For $G(L, R, E)$, consider the bipartite adj. mat.

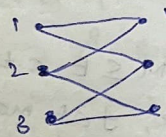
$$A \in \{0,1\}^{n \times n}, \quad n = |L| = |R|$$

$$A(i,j) = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{o.w.} \end{cases} \qquad [i \in L, j \in R]$$

$$P(i,j) = x_{ij}\, A(i,j)$$

(variable)

For



$$P = \begin{bmatrix} x_{11} & x_{12} & 0 \\ x_{21} & 0 & x_{23} \\ 0 & x_{32} & x_{33} \end{bmatrix}$$

$$\det(P) = \sum_{\sigma \in S_n} (-1)^{\operatorname{sgn}(\sigma)} \prod_{i=1}^{n} P(i, \sigma(i))$$

S_n: set of bijections from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$

$\operatorname{sgn}(\sigma)$: signature

$\operatorname{sgn}(\sigma)$ = No. of inversions in σ

$= |\{(i,j) \mid i<j,\ \sigma(i) > \sigma(j)\}|$

$\prod_{i=1}^{n} P(i, \sigma(i))$: $|E|$-variate degree n poly w/ at most $n!$ terms

Each surviving terms corresponds to a valid perfect matching

Obs: G has a p.m. iff $\det(P) \neq 0$

$\Rightarrow$ solving PIT solves P.M.! (w/ a n. simple algo)

Solving PIT

DeMillo-Lipton-Schwartz-Zippel lemma:

Let p be a non-zero n-variate deg-d poly. over $\mathbb{R}$.

Let $S \subseteq \mathbb{R}$. Then

$$\Pr_{a_1, a_2, \ldots, a_n \in S}\left[p(a_1, a_2, \ldots, a_n) = 0\right] \leq \frac{d}{|S|} \quad \text{(indep. of } n!)$$

($a_1, a_2, \ldots, a_n$: chosen indep. & uniformly at random)

3/8/23

Here, sample space $\Omega = \underbrace{S \times S \times \cdots \times S}_{n \text{ times}} = S^n$

The event of interest is the set of zeros,

i.e., $Z \subseteq S^n = \{(a_1, a_2, \ldots, a_n) \in S^n \mid p(a_1, \ldots, a_n) = 0\}$

$\Pr(Z) = ?$

The proba. space is $(\Omega, \mathcal{F}, \Pr)$

$\mathcal{F} = \mathcal{P}(\Omega) = 2^{\Omega}$ (set of all events)

$\Pr : \Omega \to [0,1]$ (proba. of choosing each tuple)

$\Pr(a_1, \ldots, a_n) = \frac{1}{|S|^n}$ (indep. & u.a.r)

It satisfies:

1. $\forall E \in \mathcal{F} \quad 0 \leq \Pr(E) \leq 1$
2. $\Pr(\Omega) = 1$
3. If $E_1, E_2, \ldots, E_n$ are pairwise disjoint,
$$\Pr(\cup E_i) = \sum \Pr(E_i)$$

Some facts:

* Let $E_1, E_2, \ldots, E_n$ be any n events

$$\Pr(\cup E_i) \leq \sum_{i=1}^{n} \Pr(E_i) \quad \text{[union bound]}$$

* **Conditional proba:**

$$\Pr(E|F) = \frac{\Pr(E \cap F)}{\Pr(F)}$$

* E & F are indep. if $\Pr(E|F) = \Pr(E)$

$\Rightarrow \Pr(E \cap F) = \Pr(E)\Pr(F)$

* **Law of total proba:**

$E_1, \ldots, E_n$ form a partition of Ω for disjoint events $E_1, E_2, \ldots, E_n$ s.t. $\cup E_i = \Omega$,

$$\Pr(F) = \sum_{i=1}^{n} \Pr(F \cap E_i)$$

Proof of DLSZ-lemma

[By induction on n]

Base case: $n = 1$

$\Pr_{a \in S}[p(a) = 0] \leq \frac{d}{|S|}$

$Z = \{a \in S \mid p(a) = 0\}$

$|Z| \leq d$

$\Rightarrow \Pr(Z) \leq \frac{d}{|S|}$

Algo: (A)

Sample $a_1, \ldots, a_n \in_r S$ (indep. & u.a.r.)

If $p(a_1, \ldots, a_n) = 0$, say $p \equiv 0$

else, say $p \not\equiv 0$

Note:

① If $p \equiv 0$, then $\Pr(A \text{ says } p \equiv 0) = 1$

② If $p \not\equiv 0$, then

$\Pr(A \text{ says } p \equiv 0) \leq \frac{d}{|S|}$ (error proba.)

$\Rightarrow$ one-sided error algo.

Induction step

$p(x_1, \ldots, x_n) = \sum_{i=0}^{d} x_1^i \, p_i(x_2, \ldots, x_n)$

Find largest i s.t. $p_i \not\equiv 0$

Z_i = set of tuples on which p_i evals. to 0

deg. $p_i \leq d - i$

$\Rightarrow \Pr(Z_i) \leq \frac{d-i}{|S|}$ (induction hypothesis)

$\Pr(Z) = \Pr(Z \cap Z_i) + \Pr(Z \cap \overline{Z_i})$

$= \Pr(Z_i) \cdot \Pr(Z \mid Z_i) + \Pr(\overline{Z_i}) \cdot \Pr(Z \mid \overline{Z_i})$

$\Pr(Z_i) \leq 1$; $\Pr(Z \mid \overline{Z_i}) \leq \frac{i}{|S|}$

we have a lower bound, but we want an upper bound, we can just choose 1

[p_i's are evaluated, now we have a poly. on x_i, w/ deg. i]

$\leq \frac{d-i}{|S|} \cdot 1 + 1 \cdot \frac{i}{|S|}$

$\leq \frac{d}{|S|}$

We can boost the success proba. by repeating the sampling.

i.e., A'

Repeat k times

- Sample $a_1, \ldots, a_n \in_r S$
- If $p(a_1, \ldots, a_n) \neq 0$, say $p \not\equiv 0$

say $p \equiv 0$

Error proba. now is $\left(\frac{d}{|S|}\right)^k$

Running time inc. by factor of k

If $p \equiv 0$, $\Pr(A' \text{ says } p = 0) = 1$

If $p \not\equiv 0$, $\Pr(A' \text{ says } p = 0)$

$= \Pr(p = 0 \text{ in every iteration})$

$= \Pr(Z_1 \cap Z_2 \cap \ldots \cap Z_k) = \Pr(Z_1) \cdot \Pr(Z_2) \cdots \Pr(Z_k)$

$\leq (d/|S|)^k$

4/8/23

Verifying matrix multiplication (Frievald's algorithm)

Given $n \times n$ matrices A, B, C,

Check if $A \cdot B = C$

- can you do this asymptotically faster than matrix-multiplication?

$O(n^3)$ - naive

$O(n^{2.37})$ - complex

conjectured to be $O(n^{2+\epsilon})$ for v. small ϵ

Frievald's algo. → randomized $O(n^2)$

Algo

- choose $\bar{r} \in \{0,1\}^n$
- check if $\underbrace{ABr}_{\underbrace{A(B\bar{r})}_{O(n^2)}} = C\bar{r}$

$\Omega = \{0,1\}^n$

$Pr(\bar{a}) = 1/2^n \quad \forall\, \bar{a} \in \Omega$

Theorem

If $A \cdot B \neq C$, → non-zero matrix $D = AB - C$

$$Pr_{\bar{r} \in \{0,1\}^n}[D\bar{r} = 0]$$

$$Pr_{\bar{r} \in \{0,1\}^n}[AB\bar{r} = C\bar{r}] \leq 1/2$$

Consider that the first row of D is non-zero (WLOG) & $D_{1,1} \neq 0$

$$(D_{1,1}\ D_{1,2}\ \cdots\ D_{1,n}) \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}$$

$$Pr_{r \in \{0,1\}^n}\left(\sum_{i=1}^{n} D_{1,i} r_i = 0\right)$$

$\leq 1/2$ by DLSZ-lemma, where variables are r_i, $i = 1, \ldots, n$ & identically zero implies $AB_{1,:} = C_{1,:}$

$$= Pr_{r \in \{0,1\}^n}\left(r_1 = \frac{-\sum_{i=2}^{n} D_{1,i} r_i}{D_{1,1}}\right)$$

or

$r_1, r_2, \ldots, r_n \in \{0,1\}$ → we choose $r_2, \ldots, r_n$ & finally choose r_1

[Principle of deferred decisions]

$\leq 1/2$

$$= \sum_{b_2, \ldots, b_n \in \{0,1\}} Pr\left(r_1 = \frac{-\sum_{i=2}^{n} D_{1,i} \cdot b_i}{D_{1,1}} \,\middle|\, r_2, \ldots, r_n = b_2, \ldots, b_n\right) \cdot Pr(r_2 = b_2, \ldots, r_n = b_n)$$

$\leq 1/2$

If we want to repeat it until error proba. is ϵ,

we want $(1/2)^n = \epsilon$

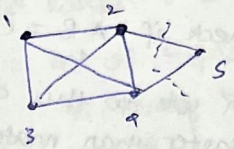
$$n = -\log_2 \epsilon = \log_2 1/\epsilon$$

Min-cut in graphs

$G(V, E)$

A cut set is a set of edges whose removal disconnects G

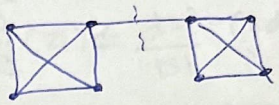
Min-cut is a cut set of smallest size/cardinality



min. cut size 2

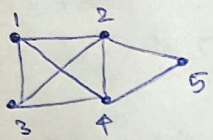
- A rand. algo. (Karger '93)

repeat until 2 vertices remain:
- choose $e \in E$ u.a.r.
- contract e
  $G \leftarrow G \setminus \{e\}$
  & keep parallel edges

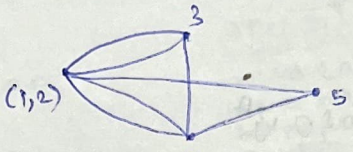


min cut size 1

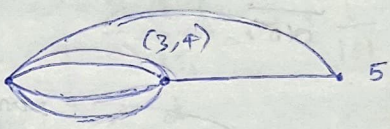
- remaining edges belong to min cut



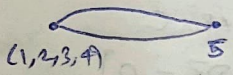
$(1,2)$



$(3,4)$



$(1,2),(3,4)$



min-cut size = 2

If k is size of min. cut,

$|E| \geq \frac{nk}{2}$ (degree of each vertex $\geq k$)

Obs: If G has min cut size k, then G has at least $\frac{kn}{2}$ edges

[If min-cut size $\geq k$, then
$\forall u \in V$, $\deg(u) \geq k$
no. of edges $= \frac{1}{2} \sum_{u \in V} \deg(u) \geq \frac{nk}{2}$]

Fix a min-cut C of size k

$\Pr(\text{no edge from } C \text{ is sampled}) \geq 1 - \frac{2}{n}$

7/8/23

E_i : Event that an edge from C was not sampled in the i^{th} step

$\Pr(E_1) \geq 1 - \frac{2}{n}$

Assume that E_1 occurred

$G' = G \setminus \{e\}$

obs: Every cut in G' is a cut in G

$\Rightarrow$ The min-cut size in G' is also k

$F_i = \bigcap_{j=1}^{i} E_j$ = the event that no edge from C was sampled in the first i iterations

We are interested in $\Pr(F_{n-2})$

$\Pr(E_2 | F_1) \geq 1 - \frac{2}{n-1}$ [no of edges in $G \setminus \{e\} \geq \frac{(n-1)k}{2}$]

$\Pr(E_i | F_{i-1}) \geq 1 - \frac{2}{n-i+1}$

$\Pr(F_{n-2}) = \Pr(F_{n-3}) \cdot \Pr(E_{n-2} | F_{n-3})$

$= \Pr(E_{n-2} | F_{n-3}) \cdot \Pr(E_{n-3} | F_{n-4}) \cdot \dots \cdot \Pr(E_1)$

$\geq \prod_{i=1}^{n-2} \frac{n-i-1}{n-i+1} = \frac{1 \times 2}{n(n-1)}$

$\geq \frac{2}{n(n-1)}$ → v. low success proba.

for some fixed min cut ⇒ $O(n^2)$ mincuts possible, more not possible

$\Pr(C \text{ was not outputted}) \leq 1 - \frac{2}{n(n-1)}$

Repeat k times

$\Pr(C \text{ was not output in any iteration}) \leq \left[1 - \frac{2}{n(n-1)}\right]^k$ ($\star$ $1-x \leq e^{-x}$)

$\leq e^{\frac{-2k}{n(n-1)}}$ $\quad k = \log 1/\epsilon \cdot \frac{n(n-1)}{2}$

⇒ success proba $1-\epsilon$

Max-cut

$G(V, E)$

Partition $V = V_1 \cup V_2$ s.t.

$|E(V_1, V_2)|$ is maximized

no. of edges across the partition/cut

we analyse a $\frac{1}{\alpha}$-approx

- $O(2^n)$ brute-force algo.
- NP-hard
- α-approximation
  - Algo. that outputs a cut of size $\geq \alpha$ (max.), $\alpha < 1$
  - Best known α for poly. time: $\alpha = 0.8$
  - $\alpha \geq 0.94$ not possible unless P = NP

Algo. A

$\forall v \in V$,

put v in V_i w.p. $1/2$ ind. of other vertices

$A(x, r)$ → rand. var.

$A : \{0,1\}^n \to \mathbb{R}$

(n-bit str. denoting a partition → cut size)

Theorem: $E(x) \geq \frac{m}{2}$ (size of the rand. cut; m = total no. of edges)

9/8/23

Expectation (AKA first moment)

- Most basic parameter for a r.v.

$$E(X) = \sum_{\omega \in \Omega} Pr(\omega) \cdot X(\omega)$$

Note: $Pr(X=a) = \sum_{\omega \in \Omega,\ X(\omega)=a} Pr(\omega)$

- Linearity of expectations:

If $X_1, X_2, \ldots, X_n$ are r.v.s, $X = \sum_{i=1}^{n} X_i$, then

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

- Independence:

Criterion: $Pr(X=a \cap Y=b) = Pr(X=a) \cdot Pr(Y=b) \quad \forall a, b$

Analysis of A

$\forall e \in E$, define X_e = indicator r.v.s → $E(X) = Pr(X=1)$ for those

$$X_e = \begin{cases} 1, & \text{if } e \in \text{cut} \\ 0, & \text{o.w.} \end{cases}$$

indep.

$$\text{Cut-size} = X = \sum_{e \in E} X_e$$

$$= \sum_{\substack{e \in E \\ e=(u,v)}} Pr(u \in V_1 \ \& \ v \in V_2) + Pr(u \in V_2 \ \& \ v \in V_1)$$

($Pr(u \in V_1 \ \& \ v \in V_2) = 1/2$, $Pr(u \in V_2 \ \& \ v \in V_1) = 1/2$)

$$= m \times \frac{1}{2} = \frac{m}{2}$$

- doesn't necessarily hold for 3 or more r.v.s ⇒ pairwise indep. r.v.s

$\{X_{e_1}$ is indep of X_{e_2} $\forall e_1, e_2 \in E$

- if e_1 & e_2 are both incident on some $v \in V$, knowing X_{e_1} does not influence X_{e_2}
- if they don't share a vertex, they are indep.

Theorem: Every graph G w/ m edges contains a cut of size $\geq \frac{m}{2}$

Pf: $E(X) = \sum_{\text{cuts } C} Pr(C) \cdot \text{size}(C) = \frac{m}{2}$

If $\text{size}(C) < \frac{m}{2} \ \forall C$

$\Rightarrow E(X) < \frac{m}{2}$

$\Rightarrow \exists C$ w/ size $\geq m/2$ [Probabilistic method]

Derandomizing the algo.

→ naive: Brute force over the 2^n possible random choices

Obs: We have assumed indep. of choosing $V_i \ \forall v \in V$, however, we only reqd. pairwise indep.

Thm: We can construct $n = 2^b - 1$ pairwise-indep. rand. bits from b completely rand. bits

Example:

$b = 3, n = 7$

rand. bits: Y_1, Y_2, Y_3

001	$X_1 = Y_1$
010	$X_2 = Y_2$
011	$X_3 = Y_1 \oplus Y_2$
100	$X_4 = Y_3$
101	$X_5 = Y_3 \oplus Y_1$
110	$X_6 = Y_3 \oplus Y_2$
111	$X_7 = Y_3 \oplus Y_1 \oplus Y_2$

(At any pt., use the newest var. & do XOR w/ a previous combo not containing it)

Modified algo.

- sample $\log n$ bits u.a.r.
- construct n pairwise indep rand-bits defining a cut
- output the size of the cut

No. of rand. choices $= 2^{\log n} = n$

$\Rightarrow$ Brute-forcing over all $\log n$ bit strings obtains a poly-time deterministic algo.

. same results as before

$\mathbb{E}(X) = m/2$ ↳ over the $\log n$ bits

10/8/23

Alternate procedure for derandomization

Method of conditional expectation

---

conditional expectation: Given r.v.s X, Y,

$$\mathbb{E}(X \mid Y = y) = \sum_x x \cdot \Pr(X = x \mid Y = y)$$

Fact: 1) $\mathbb{E}(X) = \sum_y \Pr(Y = y)\, \mathbb{E}(X \mid Y = y)$

2) $\mathbb{E}\left(\sum_{i=1}^{n} X_i \mid Y = y\right) = \sum_{i=1}^{n} \mathbb{E}(X_i \mid Y = y)$

$$\mathbb{E}(X) = \Pr(v_1 \in V_1) \cdot \mathbb{E}(X \mid v_1 \in V_1) + \Pr(v_1 \in V_2) \cdot \mathbb{E}(X \mid v_1 \in V_2)$$

$$= \frac{1}{2} \cdot \mathbb{E}(X \mid v_1 \in V_1) + \frac{1}{2} \cdot \mathbb{E}(X \mid v_1 \in V_2) = \frac{m}{2} \text{ (from analysis before)}$$

Put $v_1 \in V_1$ arbitrarily

$$\mathbb{E}(X \mid v_1 \in V_1) = \Pr(v_2 \in V_1) \cdot \mathbb{E}(X \mid v_1 \in V_1 \,\&\, v_2 \in V_1) + \Pr(v_2 \in V_2) \cdot \mathbb{E}(X \mid v_1 \in V_1 \,\&\, v_2 \in V_2)$$

($\Pr(v_2 \in V_1) = 1/2$, $\Pr(v_2 \in V_2) = 1/2$; $\mathbb{E}(X \mid v_1 \in V_1) = m/2$)

The larger of these must be $\geq m/2$ ↳ keep choosing the larger

$$\mathbb{E}(X \mid v_1, v_2, \ldots, v_{i-1}) = \Pr(v_i \in V_1) \cdot \mathbb{E}(X \mid v_1, v_2, \ldots, v_{i-1}, v_i \in V_i) + \Pr(v_i \in V_2) \cdot \mathbb{E}(X \mid v_1, v_2, \ldots, v_{i-1}, v_i \in V_2)$$

↳ $\geq \frac{m}{2}$ (from prev. steps)

↳ No. of cut edges b/w $v_1, \ldots, v_{i-1}$ + No. of cut edges incident on v_i + $\frac{1}{2} \times$ No. of other edges (almost one vertex incident on $v_1, \ldots, v_n$)

↳ recursing

sequential greedy algo:

- Fix some order of vertices $v_1, \ldots, v_n$
- For $i = 1, \ldots, n$:
  - Add v_i to V_1 if no. of cut edges from v_i to $v_1, \ldots, v_{i-1}$ is larger
  - Else, add v_i to V_2

11/9/23

Quicksort

- Randomly choose a pivot
- Partition array based on pivot $\longrightarrow O(n)$
- Recursively sort the two parts

Running time:

worst-case $O(n^2)$

Expected $\Theta(n \log n)$

Running time $\propto$ no. of comparisons

$\bar{a} = a_1 a_2 \ldots a_n \xrightarrow{\text{sorted}} b_1 b_2 \ldots b_n$

$$X_{ij} = \begin{cases} 1, & \text{if } b_i \text{ \& } b_j \text{ were compared at some pt. in Quicksort} \\ 0, & \text{o.w.} \end{cases}$$

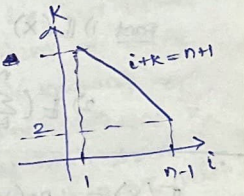
$\longrightarrow$ this occurs when the first pivot in $b_i, b_{i+1}, \ldots, b_{j-1}, b_j$ is b_i or b_j

$$X = \sum_{i<j} X_{ij}$$

$$E(X_{ij}) = Pr(X_{ij} = 1) = \frac{2}{j-i+1}$$

$$E(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1}$$

$$= \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k} = \sum_{k=2}^{n} \sum_{i=1}^{n-k+1} \frac{2}{k} = \sum_{k=2}^{n} (n-k+1)\frac{2}{k}$$



$$= 2(n+1)\sum_{k=2}^{n} \frac{1}{k} - 2(n-1)$$

$\log n + \Theta(1)$ ★ Harmonic sum

$$= 2n\log n + \Theta(n)$$

$\Rightarrow$ Expected $\Theta(n \log n)$

We will show later that

$$Pr(X > cn\log n) \leq 1/n$$

A similar analysis holds if we randomly permute the array & then choose the first/last/some fixed index element as the pivot repeatedly

A few standard r.v.s

① Bernoulli r.v. $\equiv$ Indicator r.v.

$$X = \begin{cases} 1, & \text{w.p. } p \\ 0, & \text{w.p. } 1-p \end{cases} \qquad E(X) = Pr(X=1) = p$$

② Binomial r.v.

Let $X_1, X_2, \ldots, X_n$ be Bernoulli trials, i.e., rand. expts. w/ success proba. p

A binomial r.v. counts the no. of successes in n Bernoulli trials, i.e., $X = \sum_{i=1}^{n} X_i$

Range of $X = \{0, 1, \ldots, n\}$

$$Pr(X = i) = \binom{n}{i} p^i (1-p)^{n-i} \longrightarrow \text{proba. mass fn.}$$

$$E(X) = np$$

(3) Geometric r.v.

* Counts the no. of Bernoulli trials before the first success
* Range of the r.v. = $\mathbb{N}$
* $Pr(X = i) = (1-p)^{i-1} p$
* Memorylessness

$$Pr(X = n+k \mid \underbrace{X > k}_{\text{no success in first } k \text{ trials}}) = Pr(X = n)$$

* $E(X) = Pr(Y=1) \cdot \underbrace{E(X|Y=1)}_{1} + Pr(Y=0) \cdot E(X|Y=0)$ — Y: first trial
  - $Pr(Y=1) = p$
  - $Pr(Y=0) = 1-p$
  - $E(X|Y=0)$: $X = 1 + Z$, Z being an identical geometric r.v. [memorylessness]

$$= p + (1-p)(E(1+Z))$$
$$= 1 + (1-p)E(Z) = 1 + (1-p)E(X)$$

$$\Rightarrow pE(X) = 1 \Rightarrow E(X) = \frac{1}{p}$$

Coupon collector problem

n coupons

A box of chocolate has one of the coupons u.a.r.

Q. How many boxes must be collected to obtain one copy of every coupon?

X_i = No. of boxes that are bought after $i-1$ coupons are obtained before obtaining the i^{th} coupon

X_i is a geometric r.v. with success proba. $\frac{n-i+1}{n}$

No. of boxes, $X = \sum_{i=1}^{n} X_i$

$$E(X) = \sum_{i=1}^{n} E(X_i) = \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{j=1}^{n} \frac{1}{j}$$

$$= n \log n + \Theta(n)$$

1/15/23

Markov's inequality

If X is a non-neg. r.v. & $a > 0$, then

$$Pr(X \geq a) \leq \frac{E(X)}{a}$$

i.e., $Pr(X \geq k E(X)) \leq \frac{1}{k}$

Pf: Consider the indicator r.v. $I = \begin{cases} 1, & X \geq a \\ 0, & \text{o.w.} \end{cases}$

$Pr(X \geq a) = Pr(I = 1) = E(I)$

Note that $I \leq \frac{X}{a}$

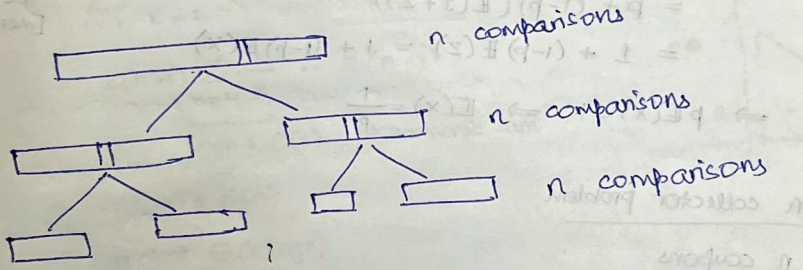
$\Rightarrow E(I) \leq E\left(\frac{X}{a}\right) = \frac{E(X)}{a}$

$\therefore Pr(X \geq a) \leq \frac{E(X)}{a}$

A bound for Quicksort

X = No. of comparisons, $E(X) = 2n\log n + \Theta(n)$

Thm: $Pr(X > k n \log n) \leq \frac{1}{n}$, k is a constant indep. of n



Each level performs n comparisons

$\Rightarrow$ Bounding no. of levels bounds no. of comparisons

Fix an element a of the array

X_i = size of the array containing a in the i^{th} level of recursion

$X_0 = n$

$$E(X_i) = \sum_{t > 0} Pr(X_{i-1} = t) \cdot E(X_i \mid X_{i-1} = t)$$

We want this to be $\alpha \cdot X_{i-1}$ to bound levels as $\Theta(\log n)$

Choosing an element in 25^{th} - 75^{th} percentile bounds the size of the array with a to be at most $\frac{3}{4} X_{i-1}$

$$\mathbb{E}(x_i \mid x_{i-1} = t) \leq \frac{1}{2} \cdot \frac{3}{4} x_i + \frac{1}{2} \cdot x_i = \frac{7}{8} t$$

$$\Rightarrow \mathbb{E}(x_i) \leq \sum_{t>0} \Pr(x_{i-1} = t) \cdot \frac{7}{8} t = \frac{7}{8} \sum_{t>0} \Pr(x_{i-1} = t) \cdot t$$

$$\mathbb{E}(x_i) \leq \frac{7}{8} \mathbb{E}(x_{i-1})$$

$$\Rightarrow \mathbb{E}(x_i) \leq \left(\frac{7}{8}\right)^i n$$

Choose $i = 3\log_{8/7} n$

$$\Rightarrow \mathbb{E}(x_i) \leq n \left(\frac{7}{8}\right)^{3\log_{8/7} n} = n\left(\frac{1}{n^3}\right)$$

$$\leq \frac{1}{n^2}$$

We have only considered the bound for a single element a.

By Markov's inequality, for $a = 1$,

$$\Pr(x_i > 1) \leq \frac{1}{n^2} \quad \text{for } i = 3\log_{8/7} n$$

Let E_a be the event that a exists in a subarray of size > 1 after $i = 3\log_{8/7} n$ recursive steps.

$$\Pr(E_a) \leq \frac{1}{n^2}$$

$$\Pr(\exists a \text{ s.t. } E_a) \leq \sum \Pr(E_a) = n\left(\frac{1}{n^2}\right)$$

$$\leq 1/n$$

Variance of a r.v.

$$\text{Var}(X) = \mathbb{E}((X - \mathbb{E}(X))^2), \quad \mathbb{E}(X) = \mu$$

$$= \mathbb{E}(X^2 - 2\mu X + \mu^2)$$

$$= \mathbb{E}(X^2) - \mu^2$$

Properties

① $\text{var}(kX) = k^2 \text{var}(X)$

② $\text{var}(X+Y) = \text{var}(X) + \text{var}(Y) + 2\text{cov}(X,Y)$,

$\text{cov}(X,Y) = \mathbb{E}((X - \mathbb{E}(X))(Y - \mathbb{E}(Y))) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$

If X & Y are independent,

$\text{var}(X+Y) = \text{var}(X) + \text{var}(Y)$

* $X \sim \text{Bin}(n, p)$

$X = X_1 + X_2 + \cdots + X_n$, $X_i \sim$ Bernoulli(p)

$\text{Var}(X_i) = \mathbb{E}(X_i^2) - \mathbb{E}(X_i)^2 = p - p^2 = p(1-p)$

$\text{Var}(X) = \sum_{i=1}^{n} \text{var}(X_i) = np(1-p)$

* $X \sim Geom(p)$

$$E(X^2) = p \cdot 1 + (1-p) \cdot E((X+1)^2) \quad \text{(memorylessness)}$$
$$= p + (1-p)\left[E(X^2) + 2\underbrace{E(X)}_{1/p} + 1\right]$$
$$= 1 + 2\left(\frac{1-p}{p}\right) + (1-p)E(X^2)$$

$$p\,E(X^2) = \frac{2-p}{p} \Rightarrow E(X^2) = \frac{2-p}{p^2}$$

$$Var(X) = E(X^2) - E(X)^2 = \frac{2-p}{p^2} - \frac{1}{p^2} = \frac{1-p}{p^2}$$

Chebyshev's inequality

Let X be any r.v. & $a > 0$

$$Pr(|X-\mu| \geq a) \leq \frac{Var(X)}{a^2}$$

i.e., $Pr(\mu - a \leq X \leq \mu + a) \geq 1 - \frac{Var(X)}{a^2}$

Pf: Consider $Y = (X-\mu)^2$

Y is a non-neg. r.v.

$\Rightarrow$ By Markov's inequality, $Pr(Y \geq a^2) \leq \frac{E(Y)}{a^2}$

$$Pr(|X-\mu| \leq a) \leq \frac{E((X-\mu)^2)}{a^2} = \frac{Var(X)}{a^2}$$

Randomness-efficient probability amplification

Consider Freivald's algo.

$$\underbrace{AB \overset{?}{=} C}_{n \times n}$$

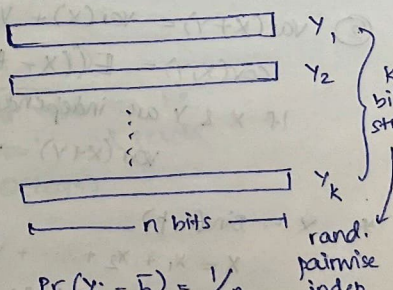
* We use n rand. bits
* $AB \neq C \Rightarrow Pr(\text{Error}) \leq 1/2$
* $AB = C \Rightarrow Pr(\text{Error}) = 0$
* Repeating the event k times, $Pr(\text{Error}) \leq 1/2^k$
  w/ no. of rand. bits used $= nk$

Alternative:

Repeating k times,

$Pr(\text{Error}) \leq 1/k$, $k \leq n$

No. of rand. bits $= 2n$



$Pr(Y_i = \bar{b}) = 1/2^n$

* The construction of k rand. pairwise indep. bit strings of len. n can be done w/ $n \log k$ bits
  - For each bit pos., generate $\log k$ purely rand. bits
  - Using these bits, gen. k pairwise indep. rand. bits
  - ~~Do the co~~ $\Rightarrow \log k \times n$

* Here, we can actually do it w/ ~~$n \log k$~~ $2n$ bits

Let $Y_1, Y_2, \dots, Y_k$ be pairwise indep. n-bit strings

$$X_i = \begin{cases} 1, & \text{if } ABY_i \neq CY_i \\ 0, & \text{o.w.} \end{cases}$$

Given $AB \neq C$, $E(X_i) \geq 1/2$

Consider $X = \sum_{i=1}^{k} X_i$

$E(X) \geq k/2$

Error proba. $= Pr(X = 0)$

$$Pr(X=0) \leq Pr\left(|X - E(X)| \geq k/2\right) \leq \frac{var(X)}{(k/2)^2}$$

$$Pr(X=0) \leq \frac{4}{k^2} var(X)$$

$$var(X) = \sum_{i=1}^{k} var(X_i) + 2 \sum_{i<j} Cov(X_i, X_j) \quad (Y_i, Y_j \text{ are pairwise indep.} \Rightarrow Cov(X_i, X_j) \to 0)$$

$$= \sum_{i=1}^{k} var(X_i)$$

$$X_i = \begin{cases} 1, & \text{w.p. } p \\ 0, & \text{w.p. } 1-p \end{cases}$$

$$var(X_i) = p(1-p) \leq 1/4$$

$$\Rightarrow Pr(X=0) \leq \frac{4}{k^2} \cdot (k \times 1/4) = 1/k \quad \Rightarrow \text{Error bound proven}$$

* Consider $\mathbb{Z}_p = \{0, 1, \dots, p-1\}, +, \cdot$ ~~a finite field~~ modulo p

If p is prime, the following properties hold.

1. associativity
2. commutativity
3. additive identity
4. multiplicative identity
5. Unique additive inverse $(\forall z \in \mathbb{Z}_p)$
6. Unique multiplicative inverse $(\forall z \in \mathbb{Z}_p \setminus \{0\})$ → p's primality is reqd here

A set w/ these operations satisfying the above properties is a field.

$\mathbb{Z}_p$ is a finite field for prime p. There also exist finite fields $\mathbb{Z}_p$ for non-prime p

17/8/23

Theorem: Let p be a prime no. (for our proof, we want $p > 2$)

choose $a, b \in_r \mathbb{Z}_p = \{0, 1, \ldots, p-1\}$

$$x_i = \underbrace{a \cdot i + b}_{\text{operations modulo } p} \quad , \; i \le p$$

$\longrightarrow$ forms a finite field

① x_i is uniformly distributed over $\mathbb{Z}_p$

② $\{x_i\}$ are pairwise ind.

① i.e., $\forall c \in \mathbb{Z}_p$

$$\Pr_{a,b \in_r \mathbb{Z}_p}[ai + b = c] = 1/p$$

Fix $a \in \mathbb{Z}_p$

$\Rightarrow b = \underbrace{c - ai}_{\text{additive inverse}}$

$\Rightarrow p$ of p^2 combos yield c

② i.e., $\forall c_1, c_2 \in \mathbb{Z}_p$,

$$\Pr_{a,b \in_r \mathbb{Z}_p}[a \cdot i + b = c_1 \wedge a \cdot j + b = c_2]$$

$\longrightarrow$ yields unique a, b

$$a = \underbrace{\frac{c_1 - c_2}{i - j}}_{\text{multiplicative inverse}}$$

$$b = \frac{c_1 j - c_2 i}{i - j}$$

$$= \Pr_{a,b \in_r \mathbb{Z}_p}[a \cdot i + b = c_1] \cdot \Pr_{a,b \in_r \mathbb{Z}_p}[a \cdot j + b = c_2]$$

$$= 1/p \cdot 1/p$$

$$= 1/p^2$$

$\Rightarrow 1/p^2$

Theorem: For every prime p & integer k, $\exists$ a unique finite field of cardinality p^k, up to isomorphism

[Galois field, $GF(p^k)$]

Construction of $GF(4)$ [smallest non-prime finite field] $(p = k = 2)$

Consider polynomials of deg. ≤ 2 over $GF(2)$,

i.e., deg. ≤ 2 & coeffs. from $GF(2)$, i.e., $\{0, 1\}$

We have 8 polynomials:

0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
000	001	010	011	100	101	110	111

Irreducible polynomials of deg. 2 over $GF(2)$

↳ cannot be resolved to a product of lesser polynomials

~~$x^2 + x$~~ $= x(x+1)$ not irreducible

$x^2 + 1$ has a root of 1, not irreducible

In fact, $(x+1)^2 = x^2 + 1 + \underbrace{(x + x)}_{0} = x^2 + 1$

x^2+x+1 is irreducible

$GF(4) = \{0, 1, x, x+1\}$ $+, \cdot$ performed modulo $1+x+x^2$ the irreducible poly. (of deg. 2)

$x \cdot x \equiv (1+x) \bmod (1+x+x^2)$

$x \cdot (1+x) \equiv 1 \bmod (1+x+x^2)$

$(1+x) \cdot (1+x) \equiv x \bmod (1+x+x^2)$

↳ this always exists, apparently

Fact: $\forall l \geq 0$, $x^{2 \cdot 3^l} + x^{3^l} + 1$ is irreducible over $GF(2)$

Chernoff - Hoeffding bounds

$X_1, X_2, \ldots, X_n$, $X_i = \begin{cases} 1, & \text{w.p. } p_i \\ 0, & \text{w.p. } 1-p_i \end{cases}$

$X = \sum_{i=1}^{n} X_i$, X_i's are indep., $\mu = E[X]$

① $\Pr(|X-\mu| \geq t) \leq e^{-2t^2/n}$, $t > 0$

② $\Pr(X \geq (1+\delta)\mu) \leq e^{-\delta^2\mu/3}$

③ $\Pr(X \leq (1-\delta)\mu) \leq e^{-\delta^2\mu/2}$

} $0 \leq \delta \leq 1$

Hoeffding's extension:

$E(X_i) = \mu$

X_i's are indep. & take values in $[a,b]$,

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| \geq \epsilon\right] \leq 2e^{-2n\epsilon^2/(b-a)^2}$$

18/8/23

Consider a 2-sided error algo.,

w/ $\Pr(\text{error}) \leq \frac{1}{2} - \epsilon$

To boost the success proba., we

- repeat the algo. k times
- take the majority as the ans.

$X_1, X_2, \ldots, X_k$, $X_i = \begin{cases} 1, & \text{if } i^{th} \text{ iteration answers correctly} \\ 0, & \text{o.w.} \end{cases}$

$E(X_i) \geq \frac{1}{2} + \epsilon$

$X = \sum_{i=1}^{k} X_i$ = No. of successful trials

$E(X) \geq \frac{k}{2} + k\epsilon$

Proba. of error = $\Pr(X \leq k/2)$

$\Pr(E(X) - X > k\epsilon) \leq e^{-2k^2\epsilon^2/k}$ [Chernoff bound ①]

$\leq e^{-2k\epsilon^2}$

When $\epsilon = 1/n^c$, $k = \frac{1}{\epsilon^2}\log n$ is poly(n)

choosing $k = \frac{1}{\epsilon^2}\log n$, $\leq 1/n^2$

Even $k = \frac{n}{\epsilon^2}$ is poly(n), w/ error $\leq e^{-n}$!

Moment generating fn.

$$M_X(t) = E(e^{tX}) = E\left(\sum_{i\geq 0} \frac{t^i X^i}{i!}\right)$$

$$= \sum_{i\geq 0} \frac{t^i}{i!} \underbrace{E(X^i)}_{i^{th} \text{ moment of } X}$$

Pf of Chernoff bounds:

$$\Pr(X \geq m) = \Pr(\underbrace{e^{tX}}_{\text{non-neg. r.v.}} \geq e^{tm}), \quad t > 0$$

$$\leq \frac{E(e^{tX})}{e^{tm}} \quad \text{(Markov's ineq.)}$$

$$E(e^{tX}) = E\left(e^{t\sum_{i=1}^{n} X_i}\right) = E\left(\prod_{i=1}^{n} e^{tX_i}\right)$$

$$= \prod_{i=1}^{n} E(e^{tX_i}) \quad [\text{indep r.v.s}]$$

$$E(e^{tX_i}) = p \cdot e^{t\cdot 1} + \underbrace{(1-p)}_{q} \cdot 1 = pe^t + q = p(e^t - 1) + 1$$

$$E(e^{tX}) = (pe^t + q)^n$$

$$\Pr(X \geq m) \leq \frac{(pe^t + q)^n}{e^{tm}}$$

Let $m = (p+r)n$, $\mu = np$

$$\Pr(X \geq (p+r)n) \leq \frac{(pe^t + q)^n}{e^{t(p+r)n}}$$

calculus ↓

$$\leq \left(\left(\frac{p}{p+r}\right)^{p+r} \left(\frac{q}{q-r}\right)^{q-r}\right)^n$$

strongest version of the bound but difficult to apply

Weakening & simplifying w/ approximations:

$$E(e^{tX_i}) = p(e^t - 1) + 1 \leq e^{p(e^t-1)}$$

$$E(e^{tX}) = \prod_{i=1}^{n} E(e^{tX_i}) \leq e^{np(e^t-1)} = e^{\mu(e^t-1)}$$

$$\Rightarrow \Pr(X \geq (1+\delta)\mu) \leq \frac{e^{\mu(e^t-1)}}{e^{t\mu(1+\delta)}} \rightarrow \text{Minimized for } t = \ln(1+\delta)$$

$$\leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}$$

calculus/analysis ↓

$$\leq e^{-\delta^2\mu/3}, \quad \delta < 1$$

* K servers, n jobs

Allocate jobs to servers in a decentralized way

simple way: Allocate each job to a server to a server uniformly at random

Expected load = n/k ⟶ "balls-in-bins"

21/8/23

Balls & bins

m balls, n bins

Throw the balls into the bins u.a.r.

Coupon collector prob. - what is the val. of m s.t. every bin contains at least one ball?

Hashing - How large should n be compared to m s.t. no bin has a lot of balls?

Birthday problem

what should m be so that the proba. that ∃ a bin w/ more than 1 ball is at least ½?

B_i - the event that the i^{th} ball lands in a bin of its own

we want $\Pr\left(\bigcap_{i=1}^{m} B_i\right)$.

$\Pr\left(\bigcap_{i=1}^{m} B_i\right)$ ⟶ we want $\Pr(B_i \mid B_{i-1}, \dots, B_1) = \frac{n-i+1}{n}$

$$= \prod_{i=1}^{m} \Pr(B_i \mid B_{i-1}, \dots, B_1) = \prod_{i=1}^{m} \frac{n-i+1}{n} = \prod_{i=1}^{m} \left(1 - \frac{i-1}{n}\right)$$

$$\leq \prod_{i=1}^{m} e^{-(i-1)/n} = e^{-\sum_{i=1}^{m} \frac{i-1}{n}} = e^{-m(m-1)/2n} < \frac{1}{2}$$

$$\Rightarrow m = \Theta(\sqrt{n})$$

For hashing, this tells that a hash table of size ~~smaller than~~ $o(n^2)$ is likely to find collisions (w.p. > ½)

E_i = event that the i^{th} ball doesn't land in ~~one of t.~~ a separate bin given that the first $i-1$ balls landed in separate bins

$\Pr(E_i) = \frac{i-1}{n}$ $\quad \Pr\left(\bigcup_{i=1}^{m} E_i\right) \leq \sum_{i=1}^{m} \frac{i-1}{n} = \frac{m(m-1)}{2n}$

If $m < \sqrt{n}$, $\Pr\left(\exists \text{ a ball that doesn't land in a separate bin}\right) < ½$

If n balls are thrown into n bins, what is the max. load? ⊖ $\frac{3\log n}{\log\log n}$ w.p. $1 - 1/n$

max. no. of balls in some bin

Thm: If n balls are thrown into n bins u.a.r., then w.p. $\to 1 - 1/n$, the max. load is at most $\frac{3\log n}{\log\log n}$

Pf: Fix a bin i.

$$\Pr(\text{bin } i \text{ has } k \text{ balls}) \le \binom{n}{k}\left(\frac{1}{n}\right)^k$$

★ $\binom{n}{k} \le \left(\frac{ne}{k}\right)^k$

$$\le \left(\frac{ne}{k}\cdot\frac{1}{n}\right)^k = \left(\frac{e}{k}\right)^k$$

$$\Pr(\exists \text{ bin } i \text{ s.t. bin } i \text{ has } k \text{ balls}) \le n\left(\frac{e}{k}\right)^k \quad \text{(Union bound)}$$

$$\le 1/n \quad \text{(reqd.)}$$

what val. of k do we choose?

we can substitute $\frac{3\log n}{\log\log n}$ to verify it works.

$X_1, X_2, \ldots, X_n$ – r.v.s corresponding to no. of balls in individual bins

dependent, their sum is fixed → something indep would be nicer to work with

$$X_i \sim \text{Bin}(m, 1/n)$$

$$\Pr(X_i = k) = \binom{m}{k}\left(\frac{1}{n}\right)^k\left(1-\frac{1}{n}\right)^{m-k}$$

$$= \frac{m!}{k!\,(m-k)!}\left(\frac{1}{n}\right)^k\left(1-\frac{1}{n}\right)^{m-k}$$

$$= \frac{(m-k+1)\cdots(m)}{k!\;n^k}\left(1-\frac{1}{n}\right)^{m-k}$$

$$\approx \left(\frac{m}{n}\right)^k\left(\frac{1}{k!}\right)e^{-m/n} \quad \text{(assuming } m, n \gg k\text{)}$$

$\lambda = m/n$

Poisson dbn.

$X \sim \text{Poi}(\lambda)$

$$\Pr(X = k) = \frac{e^{-\lambda}\lambda^k}{k!} \qquad E(X) = \lambda$$

Thm: If $X \sim \text{Bin}(n, p)$ & $\lim_{n\to\infty} np = \lambda$ (indep. of n), for every fixed k, $\lim_{n\to\infty} \Pr(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}$

23/6/23

$X_1 \sim \text{Poi}(\lambda_1)$, $X_2 \sim \text{Poi}(\lambda_2)$ [indep.]

then $X_1 + X_2 \sim \text{Poi}(\lambda_1 + \lambda_2)$

Pf: $\Pr(X_1 + X_2 = j) = \sum_{k=0}^{j} \Pr(X_1 = k) \cdot \Pr(X_2 = j-k)$

$$= \sum_{k=0}^{j} \frac{e^{-\lambda_1} \cdot \lambda_1^k}{k!} \cdot \frac{e^{-\lambda_2} \lambda_2^{j-k}}{(j-k)!}$$

$$= \frac{e^{-(\lambda_1+\lambda_2)}}{j!} \sum_{k=0}^{j} \lambda_1^k \lambda_2^{j-k} \frac{j!}{k!(j-k)!}$$

$$= \frac{e^{-(\lambda_1+\lambda_2)} (\lambda_1+\lambda_2)^j}{j!}$$

Chernoff bounds for Poisson r.v.s

$X \sim \text{Poi}(\lambda)$

$$E(e^{tX}) = \sum_{k=0}^{\infty} \Pr(X=k) \cdot e^{tk} = \sum_{k=0}^{\infty} \frac{e^{-\lambda} \cdot \lambda^k}{k!} \cdot e^{tk}$$

$$= e^{-\lambda} \sum_{k=0}^{\infty} \frac{e^{tk} \lambda^k}{k!} = e^{-\lambda} \sum_{k=0}^{\infty} \frac{(e^t \lambda)^k}{k!} \cdot e^{-\lambda e^t} \cdot e^{\lambda e^t}$$

$$= e^{\lambda(e^t-1)} \sum_{k=0}^{\infty} \frac{e^{-\lambda e^t} (e^t \lambda)^k}{k!}$$

$$= e^{\lambda(e^t-1)}$$

Let $x \geq \lambda$.

$$\Pr(X \geq x) = \Pr(e^{tX} \geq e^{tx})$$

$$\leq \frac{e^{\lambda(e^t-1)}}{e^{tx}}$$ Find t that minimizes this. $t = \log x/\lambda$

$$\leq \frac{e^{-\lambda}(e\lambda)^x}{x^x}$$

Let $x \leq \lambda$

$$\Pr(X \leq x) = \Pr(e^{tX} \geq e^{tx}), \quad t < 0$$

$$\leq \frac{e^{-\lambda}(e\lambda)^x}{x^x}$$

Thm: Let $\{X_i^m\}_{1\le i\le n}$ be the balls-in-bins dbn. w/ m balls & n bins

Let $\{Y_i^m\}_{1\le i\le n}$ be indep. ~~Poisson r.v.s w/~~ $\sim Poi(m/n)$

The dbn. $\{X_i^m\}_{1\le i\le n}$ conditioned on $\sum_{i=1}^n Y_i^m = k$

is identical to $\{X_i^k\}_{1\le i\le n}$

Pf: $\Pr(\bigcap_{i=1}^n (X_i = k_i))$ [multinomial dbn.] $(\sum_{i=1}^n k_i = k)$

$$= \binom{k}{k_1}\cdot\binom{k-k_1}{k_2}\cdot\cdots\cdot\binom{k-k_1-k_2-\cdots-k_{n-1}}{k_n}\cdot\left(\frac{1}{n}\right)^{k_1}\cdot\left(\frac{1}{n}\right)^{k_2}\cdot\cdots\cdot\left(\frac{1}{n}\right)^{k_n}$$

$$= \frac{k!}{k_1!\,k_2!\cdots k_n!}\left(\frac{1}{n}\right)^k$$

$$\Pr\left(\bigcap_{i=1}^n (Y_i^m = k_i) \,\middle|\, \sum_{i=1}^n Y_i^m = k\right) = \frac{\Pr\left[\bigcap_{i=1}^n (Y_i^m = k_i)\right]\cdot \Pr\left[\sum_{i=1}^n Y_i^m = k \,\middle|\, \bigcap_{i=1}^n (Y_i^m = k_i)\right]}{\Pr\left(\sum_{i=1}^n Y_i^m = k\right)}$$

indep. $\Rightarrow$ prod. ; $\sum Y_i^m = k$ is determined by $\bigcap (Y_i^m = k_i)$, so that factor $= 1$

↳ sum of Poisson $\Rightarrow Poi(m)$

$$= \frac{\prod_{i=1}^n \Pr(Y_i^m = k_i)}{\frac{e^{-m}m^k}{k!}} = \left(\prod_{i=1}^n \frac{e^{-m/n}\cdot\left(\frac{m}{n}\right)^{k_i}}{k_i!}\right)\cdot\frac{e^m k!}{m^k}$$

$$= \frac{e^{-m}\left(\frac{m}{n}\right)^k}{k_1!\,k_2!\cdots k_n!}\cdot\frac{e^m k!}{m^k} = \frac{k!}{k_1!\,k_2!\cdots k_n!}\left(\frac{1}{n}\right)^k$$

Thm: $\{X_i^m\}_{1\le i\le n}$, balls-in-bins dbn. w/ m balls & n bins

$\{Y_i^m\}_{1\le i\le n}$, indep. $Poi(m/n)$

Let f be a non-negative fn.

$$\mathbb{E}(f(X_1^m, X_2^m, \ldots, X_n^m)) \le e\sqrt{m}\,\underbrace{\mathbb{E}(f(Y_1^m, Y_2^m, \ldots, Y_n^m))}_{\text{no conditioning!}}$$

How to apply this?

Suppose we want to answer w/ what prob. no. of bins with no balls is $\ge k$

Define $f(x_1,\ldots,x_n) = \begin{cases} 1, \text{ if } |\{i : x_i = 0\}| \ge k \\ 0, \text{ o.w.} \end{cases}$

$\mathbb{E}(f(X_1^m,\ldots,X_n^m)) = \Pr(\text{no. of bins w/ no balls is} \ge k)$

24/8/23

Pf: $\mathbb{E}(f(Y_1^m, Y_2^m, \ldots, Y_n^m)) = \sum_{k\ge 0} \mathbb{E}(f(Y_1^m,\ldots,Y_n^m) \mid \sum_{i=1}^n Y_i^m = k)\cdot \Pr(\sum_{i=1}^n Y_i^m = k)$

$$\ge \mathbb{E}(f(Y_1^m,\ldots,Y_n^m) \mid \sum_{i=1}^n Y_i^m = m)\cdot \Pr(\sum_{i=1}^n Y_i^m = m)$$

$\Pr(\sum_{i=1}^n Y_i^m = m)$: $\sum Y_i^m \sim Poi(m)$

$$= \mathbb{E}(f(Y_1^m,\ldots,Y_n^m))\cdot\frac{e^{-m}m^m}{m!}$$

$$\mathbb{E}(f(Y_1^m, \dots, Y_n^m)) \geq \mathbb{E}(f(X_1^m, \dots, X_n^m)) \cdot \frac{e^{-m} \cdot m^m}{\sqrt{2\pi m}} \cdot \left(\frac{e}{m}\right)^m$$

$$\mathbb{E}(f(X_1^m, \dots, X_n^m)) \leq e\sqrt{m}\, \mathbb{E}(f(Y_1^m, \dots, Y_n^m))$$

$[e \leq \sqrt{2\pi}]$

Stirling's approx.
$m! \leq \sqrt{2\pi m}\left(\frac{m}{e}\right)^m$

Thm: If n balls are thrown into n bins u.a.r., then $\exists$ a bin containing $\geq \Omega\left(\frac{\log n}{\log\log n}\right)$ balls w.p. $\geq 1 - \frac{1}{n}$

Pf: $f(x_1, \dots, x_n) = \begin{cases} 1, & \text{if } x_i \leq k \ \forall\, i \in \{1, \dots, n\} \equiv \max\{x_1, \dots, x_n\} \leq k \\ 0, & \text{o.w.} \end{cases}$

$$\mathbb{E}(f(X_1^m, \dots, X_n^m)) = \Pr(\text{Every bin has} \leq k \text{ balls})$$

$$\Pr\left(\bigcap_{i=1}^n (Y_i^m \leq k)\right) = \left(\Pr(Y_i^n \leq k)\right)^n = \left(1 - \Pr(Y_i^n \geq k)\right)^n$$

$$\Pr(Y_i^n \geq k) \geq \Pr(Y_i^n = k) = \frac{e^{-1} \cdot 1^k}{k!} = \frac{1}{ek!}$$

($Y_i^n \hookrightarrow \text{Poi}(1)$)

$$\Pr\left(\bigcap_{i=1}^n (Y_i^n < k)\right) = \left(1 - \frac{1}{ek!}\right)^n \leq e^{\frac{-n}{ek!}}$$

$$\Pr(\text{Every bin has} < k \text{ balls}) \leq e\sqrt{n} \cdot e^{-n/ek!}$$

$$< \frac{1}{n} \text{ (req'd.)}$$

Verify that $k = \Omega\left(\frac{\log n}{\log\log n}\right)$ works

Static dictionary

Universe $\mathcal{U}$, set $S \subseteq \mathcal{U}$

S is static, i.e., known at the start & fixed

$m = |S| \ll |\mathcal{U}| = M$ → $O(|S|)$

we wish to compactly represent S so that membership queries ($x \overset{?}{\in} S$) can be answered efficiently → $O(1)$ [expected or worst-case]

$\log(1 - e^{-y}) - y + \frac{y}{e^y - 1}$ $\log(e^y - 1) - \frac{ye^y}{e^y - 1}$ $\log e^y - \log(e^y - 1) = ye^y$ $e^y = 1/2$ $y = -\log 2$

Bloom filter

Bit-array $A[1,2,\ldots,n]$ (i.e., size is n)

Choose k fns. $h_1,\ldots,h_k$ u.a.r. [assumes perfectly rand. hash fns.]

$\forall x \in S$, set $A[h_i(x)] = 1 \ \forall \ i \in \{1,\ldots,k\}$

i.e., $h_j(x) = i$ w.p. n^{-1}

Membership query resolution: Given x, check $A[h_i(x)] \ \forall \ i \in \{1,\ldots,k\}$

If all 1's, say $x \in S$

$x \in S \Rightarrow$ correct result guaranteed

$x \notin S \Rightarrow$ false positives may occur

Given $x \notin S$,

$\Pr[\text{we answer } x \in S] \leq \delta$

$\Pr[A(j) = 0] = \underbrace{\left(1 - \frac{1}{n}\right)^{mk}}_{=p} \approx e^{-mk/n}$ ← overall h_i's, k & x, m

For given $x \in S$ & h_i, $\Pr(h_i(x) = j) = \frac{1}{n}$

$A[j]$ & $A[j']$ are dependent (balls-in-bins w/ km balls & n bins)

$\Pr(Y_i \sim \text{Poi}(mk/n) = 0)$

25/8/23

What should be n, k so that proba. for false positives is $\leq \delta$?

Expected no. of empty posns. in the Bloom filter $= np$

Given $x \notin S$, assuming p proba. of bin being empty (frac. of empty bins), ↳ assumes independence

$\Pr(\text{Answer Yes}) = (1-p)^k$

$Y_i \sim \text{Poi}(km/n)$, $\Pr(Y_i = 0) = e^{-km/n}$

$Z_i = \begin{cases} 1, & \text{if } Y_i = 0 \\ 0, & \text{o.w.} \end{cases}$ $\quad Z_i \sim \text{Ber}(e^{-km/n})$

$Z = \sum_{i=1}^{n} Z_i \approx$ No. of ~~empty~~ posns. in A that are 0

Just to say they are close i.e., a good bound

$\Pr[|Z - E(Z)| \geq \epsilon n] \leq 2e^{-n\epsilon^2 e^{mk/n}/3}$, $[\delta = \epsilon n/\mu]$

With $f(x)$ as $|x - E(z)| \geq \epsilon n \Rightarrow 1$; 0 o.w.,

$\Pr[|X - \underbrace{E(z)}_{ne^{-km/n}}| \geq \epsilon n] \leq 2e\sqrt{km}\, e^{-n\epsilon^2 e^{mk/n}/3}$

$\Pr(\text{false positives}) \approx \left(1 - e^{-km/n}\right)^k \xrightarrow{\text{min. for } k = (\ln 2)\frac{n}{m}} = \left(\frac{1}{2}\right)^k \leq \delta$

$k = \log(1/\delta) \Rightarrow n = m\log(1/\delta)/\ln 2$

$\frac{k\log(1 - e^{-km/n})}{\log(1 - e^{-km/n}) + k\left(e^{-km/n}\right)\cdot\frac{m}{n}}$

Universal hash families (Carter-Wegman)

k-universal hash family:

Family of fns $\mathcal{H} = \{h : U \longrightarrow [n]\}$ s.t.

$\forall\ x_1, x_2, \ldots, x_k$ (all different),

$$\Pr_{h \in_r \mathcal{H}}\left[h(x_1) = h(x_2) = \cdots = h(x_k)\right] \leq \frac{1}{n^{k-1}}$$

If $\mathcal{H}$ = set of all fns. from U to $[n]$, $(|\mathcal{H}| = n^{|U|})$

$$\Pr_{h \in_r \mathcal{H}}\left[h(x_1) = \cdots = h(x_k)\right] = \frac{1}{n^{k-1}} \quad \forall k$$

(Ideally, we want $|\mathcal{H}| = \text{poly}(|U|)$)

k-wise indep. hash family: [strongly k-universal]

$\mathcal{H} = \{h : U \longrightarrow [n]\}$ s.t.

$\forall\ x_1, x_2, \ldots, x_k$ (distinct), $\forall\ y_1, y_2, \ldots, y_k \in \{1, \ldots, n\}$

$$\Pr_{h \in_r \mathcal{H}}\left[\bigcap_{i=1}^{k} (h(x_i) = y_i)\right] = \frac{1}{n^k}$$

$\Downarrow$

$\forall\ x_1, x_2, \ldots, x_k$ (distinct),

$$\Pr_{h \in_r \mathcal{H}}\left[h(x_1) = h(x_2) = \cdots = h(x_k)\right] = \frac{1}{n^{k-1}}$$

strongly k-universal $\Rightarrow$ k-universal

28/8/23

strongly k-universal $\equiv$ uniformity, i.e.,

① $\Pr_{h \in_r \mathcal{H}}[h(x) = y] = 1/n$ &

② $\Pr_{h \in_r \mathcal{H}}\left[\bigcap_{i=1}^{k} h_{x_i}(x_i) = y_i\right] = \prod_{i=1}^{k} \Pr(h_{x_i}(x_i) = y_i)$

* m balls into n bins using a 2-universal hash family

$\mathcal{H} = \{h : [m] \longrightarrow [n]\}$

i.e, choose $h \in_r \mathcal{H}$ & assign balls to bins using h

For any 2 balls i & j, let

$$X_{ij} = \begin{cases} 1, & \text{if } i \text{ \& } j \text{ land in the same bin} \\ 0, & \text{o.w.} \end{cases}$$

$$\Rightarrow \Pr(X_{ij} = 1) = \Pr_{h \in_r \mathcal{H}}[h(i) = h(j)] \leq 1/n$$

Total no. of collisions, $X = \sum_{i<j} X_{ij}$

~~collision~~ collision ~~collission~~ colision

$$E(X) \leq \binom{m}{2} \cdot \frac{1}{n} \leq \frac{m^2}{2n}$$

Let Y be the max. load

$X \geq \binom{Y}{2}$ = no. of collisions in the bin w/ max. load

$\Pr\left(X \geq \frac{m^2}{n}\right) \leq 1/2$ (Markov's ineq.)

$\Pr\left(\binom{Y}{2} \geq \frac{m^2}{n}\right) \leq 1/2$ (since $\binom{Y}{2} \leq X$)

$\Pr\left(Y \geq 1 + m\sqrt{\frac{2}{n}}\right) \leq 1/2$ $\left[\binom{Y}{2} \geq \frac{(Y-1)^2}{2}\right]$

If $n = \Theta(m^2)$, then every bin has $\Theta(1)$ balls w.p. $\geq 1/2$

for a static dict. on $S \subseteq U$, $[|S| = m]$

we can obtain a good hash fn. of size $\Theta(m^2)$

- choose a hash fn from a 2-universal hash family
- if no. of collisions is too high, resample

[Expected number of times to sample is 2 from our bound]

* Dynamic dictionary over U

- additions
- deletions
- query

(using H, a 2-universal hash family)

[we want $n = O(m)$ ideally ⇒ FKS (Fredman-Komlós-Szemerédi) hashing]

Set of all additions, $S \subseteq U$

$x \in U$, $h \in H$

No. of collisions of x w/ elements in S using hash fn. h,

$$\text{coll}(x, S, h) = |\{y \in S \mid h(x) = h(y)\}|$$

$$\mathop{E}_{h \in H}[\text{coll}(x, S, h)] = \frac{1}{|H|} \sum_{h \in H} \text{coll}(x, S, h)$$

$$= \frac{1}{|H|} \sum_{h \in H} \sum_{y \in S} 1_{[h(x)=h(y)]}$$ ← indicator fn

$$= \sum_{y \in S} \underbrace{\frac{1}{|H|} \sum_{h \in H} 1_{[h(x)=h(y)]}}_{\Pr_{h \in H}[h(x)=h(y)]}$$

$$\leq \sum_{y \in S} \frac{1}{n} = \frac{|S|}{n}$$ [$n \to$ size of hash table]

[If $n = \Theta(|S|)$, then $O(1)$ running time for query, but hash table could be sparse at times]

Explicit 2-universal family

U, $|U| = m \longrightarrow n$ sized table

Prime $p > m$, $p = \Theta(m)$ [$\exists$ prime p b/w m & $2m$]

$$\mathcal{H} = \{ h_{a,b} \mid 1 \le a \le p-1,\ 0 \le b \le p-1 \}$$

$$h_{a,b}(x) = ((ax+b) \bmod p) \bmod n$$

$$|\mathcal{H}| = \Theta(p^2)$$

$\Pr_{h \in_r \mathcal{H}} [h(x) = h(y)] \le 1/n$

$$ax + b = z_1 \pmod p$$
$$ay + b = z_2 \pmod p$$
$$z_1 = z_2 \pmod n$$

30/8/23

Thm: $\mathcal{H}$ is 2-universal

Pf: To show: $\Pr_{h \in \mathcal{H}} [h(x) = h(y)] \le 1/n$

$$ax + b = z_1 \pmod p$$
$$ay + b = z_2 \pmod p$$
$$z_1 = z_2 \pmod n$$

For every $z_1 \neq z_2$, $\exists$ unique soln. for a, b

$$|\{ z_2 \mid z_2 \neq z_1,\ z_1 = z_2 \pmod n \}| \le \frac{p}{n} - 1$$

$$\Rightarrow \Pr_{h \in \mathcal{H}} [h(x) = h(y)] \le \frac{p(\frac{p}{n} - 1)}{p(p-1)} = \frac{1}{n} \cdot \frac{(p-n)}{(p-1)}$$

$$\le 1/n$$

[same idea as pairwise indep. strings]

Pairwise-indep hash family

$|U| = 2^m \longrightarrow 2^n$

Consider the field $GF(2^m)$

$$\mathcal{H} = \{ h_{a,b} \mid a, b \in GF(2^m) \}$$

$h_{a,b}(x) = (ax + b)$ in $GF(2^m)$ → The "pairwise" comes from here. Fix y ($k-1$) $\Rightarrow$ k-wise indep.

m-bit string $\Longrightarrow$ truncate to n bits

Exercise: verify this is pairwise indep.

Perfect hash family [static dicts again]

A family $\mathcal{H} = \{h : \mathcal{U} \to [n]\}$ is perfect for sets of size $\leq n$ if $\forall S \subseteq \mathcal{U}, |S| \leq n$,

$\exists h \in \mathcal{H}$ s.t. $\forall x \neq y, h(x) \neq h(y)$

[Any n-sized set in $\mathcal{U}$ can be perfectly hashed by a fn. in $\mathcal{H}$]

$\Rightarrow$ static dictionary has an $O(n)$ data structure with $O(1)$ query time

↳ Is $A[h(x)] = x$?

Each cell of the table = $O(\log|\mathcal{U}|)$

we also want to store each h in a constant no. of cells & to index $\mathcal{H}$, we need $\log|\mathcal{H}|$ bits

$\Rightarrow \log|\mathcal{H}| = O(1) \cdot \log|\mathcal{U}| \Rightarrow |\mathcal{H}| = |\mathcal{U}|^{O(1)}$

* If $\mathcal{H} = \{h : \mathcal{U} \to [n]\}$ is a perfect hash family for sets of size n, then

$$|\mathcal{H}| = 2^{\Omega(n)}$$

$|\mathcal{U}| = n^2 \to [n]$ cannot be perfectly has

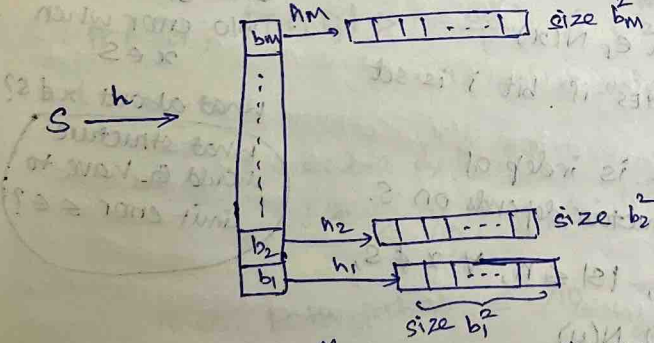
FKS hashing (Fredman - Komlos - Szemeredi, '84) [Cell probe model]

$\mathcal{U} \to [n]$, sets S, $|S| = m$

Choose $h \in \mathcal{H}$ (2-universal)

s.t. no. of collisions $\leq m$

Handle collisions using another level of hashing (2-universal again)



(Further collisions are chained, since only constant collisions now)

$$\text{Size} = m + \sum_{i=1}^{m} b_i^2$$

$$= O(m)$$

$$\sum_{i=1}^{m} \binom{b_i}{2} \leq m$$

(No. of collisions)

Bit probe model

Size, query are now in terms of bits rather than cells

FKS: $m \log|\mathcal{U}|$ $\log|\mathcal{U}|$

What should be n if we limit ourself to 1 bit query?

Buhrman, Miltersen, Radhakrishnan, Venkatesh, '00

$\Rightarrow$ size $= O\left(\frac{M}{\epsilon} \log |\mathcal{U}|\right)$ $\longrightarrow$ comparable to FKS result

1-bit query w/ 2-sided error of $\leq \epsilon$

If we restrict 1-sided error of $\leq \epsilon$, we show this

$$\text{size} = O\left(\frac{M^2}{\epsilon^2} \log |\mathcal{U}|\right)$$

↳ can show existence but can't obtain the data structure

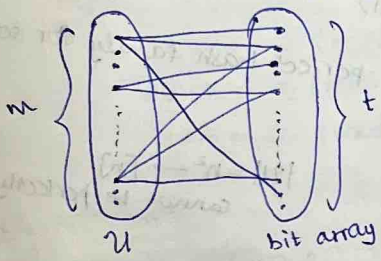
If we want to efficiently find the structure,

$$\text{size} = O\left(\frac{M^2}{\epsilon^2} \log^2 |\mathcal{U}|\right)$$

31/8/23

Change of notation for the proof: $|\mathcal{U}| = M$, $|S| = n$

1-query algo. $\longrightarrow$ bipartite graph



$x \overset{?}{\in} S$: Randomly choose $i \in_R [t]$ using the graph say, [edge wts. like proba. of choosing]

Given $G(\mathcal{U}, [t], E)$,

$S \subseteq \mathcal{U}$, $|S| \leq n$,

define $\forall x$, $N(x) = \{i \mid (x,i) \in E\}$

set all the bits in $\bigcup_{x \in S} N(x)$

query: $x \overset{?}{\in} S$

- choose $i \in_R N(x)$
- Answer "YES" if bit i is set

clearly, this works w/o error when $x \in S$

What about $x \notin S$?

What structure should G have to limit error $\leq \epsilon$?

Note that the graphs is indep. of S. Only the data structure depends on S.

We want: $\forall S \subseteq \mathcal{U}$, $|S| \leq n$, $\forall x \notin S$,

for $N(S) = \bigcup_{y \in S} N(y)$,

$$|N(x) \cap N(S)| \leq \epsilon |N(x)|$$

We show a stronger result:

Such graphs exist

$$\begin{cases} \forall x, |N(x)| = d & [d\text{-regular over } \mathcal{U}] \\ \forall x \neq y, |N(x) \cap N(y)| \leq \frac{\epsilon d}{n} & [\text{this implies the above}] \end{cases}$$

An (m, t, d, r)-combinatorial design is a family of sets

$T_1, T_2, \ldots, T_m$ s.t. $\forall i \; \forall j, i \neq j$

① $T_i \subseteq [t]$

② $|T_i| = d$

③ $|T_i \cap T_j| \leq r$

In our case,

$T_i = N(x_i)$, $x_1, x_2, \ldots, x_m \in \{0,1\}^n$

$r = \frac{\epsilon d}{n}$

We consider a random collection of sets & show that w.p. >0, the properties are satisfied, implying existence.

Pf: $\forall j \in [m]$, $\forall i \in [t]$,
put i in T_j w.p. $\frac{2d}{t}$ independently

$\Rightarrow E(|T_j|) = 2d$

$Pr(|T_j| < d) \leq e^{-2d(\frac{1}{2})^2/2} = e^{-d/4}$ [chernoff bound]

$\Rightarrow Pr(\exists j \; |T_j| < d) \leq m e^{-d/4}$ [Union bound]

< 1 [we want this, otherwise no set can have size d]

$E(|T_i \cap T_j|) = t\left(\frac{2d}{t}\right)^2$ $\left[\sum_{k \in [t]} E(|T_i \cap T_j| \mid k \in T_i) \cdot Pr(k \in T_i)\right]$ (check)

$= \frac{4d^2}{t}$

$Pr\left(|T_i \cap T_j| > \frac{8d^2}{t}\right) \leq e^{-\frac{4d^2}{3t}}$ [chernoff bound].

$Pr\left(\exists i,j \; |T_i \cap T_j| > r = \frac{8d^2}{t}\right) \leq \binom{m}{2} e^{-4d^2/3t}$

$\leq m^2 e^{-4d^2/3t}$

We find suitable d, t s.t.

$m e^{-d/4} \leq 1/10$ & $m^2 e^{-4d^2/3t} \leq 1/10$

Then, both probas $\leq 1/10$ $\Rightarrow Pr(\text{not satisfied}) \leq 1/5$

$\Rightarrow Pr(\text{satisfied}) \geq 4/5 > 0$

$r = \frac{\epsilon d}{n} = \frac{8d^2}{t} \Rightarrow t = \frac{8dn}{\epsilon}$

This is satisfied when

$d = \Theta\left(\frac{n}{\epsilon} \log m\right)$ & $t = \Theta\left(\frac{n^2}{\epsilon^2} \log m\right)$

Hence, proved

$r = \Theta(\log m)$

1/9/23

PS1 discussion

(1) Generate n distinct nos. & sort them

Assign each i the sorted posn. of k_i

$k_1, k_2, \ldots, k_n$

choose using $3\log n$ bits each

$\Rightarrow \Pr(k_i = k_j) \leq \frac{1}{n^3}$

$\therefore \Rightarrow \Theta(n \log n)$

$\Rightarrow \Pr(\exists i,j \text{ s.t. } k_i = k_j) \leq \frac{1}{n}$

(4) Intuitively, X increments every 2^x times, so in a sense, it appears to be counting the no. of bits in the representation of n

Exercise: Find variance, $\text{Var}(y)$ & show

n can be rep. ~~using~~ as $(1 \pm \epsilon)n$

using $\frac{1}{\epsilon^2} \log\log n$ bits

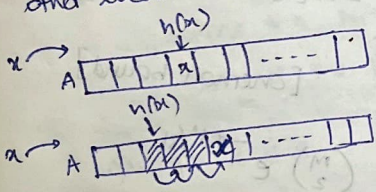
(Morris counter)

↳ way to ~~store~~ count n times using $\log\log n$ bits

4/9/23

Open addressing (Linear probing)

Our previous hashing schemes (such as FKS) handled collisions using an auxiliary data structure. This scheme maps collisions into some other location of the primary data structure.



If $A[h(x)]$ is empty

If $A[h(x)]$ contains some other element, probe linearly for an empty position

~~Addition~~: Insertion: Starting from $h(x)$, insert x at the first empty slot

Membership: Starting from $h(x)$, continue until you find x or an empty slot

[doesn't work when there are deletions, in which case we set flags/tombstones at posns. of deleted elements]

Works very well in practice.

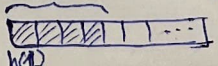
Knuth '63: If h is completely random, then expected time is $O(1)$.

2010-11: 5-wise independence is necessary & sufficient for expected $O(1)$ time.

"Run" A run is a maximal contiguous seq. of cells that are filled

Given $i \in S$, what is the length of the run containing $h(i)$?

Say, 1



Consider a k-wise indep. hash family $\mathcal{H}$, $h \in_r \mathcal{H}$

~~For the given i,~~ consider the interval of posⁿs in the hash table.

$$I_j = \left[h(1) - (2^j - 1),\ h(1) + (2^j - 1) \right] \quad \text{(dyadic intervals)}$$

$|S| = n$, hash table has size t

Expected no. of elements in $I_j = \frac{|I_j| \cdot n}{t}$

$$X_i = \begin{cases} 1, & \text{if } i \in I_j \\ 0, & \text{o.w.} \end{cases}$$

$$X = \sum_{i \in S \setminus \{1\}} X_i \quad \Rightarrow E(X) = \frac{n|I_j|}{t}$$

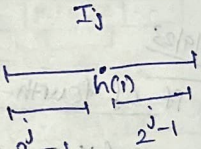
If $t = 8n$, $|I_j| = 2^{j+1} - 1 \Rightarrow E(X) \le 2^{j-2}$

Consider R, a run containing 1

$$E(|R|) = \sum_{l=0}^{n} l \cdot \Pr(|R| = l)$$

$$\le \sum_{j=0}^{\log t} 2^j \cdot \Pr\left(\underbrace{2^{j-1} \le |R| \le 2^j}_{\text{For these, } l \le 2^j} \right)$$

If $|R| \ge 2^{j-1}$, no. of elements in I_j is $\ge 2^{j-1}$



① Assume h is random $\Rightarrow X$ is a sum of Bernoulli r.v.s

$$\Pr(|R| \ge 2^{j-1}) \le \Pr\left(X \ge \underbrace{2^{j-1}}_{2E(X)}\right)$$

$$\le e^{-2^{j-2}/2}$$

$$\Rightarrow E(|R|) \le \sum_{j=0}^{\log t} \frac{2^j}{e^{2^j}} = O(1)$$

② Suppose h is picked from a 3-wise indep. hash family

Given $h(1)$, X_i's are pairwise indep.

$$\Rightarrow \text{var}(X) = \sum_{i \in S \setminus \{1\}} \text{Var}(X_i) = \sum_{i \in S \setminus \{1\}} \underbrace{E(X_i^2)}_{= E(X_i)} - [E(X_i)]^2 \quad \text{(Bernoulli)}$$

$$\le \sum_{i \in S \setminus \{1\}} E(X_i)$$

$$\Pr(|R| \ge 2^{j-1}) \le \Pr\left(X \ge \underbrace{2^{j-1}}_{2E(X) \le 2\text{var}(X)} \right) \le \frac{1}{\text{Var}(X)}$$

$$\le \frac{1}{2^{j-2}}$$

$$\Rightarrow E(|R|) \le \sum_{j=0}^{\log t} \frac{2^j}{2^{j-2}} = O(\log n)$$

Fourth-moment bound

Let $X_1, X_2, \ldots, X_n$ be 4-wise indep. s.t. ~~$E(X_i) = p$~~ $X_i \sim \text{Ber}(p)$

$$X = \sum_{i=1}^{n} X_i \Rightarrow E(X) = np = \mu$$

For $\mu \geq 1$, $\beta > \mu$

$$\Pr(X > \mu + \beta) \leq \frac{4\mu^2}{\beta^4}$$

---

(3) h is picked from a 5-wise indep. family

$\Rightarrow$ Given $h(1)$, X_i's are 4-wise indep

$$\Pr(|R| \leq 2^{j-1}) \leq \Pr(X \geq 2^{j-1})$$

$$= \Pr(X \geq 2^{j-2} + 2^{j-2})$$

$$\leq \frac{4}{\mu^2} = \frac{32}{2^{2j}}$$

$$E(|R|) \leq \sum_{j=0}^{\log t} \frac{2^j \cdot 32}{2^{2j}} = 32 \sum_{j=0}^{\log t} \frac{1}{2^j} \leq = \Theta(1)$$

6/9/23

Pf. of fourth-moment bound:

$$\Pr(X \geq \mu + \beta) = \Pr(X - \mu \geq \beta) = \Pr((X-\mu)^4 \geq \beta^4)$$

$$\leq \frac{E[(X-\mu)^4]}{\beta^4}$$

Define $Y_i = X_i - p$ Since X_i's are 4-wise indep., so are Y_i's

$$E(Y_i) = 0$$

$$E[(X-\mu)^4] = E\left[\left(\sum_{i=1}^{n}(X_i - p)\right)^4\right] = E\left[\left(\sum_{i=1}^{n} Y_i\right)^4\right]$$

$$= E\left[\sum_{i=1}^{n} Y_i^4 + \binom{4}{2}\sum_{i<j} Y_i^2 Y_j^2 + \binom{4}{3}\sum_{i\neq j} Y_i Y_j^3 + \sum_{i\neq j\neq k\neq l} Y_i Y_j Y_k Y_l + \binom{4}{2}\binom{2}{1}\sum_{i\neq j\neq k} Y_i Y_j Y_k^2\right]$$

$$= \sum_{i=1}^{n} E(Y_i^4) + \binom{4}{2}\sum_{i<j} E(Y_i^2)E(Y_j^2)$$ [others split & go to 0 due to 4-wise indep.]

$$E(Y_i^4) = p(1-p)^4 + (1-p)(0-p)^4$$

$$E(Y_i^2) = p(1-p)^2 + (1-p)p^2$$

$$E[(X-\mu)^4] \leq 4np^2 = 4\mu^2 \quad \text{(verify)}$$ Hence, proved.

Cuckoo hashing (Pagh, Rödler – '01)

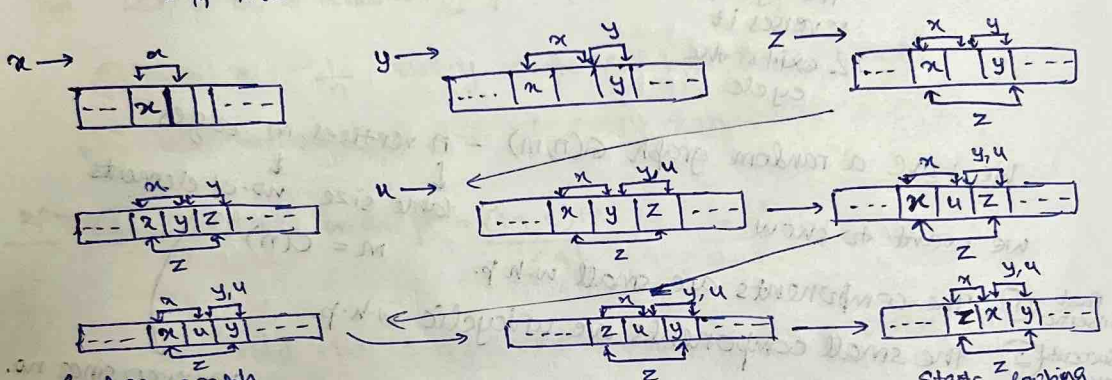
T

2 hash fns. h_1, h_2 (completely random)

* Every x is hashed to either $h_1(x)$ or $h_2(x)$
  ⇒ search for x in $O(1)$ – worst case

Insertions are slightly harder:

Insert (x):
- If one of $h_1(x), h_2(x)$ is empty, then insert in T at the empty loc.
- If both $h_1(x)$ & $h_2(x)$ are occupied,
  then $h_i(y) = h_1$ for some $y \neq x$, $i \in \{1,2\}$,
  place x in $h_1(x)$ & recursively try to insert y
- If this continues indefinitely, choose new h_1', h_2' & rehash the table



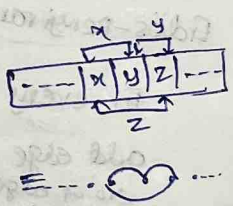
Cuckoo graph

$G(V, E)$

$V \rightarrow$ set of posns. in the hash table

$\forall x$ in the table, add edge $(h_1(x), h_2(x))$

- self loops & multi edges are possible

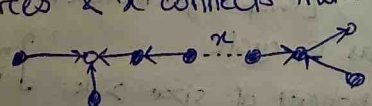


Obs: Let C be a connected component in the cuckoo graph, and x is being inserted. If the new component has >1 cycle, then insertion is impossible. [without rehashing]

Lemma: If after insertion, the component is unicyclic (at most 1 cycle), then insertion is possible in time $O(|C|)$
↳ new size of the component

Possible cases:

① 2 trees & x connects them



Directions show potential next posns. for hashed elements

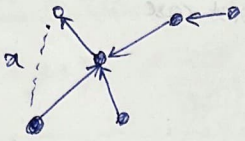
Each vertex has out-degree ≤ 1

Results in a DAG, which must have a sink ⇒ terminates

② x lies within a component C

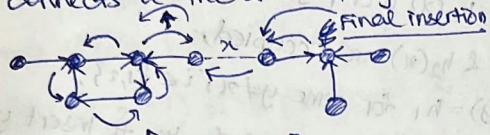
$\Rightarrow$ C must be a tree

~~Once again~~, once again, the tree becomes a DAG, & the sink node ~~gets~~ filled



7/9/23

③ x connects a tree & a ~~unicycle~~ cycle



Eventually it reaches the cycle, reverses it & exits the cycle

Insertion here we have seen

We have a random graph $G(n, m)$ – n vertices, m edges

↓ table size ↓ no. of elements

$m = O(n)$

we want to show,

Fast insertion ① The components are small w.h.p.

Successful insertion ② The small components are unicyclic w.h.p.

However, since no. of edges is fixed, existence of edges is dependent, making analysis difficult

Instead, we analyse

Erdös–Renyi random graph: $G(n, p)$

For every pair of vertices (v_1, v_2), add edge (v_1, v_2) w.p. p indep.

$\Rightarrow \mathbb{E}(\text{No. of edges}) = \binom{n}{2} p = m$, when $p = \frac{m}{\binom{n}{2}}$

Good h.p. bound too, since it is a binomial r.v.

Monotone graph property

A property $\mathcal{P}$ is monotone increasing if

$G \in \mathcal{P} \wedge G \subseteq G' \Rightarrow G' \in \mathcal{P}$

G has the property; G' is a supergraph of G; G' has the property

[Swap to get monotone decreasing]

(E.g.): $\mathcal{P}_1$ – graph has a cycle

$\mathcal{P}_2$ – If a component of size $> k$

} monotone increasing

$\mathcal{P}$ - max. indep. set is of size $\geq k$
- bipartiteness
- planarity

} monotone decreasing

Lemma: Let $\mathcal{P}$ be any monotone increasing property.

Denote by $P(n,M) = \Pr_{G \sim G(n,M)}(G \in \mathcal{P})$

& $P(n,p) = \Pr_{G \sim G(n,p)}(G \in \mathcal{P})$

If $p^+ = \frac{(1+\epsilon)M}{\binom{n}{2}}$, $p^- = \frac{(1-\epsilon)M}{\binom{n}{2}}$, then

$$P(n,p^-) - e^{-\Theta(n)} \leq P(n,M) \leq P(n,p^+) + e^{-\Theta(n)}$$

Thm: Let G be a cuckoo graph w/ $m = (1-\epsilon)\frac{n}{2}$, then

① w.p. $\geq 1 - \frac{1}{n}$, every component has size $\Theta(\log n)$

② ~~w.h.p.~~ the expected size of every component is $O(1)$

Pf: ① $G(n,p^+)$ with $p^+ = \frac{(1+\epsilon)m}{\binom{n}{2}} = \frac{1-\epsilon^2}{n-1}$

To bound the component size, we fix a vertex $v \in V$ & start a BFS

For any vertex v_i, $N_i = |N(v_i)|$ (num. neighbours) $\sim Bin(n-1, p^+)$

$v_1, v_2, \dots, v_{i-1}, v_i \xrightarrow{N_i} Bin(n-i, p^+)$

$\Pr\left(\sum_{i=1}^{k} N_i > k\right)$ ⟶ once again dep.

$B_i \sim Bin(n-1, p^+) \Rightarrow \Pr\left(\sum_{i=1}^{k} B_i > k\right)$ (Branching process)

more flexible, so can bound this to bound our target

$B = \sum_{i=1}^{k} B_i \sim Bin(k(n-i), p^+)$

$\mathbb{E}(B) = k(n-i)p^+ = k(1-\epsilon^2)$

$$\Pr(B > k) = \Pr\left(B \geq \frac{\mathbb{E}(B)}{1-\epsilon^2}\right) \leq \Pr(B \leq (1+\epsilon^2)\mathbb{E}(B))$$

8/9/23

① If $m = (1-\epsilon)\frac{n}{2}$, then the expected size of any connected component in the cuckoo graph is $O(1)$.

② w.h.p. all components of size $\Theta(\log n)$ are unicyclic

(1) Fix $v \in V$. Let S be the size of the component containing v in $G(n, M)$.

$$\mathbb{E}(S) = \sum_{k \geq 0} k \cdot \Pr(S = k)$$

$$= \Pr(S=1) + \Pr(S=2) + \Pr(S=2) + \Pr(S=3) + \Pr(S=3) + \Pr(S=3) + \cdots$$

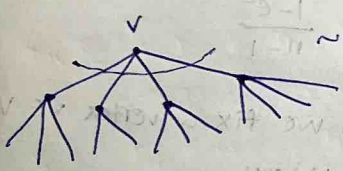
$$= \sum_{k \geq 0} \Pr(S \geq k) \rightarrow \text{monotone increasing}$$

$$\leq \sum_{k \geq 0} \Pr(S' \geq k) + ne^{-\Theta(n)} \rightarrow O(1)$$

S': component size in $G(n, p^+)$

outside summation, $k \leq n$

Branching process



$\sim Bin(n-1, p^+)$

$$p^+ = \frac{2(1+\epsilon)n(1-\epsilon)}{2n(n-1)} = \frac{1-\epsilon^2}{n-1}$$

Y_i = No. of nodes at level i

$$Y = \sum_{i \geq 0} Y_i$$

$$\mathbb{E}(Y_i) = \sum_{k \geq 0} \mathbb{E}(Y_i \mid Y_{i-1} = k) \cdot \Pr(Y_{i-1} = k)$$

$$= \sum_{k \geq 0} k \cdot (n-1)p^+ \cdot \Pr(Y_{i-1} = k) = (n-1)p^+ \mathbb{E}(Y_{i-1})$$

$$= ((n-1)p^+)^i = (1-\epsilon^2)^i$$

$$\mathbb{E}(Y) = \sum_{i \geq 0} (1-\epsilon^2)^i = \frac{1}{\epsilon^2} = O(1)$$

$$\Rightarrow \mathbb{E}(S) = O(1)$$

(2) In $G(n, m)$, how can a component not be unicyclic?

fix k vertices - $\binom{n}{k}$

Cayley's formula: There are k^{k-2} labelled trees on k vertices

Edges are placed into the tree w.p. $\binom{m-1}{k-1} \cdot (k-1)! \cdot \left(\frac{1}{n^2}\right)^{k-1}$ (choosing that edge in $G(n,m)$)

Add 2 edges $\binom{m-k+1}{2} \cdot 2! \underbrace{(k^2)^2 \left(\frac{1}{n^2}\right)^2}_{\text{within the tree}}$

We don't want any more edges to the tree (from other vertices)

$\Rightarrow \left(1 - \frac{k(n-k)}{n^2}\right)^{m-k-1}$ → can add more cycles inside k-vertex-set, which double counts ⇒ lower bound

⇒ Proba. that a component of size k isn't unicyclic:

$\leq \binom{n}{k} k^{k-2} \cdot \binom{m}{k-1} \cdot (k-1)! \cdot \binom{m-k+1}{2} \cdot 2! \cdot k^4 \left(\frac{1}{n^2}\right)^{k+1} \left(1 - \frac{k(n-k)}{n^2}\right)^{m-k}$

$\leq \frac{1}{n}$ when $k = \Theta(\log n)$, after heavy calculation

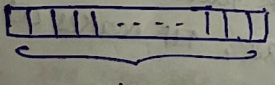
---

Midsem portions complete

Online algorithms

- Data arrives as a stream & not all at once
- At each step, we make irrevocable decisions
- We want to know how good the decisions (made with incomplete information) are

Paging problem

Cache 

k

Seq. of page reqs. $r_1, r_2, \ldots, r_m$

If r_i does not exist in the cache, ⇒ cache miss / page fault
evict some item & bring r_i in

Goal: Minimize page faults

How do we know if some strategy is good?
Optimal offline strategy
Farthest-in-the-future
⇒ compare against this

Deterministic online strategies:

1. LRU - Least recently used
2. FIFO - First-in first-out
3. LFU - Least frequently used

An adversary can design the requests to forcefully trigger faults at every step
⇒ what does it mean to minimize page faults?

Competitive ratio

An algo. A is α-competitive if $A(\sigma) \leq \alpha \cdot OPT(\sigma) + c$ (indep. of m)

Edges are placed into the tree w.p. $\binom{m-k}{k} \cdot \binom{m}{k-1} \cdot (k-1)! \cdot \left(\frac{1}{n^2}\right)^{k-1}$ choosing the edge in $G(n,m)$

Add 2 edges $\binom{m-k+1}{2} \cdot 2! \underbrace{(k^2)^2 \left(\frac{1}{n^2}\right)^2}_{\text{within the tree}}$

We don't want any more edges to the tree (from other vertices)

$\Rightarrow \left(1 - \frac{k(n-k)}{n^2}\right)^{m-k-1}$ → can add more cycles inside k-vertex-set, which double counts ⇒ lower bound

⇒ Proba. that a component of size k isn't unicyclic:

$\leq \binom{n}{k} k^{k-2} \cdot \binom{m}{k-1} \cdot (k-1)! \cdot \binom{m-k+1}{2} \cdot 2! \cdot k^4 \left(\frac{1}{n^2}\right)^{k+1} \left(1 - \frac{k(n-k)}{n^2}\right)^{m-k-1}$

$\leq \frac{1}{n}$ when $k = \Theta(\log n)$, after heavy calculation

---

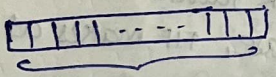
Midsem portions complete

11/9/23

Online algorithms

- Data arrives as a stream & not all at once
- At each step, we make irrevocable decisions
- We want to know how good the decisions (made with incomplete information) are

Paging problem

cache 

Seq. of page reqs. $r_1, r_2, \ldots, r_m$

If r_i does not exist in the cache, ⇒ cache miss / page fault

evict some item & bring r_i in

Goal: Minimize page faults

How do we know if some strategy is good?

optimal offline strategy: farthest-in-the-future

⇒ compare against this

Deterministic online strategies:

1. LRU - Least recently used
2. FIFO - First-in first-out
3. LFU - Least frequently used

An adversary can design the requests to forcefully trigger faults at every step

⇒ what does it mean to minimize page faults?

Competitive ratio

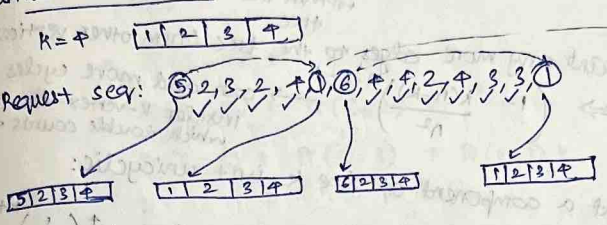
An algo. A is c-competitive if ∀ seqs. of reqs. $r_1, r_2, \ldots, r_m$, (indep. of m)

$f_A(r_1, r_2, \ldots, r_m) \leq c \cdot f_{OPT}(r_1, r_2, \ldots, r_m)$ → optimal algo., possibly offline

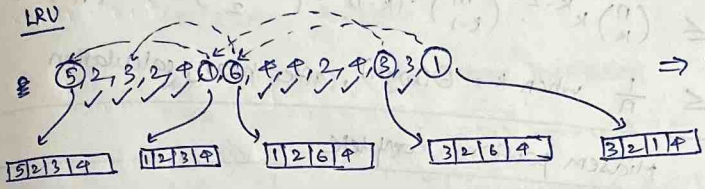
$f \approx$ loss fn.

$\Rightarrow \frac{f_A(r_1, r_2, \ldots, r_m)}{f_{OPT}(r_1, r_2, \ldots, r_m)}$ is bounded

Farthest-in-the-Future (FIF)



LRU



Thm: LRU is k-competitive

i.e., For any seq. $r_1, r_2, \ldots, r_m$, if LRU makes n faults, then FIF makes $\geq \frac{n}{k}$ faults

Idea:

Pf: Divide $r_1, r_2, \ldots, r_m$ into phases where in each phase, LRU makes k faults

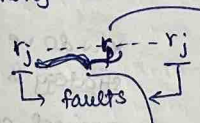
$r_1 \ldots r_{i_1}$ | $r_{i_1+1} \ldots r_{i_2}$ | $\ldots$ | $r_{i_{j-1}+1} \ldots r_m$

k faults | k faults | $\ldots$ | k faults

we want to show that in each phase, FIF makes at least one fault.

Pf: phase i

① Among the k faults, the same page r_j faulted twice



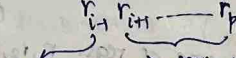
All other k-1 elements in the cache were requested

$\Rightarrow$ No. of distinct requests $\geq k-1$

In the interval $(r_j \ldots r_p)$, there were $\geq k+1$ distinct requests

$\Rightarrow$ FIF must also have faulted in that interval

② All the k faults are distinct



At the start of phase i, r_{i-1} was in the cache

(a) r_{i-1} was one of the faults

$\Rightarrow$ FIF faults, w/ a similar analysis as in (1)

(b) r_{i-1} wasn't one of the faults

From the k-1 "free" posns., there were k distinct reqs.

$\Rightarrow$ FIF faults

$\therefore$ In each of the $\frac{n}{k}$ phases, FIF faults at least once

$\Rightarrow$ FIF makes $\geq n/k$ faults

$\therefore$ LRU is k-optimal

THM: If a deterministic paging algo. is c-competitive, then $c \geq k$.

Idea: Fix an algo. A.

Construct a seq. $r_1, r_2, \ldots, r_m$ s.t.

if $f_A(r_1, r_2, \ldots, r_m) = n$,

then $f_{OPT}(r_1, r_2, \ldots, r_m) \leq \frac{n}{k}$

1	2	3	- - -	k-1	k

$1, 2, 3, \ldots, k-1, k, \underbrace{(k+1, e_1, e_2, \ldots, e_{m-k})}$, $e_i \in \{1, 2, \ldots, k+1\}$

we can construct by running A

↳ element evicted w/ prev. req.

$\Rightarrow f_A(r_1, r_2, \ldots, r_m) = m$

Divide the req. seq. into phases of size k

A faults k times.

If OPT encounters a fault, then the $(k+1)^{th}$ element (not in the cache) has been requested instead of an element (that is not requested in the phase at all) in the cache.

Hence, post this fault, there will be no more faults in this phase

$\Rightarrow f_{OPT}(r_1, r_2, \ldots, r_m) \leq m/k$ $\Rightarrow$ Best possible det. algo is k-competitive, i.e., for example, LRU

13/9/23

Randomized paging

Analysis will differ based on the power of the adversary.

We will analyse our algorithms assuming the weakest possible adversary:

Oblivious adversary → knows only the source code of the algorithm
→ unaware of the internal coin tosses
→ non-adaptive: (input seq. decided at first, essentially)

More powerful adversaries can be adaptive:

- offline adversary → strongest form, knows the optimal offline algorithm & final comparison is essentially against this [not too clear, read up a bit more]
- online adversary

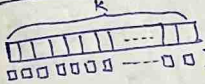
Competitive ratio

An algo. A has c.r. c if $\forall r_1, r_2, \ldots, r_M$,

$$E(f_A(r_1, r_2, \ldots, r_M)) \leq c \cdot f_{OPT}(r_1, r_2, \ldots, r_M)$$

↓ over the internal randomness of the algo. A

Marker algorithm



The algo. proceeds in rounds. At the start of each round, marker bits are 0

For each req r_i,
  If r_i is in the cache, set the marker bit to 1
  Else, choose a posn. with marker bit 0 u.a.r.,
    place r_i in that posn & set the marker bit to 1.
When all marker bits are 1, begin the next round.
(Each round has k distinct reqs.)

⇒ If a cache entry is req'd., it is not evicted in that round

Thm: Marker algo. is $2H_k$-competitive
(H_k = k^{th} Harmonic sum)

Pf: for the optimal algo:

Round i:
The cache has the following types of items:
- stale items → req'd. in round i-1 & round i (still in the cache)
- new items → req'd. in round i, not in round i-1
  denote by n_i

Let S_i be the items in the cache for OPT (at the start of round i)
S_i' be the items in the cache for Marker

$$d_i = |S_i \setminus S_i'|$$

(may contain n_i's)

OPT will fault on at least $n_i - d_i$ reqs.

$$d_{i+1} = |S_{i+1} \setminus S_{i+1}'|$$

OPT will have faulted on $\geq d_{i+1}$ reqs. [each item thrown away would've been a fault — think about this]

S_{i+1}' → queries in round i ⇒ evictions come up as missing now in S_{i+1}

⇒ OPT faults on $\geq \max\{n_i - d_i, d_{i+1}\}$

$\geq \frac{n_i - d_i + d_{i+1}}{2}$ ← non-

Total no. of faults for OPT over m rounds $\geq \frac{1}{2}\sum_{i=1}^{m} n_i$

Now, we want to show that expected no. of faults for Marker is $\leq H_k \sum_{i=1}^{m} n_i$

For Marker:

Round i:

New elements will cause page faults

Stale elements may cause page faults, if a new element evicts it before the stale element is queried

$\Rightarrow$ For the worst case, assume all the n_i new queries came first. ($=$, given the assumption)

At least one of the new queries evicts the first stale element w.p. $\leq \frac{n_i}{k}$ ($\frac{1}{k}$, union bound)

For the second stale element, there are 2 possibilities; prob. of eviction before query $\leq \frac{n_i}{k-1}$

- First stale query faulted → One new element in stale #1's pool, other $(n_i - 1)$ new queries & stale #1 can pick from $(k-1)$ locations → $\frac{n_i}{k-1}$
- First stale query didn't fault → stale #1 in own pool., n_i new queries have searched & picked from $(k-1)$ locations → $\frac{n_i}{k-1}$

For the whole round, exp. no. of faults $\leq n_i + \frac{n_i}{k} + \frac{n_i}{k-1} + \cdots + \frac{n_i}{k-(k-1)}$

$\leq n_i H_k$

$\Rightarrow$ Over all rounds, exp. no. of faults $\leq H_k \sum_{i=1}^{m} n_i$

$\Rightarrow$ c.r. $\leq 2H_k$

14/9/23

Yao's minimax principle $\longrightarrow$ std. technique to lower bound rand. algos.

The process is like a game b/w algo. designer & adversary

pay the cost = no. of pg. faults

$\Rightarrow$ a 2-player zero-sum game

Payoff matrix M

Consider rock-paper-scissors

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

Entries: How much col. player pays row player (payoff to row player)

Consider a slightly diff. version:

	R	P	S
R	0	1	2
P	-1	0	5
S	-2	-1	0

The safest option for the row player is to play R, as it guarantees 0 payoff, while P & S could lead to a payoff of -1.
For the col. player, safest option is again R, as they would have to pay at most 0.

Row player: $\max_i \min_j M_{ij}$

$\equiv$ Maximise the minimum payoff (of a row)

Col. player: $\min_j \max_i M_{ij}$

$\equiv$ Minimise the maximum amt. given away

If $\max_i \min_j M_{ij} = \min_j \max_i M_{ij}$, the game has a value

Here, both are 0 for (R,R) $\longrightarrow$ pure strategy, action (R) is fixed.

For simple RPS, $\max_i \min_j M_{ij} = -1$, $\min_j \max_i M_{ij} = 1$

$\Rightarrow$ The game doesn't have a value

$\forall M$, $\max_i \min_j M_{ij} \leq \min_j \max_i M_{ij}$

Mixed strategies

The action isn't fixed, rather they are chosen using a proba. dbn.

$p \sim$ dbn. over rows, $q \sim$ dbn. over cols.

Here, we consider the expected payoff $= \sum_{i=1}^{m} \sum_{j=1}^{n} p_i q_j M_{ij}$

$= p^T M q$

Row player: $\max_p \min_q p^T M q$

~~= best strategy for most p~~

Col player: $\min_q \max_p p^T M q$

Fixing the rand. bits for a rand. algo. makes it deterministic.

$\Rightarrow$ A rand. algo. $\sim$ dbn. over deterministic algos.

Von-Neumann minimax theorem

For every payoff matrix M (for 2-person zero-sum games),

$\max_p \min_q p^T M q = \min_q \max_p p^T M q$

Suppose we fix p.

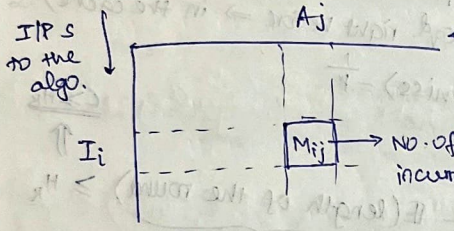
$$p^T M = \left[\sum_{i=1}^{m} p_i M_{i,1} \quad \sum_{i=1}^{m} p_i M_{i,2} \quad \cdots \quad \sum_{i=1}^{m} p_i M_{i,n} \right] = [\alpha_1 \ \alpha_2 \ \cdots \ \alpha_n]$$

We want to find q that minimises $p^T M q$ s.t. $\sum_{i=1}^{n} q_i = 1$

$$p^T M q = \sum_{i=1}^{n} \alpha_i q_i \geq \min_i \alpha_i \sum_{i=1}^{n} q_i = \min_i \alpha_i \Rightarrow$$ Choose the action that minimises

Corollary:

$$\forall M, \quad \max_p \min_j p^T M e_j = \min_q \max_i e_i^T M q$$



We want to show,

~~$\forall p$ (for every rand. algo.) ~ dbn over A~~

$\forall p$ (for every rand. algo.) ~dbn. over A

$\exists i$ (some bad input) s.t.

$\mathbb{E}(c(A, I_i))$ is large $\Rightarrow$ some row "sum" is large

(remember, pay off is still not det., because of p)

It suffices to show that all col. "sums" are large.

The principle:

$$\forall M, p, q, \quad \min_j (p^T M e_j) \leq \max_i (e_i^T M q)$$

$p^T M e_j$: exp. cost of a det. algo. when I/P is sampled acc. to p

$e_i^T M q$: expected cost of a rand. algo.

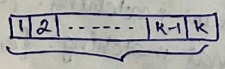
⇓ we want to show this is large

⇓ Find a worst-case I/P dbn for all det. algos.

15/9/23

Thm: For any c-competitive online paging algo., $c \geq H_k$

Pf:



(P)

I/P seq. $r_1 = k+1$

$r_2 \in_r \{1, 2, \ldots, k\}$

$r_i \in_r \{1, 2, \ldots, k+1\} \setminus \{r_{i-1}\}$

Divide the reqs. into rounds
- In each round, there are k distinct page reqs.

OPT: At most 1 fault per round (At first faults evict element that won't be req. this round)

Fix a det. algo. A

A can be uniquely characterized at each step by the element outside the cache.

A cache miss occurs if the req. is the element outside the cache. Notice that it is a part of the current valid reqs., (o.w., it would have been req. right before ⇒ in the cache)

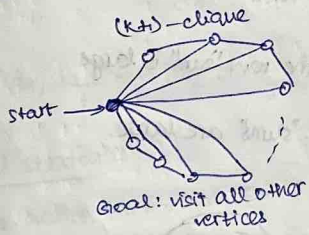
$\Pr(\text{A req causes a cache miss}) = \frac{1}{k}$

$C \geq H_k$

~~$\Rightarrow$ Exp. cache misses per rd~~

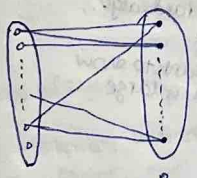
$\mathbb{E}(\text{No. of cache misses per round}) = \frac{1}{k}\,\mathbb{E}(\text{length of the round}) \geq H_k$

Each round requires k distinct reqs. $\geq kH_k$, like from coupon collector's



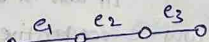
~ coupon collector's prob., but we only want to visit k of $k+1$

Online bipartite matching



Maximum matching:
- Matching having the largest cardinality over all possible matchings

Maximal matching:
- Matching that cannot be extended



Maximum matching: $\{e_1, e_3\}$
Maximal matching: $\{e_1, e_3\}$, $\{e_2\}$

Computing maximal matching online is easy, just see if there is an edge to a free vertex in L & add if so

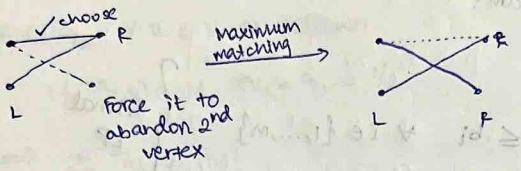
Det. greedy algo
- Add an edge if possible (maximal matching)

How good are maximal matchings, w.r.t. maximum matchings?

$\geq \frac{1}{2}$-size $\Rightarrow$ online 2-competitive algo. for maximum matching

2- competitive is the best we can get (deterministically)

M: Greedy maximal matching
OPT: Max. matching
size of



To show: $OPT \leq 2|M|$

There exists a randomized online matching algo w/ c.r. $\left(1-\frac{1}{e}\right)$

20/9/23

[rand. online $\geq \left(1-\frac{1}{e}\right)$ max. matching]

Midsem discussion

2. a) $n_1 = \#$ of voters for C_1

$n_2 = n - n_1$

$$E(X) = \sum_{k=0}^{m} k \cdot \frac{\binom{n_1}{k}\binom{n_2}{m-k}}{\binom{n_1+n_2}{m}} = \frac{n_1 m}{n_1+n_2} = p_1 m$$

Or, show that $Pr(X_i = 1) = p_1$

$$Pr(X_{i+1}=1) = \sum_{k=0}^{i} \frac{\binom{n_1}{k}\binom{n_2}{i-k}}{\binom{n_1+n_2}{i}} \cdot \frac{n_1-k}{n_1+n_2-i}$$

A linear program for matching

For every edge e, define $x_e = \begin{cases} 1, & \text{if } e \text{ is in the matching} \\ 0, & \text{o.w.} \end{cases}$

$$\max \sum_{e \in E} x_e$$

$$\text{s.t.} \sum_{u \in N(v)} x_{(u,v)} \leq 1 \quad \forall v \in V$$

$$x_e \in \{0,1\} \quad \forall e \in E$$

} defines all possible matchings

i.e., Every feasible soln is a matching

In general, integer programs are v. hard to solve

$\Rightarrow$ we usually relax constraints to reals rather than integers

A relaxed linear program: $\longrightarrow$ solvable in poly. time

$$\max \sum_{e \in E} x_e$$

$$\text{s.t.} \sum_{u \in N(v)} x_{(u,v)} \leq 1 \quad \forall v \in V$$

$$x_e \geq 0 \quad \forall e \in E$$

→ for example, (0.5, 0.5, 0.5) is feasible

However, this may beget non-integral solns., which cannot correspond to a true soln!

21/9/23

A general linear program:

$$\max \sum_{i=1}^{n} c_i x_i$$

$$\text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \forall\, i \in \{1,\dots,m\}$$

$$x_i \geq 0 \quad \forall\, i \in \{1,\dots,n\}$$

} Primal LP

Finding upper bounds on the obj. fn. allows us to estimate the soln. & tighter bounds bring us closer to the maximum. We can do this in a principled way using another LP.

$$y_1, y_2, \dots, y_m \geq 0$$

$$y_1\left(\sum_{j=1}^{n} a_{1j} x_j\right) + y_2\left(\sum_{j=1}^{n} a_{2j} x_j\right) + \dots + y_m\left(\sum_{j=1}^{n} a_{mj} x_j\right) \leq \sum_{i=1}^{m} b_i y_i$$

$$\Rightarrow x_1\left(\sum_{i=1}^{m} a_{i1} y_i\right) + x_2\left(\sum_{i=1}^{m} a_{i2} y_i\right) + \dots + x_n\left(\sum_{i=1}^{m} a_{in} y_i\right) \leq \sum_{i=1}^{m} b_i y_i$$

If we force $\geq c_1$, $\geq c_2$, $\dots$, $\geq c_n$

$$\sum_{i=1}^{n} c_i x_i \leq$$

upper bound on the obj. fn.

Dual LP

$$\min \sum_{i=1}^{m} b_i y_i$$

$$\text{s.t.} \sum_{i=1}^{m} a_{i1} y_i \geq c_1$$

$$\vdots$$

$$\sum_{i=1}^{m} a_{in} y_i \geq c_n$$

$$y_i \geq 0 \quad \forall\, i \in \{1,\dots,m\}$$

Every feasible soln. for the dual is an upper bound on the optimum (maximum) of the primal

Weak-duality

$$\text{OPT(Primal)} \leq \text{OPT(Dual)}$$

Strong-duality

$$\text{OPT(Primal)} = \text{OPT(Dual)}$$

Dual program for "matching"

$y_v, v \in V$

$$\min \sum_{v \in V} y_v$$

$$\text{s.t.} \quad y_u + y_v \geq 1 \quad \forall\, (u,v) \in E$$

[Each edge shows up in the constraint for the vertices it is incident on]

$$y_u \geq 0 \quad \forall\, u \in V$$

Given the maximal matching M, we construct a feasible soln. for the dual.

Define q_u $\forall u \in V$ as follows:

If $(u,v) \in M$, $q_u = q_v = 1/2$

$\Rightarrow \sum_{u \in V} q_u = |M|$

However, this is not a feasible assignment to y_u $(y_u = q_u)$ since edges not in the matching do not have their constraints satisfied.

What about $y_u = 2q_u$?

This is feasible. Any vertex not in the matching has a neighbor in M

$\Rightarrow$ The constraint is satisfied for all edges

$\Rightarrow OPT \le \sum_{u \in V} y_u = 2 \sum_{u \in V} q_u = 2|M|$

(weak-duality)

Primal-dual analysis (in general)

- Formulate the primal LP
- Use some algo. to construct a dual feasible soln.
- Apply weak-duality

Online fractional matching

Essentially solving the relaxed LP for matching, in an online fashion.

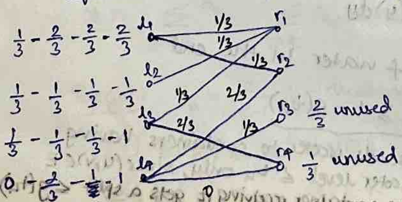
$x_e \ge 0$ instead of $x_e \in \{0,1\}$

f - fractional matching

$\Rightarrow f \ge (1 - \frac{1}{e})\, OPT$

Waterlevel algo.

The vertices in L are imagined as containers that can hold one unit of water. The vertices in R are like reservoirs containing one unit of water. Edges are like pipes. Each new vertex brings with it some pipes & transfers water to the least full container (equally divided, if not unique) until the containers fill or the reservoir empties (if water levels equalise in between, the supply is further divided).



$\Rightarrow$ Matching size: 3

Maximum matching: 4

$(l_1, r_2), (l_2, r_1), (l_3, r_4), (l_4, r_3)$

$(u,v) \in E$

Fix $l_{u'}$ $\forall u' \in L \setminus \{u\}$

Run the algo. on $G \setminus \{u\}$. Let u^* be the vertex matched to v.

Fix $\& \; l_{u^*} = 1$ if v is unmatched

Claim: If $\& \; l_u < l_{u^*}$, then u is matched in G.

Pf: (1) u is already matched by the time v arrives

(2) Else, u gets matched to v

Since u hasn't been matched yet, the run of the algo. on G & $G \setminus \{u\}$ are identical before v arrives.

$\forall v' \in N(v) \setminus \{u^*\}$, $l_{v'} > l_{u^*} \Rightarrow l_u < l_{u'} \quad \forall u' \in N(v) \setminus \{u\}$

$\Rightarrow$ u gets matched to v

[If v was not matched in $G \setminus \{u\}$, now it gets matched to u]

If u is matched, $q_u = e^{l_u - 1}$

$$\mathbb{E}(q_u) \geq \int_0^{l_{u^*}} e^{y-1}\,dy = e^{l_{u^*}-1} - \frac{1}{e}$$

$\sum q_u \cdot \Pr(l_u < l_{u^*})$ (conditioned on $l_{u'}$ $\forall u' \in L \setminus \{u\}$)

[$l_u < l_{u^*}$ is a lower bound (smaller event) for u being matched in G]

Claim: $q_v \geq 1 - e^{l_{u^*}-1}$ $\forall$ values of l_u

Pf: Run the algo. on G & $G \setminus \{u\}$ in parallel

The set of unmatched vertices in G is a superset of those in $G \setminus \{u\}$

$\Rightarrow$ u^* is present in G as well when v arrives.

$\Rightarrow$ The vertex chosen by G, say, u', is s.t.

$$l_{u'} < l_{u^*}$$

$$\Rightarrow q_v \geq 1 - e^{l_{u^*}-1}$$

$$\Rightarrow \mathbb{E}(q_u + q_v) \geq 1 - \frac{1}{e} \quad \forall (u,v) \in E$$

$\therefore$ set $y_u = \frac{e}{e-1} q_u$

$$\mathbb{E}(y_u) + \mathbb{E}(y_v) \geq 1 \quad \forall (u,v) \in E$$

$$\Rightarrow OPT \leq \frac{e}{e-1} \sum_{u \in L} \mathbb{E}(y_u) + \sum_{v \in R} \mathbb{E}(y_v) = \frac{e}{e-1} \sum_{u \in L \cup R} \mathbb{E}(q_u) = \frac{e}{e-1} \mathbb{E}(|M|)$$

27/9/23

Multiplicative weights update

Task: Make Yes/No decisions
wrong decisions incur unit cost

Goal: Minimize cost

We also have n experts $e_1, e_2, \ldots, e_n$

No. of timesteps $= T$, m_i^T = Loss incurred by expert i

$\Rightarrow$ min $\left[m^T - \min_i m_i^T \right]$ regret

($\min_i m_i^T$: Mistakes made by best expert)

① Omniscient expert

- Never makes a mistake

S: set of potential omniscient experts

$S \leftarrow \{e_1, e_2, \ldots, e_n\}$

Consider the majority answer of all $e \in S$.
If the answer is correct, then no cost is incurred.
Else, S becomes halved.

$\Rightarrow m^T - \min_i m_i^T \leq \log_2^n$ [Assuming that there is an omniscient expert]

② Weighted majority algo.

$w_i^{(t)}$: weight assigned to expert i at the t^{th} timestep.

$w_i^{(0)} = 1 \quad \forall i \in \{1, \ldots, n\}$

If $\sum_{i, e_i \text{ says yes}} w_i^{(t)} \geq \sum_{i, e_i \text{ says no}} w_i^{(t)}$, answer Yes

else, answer No

All experts i with wrong answer have $w_i^{(t+1)} = (1-\epsilon) w_i^{(t)}$

Thm: Let $m_i^{(T)}$ be the loss incurred by expert i,
$m^{(T)}$ be the loss incurred by the algo.

~~Then~~, $m^{(T)} \leq 2(1+\epsilon) m_i^{(T)} + \frac{2 \ln n}{\epsilon} \quad \forall i \in \{1, \ldots, n\}$

Thm: Every det. algo. will make at least twice as many mistakes as the best expert.

Pf: Suppose there are 2 experts e_1, e_2
e_1 always answers Yes, e_2 always answers No.

$\min_{i=1} \min_j = 1 \cdot M \leq \sum_{i=1} = T$

Adversary can always give the opposite answer as the algo.

$M^T = T$, but $\min_i m_i^T \leq \frac{1}{2}T$ $\quad \left[m_1^T + m_2^T = T \Rightarrow m_1^T > \frac{T}{2}, m_2^T > \frac{T}{2} \right]$ ∆ impossible]

Pf (of prev. thm.):

Potential fn, $\phi(t) = \sum_{i=1}^{n} w_i^{(t)}$

$\phi(0) = n$

$\phi(t+1) = \underbrace{\sum_{i, e_i \text{ said yes}} w_i^{(t+1)}}_{} + \underbrace{\sum_{i, e_i \text{ said No}} w_i^{(t+1)}}_{}$

Let correct answer at timestep t be No.

$= (1-\epsilon) \underbrace{\sum_{i, e_i \text{ said yes}} w_i^{(t)}}_{} + \underbrace{\sum_{i, e_i \text{ said No}} w_i^{(t)}}_{}$

Algo. made error in the t^{th} step ⇓

$\sum_{i, e_i \text{ said yes}} w_i^{(t)} \geq \sum_{i, e_i \text{ said No}} w_i^{(t)}$

$\geq \frac{\phi(t)}{2}$

$= \phi(t) - \epsilon \sum_{i, e_i \text{ said yes}} w_i^{(t)}$

$\leq \phi(t)\left(1 - \frac{\epsilon}{2}\right)$ (no. of mistakes $= m^{(T)}$)

Also, $w_i^{(t)} = (1-\epsilon)^{m_i^{(t)}}$

$\Rightarrow \sum_{i=1}^{n} w_i^{(T)} = \phi(T) \leq \phi(0)\left(1-\frac{\epsilon}{2}\right)^{m^{(T)}} = n\left(1-\frac{\epsilon}{2}\right)^{m^{(T)}}$ ~~future scope of this come from?~~

$\Rightarrow (1-\epsilon)^{m_i^{(T)}} \leq \phi(T) \leq \left(1-\frac{\epsilon}{2}\right)^{m^{(T)}} \cdot n$

$\therefore m^{(T)} \leq m_i^{(T)} \frac{\ln(1-\epsilon)}{\ln\left(1-\frac{\epsilon}{2}\right)} - \frac{\ln n}{\ln\left(1-\frac{\epsilon}{2}\right)}$ → complete to get the bound

4/10/23

Randomized version

Cost incurred, $m_i^{(t)} \in [-1, 1]$

Dbn. $p^{(t)}$ over the experts [which expert to go by]

⇒ choose e_i acc. $p^{(t)}$ & give the same answer as e_i

$m_1^{(t)}, m_2^{(t)}, \ldots, m_n^{(t)}$ – costs incurred by the experts

⇒ Update $p^{(t)} \longrightarrow p^{(t+1)}$

Expected cost for the algo. in time $t = \sum_{i=1}^{n} m_i^{(t)} p_i^{(t)}$

Expected total cost $= \sum_{t=1}^{T} m^{(t)} \cdot p^{(t)}$

Regret $= \sum_{t=1}^{T} m^{(t)} \cdot p^{(t)} - \min_i \sum_{t=1}^{T} m_i^{(t)}$

MWU algo.

$\eta \le 1/2$, $w_i^{(1)} = 1 \quad \forall i \in \{1, \dots, n\}$

For each t, $\Phi(t) = \sum_{i=1}^{n} w_i^{(t)}$

$$p^{(t)} = \left\{ \frac{w_1^{(t)}}{\Phi(t)}, \frac{w_2^{(t)}}{\Phi(t)}, \dots, \frac{w_n^{(t)}}{\Phi(t)} \right\}$$

Observe $m_1^{(t)}, m_2^{(t)}, \dots, m_n^{(t)}$.

$\forall i \in \{1, \dots, n\}$, $w_i^{(t+1)} = w_i^{(t)} (1 - \eta m_i^{(t)})$

[Large $m_i^{(t)}$ → large loss ⇒ Make $w_i^{(t)}$ a lot smaller
Neg. $m_i^{(t)}$ ⇒ inc. $w_i^{(t)}$]

Thm: $\forall i \in \{1, \dots, n\}$,

$$\underbrace{\sum_{t=1}^{T} m^{(t)} \cdot p^{(t)}}_{\text{Total expected cost}} \le \sum_{t=1}^{T} m^{(t)} + \eta \sum_{t=1}^{T} |m^{(t)}| + \frac{\ln n}{\eta}$$

[Similar to prev. result w/o a factor of 2]

* If $m_i^{(t)}$ corresponds to the gain instead of the loss/cost, the update rule becomes: $w_i^{(t+1)} = w_i^{(t)} (1 + \eta m_i^{(t)})$

& the bound on the total expected gain becomes:

$$\sum_{t=1}^{T} m^{(t)} \cdot p^{(t)} \ge \sum_{t=1}^{T} m^{(t)} - \eta \sum_{t=1}^{T} |m^{(t)}| - \frac{\ln n}{\eta}$$

Simply make $m_i^{(t)} \to -m_i^{(t)}$

Learning a linear classifier (winnow algo.) [winnowing → separating lol]

Datapoints $\{(\bar{a}_i, l_i)\}_{1 \le i \le m}$ [Littlestone, 1988] $\quad \bar{a}_i \in \mathbb{R}^n$, $l_i \in \{\pm 1\}$

Goal: Find $\bar{w} \in \mathbb{R}^n$ s.t.

$\forall i$, $\operatorname{sgn}\left(\sum_{j=1}^{n} a_{ij} w_j\right) = l_i$ → $\sum_{j=1}^{n} a_{ij} w_j \ge 0$, by absorbing l_i into $\bar{a}_i$

Assumptions: ① $|a_{ij}| \le 1$

Additional constraints w.l.o.g.:
② $\sum_{i=1}^{n} w_i = 1$
③ $w_i \ge 0 \quad \forall j \in [n]$ → w.l.o.g., though it may increase soln. space from $\mathbb{R}^n$ to $\mathbb{R}^{2n}$, by adding $-\bar{a}_i$ to get an extended vector in $\bar{a}_i$
④ $\exists w^*$ s.t. $\forall i \in [m]$ $\sum_{j=1}^{n} a_{ij} w_j^* \ge \epsilon$

5/10/23

Algo:

$w_i^{(1)} = \frac{1}{n} \quad \forall i \in [n]$

For $t = 1$ to T:

Check if $\forall j$ $\bar{a}_j \cdot \bar{w}^{(t)} \ge 0$ (→ normalized)

If $\exists j$ s.t. $\bar{a}_j \cdot \bar{w}^{(t)} < 0$,

$w_i^{(t+1)} = w_i^{(t)} \left(1 + \frac{\epsilon}{2} a_{ji}\right)$

T is yet to be fixed

$w^{(t)} \approx p^{(t)}$

$a_{ji}^{(t)} \approx m_i^{(t)}$ for $\bar{a}_j \cdot \bar{w}^{(t)} < 0$

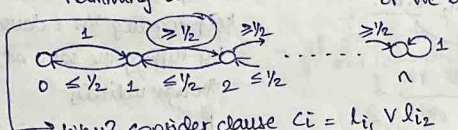
Algo. MWU

Rand. 2-SAT as a Markov chain

No. of States = $n+1$

State i is associated w/ assignments that are at a Hamming dist. of $n-i$ from a sat. assignment

Each bit flip either increases or decreases the Hamming distance by 1. [Assuming there is 1 sat. assignment, or we only analyse for one]



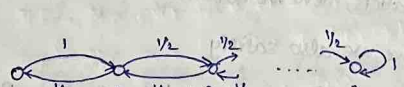
→ Why? consider clause $C_i = l_{i_1} \vee l_{i_2}$

Both l_{i_1} & l_{i_2} cannot match the sat. assignment, otherwise $C_i = 1$.

$\Rightarrow$ w.p. ½ (or 1), the flipping moves closer to the sat. assignment $\forall C_i$ (that are 0)

However, this is not a Markov chain, since the transition probas. are unknown. Instead, let us fix the $\{\geq ½, \leq ½\}$ edge wts. as ½. If this reaches state n in t steps in expectation, so will the actual process.

M



Expected time for M to go from 0 to n?

Y_i = Time for M to go from i to n

$$\mathbb{E}(Y_i) = \frac{1}{2}\left(1 + \mathbb{E}(Y_{i-1})\right) + \frac{1}{2}\left(1 + \mathbb{E}(Y_{i+1})\right)$$

$$= \frac{1}{2}\mathbb{E}(Y_{i-1}) + \frac{1}{2}\mathbb{E}(Y_{i+1}) + 1$$

$$\mathbb{E}(Y_0) = 1 + \mathbb{E}(Y_1)$$

$$\mathbb{E}(Y_n) = 0$$

Verify that $\mathbb{E}(Y_i) = n^2 - i^2$

$\Rightarrow$ By running for $2n^2$ steps, we find a sat. assignment w.p. $\geq ½$

Note that this won't work for 3-SAT, since the trans? probas. forward can only be lower-bounded by ⅓.

9/10/23

3-SAT

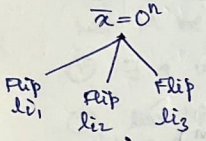
$\phi = C_1 \wedge C_2 \wedge C_3 \wedge \dots \wedge C_m$

$C_i = l_{i_1} \vee l_{i_2} \vee l_{i_3}$

$\exists\, \bar{x} \in \{0,1\}^n$ s.t. $\phi(\bar{x}) = 1$?

* NP-complete
* Trivial algo: $O^*(2^n)$ $[O(2^n \text{poly}(n))]$

Deterministic $O^*(1.732^n)$



⋮ complete in n steps

$\Rightarrow 3^n \times$ too bad

Obs: Every assignment $\bar{x} \in \{0,1\}^n$ is at a Hamming distance of $\leq \frac{n}{2}$ from 0^n or 1^n

$\Rightarrow \frac{n}{2}$ steps on 0^n, if not satisfied, start over from 1^n & do $\frac{n}{2}$ steps

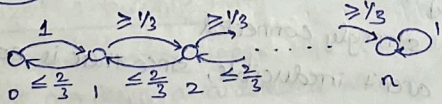
Randomized 3-SAT (1)

- choose $\bar{x} \in_r \{0,1\}^n$ → only change (random instead of arbitrary)
- If $\phi(\bar{x}) \neq 1$

  Let $C_i = l_{i_1} \vee l_{i_2} \vee l_{i_3}$ s.t. $C_i(\bar{x}) = 0$

  choose $l_{i_1}, l_{i_2}, l_{i_3}$ u.a.r. & flip

n steps



If ϕ is satisfiable,

$$\Pr(\text{success}) \geq \sum_{k=0}^{n} \underbrace{\binom{n}{k} \cdot \frac{1}{2^n}}_{\substack{\text{chosen } \bar{x} \text{ has Hamming} \\ \text{dist. } k \text{ from} \\ \text{correct } \bar{x}}} \cdot \underbrace{\left(\frac{1}{3}\right)^k}_{\substack{\text{Proba. of} \\ \text{moving forward} \\ \text{until soln.} \\ \text{v. weak!}}}$$

$$\geq \frac{1}{2^n} \cdot \left(\frac{4}{3}\right)^n = \left(\frac{2}{3}\right)^n$$

$\Rightarrow$ repeating $\left(\frac{3}{2}\right)^n$ times gives a soln. w.h.p

$\therefore O^*(1.5^n)$

Schöning's algo.

Suppose $\bar{x}$ has Hamming wt. k

Proba. that the algo. makes ℓ moves in the wrong direction $= \binom{k+2\ell}{\ell}\left(\frac{2}{3}\right)^{\ell}\left(\frac{1}{3}\right)^{k+\ell}$

(for the Markov chain)

$\Pr(\text{Algo reaches sat. assignment}) \geq \max_{\ell} \binom{k+2\ell}{\ell}\left(\frac{2}{3}\right)^{\ell}\left(\frac{1}{3}\right)^{k+\ell}$

$\geq \underbrace{\binom{3k}{k}\left(\frac{2}{3}\right)^{k}\left(\frac{1}{3}\right)^{2k}}_{\ell = k} \geq \frac{c}{\sqrt{k}}\left(\frac{1}{2}\right)^{k}$

$$\sqrt{2\pi k}\left(\frac{k}{e}\right)^{k} \leq k! \leq 2\sqrt{2\pi k}\left(\frac{k}{e}\right)^{k}$$

$$\Rightarrow \Pr(\text{success}) \geq \sum_{k=0}^{n} \binom{n}{k} \cdot \frac{1}{2^n} \cdot \underbrace{\frac{c}{\sqrt{k}}}_{\text{Ignored}} \cdot \frac{1}{2^k}$$

$$\geq \left(\frac{3}{4}\right)^{n} \Rightarrow \text{Running time } O^*(1.33^n)$$

More about Markov chains

* vector Π – dbn. over Ω

  ΠP – dbn. over Ω after one timestep (P = transition matrix)

  ΠP^n – dbn. over Ω after n timesteps of the random walk

  P^n : n-step transition matrix

* If $\exists n$ $P^n_{i,j} > 0$, $\exists$ a way to reach j from i (in n steps)

  Defn: A Markov chain M is irreducible if

  $\forall i,j$, $\exists n$ $P^n_{i,j} > 0$

  $\equiv$ The digraph is strongly connected

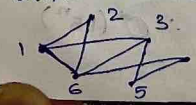
  Our satisfiability MCs aren't irreducible, since there is no way out from state n

** Defn: State j in a MC M is periodic if $\exists \Delta > 1$ s.t.

$$\Pr(X_{s+t} = j \mid X_s = j) = 0 \text{ unless } \Delta \mid t$$

A MC M is aperiodic if it does not have any periodic states.

* $G(V,E)$ – undirected



Random walk on G:

	1	2	3	4	5	6
1	0	1/3	1/3	0	0	1/3
2	1/2	0	0	0	0	1/2
3						
4						
5						
6						

If G is connected, the random walk MC is irreducible

If G is bipartite, the random walk MC is periodic ($\Delta = 2$)

13/10/23

Exercise: show that the random walk on an undirected graph is aperiodic iff the graph is not bipartite

[For non-bipartite graphs, we use the odd cycle to show it]

* Stationary dbn.

A dbn π on the MC is stationary if

$$\pi P = \pi$$

* Fundamental thm. of MCs

Let M be a finite irreducible aperiodic MC.

① $\exists$ a unique stationary dbn. π

② $\lim_{t \to \infty} P^t_{i,j}$ exists & is indep. of i [Proba. of reaching j, indep. of where we start]

③ $\pi_j = \lim_{t \to \infty} P^t_{i,j} = \frac{1}{h_{j,j}}$

$h_{j,j}$ = Expected time for the MC to return to j starting from j

* Undirected, non-bipartite, connected G

⇓ aperiodic ⇓ irreducible

What is π s.t. $\pi P = \pi$?

we know, $P_{u,v} = \frac{1}{deg(u)}$

Verify: $\pi(u) = \frac{deg(u)}{2M}$, $M = |E|$

π is uniform when G is regular!

Randomized s-t connectivity

- BFS/DFS takes linear time & space
- we can design a randomized algo. via a random walk, using no extra space (besides the curr. index)

Algo:

Perform a random walk for l steps from s & check if t was visited

- $O(\log n)$ space ⟶ RL space (randomized logspace)
- $O(l)$ time $= O(mn)$

↳ in 2004, shown to be solvable in det. $O(\log n)$ space

Cover time: Expected time for a random walk to visit all vertices

Claim: $\forall (u,v) \in E,\ \underbrace{h_{u,v} + h_{v,u}}_{\text{commute time}} \leq 2|E|$

$h_{u,v}$: Expected time for a random walk starting from u to reach v

Pf: Consider M' over directed edges by charging $(u,v) \longrightarrow \underbrace{u \to v,\ v \to u}_{\text{states of } M'}$

Assume M is not bipartite (can do so by adding self-loops)

No. of states of M' is $2|E|$

M' is aperiodic & irreducible.

$P_{u \to v, v \to w} = \frac{1}{\deg(v)}$

Exercise: Stationary dbn. is uniform $\frac{1}{2|E|}$.

$\Rightarrow h_{u,u} = 2|E|$

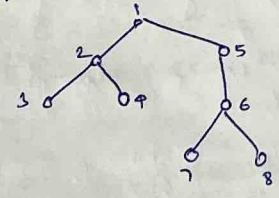
$\Rightarrow h_{u,v} + h_{v,u} \leq 2|E|$ $\quad \forall (u,v) \in E$,

16/10/23

Lemma: Cover time of the random walk $\leq 2m(n-1)$

Expected time to visit all vertices in the component

Pf: Spanning tree of G : T.



DFS traversal:

1, 2, 3, 2, 4, 2, 1,
5, 6, 7, 6, 8, 6, 5, 1.

Consider a random walk, forcing a visiting in this DFS order, i.e, we are first interested in $1 \to 2$, then $2 \to 3$, then $3 \to 2$, so on.

Expected len. of the random walk $\leq \sum_{(u,v) \in T} h_{u,v} + h_{v,u}$

$= 2m(n-1)$

$< 2n^3$ $\Rightarrow$ can random walk for $8n^3$ steps & get error rate of $\leq \frac{1}{4}$

Monte Carlo methods – counting & sampling $\rightsquigarrow$ Idea: "Throw darts", based on sampling & identifying properties

Defn: Let V be a qty. to be estimated. X is an (ϵ, δ)-approx. (r.v.) of V if $\Pr(|X - V| \geq \epsilon V) \leq \delta$.

Fully Poly-time Randomized Approx Scheme (FPRAS)

Task: Given x, calculate $V(x)$

An algo. A is an FPRAS if it takes x, ϵ, δ as I/P s.t.

$A(x, \epsilon, \delta)$ is an (ϵ, δ)-approx. of $V(x)$ & runs in time $\text{poly}\left(\frac{1}{\epsilon}, \ln\left(\frac{1}{\delta}\right), |x|\right)$

DNF counting

$\phi = T_1 \vee T_2 \vee \dots \vee T_m \longrightarrow$ DNF

$T_i = l_{i_1} \wedge l_{i_2} \wedge \dots \wedge l_{i_r}$

* DNF-SAT $\in P$
* #-DNF (Sharp DNF) $= |\{\bar{a} \mid \phi(\bar{a}) = 1\}| \in$ #P-complete

No. of satisfying assignments

Trivial algo: Sample $\bar{a} \in \{0,1\}^n$ & check if $\phi(\bar{a}) = 1$ } Reqs. $\frac{n^{O(1)}}{2^n}$ repeats if no. of sat. assignments is poly(n)

Expressing the problem differently:

Sets $S_1, S_2, \dots, S_m \equiv$ sat. assignments of $T_1, T_2, \dots \to T_m$

Given x, check $x \overset{?}{\in} S_i$

Computing $|\bigcup_{i=1}^{m} S_i|$

- $S_i = \{\bar{a} \mid T_i(\bar{a}) = 1\}$

  $T_i = \overline{x_1} \wedge x_2 \wedge x_3 \wedge \overline{x_4}$ (say)
  0 1 1 0 $\longrightarrow$ necessary to satisfy

  other variables can be set arbitrarily

  $\Rightarrow$ computing $|S_i|$ is easy enough

  $|S_i| = 2^{n-k}$, if T_i is satisfiable & T_i has k literals

- Can sample u.a.r. from S_i

$C_\phi = \{(i, \bar{a}) \mid T_i(\bar{a}) = 1\}$

$|C_\phi| = \sum_{i=1}^{m} |S_i|$

Obs: $|C_\phi| \leq m |\bigcup_{i=1}^{m} S_i|$

$S_\phi = \{(i, \bar{a}) \mid T_i(\bar{a}) = 1 \wedge \forall j < i,\ T_j(\bar{a}) = 0\}$

$\equiv T_i$ is the first term satisfied by $\bar{a}$

Obs: $|S_\phi| = |\bigcup_{i=1}^{m} S_i| \Longrightarrow S_\phi$ is similar in size to C_ϕ, which avoids the problem we faced before

Algo:

- Sample i w.p. $|S_i| / \sum |S_i|$ } sample $(i, \bar{a})$ u.a.r. from C_ϕ
- sample $\bar{a}$ u.a.r. from S_i
- check if $(i, \bar{a}) \in S_\phi$

18/10/23 #SAT ↷ generic boolean formula satisfiability

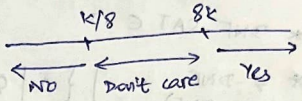
Thm: If SAT ∈ P, then there is an FPRAS for #SAT.

GapSAT

I/P: ϕ, k

O/P: Yes, if $\#(\phi)$ (No. of sat-assignments) $> 8k$

No, if $\#(\phi) < k/8$



Lemma 1: If SAT ∈ P, GapSAT has an efficient randomized algo.

Lemma 2: If ∃ an efficient rand.-algo for GapSAT, then ∃ algo. A & $i \in \mathbb{N}$ s.t. $\Pr\left[\frac{\#(\phi)}{2^i} \le A(\phi) \le \#(\phi).2^i\right] \ge 1 - \frac{1}{n}$

Lemma 3: If ∃ algo A as described in lemma 2, then #-SAT has an FPRAS.

Pf. of lemma 3:

Given ϕ, define $\phi' = \bigwedge_{K=1}^{j} \phi(\bar{x}^{(K)})$

$\#(\phi') = \#(\phi)^j$

- choose $j = i/\epsilon$
- construct ϕ'
- compute $A(\phi')^{1/j}$

$$\Pr\left[\frac{\#(\phi')}{2^i} \le A(\phi') \le \#(\phi').2^i\right] \ge 1 - \frac{1}{n}$$

$$\Rightarrow \Pr\left[\underbrace{\frac{\#(\phi')^{1/j}}{2^{i/j}}}_{\frac{\#(\phi)}{2^{i/j}}} \le \underbrace{(A(\phi'))^{1/j}}_{\text{FPRAS O/P}} \le \underbrace{\#(\phi')^{1/j}.2^{i/j}}_{\#(\phi).2^{i/j}}\right] \ge 1 - \frac{1}{n}$$

Pf. of lemma 2:

Find smallest i s.t.:

GapSAT$(\phi, 8^i)$ = Yes & GapSAT$(\phi, 8^{i+1})$ = No

$\Downarrow$ $\Downarrow$

$\#(\phi) \ge 8^{i-1} = \frac{A(\phi)}{8}$ $\quad$ $\#(\phi) \le 8^{i+2} = 64 A(\phi)$

choosing $A(\phi) = 8^i$

$\Rightarrow \frac{\#(\phi)}{\quad} \le A(\phi) \le 8\#(\phi) \le 64\#(\phi)$, i.e., $i = 6$

Lemma: Let $\mathcal{H}_{n,m}$ be a family of pairwise-indep. hash fns.

$$h: \{0,1\}^n \to \{0,1\}^m, \quad \text{let } \epsilon > 0.$$

Let $S \subseteq \{0,1\}^n$ s.t. $|S| \geq 2^m/\epsilon^3$.

$$\Pr_{h \in \mathcal{H}_{n,m}}\left[(1-\epsilon)\frac{|S|}{2^m} \leq \left|\{x \in S \mid h(x) = 0^m\}\right| \leq (1+\epsilon)\frac{|S|}{2^m}\right] \geq 1-\epsilon$$

Pf of lemma 1:

Given ϕ, k,

choose $m = \log k$

Define $\phi'(x) = \phi(x) \wedge (h(x) = 0^m)$, $h \in_r \mathcal{H}_{n,m}$

If $\#(\phi) \geq 8k$, then ϕ' is satisfiable.

If $\#(\phi) \leq \frac{k}{8}$, then ϕ' is not satisfiable.

$S = \{x \mid \phi(x) = 1\}$, $|S| \geq \frac{8 \cdot 2^m}{8k}$

Choose $\epsilon = 1/2$,

$\Pr\left(\left|\{x \in S \mid h(x) = 0^m\}\right| > 1\right) \geq 1/2$

$\to |S| \leq \frac{2^m}{8}$

$\Pr(\exists x \in S, h(x) = 0^m) \leq \Pr\left(\left|\{x \in S \mid h(x) = 0^m\}\right| \geq 1\right)$

$1 - \epsilon =$

$\leq \sum_{x \in S} \Pr(h(x) = 0^m)$

$\leq \frac{2^m}{8} \cdot \frac{1}{2^m}$

$\leq 1/8$

4/10/23

DNF-sampling

Given $\phi = T_1 \vee T_2 \vee \ldots \vee T_m$

Goal: Sample $\bar{x}$ u.a.r. s.t. $\phi(\bar{x}) = 1$

If $m = 1$, i.e., $\phi = T_1 = $ (say) $\bar{x}_1 \wedge x_2 \wedge \bar{x}_3 \wedge x_4$

$\phi(\bar{x}) = 1 \Rightarrow x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1$

$\Rightarrow$ The remaining $n-4$ bits can be set u.a.r.

$$\left|\{\bar{x} \mid \phi(\bar{x}) = 1\}\right| = 2^{n-4}.$$

For a more general ϕ, we sample a T_i & then sample $x \in \{0,1\}^n$

- Sample i w.p. $\frac{|S_i|}{\sum_{i=1}^{m} |S_i|}$ $\left(S_i = \{\bar{x} \mid T_i(\bar{x}) = 1\}\right)$

- Sample u a.r. from S_i

Fix $\bar{a} \in \{0,1\}^n$ s.t. $\phi(\bar{a}) = 1$.

$$\Pr(\bar{a} \text{ is sampled}) = \sum_{i \mid T_i(\bar{a})=1} \frac{|S_i|}{\sum_{i=1}^{m} |S_i|} \cdot \frac{1}{|S_i|}$$

$$= \frac{m(\bar{a})}{\sum_{i=1}^{m} |S_i|} \qquad \left(m(\bar{a}) = |\{i \mid T_i(\bar{a}) = 1\}|\right)$$

This is not yet uniform. In order to make it uniform, we need another sampling step w.p. $\frac{1}{m(\bar{a})}$, which is

- Accept $\bar{a}$ w.p. $\frac{1}{m(\bar{a})}$. Else continue

(rejection sampling)

Fully Polynomial Almost Uniform Sampler (FPAUS)

I/P : x

Associated w/ x is a set $\Omega(x)$, which we want to sample from. x may be a graph & $\Omega(x)$ may be the set of all indep. sets or the set of all matchings in x.

O/P : $w \in \Omega(x)$ (ϵ-sampler)

A sampling algo. generates an ϵ-uniform sample of Ω if w satisfies $\forall S \subseteq \Omega$, $\left|\Pr(w \in S) - \underbrace{\frac{|S|}{|\Omega|}}_{\text{actually uniform}}\right| \leq \epsilon$

we also want our algo. to run in $\text{poly}\left(\ln\left(\frac{1}{\epsilon}\right), |x|\right)$

This can also be expressed as follows (not proved in class yet)

Uniform dbn. over Ω, $\Pr(w) = \frac{1}{|\Omega|}$, $w \in \Omega$

Sampling algo. A defines a dbn. over Ω, $A_\Omega(w)$, $w \in \Omega$

s.t. $\frac{1}{2} \sum_{w \in \Omega} \left| \frac{1}{|\Omega|} - A_\Omega(w) \right| \leq \epsilon$

Thm: (Jerrum-Sinclair)

FPAUS $\iff$ FPRAS, for self-reducible problems
- satisfiability
- matching
- indep. set

Thm: If $\mathcal{J}$ an FPAUS for indep. sets, then $\mathcal{J}$ an FPRAS for indep. sets

(sampling $\Rightarrow$ counting)

Idea:

Given G, $\Omega(G)$ = set of all indep. sets in G

Suppose G has m edges $e_1, e_2, \ldots, e_m$.

G_i = Graph with $e_1, e_2, \ldots, e_i$.

$$|\Omega(G)| = \frac{|\Omega(G_m)|}{|\Omega(G_{m-1})|} \cdot \frac{|\Omega(G_{m-1})|}{|\Omega(G_{m-2})|} \cdot \ldots \cdot \frac{|\Omega(G_1)|}{|\Omega(G_0)|} \cdot |\Omega(G_0)|$$

$|\Omega(G_0)| = 2^n$, since $\forall V' \subseteq V$, V' is an indep set of G_0

This is where self-reducibility plays its role

↳ estimated by sampling from denominator using FPAUS & checking membership in numerator

We need to show that these ratios aren't too small

$$r_i = \frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|} \xrightarrow{\text{estimate}} \tilde{r}_i$$

Monte-Carlo

Output $2^n \prod_{i=1}^{m} \tilde{r}_i \approx 2^n \prod_{i=1}^{m} r_i$

↳ we want

$$R = \prod_{i=1}^{m} \frac{\tilde{r}_i}{r_i}$$

Lemma: If $\tilde{r}_i$ is an $\left(\frac{\epsilon}{2m}, \frac{\delta}{m}\right)$-approx. of r_i, then $\Pr(|R-1| \le \epsilon) \ge 1-\delta$

$\Rightarrow \Pr\left(|\tilde{r}_i - r_i| \le \frac{\epsilon}{2m} r_i\right) \ge 1 - \frac{\delta}{m}$

$\Downarrow$

w.p. $\ge 1-\delta$, $\forall i \; |\tilde{r}_i - r_i| \le \frac{\epsilon}{2m} r_i$

$\Rightarrow \left|\frac{\tilde{r}_i}{r_i} - 1\right| \le \frac{\epsilon}{2m}$

$\Rightarrow 1 - \frac{\epsilon}{2m} \le \frac{\tilde{r}_i}{r_i} \le 1 + \frac{\epsilon}{2m}$

↓ Product

we need a lower bound for $\frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|}$

* $\Omega(G_i) \subseteq \Omega(G_{i-1})$

Let (u,v) be the i^{th} edge e_i.

Consider $\Omega(G_{i-1}) \setminus \Omega(G_i)$.

Every $I \in \Omega(G_{i-1}) \setminus \Omega(G_i)$ contains both u & v.

[If it contains at most one of u,v, its presence in $\Omega(G_{i-1})$ $\Rightarrow$ presence in $\Omega(G_i)$]

Claim: ~~for~~ $|\Omega(G_{i-1}) \setminus \Omega(G_i)| \leq |\Omega(G_i)|$

$I \in \Omega(G_{i-1}) \setminus \Omega(G_i)$

I contains u & v

Remove v: $I \setminus \{v\} \in \Omega(G_i)$

$$\Rightarrow \frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|} \geq \frac{1}{2}$$

20/10/23

- Sample from $\Omega(G_{i-1})$ using FPAUS ⟶ How many times?
- Count the fraction of samples in $\Omega(G_i)$

Say we sample M ~~seen~~ times using an $\frac{\epsilon}{6m}$-sampler for $\Omega(G_{i-1})$

Let $X_k = \begin{cases} 1, & \text{the } k^{th} \text{ sample} \in \Omega(G_i) \\ 0, & \text{o.w.} \end{cases}$

By FPAUS defn., $(\Omega = \Omega(G_{i-1}),\ S = \Omega(G_i))$

$$\left| \Pr(X_k = 1) - \frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|} \right| \leq \frac{\epsilon}{6m}$$

$$\Rightarrow \left| E(X_k) - \frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|} \right| \leq \frac{\epsilon}{6m}$$

$$\Rightarrow \left| E\left(\frac{1}{M}\sum X_k\right) - \frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|} \right| \leq \frac{\epsilon}{6m}$$

$$\Rightarrow |E(\tilde{r}_i) - \underbrace{r_i}_{\geq 1/2}| \leq \frac{\epsilon}{6m}$$

For large enough m, $E(\tilde{r}_i) \geq \frac{1}{3}$

What should M be so that

$$|\tilde{r}_i - E(\tilde{r}_i)| \leq \frac{\epsilon}{12m} E(\tilde{r}_i)$$

we bound $\Pr\left[|\tilde{r}_i - E(\tilde{r}_i)| \leq \frac{\epsilon}{12m} E(\tilde{r}_i)\right] \geq 1 - \frac{\delta}{m}$

Using chernoff bounds, $M = O\left(\frac{m^2}{\epsilon^2} \ln\left(\frac{2m}{\delta}\right)\right)$

$$\Rightarrow 1 - \frac{\epsilon}{12m} \leq \frac{\tilde{r}_i}{E(\tilde{r}_i)} \leq 1 + \frac{\epsilon}{12m}, \text{ w.p. } \geq 1 - \frac{\delta}{m}$$

$$1 - \frac{\epsilon}{6mr_i} \leq \frac{E(\tilde{r}_i)}{r_i} \leq 1 + \frac{\epsilon}{6mr_i}$$

Using $r_i \geq 1/2$, we get $\boxed{1 - \frac{\epsilon}{2m} \leq \frac{\tilde{r}_i}{r_i} \leq 1 + \frac{\epsilon}{2m}}$

This is not yet a complete algo., since we don't have an algo. for FPAUS yet, and we have only constructed a red^{n}. from FPAUS to FPRAS. To construct an FPAUS, we want to design a Markov chain with states corresponding to the set we sample from & a uniform stationary dbn.

Markov chain Monte Carlo

Let Ω be a finite state space.

Consider a directed graph on Ω that is strongly connected.

$N(x) = \{ y \mid (x,y) \in E \}$

$M \geq \max_x |N(x)|$

Define the trans^{n} matrix P as

$$P_{x,y} = \begin{cases} \frac{1}{M}, & x \neq y,\ y \in N(x) \\ 1 - \frac{|N(x)|}{M}, & x = y \\ 0, & \text{o.w.} \end{cases}$$

P has a uniform stationary dbn

* We want to construct such an MC for indep. sets, but we cannot do it explicitly by finding all indep. sets, which may be exponential in number & hence, impractical.

Let X_i be an indep. set.

$\forall v \in V$, ⟶ defines the transition

- If $v \in X_i$, then $X_{i+1} = X_i \setminus \{v\}$
- If $v \notin X_i$ & $\{v\} \cup X_i$ is an indep. set, then $X_{i+1} = X_i \cup \{v\}$
- Else, $X_{i+1} = X_i$

If we show that this MC is irreducible & aperiodic, its stationary dbn. is uniform & we are done.

Irreducible:

$I_1 \longrightarrow I_2$

Remove all $v \in I_1 \setminus I_2$

Add all $v \in I_2 \setminus I_1$

Aperiodic:

If the graph has an edge, the MC has a self-loop, allowing arbitrarily many timesteps at the state w/ a self-loop ⇒ aperiodic

We also want to bound the time to convergence to the stationary dbn., since our FPAUS should run in poly(n), & $|\Omega| = O(2^n)$

25/10/23

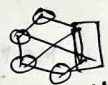
Monomer-Dimer problem (Mark Jerrum & Alistair Sinclair)

Graph $G(V, E)$, $|V| = 2n$

Molecules are placed on the graph s.t.

2 types of molecules { Monomers occupy one vertex each,
Dimers occupy adjacent vertices (i.e, an edge)

A <u>config.</u> is a placement of monomers & dimers s.t. every vertex has only one molecule.



an example config.

M_k = set of configs. w/ k dimers ($\Rightarrow 2(n-k)$ monomers)

$|M_k| = m_k$

Note that M_n = set of all perfect matchings

For a config. M, $|M|$ = no. of dimers in M

Our example config. has $|M| = 1$

Statistical Physicists study these & make use of a partition fn.

$$Z(\lambda) = \sum_M w(M)$$

$$w(M) = \lambda^{|M|}$$

$$Z(\lambda) = \sum_{k=0}^{n} m_k \cdot \lambda^k \longrightarrow \text{A generating fn. for the no. of matchings}$$

The task of computing $Z(\lambda)$ given λ is #P-complete.

Jerrum-Sinclair provided an FPRAS for the problem.

The idea is similar to the previous, writing as a telescopic product.

$$Z(\lambda) = \frac{Z(\lambda_r)}{Z(\lambda_{r-1})} \cdot \frac{Z(\lambda_{r-1})}{Z(\lambda_{r-2})} \cdot \cdots \cdot \frac{Z(\lambda_1)}{Z(\lambda_0)} \cdot Z(\lambda_0)$$

s.t. $0 = \lambda_0 < \lambda_1 < \lambda_2 < \cdots < \lambda_r = \lambda$

Sampling from a Gibbs dbn. $\pi_\lambda(M) = \frac{\lambda^{|M|}}{Z(\lambda)}$

allows us to deal w/ this product well $\rightsquigarrow$ we need to construct a Markov chain w/ such a stationary dbn.

Using $\lambda_i = \left(1 + \frac{1}{n}\right)^{i-1} \lambda_1$, $\lambda_1 = \frac{1}{|E|}$

Estimating $\frac{Z(\lambda_i)}{Z(\lambda_{i-1})}$. π_{λ_i}

Apply a few times
- Sample M acc. to the Gibbs dbn. (via the MC)
- Estimate $X = \left(\frac{\lambda_{i-1}}{\lambda_i}\right)^{|M|}$

we claim this is a good estimator

$$\mathbb{E}(X) = \sum_M \left(\frac{\lambda_{i-1}}{\lambda_i}\right)^{|M|} \cdot \frac{\lambda_i^{|M|}}{Z(\lambda_i)} = \frac{1}{Z(\lambda_i)} \underbrace{\sum_M (\lambda_{i-1})^{|M|}}_{Z(\lambda_{i-1})}$$

$$= \frac{Z(\lambda_{i-1})}{Z(\lambda_i)}$$

Now, we need a method to convert an MC w/ a uniform stationary dbn. to one w/ an arbitrary stationary dbn.

Metropolis algo.

Given a finite, irreducible, aperiodic MC w/ uniform stationary dbn. & given a dist. π over the states $\longrightarrow$ A new MC w/ the same state space w/ stationary dbn. π

A MC for matchings

M_i = current matching

What are the transitions M_{i+1}?

- w.p. $1/2$, $M_{i+1} = M_i$
- w.p. $1/2$, $\forall$ edges $e = (u,v) \in E$,
  - $M_{i+1} = M_i \setminus \{e\}$, if $e \in M_i$
  - $M_{i+1} = M_i \cup \{e\}$, if both u & v are unmatched in M_i
  - $M_{i+1} = M_i \setminus \{e'\} \cup \{e\}$, if exactly one of u & v is matched in M_i via $e' \in E$ (e chosen w.p. $1/|E|$)
  - $M_{i+1} = M_i$, o.w.

The MC is irreducible (can remove all matching edges & then add new ones) & aperiodic (self loops exist) $\Rightarrow$ it has a stationary dbn.

There are $|E|$ out edges in the second transition

$\Rightarrow |E|$ more from the first (self-loops)

$\Rightarrow 2|E|$ out-edges per state of MC

$\Rightarrow$ The MC is uniform over the set of all matchings

Recall the uniform MC we earlier constructed.

state space Ω, $N(x)$ = set of nbrs. of x, $x \in \Omega$

$$M \geq \max_x |N(x)|$$

$$P_{x,y} = \begin{cases} \frac{1}{M}, & \text{if } y \in N(x) \\ 0, & \text{if } y \notin N(x) \\ 1 - \sum_{y \in N(x)} P_{x,y}, & \text{if } y = x \end{cases}$$

↓ To get a dbn. π

$$P_{x,y} = \begin{cases} \frac{1}{M} \min\left\{1, \frac{\pi_y}{\pi_x}\right\}, & \text{if } y \in N(x) \\ 0, & \text{if } y \notin N(x) \\ 1 - \sum_{y \in N(x)} P_{x,y}, & \text{if } y = x \end{cases}$$

This needs the dbn. π. But, for matchings, notice that the transitions do not change the size of the matching by more than 1.

$$\Rightarrow \frac{\pi_y}{\pi_x} = \frac{\pi_{\lambda}(M')}{\pi_{\lambda}(M)} = \lambda^{\overbrace{|M'|-|M|}^{\in\{-1,0,1\}}} \in \left\{\frac{1}{\lambda}, 1, \lambda\right\}$$

$\Rightarrow$ A simple additional step in the MC is sufficient (for matchings)

— w.p. $\min\left\{1, \frac{\pi_{\lambda}(M_{i+1})}{\pi_{\lambda}(M_i)}\right\}$, move to M_{i+1}

26/4/23

Lemma: Let M be a finite, irreducible, aperiodic MC w/ transition matrix P. Let π be some dbn. over the state space Ω.

If $\forall x, y \quad \pi_x P_{x,y} = \pi_y P_{y,x}$,

then π is the stationary dbn.

Pf: $\sum_{x \in \Omega} \pi_x P_{x,y} = \sum_{x \in \Omega} \pi_y P_{y,x} = \pi_y \underbrace{\sum_{x \in \Omega} P_{y,x}}_{1} = \pi_y \Rightarrow \pi$ is the stationary dbn.

To verify that the above MC has π as its stationary dbn., notice that if $\pi_x < \pi_y$,

$$P_{x,y} = \frac{1}{M} \quad \& \quad P_{y,x} = \frac{1}{M} \cdot \frac{\pi_x}{\pi_y}$$

$$\Rightarrow \pi_x P_{x,y} = \pi_y P_{y,x}$$

Now, we are interested in using our estimate of $Z(\lambda)$ to approximate m_n, the no. of perfect matchings.

i.e, given algo. for (ϵ, δ)-approx. of $Z(\lambda) \equiv \hat{Z}(\lambda)$,

approximate m_n.

- sample M from the Gibbs dbn.
- count the frac. of matchings that are perfect matchings (say, X)

$$\mathbb{E}(X) = \frac{m_n \lambda^n}{Z(\lambda)}$$

Estimate of m_n, $Y = \frac{X \cdot \hat{Z}(\lambda)}{\lambda^n}$

Lemma: For a graph G w/ no. of matchings of size k being m_k,

$\forall k$, $m_{k-1} \cdot m_{k+1} \leq m_k^2$ (log-concavity)

$$\Rightarrow \frac{m_{k-1}}{m_k} \leq \frac{m_k}{m_{k+1}} \leq \cdots \leq \frac{m_{n-1}}{m_n}$$

If this is not too large, we get good bounds since the MC runs in poly(n) time

$$\frac{m_k}{m_n} = \frac{m_k}{m_{k+1}} \cdot \frac{m_{k+1}}{m_{k+2}} \cdot \cdots \cdot \frac{m_{n-1}}{m_n} \leq \left(\frac{m_{k-1}}{m_n}\right)^{n-k}$$

$$\mathbb{E}(X) = \frac{\lambda^n}{\sum_k \left(\frac{m_k}{m_n}\right)\lambda^k} \geq \frac{\lambda^{n-k}}{(n+1)\left(\frac{m_{n-1}}{m_n}\right)^{n-k}}$$

$$\lambda = \frac{m_{n-1}}{m_n} \Rightarrow \mathbb{E}(X) \geq \frac{1}{\sum_k \left(\frac{m_k}{m_n}\right)\cdot\left(\frac{m_{n-1}}{m_n}\right)^k} \leq (n+1)\cdot\lambda^k$$

$$\Rightarrow \mathbb{E}(X) \geq \frac{1}{n+1}$$

27/10/23

Mixing times of Markov chains

Total variation distance of dbns. D_1 & D_2 supported on Ω,

$$\|D_1 - D_2\| = \frac{1}{2}\sum_{x \in \Omega} |D_1(x) - D_2(x)|$$

$$= \max_{A \subseteq \Omega} |D_1(A) - D_2(A)| \quad \text{(can be shown)}$$

$$\downarrow$$

$$\sum_{x \in A} D_1(x)$$

Let π be the stationary dbn. of an MC M.

P_x^t = Dbn. over Ω after t steps of M starting from x

$\Delta_x^{\epsilon}(t) = \|P_x^t - \pi\|$

$\Delta(t) = \max_{x \in \Omega} \Delta_x(t)$

Mixing time for a state $x \in \Omega$,

$$T_x(\epsilon) = \min\{t \mid \Delta_x(t) \leq \epsilon\}$$

Mixing time of M, $T(\epsilon) = \max_{x \in \Omega} T_x(\epsilon)$ ⟶ Bounding this helps bound our algo.'s running time

M is rapidly mixing if $T(\epsilon)$ is poly$(\ln(1/\epsilon), n)$
↳ I/p size, depends on context, not necess-arily $|\Omega|$

Coupling of Markov chains

A coupling of an MC M_t on the state space Ω is a MC Z_t on the state space $\Omega \times \Omega$ s.t.
↳ (X_t, Y_t)

$$\Pr(X_{t+1} = x' \mid Z_t = (x,y)) = \Pr(M_{t+1} = x' \mid M_t = x)$$

& $$\Pr(Y_{t+1} = y' \mid Z_t = (x,y)) = \Pr(M_{t+1} = y' \mid M_t = y)$$

A simple coupling could be just 2 indep. copies of an MC.

* Shuffling of cards

M_t : Pick $i \in_r \{1, \ldots, n\}$, move the card in pos^n i to the top

Coupling (X_t, Y_t)

X_t : M_t Let C be the card that was chosen

Y_t : Move the card C to the top

$X_{t+1} \sim M_{t+1}$ is clearly satisfied

$Y_{t+1} \sim M_{t+1}$ is because C is chosen u.a.r.
The fact that the choice was done by X_t doesn't affect the probas.

Notice that,
In expectation, after $n\log n + \Theta(n)$ steps, since this is just the coupon collector's problem.

Coupling lemma

Let $Z_t = (X_t, Y_t)$ be a coupling of an MC M_t on Ω.

Suppose that $\exists T$ s.t. $\forall x, y \in \Omega$,

$$\Pr(X_T \neq Y_T \mid X_0 = x, Y_0 = y) \leq \epsilon$$

Then, $\underbrace{\tau(\epsilon)}_{\text{Mixing time for } M_t} \leq T$.

Pf: Let Y_0 be chosen acc. to the stationary dbn. π of M_t.

Let $A \subseteq \Omega$. Given T, ϵ,

$$\begin{aligned}
\Pr(X_T \in A) &\geq \Pr((X_T = Y_T) \cap (Y_T \in A)) \\
&\geq 1 - \Pr(X_T \neq Y_T) - \Pr(Y_T \notin A) \quad \text{[union bound]} \\
&= \Pr(Y_T \in A) - \underbrace{\Pr(X_T \neq Y_T)}_{\leq \epsilon} \\
&\geq \pi(A) - \epsilon
\end{aligned}$$

Replacing A w/ $\Omega \setminus A$,

$$\Pr(X_T \in \Omega \setminus A) \geq \pi(\Omega \setminus A) - \epsilon$$

$$1 - \Pr(X_T \in A) \geq 1 - \pi(A) - \epsilon \implies \Pr(X_T \in A) \leq \pi(A) + \epsilon$$

$$\implies \pi(A) - \epsilon \leq \Pr(X_T \in A) \leq \pi(A) + \epsilon \implies \tau(\epsilon) \leq \epsilon$$

Independent sets of a fixed size k

$k \leq \frac{n}{3\Delta + 3}$, where $n = |V|$, Δ = max. deg.

X_i – indep. set, $|X_i| = k \longrightarrow |X_{i+1}| = k$

$v \in_r X_i$, $w \in_r V$

If $w \notin X_i$ & $X_i \setminus \{v\} \cup \{w\}$ is an indep. set, $X_{i+1} = X_i \setminus \{v\} \cup \{w\}$

Else, $X_{i+1} = X_i$

The MC is aperiodic ($w = v$)

It is irreducible because there are enough vertices outside X (src.) & Y (dest.) to make appropriate transitions.

$$|X \cup Y| \leq 2k$$

$$\implies |X \cup Y \cup N(X \cup Y)| \leq 2k(\Delta + 1) \leq 2n/3$$

Hence, it has a stationary dbn

30/10/23 CS6170

Coupling

A coupling of an MC M_t on the state space Ω is an MC $Z_t = (X_t, Y_t)$ over $\Omega \times \Omega$ s.t.

$$\Pr(X_{t+1} = x' \mid Z_t = (x,y)) = \Pr(M_{t+1} = x' \mid M_t = x)$$
$$\& \Pr(Y_{t+1} = y' \mid Z_t = (x,y)) = \Pr(M_{t+1} = y' \mid M_t = y)$$

Coupling lemma

Let $Z_t = (X_t, Y_t)$ be a coupling of an MC M_t on Ω.

Suppose $\exists T$ s.t. $\forall x, y \in \Omega$,

$$\Pr(X_T \neq Y_T \mid X_0 = x, Y_0 = y) \leq \epsilon.$$

Then, $\tau(\epsilon) \leq T$.

Independent sets of size k

Max. deg. $\leq \Delta$, $k \leq \frac{n}{3(\Delta+1)}$

MC: X_i (ind. set of size k)

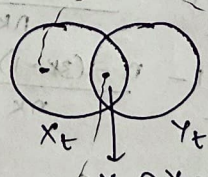
- choose $v \in_r X_i$, $w \in_r V$
- if $X_i \setminus \{v\} \cup \{w\}$ is an ind. set,

$$X_{i+1} = X_i \setminus \{v\} \cup \{w\}$$

Else, $X_{i+1} = X_i$

This is aperiodic & irreducible. We want to show it mixes well.

coupling (X_t, Y_t)



X_t: $v \in_r X_t$, $w \in_r V$
& same as M_t

Y_t: if $v \in Y_t$ (same w as X_t)
move acc. to M_t
~~if $v \in X_t$~~
Else, $v' \in_r Y_t \setminus X_t$, move acc. to v', w

We want a bound on the no. of steps before $X_t = Y_t$,

i.e, $d_t = |X_t - Y_t| = 0$

Notice that $d_{t+1} \in \{d_t - 1, d_t, d_t + 1\}$ [Think about this again]

$\Rightarrow$ To bound: $\Pr(d_{t+1} = d_t^{+1} \mid d_t > 0) \leq ?$

$\Pr(d_{t+1} = d_t - 1 \mid d_t > 0) \geq ?$

when is $d_{t+1} = d_t + 1$? Something must be removed from $X_t \cap Y_t$

i.e., $v \in X_t \cap Y_t$

$\& \; w \in (X_t \setminus Y_t) \cup (Y_t \setminus X_t)$

$\cup N(X_t \setminus Y_t) \cup N(Y_t \setminus X_t)$

$\Pr(d_{t+1} = d_t + 1 \mid d_t > 0)$

$\leq \underbrace{\frac{k - d_t}{k}}_{v \in X_t \cap Y_t} \cdot \frac{2d_t(\Delta+1)}{n}$

when is $d_{t+1} = d_t - 1$? Something must be added to $X_t \cap Y_t$

i.e., $v \in X_t \setminus Y_t$, $v' \in Y_t \setminus X_t$

$\& \; w \in X_{t+1} \cap Y_{t+1}$

$\Rightarrow$ It is sufficient that

$w \notin (X_t \cup Y_t) \cup N(X_t \cup Y_t)$

$(d_t + k)$ size

$\Pr(d_{t+1} = d_t - 1 \mid d_t > 0)$

$\geq \frac{d_t}{k} \cdot \frac{n - (d_t + k)(\Delta+1)}{n}$

$$\begin{aligned} E(d_{t+1} \mid d_t) &= d_t \cdot \Pr(d_{t+1} = d_t) \\ &\quad + (d_t + 1) \cdot \Pr(d_{t+1} = d_t + 1) \quad + (d_t - 1) \cdot \Pr(d_{t+1} = d_t - 1) \\ &= d_t + \Pr(d_{t+1} = d_t + 1) - \Pr(d_{t+1} = d_t - 1) \\ &\leq d_t \left[1 + \frac{k - d_t}{k} \cdot \frac{2(\Delta+1)}{n} + \frac{n - (d_t + k)(\Delta+1)}{nk} \right] \\ &\leq d_t \left[1 - \frac{n - (3k-1)(\Delta+1)}{nk} \right] \quad \text{(verify)} \end{aligned}$$

$$\Rightarrow E(d_{t+1}) \leq E(d_t) \left[1 - \frac{n - (3k-1)(\Delta+1)}{nk} \right]$$

$$E(d_{t+1}) \leq \underbrace{E(d_0)}_{\leq k} \left[1 - \frac{n - (3k-1)(\Delta+1)}{nk} \right]^{t+1}$$

$\Pr(d_t \geq 1) \leq \underbrace{E(d_t)}_{\text{verify that we get } t = \text{poly}(n, \ln(1/\epsilon))} \leq \epsilon$

Vertex coloring

$c : V \to \zeta$ Goal: color vertices s.t. adj. vertices get diff. colors

max. deg. $= \Delta$

No. of colors $\geq 4\Delta + 1$

Task is to sample colorings

MC M_t : Choose $v \in_r V$, color ℓ u.a.r.,
try to color v with ℓ

Picking the same color allows aperiodicity.

Ex: Show that it is irreducible.

Each state has $|V| \cdot |C|$ out-edges $\Rightarrow$ uniform stationary dbn.

Coupling (X_t, Y_t)

X_t & Y_t will behave exactly the same, i.e, same v & ℓ

d_t = No. of vertices w diff. colors

Once again, $d_{t+1} \in \{d_t - 1, d_t, d_t + 1\}$

$$\Pr(d_{t+1} = d_t - 1 \mid d_t > 0) \geq \underbrace{\frac{d_t}{n}}_{\substack{v \text{ should be colored differently, i.e, it is one of the } d_t}} \cdot \underbrace{\frac{C - 2\Delta}{C}}_{\substack{\text{New color must not be used in its nbrs.} \Rightarrow 2\Delta \text{ may be used}}}$$

$$\Pr(d_{t+1} = d_t + 1 \mid d_t > 0) \leq \underbrace{\frac{n - d_t}{n}}_{\substack{\text{Must be colored the same}}} \cdot \underbrace{\frac{2\Delta}{C}}_{\substack{\downarrow \\ \text{color should be in one of the two colorings}}}$$

We can now bound the expectation & consequently, the no. of steps, using a similar analysis as for ind. sets.

1/11/23

Canonical paths

MC M over Ω = irreducible, aperiodic

s.t. $\forall x, y$ $\pi(x) P_{x,y} = \pi(y) P_{y,x} = Q(x,y)$

π ↓ stationary dbn.

Graph $\tilde{G}$: $V = \Omega$, $E = \{(x,y) \mid Q(x,y) > 0\}$ (undirected)

$\gamma_{x,y}$ = Canonical path from x to y in $\tilde{G}$.

$\Gamma = \{\gamma_{x,y} \mid x, y \in \Omega\}$

Load for an edge $e \in \tilde{G}$,

$$\rho(e) = \frac{1}{Q(e)} \sum_{(x,y) : e \in \gamma_{x,y}} \pi(x)\,\pi(y)\,|\gamma_{x,y}|$$

↳ len. of canonical path