Lecture 13

- finish saving random bits via random walks

- Linearity testing intro

From last time:

Linear Algebra Review

def v is an eigenvector of A with corresponding eigenvalue λ iff

$$vA = \lambda v$$

def ℓ_2-norm of $v = (v_1 \cdots v_n) = \sqrt{\sum_{i=1}^{n} v_i^2} = v \cdot v$

def $v^{(1)} \dots v^{(m)}$ orthonormal if

$$\underbrace{v^{(i)} \cdot v^{(j)}}_{\text{inner product}} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases}$$

$$= \sum_{\ell} v_{\ell}^{(i)} \cdot v_{\ell}^{(j)}$$

Thm Transition matrix P real + symmetric

$\Rightarrow$ $\exists$ e-vecs $v^{(1)} \dots v^{(n)}$

forming orthonormal basis with corresponding e-values $1 = \lambda_1 \geq |\lambda_2| \geq \dots \geq |\lambda_n|$

$+$ $v^{(1)} = \frac{1}{\sqrt{n}} (1 \dots 1)$

chosen so that $\|v^{(1)}\|_2 = 1$

$\Rightarrow$ any vector w is expressible as linear combination of $v^{(i)}$'s

$$w = \sum \alpha_i v^{(i)}$$

$+$ L_2-norm of w is $\sqrt{\sum \alpha_i^2}$ (*)

From last time:

Useful Facts:

Assume P has all positive entries & e-vecs $v^{(1)} \dots v^{(n)}$ with corresponding e-vals $\lambda_1 \cdots \lambda_n$

Facts

(1) αP has e-vecs $v^{(1)} \dots v^{(n)}$ with corresponding evals $\alpha\lambda_1, \dots, \alpha\lambda_n$

(2) $P+I$ " " " " " " $\lambda_1+1, \dots, \lambda_n+1$

(3) P^k " " " " " " $\lambda_1^k, \dots, \lambda_n^k$ ← Useful today

(4) P stochastic $\Rightarrow$ $|\lambda_i| \le 1 \quad \forall i$

Note: add self-loops; $\frac{P+I}{2}$ = "stay put with prob ½ & walk with prob ½"

$\Rightarrow$ new eigenvalues $\frac{\lambda_1+1}{2}, \dots, \frac{\lambda_n+1}{2}$

Thm P is transition matrix of undirected, non k-partite, d-reg connected graph

can put self loop on each node →

π_0 is start dist.

π is stationary dist $= (\frac{1}{n}, \dots, \frac{1}{n})$

(so $\pi P = \pi$)

Then $\|\pi_0 P^t - \pi\|_2 \le |\lambda_2|^t$

Reducing Randomness via Random Walks:

For language L,

let $\mathcal{A}$ be algorithm s.t.

(1) $\forall x \in L \quad \Pr_{\mathcal{A}\text{'s coins}}[\mathcal{A}(x)=1] \geq 99/100$ usually correct

(2) $\forall x \notin L \quad \Pr_{\mathcal{A}\text{'s coins}}[\mathcal{A}(x)=0] = 1$ alway correct

To get error $< 2^{-k}$

Method	# random bits used
1) run k times & output "$x \notin L$" if see 0 else output "$x \in L$"	$k \cdot r$
2) use pairwise ind random bits	$O(k+r)$
3) today: use random walks to choose bits	$r + O(k)$

The graph G: $\leftarrow$ we get to pick G!!!

- constant degree d-regular, connected, nonbipartite

- transition matrix P for r.w. on G
  has $|\lambda_2| \le \frac{1}{10}$

  d-reg $\Rightarrow$ stat dist π is uniform

- # nodes $= 2^r$ corresponds to all possible choices of r random bits

The Algorithm

 # random bits

- Pick random start node $w \in \{0,1\}^r$ — r
- Repeat K times:

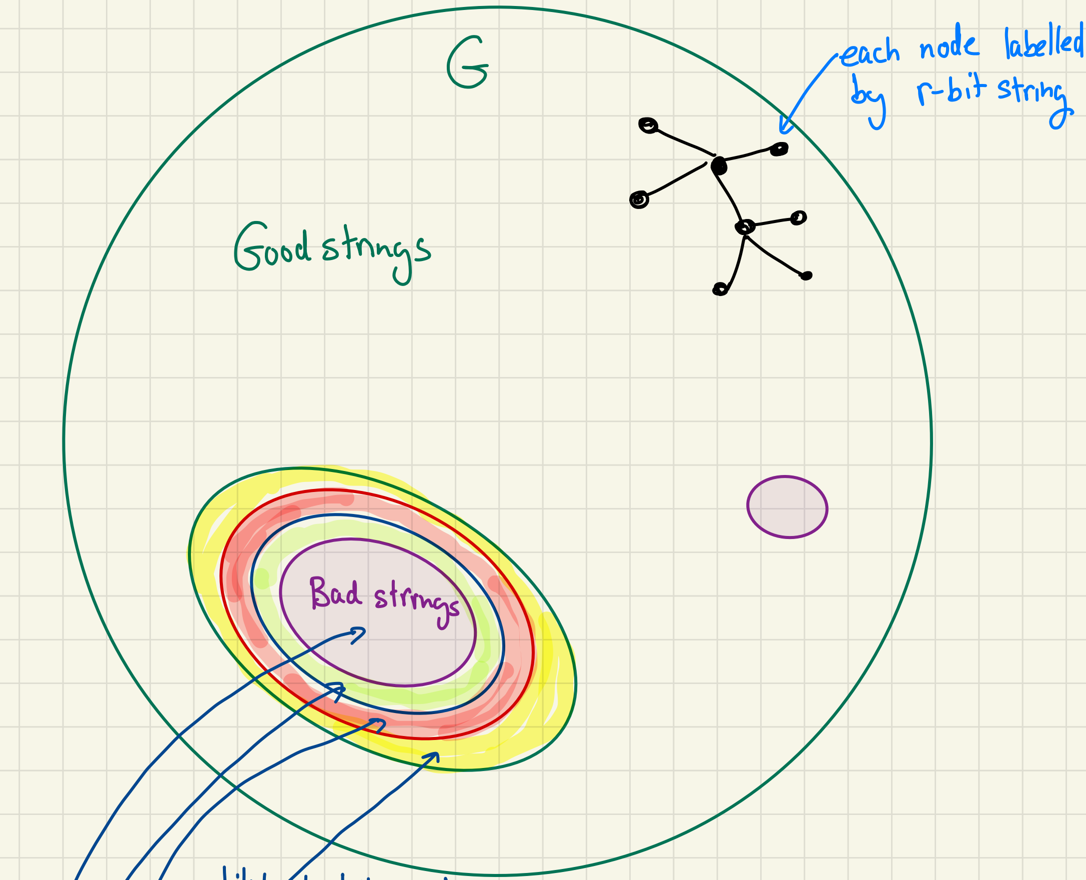
  $w \leftarrow$ random nbr of w — $O(1) \times k$ (d is const; # loops)

  run $\mathcal{A}(x)$ with w as random bits.

  If $\mathcal{A}(x)$ outputs "$x \in L$", output "$x \in L$" & halt
  else continue
- Output "$x \notin L$"

total: $r + O(k)$

Behavior: Claim: error of new algorithm is $\leq \left(\frac{1}{5}\right)^k$ for $x \in L$
(still 0 error for $x \notin L$)



likely to hit good string after 1 step

after 2 steps

after 3 steps

⋮

after k steps ← fewer & fewer of these as k gets bigger

if $|\lambda_2| \ll 1$ then

Main Idea

unlikely to pick start location that is bad after k-steps

bad case: walk only on "bad strings" & never reach good strings

why is this possible if G arbitrary? e.g. line

↑ λ_2 is close to 1

Proof of Claim

$x \notin L$: algorithm never errs (no bad strings)

$x \in L$:

most random bits say $x \in L$: $\geq \frac{99}{100} \cdot 2^r$

define $B = \left\{ w \mid A(x) \text{ with random bits } w \text{ is incorrect. i.e. says } x \notin L \right\}$

"bad w's"

$|B| \leq \frac{2^r}{100}$

need lin. alg. way of describing walks that stay in bad set:

define N diagonal matrix

$$N_w = \begin{cases} 1 & \text{if } w \in B \quad \leftarrow \text{incorrect} \\ 0 & \text{o.w.} \quad \leftarrow \text{correct} \end{cases}$$

Bad w's

$$N = \begin{pmatrix} 1 & & & & \\ & 1 & & O & \\ & & 0\,0\,0\,1\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0 & & \\ & O & & & \\ & & & & 1 \end{pmatrix}$$

For q <u>any</u> probability dist:

$q \cdot N$ is??

example:

$q = \left(\tfrac{1}{4} \;\; \tfrac{3}{4}\right) \qquad N = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$

$q \cdot N = \left(\tfrac{1}{4} \;\; 0\right)$

this 0-entry zeroed out the 3/4

$q \cdot N$ deletes weight that finds a witness to $x \in L$

$$\|q \cdot N\|_1 = \Pr_{w \in q}[\, w \text{ is bad} \,]$$

Can compose:

$$\| q \cdot PN \|_1 = \Pr_{w \in q}[\text{start at } q\text{, take a step \& land on "bad"}]$$

$\vdots$

$$\| q (PN)^i \|_1 = \Pr_{w \in q}[\text{start at } q\text{, take } i \text{ steps \& each is "bad"}]$$

ignores whether start node bad. this just hurts us, so ok to ignore.

<u>Lemma</u> $\forall \pi \quad \| \pi P N \|_2 \le \frac{1}{5} \| \pi \|_2$

First: how do we use lemma?

answer incorrect only if always see bad w's

$$\Rightarrow \Pr[\text{incorrect}] \le \| p_0 (PN)^k \|_1$$

$$\le \sqrt{2^r} \, \| p_0 (PN)^k \|_2$$

since $\|p\|_1 \le \sqrt{\text{domain size}} \cdot \|p\|_2$

$$\leq \cancel{\sqrt{2^r}}\ \underbrace{\|p_0\|_2}_{= \frac{1}{\cancel{\sqrt{2^r}}}} \left(\frac{1}{5}\right)^k$$

apply lemma k times

since start at uniform + L_2 norm of uniform $= \sqrt{\sum \left(\frac{1}{2^r}\right)^2} = \sqrt{\frac{1}{2^r}}$

$$= \left(\frac{1}{5}\right)^k$$

Proof of lemma:

let $V_1 \cdots V_{2^r}$ be e-vecs of P

+ V_1 is s.t. $\|V_1\|_2 = 1$ (so $V_1 = \left(\frac{1}{\sqrt{2^r}}, \dots, \frac{1}{\sqrt{2^r}}\right)$)

then $\pi = \sum_{i=1}^{2^r} \alpha_i V_i$

note: 1) $\|\pi\|_2 = \sqrt{\alpha_i^2}$ by (*) proved previously

2) $\forall\, w \quad \|wN\|_2 = \sqrt{\sum_{i \in B} w_i^2} \leq \sqrt{\sum_i w_i^2} = \|w\|_2$

So:

$$\|\pi P N\|_2 = \left\| \sum_{i=1}^{2^n} \alpha_i v_i P N \right\|_2$$ since any π is lin comb of basis vectors

$$= \left\| \sum_{i=1}^{2^n} \alpha_i \lambda_i v_i N \right\|_2$$

$$\leq \underbrace{\| \alpha_1 \lambda_1 v_1 N \|_2}_{A} + \underbrace{\left\| \sum_{i=2}^{2^n} \alpha_i \lambda_i v_i N \right\|_2}_{B}$$ Cauchy-Schwarz

bound A:

$$\| \alpha_1 \lambda_1 v_1 N \|_2 = \| \alpha_1 v_1 N \|_2$$ since $\lambda_1 = 1$

$$= |\alpha_1| \cdot \sqrt{\sum_{i \in B} \left(\frac{1}{\sqrt{2^r}}\right)^2}$$ since $v_1 = \left(\frac{1}{\sqrt{2^r}}, \dots, \frac{1}{\sqrt{2^r}}\right)$ & $N = \begin{pmatrix} 0 0 \ddots & 0 \\ 0 & 0 0 \ddots 0 0 \end{pmatrix}$

$$= |\alpha_1| \cdot \sqrt{\frac{|B|}{2^r}}$$

uses that uniform dist is unlikely to be on a bad string

$$\leq \frac{|\alpha_1|}{10}$$ since $\frac{|B|}{2^r} \leq \frac{1}{100}$

$$\leq \frac{\|\pi\|_2}{10}$$ since $\|\pi\|_2 = \sqrt{\sum_{i=1} \alpha_i^2}$

bound Ⓑ:

$$\left\| \sum_{i=2}^{2^n} \alpha_i \lambda_i v_i N \right\|_2 \leq \left\| \sum_{i=2}^{2^n} \alpha_i \lambda_i v_i \right\|_2$$ from note

$$= \sqrt{\sum (\alpha_i \lambda_i)^2}$$ (*)

$$\leq \sqrt{\sum \alpha_i^2 \cdot \left(\frac{1}{10}\right)^2}$$ $\lambda_i \leq 1/10$

$$\leq \frac{1}{10} \cdot \|\pi\|_2$$ (*)

uses "mixing" of v_i's for $i > 2$.

(v_i could have lots of weight in bad areas, but "expansion" of graph causes it to step out of bad area)

So: $\|\pi P N\|_2 \leq \frac{\|\pi\|_2}{5}$ ∎

New topic

Linearity Testing

$$f: G \to \cancel{G}\ H$$

G is finite group

H " " "

def. f is "linear" (homomorphism) if

$$\forall\ x, y \in G \qquad f(x) +_H f(y) = f(x +_G y)$$

$+_H$ is "plus" in group H

$+_G$ is "plus" in group G

e.g. $f(x) = x$

$f(x) = ax \bmod p$ for $G = \mathbb{Z}_p$

$f_{\bar{a}}(x) = \sum a_i x_i \bmod 2$

def f is "ε-linear" if $\exists$ linear g s.t. f + g agree on $\geq 1-\varepsilon$ fraction of inputs.

Notation note that the following are equivalent statements:

- f & g agree on $\geq 1-\varepsilon$ fraction of inputs
- $$\frac{|\{x \mid f(x)=g(x),\ x\in G\}|}{|G|} \geq 1-\varepsilon$$
- $$\Pr_{x\in G}[f(x)=g(x)] \geq 1-\varepsilon$$

How hard is it to test linearity?

do we need to try all $x, y, x+y$ tuples?

if domain is size n, this requires n^2 tests of $f(x)+f(y)=f(x+y)$

Proposed test: Pick random x, y
Test $f(x)+f(y)=f(x+y)$

repeat how many times?