# Reducing Randomness Via Random Walks on special graphs

eigen(

# Reducing Randomness

For decision problem $L$,

Let $A$ be algorithm s.t. 1) $\forall x \in L$ $\Pr[A(x)=1] \geq 99/100$ almost always correct

2) $\forall x \notin L$ $\Pr[A(x)=0] = 1$ always correct

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

**Method:** | **# random bits used**
---|---
1) run $k$ times & output "$x \notin L$" if ever see "$x \notin L$" else output "$x \in L$" | $O(kr)$
2) use p.i. random bits | $O(k+r)$
3) today: use random walk on graph to choose random bits | $r + O(k)$

## Plan:

- associate all (random) strings in $\{0,1\}^n$ with nodes of a graph $G$
- problem of picking a random string is now equivalent to problem of picking a random node ← easier

  picking several random strings $\Rightarrow$ picking several nodes ← easier

  picking several strings, one of which is "good" $\Rightarrow$ picking several nodes one of which is "good" ← "easier"!

## The graph $G$:

- constant degree $d$-regular, connected, nonbipartite
- transition matrix $P$ for r.w. on $G$ has $|\lambda_2| \le \frac{1}{10}$
  $\pi$ uniform since $d$-reg
- $\#$nodes $= 2^r$ $\sim$ $r$ random bits

## The Algorithm:

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

  $w \Leftarrow$ random neighbor of $w$ — $O(1)$ bits $\times K$

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

  if $\mathcal{A}$ outputs "$x \in L$" then output "$x \in L$" + halt

  else continue

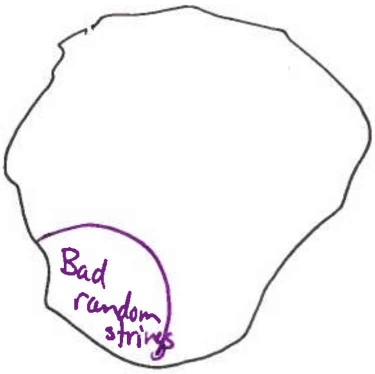
- Output "$x \notin L$"

total random bits: $r + O(K)$

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

## Behavior :

**idea :**




bad case - walk only on "bad" random strings
& never get out to "good" random strings

why would this <u>not</u> work on arbitrary $G$ ?
e.g. $G$ = line

if $x \notin L$ : algorithm <u>never</u> errs (there are no bad strings)

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

define $B \leftarrow \{ w \mid A(x)$ with random bits $w$ is incorrect
i.e. says $x \notin L \}$
" Bad w's "

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

Want linear algebraic way of describing walks that stay in badset :
define $N$ diagonal matrix such that

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

Bad w's

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

$q$ any probability distribution

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

i.e. $pN$ deletes weight that finds a witness to $x \in L$

Can compose:

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

$\vdots$

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

ignores whether start node is bad, this jus hurts us so it is ok to ignore

**Lemma** $\forall \pi \quad \|\pi PN\|_2 \le \frac{1}{5}\|\pi\|_2$

First: How do we use the lemma?

If always see bad w's, then answer incorrect

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

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

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

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

apply lemma k times

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

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

**Proof of lemma** let $V_1 \dots V_{2^r}$ be e-vects of $P$, + $V_1$ is s.t. $\|V_1\|_2 = 1$

note, $V_1 = \left(\frac{1}{\sqrt{2^r}}, \dots, \frac{1}{\sqrt{2^r}}\right)$

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

Note: 1) $\|\Pi\|_2 = \sqrt{\alpha_i^2}$ (from before)

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

So:

$$\|\Pi P N\|_2 = \left\| \sum_{i=1}^{2^r} \alpha_i v_i P N \right\|_2$$

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

$$\le \underbrace{\|\alpha_1 \lambda_1 v_1 N\|_2}_{A} + \underbrace{\left\| \sum_{i=2}^{2^r} \alpha_i \lambda_i V_i N \right\|_2}_{B} \quad \text{Cauchy-Schwarz}$$

bounding (A):

$$\|\alpha_1 \lambda_1 v_1 N\|_2 = \|\alpha_1 v_1 N\|_2 \quad \text{since } \lambda_1 = 1$$

$$= |\alpha_1| \sqrt{\sum_{i \in B} \left(\frac{1}{\sqrt{2^r}}\right)^2} \quad \text{since } V_1 = \left(\frac{1}{\sqrt{2^r}}, \dots, \frac{1}{\sqrt{2^r}}\right)$$

use that uniform is unlikely to be on bad string

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

$$\le \frac{|\alpha_1|}{10} \quad \text{since } \frac{|B|}{2^r} \le \frac{1}{100}$$

$$\le \frac{\|\Pi\|_2}{10} \quad \text{since } \|\Pi\|_2 = \sqrt{\Sigma \alpha_i^2}$$

Bounding (B):

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

use "mixing"

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

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

$$\leq \frac{1}{10} \|\pi\|_2$$

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