Lecture 5

- linear algebra & random walks
- randomized complexity classes
- derandomization via enumeration

Linear Algebra Review

← this will be useful for analyzing random walks!

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

$$vA = \lambda v$$

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

($v \cdot v$: inner product)

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

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

($v^{(i)} \cdot v^{(j)}$: inner product) $= \sum_{\ell} v_\ell^{(i)} \cdot v_\ell^{(j)}$

example P = transition matrix of d-reg undir graph (doubly stochastic)

ℓ_1-norm = 1

$$\left(\frac{1}{n} \cdots \frac{1}{n}\right) \cdot P = 1 \cdot \left(\frac{1}{n} \cdots \frac{1}{n}\right)$$

↖ doesn't this seem more natural? its the probability distribution vector

also $$\left(\frac{1}{\sqrt{n}} \cdots \frac{1}{\sqrt{n}}\right) \cdot P = 1 \cdot \left(\frac{1}{\sqrt{n}} \cdots \frac{1}{\sqrt{n}}\right)$$

ℓ_2-norm = 1 $\Rightarrow$ so this gets used a lot!

Important Theorem

Thm Transition matrix P real + symmetric

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

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

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

↖ chosen so that $\|v^{(1)}\|_2 = 1$

(won't prove here)

Useful Facts:

Assume P has all positive entries

$\exists$ 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 " " " " " " $\lambda_1^k, \dots, \lambda_n^k$

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

Why?

(1) $vP = \lambda v \iff v \cdot \alpha \cdot P = \lambda \cdot \alpha \cdot P$

(2) $v(P+I) = vP + vI = \lambda v + v = (\lambda+1)v$

Note: add self-loops: $\frac{P+I}{2}$ = "stay put with prob $\frac{1}{2}$ & walk with prob $\frac{1}{2}$"

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

(3) $v P^k = (vP)P^{k-1} = \lambda v P^{k-1} = \lambda^2 v P^{k-2} = \dots = \lambda^k v$

k-step walks

(4) $\forall i$, let $I = \{j \mid v_j^{(i)} > 0\}$

then $\lambda \sum_{j \in I} v_j^{(i)} = \sum_{j \in I} \sum_k v_k^{(i)} P_{kj}$ (computes j^{th} entry of $v \cdot P$)

$\leq \sum_{\substack{j,k \\ \text{s.t. } j,k \in I}} v_k^{(i)} P_{kj}$ (since entries of v not in I are ≤ 0 & P_{kj} is always ≥ 0)

$\leq \sum_{k \in I} v_k^{(i)} \underbrace{\sum_{j \in I} P_{ij}}_{\leq 1 \text{ since stochastic}} \leq \sum_{k \in I} v_k^{(i)}$

$\Rightarrow \lambda \leq 1$

Note if $v^{(1)} \dots v^{(n)}$ orthonormal basis then

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

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

& L_2-norm of w is $\sqrt{\sum \alpha_i^2}$

why? (see below)

L_2-norm Calculation:

$$\|w\|_2 = \sqrt{\left(\sum_i \alpha_i v^{(i)}\right) \cdot \left(\sum_j \alpha_j v^{(j)}\right)}$$

$$= \sqrt{\sum_{i,j} \alpha_i \alpha_j \underbrace{v^{(i)} v^{(j)}}_{\begin{array}{l}=0 \text{ if } i \neq j \\ =1 \text{ if } i = j\end{array}}}$$

$$= \sqrt{\sum_i \alpha_i^2}$$

Call this: (*) will use this soon

Recall:

Stationary distribution:

π s.t. $\pi = \pi P$ (so π is eigenvector with e-val = 1)

(taking more steps in r.w. keeps you in same distribution)

recall: P ergodic $\Rightarrow$ π exists & unique

(for graphs, can always take $\frac{P+I}{2}$)

Mixing Times

How long does it take to reach stationary distribution?

def. $\varepsilon > 0$

Mixing time, $T(\varepsilon)$, of M.C. A with stationary dist π is min t s.t.

$\forall\ \pi^{(0)}$, $\|\pi - \pi^{(0)} A^t\|_1 < \varepsilon$

↑ any start distribution

def. M.C. A is rapidly mixing if

$$T(\varepsilon) = \text{poly}(\log n, \log 1/\varepsilon)$$

↑ # states

examples: r.w. on complete graph, random graph

note that mixing time of $\frac{P+I}{2}$ is at most 2x more

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 $\pi P = P$)

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

exponentially decreasing dist if $1-\lambda_2$ is const!!

$\Rightarrow$ rapid mixing

Proof

P real, symmetric $\Rightarrow$

$\exists$ evecs $v^{(1)} \dots v^{(n)}$ are orthonormal basis with e-vals $1 = \lambda_1 \geq |\lambda_2| \geq \dots \geq |\lambda_n|$

$\forall\ v^{(1)} = \frac{1}{\sqrt{n}}(1, 1, \dots, 1)$ (**)

so any distribution vector, in particular π_0, can be expressed as lin comb of $v^{(i)}$'s:

$$\pi_0 = \sum_{i=1}^{n} \alpha_i v^{(i)} \qquad (***)$$

so $\pi_0 P^t = \sum_{i=1}^{n} \alpha_i \underbrace{v^{(i)} \cdot P^t}_{= \lambda_i^t v^{(i)}}$

main idea: for $i \geq 2$, λ_i^t is very small!

$$= \underbrace{\alpha_1}_{=\frac{1}{\sqrt{n}}} \underbrace{\lambda_1^t}_{=1} v^{(1)} + \alpha_2 \lambda_2^t v^{(2)} + \dots$$

what is α_1? what is $\alpha_1 \lambda_1^t v^{(1)}$?

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

Using → (***)

$$\pi_0 \cdot v^{(1)} = \alpha_1 \underbrace{v^{(1)} \cdot v^{(1)}}_{=1} + \sum_{i=2}^{n} \alpha_i \underbrace{v^{(i)} \cdot v^{(1)}}_{=0} = \alpha_1$$

Using (**) + that π_0 is a distribution →

$$\text{also, } \pi_0 \cdot v^{(1)} = \pi_0 \cdot \frac{1}{\sqrt{n}}(1 1 \dots 1) = \frac{1}{\sqrt{n}} \underbrace{\pi_0 \cdot (1 1 \dots 1)}_{=1}$$

$$\text{so } \alpha_1 = \frac{1}{\sqrt{n}} \quad + \quad \alpha_1 \lambda_1^t v^{(1)} = \left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right) = \pi$$

note that this argument does not use any knowledge of π_0, other than it is a distribution.

Continuing... π

$$\|\pi_0 P^t - \alpha_1 \cdot v^{(1)}\|_2 = \left\| \sum_{i=2}^{n} \alpha_i \lambda_i^t v^{(i)} \right\|_2$$

$$= \sqrt{\sum_{i=2}^{n} \alpha_i^2 \lambda_i^{2t}} \qquad \text{by } (*)$$

$$\leq |\lambda_2|^t \sqrt{\sum_{i=2}^{n} \alpha_i^2} \qquad \text{since } |\lambda_2| \geq |\lambda_3| \geq \dots$$

$$\leq |\lambda_2|^t \, \|\pi_0\|_2 \qquad \text{by } (*) \text{ + since } \|\pi_0\|_2 = \sum_{i=1}^{n} \alpha_i^2 > \sum_{i=2}^{n} \alpha_i^2$$

$$\leq |\lambda_2|^t \qquad \text{since } \|w\|_2 \leq \|w\|_1 = 1 \text{ when entries} \leq 1$$

∎

We are going to use this to "save" randomness...

but first, some background on derandomization

Randomized Complexity Classes

def. language L is subset of $\{0,1\}^*$

e.g. $\{x \mid x$ is graph with Hamilton path$\}$

$\{x \mid x$ is collection of sets with proper 2-coloring$\}$

def P is class of languages L with polytime deterministic algorithm A

s.t. $x \in L \Rightarrow A(x)$ accepts

$x \notin L \Rightarrow A(x)$ rejects

def RP is class of languages L with

polytime probabilistic algorithm A

s.t. $x \in L \Rightarrow \Pr[A(x) \text{ accepts}] \geq 1/2$

$x \notin L \Rightarrow \Pr[A(x) \text{ accepts}] = 0$

} 1-sided error: if $A(x)$ = accept you know answer is correct

def BPP is class of languages L with

polytime probabilistic algorithm A

s.t. $x \in L \Rightarrow \Pr[A(x) \text{ accepts}] \geq 2/3$

$x \notin L \Rightarrow \Pr[A(x) \text{ accepts}] \leq 1/3$

} 2-sided error

Comments

- constants arbitrary

  with multiplicative overhead of $O(\log 1/\beta)$

  can get error $\leq \beta$

- Clearly $P \subseteq RP \subseteq BPP$

OPEN: is $P = BPP$?

Derandomization via Enumeration

Given: probabilistic algorithm A + input x

Algorithm:

Run A on **every** possible random string of length $r(n)$

($r(n)$: at most time bound of A. ⇑ is there a better bound?)

Output majority answer

Behavior:

if $x \in L$, $\geq 2/3$ of random strings cause A to accept

$\Rightarrow$ majority answer is accept

if $x \notin L$, $\geq 2/3$ of random strings cause A to reject

$\Rightarrow$ majority answer is reject

Runtime: $O(2^{r(n)} \cdot t(n)) \leq O(2^{t(n)} \cdot t(n))$

$t(n)$: time bound of A

note $r(n) \leq t(n)$ but if could get a better bound on $r(n)$, would improve runtime. e.g. $r'(n) = O(\log n)$ & $t(n) = \text{poly}(n)$

$\rightarrow$ total time is $\text{poly}(n)$

Corollary: $BPP \subseteq EXP$

$\| \|$

$DTIME(\bigcup_c 2^{n^c})$

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

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

Method

run k times & output "$x \notin L$" if see 0
else output "$x \in L$"

 # random bits used

$k \cdot r$

today: use random walks to choose bits $r + O(k)$

Plan

- $\forall$ (random) string in $\{0,1\}^n$, assign it to node in graph G

- picking random n-bit string
  $\Rightarrow$ picking random node in G

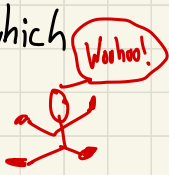
  easier?

picking several random n-bit strings
$\Rightarrow$ picking several random nodes in G

easier?

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

Easier!!



The graph G: ← we get to pick G!!!

- constant degree d-regular, connected, nonbipartite
- transition matrix P for r.w. on G has $|\lambda_2| \leq \frac{1}{10}$

  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

  run $A(x)$ with w as random bits.

  If $A(x)$ outputs "$x \in L$", output "$x \in L$" & halt
  else continue

  $O(1) \times k$ — d is const; # loops
- Output "$x \notin L$"

total: $r + O(k)$

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