Lecture 22: Random walks on graphs I: Markov Chains

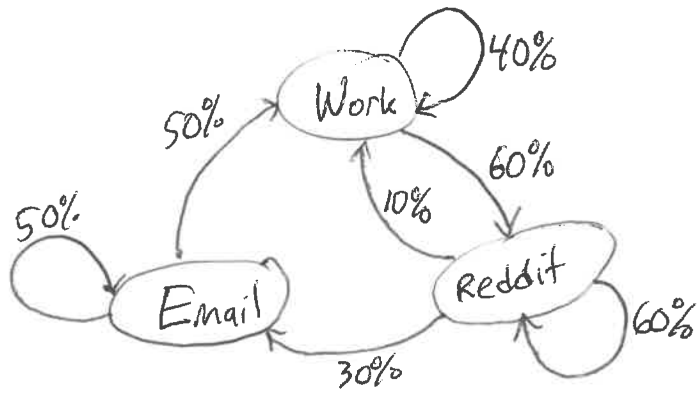
[[A day in the life of me....]]

9:00am

9:01am

9:02am

⋮



"Markov chain": • Directed graph – for simplicity, finite & strongly connected [any node can reach any other node]

• Self-loops OK

• Each edge labeled by positive probability

• At each node/"state", outgoing probabilities add to 1.

[[Life rule: always use linear algebra. Or, since we have a graph, make what's basically its adjacency matrix.]]

Def: Transition matrix K: $n \times n$ matrix (n = # states)

$K[i,j] = \Pr[j \to i \text{ in one step}]$. E.g.:

$$\begin{array}{c|ccc} & \text{Work} & \text{Reddit} & \text{Email} \\ \hline W & .4 & .1 & .5 \\ R & .6 & .6 & 0 \\ E & 0 & .3 & .5 \end{array}$$

[Why not $i \to j$? Yeah, this is a lifelong struggle, arising b/c it's super-traditional to do

$[\quad][\,] = [\,]$ and not

$[\,][\quad] = [\quad]$]

["stochastic matrix": cols sum to 1]

[[Given a Markov Chain, we always imagine the following process...]]

$X_0 :=$ .... // some state [[how this is chosen may vary]]

for $t = 1, 2, 3, \ldots$

$X_t :=$ pick i randomly with prob. $K[i, X_{t-1}]$

[["Markov property": state you're at @ time t only depends on where you were @ time $t-1$]]

[[Why matrices? Because...]]

$$\Pr[X_1 = j \mid X_0 = l] = K[j, l]$$

[[& $X_0 = l$, but don't need it]]

$$\Pr[X_2 = i \mid X_0 = l] = \sum_{j=1}^{n} \Pr[X_2 = i \mid X_1 = j] \cdot \Pr[X_1 = j \mid X_0 = l]$$

$$= \sum_{j=1}^{n} K[i,j] \cdot K[j,l] = (K \cdot K)[i,l]$$

def$^{\underline{n}}$ of matrix multiplication!

$$\begin{array}{cc} & l \\ i & \begin{bmatrix} K^2 \end{bmatrix} \end{array} = \begin{array}{c} \\ i \end{array}\begin{bmatrix} \quad K \quad \end{bmatrix} \begin{array}{c} l \\ \begin{bmatrix} K \end{bmatrix} \end{array} = K^2[i,l]$$

More generally, $\Pr[X_t = i \mid X_0 = j] = K^t[i,j]$

X_0 might be fixed, or we might also pick it randomly somehow.

E.g. $X_0 := \begin{cases} \text{Work} & \text{w.p. } 50\% \\ \text{Reddit} & \text{w.p. } 20\% \\ \text{Email} & \text{w.p. } 30\% \end{cases}$ $\qquad \pi_0 := \begin{bmatrix} .5 \\ .2 \\ .3 \end{bmatrix}$

typical notation for starting distribution vector

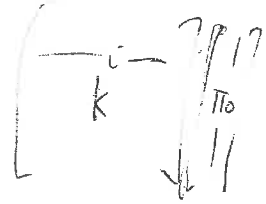
[[distribution vector: length n, nonneg. #'s adding to 1]]

Given π_0, what's $Pr[X_1 = i]$?

$$\sum_{j=1}^{\hat{}} \underbrace{Pr[X_0 = j]}_{\pi_0[j]} \cdot \underbrace{Pr[X_1 = i \mid X_0 = j]}_{K[i,j]}$$

$$= \sum_{j=1}^{\hat{}} K[i,j]\, \pi_0[j]$$

$$= (K\pi_0)[i]$$



So $K\pi_0$ is distrib. vector for X_1

& $K^t \pi_0$ " " " " X_t.

For my W/R/E chain, $K^{10}[i,j] = Pr[X_{10} = i \mid X_0 = j]$

$$K^{10} = \begin{array}{c} W \\ R \\ E \end{array} \begin{bmatrix} .2940 & .2942 & .2942 \\ .4413 & .4411 & .4413 \\ .2648 & .2648 & .2646 \end{bmatrix}$$

[[All columns more or less the same! Interpretation?
Where I am at time 10 hardly depends on the state I start in. After a while (10 mins), I'm pretty much 29% Work, 44% Reddit, 26% Email no matter what X_0 was.]]

The limiting column of K^t $(t \to \infty)$ (assuming limit exists)
is called <u>stationary</u> distribution π
(or "invariant")
[[if you start X_0 in this distrib., do 1 more step, you're still in this distrib]]

$$K \begin{pmatrix} \pi & \pi & \pi \end{pmatrix} = \begin{pmatrix} \pi & \pi & \pi \end{pmatrix}$$

$\Rightarrow$ $K\pi = \pi$: π is an <u>eigenvector</u> of <u>eigenvalue</u> 1.

[[can solve these eq^n's, plus $\sum_i \pi[i] = 1$, to get a precise π]]

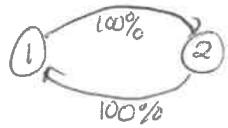
Fundamental Theorem: [[Given a finite, strongly connected Markov chain,]]

[[takes 1/2 an hour to prove; we'll skip]]

there is a unique π satisfying $K\pi = \pi$;
it's an invariant prob. distrib. with $\pi[i] > 0 \ \forall i$.

Also, it's the limiting col. of K^t as $t \to \infty$ unless
chain has stupid "periodicity"

(e.g.



$$K = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = K^3$$

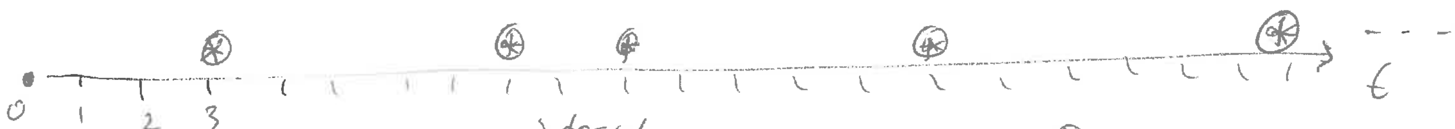
$$K^2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = K^4$$

K^t has no limiting row, but

$\pi = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$ is (unique) invariant distrib.)

[[Before seeing an application let's record a theoretical result.]]

Say you run a M.C. for a looong time, & "mark" every time you're at state "u".



⊛ = @ state u

In long interval of time, T, [[also long after time 0]],
for each particular $t \in T$, $\Pr[X_t = u] \underset{(\to)}{\approx} \pi[u]$.

$\therefore$ [[linearity of expectation/indicators trick]]

\# of times you're @ u $\approx \pi[u] \cdot |T|$.

$\therefore$ average spacing between ⊛'s is $\underset{\to}{\approx} \dfrac{|T|}{\pi[u]\,|T|} = \dfrac{1}{\pi[u]}$

[[adding about 20% more rigor, which I'm skipping]] $\Longrightarrow$

thm ("Mean 1st Recurrence Thm"):

In M.C. with invariant dist π,
"$M_{u,u}$" := E[# steps to hit u (again) when starting from u]
$= 1/\pi[u]$.

Application: PageRank ⟦the original algorithm used by Google to rank search results; first suggested by Lasker in '85 for ranking chess players; also used ca. 2013 for Twitter's "who-to-follow" suggestions.⟧

1997: ⟦when searching the web for "CMU", search engines returned pages ranked by how many occurrences of "CMU". Easy to spam, and also not very good ~~~~ anyway⟧

How to "rank" all web pages? ⟦you'll return the relevant pages, ordered by rank⟧

Idea: • web page is "important" if a lot of "important" pages link to it ⟦recursive!⟧

Say page has d outgoing links, to pages $p_1, \ldots, p_d$. Like it's voting, giving $1/d$ points to each. Let ~~K[p,q] be matrix where~~

K be matrix where $K[p,q]$ = fraction of page q's links pointing to p.

If $\pi[p]$ is "importance" of page p, we want

$$\pi[p] = \sum_q \pi[q] \cdot K[p,q] = (K\pi)[p] \quad !$$

$\rightarrow$ π is invariant distribution for the "random surfer"

(fraction of time a random surfer is at page p)

$\pi[p] =$

To combat spammy link cliques & also pages with 0 outgoing links, introduced "damping factor" α ($\approx .85$):

surfer follows random link w.p. α, goes to totally random page w.prob. $1-\alpha$ (or w.p. 100% if no out links)

⟦$\Rightarrow$ strong connectivity & no periodicity :)⟧

Q: How to compute π?

A: ⟦We studied this before, albeit for symmetric ~~~~ matrices...⟧

Solving sys of equations → bad! [[trillion × trillion system is no good]]

"Power method" = ~~simulate random surfer = few thousand steps~~

compute K^t and take (any) column

for some $t \approx 1000$s.

→ much faster!