Page Rank

25 September 2024 16:28

Discrete Markov Process

Discrete → discrete states of time → $t, t+\Delta t, t+2\Delta t \dots$

→ Series of experiments — regular time intervals — same set of possible outcomes

→ proceeds in steps (think: probability tree)

outcomes → "States" → Model can be in any one of these states.

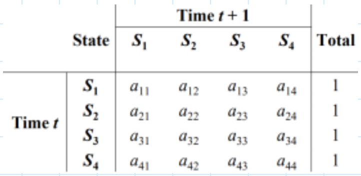
Movement b/w steps defined by probability { Next step → back to same state; Next step → Next state }

↓

Make stochastic transition probability matrix

Σa_{ij} for any col/row $= 1$

Also, all entries are non-negative.



	State	Time $t+1$ S_1	S_2	S_3	S_4	Total
Time t	S_1	a_{11}	a_{12}	a_{13}	a_{14}	1
	S_2	a_{21}	a_{22}	a_{23}	a_{24}	1
	S_3	a_{31}	a_{32}	a_{33}	a_{34}	1
	S_4	a_{41}	a_{42}	a_{43}	a_{44}	1

Eigen Decomposition Theorem:

"Square matrix can be decomposed to eigenvectors + eigenvalues."

$XC = \lambda X$ — eigenvector.

λ — eigenvalue] — no. of nonzero λ = rank(C) → no. of linearly independent rows/cols.

$XC = \lambda X$ — Eigenvector corresponding to largest $|\lambda|$

= principal right eigenvector

(used interchangeably w/ eigenvector)

also: column vector.

$y^T C = \lambda y^T$

y = left eigenvectors

= transpose cuz row vector

Also: $x_L C = \lambda x_L$

if x_L is steady state dist, $\lambda = 1$

Steady state + Stochastic Matrices:

probability of transitioning to a state approaches lim. as $t \to \infty$

- limiting values = stable probabilities
- system reaches = steady state.

steady state → eigenvector for stochastic matrix

Ergodic Markov chains:

any markov chain where it's possible to get from any A to any B in a finite no. of steps.

→ Conditions → Irreducibility = connected chain

→ Aperiodicity = transitions b/w states doesn't occur in a regular, repeating pattern.

→ Steady state theorem:

"for any ergodic MC, there is a unique steady state probability vector Π that is the principle left eigenvector of transition matrix P"

(no further change in probability vector)

$\Pi P = \lambda \Pi$

Random Walk + Probability Vectors.

→ MCs are abstractions of random walks.

$\bar{x}$ probability (row) vector tells us where we are

$$\begin{pmatrix} \underset{1}{0} & \underset{2}{0} & \underset{3}{0} \dots & \underset{i}{1} \dots & \underset{n}{0} \end{pmatrix}$$

1 → @ i^{th} position

$\Sigma x_i = 1$ (row stochastic)

walk is in state i w/ probability x_i (1 in above case)

If probability vector is $(x_1, x_2 \dots x_n)$ at this step, where in next step? Matrix P tells us.

from x, next step is distributed as xP

then xP^2, $xP^3 \dots$] Converges @ $xP = x$

from x, next step is distributed as xP
then xP^2, xP^3 ...] Converges @ $xP = x$ (steady state)

for any ergodic markov chain,
"long-term visit rate" for each state
= steady state probability distribution.

Over a long time,
we visit each state in proportion to this rate.

Steady state?
ie vector of probabilities.

PAGERANK IS A SUPER LONG VECTOR W/ ONE ENTRY/PAGE. ← $[P = (p_1, p_2 \dots p_N)$ → p_i = long term visit rate of site i

Pagerank

└ entire web = directed graph? Si.] — PR represents how impt. a page is on the web.

How do you model it? — [Page A links to B ⇒ one vote for page B.
votes ∝ importance

Eigenvector centrality
└ generalize degree centrality
by incorporating neighbors (undirected) } recursive.

X = popularity vector of all nodes
X_j = popularity of adj. nodes @ time t.

@ t=1, popularity depends on neighbors:

$$X_i(1) = \sum A_{ij} X_j(0) \quad \text{or} \quad X(1) = \sum A X(0)$$

$X(t) = A^t \cdot X(0)$, when t is large, it'll stabilize.
wkt $X(0) = \sum c_i v_i$ and $Av = \lambda v$
⇒ $A^t v = \lambda^t v$

$$\Rightarrow X(t) = A^t \sum c_i v_i$$

$$\Rightarrow X(t) = \sum \lambda_i^T c_i v_i$$

Let λ_1 be the principal eigenvalue.

Now make this more digraph friendly-

$$\Rightarrow \frac{X(t)}{\lambda_1^t} = \sum \left(\frac{\lambda_i}{\lambda_1}\right)^t c_i v_i$$

as $t \to \infty$, λ_1 is the largest, so
$X(t) = c_i v_i$
(assumed that adj matrix is symm)

for directed graphs
└ popularity passed by INCOMING edges.
└ so : $C_{katz}(v_i) = \alpha \sum A_{ji} C_{catz}(v_j) + \beta$

depends on neighborhood. (α)
initial popularity (β)

But there's a problem w/ Katz:
Everyone known by Mr. Popular is expected to be popular.

Rewrite in vector form:

$$C_{katz} = \beta(I - \alpha A^T)^{-1}$$

iterative version: $C(t) = \alpha A C(t-1) + \beta$

⟶ So divide by no. of outgoing links:

$$C_p = \alpha \sum A_{ji} \frac{C_p(v_j)}{d_j^{out}} + \beta$$

* pagerank + Katz centrality are variations of eigenvector centrality

Phew.

Phew.

Now let's look @ the underlying process.

Net-net, PR = model of user behavior.

1. Initially, every webpage chosen @ random
2. w/ probability α choose hyperlink (adjacent node)
3. w/ probability $(1-\alpha)$ perform random node for random walk.

Basic PR Update Rule:

① network of n nodes — all get same initial $PR = \frac{1}{n}$
② choose k no. of steps.
③ Perform k updates:
  - each page divides PR equally across outgoing links.
  - each page updates PR to be sum of shares it gets.
④ Constant PR, just moved around.
⑤ as $k \to \infty$, PR of all nodes converges.

Scaled PR update Rule:

① Apply basic PR
② Scale down all PR values by factor of s, when $s<1$.
  $\Rightarrow$ total PR has gone down from 1 to s.
③ Divide residual PR into n nodes as $(1-s)/n$.

Not done yet.

Why does PR converge?

Perron Frobenius Theorem:

for matrix M, non-negative entries, stochastic —

a) largest eigenvalue $c=1$, such that $c>|c'|$ for all other c'
b) 1 is eigenvalue w/ multiplicity 1.
c) for eigenvalue $=1$, eigenvector exists where $\Sigma x_i = 1$

most impt d) for any starting vector $x \neq 0$, sequence vector $M^k x$ converges to eigenvector x corresponding to largest eigenvalue $c=1$ as $k \to \infty$

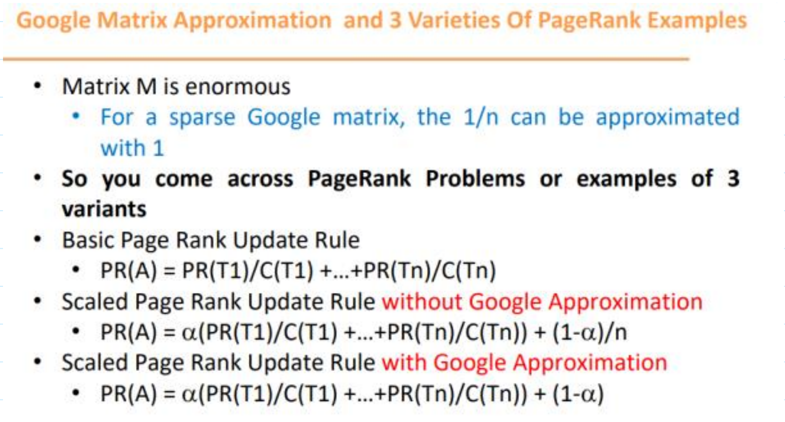
Limitations of early PR:

① Rank sinks or leaks — many incoming, no outgoing — selfish. no share.

② Hoarding | circular reference — group of pages only link within group.

③ Dangling nodes — random walk no work ☹

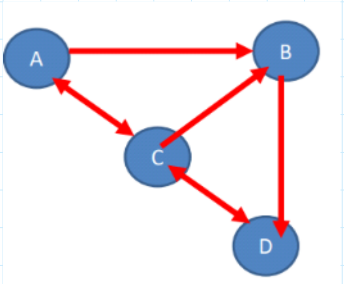
Solve by teleporting to a new random URL.



Google Matrix Approximation and 3 Varieties Of PageRank Examples

- Matrix M is enormous
  - For a sparse Google matrix, the 1/n can be approximated with 1
- **So you come across PageRank Problems or examples of 3 variants**
- Basic Page Rank Update Rule
  - PR(A) = PR(T1)/C(T1) +…+PR(Tn)/C(Tn)
- Scaled Page Rank Update Rule without Google Approximation
  - PR(A) = α(PR(T1)/C(T1) +…+PR(Tn)/C(Tn)) + (1-α)/n
- Scaled Page Rank Update Rule with Google Approximation
  - PR(A) = α(PR(T1)/C(T1) +…+PR(Tn)/C(Tn)) + (1-α)

Pagerank numerical:



	Iteration 0	Iteration 1
A	$1/4$	$1/12$
B	$1/4$	
C	$1/4$	
D	$1/4$	

Iteration 1:

$$PR(A) = \frac{PR(C)}{out(C)} = \frac{1/4}{3} = \frac{1}{12}$$ cuz only C incoming + sharing.

$$PR(B) = \frac{PR(A)}{out(A)} + \frac{PR(C)}{out(C)} = \frac{1/4}{2} + \frac{1/4}{3} = \frac{2.5}{12}$$ cuz A, C coming + sharing

Get it?

Now, if we wanna add damping factor α : (say α = 0.05)

$$PR(A) = (1-\alpha)\left[\alpha \cdot \left(\frac{PR(C)}{out(C)}\right)\right]$$

$$PR(B) = (1-\alpha)\left[\alpha\left(\frac{PR(A)}{out(A)} + \frac{PR(C)}{out(C)}\right)\right]$$