Information Retrieval
Tutorial 7: PageRank

Professor: Michel Schellekens

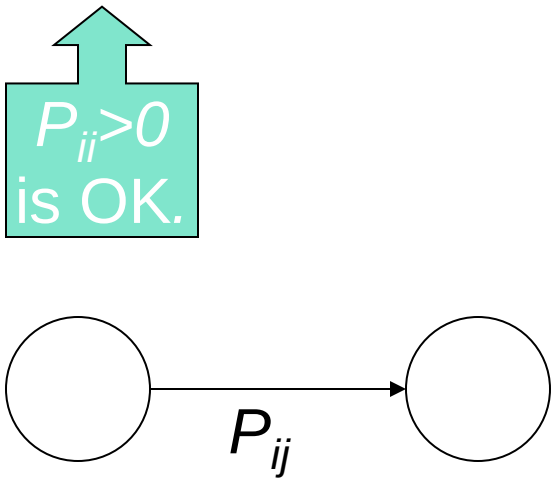
TA: Ang Gao

University College Cork

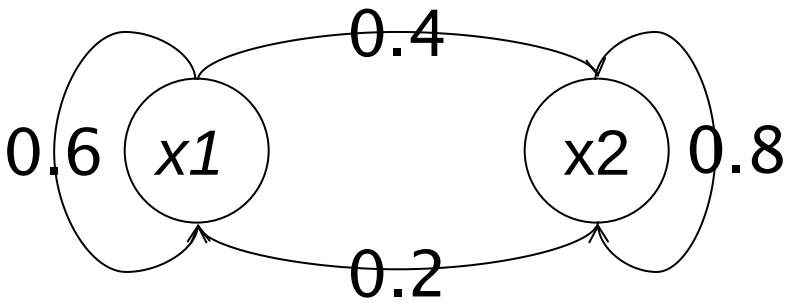
2012-12-07

Markov chains

- A Markov chain consists of *n* states, plus an *n*×*n* transition probability matrix **P**.
- At each step, we are in exactly one of the states.
- For $1 \leq i,j \leq n$, the matrix entry P_{ij} tells us the probability of *j* being the next state, given we are currently in state *i*.



Example: Markov chains



	x1	x2
x1	0.6	0.4
x2	0.2	0.8

$P_0(x1)=1 \quad P_0(x2)=0$

What is $P_1(x1)$ and $P_1(x2)$

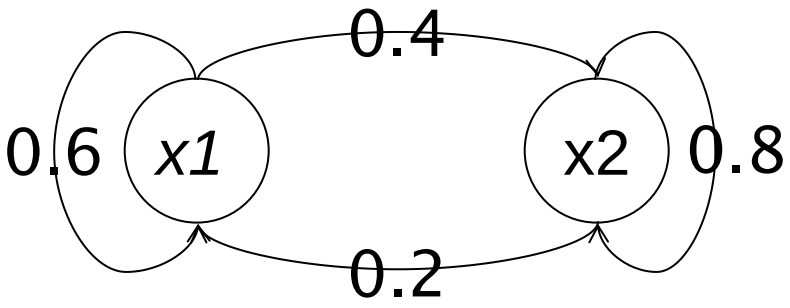
$$P_1(x1) = P_0(x1) * P_{x1x1} + P_0(x2) * P_{x2x1} = 1*0.6+0*0.2 = 0.6$$

$$P_1(x2) = P_0(x1) * P_{x1x2} + P_0(x2) * P_{x2x2} = 1*0.4+0*0.8 = 0.4$$

$$P_1(x2) = 1 - P_1(x1)$$

$$P_t(x_1) = P_{t-1}(x_1) * P_{x1x1} + P_{t-1}(x_2) * P_{x2x1}$$

Example: Markov chains



	x1	x2
x1	0.6	0.4
x2	0.2	0.8

$P_1(x1)=0.6$ $P_1(x2)=0.4$

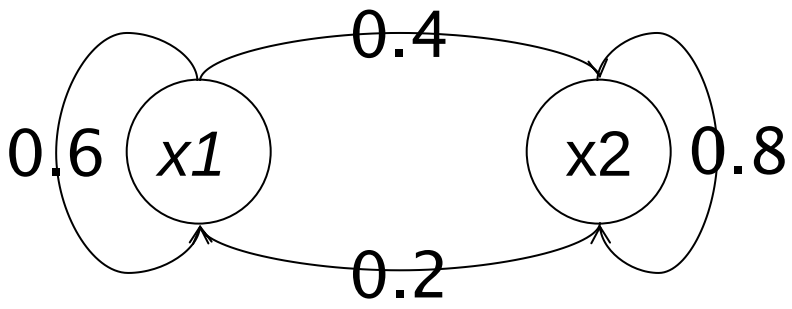
What is $P_2(x1)$ and $P_3(x1)$?

$P_2(x1) = P_1(x1) * P_{x1x1} + P_1(x2) * P_{x2x1} = 0.6*0.6+0.4*0.2 = 0.44$

$P_3(x1) = P_2(x1) * P_{x1x1} + P_2(x2) * P_{x2x1} = 0.44*0.6+0.56*0.2 = 0.376$

How to calculate $P_\infty(x1)$?

Example: Markov chains



	x1	x2
x1	0.6	0.4
x2	0.2	0.8

$P_t(x_1) = P_{t-1}(x_1) * P_{x1x1} + P_{t-1}(x_2) * P_{x2x1}$

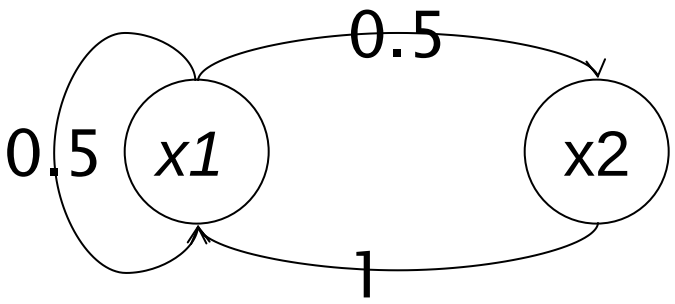
When t goes to ∞ notice $P_t(x_1) = P_{t-1}(x_1)$!

$P_t(x_1) = P_t(x_1) * 0.6 + (1 - P_t(x_1)) * 0.2$

$P_t(x_1) = \frac{1}{3}$ when t -> ∞

steady state probability

Exercise: Markov chains



	x1	x2
x1	0.5	0.5
x2	1	0

Calculate steady state probability for x1 and x2

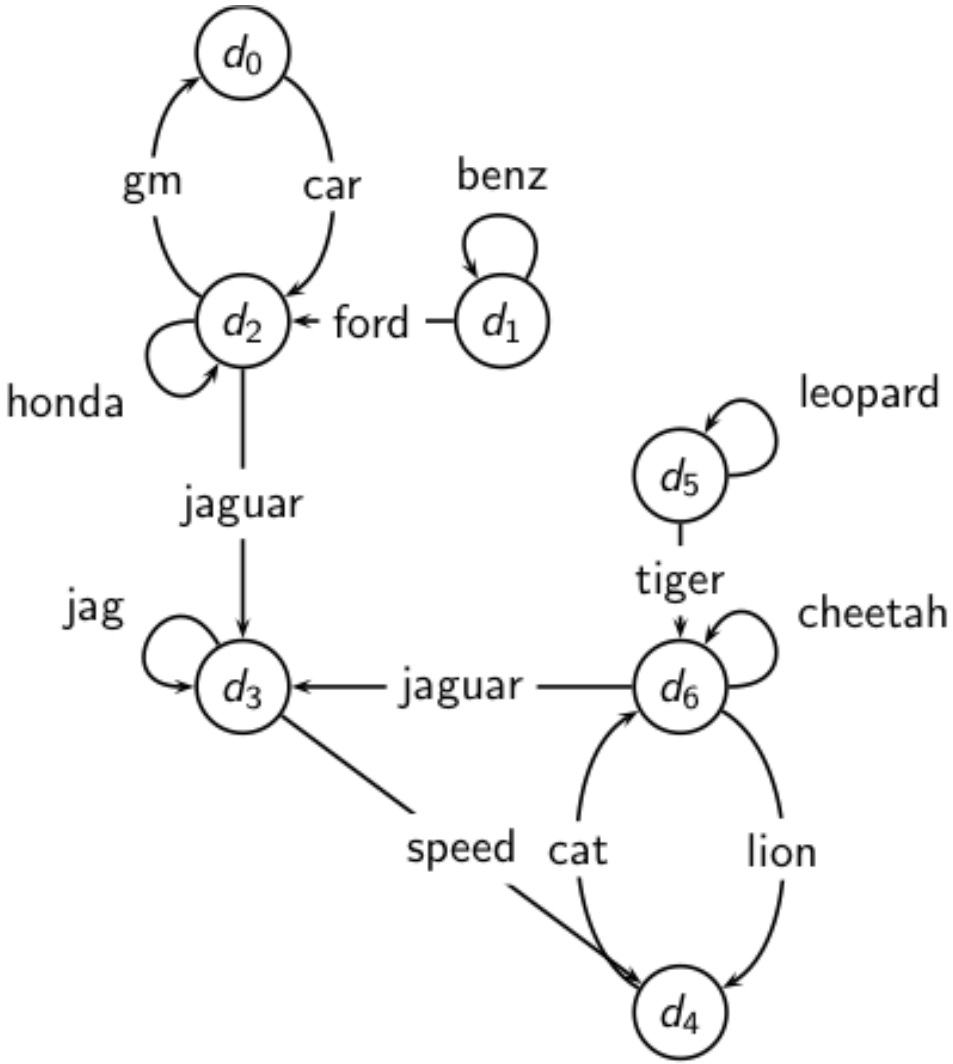
$P_t(x_1) = \frac{2}{3}$ when t -> ∞

$P_t(x_2) = \frac{1}{3}$ when t -> ∞

Model behind PageRank: Random walk

- Imagine a web surfer doing a random walk on the web
  - Start at a random page
  - At each step, go out of the current page along one of the links on that page, equiprobably
- In the steady state, each page has a long-term visit rate.
- This long-term visit rate is the page's PageRank.
- PageRank = long-term visit rate = steady state probability.

Example web graph



Link matrix for example

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0	0	1	0	0	0	0
d_1	0	1	1	0	0	0	0
d_2	1	0	1	1	0	0	0
d_3	0	0	0	1	1	0	0
d_4	0	0	0	0	0	0	1
d_5	0	0	0	0	0	1	1
d_6	0	0	0	1	1	0	1

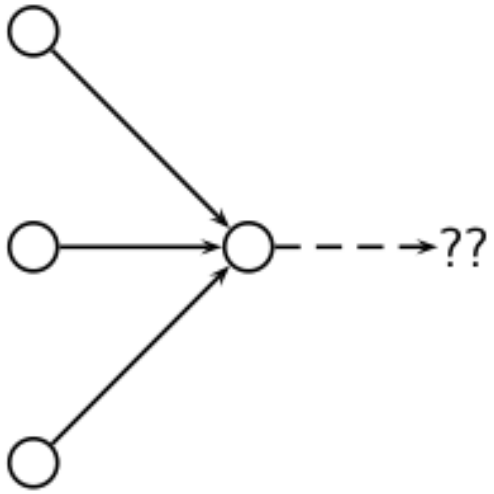
Transition probability matrix P for example

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.00	0.00	1.00	0.00	0.00	0.00	0.00
d_1	0.00	0.50	0.50	0.00	0.00	0.00	0.00
d_2	0.33	0.00	0.33	0.33	0.00	0.00	0.00
d_3	0.00	0.00	0.00	0.50	0.50	0.00	0.00
d_4	0.00	0.00	0.00	0.00	0.00	0.00	1.00
d_5	0.00	0.00	0.00	0.00	0.00	0.50	0.50
d_6	0.00	0.00	0.00	0.33	0.33	0.00	0.33

Long-term visit rate

- Recall: PageRank = long-term visit rate.
- Long-term visit rate of page *d* is the probability that a web surfer is at page *d* at a given point in time.
- Next: what properties must hold of the web graph for the long-term visit rate to be well defined?
- The web graph must correspond to an ergodic Markov chain.
- First a special case: The web graph must not contain dead ends.

Dead ends



- The web is full of dead ends.
- Random walk can get stuck in dead ends.
- If there are dead ends, long-term visit rates are not well-defined (or non-sensical).

Teleporting – to get us of dead ends

- At a dead end, jump to a random web page with prob. $0.1/N$.
- At a non-dead end, with probability 10%, jump to a random web page (to each with a probability of $0.1/N$).
- With remaining probability (90%), go out on a random hyperlink.
  - For example, if the page has 4 outgoing links: randomly choose one with probability (1-0.10)/4=0.225
- 10% is a parameter, the teleportation rate.
- Note: "jumping" from dead end is independent of teleportation rate.

Transition matrix with teleporting

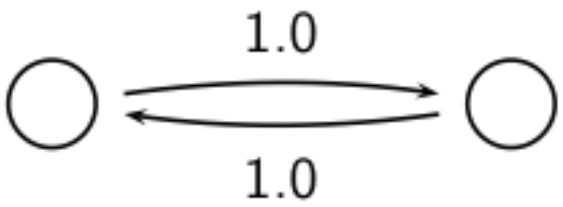
	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.02	0.02	0.88	0.02	0.02	0.02	0.02
d_1	0.02	0.45	0.45	0.02	0.02	0.02	0.02
d_2	0.31	0.02	0.31	0.31	0.02	0.02	0.02
d_3	0.02	0.02	0.02	0.45	0.45	0.02	0.02
d_4	0.02	0.02	0.02	0.02	0.02	0.02	0.88
d_5	0.02	0.02	0.02	0.02	0.02	0.45	0.45
d_6	0.02	0.02	0.02	0.31	0.31	0.02	0.31

Result of teleporting

- With teleporting, we cannot get stuck in a dead end.
- But even without dead ends, a graph may not have well-defined long-term visit rates.
- More generally, we require that the Markov chain be ergodic.

Ergodic Markov chains

- A Markov chain is ergodic if it is irreducible and aperiodic.
- Irreducibility. Roughly: there is a path from any other page.
- Aperiodicity. Roughly: The pages cannot be partitioned such that the random walker visits the partitions sequentially.
- A non-ergodic Markov chain:



Ergodic Markov chains

- Theorem: For any ergodic Markov chain, there is a unique long-term visit rate for each state.
- This is the steady-state probability distribution.
- Over a long time period, we visit each state in proportion to this rate.
- It doesn't matter where we start.
- Teleporting makes the web graph ergodic.
- $\Longrightarrow$Web-graph+teleporting has a steady-state probability distribution.
- $\Longrightarrow$Each page in the web-graph+teleporting has a PageRank.

Formalization of "visit": Probability vector

- A probability (row) vector $\vec{x} = (x_1, \ldots, x_N)$ tells us where the random walk is at any point.
- Example:

(	0	0	0	…	1	…	0	0	0	)
	1	2	3	…	*i*	…	N-2	N-1	N	

- More generally: the random walk is on the page *i* with probability x_i.
- Example:

(	0.05	0.01	0.0	…	0.2	…	0.01	0.05	0.03	)
	1	2	3	…	*i*	…	N-2	N-1	N	

- $\Sigma\, x_i = 1$

Change in probability vector

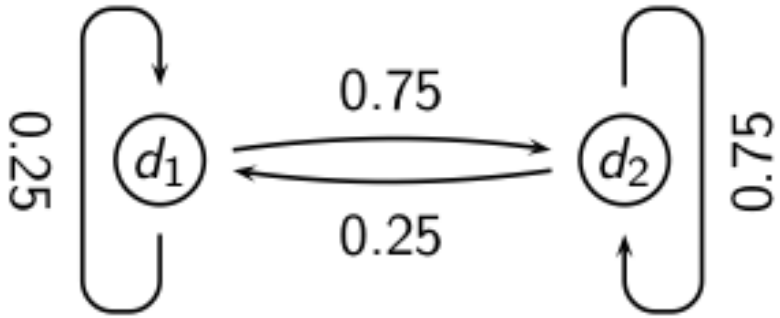
- If the probability vector is $\vec{x} = (x_1, \ldots, x_N)$, at this step, what is it at the next step?
- Recall that row i of the transition probability matrix P tells us where we go next from state i.
- So from $\vec{x}$, our next state is distributed as $\vec{x}P$.

Steady state in vector notation

- The steady state in vector notation is simply a vector $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ of probabilities.
- (We use $\vec{\pi}$ to distinguish it from the notation for the probability vector $\vec{x}$.)
- π is the long-term visit rate (or PageRank) of page *i*.
- So we can think of PageRank as a very long vector – one entry per page.

Steady-state distribution: Example

- What is the PageRank / steady state in this example?



One way of computing the PageRank $\vec{\pi}$

- Start with any distribution $\vec{x}$, e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After k steps, we're at $\vec{x}P^k$.
- Algorithm: multiply $\vec{x}$ by increasing powers of P until convergence.
- This is called the power method.
- Recall: regardless of where we start, we eventually reach the steady state $\vec{\pi}$.
- Thus: we will eventually (in asymptotia) reach the steady state.

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76	0.252	0.748	$= \vec{x}P^3$
t_3	0.252	0.748	0.2496	0.7504	$= \vec{x}P^4$
			. . .		. . .
t_∞	0.25	0.75	0.25	0.75	$= \vec{x}P^\infty$

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

PageRank summary

- Preprocessing
  - Given graph of links, build matrix P
  - Apply teleportation
  - From modified matrix, compute $\vec{\pi}$
  - $\vec{\pi}_i$ is the PageRank of page i.
- Query processing
  - Retrieve pages satisfying the query
  - Rank them by their PageRank
  - Return reranked list to the user

PageRank issues

- Real surfers are not random surfers.
  - Examples of nonrandom surfing: back button, short vs. long paths, bookmarks, directories – and search!
  - → Markov model is not a good model of surfing.
  - But it's good enough as a model for our purposes.
- Simple PageRank ranking produces bad results for many pages.
  - Consider the query [video service].
  - The Yahoo home page (i) has a very high PageRank and (ii) contains both *video* and *service*.
  - If we rank all Boolean hits according to PageRank, then the Yahoo home page would be top-ranked.
  - Clearly not desireble.

PageRank issues

- In practice: rank according to weighted combination of raw text match, anchor text match, PageRank & other factors.
- need more lecture on Learning to Rank.

How important is PageRank?

- Frequent claim: PageRank is the most important component of web ranking.
- The reality:
  - There are several components that are at least as important: e.g., anchor text, phrases, proximity, tiered indexes ...
  - Rumor has it that PageRank in his original form (as presented here) now has a negligible impact on ranking!
  - However, variants of a page's PageRank are still an essential part of ranking.
  - Addressing link spam is difficult and crucial.