# Information Retrieval


**BITS** Pilani
Pilani Campus

Abhishek
March 2020

# Current Status

- Assignments
- Mid-semester
- Syllabus

# Syllabus

M1: Basic Information Retrieval concepts

M2: Text Mining

M3: Web Search and Link Analysis

M4: Multimedia and Cross Lingual IR

M5: Recommender Systems

# Plan for the next few Weeks

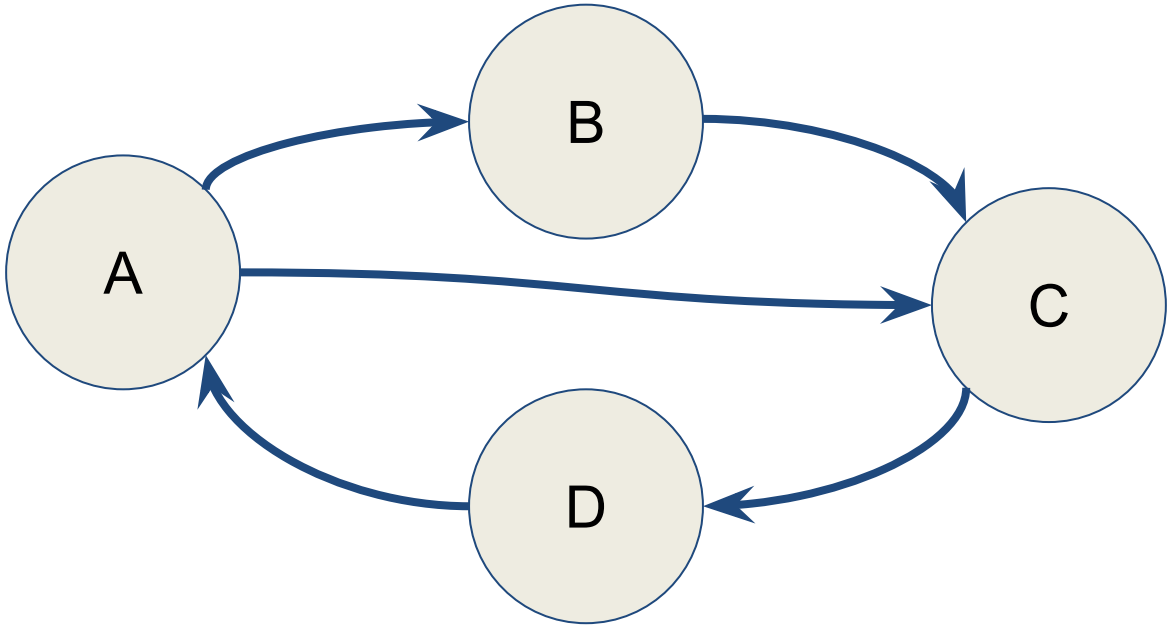
- No Assignment 3 till the classes are online
- Assignment 2 evaluation: 1st to 7th April
- Lectures:
  - A combination on online live sessions + Pre-recorded lecture

# PageRank

# PageRank

- **PageRank** (**PR**) was developed by Larry Page (hence the name Page-Rank) and Sergey Brin, founders of Google.
- It was first part of a research project about a new kind of search engine. That project started in 1995 and led to a functional prototype in 1998.
- PageRank is a way of measuring the importance of website pages.
- Currently, PageRank is not the only algorithm used by Google to order search results, but it is the first algorithm that was used by the company, and it is the best known.
- Paper: http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf

# Web as a graph

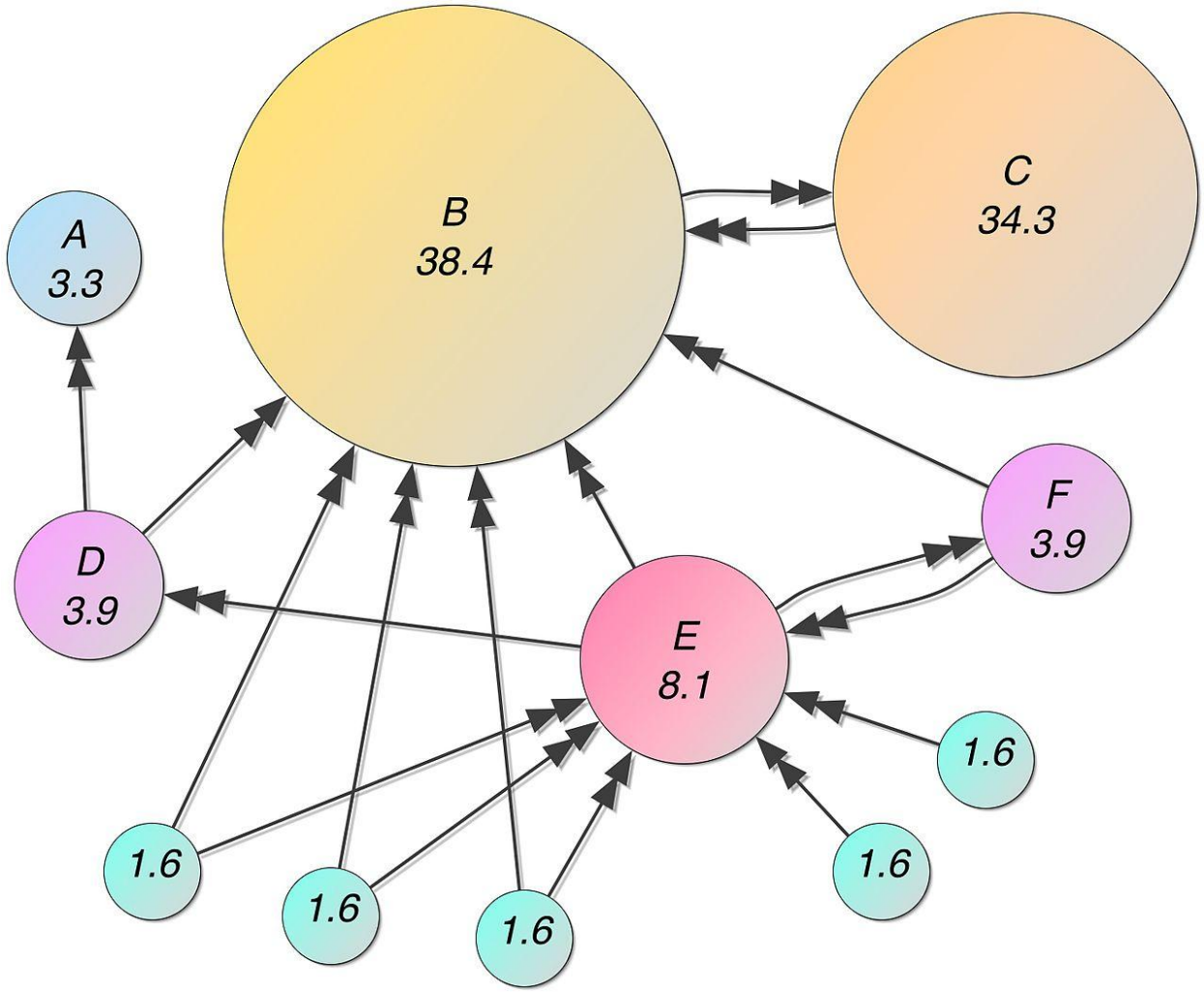


# Model behind PageRank: Random walk



- 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, equiprobable.
- 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.

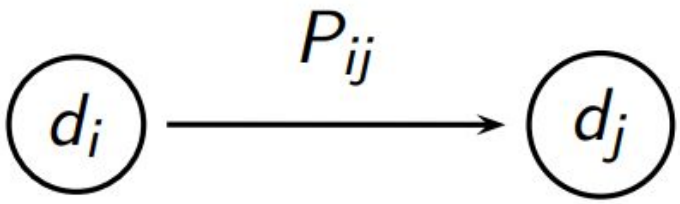
# Model behind PageRank: Random walk



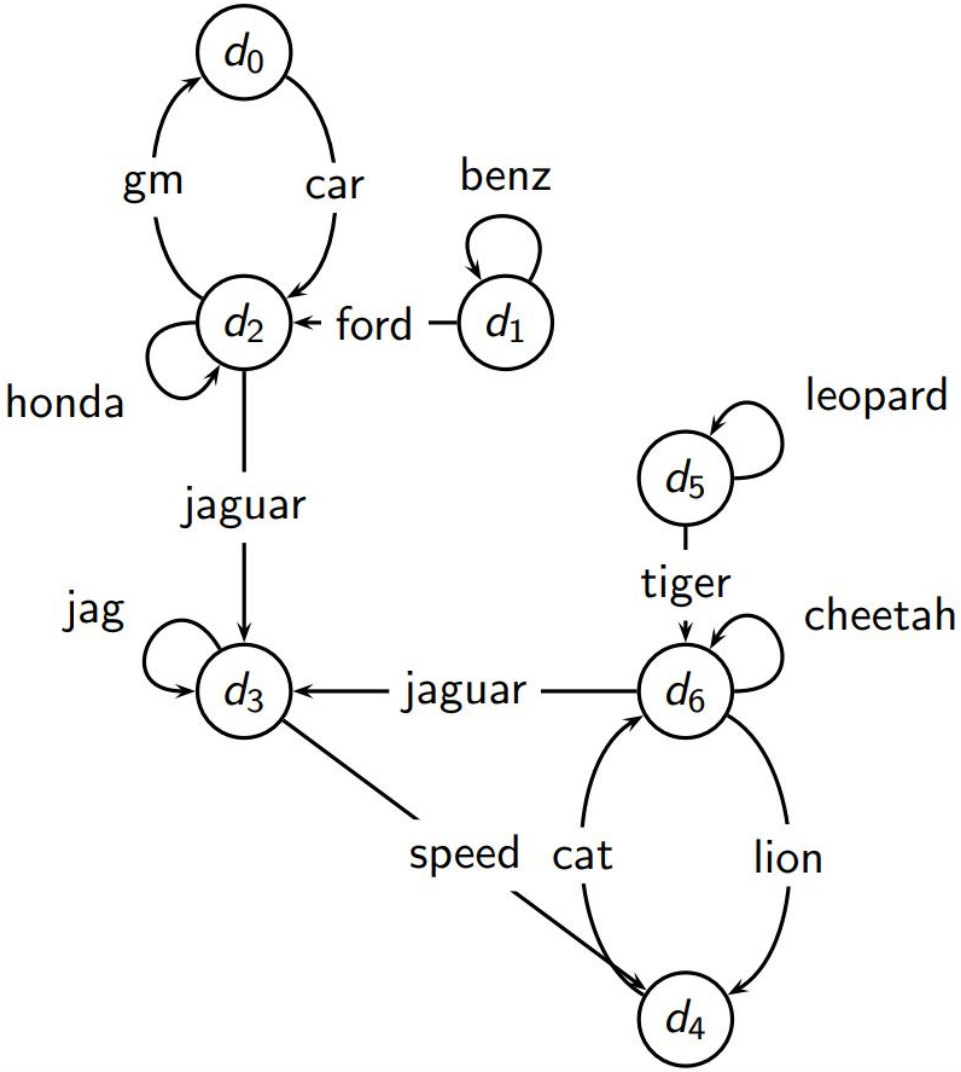
# Formalization of random walk: Markov chains



- A Markov chain consists of N states, plus an N × N transition probability matrix P.
- state = page.
- At each step, we are on exactly one of the pages.
- For 1 ≤ i, j ≤ N, the matrix entry $P_{ij}$ tells us the probability of j being the next page, given we are currently on page i.
  - Clearly, for all i, $\sum_{j=1}^{N} P_{ij} = 1$



# Example web graph



# Link matrix and P matrix 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 |

Link matrix for example

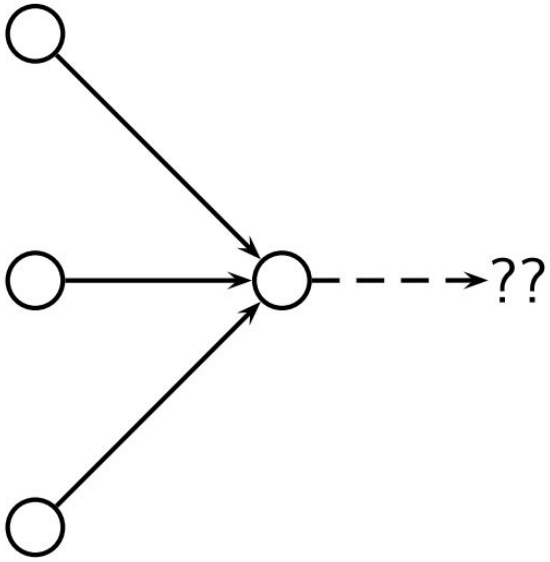
| | $d_0$ | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ |
|---|---|---|---|---|---|---|---|
| $d_0$ | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $d_1$ | 0.00 | 0.50 | 0.50 | 0.00 | 0.00 | 0.00 | 0.00 |
| $d_2$ | 0.33 | 0.00 | 0.33 | 0.33 | 0.00 | 0.00 | 0.00 |
| $d_3$ | 0.00 | 0.00 | 0.00 | 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 |

Transition probability matrix P for example

# 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.

# Teleporting – to get us out of dead ends



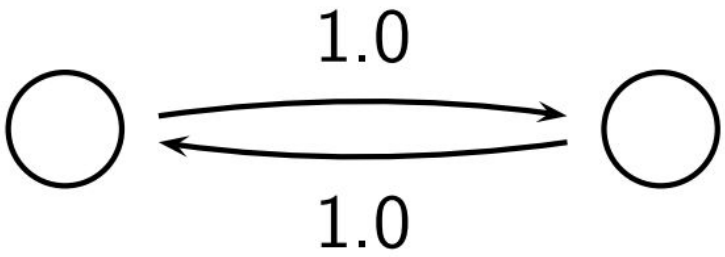
- At a dead end, jump to a random web page with prob. 1/N.
- At a non-dead end, with probability 10%, jump to a random web page (to each with a probability of 0.1/N).
- With remaining probability (90%), go out on a random hyperlink.
- For example, if the page has 4 outgoing links: randomly choose one 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.

# 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 iff it is irreducible and aperiodic.
- Irreducibility. Roughly: there is a path from any page to 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.
- ⇒ Web-graph+teleporting has a steady-state probability distribution.
- ⇒ Each page in the web-graph+teleporting has a PageRank.

# Where we are

- We now know what to do to make sure we have a well-defined PageRank for each page.
- Next: how to compute 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 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 | |

- $\sum x_i = 1$

# 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 $\vec{x}P$. □

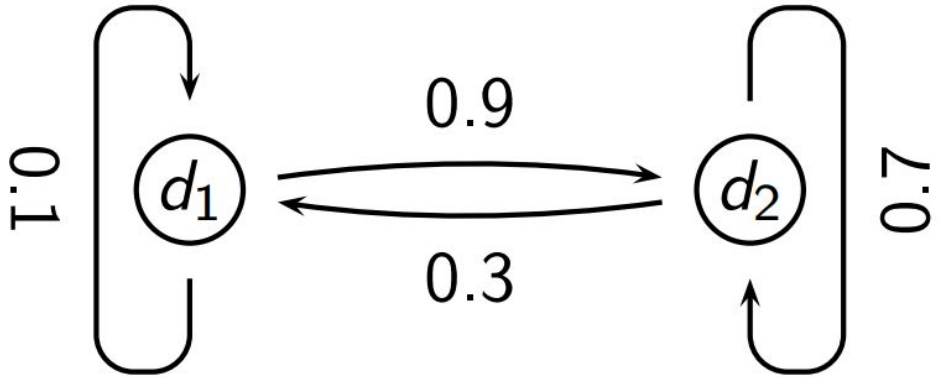
# One way of computing the PageRank



- 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.

# Power method: Example

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




- The steady state distribution (= the PageRanks) in this example are 0.25 for $d_1$ and 0.75 for $d_2$.

# Computing PageRank: Power method

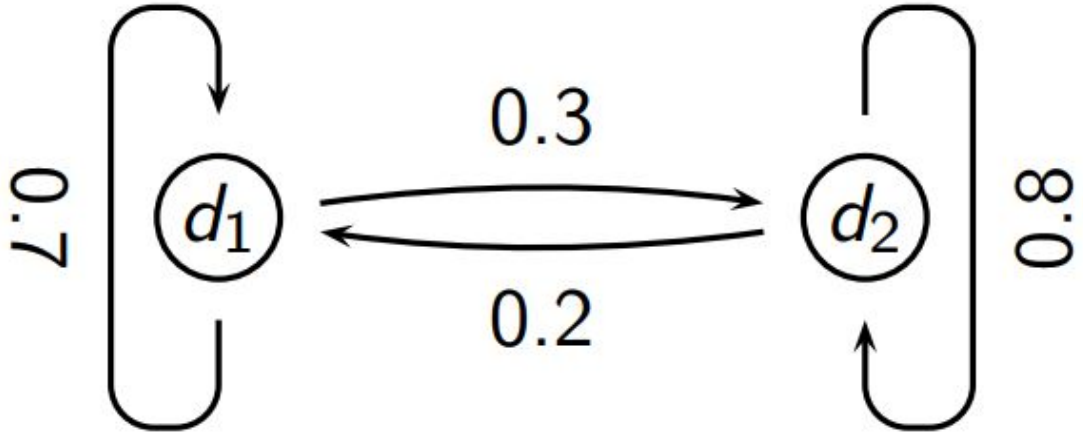
| | $x_1$ $P_t(d_1)$ | $x_2$ $P_t(d_2)$ | | | |
|---|---|---|---|---|---|
| | | | $P_{11} = 0.1$ $P_{21} = 0.3$ | $P_{12} = 0.9$ $P_{22} = 0.7$ | |
| $t_0$ | 0 | 1 | 0.3 | 0.7 | $= \vec{x}P$ |
| $t_1$ | 0.3 | 0.7 | 0.24 | 0.76 | $= \vec{x}P^2$ |
| $t_2$ | 0.24 | 0.76 | 0.252 | 0.748 | $= \vec{x}P^3$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

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

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

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

# Exercise: Compute PageRank using power method



# Solution

| | $x_1$ $P_t(d_1)$ | $x_2$ $P_t(d_2)$ | | |
|---|---|---|---|---|
| | | | $P_{11} = 0.7$ $P_{21} = 0.2$ | $P_{12} = 0.3$ $P_{22} = 0.8$ |
| $t_0$ | 0 | 1 | 0.2 | 0.8 |
| $t_1$ | 0.2 | 0.8 | 0.3 | 0.7 |
| $t_2$ | 0.3 | 0.7 | 0.35 | 0.65 |
| $t_3$ | 0.35 | 0.65 | 0.375 | 0.625 |
| | | | | ... |
| $t_\infty$ | 0.4 | 0.6 | 0.4 | 0.6 |

PageRank

$$\text{vector} = \vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$$

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

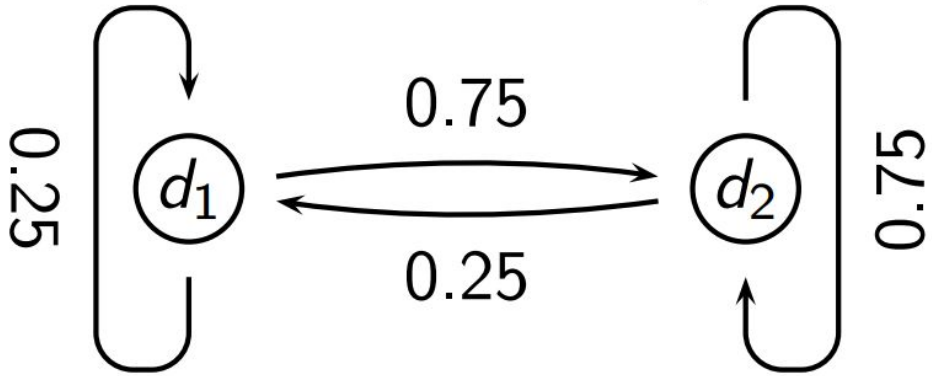
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

# 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 $\vec{x}$.)
- $\pi_i$ 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?



# Steady-state distribution: Example

| | $x_1$ $P_t(d_1)$ | $x_2$ $P_t(d_2)$ | | |
|---|---|---|---|---|
| | | | $P_{11} = 0.25$ | $P_{12} = 0.75$ |
| | | | $P_{21} = 0.25$ | $P_{22} = 0.75$ |
| $t_0$ | 0.25 | 0.75 | 0.25 | 0.75 |
| $t_1$ | 0.25 | 0.75 | (convergence) | |

PageRank

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

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

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

# 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

# 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 its 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.

# Thank You!