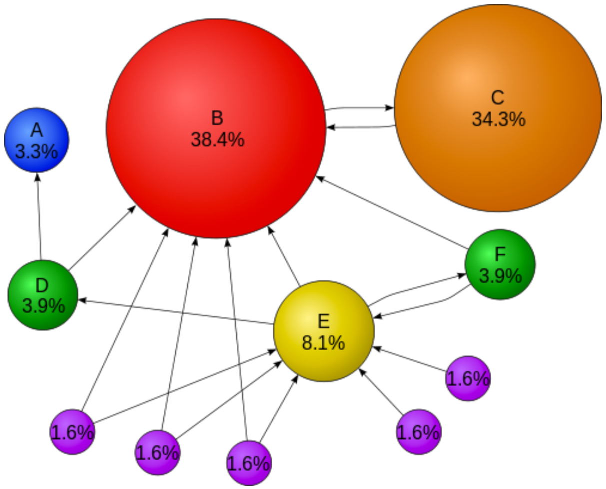


Image:
https://commons.wikimedia.org/wiki/File:PageRanks-Example.svg

COMP90042 LECTURE 18

THE WEB AS A GRAPH: LINK ANALYSIS FOR RETRIEVAL

OVERVIEW

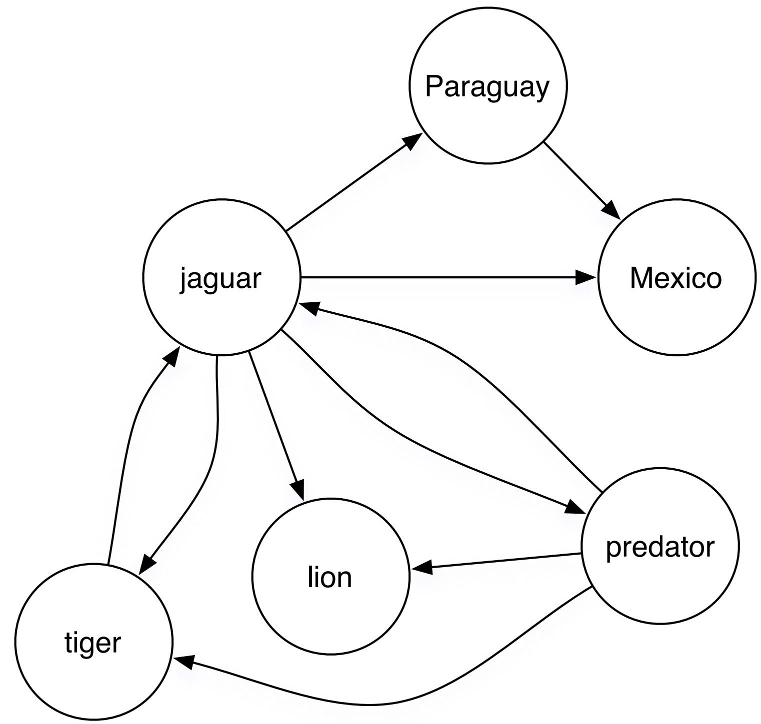
- The web as a graph
- Two methods for automatically determining web-page importance
  - Page-rank
  - Hubs and authorities (HITS)

NOT ALL DOCUMENTS ARE EQUAL

- Documents assumed to be ‘equal’
  - Usefulness for ranking only affected by how well the terms in the document match with the query terms
  - e.g., assumed $P(r|d)$ uniform prior in probabilistic methods
- Can we do better than this?
  - some documents are authoritative and should be ranked higher than others *for any query*

THE WEB AS A GRAPH

- Pages on the Web do not stand alone
  - Treatment as independent "documents" is over-simplification
  - Considerable information in hyperlink structure

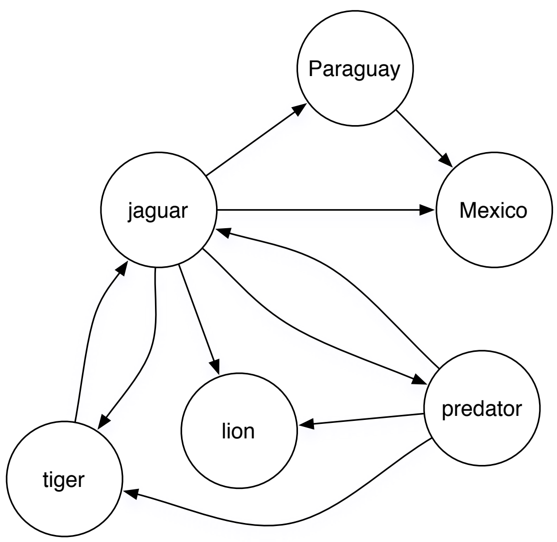


WHAT INFORMATION DOES A HYPERLINK CONVEY?

- Directs user's attention to other pages
- Conferral of authority (not always!)
- Anchor text to explain why linked page is of interest
  - "*IBM computers*" links to www.ibm.com
  - "*search portal*" links to www.yahoo.com
  - "*click here*" links to Adobe Acrobat
- Additional source of terms for indexing
- Perhaps the most important pages have more incoming links?

THE WEB AS A DIRECTED GRAPH

- Formally, consider
  - in-links Number of incoming edges
  - out-links Number of outgoing edges
  - connected components Path connects all pairs of nodes



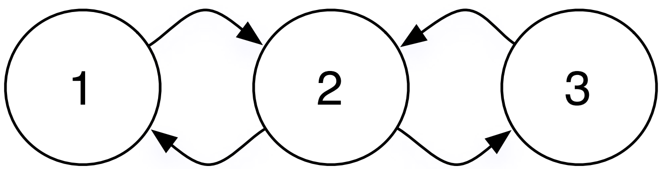
NOT ALL LINKS ARE EQUAL

- Who and what to trust?
  - outgoing links from reputable sites should carry more weight than user-generated content and links from unknown websites
- Web has "bow-tie" structure, comprising
  - in pages that only have outgoing edges to
  - strongly connected component whose pages are highly interlinked, and also link to
  - out pages that only have incoming edges
- Typically don't consider internal links within a web-site (why?)

PAGE RANK

- Assumptions
  - links convey authority of the source page
  - pages with more in links from authoritative sources are more important
- Formalised using model of Random web surfer
  - Consider surfer who visits a web page
    - then follows a random out link, uniformly (repeat above)
    - occasionally types a new random URL into the address bar (called “teleporting” to a new random page)
- Inference problem: which pages does the surfer visit most often?

EXAMPLE GRAPH

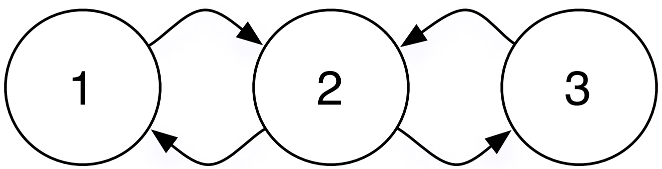


- Transition probabilities (no teleport for now)

$$
\begin{array}{lll}
P(1 \to 1) = 0 & P(1 \to 2) = 1 & P(1 \to 3) = 0 \\
P(2 \to 1) = \frac{1}{2} & P(2 \to 2) = 0 & P(2 \to 3) = \frac{1}{2} \\
P(3 \to 1) = 0 & P(3 \to 2) = 1 & P(3 \to 3) = 0
\end{array}
$$

- Example from MRS, Ch 21

EXAMPLE GRAPH



Represent as matrix, $P_{ij} = P(i \to j)$. I.e.,

$$P = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix}$$

Note that P is the adjacency matrix $A_{ij} = [\![\text{edge exists } i \to j]\!]$, normalised such that each row sums to 1.

ADDING TELEPORTATION

- If at each time step we randomly jump to another node in the graph with probability α
  - scale our original P matrix by 1 − α
  - add α/N to all cells of the resulting matrix

Overall our transition matrix is

$$P_{ij} = (1-\alpha)\frac{A_{ij}}{\sum_{j'} A_{ij'}} + \alpha\frac{1}{N}$$

For the example with $\alpha = 0.5$

$$P = \begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \frac{5}{12} & \frac{1}{6} & \frac{5}{12} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix}$$

MARKOV CHAIN

- Formally we have defined a *Markov chain*
  - a discrete time stochastic process consisting of N states, one per web page
  - assuming the prob. of reaching a state (page) is based only on previous state (page)

$$p(x_t | x_1, x_2, \ldots, x_{t-1}) = p(x_t | x_{t-1})$$

  - characterised by the transition matrix
- Déjà vu: seen before for HMMs

TRANSITIONS AS MATRIX MULT.

Probabilty chain rule can be expressed using matrix multiplication

$$\vec{x}P = \left[P(1)\ P(2)\ P(3)\right] \times \begin{bmatrix} P(1\to 1) & P(1\to 2) & P(1\to 3) \\ P(2\to 1) & P(2\to 2) & P(2\to 3) \\ P(3\to 1) & P(3\to 2) & P(3\to 3) \end{bmatrix}$$

$$= \begin{bmatrix} P(1)P(1\to 1) + P(2)P(2\to 1) + P(3)P(3\to 1) \\ P(1)P(1\to 2) + P(2)P(2\to 2) + P(3)P(3\to 2) \\ P(1)P(1\to 3) + P(2)P(2\to 3) + P(3)P(3\to 4) \end{bmatrix}^{\top}$$

$$= [P(X_{t+1}=1)\ \ P(X_{t+1}=2)\ \ P(X_{t+1}=3)]$$

where $P(1) = P(X_t = 1)$ and $P(2\to 3) = P(X_{t+1} = 3 | X_t = 2)$.

EXAMPLE

- Start at state **x**
  - after one time step, prob on next state is **x**P
  - after two time steps, now (**x**P)P = **x**P^2
- Example
  - **x** = [0 1 0]
  - **x**P = [0.4167 0.1667 0.4167]
  - **x**P^2 = [0.2083 0.5833 0.2083]
  - ...
  - **x**P^{99} = [0.2778 0.4444 0.2778]
  - **x**P^{100} = [0.2778 0.4444 0.2778] (denoted π)

MARKOV CHAIN CONVERGENCE

- Run sufficiently long, state membership converges
  - reaches a steady-state, denoted π
  - transitions from this state leave the state unmodified
  - π record frequency of visiting each page for random surfer in the limit as t $\rightarrow \infty$
- [JFF] When will the Markov Chain converge? Must have the properties:
  - ergodicity: For any start state i, all states j must be reachable with non-zero probability in finite steps. Ergodicity in turn requires
  - irreducibility: reachability between i and j; and
  - aperiodicity: relating to partitioning into sets with internal cycles.

COMPUTING PAGERANK

- Definition of a steady-state is

$$\vec{\pi}P = \vec{\pi}$$

- This is a classic linear algebra problem (finding the left eigenvalues), of the form

$$\vec{\pi}P = \lambda\vec{\pi}$$

  - Can recover several solution vectors for different values of λ
  - Want the principle eigenvector, for which λ = 1
- In practice, may use the power iteration method to handle large graphs

PAGERANK IN PYTHON

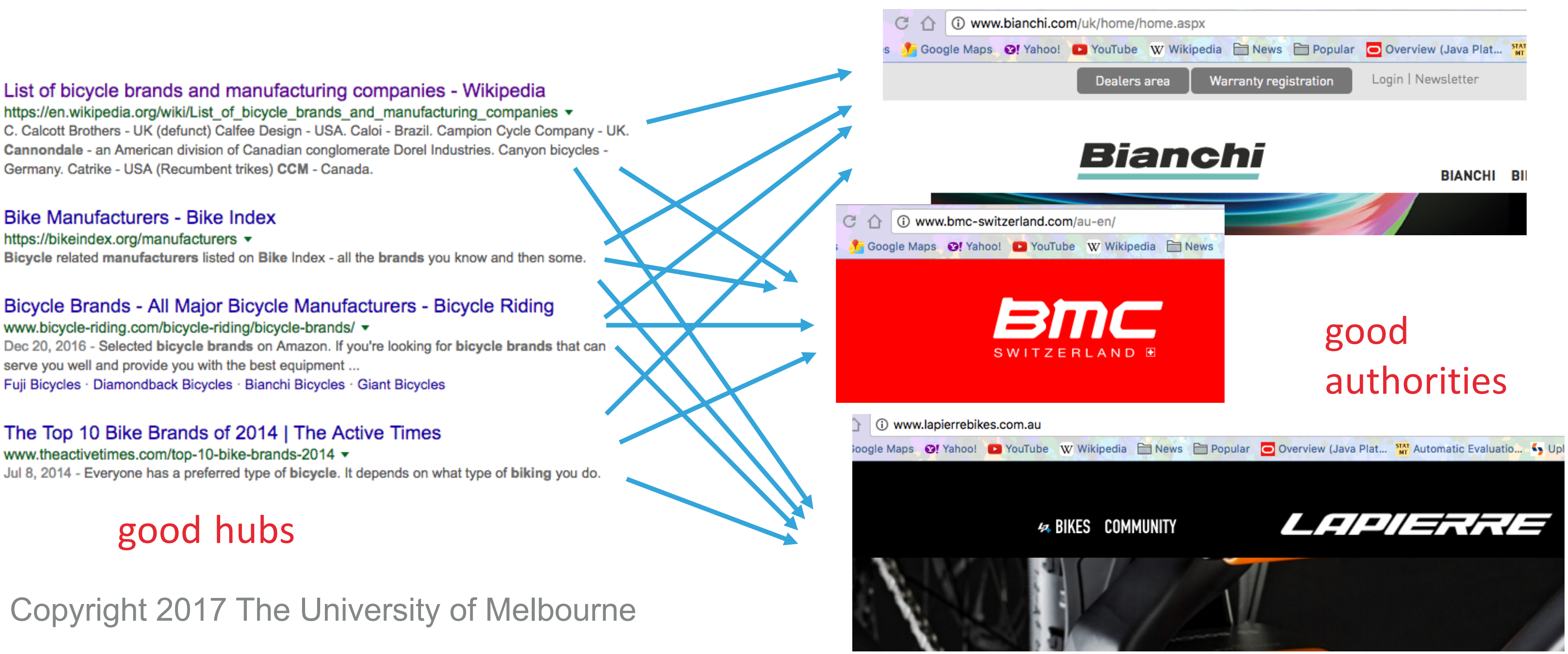
Using `numpy`, e.g., `ipython --pylab`

```
In [1]: P = np.array([[1./6, 2./3, 1./6],
                      [5./12, 1./6, 5./12],
                      [1./6, 2./3, 1./6]])

In [2]: D, W = np.linalg.eig(P.T)

In [3]: W[:,0]/sum(W[:,0])
Out[3]: array([ 0.27777778,  0.44444444,  0.27777778])
```

Note that the transpose is needed to convert the right eigenvectors returned by `np.linalg.eig` to left eigenvectors.

HUBS AND AUTHORITIES (HITS)

- Assumes there are two kinds of pages on web
  - authorities providing authoritative and detailed information
  - hubs containing mainly links to lots of pages about a topic



HITS MOTIVATION

- Depending on our query, we might prefer one type or the other
  - broad topic query, e.g., *information about biking in Melbourne* … → hub
  - specific query, e.g., *is the Yarra trail sealed*? → authority
- More generally, both types of page can be informative
  - but don't know what kind each page is!

FORMULATING HUBS AND AUTHORITIES

- Circular definition
  - A good Hub links to many authorities
  - A good Authority is linked to from many hubs.
- Define
  - $\vec{h}$ vector of hub scores for each web page
  - $\vec{a}$ vector of authority scores for each web page
- Mutually recursive definition for all pages i

$$h_i \leftarrow \sum_{i \to j} a_j$$

$$a_i \leftarrow \sum_{j \to i} h_j$$

INFERENCE

Define

A adjacency matrix, as before $A_{ij} = 1$ denotes edge $i \to j$

Leads to the relations

$$\vec{h} \leftarrow A\vec{a}$$

$$\vec{a} \leftarrow A^{\top}\vec{h}$$

Combining the definitions for $\vec{h}$ into $\vec{a}$,

$$\vec{a} \leftarrow A^{\top}A\vec{a}$$

which is another eigenvalue problem. The principle eigenvalue of $A^{\top}A$ can be used to solve for $\vec{a}$, provided there is a steady-state solution (find $\vec{h}$ in similar way).

HITS EXAMPLE

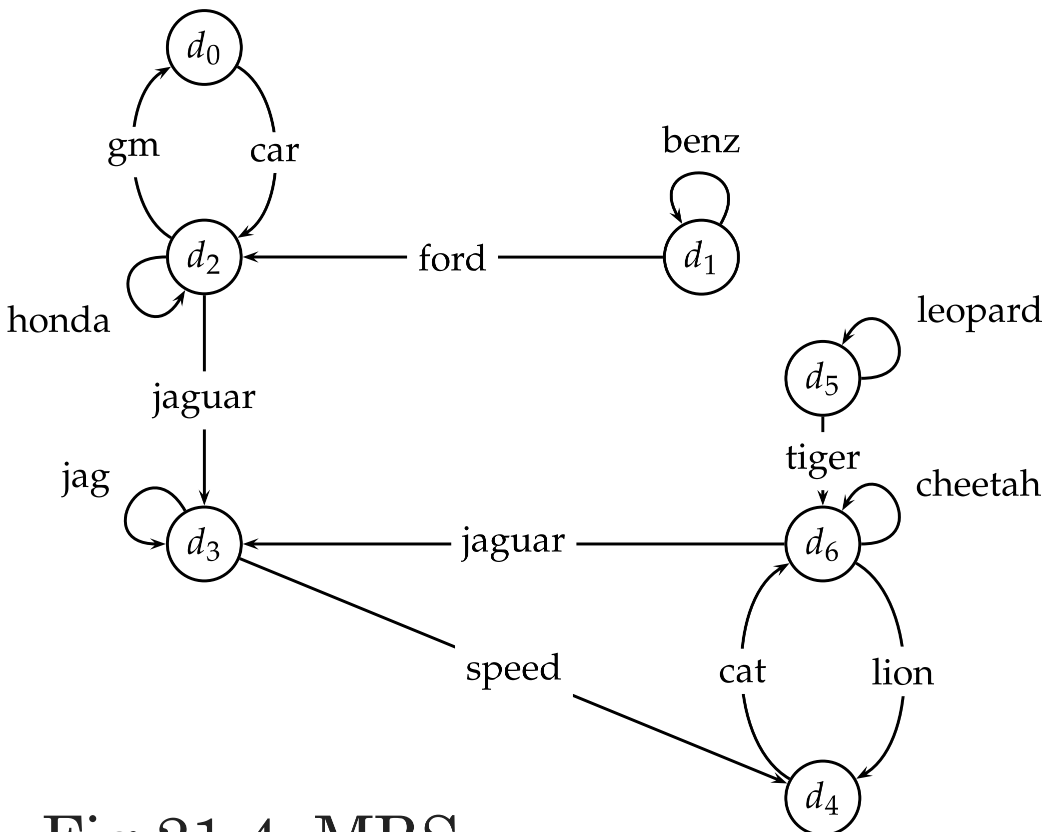


Fig 21.4, MRS

Steps:

1. Compute adjacency matrix, A
2. (optional) Adjust A based on query
3. Solve eigenvalue problem to find a, h

COMPARING PAGERANK AND HITS

- Both static query-independent measures of web page quality
- Can be run offline to score each web page
- Based on latent (unobserved) quality metric for each page
  - single importance score (pagerank)
  - hub and authority scores (hits)
- Plus a specific transition mechanism
- Gives rise to document rankings

USE IN A RETRIEVAL SYSTEM?

- Use as part of a feature representation, e.g.,

$$RSV_d = \sum_{i=1}^{I} \alpha_i h_i(\vec{f}_{:}, \vec{f}_{d,:}, \ldots)$$

  - Features for TF*IDF, BM25 factors, LM, PageRank, HITS etc, each with their own weight α
- Learn α values using machine learned scoring function
  - to match binary relevance judgements
  - to match click-throughs or query reformulations
- Caveat: graph methods can be exploited, e.g., link spam, Google bombs etc.

SUMMARY

- Link structure of web gives rise to a graph, which conveys information about page importance
  - PageRank models a random surfer in the limit, with Markov Chain
  - HITS models hub and authority status of pages
  - Both solved for steady-state using iteration/linear algebra
- Reading
  - (Review of Eigen decompositions) 18.1, “Linear algebra review” of Manning, Raghavan, and Schutze, Introduction to Information Retrieval.
  - Chapter 21, “Link Analysis” of Manning, Raghavan, and Schutze, Introduction to Information Retrieval.