# MSAN 602: Computational Analytics I

Dr. Matthew Dixon[1]

[1]Graduate Program in Analytics, University of San Francisco
MSAN602.analytics.usfca.edu
Email: mfdixon@usfca.edu

Lecture 11: HITS and Pagerank
2nd October, 2012

# Reading

- Section 10.5 of CT1
- Section 6 of BR2

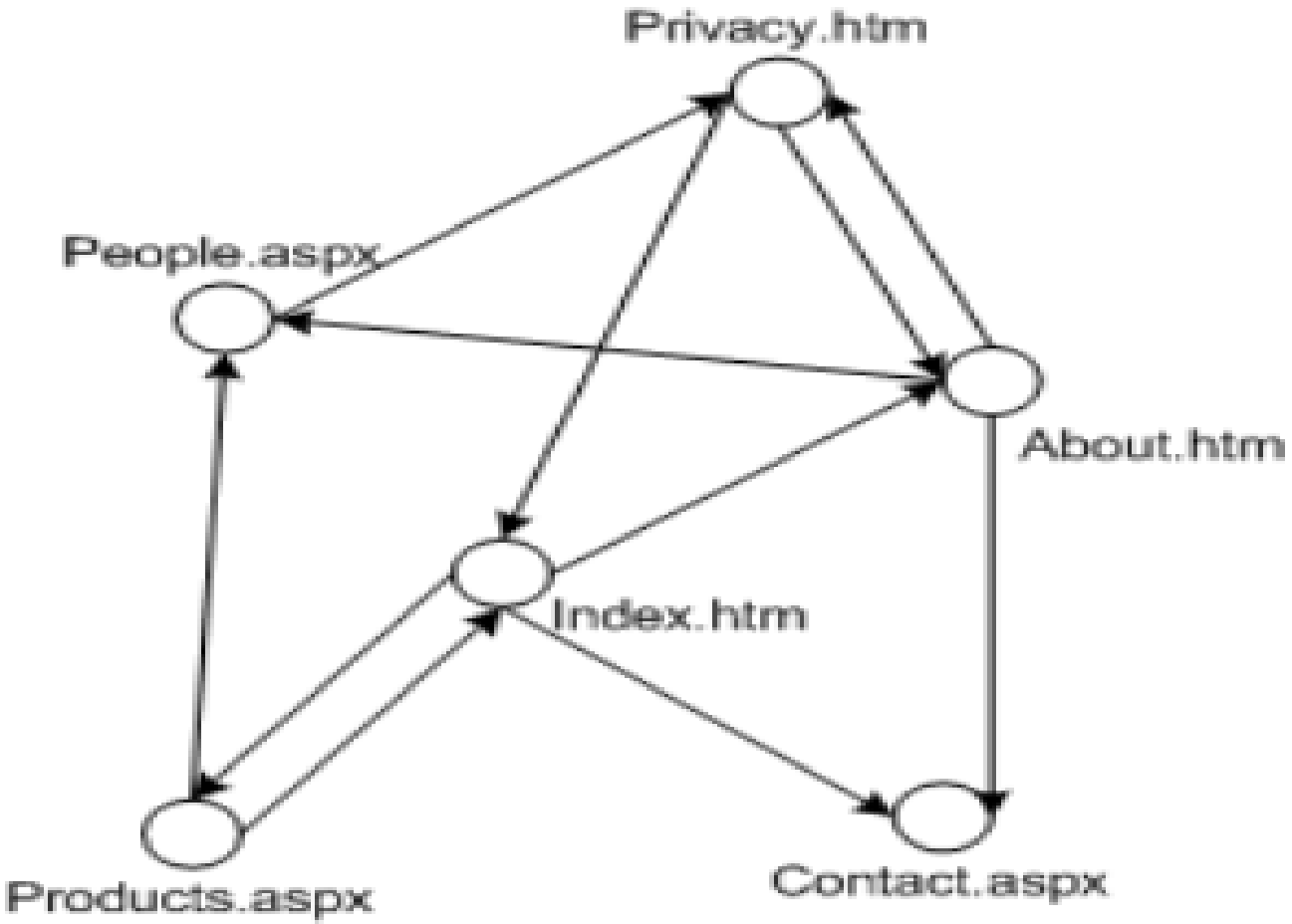
# Agenda: Data Mining with Graphs

- Web search engines use the HITS and pagerank algorithm to rank results
- Use the `arulesViz` package to visualize rules produced by the apriori algorithm
- Use the `igraph` package to view the results from extracting co-located words in a corpus
- Use the `igraph` package to identify structures in networks

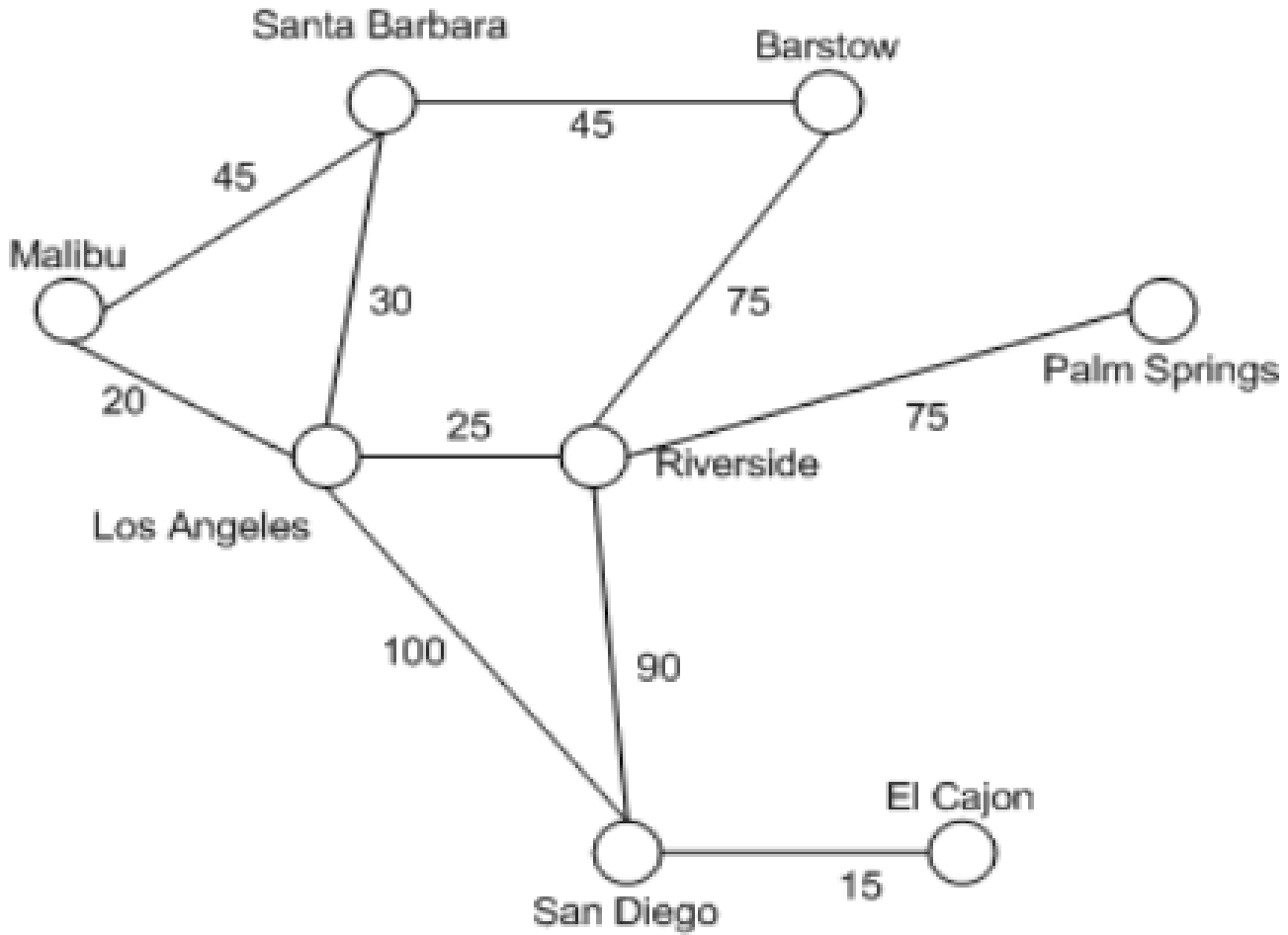
# Introduction to Graphs

- Let $G(V, E)$ denote an n-vertex directed graph representing the web with vertex-set V and edge-set E.
- Let $A = \{a_{ij}\}$ denote the $n \times n$ adjacency matrix of graph $G(V, E)$: that is, if the vertices are $V = \{v_1, v_2, \ldots, v_n\}$, then $a_{ij} = 1$ if $v_i$ is connected to $v_j$ and 0 otherwise.
- Graphs can be weighted or unweighted and directed or undirected

# Example of an unweighted, directed graph



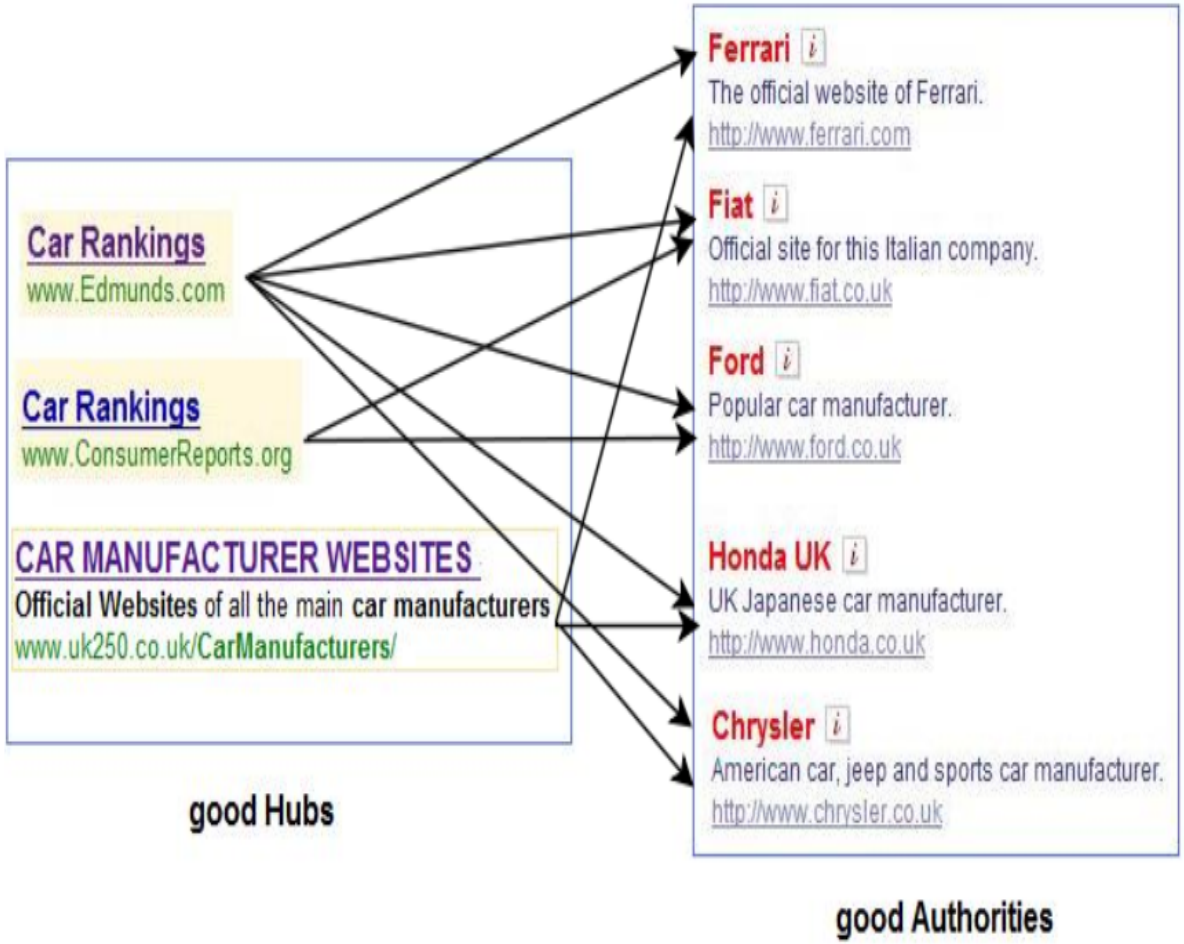
# Example of a weighted, undirected graph



# HITS algorithm

- Jon Kleinberg's algorithm called HITS (hyperlink-induced topic search) is now part of the Ask search engine (www.Ask.com).
- HITS is applied on a subgraph after a search is done on the complete graph
- Page i is called an *authority* for a query $Q$ if it contains valuable information on the subject.
- *Hubs* contain useful links towards the authoritative pages. Hubs point the search engine in the "right direction".
- HITS defines hubs and authorities recursively

# Hubs and Authorities



# HITS: conceptual description

- HITS constructs a root set $R_Q$ of all pages which have a high occurence of Q.
- Extend the subgraph $R_Q$ by including all edges coming from or pointing to nodes from $R_Q$.
- The resulting subgraph $S_Q$ is referred to as the *seed* of the search.

# HITS: conceptual description

- HITS identifies good authorities and hubs for a topic by assigning two numbers to a page $p$: an authority weight $a_p$ and a hub weight $h_p$

$$a_p = \sum_{q \mid q \to p} h_q, \quad h_p = \sum_{q \mid q \leftarrow p} a_q$$

- A good hub increases the authority weight of the pages it points to. If a page is pointed to by many good hubs, its authority weight should increase
- A good authority increases the hub weight of the pages that point to it. If a page is pointing to many good authorities, its hub weight should increase
- Apply the two operations above alternatively until equilibrium values for the hub and authority weights are reached.

# HITS: solution approach

- Initialize all $a_p$ and $h_p$ values to a uniform constant.
- Recasting the above relations in matrix vector notation:

$$\mathbf{h} = A\mathbf{a}, \qquad a = A^T\mathbf{h}$$

- The following sequences hold

$$\begin{aligned} \mathbf{h} &= A\mathbf{a} = AA^T\mathbf{h} = (AA^T)^2\mathbf{h} = \cdots = (AA^T)^k\mathbf{h} \quad (1) \\ \mathbf{a} &= A^T\mathbf{h} = A^TA\mathbf{a} = (A^TA)^2\mathbf{a} = \cdots = (A^TA)^k\mathbf{a} \quad (2) \end{aligned}$$

# HITS: solution approach

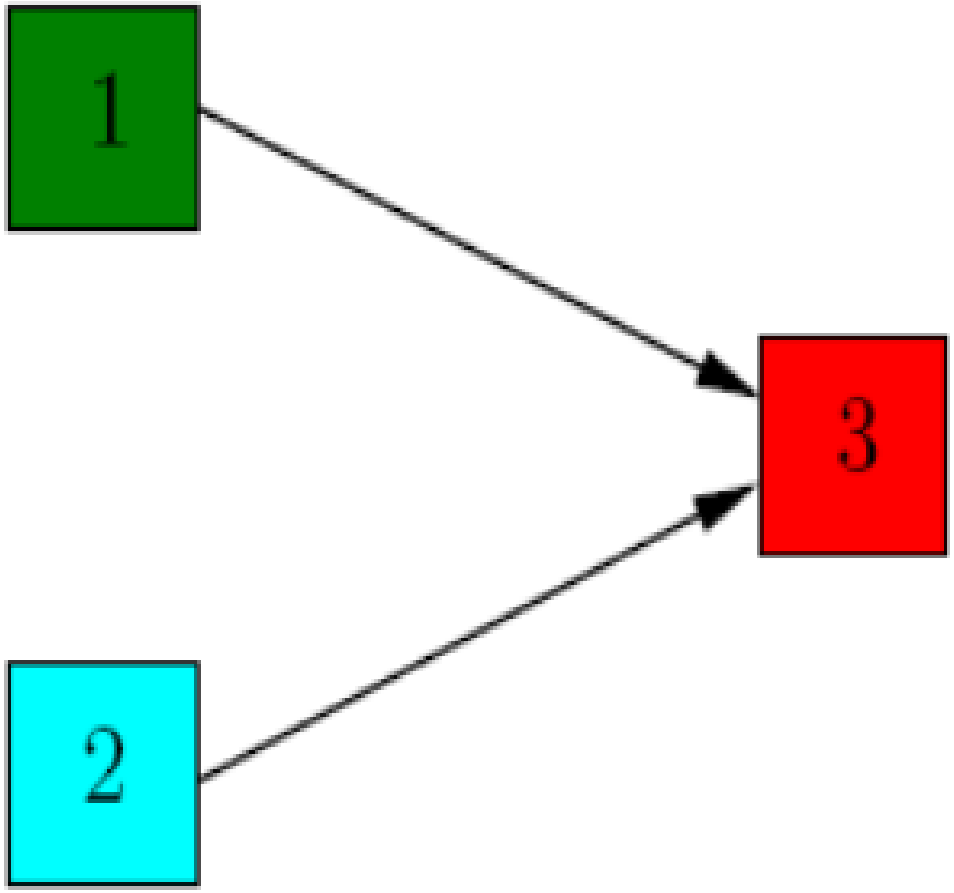
- **a** is the normalized eigenvector (sum of elements is unity) corresponding to the largest eigenvalue of $A^T A$
- **h** is the normalized eigenvector (sum of elements is unity) corresponding to the largest eigenvalue of $AA^T$
- Using the singular value decomposition on $A = USV^T$

$$A^T A = VS^T U^T USV^T = V(S^T S)V^T = V\Sigma V^T$$

$$AA^T = USV^T VS^T U^T = U(SS^T)U^T = U\Sigma U^T$$

- First vectors of $U$ and $V$ are the first eigenvectors of $AA^T$ and $A^T A$, i.e. the (unnormalized) hub and authority weight vectors
- Normalize weight vectors with $||\cdot||_2$ or with the sum of the elements

# 3-vertex directed graph example



# 3-vertex directed graph example

Adjacency matrix:

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

# 3-vertex directed graph example

Authority weight

$$\mathbf{a} = A^T\mathbf{h} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

# 3-vertex directed graph example

Updated hub weight

$$\mathbf{h} = A\mathbf{a} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}$$

# 3-vertex directed graph example

```
A <- t(matrix(c(0,0,1,0,0,1,0,0,0), nr=3))
U<-svd(A)$u
V<-svd(A)$v

h<-U[,1]/sum(U[,1])
a<-V[,1]/sum(V[,1])
```

# The PageRank Algorithm

- PageRank developed by Larry Page and Sergey Brin at Stanford University[1]
- Based on the idea of a random surfer
- PageRank is used for ranking all the nodes of the complete graph and then applying a search
- Probability for moving from a page to another page modelled as a state transition probability

[1] Google came from googol which is $10^{100}$

# The PageRank Algorithm

- The probability that the random surfer visits a page is its PageRank.
- The PageRank $r_i$ of node $i$ is given by the recursive formula

$$r_i = (1 - d) + d \sum_{j \in \{(j,i)\}} \frac{r_j}{o_j}$$

- $o_j$ is the out-degree of node $j$
- $d$ is a damping factor between 0 and 1 (typically set to 0.85) is the probability at each page the random surfer will get bored and request another random page

# The PageRank Algorithm

- In vector notation:

$$\mathbf{r} = (1 - d)\mathbf{e} + dT\mathbf{r}$$

- The Transition matrix $T$ is the transpose of the adjacency matrix with the columns normalized so that the sum of each column vector is unity

# Power Iteration for PageRank

**PageRank-Iterate($C$)**

$P_0 \leftarrow e/n$

$k \leftarrow 1$

**repeat**

$P_k \leftarrow (1-d)e + dA^T P_{k-1}$;

$k \leftarrow k+1$;

**until** $\|P_k - P_{k-1}\|_1 < \varepsilon$

**return** $P_k$

# 3-vertex directed graph example

Adjacency matrix:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

# 3-vertex directed graph example

- The Transition matrix is the probability of moving from node $i$ to node $j$

$$T = \begin{bmatrix} 0 & 0 & .5 \\ 1 & 0 & .5 \\ 0 & 1 & 0 \end{bmatrix}$$

# 3-vertex directed graph example

Find the principal eigenvector[2] of the transition matrix

$$Tr = r$$

[2] with a corresponding unit eigenvalue

# 3-vertex directed graph example

Apply the power iteration method to find the principal eigenvector

$$r = T^k r_0 = T(\ldots(Tr_0))$$

Choose

$$r_0 = \begin{bmatrix} 0.33 \\ 0.33 \\ 0.33 \end{bmatrix}$$

# 3-vertex directed graph example

$$r_1 = Tr_0 = \begin{bmatrix} 0 & 0 & 0.5 \\ 1 & 0 & 0.5 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.33 \\ 0.33 \\ 0.33 \end{bmatrix}$$

$$r_1 = Tr_0 = \begin{bmatrix} 0 \cdot 0.33 + 0 \cdot 0.33 + 0.5 \cdot 0.33 \\ 1 \cdot 0.33 + 0 \cdot 0.33 + 0.5 \cdot 0.33 \\ 0 \cdot 0.33 + 1 \cdot 0.33 + 0 \cdot 0.33 \end{bmatrix}$$