CSE 5243 INTRO. TO DATA MINING

Graph Data

Yu Su, CSE@The Ohio State University

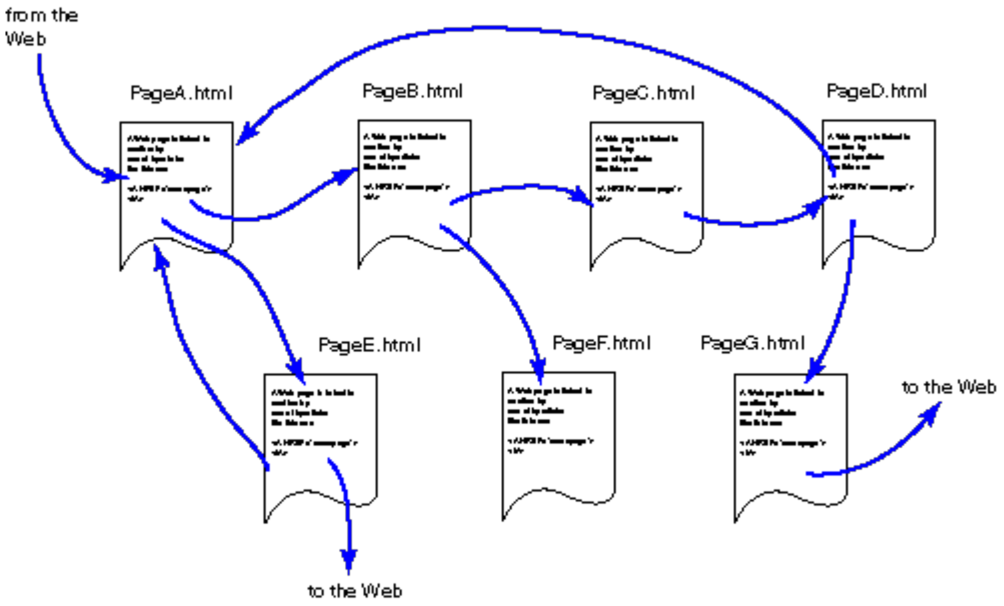
Slides adapted from UIUC CS412 by Prof. Jiawei Han and OSU CSE5243 by Prof. Huan Sun

Chapter 4 Graph Data:
http://www.dataminingbook.info/pmwiki.php/Main/BookPathUploads?action=downloadman&upname=book-20160121.pdf ,
http://www.dataminingbook.info/pmwiki.php

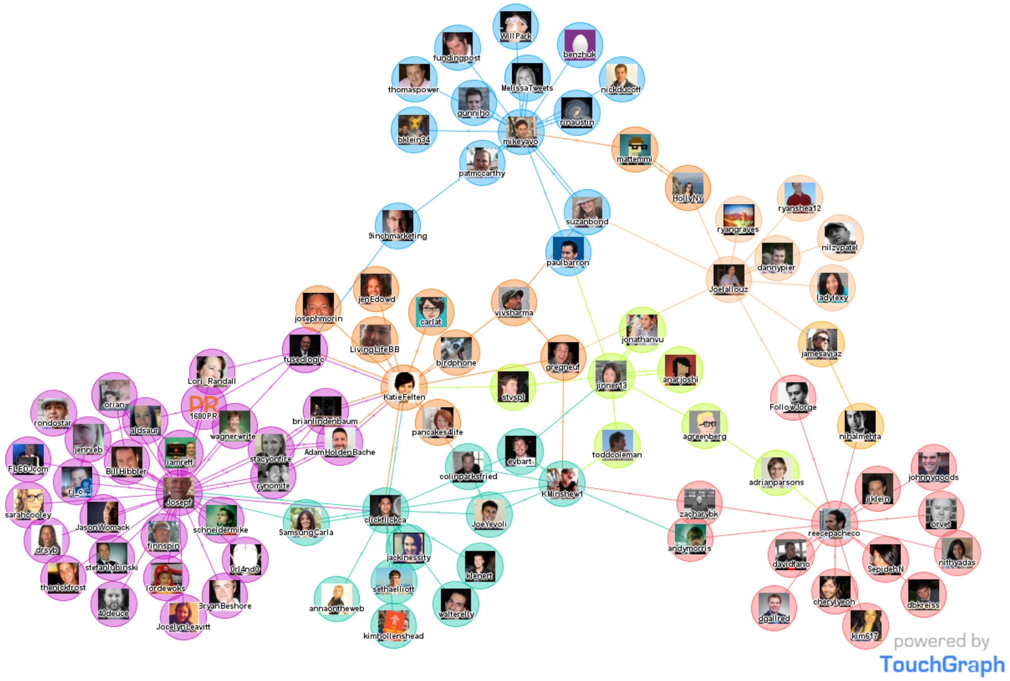
GRAPH BASICS AND A GENTLE INTRODUCTION TO PAGERANK

Slides adapted from Prof. Srinivasan Parthasarathy @OSU

Graphs from the Real World



The Web: hyperlinked docs



Social networks

https://chortle.ccsu.edu/Java5/Notes/appendixA/htmlPart2_6.html
http://www.touchgraph.com/news

Primitives and Notations

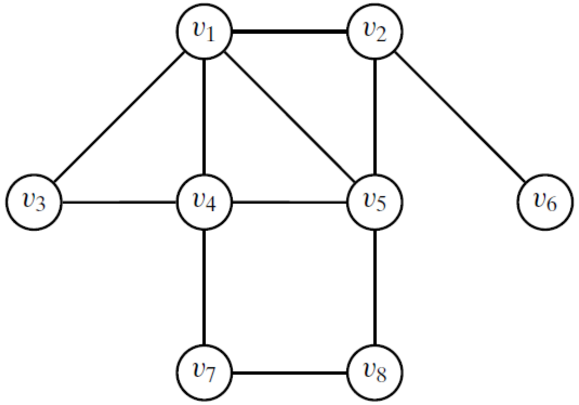
- G = (V, E)
  - $E \subseteq V \times V$, and can also be represented as an adjacency matrix.
- Undirected vs. directed graph

A directed edge (v_i, v_j) is also called an *arc*, and is said to be *from* v_i *to* v_j. We also say that v_i is the *tail* and v_j the *head* of the arc.

Primitives and Notations

- G = (V, E)
  - E can also be represented as an adjacency matrix
- Undirected vs. directed graph
- Degree

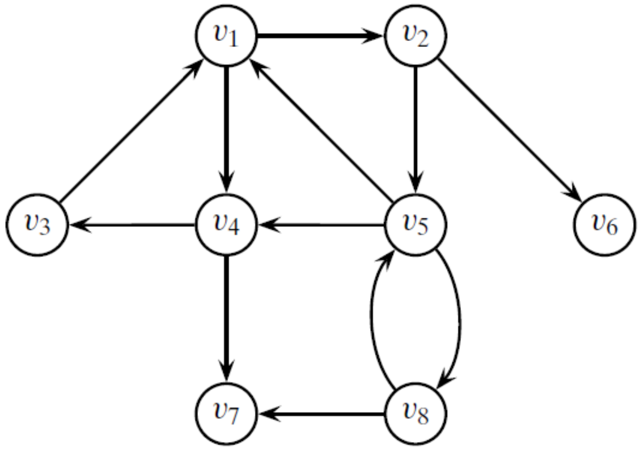
The *degree* of a node $v_i \in V$ is the number of edges incident with it



(a)

Primitives and Notations

- G = (V, E)
  - E can also be represented as an adjacency matrix
- Undirected vs. directed graph
- Degree



(b)

For directed graphs, the *indegree* of node v_i, denoted as $id(v_i)$, is the number of edges with v_i as head, that is, the number of incoming edges at v_i. The *outdegree* of v_i, denoted $od(v_i)$, is the number of edges with v_i as the tail, that is, the number of outgoing edges from v_i.

Primitives and Notations

- G = (V, E)
  - E can also be represented as an adjacency matrix
- Undirected vs. directed graph
- Degree
- (Shortest) distance between two vertices

The *eccentricity* of a node v_i is the maximum distance from v_i to any other node in the graph:

$$\text{Eccentricity}(v) = \max_{u \neq v} dist(u, v)$$

Primitives and Notations

- G = (V, E)
  - E can also be represented as an adjacency matrix
- Undirected vs. directed graph
- Degree
- (Shortest) distance between two vertices

The *eccentricity* of a node v_i is the maximum distance from v_i to any other node in the graph:

$$\text{Eccentricity}(v) = \max_{u \neq v} dist(u, v)$$

Primitives and Notations

- G = (V, E)
  - E can also be represented as an adjacency matrix
- Undirected vs. directed graph
- Degree
- (Shortest) distance between two vertices

The *radius* of a connected graph, denoted $r(G)$, is the minimum eccentricity of any node in the graph:

$$\text{Radius}(G) = \min_{v \in V} \text{Eccentricity}(v)$$

Primitives and Notations

- G = (V, E)
  - E can also be represented as an adjacency matrix
- Undirected vs. directed graph
- Degree
- (Shortest) distance between two vertices

The *diameter*, denoted $d(G)$, is the maximum eccentricity of any vertex in the graph:

$$\text{Diameter}(G) = \max_{v \in V} \text{Eccentricity}(v)$$

Properties of Nodes

- Centrality: how "central" or important a node is in the graph
  - How close the node is to all other nodes?

$$\text{Closeness Centrality}(v) = \frac{1}{\sum_{u \neq v} dist(u, v)}$$

A node v_i with the smallest total distance, $\sum_j d(v_i, v_j)$, is called the *median node*.

Properties of Nodes

- Centrality: how “central” or important a node is in the graph
  - How close the node is to all other nodes?
  - How much is a node a “choke point”?

Betweenness centrality: How many shortest paths between all pairs of vertices include vi.

$$\gamma_{jk}(v_i) = \frac{\eta_{jk}(v_i)}{\eta_{jk}}$$: the fraction of shortest paths between vertices v_j and v_k through v_i

The betweenness centrality for a node v_i is defined as

$$c(v_i) = \sum_{j \neq i} \sum_{\substack{k \neq i \\ k > j}} \gamma_{jk}(v_i) = \sum_{j \neq i} \sum_{\substack{k \neq i \\ k > j}} \frac{\eta_{jk}(v_i)}{\eta_{jk}}$$

Properties of Nodes

- Clustering coefficient: how much does a node cluster with neighbors
  - Local clustering coefficient

The **local clustering coefficient** of a vertex (node) in a graph quantifies how close its neighbors are to being a clique (complete graph).

The proportion of links between the vertices within its neighbourhood divided by the number of links that could possibly exist between them.

Background

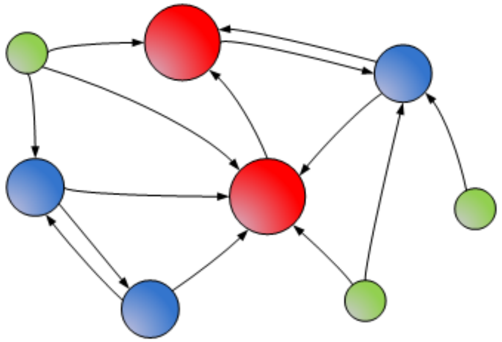
- Besides the keywords, what other evidence can one use to rate the importance of a webpage?

Background

- Besides the keywords, what other evidence can one use to rate the importance of a webpage?
- Solution: Use the hyperlink structure
- E.g. a webpage linked by many webpages is probably important.
  - but this method is not global (comprehensive).
- PageRank was developed by Larry Page and Sergey Brin in 1998.

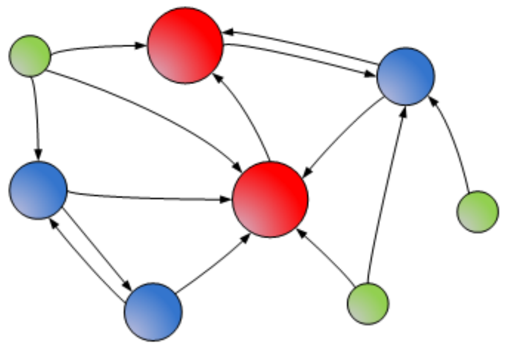
Idea

- A graph representing WWW
  - Node: webpage
  - Directed edge: hyperlink



Idea

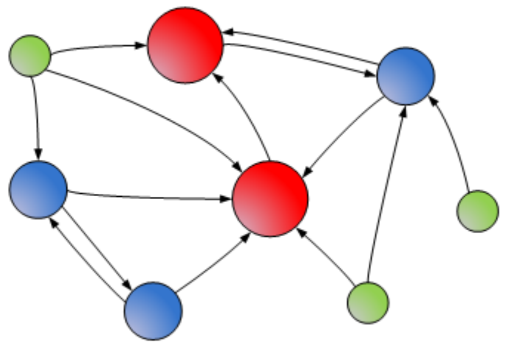
- A graph representing WWW
  - Node: webpage
  - Directed edge: hyperlink



- A user randomly clicks the hyperlink to surf WWW.
  - The probability a user stop in a particular webpage is the PageRank value.

Idea

- A graph representing WWW
  - Node: webpage
  - Directed edge: hyperlink



- A user randomly clicks the hyperlink to surf WWW.
  - The probability a user stop in a particular webpage is the PageRank value.

- A node that is linked by many nodes with high PageRank value receives a high rank itself;
  If there are no links to a node, then there is no support for that page.

Formal Formulation

Let $G = (V, E)$ be a directed graph, with $|V| = n$. The adjacency matrix of G is an $n \times n$ asymmetric matrix $\mathbf{A}$ given as

$$\mathbf{A}(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 0 & \text{if } (u, v) \notin E \end{cases}$$

Let $p(u)$ be a positive real number, called the *prestige* score for node u.

$$\begin{aligned} p(v) &= \sum_u \mathbf{A}(u, v) \cdot p(u) \\ &= \sum_u \mathbf{A}^T(v, u) \cdot p(u) \end{aligned}$$

the prestige of a node depends on the prestige of other nodes pointing to it.

Formal Formulation

Let $p(u)$ be a positive real number, called the *prestige* score for node u.

$$p(v) = \sum_u \mathbf{A}(u, v) \cdot p(u)$$

$$= \sum_u \mathbf{A}^T(v, u) \cdot p(u)$$

the prestige of a node depends on the prestige of other nodes pointing to it.

Across all the nodes, we can recursively express the prestige scores as

$$\mathbf{p}' = \mathbf{A}^T\mathbf{p}$$

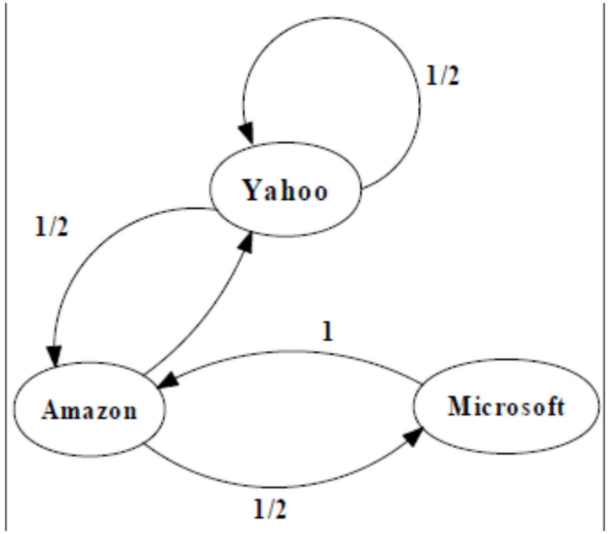
where $\mathbf{p}$ is an n-dimensional column vector corresponding to the prestige scores for each vertex.

Iterative Computation

$$
\begin{aligned}
\mathbf{p}_k &= \mathbf{A}^T \mathbf{p}_{k-1} \\
&= \mathbf{A}^T (\mathbf{A}^T \mathbf{p}_{k-2}) = (\mathbf{A}^T)^2 \mathbf{p}_{k-2} \\
&= (\mathbf{A}^T)^2 (\mathbf{A}^T \mathbf{p}_{k-3}) = (\mathbf{A}^T)^3 \mathbf{p}_{k-3} \\
&= \vdots \\
&= (\mathbf{A}^T)^k \mathbf{p}_0
\end{aligned}
$$

where $\mathbf{p}_0$ is the initial prestige vector. It is well known that the vector $\mathbf{p}_k$ converges to the dominant eigenvector of $\mathbf{A}^T$ with increasing k.

Example 1



$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

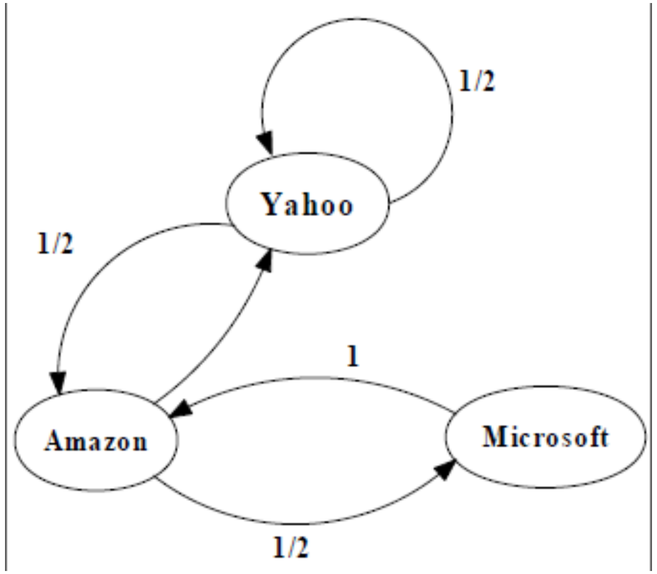
=the transpose of A (adjacency matrix)

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

PageRank Calculation: first iteration

Example 1



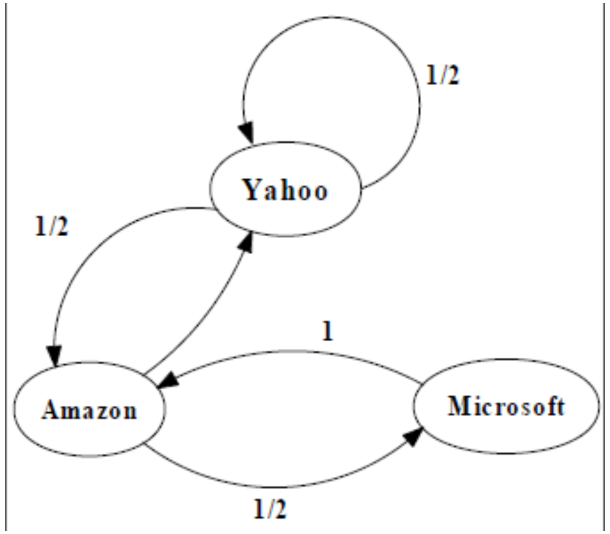
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 5/12 \\ 1/3 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix}$$

PageRank Calculation: second iteration

Example 1



$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 3/8 \\ 11/24 \\ 1/6 \end{bmatrix} \begin{bmatrix} 5/12 \\ 17/48 \\ 11/48 \end{bmatrix} \ldots \begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix}$$

Convergence after some iterations

A simple version

$$R(u) = \sum_{v \in B_u} \frac{R(v)}{N_v}$$

- u: a webpage
- B_u: the set of u's backlinks
- N_v: the number of forward links of page v

- Initially, R(u) is 1/N for every webpage
- Iteratively update each webpage's PR value until convergence.

A little more advanced version

- Adding a damping factor *d*
- Imagine that a surfer would stop clicking a hyperlink with probability 1-*d*

$$R(u) = \frac{(1-d)}{N-1} + d \sum_{v \in B_u} \frac{R(v)}{N_v}$$

- R(u) is at least (1-d)/(N-1)
  - N is the total number of nodes.

Other applications

- Social network (Facebook, Twitter, etc)
  - Node: Person; Edge: Follower / Followee / Friend
  - Higher PR value: Celebrity
- Citation network
  - Node: Paper; Edge: Citation
  - Higher PR values: Important Papers.
- Protein-protein interaction network
  - Node: Protein; Edge: Two proteins bind together
  - Higher PR values: Essential proteins.