Analytics and Visualization of Big Data

Fadel M. Megahed

Lecture 19: Link Analysis

AUBURN UNIVERSITY
SAMUEL GINN
COLLEGE OF ENGINEERING

Department of Industrial and Systems Engineering

Spring 13

Spring Break Refresher: Course Objectives

- Explain the basics behind the hardware and software needed for “big data” analytics.
- Analyze high-dimensional data.
- Develop visualizations that makes the data “sing”☺.
- Describe the components of successful search engines.
- Mine the web using structured and unstructured data.
- Train algorithms that can be used to extract new knowledge from data.

Spring Break Refresher: Analytics Based on Data Type

High dim. data	Graph data	Infinite data	Machine learning	Apps
Locality sensitive hashing	PageRank, SimRank	Filtering data streams	SVM	Recommender systems
Clustering	Community Detection	Web advertising	Decision Trees	Association Rules
Dimensionality reduction	Spam Detection	Queries on streams	Perceptron, kNN	Duplicate document detection

Early Search Engines

- There were many search engines before Google
  - Typically, based on the concept of an **inverted index**
  - With a **search query**, the old engines returned the results in an order that reflected the use of terms within a page

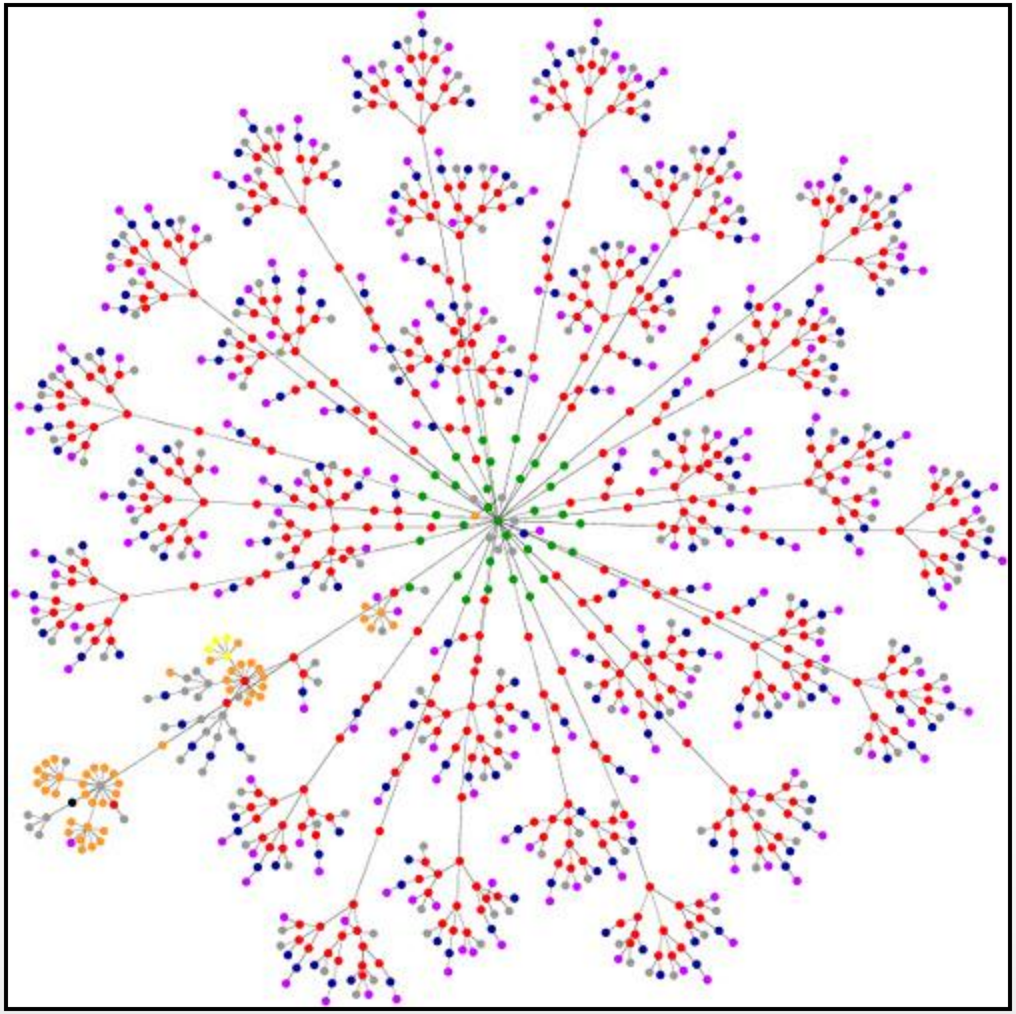
- It was easy to trick these search engines to believe that a page about *selling t-shirts* was actually about *movies* → **How?**

Early Search Engines Almost Made the Internet Useless

- As the internet gained popularity in the mid to late 1990s, it started to become so easy for **term spammers** to operate.

- To combat this, Larry Page and Sergey Brin came up with a simple (but genius idea)!!

Side Note – Structure of the Web



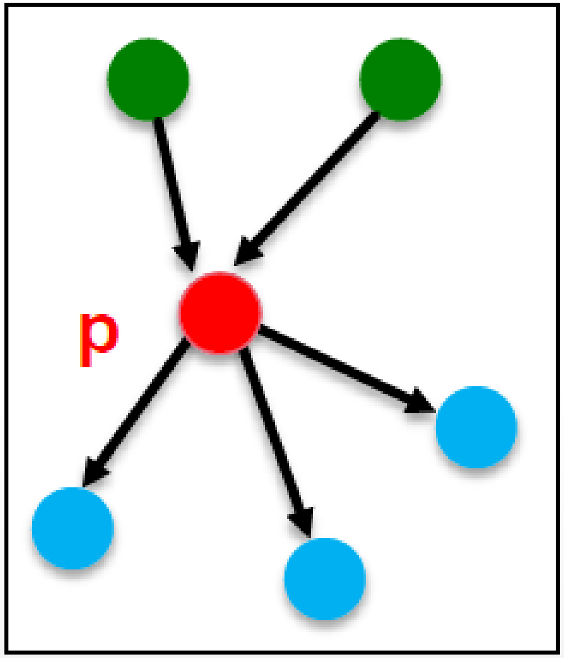
For more details, please read this one-page description: http://www.cs.cornell.edu/home/kleinber/sci01.pdf

The Brilliant Idea that Made Google: PageRank

- **Idea: Links as votes**
  - Page is more important if it has more links
    - In-coming links? Out-going links?

- **Think of in-links as votes**:
  - www.auburn.edu
  - www.joe-schmoe.com

- **Are all in-links are equal?**
  - Links from important pages count more
  - Recursive question!



What do we mean by recursive? (PageRank Cont.)

- Each link's vote is proportional to the **importance** of its source page
- If page ***p*** with importance ***x*** has ***n*** out-links, each link gets ***x/n*** votes
- Page ***p***'s own importance is the sum of the votes on its in-links

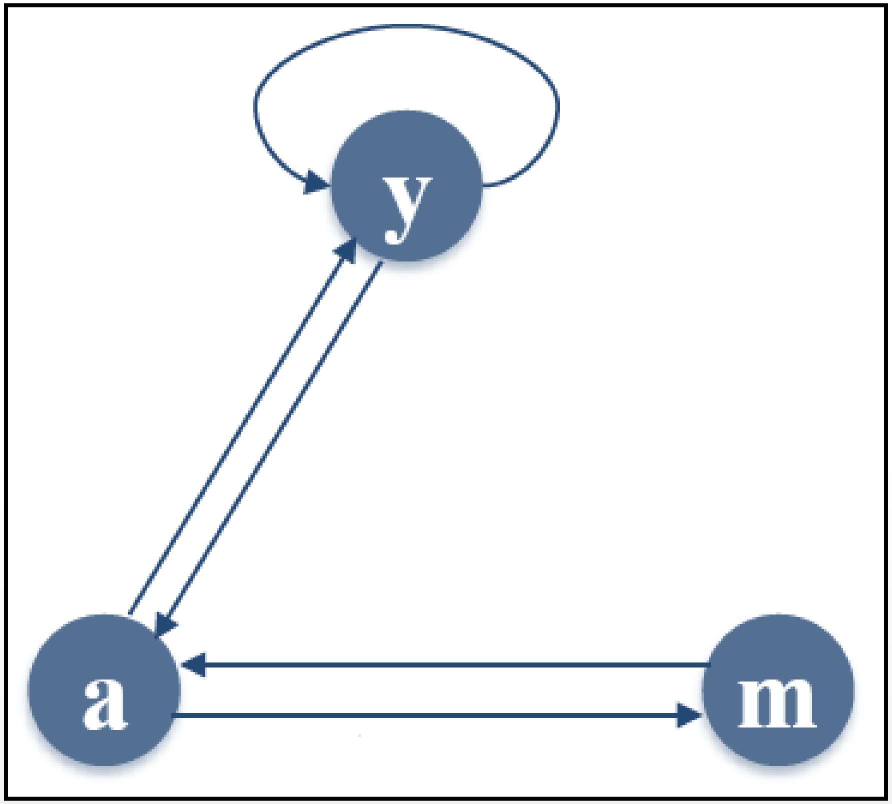
PageRank: The "Flow" Model

- A "vote" from an important page is worth more

- A page is important if it is pointed to by other important pages

- Define a "rank" r_j for node j

$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d}_{\mathrm{out}}(\mathrm{i})}$$



Pop Quiz (☺): Solving the Flow Equations

In teams of two, please solve for r_y, r_a, and r_m. (5 mins)

For distance students, please email me your answers (only if you are watching the class live)

Note that this quiz is not graded; it is only to assess your understanding so far ☺

Solving the Flow Equations

- **3 equations, 3 unknowns, no constants**
  - No unique solution
  - All solutions equivalent modulo scale factor
- **Additional constraint forces uniqueness**
  - $r_y + r_a + r_m = 1$
  - Solution: $r_y = 2/5$, $r_a = 2/5$, $r_m = 1/5$
- Gaussian elimination method works for small examples, but we need a better method for large web-size graphs

PageRank: From a Markov Chain Perspective

- **Stochastic adjacency matrix *M***
  - Let page j has d_j out-links
  - If $j \rightarrow i$, then $M_{ij} = 1/d_j$ else $M_{ij} = 0$
    - M is a **column stochastic matrix**
      - Columns sum to 1

- **Rank vector *r***: vector with an entry per page
  - r_i is the importance score of page i
  - $\sum_i r_i = 1$

- **The flow equations can be written (see book for proof)**

$$\underline{r} = \underline{M} * \underline{r}$$

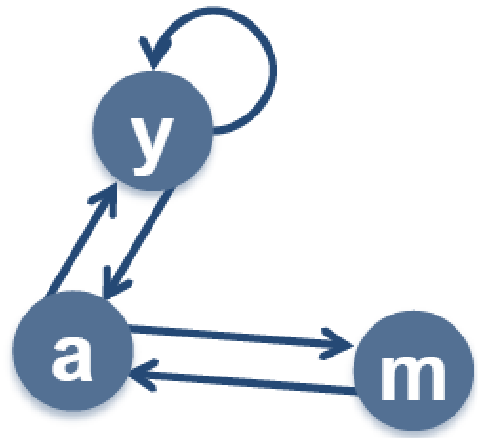
Eigenvector Formulation

- **The flow equations can be written**

$$\underline{r} = \underline{M}^{*}\underline{r}$$

- So the rank vector is an eigenvector of the stochastic web matrix
  - In fact, its first or principal eigenvector, with corresponding eigenvalue 1

Formulating the Previous Problem Using Markov Chains ☺



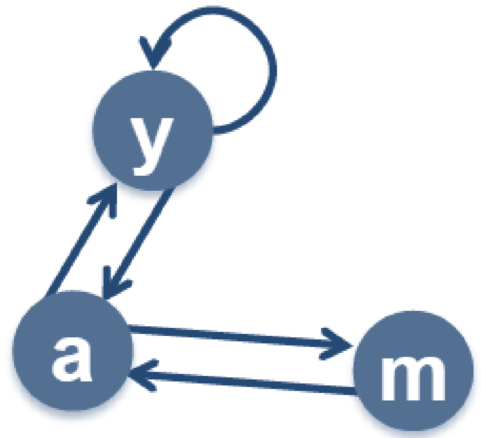
$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

Solution Approach 1: The Power Iteration Method

- **Given a web graph with n nodes, where the nodes are pages and edges are hyperlinks**

- **Power iteration:** a simple iterative scheme
  - **Suppose there are *N* web pages**
  - **Initialize: $r^{(0)} = [1/N,\ldots.,1/N]^T$**
  - **Iterate: $r^{(t+1)} = M \cdot r^{(t)}$**
  - **Stop when $|r^{(t+1)} - r^{(t)}|_1 < \varepsilon$**

Solution for the Example



$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

Solution for the Example: Using MATLAB

- To get the rank in MATLAB:
  - Define the Transition Probability Matrix (say we call it M)
  - [VectorMatrix, ValueMatrix]=eigs(M);
  - rankVector=VectorMatrix(:,1)

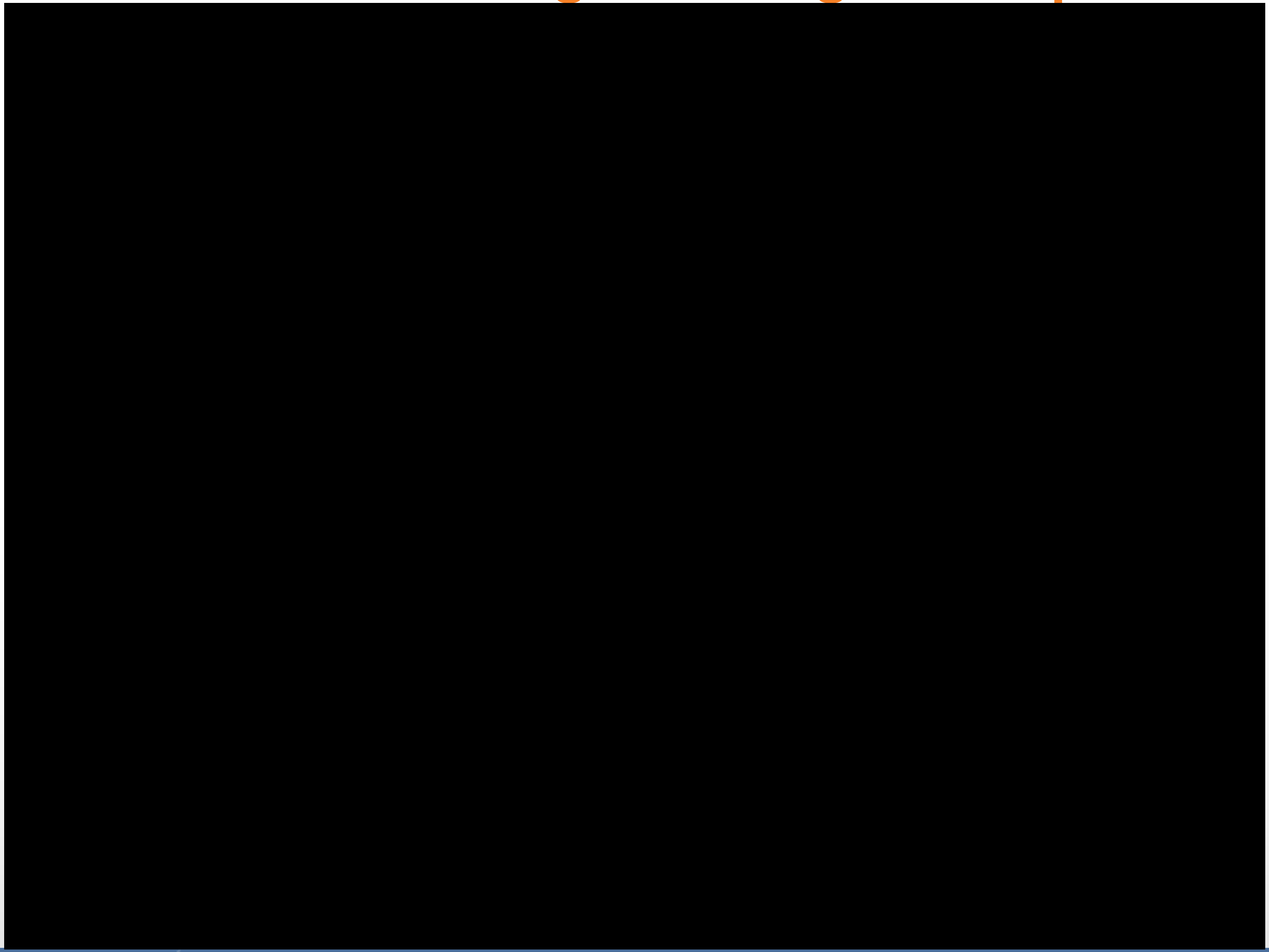
Since, r is the first/principal eigenvector, with a corresponding Eigenvalue of 1

Three Questions that Will be Addressed Next Class

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d}_\mathrm{i}} \quad \text{or equivalently} \quad r = Mr$$

1. Does this converge?
2. Does it converge to what we want?
3. Are results reasonable?

The Evolution of Search Engines: Google's Perspective



HW 04 – Deadline (3/26/2013 – Submit PDF to my Email)

- Watch the following videos:
  - http://www.youtube.com/watch?v=0v4v55OEZCQ (History of Internet Search and Google ~43 mins)
  - http://youtu.be/no3Cd0kG8uU (The Science of Search, ~ 5mins)

- In bullet points, identify the 10 main points in Video 1 and the 3 main points in Video 2.

- The Future of Search Series (Interesting perspective from 2007 , still valid, not part of the HW)
  - http://youtu.be/vst_Iombu0E (Yahoo's Perspective, Note the voice cuts out for a minute)
  - http://youtu.be/0zRUozxcOxo (Google's Perspective)
  - http://youtu.be/Nkl-rUCuNJk (Microsoft's Perspective)