General Link Analysis: PageRank and Its Relatives

Susan B. Davidson and Zachary G. Ives

University of Pennsylvania

CIS 545 – Big Data Analytics

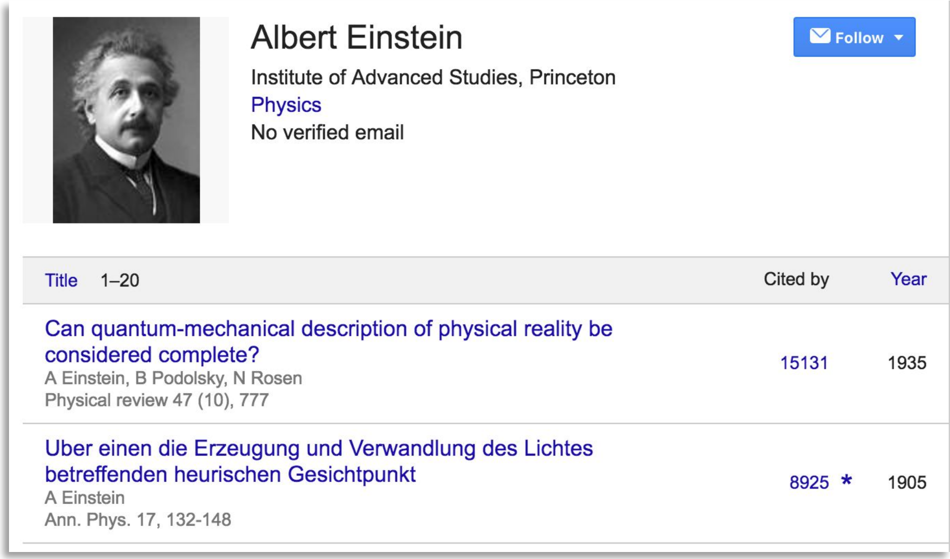
Penn Engineering

ODPi OpenDS4All

Portions of this lecture have been contributed to the OpenDS4All project, piloted by Penn, IBM, and the Linux Foundation

https://tinyurl.com/cis545-lecture-02-23-22

Measuring "Importance" in a Field



Albert Einstein

Institute of Advanced Studies, Princeton

Physics

No verified email

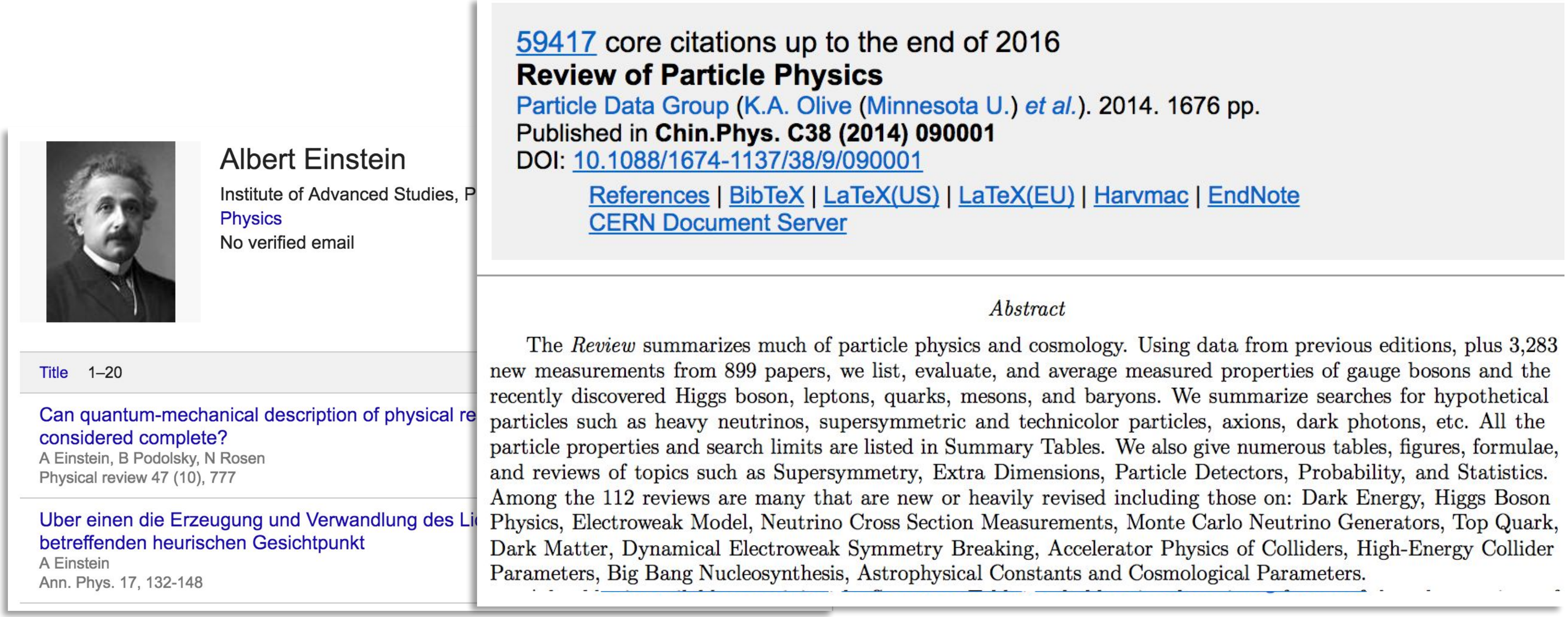
Title 1–20	Cited by	Year
Can quantum-mechanical description of physical reality be considered complete? A Einstein, B Podolsky, N Rosen Physical review 47 (10), 777	15131	1935
Uber einen die Erzeugung und Verwandlung des Lichtes betreffenden heurischen Gesichtpunkt A Einstein Ann. Phys. 17, 132-148	8925 *	1905

tells how well connected it

was an influential physicist?

https://tinyurl.com/cis-545-lecture-02-23-22

Measuring “Importance”



Albert Einstein

Institute of Advanced Studies, P

Physics

No verified email

Title 1–20

Can quantum-mechanical description of physical re considered complete?
A Einstein, B Podolsky, N Rosen
Physical review 47 (10), 777

Uber einen die Erzeugung und Verwandlung des Li betreffenden heurischen Gesichtpunkt
A Einstein
Ann. Phys. 17, 132-148

59417 core citations up to the end of 2016

Review of Particle Physics

Particle Data Group (K.A. Olive (Minnesota U.) *et al.*). 2014. 1676 pp.
Published in **Chin.Phys. C38 (2014) 090001**
DOI: 10.1088/1674-1137/38/9/090001

References | BibTeX | LaTeX(US) | LaTeX(EU) | Harvmac | EndNote
CERN Document Server

Abstract

The *Review* summarizes much of particle physics and cosmology. Using data from previous editions, plus 3,283 new measurements from 899 papers, we list, evaluate, and average measured properties of gauge bosons and the recently discovered Higgs boson, leptons, quarks, mesons, and baryons. We summarize searches for hypothetical particles such as heavy neutrinos, supersymmetric and technicolor particles, axions, dark photons, etc. All the particle properties and search limits are listed in Summary Tables. We also give numerous tables, figures, formulae, and reviews of topics such as Supersymmetry, Extra Dimensions, Particle Detectors, Probability, and Statistics. Among the 112 reviews are many that are new or heavily revised including those on: Dark Energy, Higgs Boson Physics, Electroweak Model, Neutrino Cross Section Measurements, Monte Carlo Neutrino Generators, Top Quark, Dark Matter, Dynamical Electroweak Symmetry Breaking, Accelerator Physics of Colliders, High-Energy Collider Parameters, Big Bang Nucleosynthesis, Astrophysical Constants and Cosmological Parameters.

https://tinyurl.com/cis-545-lecture-02-23-22

Measuring "Importance" in a Graph

Revisiting our discussion of measures of **centrality** –

- **eigenvector centrality** gives us a *recursive* measure of importance, i.e., do I connect to important nodes, do they connect to important nodes, etc.
  - In the Web graph, this is called **link analysis**

https://tinyurl.com/cis-545-lecture-02-23-22

Link Analysis for the Web

Suppose a search engine processes a query for “physics"

- Problem: Millions of pages contain these words!
- Which ones should we return first?

Idea: Hyperlinks encode a considerable amount of human judgment, much as citations do

- What does it mean when a web page links another page?
- Intra-domain links: Often created primarily for navigation
- Inter-domain links: Confer some measure of authority

It’s more than looking at the count of the links!

https://tinyurl.com/cis-545-lecture-02-23-22

Brief Review

https://canvas.upenn.edu/courses/1636888/quizzes/2771486 (11A)

Link analysis for the web defines a node's influence in terms of:

a. influence of a node's neighbors

b. direct neighbors

c. connecting paths

In the web, the links that matter for ranking are considered to be those:

d. between sites

e. within sites

f. between or within sites

https://tinyurl.com/cis-545-lecture-02-23-22

PageRank

Susan B. Davidson and Zachary G. Ives

University of Pennsylvania

CIS 545 – Big Data Analytics

Penn Engineering

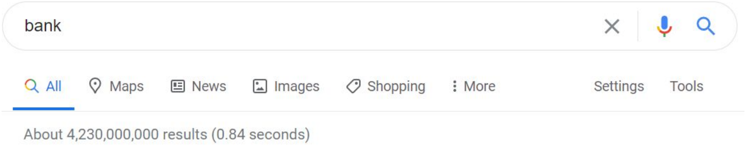
ODPi OpenDS4All

Portions of this lecture have been contributed to the OpenDS4All project, piloted by Penn, IBM, and the Linux Foundation

https://tinyurl.com/cis545-lecture-02-23-22

Web Search, B.G.

(Before Google)



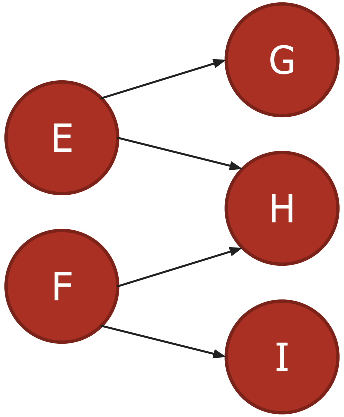
bank

About 4,230,000,000 results (0.84 seconds)

- 1990s Web search –matched keywords against documents, ranked by how unusual keywords were
  - Focus was on scale – how many millions of pages
  - Problem: too many hits, how do we differentiate?
- Needed better quality – Larry Page and Sergey Brin decided that we needed to know *importance* of a page!

https://tinyurl.com/cis-545-lecture-02-23-22

PageRank: Intuition

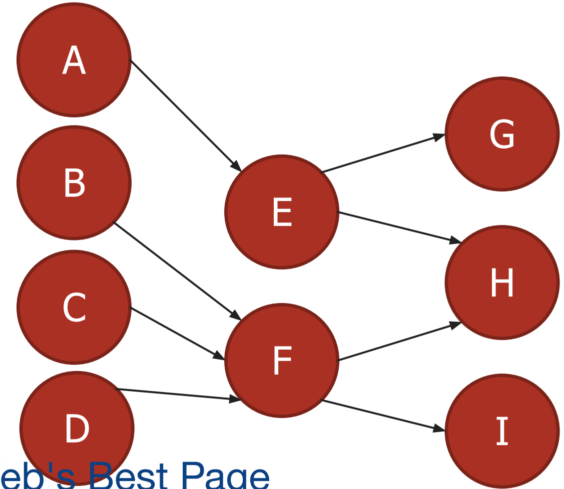


Imagine a contest for The Web's Best Page

- Initially, each page has one vote
- Each page votes for all the pages it has a link to
- To ensure fairness, pages voting for more than one page must split their vote equally between them

https://tinyurl.com/cis-545-lecture-02-23-22

PageRank: Intuition



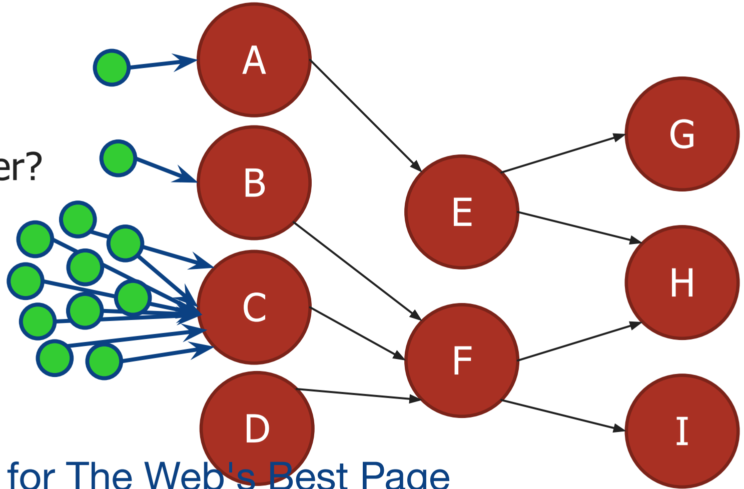
Imagine a contest for The Web's Best Page

- Initially, each page has one vote
- Each page votes for all the pages it has a link to
- To ensure fairness, pages voting for more than one page must split their vote equally between them

https://tinyurl.com/cis-545-lecture-02-23-22

PageRank: Intuition

How many levels should we consider?



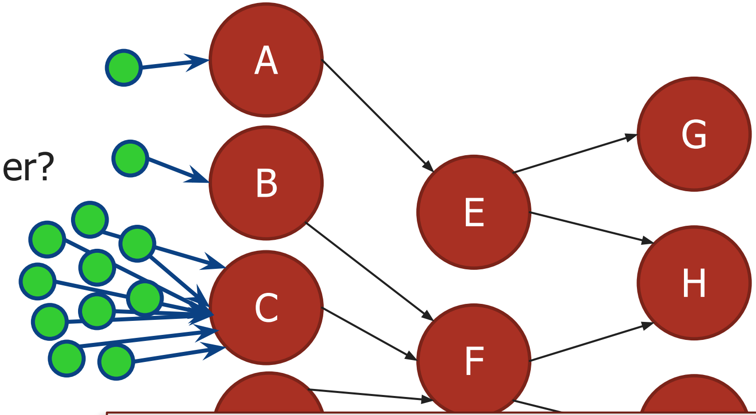
Imagine a contest for The Web's Best Page

- Initially, each page has one vote
- Each page votes for all the pages it has a link to
- To ensure fairness, pages voting for more than one page must split their vote equally between them

https://tinyurl.com/cis-545-lecture-02-23-22

PageRank: Intuition

How many levels
should we consider?



- Voting proceeds in rounds
- In each round, each page has the number of votes it received in the previous round

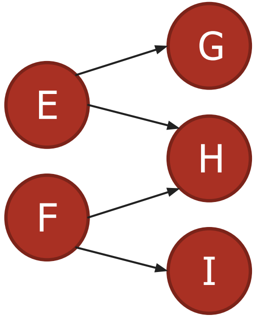
Imagine a contest for T

- Initially, each page
- Each page votes for all the pages it has a link to
- To ensure fairness, pages voting for more than one page must split their vote equally between them

https://tinyurl.com/cis-545-lecture-02-23-22

Simplified Version of PageRank

- Each page *x* is given a rank PageRank(*x*)
- Goal: Assign the PageRank(*x*) such that the rank of each page is governed by the **ranks of the pages linking to it**:



Rank of page x

Every page j that links to x

Number of links out from page j

Rank of page j

How do we compute the rank values?

https://tinyurl.com/cis-545-lecture-02-23-22

Other Applications of the Same Idea

This question occurs in several other areas:

- How do we measure the "impact" of a researcher? (#papers? #citations?)
- What are the most useful datasets? (# downloads?)
- Who are the most "influential" individuals in a social network? (#friends?)
- Which programmers are writing the "best" code? (#uses?)

https://tinyurl.com/cis-545-lecture-02-23-22

PageRank Is a Recursive Measure of Importance

- Influential pages link to influential pages

- Two important properties:
  - It converges!
  - It can be computed independently of the query

- Caveat: query independence means it only looks at **structure**!

https://tinyurl.com/cis-545-lecture-02-23-22

Brief Review

https://canvas.upenn.edu/courses/1636888/quizzes/2771488 (11B)

The PageRank of a node, as it "flows" out from a node, is:

a. assigned in full to each out-link
b. split across out-links in order of importance
c. not considered
d. split uniformly across all out-links

What properties of PageRank are important for web search applications:

e. independence of the query and guaranteed convergence
f. speed
g. self-importance
h. ability to take query semantics into account

https://tinyurl.com/cis-545-lecture-02-23-22

A Basic PageRank Implementation

Susan B. Davidson and Zachary G. Ives

University of Pennsylvania

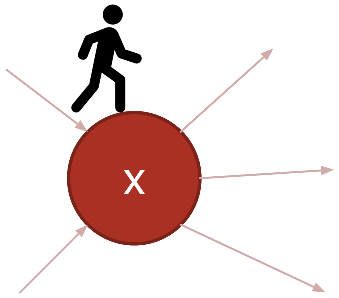
CIS 545 – Big Data Analytics

Portions of this lecture have been contributed to the OpenDS4All project, piloted by Penn, IBM, and the Linux Foundation

https://tinyurl.com/cis545-lecture-02-23-22

Restating the Simplified Model as a Browsing Experience

- Let’s imagine we “randomly walk” through a graph of nodes and connections
- At each point, we look at the out-edges and follow each with equal probability



https://tinyurl.com/cis-545-lecture-02-23-22

Iterative PageRank (simplified)

Initialize all ranks:

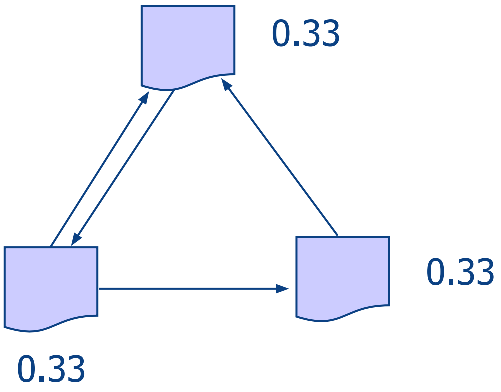
Iterate until convergence

No need to decide how many iterations to consider!

https://tinyurl.com/cis-545-lecture-02-23-22

Example: Step 0

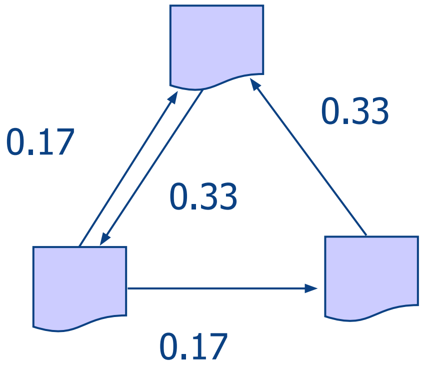
Initialize all ranks



https://tinyurl.com/cis-545-lecture-02-23-22

Example: Step 1

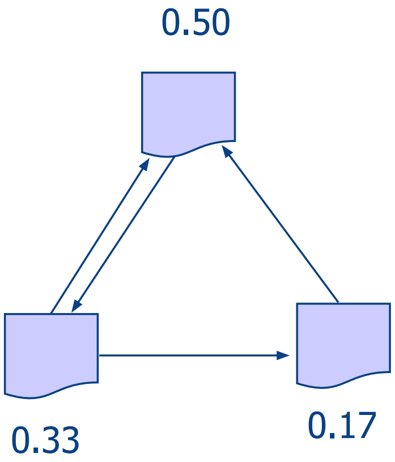
Propagate weights across out-edges



https://tinyurl.com/cis-545-lecture-02-23-22

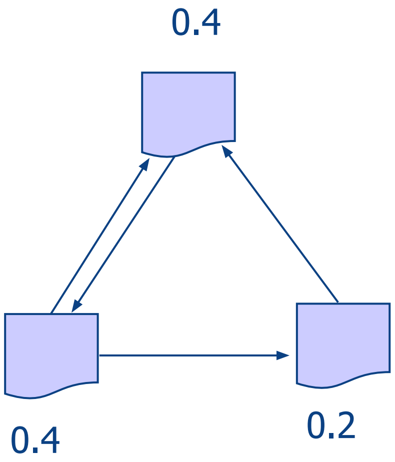
Example: Step 2

Compute weights based on in-edges



https://tinyurl.com/cis-545-lecture-02-23-22

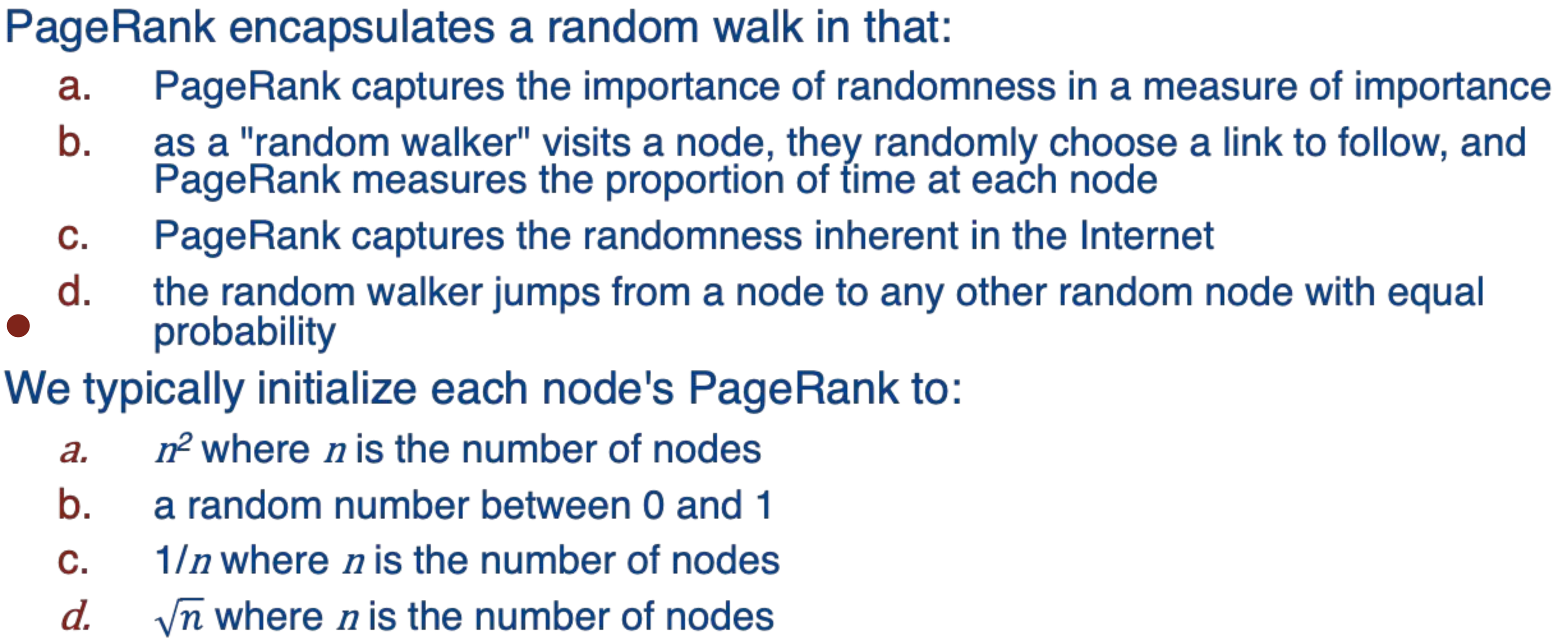
Example: Convergence



https://tinyurl.com/cis-545-lecture-02-23-22

Brief Review

https://canvas.upenn.edu/courses/1636888/quizzes/2771491 (11D)



PageRank encapsulates a random walk in that:

a. PageRank captures the importance of randomness in a measure of importance

b. as a "random walker" visits a node, they randomly choose a link to follow, and PageRank measures the proportion of time at each node

c. PageRank captures the randomness inherent in the Internet

d. the random walker jumps from a node to any other random node with equal probability

We typically initialize each node's PageRank to:

a. n^2 where n is the number of nodes

b. a random number between 0 and 1

c. $1/n$ where n is the number of nodes

d. $\sqrt{n}$ where n is the number of nodes

https://tinyurl.com/cis-545-lecture-02-23-22

Great! We Can Compute PageRank Iteratively

- e.g., using the recursive join computations we saw for Spark
  - “Propagate” a node’s weight along its edges
  - Sum up incoming edge weights
- But some pieces of this can be thought of in a more general way…

https://tinyurl.com/cis-545-lecture-02-23-22

PageRank Using Matrices

Susan B. Davidson and Zachary G. Ives

University of Pennsylvania

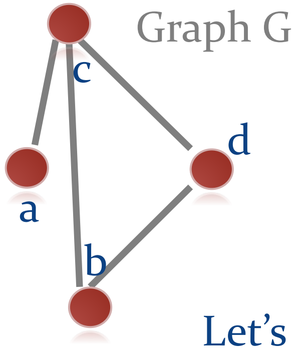
CIS 545 – Big Data Analytics

Portions of this lecture have been contributed to the OpenDS4All project, piloted by Penn, IBM, and the Linux Foundation

https://tinyurl.com/cis545-lecture-02-23-22

Graphs and Adjacency Matrices

Recall that we can use an **adjacency matrix** to describe connectivity

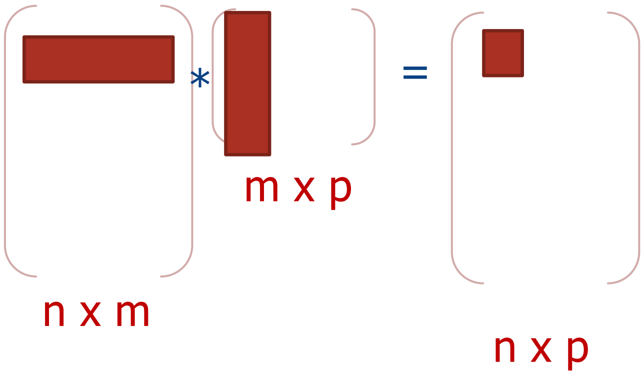


$$\begin{array}{ccccc} & a & b & c & d \\ a & 0 & 0 & 1 & 0 \\ b & 0 & 0 & 1 & 1 \\ c & 1 & 1 & 0 & 1 \\ d & 0 & 1 & 1 & 0 \end{array}$$

Let's generalize from this idea, adding **direction** and **weight** to the edges…

https://tinyurl.com/cis-545-lecture-02-23-22

Brief Refresher on Matrix Multiplication



Bulk operation, much like those over relations!

```
import numpy
mat1 = np.zeros((n,m))
mat2 = np.array([1,0, …], […], …])
mat3 = np.dot(mat1,mat2)
```

https://tinyurl.com/cis-545-lecture-02-23-22

PageRank Linear Algebra formulation

Create an m x m “weight transfer matrix” M to capture links:

M(i, j) = 1 / n$_j$ if page i **is pointed to** by page j and page j has **n$_j$ outgoing links**

= 0 otherwise

Initialize all PageRanks to 1, multiply by M repeatedly until all values converge:

$$\begin{bmatrix} PageRank(p_1') \\ PageRank(p_2') \\ \ldots \\ PageRank(p_m') \end{bmatrix} = M \begin{bmatrix} PageRank(p_1) \\ PageRank(p_2) \\ \ldots \\ PageRank(p_m) \end{bmatrix}$$

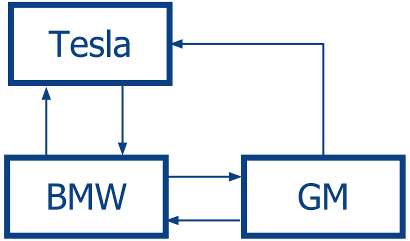
Iterative computation:

$$PageRank^{(i)} = M \cdot PageRank^{(i-1)}$$

Computes principal eigenvector via power iteration

https://tinyurl.com/cis-545-lecture-02-23-22

A Brief Example



$$\begin{vmatrix} t' \\ g' \\ b' \end{vmatrix} = \begin{array}{ccc} & \text{GM} & \\ \boxed{\begin{matrix} 0 \\ 0 \\ 1 \end{matrix}} & \boxed{\begin{matrix} 0.5 \\ 0 \\ 0.5 \end{matrix}} & \begin{matrix} 0.5 \\ 0.5 \\ 0 \end{matrix} \\ \text{Tesla} & & \end{array} * \begin{vmatrix} t \\ g \\ b \end{vmatrix}$$

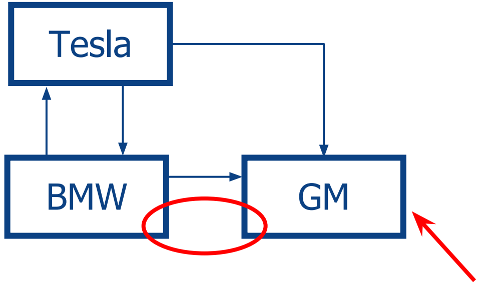
Running for multiple iterations:

$$\begin{vmatrix} t \\ g \\ b \end{vmatrix} = \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix}, \begin{vmatrix} 1 \\ 0.5 \\ 1.5 \end{vmatrix}, \begin{vmatrix} 1 \\ 0.75 \\ 1.25 \end{vmatrix}, \ldots \begin{vmatrix} 1 \\ 0.67 \\ 1.33 \end{vmatrix}$$

Total rank sums to number of pages

https://tinyurl.com/cis-545-lecture-02-23-22

Oops #1
– PageRank *sinks*



'dead end' - PageRank
is lost after each round

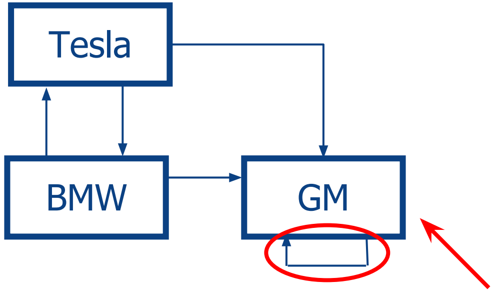
$$\begin{vmatrix} t' \\ g' \\ b' \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0 \end{vmatrix} * \begin{vmatrix} t \\ g \\ b \end{vmatrix}$$

Running for multiple iterations:

$$\begin{vmatrix} t \\ g \\ b \end{vmatrix} = \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix}, \begin{vmatrix} 0.5 \\ 1 \\ 0.5 \end{vmatrix}, \begin{vmatrix} 0.25 \\ 0.5 \\ 0.25 \end{vmatrix}, \ldots, \begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix}$$

https://tinyurl.com/cis-545-lecture-02-23-22

Oops #2
– PageRank hogs



PageRank cannot flow out and accumulates

$$\begin{vmatrix} t' \\ g' \\ b' \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0 & 0 \end{vmatrix} * \begin{vmatrix} t \\ g \\ b \end{vmatrix}$$

Running for multiple iterations:

$$\begin{vmatrix} t \\ g \\ b \end{vmatrix} = \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix}, \begin{vmatrix} 0.5 \\ 2 \\ 0.5 \end{vmatrix}, \begin{vmatrix} 0.25 \\ 2.5 \\ 0.25 \end{vmatrix}, \ldots \begin{vmatrix} 0 \\ 3 \\ 0 \end{vmatrix}$$

https://tinyurl.com/cis-545-lecture-02-23-22

Reducing Rank Hogs and Dead-Ends

- Remove **out-degree 0 nodes** (or consider them to refer back to referrer)
- Add damping or decay factor α to deal with sinks

$$PageRank^{(i)} = \alpha \cdot M \cdot PageRank^{(i-1)} + \beta$$

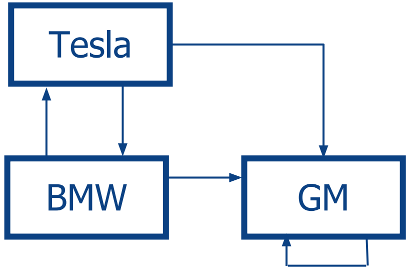
- Typical values: α =0.85 β = *m* element vector with values 1 - α = 0.15

Non-matrix form is

$$PageRank^{(i)}(x) = \alpha \sum_{j \in B(x)} \frac{1}{N_j} PageRank^{(i-1)}(j) + \beta$$

https://tinyurl.com/cis-545-lecture-02-23-22

Example: Reducing the Hog



$$\begin{vmatrix} t' \\ g' \\ b' \end{vmatrix} = \overset{\alpha}{0.85} \begin{vmatrix} 0 & 0 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0 & 0 \end{vmatrix} * \begin{vmatrix} t \\ g \\ b \end{vmatrix} + \overset{\beta}{\begin{vmatrix} 0.15 \\ 0.15 \\ 0.15 \end{vmatrix}}$$

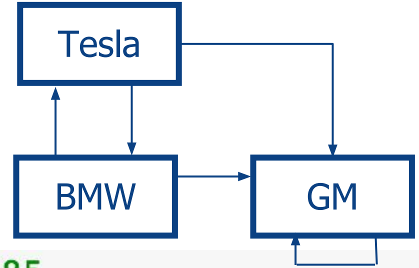
Running for multiple iterations:

$$\begin{vmatrix} t \\ g \\ b \end{vmatrix} = \begin{vmatrix} 0.57 \\ 1.85 \\ 0.57 \end{vmatrix}, \begin{vmatrix} 0.39 \\ 2.21 \\ 0.39 \end{vmatrix}, \begin{vmatrix} 0.32 \\ 2.36 \\ 0.32 \end{vmatrix}, \cdots, \begin{vmatrix} 0.26 \\ 2.48 \\ 0.26 \end{vmatrix}$$

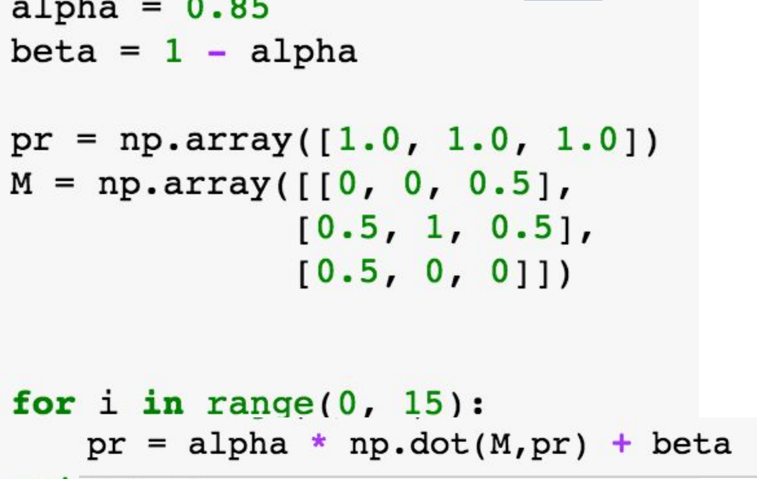
https://tinyurl.com/cis-545-lecture-02-23-22

… though does this seem right?

PageRank Example in Python



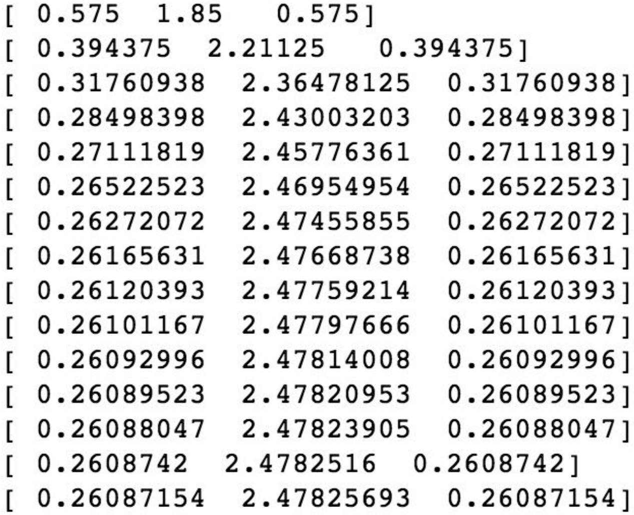
$$\begin{vmatrix} t' \\ g' \\ b' \end{vmatrix} = 0.85 \begin{vmatrix} 0 & 0 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0 & 0 \end{vmatrix} * \begin{vmatrix} t \\ g \\ b \end{vmatrix} + \begin{vmatrix} 0.15 \\ 0.15 \\ 0.15 \end{vmatrix}$$



```
alpha = 0.85
beta = 1 - alpha

pr = np.array([1.0, 1.0, 1.0])
M = np.array([[0, 0, 0.5],
              [0.5, 1, 0.5],
              [0.5, 0, 0]])


for i in range(0, 15):
    pr = alpha * np.dot(M,pr) + beta
```



```
[ 0.575   1.85    0.575]
[ 0.394375  2.21125   0.394375]
[ 0.31760938  2.36478125  0.31760938]
[ 0.28498398  2.43003203  0.28498398]
[ 0.27111819  2.45776361  0.27111819]
[ 0.26522523  2.46954954  0.26522523]
[ 0.26272072  2.47455855  0.26272072]
[ 0.26165631  2.47668738  0.26165631]
[ 0.26120393  2.47759214  0.26120393]
[ 0.26101167  2.47797666  0.26101167]
[ 0.26092996  2.47814008  0.26092996]
[ 0.26089523  2.47820953  0.26089523]
[ 0.26088047  2.47823905  0.26088047]
[ 0.2608742  2.4782516  0.2608742]
[ 0.26087154  2.47825693  0.26087154]
```

https://tinyurl.com/cis-545-lecture-02-23-22

Intuition Behind PageRank: Random Surfer Model

- PageRank has an intuitive basis in **random walks on graphs**

- Imagine a random surfer, who starts on a **random page with equal probability** and, in each step,
  - **with probability α**, clicks on a random link on the page
  - **with probability β = 1 - α**, jumps to a random page (bored?)

- The PageRank of a page can be interpreted as the fraction of steps the surfer spends on the corresponding page
  - Transition matrix can be interpreted as a Markov Chain

https://tinyurl.com/cis-545-lecture-02-23-22

Brief Review

https://canvas.upenn.edu/courses/1636888/quizzes/2771546 (11E)

The PageRank weight transfer matrix initializes each M[i,j] to be:

a. $1/N_i$ if j points to i, where N_i is the number of out-edges from i

b. $1/N_j$ if i points to j, where N_j is the number of out-edges from j

c. $1/n$

d. $1/N_j$ if j points to i, where N_j is the number of out-edges from j

The *decay factor* α, as defined in the slides, can be considered to be:

e. the proportion of time the user randomly jumps at uniform to another page

f. the proportion of pages that are important

g. the proportion of PageRank that is important

h. the proportion of the time the user traverses a link from the page

https://tinyurl.com/cis-545-lecture-02-23-22

Variations on PageRank

- Many have been studied!

- What if we don’t randomly jump with equal probability?
  - *Personalized PageRank* makes the Beta term non-uniform

- What if we want to “personalize” PageRank or measure it relative to certain start points?
  - *Label propagation* starts at labeled nodes, estimates how often we end up at a destination if we randomly walked from each labeled node

Recap and Take-aways

- Link analysis schemes are much more sophisticated than direct methods of measuring centrality
- They look at *structure* to understand importance
- They don't look at semantics – so they are vulnerable to manipulation of structure!

https://tinyurl.com/cis-545-lecture-02-23-22