

Social Networks and Ranking

1

Generalized Social Networks

- Represent relationship between entities
 - paper cites paper
 - html page links to html page
 - A supervises B

} directed graph

- A and B are friends
- papers share an author
- A and B are co-workers

} undirected graph

2

Hypertext

- document or part of document links to other parts or other documents
 - construct documents of interrelated pieces
 - relate documents to each other
- pre-dates Web
- Web “killer app.”

3

How use links to improve information search?

- use **structure** to **compute score** for ranking
- **include more objects** to rank
 - redefines “satisfying” of query?
- **add** to the **content** of a document

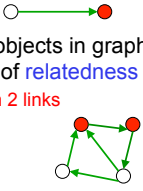
✧ can deal with objects of mixed types

- images, PDF, ...

4

Scoring using structure

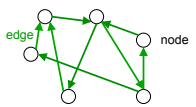
- Ideas
 1. **link to** object suggests it **valuable** object
 2. **distance** between objects in graph represents degree of **relatedness**
 reachable by all in 2 links



5

Pursuing linking and value

- **Intuition:** when Web page **points** to another Web page, it **confers status/authority/popularity** to that page
- Find a measure that **captures intuition**



- Not just web linking
 - Citations in books, articles
 - Doctors referring to other doctors

6

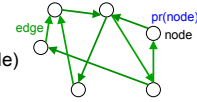
Indegree

- Indegree = number of links into a node
- Most obvious idea:
 - higher indegree => better node
- Doesn't work well
- Need some feedback in system
- Leads us to Page and Brin's PageRank

7

PageRank

- Algorithm that gave Google the **leap in quality**
 - link structure centerpiece of scoring
- Framework
 - Given a directed graph with n nodes
 - Assign each node a score that represents its importance in structure: PageRank: $pr(\text{node})$



8

Conferring importance

Core ideas:

- A node should confer some of its importance to the nodes to which it points
 - If a node is important, the nodes it links to should be important
- A node should not transfer more importance than it has

9

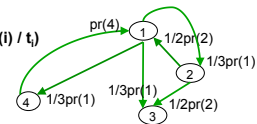
Attempt 1

Refer to nodes by numbers $1, \dots, n$ (arbitrary numbering)
 Let t_i denote the number of edges out of node i (outdegree)
 Node i transfers $1/t_i$ of its importance on each edge out of it

Define

$$pr_{\text{new}}(\mathbf{k}) = \sum_{i \text{ with edge from } i \text{ to } k} (pr(i) / t_i)$$

Iterate until converges



Problems

- Sinks (nodes with no edges out)
- Cyclic behavior

10

Attempt 2

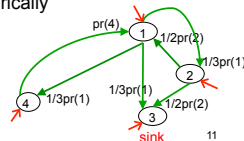
Random walk model

- Attempt 1 gives movement from node to linked neighbor with probability $1/\text{outdegree}$
- Add random jump to any node

$$pr_{\text{new}}(\mathbf{k}) = \alpha/n + (1-\alpha) \sum_{i \text{ with edge from } i \text{ to } k} (pr(i) / t_i)$$

– α parameter chosen empirically

- Break cycles
- Escape from sinks



11

Normalized?

- Would like $\sum_{1 \leq k \leq n} (pr(\mathbf{k})) = 1$
- Consider $\sum_{1 \leq k \leq n} (pr_{\text{new}}(\mathbf{k}))$

$$\begin{aligned} &= \sum_{1 \leq k \leq n} (\alpha/n + (1-\alpha) \sum_{i \text{ with edge from } i \text{ to } k} (pr(i) / t_i)) \\ &= \sum_{1 \leq k \leq n} (\alpha/n) + \sum_{1 \leq k \leq n} ((1-\alpha) \sum_{i \text{ with edge from } i \text{ to } k} (pr(i) / t_i)) \\ &= \alpha + (1-\alpha) \sum_{1 \leq k \leq n} \sum_{i \text{ with edge from } i \text{ to } k} (pr(i) / t_i) \\ &= \alpha + (1-\alpha) \sum_{1 \leq i \leq n} \sum_{k \text{ with edge from } i \text{ to } k} (pr(i) / t_i) \\ &= \alpha + (1-\alpha) \sum_{i \text{ with edge from } i} pr(i) \end{aligned}$$

*inner sum \sum_i over incoming edges for one k

*inner sum \sum_k over outgoing edges for one i



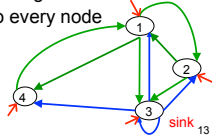
12

Problem for desired normalization

- Have $\sum_{1 \leq k \leq n} (\mathbf{pr}_{\text{new}}(\mathbf{k})) = \alpha + (1-\alpha) \sum_{i \text{ with edge from } i} \mathbf{pr}(i)$
- **Missing $\mathbf{pr}(i)$** for nodes with no edges from them
 - sinks!
- **Solution:** add n edges out of every sink
 - Edge to every node including self
 - Gives $1/n$ contribution to every node

Gives desired normalization:

If $\sum_{1 \leq k \leq n} (\mathbf{pr}_{\text{initial}}(\mathbf{k})) = 1$
 then $\sum_{1 \leq k \leq n} (\mathbf{pr}(\mathbf{k})) = 1$



Matrix formulation

- Let E be the n by n adjacency matrix
 $E(i,k) = 1$ if there is an edge from node i to node k
 $= 0$ otherwise
- Define **new matrix L** :
 For each row i of E ($1 \leq i \leq n$)
 If row i contains $t_i > 0$ ones, $L(i,k) = (1/t_i) E(i,k)$, $1 \leq k \leq n$
 If row i contains 0 ones, $L(i,k) = 1/n$, $1 \leq k \leq n$
- Vector \mathbf{pr} of PageRank values defined by
 $\mathbf{pr} = (\alpha/n, \alpha/n, \dots, \alpha/n)^T + (1-\alpha) L^T \mathbf{pr}$
- has a solution representing the **steady-state values $\mathbf{pr}(\mathbf{k})$**

14

Calculation

- Choose α
 - No single best value
 - Page and Brin originally used $\alpha = .15$
- Simple iterative calculation
 - Initialize $\mathbf{pr}_{\text{initial}}(\mathbf{k}) = 1/n$ for each node k
 so $\sum_{1 \leq k \leq n} (\mathbf{pr}_{\text{initial}}(\mathbf{k})) = 1$
 - $\mathbf{pr}_{\text{new}}(\mathbf{k}) = \alpha/n + (1-\alpha) \sum_{1 \leq i \leq n} L(i,k) \mathbf{pr}(i)$
- Converges
 - Has necessary mathematical properties
 - In practice, choose convergence criterion
 - Stops iteration

15

PageRank Observations

- Can be calculated for *any* directed graph
- Google calculates on **entire Web graph**
 - **query independent** scoring
- Huge calculation for Web graph
 - precomputed
 - 1998 Google published:
 - 52 iterations for 322 million links
 - 45 iterations for 161 million links
- PageRank must be combined with query-based scoring for final ranking
 - Many variations
 - What Google exactly does secret
 - Can make some guesses by results

16

HITS

Hyperlink Induced Topic Search

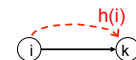
- Second well-known algorithm
- By Jon Kleinberg while at IBM Almaden Research Center
- Same general goal as PageRank
- Distinguishes **2 kinds of nodes**
 - **Hubs:** resource pages
 - **Point to many authorities**
 - **Authorities:** good information pages
 - **Pointed to by many hubs**

17

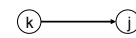
Mutual reinforcement

- Authority weight node j : $\mathbf{a}(j)$
 - Vector of weights \mathbf{a}
- Hub weight node j : $\mathbf{h}(j)$
 - Vector of weights \mathbf{h}
- Update:

$$\mathbf{a}_{\text{new}}(\mathbf{k}) = \sum_{i \text{ with edge from } i \text{ to } k} (\mathbf{h}(i))$$



$$\mathbf{h}_{\text{new}}(\mathbf{k}) = \sum_{j \text{ with edge from } k \text{ to } j} (\mathbf{a}(j))$$



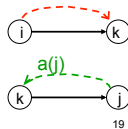
18

Mutual reinforcement

- Authority weight node j : $a(j)$
 - Vector of weights \mathbf{a}
- Hub weight node j : $h(j)$
 - Vector of weights \mathbf{h}
- Update:

$$\mathbf{a}_{\text{new}}(k) = \sum_{i \text{ with edge from } i \text{ to } k} (h(i))$$

$$\mathbf{h}_{\text{new}}(k) = \sum_{j \text{ with edge from } k \text{ to } j} (a(j))$$



19

Matrix formulation

Steady state:

$$\begin{aligned} \mathbf{a} &= \mathbf{E}^T \mathbf{h} & \mathbf{a} &= \mathbf{E}^T \mathbf{E} \mathbf{a} \\ \mathbf{h} &= \mathbf{E} \mathbf{a} & \mathbf{h} &= \mathbf{E} \mathbf{E}^T \mathbf{h} \end{aligned}$$

Interpretation:

- $\mathbf{E}^T \mathbf{E}(i,j)$: number nodes **point to** both node i and node j
 - “Co-citation”
- $\mathbf{E} \mathbf{E}^T(i,j)$: number nodes **pointed to by** both node i and node j
 - “Bibliographic coupling”

20

Iterative Calculation

$$\mathbf{a} = \mathbf{h} = (1, \dots, 1)^T$$

While (not converged) {

$$\mathbf{a}_{\text{new}} = \mathbf{E}^T \mathbf{h}$$

$$\mathbf{h}_{\text{new}} = \mathbf{E} \mathbf{a}$$

$$\mathbf{a} = \mathbf{a}_{\text{new}} / \|\mathbf{a}_{\text{new}}\| \quad \text{normalize to unit vector}$$

$$\mathbf{h} = \mathbf{h}_{\text{new}} / \|\mathbf{h}_{\text{new}}\| \quad \text{normalize to unit vector}$$

}

Provable convergence by linear algebra

21

Use of HITS

original use **after** find Web pages satisfying query:

1. Retrieve documents satisfy query and **rank by term-based** techniques
2. Keep **top c documents**: root set of nodes
 - c a chosen constant - tunable
3. Make base set:
 - a) Root set
 - b) **Plus nodes pointed to by** nodes of **root set**
 - c) **Plus nodes pointing to** nodes of **root set**
4. Make base graph: base set plus edges from Web graph between these nodes
5. Apply HITS to base graph

using links to expand matches!

22

Results using HITS

- Documents ranked by authority score $a(\text{doc})$ and hub score $h(\text{doc})$
 - Authority score primary score for search results
- Heuristics:
 - delete all links between pages in same domain
 - Keep only pre-determined number of pages linking into root set (~200)
- Findings (original paper)
 - Number iterations in original tests ~50
 - most authoritative pages **do not** contain initial query terms

23

Observations

- HITS can be applied to any directed graph
- Base graph **much smaller** than Web graph
- Kleinberg identified bad phenomena
 - Topic diffusion: generalizes topic when expand root graph to base graph
 - example: want *compilers* - generalized to *programming*

24

Revisit: How use links in ranking documents?

- use **structure** to compute score for ranking
 - PageRank, HITS
- include **more objects** to rank
 - saw in use of HITS
- use **anchor text** (HTML)
 - anchor text **labels link**
 - include anchor text as text of **document pointed to**

25

Using anchor text

“homework” may not occur in *content* of doc b

terms in doc b for building index:

- homework: anchor
- problem: title 1
- set: title 2

26

Summary

- Link analysis
 - a principal component of ranking by modern Web search engines
 - must be combined with content analysis
- Extend document content with link info
 - anchor text
 - text of URLs
 - e.g. princeton.edu, aardvarksportsshop.com
- Expand set of satisfying docs using links
 - less often used

27

Ranking documents w.r.t. query

28

General Framework

- Have set of **n** features (aka **signals**) to use in **determining ranking** score
 - Features depend on query:
 - vector $\Psi(d, q)$ of feature values f_i for doc d , query q
 - eg tf.idf score is feature
 - Features are conditioned to be comparable
- Have **parameterized function** to **combine** signals
 - simple: linear $\alpha_0 + \sum_{i=1}^n \alpha_i * (f_i)$
 - α_i are adjustable weights - **how choose?**
 - intuition
 - experimentation
 - machine learning

29

Machine Learning

Many possibilities – overview of one **Ordinal Regression Model**

- Goal: get **comparison** of doc.s correct
- capture goal
 - Let ω represent vector $(\alpha_1, \dots, \alpha_n)$
 - want $\omega^T \cdot \Psi(d_i, q) - \omega^T \cdot \Psi(d_j, q) > 0$ if and only if d_i more relevant than d_j for query q
 - find ω that works
- techniques **train** on known correct data:
 - **humans rank** a set of documents for various queries

30