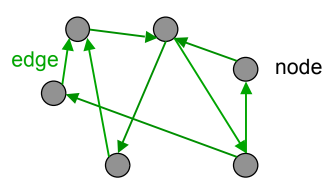
Link-based ranking
Part 2

1

Goal

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

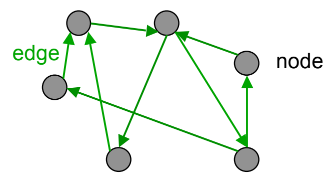


- Not just web linking
  - Citations in books, articles
  - Doctors referring to other doctors

2

Review: first measure PageRank

- Given a directed graph with *n* nodes
- Assign each node a score that represents its importance in structure
  - Call score PageRank: pr(node)



3

Conferring importance

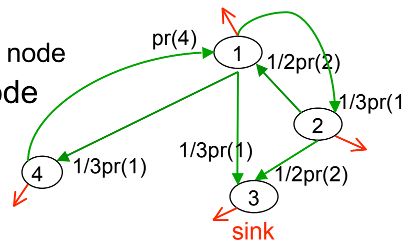
Core ideas:

- 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
- Address problems with:
  - Sinks (nodes with no edges out)
  - Cyclic behavior

4

Random walk model (review)

1. Move from node to linked neighbor with probability 1/outdegree
   Outdegree of a node
   = number of edges out of a node
2. Randomly jump to any node
   - Break cycles
   - Escape from sinks



Captured with:

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

- α parameter chosen empirically
- $\mathbf{t_i}$ outdegree of node i

Steady state probability of being at a node = pr(node)

5

Normalized?

- Would like $\mathbf{\Sigma_{1 \le k \le n}\,(pr\,(k)) = 1}$
- Consider $\mathbf{\Sigma_{1 \le k \le n}\,(pr_{new}(k))}$

$= \Sigma_{1 \le k \le n}(\ \alpha/n + (1\text{-}\alpha)\Sigma_{i \text{ with edge from } i \text{ to } k}(pr(i) / t_i)\ )$

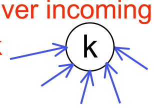
$= \Sigma_{1 \le k \le n}(\ \alpha/n)+\Sigma_{1 \le k \le n}((1\text{-}\alpha)\Sigma_{i \text{ with edge from } i \text{ to } k}(pr(i) / t_i))$ *

$= \quad \alpha \quad + (1\text{-}\alpha)\Sigma_{1 \le k \le n}\ \Sigma_{i \text{ with edge from } i \text{ to } k}(pr(i) / t_i)$

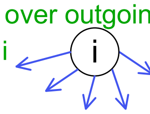
$= \quad \alpha \quad + (1\text{-}\alpha)\Sigma_{1 \le i \le n}\Sigma_{k \text{ with edge from } i \text{ to } k}(pr(i) / t_i)$ *

$= \quad \alpha \quad + (1\text{-}\alpha)\Sigma_{i \text{ with edge from } i}\ pr(i))$

*inner sum Σ_i over incoming edges for one k



*inner sum Σ_k over outgoing edges for one i

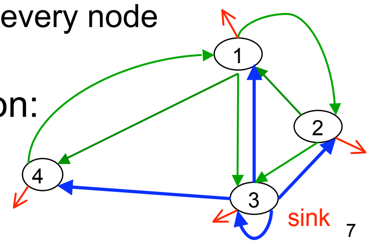


6

Problem for desired normalization

- Have
  $\Sigma_{1 \le k \le n}(pr_{new}(k)) = \alpha + (1-\alpha)\Sigma_{i \text{ with edge from } i}\, pr(i))$
- Missing **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 $\Sigma_{1 \le k \le n}(pr_{initial}(k)) = 1$
then $\Sigma_{1 \le k \le n}(pr(k)) = 1$



7

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 ($1 \le i \le n$)
  If row i contains $t_i > 0$ ones, $L(i,k) = (1/t_i)\, E(i,k)$, $1 \le k \le n$
  If row i contains 0 ones, $L(i,k) = 1/n$, $1 \le k \le n$
- Vector ***pr*** of PageRank values defined by
  $$\boldsymbol{pr} = (\alpha/n, \alpha/n, \dots \alpha/n)^T + (1-\alpha)\, L^T \boldsymbol{pr}$$
- has a solution representing the steady-state values pr(k)

8

Calculation

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

9

PageRank Observations

- PageRank can be calculated for *any* graph
- Google calculates on entire Web graph
- Huge calculation for Web graph
  - precomputed
  - 1998 Google:
    - 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

10

HITS

Hyperlink **I**nduced **T**opic **S**earch

- 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
    - **Point to many hubs**

11

Mutual reinforcement

- Authority weight node j: a(j)
  - Vector of weights ***a***
- Hub weight node j: h(j)
  - Vector of weights ***h***
- Update:

$a_{new}(k) = \Sigma_{\text{i with edge from i to k}} (h(i))$

$h_{new}(k) = \Sigma_{\text{j with edge from k to j}} (a(j))$

12

Matrix formulation

Steady state:

$\boldsymbol{a} = E^T\boldsymbol{h}$ $\qquad$ $\boldsymbol{a} = E^TE\boldsymbol{a}$

$\boldsymbol{h} = E\boldsymbol{a}$ $\qquad$ $\boldsymbol{h} = EE^T\boldsymbol{h}$

Interpretation:

- $E^TE(i,j)$: number nodes point to both node i and node j
  - **"Co-citation"**
- $EE^T(i,j)$: number nodes pointed to by both node i and node j
  - **"Bibliographic coupling"**

13

Iterative Calculation

$\boldsymbol{a} = \boldsymbol{h} = (1, \ldots, 1)^T$

While (not converged) {

$\boldsymbol{a}_{new} = E^t\boldsymbol{h}$

$\boldsymbol{h}_{new} = E\boldsymbol{a}$

$\boldsymbol{a} = \boldsymbol{a}_{new} / ||\boldsymbol{a}_{new}||$ normalize to unit vector

$\boldsymbol{h} = \boldsymbol{h}_{new} / ||\boldsymbol{h}_{new}||$ normalize to unit vector

}

14

Convergence

- Linear algebra - eigenvalues
- Kleinberg uses slightly different iteration and slightly different proof than in *Intro to IR* book
  - Normalization important

$\boldsymbol{a}_0 = \boldsymbol{h}_0 = (1, \ldots, 1)^T$

For k^{th} iteration {

$\boldsymbol{a}_k$ = normalized ($E^T \boldsymbol{h}_{k-1}$)

$\boldsymbol{h}_k$ = normalized ($E \boldsymbol{a}_k$) uses **new value of *a***

}

Then $\boldsymbol{a}_k$ = normalized ($(E^T E)^{k-1} E^T \boldsymbol{a}_0$)

Then $\boldsymbol{h}_k$ = normalized ($(EE^T)^k \boldsymbol{h}_0$)

15

General Theorem:

If M is a symmetric n by n matrix and ***v*** is a vector not orthogonal to the principal eigenvector $\boldsymbol{w}_1$ of M,

then the unit vector in the direction of $M^k \boldsymbol{v}$ converges to $\boldsymbol{w}_1$ as k goes to infinity.

Application:

Since $\boldsymbol{h}_0 = (1,1, \ldots 1)^T$, $\boldsymbol{h}_0$ is not orthogonal to the principal eigenvector of EE^T

$\Rightarrow \boldsymbol{h}_k$ converges

$\boldsymbol{a}_k$ similar but little more work because first vector $E^T \boldsymbol{a}_0$

16

Use of HITS

- Actual use of HITS by IBM people was **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:
     1. Root set
     2. *Plus* nodes pointed to by nodes of root set
     3. *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

17

Results using HITS

- Documents ranked by authority score a(doc) and hub score h(doc)
  1. 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
    - Compare LSI “concepts”

18

Observations

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

19

HITS and clustering

- Non-principal eigenvectors of EE^T and E^TE have positive and negative component values
  - Denote a_{e2}, a_{e3}, …
    matching h_{e2}, h_{e3}, …
- For a matched pair of eigenvectors $\boldsymbol{a}_{ej}$ and $\boldsymbol{h}_{ej}$
  - Denote k^{th} component of j^{th} eigenvector $\boldsymbol{a}_{ej}(k)$ and $\boldsymbol{h}_{ej}(k)$
  - Make a "community" of size *c* (a chosen constant):
    - Choose c pages with most positive $\boldsymbol{h}_{ej}(k)$ - hubs
    - Choose c pages with most positive $\boldsymbol{a}_{ej}(k)$ - authorities
  - Make another "community" of size *c*:
    - Choose c pages with most negative $\boldsymbol{h}_{ej}(k)$ - hubs
    - Choose c pages with most negative $\boldsymbol{a}_{ej}(k)$ - authorities
- Compare LSI

20

Eigenvalues and clustering

General class of techniques for clustering a graph using eigenvectors of adjacency matrix (or similar matrix) called

Spectral clustering

First described in 1973

More later, maybe …

21