DA-Final-6) Page rank & Link analysis

1. Page rank

1-1. Page rank 정의

Early Search Engines and Term Spam

- Search query =>
  (1) find pages with the terms using inverted index (O(logN))
  (2) rank according to term frequency in header, body, etc.
- How to trick? 조작!

Google's two innovations : Listen to what others say about him rather than what he says about himself

- Rank using the terms near the links to that page (해당 page을 linking하는 page)
  => spammers often do not control over the pages that link to their own page
- How about counting # of in-links?
  => "spam farm" problem
- PageRank: Random surfers, starting at a random page and follow a randomly chosen outlinks
  => frequently visited pages are more important than rarely visited pages

Why PageRank works?

- Users tend to place links to pages they think are good or useful
- Random surfer tends to visit such pages

- Starting probability $V_0 = \left[\frac{1}{n}, \ldots\right]$ uniform
  - What are probabilities that the surfer will next be at each of the pages?
- Transition matrix M A로 들어올 확률(row):inlink

A에서 나갈 확률 (col):outlink

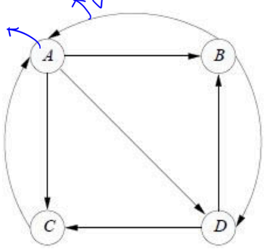
- $$M = \begin{array}{c|c} & \begin{array}{cccc} (A) & (B) & (C) & (D) \end{array} \\ \begin{array}{c} (A) \\ (B) \\ (C) \\ (D) \end{array} & \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \end{array}$$

in/outlink에 대해 uniform-dist.

- V_0 = column vector of $[1/n, \ldots 1/n]$
- $V_1 = M * V_0$
- $V_2 = M^2 * V_0$
- $V_i = M^i * V_0$

linking

page

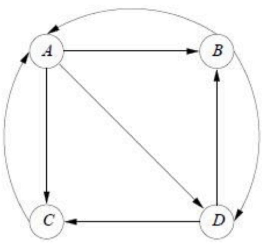


$$V_0 \begin{pmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{pmatrix} \rightarrow V_1 \begin{pmatrix} \frac{1}{4}\cdot\frac{1}{2} + \frac{1}{4}\cdot 1 \\ \frac{1}{4}\cdot\frac{1}{3} + \frac{1}{4}\cdot\frac{1}{2} \\ \frac{1}{4}\cdot\frac{1}{3} + \frac{1}{4}\cdot\frac{1}{2} \\ \frac{1}{4}\cdot\frac{1}{3} + \frac{1}{4}\cdot\frac{1}{2} \end{pmatrix} \rightarrow \cdots$$

- V will converge, i.e. $V = M * V$, if ⇐ Markov process!
  - The graph is strongly connected (reachable from any to any)
- V is an eigenvector of M. Inversion 계산 필요 (비효율적)
  - $M * V = \lambda * V$ (λ : eigenvalue)
  - Because M is stochastic, i.e., its columns add up to 1, V is the principal eigenvector (its eigenvalue is 1, the largest)
  - The principal eigenvector of M tells us where the surfer is most likely to be after a long time
  - 50-75 iterations from V_0 are sufficient to converge in practice

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 9/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix} \begin{bmatrix} 15/48 \\ 11/48 \\ 11/48 \\ 11/48 \end{bmatrix} \begin{bmatrix} 11/32 \\ 7/32 \\ 7/32 \\ 7/32 \end{bmatrix} \cdots \begin{bmatrix} 3/9 \\ 2/9 \\ 2/9 \\ 2/9 \end{bmatrix}$$

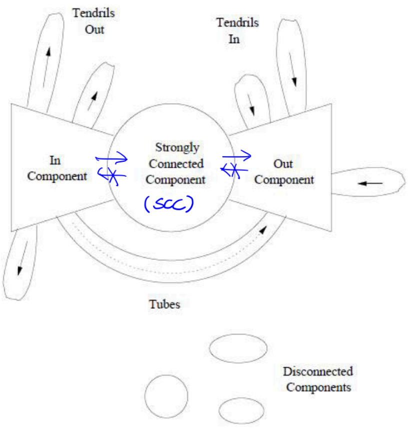
- $$M = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$
- V_0 = column vector of $[1/n, \ldots 1/n]$
- $V_1 = M * V_0$
- $V_2 = M^2 * V_0$
- $V_i = M^i * V_0$



1-2. EX) Structure of the Web

- In-component: reach SCC/but not reachable from SCC
- Out-component: reachable from SCC/but not reach SCC
- Tendrils, Tubes, Isolated components

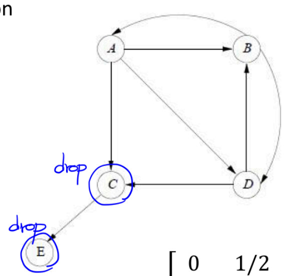
- Surfers will wind up in either the out-component and tendrils => In-component or SCC are not important? X



1-3. Dead ends 배제

1-3-(1). Dead ends를 recursively drop

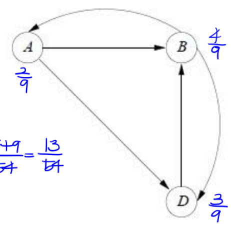
1. Recursively drop dead ends
2. Taxation
(과세)



$P(C) = \frac{2}{9}\cdot\frac{1}{3} + \frac{3}{9}\cdot\frac{1}{2}$

$= \frac{2}{27} + \frac{3}{18} = \frac{4+9}{54} = \frac{13}{54}$

$P(E) = P(C)$



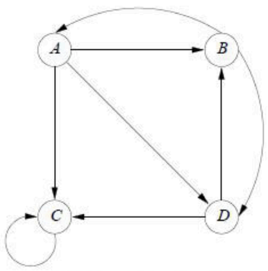
$$M = \begin{bmatrix} 0 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 1/2 & 1/2 & 0 \end{bmatrix} \quad \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 1/6 \\ 3/6 \\ 2/6 \end{bmatrix} \begin{bmatrix} 3/12 \\ 5/12 \\ 4/12 \end{bmatrix} \begin{bmatrix} 5/24 \\ 11/24 \\ 8/24 \end{bmatrix} \cdots \begin{bmatrix} 2/9 \\ 4/9 \\ 3/9 \end{bmatrix}$$

What about C and E?

- C => 1/3*2/9 + ½*3/9 = 13/54
- E => C

⇒ Drop!

1-3-(2). Taxation



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

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 3/24 \\ 5/24 \\ 11/24 \\ 5/24 \end{bmatrix} \begin{bmatrix} 5/48 \\ 7/48 \\ 29/48 \\ 7/48 \end{bmatrix} \begin{bmatrix} 21/288 \\ 31/288 \\ 205/288 \\ 31/288 \end{bmatrix} \cdots \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

② **Taxation**

- $V' = \beta * M * V + (1-\beta) * e/n$ (random jump: Uniform probability)
  - β is constant (0.8~0.9)
  - e is vector of 1
  - n is # of nodes
  - With probability $1-\beta$, the surfer moves to a random page
- When β = 0.8

$\frac{1}{5}\cdot\frac{1}{4}$

$$\mathrm{v}' = \begin{bmatrix} 0 & 2/5 & 0 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 4/5 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{bmatrix} \mathrm{v} + \begin{bmatrix} 1/20 \\ 1/20 \\ 1/20 \\ 1/20 \end{bmatrix}$$

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 9/60 \\ 13/60 \\ 25/60 \\ 13/60 \end{bmatrix} \begin{bmatrix} 41/300 \\ 53/300 \\ 153/300 \\ 53/300 \end{bmatrix} \begin{bmatrix} 543/4500 \\ 707/4500 \\ 2543/4500 \\ 707/4500 \end{bmatrix} \cdots \begin{bmatrix} 15/148 \\ 19/148 \\ 95/148 \\ 19/148 \end{bmatrix}$$

- Let
  - D be the set of all Web pages
  - $I(p)$ be the set of pages that link to the page p (inlinks)
  - $|O(q)|$ be the total number of links going out of page q (outlinks)

- The PageRank score of page p, denoted by $PR(p)$, is

$$PR(p) = d\left[\sum_{q \in I(p)} \frac{PR(q)}{|O(q)|}\right] + (1-d)\frac{1}{|D|}$$

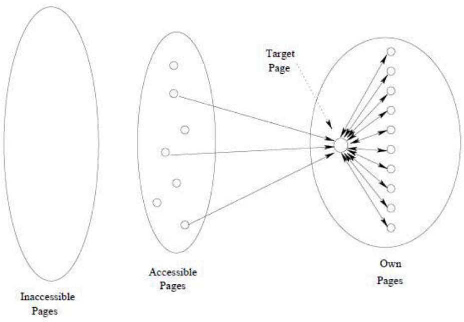
→ p로 들어오는 pages들

$\boxed{q} \to (p)$

2. Link spam

2-1. Link spam structure

- Architecture of a Spam Farm
  - Inaccessible pages: cannot be affected by spammer
  - Accessible pages: can be affected but not controlled by spammer
  - Own pages: can be controlled by spammer



2-2. Trust rank & Spam mass - Link spam 방지

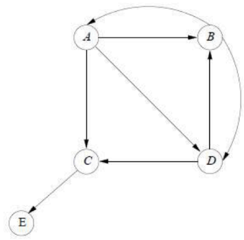
- Detect & eliminate the structure where one page links to many pages with links back to it
  => Spammer develop different structure, a variation
  => War between spammer and search engine
- Modify the definition of PageRank to lower the rank of link-spam
- TrustRank: a variation of topic-sensitive PageRank (trustable pages로 random jump)
- Spam mass: a calculation to identify spam-like pages

- TrustRank
  - Similar to topic-sensitive PageRank where topic is a set of pages believed to be trustworthy
  - Select "topic" from the top list of PageRank
  - Pick a controlled domain such as ".edu"
- Spam Mass = (PageRank - TrustRank) / PageRank

Node	PageRank	TrustRank	Spam Mass
A	3/9	54/210	0.229
B	2/9	59/210	-0.264
C	2/9	38/210	0.186
D	2/9	59/210	-0.264

3. HITS (Hyperlink-induced topic search)

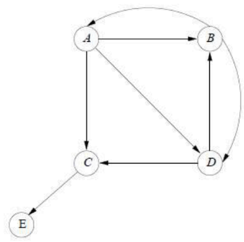
- Authorities(권위): valuable pages providing information about a topic (e.g. course page)
- Hubs: valuable pages providing links to Authorities (e.g. course list page) → cheating이 쉬울 수 있다(단점)
- "A page is a good hub if it links to good authorities, and a page is a good authority if it is linked by good hubs"
- Compute h (hubbiness) and a (authority) scores iteratively
- h: hubbiness score, a: authority score
- Start with h, a vector of all 1's
- $\mathrm{a} = L^T\mathrm{h}$ and scale so the largest component to 1 (hub→authority)
- $\mathrm{h} = L\mathrm{a}$ (authority←hub)
- Repeat until the changes are small



$$
L = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array}
\begin{array}{c}
\begin{array}{ccccc} A & B & C & D & E \end{array} \\
\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{array}
\qquad
L^{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}
$$

{ row : outlink, column : inlink } { row : inlink, column : outlink }

$$
L = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\qquad
L^{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}
$$



$$
\underset{\mathbf{h}}{\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}}
\quad
\underset{L^{\mathrm{T}}\mathbf{h}}{\begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix}}
\xrightarrow{\text{scaling}}
\underset{\mathbf{a}}{\begin{bmatrix} 1/2 \\ 1 \\ 1 \\ 1 \\ 1/2 \end{bmatrix}}
\quad
\underset{L\mathbf{a}}{\begin{bmatrix} 3 \\ 3/2 \\ 1/2 \\ 2 \\ 0 \end{bmatrix}}
\xrightarrow{\text{scaling}}
\underset{\mathbf{h}}{\begin{bmatrix} 1 \\ 1/2 \\ 1/6 \\ 2/3 \\ 0 \end{bmatrix}}
$$

$$
\underset{L^{\mathrm{T}}\mathbf{h}}{\begin{bmatrix} 1/2 \\ 5/3 \\ 5/3 \\ 3/2 \\ 1/6 \end{bmatrix}}
\quad
\underset{\mathbf{a}}{\begin{bmatrix} 3/10 \\ 1 \\ 1 \\ 9/10 \\ 1/10 \end{bmatrix}}
\quad
\underset{L\mathbf{a}}{\begin{bmatrix} 29/10 \\ 6/5 \\ 1/10 \\ 2 \\ 0 \end{bmatrix}}
\quad
\underset{\mathbf{h}}{\begin{bmatrix} 1 \\ 12/29 \\ 1/29 \\ 20/29 \\ 0 \end{bmatrix}}
$$

$\mathrm{h} = ..\ LL^{\mathrm{T}}\,\mathrm{h}$

$\mathrm{a} = ..\ L^{\mathrm{T}}\,\mathrm{h}$

$$
\mathbf{h} = \begin{bmatrix} 1 \\ 0.3583 \\ 0 \\ 0.7165 \\ 0 \end{bmatrix}
\qquad
\mathbf{a} = \begin{bmatrix} 0.2087 \\ 1 \\ 1 \\ 0.7913 \\ 0 \end{bmatrix}
$$

$$
LL^{\mathrm{T}} = \begin{bmatrix} 3 & 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
$$