20230727

Machine Learning with Graphs

서수원

Business Intelligence Lab.

산업경영공학과, 명지대학교

01

PageRank

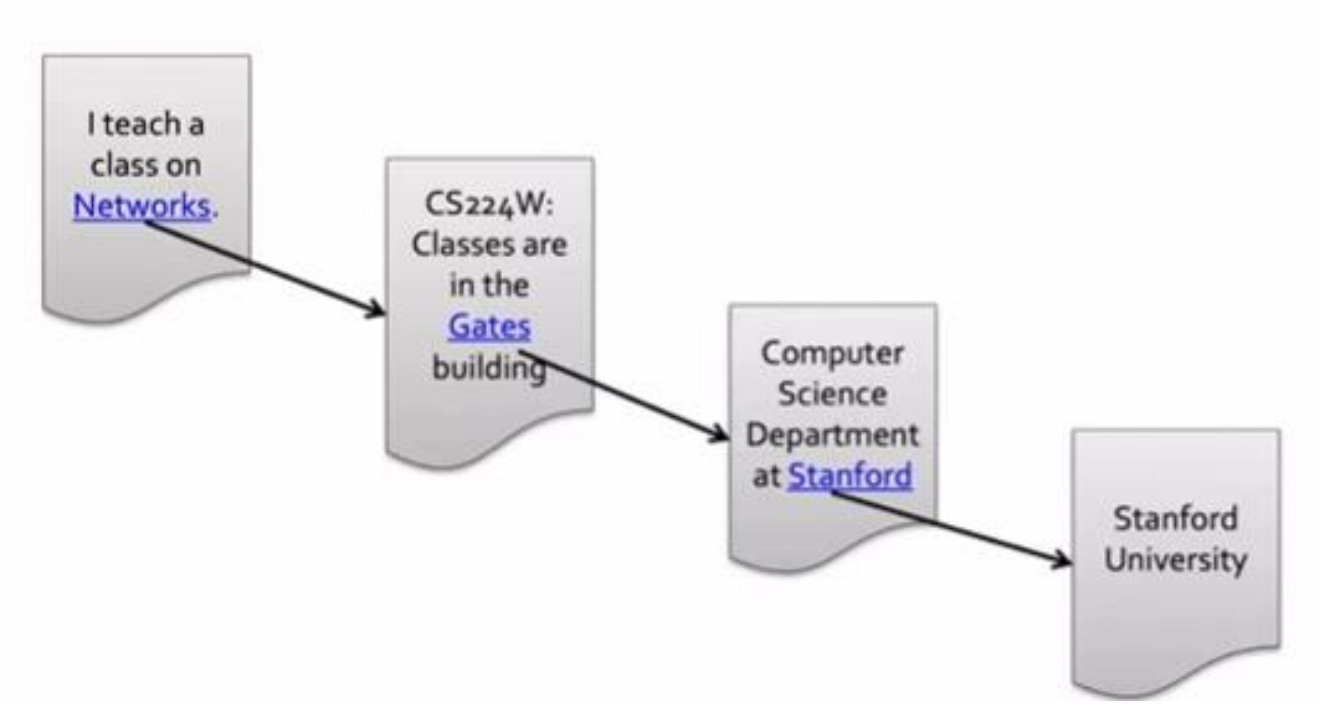
PageRank

- Page Rank
  - 구글에서 만들었다.
  - 모든 노드(웹 혹은 문서)가 동일한 중요도를 갖지 않는다고 생각한다.
  - 노드는 웹 페이지, 엣지는 하이퍼링크로 생각 할 수 있다.

- **Web as a graph:**
  - Nodes = web pages
  - Edges = hyperlinks
  - **Side issue:** What is a node?
    - Dynamic pages created on the fly
    - "dark matter" – inaccessible database generated pages

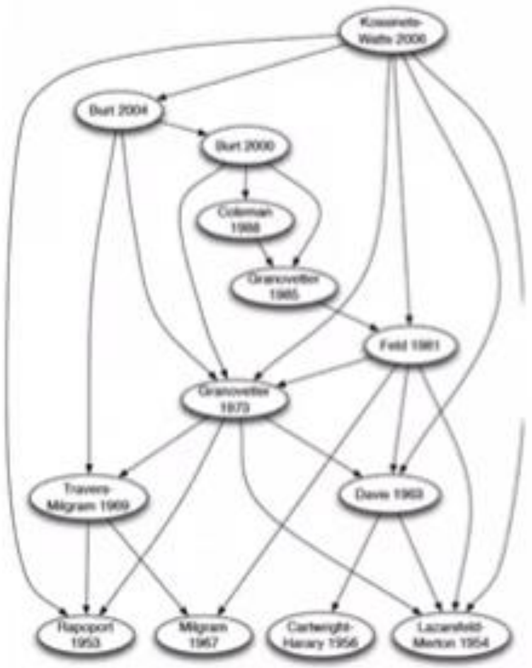
PageRank

- Page Rank
  - 과거에는 웹이 단지 웹에서 웹으로, 하이퍼링크를 통해 이동이 가능했다.
    - ✓ 이를 정적 웹페이지와 링크로 구성되는 방식이라 한다.
  - 현재는 단순히 웹에서 웹으로 이동하는 것이 아닌, 게시물, 좋아요 등의 여러 형태가 생겼다.

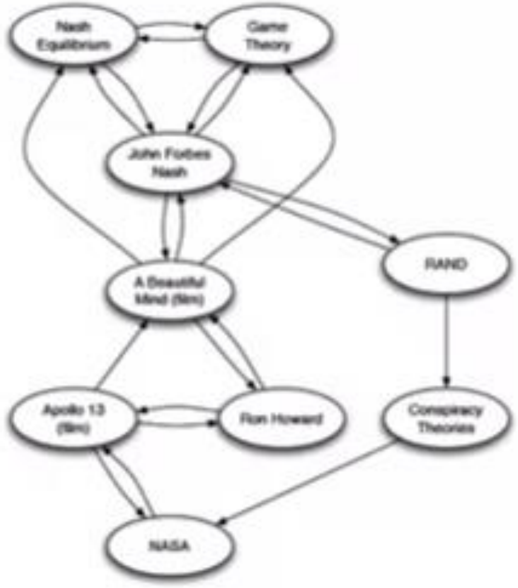


PageRank

- Page Rank
  - 다른 예) 인용 피인용관계, 참고문헌의 관계 등도 나타낼 수 있다.



Citations



References in an Encyclopedia

PageRank

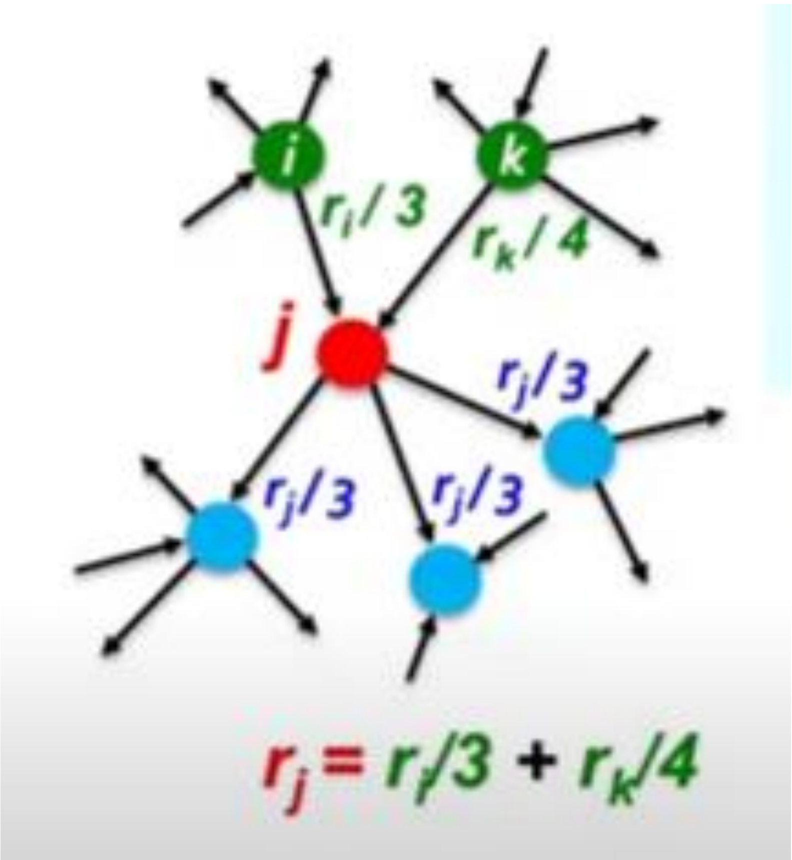
- Link Analysis Algorithms
  - 노드의 중요성을 계산하는 링크 수준의 분석을 다룰 예정이다.
  - PageRank, Personalized PageRank, Random walk with Restarts가 있다.
- Links as Votes
  - 링크가 많이 연결되어 있다면 더 중요하다고 생각한다.
  - 링크에는 들어오는 링크, 나오는 링크가 있다.
    - ✓ 들어오는 링크는 위변조가 어렵고, 나오는 링크는 상대적으로 위변조가 쉽다.(우리가 생성하는 것이기 때문이다.)
- Are all in-links equal?
  - 그렇지 않다.
  - 다른 웹에 중요도에 따라 링크의 강도?가 달라진다.

PageRank

- Page Rank : The “Flow”Model
  - 중요한 페이지(노드)와 연결되는 것이 더 가치 있다.
  - 각각의 링크가 노드의 중요도에 따른 비율의 가치를 갖는다.
    - ✓ Out-links와 연관이 있다.
  - 자신의 중요도는 자신에 들어오는 링크의 중요도에 따라 다르다.

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

d_i … out-degree of node i

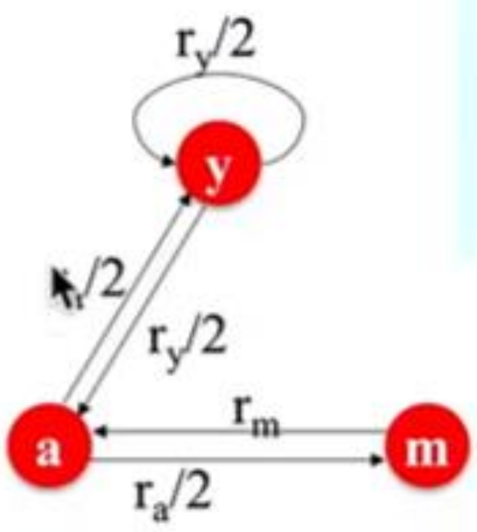


PageRank

- Page Rank : The “Flow”Model
  - 페이지의 중요도는, 중요한 다른 페이지(노드)와 연결되는 것이 더 가치 있다.
  - 공식화 하면 밑과 같다.

$$r_j = \sum_{i \rightarrow j} \frac{r_i}{d_i}$$

d_i ... out-degree of node i



“Flow” equations:

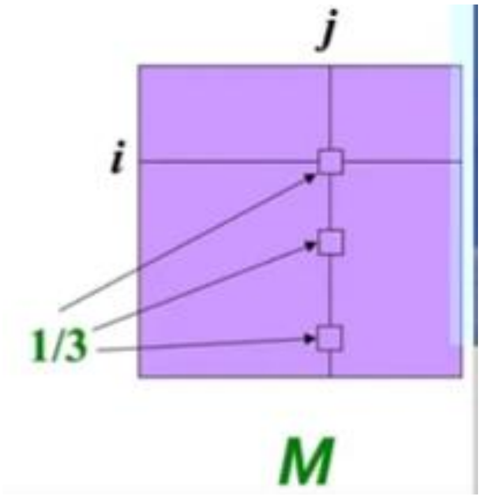
$r_y = r_y/2 + r_a/2$

$r_a = r_y/2 + r_m$

$r_m = r_a/2$

PageRank

- Page Rank : Matrix Formulation
  - 페이지를 행렬 형식으로 변환한다.
  - Page J have Dj out-links
  - 만약 j의 out-links가 i에 연결 된다면, Mij = 1/dj
  - J열의 모든 값을 합치면 1이 된다.

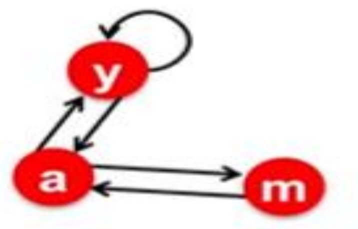


- **Rank vector r:** An entry per page
  - r_i is the importance score of page i
  - $\sum_i r_i = 1$

$$r = M \cdot r$$

PageRank

- Page Rank : Matrix Formulation
  - 페이지를 행렬 형식으로 변환한다.
  - Page J have Dj out-links
  - 만약 j의 out-links가 i에 연결 된다면, Mij = 1/dj
  - J열의 모든 값을 합치면 1이 된다.



	r_y	r_a	r_m
r_y	½	½	0
r_a	½	0	1
r_m	0	½	0

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

d_i … out-degree of node i

$$\begin{aligned} r_y &= r_y/2 + r_a/2 \\ r_a &= r_y/2 + r_m \\ r_m &= r_a/2 \end{aligned}$$

$$\underbrace{\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix}}_{r} = \underbrace{\begin{bmatrix} ½ & ½ & 0 \\ ½ & 0 & 1 \\ 0 & ½ & 0 \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix}}_{r}$$

PageRank

- 참고할 것
  - 고윳값과 고유벡터
    - ✓ 임의의 n×n 행렬 A 에 대하여, 0이 아닌 솔루션 벡터 x 가 존재한다면 숫자 λ는 행렬 A 의 고윳값라고 할 수 있다.

      λx= Ax

      이 때, 솔루션 벡터 x는 고윳값 λ 에 대응하는 고유벡터이다. 이때 Ax=λx식은 (A−λI)x=0 로 변환이 가능하다.

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$det(A - \lambda I) = det\left(\begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix}\right) = 0$$

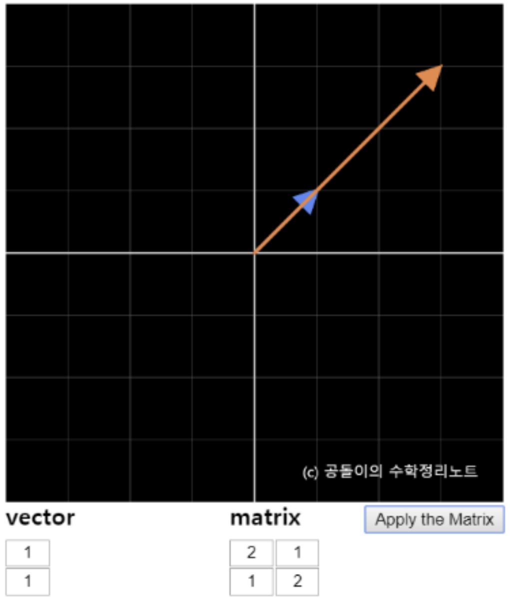
$$\Rightarrow (2-\lambda)^2 - 1$$

$$= (4 - 4\lambda + \lambda^2) - 1$$

$$= \lambda^2 - 4\lambda + 3 = 0$$

$$, \lambda_1 = 1, \lambda_2 = 3$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



PageRank

- Eigenvector Formulation
  - 1 x r = M x r

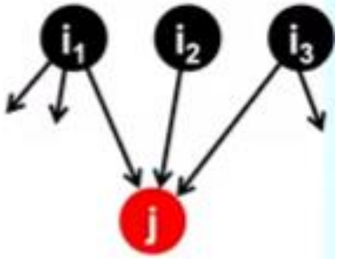
$$\underset{r}{\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix}} = \underset{M}{\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}} \underset{r}{\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix}}$$

  - 만약 임의의 r벡터에서 시작을 하고 계속 서치를 한다면, M(M(,,,M(M r)))의 형태가 된다.
  - r은 고윳값이 1인 M행렬의 고유벡터 이다.

PageRank

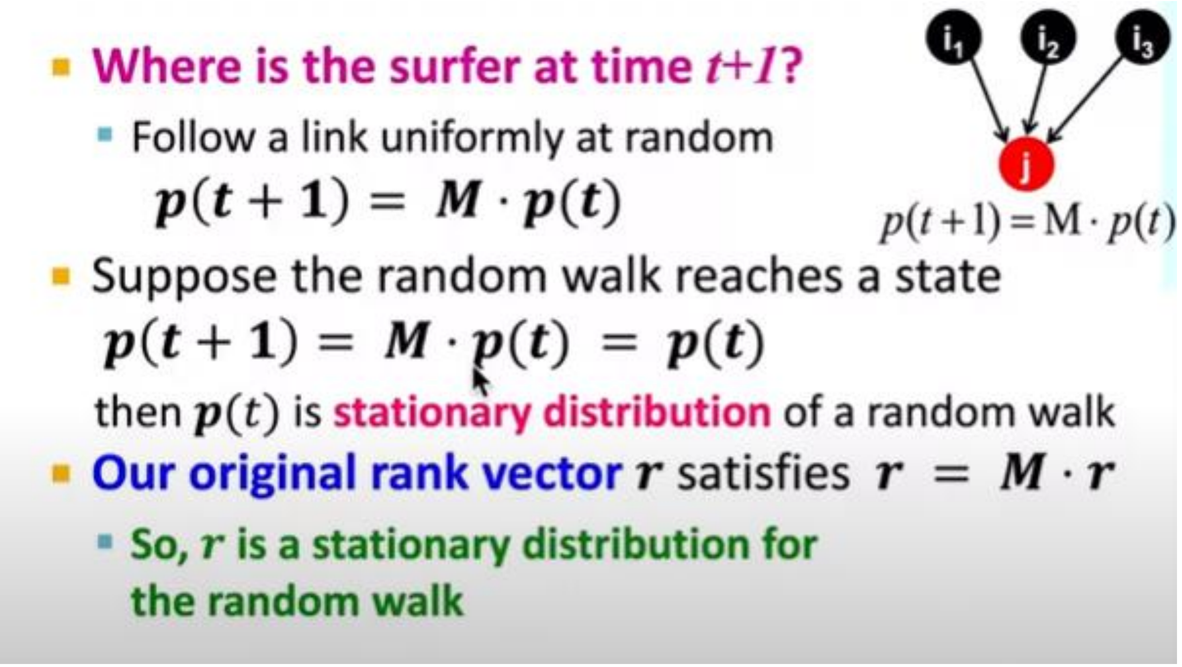
- Page Rank : Connection to Random Walk
  - T시점에 이용자는 I page에 있다.
  - T+1점에 이용자는 out-link를 따라 랜덤하게 나간다.
  - 결국 I 와 연결된 J page 까지 도달 하게 된다.
  - 무한반복 한다.

- **Imagine a random web surfer:**
  - At any time t, surfer is on some page i
  - At time $t+1$, the surfer follows an out-link from i uniformly at random
  - Ends up on some page j linked from i
  - Process repeats indefinitely



PageRank

- Page Rank : Connection to Random Walk
  - 이용자가 t+1시점에 어디에 있는지 알고 싶다
  - P(T)는 t시점에 이용자가 어디에 있던, 이용자가 있는 곳과 연결된 곳으로 랜덤하게 이동하는 것을 의미한다.
  - 반복 후 안정상태가 되면 2번째 식과 같이 나타낼 수 있다.



02

PageRank : How to solve?

PageRank : How to solve?

- PageRank : Solve Method
  - 처음에 노드들에 임의의 랭크값을 부여 해준다.
  - 오른쪽과 같은 식이 성립 할 때 까지 반복한다.
    - ✓ 값들의 변경된 정도의 합이 입실론 보다 작으면 된다는 의미이다. $(\sum_i |r_i^{t+1} - r_i^t| < \epsilon)$
    - ✓ 즉 수렴할 때 까지 한다는 의미이다.
    - ✓ 이 과정을 Power Iteration Method라고 한다.
    - ✓ 평균적으로 50번 정도 반복 하면 그만 한다.
    - ✓ 구글은 매일 이 과정을 통해 웹사이트의 순위를 구한다고 한다.

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

- Initialize: $\boldsymbol{r}^0 = [1/N, \ldots, 1/N]^T$
- Iterate: $\boldsymbol{r}^{(t+1)} = \boldsymbol{M} \cdot \boldsymbol{r}^t$
- Stop when $|\boldsymbol{r}^{(t+1)} - \boldsymbol{r}^t|_1 < \varepsilon$

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

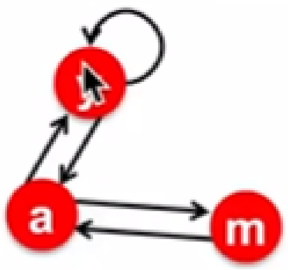
d_i .... out-degree of node i

$|x|_1 = \sum_1^N |x_i|$ is the L₁ norm
Can use any other vector norm, e.g., Euclidean

PageRank : How to solve?

- Power Iteration Method example
  - 간단한 반복 기법이다.

- **Power Iteration:**
  - Set $r_j \leftarrow 1/N$
  - **1:** $r'_j \leftarrow \sum_{i \to j} \frac{r_i}{d_i}$
  - **2:** If $|r - r'| > \varepsilon$:
    - $r \leftarrow r'$
  - **3:** go to **1**



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

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

- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 1/3 \\ 3/6 \\ 1/6 \end{bmatrix} \begin{bmatrix} 5/12 \\ 1/3 \\ 3/12 \end{bmatrix} \begin{bmatrix} 9/24 \\ 11/24 \\ 1/6 \end{bmatrix} \ldots \begin{bmatrix} 6/15 \\ 6/15 \\ 3/15 \end{bmatrix}$$

PageRank : How to solve?

- Power Iteration Method
  - 세가지가 중요하다.
    - ✓ 결과가 합리적인가?
    - ✓ 수렴 하는가?
    - ✓ 우리가 원하는 대로 수렴 하는가?
  - 하나라도 만족하지 않으면 안된다.

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

or equivalently

$$r = Mr$$

PageRank : How to solve?

- PageRank : Problems
  - Some pages are dead ends
    - ✓ Out-links가 없다는 의미이다.
    - ✓ 수렴하지 않는다. 수학적 문제가 발생한다.
  - Spider traps
    - ✓ 모든 Out-links 가 자신에게 온다.
    - ✓ 수렴은 하기에 이것 자체의 문제는 아니지만 결과가 합리적이지 않다.

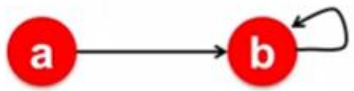
- **The "Dead end" problem:**

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

- **Example:**

		Iteration: 0,	1,	2,	3...
r_a	=	1	0	0	0
r_b		0	1	0	0

- **The "Spider trap" problem:**



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

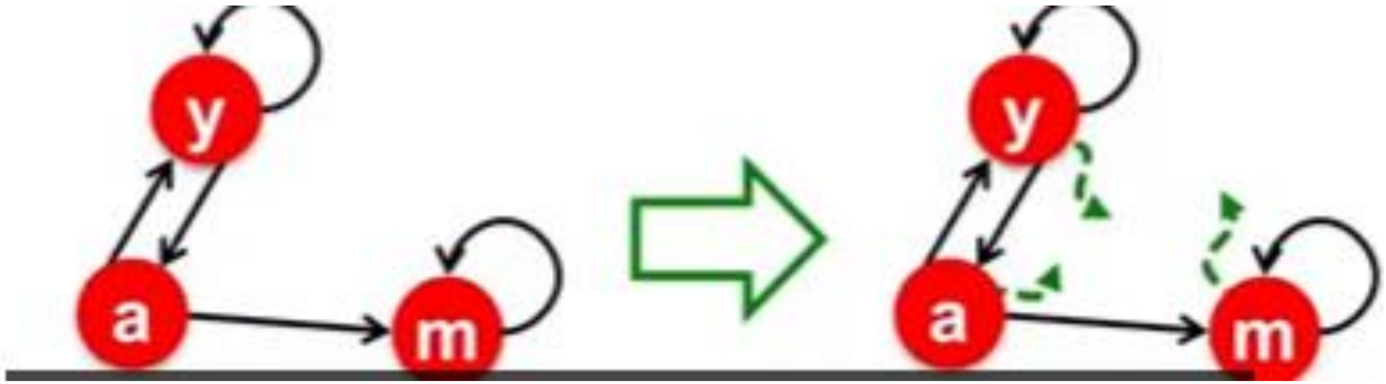
- **Example:**

		Iteration: 0,	1,	2,	3...
r_a	=	1	0	0	0
r_b		0	1	1	1

- 세가지가 중요하다.
  - ✓ 결과가 합리적인가?
  - ✓ 수렴 하는가?
  - ✓ 우리가 원하는 대로 수렴 하는가?

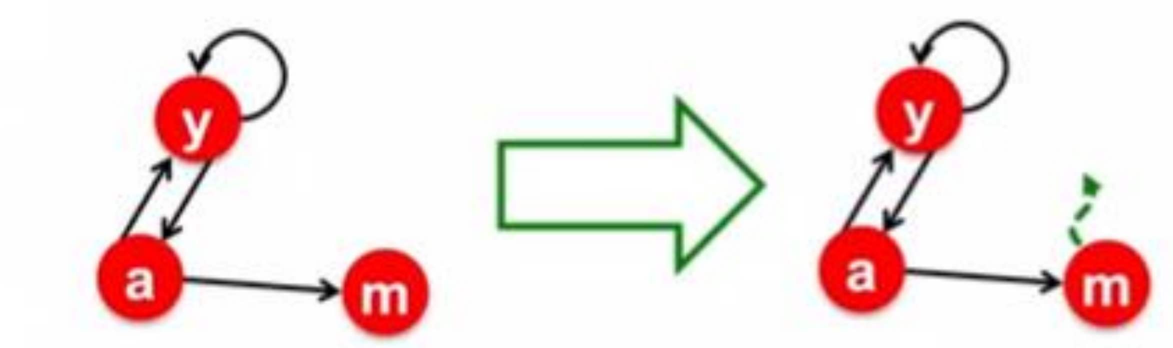
PageRank : How to solve?

- Solution to Spider Traps
  - Random surfer 에 옵션을 추가한다.
    - ✓ 베타라는 확률값을 추가한다.
    - ✓ 1-베타 만큼의 확률로 랜덤 페이지로 점프한다.
      - ❖ 순간이동 한다고 생각하면 된다.
      - ❖ 베타값은 주로 0.8,0.9 로 책정한다.
    - ✓ 베타 확률만큼 랜덤하게 연결 되어있는 out-link로 이동한다.



PageRank : How to solve?

- Solution to Dead Ends
  - Random surfer 에 텔레포트의 개념을 추가한다.
    - ✓ Dead Ends 구간에서 합이 1인 유니폼 한 랜덤값을 부여한다.
      - ❖ 이는 다른 페이지로 이동 할 수 있게 도와준다.



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

PageRank : How to solve?

- Final Solution
  - J의 중요도는 베타에 노드 i의 중요도를 곱한 것과 1-베타에 1/전체 문서의 수를 한 것과 같다.
  - 뒷부분의 식을 설명하면1-베타는 랜덤하게 문서로 넘어갈 확률인데, 이는 1-베타가 채택 되었을 때 전체 문서에서 j문서로 갈 확률을 의미한다.

- **PageRank equation** [Brin-Page, 98]

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{N}$$

d_i … out-degree of node i

PageRank : How to solve?

- The Google Matrix G
  - 앞선 식을 행렬로 변환하면 밑과 같게 된다.
    - ✓ G가 이제 M의 역할을 한다고 보면 된다.

- **PageRank equation** [Brin-Page, 98]

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{N}$$

d_i … out-degree of node i

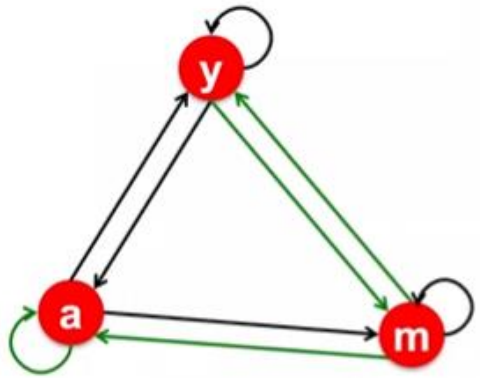
- **The Google Matrix *G*:**

$[1/N]_{N \times N}$…N by N matrix where all entries are 1/N

$$P = \beta M + (1-\beta)\left[\frac{1}{N}\right]_{N \times N}$$

PageRank : How to solve?

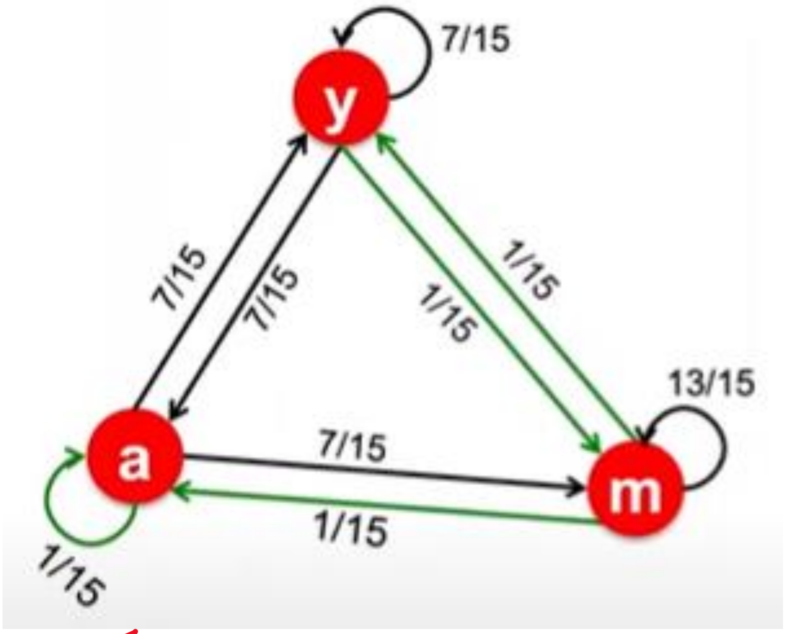
- Example
  - 베타가 0.8이라고 가정한다.



$$0.8\ \underbrace{\begin{bmatrix}1/2 & 1/2 & 0\\ 1/2 & 0 & 0\\ 0 & 1/2 & 1\end{bmatrix}}_{\mathbf{M}} + 0.2\ \underbrace{\begin{bmatrix}1/3 & 1/3 & 1/3\\ 1/3 & 1/3 & 1/3\\ 1/3 & 1/3 & 1/3\end{bmatrix}}_{[1/N]_{N\times N}}$$

$$\mathbf{G} = \begin{array}{c} y\\ a\\ m \end{array}\begin{bmatrix}7/15 & 7/15 & 1/15\\ 7/15 & 1/15 & 1/15\\ 1/15 & 7/15 & 13/15\end{bmatrix}$$

$$\begin{array}{c} y\\ a\\ m \end{array} = \begin{array}{c} 1/3\\ 1/3\\ 1/3 \end{array}$$



$$\begin{array}{c} y\\ a\\ m \end{array} = \begin{array}{cccccc} 1/3 & 0.33 & 0.24 & 0.26 & & 7/33\\ 1/3 & 0.20 & 0.20 & 0.18 & \dots & 5/33\\ 1/3 & 0.46 & 0.52 & 0.56 & & 21/33 \end{array}$$

PageRank : How to solve?

- Example
  - 모든 문서엔 중요도가 있는 것을 볼 수 있다.
    - ✓ B엔 많은 문서로 부터 들어오기 때문에 중요도가 높다.
    - ✓ B로부터 값을 받는 C도 중요도가 높다.
    - ✓ Dead End에도 중요도가 있다.