# The PageRank Citation Ranking: Bring Order to the Web

2022.03.13

발표자 | 김진영

# PageRank

: 웹 페이지의 상대적인 중요성을 측정하는 방법 (measure the relative importance of web pages)

## 어떤 웹 페이지가 중요한 웹페이지인가?

어떤 페이지의 모든 백링크를 찾아내는 것은 불가능 하지만, 모든 페이지를 크롤링한뒤, 포워드 링크는 알아낼 수 있다 라는 생각에서 출발

~~(사실 "전세계의 모든 웹페이지를 크롤링한다는 생각" 이건 98년도 논문이기에 가능한 일이지 않을까 싶음)~~

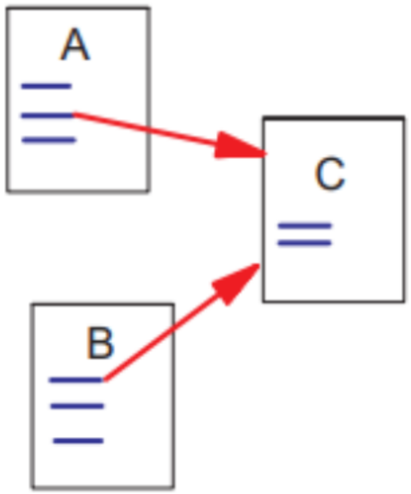



Figure 1: A and B are Backlinks of C

* 사전지식 점검)
  포워드 링크 : 해당 페이지로부터 밖으로 나가는 링크
  백 링크 : 해당 페이지를 가리키는 링크

* 사진 설명)
  C는 2개의 백링크(A,B)를 가진다.

# PageRank

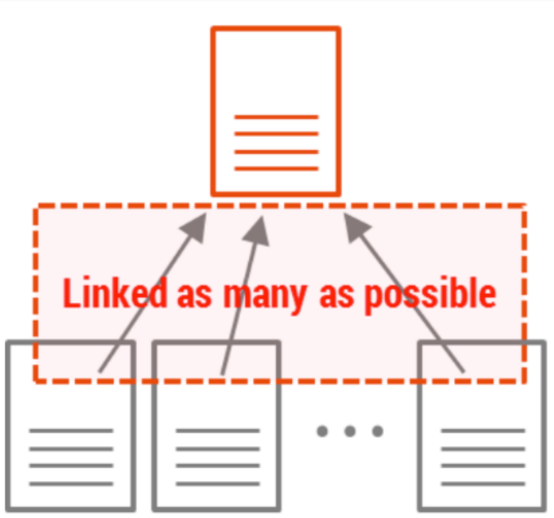
: 웹 페이지의 상대적인 중요성을 측정하는 방법 (measure the relative importance of web pages)

## 어떤 웹 페이지가 중요한 웹페이지인가?

어떤 페이지의 모든 백링크를 찾아내는 것은 불가능 하지만, 모든 페이지를 크롤링한뒤, 포워드 링크는 알아낼 수 있다 라는 생각에서 출발

~~(사실 "전세계의 모든 웹페이지를 크롤링한다는 생각" 이건 98년도 논문이기에 가능한 일이지 않을까 싶음)~~

- 얼마나 많은 페이지가 참조하였는가?
(when a page has many backlinks)




- 얼마나 많은 "중요하다고 평가된" 페이지가 참조하였는가?
(when a page has a few highly ranked backlinks)

# PageRank

: Definition of PageRank (Simple)

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

← backlink의 영향력

← 가리키는 페이지의 rank에 동일하게 기여하기 위함 (참조의 신중성)

얼마나 많은 backlink를 가지고 있는지

u : 웹 페이지

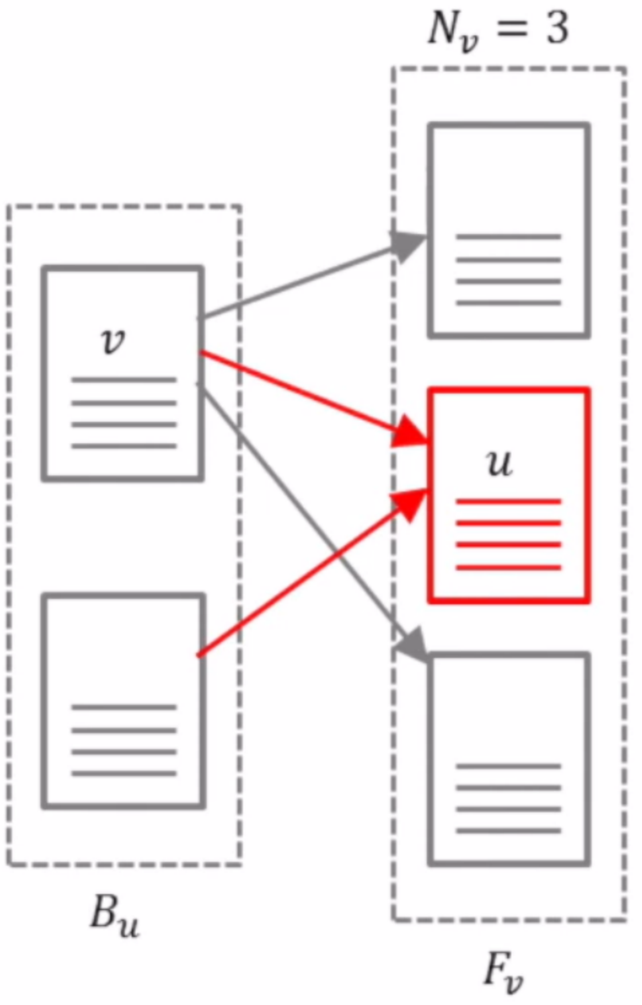
$F_u$ : u 페이지가 가리키는 페이지들의 집합 (u페이지의 포워드 링크 집합)

$B_u$ : u 페이지를 가리키는 페이지의 집합 (u페이지의 백 링크 집합)

$N_u$ : u 페이지로부터 나가는 링크의 개수 (= $F_u$ 의 갯수)

c : 정규화를 하기 위해 사용하는 상수

(전체 웹 페이지의 rank 합을 일정하게 하기 위해 정규화 사용)



# PageRank

: Definition of PageRank (Simple)

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

← backlink의 영향력

← 가리키는 페이지의 rank에 동일하게 기여하기 위함 (참조의 신중성)

얼마나 많은 backlink를 가지고 있는지

u : 웹 페이지

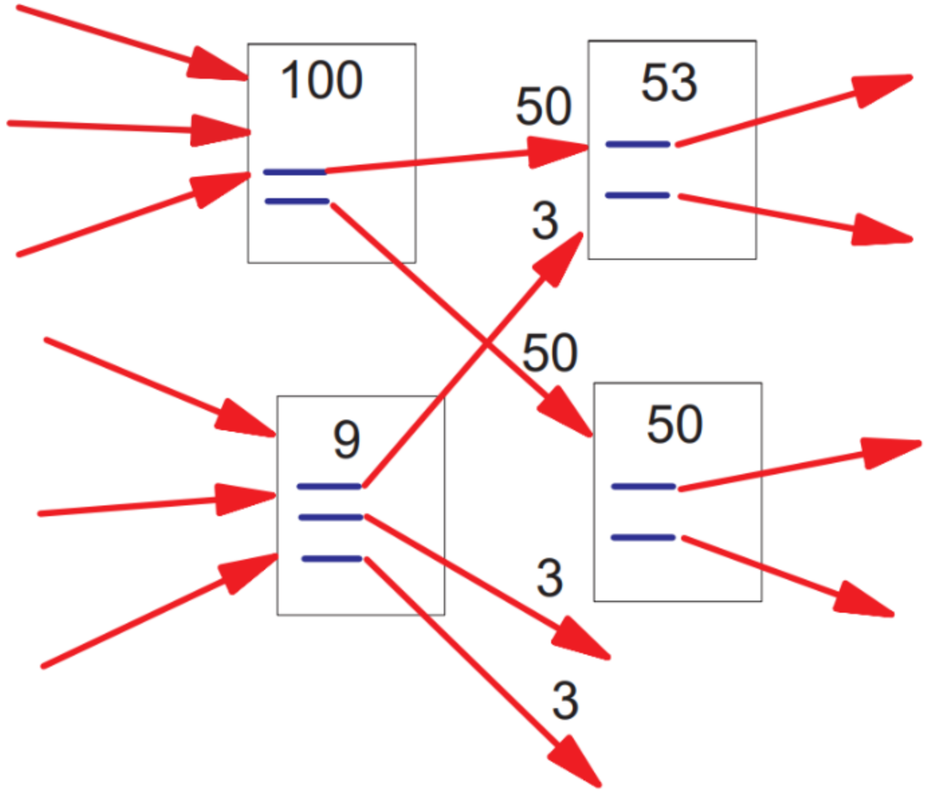
$F_u$ : u 페이지가 가리키는 페이지들의 집합 (u페이지의 포워드 링크 집합)

$B_u$ : u 페이지를 가리키는 페이지의 집합 (u페이지의 백 링크 집합)

$N_u$ : u 페이지로부터 나가는 링크의 개수 (= $F_u$ 의 갯수)

c : 정규화를 하기 위해 사용하는 상수

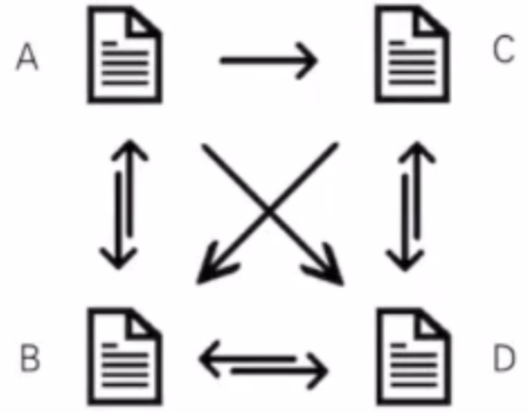
(전체 웹 페이지의 rank 합을 일정하게 하기 위해 정규화 사용)



slide 참고) https://www.youtube.com/watch?v=2CWnZfBSj0Q

# PageRank

: Definition of PageRank (Matrix)




(A) Adjacency Matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

(H) Hyperlink Matrix

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

(r) Rank Vector

$$\begin{bmatrix} R(A) \\ R(B) \\ R(C) \\ R(D) \end{bmatrix}$$

$$0 \cdot R(A) + \frac{1}{2}R(B) + 0 \cdot R(C) + 0 \cdot R(D)$$

$$\frac{1}{3} \cdot R(A) + 0 \cdot R(B) + \frac{1}{2} \cdot R(C) + \frac{1}{2} \cdot R(D)$$

$$\frac{1}{3} \cdot R(A) + 0 \cdot R(B) + 0 \cdot R(C) + \frac{1}{2} \cdot R(D)$$

$$\frac{1}{3} \cdot R(A) + \frac{1}{2}R(B) + \frac{1}{2} \cdot R(C) + 0 \cdot R(D)$$

$$R(u) = \sum_{\{v \in B_u\}} \frac{R(v)}{N_v}$$

$R(v)$: BackLink의 영향력

$N_v$: 참조의 신중성

$\{v \in B_u\}$: 얼마나 많은 BackLink가 있는가?

# PageRank

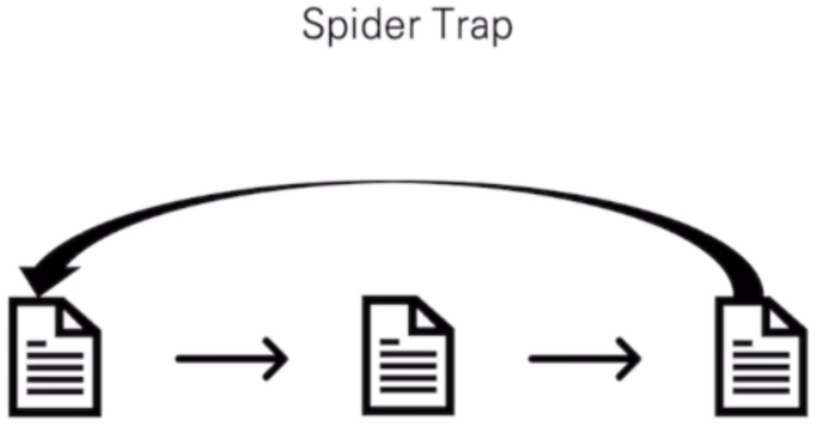
: problem of ranking function - Rank Sink

**case1. Dead End (= Dangling Node)**

: 노드가 연결이 되어 있으나, 해당 Dangling Node는 다른 노드로의 연결이 되어 있지 않은 경우


**case2. Spider Trap**

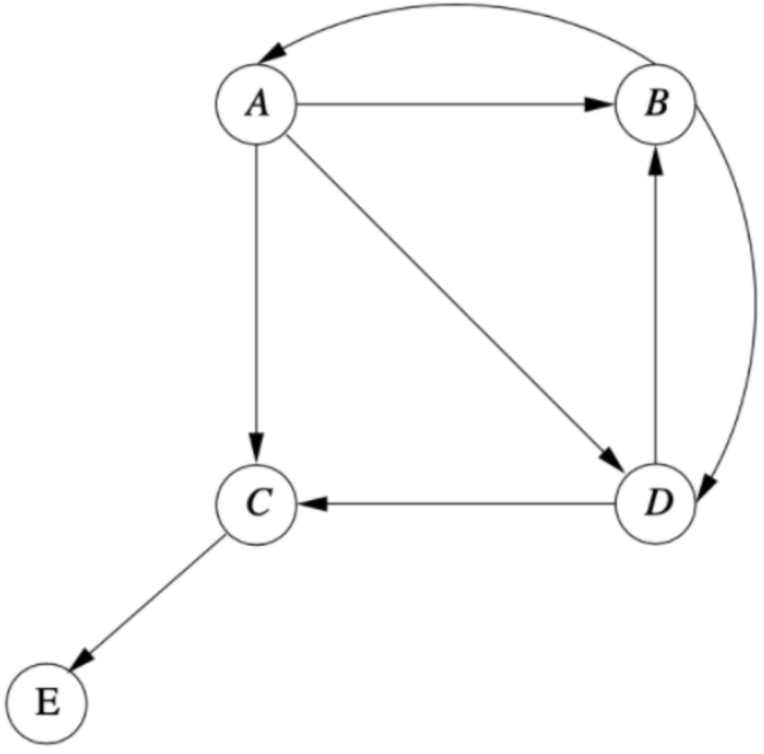
: 노드들이 자신들끼리만 linked 되어 있는 경우



# PageRank

: problem of ranking function - Rank Sink

## case1. Dead End (= Dangling Node) 가 문제가 되는 이유




이미지 출처 : Mining of Massive Datasets - Figure 5.4

**(A)** Adjacency Matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

**(H)** Hyperlink Matrix

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

**(R)** Rank Vector

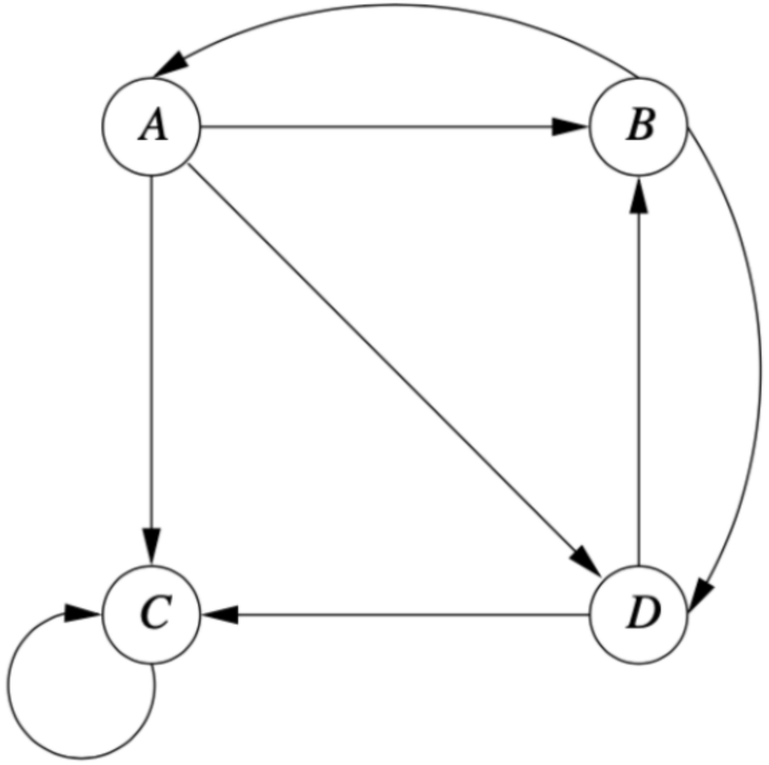
$$\begin{bmatrix} 1/3 * R(B) \\ 1/2 * R(A) + 1/2 * R(D) \\ 0 \\ 1/2 * R(B) \\ 0 \end{bmatrix}$$

웹의 변환행렬(H)의 특정 열의 합이 1이 아닌 0이 된다.
-> 변환행렬은 더이상 확률적(stochatic)하지 않는다.
-> web 페이지에 대한 상대 중요성 정보를 얻을 수 없다.

# PageRank

: problem of ranking function - Rank Sink

## case2. Spider Trap 이 문제가 되는 이유




이미지 출처 : Mining of Massive Datasets - Figure 5.6

PageRank 계산 계속 반복하다 보면...

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

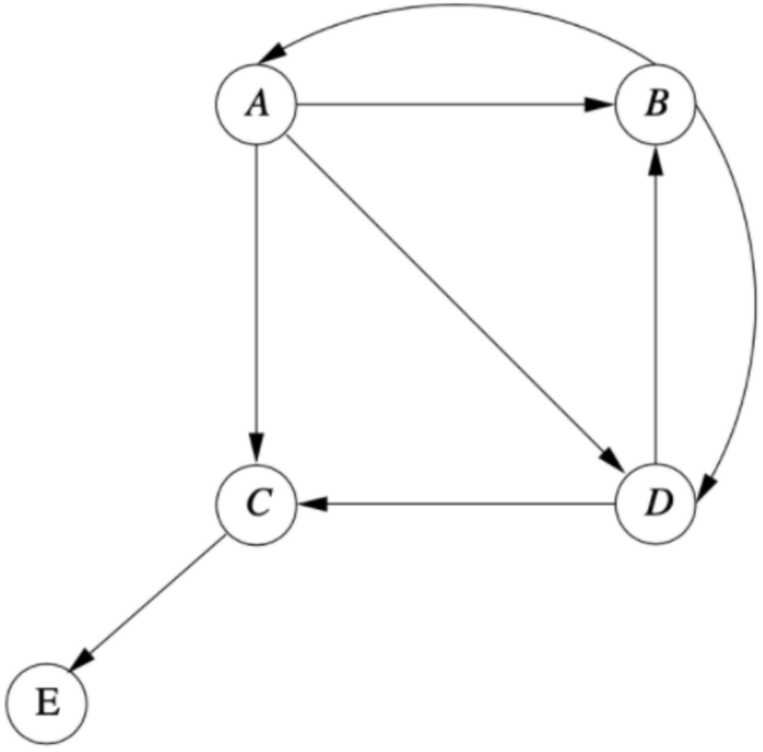
모든 페이지랭크가 C에 있게 된다

-> 페이지가 트랩에 갇혀서 다른 페이지로 갈 수 없다.

# PageRank

: solution of Rank Sink problem

## Dead End 의 해결방법




**(H)**
Hyperlink
Matrix

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

**(H)**
Hyperlink
Matrix

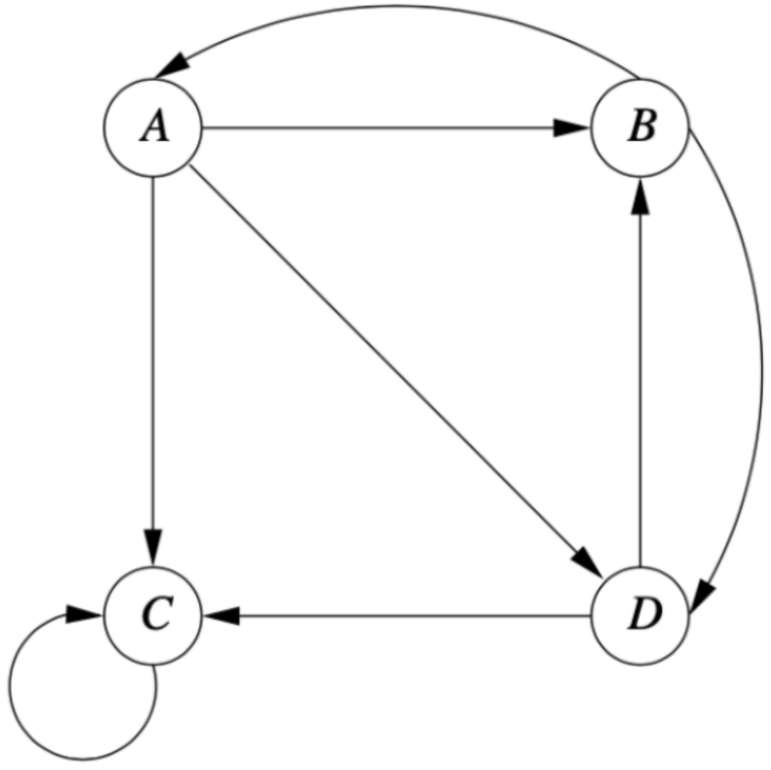
$$\begin{bmatrix} 0 & 1/2 & 0 & 0 & 1/5 \\ 1/3 & 0 & 0 & 1/2 & 1/5 \\ 1/3 & 0 & 0 & 1/2 & 1/5 \\ 1/3 & 1/2 & 0 & 0 & 1/5 \\ 0 & 0 & 1 & 0 & 1/5 \end{bmatrix}$$

이미지 출처 : Mining of Massive Datasets - Figure 5.4

# PageRank

: solution of Rank Sink problem

## Spider Trap 의 해결방법 - Random Suffer Model




Taxation :

suffer가 trap에서 돌아도, 다른 페이지로 갈 수 있는 확률을 만들어 주는 것

**(H)**
Hyperlink
Matrix

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

k(0.8 ~ 0.9) *

**(H)**
Hyperlink
Matrix

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

+ (1-k) *

$$\begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix}$$

이미지 출처 : Mining of Massive Datasets - Figure 5.6

참고한 자료) https://www.youtube.com/watch?v=2CWnZfBSj0Q

# PageRank

: 논문에서의 pagerank 정리

$$R(u) = c \sum_{\{v \in B_u\}} \frac{R(v)}{N_v}$$

$R(v)$ : BackLink의 영향력

$N_v$ : 참조의 신중성

$\{v \in B_u\}$ : 얼마나 많은 BackLink가 있는가?

---

$$R'(u) = c \sum_{\{v \in B_u\}} \frac{R(v)}{N_v} + \boxed{c\,E(u)}$$

random suffer 모델을 사용하여, spider trap은 해결
but, dead node에 대한 해결책은 적용되지 않은 rank 식

---

$$R' = c(AR' + E)$$

Matrix Form

A : transition matrix
E : random suffer model을 의미

---

$$R' = c(A + E \cdot 1)\,R'$$

$$R' = \|R'\|_1$$

Final Matrix Form

Transition Matrix · Rank

# Markov chain

: 마르코프 체인의 사용 예로 볼 수 있는 PageRank 알고리즘

## Markov Chain?

: 어떤 사건이 발생할 확률값이 시간에 따라 변화해 가는 과정
Markov Chain은 마르코프 성질을 가진 이산 확률 과정을 의미한다.

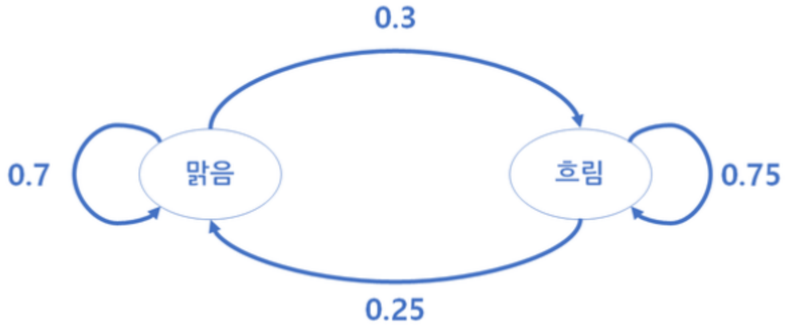
특정 상태의 확률은 오직 과거의 상태에 의존한다.
(조건1) 시간은 이산적으로 변한다.
(조건2) 시간 t+1에 사건이 발생할 확률이 시간 t에 발생한 사건에만 영향을 받는다.

## Markov Chain의 중요 특징

markov chain 계산을 충분히 많은 횟수를 반복하다보면, 어느 순간에 전이행렬이 변하지 않는 상태가 된다. (수렴하게 된다.)

# Markov chain

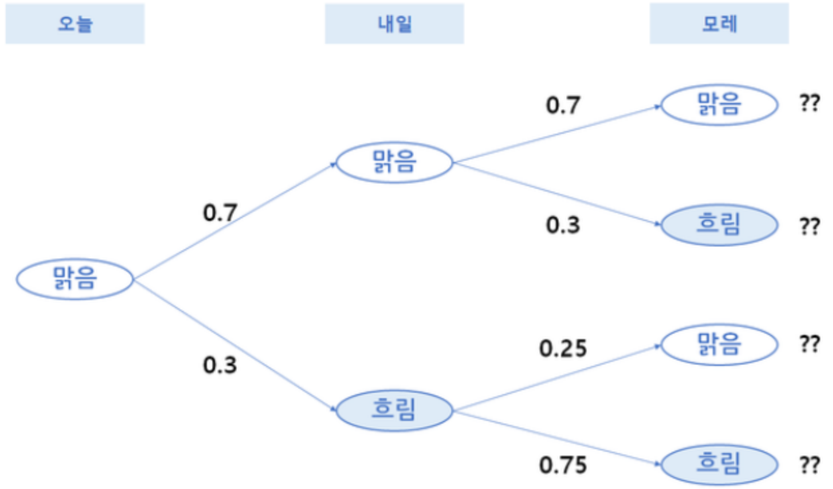
: 마르코프 체인 계산 ex.




상태 전이도 ex

$$P = \begin{bmatrix} 0.7 & 0.3 \\ 0.25 & 0.75 \end{bmatrix}$$

전이행렬(transition matrix)으로의 표현



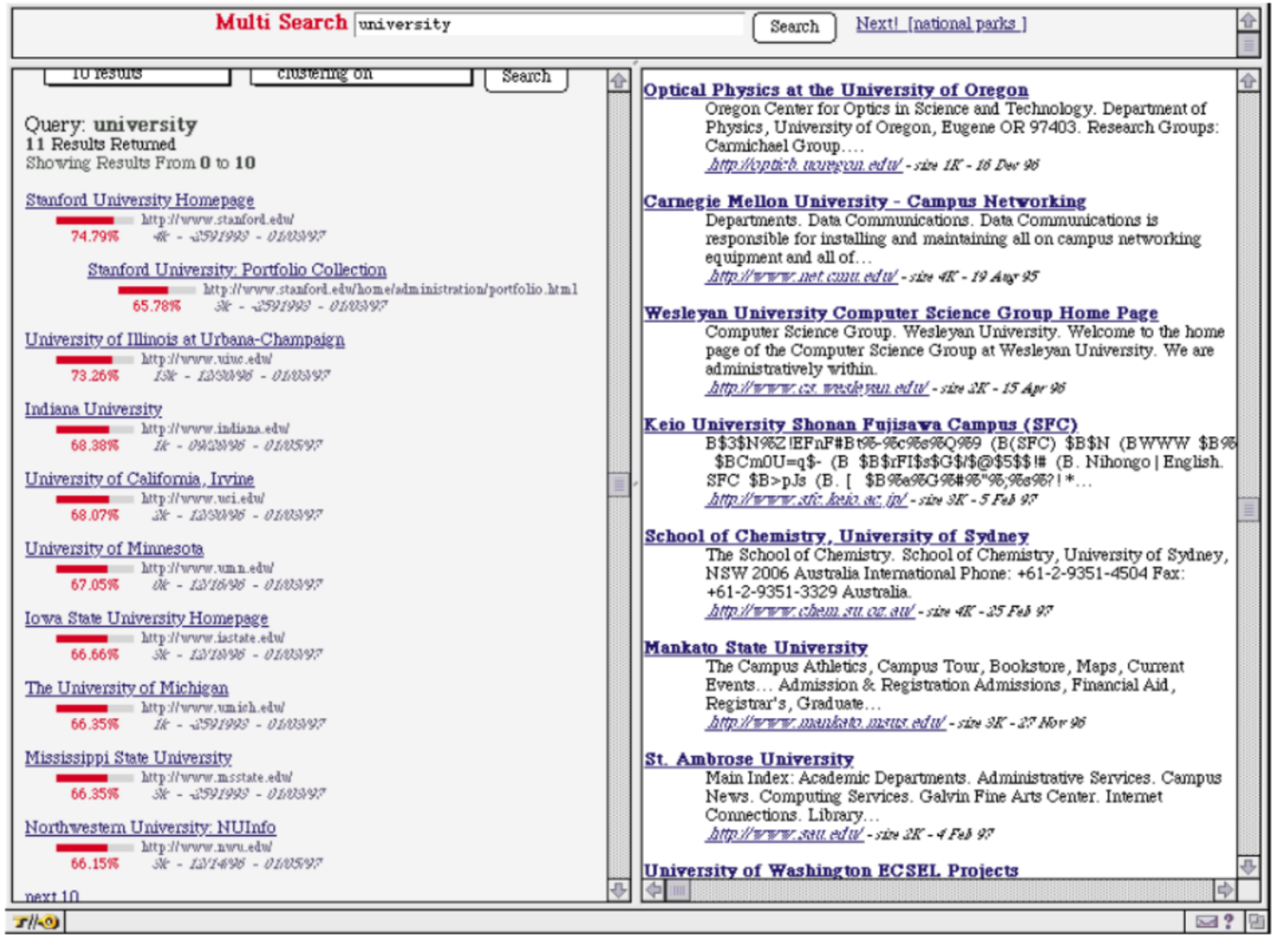

$$\begin{bmatrix} 0.7 & 0.3 \\ 0.25 & 0.75 \end{bmatrix} \cdot \begin{bmatrix} 0.7 & 0.3 \\ 0.25 & 0.75 \end{bmatrix} = \begin{bmatrix} 0.565 & 0.435 \\ 0.362 & 0.637 \end{bmatrix}$$

모레의 날씨 전이행렬(transition matrix)

# PageRank

: searching with pagerank

## Title Search




PageRank를 사용하여 제목기반/ 전체 검색어 기반의
검색 엔진을 개발하였고, 전체 검색의 기반 검색엔진이 구글임.

PageRank는 불특정 검색어에 대하여 우수한 rank로 정렬된
결과를 보여준다는 장점을 가진다.

# PageRank

: personalized pagerank

$$R' = c(A + E \cdot 1)\, R'$$
$$R' = ||R'||_1$$

PageRank에서의 E

: ranksink에서, spider trap을 해결하기 위한 하나의 방법인 (random suffer model)

: page rank 값을 조정하는 강력한 parameter

-> random suffer가 주기적으로 이동하는 웹 페이지의 분포

-> 웹을 거시적인 관점에서 바라보거나, 특정 부분에 대한 개인화된 관점에서 파악 가능

개인화된 검색 엔진이 가지게 되는 특징

- 단순히 북마크나 홈페이지를 입력하는 것 만으로, 사용자의 취향 추측 가능
- 상업적 이익을 위한 검색순위 조작을 방지할 수 있다.

  높은 페이지 랭크 : 중요한 페이지에서 언급 or 중요하지 않은 페이지에서 언급X
  (중요한 페이지의 광고(링크)를 구매하여 조작하는 경우가 생길 수 있으나, 비용이 소요되므로 어느정도의 상업적 이익을 위한 조작을 방지한다고 볼 수 있다.)

# PageRank

: Applications

## 1. Estimating Web Traffic

PageRank와 실제 웹 페이지의 사용도 사이의 차이점을 살펴 보는 것을 통하여,
사람들이 보고는 싶어하는데, 자신의 웹페이지에서는 언급하고 싶지 않은 것이 무엇이 있는지 파악할 수 있다.

## 2. PageRank as Backlink Predictor

## 3. User Navigation : The PageRank Proxy


web proxy?

: 사용자가 각 링크의 pagerank와 함께 부가적인 정보도 함께 볼 수 있음

: 사용자가 링크를 클릭하기 이전에 관련 정보를 미리 알 수 있다는 장점有

# PageRank

: Conclusion

PageRank를 사용함으로써, 더 중요하고 중심적인 웹 페이지들의 검색 결과를 정렬할 수 있다.
PageRank에 기반이 되는 개념은, 웹페이지 외부의 정보(=백링크)를 사용한다는 것이다.
중요한 페이지들로부터의 백링크는 평균적인 페이지로부터의 백링크보다 더 중요하고, pagerank는 이러한 특징을 반영하여 구현되어 있다.
PageRank는 대부분의 질의어들을 사용하여, 소수에게만 자주 사용되는 문서들을 분리해내는 데에도 용이하게 사용될 수 있다.
PageRank는 검색 이외에도 트래픽 추정이나 사용자 네비게이션과 같이 다양한 곳에 적용될 수 있다.

**웹 그래프의 구조가 다양한 정보 검색 작업에서 유용하게 사용될 수 있다.**