Google Ranking System :

우리 집 앞마당 텃밭을 지나는 금맥

이 인 석

제 31 차 대수 캠프

2014 년 2 월 6 일

Google Ranking System : PageRank

- Google = Google search engine + PageRank
  - search engine은 모두 오십보 백보
- PageRank
  - Sergei Brin and Lawrence Page et al., January 29, 1998
  - "The PageRank citation ranking : bringing order to the web"
  - Technical Report, Stanford InfoLab, 1999

PageRank

- 1995 년 (?) Stanford 에서 Golub 의 선대 강의 듣다 착안
  - 시가 총액 300조원짜리 리포트!
- 1998 년 Google 시범서비스 시작
- idea : "좋은 벗 하나면 열 친구 안 부럽다!"
  - 친구 숫자 < 친구들의 질

선대 복습

- 선대에 등장하는 극한 3 개?
  -
  -
  -

선대에 등장하는 극한

- $\exp(A)$
- eigenvector 의 존재 motivation
  - (그림)
  - 예를 들어 회전변환 R_θ 는?
- Markov chain

선대에 등장하는 극한

- $A \in M_n(\mathbb{C}),\ \ X \in \mathbb{C}^n$
- $\lim_{m\to\infty} A^m X$? No way (in general) ! ?
- $\lim_{m\to\infty} \dfrac{A^m X}{\|A^m X\|}$?

표기법

- $A \in M_n(\mathbb{C}),\ \ \lambda \in \mathbb{C}$
- $E_\lambda = E_{\lambda,A} = \{X \in \mathbb{C}^n \mid AX = \lambda X\}$
  - λ-eigenspace for A
- $\mathrm{mult}_A(\lambda) = m$ if $\phi_A(t) = (t-\lambda)^m g(t)$ and $g(\lambda) \neq 0$
  - the multiplicity of λ for A
- $\dim(E_\lambda) = \mathrm{mult}_A(\lambda)$??
  - 만약 $\mathrm{mult}_A(\lambda) = 1$ 이면 ?

Dominant Eigenvector (Principal Eigenvector)

- dominant eigenvalue = 절댓값 가장 큰 eigenvalue
  - dominant eigenvector

- 만약 dominant eigenvalue $\lambda > 0$ 유일하고,
  $\text{mult}_A(\lambda) = 1$ 이면

$$\lim_{m \to \infty} \frac{A^m X}{\|A^m X\|} = \text{the dominant eigenvector}$$

  - (증명) "거의" 언제나……
  - L^2-norm 이나 L^1-norm 이나……

- dominant eigenvector 여럿이면?

Dominant Eigenvector

- $U^{-1}AU = D = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \quad A^m X = UD^m U^{-1} X$
  - assume $\lambda_1 > 0$ is the unique dominant eigenvalue
  - assume $\mathrm{mult}_A(\lambda_1) = 1$
  - write $U = (u_{ij}), \ U^{-1}X = (y_j),$ assume $y_1 \neq 0$

- $\lim_{m\to\infty} A^m X / \|A^m X\|$ 의 k-좌표는 ?
  - $\dfrac{\sum_j u_{kj}\lambda_j^m y_j}{\sqrt{\sum_i |\sum_j u_{ij}\lambda_j^m y_j|^2}} = \dfrac{\lambda_1^m \sum_j u_{kj}\left(\frac{\lambda_j}{\lambda_1}\right)^m y_j}{\lambda_1^m \sqrt{\sum_i \left|\sum_j u_{ij}\left(\frac{\lambda_j}{\lambda_1}\right)^m y_j\right|^2}} \ \to \ \dfrac{y_1}{\sqrt{\sum_i |u_{i1}y_1|^2}}\, u_{k1}$
  - 따라서 $\lim_{m\to\infty} A^m X / \|A^m X\| = (U$ 의 첫번째 열의 상수배$)$
  - U 의 첫번째 column $= A$ 의 dominant eigenvector

Dominant Eigenvector

- $\lim_{m\to\infty} A^m X / \|A^m X\|$ 는 X 와 무관
- "Power Method" :
  - '적당한' X 에 대해
  - m 을 키워 가면서 dominant eigenvector 의 근삿값 구한다
  - 왜 X 필요 ?
- 참고 : (L^1-norm 이라면) $\|A^m X\| = 1$
  - if A is a Markov matrix and X is a probability vector

Dominant Eigenvector

- ("Matrix Computation", "Applied Linear Algebra") $\times \frac{2}{3}$
  $\approx$ dominant eigenvector & power method

- Brin-Page : Golub 의 응용선대 강의 듣다가 · · · · · ·
  - 우리는 · · · · · ·

Positive Markov Matrix

- $X = (x_i) \in \mathbb{R}^n$ is a **probability vector** if $x_i \geq 0$ and $\sum_i x_i = 1$
- $A \in M_n(\mathbb{R})$ is a **Markov matrix** if every column is a probability vector
  - Markov matrix = stochastic matrix = transition matrix
- $X \in \mathbb{R}^n$ or $A \in M_n(\mathbb{R})$ is **positive** if every entry is positive

- $\lim_{m\to\infty} A^m X$ 에 관한 '그럴듯한' **theory** 는
  A 가 **positive Markov matrix** 인 경우에만 존재 **!**
  - regular Markov matrix $\cdots\cdots$

Rank (인격)

- $r_i = (\text{rank (인격) of } i),\ 0 < r_i \in \mathbb{R}$
  - rank (인격)는 숫자 클수록 높다
  - rank = **reputation** = 인기도 = 중요도 $\neq$ 등수
- 정의 $r_i = \sum_{j \to i} \frac{1}{N_j} r_j$
  - $j \to i$: j는 i를 존경
  - $N_j = \left|\{k \mid j \to k\}\right|$
- 존재? 유일? 어떻게 계산?

PageRank

- $r_i = (\text{ PageRank of the web page } i) > 0$
- 300 조원짜리 정의 $r_i = \sum_{j \to i} \frac{1}{N_j} r_j$
  - $j \to i$: j 에서 i 로 가는 link 존재
  - $N_j = \left| \{k \mid j \to k\} \right|$
- 존재? 유일? 어떻게 계산?

선형대수학

- put $a_{ij} = \begin{cases} \frac{1}{N_j} & \text{if } i \neq j \text{ and } j \to i \\ 0 & \text{otherwise} \end{cases}$
  - $a_{ii} = 0$, 즉 self-link 는 무시
- 300 조원짜리 정의 $r_i = \sum_j a_{ij} r_j$ (1 차 연립방정식)
- $R = AR$, where $R = (r_i) \in \mathbb{R}^n$, $A = (a_{ij}) \in M_n(\mathbb{R})$
  - A 의 **eigenvalue 1 인 eigenvector** R 찾는 문제 (n = 수억)
  - R 의 이름은 rank vector

연습문제

- Brin-Page의 리포트에는 $r_i = c \sum_j a_{ij} r_j$
  - "c is a factor used to normalize 어쩌구 ……"
  - "$R = cAR$이므로 c는 A의 eigenvalue 저쩌구 ……"
  - 리포트 학점은…… ? Honor system !

- 숙제 : If A is Markov and $R = (r_i)$ exists s.t. $R = cAR$ and $\sum_i r_i \neq 0$, show $c = 1$.
  - recall $r_i > 0$

Uniqueness Problem

- eigenvalue 1 인 A 의 eigenvector 여럿이면?
  - A 가 block-diagonal matrix 이면?
  - Brin-Page: "there is a small problem ····· two web pages that point to each other but to no other page ····· "
- A 의 dominant eigenvalue 가 여럿이더라도 문제
- 무언가 보완(땜질) 이 필요
  - note: A is a very very sparse matrix
  - 경험: 0 이 많은 행렬은 대개 '문제'가 많다·····

Markov Matrix

- 1차 조작(땜질): 행렬 $A \in M_n(\mathbb{R})$에 zero column 있으면 zero column을 ${}^t\left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)$로 대체
  - 목적: A를 **Markov matrix**로 만들기 위해
  - outlink가 하나도 없는 page는 모든 page(자기 자신 포함)로 가는 outlink가 있는 것으로 간주

- A는 이제 Markov matrix
  - 즉 A의 모든 좌표는 0 이상이고 각 column의 합은 1
  - 즉 A의 모든 column vector는 **probability vector**

Existence of eigenvalue 1

- 1 은 항상 Markov matrix 의 eigenvalue
  - $(A - I)$ 의 row 들의 합은 0 이므로, $\det(A - I) = 0$
- Gerschgorin's disk theorem
  - $\lambda \in \mathbb{C}$ 가 Markov matrix 의 eigenvalue 이면, $|\lambda| \leq 1$
  - i.e. 1 is **a** dominant eigenvalue of A
  - 증명은 trivial (결과는 매우 유용)

Gerschgorin's Disk Theorem (1931)

- ${}^{\mathbf{t}}A = B = (b_{ij})$, $BX = \lambda X$, $X = (x_i) \neq 0$, $\rho_i = \sum_j |b_{ij}|$
  - choose k s.t. $|x_k| = \max_j |x_j|$, thus $x_k \neq 0$
  - $\sum_j b_{kj} x_j = \lambda x_k$, $\sum_{j \neq k} b_{kj} x_j = \lambda x_k - b_{kk} x_k$
  - $|\lambda - b_{kk}| = \left| \frac{\sum_{j \neq k} b_{kj} x_j}{x_k} \right| \leq \sum_{j \neq k} |b_{kj}| = \rho_k - |b_{kk}|$
  - $|\lambda| = |\lambda - b_{kk} + b_{kk}| \leq |\lambda - b_{kk}| + |b_{kk}| \leq \rho_k$

- *Gerschgorin is a Belarusian Jew*
  - *transliteration from Yiddish spelling*
  - *Gershgorin, Geršgorin, Hershhorn, Herschhorn, ...*

Google Matrix와 Damping Factor

- 2차 땜질 : $G = dA + \frac{1-d}{n}\mathbf{1}$, (**1**은 모든 좌표가 1인 행렬)
  - G = Google matrix
  - $d = 0.85$ = **damping factor**
    - 'damp' = 둔화시키다(진동(발산)을 멈추다), 낙담시키다

- damping factor $d = 0.85$를 도입한 목적
  - $0.15/n$는 거의 zero(무의미한 수)
  - G는 **positive** Markov matrix(모든 좌표 양수)
  - positive Markov matrix의 경우에만 '멋진' theory 존재
  - 다른 '변명'은 잠시 후에

Positive Markov Matrix

- 정리 : G 가 positive Markov matrix 이면,
  - 1 은 G 의 유일한 dominant eigenvalue
    - 즉 λ 가 G 의 eigenvalue 이고 $|\lambda| = 1$ 이면, $\lambda = 1$
  - $\dim E_{1,G} = 1$, $\mathrm{mult}_G(1) = 1$
- G 의 dominant eigenvector 구하면 된다!
  - "Power Method"
- 숙제 (위 정리 증명에 필요)
  - $\dim E_{\lambda,B} = \dim E_{\lambda,{}^tB}$
  - $\alpha_i \in \mathbb{C}$, $|\alpha_1 + \cdots + \alpha_r| = |\alpha_1| + \cdots + |\alpha_r| \implies$??

Markov Chain

- 정리 : G 가 positive Markov matrix 이면,
  - $L = \lim_{m\to\infty} G^m$ 존재 ($GL = LG = L$)
  - $L = (R, R, \ldots, R)$, 단 $R \in \mathbb{R}^n$ 은 (positive) probability vector 이고 G 의 dominant eigenvector (PageRank vector)
  - for any probability vector $S \in \mathbb{R}^n$, $\lim_{m\to\infty} G^m S = R$
- 증명 : Friedberg-Insel-Spence §5.3, §7.2 참조
  - G 가 대각화 안 되면 Jordan canonical form 필요
  - key 는 $\mathrm{mult}_G(1) = 1$

KEY : $\mathrm{mult}_G(1) = 1$

- 만약 $\dim(E_{1,G}) < \mathrm{mult}_G(1)$ 이면
- 예를 들어, G 에 Jordan block $\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ 이 있으면,

  G^m 에는 $\begin{pmatrix} 1 & 0 & 0 \\ m & 1 & 0 \\ * & m & 1 \end{pmatrix}$ 나타남 : 발산

λ-Jordan block with $|\lambda| < 1$

- 예를 들어, G 에 λ-Jordan block $\begin{pmatrix} \lambda & 0 & 0 \\ 1 & \lambda & 0 \\ 0 & 1 & \lambda \end{pmatrix}$ 가 있으면,

$$G^m = \begin{pmatrix} \lambda^m & 0 & 0 \\ m\lambda^{m-1} & \lambda^m & 0 \\ \frac{m(m-1)}{2}\lambda^{m-2} & m\lambda^{m-1} & \lambda^m \end{pmatrix} \longrightarrow 0$$

Power Method

- PageRank의 경우 $G^{52}S$로 충분하다고 함 (즉 $G^{52}S \approx R$)
  - 임의의 probability vector S로 시작하거나
  - $S = {}^{\mathbf{t}}\left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)$로 시작하거나
  - 기존의 PageRank vector에서 시작하거나
- power method : GS, $G(GS)$, $G(G(GS))$, $\cdots$
- PageRank vector R의 이름들
  - Perron-Frobenius vector, fixed probability v, stationary v, $\cdots$

Markov Chain : an Example

- Pr(city-to-city)=0.90, Pr(suburbs-to-city)=0.02,
  Pr(city-to-suburbs)=0.10, Pr(suburbs-to-suburbs)=0.98,
- $G = \begin{pmatrix} 0.90 & 0.02 \\ 0.10 & 0.98 \end{pmatrix}$, $U = \begin{pmatrix} \frac{1}{6} & -\frac{1}{6} \\ \frac{5}{6} & \frac{1}{6} \end{pmatrix}$, $D = \begin{pmatrix} 1 & 0 \\ 0 & 0.88 \end{pmatrix}$
- $U^{-1}GU = D$, note $\lim_{m\to\infty} D^m = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$

Markov Chain : an Example

- $L = \lim_{m\to\infty} G^m = \lim_{m\to\infty} UD^mU^{-1} = U \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} U^{-1} = \begin{pmatrix} \frac{1}{6} & \frac{1}{6} \\ \frac{5}{6} & \frac{5}{6} \end{pmatrix}$
- 임의의 (!) '인구분포 확률벡터' $S = {}^{\mathbf{t}}(c, s)$ 에 대해,

$$R = \lim_{m\to\infty} G^m S = LS = {}^{\mathbf{t}}(\tfrac{1}{6}, \tfrac{5}{6})$$

  - 현재 인구분포 확률벡터 $= S$
  - 1 년 후 인구분포 확률벡터 $= GS$
  - 2 년 후 인구분포 확률벡터 $= G^2S$
  - $\vdots$
  - ∞ 년 후 인구분포 확률벡터 $= R$

Random Surfer Model

- random surfer 가 임의의 web page 에서 시작해서
- 계속 다른 web page 로 randomly 이동 (이사)
  - 현재 page 의 link 를 click 하거나
  - link 를 click 하지 않고 임의의 다른 page 로 이동하거나
- g_{ij} = j-page 에서 i-page 로 옮겨갈 (이사갈) 확률
  - 단, $G = (g_{ij})$
- R 은 ∞ 번 이동 후 surfer 의 '위치 확률벡터'
  - surfer 는 중요한 (rank 가 높은) page 에 있을 확률이 크다

Random Surfer Model

- 통계조사에 따르면?!
  - surfer가 현재 page의 link를 click할 확률 85%
  - link를 click하지 않고 다른 page로 이동할 확률 15%
- damping factor $d = 0.85$ (2차 땜질)를 위한 변명 ……?
  - 아니면 진짜 천재적인 발상?
  - $g_{ii} \neq 0$은 어떻게 해석? (자기 집에서 자기 집으로 이사?)

참고: Perron-Frobenius Theory

- A가 positive matrix이면
  - unique dominant eigenvalue $\lambda > 0$
  - a positive dominant eigenvector
  - $\text{mult}_A(\lambda) = 1$
- A가 non-negative matrix이면, $\cdots\cdots$.
- 참고도서
  - C. D. Meyer, Matrix analysis and applied linear algebra

질문

- 질문 1 : keyword search 먼저 ? 전체 rank vector 계산 먼저 ?
  - keyword search 먼저라면 n 은 별로 크지 않음
  - 전체(n=수억) rank vector 는, 예를 들어, 하루 한 번만 계산 ?
  - 언어별 rank 따로따로 ?
  - search for perfect matches only ?
- 질문 2 : webpage 방문자 숫자는 무시하고, link 만 고려 ?
  - '가짜 존경' (광고 등) 어떻게 제거 ?
- 질문 3 : SCI impact factor · · · · · · ? SNS follower 숫자 ?