선형대수학 소개

# “선형대수학과 구글(Google) 검색엔진” – 페이지랭크 알고리즘

이상구 (성균관대학교)[1]

영국 파이낸셜타임스(FT)와 컨설팅그룹 밀워드브라운이 공동 조사한 ‘2010 글로벌 100대 브랜드 기업’을 보면 랭킹 1위는 미국의 인터넷 기업 구글로, 브랜드 가치가 1,143억달러를 기록해 2007년부터 4년 연속 최고의 자리를 지켰다. [2] 최근 구글의 모토로라 인수가 삼성과 한국의 경제에 큰 영향을 줄 것이라는 언론 보도를 접하였듯이 격변하는 환경에서 ‘모든 학생은 왜 수학을 배워야 하는가?’에 대한 답의 하나로 본 원고에서는 경제와 생활에 막대한 영향을 미치는 “구글 검색 엔진 속의 수학이론”에 대하여 소개한다.

구글(Google)의 역사는 Matrix Computation의 저자 Gene Golub이 강의하던 스탠퍼드대의 20대 초반의 대학원생 래리 페이지(Larry Page)와 세르게이 브린(Sergey Brin)이 1995년 기존의 검색엔진들이 보여주는 정리되지 않는 결과물에 불만을 가지고 새로운 검색기법 연구에 몰두하여 마침내 페이지랭크(PageRank) 기술과 링크(link)의 매칭기술을 개발하는 데 성공하면서 시작된다. 이들은 ‘이 사이트가 링크를 만들어 연결시켜 줄 얼마만큼의 가치를 가지고 있을까?’라는 기준을 가지고 검색된 사이트들의 순위를 매기는 구글 특유의 차별화된 검색방법을 확립한다.

페이지와 브린이 인터넷의 광대한 정보를 구글이 모두 담겠다는 의지를 담고, 미국인 수학자 Edward Kasner의 조카인 Milton Sirota가 10의 100승을 뜻하는 말로 만든 'googol'이라는 신조어를 변형시켜 처음 구글이란 말을 쓴 것은 1997년이었다.

1998년 4월 개최된 월드와이드웹 컨소시엄(W3C)에서 페이지와 브린은 자신들의 연구 결과를 발표하게 되고, 이때부터 검색엔진 업계와 학계에서 관심을 끌기 시작했다. 구글은 사이트가 가지고 있는 메타태그나 키워드에만 의지하지 않고 페이지랭크라는 독특한 기법을 사용하

1) 본 원고는 저자가 수학적모델링 강좌용으로 개발하여, 일부분은 2011년 정민철군의 석사지도에 활용하였음.

2) 2011년에는 애플이 구글을 제치고 브랜드 가치 세계 1위를 차지했다고 보도했다. 구글은 2위(1,724억 달러,약 180조원)로 밀려났다. 유일하게 포함된 국내 기업 삼성은 67위를 차지했다.

여 웹페이지의 공정한 순위를 매긴다.

순수하게 수학적으로 접근한 구글 검색엔진의 새로운 아이디어와 수학적 알고리즘인 구글행렬과 페이지랭크를 계산하는 방법은 인터넷 검색엔진 시장에 획기적인 새 패러다임을 제공하였다. 이를 통하여 학생들이 대학에서 배우는 선형대수학의 기본적인 내용이 생활 속에 널리 이용되고 있음을 확인 할 수 있다.

## 1. 구글 검색엔진의 기본 아이디어

우리는 구글 검색엔진에 사용된 페이지랭크를 통한 웹페이지의 외부인기도가 검색 순위에 반영될 때 페이지랭크 점수가 어떻게 구해지는지 알아보고 그 뒤에 숨어있는 수학적 아이디어의 핵심을 찾아 설명하도록 한다.

웹페이지에 대한 객관적인 순위를 만들어 내기 위하여 구글은 인터넷의 광범위한 구조를 직접 이용한다. 대부분의 다른 검색엔진들은 관계있는 사이트를 결정하기 위해 웹페이지의 제목과 내용을 체크한다. 그러나 구글은 한 웹페이지에서 다른 웹페이지로 연결하는 링크가 있으면, 그 링크를 일종의 투표로 본다. 많이 투표된 웹페이지를 중심으로 구글이 평가를 하게 된다. 구글의 페이지랭크는 어떤 웹페이지가 다른 웹페이지와 밀접한 관계가 있는지를 결정하고 관련된 구조를 표현하기 위해 하이퍼링크(Hyperlink)[3] 행렬을 만들고 행렬의 가장 큰 고유값을 이용해서 검색에서 가장 근접한 사이트들을 찾아내는 것이다. 이 과정에서 구글은 Power Method(거듭제곱법)와 페이지랭크 연산법을 이용한다.

구글과 같은 검색엔진이 수행하는 기본적인 업무중 첫 번째는 인터넷에 상주하며 사용자들이 접근하는 웹페이지를 파악하는 것이다. 두 번째 작업은 파악된 웹페이지에 대한 데이터를 수집하는 것이다. 수집된 데이터들은 검색어와 관련된 단어나 문구를 검색하는데 효과적으로 사용하게 된다. 세 번째 단계는 데이터로 수집한 웹페이지에 중요도(중요도: 웹페이지가 가지고 있는 중요성을 다른 웹페이지와 비교하여 점수로 표현한 것)를 기록하고 사용자가 검색을 할 때 수집하여 가지고 있던 웹페이지들의 데이터 중에 좀 더 중요한 자료를 검색결과의 상단에 보여주는 것이다.

우리는 이 세 번째 과정에서 수집된 웹페이지 데이터들의 집합에서 각각의 웹페이지에 대한 중요도를 어떠한 방법으로 결정하며 그 비율을 어떻게 정했는지에 대해 알아본다.

## 2. 페이지랭크(PageRank)공식의 건설

구글의 페이지랭크라는 개념은 쉽게 이야기해서 '사람들이 링크를 많이 거는 웹페이지는 사람들이 많이 찾아가는 곳일 테고, 그 만큼 정확한 정보가 있는 웹페이지일 것이므로 웹페이지의 페이지랭크 점수를 높게 주자.' 라는 이론이다.

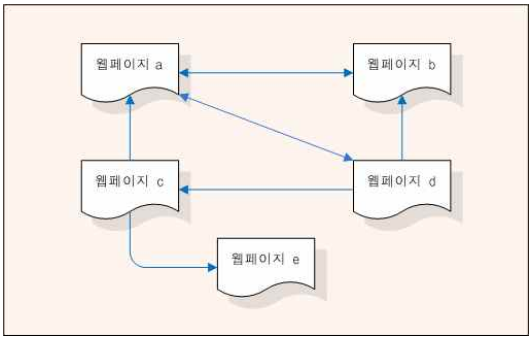
페이지랭크는 임의의 방문자가 어떤 웹페이지를 방문하는 빈도로 이해될 수도 있다. 새로운 웹페이지는 자신이 담고 있는 링크의 적절성에 따라 페이지랭크를 증가시키기도 하고, 감소시키

3) 하이퍼링크(Hyperlink): <컴퓨터 전문용어> 하이퍼텍스트 문서 내의 단어나 구, 기호, 화상과 같은 요소를 그 하이퍼텍스트 내의 다른 요소 또는 다른 하이퍼텍스트 내의 다른 요소와 연결한 것. 하이퍼링크 행렬을 링크행렬이라 부르기로 한다.

기도 한다. (Avrachenkov & Litvak, 2005)

이제 구글 검색엔진안의 수학적 모델에 대하여 살펴보자.

**1단계: 인접(adjacency) 행렬의 건설**




[그림 1] 5개 웹페이지로 이루어진 간단한 웹

위의 그림은 간단한 웹을 표현한 것이다. 주어진 웹페이지들로부터 얻어진 위의 연결관계를 다음과 같이 $n\times n$ 인접행렬 $A$를 사용하여 나타낼 수 있다.(Page; Brin; Motwani & Winograd, 1998)

$$A=\begin{bmatrix}0&1&1&1&0\\1&0&0&1&0\\0&0&0&1&0\\1&0&0&0&0\\0&0&1&0&0\end{bmatrix} \qquad (2)$$

이 경우 [그림 1]의 웹페이지 e는 다른 웹페이지로의 링크를 가지고 있지 않으므로 행렬 $A$의 마지막 열은 0의 값을 갖는다. dangling node[4]라 불리는 외부로의 링크를 가지지 않는 웹페이지는 특별한 케이스이다. 게다가 인접행렬을 만들 때 각 웹페이지의 자기 자신으로의 링크는 무시하므로 행렬 $A$의 대각선 성분은 모두 0의 값을 갖는 것을 알 수 있다.

**2단계: 열정규화된 인접행렬 $H$의 건설**

행렬 $A$의 각각의 열 $A_j$를 $A_j$의 성분들의 합으로 나누어 새로운 행렬 $H$를 얻는다. 이 행렬을 열정규화된 인접행렬이라고 한다. 행렬 $H$의 열을 $H_j$라 하면 다음과 같은 공식에 의해 얻어진다.

$$H_j=\frac{A_j}{\sum_{k=1}^{n}A_{kj}},\quad j=1,\cdots,n$$

이 과정을 식 (2)의 행렬 $A$에 적용하면

$$H=\begin{bmatrix}0&1&\frac{1}{2}&\frac{1}{3}&0\\\frac{1}{2}&0&0&\frac{1}{3}&0\\0&0&0&\frac{1}{3}&0\\\frac{1}{2}&0&0&0&0\\0&0&\frac{1}{2}&0&0\end{bmatrix} \qquad (3)$$

를 얻는다.

**3단계: 행렬 $H$의 열 stochastic화**

전 단계에서 얻어진 행렬 $H$가 바로 페이지랭크 알고리즘에 이용되는 것은 아니다. 그 이유는 페이지랭크 알고리즘의 수렴성을 확보하기 위해서는 stochastic 행렬이 필요한데, 행렬 $H$는 dangling node에 의해 열stochastic 행렬이 아닐 수 있기 때문이다. 열stochastic은 '행렬의 모든 성분이 음수가 아니어야 하고 열의 합이 모두 1이어야 한다'를 만족해야 한다. 일단 위의 행렬 $H$는 dangling node인 웹페이지에 의해 마지막 열의 합이 0이므로 열 stochastic 행렬이 아니다. 그래서 행렬 $H$로부터 모든 열의 합을 1이 되는 열 stochastic 행렬 $S$를 다음과 같이 정의 한다. $S=H+\frac{\mathbf{e}\mathbf{a}^T}{n}$

4) dangling node: 다른 웹페이지들과 연결이 끊겨 버린 상태

여기서 벡터 $\mathbf{e}$는 모든 성분이 1인 열벡터이고, $\mathbf{a}$는 $\sum_{i=1}^{n} H_{ij} = 0$이면 $a_j = 1$이고 $\sum_{i=1}^{n} H_{ij} \neq 0$이면 $a_j = 0$인 열벡터이다. 그러면 행렬 $H$로 부터 열 stochastic행렬 $S$를 건설할 수 있다. Gerschgorin정리에 의해 열 stochastic 행렬(혹은 Markov행렬) $S$의 가장 큰 고유값의 크기 $\rho(S)$는 1보다 작거나 같다. 또한 $S-I$의 행벡터(혹은 열벡터)의 합은 0이 되는데, 이것은 trivial sum이 아니다. 따라서 $\det(S-I)=0$으로 부터 어떤 $i \in 1,\ldots,n$ 에 대하여 $\lambda_i = 1$인 고유값이 존재함을 알 수 있다. 그런데 문제점은 링크행렬이 크기가 매우 큰 sparse행렬이라는 것이다. 따라서 일반적인 계산을 통해 고유벡터를 구하기보다는 거듭제곱법(Power method)를 사용하는 것이 효율적이다. 우선 식 (3)의 행렬 $H$를 사용하여 행렬 $S$를 만들면 다음과 같이 된다.

$$S = \begin{bmatrix} 0 & 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{5} \\ \frac{1}{2} & 0 & 0 & \frac{1}{3} & \frac{1}{5} \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{5} \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{5} \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{5} \end{bmatrix} \qquad (4)$$

**4단계: 구글행렬 $G$의 건설**

행렬 $S$가 유일한 stationary distribution 벡터를 갖는다는 것을 보증할 수 없기 때문에 아직 끝난 것이 아니다. (즉, 정확한 하나의 페이지랭크 벡터가 없을 수도 있다는 의미) 따라서 수렴하는 하나의 stationary distribution 벡터 $\mathbf{x}$가 있다는 것을 보장하기 위해 행렬 $S$가 irreducible인 동시에 stochastic임을 확실히 해야 한다. 정의에 의하여 "주어진 정사각행렬 $A$에 대하여, $P^T A P = \begin{pmatrix} X & Y \\ 0 & Z \end{pmatrix}$인 치환 행렬 $P$와 정사각행렬 $X$, $Z$가 존재하지 않을 때 행렬 $A$는 irreducible이다." 행렬이 irreducible인 것은 대응하는 그래프가 strongly connected하다는 것과 동치이다.(Horn & Johnson, 1985) 또 $A$가 primitive 행렬이면, 언제나 $A$는 irreducible 행렬이다. 이에 보태서 주어진 정사각행렬 $A$가 primitive일 필요충분조건은 어떤 자연수 $k$에 대하여 $A^k$가 양의 행렬이 되는 것이다.[5] 그래서 우리는 식 (4)의 행렬 $S$를 이용하여 irreducible한 열 stochastic 행렬 $G$를 다음과 같이 정의한다.

$$G = mS + (1-m)E \qquad (5)$$

여기서 $m$은 $0 \leq m \leq 1$이고 $E = \frac{\mathbf{e}\mathbf{e}^T}{n}$이다. 구글검색에서는 보통 $m$을 0.85로 이용한다. 이 행렬 $G$가 바로 우리가 필요로 했던 구글행렬이 된다.

이제 페이지랭크를 구하는 기본 알고리즘인 식(1) $\mathbf{x}_{k+1} = H\mathbf{x}_k$에 행렬 $H$대신 행렬 $G$를 대입하여 새 페이지랭크 알고리즘을 다음과 같이 정의한다. $\mathbf{x}_{k+1} = G\mathbf{x}_k$

그러나 크기가 1인 고유값이 2개 이상 존재하면 거듭제곱법의 수렴성이 보장되지 않는다. 또한 행렬의 주위에 블록 $O$행렬이 있다면, 페이지랭킹에서 아예 제외되는 웹페이지가 생기는데, 이런 경우 현실 문제를 적절히 반영한다고 볼 수 없다. 따라서 우리가 다루기를 원하는 행렬의

5) http://matrix.skku.ac.kr/sglee/perron_frobenius/perron_frobenius.html

의미를 모두 담고 있으면서 가장 큰 고유값이 1개뿐인(대수적 중복도가 1인)행렬인 primitive 행렬로 변환시킨 것이 구글행렬이다. 이 부분의 이론에 대하여 소개하고 다음 단계로 넘어간다.

집합 $\sigma(G)$를 행렬 $G$의 고유값들의 집합, 정사각행렬$G$의 spectral radius를 $\rho(G)=\max_{\lambda\in\sigma(G)}|\lambda|$라 하면 Perron-Frobenius정리에 의해 구글행렬 $G=mS+(1-m)E$에 대하여 $\rho(G)$=1인 고유값 1은 $G$의 Gerschgorin circle 상의 유일한 고유값이고, 고유값 1의 대수적 중복도는 1임을 알게 된다. 그리고 $\lambda=\rho(G)$는 행렬 $G$의 유일한 가장 큰 고유값이 된다. 더구나 대응하는 그래프는 strongly connected이고 어떤 자연수 $k$에 대하여 $G^k$가 양의 행렬이 되는 irreducible행렬 $G$는 primitive 행렬이다. 즉, 구글 행렬 $G=mS+(1-m)E$가 primitive 행렬이라는 의미이다.(Horn & Johnson, 1985)

여기서 $\lambda>0$이고 $\lambda\in\sigma(G)$인 대수적 중복도가 1인 가장 큰 고유값 $\lambda=\rho(G)$를 $G$의 Perron 근(root)라 하고, $G\mathbf{x}=\lambda\mathbf{x}$인 양의 고유벡터 $\mathbf{x}>0$가 존재한다. 그리고 $G\mathbf{x}=\lambda\mathbf{x}$이면서 $\mathbf{x}>0$이고 $\|\mathbf{x}\|_1=1$인 $G$의 고유벡터를 $G$의 Perron 벡터 라 한다.

위의 건설 과정에서 우리가 건설한 구글행렬이 수렴하는 유일한 페이지랭크 벡터를 갖는다는 것을 보인 셈이다.

이제 수렴성을 분석해 보자. 만일 행렬 $P$가 ($G$와 마찬가지로) primitive 행렬이라 하자. 그러면 $\lim_{k\to\infty}P^k$가 존재한다. 특히, $r=\rho(B)$가 이 행렬 $B$의 Perron 근이고, $\mathbf{p}$ 와 $\mathbf{q}$ 를 각각 primitive 행렬 $B$ 와 $B^T$의 Perron 벡터 라 할 때 $\lim_{k\to\infty}(B/r)^k=\frac{\mathbf{p}\mathbf{q}^T}{\mathbf{q}^T\mathbf{p}}$임을 알고 있다.(Meyer, 2000) 이를 이용하면 $\mathbf{x}$가 $P$의 (right) Perron 벡터라 하고, $\mathbf{e}/n$를 $P^T$의 Perron 벡터라 하면, $\lim_{k\to\infty}P^k=\frac{(\mathbf{e}/n)\mathbf{x}^{\mathrm{T}}}{\mathbf{x}^{\mathrm{T}}(\mathbf{e}/n)}=\frac{(\mathbf{e})\mathbf{x}^T}{\mathbf{x}^T(\mathbf{e})}=\mathbf{e}\mathbf{x}^T>0$를 얻는다. 벡터 $\mathbf{e}$는 모든 성분을 1로 갖는 열 벡터이다.

이것으로부터 $P\mathbf{x}=\mathbf{x}$ 즉, $\mathbf{x}$가 $P$의 (right) Perron 벡터임이 확인되고, 따라서 $\mathbf{x}^TP^T=\mathbf{x}^T$가 되어 $\mathbf{x}^T$는 $P^T$의 (left) Perron 벡터임이 확인된다. 따라서 $\|\mathbf{x}_0\|_1=1$인 임의의 초기벡터 $\mathbf{x}_0$ 에 대하여 $P^k\mathbf{x}_0=\mathbf{x}_k$이며, primitive 행렬 $P$에 대하여 $\lim_{k\to\infty}P^k=\mathbf{e}\mathbf{x}^T$이고 $\lim_{k\to\infty}\mathbf{p}_k=\mathbf{x}$이다. $P\mathbf{x}=1\mathbf{x}$인 유일한 단위 벡터 $\mathbf{x}$가 존재한다는 의미이다.

수렴성을 보여주는 위의 설명은 구글행렬 $G$가 벡터 $\mathbf{x}$를 (right) Perron 벡터로 가지는 것과 그에 대응하는 대수적 중복도가 1인 고유값 $\lambda=\rho(G)=1$을 가진다는 것을 확인해 준다. 그리고 이 알고리즘을 이용하면 반복연산을 통하여 $\mathbf{x}_{k+1}\approx G\mathbf{x}_k$ 를 만족하는 근사값을 페이지랭크 벡터 $\mathbf{x}$로 언제나 구할 수 있다는 것이다. 이 $\mathbf{x}$가 구글행렬 $G$의 Perron root라 하고, $G\mathbf{x}=\lambda\mathbf{x}$인 양의 고유벡터 $\mathbf{x}>0$가 존재한다. 그리고 $G\mathbf{x}=\lambda\mathbf{x}$이면서 $\mathbf{x}>0$이고 $\|\mathbf{x}\|_1=1$인 행렬 $G$의 고유벡터는 행렬 $G$의 페이지랭크 벡터이고, Perron 벡터이다. 따라서 행렬 $G$의 페이지랭크 벡터를 찾는 것은 행렬 $G$의 (dominant right) 고유벡터 또는 $G^T$의 (dominant left) 고

유벡터, 즉, Perron 벡터를 찾는 것과 동치이다.

이제 구글행렬 $G$의 페이지랭크 벡터를 구하는 문제를 생각하자, 구글은 수십억대의 홈페이지를 다루므로, 실제로 그런 크기의 행렬 $G$를 만들어 Perron 벡터를 구하는 것은 현실과 다르다. 인터넷상의 웹페이지는 적어도 80억 이상의 개수를 가지므로 간단하게 계산되어질 수 없다. 따라서 실제로는 페이지랭크 벡터를 아래에 설명하는 거듭제곱법을 이용하여 구한다.(Langville & Meyer, 2004)

**단계 5: 거듭제곱법(Power method)**

$$\begin{aligned}\mathbf{x}_k &= G\mathbf{x}_{k-1}\\ &= \left(mS+(1-m)\frac{\mathbf{e}}{n}\mathbf{e}^T\right)\mathbf{x}_{k-1}\\ &= mS\mathbf{x}_{k-1}+(1-m)\frac{\mathbf{e}}{n}\mathbf{e}^T\mathbf{x}_{k-1}\\ &= mS\mathbf{x}_{k-1}+(1-m)\frac{\mathbf{e}}{n}\\ &= m\left(H+\frac{\mathbf{e}\mathbf{a}^T}{n}\right)\mathbf{x}_{k-1}+(1-m)\frac{\mathbf{e}}{n}\\ &= mH\mathbf{x}_{k-1}+\frac{\mathbf{e}}{n}\left(m\mathbf{a}^T\mathbf{x}_{k-1}+(1-m)\right)\end{aligned}$$

거듭제곱법(이상구, 2009)을 이용한 수렴속도는 $\tau$를 $\mathbf{x}_{(k)}-\mathbf{x}_{(k-1)}$라고 할 때 $\log_{10}\tau/\log_{10}m$ 임과 페이지와 브린이 $m=0.85$로 놓으면 대개 50번에서 100번의 연산을 통하여 원하는 페이지랭크 벡터를 구할 수 있음을 보였다.(Langville & Meyer, 2003)

이제 어떻게 페이지랭크 벡터가 계산되는지를 확인해 보자. 첫째 웹페이지 $i$의 페이지랭크는 페이지랭크 벡터 $\mathbf{x}$의 $i$번째 성분인 값 $x_i$이다. 여기서 벡터 $\mathbf{x}$는 $\|\mathbf{x}\|_1=1$인 Perron 벡터이다. 이 벡터 $\mathbf{x}$를 통해 웹상에서의 페이지 $i$의 순위를 가늠해 볼 수 있는 점수를 구한다.

위의 전 과정을 요약하면 다음과 같다. 웹의 연결성으로부터 얻은 행렬 $A$의 각 성분을 $H_j=\dfrac{A_j}{\sum_{k=1}^{n}A_{kj}}$를 이용하여 정규화한 행렬 $H$를 만들게 된다. 그리고 행렬 $H$로부터 열 stochastic 행렬인 $S$를 $S=H+\dfrac{\mathbf{e}\mathbf{a}^T}{n}$로 정의하고, 이 때 벡터 $\mathbf{e}$는 모든 성분이 1인 열(column) 벡터이고, 벡터 $\mathbf{a}$는 $\sum_{i=1}^{n}H_{ij}=0$이면 $a_j=1$이고 $\sum_{i=1}^{n}H_{ij}\neq 0$이면 $a_j=0$인 열벡터이다. 이로부터 여기서 $m$은 $0\le m\le 1$의 값인 0.85로 하고 $E=\dfrac{\mathbf{e}\mathbf{e}^T}{n}$로 정의하여 구글행렬 $G=mS+(1-m)E$ 를 얻은 것이다. $m$을 0.85로 선택한 것은 임의로 한 것이 아니다. 이와 관련된 자세한 내용은 (Haveliwala & Kamvar, 2003)에 소개 되었다.

이와 같이 문제 표현형식을 바꾸면, 웹페이지의 순위를 매기기 위한 문제가 정사각행렬에서 고유벡터를 찾는 우리가 잘 아는 문제로 바뀌게 된다.(정의에 의해 행렬 $G$의 고유값 $\lambda$와 고유벡터 $\mathbf{x}$는 방정식 $G\mathbf{x}=\lambda\mathbf{x}$를 만족한다.) 위 행렬 $G$를 구글행렬이라 한다. 이제는 열 stochastic 행렬 $G$의 가장 큰 고유값 1에 대응하는 고유벡터 $\mathbf{x}$를 구하는 일이 남는다.

위의 전 과정을 예를 들어 설명하면 다음과 같다. 미리 언급하였듯이 구글에서는 $m=0.85$을 사용한다. 그래서 구글에서 사용하는 $m=0.85$의 값을 사용하여 얻은 식 (5)의 행렬 $S$를 이용하여 행렬 $G$를 구하면 아래와 같다.

$$G=\begin{bmatrix} \frac{3}{100} & \frac{22}{25} & \frac{91}{200} & \frac{47}{150} & \frac{1}{5} \\ \frac{91}{200} & \frac{3}{100} & \frac{3}{100} & \frac{47}{150} & \frac{1}{5} \\ \frac{3}{100} & \frac{3}{100} & \frac{3}{100} & \frac{47}{150} & \frac{1}{5} \\ \frac{91}{200} & \frac{3}{100} & \frac{3}{100} & \frac{3}{100} & \frac{1}{5} \\ \frac{3}{100} & \frac{3}{100} & \frac{91}{200} & \frac{3}{100} & \frac{1}{5} \end{bmatrix}$$

페이지랭크가 담고 있는 내용을 정리하면 웹페이지 $i$의 페이지랭크는 $\|\mathbf{x}\|_1=1$인 Perron 벡터 $\mathbf{x}$의 $i$번째 $x_i$성분의 값이고 페이지랭크 벡터는 각 웹페이지에 접근하게 될 가능성을 반영하는 Probability 벡터이다.(Langville & Meyer, 2003)

우리가 사용하고 있는 인터넷의 모든 사이트(웹페이지)가 $n$개의 웹페이지로 만들어진 웹이라 하자. 그러면 인터넷 사용자들은 어떠한 웹페이지를 시작페이지로 설정하여 인터넷을 이용하게 될 것이다.

인터넷을 이용하던 중에 $r$개의 링크를 가지고 있는 어느 특정 웹페이지를 열어보게 된다면 그 웹페이지에서 $\frac{m}{r}$의 확률(구글에서는 $m$을 0.85를 사용하였다.)을 가지고 웹페이지가 가지고 있는 임의의 링크를 선택을 하게 된다. 웹페이지에 있는 링크를 선택하지 않을 확률은 $\frac{1-m}{n}$이 되고 인터넷 사용자가 선택할 수 있는 모든 경우의 수는 두 확률로 합으로 계산이 된다. 즉, $r\frac{m}{r}+n\frac{(1-m)}{n}=1$ (여기서 $r$은 웹페이지가 가진 링크의 개수, $n$은 인터넷 웹상의 모든 웹페이지의 개수)가 된다.

구글의 페이지랭크 알고리즘에서 기본 전제로 한 '중요한 웹페이지는 다른 웹페이지로부터 더 많이 연결되어있다.'에 따라 인터넷 사용자는 자주 페이지랭크 점수가 높은 웹페이지로 연결되는 링크를 더 자주 선택하게 되고 페이지랭크 점수가 높은 페이지에 많은 인터넷 사용자가 접근을 하게 된다.

접근을 하는 횟수가 다른 웹페이지에 비하여 많다는 것은 바꿔 생각하면 그만큼 인터넷 사용자들이 해당 웹페이지에 머물게 된다는 것이다. 즉, 식 $\mathbf{x}=G\mathbf{x}$에서 정규화 된 벡터 $\mathbf{x}$의 성분 $x_i$는 인터넷 사용자들이 웹페이지 $i$에서 머무르는 짧은 시간을 의미하는 것으로 판단 생각할 수 있게 된다. 링크구조를 데이터베이스로 가지는 검색엔진의 설계에서 링크구조는 고정적이다. 고정된 링크구조를 가지는 웹에서는 한번 정의된 $\mathbf{x}$는 계속해서 사용할 수 있다. 하지만 인터넷에서의 웹페이지는 단 하나의 구조로 고정되어 있지 않는다. 수많은 사이트와 웹페이지는 생성되었다가 사라지기도 하고 그들 사이를 잇는 링크들 또한 필요성에 따라 계속해서 생겨나고 사라지는 동적인 공간이기 때문에 웹페이지의 페이지랭크 계산을 통한 고유벡터 $\mathbf{x}$의 개선은 매우 중요한 문제가 된다. 페이지랭크 기술은 키워드별로 인터넷 사용자가 클릭하는 웹페이지의 경로를 알고리즘화해 웹페이지들 간의 상호 관련성을 계산해내고, 웹페이지간의 관련성과 웹페이지내의 관련어 배치 등을 고려해 고객이 원하는 결과를 가장 빠르고 정확하게 제공하는 기술이다. 구글 검색엔진의 경우 어떤 내용을 검색창에 넣어도 0.5초면 자신이 찾는 가장 근접한 검색결과를 제시해준다. 이 기능은 아직 어떤 검색엔진도 따라올 수 없는 독보적 기술로 평가받고 있다.

현재 구글은 다양한 분야로 사업을 확장하고 있다. 수학자는 '구글 북스 라이브러리 프로젝트'[6]에 대하여도 관심을 가져야 한다. 구글의 목표 중 하나가 20세기에 발간된 모든 책을 전자화 한다는 것이라고 들었다. 구글은 이미 일본 게이오대학과 미국 하버드대학 등 세계 40곳 이상의 도서관 및 3만 곳 이상의 출판사와 연계해 지금까지 1500만 권 이상을 전자화했다. 2011년 3월에는 영국의 대영도서관 소장 장서 25만 권을 전자화하기로 합의했다고 발표했다. 이런 변화가 수학을 하는 우리에게 조만간 또 다른 큰 영향을 미칠 것이라는 것을 느낄 수 있다.

## 3. 결론

인류 역사의 혁명적 발전 시기마다 그 사회가 안고 있던 문제를 해결하는 과정의 중심에 수학이 있었다. 우리나라의 학생들은 보통 수학공부를 하는 이유를 '입학시험을 잘 보기위해서'라고 생각하고 있다. 따라서 수학은 우리의 삶에 불필요한 과목이라고 생각하기도 한다. 본 원고에서는 구글 검색 엔진 속의 존재하는 선형대수학 이론의 일부를 간략하게 소개했다. 이는 수학의 발전이 인류 사회의 발전 역사와 함께한다는 것을 보여주는 좋은 예가 될 수 있다.

## 4. 참고문헌


*이상구 (2009).* 현대선형대수학 3판, 서울 : 경문사

*Avrachenkov K. & Litvak N. (2005).* The Effect of New Links on Google PageRank http://wwwhome.math.utwente.nl/~litvakn/StochModels06.pdf

*Brin S.; Page L.; Motwami R. & Winograd T. (1998).* The PageRank Citation Ranking: Bringing Order to the Web, Technical report, Computer Science Department, Stanford University, Stanford, CA

*Haveliwala T. H. & Kamvar S. D. (2003).* The Second Eigenvalue of the Google Matrix, Stanford University Technical Report http://www.stanford.edu/~sdkamvar/papers/secondeigenvalue.pdf

*Horn R. & Johnson C.R. (1985).* Matrix Analysis, Cambridge, 1985 ISBN 0521305861

*Langville A. N. & Meyer C. D. (2003).* Fiddling with PageRank http://meyer.math.ncsu.edu/Meyer/PS_Files/FiddlingPageRank.pdf

*Langville A. N. & Meyer C. D. (2004).* The Use of the Linear Algebra by Web Search Engines http://meyer.math.ncsu.edu/Meyer/PS_Files/IMAGE.pdf

*Langville A. N. & Meyer C. D. (2005).* Deeper inside PageRank, Internet Math.

*Meyer C. D. (2000).* Matrix Analysis and Applied Linear Algebra, SIAM, Philadelphia.


■

6) http://en.wikipedia.org/wiki/Google_Books_Library_Project