Google's PageRank[1]

Data Science in Quantitative Finance

Ivailo Dimov and Petter Kolm

November 3 & 10, 2020

Readings

In addition to the lecture notes, the following are required readings:

- Chapter 3 and 4 in Langville and Meyer (2011) (available online via NYU Libraries)

Recommended reading:

- Perron-Frobenius: Chapter 15.2 in Langville and Meyer (2011)
- Markov chains: Chapter 15.3 in Langville and Meyer (2011)
- Details and proofs of the Perron and Perron-Frobenius theorems, as well as a detailed discussion of Markov chains, see Meyer (2000)

Google's PageRank: Part I

The Elements of a Search Engine

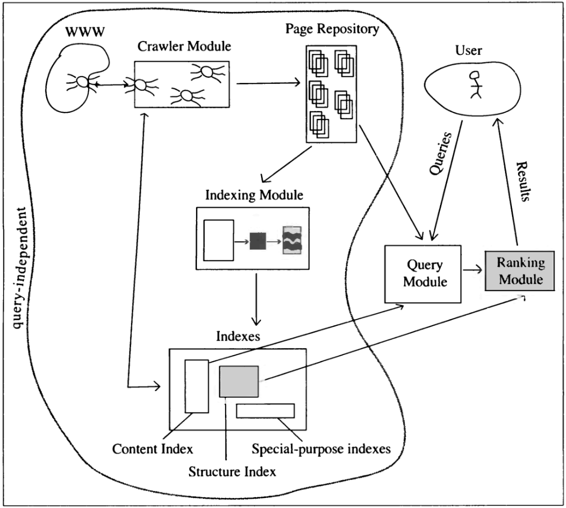

Figure 1: The elements of a search engine. (Source: Langville and Meyer (2011).)

The Web as a Graph

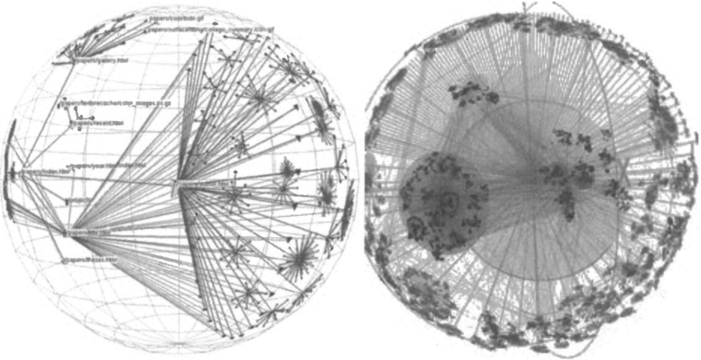

Figure 2: The web as a graph. (Source: Langville and Meyer (2011).)

Intuition: How Google PageRank Works I

Core idea:

- A webpage is important if it is linked to by other important pages, that in turn are linked to other important pages

Intuition: How Google PageRank Works II

- In the late nineties, the founders of Google, Sergey Brin and Larry Page, recognized the web graph's potential to build a search engine
- How can the web graph be used to build a search engine?
  - View "inlink" hyperlinks as "recommendations" of a page
    - A hyperlink from one homepage to someone else's page is an endorsement of the page
    - Therefore, pages with more recommendations (i.e. more inlinks) must be more important than a page with fewer inlinks
  - However, the status of the recommender is also important
    - For example, a personal endorsement from Bill Gates probably does more to strengthen a job application than 20 endorsements from 20 unknown teachers and colleagues
    - However, if the job interviewer learns that Bill Gates is very generous with his praises of employees, and he (or his secretary) has written over 40,000 recommendations in his life, then his recommendations would decrease in importance

Intuition: How Google PageRank Works III

- It makes sense that the weights signifying the status of a recommender must be lowered for recommenders who are less selective. To account for this, we can lower the weight of each endorsement based on the total number of recommendations made by the recommender

Intuition: How Google PageRank Works IV

- We may also want *query independence*
  - This means that the popularity score for each page is determined offline and remains constant (until the next update) regardless of the query
  - Consequently, at query time, when milliseconds are precious, no time is spent computing the popularity scores for relevant pages. Instead, the scores are "looked up" in a pre-computed table
  - In contrast, *query dependence* means the popularity score is computed at the time of each search

The Original Summation Formula for PageRank I

The PageRank of a page P_i, denoted $r(P_i)$, is the sum of the normalized PageRanks of all the pages pointing to it, that is

$$r(P_i) = \sum_{P_j \in \mathcal{B}_{P_i}} \frac{r(P_j)}{|P_j|}$$

where
$\mathcal{B}_{P_i}$ is set of pages pointing to P_i, and
$|P_j|$ is the number of outlinks from page P_j.

How do we solve this equation?

The Original Summation Formula for PageRank II

We could try an iterative procedure:

1. Set $r_0(P_i) = 1/n$ (where n is the number of pages in the index)
2. For each i, iteratively compute

$$r_{k+1}(P_i) = \sum_{P_j \in \mathcal{B}_{P_i}} \frac{r_k(P_j)}{|P_j|}$$

   until convergence

Will this work? What do you think?

The Original Summation Formula for PageRank III

Let us introduce some matrix notation for the summation formula

$$r_{k+1}(P_i) = \sum_{P_j \in \mathcal{B}_{P_i}} \frac{r_k(P_j)}{|P_j|}\,.$$

We define the matrix $\mathbf{H}$ via

$$\mathbf{H}_{ij} := \begin{cases} \frac{1}{|P_i|}\,, & \text{if there is a link from page } i \text{ to page } j\,, \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{1}$$

and the vector

$$\boldsymbol{\pi} = \begin{pmatrix} r(P_1) \\ \vdots \\ r(P_n) \end{pmatrix}.$$

Then the iterative scheme for the summation formula becomes

$$\boldsymbol{\pi}'_{k+1} = \boldsymbol{\pi}'_k \mathbf{H}\,.$$

Example: From the Web Graph to the $\mathbf{H}$ Matrix I

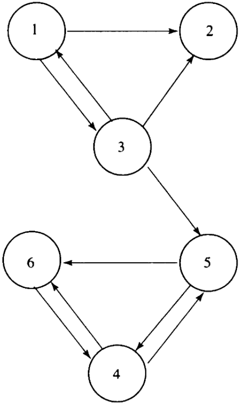

Figure 3: From the web graph to the $\mathbf{H}$ matrix. (Source: Langville and Meyer (2011).)

$$\mathbf{H} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Example: From the Web Graph to the $\mathbf{H}$ Matrix II

We observe that the matrix $\mathbf{H}$ is similar to a stochastic transition probability matrix for a Markov chain. In particular,

- rows corresponding to *nondangling nodes* result is stochastic rows, i.e. rows where the elements sum to 1; and
- rows corresponding to *dangling nodes* (nodes with no outlinks) result in rows with zeros, and hence are non-stochastic rows.

$\mathbf{H}$ is called a *sub stochastic* matrix.

Convergence of the Iterative Scheme I

So let us ask this question again: Does the iteration

$$\boldsymbol{\pi}'_{k+1} = \boldsymbol{\pi}'_k \mathbf{H} \, ,$$

with $\mathbf{H}$ sub stochastic converge?

What do you think?

Convergence of the Iterative Scheme II

No, we do not expect the iteration to converge in this case:

- Dangling nodes lead to a singular matrix.
- Rank sinks: Pages that accumulate more and more PageRank at each iteration (so-called *rank sinks*). For example, here is a graph where the dangling note 3 is a rank sink.

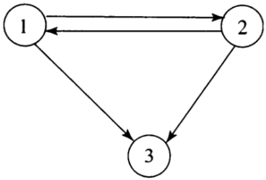

Figure 4: Rank sinks. (Source: Langville and Meyer (2011).)

Convergence of the Iterative Scheme III

- Cycles: Prevent convergence as the ranks “flip flop.”

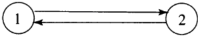

Figure 5: Cycles. (Source: Langville and Meyer (2011).)

Convergence of the Iterative Scheme IV

Take-away: The PageRank convergence problems caused by sinks and cycles can be overcome if $\mathbf{H}$ is modified slightly so that it satisfies the properties above.

In "ML-speak," we need to *regularize* $\mathbf{H}$. The properties 1-3 above hint at how we could perform the regularization.

Convergence of the Iterative Scheme V

It turns out that the iterative scheme

$$\boldsymbol{\pi}'_{k+1} = \boldsymbol{\pi}'_k \mathbf{P} \,,$$

converges to a unique positive vector if[2]

1. $\mathbf{P}$ is stochastic (each row sums up to one),
2. $\mathbf{P}$ is irreducible, and
3. $\mathbf{P}$ is aperiodic.

To obtain this result, we need the *Perron-Frobenius theorem* and some results from *discrete-time finite-state Markov chains*.

[2] Note that aperiodicity plus irreducibility implies primitivity.

Perron-Frobenius

Definitions I

- A matrix $\mathbf{A}$ is *nonnegative* when $a_{ij} \geq 0$ for all i, j. We denote this by $\mathbf{A} \geq 0$.
- A matrix $\mathbf{A}$ is *positive* when $a_{ij} > 0$ for all i, j. We denote this by $\mathbf{A} > 0$.
- The *spectral radius* of a matrix is defined by

$$\rho(\mathbf{A}) = \max \{|\lambda_1|, \ldots, |\lambda_n|\}$$

  where λ_i are the eigenvalues of $\mathbf{A}$.
- We denote by $\sigma(\mathbf{A})$ the set of all eigenvalues of $\mathbf{A}$. $\sigma(\mathbf{A})$ is also referred to as the *spectrum* of matrix $\mathbf{A}$.

Perron's Theorem for Positive Matrices I

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{A} > 0$ with $r = \rho(\mathbf{A})$. Then we have:

1. $r > 0$.
2. $r \in \sigma(\mathbf{A})$ (i.e. r is an eigenvalue of $\mathbf{A}$).
3. r is a simple eigenvalue (i.e. it does not appear as multiple roots of the characteristic polynomial).
4. There exists a positive eigenvector $\mathbf{x} > 0$ such that $\mathbf{A}\mathbf{x} = r\mathbf{x}$.
5. The positive eigenvector $\mathbf{p} > 0$ satisfying $\mathbf{A}\mathbf{p} = r\mathbf{p}$ and $\|\mathbf{p}\|_1 = 1$ is unique. Except for positive multiples of $\mathbf{p}$, there are no other nonnegative eigenvectors for $\mathbf{A}$ (regardless of the eigenvalue).
6. r is the only eigenvalue on the spectral circle of $\mathbf{A}$.
7. r can be computed by the Collatz-Wielandt formula

$$r = \max_{\mathbf{x} \in \{\mathbf{x} | \mathbf{x} \geq 0 \text{ with } \mathbf{x} \neq 0\}} \min_{\substack{1 \leq i \leq n \\ x_i \neq 0}} \frac{(\mathbf{A}\mathbf{x})_i}{x_i}.$$

Perron's Theorem for Positive Matrices II

- The eigenvalue r is called the *Perron root*, and
- the eigenvector $\mathbf{p}$ is called the *Perron vector*.

Perron's Theorem for Nonnegative Matrices

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{A} \geq 0$ with $r = \rho(\mathbf{A})$. Then we have:

1. $r \in \sigma(\mathbf{A})$ (i.e. r is an eigenvalue of $\mathbf{A}$) but $r = 0$ is possible.
2. There exists a nonnegative eigenvector $\mathbf{x} \geq 0$ such that $\mathbf{A}\mathbf{x} = r\mathbf{x}$.
3. r can be computed by the Collatz-Wielandt formula

$$r = \max_{\mathbf{x} \in \{\mathbf{x} | \mathbf{x} \geq 0 \text{ with } \mathbf{x} \neq 0\}} \min_{\substack{1 \leq i \leq n \\ x_i \neq 0}} \frac{(\mathbf{A}\mathbf{x})_i}{x_i} .$$

In other words, this theorem says that a portion of Perron's theorem for positive matrices can be extended to nonnegative matrices by sacrificing the existence of a positive eigenvector for a nonnegative one.

Irreducible Matrices I

Definition

$\mathbf{A} \in \mathbb{R}^{n \times n}$ is a *reducible matrix* if there exists a permutation matrix $\mathbf{P}$ such that

$$\mathbf{P}'\mathbf{A}\mathbf{P} = \begin{pmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{0} & \mathbf{Z} \end{pmatrix}, \tag{2}$$

where $\mathbf{X}$ and $\mathbf{Z}$ are square matrices.

A matrix is *irreducible* if it is not reducible.

Irreducible Matrices II

Proposition

$\mathbf{A} \in \mathbb{R}^{n \times n}$ is *irreducible* if and only if for each pair of indices (i, j) there is a sequence of entries in $\mathbf{A}$ such that

$$a_{ik_1} \cdot a_{k_1 k_2} \cdot \ldots \cdot a_{k_t j} \neq 0 \,.$$

The Perron-Frobenius Theorem I

Let $\mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{A} \geq 0$, and irreducible. Then we have:

1. $r = \rho(\mathbf{A}) > 0$.
2. $r \in \sigma(\mathbf{A})$ (i.e. r is an eigenvalue of $\mathbf{A}$).
3. r is a simple eigenvalue (i.e. it does not appear as multiple roots of the characteristic polynomial).
4. There exists a positive eigenvector $\mathbf{x} > 0$ such that $\mathbf{Ax} = r\mathbf{x}$.
5. The positive eigenvector $\mathbf{p} > 0$ satisfying $\mathbf{Ap} = r\mathbf{p}$ and $\|\mathbf{p}\|_1 = 1$ is unique. Except for positive multiples of $\mathbf{p}$, there are no other nonnegative eigenvectors for $\mathbf{A}$ (regardless of the eigenvalue).
6. r need not to be the only eigenvalue on the spectral circle of $\mathbf{A}$.
7. r can be computed by the Collatz-Wielandt formula

$$r = \max_{\mathbf{x}\in\{\mathbf{x}|\mathbf{x}\geq 0 \text{ with } \mathbf{x}\neq 0\}} \min_{\substack{1\leq i\leq n \\ x_i\neq 0}} \frac{(\mathbf{Ax})_i}{x_i}.$$

The Perron-Frobenius Theorem II

Remark

Note that with irreducibility we are able to recover the results $\#1$, $\#3$, $\#4$ in Perron's theorem for positive matrices.

However, irreducibility alone does not lead to result $\#6$. To recover it, we need one more property of the matrix to be satified.

For you: Provide an example of a nonnegative and irreducible matrix, but with several eigenvalues on the unit circle.

Primitive Matrices I

Definition

$\mathbf{A}$ is *primitive* if and only if $\mathbf{A}^m > 0$ for some $m > 0$

- Nonnegative irreducible matrices are split into two groups:
    1. Primitive: The matrix only has one eigenvalue on its spectral radius.
    2. Imprimitive: The matrix has several eigenvalues on it its spectral radius.
- For the power method to converge, we need the matrix to be primitive. So the rule is, "if $\mathbf{A}$ is a nonnegative, irreducible and primitive square matrix, then (a) it has a simple eigenvalue on the spectral radius and (b) the power methods converges."

Primitive Matrices II

Proposition (A Test for Primitivity)

Let $\mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{A} \geq 0$. If $\mathbf{A}$ is irreducible and has at least one positive diagonal element, then $\mathbf{A}$ is primitive.

Markov Chains

Introduction

Google's PageRank algorithm leverages tools from discrete-time finite-state Markov chains. Therefore, to understand and analyze the mathematics of PageRank, it is helpful to review some of the basic concepts of Markov chains.

Definitions and Concepts I

Definitions

- A *discrete and finite stochastic process* is a set of random variables $\{x_t\}_{t\in\mathbb{N}}$ having a common state space $\mathcal{S} = \{s_1, \ldots, s_n\}$ of discrete states. Typically, we think about t as time, and x_t representing the state of the process at time t
- A *Markov chain* is a stochastic process that satisfies the *Markov property*

$$\begin{gathered}p\left(x_{t+1} = s_j \mid x_t = s_{i_t}, x_{t-1} = s_{i_{t-1}}, \ldots, x_0 = s_{i_0}\right) \\ = p\left(x_{t+1} = s_j \mid x_t = s_{i_t}\right)\end{gathered}$$

  for all t.
- A *stochastic matrix* is a nonnegative matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ where each row sum is equal to 1.

Definitions and Concepts II

- The probability of being in state s_j at time t given that the chain was in state s_i at time $t-1$, is called the *transition probability*

$$p_{ij}(t) := p\left(x_t = s_j \mid x_{t-1} = s_i\right).$$

The matrix $\mathbf{P}(t) \in \mathbb{R}^{n \times n}$ of all transition probabilities is a stochastic matrix. Every Markov chain defines a stochastic matrix. Conversely, every stochastic matrix defines a Markov chain.

- A *stationary Markov chain* (also called a *homogeneous chain*) is a Markov chain in which the transition probabilities do not change with time, that is

$$p_{ij}(t) \equiv p_{ij} \quad \text{and} \quad \mathbf{P}(t) \equiv \mathbf{P}. \tag{3}$$

We will only consider stationary Markov chains here.

Definitions and Concepts III

- An *irreducible Markov chain* is a chain for which the matrix $\mathbf{P}$ is an irreducible matrix. We distinguish two cases:
  - A *periodic Markov chain* is an irreducible chain whose transition probability matrix $\mathbf{P}$ is an *imprimitive matrix*.[3]
  - An *aperiodic Markov chain* is an irreducible chain whose transition probability matrix $\mathbf{P}$ is a primitive matrix.
- A *probability distribution vector* (or *probability vector*) is a nonnegative row vector $\mathbf{p}' := (p_1, p_2, \ldots, p_n)$ such that $\mathbf{p}'\mathbf{e} = 1$.
- The *k-th step probability distribution vector* for an *n*-state chain is defined as

$$\mathbf{p}(k) := (p_1(k), p_2(k), \ldots, p_n(k))' \quad \text{where} \quad p_j = p\,(x_k = s_j)$$

  The vector $\mathbf{p}(0) := (p_1(0), p_2(0), \ldots, p_n(0))'$ is called the *initial distribution vector*.
- $\boldsymbol{\pi}$ is a *stationary probability distribution vector* for a Markov chain if and only if $\boldsymbol{\pi}'\mathbf{P} = \boldsymbol{\pi}'$.

[3] These chains are called periodic because each state can be occupied only at periodic points in time, where the period is the index of imprimitivity.

Analysis of Markov Chains

Much of Markovian analysis revolves around questions concerning the transient behavior of the chain as well as its limiting behavior. Some standard questions include:

- Characterize the k-th step distribution $\mathbf{p}'(k)$ for any initial distribution vector $\mathbf{p}'(0)$.
- Determine if $\lim_{k\to\infty} \mathbf{p}'(k)$ exists, and if so, find its value (the limiting probability vector).
- etc.

Example

The following example demonstrates that Markov chains can be used to describe someone who is surfing the web.

Example: Consider the process of crawling the Web by successively clicking on links to move from one Web page to another. The state space is the set of all Web pages, and the random variable x_t is the Web page being viewed at time t.

The Markov property asserts that the process is memoryless in the sense that the state of the chain at the next time period depends only on the current state and not on the past history of the chain. For example, the process of surfing the Web is a Markov chain provided that the next page that the Web surfer visits doesn't depend on the pages that were visited in the past (before $t - 1$).

Google's PageRank: Part II

The Stochasticity Adjustment of $\mathbf{H}$

Idea: Replace rows of zeros with $(\frac{1}{n}, \ldots, \frac{1}{n})$.

Formally, we replace $\mathbf{H}$ by the matrix

$$\mathbf{S} := \mathbf{H} + \frac{1}{n}\mathbf{a}\mathbf{e}'$$

where the vector $\mathbf{a}$ is defined as

$$a_i := \begin{cases} 1\,, & \text{if row } i \text{ is a zero row (i.e. dangling node)}\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

and $\mathbf{e} := (1, \ldots, 1)'$. This regularization guarantees that $\mathbf{S}$ is stochastic.

For you: How do you interpret this regularization?

Example: The Stochasticity Adjustment of $\mathbf{H}$

$$\mathbf{H} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

The Primitivity Adjustment of $\mathbf{S}$ I

Next: We need a regularization that removes any cycles.

Here is one way to do this, by replacing $\mathbf{S}$ by the matrix

$$\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\mathbf{E}$$

where $0 < \alpha < 1$ and

$$\mathbf{E} := \frac{1}{n}\mathbf{e}\mathbf{e}'\,.$$

$\mathbf{G}$ is referred to as the *Google matrix* and $\mathbf{E}$ is referred to as the *teleportation matrix*.[4]

With this adjustment, the resulting matrix is both stochastic and primitive. Then, as a primitive matrix is irreducible and aperiodic, a unique stationary vector of the Markov chain exists and can be found by the iterative scheme above.[5]

The Primitivity Adjustment of $\mathbf{S}$ II

Properties of the Google matrix:

- $\mathbf{G}$ is stochastic (sum of two stochastic matrices is stochastic);
- $\mathbf{G}$ is irreducible (as each page is connected to all pages);
- $\mathbf{G}$ is aperiodic (the "self-loops" create aperiodicity; that is $G_{ii} > 0$ for all i; and
- $\mathbf{G}$ is primitive (i.e. $\mathbf{G}^k > 0$ for some k). This implies we can apply the Perron-Frobenius theorem.

---

[4] Intuition: α controls the proportion of time a "random surfer" follows the hyperlinks as opposed to being teleported. The teleportation matrix is uniform, meaning the surfer is equally likely (when being teleported) to jump to any page.

[5] Can you show that $\mathbf{S}$ defined in this way is stochastic, irreducible, aperiodic and primitive?

The Personalization Vector I

More generally we could also define

$$\mathbf{G} = \alpha \mathbf{S} + (1-\alpha)\mathbf{e}\mathbf{v}'$$

with $\mathbf{v}$ a positive vector (i.e. $v_i > 0$ for all i) and $\mathbf{v}'\mathbf{e} = 1$. In this case $\mathbf{v}$ is referred to as the personalization vector.[6]

---

[6] It is interesting to explore the uses of $\mathbf{v}$. Some questions that come to mind include: (1) Should we all be subject to the same ranking of webpages, or should we be able to create or tweak or own? (2) What if the search engine chooses one for us? For example, if you like to surf for pages about financial news and current events on data science, you could construct $\mathbf{v}$ such that v_i is large for pages P_i on those topics and close to zero for all other pages. The resulting "personalized" PageRank vector would then be tailored to your interests. Based on the discussion in Langville and Meyer (2011), this seems to have been Google's original intent. However, personalization makes the search query dependent.

The Power Method Used on $\mathbf{G}$ I

The Google PageRank solution is the stationary vector of the iteration

$$\boldsymbol{\pi}'_{k+1} = \boldsymbol{\pi}'_{k}\mathbf{G}\,.$$

This is the *power method* applied to the Google matrix.

The Power Method Used on $\mathbf{G}$ II

While $\mathbf{G}$ is a dense matrix, we can still perform matrix vector multiplication much more efficiently than $O\left(n^2\right)$

$$
\begin{aligned}
\boldsymbol{\pi}'_{k+1} &= \boldsymbol{\pi}'_k \mathbf{G} \\
&= \alpha \boldsymbol{\pi}'_k \mathbf{S} + \frac{(1-\alpha)}{n} \boldsymbol{\pi}'_k \mathbf{e}\mathbf{e}' \\
&= \alpha \boldsymbol{\pi}'_k \left(\mathbf{H} + \frac{1}{n} \mathbf{a}\mathbf{e}' \right) + \frac{(1-\alpha)}{n} \boldsymbol{\pi}'_k \mathbf{e}\mathbf{e}' \\
&= \alpha \boldsymbol{\pi}'_k \mathbf{H} + \boldsymbol{\pi}'_k (\alpha \mathbf{a} + (1-\alpha)\mathbf{e}) \frac{1}{n} \mathbf{e}'
\end{aligned}
$$

This matrix multiplication is $O(n)$.[7]

[7] The Google matrix is a very sparse matrix. As reported by Langville and Meyer (2011) the average number of a page's outlinks is about 10. This means that the complexity of computing the matrix vector multiplication of $\mathbf{Hx}$ is about $O(10n) = O(n)$ in comparison to $O\left(n^2\right)$ for a dense matrix.

Computational Considerations I

- The iterative method presented here is storage-friendly. In addition to the sparse matrix $\mathbf{H}$ and the dangling node vector $\mathbf{a}$, only one vector, the current iterate $\boldsymbol{\pi}_k$ must be stored.
- How many iterations does the iteration take until it has converged to a usable accuracy? Brin and Page reported in their 1998 papers, and others have confirmed, that only $50 - 100$ power iterations are needed before the iterates have converged, giving a satisfactory approximation to the exact PageRank vector. As a result, it is hard to find a method that can beat it in this situation.

Computational Considerations II

- Why are only about 50 iterations necessary for convergence? Langville and Meyer (2011) discuss this in quite some detail. Briefly, the asymptotic rate of convergence of the iterative scheme is determined by the ratio $|\lambda_2/\lambda_1|$ where λ_1 and λ_2 are the largest and second largest eigenvalues of matrix used in the iteration. For $\mathbf{G}$, $\lambda_1 = 1$ and $|\lambda_2| < 1$. Langville and Meyer (2011, Theorem 4.7.1, p. 46) argue that in most cases $|\lambda_2| \approx \alpha$. In their papers, Brin and Page use $\alpha = 0.85$. Note that $\alpha^{50} = 0.85^{50} \approx 0.000296$ which implies that with 50 iterations we can expect roughly $2 - 3$ places of accuracy in the approximate PageRank vector.

The Power Method I

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a diagonalizable matrix with eigenvalues

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_k| \ .$$

While we will keep this discussion general, recall that for the Google matrix we have $\lambda_1 = 1$.

Note that $|\lambda_1| > |\lambda_2|$ implies that λ_1 must be real.[8]

We can write $\mathbf{A}$ as

$$\mathbf{A} = \lambda_1 \mathbf{G}_1 + \ldots + \lambda_k \mathbf{G}_k$$

where $\mathbf{G}_i$ is the orthogonal projection onto $\mathrm{null}\,(\mathbf{A} - \lambda_i \mathbf{I})$.

The Power Method II

What happens when we take powers of this matrix?

$$\mathbf{A}^m = (\lambda_1 \mathbf{G}_1 + \ldots + \lambda_k \mathbf{G}_k)^m \; .$$

We see that

$$\begin{aligned} \mathbf{A}^m &= \mathbf{A}^{m-1} (\lambda_1 \mathbf{G}_1 + \ldots + \lambda_k \mathbf{G}_k) \\ &= \lambda_1^m \mathbf{G}_1 + \ldots + \lambda_k^m \mathbf{G}_k \, . \end{aligned}$$

Thus, dividing through by λ_1, we obtain

$$\begin{aligned} \left(\frac{\mathbf{A}}{\lambda_1}\right)^m &= \mathbf{G}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^m \mathbf{G}_2 + \ldots + \left(\frac{\lambda_k}{\lambda_1}\right)^m \mathbf{G}_k \\ &\to \mathbf{G}_1 \quad (m \to \infty) \end{aligned}$$

so that for any vector $\mathbf{x}_0$

$$\left(\frac{\mathbf{A} x_0}{\lambda_1}\right)^m \to \mathbf{G}_1 x_0 \in \text{null}\,(\mathbf{A} - \lambda_1 \mathbf{I}) \; . \tag{4}$$

The Power Method III

To summarize, we have the following results:

1. As long as $\mathbf{G}_1\mathbf{x}_0 \neq 0$, $\left(\frac{\mathbf{A}x_0}{\lambda_1}\right)^m$ converges to an eigenvector associated with λ_1.
2. The speed of convergence depends on how fast $\left(\frac{\lambda_2}{\lambda_1}\right)^m \to 0$.[9]

[8] Otherwise $\bar{\lambda}_1$ is another eigenvalue with $|\lambda_1| = |\bar{\lambda}_1|$

[9] Recall that for the Google matrix we have that $\lambda_1 = 1$ and $\lambda_2 \approx \alpha$ so the speed of convergence can be controlled by the choice of α.

The World's Largest Eigenvalue Problem: How Big is n in Practice?

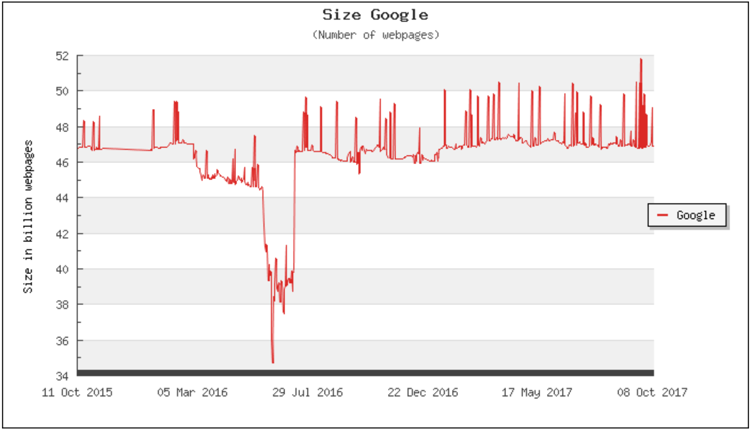

Figure 6: The world's largest eigenvalue problem.[10] (Source: http://www.worldwidewebsize.com.)

[10]You may wonder how often Google crawls a particular site. For more on this topic, see https://goo.gl/ks345K and https://www.google.com/webmasters/tools.

References

Langville, Amy N. and Carl D. Meyer (2011). *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press.

Meyer, Carl D. (2000). *Matrix Analysis and Applied Linear Algebra*. Vol. 71. SIAM.