The PageRank Eigenvalue Problem

Rafikul Alam
Department of Mathematics
Indian Institute of Technology Guwahati
Guwahati - 781039, INDIA

Outline

- The PageRank Problem
- PageRank as an eigenvalue problem
- Power method for PageRank vector
- Single vector iterations for eigenvalue problems

The PageRank problem

The task of extracting information from all the web pages available on the Internet is performed by search engines.

The PageRank problem

The task of extracting information from all the web pages available on the Internet is performed by search engines.

Problem: If it is found, after a query, that more than a thousand web pages contain information that might be relevant to the user's query, then *in which order should the pages be displayed?*

The PageRank problem

The task of extracting information from all the web pages available on the Internet is performed by search engines.

Problem: If it is found, after a query, that more than a thousand web pages contain information that might be relevant to the user's query, then *in which order should the pages be displayed?*

Part of the magic behind Google search engine is its PageRank algorithm, which ranks all the web pages with public access independently of user's query.

The PageRank problem

The task of extracting information from all the web pages available on the Internet is performed by search engines.

Problem: If it is found, after a query, that more than a thousand web pages contain information that might be relevant to the user's query, then *in which order should the pages be displayed?*

Part of the magic behind Google search engine is its PageRank algorithm, which ranks all the web pages with public access independently of user's query.

The PageRank algorithm ranks web pages by solving the PageRank eigenvalue problem:

$$\mathbf{G}\mathbf{v} := \underbrace{(\alpha\mathbf{S} + (1-\alpha)\mathbf{E})}_{\text{Google matrix}}\mathbf{v} = \mathbf{v}.$$

The page with highest rank is displayed first. This is amazingly effective!

Non-negative matrices and vectors

Let $\mathbf{x} \in \mathbb{R}^n$. Then $\mathbf{x}$ is called

- non-negative $(\mathbf{x} \succeq \mathbf{0})$ if $x_j \geq 0$ for $j = 1 : n$.
- positive $(\mathbf{x} \succ \mathbf{0})$ if $x_j > 0$ for $j = 1 : n$.

Non-negative matrices and vectors

Let $\mathbf{x} \in \mathbb{R}^n$. Then $\mathbf{x}$ is called

- non-negative $(\mathbf{x} \succeq \mathbf{0})$ if $x_j \geq 0$ for $j = 1 : n$.
- positive $(\mathbf{x} \succ \mathbf{0})$ if $x_j > 0$ for $j = 1 : n$.

Let $A \in \mathbb{R}^{m \times n}$. Then A is called

- non-negative $(A \succeq \mathbf{0})$ if $A\mathbf{e}_j \succeq \mathbf{0}$ for $j = 1 : n$.
- positive $(A \succ \mathbf{0})$ if $A\mathbf{e}_j \succ 0$ for $j = 1 : n$.

Non-negative matrices and vectors

Let $\mathbf{x} \in \mathbb{R}^n$. Then $\mathbf{x}$ is called

- non-negative $(\mathbf{x} \succeq \mathbf{0})$ if $x_j \geq 0$ for $j = 1 : n$.
- positive $(\mathbf{x} \succ \mathbf{0})$ if $x_j > 0$ for $j = 1 : n$.

Let $A \in \mathbb{R}^{m \times n}$. Then A is called

- non-negative $(A \succeq \mathbf{0})$ if $A\mathbf{e}_j \succeq \mathbf{0}$ for $j = 1 : n$.
- positive $(A \succ \mathbf{0})$ if $A\mathbf{e}_j \succ 0$ for $j = 1 : n$.

Fact: If $A \succ \mathbf{0}$ and $\mathbf{x} \succeq \mathbf{0}$ is nonzero then $A\mathbf{x} \succ \mathbf{0}$.

Non-negative matrices and vectors

Let $\mathbf{x} \in \mathbb{R}^n$. Then $\mathbf{x}$ is called

- non-negative $(\mathbf{x} \succeq \mathbf{0})$ if $x_j \geq 0$ for $j = 1 : n$.
- positive $(\mathbf{x} \succ \mathbf{0})$ if $x_j > 0$ for $j = 1 : n$.

Let $A \in \mathbb{R}^{m \times n}$. Then A is called

- non-negative $(A \succeq \mathbf{0})$ if $A\mathbf{e}_j \succeq \mathbf{0}$ for $j = 1 : n$.
- positive $(A \succ \mathbf{0})$ if $A\mathbf{e}_j \succ 0$ for $j = 1 : n$.

Fact: If $A \succ \mathbf{0}$ and $\mathbf{x} \succeq \mathbf{0}$ is nonzero then $A\mathbf{x} \succ \mathbf{0}$.

Let $\mathbf{e} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^\top$ and $A \in \mathbb{R}^{m \times n}$ be non-negative. Then A is called

- column stochastic if $\mathbf{e}^\top A = \mathbf{e}^\top$, that is, $\mathbf{e}^\top A\mathbf{e}_j = 1$ for $j = 1 : n$.
- column sub-stochastic if $\mathbf{e}^\top A \preceq \mathbf{e}^\top$, that is, $\mathbf{e}^\top A\mathbf{e}_j \leq 1$ for $j = 1 : n$.

Positive eigenpair of a positive matrix

The number $\rho(A) := \max\{|\lambda| : \lambda \in \Lambda(A)\}$ is called the spectral radius of A.

Positive eigenpair of a positive matrix

The number $\rho(A) := \max\{|\lambda| : \lambda \in \Lambda(A)\}$ is called the spectral radius of A. Note that $\rho(A)$ may or may not be an eigenvalue of A.

Positive eigenpair of a positive matrix

The number $\rho(A) := \max\{|\lambda| : \lambda \in \Lambda(A)\}$ is called the spectral radius of A. Note that $\rho(A)$ may or may not be an eigenvalue of A. If λ is called a dominant eigenvalue of A if $|\lambda| = \rho(A)$.

Positive eigenpair of a positive matrix

The number $\rho(A) := \max\{|\lambda| : \lambda \in \Lambda(A)\}$ is called the spectral radius of A. Note that $\rho(A)$ may or may not be an eigenvalue of A. If λ is called a dominant eigenvalue of A if $|\lambda| = \rho(A)$.

For $\mathbf{x} := \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$, define $|\mathbf{x}| := \begin{bmatrix} |x_1| & \cdots & |x_n| \end{bmatrix}^\top$. Similarly, for $A = [a_{ij}]$, define $|A| := [|a_{ij}|]$. Then $|A\mathbf{x}| \leq |A|\,|\mathbf{x}|$.

Positive eigenpair of a positive matrix

The number $\rho(A) := \max\{|\lambda| : \lambda \in \Lambda(A)\}$ is called the spectral radius of A. Note that $\rho(A)$ may or may not be an eigenvalue of A. If λ is called a dominant eigenvalue of A if $|\lambda| = \rho(A)$.

For $\mathbf{x} := \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$, define $|\mathbf{x}| := \begin{bmatrix} |x_1| & \cdots & |x_n| \end{bmatrix}^\top$. Similarly, for $A = [a_{ij}]$, define $|A| := [|a_{ij}|]$. Then $|A\mathbf{x}| \leq |A|\,|\mathbf{x}|$.

Fact: If $A \succ \mathbf{0}$ then the following hold:

- $\rho(A) \in \Lambda(A)$ and $\rho(A) > 0$
- If $A\mathbf{v} = \rho(A)\mathbf{v}$ then $A|\mathbf{v}| = \rho(A)|\mathbf{v}|$ and $|\mathbf{v}| > 0$.

Positive eigenpair of a positive matrix

The number $\rho(A) := \max\{|\lambda| : \lambda \in \Lambda(A)\}$ is called the spectral radius of A. Note that $\rho(A)$ may or may not be an eigenvalue of A. If λ is called a dominant eigenvalue of A if $|\lambda| = \rho(A)$.

For $\mathbf{x} := \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$, define $|\mathbf{x}| := \begin{bmatrix} |x_1| & \cdots & |x_n| \end{bmatrix}^\top$. Similarly, for $A = [a_{ij}]$, define $|A| := [|a_{ij}|]$. Then $|A\mathbf{x}| \leq |A|\,|\mathbf{x}|$.

Fact: If $A \succ \mathbf{0}$ then the following hold:

- $\rho(A) \in \Lambda(A)$ and $\rho(A) > 0$
- If $A\mathbf{v} = \rho(A)\mathbf{v}$ then $A|\mathbf{v}| = \rho(A)|\mathbf{v}|$ and $|\mathbf{v}| > 0$.

Proof: If $A\mathbf{x} = \lambda\mathbf{x}$ and $|\lambda| = \rho(A)$ then $\rho(A)|\mathbf{x}| = |A\mathbf{x}| \leq A|\mathbf{x}|$.

Positive eigenpair of a positive matrix

The number $\rho(A) := \max\{|\lambda| : \lambda \in \Lambda(A)\}$ is called the spectral radius of A. Note that $\rho(A)$ may or may not be an eigenvalue of A. If λ is called a dominant eigenvalue of A if $|\lambda| = \rho(A)$.

For $\mathbf{x} := \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$, define $|\mathbf{x}| := \begin{bmatrix} |x_1| & \cdots & |x_n| \end{bmatrix}^\top$. Similarly, for $A = [a_{ij}]$, define $|A| := [|a_{ij}|]$. Then $|A\mathbf{x}| \leq |A|\,|\mathbf{x}|$.

Fact: If $A \succ \mathbf{0}$ then the following hold:

- $\rho(A) \in \Lambda(A)$ and $\rho(A) > 0$
- If $A\mathbf{v} = \rho(A)\mathbf{v}$ then $A|\mathbf{v}| = \rho(A)|\mathbf{v}|$ and $|\mathbf{v}| > 0$.

Proof: If $A\mathbf{x} = \lambda\mathbf{x}$ and $|\lambda| = \rho(A)$ then $\rho(A)|\mathbf{x}| = |A\mathbf{x}| \leq A|\mathbf{x}|$. Set $\mathbf{y} := A|\mathbf{x}| - \rho(A)|\mathbf{x}|$. Then $\mathbf{y} \succeq \mathbf{0}$. Now $\mathbf{y} \neq \mathbf{0}$ leads to a contradiction.

Positive eigenpair of a positive matrix

The number $\rho(A) := \max\{|\lambda| : \lambda \in \Lambda(A)\}$ is called the spectral radius of A. Note that $\rho(A)$ may or may not be an eigenvalue of A. If λ is called a dominant eigenvalue of A if $|\lambda| = \rho(A)$.

For $\mathbf{x} := \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$, define $|\mathbf{x}| := \begin{bmatrix} |x_1| & \cdots & |x_n| \end{bmatrix}^\top$. Similarly, for $A = [a_{ij}]$, define $|A| := [|a_{ij}|]$. Then $|A\mathbf{x}| \leq |A|\,|\mathbf{x}|$.

Fact: If $A \succ \mathbf{0}$ then the following hold:

- $\rho(A) \in \Lambda(A)$ and $\rho(A) > 0$
- If $A\mathbf{v} = \rho(A)\mathbf{v}$ then $A|\mathbf{v}| = \rho(A)|\mathbf{v}|$ and $|\mathbf{v}| > 0$.

Proof: If $A\mathbf{x} = \lambda\mathbf{x}$ and $|\lambda| = \rho(A)$ then $\rho(A)|\mathbf{x}| = |A\mathbf{x}| \leq A|\mathbf{x}|$. Set $\mathbf{y} := A|\mathbf{x}| - \rho(A)|\mathbf{x}|$. Then $\mathbf{y} \succeq \mathbf{0}$. Now $\mathbf{y} \neq \mathbf{0}$ leads to a contradiction.

Hence $\mathbf{y} = \mathbf{0} \implies A|\mathbf{x}| = \rho(A)|\mathbf{x}| \implies \rho(A) > 0$ and $|\mathbf{x}| > 0$. ■

Positive eigenpair of a positive matrix

The number $\rho(A) := \max\{|\lambda| : \lambda \in \Lambda(A)\}$ is called the spectral radius of A. Note that $\rho(A)$ may or may not be an eigenvalue of A. λ is called a dominant eigenvalue of A if $|\lambda| = \rho(A)$.

For $\mathbf{x} := \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$, define $|\mathbf{x}| := \begin{bmatrix} |x_1| & \cdots & |x_n| \end{bmatrix}^\top$. Similarly, for $A = [a_{ij}]$, define $|A| := [|a_{ij}|]$. Then $|A\mathbf{x}| \leq |A|\,|\mathbf{x}|$.

Fact: If $A \succ \mathbf{0}$ then the following hold:

- $\rho(A) \in \Lambda(A)$ and $\rho(A) > 0$
- If $A\mathbf{v} = \rho(A)\mathbf{v}$ then $A|\mathbf{v}| = \rho(A)|\mathbf{v}|$ and $|\mathbf{v}| > 0$.

Proof: If $A\mathbf{x} = \lambda\mathbf{x}$ and $|\lambda| = \rho(A)$ then $\rho(A)|\mathbf{x}| = |A\mathbf{x}| \leq A|\mathbf{x}|$. Set $\mathbf{y} := A|\mathbf{x}| - \rho(A)|\mathbf{x}|$. Then $\mathbf{y} \succeq \mathbf{0}$. Now $\mathbf{y} \neq \mathbf{0}$ leads to a contradiction.

Hence $\mathbf{y} = \mathbf{0} \implies A|\mathbf{x}| = \rho(A)|\mathbf{x}| \implies \rho(A) > 0$ and $|\mathbf{x}| > 0$. ■

Fact: If $A \succ \mathbf{0}$ then $\rho(A)$ is a simple eigenvalue of A. Thus

$$\text{algebraic multiplity of } \rho(A) = \text{ geometric multiplicity of } \rho(A) = 1.$$

Perron vector and Perron root

Thus $N(A - \rho(A)I) = \mathrm{span}(\mathbf{v})$ and either $\mathbf{v} \succ \mathbf{0}$ or $-\mathbf{v} \succ \mathbf{0}$.

Perron vector and Perron root

Thus $N(A - \rho(A)I) = \mathrm{span}(\mathbf{v})$ and either $\mathbf{v} \succ \mathbf{0}$ or $-\mathbf{v} \succ \mathbf{0}$. Consequently, there is a unique vector $\mathbf{p} \succ \mathbf{0}$ such that

$$A\mathbf{p} = \rho(A)\mathbf{p} \quad \text{and} \quad \mathbf{e}^\top \mathbf{p} = p_1 + \cdots + p_n = 1.$$

Perron vector and Perron root

Thus $N(A - \rho(A)I) = \mathrm{span}(\mathbf{v})$ and either $\mathbf{v} \succ \mathbf{0}$ or $-\mathbf{v} \succ \mathbf{0}$. Consequently, there is a unique vector $\mathbf{p} \succ \mathbf{0}$ such that

$$A\mathbf{p} = \rho(A)\mathbf{p} \quad \text{and} \quad \mathbf{e}^\top \mathbf{p} = p_1 + \cdots + p_n = 1.$$

Definition: The special vector $\mathbf{p}$ is called the Perron vector of $A \succ \mathbf{0}$ and the associated eigenvalue $\rho(A)$ is called the Perron root of A.

Perron vector and Perron root

Thus $N(A - \rho(A)I) = \mathrm{span}(\mathbf{v})$ and either $\mathbf{v} \succ \mathbf{0}$ or $-\mathbf{v} \succ \mathbf{0}$. Consequently, there is a unique vector $\mathbf{p} \succ \mathbf{0}$ such that

$$A\mathbf{p} = \rho(A)\mathbf{p} \quad \text{and} \quad \mathbf{e}^\top \mathbf{p} = p_1 + \cdots + p_n = 1.$$

Definition: The special vector $\mathbf{p}$ is called the Perron vector of $A \succ \mathbf{0}$ and the associated eigenvalue $\rho(A)$ is called the Perron root of A.

Fact: If $A \succ \mathbf{0}$ then there are no nonnegative eigenvectors of A other than the Perron vector $\mathbf{p}$ and its positive scalar multiples.

Perron vector and Perron root

Thus $N(A - \rho(A)I) = \mathrm{span}(\mathbf{v})$ and either $\mathbf{v} \succ \mathbf{0}$ or $-\mathbf{v} \succ \mathbf{0}$. Consequently, there is a unique vector $\mathbf{p} \succ \mathbf{0}$ such that

$$A\mathbf{p} = \rho(A)\mathbf{p} \quad \text{and} \quad \mathbf{e}^\top \mathbf{p} = p_1 + \cdots + p_n = 1.$$

Definition: The special vector $\mathbf{p}$ is called the Perron vector of $A \succ \mathbf{0}$ and the associated eigenvalue $\rho(A)$ is called the Perron root of A.

Fact: If $A \succ \mathbf{0}$ then there are no nonnegative eigenvectors of A other than the Perron vector $\mathbf{p}$ and its positive scalar multiples.

Proof: Let $A\mathbf{y} = \lambda \mathbf{y}$ and $\mathbf{y} \succeq \mathbf{0}$. Let $\mathbf{x} \succ 0$ be the Perron vector of $A^\top$. Then $\mathbf{x}^\top \mathbf{y} > 0$. Hence

$$\rho(A)\mathbf{x}^\top = \mathbf{x}^\top A \Longrightarrow \rho(A)\mathbf{x}^\top \mathbf{y} = \mathbf{x}^\top A\mathbf{y} = \lambda \mathbf{x}^\top \mathbf{y} \Longrightarrow \rho(A) = \lambda. \blacksquare$$

Perron's Theorem

Theorem: Let $A \succ \mathbf{0}$. Then the following hold:

- $\rho(A) > 0$
- $\rho(A) \in \Lambda(A)$ and $\rho(A)$ is simple ($\rho(A)$ is called Perron root)

Perron's Theorem

Theorem: Let $A \succ \mathbf{0}$. Then the following hold:

- $\rho(A) > 0$
- $\rho(A) \in \Lambda(A)$ and $\rho(A)$ is simple ($\rho(A)$ is called Perron root)
- There is a unique vector $\mathbf{p}$ (called Perron vector) such that $A\mathbf{p} = \rho(A)\mathbf{p}$, $\mathbf{p} \succ \mathbf{0}$ and $\mathbf{e}^\top \mathbf{p} = 1$

Perron's Theorem

Theorem: Let $A \succ \mathbf{0}$. Then the following hold:

- $\rho(A) > 0$
- $\rho(A) \in \Lambda(A)$ and $\rho(A)$ is simple ($\rho(A)$ is called Perron root)
- There is a unique vector $\mathbf{p}$ (called Perron vector) such that $A\mathbf{p} = \rho(A)\mathbf{p}, \;\; \mathbf{p} \succ \mathbf{0}$ and $\mathbf{e}^\top \mathbf{p} = 1$
- There are no nonnegative eigenvectors of A other than the Perron vector $\mathbf{p}$ and its positive scalar multiples

Perron's Theorem

Theorem: Let $A \succ \mathbf{0}$. Then the following hold:

- $\rho(A) > 0$
- $\rho(A) \in \Lambda(A)$ and $\rho(A)$ is simple ($\rho(A)$ is called Perron root)
- There is a unique vector $\mathbf{p}$ (called Perron vector) such that $A\mathbf{p} = \rho(A)\mathbf{p}, \;\; \mathbf{p} \succ \mathbf{0}$ and $\mathbf{e}^\top \mathbf{p} = 1$
- There are no nonnegative eigenvectors of A other than the Perron vector $\mathbf{p}$ and its positive scalar multiples
- $\rho(A)$ is the only eigenvalue on the spectral circle $|z| = \rho(A)$

Perron's Theorem

Theorem: Let $A \succ \mathbf{0}$. Then the following hold:

- $\rho(A) > 0$
- $\rho(A) \in \Lambda(A)$ and $\rho(A)$ is simple ($\rho(A)$ is called Perron root)
- There is a unique vector $\mathbf{p}$ (called Perron vector) such that $A\mathbf{p} = \rho(A)\mathbf{p}, \;\; \mathbf{p} \succ \mathbf{0}$ and $\mathbf{e}^\top \mathbf{p} = 1$
- There are no nonnegative eigenvectors of A other than the Perron vector $\mathbf{p}$ and its positive scalar multiples
- $\rho(A)$ is the only eigenvalue on the spectral circle $|z| = \rho(A)$
- Collatz-Weilandt formula

$$\rho(A) = \max\left\{ \min_{i, x_i \neq 0} \frac{(A\mathbf{x})_i}{x_i} : \mathbf{x} \succeq \mathbf{0} \text{ and } \mathbf{x} \neq \mathbf{0} \right\}$$

Google Search Engine

Google search engine does three basic things:

- Crawl the web and locate all web pages with public access.

Google Search Engine

Google search engine does three basic things:

- Crawl the web and locate all web pages with public access.
- Index the data (create term-document matrix) so that they can be searched efficiently for relevant key words or phrases.

Google Search Engine

Google search engine does three basic things:

- Crawl the web and locate all web pages with public access.
- Index the data (create term-document matrix) so that they can be searched efficiently for relevant key words or phrases.
- Rank each page in the database, so that when a user does a search and all the pages in the database with the desired information have been found, the most relevant pages can be presented first.

Google Search Engine

Google search engine does three basic things:

- Crawl the web and locate all web pages with public access.
- Index the data (create term-document matrix) so that they can be searched efficiently for relevant key words or phrases.
- Rank each page in the database, so that when a user does a search and all the pages in the database with the desired information have been found, the most relevant pages can be presented first.

Google's PageRank algorithm assigns ranks to all the web pages and is formulated as a matrix eigenvalue problem.

World Wide Web as a Graph

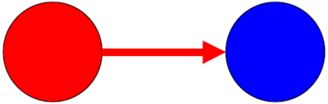
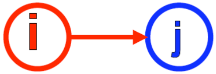


Figure : Link from one web page to another web page.

Web graph:

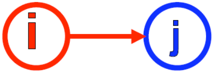
- Nodes = Web pages
- Edges = links

Ranking of web pages



The web is an example of a directed graph. Let all the web pages be ordered as $P_1, \ldots, P_n$. A link from P_i to P_j represents an arrow. Google assigns rank to a page based on its in-links (incoming links) and out-links (outgoing links).

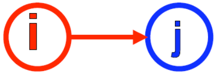
Ranking of web pages



The web is an example of a directed graph. Let all the web pages be ordered as $P_1, \ldots, P_n$. A link from P_i to P_j represents an arrow. Google assigns rank to a page based on its in-links (incoming links) and out-links (outgoing links).

In-links of P_j: All the incoming links (pages) to P_j.
Out-links of P_j: All the outgoing links (pages) from P_j.

Ranking of web pages



The web is an example of a directed graph. Let all the web pages be ordered as $P_1, \ldots, P_n$. A link from P_i to P_j represents an arrow. Google assigns rank to a page based on its in-links (incoming links) and out-links (outgoing links).

In-links of P_j: All the incoming links (pages) to P_j.
Out-links of P_j: All the outgoing links (pages) from P_j.

Google assigns a high rank to a web page if it has inlinks from other pages that have a high rank. This self-referencing statement can be formulated mathematically as an eigenvalue problem.

Ranking of web pages

The web is an example of a directed graph. Let all the web pages be ordered as $P_1, \ldots, P_n$. A link from P_i to P_j represents an arrow. Google assigns rank to a page based on its in-links (incoming links) and out-links (outgoing links).

In-links of P_j: All the incoming links (pages) to P_j.
Out-links of P_j: All the outgoing links (pages) from P_j.

Google assigns a high rank to a web page if it has inlinks from other pages that have a high rank. This self-referencing statement can be formulated mathematically as an eigenvalue problem.

However, a ranking system based only on the number of inlinks is easy to manipulate. Google overcomes this problem as follows.

PageRank eigenvalue problem

Let $x_j \geq 0$ be the rank of page P_j. Then $x_j > x_k \Longrightarrow P_j$ is more important than page P_k.

PageRank eigenvalue problem

Let $x_j \geq 0$ be the rank of page P_j. Then $x_j > x_k \implies P_j$ is more important than page P_k. If page P_j contains n_j out-links, one of which links to page P_k, then page P_k's score is boosted by x_j/n_j.

PageRank eigenvalue problem

Let $x_j \geq 0$ be the rank of page P_j. Then $x_j > x_k \implies P_j$ is more important than page P_k. If page P_j contains n_j out-links, one of which links to page P_k, then page P_k's score is boosted by x_j/n_j.

Thus each web page gets a total of one vote, weighted by that web page's score, that is evenly divided up among all of its outgoing links.

PageRank eigenvalue problem

Let $x_j \geq 0$ be the rank of page P_j. Then $x_j > x_k \implies P_j$ is more important than page P_k. If page P_j contains n_j out-links, one of which links to page P_k, then page P_k's score is boosted by x_j/n_j.

Thus each web page gets a total of one vote, weighted by that web page's score, that is evenly divided up among all of its outgoing links.

Let L_k be the set of in-links of P_k. Then the rank x_k of P_k is defined by

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j} \tag{1}$$

where n_j is the number of outgoing links (outlinks) from page P_j. Setting $\mathbf{v} := [x_1, \ldots, x_n]^T$, equation (1) is rewritten as

PageRank eigenvalue problem

Let $x_j \geq 0$ be the rank of page P_j. Then $x_j > x_k \implies P_j$ is more important than page P_k. If page P_j contains n_j out-links, one of which links to page P_k, then page P_k's score is boosted by x_j/n_j.

Thus each web page gets a total of one vote, weighted by that web page's score, that is evenly divided up among all of its outgoing links.

Let L_k be the set of in-links of P_k. Then the rank x_k of P_k is defined by

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j} \tag{1}$$

where n_j is the number of outgoing links (outlinks) from page P_j. Setting $\mathbf{v} := [x_1, \ldots, x_n]^T$, equation (1) is rewritten as

$$\text{Eigenvalue problem} \quad \mathbf{Hv} = \mathbf{v} \quad ,$$

where $\mathbf{H}$ is the hyperlink matrix and $\mathbf{v}$ is the PageRank vector.

World Wide Web as a Hyperlink matrix

Let n_j be the number of outlinks of the web page P_j. Then

$$\mathbf{H} = \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \dots & \vdots \\ h_{n1} & \cdots & h_{nn} \end{bmatrix},$$

$h_{ij} = 1/n_j$ if there is a link from P_j to page P_i else $h_{ij} = 0$.

World Wide Web as a Hyperlink matrix

Let n_j be the number of outlinks of the web page P_j. Then

$$\mathbf{H} = \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \cdots & \vdots \\ h_{n1} & \cdots & h_{nn} \end{bmatrix},$$

$h_{ij} = 1/n_j$ if there is a link from P_j to page P_i else $h_{ij} = 0$.

Number of nonzero entries in $\mathbf{H}\mathbf{e}_j$ = number of outlinks of P_j.
Number of nonzero entries in $\mathbf{e}_i^\top \mathbf{H}$ = number of inlinks of P_i.

World Wide Web as a Hyperlink matrix

Let n_j be the number of outlinks of the web page P_j. Then

$$\mathbf{H} = \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \cdots & \vdots \\ h_{n1} & \cdots & h_{nn} \end{bmatrix},$$

$h_{ij} = 1/n_j$ if there is a link from P_j to page P_i else $h_{ij} = 0$.

Number of nonzero entries in $\mathbf{He}_j$ = number of outlinks of P_j.
Number of nonzero entries in $\mathbf{e}_i^\top \mathbf{H}$ = number of inlinks of P_i.

Thus $\mathbf{H}$ is non-negative: $\mathbf{H} \succeq 0$ and $\mathbf{He}_k = \mathbf{0}$ if P_k has no outlink.
On the other hand, $\mathbf{e}^\top \mathbf{He}_k = 1$ if the page P_k has outlinks.

Hence $\mathbf{H}$ is column sub-stochastic: $\mathbf{e}^\top \mathbf{He}_j \leq 1$ for $j = 1 : n$.

World Wide Web as a Hyperlink matrix

Let n_j be the number of outlinks of the web page P_j. Then

$$\mathbf{H} = \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \cdots & \vdots \\ h_{n1} & \cdots & h_{nn} \end{bmatrix},$$

$h_{ij} = 1/n_j$ if there is a link from P_j to page P_i else $h_{ij} = 0$.

Number of nonzero entries in $\mathbf{He}_j$ = number of outlinks of P_j.
Number of nonzero entries in $\mathbf{e}_i^\top \mathbf{H}$ = number of inlinks of P_i.

Thus $\mathbf{H}$ is non-negative: $\mathbf{H} \succeq 0$ and $\mathbf{He}_k = \mathbf{0}$ if P_k has no outlink.
On the other hand, $\mathbf{e}^\top \mathbf{He}_k = 1$ if the page P_k has outlinks.

Hence $\mathbf{H}$ is column sub-stochastic: $\mathbf{e}^\top \mathbf{He}_j \leq 1$ for $j = 1 : n$.

Dangling node: If a page P_k has no outlink ($\mathbf{He}_k = \mathbf{0}$) then it is called a dangling node.

World Wide Web as a Hyperlink matrix

Let n_j be the number of outlinks of the web page P_j. Then

$$\mathbf{H} = \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \ldots & \vdots \\ h_{n1} & \cdots & h_{nn} \end{bmatrix},$$

$h_{ij} = 1/n_j$ if there is a link from P_j to page P_i else $h_{ij} = 0$.

Number of nonzero entries in $\mathbf{He}_j$ = number of outlinks of P_j.
Number of nonzero entries in $\mathbf{e}_i^\top \mathbf{H}$ = number of inlinks of P_i.

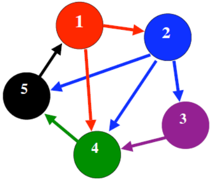
Thus $\mathbf{H}$ is non-negative: $\mathbf{H} \succeq 0$ and $\mathbf{He}_k = \mathbf{0}$ if P_k has no outlink.
On the other hand, $\mathbf{e}^\top \mathbf{He}_k = 1$ if the page P_k has outlinks.

Hence $\mathbf{H}$ is column sub-stochastic: $\mathbf{e}^\top \mathbf{He}_j \leq 1$ for $j = 1 : n$.

Dangling node: If a page P_k has no outlink ($\mathbf{He}_k = \mathbf{0}$) then it is called a dangling node. If there are no dangling nodes then $\mathbf{e}^\top \mathbf{H} = \mathbf{e}^\top$. Hence $\mathbf{H}$ is column stochastic.

Example

Column stochastic hyperlink matrix $\mathbf{H}$.



$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 1 & 0 \end{bmatrix} \text{ and } \mathbf{e}^\top \mathbf{H} = \mathbf{e}^\top.$$

There is a unique PageRank vector $\mathbf{v} \succ 0$ such that $\mathbf{H}\mathbf{v} = \mathbf{v}$ and $\mathbf{e}^\top \mathbf{v} = 1$. The entries of $\mathbf{v}$ are the ranks of web pages.

Nonunique ranking

If $\mathbf{H}$ is column stochastic then $\mathbf{e}^\top \mathbf{H} = \mathbf{e}^\top$ which shows that $\mathbf{e}$ is a left eigenvector of $\mathbf{H}$ corresponding the eigenvalue 1.

Nonunique ranking

If $\mathbf{H}$ is column stochastic then $\mathbf{e}^\top \mathbf{H} = \mathbf{e}^\top$ which shows that $\mathbf{e}$ is a left eigenvector of $\mathbf{H}$ corresponding the eigenvalue 1.

Now $\dim N(\mathbf{H} - I) > 1$ leads to non-unique ranking. For ranking, we require $\dim N(\mathbf{H} - I) = 1$ and a vector $\mathbf{v} \succ \mathbf{0}$ such that $\mathbf{H}\mathbf{v} = \mathbf{v}$ and $\mathbf{e}^\top \mathbf{v} = 1$.

Nonunique ranking

If $\mathbf{H}$ is column stochastic then $\mathbf{e}^\top \mathbf{H} = \mathbf{e}^\top$ which shows that $\mathbf{e}$ is a left eigenvector of $\mathbf{H}$ corresponding the eigenvalue 1.

Now $\dim N(\mathbf{H} - I) > 1$ leads to non-unique ranking. For ranking, we require $\dim N(\mathbf{H} - I) = 1$ and a vector $\mathbf{v} \succ \mathbf{0}$ such that $\mathbf{H}\mathbf{v} = \mathbf{v}$ and $\mathbf{e}^\top \mathbf{v} = 1$.

Theorem: Suppose that $\mathbf{H} \succ \mathbf{0}$ and $\mathbf{e}^\top \mathbf{H} = \mathbf{e}^\top$. Then

- 1 is the simple eigenvalue of $\mathbf{H}$ and there exists $\mathbf{v} \succ 0$ such that $\mathbf{H}\mathbf{v} = \mathbf{v}$. If $\mathbf{v} \in N(\mathbf{H} - I)$ is nonzero then either $\mathbf{v} \succ \mathbf{0}$ or $-\mathbf{v} \succ \mathbf{0}$.

Nonunique ranking

If $\mathbf{H}$ is column stochastic then $\mathbf{e}^\top \mathbf{H} = \mathbf{e}^\top$ which shows that $\mathbf{e}$ is a left eigenvector of $\mathbf{H}$ corresponding the eigenvalue 1.

Now $\dim N(\mathbf{H} - I) > 1$ leads to non-unique ranking. For ranking, we require $\dim N(\mathbf{H} - I) = 1$ and a vector $\mathbf{v} \succ \mathbf{0}$ such that $\mathbf{Hv} = \mathbf{v}$ and $\mathbf{e}^\top \mathbf{v} = 1$.

Theorem: Suppose that $\mathbf{H} \succ \mathbf{0}$ and $\mathbf{e}^\top \mathbf{H} = \mathbf{e}^\top$. Then

- 1 is the simple eigenvalue of $\mathbf{H}$ and there exists $\mathbf{v} \succ 0$ such that $\mathbf{Hv} = \mathbf{v}$. If $\mathbf{v} \in N(\mathbf{H} - I)$ is nonzero then either $\mathbf{v} \succ \mathbf{0}$ or $-\mathbf{v} \succ \mathbf{0}$.
- 1 is the dominant eigenvalue, that is, if $\lambda_2, \ldots, \lambda_n$ are eigenvalues of $\mathbf{H}$ then $|\lambda_j| < 1$ for $j = 2 : n$.

Nonunique ranking

If $\mathbf{H}$ is column stochastic then $\mathbf{e}^\top \mathbf{H} = \mathbf{e}^\top$ which shows that $\mathbf{e}$ is a left eigenvector of $\mathbf{H}$ corresponding the eigenvalue 1.

Now $\dim N(\mathbf{H} - I) > 1$ leads to non-unique ranking. For ranking, we require $\dim N(\mathbf{H} - I) = 1$ and a vector $\mathbf{v} \succ \mathbf{0}$ such that $\mathbf{H}\mathbf{v} = \mathbf{v}$ and $\mathbf{e}^\top \mathbf{v} = 1$.

Theorem: Suppose that $\mathbf{H} \succ \mathbf{0}$ and $\mathbf{e}^\top \mathbf{H} = \mathbf{e}^\top$. Then

- 1 is the simple eigenvalue of $\mathbf{H}$ and there exists $\mathbf{v} \succ 0$ such that $\mathbf{H}\mathbf{v} = \mathbf{v}$. If $\mathbf{v} \in N(\mathbf{H} - I)$ is nonzero then either $\mathbf{v} \succ \mathbf{0}$ or $-\mathbf{v} \succ \mathbf{0}$.
- 1 is the dominant eigenvalue, that is, if $\lambda_2, \ldots, \lambda_n$ are eigenvalues of $\mathbf{H}$ then $|\lambda_j| < 1$ for $j = 2 : n$.
- any other positive eigenvector of $\mathbf{H}$ is a scalar multiple of $\mathbf{v}$.

Adjustment of hyperlink matrix

The web is full of dangling nodes (e.g., pdf flies, image files, etc.). To fix this problem, Brin and Page replaced zero columns of $\mathbf{H}$ with $\frac{1}{n}\mathbf{e}$. Define

$$\mathbf{S} := \mathbf{H} + \frac{1}{n}\mathbf{e}\,\mathbf{a}^\top,$$

where $a_j = 1$ if P_j is a dangling node (i.e., $\mathbf{He}_j = \mathbf{0}$) and $a_j = 0$ else.

Adjustment of hyperlink matrix

The web is full of dangling nodes (e.g., pdf flies, image files, etc.). To fix this problem, Brin and Page replaced zero columns of $\mathbf{H}$ with $\frac{1}{n}\mathbf{e}$. Define

$$\mathbf{S} := \mathbf{H} + \frac{1}{n}\mathbf{e}\,\mathbf{a}^\top,$$

where $a_j = 1$ if P_j is a dangling node (i.e., $\mathbf{H}\mathbf{e}_j = \mathbf{0}$) and $a_j = 0$ else. Then $\mathbf{e}^\top\mathbf{S} = \mathbf{e}^\top$.

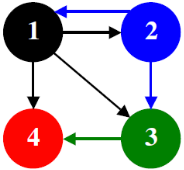
Adjustment of hyperlink matrix

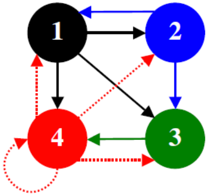
The web is full of dangling nodes (e.g., pdf flies, image files, etc.). To fix this problem, Brin and Page replaced zero columns of $\mathbf{H}$ with $\frac{1}{n}\mathbf{e}$. Define

$$\mathbf{S} := \mathbf{H} + \frac{1}{n}\mathbf{e}\,\mathbf{a}^\top,$$

where $a_j = 1$ if P_j is a dangling node (i.e., $\mathbf{H}e_j = \mathbf{0}$) and $a_j = 0$ else. Then $\mathbf{e}^\top\mathbf{S} = \mathbf{e}^\top$.

The binary vector $\mathbf{a}$ is called the dangling node vector. Note that $\mathbf{S}$ is column stochastic and the adjustment amounts to adding artificial links to the dangling nodes.





The Google matrix

The matrix $\mathbf{S}$, however, cannot guarantee existence of a unique PageRank vector. So, Brin and Page made the final adjustment to obtain the Google matrix

$$\mathbf{G} := \alpha\mathbf{S} + (1-\alpha)\mathbf{E}, \; 0 \leq \alpha \leq 1,$$

where $\mathbf{E} = \dfrac{1}{n}\mathbf{e}\mathbf{e}^{\top}$ is the teleportation matrix which enables jumping to any page.

The Google matrix

The matrix $\mathbf{S}$, however, cannot guarantee existence of a unique PageRank vector. So, Brin and Page made the final adjustment to obtain the Google matrix

$$\mathbf{G} := \alpha\mathbf{S} + (1-\alpha)\mathbf{E}, \ 0 \leq \alpha \leq 1,$$

where $\mathbf{E} = \frac{1}{n}\mathbf{e}\mathbf{e}^\top$ is the teleportation matrix which enables jumping to any page. If $\alpha = 0.6$ then 60% of the time the surfer follows hyperlink structure of web and 40% of the time teleports to a random new page.

The Google matrix

The matrix $\mathbf{S}$, however, cannot guarantee existence of a unique PageRank vector. So, Brin and Page made the final adjustment to obtain the Google matrix

$$\mathbf{G} := \alpha\mathbf{S} + (1-\alpha)\mathbf{E}, \; 0 \le \alpha \le 1,$$

where $\mathbf{E} = \frac{1}{n}\mathbf{e}\mathbf{e}^\top$ is the teleportation matrix which enables jumping to any page. If $\alpha = 0.6$ then 60% of the time the surfer follows hyperlink structure of web and 40% of the time teleports to a random new page.

Fact: The Google matrix $\mathbf{G}$ has the following properties.

- $\mathbf{G}$ is positive $(\mathbf{G} \succ \mathbf{0})$ and column stochastic $(\mathbf{e}^\top\mathbf{G} = \mathbf{e}^\top)$.
- There is a unique vector $\mathbf{v}$ such that $\mathbf{G}\mathbf{v} = \mathbf{v}, \; \mathbf{v} \succ \mathbf{0}$ and $\mathbf{e}^\top\mathbf{v} = 1.$
- $\mathbf{G}$ is the rank-1 update of $\mathbf{H}$:

$$\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\frac{1}{n}\mathbf{e}\mathbf{e}^\top = \alpha\mathbf{H} + \frac{1}{n}\mathbf{e}(\alpha\mathbf{a} + (1-\alpha)\mathbf{e})^\top.$$

The Google matrix

Theorem: Let $\mathbf{G} := \alpha\mathbf{S} + (1-\alpha)\mathbf{E}$ be the Google matrix.

- If $\Lambda(\mathbf{S}) = \{1, \lambda_2, \ldots, \lambda_n\}$ then $\Lambda(\mathbf{G}) = \{1, \alpha\lambda_2, \ldots, \alpha\lambda_n\}$.

The Google matrix

Theorem: Let $\mathbf{G} := \alpha \mathbf{S} + (1-\alpha)\mathbf{E}$ be the Google matrix.

- If $\Lambda(\mathbf{S}) = \{1, \lambda_2, \ldots, \lambda_n\}$ then $\Lambda(\mathbf{G}) = \{1, \alpha\lambda_2, \ldots, \alpha\lambda_n\}$.
- $\mathbf{G}\mathbf{v} = \mathbf{v}$ and $\mathbf{e}^\top \mathbf{v} = 1$ yields

$$\mathbf{v} = \alpha \mathbf{S}\mathbf{v} + (1-\alpha)\frac{1}{n}\mathbf{e} \Longrightarrow (I - \alpha\mathbf{S})\mathbf{v} = (1-\alpha)\frac{1}{n}\mathbf{e}.$$

The Google matrix

Theorem: Let $\mathbf{G} := \alpha \mathbf{S} + (1-\alpha)\mathbf{E}$ be the Google matrix.

- If $\Lambda(\mathbf{S}) = \{1, \lambda_2, \ldots, \lambda_n\}$ then $\Lambda(\mathbf{G}) = \{1, \alpha\lambda_2, \ldots, \alpha\lambda_n\}$.
- $\mathbf{Gv} = \mathbf{v}$ and $\mathbf{e}^\top \mathbf{v} = 1$ yields

$$\mathbf{v} = \alpha \mathbf{S}\mathbf{v} + (1-\alpha)\frac{1}{n}\mathbf{e} \Longrightarrow (I - \alpha\mathbf{S})\mathbf{v} = (1-\alpha)\frac{1}{n}\mathbf{e}.$$

- If $\mathbf{e}^\top \mathbf{x} = 1$ then $\mathbf{Gx} = \alpha\mathbf{Sx} + (1-\alpha)\frac{1}{n}\mathbf{e}$. Hence

The Google matrix

Theorem: Let $\mathbf{G} := \alpha\mathbf{S} + (1-\alpha)\mathbf{E}$ be the Google matrix.

- If $\Lambda(\mathbf{S}) = \{1, \lambda_2, \ldots, \lambda_n\}$ then $\Lambda(\mathbf{G}) = \{1, \alpha\lambda_2, \ldots, \alpha\lambda_n\}$.
- $\mathbf{Gv} = \mathbf{v}$ and $\mathbf{e}^\top\mathbf{v} = 1$ yields

$$\mathbf{v} = \alpha\mathbf{Sv} + (1-\alpha)\frac{1}{n}\mathbf{e} \Longrightarrow (I - \alpha\mathbf{S})\mathbf{v} = (1-\alpha)\frac{1}{n}\mathbf{e}.$$

- If $\mathbf{e}^\top\mathbf{x} = 1$ then $\mathbf{Gx} = \alpha\mathbf{Sx} + (1-\alpha)\frac{1}{n}\mathbf{e}$. Hence

$$\mathbf{Gx} - \mathbf{Gv} = \alpha\mathbf{S}(\mathbf{x}-\mathbf{v}).$$

Consequently, we have

$$\|\mathbf{G}(\mathbf{x}-\mathbf{v})\|_1 = \alpha\|\mathbf{S}(\mathbf{x}-\mathbf{v})\|_1 \leq \alpha\|\mathbf{x}-\mathbf{v}\|_1,$$

where $\|\mathbf{x}\|_1 = |x_1| + \cdots + |x_n|$.

The Power method

Power method: Choose $\mathbf{v}_0 \succeq \mathbf{0}$ such that $\mathbf{e}^\top \mathbf{v}_0 = 1$. Define $\mathbf{u}_j = \alpha \mathbf{S} \mathbf{v}_{j-1} + (1-\alpha)\frac{1}{n}\mathbf{e}$ and $\mathbf{v}_j = \mathbf{u}_j / \mathbf{e}^\top \mathbf{u}_j$ for $j = 1, 2, \ldots$. Then

$$\|\mathbf{v}_j - \mathbf{v}\|_1 \leq \alpha^j \|\mathbf{v} - \mathbf{v}_0\|_1.$$

Algorithm: The Power Method
Input: Vector $\mathbf{v}_0 \succeq \mathbf{0}$ such that $\mathbf{e}^\top \mathbf{v}_0 = 1$ and number of iterations ℓ
Output: PageRank vector $\mathbf{v}_\ell$)

---

for $j = 1 : \ell$
  $\mathbf{u}_j := \alpha \mathbf{S} \mathbf{v}_{j-1} + (1-\alpha)\mathbf{e}/n$ (application of $\mathbf{G}^j$)
  $\mathbf{v}_j := \mathbf{u}_j / \mathbf{e}^\top \mathbf{u}_j$ (normalization)
end

Further reading

Want to know more about Google search engine?

Further reading

Want to know more about Google search engine?

Read the following paper/book.

- K. Bryan and T. Leise, The $25,000,000,000 Eigenvector: The Linear Algebra behind Google, SIAM Review, 48(2006), pp.569-581.
- Amy N. Langville and Carl D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings,* Princeton University Press, 2006

The power method

The power method (or power iteration) is one of the oldest and simplest iterative method for computing an eigenpair of A. It is designed for a matrices A with one dominant eigenvalue. Starting with an arbitrary nonzero vector $v_0 \in \mathbb{C}^n$, the power method computes the sequence

$$v_0,\ Av_0,\ \ldots,\ A^j v_0,\ \ldots$$

Setting $v_j := A^j v_0$, we have $v_j = Av_{j-1}$ for $j = 1, 2, \ldots$.

The power method

The power method (or power iteration) is one of the oldest and simplest iterative method for computing an eigenpair of A. It is designed for a matrices A with one dominant eigenvalue. Starting with an arbitrary nonzero vector $v_0 \in \mathbb{C}^n$, the power method computes the sequence

$$v_0, \; Av_0, \; \ldots, \; A^j v_0, \; \ldots$$

Setting $v_j := A^j v_0$, we have $v_j = Av_{j-1}$ for $j = 1, 2, \ldots.$

Algorithm: The Power Method
Input: Nonzero vector v_0 and the number of iterations ℓ
Output: Dominant eigenpair (μ_ℓ, v_ℓ)

---

for $j = 1 : \ell$
  $w_j := Av_{j-1}$ (application of A^j)
  $v_j := w_j / \|w_j\|_2$ (normalization)
  $\mu_j := v_j^* A v_j$ (Rayleigh quotient)
end

The inverse iteration

The power method can only be used to compute a dominant eigenpair of A. It turns out that a simple *shift-and-invert* strategy can be utilized to compute any desired eigenvalue of A and an associated eigenvector. Assume for the moment that A is invertible. Then

$$Av = \lambda v \Longleftrightarrow A^{-1}v = \frac{1}{\lambda}v.$$

This shows that $1/\lambda$ is the dominant eigenvalue of $A^{-1} \iff \lambda$ is the smallest eigenvalue of A.

The inverse iteration

The power method can only be used to compute a dominant eigenpair of A. It turns out that a simple *shift-and-invert* strategy can be utilized to compute any desired eigenvalue of A and an associated eigenvector. Assume for the moment that A is invertible. Then

$$Av = \lambda v \Longleftrightarrow A^{-1}v = \frac{1}{\lambda}v.$$

This shows that $1/\lambda$ is the dominant eigenvalue of $A^{-1} \iff \lambda$ is the smallest eigenvalue of A.

If s is not an eigenmvalue of A then

$$Av = \lambda v \iff (A - sI)v = (\lambda - s)v \iff (A - sI)^{-1}v = \frac{1}{\lambda - s}v.$$

Thus with the help of a shift s that is closer to a simple eigenvalue λ than to the rest of the spectrum of A, the transformed eigenvalue $(\lambda - s)^{-1}$ can be made dominant for the transformed matrix $(A - sI)^{-1}$. Hence we can apply the power iteration to the shifted matrix $(A - sI)^{-1}$ and arrive at the shifted inverse iteration.

The inverse iteration

Algorithm: Shifted inverse iteration
Input: Nonzero vector v_0, a shift s and the number of iterations ℓ
Output: An eigenpair (μ_ℓ, v_ℓ) such that μ_ℓ closest to s

```
for j = 1:ℓ
    (A − sI)w_j = v_{j−1}      (application of (A − sI)^{−j})
    v_j := w_j/‖w_j‖_2         (normalization)
    μ_j := v_j^* A v_j         (Rayleigh quotient)
end
```

$$\begin{array}{ll}
\text{for } j = 1{:}\,\ell & \\
\quad (A - sI)w_j = v_{j-1} & \text{(application of } (A - sI)^{-j}\text{)} \\
\quad v_j := w_j/\|w_j\|_2 & \text{(normalization)} \\
\quad \mu_j := v_j^* A v_j & \text{(Rayleigh quotient)} \\
\text{end} &
\end{array}$$

The inverse iteration

Algorithm: Shifted inverse iteration
Input: Nonzero vector v_0, a shift s and the number of iterations ℓ
Output: An eigenpair (μ_ℓ, v_ℓ) such that μ_ℓ closest to s

for $j = 1 : \ell$
  $(A - sI)w_j = v_{j-1}$ (application of $(A - sI)^{-j}$)
  $v_j := w_j / \|w_j\|_2$ (normalization)
  $\mu_j := v_j^* A v_j$ (Rayleigh quotient)
end

The computation of w_j requires solution of $(A - sI)w_j = v_j$. Since $(A - sI)$ is not changed during the iteration, we compute LU factorization $P(A - sI) = LU$ only once, where P is a permutation matrix, L is a unit lower triangular matrix and U is an upper triangular matrix.

The inverse iteration

Algorithm: Shifted inverse iteration
Input: Nonzero vector v_0, a shift s and the number of iterations ℓ
Output: An eigenpair (μ_ℓ, v_ℓ) such that μ_ℓ closest to s

---

for $j = 1\colon \ell$
$\quad (A - sI)w_j = v_{j-1} \qquad$ (application of $(A - sI)^{-j}$)
$\quad v_j := w_j/\|w_j\|_2 \qquad$ (normalization)
$\quad \mu_j := v_j^* A v_j \qquad$ (Rayleigh quotient)
end

The computation of w_j requires solution of $(A - sI)w_j = v_j$. Since $(A - sI)$ is not changed during the iteration, we compute LU factorization $P(A - sI) = LU$ only once, where P is a permutation matrix, L is a unit lower triangular matrix and U is an upper triangular matrix.

This will reduce the cost of computation to $2n^2$ flops per iteration. This is an enormous saving as solving the system without exploiting LU factorization would cost $2n^3/3$ flops per iteration. Hence the total cost of performing ℓ iterations of the inverse iteration is $(2n^3/3 + 4n^2\ell)$ flops.

Rayleigh quotient iteration (RQI)

The shifted inverse iteration converges rapidly if the shift parameter $s \in \mathbb{C}$ is close to an eigenvalue of A. The Rayleigh quotient $q(x) := x^*Ax/x^*x$ approximates an eigenvalue A. Hence it makses sense to use Rayleigh quotient as shift in the shifted inverse iteration. The resulting algorithm is called the *Rayleigh quotient iteration* (RQI).

Rayleigh quotient iteration (RQI)

The shifted inverse iteration converges rapidly if the shift parameter $s \in \mathbb{C}$ is close to an eigenvalue of A. The Rayleigh quotient $q(x) := x^*Ax/x^*x$ approximates an eigenvalue A. Hence it makses sense to use Rayleigh quotient as shift in the shifted inverse iteration. The resulting algorithm is called the *Rayleigh quotient iteration* (RQI).

Algorithm: Rayleigh quotient iteration (RQI)
Input: Nonzero vector v_0 and the number of iterations ℓ
Output: Approximate eigenpair (μ_ℓ, v_ℓ)

$v_0 := v_0/\|v_0\|_2$ (normalize v_0)
$\mu_0 := v_0^* A v_0$ (Rayleigh quotient)
for $j = 1:\ell$
  $(A - \mu_{j-1} I)w_j = v_{j-1}$ (application of $(A - \mu_{j-1} I)^{-j}$)
  $v_j := w_j/\|w_j\|_2$ (normalization)
  $\mu_j := v_j^* A v_j$ (Rayleigh quotient)
end

Remark: RQI requires a new LU factorization in each iteration.