# Numerical Computing Cheatsheet

Edoardo Riggio

January 10, 2022

Numerical Computing - SA. 2021
Computer Science
Università della Svizzera Italiana, Lugano

# Contents

# 1 PageRank Algorithm

The PageRank algorithm is entirely determined by the link structure of the World Wide Web. A page will thus have a high rank if other pages with high ranks link to it.

## 1.1 Random Surfer Model

This algorithm is based on the **Random Surfer Model**. Here we imagine a user going from one page to the other by randomly choosing an outgoing link from one page to the next. This process is also called **exploitation**. The problem with exploitation is that it could lead to dead-ends – i.e. a page with no outgoing links, or cycles around cliques of interconnected pages. For this reason, we sometimes choose a random page from the web to navigate to. This process is called **exploration**.

## 1.2 Markov Chains

The random walk generated by the combination of exploitation and exploration is known as a **Markov Chain**. A Markov Chain – or Markov Process, is a stochastic process. Differently from other stochastic processes, it has the property of being memory-less. This means that the probability of future states are not dependent upon the steps that led up to the present state.

Let $W$ be the set of webpages that can be reached by a Markov Chain of hyperlinks, $n$ the number of pages in $W$, ad $G$ the $n \times n$ connectivity matrix of a portion of the Web. The matrix $G$ will be composed as follows:

$$g_{ij} = \begin{cases} 1 & \text{if there is a hyperlink from } i \text{ to } j \\ 0 & \text{otherwise} \end{cases}$$

From matrix $G$ we can determine the in-degree and out-degree of a page $j$. This can be computed as follows:

$$r_i = \sum_j g_{ij}$$

$$c_J = \sum_i g_{ij}$$

Where $r_i$ is the in-degree and $c_j$ is the out-degree. Let now $p$ be the probability that the random walk follows a link – i.e. performs exploitation. A typical value for $p$ is 0.85. Let $\delta$ be the probability that a particular random page is chosen, and $1 - p$ is the probability that some random page is chosen – i.e. perform exploration. Then $\delta$ will have the following formulation:

$$\delta = \frac{1-p}{n}$$

Now let $A$ be a matrix that comes from scaling $G$ by its column sums. The elements of matrix $A$ will be:

$$a_{ij} = \begin{cases} \rho \cdot \frac{g_{ij}}{c_j} + \delta & \text{if } c_j \neq 0 \\ \frac{1}{n} & \text{if } c_j = 0 \end{cases}$$

Matrix $A$ is the transition probability matrix for the Markov Chain – for this reason this matrix is also known as the Markov matrix. All of its elements are strictly between 1 and 0, and its column sums are all equal to 1.

### 1.3 Power Method

The **power method** is an algorithm used in order to produce the dominant eigenvector of matrix $A$. In order to do so, we need to repeat the following computation:

$$x = G \cdot (x + e) \cdot (z + x)$$

Until $x$ settles down to several decimal places.

### 1.4 Inverse Iteration

The **inverse iteration** is an algorithm which has the same goal of the power method. In this case, we need to find the dominant eigenvector of $(A - \alpha)^{-1}$ rather than of $A$. By doing so, the convergence is faster, but the computations are more expensive – this method solves a system of equations, while the power method simply multiplies matrices together.

## 2 Spectral Graph Partitioning

In order to find a partition in a graph, we need to use the second smallest eigenvector – i.e. eigenvector $v_2$. This eigenvector is also known as the **Fiedler eigenvector**, and plays a very important role in graph partitioning. This is because all of the indices corresponding to vector entries can be divided as follows:

$$v_{2i} \in \begin{cases} \text{Set 1} & \text{if } v_{2i} > 0 \\ \text{Set 2} & \text{if } v_{2i} < 0 \end{cases}$$

The arising partition minimizes the number of edges between the two sets. The second smallest eigenvalue $\lambda_2$ is greater than 0 iff we are considering a connected graph. The magnitude of $\lambda_2$ gives us an indication on how well-connected the overall graph is.

## 2.1 Laplacian Matrix

The **Laplacian Matrix** is the matrix representation of a graph. It is computed by doing the following:

$$L = D - A$$

Where $A$ is the adjacency matrix of the graph $G$, and $D$ is the degree matrix of $A$. The **degree matrix** is a diagonal matrix which contains information about the degree of each vertex.

From the Laplacian Matrix, we can compute the eigenvector $w_2$ and the respective eigenvalue $\lambda_2$.

## 2.2 Degree Centrality

The **degree centrality** is defined as the number of links incident upon a node. In other words, it represents the number of vertices a node has.

## 2.3 Reverse Cuthill McKee Ordering and Cholesky Factor

The **Reverse Cuthill McKee Ordering** is an algorithm used to permute a sparse matrix that has a symmetric sparsity pattern, into a band matrix form with a small bandwidth. In other words, the non-zero elements of the matrix get nearer to the diagonal of the matrix.

The **Cholesky Factorization** is the decomposition of a positive-definite matrix into the product of a lower-triangular matrix and its conjugate transpose.

# 3 Graph Partitioning

## 3.1 Spectral Partitioning

The most common example of a global approach to graph partitioning is **spectral partitioning**. This is carried out through a method known as spectral bisection. The spectral method utilizes the spectral graph theorem of linear algebra. This enables the decomposition of a real symmetric matrix into eigenvalues, within an orthonormal basis of eigenvectors.

A spectral bisection is computed using the second eigenvector corresponding with the second eigenvalue of the graph Laplacian Matrix. By thresholding the values of $w_2$ around 0, it will result in two roughly balanced partitions with minimum edgecut. On the other hand, by thresholding the values around the median of $w_2$, it will result in two strictly balanced partitions.

## 3.2 Inertial Bisection

Another common approach is **inertial bisection**. In this case the nodes of the graph are associated with a coordinate list. it thus relies on the geometric layout of the graph.

The main goal of this method is to find a hyperplane running through the centre of gravity of the points. Such a line is chosen in a way that the sum of squares of the distances of the nodes to the line is minimized.

## 3.3 Recursive Partitioning

**Recursive bisection** is a technique which is highly dependent on the decisions made during the earlier stages of the partitioning process. Furthermore, it lacks of global information.

On the other hand, **K-way partitioning** starts with the partitioning of a small set of vertices. After this, the obtained partition is projected back towards the original set of nodes in order to refine it.

The main difference between these two recursive approaches is that k-way partitioning stores global information about the graph – i.e. it takes into consideration the whole graph when creating the partitions, while recursive partitioning does not.