

# R5.A.12/R5.B.10 Modélisations mathématiques

Séquence 2 : Valeurs propres - CM3 : PageRank et SVD

Thibault Godin, Lucie Naert

IUT de Vannes

28 septembre 2024

# Avancement

- ▶ Semaine 1 : Mariages stables avec Gale-Shapley
- ▶ Semaine 2 : Mariages stables équitables avec Selkow
- ▶ Semaine 3 : Flots et Affectations avec Edmonds-Karp
- ▶ Semaine 4 : Initiations aux valeurs propres
- ▶ Semaine 5 : Clustering spectral et découpage de vaches
- ▶ Semaine 6 : Classement des pages Web avec PageRank
- ▶ Semaine 7 : Compression et débruitage d'images avec SVD
- ▶ Semaine 8 : Évaluation avec sujet surprise

# Plan

Avancement

PageRank

Objectif

Algorithme

Décomposition en valeurs singulières

Valeurs singulières

Application à la compression d'images

# PageRank

PageRank est un algorithme qui permet de classer des pages web par importance relative.

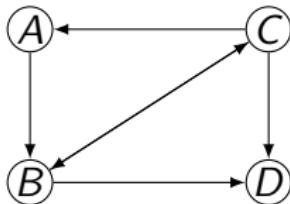
Pour mesurer cette importance, PageRank utilise les liens entre les pages et se basent sur les hypothèses suivantes :

- ▶ Si le site 1 renvoie vers le site 2, le site 1 considère le site 2 comme intéressant.
- ▶ Plus il y a de sites qui renvoient vers le site 2, plus celui-ci doit être important.
- ▶ Si un site important renvoie vers un site 2, le site 2 est sûrement important aussi
- ▶ Si le site 1 renvoie vers beaucoup d'autres sites, il faut répartir l'importance entre ces sites.

## Graphes orientés et matrices d'adjacence

On peut représenter les liens entre des pages web sous forme d'un graphe orienté.

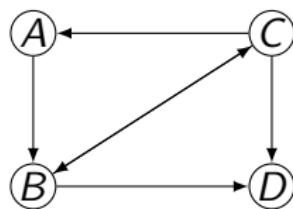
Par exemple : Soient 4 pages web  $A$ ,  $B$ ,  $C$ ,  $D$  dont les liens sont représentés par le graphe orienté ci-dessous :



**Lecture** : Le site  $A$  a un lien vers le site  $B$ ,  $B$  possède des liens vers  $C$  et  $D$ , etc.

## Graphes orientés et matrices d'adjacence

Un tel graphe est représenté par la matrice d'adjacence suivante :



$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

# Plan

Avancement

PageRank

Objectif

Algorithme

Décomposition en valeurs singulières

Valeurs singulières

Application à la compression d'images

# Algorithme de PageRank (version simplifiée)

**Entrées :**

- ▶ Graphe orienté  $G$  (représentant le réseau de sites web)
- ▶ Nombre d'itérations souhaitées  $iter$

**Sortie :** Mesure d'importance pour chaque nœud du graphe

## Déroulé de l'algorithme (version simplifiée)

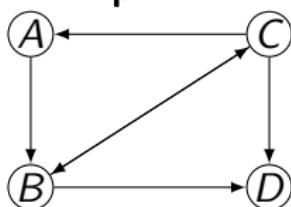
PageRank fonctionne sur le principe d'une marche aléatoire dans le graphe :

1. Nombre d'itération  $i = 0$
2. Choix d'un nœud  $n_0$  au hasard dans  $G$
3. Tant que  $i < iter$  :
  - ▶ choix d'un nouveau nœud  $n_i$  parmi les voisins de  $n_{i-1}$  via l'un de ses liens sortants tiré uniformément.
  - ▶ on conserve le nœud visité dans une liste
  - ▶  $i = i + 1$
4. Calcul de la fréquence d'occupation de chaque nœud.

## Fréquence d'occupation

On appelle **fréquence d'occupation** d'un nœud  $k$ , le nombre de fois où  $k$  a été visité divisé par le nombre d'itérations. Cette fréquence donne l'importance du nœud.

**Exemple :**



Sur 10 itérations, imaginons que la marche aléatoire donne : [ABCBCABDDD]

Les scores d'importance seront les suivants :

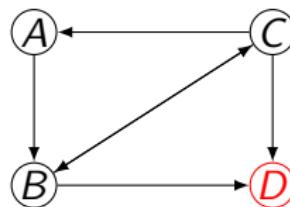
$$f_A = \frac{2}{10} = 0.2, f_B = \frac{3}{10} = 0.3,$$

$$f_C = \frac{2}{10} = 0.2, f_D = \frac{3}{10} = 0.3$$

Que remarquez-vous ?

## Le "hic"

**Problème** : si un nœud ne contient pas de liens sortants, ou si un ensemble de nœuds forme une boucle sans liens sortants, le marcheur aléatoire se retrouve piégé indéfiniment et un site sans grande importance peut se retrouver avec un score énorme.



## Solution

Pour éviter cela, la version native de PageRank modifie légèrement la marche aléatoire présentée plus haut afin qu'à chaque étape, il y ait une certaine probabilité que le marcheur saute vers un nœud aléatoire plutôt que de suivre un lien. On appelle cela une **téléportation**.

Cette probabilité est déterminée par un paramètre,  $\alpha$ , qui est la probabilité de suivre un lien. Ainsi  $1 - \alpha$  est la probabilité de faire un saut aléatoire.

# Algorithme de PageRank (version avec téléportation)

Entrées :

- ▶ Graphe orienté  $G$  (représentant le réseau de sites web)
- ▶ Nombre d'itérations souhaitées  $iter$
- ▶ **Probabilité  $\alpha$  de ne pas se téléporter ( $0 \leq \alpha \leq 1$ )**

Sortie : Mesure d'importance pour chaque nœud du graphe

## Déroulé de l'algorithme (version avec téléportation)

1. Nombre d'itération  $i = 0$
2. Choix d'un nœud  $n_0$  au hasard dans  $G$
3. Tant que  $i < iter$  :
  - ▶ **Tir d'un nombre  $x$  au hasard entre 0 et 1**
    - ▶ si  $x < \alpha$  : choix d'un nouveau nœud  $n_i$  parmi les voisins de  $n_{i-1}$  via l'un de ses liens sortants tiré uniformément.
    - ▶ sinon, **téléportation** : choix d'un nouveau nœud  $n_i$  parmi tous les noeuds du graphe
  - ▶ on conserve le nœud visité dans une liste
  - ▶  $i = i + 1$
4. Calcul de la fréquence d'occupation de chaque nœud.

## Lien avec les valeurs propres

L'implémentation de la marche aléatoire de PageRank est conceptuellement simple, mais pas très efficace à calculer.

Une alternative consiste à utiliser la matrice d'adjacence du graphe et de calculer ses vecteurs propres.

Ce sera vu en TP la semaine prochaine...

# Plan

Avancement

PageRank

Objectif

Algorithme

Décomposition en valeurs singulières

Valeurs singulières

Application à la compression d'images

# Plan

Avancement

PageRank

Objectif

Algorithme

Décomposition en valeurs singulières

Valeurs singulières

Application à la compression d'images

## Valeurs singulières

Les valeurs singulières sont (approximativement) une généralisation des valeurs propres aux matrices rectangulaires.

Toute matrice  $M \in \mathcal{M}_{n,m}(\mathbb{R})$  admet une décomposition en valeurs singulières (SVD).

## Matrice transposée

La transposée d'une matrice  $M \in \mathcal{M}_{n,m}(\mathbb{R})$  est la matrice  $M^\top$  obtenue en échangeant les lignes et les colonnes de  $M$ .

Donner la transposée  $M^\top$  de  $M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$

A quoi est égal  $(M^\top)^\top$  ?

## Matrices orthogonales

$U \in \mathcal{M}_n(\mathbb{R})$  est orthogonale si et seulement elle est inversible et son inverse est égale à sa transposée ce qui peut s'écrire :

$$UU^\top = U^\top U = I_n$$

## Matrices orthogonales

Montrer que  $U = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$  est orthogonale.

## SVD

Toute matrice  $M \in \mathcal{M}_{n,m}(\mathbb{R})$  admet une décomposition en valeurs singulières donnée par :

$$M = U\Sigma V^\top$$

où  $U \in \mathcal{M}_n(\mathbb{R})$ ,  $V \in \mathcal{M}_m(\mathbb{R})$  sont des matrices **orthogonales** et  $\Sigma \in \mathcal{M}_{n,m}(\mathbb{R})$  est une matrice **diagonale** (au sens des matrices rectangulaires) contenant les **valeurs singulières** de  $M$ .

# SVD

$$\underset{n \times m}{\left( \mathbf{M} \right)} = \underset{n \times n}{\left( \mathbf{U} \right)} \underset{m \times m}{\left( \Sigma \right)} \underset{m \times n}{\left( \mathbf{V}^T \right)}$$

## En Python

```
# imports  
import numpy as np  
from numpy import linalg as la  
#déclaration de la matrice à décomposer  
M = ...  
  
# Decomposition en valeurs singulières  
U,S,Vt=la.svd(M)
```

Où  $U$  et  $V^T$  représentent respectivement les matrices orthogonales  $U$  et  $V^T$  et  $S$  est la liste des valeurs singulières de  $M$  triées dans l'ordre décroissant.

# Plan

Avancement

PageRank

Objectif

Algorithme

Décomposition en valeurs singulières

Valeurs singulières

Application à la compression d'images

## Images et matrices

Une image en noir et blanc peut-être représentée sous forme matricielle où chaque nombre représente le niveau de gris d'un pixel ( $0 \rightarrow$  noir,  $255 \rightarrow$  blanc).



```
[ [98 94 90 ... 24 28 22]  
[93 91 89 ... 29 34 21]  
[90 89 88 ... 47 48 39]  
...  
[ 2  4  3 ... 37 43 41]  
[ 2  4  3 ... 41 42 44]  
[ 2  4  3 ... 42 38 43]]
```

## Poids d'une image et taux de compression

Le poids d'une image est défini par le nombre de coefficients de la matrice correspondante. Par exemple, l'image du slide précédent fait 798 par 1 280 pixels donc 1 021 440 coefficients au total.

Le taux de compression  $\rho$  est le ratio entre le poids de l'image compressée et le poids de l'image initiale.

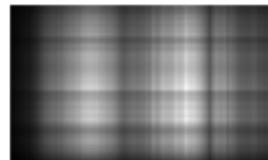
## Compression d'images

Il est possible de réduire le poids d'une image en utilisant sa décomposition SVD par la conservation des  $k$  plus grandes valeurs singulières de la matrice  $\Sigma$  (les autres coefficients sont mis à 0).

On note  $\Sigma^{(k)}$  la matrice ainsi obtenue.



Image initiale (798 valeurs singulières)



1



10



50



100

Nombre de valeurs singulières conservées

## Compression d'images

Concrètement, comme on a conservé uniquement  $k$  plus grandes valeurs singulières pour  $\Sigma$ , nous n'avons plus besoin de stocker certaines informations dans  $U$  et  $V$ .

Plus précisément, dans  $U\Sigma^{(k)}V^\top \approx M$ , on peut ne conserver que les  $k$  premières colonnes de  $U$ , les  $k$  premières lignes et colonnes de  $\Sigma^{(k)}$  et les  $k$  premières lignes de  $V$ .

On notera  $U_k, \Sigma_k^{(k)}$  et  $V_k$  les matrices ainsi obtenue.

# Compression d'images

**Avant compression** - poids de l'image :  $n \times m$

$$\begin{matrix} n \\ \updownarrow \end{matrix} \left( \begin{array}{c} \text{M} \end{array} \right) = \begin{matrix} n \\ \updownarrow \end{matrix} \left( \begin{array}{c} \text{U} \end{array} \right) \left( \begin{array}{c} \Sigma \end{array} \right) \left( \begin{array}{c} \text{V}^T \end{array} \right) \begin{matrix} m \\ \updownarrow \end{matrix}$$

**Après compression** - poids de l'image :  $k(n + k + m)$

$$\begin{matrix} n \\ \updownarrow \end{matrix} \left( \begin{array}{c} \text{M} \end{array} \right) \approx \begin{matrix} n \\ \updownarrow \end{matrix} \left( \begin{array}{c} \text{U}_k \end{array} \right) \left( \begin{array}{c} \Sigma_k^{(k)} \end{array} \right) \left( \begin{array}{c} \text{V}_k^T \end{array} \right) \begin{matrix} m \\ \updownarrow \end{matrix}$$

## Taux de compression

Quel est le taux de compression ?