Algorithmique numérique

Éléments propres et éléments singuliers

Gilles Marait & The IS104 team

12/04/2024

Plan du cours

Rappel sur les complexes

Produit scalaire

Dans un espace hermitien, le produit scalaire de deux vecteurs x et $y \in \mathbb{C}^n$:

$$x \cdot y = \overline{{}^t x} y$$

On appelle trans-conjugué ou adjoint de x, noté x^* :

$$\boxed{x^* = \overline{{}^t x}}$$

Remarque : cette définition permet de concerver les bonnes propriétés du produit scalaire, en particulier $x^* x = ||x||_2^2$, alors que ${}^t xx$ est en général un complexe.

Matrices unitaires

L'équivalent des matrices orthogonales pour les complexes sont les matrices unitaires, qui ont des propriétés similaires :

$$Q^* Q = Id$$

$$Q^* = Q^{-1}$$

...

Plan

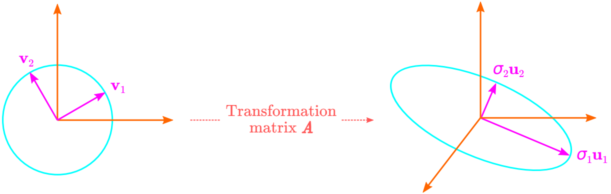
Bordeaux INP AQUITAINE

Plan

Motivation géométrique

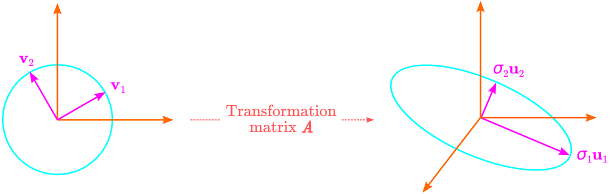
L'image de la sphère unité de $\mathbb{R}^n$ par une matrice $m \times n$ est une hyperellipse dans $\mathbb{R}^m$.

Avec une matrice A de dimensions 3×2, on prend l'image de deux vecteurs orthogonaux de norme 1 : v_1 et v_2



Source : https://pabloinsente.github.io/intro-linear-algebra, Pablo Caceres (modifié)

Définitions



Définitions

On appelle, en supposant (pour le moment) $m \geq n$:

- valeurs singulières : $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ les longueurs des semi-axes de l'hyperellipse.
- vecteurs singuliers à gauche : $\{u_1, u_2, \ldots, u_n\}$ les directions des semi-axes de l'hyperellipse (vecteurs unitaires).
- vecteurs singuliers à droite : $\{v_1, v_2, \ldots, v_n\}$ les vecteurs de la base orthonormale de $\mathbb{R}^n$.

Oui, c'est normal que les vecteurs singuliers à droite soient à gauche sur l'illustration et vice versa.

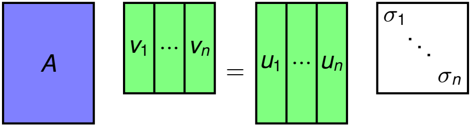
SVD réduite

On a $\forall j \in [1, \ldots, n]$:

$$\boxed{Av_j = \sigma_j u_j}$$

En écriture matricielle :

$$\boxed{AV = \hat{U}\hat{\Sigma}}$$



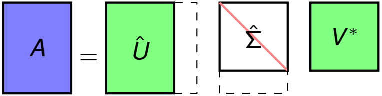
- A une matrice $m \times n$ quelconque
- $\hat{\Sigma}$ est $n \times n$ diagonale, avec des valeurs > 0
- $\hat{U}$ est $m \times n$ avec des colonnes orthonormales
- V est $n \times n$, unitaire

En multipliant à droite par V^*, on peut écrire la décomposition en valeurs singulières réduite :

$$\boxed{A = \hat{U}\hat{\Sigma}V^*}$$

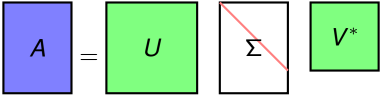
SVD pleine

Comme pour la factorisation QR, on peut compléter la base de $\hat{U}$ pour obtenir une matrice U unitaire :



Qui devient alors :

$$\boxed{A = U \Sigma V^*}$$



- On a U unitaire !
- Plus coûteux à stocker

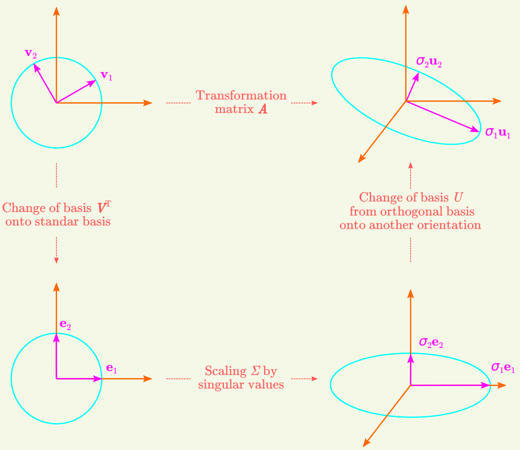
Historique

Découverte indépendante par Eugenio Beltrami (italien) en 1873 et Camille Jordan (français) en 1874.

Description géométrique

Exemple

Avec A de dimensions 3×2 :



Source : https://pabloinsente.github.io/intro-linear-algebra, Pablo Caceres (modifié)

Plan

Définition formelle

Soit $A \in \mathbb{C}^{m \times n}$, pas forcément avec $m \geq n$, sans propriété particulière.
On appelle la décomposition en valeurs singulières de A la factorisation :

$$\boxed{A = U\Sigma V^*}$$

où :

- U est unitaire $\mathbb{C}^{m \times m}$
- V est unitaire $\mathbb{C}^{n \times n}$
- Σ est diagonale $\mathbb{R}^{m \times n}$
  De plus, les valeurs diagonales de Σ sont positives ou nulles et classées :

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0, \text{ où } p = min(m, n)$$

SVD

En anglais, on parle de *Singular Value Decomposition*, ou SVD.

Bordeaux INP AQUITAINE

Existence et unicité de la SVD

Théorèmes

- Pour toute matrice $A \in \mathbb{C}^{m \times n}$ il existe une décomposition en valeurs singulières.
- De plus, les valeurs singulières $\{\sigma_j\}$ sont déterminées de manière unique.
- Si A est carrée et les σ_j sont distincts, les vecteurs singuliers droits $\{v_j\}$ et gauches $\{u_j\}$ sont déterminés de manière unique au signe complexe près (à un facteur $z \in \mathbb{C}$ tel que $|z| = 1$ près).

Changement de base

La SVD permet de manipuler des matrices diagonales en se plaçant dans une base appropriée. Considérons l'expression

$$b = A.x \qquad \text{et la SVD de } A : \qquad A = U\Sigma V^*$$

Écrivons maintenant :

$$b' = U^* b \qquad \text{et} \qquad x' = V^* x$$

La relation devient :

$$\begin{aligned} b &= A.x \\ U^*.b &= U^*.A.x \\ \underbrace{U^*.b}_{b'} &= \underbrace{U^*.U}_{Id}.\Sigma.\underbrace{V^*.x}_{x'} \\ b' &= \Sigma x' \end{aligned}$$

Plan

Vocabulary

Comparons les deux décomposition : en valeurs propres et en valeurs singulières.
Commençons par un point vocabulaire en anglais :

Français	Anglais
Valeur propre	*Eigenvalue*
Vecteur propre	*Eigenvector*
Décomposition en valeurs propres	*Eigendecomposition* ou *Eigenvalue decomposition* ou *EVD*
Valeur singulière	*Singular value*
Vecteur singulier à droite	*Right singular vector*
Décomposition en valeurs singulières	*Singular value decomposition* ou *SVD*

Définition EVD

Définition EVD

Avec $A \in \mathbb{C}^{m \times m}$ une matrice diagonalisable et $X \in \mathbb{C}^{m \times m}$:

$$A = X \Lambda X^{-1}$$

où Λ est une matrice diagonale $m \times m$ dont les valeurs sont les valeurs propres de A.

Changement de base

On peut aussi se placer dans une base appropriée pour manipuler des matrices diagonales.

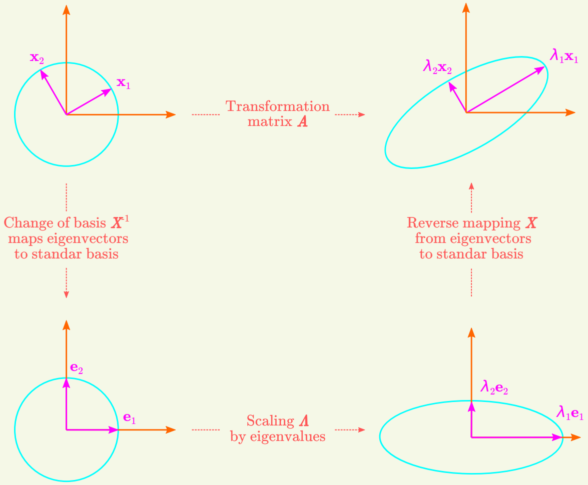
Avec A diagonalisable : $A = X \Lambda X^{-1}$, on considère $b = A.x$ comme précédemment.

$$b' = X^{-1}.b \qquad \text{et} \qquad x' = X^{-1}.x$$

pour obtenir : $b = A.x \iff b' = \Lambda x'$

Visualisation géométrique

Exemple 2D



Source : https://pabloinsente.github.io/intro-linear-algebra, Pablo Caceres (modifié)

SVD versus EVD

	SVD	EVD
Formule	$A = U\Sigma V^*$	$A = X\Lambda X^{-1}$
Nombre de bases	2	1 (endomorphisme)
Bases orthonormales	Oui	Non
Il existe toujours une décomposition	Oui	Non
Applications	Utilisation de A ou A^{-1} Analyse de données sur A	Calcul d'itération A^k e^{tA} Polynômes en A

Relation SVD - EVD

Lien valeur propres - valeurs singulières

Les valeurs singulières non nulles de A sont les racines carrées des valeurs propres de A^*A ou AA^*.

En effet :

$$\begin{aligned} A^*A &= (U\Sigma V^*)^*(U\Sigma V^*) \\ &= V\Sigma^* U^* U\Sigma V^* \\ &= V(\Sigma^*\Sigma)V^* \end{aligned}$$

- V est une base de vecteurs propres de A^*A
- U est une base de vecteurs propres de AA^*
- $\sigma_i^2 = \lambda_i$

Remarque

Attention, c'est une très mauvaise méthode pour calculer les valeurs singulières : en général $\kappa(A^*A) \approx \kappa(A)^2$.

Plan

Bordeaux INP AQUITAINE

SVD et rang

Théorème

Le nombre de valeurs singulières non nulles de la SVD de A vaut r : le rang de A.

$$A = U.\begin{pmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix}.V^*$$

Preuve

$$A = \underbrace{U}_{\text{de rang plein}} \Sigma \underbrace{V^*}_{\text{de rang plein}}$$

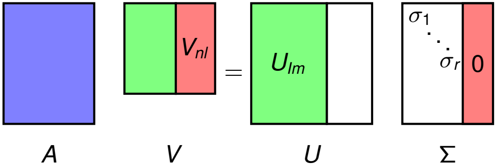
$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0$ donc $rg(A) = rg(\Sigma) = r$

Image et noyaux

Théorème

$$Im(A) = Vect(u_1, \dots, u_r)$$

$$Ker(A) = Vect(v_{r+1}, \dots, v_n)$$



Preuve

Conséquence de :

$$Im(\Sigma) = Vect(e_1, \dots, e_r) \subseteq \mathbb{C}^m$$

$$Ker(\Sigma) = Vect(e_{r+1}, \dots, e_n) \subseteq \mathbb{C}^n$$

Normes

Norme induite

En norme induite par la norme 2 : $|||A||| = \sigma_1$

Norme de Frobenius

La norme de Frobenius de A est définie :

$$||A||_F = \sqrt{\sum_{i=1}^{m}\left(\sum_{j=1}^{n}|a_{i,j}|^2\right)}$$

On peut aussi l'exprimer :

$$||A||_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \ldots + \sigma_r^2}$$

Déterminant, Approximation de rang faible

Déterminant

Pour $A \in \mathbb{C}^{m \times m}$ (comme pour les valeurs propres) :

$$|det(A)| = \prod_{i=1}^{m} \sigma_i$$

Approximation de rang faible

A est la somme de r matrices de rang 1 :

$$A = \sum_{j=1}^{r} \sigma_j u_j v_j^*$$

Application : la compression d'image (projet) !

Compression

Image originale

Image compressée par SVD (rang 50)

Calcul de la SVD

Comment obtenir la décomposition en valeurs singulières ?

Voir projet !

On va maintenant s'intéresser au calcul de la décomposition en valeurs propres, qui a des similarités avec le calcul de la SVD.

Plan

Plan

2 Décomposition en valeurs propres
- Présentation du problème
- Difficulté du problème

Présentation du problème

On considère $A \in \mathbb{C}^{n \times n}$, diagonalisable. On cherche l'ensemble des valeurs propres λ et des vecteurs propres x_λ tels que :

$$\begin{cases} Ax_\lambda = \lambda x_\lambda \\ x_\lambda \neq 0 \end{cases}$$

Tony Stark et la EVD

Dans *Avenger : endgame*, Tony Stark demande à son assistante, l'intelligence artificielle F.R.I.D.A.Y. :

"F.R.I.D.A.Y., compute this eigenvalue"

Il s'agit d'un calcul de physique quantique sur la topologie d'un ruban de Möbius, pour trouver comment voyager dans le temps !

En effet, en physique quantique, on peut chercher les états propres à partir de l'équation de Schrödinger indépendante du temps :

$$H\varphi = E\varphi$$

où E est l'énergie, valeur propre de l'opérateur hamiltonien H, et φ la fonction d'onde de la particule dans un état propre !

Plan

2 Décomposition en valeurs propres
- Présentation du problème
- Difficulté du problème

Difficulté du problème

Sondage : à votre avis, existe-t-il des méthodes directes pour le calcul des valeurs propres ?

A oui, comme pour la résolution de systèmes linéaires

B oui, mais elles sont trop coûteuse donc pas utilisées

C non, il n'en existe pas

Difficulté du problème

On peut résoudre le problème en trouvant les racines d'un polynôme de degré n : le polynôme caractéristique.

$$P_A(z) = det(zId - A)$$

$$P_A(z) = z^n + \alpha_{n-1}z^{n-1} + ... + \alpha_0$$

Réciproquement, si on a un tel polynôme $P(z)$, il existe une matrice compagnon dont P est le polynôme caractéristique :

$$A_P = \begin{pmatrix} 0 & 0 & \cdots & 0 & -\alpha_0 \\ 1 & 0 & \cdots & 0 & -\alpha_1 \\ 0 & 1 & \cdots & 0 & -\alpha_2 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & -\alpha_{n-1} \end{pmatrix}$$

Donc les deux problèmes sont **équivalents** !

$\boxed{\text{Calcul de EVD}} \iff \boxed{\text{Calcul des racines d'un polynôme}}$

Difficulté du problème

Existe-t-il donc une méthode directe pour le calcul des racines d'un polynôme ?

Abel (1802 - 1829), mathématicien norvégien, a montré qu'il n'existe pas de formule générale pour calculer les racines d'un polynôme de degré ≥ 5 avec les opérations usuelles $(+, -, *, /, \sqrt{})$ (**théorème d'Abel-Ruffini**)

Galois (1811 - 1832), mathématicien français, confirme et généralise ce résultat dans sa **théorie de Galois**.

Voir le chapitre 4 pour le calcul des racines d'un polynôme en pratique.

Difficulté du problème

Existe-t-il des méthodes directes pour le calcul des valeurs propres ?

A ~~oui, comme pour la résolution de systèmes linéaires~~

B ~~oui, mais elles sont trop coûteuse donc pas utilisées~~

C non, il n'en existe pas

Heureusement, il existe des méthodes itératives assez efficaces.

Plan

Plan

Bordeaux INP AQUITAINE

Méthode de Jacobi

Commençons par une première méthode itérative : la méthode de Jacobi.

Principe

Dans le cas symétrique réel, la méthode de Jacobi permet de se ramener à la forme diagonale à partir de rotations.

$$ {}^tO_k.A.O_k \underset{k\to\infty}{\longrightarrow} D $$

où :

- D est une matrice diagonale.
- les O_k sont des matrices de rotations, donc orthogonales.

Méthode de Jacobi

Soit p un indice de ligne et q un indice de colonne, et un réel $\theta \in]-\pi;\pi]$, alors :

$$O = \begin{pmatrix}
1 & & & & & & & & & \\
& \ddots & & & & & & & & \\
& & \cos\theta & & & & \sin\theta & & & \\
& & & 1 & & & & & & \\
& & & & \ddots & & & & & \\
& & & & & 1 & & & & \\
& & -\sin\theta & & & & \cos\theta & & & \\
& & & & & & & 1 & & \\
& & & & & & & & \ddots & \\
& & & & & & & & & 1
\end{pmatrix}$$

où :

- $O_{p,p} = O_{q,q} = \cos(\theta)$
- $O_{p,q} = \sin\theta$
- $O_{q,p} = -\sin\theta$

Méthode de Jacobi

Exemple de matrice Q_k

Considérons la transformation d'une matrice A, restreinte aux lignes p et q. On a :

$$B = {}^tO.A.O$$

$$B = \begin{pmatrix} b_{p,p} & b_{p,q} \\ b_{q,p} & b_{q,q} \end{pmatrix}$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} . \begin{pmatrix} a_{p,p} & a_{p,q} \\ a_{q,p} & a_{q,q} \end{pmatrix} . \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

Choisir $\theta \in]-\frac{\pi}{4}; \frac{\pi}{4}[$ tel que $b_{p,q}$ et $b_{q,p}$ soient nuls. Si $b_{p,q}$ et $b_{q,p}$

sont nuls alors $\theta = 0$, sinon,

$$b_{p,q} = b_{q,p} = a_{p,q} cos(2\theta) + \frac{a_{p,p} - a_{q,q}}{2} sin(2\theta)$$

$$\Rightarrow cotan(2\theta) = \frac{a_{q,q} - a_{p,p}}{2a_{p,q}}$$

Méthode de Jacobi

Itérations

Soient les matrices $A_0 = A$ et $O_0 = Id$, on construit la suite telle que :

$$\begin{cases} A_{k+1} = {}^tO.A_k.O \\ O_{k+1} = O_k.O \end{cases}$$

Étapes de l'algorithme :

- choisir un élément extradiagonal $a_{p,q}$ de A non nuls;
- construire la matrice de rotation O adaptée à cet élément;
- construire la matrice ${}^tO.A.O$.

Méthode de Jacobi

Optimisations

Seules les p -ièmes et q -ièmes lignes et colonnes de la matrice A sont modifiées à chaque itération :

$$\begin{cases} b_{i,j} = a_{i,j} & \text{si } i,j \neq p,q \\ b_{p,i} = a_{p,i}\cos\theta - a_{q,i}\sin\theta & \text{si } i \neq p,q \\ b_{q,i} = a_{p,i}\sin\theta - a_{q,i}\cos\theta & \text{si } i \neq p,q \\ b_{p,p} = a_{p,p} - a_{p,q}\tan\theta & \\ b_{q,q} = a_{p,p} + a_{p,q}\tan\theta & \\ b_{p,q} = b_{q,p} = 0 & \end{cases}$$

Bilan

- Choix du pivot : élément extra-diagonal de norme maximal $\Rightarrow$ assure la convergence
- Complexité : $\mathcal{O}(n)$ à chaque étape

Bordeaux INP AQUITAINE

Plan

3 Calcul des valeurs propres
- Méthode de Jacobi
- Transformations vers des matrices plus simples
- Méthode dichotomique
- Méthode QR

Transformations vers des matrices plus simples

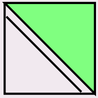
Avec un algorithme direct, on peut néanmoins simplifier la forme de la matrice :

- EVD d'une matrice symétrique : se ramener à une matrice tridiagonale
- EVD d'une matrice non symétrique : se ramener à une matrice de Hessenberg
- SVD : se ramener à une matrice bidiagonale (cf. projet)

Matrice de Hessenberg

C'est une matrice «presque triangulaire».

Une matrice de Hessenberg supérieure est une matrice triangulaire supérieure avec une sous-diagonale.



Transformations vers des matrices plus simples

On va procéder en transformant A à l'aide d'un algorithme direct puis un algorithme itératif :

Décomposition Matrice A	EVD Non symétrique	EVD Symétrique	SVD Quelconque
Algo direct	$\downarrow$	$\downarrow$	$\downarrow$
Forme inter.	Hessenberg	Tridiagonale	Bidiagonale
Algo itératif	$\downarrow$	$\downarrow$	$\downarrow$
Forme finale	Triangulaire	Diagonale	Diagonale

Et à partir de ces formes finales, on peut trouver les valeurs propres et singulières sur la diagonale.

Bordeaux INP AQUITAINE

Bordeaux INP AQUITAINE

Réduction à la forme de Hessenberg ou tridiagonale

Pour concerver les valeurs propres, on applique des transformations unitaires : des changements de base avec des matrices Q unitaires (rotations, réflexions).

$$A \to Q^* AQ$$

Une mauvaise idée (a priori)

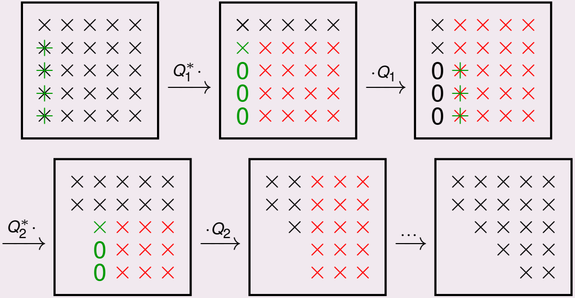
Utiliser une matrice de Householder pour triangulariser :

$$\begin{bmatrix} \ast & \times & \times & \times & \times \\ \ast & \times & \times & \times & \times \\ \ast & \times & \times & \times & \times \\ \ast & \times & \times & \times & \times \\ \ast & \times & \times & \times & \times \end{bmatrix} \xrightarrow{Q_1^* \cdot} \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \xrightarrow{\cdot Q_1} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}$$

On modifie toutes les lignes en multipliant par Q_1^* à gauche, donc toutes les colonnes en multipliant par Q_1 à droite.

Réduction à la forme de Hessenberg ou tridiagonale

Une bonne idée



On utilise Householder pour se ramener à la forme Hessenberg.

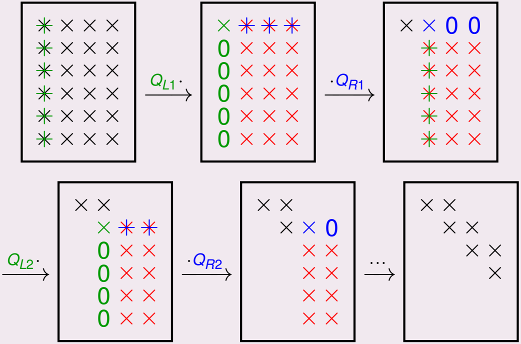
Dans le cas symétrique, la multiplication par Q_k éliminera aussi les coefficients de la ligne k $\rightarrow$ forme finale tridiagonale.

Réduction à la forme bidiagonale pour la SVD

Pour le calcul de la SVD, le problème est différent car on construit 2 bases distinctes. On n'est pas obligé d'appliquer 2 similitudes inverses à droite et à gauche, mais simplement 2 transformations unitaires :

$$A \to Q_L \cdot A \cdot Q_R$$

Bidiagonalisation



Plan

Bordeaux INP AQUITAINE

Méthode dichotomique

Dans le cas symétrique, considérons la matrice tridiagonale obtenue :

$$\begin{pmatrix} b_1 & c_1 & & & \\ c_1 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & b_{n-1} & c_{n-1} \\ & & & c_{n-1} & b_n \end{pmatrix}$$

Polynôme caratéristique

Si on considère A_k la matrice restreinte aux k premières lignes et colonnes, on a les polynôme caratéristiques suivants :

$$\begin{cases} P_0(X) = 1 \\ P_1(X) = b_1 - X \\ P_k(X) = (b_k - X)P_{k-1}(X) - c_{k-1}^2 P_{k-2}(X) \end{cases}$$

Méthode dichotomique

Supposons avoir des valeurs propres toutes distinctes.

Propriétés des polynômes caractéristiques P_k

- P_k de degré k
- $\lim_{x \to -\infty} P_k(x) = +\infty$ et $\lim_{x \to +\infty} P_k(x) = (-1)^k \infty$
- si x est racine du polynôme P_k, alors $P_{k-1}(x)$ et $P_{k+1}(x)$ sont non nuls et de signes opposés

Calcul des valeurs propres

Recherche des racines des $P_k(X)$

Recherche des racines

Tableau des variations de ces polynômes :

	$-\infty$		e_1	d_1	e_2	b_1	d_2	e_3		$+\infty$
Variations de P_0 :	1					$+$				1
Variations de P_1 :	$+\infty$			$+$		0	$-$			$-\infty$
Variations de P_2 :	$+\infty$		$+$	0	$-$		0	$+$		$+\infty$
Variations de P_3 :	$+\infty$	$+$	0	$-$	0		$+$	0	$-$	$-\infty$

Voir chapitre 4 pour la méthode de localisation des racines de cette suite.

Plan

Algorithme QR

Un algorithme itératif simple à implémenter en utilisant la factorisation QR sur la matrice Hessenberg ou tridiagonale obtenue :

Algorithme

- $A^{(0)} = A$
- Pour $k = 1, 2, \ldots$
- $\quad Q^{(k)}, R^{(k)} \leftarrow factoQR(A^{(k-1)})$
- $\quad A^{(k)} \leftarrow R^{(k)}.Q^{(k)}$

Remarque

Condition d'arrêt pas évidente à choisir.

Algorithme QR

Théorème : convergence de la méthode QR

Si A est diagonalisable et ses valeurs propres $\{\lambda_1, \ldots, \lambda_n\}$ sont distinctes, telles que : $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$, alors :

$$\begin{cases} \lim_{k\to\infty} a_{i,i}^{(k)} = \lambda_i \\ \lim_{k\to\infty} a_{i,j}^{(k)} = 0 \qquad \forall j < i \end{cases}$$

On converge vers une matrice :

$$A^{(k)} = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

Algorithme QR

Remarque

L'algorithme préserve la forme de la matrice (tridiagonale ou Hessenberg) à chaque étape.

Remarque

Plus les valeurs propres seront de module proche, plus la convergence sera lente. Il existe des algorithmes optimisés qui utilisent des décalages :

$$Q^{(k)}, R^{(k)} \leftarrow factoQR(A^{(k-1)} - \sigma_{(k-1)} Id)$$

où $\sigma_{(k-1)}$ est l'estimation d'une valeur propre.

Convergence

Les itérations sont chères (rappel : factorisation QR en $\mathcal{O}(n^3)$ en général, même si on peut optimiser dans le cas tridiagonal), mais la convergence est généralement rapide.

Plan

Quotient de Rayleigh

Définition

Le quotient de Rayleigh d'un vecteur $x \in \mathbb{R}^n$ non nul est :

$$\boxed{r(x) = \frac{{}^t xAx}{{}^t xx}}$$

- Si x est un vecteur propre : $r(x) = \lambda$ la valeur propre associée.
- Sinon, $r(x)$ est la meilleure valeur propre à considérer si x est proche, mais pas forcément égal à un vecteur propre.

Remarque

Avec un vecteur normalisé en norme 2, on a $r(x) = {}^t xAx$.

Plan

Méthode de la puissance itérée

Idée

Un vecteur propre en particulier est facilement identifiable : celui associé à la plus grande valeur propre (en module).

Soient $(e_1, \ldots, e_n)$ une base de vecteurs propres associés aux valeurs propres $\{\lambda_1, \ldots, \lambda_n\}$, classées : $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$

Soit $x_0 \neq 0$, on étudie la suite des $x_k = A^k x_0$. Dans la base (e_i) :

$$\begin{cases} x_0 = \alpha_1 e_1 + \alpha_2 e_2 + \ldots + \alpha_n e_n \\ x_k = \lambda_1^k \alpha_1 e_1 + \lambda_2^k \alpha_2 e_2 + \ldots + \lambda_n^k \alpha_n e_n \end{cases}$$

Si $|\lambda_1| > |\lambda_2|$, asymptotiquement $||x_k|| \xrightarrow[k \to \infty]{} |\lambda_1|^k$

Méthode de la puissance itérée

Algorithme

- Choisir $x_0 \neq 0$
- Pour $k = 1, \dots$
- $x_{k+1} \leftarrow \frac{Ax_k}{||Ax_k||}$: multiplication par A et normalisation
- $\lambda_{k+1} \leftarrow {}^t x_{k+1} A x_{k+1}$: quotient de Rayleigh

Convergence

Si on a bien $|\lambda_1| > |\lambda_2|$, on converge vers le vecteur propre associé à λ_1.

Remarque

Si on effectue l'algorithme sur A^{-1}, on converge vers le vecteur propre associé à la plus petite valeur propre de A (en module). C'est l'algorithme de la puissance inverse. On remplace la multiplication $x_{k+1} = A^{-1} x_k$ par la résolution du système $Ax_{k+1} = x_k$.

Plan

4 Obtention des vecteurs propres
- Méthode de la puissance itérée
- Méthode de la puissance inverse avec décalage
- Résumé
- Un vecteur propre à quelques milliards de dollars

Bordeaux INP AQUITAINE

Méthode de la puissance inverse avec décalage

Cette méthode permet de chercher un vecteur propre en ciblant la valeur propre λ qui lui est associée.

Soit $\tilde{\lambda}$ une valeur proche mais différente de λ (plus proche que les autres valeurs propres).
Soit x un vecteur, et on cherche y solution du système :

$$(A - \tilde{\lambda}.Id)y = x$$

Dans la base de vecteurs propres :

$$\left\{\begin{array}{ccccccccc} x & = & \alpha_1 e_1 & + & \alpha_2 e_2 & + & \dots & + & \alpha_n e_n \\ y & = & \gamma_1 e_1 & + & \gamma_2 e_2 & + & \dots & + & \gamma_n e_n \\ (A - \tilde{\lambda}.Id)y & = & (\lambda_1 - \tilde{\lambda})\gamma_1 e_1 & + & (\lambda_2 - \tilde{\lambda})\gamma_2 e_2 & + & \dots & + & (\lambda_n - \tilde{\lambda})\gamma_n e_n \end{array}\right.$$

d'où : $\forall k$

$$\boxed{\gamma_k = \frac{\alpha_k}{\lambda_k - \tilde{\lambda}}}$$

Construction : $x_0 \neq 0$ et une suite (x_k) :

$$\boxed{(A - \tilde{\lambda}.Id)x_{k+1} = x_k}$$

Méthode de la puissance inverse avec décalage

Compostante par composante, (x_k) est une suite géométrique de raison $\frac{1}{\lambda_k - \tilde{\lambda}}$.

Algorithme

- Choisir $x_0 \neq 0$
- Pour $k = 1, \dots$
-     $x_{k+1} \leftarrow$ Résolution de $(A - \tilde{\lambda}.Id)x_{k+1} = x_k$
-     $x_{k+1} \leftarrow \frac{x_{k+1}}{||x_{k+1}||}$
-     $\tilde{\lambda} \leftarrow {}^t x_{k+1} A x_{k+1}$: quotient de Rayleigh (optionnel)

Remarques

- C'est la méthode de la puissance itérée sur la matrice $(A - \tilde{\lambda}.Id)^{-1}$
- On fait une fois la factorisation LU sur $A - \tilde{\lambda}.Id$, et on fait les résolutions triangulaires $L.U.x_{k+1} = x_k$ à chaque étape.
- On peut modifier $\tilde{\lambda}$ à la dernière étape, mais il faut alors refaire la factorisation LU de la nouvelle matrice pour la résolution suivante.

Bordeaux INP AQUITAINE

Plan

Résumé de convergence

	Méthode	Coût	Convergence $\|\|e_{k+1}\|\|/\|\|e_k\|\|$
Puissance itérée	$x_{k+1} = Ax_k$	kn^2	$\|\lambda_2\|/\|\lambda_1\|$
Puissance inverse	$Ax_{k+1} = x_k$	$n^3 + kn^2$	$\|\lambda_n\|/\|\lambda_{n-1}\|$
Puissance inverse avec décalage	$(A - \tilde{\lambda} I)x_{k+1} = x_k$	$n^3 + kn^2$	$\frac{\|\lambda_c - \tilde{\lambda}\|}{\|\lambda_{c2} - \tilde{\lambda}\|}$

Avec :

- λ_1 : valeur propre la plus grande (en module)
- λ_2 : deuxième valeur propre la plus grande (en module)
- λ_n : valeur propre la plus petite (en module)
- λ_{n-1} : deuxième valeur propre la plus petite (en module)
- λ_c : valeur propre la plus proche de $\tilde{\lambda}$
- λ_{c2} : deuxième valeur propre la plus proche de $\tilde{\lambda}$

Plan

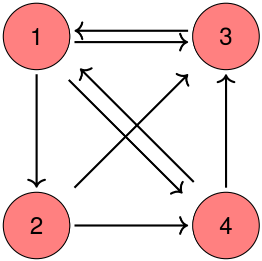
4. Obtention des vecteurs propres
   - Méthode de la puissance itérée
   - Méthode de la puissance inverse avec décalage
   - Résumé
   - Un vecteur propre à quelques milliards de dollars

Le problème

Le problème d'un moteur de recherche :

- localiser les pages web;
- les indexer;
- les classer selon leur ordre d'importance.

Un modèle de classement démocratique



Première idée

Classer une page selon le nombre de pages qui pointent dessus : nombre de pointeurs entrants.

Problème : un lien qui vient d'un «petit site» a le même poids qu'un lien qui vient d'un «gros site».

Pondération des pointeurs

Deuxième idée

Affecter un poids à chaque lien et noter les pages en sommant les poids des pointeurs entrants.

On écrit le système linéaire :

$$\begin{cases} x_1 = x_3 + x_4 \\ x_2 = x_1 \\ x_3 = x_1 + x_2 + x_4 \\ x_4 = x_1 + x_2 \end{cases}$$

Modélisation du problème

Quelle propriété doit avoir la solution de ce problème ?

$$x = A.x$$

Autrement dit, on cherche le vecteur propre associé à la valeur propre 1, où A est la matrice :

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

Existence d'une solution

Ce problème admet-il une solution ?

$$\Pi_A(y) = det(A - yI) = \begin{vmatrix} -y & 0 & 1 & 1 \\ 1 & -y & 0 & 0 \\ 1 & 1 & -y & 1 \\ 1 & 1 & 0 & -y \end{vmatrix}$$

$$\Pi_A(y) = y^4 - 2y^2 - 3y - 1$$

Donc $\Pi_A(1) \neq 0$ et on n'a pas de solution.

Remarque

Un autre inconvénient de cette méthode (en plus de ne pas admettre de solution. . .) est qu'elle donne un «poids électoral» plus important à un site qui a beaucoup de pointeurs sortants.

Normalisation des poids

Troisième idée

On normalise les poids : si n_i est le nombre de pointeurs sortants du site i, on divise le poids de ses pointeurs par n_i.

$$\begin{cases} x_1 = x_3 + x_4/2 \\ x_2 = x_1/3 \\ x_3 = x_1/3 + x_2/2 + x_4/2 \\ x_4 = x_1/3 + x_2/2 \end{cases}$$

Notre nouveau A est :

$$A = \begin{pmatrix} 0 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix}$$

Existence de la solution

On peut vérifier que $\Pi_A(1) = 0$ cette fois-ci.

Plus généralement, en prenant le vecteur $\mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$, on peut voir que :

$$^{t}A\mathbf{1} = \mathbf{1}$$

Donc 1 est valeur propre de ^{t}A. Comme A et ^{t}A ont les mêmes valeur propres, 1 est aussi valeur propre de A.

Valeur propre de plus grand module

On peut même montrer que 1 est la valeur propre de plus grand module.

On cherche alors le vecteur propre associé à la plus grande valeur propre $\rightarrow$ algorithme de la puissance itérée.

Ce serait cette modélisation qui aurait permis au moteur de recherche *Google* de se démarquer des autres au début des années 2000.

Référence

K. Bryan and T. Leise,The 25,000,000,000 dollars* eigenvector : the linear algebra behind Google, *SIAMReview, 48 (3), 569-581, Sept. 2006*

* valeur estimée de *Google* en 2004.