Projet CLANU en 3GE: Compléments d'algèbre linéaire numérique

Année 2008/2009

1 Décomposition QR

On rappelle que la multiplication avec une matrice unitaire $Q \in \mathbb{C}^{n \times n}$ (c'est-à-dire $Q^{-1} = Q^* = \bar{Q}^T$) ne change pas la norme euclidienne d'un vecteur $x \in \mathbb{C}^n$, $\|Qx\| = \|x\|$. Soit $A \in \mathbb{C}^{n \times p}$. Il est souvent utile d'avoir une décomposition $A = QR$ où $R \in \mathbb{C}^{n \times p}$ est une matrice triangulaire supérieure et $Q \in \mathbb{C}^{n \times n}$ est unitaire.

Soient $A \in \mathbb{R}^{n \times p}$, $b \in \mathbb{R}^n$ donnés. Le problème des moindres carrés pour ces données est de trouver $x \in \mathbb{R}^p$ tel que

$$\|b - Ax\| = \min_{y \in \mathbb{R}^p} \|b - Ay\|.$$

Avec la décomposition $A = QR$ (toutes les matrices réelles) ce problème est équivalent au problème de trouver

$$\min_{y \in \mathbb{R}^p} \|Q^T b - Ry\|.$$

Puisque R est triangulaire supérieure on peut résoudre $Rx = Q^T b$ avec l'algorithme de remontée.

1.1 L'algorithme de Householder

L'algorithme de Householder est une méthode stable pour calculer la décomposition QR.

Définition :
Soit $v \in \mathbb{C}^n, v \neq 0$ un vecteur. La matrice de Householder associée à v, notée $H(v)$ est définie par $H(v) = \mathbb{1} - 2vv^*/\|v\|^2$. On pose $H(0) = \mathbb{1}$.

On observe que les matrices de Householder sont hermitiennes :

$$H(v)^* = \mathbb{1} - 2(v^*)^* v^*/\|v\|^2 = H(v).$$

De plus $H(v)^2 = \mathbb{1}$, c'est-à-dire les matrices de Householder sont unitaires. Soit $w = v + \|v\|e_1$. Un petit calcul montre alors que $H(w)v = -\|v\|e_1$. De façon similaire, on obtient

$H(v-\|v\|e_1)v = +\|v\|e_1$. Donc, en multipliant v avec $H(v+\|v\|e_1)$ ou $H(v-\|v\|e_1)$, on obtient un vecteur aligné à e_1. Le but est de transformer la matrice A successivement en une matrice triangulaire supérieure en la multipliant avec des matrices de Householder. Soit $A = [a_1, \ldots, a_p]$ où $a_k \in \mathbb{R}^n$ est la k-ième colonne de A. On pose $A_1 = A$ et $H_1 = H(a_1 \pm \|a_1\|e_1)$. Ensuite on pose $A_2 = H_1 A_1$. Avec les propriétés de la matrice de Householder, la première colonne de A_2 devient

$$a_1^{(2)} = \begin{pmatrix} \mp\|a_1\| \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Le choix du signe dans H_1 se fera en fonction de la stabilité numérique. On considère maintenant la sous-matrice suivante $\tilde{A}_2 \in \mathbb{C}^{(n-1)\times(p-1)}$ de A_2 :

$$\tilde{A}_2 = \begin{pmatrix} a_{2,2}^{(2)} & \ldots & a_{2,p}^{(2)} \\ \vdots & & \vdots \\ a_{n,2}^{(2)} & \ldots & a_{n,p}^{(2)} \end{pmatrix}$$

Soit $\tilde{a}_1^{(2)} \in \mathbb{C}^{n-1}$ la première colonne de $\tilde{A}_2$. On pose $\tilde{H}_2 = H(\tilde{a}_1^{(2)} \pm \|\tilde{a}_1^{(2)}\|e_1)$ où $e_1 \in \mathbb{C}^{n-1}$ et

$$H_2 = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & & \tilde{H}_2 & \end{pmatrix}$$

Donc

$$A_3 = H_2 A_2 = \begin{pmatrix} \mp\|a_1\| & a_{1,2}^{(3)} & a_{1,3}^{(3)} & \ldots & a_{1,p}^{(3)} \\ 0 & \mp\|\tilde{a}_1^{(2)}\| & a_{2,3}^{(3)} & \ldots & a_{2,p}^{(3)} \\ \vdots & 0 & a_{3,3}^{(3)} & \ldots & a_{3,p}^{(3)} \\ 0 & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a_{n,3}^{(3)} & \ldots & a_{n,p}^{(3)} \end{pmatrix}$$

Supposons qu'au bout de $k-1$ étapes correspondantes à $k-1$ multiplications avec certaines matrices H_i, on est arrivé à transformer la matrice originale A en la matrice A_k,

$$A_k = \begin{pmatrix} a_{1,1}^{(k)} & \ldots & \ldots & \ldots & \ldots & a_{1,p}^{(k)} \\ 0 & \ddots & & & & \\ 0 & & a_{k-1,k-1}^{(k)} & a_{k-1,k}^{(k)} & \ldots & a_{k-1,p}^{(k)} \\ \vdots & & 0 & a_{k,k}^{(k)} & \ldots & a_{k,p}^{(k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & a_{n,k}^{(k)} & \ldots & a_{n,p}^{(k)} \end{pmatrix}$$

Pour la k-ième étape on pose

$$\tilde{A}_k = \begin{pmatrix} a_{k,k}^{(k)} & \dots & a_{k,p}^{(k)} \\ \vdots & & \vdots \\ a_{n,k}^{(k)} & \dots & a_{n,p}^{(k)} \end{pmatrix}$$

et

$$H_k = \begin{pmatrix} \mathbb{1}_{k-1} & 0 \\ 0 & \tilde{H}_k(\tilde{a}_1^{(k)} \pm \|\tilde{a}_1^{(k)}\| e_1) \end{pmatrix}$$

où $\tilde{a}_1^{(k)}$ est la première colonne de $\tilde{A}_k$ et $\mathbb{1}_{k-1}$ est l'identité dans $\mathbb{C}^{(k-1)\times(k-1)}$. Ensuite $A_{k+1} = H_k A_k$. Remarquons enfin que pour des raisons de stabilité, on choisit le signe dans $\tilde{H}_k$ comme celui de $a_{k,k}^{(k)}$

Au bout de $p-1$ multiplications la matrice $R = H_{p-1} \dots H_1 A$ est triangulaire supérieure. Donc $Q^* = H_{p-1} \dots H_1$ et $Q = H_1 \dots H_{p-1}$ (puisque les matrices de Householder sont hermitiennes).

2 Calcul de valeurs propres

On rappelle qu'un scalaire $\lambda \in \mathbb{C}$ est une valeur propre de la matrice $A \in \mathbb{C}^{n\times n}$ s'il existe un vecteur $x \in \mathbb{C}^n, x \neq 0$ tel que $Ax = \lambda x$. Dans ce cas x est appelé vecteur propre. Par ailleurs, λ est une valeur propre si et seulement si c'est un zéro du polynôme caractéristique $p(\lambda)$ de degré n, $p(\lambda) = \det(A - \lambda \mathbb{1})$. Donc il y a n valeurs propres (qui ne sont pas forcément différentes).

2.1 Méthode de la puissance

La méthode la plus simple pour calculer la valeur propre la plus grande d'une matrice est la méthode de la puissance. On suppose que les valeurs propres $\lambda_1, \dots, \lambda_n$de $A \in \mathbb{R}^{n\times n}$ sont ordonnées :

$$0 \le |\lambda_1| \le \dots \le |\lambda_{n-1}| < \lambda_n.$$

Si les vecteur propres correspondants $v_1, \dots, v_n$ sont linéairement indépendants, on peut calculer λ_n et v_n avec l'algorithme suivant :

`choisir` $x^{(0)} \in \mathbb{R}^n$ `avec` $\|x^{(0)}\| = 1$ `et une tolérance` ϵ.
`Faire` $y^{(k)} = Ax^{(k-1)}$`,` $x^{(k)} = y^{(k)}/\|y^{(k)}\|$
`jusqu'à ce que` $\|x^{(k)} - x^{(k-1)}\| \le \epsilon$.

TAB. 1: Algorithme pour la méthode de la puissance.

Sous les hypothèses indiquées, $x^{(k)}$ converge vers un vecteur propre pour λ_n et $\|y^{(k)}\|$ converge vers λ_n. Pour plus de détails voir la littérature, par exemple le livre de G. Allaire, « Algèbre linéaire numérique ».

2.2 La méthode QR

La méthode QR est peut-être la plus utilisée pour calculer toutes les valeurs propres d'une matrice $A \in \mathbb{R}^{n\times n}$. Nous allons introduire ici seulement une version basique de cette méthode et nous faisons les mêmes hypothèses que dans le paragraphe précédent : D'abord on pose $A_1 = A$. On calcule la décomposition $A_1 = Q_1R_1$ puis on pose $A_2 = R_1Q_1$. Ensuite on décompose $A_k = Q_kR_k$ et on pose $A_{k+1} = R_kQ_k$, $k \geq 1$. Si toutes les valeurs propres de A sont réelles et différentes, A_k converge vers une matrice triangulaire supérieure avec les valeurs propres de A sur la diagonale, donc on arrête l'itération quand $\max_{i>j} \left|a_{i,j}^{(k)}\right| < \epsilon$ où ϵ est une tolérance. S'il y a des paires de valeurs propres complexes avec des valeurs absolues différentes, la suite A_k « converge » vers une matrice presque triangulaire supérieure, c'est-à-dire une paire $\lambda,\bar{\lambda}$ de valeurs propres complexes donne lieu à un bloc B,

$$B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix},$$

où b_{11} et b_{22} sont sur la diagonale dans la limite $k \to \infty$ et les valeurs propres de B convergent vers λ, $\bar{\lambda}$. Dans ce cas il faut adapter le critère d'arrêt puisque b_{21} est sous la diagonale et non zéro en général. Entre ces blocs on obtient toujours les valeurs propres réelles sur la diagonale.

3 Exercices

3.1 Programmation

On vous demande d'écrire en langage C :

- une fonction qui calcule la décomposition QR d'une matrice $A \in \mathbb{R}^{n\times p}$ avec l'algorithme de Householder où la matrice A peut être entrée à la main ou lue à partir d'un fichier. On se restreint au cas où A est réelle.

- une fonction qui calcule la valeur propre la plus grande et un vecteur propre correspondant d'une matrice $A \in \mathbb{R}^{n\times n}$ avec la méthode de la puissance pour une tolérance ϵ spécifiée (si possible).

- une fonction qui calcule toutes les valeurs propres d'une matrice $A \in \mathbb{R}^{n\times n}$ avec la méthode QR.

Attention aux boucles infinies ! Attention au portage informatique : tout le monde n'a pas le même système.

3.2 Problème inverse linéaire

Un problème inverse est une situation dans laquelle on tente de déterminer certains paramètres d'un modèle à partir des observations. Considérons, par exemple, les notes du module MA1. On suppose que les différentes notes sont stockées dans une matrice $A = (a_{ij})_{i=1\ldots,n,j=1,2,3}$ où n est le nombre d'élèves et a_{i1} est la note du DM de l'élève i,

a_{i2} est la note du DS d'analyse numérique et a_{i3} est la note du DS de mathématiques générales. Les notes finales sont stockées dans le vecteur $b \in \mathbb{R}^n$. Nous savons que la note finale est une combinaison linéaire des différentes notes, $b_i = c_1 a_{i1} + c_2 a_{i2} + c_3 a_{i3}$. Bien sûr, selon le reglement d'études, $c_1 = c_2 = c_3 = 1/2$, mais essayons de déterminer ces coefficients à partir des observations. Pour cela, on cherche $x \in \mathbb{R}^3$ tel que

$$\|b - Ax\| = \min_{y \in \mathbb{R}^3} \|b - Ay\|.$$

Question :
Résoudre le problème des moindres carrés ci-dessus avec la décomposition QR. Les données sont disponibles sur le site `math.univ-lyon1.fr/~reichert/ teaching.html`).

3.3 Modèle d'une population

On considère un modèle à temps discret d'une population structurée au sens qu'elle est divisée en n tranches d'âge. On note $x_i(t)$, $t = 0, 1, 2, \ldots$, le nombre d'individus dans la i-ième tranche au temps t. Soit s_i le taux de survie de la i-ième tranche entre t et $t+1$ et soit m_i son taux de reproduction. Alors

$$\begin{aligned} x_{i+1}(t+1) &= s_i x_i(t),\ i = 1, \ldots, n-1, \\ x_1(t+1) &= \sum_{i=1}^{n} m_i x_i(t). \end{aligned}$$

Sous forme matricielle, cela donne

$$x(t+1) = Ax(t) \text{ où}$$
$$A = \begin{pmatrix} m_1 & m_2 & \ldots & \ldots & m_n \\ s_1 & 0 & \ldots & \ldots & 0 \\ 0 & s_2 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & s_{n-1} & 0 \end{pmatrix}.$$

On peut montrer qu'il y a une valeur propre λ_n avec $0 \leq |\lambda_1| \leq \cdots \leq |\lambda_{n-1}| < \lambda_n$ et un vecteur propre v_n correspondant avec des entrées positives. Supposons qu'il y a n vecteurs propres $v_1, \ldots, v_n$ linéairement indépendants qui forment alors une base de $\mathbb{R}^n$. Par conséquent, $x(0) = \beta_1 v_1 + \cdots + \beta_n v_n$ pour certains coefficients $\beta_1, \ldots, \beta_n \in \mathbb{R}$ et

$$x(k) = A^k x(0) = \beta_1 \lambda_1^k v_1 + \cdots + \beta_n \lambda_n^k v_n.$$

Puisque λ_n est la valeur propre la plus grande, $x(k)$ s'aligne à v_n quand $k \to \infty$. Si $\lambda_n > 1$ la population croît indéfiniment, par contre, même dans ce cas il s'installe un équilibre dans le sens que la proportion de la i-ième tranche d'âge de la population totale tend vers une constante donnée par le i-ième élément de v_n qui donne alors la distribution normalisée.

Question :
On considère les données suivantes pour une population de poissons : $x(0) = (6, 12, 8, 4)$, $m_1 = 0$, $m_2 = 0.5$, $m_3 = 0.8$, $m_4 = 0.3$, $s_1 = 0.2$, $s_2 = 0.4$, $s_3 = 0.8$. En utilisant la méthode de la puissance, trouver la valeur propre la plus grande de la matrice A correspondante et la distribution normalisée.

Détails techniques Le projet peut se faire en binôme. Chaque groupe doit rendre deux dossiers, un pour l'enseignant en Analyse Numérique et le deuxième pour l'enseignant en Informatique. Les dossiers seront composés d'un exposé sur papier du travail effectué par le binôme et d'un CD avec les logiciels qui ont été implémentés et utilisés pour résoudre les exercices. Il sera tenu le plus grand compte de la qualité et de la clarté de la présentation. Les dossiers sont à rendre **au plus tard le vendredi 12 juin**, mais de préférence au début d'un cours correspondant.