Notes de cours
Algèbre Linéaire et Analyse Matricielle
3BIM INSA Lyon
2014-2015
v05122014

Ces notes de cours sont inspirées des livres de Algèbre des matrices de J Fresnel [1] et Numerical Linear Algebra de LN Trefethen et D Bau III [2].

5 Factorisation QR

Un **projecteur** est une matrice carrée P qui satisfait $P^2 = P$. Cette matrice est dite **idempotente**. Un projecteur "projette" un point x sur le sous-espace $\operatorname{im} P$. Si $v = Px$ alors $Pv = v$, c.-à-d. que P est constant sur les éléments de son image. Les projecteurs ont des propriétés utiles. Si P est un projecteur, alors

- $I - P$ est un projecteur. *Calculer* $(I - P)^2$.
- $\operatorname{im} P = \ker(I - P)$ et $\operatorname{im}(I - P) = \ker P$. *Considérez* $Pv - v$ *pour* v $\operatorname{im} P$.
- $\operatorname{im} P \cap \ker P = \{0\}$

Un projecteur est **orthogonal** si $\operatorname{im} P$ est orthogonal à $\ker P$. On a que P est un projecteur orthogonal si et seulement si $P = P^*$ (voir Quiz 0, question 3).

Sauf pour la matrice identité (qui est un projecteur), un projecteur P n'a pas plein rang. On peut se servir de projecteur pour décomposer un vecteur selon une famille orthonormée. Soit $q_1, q_2, ..., q_n$ une famille orthonormée avec $q_i \in \mathbb{C}^m$, $m \geq n$. On a pour tout $v \in \mathbb{C}^m$

$$v = r + \sum_{i=1}^{n} (q_i^* v) q_i$$

le terme $(q_i^* v)$, on peut donc commuter avec q_i,

$$\begin{aligned}
&= r + \sum_{i=1}^{n} q_i (q_i^* v) \\
&= r + \sum_{i=1}^{n} q_i q_i^* v \\
&= r + \sum_{i=1}^{n} (q_i q_i^*) v \\
&= r + \Big[\sum_{i=1}^{n} (q_i q_i^*) \Big] v
\end{aligned}$$

Pour $n < m$, la famille orthonormée ne forme pas une base; il y donc un vecteur résiduel $r \neq 0$. Les coefficients $(q_i^* v)$ représentent les composantes de v le long des axes formés par les vecteurs q_i. Cette décomposition peut-être réécrite sous forme matricielle. Les termes $(q_i q_i^*)$ sont des matrices $m \times m$ avec coefficients $[q_{i(j)} q_{i(k)}^*]_{jk}$. Si $\hat{Q} = [q_1|q_2|...|q_n]$ est une matrice $m \times n$, alors on peut vérifier que

$$\hat{Q}\hat{Q}^* = \Big[\sum_{i=1}^{n} (q_i q_i^*)\Big].$$

La matrice $P = QQ^*$ est une matrice hermitienne: $P^* = (\hat{Q}\hat{Q}^*)^* = (\hat{Q}^*)^*\hat{Q}^* = \hat{Q}\hat{Q}^* = P$. De plus P est un projecteur: $P^2 = \hat{Q}\hat{Q}^*\hat{Q}\hat{Q}^* = \hat{Q}(\hat{Q}^*\hat{Q})\hat{Q}^* = \hat{Q}I\hat{Q}^* = \hat{Q}\hat{Q}^*$. C'est donc un projecteur orthogonal. La matrice P projette un vecteur v orthogonalement sur le sous-espace vectoriel engendré par les colonnes de Q, c.-à-d. que $v = r + Pv$ avec r orthogonal à Pv. (Est-ce que cela rappelle le problème des moindres carrés ?). L'égalité $P = \hat{Q}\hat{Q}^*$ nous donne presque une décomposition en valeurs singulières de P. La matrice P est carrée $m \times m$. Soit Q une matrice orthonormale dont les n premières colonnes forment $\hat{Q}$. Alors $P = Q\Sigma Q^*$ avec Σ une matrice diagonale $m \times m$ avec n 1s sur les n premières diagonales et 0 ailleurs, est une décomposition en valeurs singulière de P. Les n valeurs singulière de P sont 1 (P est donc de rang n).

En procédant à l'envers, on peut montrer que tout projecteur orthogonal admet une décomposition $P = \hat{Q}\hat{Q}^*$, où $\hat{Q}$ est une base orthonormée de im P.

5.1 Orthogonalisation de Gram-Schmidt

Soit une matrice A de taille $m \times n$ avec $m \geq n$. On considère les vecteurs colonnes de la matrice $a_1, ..., a_n$ avec $a_i \in \mathbb{C}^m$. On veut construire une suite d'espaces engendrés par les vecteurs a_i:

$$\text{Vect}(a_1) \subseteq \text{Vect}(a_1, a_2) \subseteq ... \subseteq \text{Vect}(a_1, ..., a_i)...$$

L'idée de la factorisation QR est de trouver construire une suite de vecteurs orthonormés $q_1, q_2, ..., q_n$ qui engendres les sous-espaces successifs des colonnes de A:

$$\begin{aligned}
\text{Vect}(q_1) &= \text{Vect}(a_1), \\
\text{Vect}(q_1, q_2) &= \text{Vect}(a_1, a_2), \\
&... \\
\text{Vect}(q_1, ..., q_i) &= \text{Vect}(a_1, ..., a_i).
\end{aligned}$$

De cette façon, a_i doit être une combinaison linéaire des i premiers vecteurs $q_1, ..., q_i$,

$i = 1, ..., n$. Sous forme matricielle, on a

$$\left[\begin{array}{c|c|c|c} \\ \\ a_1 & a_2 & \cdots & a_n \\ \\ \\ \end{array}\right] = \left[\begin{array}{c|c|c|c} \\ \\ q_1 & q_2 & \cdots & q_n \\ \\ \\ \end{array}\right] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & & \vdots \\ & & \ddots & \\ & & & r_{nn} \end{bmatrix},$$

De façon équivalente,

$$\begin{aligned} a_1 =& r_{11}q_1, \\ a_2 =& r_{12}q_1 + r_{22}q_2, \\ & ... \\ a_i =& r_{1i}q_1 + r_{2i}q_2 + ... + r_{ni}q_i, \end{aligned}$$

ou encore:

$$A = \hat{Q}\hat{R}, \tag{1}$$

avec $\hat{Q}$ la matrice des vecteurs colonnes q_i et $\hat{R}$ la matrice des coefficients r_{ij}. C'est la **factorisation QR réduite**. $\hat{Q}$ est une matrice $m \times n$ et $\hat{R}$ est une matrice triangulaire supérieure $n \times n$.

La **factorisation QR complète** est une factorisation $A = QR$ avec Q unitaire la matrice augmentée de $\hat{Q}$ avec des vecteurs orthonormés et R triangulaire la matrice augmentée de $\hat{R}$ avec des lignes de zéros.

La méthode d'orthogonalisation de Gram-Schmidt pour la décomposition QR réduite est la suivante. On suppose qu'on a construit les vecteurs $q_1, ..., q_{j-1}$. A l'étape j in veut trouver une vecteur orthogonal aux q_i, $i < j$, qui se trouve dans le sous-espace engendré par les j premiers vecteurs a_i. Or,

$$v_j = a_j - \sum_{i=1}^{j-1} (q_i^* a_j) q_i \tag{2}$$

est un vecteur orthogonal à l'espace engendré par les $j-1$ premiers q_i, et est clairement dans le sous-espace engendré par les j premiers a_i. En divisant par $||v_j||$ on obtient le vecteur q_j recherché. Les coefficients

$$\begin{aligned} r_{ij} =& q_i^* a_j, \ i < j, \\ r_{jj} =& ||a_j - \sum_{i=1}^{j-1} r_{ij} q_i||. \end{aligned}$$

On peut écrire l'algorithme pour la factorisation QR réduite.

Algorithme Gram-Schmidt Classique.
Data: une matrice A de plein rang
Result: une décomposition QR réduite de A
for $j = 1$ **to** n **do**
  $v_j = a_j$;
  for $i = 1$ **to** $j - 1$ **do**
    $r_{ij} = q_i^* a_j$;
    $v_j = v_j - r_{ij} q_i$;
  end
  $r_{jj} = ||v_j||$;
  $q_j = v_j / r_{jj}$;
end

Cet algorithme montre que toute matrice de plein rang possède une factorisation QR réduite unique avec $r_{jj} > 0$. En pratique, cet algorithme n'est pas numériquement stable. Une autre façon de calculer les vecteurs q_i nous est donné en formant des projecteurs orthogonaux. Ici, on considère q_j comme le résultat de la projection orthogonale de a_j sur l'espace orthogonal à $q_1, q_2, ..., q_{j-1}$.

$$q_1 = P_1 a_1 / ||P_1 a_1||, ..., q_j = P_j a_j / ||P_j a_j||, ...$$

avec P_j la matrice de projection orthogonale sur l'espace orthogonal à $q_1, ..., q_{j-1}$. On a vu que le projecteur orthogonal qui projette sur l'espace engendré par les $j-1$ premiers q_i était $\hat{Q}_{j-1}\hat{Q}_{j-1}^*$ avec $\hat{Q}_{j-1} = [q_1|...|q_{j-1}]$. Le projecteur orthogonal qui projette sur l'espace orthogonal est le projecteur complémentaire

$$P_j = I - \hat{Q}_{j-1}\hat{Q}_{j-1}^*.$$

P_j est un projecteur orthogonal. Sa projection est orthogonale aux vecteurs q_i, $i < j$. En effet, P_j est tout simplement une autre façon d'écrire l'équation pour les v_j de Gram-Schmidt:

$$\begin{aligned} v_j = P_j a_j =& a_j - \hat{Q}_{j-1}\hat{Q}_{j-1}^* a_j, \\ =& a_j - \sum_{i=1}^{j-1} (q_i^* a_j) q_i. \end{aligned}$$

Cette réécriture permet d'avoir un algorithme de Gram-Schmidt numériquement stable en calculant les P_j comme une succession de projections. Si on définit $P_{\perp q}$ le projecteur orthogonal qui projette sur l'espace orthogonal à q, alors P_j s'écrit comme le produit des projecteurs $P_{\perp q_i}$, $i < j$,

$$P_j = P_{\perp q_{j-1}} P_{\perp q_2} ... P_{\perp q_1}$$

On exploite le fait que $P_j = P_{\perp q_{j-1}} P_{j-1}$ pour l'algorithme. Les matrices

$$P_{\perp q} = I - qq^*$$

et

$$
\begin{aligned}
P_{\perp q} v =& v - qq^* v, \\
=& v - (q^* v) q.
\end{aligned}
$$

On commence par initialiser les vecteurs $v_i^{(1)} = a_i$, $i = 1, ..., n$. Le premier vecteur v_1 est obtenu directement $v_1 = v^(1)_1$. Ensuite on calcule les produit $P_{\perp q_1} v_j^{(1)}$ qu'on nomme $v_j^{(2)}$ et on itère. À chaque étape i, on obtient le vecteur $v_{i+1} = v_{i+1}^{(i+1)}$ qu'on renormalisera à l'étape $i + 1$.

$$
\begin{array}{llllll}
 & & j=2 & j=3 & \dots & j=n \\
i=1 & q_1 = v_1^{(1)}/||v_1^{(1)}|| & \underbrace{P_{\perp q_1} v_2^{(1)}}_{=v_2^{(2)}=v_2} & \underbrace{P_{\perp q_1} v_3^{(1)}}_{=v_3^{(2)}} & \dots & \underbrace{P_{\perp q_1} v_n^{(1)}}_{=v_n^{(2)}} \\
i=2 & & q_2 = v_2/||v_2|| & \underbrace{P_{\perp q_2} v_3^{(2)}}_{=v_3^{(3)}=v_3} & \dots & \underbrace{P_{\perp q_2} v_n^{(2)}}_{=v_n^{(3)}} \\
i=3 & & & q_3 = v_3/||v_3|| & \dots & \underbrace{P_{\perp q_3} v_n^{(3)}}_{=v_n^{(4)}} \\
\vdots & & & & & \\
i=n & & & & & q_n = v_n/||v_n||.
\end{array}
$$

L'implémentation de l'algorithme est la suivante.

Algorithme Gram-Schmidt Modifié.
Data: une matrice A de plein rang
Result: une décomposition QR réduite de A
for $j = 1$ **to** n **do**
  $v_j = a_j$;
end
for $i = 1$ **to** n **do**
  $r_{ii} = ||v_i||$;
  $q_i = v_i / r_{ii}$;
  for $j = i + 1$ **to** n **do**
    $r_{ij} = q_i^* v_j$;
    $v_j = v_j - r_{ij} q_i$;
  end
end

5.2 Triangularisation de Householder

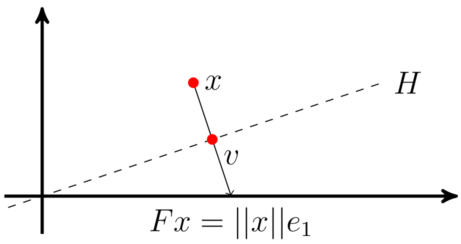
La méthode de Householder applique successivement des transformations $Q_1 A$, $Q_2 A$, ... sur les colonnes de A pour introduire des zéros sous la diagonale. Chaque matrice Q_k est unitaire et agit en introduisant des zéros sous la diagonale de la colonne k, sans affecter les premières $k-1$ colonnes. La matrice Q_k a la forme suivante

$$Q_k = \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}, \tag{3}$$

où I est la matrice identité de taille $(k-1) \times (k-1)$ et F est une matrice unitaire de taille $(m-k+1) \times (m-k+1)$. Si on a une matrice avec des zéros sous la diagonale des $k-1$ premières colonnes, le produit par Q_k est de la forme

$$\begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix} \begin{bmatrix} T & B \\ 0 & C \end{bmatrix} = \begin{bmatrix} T & B \\ 0 & FC \end{bmatrix}. \tag{4}$$

On voit que le produit par Q_k ne modifie que la sous-matrice inférieure droite. La sous-matrice FC doit avoir des zéros sous la diagonale de la première colonne. Si x est le vecteur première colonne de C, on veut $Fx = cste_1$, avec $e_1 = {}^t(1, 0, ..., 0)$. Comme F est unitaire, F doit préserver la norme de x, donc $Fx = \pm||x||e_1$. On veut donc envoyer x sur l'axe de la première coordonnée. Pour ce faire, on utilise le **réflecteur de Householder**.



L'idée du réflecteur est de réfléchir x orthogonalement à travers un hyperplan H. Le vecteur orthogonal à l'hyperplan H est $v = ||x||e_1 - x$. On sait déjà comment envoyer x *sur* le plan H, à l'aide des projecteurs orthogonaux. Ce projecteur est le projecteur qui projette sur l'espace othogonal à v, c.-à-d. $P_{\perp v}$.

$$P_{\perp v} = I - \frac{vv^*}{v^*v}$$

(On divise pour normaliser v.) On ne veut pas aller sur H mais de l'autre côté. Le réflecteur sera donc

$$F = I - 2\frac{vv^*}{v^*v}$$

F est unitaire: On a $F = F^*$, ce qui implique que et $F^*F = F^2$ et $F^2 = I$. Le réflecteur F n'est pas unique. On aurait pu envoyer x sur $-||x||e_1$. Pour des raisons de stabilité numérique, on veut envoyer x le plus loin possible. Pour cela on choisit le signe inverse du signe de x_1, la première composante de x. Le vecteur v devient $v = -\text{sign}(x_1)||x_1||e_1 - x$, ou de façon équivalente

$$v = \text{sign}(x_1)||x||e_1 + x. \tag{5}$$

L'implémentation de l'algorithme est la suivante.

Algorithme de Householder
Data: une matrice A de plein rang
Result: une matrice R de la décomposition QR réduite de A et une matrice de réflexions W. La matrice R est directement stockée dans A, et W est composée des vecteurs colonnes v_k.
for $k = 1$ **to** n **do**
  $x = A_{k:m,k}$ $v_k = \text{sign}(x_1)||x||e_1 + x$;
  $v_k = v_k/||v_k||$;
  $A_{k:m,k:m} = A_{k:m,k:m} - 2v_k(v_k^* A_{k:m,k:m})$;
end

Cet algorithme ne calcule pas explicitement la matrice Q, seulement les élements v_k pour former les Q_k. On a cependant les relations $Q^* = Q_n \cdots Q_2 Q_1$. En pratique on a souvent à calculer $Q^* b$. L'algorithme pour le calculer est

Calcul de $Q^* b$
Data: Un vecteur b et les vecteurs v_k
Result: Le produit $Q^* b$.
for $k = 1$ **to** n **do**
  $b_{k:m} = b_{k:m} - 2v_k(v_k^* b_{k:m})$;
end

References

[1] Jean Fresnel. *Algebre des matrices*. Hermann, 2013.

[2] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.