

ML 12 Factorisation QR

1 Factorisation QR

Nous introduisons une décomposition matricielle qui nous permettra plus tard d'implémenter les calculs des décompositions en valeurs propres et en valeurs singulières.

Nous cherchons des vecteurs orthonormaux qui forment une base pour les espaces couverts par les colonnes successives d'une matrice $\mathbf{A} \in \mathbb{R}^{M \times N}$. Notons :

- $\langle \mathbf{a}_1 \rangle$: l'espace (la ligne) couvert par la première colonne de \mathbf{A}
- $\langle \mathbf{a}_1, \mathbf{a}_2 \rangle$: l'espace (le plan) couvert par les deux premières colonnes de \mathbf{A}
- Etc. : Etc.

Nous avons $\langle \mathbf{a}_1 \rangle \subseteq \langle \mathbf{a}_1, \mathbf{a}_2 \rangle \subseteq \dots \subseteq \langle \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N \rangle$.

Nous cherchons des vecteurs orthogonaux \mathbf{q}_i tels que :

$$\langle \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_j \rangle = \langle \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_j \rangle \quad \text{pour } j = 1, 2, \dots, N$$

Ce qui peut s'écrire sous une forme matricielle :

$$(\mathbf{a}_1 \ \mathbf{a}_2 \ \dots \ \mathbf{a}_N) = (\mathbf{q}_1 \ \mathbf{q}_2 \ \dots \ \mathbf{q}_N) \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1N} \\ 0 & r_{22} & \dots & r_{2N} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & r_{NN} \end{pmatrix}$$

Sans perte de généralité, considérons la situation où $M > N$. Nous appelons alors cette expression, $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ (avec $\hat{\mathbf{Q}} \in \mathbb{R}^{M \times N}$ et $\hat{\mathbf{R}} \in \mathbb{R}^{N \times N}$), la forme réduite de la factorisation QR.

Nous pouvons aussi considérer la forme dite complète de la factorisation QR, avec $\mathbf{Q} \in \mathbb{R}^{M \times M}$ et $\mathbf{R} \in \mathbb{R}^{M \times N}$. Dans ce cas, les $M - N$ dernières lignes de \mathbf{R} sont nulles et il suffit de compléter $\hat{\mathbf{Q}}$ en lui ajoutant $M - N$ colonnes orthogonales.

2 Orthogonalisation de Gram-Schmidt

Nous présentons l'algorithme de Gram-Schmidt pour la décomposition QR.

Le principe est de construire \mathbf{q}_j orthogonal à $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{j-1}$ en retirant à la colonne \mathbf{a}_j ses composantes selon les directions de $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{j-1}$:

$$\begin{aligned} \mathbf{v}_j &= \mathbf{a}_j - (\mathbf{q}_1^T \mathbf{a}_j) \mathbf{q}_1 - (\mathbf{q}_2^T \mathbf{a}_j) \mathbf{q}_2 - \dots - (\mathbf{q}_{j-1}^T \mathbf{a}_j) \mathbf{q}_{j-1} \\ \mathbf{q}_j &= \mathbf{v}_j / \|\mathbf{v}_j\| \end{aligned}$$

Nous obtenons ainsi la matrice $\hat{\mathbf{Q}}$. Pour $\hat{\mathbf{R}}$, nous avons, par construction des \mathbf{q}_j :

$$r_{ij} = \mathbf{q}_i^T \mathbf{a}_j \quad i \neq j$$

$$|r_{jj}| = \|\mathbf{a}_j - \sum_{i=1}^{j-1} r_{ij} \mathbf{q}_i\| \quad \text{par convention, nous choisissons } r_{jj} > 0$$

Nous avons ainsi dérivé l'algorithme classique de Gram-Schmidt :

Gram-Schmidt classique (GS)

Résultat: Décomposition QR de la matrice \mathbf{A}

Entrée: $\mathbf{A} \in \mathbb{R}^{M \times N}$, avec $M \geq N$

Sortie : $\mathbf{Q} \in \mathbb{R}^{M \times N}$ une matrice orthonormale

$\mathbf{R} \in \mathbb{R}^{N \times N}$ une matrice triangulaire supérieure

telles que $\mathbf{A} = \mathbf{QR}$

```

1 pour j ← 1 à N faire
2   vj ← aj
3   pour i ← 1 à j - 1 faire
4     rij ← qiT aj
5     vj ← vj - rij qi
6   finpour
7   rjj ← ||vj||
8   qj ← vj/rjj
9 finpour

```

3 Algorithme de Gram-Schmidt modifié

3.1 Dérivation de l'algorithme de Gram-Schmidt modifié

Nous présentons une autre manière de voir ce résultat qui nous mènera à un algorithme équivalent mais numériquement plus stable.

Nous pouvons considérer que \mathbf{q}_j est obtenu en projetant \mathbf{a}_j sur l'espace orthogonal à $\langle \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{j-1} \rangle$, cet espace étant par construction égal à $\langle \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{j-1} \rangle$:

$$\mathbf{q}_j = \frac{\mathbf{P}_j \mathbf{a}_j}{\|\mathbf{P}_j \mathbf{a}_j\|} \quad ; \quad \mathbf{P}_j = \mathbf{I} - \mathbf{Q}_{j-1} \mathbf{Q}_{j-1}^T \quad \text{avec } \mathbf{Q}_{j-1} = (\mathbf{q}_1 \ \mathbf{q}_2 \ \dots \ \mathbf{q}_{j-1})$$

De façon équivalente, \mathbf{P}_j peut s'exprimer comme le produit de projecteurs sur les espaces orthogonaux à \mathbf{q}_1 , puis \mathbf{q}_2 , etc. jusqu'à \mathbf{q}_{j-1} :

$$\mathbf{P}_j = \mathbf{P}_{\perp \mathbf{q}_{j-1}} \mathbf{P}_{\perp \mathbf{q}_{j-2}} \dots \mathbf{P}_{\perp \mathbf{q}_2} \mathbf{P}_{\perp \mathbf{q}_1} \quad \text{avec } \mathbf{P}_{\perp \mathbf{q}} = \mathbf{I} - \mathbf{q} \mathbf{q}^T$$

Dans la version classique de Gram-Schmidt, $\mathbf{v}_j = \mathbf{P}_j \mathbf{a}_j$.

Dans la version modifiée de Gram-Schmidt :

$$\mathbf{v}_j = \mathbf{P}_{\perp \mathbf{q}_{j-1}} \mathbf{P}_{\perp \mathbf{q}_{j-2}} \dots \mathbf{P}_{\perp \mathbf{q}_2} \mathbf{P}_{\perp \mathbf{q}_1} \mathbf{a}_j$$

Ainsi, nous observons que dès que \mathbf{q}_1 est connu ($\mathbf{q}_1 = \mathbf{v}_1 / \|\mathbf{v}_1\|$, $\mathbf{v}_1 = \mathbf{a}_1$), nous pouvons appliquer $\mathbf{P}_{\perp \mathbf{q}_1}$ à $\mathbf{a}_2, \mathbf{a}_3, \dots, \mathbf{a}_N$. Puis, quand \mathbf{q}_2 est connu, nous pouvons appliquer $\mathbf{P}_{\perp \mathbf{q}_2}$ à $\mathbf{P}_{\perp \mathbf{q}_1} \mathbf{a}_3, \mathbf{P}_{\perp \mathbf{q}_1} \mathbf{a}_4, \dots, \mathbf{P}_{\perp \mathbf{q}_1} \mathbf{a}_N$. Etc. Nous en déduisons l'algorithme de Gram-Schmidt modifié.

Gram-Schmidt modifié (MGS)

Résultat: Décomposition QR de la matrice \mathbf{A}

Entrée: $\mathbf{A} \in \mathbb{R}^{M \times N}$, avec $M \geq N$

Sortie : $\mathbf{Q} \in \mathbb{R}^{M \times N}$ une matrice orthonormale

$\mathbf{R} \in \mathbb{R}^{N \times N}$ une matrice triangulaire supérieure

telles que $\mathbf{A} = \mathbf{QR}$

```
1 pour i ← 1 à N faire
2   | vi ← ai
3 finpour
4 pour i ← 1 à N faire
5   | rii ← ||vi||
6   | qi ← vi/rii
7   | pour j ← i + 1 à N faire
8     | rij ← qiTvj
9     | vj ← vj - rijqi
10  | finpour
11 finpour
```

3.2 Test de l'algorithme

Nous comparons notre implémentation (voir la fonction `mgs` dans le code source associé à ce module) avec le résultat d'un petit exemple issu de Wikipedia.

```
mat <- matrix(c(3,2,1,2),nrow=2,ncol=2,byrow = TRUE)
qr <- mgs(mat)
all.equal(qr$Q, (1/(sqrt(10))) * matrix(c(3,1,-1,3),nrow=2,ncol=2))

## [1] TRUE
```

3.3 Comparaison de la stabilité numérique des deux versions de l'algorithme de Gram-Schmidt

Observons comment de petites erreurs de calculs se propagent dans le cas de l'algorithme de Gram-Schmidt (GS). Mettons qu'une petite erreur ait été commise sur le calcul de \mathbf{q}_2 qui s'en trouve de ce fait n'être pas parfaitement perpendiculaire à \mathbf{q}_1 :

$$\mathbf{q}_1^T \mathbf{q}_2 = \epsilon > 0$$

Quelle est l'effet de cette erreur sur la valeur de \mathbf{v}_3 ?

$$\begin{aligned}\mathbf{v}_3 &= \mathbf{a}_3 - (\mathbf{q}_1^T \mathbf{a}_3)\mathbf{q}_1 - (\mathbf{q}_2^T \mathbf{a}_3)\mathbf{q}_2 \\ \mathbf{q}_2^T \mathbf{v}_3 &= \mathbf{q}_2^T \mathbf{a}_3 - (\mathbf{q}_1^T \mathbf{a}_3)\epsilon - (\mathbf{q}_2^T \mathbf{a}_3) = -(\mathbf{q}_1^T \mathbf{a}_3)\epsilon \\ \mathbf{q}_1^T \mathbf{v}_3 &= \mathbf{q}_1^T \mathbf{a}_3 - (\mathbf{q}_1^T \mathbf{a}_3) - (\mathbf{q}_2^T \mathbf{a}_3)\epsilon = -(\mathbf{q}_2^T \mathbf{a}_3)\epsilon\end{aligned}$$

L'erreur commise sur \mathbf{q}_2 s'est transmise à \mathbf{q}_3 qui n'est plus parfaitement orthogonal à \mathbf{q}_1 et \mathbf{q}_2 .

Pour l'algorithme de Gram-Schmidt modifié (MGS), nous avons :

$$\begin{aligned}
\mathbf{v}_3^{(0)} &= \mathbf{a}_3 \\
\mathbf{v}_3^{(1)} &= \mathbf{v}_3^{(0)} - \mathbf{q}_1^T \mathbf{v}_3^{(0)} \mathbf{q}_1 \\
\mathbf{v}_3 &= \mathbf{v}_3^{(1)} - \mathbf{q}_2^T \mathbf{v}_3^{(1)} \mathbf{q}_2 \\
\mathbf{q}_2^T \mathbf{v}_3 &= \mathbf{q}_2^T \mathbf{v}_3^{(1)} - \mathbf{q}_2^T \mathbf{v}_3^{(1)} = 0
\end{aligned}$$

Il n'y a pas d'erreur pour l'orthogonalité entre \mathbf{q}_3 et \mathbf{q}_2 . Etudions maintenant l'erreur commise sur l'orthogonalité entre \mathbf{q}_3 et \mathbf{q}_1 .

$$\begin{aligned}
\mathbf{q}_1^T \mathbf{v}_3 &= \mathbf{q}_1^T \mathbf{v}_3^{(1)} - \mathbf{q}_2^T \mathbf{v}_3^{(1)} \epsilon \\
\mathbf{q}_2^T \mathbf{v}_3^{(1)} &= \mathbf{q}_2^T \mathbf{v}_3^{(0)} - \mathbf{q}_1^T \mathbf{v}_3^{(0)} \epsilon \\
\mathbf{q}_1^T \mathbf{v}_3^{(1)} &= \mathbf{q}_1^T \mathbf{v}_3^{(0)} - \mathbf{q}_1^T \mathbf{v}_3^{(0)} \epsilon = 0 \\
\mathbf{q}_1^T \mathbf{v}_3 &= -(\mathbf{q}_2^T \mathbf{v}_3^{(0)} - \mathbf{q}_1^T \mathbf{v}_3^{(0)} \epsilon) = -\mathbf{q}_2^T \mathbf{v}_3^{(0)} \epsilon + \mathbf{q}_1^T \mathbf{v}_3^{(0)} \epsilon^2
\end{aligned}$$

Ainsi, comme pour la version classique de Gram-Schmidt, une erreur de l'ordre de grandeur de ϵ est commise sur l'orthogonalité entre \mathbf{q}_1 et \mathbf{q}_3 . Par contre, il n'y a pas de propagation de l'erreur pour l'orthogonalité entre \mathbf{q}_2 et \mathbf{q}_3 . C'est pourquoi l'algorithme de Gram-Schmidt modifié est numériquement plus stable que l'algorithme de Gram-Schmidt classique.

4 Autres algorithmes

Notons qu'une autre approche très souvent employée pour le calcul de la décomposition QR est la méthode dite de Householder. Elle assure le maintien d'une quasi parfaite orthogonalité des colonnes de \mathbf{Q} . Il existe également l'approche dite des rotations de Givens, plus facilement parallélisable et permettant la mise à jour de la décomposition suite à l'ajout d'une ligne à la matrice \mathbf{X} .

Par contre, Gram-Schmidt permet de calculer la forme réduite de la décomposition QR, alors que les algorithmes de Givens et Householder calculent la forme complète. Ainsi, Gram-Schmidt n'est pas tout à fait aussi stable mais peut parfois être plus efficace.