ECN 6338 Cours 2

La résolution de systèmes d'équations linéaires

William McCausland

2022-01-19

Systèmes triangulaires

- Notation :
  - L, une matrice triangulaire inférieure $n \times n$;
  - R, une matrice triangularie supérieure $n \times n$.
- Il est facile de voir si la matrice est inversible : ssi aucun élément diagonal n'est nul.
- Pour le n-vecteur colonne y, la résolution de
  - $y = Lx$ est facile par substitution avant
  - $y = Rx$ est facile par substitution arrière
- Pour le n-vecteur ligne y, la solution de $y = xL$ est la transposée de la solution de $y^\top = L^\top x^\top$.

Systèmes orthogonaux I

- Une matrice $n \times n$ Q est *orthogonal* ssi $Q^{-1} = Q^\top$.
- Définition équivalente : ... ssi $Q^\top Q = QQ^\top = I_n$.
- Exemple, reflections :

$$Q = Q^\top = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad Q = Q^\top = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}.$$

- Exemple, permutation :

$$Q = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad Q^\top = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

- Exemple, rotation par un angle θ :

$$Q = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

$$Q^\top = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \cos -\theta & -\sin -\theta \\ \sin -\theta & \cos -\theta \end{bmatrix}$$

Systèmes orthogonaux II

Notes :

- La solution du système $y = Qx$ est $x = Q^\top y$.
- Si Q_1 et Q_2 sont orthogonales,
  - $Q_1^\top$ l'est aussi,
  - $Q_1 Q_2$ l'est aussi :

$$(Q_1 Q_2)^\top (Q_1 Q_2) = Q_2^\top Q_1^\top Q_1 Q_2 = Q_2^\top Q_2 = I_n,$$

$$Q_1 Q_2 (Q_1 Q_2)^\top = Q_1 Q_2 Q_2^\top Q_1^\top = I_n.$$

Décompositions

Trois décompositions permettent la division de la résolution d'un système plus générale en systèmes triangulaires ou orthogonaux :

- Décomposition LU : $A = LU$ pour A général $n \times n$
- Décomposition Cholesky : $A = LL^\top = R^\top R$, pour A symmétrique et définie positive $n \times n$.
- Décomposition QR : $A = QR$ pour A général $m \times n$

La résolution de systèmes avec la décomposition LU

- En théorie, la décomposition est $A = LU$, où
  - L est triangulaire inférieure $n \times n$.
  - U est triangulaire supérieure $n \times n$.
- En pratique, pour garder contre la division par un nombre près de zéro, la décomposition est $A = PLU$, où
  - P est une matrice de permutation.
- Pour résoudre le système $y = Ax$,
  - Calculer la décomposition $A = PLU$.
  - Le systéme s'écrit $y = PLUx$
  - Permuter y avec $P^\top = P^{-1}$: $\tilde{y} = P^\top y$.
  - Le système s'écrit $\tilde{y} = LUx$.
  - Résoudre le système $\tilde{y} = Lz$ pour trouver $z \equiv Ux$.
  - Résoudre le système $Ux = z$ pour trouver x.
- Pour les systèmes $n \times n$, toutes les décompositions sont $O(n^3)$, la substitution (avant et arrière) est $O(n^2)$ et la multiplication matrice-vecteur est $O(n^2)$.

Exemple, oligopole (Judd, chapitre 3, exo 7)

Voici les fonctions de meilleure réponse de trois firmes en jeu de Cournot

$$
\begin{aligned}
q_1 &= 5 - 0.5q_2 - 0.3q_3, \\
q_2 &= 7 - 0.6q_1 - 0.1q_3, \\
q_3 &= 4 - 0.2q_1 - 0.4q_2.
\end{aligned}
$$

En équation matriciel $Aq = y$, où

$$
A = \begin{bmatrix} 1.0 & 0.5 & 0.3 \\ 0.6 & 1.0 & 0.1 \\ 0.2 & 0.4 & 1.0 \end{bmatrix}, \quad y = \begin{bmatrix} 5 \\ 7 \\ 4 \end{bmatrix}.
$$

L'équilibre Cournot, calculé numériquement

```r
data = c(1.0, 0.6, 0.2, 0.5, 1.0, 0.4, 0.3, 0.1, 1.0)
A = matrix(data, nrow = 3, ncol=3, byrow=FALSE)
y = c(5, 7, 4)
q = solve(A, y)
q
```

```
## [1] 1.671554 5.865103 1.319648
```

Notes :

- `solve` utilise la commande `DGESV` de `LAPACK`.
- De la documentation `LAPACK` : "LU decomposition with partial pivoting and row interchanges"
- Pas d'accès à la décomposition PLU, paraît-il.
- Cependant, si on veut résoudre $Ax^i = y^i$ pour plusieurs i, on peut résoudre $AX = Y$, où $X = [x^1 \cdots x^N]$ et $Y = [y^1 \cdots y^N]$, avec `X = solve(A, Y)`.

La décomposition QR

Pour une matrice A, $n \times n$, la décomposition suivante existe toujours :

$$A = QR,$$

où Q est orthogonal, R est triangulaire supérieur.

- Normalisation habituelle : choisir Q pour que les éléments diagonaux de R soient non-négatifs.
- Pour A inversible, la décomposition normalisée est unique.
- Il y a des décompositions analogues QL, RQ, LQ.

Pour une matrice X, $n \times k$, $n > k$ (souvent $n \gg k$), on peut faire la décomposition

$$A = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1$$

où Q est $n \times n$, R est $n \times k$, Q_1 est $n \times k$ et R_1 est $k \times k$.

La résolution de systèmes avec la décomposition QR, I

Pour A carré est inversible, tout est simple:

- Écrire le système à résoudre, $y = Ax$, comme $y = QRx$.
- Définir $\tilde{y} = Q^\top y$ et maintenant le système s'écrit $\tilde{y} = Rx$.
- Résoudre ce système triangulaire avec la substituion arrière.

Pour A grande et mince, l'application principale en économie le calcul des coefficients moindres carrées linéaire.

Aparté, régression linéaire

- Équation de régression pour une unité d'observation :

$$y_i = x_{i1}\beta_1 + \ldots + x_{ik}\beta_k + \epsilon_i, \quad E[\epsilon_i | x_{i1}, \ldots, x_{ik}] = 0.$$

- Cette équations en vecteurs :

$$y_i = x_i^\top \beta + \epsilon_i.$$

- Toutes les équations, $i = 1, \ldots, n$, en matrices :

$$y = X\beta + \epsilon,$$

  où y et ϵ sont $n \times 1$, X est $n \times k$, β est $k \times 1$.
- L'estimateur MCO b de β est $(X^\top X)^{-1} X^\top y$.

La résolution de systèmes avec la décomposition QR, II

- On veut calculer $b \equiv (X^\top X)^{-1} X^\top y$.
- Soit $X = QR$ la décomposition QR de X.
- Rappeler que $QR = Q_1 R_1$, pour la sous-matrice $n \times k$ Q_1 et la sous-matrice $k \times k$ R_1.
- On peut écrire

$$\begin{aligned}
(X^\top X)^{-1} X^\top y &= (R^\top Q^\top Q R)^{-1} R^\top Q^\top y \\
&= (R^\top R)^{-1} R^\top Q^\top y \\
&= (R_1^\top R_1)^{-1} R_1^\top Q_1^\top y \\
&= R_1^{-1} (R_1^\top)^{-1} R_1^\top Q_1^\top y \\
&= R_1^{-1} Q_1^\top y
\end{aligned}$$

- On peut calculer Q_1 sans calculer Q_2 (Tant mieux, Q a $n \times n$ éléments à calculer)
- Les résultats sont numériquement plus stables que si on décompose $X^\top X$.

La décomposition Cholesky

- Soit Σ une matrice symmétrique et positive définie $n \times n$.
- La décomposition est $\Sigma = LL^\top = R^\top R$, où
  - $L = R^\top$ est triangulaire inférieure $n \times n$.
- C'est une décomposition LU sans permutation.
- Le calcul est numériquement plus stable, sans recours au pivots.
- Deux opérations intéressantes dans le contexte des variables aléatoires gaussiennes multivariées.
  - Multiplication directe Lx
  - Résolution du système $Lx = y$.

Exemple, tirage de variables aléatoires gaussiennes multivariées

Problème : tirer $X \sim N(\mu, \Sigma)$, où la variance Σ de X est $n \times n$ et positive définie.

- La décomposition Cholesky est $\Sigma = LL^\top$.
- Pour $u \sim N(0, I_n)$, $Lu \sim N(0, \Sigma)$, parce que $LI_nL^\top = LL^\top = \Sigma$.
- Alors $Lu + \mu \sim N(\mu, \Sigma)$.

Tirage de variables aléatoire $N(\mu, \Sigma)$ en R

```
M = 100   # Nombre de tirages
n = 3     # Nombre d'éléments du vecteur aléatoire
mu = c(4, 1, 3)  # Moyenne mu, variance Sigma
Sigma = matrix(c(4,1,3, 1,1,1.5, 3,1.5,9), nrow=3, ncol=3)
R = chol(Sigma)  # Facteur de Cholesky supérieure

U = matrix(rnorm(n*M), nrow=n, ncol=M) # U_i ~ N(0,I)
X = t(R) %*% U + mu # X_i ~ N(mu, Sigma), i=1,...,M
rowMeans(X)
```

```
## [1] 3.930382 1.053437 3.155961
```

```
var(t(X))
```

```
##           [,1]      [,2]     [,3]
## [1,] 3.8968993 0.9176063 2.495346
## [2,] 0.9176063 1.0130663 1.292524
## [3,] 2.4953462 1.2925241 7.314193
```

Exemple, évaluation de la densité gaussienne multivariée

Si $X \sim N(\mu, \Sigma)$, la densité multivariée est

$$f(x) = |\Sigma|^{-1/2}(2\pi)^{-k/2} \exp\left[-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right]$$

Pour calculer $(x-\mu)^\top \Sigma^{-1}(x-\mu)$:

$$\begin{aligned}
(x-\mu)^\top \Sigma^{-1}(x-\mu) &= (x-\mu)^\top (LL^\top)^{-1}(x-\mu) \\
&= (x-\mu)^\top L^{-\top} L^{-1}(x-\mu)\,, \\
&= \|L^{-1}(x-\mu)\|
\end{aligned}$$

et $L^{-1}(x-\mu)$ est la solution du sytème triangulaire $Lu = (x-\mu)$.

Pour calculer $|\Sigma|$: $|\Sigma| = |L||L^\top|$ où $|L| = |L^\top|$ est le produit des éléments diagonaux de L, parce qu'elle est triangulaire.

Matrice creuse (sparse matrix), format triple

Matrice creuse A, $m \times n$, avec N éléments $(m = 5, n = 6, N = 5)$:

$$A = \begin{bmatrix} 0 & 0 & A_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{32} & 0 & 0 & A_{35} & 0 \\ 0 & 0 & A_{43} & 0 & 0 & 0 \\ 0 & A_{52} & 0 & 0 & 0 & 0 \end{bmatrix}$$

Triples en ordre colonne par colonne (column major order) :

i	j	x
3	2	A_{32}
5	2	A_{52}
1	3	A_{13}
4	3	A_{43}
3	5	A_{35}

Matrice creuse, format colonne compressée

$$A = \begin{bmatrix} 0 & 0 & A_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{32} & 0 & 0 & A_{35} & 0 \\ 0 & 0 & A_{43} & 0 & 0 & 0 \\ 0 & A_{52} & 0 & 0 & 0 & 0 \end{bmatrix}$$

i	x
3	A_{32}
5	A_{52}
1	A_{13}
4	A_{43}
3	A_{35}

j	0	1	2	3	4	5	6
p	0	0	2	4	4	5	5

Mise en oeuvre en R, format collonne compressé par défaut

```r
library(Matrix)

m <- Matrix(nrow=5, ncol=6, data=0, sparse=TRUE)
m[1, 3] <- 13
m[3, 2] <- 32
m[3, 5] <- 35
m[4, 3] <- 43
m[5, 2] <- 52
m
```

```
## 5 x 6 sparse Matrix of class "dgCMatrix"
##
## [1,] .  . 13 .  . .
## [2,] .  .  . .  . .
## [3,] . 32  . . 35 .
## [4,] .  . 43 .  . .
## [5,] . 52  . .  . .
```

Représentation en format triple

```r
mT <- as(m, "dgTMatrix")
str(mT)
```

```
## Formal class 'dgTMatrix' [package "Matrix"] with 6 slots
##   ..@ i       : int [1:5] 2 4 0 3 2
##   ..@ j       : int [1:5] 1 1 2 2 4
##   ..@ Dim     : int [1:2] 5 6
##   ..@ Dimnames:List of 2
##   .. ..$ : NULL
##   .. ..$ : NULL
##   ..@ x       : num [1:5] 32 52 13 43 35
##   ..@ factors : list()
```

Représentation en format colonne compressée

```r
str(m)
```

```
## Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
##   ..@ i       : int [1:5] 2 4 0 3 2
##   ..@ p       : int [1:7] 0 0 2 4 4 5 5
##   ..@ Dim     : int [1:2] 5 6
##   ..@ Dimnames:List of 2
##   .. ..$ : NULL
##   .. ..$ : NULL
##   ..@ x       : num [1:5] 32 52 13 43 35
##   ..@ factors : list()
```

Opérations avec les matrices creuses

Opérations rapides et simples

- Soit `A` une matrice creuse $m \times n$.
- Quand $N = O(n)$ et les éléments sont bien dispersés, la recherche, l'insertion et la suppression (lookup, insertion, deletion) sont des problèmes
  - $O(1)$ pour le format colonne compressée,
  - $O(\log N)$ pour le format triple.
- Opérations rapides :
  - Recherche : `x = A[i,j]`
  - Insertion : `A[i,j] = Aij`
  - Suppression : `A[i,j] = 0`
  - Multiplication par un vecteur : `y = A %*% x` ou `y = x %*% A`

Opérations lentes ou difficiles

- Les décompositions LU, QR, Cholesky sont possibles (et disponibles en R) mais les résultats sont souvent des matrices denses.

Multiplication pour une matrice en format triple

- Soit `A` une matrice creuse en format triple, $n \times n$ avec N éléments non-nuls.
- Soit `x` un vecteur, $n \times 1$.
- On veut calculer le vecteur dense `y = A %*% x`.
- Algorithme : $y \leftarrow 0$ puis pour $k = 1, \ldots, N$,

$$y_{i_k} \leftarrow y_{i_k} + A_{i_k, j_k} x_{j_k}.$$