feuille-exercices-1

October 10, 2023

1 Feuille d'exercices 1

1.1 Exercice 1. Algorithme pour la décomposition de Cholesky

On rappelle (voir notes de cours) que les matrices symmétriques $M = (m_{i,j})_{0 \leq i,j \leq n-1} \in \mathbb{R}^{n \times n}$ définies positives admettent une decomposition dite de Cholesky :

$$M = LL^T \qquad \text{avec} \quad L = \begin{pmatrix} l_{0,0} & 0 & . & . & (0) \\ l_{1,0} & l_{1,1} & 0 & . & . \\ . & . & . & . & . \\ l_{n-2,0} & l_{n-2,1} & . & . & 0 \\ l_{n-1,0} & l_{n-1,1} & . & . & l_{n-1,n-1} \end{pmatrix}$$

où les coefficients diagonaux de la matrice triangulaire inférieure L sont non nuls. Cet exercice vise à implémenter un algorithme pour calculer cette décomposition.

1. Montrez à la main que pour $i = 0, ..., n-1$ et $j = 0, ..., i$:

$$m_{i,j} = \sum_{k=0}^{j} l_{i,k} l_{j,k}$$

2. En déduire que pour $j = 0, ..., n-1$:

$$(1) \qquad l_{j,j} = \sqrt{m_{j,j} - \sum_{k=0}^{j-1} l_{j,k}^2}$$

   et pour tout $i = j+1, ..., n-1$:

$$(2) \qquad l_{i,j} = \frac{m_{i,j} - \sum_{k=0}^{j-1} l_{i,k} l_{j,k}}{l_{j,j}}.$$

3. Grâce aux identités (1) et (2), on remarque que si l'on connait les j premières colonnes de L, on peut en déduire la $j+1$-ième. En effet, si on connaît les coefficients $l_{i,j'}$ pour tous $j' \leq j-1$ et $i = j', ..., n-1$, alors on en déduit la valeur de $l_{j,j}$ en utilisant (1), et la valeur de $l_{i,j}$ pour $i > j$ en utilisant (2).

Ecrire en Python une fonction `cholesky(M)` qui retourne la matrice L de la décomposition de Cholesky d'une matrice symmétrique définie positive M en implémentant cette méthode.

4. Tester votre fonction `cholesky(M)` sur deux exemples de matrices M symmétriques définies positives de votre choix.

5. En combien d'opérations la fonction `cholesky(M)` calcule-t-elle la matrice U ? Donner une réponse sous la forme $O(n^\alpha)$ où α est à déterminer.

1.2 Exercice 2. Introduction à l'analyse par composantes principales

(pour plus d'informations sur l'analyse par composantes principales, on pourra consulter "Analyse factorielles simples et multiples, B. Escofier et J. Pagès, Dunod")

Cet exercice s'attaque au problème suivant d'analyse de données : étant donné un nuage de points dans $\mathbb{R}^N$ (avec N grand) peut-on projeter ce nuage sur seulement deux dimensions de manière à garder un maximum d'informations ? Nous allons mettre en oeuvre l'analyse par composantes principales (ACP).

Le tableau T suivant représente l'évolution de la dette publique entre 1990 et 2014 pour certains pays (données du Fonds Monétaire International). La dette publique est mesurée en pourcentage par rapport au produit intérieur brut. On écrira $T_{pays,annee}$ pour repérer les entrées, i.e. $T_{Japon,1992} = 71$.

$$T = \begin{pmatrix} & 1990 & 1992 & 1994 & 1996 & 1998 & 2000 & 2002 & 2004 & 2006 \\ \text{Etats Unis} & 62 & 69 & 69 & 68 & 62 & 53 & 55 & 65 & 64 \\ \text{France} & 35 & 40 & 50 & 60 & 61 & 59 & 60 & 66 & 65 \\ \text{Espagne} & 41 & 44 & 57 & 66 & 63 & 58 & 51 & 45 & 39 \\ \text{Allemagne} & 42 & 41 & 47 & 58 & 59 & 59 & 59 & 65 & 66 \\ \text{Royaumes-Unis} & 29 & 34 & 41 & 45 & 42 & 37 & 34 & 39 & 41 \\ \text{Canada} & 75 & 89 & 98 & 101 & 94 & 81 & 80 & 72 & 70 \\ \text{Japon} & 67 & 71 & 86 & 102 & 122 & 144 & 164 & 181 & 186 \\ \text{Coree du sud} & 8 & 8 & 8 & 7 & 15 & 17 & 18 & 23 & 29 \\ \text{Singapour} & 73 & 79 & 67 & 70 & 80 & 81 & 95 & 97 & 88 \\ \text{Inde} & 48 & 77 & 73 & 66 & 68 & 74 & 83 & 83 & 77 \\ \text{Chine} & 7 & 5 & 6 & 21 & 20 & 23 & 26 & 26 & 25 \\ \text{Malaisie} & 75 & 60 & 44 & 33 & 34 & 33 & 40 & 42 & 40 \\ \text{Senegal} & 63 & 62 & 94 & 77 & 83 & 74 & 68 & 48 & 22 \\ \text{Cote d'Ivoire} & 110 & 117 & 151 & 107 & 97 & 98 & 89 & 78 & 79 \end{pmatrix}$$

	2008	2010	2012	2014
Etats Unis	73	95	102	105
France	68	82	90	95
Espagne	39	60	85	99
Allemagne	65	81	80	74
Royaumes-Unis	50	76	85	88
Canada	68	81	85	86
Japon	192	216	238	249
Coree du sud	28	31	32	36
Singapour	94	100	106	100
Inde	75	67	69	68
Chine	27	33	34	40
Malaisie	40	52	55	56
Senegal	24	36	43	54
Cote d'Ivoire	71	63	45	47

À chaque pays correspond un vecteur de la dette par années $x_{pays} = (T_{pays,annees})_{annees \in \{1990,1992,\ldots,2014\}} \in \mathbb{R}^{13}$. Le nuage des points formé par la distribution de la dette par années pour chacun des pays est $\{x_{pays}\}_{pays \in \{Etats\ Unis,\ldots,Cote\ d'Ivoire\}}$.

Il est très difficile de discerner ce qui distingue l'évolution de la dette entre les pays, à cause du très grand nombre de données. On se demande donc dans cet exercice si l'on peut représenter efficacement toutes ces données avec seulement deux dimensions, et en tirer des conclusions. Le code ci-dessous donne le tableau T en code python adapté au module numpy :

[ ]:
```
import numpy as np
T=np.array([
[  62  , 69  , 69  , 68  , 62  , 53  , 55  , 65  , 64  , 73  , 95  , 102 , 105 ],
[  35  , 40  , 50  , 60  , 61  , 59  , 60  , 66  , 65  , 68  , 82  , 90  , 95   ],
[  41  , 44  , 57  , 66  , 63  , 58  , 51  , 45  , 39  , 39  , 60  , 85  , 99 ],
[  42  , 41  , 47  , 58  , 59  , 59  , 59  , 65  , 66  , 65  , 81  , 80  , 74 ],
[  29  , 34  , 41  , 45  , 42  , 37  , 34  , 39  , 41  , 50  , 76  , 85  , 88 ],
[  75  , 89  , 98  , 101 , 94  , 81  , 80  , 72  , 70  , 68  , 81  , 85  , 86 ],
[  67  , 71  , 86  , 102 , 122 , 144 , 164 , 181 , 186 , 192 , 216 , 238 , 249 ],
[  8   , 8   , 8   , 7   , 15  , 17  , 18  , 23  , 29  , 28  , 31  , 32  , 36 ],
[  73  , 79  , 67  , 70  , 80  , 81  , 95  , 97  , 88  , 94  , 100 , 106 , 100 ],
[  48  , 77  , 73  , 66  , 68  , 74  , 83  , 83  , 77  , 75  , 67  , 69  , 68 ],
[  7   , 5   , 6   , 21  , 20  , 23  , 26  , 26  , 25  , 27  , 33  , 34  , 40 ],
[  75  , 60  , 44  , 33  , 34  , 33  , 40  , 42  , 40  , 40  , 52  , 55  , 56 ],
[  63  , 62  , 94  , 77  , 83  , 74  , 68  , 48  , 22  , 24  , 36  , 43  , 54 ],
[  110 , 117 , 151 , 107 , 97  , 98  , 89  , 78  , 79  , 71  , 63  , 45  , 47   ]])
T
```

[ ]:
```
array([[ 62,  69,  69,  68,  62,  53,  55,  65,  64,  73,  95, 102, 105],
       [ 35,  40,  50,  60,  61,  59,  60,  66,  65,  68,  82,  90,  95],
       [ 41,  44,  57,  66,  63,  58,  51,  45,  39,  39,  60,  85,  99],
```

```
[ 42,  41,  47,  58,  59,  59,  59,  65,  66,  65,  81,  80,  74],
[ 29,  34,  41,  45,  42,  37,  34,  39,  41,  50,  76,  85,  88],
[ 75,  89,  98, 101,  94,  81,  80,  72,  70,  68,  81,  85,  86],
[ 67,  71,  86, 102, 122, 144, 164, 181, 186, 192, 216, 238, 249],
[  8,   8,   8,   7,  15,  17,  18,  23,  29,  28,  31,  32,  36],
[ 73,  79,  67,  70,  80,  81,  95,  97,  88,  94, 100, 106, 100],
[ 48,  77,  73,  66,  68,  74,  83,  83,  77,  75,  67,  69,  68],
[  7,   5,   6,  21,  20,  23,  26,  26,  25,  27,  33,  34,  40],
[ 75,  60,  44,  33,  34,  33,  40,  42,  40,  40,  52,  55,  56],
[ 63,  62,  94,  77,  83,  74,  68,  48,  22,  24,  36,  43,  54],
[110, 117, 151, 107,  97,  98,  89,  78,  79,  71,  63,  45,  47]])
```

1.2.1 Normalisation des données

Les propriétés géométrique d'un nuage de point sont préservées si l'on ajoute un même vecteur à tous les points du nuage. On va donc commencer par retirer à chaque colonne sa moyenne sur les pays. Cela revient, pour chaque année, à centrer en 0 la distribution de la dette pour les pays.

1. Ecrire une fonction `moy(M)` qui, donnée une matrice

$$M = \begin{pmatrix} m_{0,0} & \dots & m_{0,m-1} \\ \dots & \dots & \dots \\ m_{n-1,0} & \dots & m_{n-1,m-1} \end{pmatrix} \in \mathbb{R}^{n \times m}$$

   renvoie le vecteur $u \in \mathbb{R}^m$ dont la j-ieme coordonnée est la moyenne de la j-ieme colonne de M, c'est-à-dire

$$u_j = \frac{1}{n} \sum_{i=0}^{n-1} m_{i,j}$$

2. Ecrire une fonction `eca(M)` qui renvoie le vecteur $v \in \mathbb{R}^m$ dont la j-ieme coordonnée est l'ecart-type de la j-ieme colonne de M:

$$v_j = \sqrt{\frac{1}{n} \sum_{i=0}^{n} (m_{i,j} - u_j)^2}$$

   où u est définit dans la question précédente.

3. Ecrire une fonction `nor(M)` qui renvoie la matrice N correspondant à la normalisation de la matrice M:

$$N_{i,j} = \frac{M_{i,j} - u_j}{v_j}$$

   où u et v sont définis en 1. et 2. S'en servir pour obtenir la matrice normalisée de la matrice

des températures `N=nor(T)`. Indice, la première colonne doit être

$$\begin{pmatrix} 0.34987766 \\ -0.64451148 \\ -0.42353612 \\ -0.38670689 \\ -0.86548685 \\ 0.82865762 \\ 0.5340238 \\ -1.63890062 \\ 0.75499916 \\ -0.16573152 \\ -1.67572985 \\ 0.82865762 \\ 0.38670689 \\ 2.11768058 \end{pmatrix}$$

1.2.2 Calcul des deux premiers facteurs

On note $(N_{i,j})_{0\leq i\leq 13, 0\leq j\leq 12}$ les entrées de la matrice N, donc $N_{1,2}$ représente la dette normalisée de la France en 1994. Le i-ieme pays a donc $\tilde{x}_i = (N_{i,j})_{0\leq j\leq 12}$ comme distribution de dette normalisée au cours du temps. On s'intéresse maintenant aux propriétés géométriques du nuage de points normalisé $\{\tilde{x}_i\}_{0\leq i\leq 13}$.

Soit $u \in \mathbb{R}^{13}$ un vecteur colonne unitaire. La norme de la projection de $\tilde{x}_i$ sur u est $|\tilde{x}_i u|$. Plus elle est grande, plus $\tilde{x}_i$ est bien approximé par sa projection sur $Vect(u)$. De manière analogue, plus la quantité

$$\sum_{0\leq i\leq 13} |\tilde{x}_i u|^2 = u^T N^T N u$$

est grande, plus le nuage de points $\{\tilde{x}_i\}_{0\leq i\leq 13}$ est bien approximé par le nuage de points de ses projections sur $Vect(u)$, qui est $\{(\tilde{x}_i u)u^T\}_{0\leq i\leq 13}$. Le meilleur vecteur u est donc solution du problème d'optimisation suivant :

$$(*) \qquad \text{Trouver } u \in \mathbb{R}^{13} \text{ avec} \quad |u| = 1 \text{ et} \quad u^T N^T N u = \max_{u'\in\mathbb{R}^{13},\ |u'|=1} u'^T N^T N u'$$

où $|u|$ désigne la norme Euclidienne de u.

4. Rappeler pourquoi $N^T N$ admet 13 vecteurs propres $(v_1, ..., v_{13})$ qui forment une base orthonormale de $\mathbb{R}^{13}$, associés à des valeurs propres positives ou nulles que l'on range dans l'ordre décroissant
$$0 \leq \lambda_{13} \leq \lambda_{12} \leq ... \leq \lambda_2 \leq \lambda_1.$$
Montrer ensuite à la main que le choix $u = v_1$ résout le problème de minimisation $(*)$.

5. Utiliser `numpy.linalg.eigh` pour obtenir v_1.

v_1 est appelé le premier facteur dans l'analyse factorielle du nuage de points, et v_2 est appelé le second facteur. Un résultat que l'on ne démontrera pas, analogue à la question 4., est que Vect(v_1,v_2) est le plan qui parmi les plans de $\mathbb{R}^{13}$ permet de bien approcher le nuage de point $\{\tilde{x}_i\}_{0\leq i\leq 13}$.

6. Calculer v_2 en utilisant `numpy.linalg.eigh`.

1.2.3 Analyse : représentation des pays dans le premier plan factoriel

7. Calculer, pour chaque pays, les composantes de sa projection le long des vecteurs v_0 et v_1, qui est $(\tilde{x}_i v_1, \tilde{x}_i v_2) \in \mathbb{R}^2$. C'est-à-dire, par exemple, que l'Espagne est représenté par le point $(\tilde{x}_2 v_1, \tilde{x}_2 v_2)$. Indice, les Etats Unis sont représentés par le point $(-0.3979777, -0.01505147)$.

Placer les villes sur un même graphique, où chaque ville est représentée par son nom écrit au point dont les coordonnées sont les composantes de sa projection calculées ci-dessus. On utilisera `pyplot.text` pour placer les noms.

8. Interpréter le vecteur propre v_1 : que représente-t-il ? Interpréter également le vecteur propre v_2.

[ ]: