Informatique tronc commun
Devoir n° 4 – Partie rédigée

25 mai 2019

Durée : 60 minutes, documents interdits.

Vous écrirez les fonctions demandées dans le langage `Python` (version 3) et rendrez sur papier votre composition.

Vous rédigerez soigneusement la ou les fonctions `Python` que vous devrez écrire, sans oublier les indentations, chaînes de documentation et commentaires nécessaires.

On numérotera chaque ligne de chaque bloc de code `Python`.

Lorsque vous écrivez une fonction `Python`, on ne vous demande pas de vérifier que les arguments donnés sont corrects. Cependant, il sera apprécié d'indiquer les préconditions vérifiées par ces arguments dans la chaîne de documentation de la fonction.

Les fonctions `Python` permettant de faire un calcul sur les listes (ex : `max`, `sum`, *etc.*) ne sont pas autorisées, hormis la fonction `len`.

I. Questions de cours.

Q1 Donner la complexité de l'aglorithme du pivot de Gauss permettant de résoudre un système linéaire d'équations de taille n. Justifier brièvement votre résultat en précisant la complexité de chaque étape de l'algorithme.

II. Représentation CSR des matrices creuses et produit par un vecteur.

Beaucoup de problèmes technologiques actuels mettent en jeu des données de grandes dimensions et font souvent intervenir de grandes matrices.

Beaucoup de ces matrices sont *creuses*, c'est-à-dire qu'elles ne contiennent que peu de valeurs non nulles. Nous allons développer une méthode de codage de telles matrices et étudier leur intérêt.

Dans tout ce problème, on s'interdira d'utiliser les opérations entre vecteurs et matrices `numpy` (méthode `.dot()` par exemple).

Dans tout ce problème, on utilisera le type `array` de `numpy` pour représenter des vecteurs. On pourra supposer que la ligne suivante a été écrite :

```
from numpy import array, zeros
```

Dans tout ce problème, on veillera à l'optimalité des fonctions écrites en termes de complexité temporelle asymptotique. Les réponses clairement sous-optimales seront pénalisées.

Soit une matrice réelle $A \in \mathcal{M}_{n,p}(\mathbb{R})$ de dimension $n \times p$ et possédant s coefficients non nuls. Ce coefficient s est appelé *niveau de remplissage* de la matrice A. Comme toujours en `Python`, on numérotera les lignes et les colonnes en partant de 0.

Une telle matrice se code usuellement comme un tableau de nombres de dimension $n \times p$.

Le codage CSR (Compress Sparse Raw) code la matrice A par trois listes V, L et C définies comme suit.

— La liste V contient toutes les valeurs des coefficients non nuls de A, en les listant ligne par ligne (de la première à la dernière ligne puis de la première à la dernière colonne). Ainsi, V est de longueur s.

— La liste L est de longueur $n+1$, L_0 vaut toujours 0 et pour chaque $1 \leq i \leq n$, L_i vaut le nombre de coefficients non nuls dans les i premières lignes de A (*i.e.* de la ligne d'indice 0 à celle d'indice $i-1$, inclu).

— La liste C contient les numéros de colonne de chaque coefficients non nuls de A, en les listant ligne par ligne (de la première à la dernière ligne puis de la première à la dernière colonne). Ainsi, C est de longueur s.

Par exemple, avec

$$A = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 2 & 0 & 4 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix},$$

on a $n = 3$, $p = 4$, $s = 4$ et sa représentation CSR est donnée par les trois listes

$$\begin{aligned} V &= [-1, 2, 4, -1], \\ L &= [0, 1, 3, 4], \\ C &= [1, 0, 2, 3]. \end{aligned}$$

On remarquera qu'ajouter une colonne nulle à droite de A ne change pas sa représentation CSR.

Q2 Déterminer la représentation CSR de la matrice

$$A = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Q3 Donner une matrice dont la représentation CSR est

$$V = [2, -3, 1, 1],$$
$$L = [0, 0, 2, 4],$$
$$C = [0, 3, 2, 3].$$

Q4 Soit (V, L, C) la représentation CSR de A, soit $0 \leq i \leq n-1$. Combien y a-t-il de coefficients non nuls sur la ligne n$^{\text{o}}$ i de A ? Donner deux expressions `Python` (en fonction de V, L, C et i) permettant d'obtenir respectivement la liste des valeurs de ces coefficients et la liste des indices colonnes de ces coefficients.

On rappelle la formule du produit matriciel : pour une matrice $A \in \mathcal{M}_{n,p}(\mathbb{R})$ de coefficients $a_{i,j}$ et un vecteur $X \in \mathcal{M}_{p,1}(\mathbb{R})$ de coefficients x_j, si $0 \leq i \leq n-1$, alors la i^{e} coordonnée de AX est

$$\sum_{j=0}^{p-1} a_{i,j} x_j.$$

Q5 Écrire une fonction `coeff_prod(V,L,C,X,i)` renvoyant la i^{e} coordonnée du produit AX, où le triplet (V, L, C) est la représentation CSR de A. On supposera que les dimension de A et X sont compatibles pour effectuer le produit AX.

Donner la complexité asymptotique de cette fonction, en fonction de n, p et ℓ_i, où ℓ_i est le nombre d'éléments non nuls sur la i^{e} ligne de A.

Q6 Écrire une fonction `prod(V,L,C,X)` renvoyant le produit AX, où le triplet (V, L, C) est la représentation CSR de A. On supposera que les dimension de A et X sont compatibles pour effectuer le produit AX.

Donner la complexité asymptotique de cette fonction, en fonction de n, p et s.

Q7 Écrire une fonction `prod_naif(A,X)` renvoyant le produit AX, où A est codée usuellement comme un tableau à double dimension.

Donner la complexité asymptotique de cette fonction, en fonction de n, et p.

Q8 En les comparant, discuter des deux complexités précédentes, notamment en fonction du niveau de remplissage s.

On prend maintenant pour exemple celui de la dérivation discrète, typique en traitement du signal. Pour un vecteur de longueur n

$$X = \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

on définit la dérivée discrète de X comme le vecteur Y de longueur n définit par :

$$y_0 = x_1 - x_0,\ y_{n-1} = x_{n-1} - x_{n-2} \text{ et } \forall i \in [\![1, n-1[\![,\ y_i = \frac{1}{2}(x_{i+1} - x_{i-1}).$$

Q9 Déterminer une matrice A vérifiant $Y = AX$.

Q10 Écrire une fonction `CSR_A(n)` renvoyant les trois vecteurs `V`, `L` et `C` codant A au format CSR.

Indication : on pourra écrire une fonction pour la création de chaque vecteur