NumPy pour l’algèbre linéaire

Maîtriser NumPy : Le compagnon indispensable pour les Data Lovers · https://moncoachdata.com/tutos/numpy-pour-lalgebre-lineaire/

NumPy dispose d'un large éventail de fonctions pour l'**algèbre linéaire**, regroupées dans le module `np.linalg`. Après avoir créé des vecteurs et des matrices et effectué des opérations de base, voici des opérations matricielles plus complexes.

Opérations matricielles avancées

- **Inversion de matrice :** `np.linalg.inv()` calcule l'inverse d'une matrice carrée inversible ;
- **Valeurs propres et vecteurs propres :** `np.linalg.eig()` ;
- **Normes et distances :** `np.linalg.norm()` calcule des normes (par exemple la norme de Frobenius pour une matrice), et peut servir à mesurer des distances entre vecteurs ou matrices ;
- **Rang d'une matrice :** `np.linalg.matrix_rank()`.



```python
import numpy as np

A = np.array([[2, 3],
              [4, 5]])

# Inverse de la matrice
A_inv = np.linalg.inv(A)
print(A_inv)

# Vérification : A @ A_inv donne (à l'arrondi près) la matrice identité
print(np.round(A @ A_inv, 10))

# Valeurs propres et vecteurs propres
valeurs_propres, vecteurs_propres = np.linalg.eig(A)
print(np.round(valeurs_propres.real, 4))
print(np.round(vecteurs_propres.real, 4))

# Norme de Frobenius de la matrice
print(np.linalg.norm(A))

# Distance entre deux vecteurs
u = np.array([1, 2, 3])
v = np.array([4, 6, 3])
print(np.linalg.norm(u - v))

# Rang de la matrice
print(np.linalg.matrix_rank(A))
```

Python

```
[[-2.5  1.5]
 [ 2.  -1. ]]
[[1. 0.]
 [0. 1.]]
[-0.2749  7.2749]
[[-0.7968 -0.4944]
 [ 0.6042 -0.8693]]
7.3484692283495345
5.0
2
```

Résoudre des équations linéaires

Tu peux aussi résoudre un système d'équations linéaires avec `np.linalg.solve()`. Ici, on résout Ax = b pour trouver x :

```python
A = np.array([[2, 3], [4, 5]])
b = np.array([6, 7])

# Résoudre Ax = b pour x
x = np.linalg.solve(A, b)
print(x)

# Vérification : A @ x doit redonner b
print(A @ x)
```

Python

```
[-4.5  5. ]
[6. 7.]
```

Préfère `np.linalg.solve(A, b)` au calcul de `np.linalg.inv(A) @ b` : il est plus rapide et numériquement plus stable.

La suite de la série porte sur les techniques avancées pour NumPy. Documentation : algèbre linéaire avec NumPy.

Pour aller plus loin, consulte aussi notre guide des 45 méthodes NumPy.

L'algèbre linéaire est la base du Machine Learning. Sur DataXp, tu passes de NumPy à pandas puis au Machine Learning, avec des exercices que tu exécutes dans ton navigateur.


NUMPY