CIPA4

TP3 : Apprentissage non-supervisé

- Réduction de la dimensionnalité -

Khadidja OULD AMER (Khadidja.ouldamer@isen-ouest.yncrea.fr)

Objectifs du TP

- Assimiler le principe de la réduction de la dimensionnalité
- L'interprétation des résultats (et non pas juste leur affichage)
- Prise en main des bibliothèques de la partie Liens utiles et savoir utiliser, idéalement, leurs documentations

Liens utiles

- Pandas
- Matplotlib
- Scikit-learn
- Numpy

I- Affichage des données MNIST

1. Depuis votre Google Drive, créez un notebook, sur GoogleColab, nommé tp3_IA
2. Créez une section intitulée I- Affichage des données MNIST
3. L'objectif de cette partie est de pouvoir afficher les instances de la base de données MNIST utilisée dans le TP2. Pour ce faire, vous allez tester deux méthodes : PCA et t-SNE.
   L'affichage de toutes les instances va être chronophage, par conséquent nous allons afficher que 10000 instances choisies aléatoirement. Pour réaliser ceci, utilisez le code suivant (avec mnist est la variable qui contient la base de données MNIST) :

```
from sklearn.datasets import fetch_openml
```

```
import numpy as np

mnist = fetch_openml('mnist_784', version=1, as_frame=False)
mnist["target"] = mnist["target"].astype(np.uint8)

np.random.seed(42)
m = 10000
idx = np.random.permutation(60000)[:m]
X = mnist['data'][idx]
y = mnist['target'][idx]
```

1- Utilisation de la méthode PCA

4. Créez un titre intitulé 1- Utilisation de la méthode PCA
5. Afin de réduire la dimensionnalité des données, appliquez la méthode PCA sur les données (X) en choisissant un nombre de composants de 2. Pensez à utiliser la fonction fit_transform (et non pas fit) de scikit-learn pour stocker les données réduites dans une variable nommée X_pca_reduced.
6. Vérifiez si la nouvelle dimension des données réduites est bien 2.
7. Affichez les données réduites (les données après l'application de PCA) comme suit :

```
import matplotlib.pyplot as plt

plt.figure(figsize=(13,10))
plt.scatter(X_pca_reduced[:, 0], X_pca_reduced[:, 1], c=y)
plt.axis('off')
plt.colorbar()
```

8. Le résultat d'affichage permet de donner une idée claire sur la distribution des instances de MNIST ?

2- Utilisation de la méthode t-SNE

9. Créez un titre intitulé 2- Utilisation de la méthode t-SNE
10. Appliquez la méthode t-SNE sur les données (X) en choisissant un nombre de composant de 2
11. Affichez les données réduites en utilisant le code de la question 7
12. Existe-t-il des chevauchements entre quelques chiffres ? si oui lesquels ?
13. Comparez les résultats d'affichage de la méthode PCA et t-SNE.

II- PCA sur les données MNIST

14. Créez une section intitulée II- PCA sur les données MNIST

1- Résultats de RandomForest SANS la réduction de la dimensionnalité des données

15. Créez une section intitulée 1- Résultats de RandomForest SANS la réduction de la dimensionnalité des données
16. Divisez la base de données MNIST en base d'apprentissage et base de test comme suit :

    ```
    X_train = mnist['data'][:60000]
    y_train = mnist['target'][:60000]
    X_test = mnist['data'][60000:]
    y_test = mnist['target'][60000:]
    ```

17. Appliquez la méthode de classification RandomForest sur les données d'apprentissage tout en calculant le temps d'exécution nécessaire pour l'apprentissage. Pour ce deuxième objectif, calculez :

    a. avant l'apprentissage du modèle, le nombre de secondes passé depuis le 01/01/1970 en utilisant la fonction "time()" du module time
    b. après l'apprentissage du modèle, le nombre de secondes passé depuis le 01/01/1970 en utilisant la fonction "time()" du module time
    c. le temps d'exécution nécessaire pour l'apprentissage an appliquant la différence des résultats de la question 2.a et 2.b. Pensez à n'afficher que 2 chiffres après la virgule en utilisant la méthode "format()" (ci-dessous un exemple d'utilisation) :

    ```
    format(math.pi, '.2f')   # 3.14
    ```

    d. Évaluez le modèle d'apprentissage sur la base de test en affichant le taux de classification. Pour ce faire :

        i. prédisez les labels de la base de test avec la fonction "predict"
        ii. utilisez la fonction accuracy_score du sous-module metrics du module sklearn tout en donnant en argument les labels réels de la base de test et les labels prédits

2- Résultats de RandomForest AVEC la réduction de la dimensionnalité des données

18. Créez une section intitulée 2- Résultats de RandomForest AVEC la réduction de la dimensionnalité des données
19. Appliquez la méthode PCA sur la base d'apprentissage avec une variance ratio de 95%. Cela va permettre de définir le nombre minimum de dimensions requises pour préserver 95% de la variance de l'ensemble d'apprentissage.

20. Appliquez à nouveau la méthode de classification RandomForest sur les données d'apprentissage réduites (après l'application du PCA) tout en calculant le temps d'exécution nécessaire pour l'apprentissage (suivez les étapes de la question 16). Le temps d'apprentissage est plus rapide que celui du II-1 ? C'est le résultat attendu ?
21. Appliquez la méthode PCA sur la base de test avec un variance ratio de 95% (utilisez la fonction transform et non pas fit_trannsform sur l'objet instancié dans la question 19 pour avoir la même dimension des données d'apprentissage réduites de la question 19).
22. Évaluez le modèle d'apprentissage sur la base de test en affichant le taux de classification. Comparez le résultat avec celui de la II-1.
23. L'application du PCA sur les données MNIST était fructueuse pour le temps d'apprentissage et le taux de classification dans le cas de RandomForest ?

3- Résultats de Softmax SANS la réduction de la dimensionnalité des données

24. Créez une section intitulée 3- Résultats de Softmax SANS la réduction de la dimensionnalité des données
25. Appliquez la méthode de classification LogisticRegression sur les données d'apprentissage tout en calculant le temps d'exécution nécessaire pour l'apprentissage.
26. Évaluez le modèle d'apprentissage sur la base de test en affichant le taux de classification

4- Résultats de Softmax AVEC la réduction de la dimensionnalité des données

27. Créez une section intitulée 4- Résultats de Softmax AVEC la réduction de la dimensionnalité des données
28. Appliquez à nouveau la méthode de classification LogisticRegression sur les données d'apprentissage réduites (après l'application du PCA) tout en calculant le temps d'exécution nécessaire pour l'apprentissage. Le temps d'apprentissage est plus rapide que celui du II-3 ?
29. Appliquez la méthode PCA sur la base de test avec un variance ratio de 95% (utilisez la fonction transform et non pas fit_trannsform)
30. Évaluez le modèle d'apprentissage sur la base de test en affichant le taux de classification. Comparez le résultat avec celui de la II-3.
31. L'application du PCA sur les données MNIST était fructueuse pour le temps d'apprentissage et le taux de classification dans le cas de Softmax ?
32. L'application du PCA sur les données contribue toujours à accélérer le temps de calcul du modèle d'apprentissage ?