M1 MIAGE

TD 1 – Apprentissage non supervisé

Objectifs :

- Utiliser les algorithmes de clustering proposés par scikit learn sur différents datasets
- Utiliser les algorithmes de réduction de dimensionnalité proposés par scikit learn sur différents datasets
- Evaluer l'efficacité de ces algorithmes
- Notion de recherche paramétrique

Exercice 1 : Clustering - Prédiction des nombres

Dans cet exercice, on souhaite utiliser un algorithme de clustering pour reconnaitre des nombres écrits à main levé. Pour cela, un dataset labélisé existe déjà et vous permettra ainsi d'évaluer simplement la qualité de votre clustering.

1/ En utilisant la fonction load_digit de sklearn, charger le dataset et afficher la forme (shape) de celui-ci

2/ En utilisant la fonction imshow de matplotlib, afficher quelques samples du dataset chargé. Le titre doit contenir le nombre correspondant à l'image chargée.

3/ En utilisant un des algorithmes vus en cours (KMeans), effectuer un clustering du dataset. Quel nombre de cluster doit-on utiliser dans ce cas précis ?

4/ Afficher quelques exemples d'images avec leur label prédit.

5/ Afficher les centres des clusters trouvés

Exercice 2 : Clustering - Recherche du paramètre k optimal

1/ En partant du code de l'exercice 1, écrire une fonction permettant de calculer la moyenne des inerties des clusters générés.

2/ Généraliser cette fonction en un code permettant de générer le graphe ci-dessous (figure 1) : il affiche l'inertie du clustering en fonction du nombre de clusters passé en paramètre de KMeans.

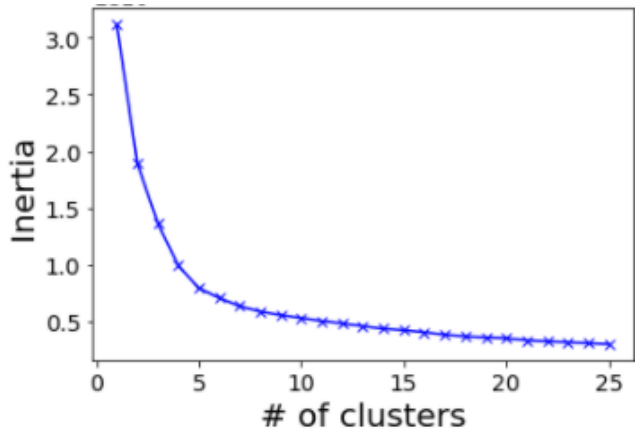

Figure 1. *La meilleure valeur du paramètre k est obtenu en observant le point à partir duquel l'inertie ne diminue plus drastiquement lorsque la valeur de k augment. Sur le graphe précédent, cette valeur est par exemple k=5. Il n'est pas toujours facile de déterminer précisément ce nombre mais cette méthode permet d'en avoir une bonne estimation.*

3/ Faire le même travail que dans la question 2 en calculant cette fois ci la distorsion des clusters.

4/ Utiliser les algorithmes précédemment développés pour trouver la meilleure valeur du paramètre k pour les datasets suivants :

4.a/ Sur un dataset artificiel généré avec des blobs

4.b/ Sur le dataset IRIS (voir la fonction load_iris pour cela et explorer le dataset)

4.c/ Sur le dataset Experience and Salary téléchargé depuis Kaggle

Exercice 3 : Reduction de dimentionalité - Visualisation du dataset IRIS avec PCA

Dans cet exercice, l'objectif est d'apprendre à utiliser la PCA sur un exemple simple.

1. Charger le dataset IRIS.
2. Afficher et analyser le dataset les premières lignes du dataset. Que représente-t-il ?
3. Afficher les variables par paire (longueur des sépales vs largeur des pétales, longueur des pétales vs largeur des sépales, etc.). Chaque point doit être coloré par la classe correspondant au type de fleur.
4. Afficher un pairplot. Que représente-t-il ?
5. Calculer et afficher la matrice de corrélation de ce dataset. Que représente-t-elle ?
6. Appliquer une PCA vers 2 dimensions au dataset
7. Afficher le résultat de la PCA. Que pensez-vous de la capacité à identifier les types de fleurs ?

Exercice 4 : Reduction de dimentionalité - Visualisation du dataset avec PCA et t-SNE

Dans cet exercice, nous allons comparer différents algorithmes de réduction de dimensionnalité, linéaire ou non.

1. Charger le dataset wine. Explorer le dataset. Que représente-t-il ?
2. Afficher les pairplot de ce dataset. Quel est votre analyse ?
3. Afficher la matrice de corrélation de ce dataset et l'analyser.
4. Appliquer une PCA vers 2 dimensions aux datasets.
5. Afficher le résultat de la PCA. Que pensez-vous de la capacité des algorithmes linéaires à prédire les classes ?
6. Appliquer une t-SNE vers 2 dimensions sur le dataset en faisant varier la perplexité
7. Afficher le résultat de la t-SNE. Que pensez-vous de la capacité des algorithmes non-linéaires à prédire les classes ?
8. Tester l'algorithme UMap.
9. Refaire les questions de 1 à 6 avec le dataset breast_cancer.

Exercice 5 : Clustering - Autres méthodes de clustering

1/ En utilisant la fonction make_classification, générer un dataset artificiel contenant 2 ou 3 caractéristiques (pour que le dataset soit affichable sur un graphe), 2 classes avec 1 cluster par classe et 1000 exemples.

2/ Explorer, analyser et tester les algorithmes de clustering suivants

2.a/ KMeans

2.b/ DBScan (Density-Based Spatial Clustering of Application and Noise)

2.c/ GMM (Gaussian Mixture Model)

Exercice 6 : Reduction de dimentionalité - Débruitage de données avec PCA

Dans cet exercice, nous allons voir comment utiliser une PCA pour réduire le bruit dans des données, ici une image afin que ce soit visuel.

1. Charger le dataset *MNist*
2. Afficher quelques images de ce dataset.
3. En utilisant la fonction *random.normal()* de numpy, appliquer à chaque pixel de chaque image du dataset un bruit gaussien. Garder ce nouveau dataset dans une nouvelle variable et afficher quelques images.
4. En faisant varier le nombre de dimension de l'espace de projection :
    a. Appliquer une PCA sur le dataset bruité.
    b. En utilisant la fonction *inverse_transform()* de la PCA, revenir à l'image d'origine.

Visualiser le résultat obtenu.