

Visualisation de données

Contrôle de Travaux pratiques

(le code python associé à ce sujet est disponible
dans votre environnement de travail)

Mise en œuvre de la méthode *Locally Linear Embedding (LLE)*

Au cours de cet exercice vous allez programmer la méthode LLE en complétant le programme my_LLE.py qui est fourni.

On rappelle que la méthode se décompose en 3 étapes

Etape 1 : détermination du graphe des plus proches voisins

Etape 2 : calcul des vecteurs de poids des voisins, réalisée par la méthode LLE_weight(). En pratique pour éviter les problèmes de valeur singulière on calcule les poids

Etape 3 : calcul des points projetés

➤ Dans la méthode LLE_weight() on vous demande d'insérer à l'endroit indiqué, le calcul de la matrice de Gram G_i associée à chaque exemple, puis de calculer les valeurs propres et vecteurs de poids. Pour éviter les problèmes d'instabilité numérique on programmera les formules suivantes

$$w_i = \frac{(G_i + \alpha I_k)^{-1} \mathbf{1}_k}{\mathbf{1}^T (G_i + \alpha I_k)^{-1} \mathbf{1}_k}$$

avec α un réel de faible valeur (10^{-3} par exemple)

➤ Comparer vos résultats à ceux de la méthode de scikitlearn pour les deux dataset IRIS et MNIST

➤ faire vérifier vos résultats par l'enseignant

➤ sauvegarder votre programme dans votre environnement de travail

Mise en œuvre de la méthode Out of Sample *Locally Linear Embedding (OS_LLE)*

Dans le code fourni, la méthode OS_LLE() calcule l'embedding du point passé en argument.

- ↳ On vous demande de programmer l'étape 2 qui calcule le vecteur W des poids des k plus proches voisins.
- ↳ *faire vérifier vos résultats par l'enseignant*
- ↳ *sauvegarder votre programme dans votre environnement de travail*