

Local Sensitive Discriminant Analysis

Sommaire

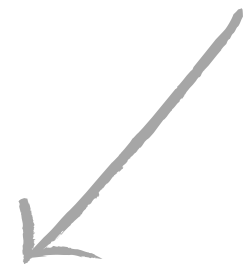
- 1. Présentation générale
- 2. Théorie
- 3. Implémentation

Présentation générale

La LSDA est une **version modifiée de la LDA** qui permet de prendre en compte la **structure locale** des données.

Présentation générale

La LSDA est une **version modifiée de la LDA** qui permet de prendre en compte la **structure locale** des données.



Utile lorsque peu
de données
disponibles



Basé sur une méthode
de réduction de
dimension usuelle



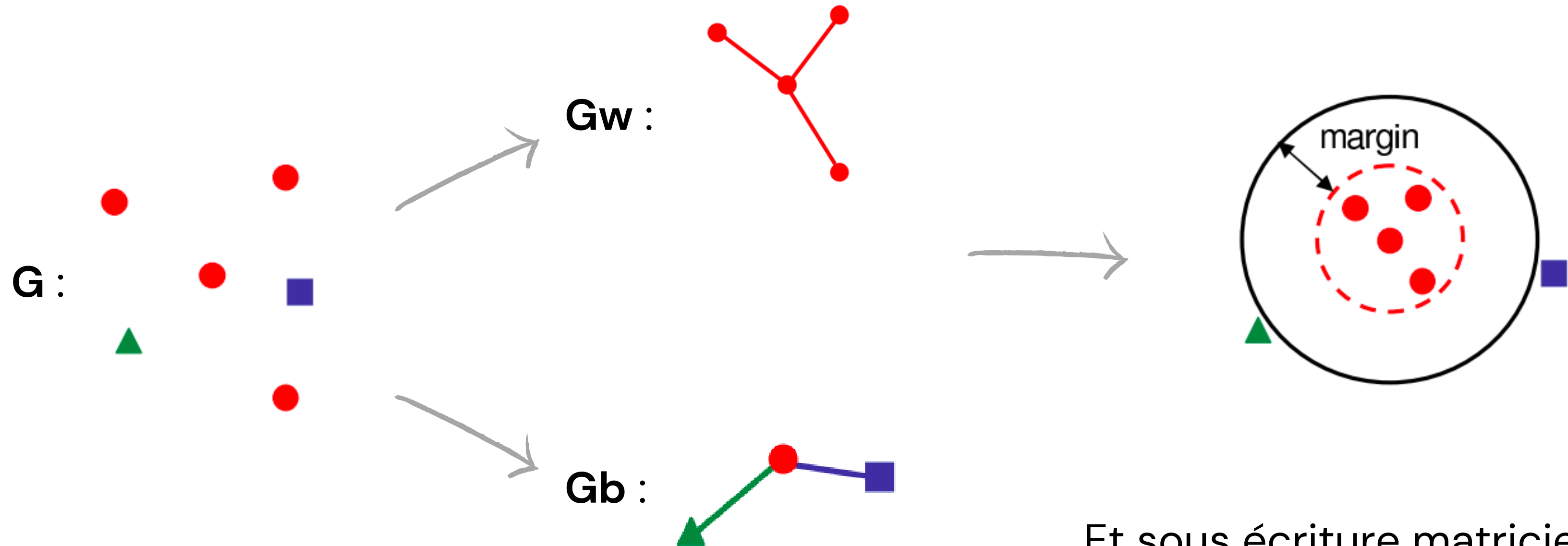
Facile à mettre en oeuvre

Théorie

Objectif : Trouver une projection qui maximise la variance entre les individus de classes différentes dans chaque voisinage local.

Comment ? Via une structure de graphe.

Théorie



Théorie

On définit :

- la matrice $X = (\mathbf{x}_1, \dots, \mathbf{x}_m)$ des individus
- l'ensemble $N(\mathbf{x}_i) = \{\mathbf{x}_i^1, \dots, \mathbf{x}_i^k\}$ des k plus proches voisins de \mathbf{x}_i
- l'ensemble $N_w(\mathbf{x}_i)$: les individus de $N(\mathbf{x}_i)$ de même label que \mathbf{x}_i
- l'ensemble $N_b(\mathbf{x}_i)$: les individus de $N(\mathbf{x}_i)$ de label différent de \mathbf{x}_i

On a alors :

$$\mathbf{G} : W_{i,j} = \begin{cases} 1 & \text{si } \mathbf{x}_i \in N(\mathbf{x}_j) \text{ ou } \mathbf{x}_j \in N(\mathbf{x}_i) \\ 0 & \text{sinon} \end{cases}$$



$$\mathbf{G}_w : W_{b,ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N_b(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_b(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathbf{G}_b : W_{w,ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N_w(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_w(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases}$$

Théorie

On souhaite obtenir une représentation où les points de **même label** sont **proches** et où les points de **labels différents** sont **éloignés**.

Ainsi on choisit d'optimiser les fonctions suivantes :

$$\min \sum_{ij} (y_i - y_j)^2 W_{w,ij}$$

$$\max \sum_{ij} (y_i - y_j)^2 W_{b,ij}$$

Théorie

Ce problème d'optimisation se réécrit sous la forme :

$$\arg \max_{\substack{\mathbf{a} \\ \mathbf{a}^T X D_w X^T \mathbf{a} = 1}} \mathbf{a}^T X (\alpha L_b + (1 - \alpha) W_w) X^T \mathbf{a} \quad \text{avec} \quad \begin{cases} 0 \leq \alpha \leq 1 \\ D_{w,ii} = \sum_j W_{w,ij} \\ D_{b,ii} = \sum_j W_{b,ij} \\ L_b = D_b - W_b \end{cases}$$

... et est minimisé par la plus plus grande valeur propre λ solution de l'équation :

$$X (\alpha L_b + (1 - \alpha) W_w) X^T \mathbf{a} = \lambda X D_w X^T \mathbf{a} \quad (1)$$

Théorie

On obtient la projection souhaitée en effectuant la multiplication matricielle

$$X_{projected} = XA$$

où $A = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_d)$ est la matrice des d vecteurs propres associés aux d plus grandes valeurs propres $\lambda_1 > \dots > \lambda_d$ solutions de l'équation (1).

$X_{projected}$ est alors de taille $m \times d$.

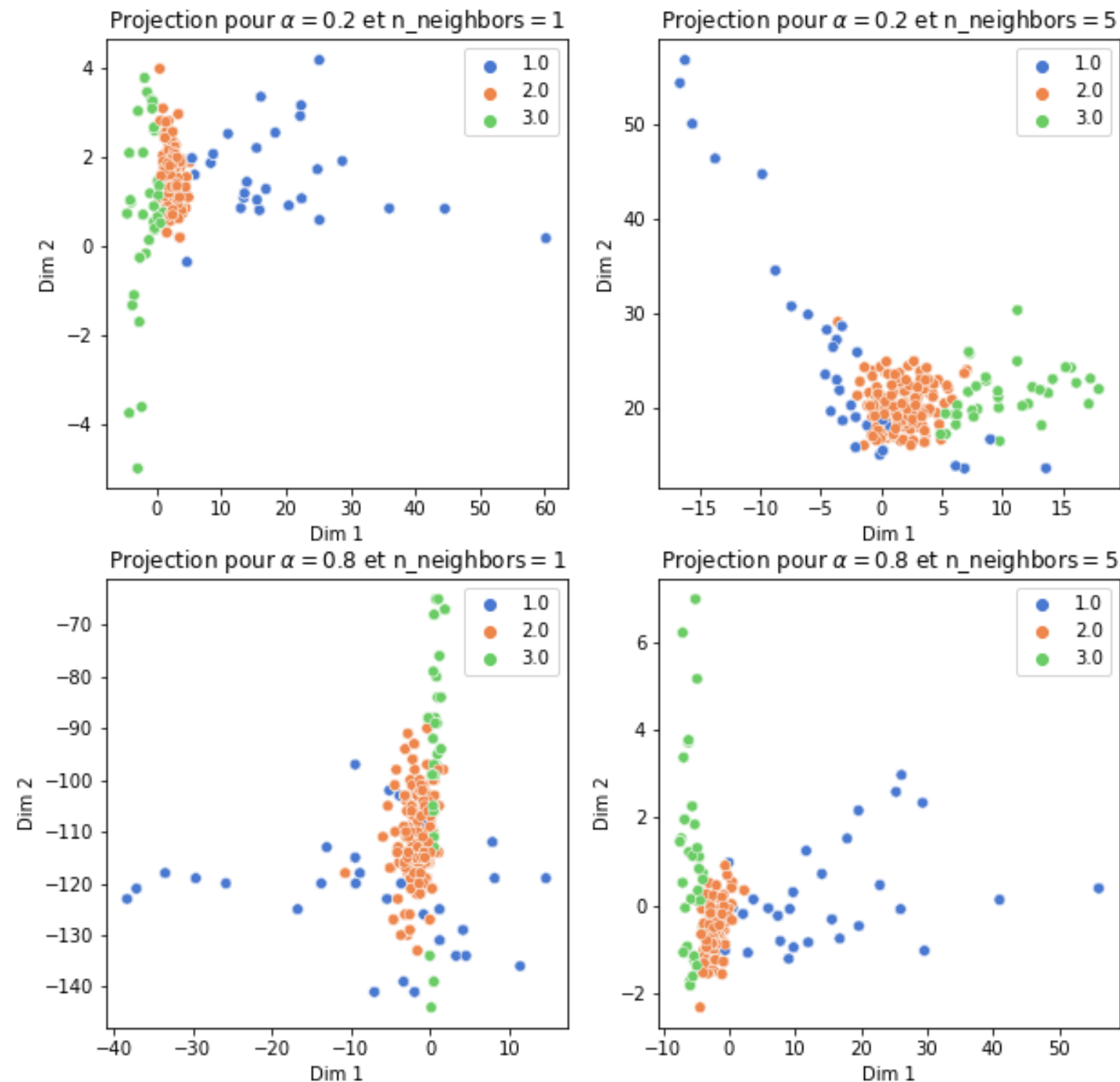
Implémentation

```
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from lsda import LSDA

X_scaled = StandardScaler().fit_transform(X, y)
model = LSDA(n_neighbors=5, alpha=0.5, n_components=2)
X_projected = model.fit_transform(X_scaled, y)
plt.plot(X_projected[:,0], X_projected[:,1], c=y)
```

Note : voir lsda.py pour le code de la méthode

Exemple de validation croisée



Jeu de données : **newthyroid**

- Nb. d'individus
 - Entrainement : 4830
 - Test : 1620
- Nb. de variables : 5
- Nb. de classes : 3
- Projeté sur un espace de dimension 2
- Temps d'exécution : ~7 s par passe de CV

Remarques :

- l'ordinalité des classes est conservée
- bonne séparation des classes pour $\alpha=0.8$ et $n_neighbors=5$

Quelques commentaires

Il est possible :

- d'appliquer une ACP pour éliminer les valeurs propres nulles lorsque il y a plus de variables que d'individus.
- d'utiliser l'astuce du noyau si les variétés sous-jacentes sont non linéaires.
- de normaliser les données pour obtenir de meilleurs résultats.
- de faire de la classification avec LSDA.

Références

1. D. Cai, X. He, K. Zhou, J. Han, and H. Bao, "Locality sensitive discriminant analysis," in International Joint Conference on Artificial Intelligence (IJCAI'07), 2007

Local Sensitive Discriminant Analysis