# Géométrie des Données et Apprentissage Machine

## Module 2 – Graphe et Diffusion sur Données

Youssef MESRI - MINES Paris - PSL

November 23, 2025

# Plan

# Module 2 : plan

- Construction de graphes (k-NN, $\varepsilon$-graph, pondérés)
- Laplacien de graphe : définitions et interprétations
- Diffusion sur graphe, noyau de chaleur
- Équation de Poisson discrète : apprentissage semi-supervisé
- Courbure de graphe et topologie intuitive
- Exemples : classification digits, segmentation
- Applications modernes : NLP, bioinformatique
- TP: Classification semi-supervisée sur graphes (ex : digits).

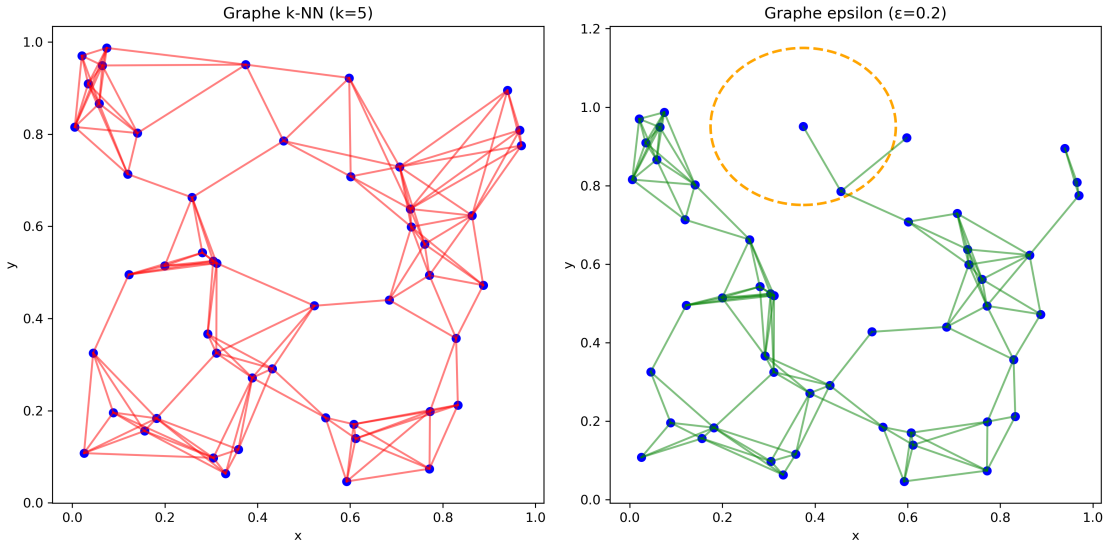
# Construction de graphes sur données

- Objectif : représenter les relations locales entre points d'un nuage de données.
- Méthodes courantes :
  1. **k-NN graph** : chaque point est relié à ses $k$ plus proches voisins.
  2. $\varepsilon$-**graph** : relier tous les points dont la distance est inférieure à $\varepsilon$.
- Graphes pondérés : assigner un poids de similarité à chaque arête

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

# Exemple : k-NN vs $\varepsilon$-graph

- k-NN : garantit $k$ voisins par point, graphe potentiellement asymétrique (symétriser si nécessaire).
- $\varepsilon$-graph : connecte tous les points à distance $< \varepsilon$, nombre de voisins variable.



# Graphes pondérés

- Utiliser des poids pour refléter la similarité :

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

- Propriétés :
  - $w_{ij} \in (0, 1]$, $w_{ii} = 0$.
  - Capture la force de connexion locale.
  - Symétriser si nécessaire : $w_{ij} = w_{ji} = \max(w_{ij}, w_{ji})$ ou $\frac{w_{ij}+w_{ji}}{2}$.
- Préparation pour Laplacien, diffusion et autres méthodes spectral-based.

# Laplacien de graphe : définition

- Soit $G = (V, E, W)$ un graphe pondéré avec matrice de poids $W$ et matrice de degrés $D$ :

$$D_{ii} = \sum_j w_{ij}$$

- Laplacien non normalisé :

$$L = D - W$$

- Laplacien normalisé :

$$L_{\text{sym}} = D^{-1/2} L D^{-1/2}, \quad L_{\text{rw}} = D^{-1} L$$

- Propriétés :
  - $L$ est semi-définie positive.
  - $\lambda_0 = 0$, vecteur propre constant 1.

# Interprétation du Laplacien

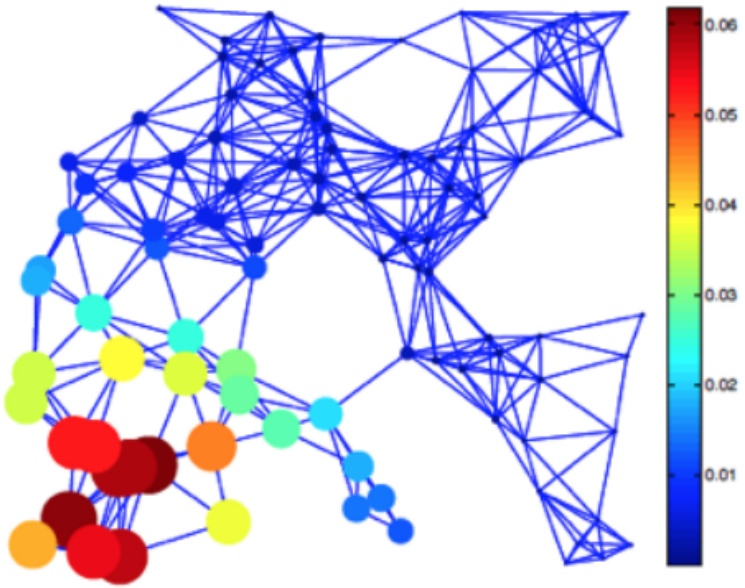
- Analogue discret du Laplacien continu sur une variété.
- Diffusion / propagation sur le graphe :

$$f^\top L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2$$

  - Petit $f^\top L f \Rightarrow f$ varie peu entre voisins fortement connectés.
- Eigenmaps :
  - Les vecteurs propres associés aux plus petites valeurs propres $(\lambda_1, \lambda_2, \dots)$ capturent la structure globale du graphe.
  - Utilisés pour réduction de dimension ou clustering spectral.

# Laplacien : intuition graphique

- Chaque arête du graphe agit comme un ressort entre points.
- Les valeurs propres et vecteurs propres du Laplacien capturent la **dynamique des flux** sur le graphe.




Flux : les différences $f_i - f_j$ tendent à se réduire via diffusion

# Diffusion sur graphe

- Soit un graphe pondéré $G = (V, E, W)$ avec matrice de transition

$$P = D^{-1}W \quad \text{(random walk)}$$

- Diffusion d'une fonction $f$ sur le graphe :

$$f(t+1) = Pf(t)$$

- Intuition : la valeur de $f$ se propage le long des arêtes en fonction de leur poids.
- Méthode symétrisée (pour propriétés spectrales) :

$$P_{\text{sym}} = D^{-1/2}WD^{-1/2}$$

# Noyau de chaleur discret

- Noyau de chaleur : solution discrète de l'équation de diffusion

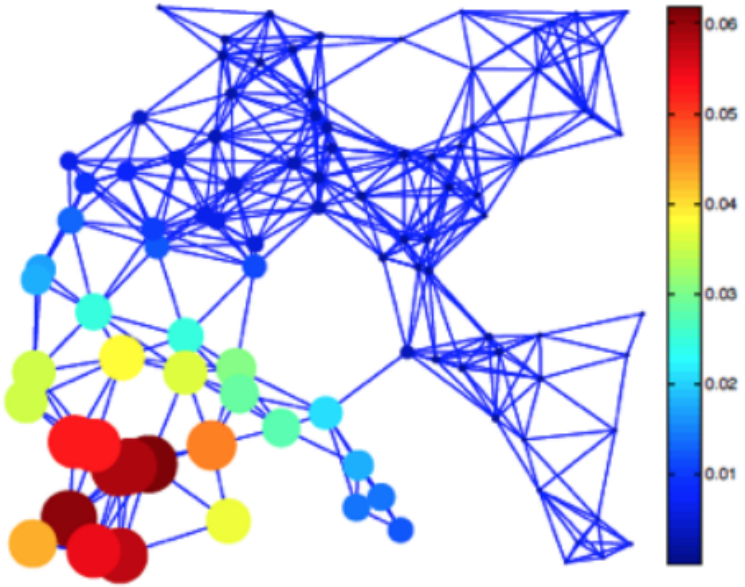
$$H_t = e^{-tL} \approx \sum_{k=0}^{n-1} e^{-t\lambda_k} \phi_k \phi_k^\top$$

où $(\lambda_k, \phi_k)$ sont les valeurs et vecteurs propres du Laplacien.

- Interprétation :
  - $H_t(i,j)$ mesure la diffusion de la chaleur de $i$ vers $j$ après temps $t$.
  - Les modes rapides ($\lambda_k$ grands) décroissent rapidement.
  - Les modes lents ($\lambda_k$ petits) dominent pour grandes échelles.

# Illustration de la diffusion sur graphe

- Initialisation : chaleur concentrée sur un ou quelques noeuds.
- Après $t$ pas de diffusion : la chaleur se propage proportionnellement aux poids des arêtes.
- Permet d'extraire des structures multi-échelle et de mesurer la proximité entre points.




Noeud initial : • chaud

Propagation : •--• (valeurs augmentent)

Flux : les voisins proches reçoivent progressivement la

# Apprentissage semi-supervisé : idée générale

- Objectif : propager des labels connus sur un petit sous-ensemble de points vers l'ensemble du graphe.
- Notation :
  - $X = \{x_1, \ldots, x_n\}$ : points du graphe
  - $L$ : Laplacien du graphe
  - $Y_\ell$ : labels connus sur un sous-ensemble $\ell \subset \{1, \ldots, n\}$
  - $f$ : fonction de labels à propager

# Équation de Poisson discrète sur graphe

- Formulation :

$$Lf = 0 \quad \text{sur les points non étiquetés } u = V \setminus \ell$$

avec conditions de Dirichlet :

$$f_i = Y_i \quad \text{pour } i \in \ell$$

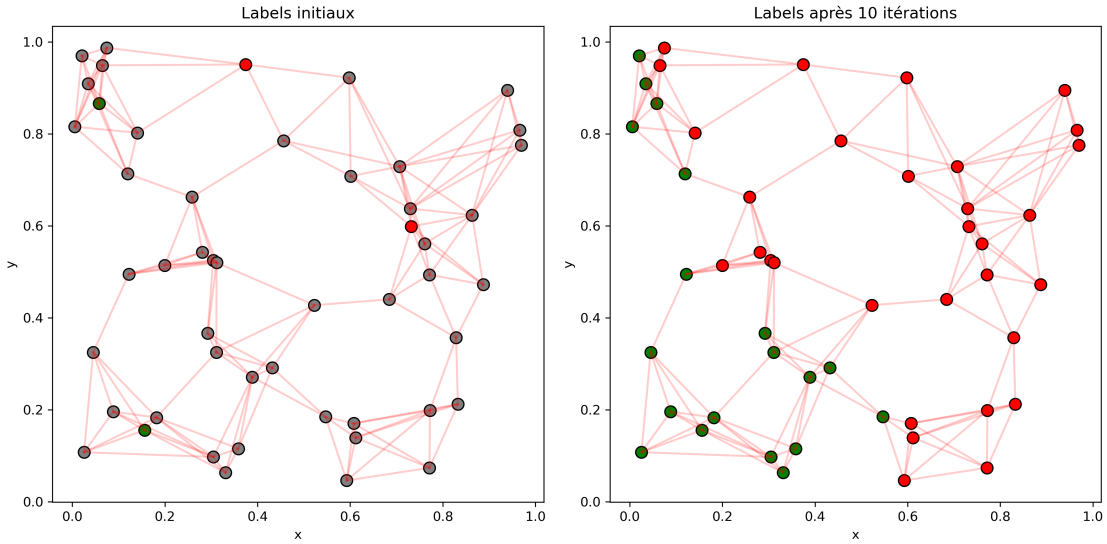
- Interprétation : trouver une fonction $f$ qui varie le moins possible entre voisins fortement connectés, tout en respectant les labels connus.
- Solution pratique : résoudre le système linéaire restreint

$$L_{uu} f_u = -L_{u\ell} Y_\ell$$

où $L$ est partitionné selon $u$ (non-étiquetés) et $\ell$ (étiquetés).

# Propagation de labels sur graphe

- Graphe avec quelques points étiquetés (rouge/vert)
- Résultat : labels propagés vers tous les points via Poisson discrete
- Intuition : labels se diffusent le long des arêtes pondérées



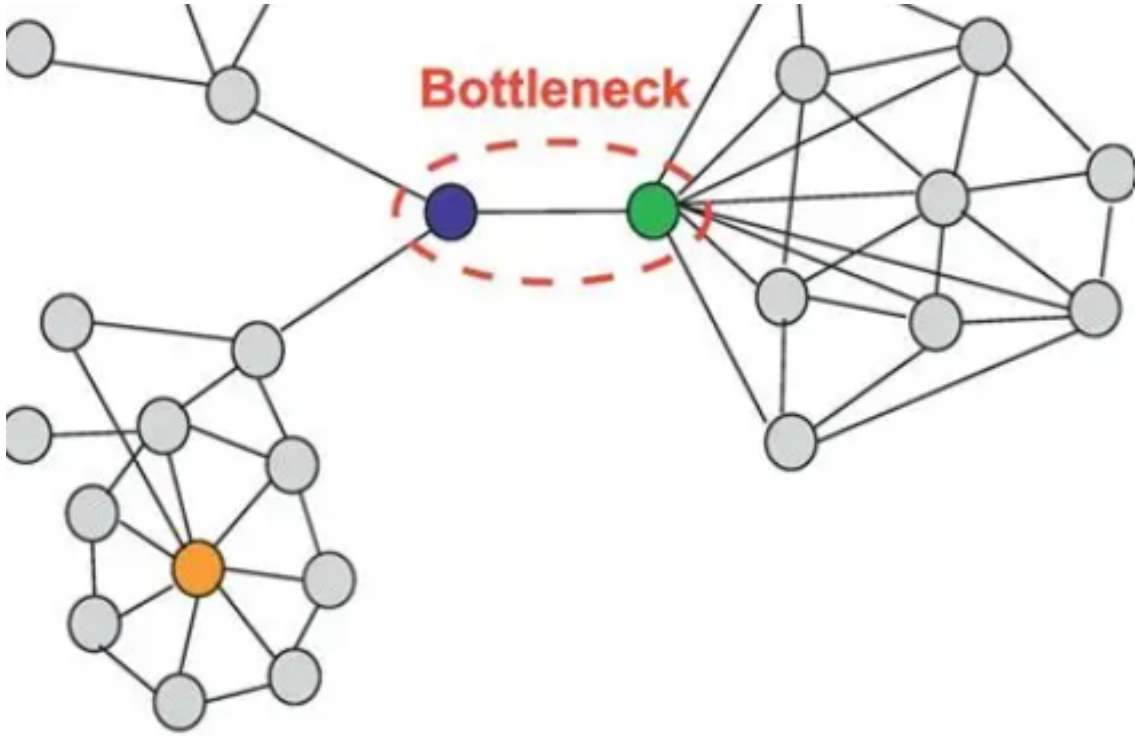
# Courbure de graphe : notion intuitive

- Concept inspiré de la courbure des surfaces continues.
- Mesure comment les voisins d'un noeud sont connectés entre eux.
- Idée :
  - "Bottleneck" : points qui relient deux clusters $\rightarrow$ faible connectivité locale.
  - Graphes très "tordus" $\rightarrow$ distances de diffusion plus grandes entre certains points.
- La courbure discrète peut guider :
  - La détection de communautés
  - La compréhension des structures locales et globales

# Topologie intuitive et diffusion

- Les graphes avec "bottlenecks" ralentissent la diffusion : la chaleur met plus de temps à traverser.
- Les vecteurs propres du Laplacien capturent cette topologie :
  - Petites valeurs propres $\rightarrow$ modes lents $\rightarrow$ grandes structures
  - Grandes valeurs propres $\rightarrow$ modes rapides $\rightarrow$ détails locaux
- Exemple : deux clusters reliés par un petit nombre d'arêtes
  - Distance de diffusion entre clusters $>$ distance intra-cluster
  - Permet de détecter naturellement les communautés

# Illustration : bottleneck et diffusion




`Diffusion : chaleur traverse lentement le bottleneck`

- La structure topologique influence les distances de diffusion et le comportement des méthodes spectral-based.

# Exemple : classification de digits (MNIST)

- Construire un graphe k-NN ou $\varepsilon$-graph à partir des images.
- Pondérer les arêtes par similarité (ex. distance Euclidienne ou cosine) :

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

- Appliquer Laplacian Eigenmaps ou diffusion sur graphe pour réduire la dimension :

$$L = D - W, \quad L\phi_k = \lambda_k \phi_k$$

- Labels connus sur un petit sous-ensemble $\rightarrow$ Poisson discrete pour propagation

# Résultat : embedding spectral

- Les vecteurs propres associés aux plus petites valeurs propres de $L$ fournissent un embedding 2D ou 3D.
- Points proches dans l'espace réduit $\rightarrow$ images similaires (mêmes digits).
- Semi-supervised : labels propagés par Poisson discrete sur graphe pondéré.

```
Exemple schématique :
0 0 0 1 1 1 2 2 2 ...
Clusters de couleurs indiquent labels propagés
```

# Exemple : segmentation d'image

- Pixels = noeuds du graphe, arêtes = voisinage spatial et/ou similarité de couleur
- Pondération :

$$w_{ij} = \exp\left( - \frac{\|I_i - I_j\|^2}{2\sigma_c^2} - \frac{\|p_i - p_j\|^2}{2\sigma_s^2} \right)$$

  où $I_i$ couleur et $p_i$ position du pixel $i$.
- Appliquer diffusion / Poisson discrete pour propager labels initiaux (ex. superpixels, annotations)
- Résultat : segmentation cohérente en clusters homogènes.

# Applications modernes : NLP

- Graphe de mots ou documents :
  - Noeuds = mots ou documents
  - Arêtes = co-occurrence ou similarité sémantique
- Pondération des arêtes par cosine similarity ou embedding pré-entraîné :

$$w_{ij} = \frac{\langle v_i, v_j \rangle}{\|v_i\| \|v_j\|}$$

- Diffusion sur graphe pour :
  - Propagation de labels (ex. classification de documents)
  - Extraction de communautés sémantiques
  - Représentations low-dimensional des mots ou documents

# Applications modernes : Bioinformatique

- Réseaux de gènes ou protéines :
  - Noeuds = gènes / protéines
  - Arêtes = interactions ou corrélations
- Utilisation :
  - Propagation de labels (fonction connue de quelques gènes $\rightarrow$ prédiction pour les autres)
  - Clustering / détection de modules biologiques
  - Analyse multi-échelle via diffusion et valeurs propres du Laplacien
- Exemples : propagation de pathologies, analyse de single-cell RNA-seq

# Module 2 : résumé

- Construction de graphes : k-NN, $\varepsilon$-graph, pondérés
- Laplacien de graphe : définitions, interprétations et Eigenmaps
- Diffusion et noyau de chaleur : distances multi-échelles, modes lents
- Poisson discrete : propagation de labels (apprentissage semi-supervisé)
- Courbure et topologie : bottlenecks, structures locales et globales
- Exemples pratiques : classification de digits, segmentation d'images
- Applications modernes : NLP, bioinformatique