l'Analyse de Données Topologiques (TDA) Homologie persistante et ses applications

Mohamed Amine Fantassi

Faculté des Sciences de Tunis

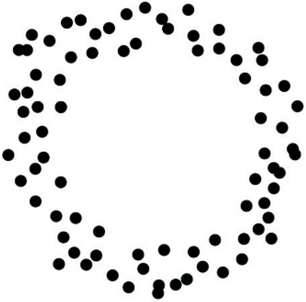
Année universitaire 2024-2025

Introduction à la TDA

L'Analyse de Données Topologiques (TDA) est une méthode qui utilise des concepts de **topologie algébrique** pour extraire des invariants géométriques et topologiques robustes des données.

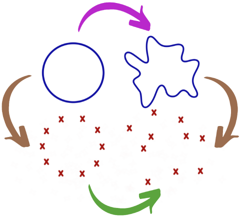
Motivation

Phrase clé : *La TDA permet de capturer la structure des données à différentes échelles tout en étant robuste au bruit.*



Les avantages de la topologie comme un outil

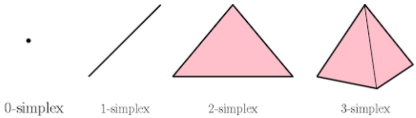
- **Indépendance des systèmes de coordonnées :** La topologie analyse les relations entre les points sans dépendre des choix de coordonnées spécifiques.
- **Invariance par homéomorphisme :** Les propriétés étudiées restent les mêmes sous des transformations continues (étirements, torsions, etc.).
- **Préservation de la structure topologique globale :** Elle permet de capturer les caractéristiques essentielles des données, telles que la connectivité et les trous, même en présence de bruit.



Complexes simpliciaux (Définition)

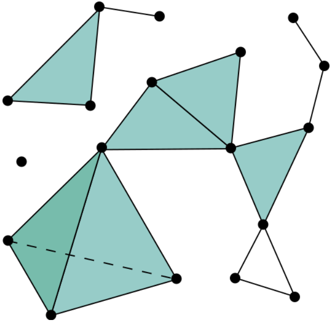
Définition : Un **complexe simplicial** est une structure combinatoire composée de :

- Sommets (points),
- Arêtes (liens entre deux points),
- Simplexes de dimension supérieure (triangles, tétraèdres, etc.).



Construction de complexes simpliciaux à partir de données

Exemple : Considérons un nuage de points.
Étant donné un ensemble de données, ou plus généralement un espace topologique ou métrique, il existe de nombreuses façons de construire des complexes simpliciaux. Nous présentons ici quelques exemples classiques qui sont largement utilisés en pratique.

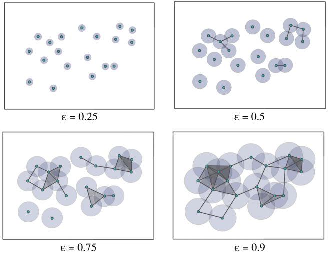


Les points sont les sommets, les connexions forment des simplex de dimension (0,1,2,...)

Vietoris-Rips complex

Le **complexe de Vietoris-Rips** $Rips_\alpha(X)$ est l'ensemble des simplexes $[x_0, \ldots, x_k]$ tels que $d_X(x_i, x_j) \leq \alpha$ pour tout (i, j).

Il découle immédiatement de la définition qu'il s'agit d'un *complexe simplicial abstrait*. Cependant, en général, même lorsque X est un sous-ensemble fini de $\mathbb{R}^d$, le complexe $Rips_\alpha(X)$ n'admet pas de réalisation géométrique dans $\mathbb{R}^d$; en particulier, il peut être de dimension supérieure à d.



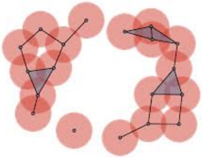
Complexe de Čech

Étroitement lié au complexe de Vietoris-Rips, le **complexe de Čech** $Cech_\alpha(X)$ est défini comme l'ensemble des simplexes $[x_0, \ldots, x_k]$ tels que les $k+1$ boules fermées $B(x_i, \alpha)$ ont une intersection non vide.

Relation entre les complexes :

$$Rips_\alpha(X) \subseteq Cech_\alpha(X) \subseteq Rips_{2\alpha}(X)$$

Lien avec $\mathbb{R}^d$: Si $X \subseteq \mathbb{R}^d$, alors $Cech_\alpha(X)$ et $Rips_{2\alpha}(X)$ partagent le même squelette 1-dimensionnel, c'est-à-dire le même ensemble de sommets et d'arêtes.



Homologie

Définition : L'**homologie** est un concept classique en topologie algébrique qui fournit un outil puissant pour formaliser et traiter les notions de caractéristiques topologiques d'un espace topologique ou d'un complexe simplicial de manière algébrique.

Pour chaque dimension k, les trous k-dimensionnels sont représentés par un espace vectoriel H_k, dont la dimension correspond intuitivement au nombre de telles caractéristiques indépendantes.

Exemples :

- Le groupe d'homologie 0-dimensionnelle H_0 représente les **composantes connexes** du complexe.
- Le groupe d'homologie 1-dimensionnelle H_1 représente les **lacets** (ou boucles) 1-dimensionnels.
- Le groupe d'homologie 2-dimensionnelle H_2 représente les **cavités** 2-dimensionnelles.

Nombres de Betti - H0 et H1

Définition des groupes de Betti :

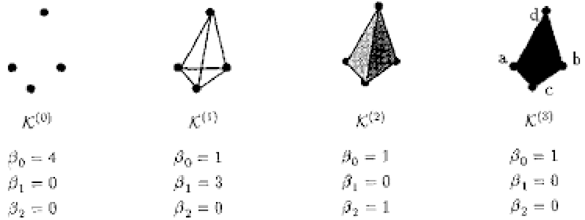
- **H0** : Le groupe d'homologie de dimension 0 capture le nombre de composants connexes d'un espace.
- **H1** : Le groupe d'homologie de dimension 1 capture les lacets (ou "boucles") dans l'espace.

Exemple : Un cercle a :

- $H_0 \cong \mathbb{Z}$ (un composant connexe),
- $H_1 \cong \mathbb{Z}$ (une boucle autour du cercle).

Illustration de H0,H1 et H2

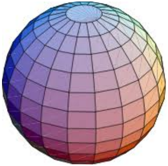
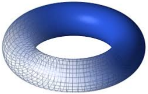
Exemple visuel : calculant les premiers groupes d'homologie des simplex.



Torus et Sphère

Exemples :

- La **sphère** : $H_0 \cong \mathbb{Z}$ (1 composant connexe), $H_1 = 0$ (pas de boucle), $H_2 \cong \mathbb{Z}$ (surface fermée).
- Le **torus** : $H_0 \cong \mathbb{Z}$ (1 composant connexe), $H_1 \cong \mathbb{Z}^2$ (2 boucles indépendantes), $H_2 \cong \mathbb{Z}$ (surface fermée).

Nombres de Betti et caractéristique d'Euler

Introduction : Les **nombres de Betti** forment une famille d'invariants topologiques qui mesurent la connectivité dans une dimension donnée. À l'aide de ces nombres, nous pouvons interpréter la **caractéristique d'Euler** de manière précise.

Définition : Soit $n \geq 0$ un entier et K un complexe simplicial. L'**espace vectoriel des n-chaînes** dans K, noté $C_n(K)$, est l'espace vectoriel libre sur $\mathbb{Z}_2$, engendré par les n-simplexes de K.

Opérateur bord : En définissant le bord d'un n-simplexe comme la somme formelle de ses faces $(n-1)$-dimensionnelles, nous obtenons une transformation linéaire :

$$\partial_n : C_n(K) \to C_{n-1}(K).$$

Notation : Pour un simplexe $\{p_0, \dots, p_n\}$, le $(n-1)$-simplexe obtenu en omettant le sommet p_i est noté $\{p_0, \dots, \hat{p}_i, \dots, p_n\}$.

L'opérateur bord

Définition : L'**opérateur bord** $\partial_n : C_n(K) \to C_{n-1}(K)$ est la transformation linéaire définie sur les simplexes par :

$$\partial_n(\{p_0, \ldots, p_n\}) = \sum_{i=0}^{n} \{p_0, \ldots, \hat{p}_i, \ldots, p_n\},$$

où $\{p_0, \ldots, \hat{p}_i, \ldots, p_n\}$ désigne le $(n-1)$-simplexe obtenu en omettant le sommet p_i.

Lemme clé : Intuitivement, ce lemme affirme que **le bord d'un bord est nul** :

$$\partial_{n-1} \circ \partial_n = 0.$$

Ce résultat fondamental est ce qui permet de construire la théorie de l'homologie, en établissant une structure cohérente pour les cycles et les bords.

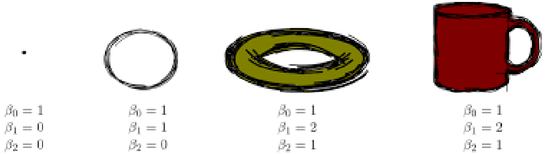
Cycles et bords

Définition : Un élément $c \in C_n(K)$ est un n-**cycle** si $\partial_n(c) = 0$. L'espace vectoriel associé aux n-cycles est défini par :

$$Z_n(K) = \ker \partial_n = \{c \in C_n(K) : \partial_n(c) = 0\}.$$

Si c appartient à l'image de ∂_{n+1}, c'est-à-dire $c = \partial_{n+1}(d)$ pour une $(n+1)$-chaîne d, alors c est un n-**bord**. L'espace vectoriel associé aux n-bords est défini par :

$$B_n(K) = \mathrm{Im}\, \partial_{n+1} = \{\partial_{n+1}(d) : d \in C_{n+1}(K)\}.$$



Le théorème de Nerve

Théorème : Si un recouvrement ouvert $\{U_i\}$ satisfait $\bigcap U_i$ est contractile, alors le complexe nerve est homotope à l'espace initial.

- **Le complexe nerve** est constitué des simples intersections des ensembles de recouvrement.

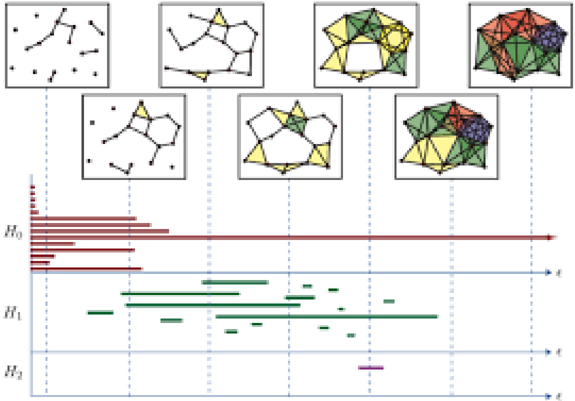
Illustration du Nerve et Lien avec l'Homologie

Exemple : Le complexe nerve construit à partir d'un recouvrement ouvert.

Le complexe nerve construit à partir des recouvrements permet de retrouver l'homologie de l'espace initial.

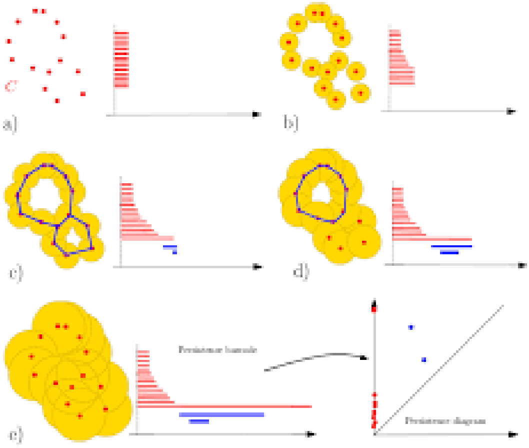
Diagrammes de persistance

Définition : Les diagrammes de persistance sont utilisés pour capturer l'apparition et la disparition des structures topologiques dans les données au cours d'une filtration. Chaque barres dans le diagramme représente une caractéristique topologique persistante.



Exemple de diagramme de persistance

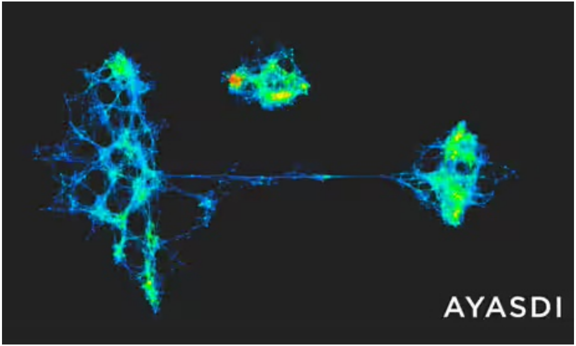
Exemple : Visualisation d'un diagramme de persistance sous forme de "codes-barres".



Chaque barre représente une structure topologique qui persiste sur une échelle donnée.

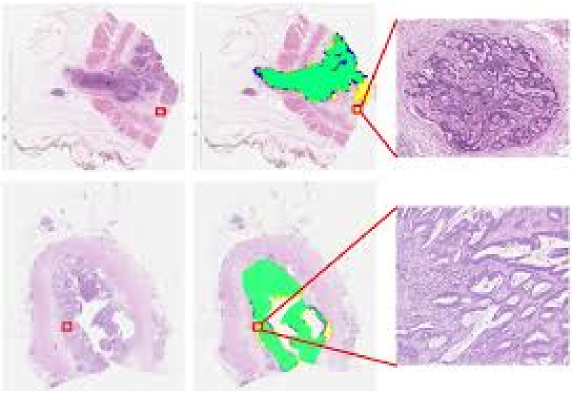
Applications : Neurosciences

Exemple : Étude des connectomes pour détecter des motifs associés à des maladies neurologiques.



Applications : Cancer du sein

Exemple : Analyse de données histologiques pour classer les tissus cancéreux.



Bibliothèque Gudhi pour calculer l'homologie

Gudhi (*Geometric Understanding in Higher Dimensions*) est une bibliothèque Python puissante pour l'analyse topologique des données.

Fonctionnalités principales :

- Construction et manipulation de complexes simpliciaux, tels que les complexes de Rips-Vietoris.
- Calcul des groupes d'homologie persistants à partir de filtrations.
- Visualisation des diagrammes de persistance et des codes-barres.

Exemple d'utilisation pour un complexe de Rips-Vietoris :
Avantages :

- Interface intuitive en Python.
- Performances élevées pour des données volumineuses.

Exemple 1 : Nuage de points

Considérons un nuage de points dans $\mathbb{R}^2$:

```
data = [[0, 0], [1, 0], [0, 1], [1, 1], [2, 1]].
```

Étapes pour calculer l'homologie avec Gudhi :

1. Créer un complexe de Rips-Vietoris avec une longueur maximale d'arête (`max_edge_length`).
2. Calculer la persistance à l'aide de la méthode `persistence()`.
3. Visualiser le diagramme de persistance avec `gudhi.plot_persistence_diagram`.

Code :

- `rips = gudhi.RipsComplex(points=data, max_edge_length=1.5)`
- `simplex_tree = rips.create_simplex_tree(max_dimension=2)`
- `persistence = simplex_tree.persistence()`
- `gudhi.plot_persistence_diagram(persistence)`

Cela permet de détecter les composantes connexes et les lacets.

Exemple 2 : Points dans $\mathbb{R}^3$

Considérons un ensemble de points distribués autour d'une sphère dans $\mathbb{R}^3$. Ces points peuvent représenter une structure creuse avec des cavités.

Étapes pour calculer l'homologie persistante :

1. Charger les points à partir d'un fichier ou les générer aléatoirement.
2. Construire le complexe de Rips-Vietoris avec `max_dimension = 3`.
3. Calculer la persistance et extraire les caractéristiques topologiques.

Code :

- `data = generate_sphere_points()` (fonction pour générer les points sur une sphère).
- `rips = gudhi.RipsComplex(points=data, max_edge_length=2.0)`
- `simplex_tree = rips.create_simplex_tree(max_dimension=3)`
- `persistence = simplex_tree.persistence()`
- `gudhi.plot_persistence_diagram(persistence)`

Cet exemple illustre comment détecter les composantes connexes (H_0), les lacets (H_1), et les cavités (H_2).

Références

- Gunnar Carlsson, "Topology and Data," Bulletin of the American Mathematical Society, 2009.
- Robert Ghrist, "Barcodes: The Persistent Topology of Data," Bulletin of the AMS, 2008.
- Afra Zomorodian, "Topology for Computing," Cambridge University Press, 2005.