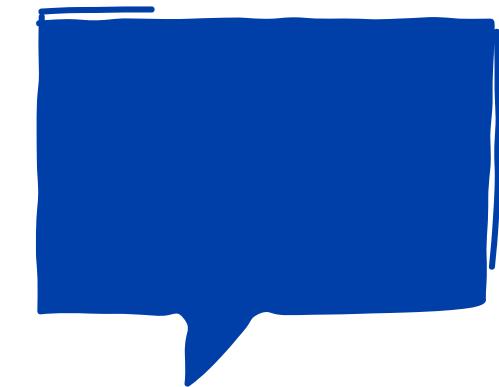
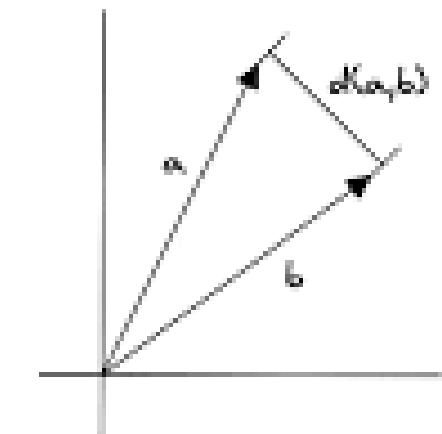


Analyse et Exploration des Données avec similarity measures

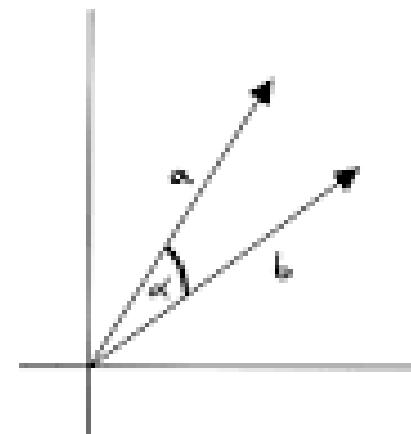
Présenter par : Manel Trabelsi 2 IKM
Mariem Hamdi 2 DSB
May Nebli 2 DSB



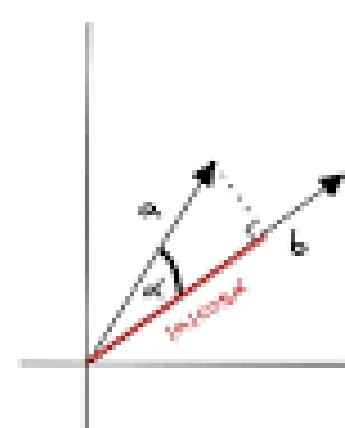
Similarity Metrics



Euclidean Distance



Cosine Similarity



Dot Product

Plan du travail

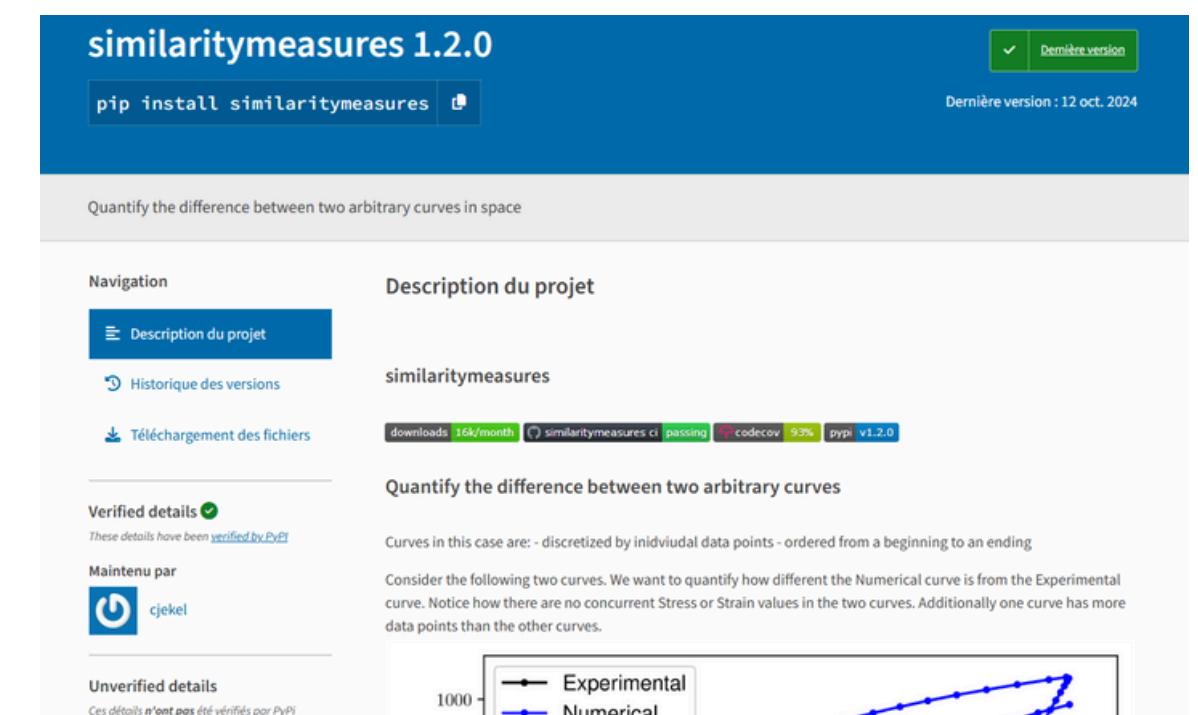
- 1.** Introduction
- 2.** Les fonctionnalités principales
- 3.** Les métriques de similarité
- 4.** Un exemple pratiques avec du code Python
- 5.** Conclusion



Introduction

Le package `similaritymeasures` est une bibliothèque Python dédiée à la comparaison de courbes en deux dimensions (2D).

Puissante et conviviale elle est particulièrement utile dans des domaines variés tels que l'analyse de données, l'ingénierie et les sciences des données. Grâce à des métriques mathématiques robustes, cette bibliothèque permet de quantifier avec précision la similarité entre deux trajectoires ou ensembles de points.



Pourquoi utiliser des mesures de similarité ?

1 Classification

Classer des données en fonction de leur similarité avec des classes connues.

2 Clustering

Grouper des données similaires en fonction de leurs caractéristiques.

3 Recherche d'informations

Trouver des documents similaires en fonction de leur contenu ou de leurs métadonnées.

4 Recommandation

Recommander des articles ou des produits similaires en fonction des préférences de l'utilisateur.

Avantages itératifs des mesures de similarité

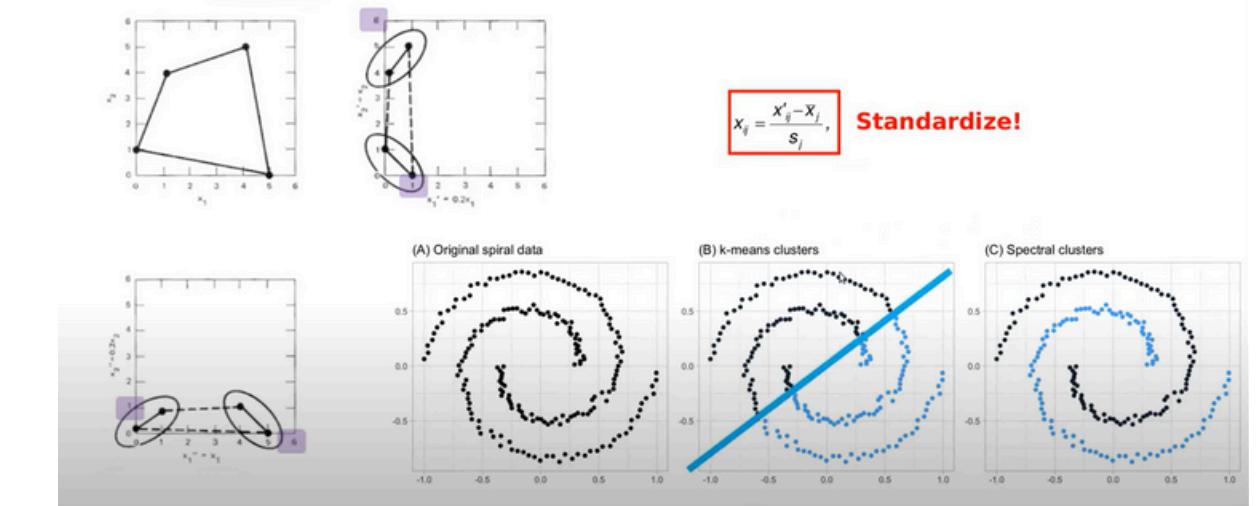
Simplifier les calculs



Appliquer dans des scénarios pratiques

Évaluer l'alignement des données

Fonctionnalités de la bibliothèque Python "Similarity Measures"



```

import numpy as np
import similaritymeasures
import matplotlib.pyplot as plt

# Generate random experimental data
x = np.random.random(100)
y = np.random.random(100)
exp_data = np.zeros((100, 2))
exp_data[:, 0] = x
exp_data[:, 1] = y

# Generate random numerical data
x = np.random.random(100)
y = np.random.random(100)
num_data = np.zeros((100, 2))
num_data[:, 0] = x
num_data[:, 1] = y

# quantify the difference between the two curves using PCM
pcm = similaritymeasures.pcm(exp_data, num_data)

# quantify the difference between the two curves using
# Discrete Frechet distance
df = similaritymeasures.frechet_dist(exp_data, num_data)

# quantify the difference between the two curves using
# area between two curves
area = similaritymeasures.area_between_two_curves(exp_data, num_data)

# quantify the difference between the two curves using
# Curve Length based similarity measure
  
```

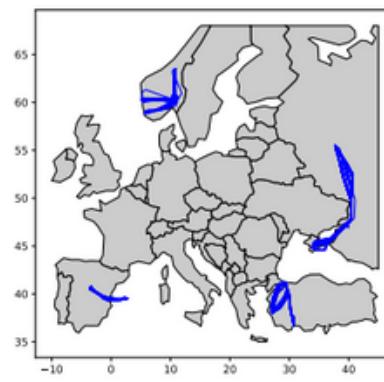
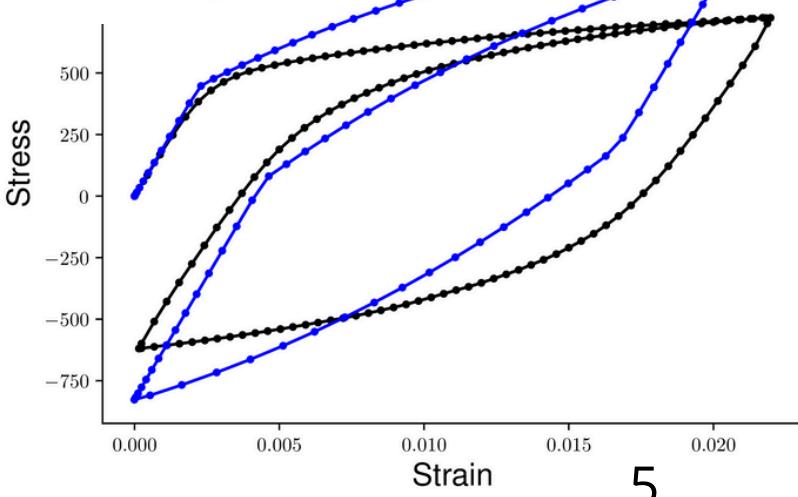
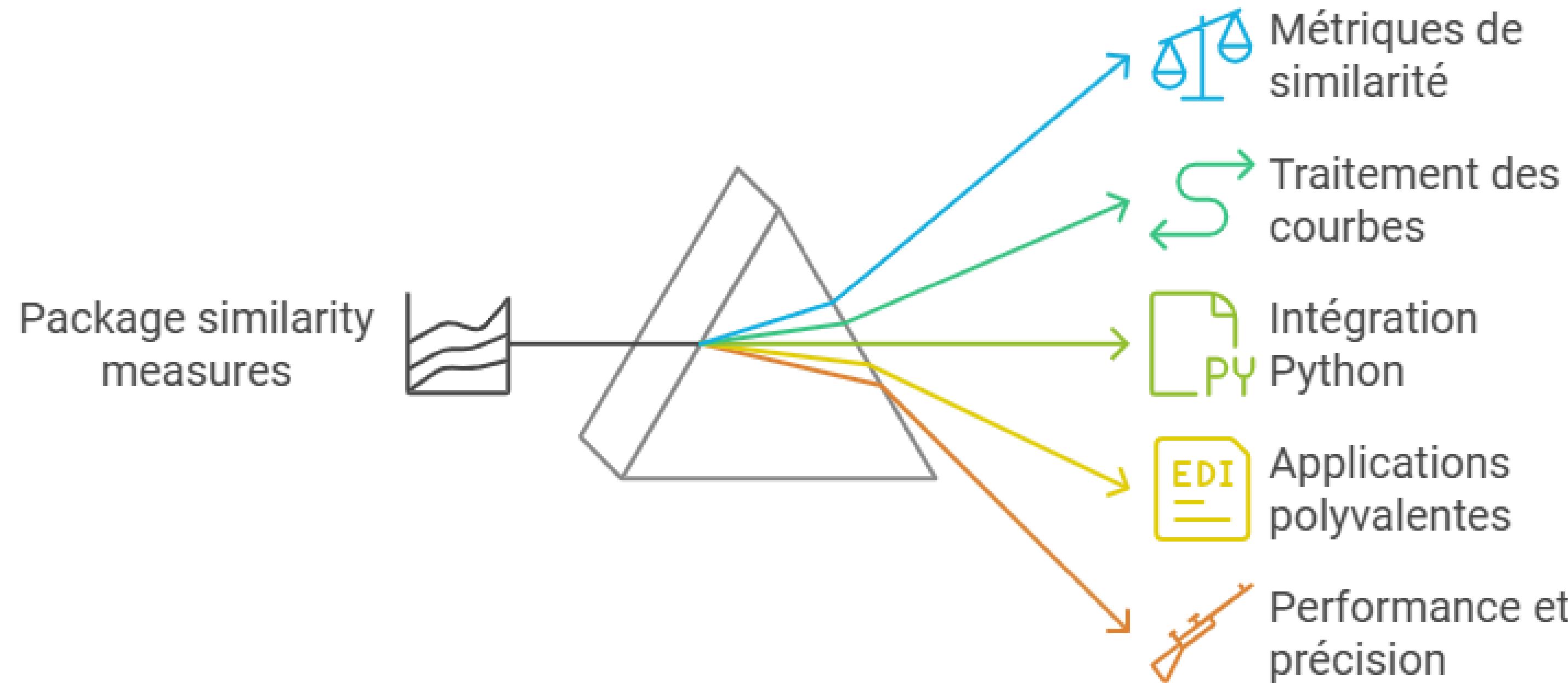


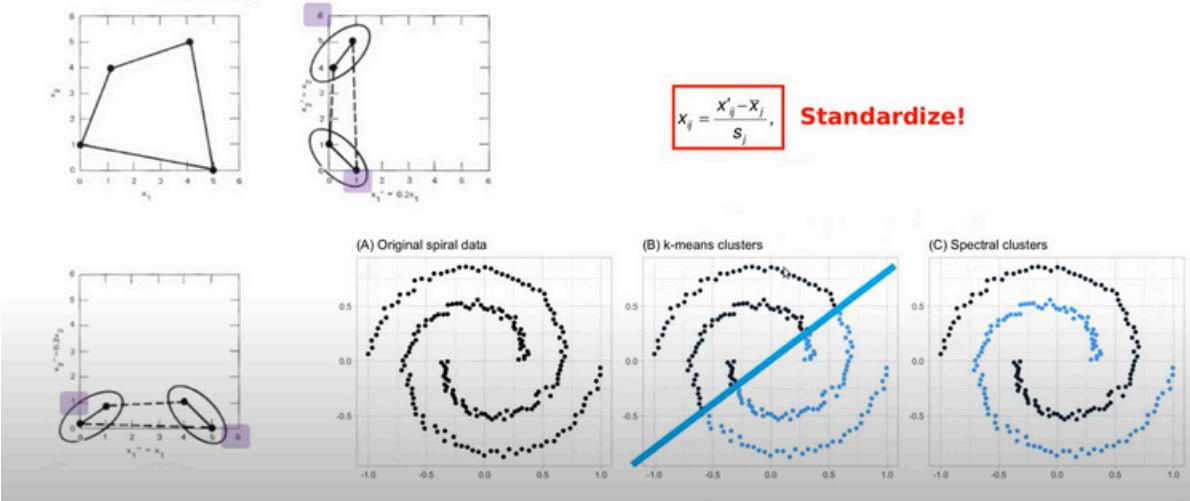
Fig. 1. Trajectories of selected commercial flights.



Fonctionnalités principales du package similaritymeasures



Les métriques de similarité



```

import numpy as np
import similaritymeasures
import matplotlib.pyplot as plt

# Generate random experimental data
x = np.random.random(100)
y = np.random.random(100)
exp_data = np.zeros((100, 2))
exp_data[:, 0] = x
exp_data[:, 1] = y

# Generate random numerical data
x = np.random.random(100)
y = np.random.random(100)
num_data = np.zeros((100, 2))
num_data[:, 0] = x
num_data[:, 1] = y

# quantify the difference between the two curves using PCM
pcm = similaritymeasures.pcm(exp_data, num_data)

# quantify the difference between the two curves using
# Discrete Frechet distance
df = similaritymeasures.frechet_dist(exp_data, num_data)

# quantify the difference between the two curves using
# area between two curves
area = similaritymeasures.area_between_two_curves(exp_data, num_data)

# quantify the difference between the two curves using
# Curve Length based similarity measure

```

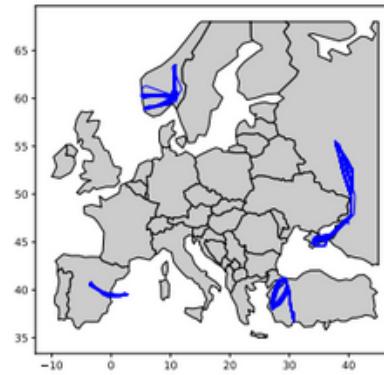
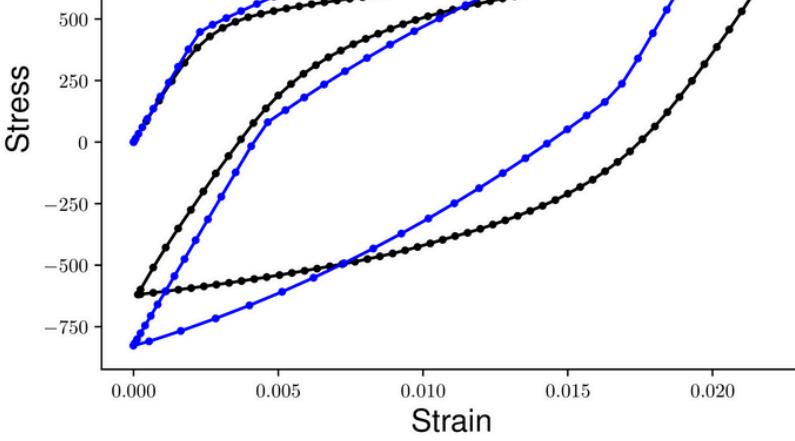
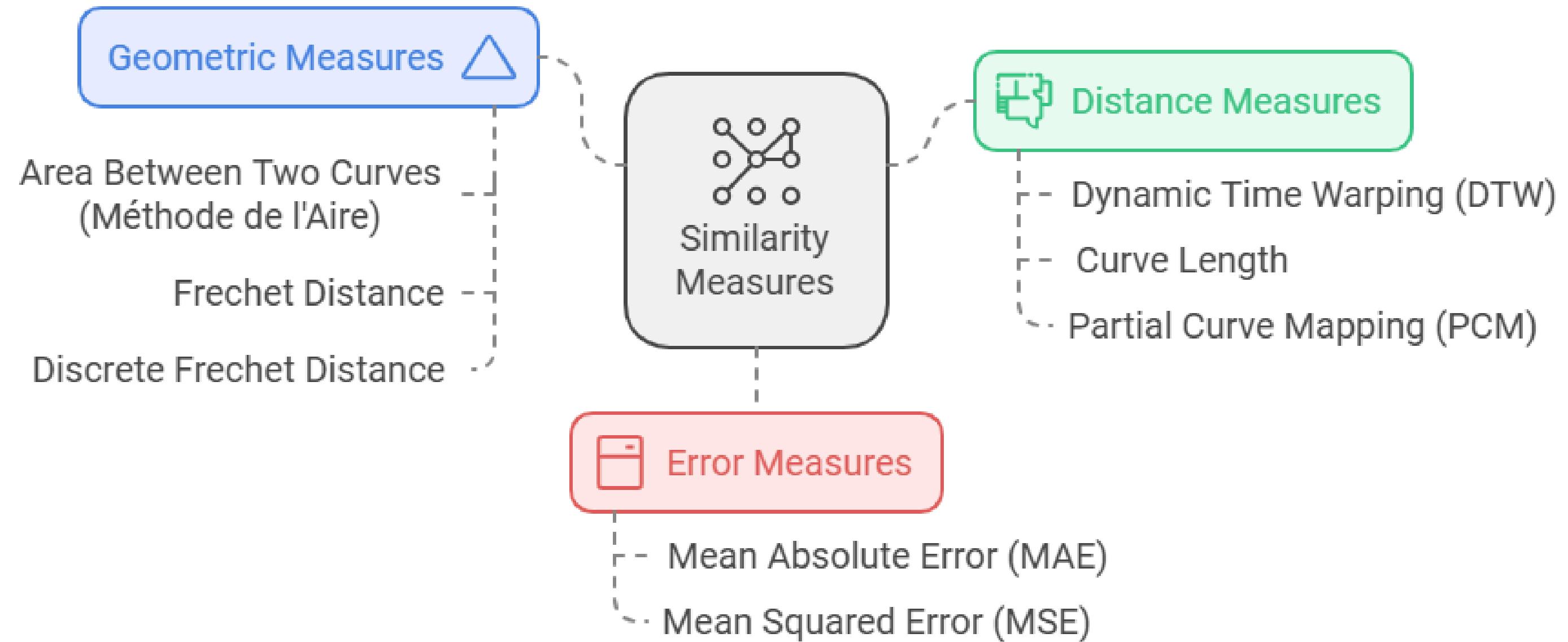


Fig. 1. Trajectories of selected commercial flights.



Métriques pour la comparaison de données

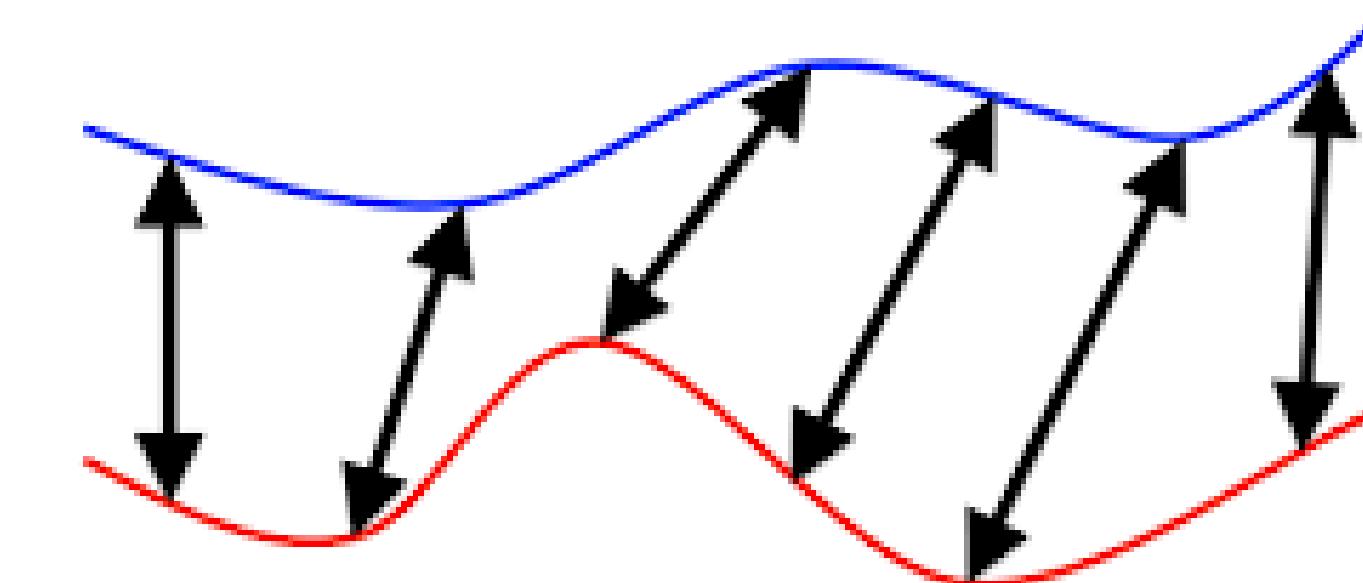


Métrique de similarité : Dynamic Time Warping (DTW)

Le Dynamic Time Warping (DTW) est une méthode qui évalue la similarité entre deux séries temporelles ou trajectoires, même en cas de décalages temporels ou d'échelles différentes.

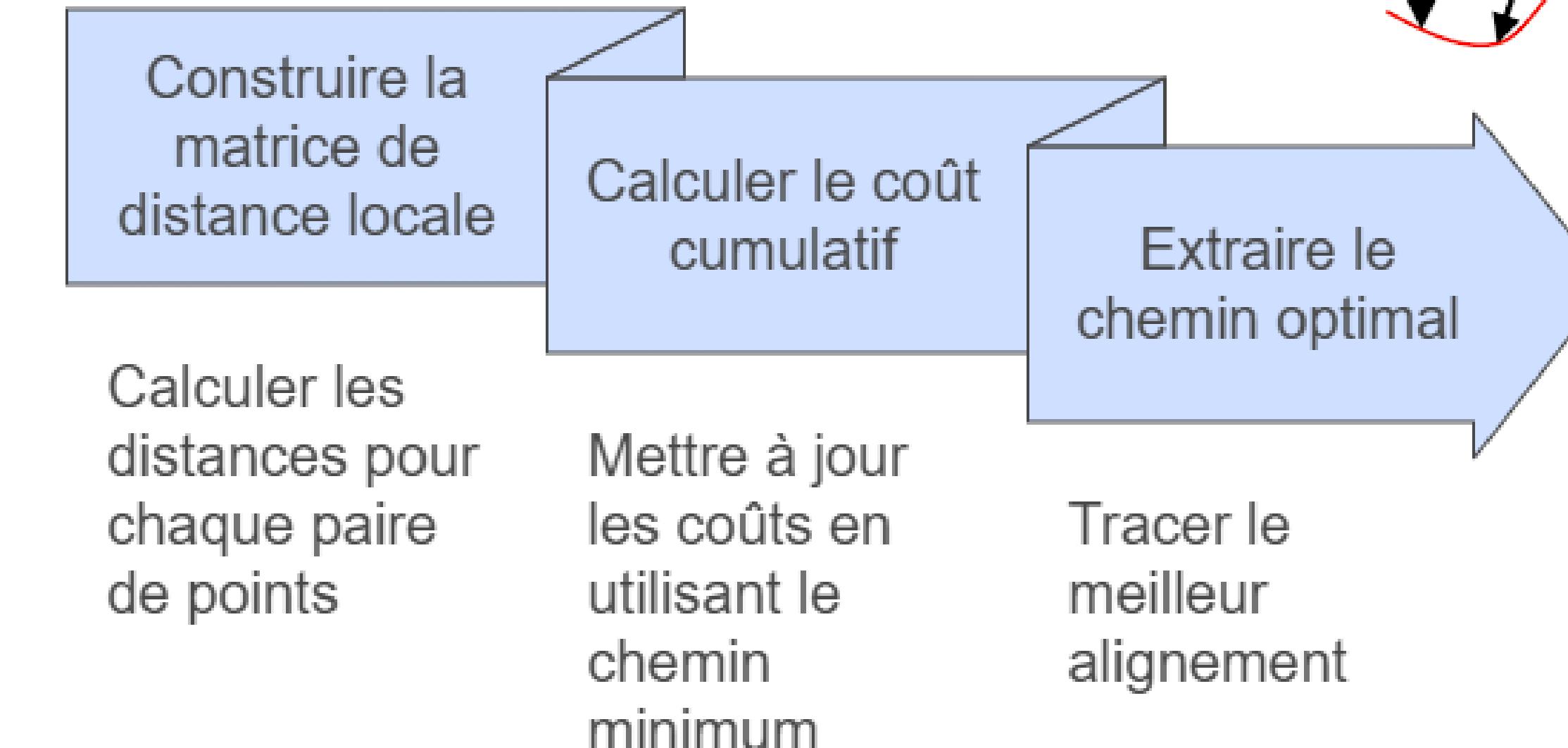
DTW trouve un alignement optimal entre les points des deux séries en minimisant la distance cumulée entre eux.

dynamic time warping

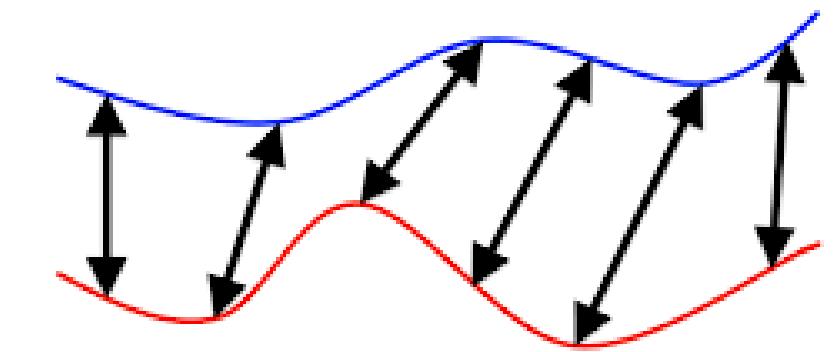


Métrique de similarité : Dynamic Time Warping (DTW)

Processus de calcul DTW:



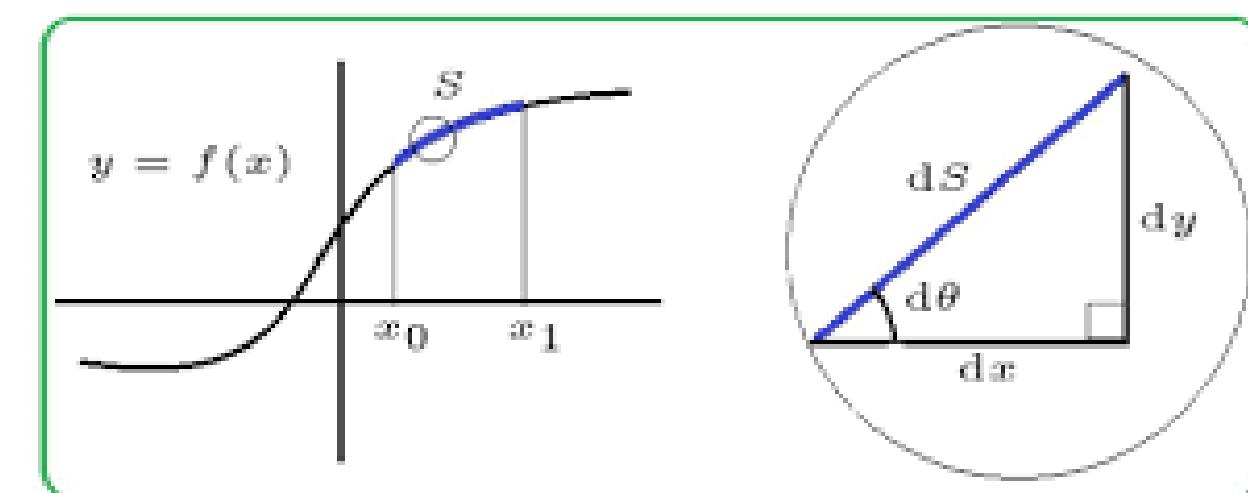
dynamic time warping



Métrique de similarité : Curve length (Longueur de courbe)

La longueur de courbe (curve length) est une mesure utilisée pour déterminer la distance totale parcourue le long d'une courbe ou d'une trajectoire, qu'elle soit dans un espace uni- ou multidimensionnel. En tant que métrique de similarité, elle sert à comparer la structure ou la forme globale de deux trajectoires ou courbes.

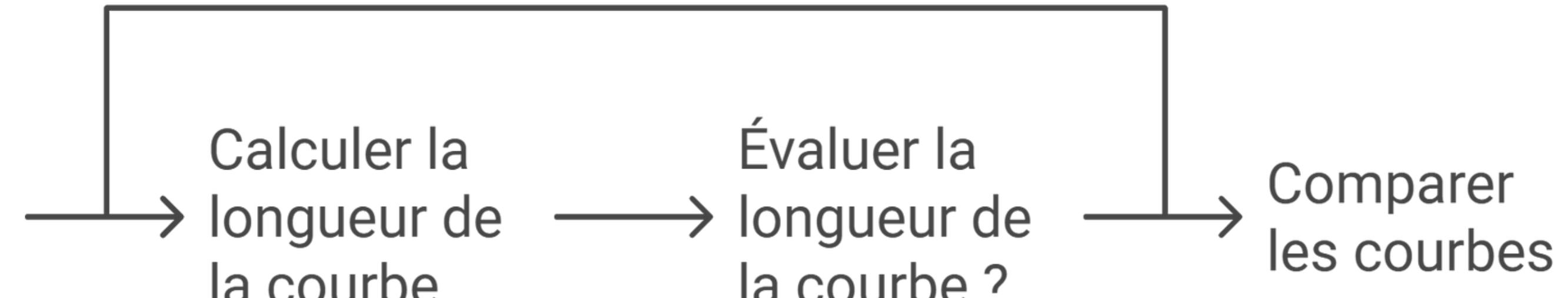
The Length of a Curve



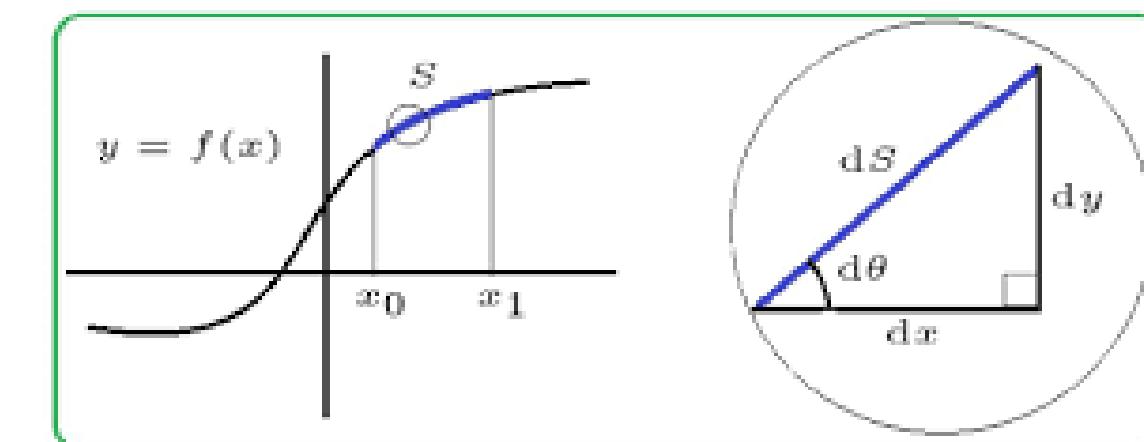
Métrique de similarité : Curve length (Longueur de courbe)

Processus de Longueur de courbe:

Définir la courbe avec des points 2D



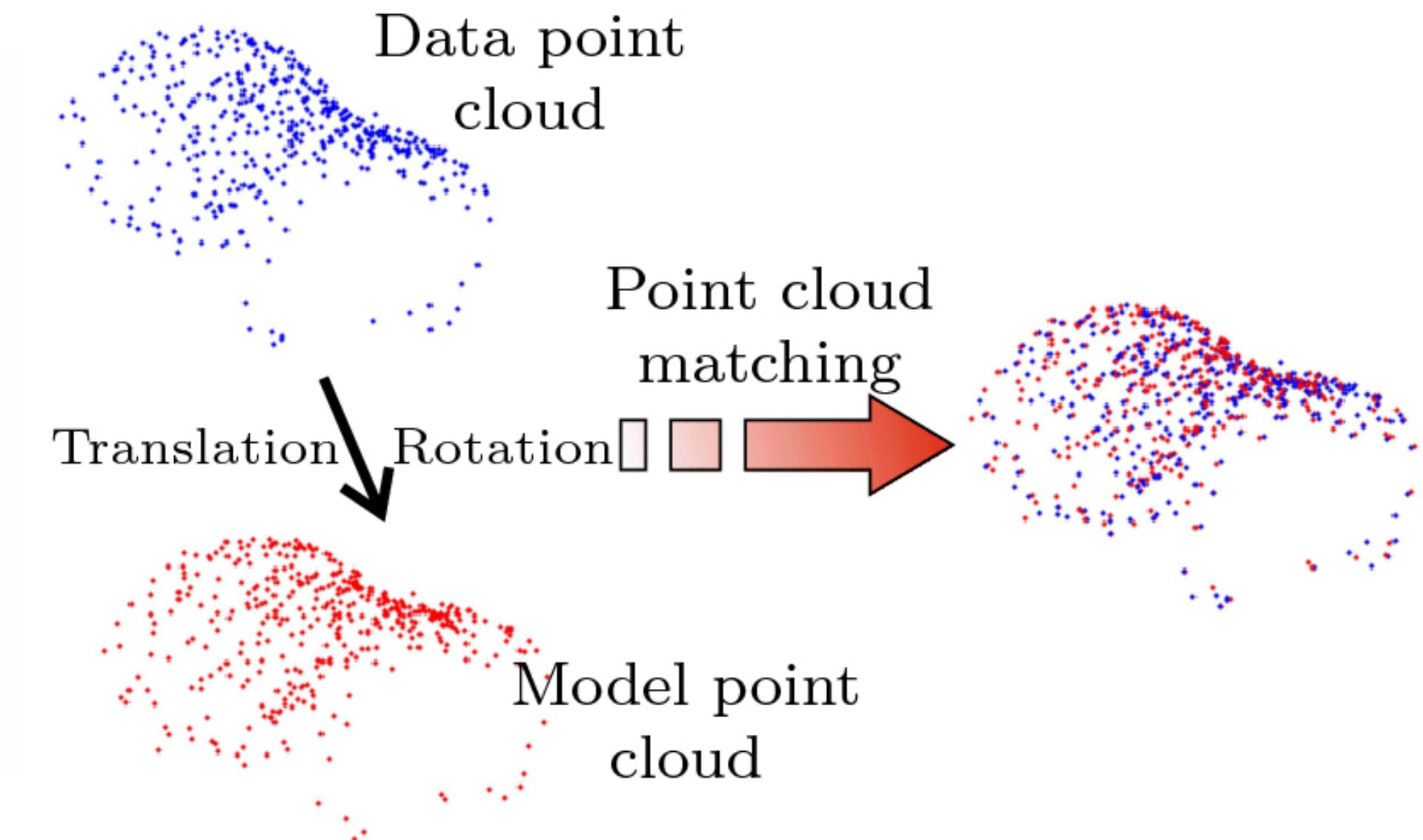
The Length of a Curve



Les métriques de similarité : PCM

PCM (Point-Cloud Matching)

PCM est une méthode utilisée pour comparer des formes basées sur des points dans l'espace. Elle consiste à trouver une correspondance optimale entre les points des deux formes, minimisant la distance entre les points correspondants.



Les métriques de similarité : PCM

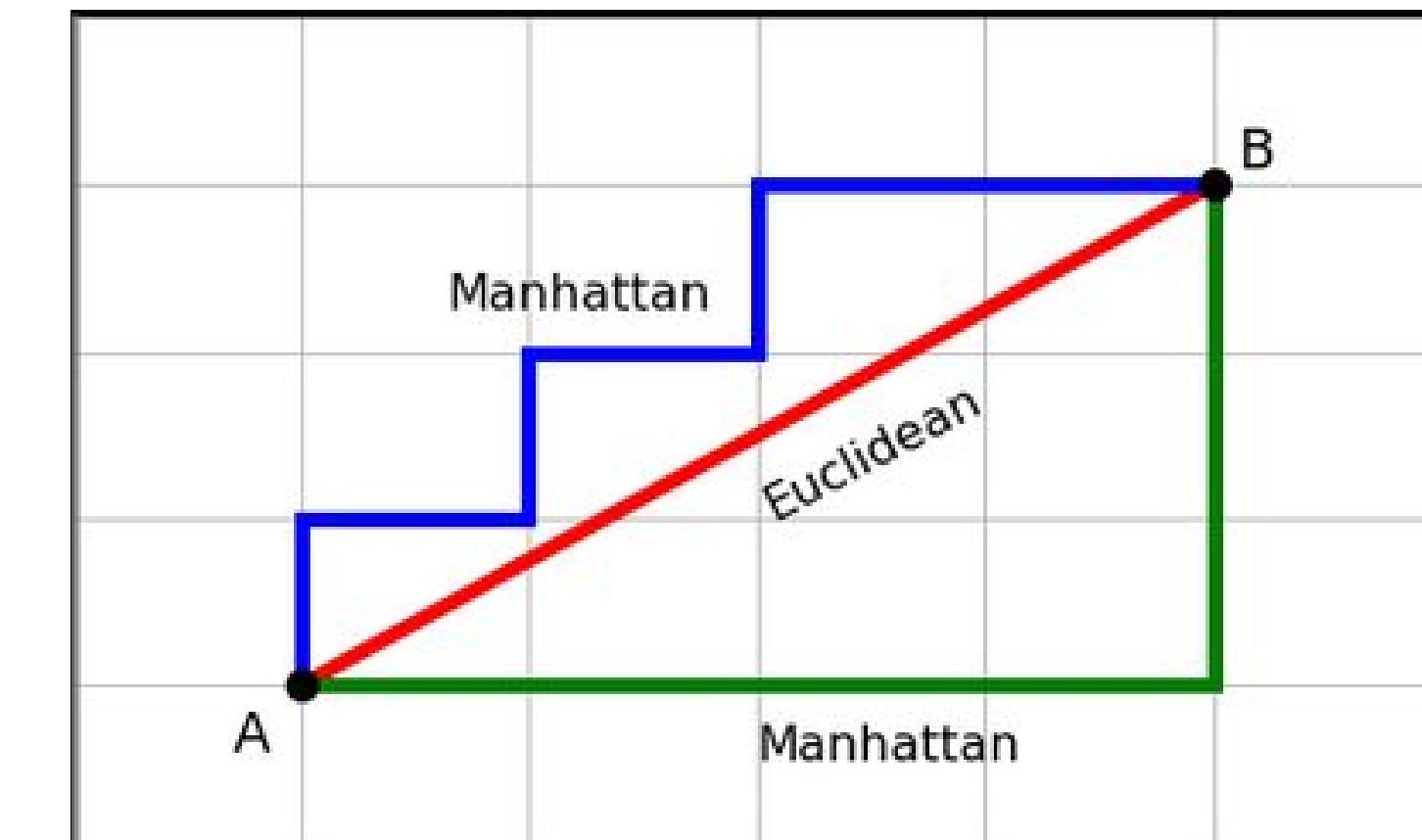
Étapes du Fonctionnement de PCM



-  Diviser les courbes en segments
-  Aligner les segments
-  Calculer les distances entre les segments
-  Ajuster pour les segments partiels
-  Combiner les distances pour un score global

Métrique de similarité : distance de manhattan

La distance de Manhattan est une mesure utilisée pour déterminer la distance entre deux points sur un chemin en forme de grille. Contrairement à la distance euclidienne, qui mesure la ligne la plus courte possible entre deux points, la distance de Manhattan mesure la somme des différences absolues entre les coordonnées des points.



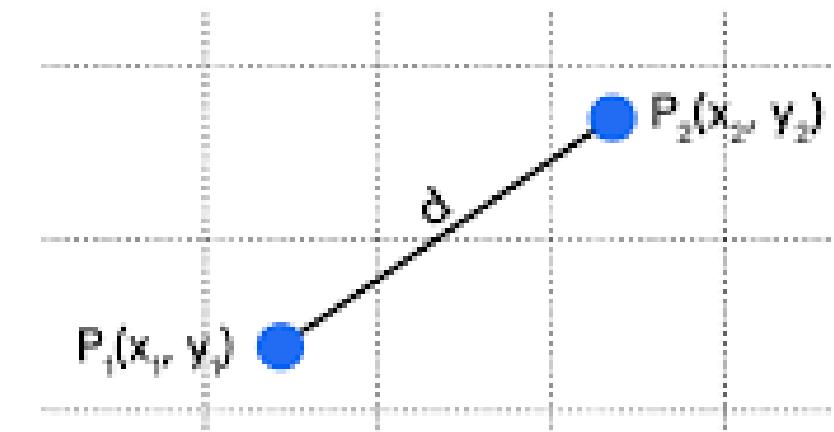
Propriétés mathématiques de la distance de Manhattan

1. **Non-négativité** : La distance entre deux points quelconques est toujours non négative.
 $d(x, y) \geq 0$ pour tout x et tout y .
2. **Identité des indiscernables** : La distance entre un point et lui-même est nulle, et si la distance entre deux points est nulle, il s'agit du même point. $d(x, y) = 0$ si et seulement si $x = y$.
3. **Symétrie** : La distance du point A au point B est la même que la distance de B à A. $d(x, y) = d(y, x)$ pour tout x et y .

Métrique de similarité : distance euclidienne

- La distance Euclidienne est la distance la plus couramment utilisée. Elle représente la longueur directe entre deux points dans un espace multidimensionnel.
- C'est l'équivalent d'une "ligne droite" entre deux points.

Euclidean Distance



$$\text{Euclidean Distance (d)} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Utilisation de la distance euclidienne pour le calcul de similarité

1 Mesure de Proximité

La distance euclidienne permet de quantifier la proximité entre deux objets (points, vecteurs, etc.) dans un espace à plusieurs dimensions.

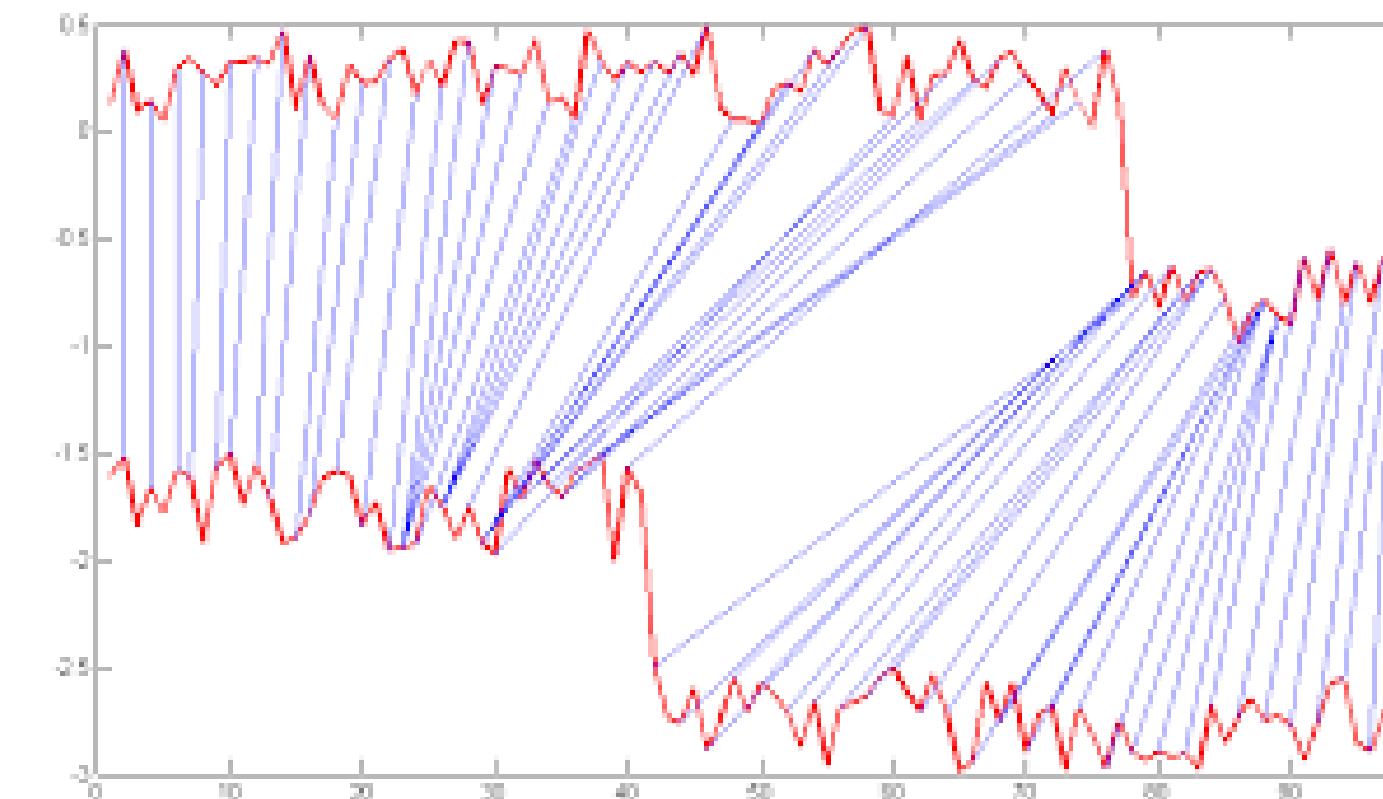
2 Clustering et Classification

Elle est utilisée dans de nombreux algorithmes de clustering et de classification pour regrouper les données en fonction de leur similarité.

Métrique de similarité : Area Method (Méthode de l'aire)

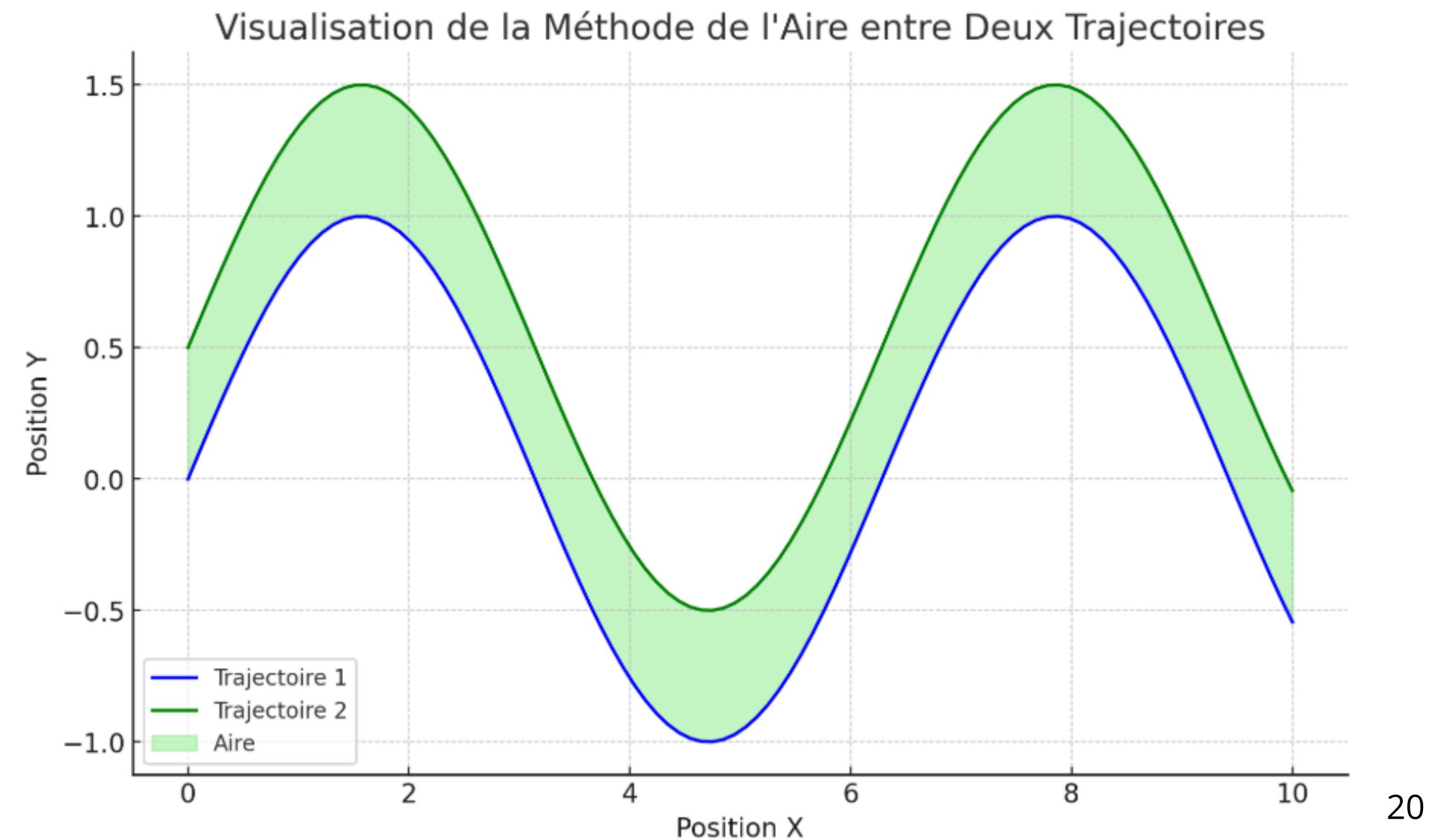
est une technique de comparaison entre deux trajectoires ou courbes. Elle évalue la similitude entre deux trajectoires en mesurant l'aire comprise entre elles tout au long de leurs parcours respectifs.

L'aire entre deux courbes est définie comme la somme des distances perpendiculaires entre les points correspondants des deux courbes, intégrée sur la longueur totale.



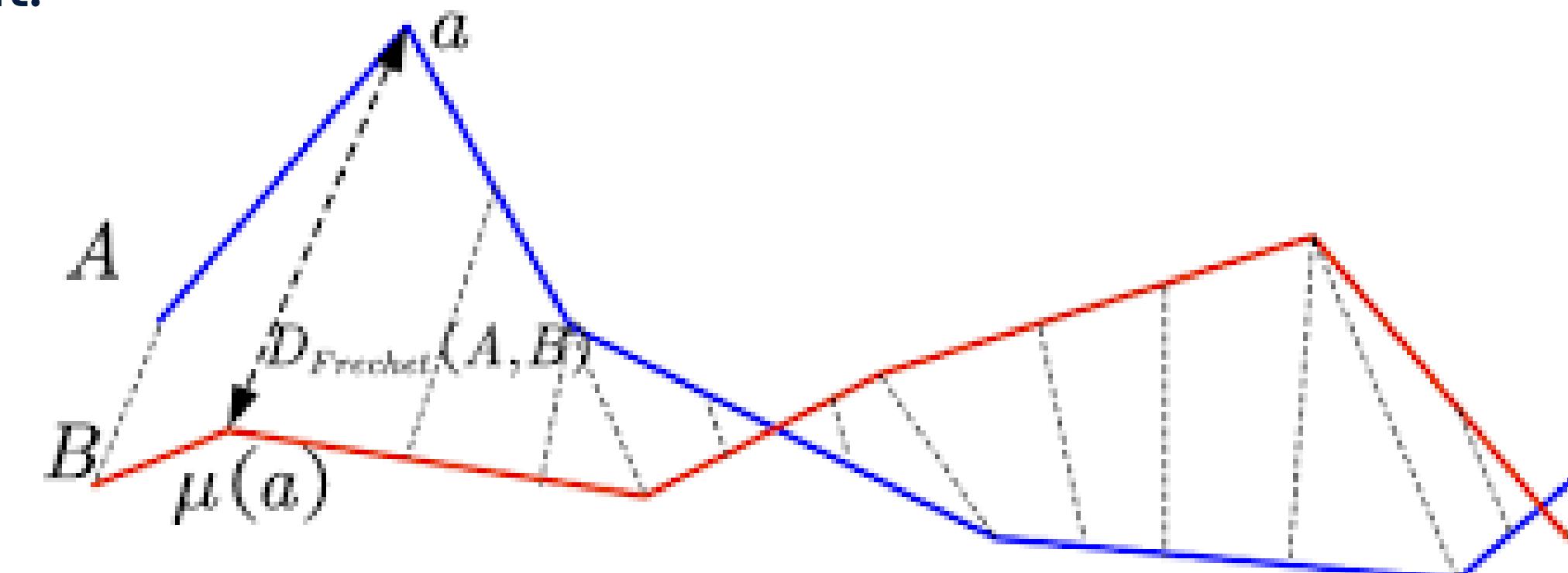
Métrique de similarité : Area Method (Méthode de l'aire)

L'idée principale est de quantifier l'aire entre deux courbes ou deux trajectoires. Plus cette aire est petite, plus les courbes sont similaires. Inversement, une aire plus grande indique une différence plus marquée entre les deux trajectoires.



Les métriques de similarité : Discrete Frechet distance

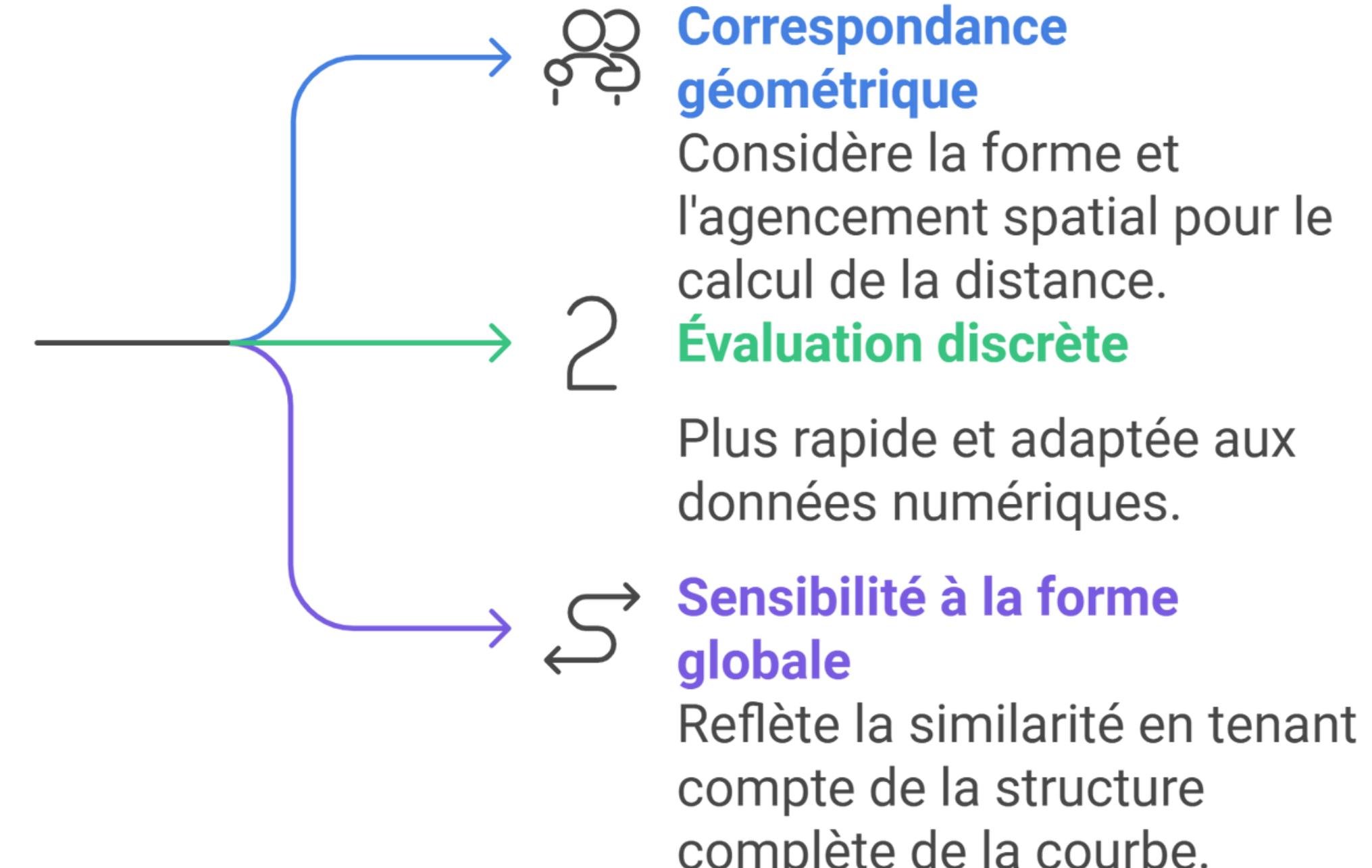
La distance de Fréchet discrète est une métrique utilisée pour mesurer la similarité entre deux courbes ou trajectoires. Contrairement à d'autres distances (comme DTW ou Euclidienne), elle prend en compte à la fois la forme des courbes et l'ordre des points. Elle est souvent utilisée pour comparer des trajectoires géométriques, comme les chemins suivis par des objets en mouvement.



Les métriques de similarité : Discrete Frechet distance



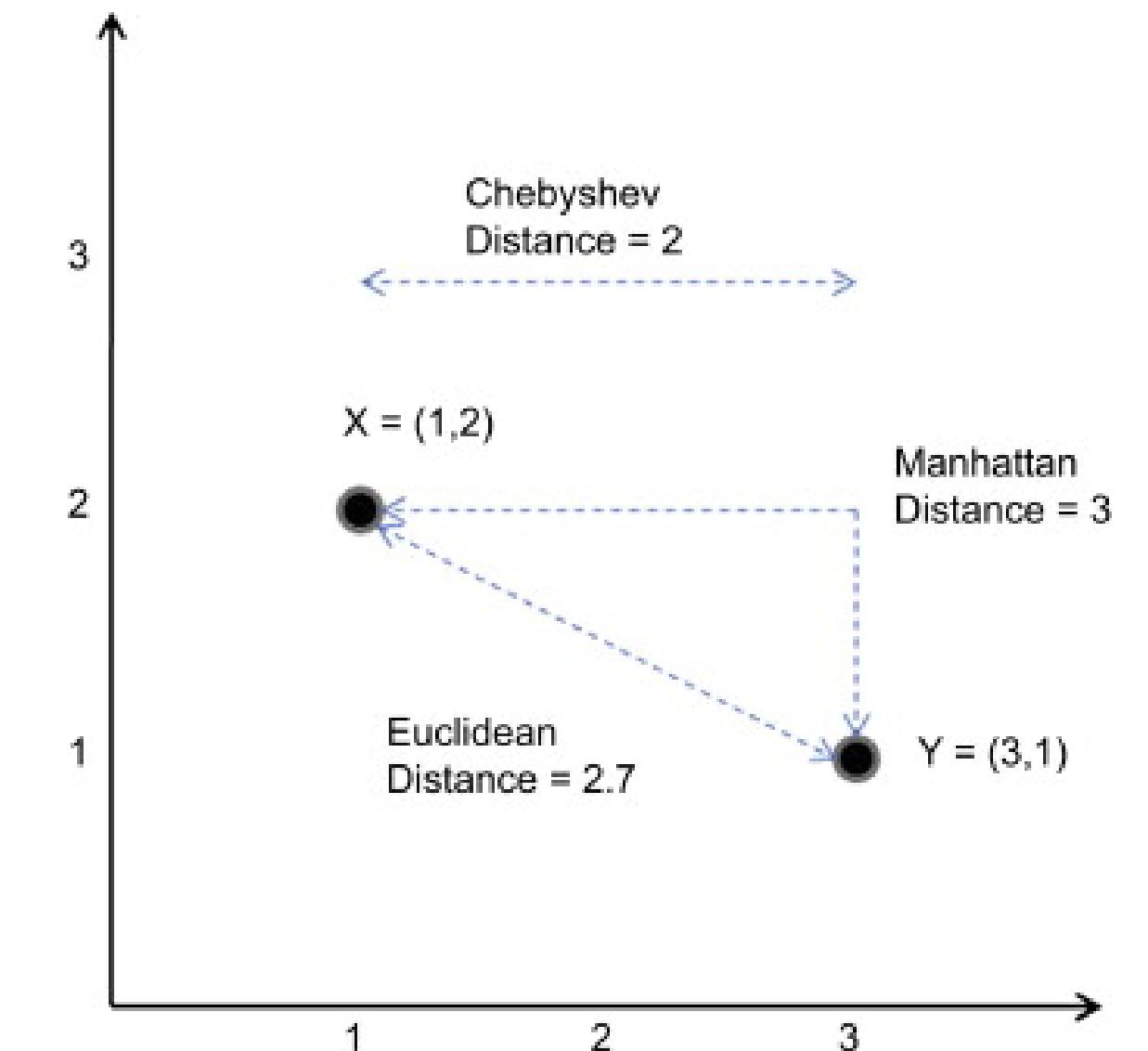
Quelle méthode de correspondance géométrique devrait être utilisée ?



Les métriques de similarité : Distance de Chebyshev

La mesure de similarité de Chebyshev est une métrique utilisée pour calculer la distance entre deux points dans un espace multidimensionnel .elle est définie comme la différence maximale entre les coordonnées de deux points le long d'un seul axe. Mathématiquement, pour deux points

$$d_{\text{Chebyshev}}(P, Q) = \max_i(|p_i - q_i|)$$



Métriques de similarité : MAE et MSE

Mean Absolute Error (MAE)

Le MAE mesure l'erreur moyenne entre les points correspondants de deux ensembles. C'est une métrique simple et intuitive qui indique la magnitude moyenne des erreurs, indépendamment de leur direction (positive ou négative).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Mean Squared Error (MSE)

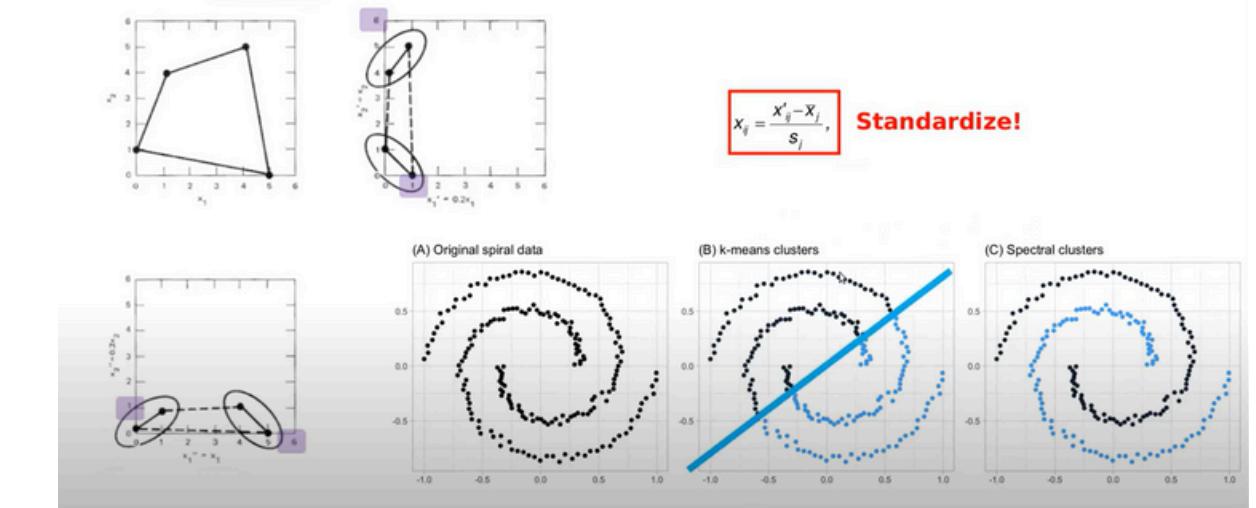
Le MSE mesure l'erreur quadratique moyenne entre deux ensembles. Contrairement au MAE, cette métrique punit davantage les grandes erreurs, car les écarts sont mis au carré.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Tableau comparatif des métriques de similarité disponibles dans similaritymeasures

Métrique	Description	Prend en compte l'ordre des points	Robuste aux anomalies	Sensibilité aux grandes erreurs	Complexité calculatoire	Applications courantes
Area Between Curves	Mesure l'aire entre deux courbes en sommant les écarts verticaux sur toute leur longueur.	Non	Oui	Non	Faible	Analyse globale des écarts entre trajectoires.
Frechet Distance	Distance minimale pour "parcourir" les deux courbes en suivant leur ordre, sans revenir en arrière.	Oui	Modérée	Non	Élevée	Comparaison de trajectoires continues, analyse géographique.
Discrete Frechet Distance	Version discrète de la distance de Frechet, calculée entre des points spécifiques.	Oui	Modérée	Non	Moyenne	Analyse rapide de trajectoires discrètes.
Dynamic Time Warping (DTW)	Trouve le meilleur alignement entre deux courbes en considérant des décalages temporels.	Oui	Modérée	Non	Élevée	Séries temporelles, reconnaissance vocale, biométrie.
Curve Length	Compare les longueurs absolues des deux courbes.	Non	Oui	Non	Faible	Évaluation des écarts de taille entre trajectoires.
Partial Curve Mapping (PCM)	Compare les parties correspondantes de deux courbes pour évaluer leur similarité locale.	Oui	Modérée	Non	Moyenne	Comparaison segmentée ou locale de trajectoires.
Mean Absolute Error (MAE)	Moyenne des différences absolues entre les points correspondants des deux courbes.	Non	Oui	Non	Faible	Validation de modèles simples, analyse d'écarts globaux.
Mean Squared Error (MSE)	Moyenne des carrés des différences entre les points correspondants des deux courbes.	Non	Non	Oui	Faible	Sensibilité aux grandes erreurs, évaluation fine.

Un exemple pratiques avec du code Python



```

import numpy as np
import similaritymeasures
import matplotlib.pyplot as plt

# Generate random experimental data
x = np.random.random(100)
y = np.random.random(100)
exp_data = np.zeros((100, 2))
exp_data[:, 0] = x
exp_data[:, 1] = y

# Generate random numerical data
x = np.random.random(100)
y = np.random.random(100)
num_data = np.zeros((100, 2))
num_data[:, 0] = x
num_data[:, 1] = y

# quantify the difference between the two curves using PCM
pcm = similaritymeasures.pcm(exp_data, num_data)

# quantify the difference between the two curves using
# Discrete Frechet distance
df = similaritymeasures.frechet_dist(exp_data, num_data)

# quantify the difference between the two curves using
# area between two curves
area = similaritymeasures.area_between_two_curves(exp_data, num_data)

# quantify the difference between the two curves using
# Curve Length based similarity measure

```

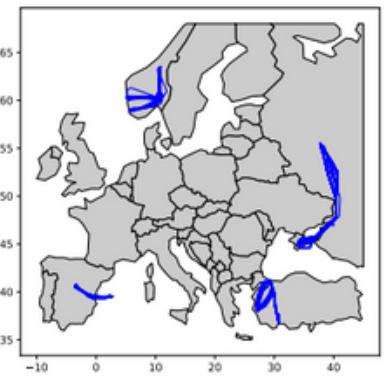
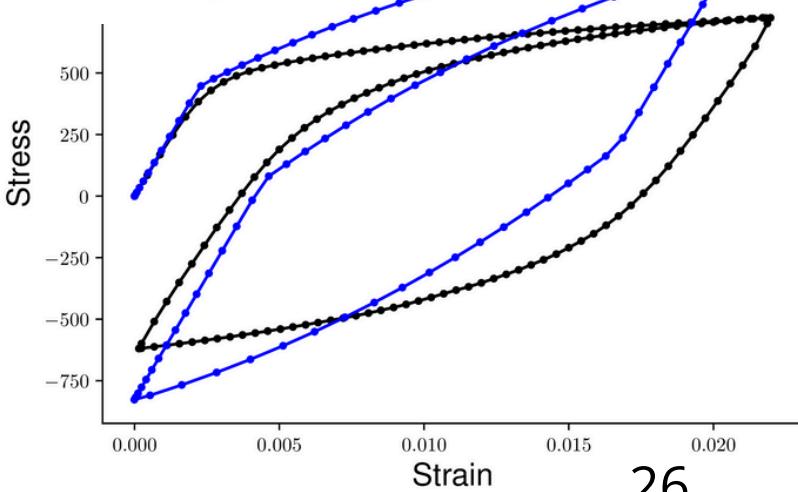


Fig. 1. Trajectories of selected commercial flights.



conclusion

Le package `similaritymeasures` est une solution performante et intuitive pour comparer des courbes en 2D grâce à des métriques comme le DTW, la distance de Frechet ou l'aire entre courbes. Polyvalent et facile à intégrer dans des projets Python, il répond aux besoins d'analyse de trajectoires et de séries temporelles dans des domaines variés, alliant précision, robustesse et simplicité.

