

Méthodologie pour le Traitement des Données de Vol

Ce rapport présente une méthodologie en trois étapes essentielles pour le traitement des données de vol :

1. Statistiques descriptives
2. Filtrage
3. Analyse cinétique

L'objectif principal est d'optimiser la compression et la synchronisation des vols tout en minimisant la perte d'information et l'erreur de reconstruction.

Rééchantillonnage des Données

Le rééchantillonnage est une technique statistique permettant de générer de nouveaux échantillons à partir d'un jeu de données existant. Dans ce projet, il est utilisé pour équilibrer les ensembles de données et améliorer la cohérence temporelle des vols.

Paramètres utilisés :

- ds (Opset) : Jeu de données issu de la toolbox Tabata.
- N : Taille du rééchantillonnage.

Analyse Préliminaire des Variables des Vols

Avant d'appliquer les techniques de compression, il est essentiel d'étudier les caractéristiques des variables des vols. Nous avons analysé :

- La fréquence d'échantillonnage de chaque variable.
- La détermination d'une référence temporelle unique ou l'estimation d'un alignement optimal.
- Les méthodes d'alignement possibles :
 - Interpolation
 - Dynamic Time Warping (DTW)
 - Rééchantillonnage uniforme

Évaluation des Performances

Pour évaluer la qualité de reconstruction des vols, nous avons utilisé trois métriques d'erreur :

- MSE (Mean Squared Error)
- RMSE (Root Mean Squared Error)
- MAE (Mean Absolute Error)

Nous avons ensuite comparé ces résultats à une baseline simple et étudié l'impact sur la synchronisation après reconstruction.

Implémentation et Scripts Python

Nous avons utilisé la toolbox Tabata pour le chargement et le prétraitement efficace des données. Un pipeline reproductible a été développé, intégrant :

- Le prétraitement des données
- La compression
- L'évaluation des performances

Interpolation Frustrée et Alignement Temporel

L'interpolation frustrée est une méthode simplifiée permettant de reconstruire un signal à partir d'un sous-échantillonnage partiel tout en minimisant la perte d'information.

Calculs et Implémentation en Python

Nous avons développé plusieurs fonctions Python pour effectuer les calculs suivants :

1. Calcul de la distance DTW et du chemin d'alignement

- `calculate_dtw()` : compare deux séries temporelles et retourne la distance DTW ainsi que le chemin d'alignement.
- Possibilité d'inverser l'ordre des séries si nécessaire.
- Utilisation de la bibliothèque `dtadistance` pour le calcul du DTW.

2. Alignement Temporel Basé sur DTW

- `align_with_dtw()` : réaligne la série cible en suivant le chemin d'alignement DTW.
- Application des méthodes `forward fill` et `backward fill` pour compléter les valeurs manquantes.

3. Visualisation de l'Alignement

- `plot_dtw_alignment()` : compare la série de référence, la cible initiale et la cible alignée après DTW.

Détection et Synchronisation des Phases de Vol

Le code développé permet de détecter les phases de vol et de synchroniser les vols en rééchantillonnant les données sur des segments identifiés.

- Détection des changements non linéaires grâce à l'utilisation de ruptures avec un modèle RBF.
- Visualisation des phases détectées pour valider la cohérence de l'alignement des vols.

Conclusion et Perspectives

Ce projet propose une approche structurée pour l'analyse et la compression des données de vol. L'application du DTW et du rééchantillonnage a permis d'améliorer la synchronisation et la précision des reconstructions.