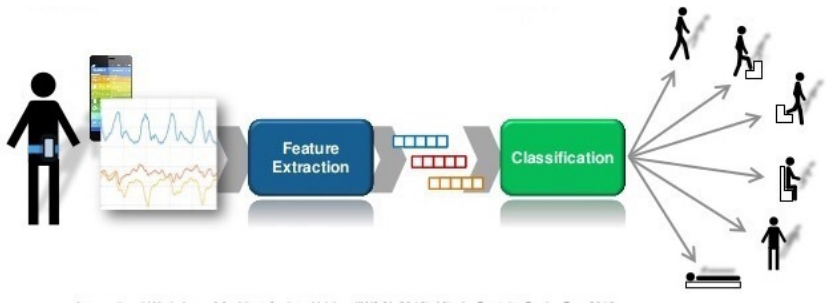


Projet Fil Rouge :

Détection d'activités humaines via capteur smartphone

N8EAN04 : Analyse de données et classification

Sandrine MOUYSET, Sadok JERAD



Objectif

Prédire l'activité humaine en utilisant les capteurs d'accéléromètre du smartphone.

(1-Descendre, 2-Courir, 3-Assis, 4-Debout, 5-Monter ou 6-Marcher)

Il est fourni pour chaque enregistrement de l'ensemble de données suivant :

- L'accélération triaxiale de l'accéléromètre (accélération totale)
- Son étiquette d'activité.
- Un identifiant du sujet qui a réalisé l'expérience.

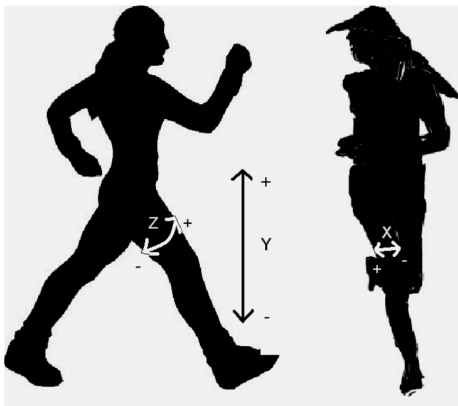
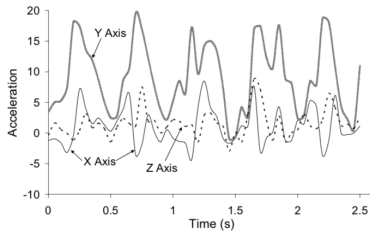
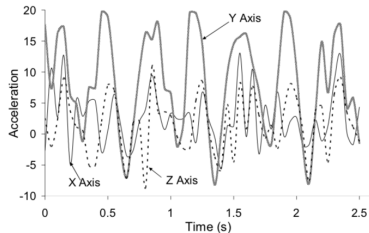


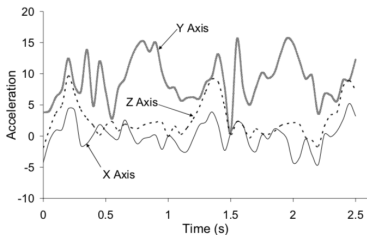
Figure 1: Axes of Motion Relative to User



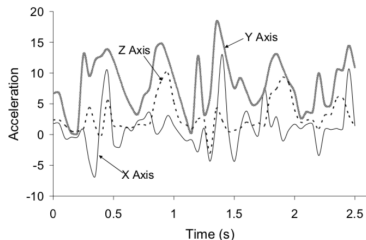
(a) Walking



(b) Jogging



(c) Ascending Stairs



(d) Descending Stairs

Organisation :

Ce projet en 6 séances se décomposent en 5 parties :

- *Partie I* : Régression logistique et différentes pénélisation associées.
- *Partie II* : Classification à l'aide des méthodes à noyau.
- *Partie III* : Classification par Méthodes d'Ensemble.
- *Partie IV* : Classification par réseaux de neurones.
- *Partie V* : Votre étude.

Projet :

- **Langage** : Notebook Python
 - Possibilité de partager le notebook via *Google Colab*.
- Travail à réaliser **en binôme**
- **Livrables du projet** : un notebook par binôme et un rapport au format pdf de 10 pages max.
- **Date limite de rendu : le 25 Juin !**

Données Brutes sous forme temporelle.

→ Traitement en échelle fréquentielle possible avec la transformation de Fourier en lien avec le cours d'analyse Hilbertienne.

Un effort sur le prétraitement des données doit être visible dans votre compte rendu.

① Utilisation de la Régression logistique pour la classification supervisée

On va chercher à appliquer l'algorithme vu en cours en variant les différents hyperparamètres (optimiseurs/ puissance de la régularisation) afin d'obtenir une meilleur performance en utilisant les fonctions *confusion_matrix*, *accuracy_score* et *GridSearchCV* de la librairie python *scikit-learn*.

② Ajout de la pénalisation en norme ℓ_1

Validation croisée pour choisir le meilleur mix entre Lasso et Ridge et le coefficient de régularisation approprié.

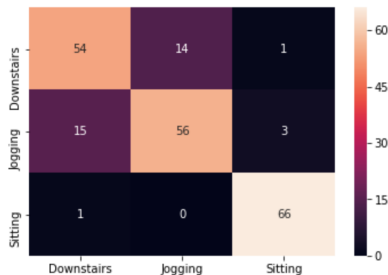
Matrice de confusion : consiste à compter le nombre de fois où des observations de la classe A ont été rangées dans la classe B .

- Chaque ligne de la matrice de confusion représente la *classe réelle* tandis que chaque colonne représente une *classe prédite*.
- Les éléments diagonaux représenteront le nombre d'éléments bien classés.

Classification parfaite = matrice de confusion diagonale

- Les éléments hors diagonaux représenteront les erreurs de classification.

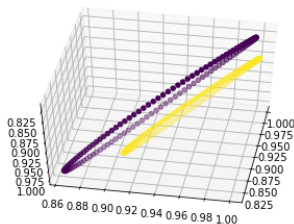
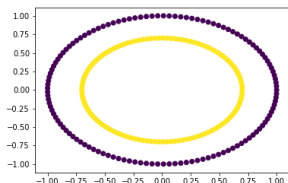
Illustration



Si on souhaite connaître le nombre de fois où le classifieur a pris des activités Jogging pour l'activité Sitting, on regardera l'élément hors diagonal (2,3) dans la matrice de confusion.

Utilisation de la librairie *scikit learn* de Python avec les kernels prédéfinis (mais vous pouvez essayer vos propres noyaux.)

- Classification par SVM.
- Clustering Spectrale.



Exemple de séparation par méthode à noyau Les données brutes ne sont pas séparables linéairement. Grâce au plongement dans le RKHS, Elles deviennent linéairement séparables.

- **Parties III & IV** : Classification par **forêts aléatoires**, **boosting** et **réseaux de neurones** ; **Pour le mois de Juin**.
- **Partie V** : Réalisez **votre propre étude**, par exemple, en :
 - en augmentant les bases de test et/ou d'apprentissage et/ou la durée des segments,
 - en proposant des variantes des méthodes proposées et/ou en testant d'autres méthodes de classification,
 - en rajoutant davantage de classes (jusqu'à 3 supplémentaires : 4-Debout, 5-Monter ou 6-Marcher),

le tout en testant les approches (parties I à IV) et en interprétant les résultats via les mesures d'évaluation (matrice de confusion et pourcentage de données bien classées) et synthétiser votre étude dans un rapport (10 pages en pdf).