

TP – Reconnaissance de configurations manuelles

Python et Scikit-Learn

S. Gibet & L. Naert

L'objectif de ce TP est de travailler sur des données de mouvements des mains (données de MoCap) et de faire de la reconnaissance de configurations manuelles sur chaque frame de ces données. Vous pourrez utiliser tous les outils et méthodes présentés dans les TPs/cours précédents. Vous utiliserez la librairie Python Scikit-Learn dans l'environnement de programmation Anaconda.

Les données exploitées ici sont celles fournies par Lucie Naert (actuellement en thèse dans l'équipe Expression). Ces données ont fait l'objet d'un protocole expérimental précis, d'enregistrements en salle MoCap, de post-traitements et d'une segmentation manuelle. Elles sont la propriété du laboratoire et ne doivent en aucun cas être diffusées en dehors de ce cours.

Capture et traitement des données

Vous travaillerez sur des données de mouvements des mains représentées par les vecteurs de position des articulations des mains dans l'espace suivant 3 coordonnées (x, y et z). Nous avons défini 26 articulations par main (5 articulations par doigts + l'articulation du poignet, cf. figure 3). Chaque vecteur contient ainsi $3 * 26 = 78$ valeurs réelles correspondant chacune à une position suivant un axe pour une articulation.

Pour obtenir les positions des articulations à partir des données de MoCap (qui donnent les positions des marqueurs et non des articulations), nous avons traité nos données en plusieurs étapes (cf. figure 1) :

- Étiquetage des trajectoires des marqueurs : il s'agit d'attribuer un identifiant de marqueur à chaque point repéré par les caméras pour chaque frame.
- Remplissage des "trous" dans les trajectoires : les caméras peuvent subir des occlusions et certains marqueurs peuvent ne pas être repérés par les caméras à un instant donné.
- Filtrage (passe-bas) des données pour retirer les micros mouvements induits par le système de capture.
- Définition d'un squelette en fonction de la position des marqueurs.
- Calcul des positions et orientations des articulations de ce squelette en fonction des données de MoCap (format de fichier de type bvh ou fbx).
- Segmentation et annotation des données. Les données de mouvement sont divisées en segments auxquels des étiquettes sont attribuées. Les annotations sont réalisées à l'aide d'un logiciel spécialisé et selon une nomenclature bien définie.

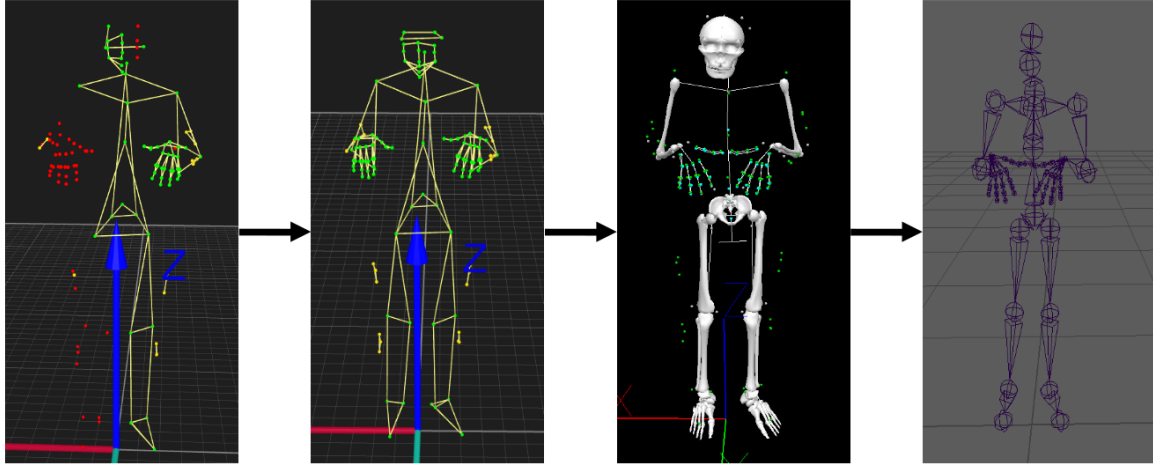


FIGURE 1 – Chaîne de traitement : de l'identification des marqueurs au calcul des positions/orientations des articulations en bvh.

Configuration des mains

Une configuration de la main correspond à une certaine disposition des doigts à un moment donné. La configuration '5', par exemple, correspond à un état de la main dans lequel tous les doigts sont tendus et écartés (faisant le chiffre 5) tandis que la configuration 'V' sera faite en tendant et écartant l'index et le majeur (les autres doigts étant pliés). Nous avons répertorié 32 classes de configurations différentes. Celles-ci sont listées et illustrées sur la figure 2.

Données

Nous avons extrait 600 configurations manuelles appartenant à ces 32 classes en nous basant sur l'annotation du corpus. Ces configurations manuelles ont été exécutées durant la réalisation de phrases (aussi appelées "énoncés") en Langue des Signes Française. Un fichier de type .csv correspondant à chacune de ces configurations a été créé. Chaque fichier suit le même schéma : la première ligne est un en-tête contenant la liste ordonnée des noms des articulations et d'un axe (x, y, ou z), les lignes suivantes décrivent la position des articulations pour chaque frame. Chaque ligne (frame) est composée des 78 valeurs de position. Un fichier .csv correspond donc à un ensemble de n frames consécutives auxquelles on a assigné la même étiquette.

Exemple du contenu d'un fichier pour $n = 3$ avec 2 articulations (poignet et première articulation de l'auriculaire) :

```

1 Left_Hand_x Left_Hand_y Left_Hand_z Left_Pinky0_x Left_Pinky0_y Left_Pinky0_z
2 27 1.30285e-06 1.33303e-06 28.1445 2.15521 -2.85254
3 27 -1.24972e-06 8.43971e-07 28.1407 2.13847 -2.86662
4 27 -2.384e-06 -6.85626e-06 28.147 2.11799 -2.87928
```

Pour récupérer les données, il faut lire les fichiers contenus dans le dossier PositionHand-Joints (téléchargement de "MoCap Data" sous moodle). Les fichiers sont nommés de la manière suivante : AH_Sx_fy_lz_label.csv, avec :

- AH = LH ou RH en fonction de la main dont il est question : gauche (LH = Left Hand) ou droite (RH = Right Hand).

- x = numéro de la séquence de capture dont est extraite la configuration.
- y = frame de début.
- z = longueur de la séquence.
- label = label du segment.

Exemple

LH_S1_f6827_l61_Oui.csv : Données de la main gauche (LH) extraites de la séquence de mouvement "S1" commençant à la frame n°6827 d'une longueur de 61 frames et étiquetées "Oui" (appartenant à la classe 'Oui' selon les annotateurs).

1. Écrivez une/des fonction(s) ou une classe pour lire et stocker le contenu des fichiers du dossier *PositionHandJoints* dans une variable. Vous pourrez vous servir la librairie *glob* (import *glob*) qui permet d'utiliser des expressions régulières pour trouver des fichiers et les fonctions *strip* et *split* qui permettent de transformer des chaînes de caractères.

Apprentissage automatique pour les données de configuration des mains

Nous allons travailler sur les données de position des articulations des mains pour traiter un problème de classification supervisée. Nous cherchons à entraîner des classifieurs de façon à reconnaître la configuration manuelle réalisée **pendant une frame**.

La définition des tâches de reconnaissance servira à l'évaluation du TP. On peut en effet proposer des tâches différentes en tenant compte de différents critères, comme par exemple :

- le choix de la ou des main(s) : LH, RH, ou les deux ;
- le choix des *features* : toutes les distances, tous les angles, un sous-ensemble pertinent, des distances **et** des angles... ;
- le choix des ensembles d'apprentissage et de test ;
- l'utilisation de différentes méthodes de classification supervisée vues en cours et TPs
- la visualisation des résultats (différentes métriques, matrices de confusion, courbe ROC)

Bien sûr vous ne testerez pas toute les combinaisons possibles de ces tâches, vous choisirez celles qui vous paraissent les plus pertinentes !

2. **Statistiques sur les données** – Vous définirez vos propres critères pour donner une représentation statistique des données : proportion des différentes configurations dans le corpus de données, combien de segments d'une certaine classe ? Combien de frames d'un certain type ? La main gauche et la main droite sont-elles équitablement représentées ? ... Cela permettra de construire des ensembles d'apprentissage et de test équilibrés.
3. **Détermination des caractéristiques (*features*) de classification** – Les positions absolues des articulations dans l'espace ne permettent pas, telles quelles, de distinguer les configurations de la main. En effet, deux configurations identiques peuvent être réalisées à deux endroits différents de l'espace. Il faut donc réfléchir à un ensemble de caractéristiques calculables à partir de ces positions qui sera adapté à la tâche de classification. Les distances entre les articulations et/ou les angles formés par ces articulations peuvent être utilisés. Il peut être intéressant de réfléchir à des (sous-)ensembles pertinents de ces *features* (se référer à la figure 3 pour avoir le squelette de la main et la position des articulations).

Les questions suivantes (questions 4 à 8) sont à réaliser pour chaque ensemble de caractéristiques :

4. **Construction des matrices X et y en fonction du set de features choisi** – La matrice X sera de taille `nbFeatures * nbFrames` et contiendra les valeurs des features choisis pour l'ensemble des frames considérées pour la classification et le vecteur y les étiquettes (labels) de chaque frame. Dans le cas où les *features* choisis seraient de nature différentes (angles et distances par exemple), il peut être nécessaire de normaliser les valeurs de la matrices X .
5. **Construction des ensembles d'apprentissage et de test** – Vous réfléchirez à une répartition des données entre l'ensemble d'apprentissage et l'ensemble de test (une manière classique consiste à réserver 80% des données pour la partie apprentissage et 20% pour l'évaluation des classifieurs). Il n'est évidemment pas raisonnable d'entraîner un modèle sur 100% des données et de faire les tests sur ces mêmes données. Les individus pris pour le test seront déjà connus par le classifieur et le test ne sera pas valable.
6. **Classification** – Utiliser des algorithmes de classification (e.g. régression logistique, K-NN, Naive Bayes, SVM) entraînés sur le *training set* pour reconnaître les configurations réalisées sur le *test set*.


```
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
```
7. **Mesures de performance numériques pour la classification** – Calculez les valeurs d'accuracy, de precision, de recall, de F-score, ... en comparant les configurations prédites avec les configurations annotées.
8. **Mesures de performance graphiques** – Affichez les matrices de confusion, courbes ROC, ...
9. **Discussion sur les résultats** – Vous rappellerez les résultats et conclurez sur la capacité de chaque classifieur, pour chaque jeu de données, de bien discriminer les configurations manuelles. Quelles sont les configurations les moins bien reconnues ? Pour quel classifieur ?
10. **Question Bonus** : Pendant le TP, nous avons fait de la reconnaissance de configuration manuelle pour chaque frame. Or, chaque fichier .csv correspond à un segment étiqueté (= ensemble de frames consécutives avec la même configuration). Comment feriez vous pour faire de la reconnaissance par segment ? Proposez une solution et testez-la sur les données.

Suite à ce TP, vous devrez rendre une archive (*NOM_Prenom_TPReco*) composée :

- du code python du TP,
- d'un README indiquant comment faire fonctionner votre code (chemin vers les données, modifications éventuelles à apporter...), et
- du compte-rendu du TP récapitulant vos choix, vos résultats et vos conclusions.



FIGURE 2 – Classes de configuration présentes dans les données.

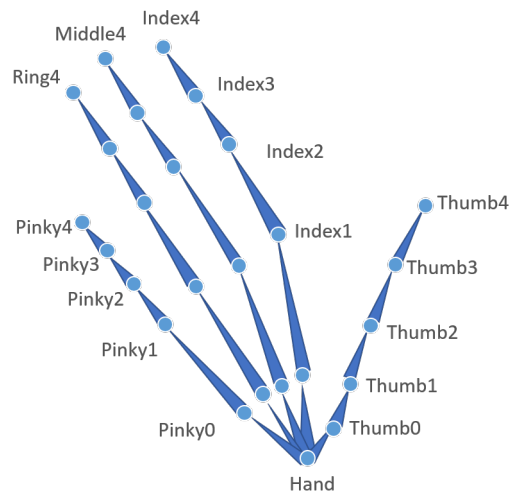


FIGURE 3 – Squelette de la main : position des articulations.