GENERATIVE AI RECREATING FOOTBALL GAME

Mattéo Hochart

Data analysis

When running main.py, the data analysis is plotted with matplotlib.

The duration of each action is pretty similar for all the actions (mean value between 0.6 and 1.1 s) except ‘no action’ and ‘rest’ (mean values of 2.7 and 3.6 s) which are not real football actions.

With the norm mean analysis we regroup the actions in 3 groups:

- Low intensity: no action, rest & walk
- Medium intensity: pass, run, tackle & dribble
- High intensity: cross & shot

There is a consecutiveness of the action for dribble, walk and run, that are not instantaneous in opposition with shot, tackle, cross and shot.

Game recreation approaches

Several approaches are conceivable:

- Comparing the action of each player at each time step and only keeping the most impacting action on the game for the game recreation. For example if at the same time player 1 is running, player 2 shooting, player 3 running and player 4 Walking, we will keep the action ‘shot’. This is the approach that is used for the test.

- We could also return more information, like which player is doing the action returned.

- It is also possible to regroup actions to have a simpler game recreation (if you only consider ‘shot’, ‘pass’ (pass & cross), ‘dribble’, ‘tackle’ and ‘no_ball’ (no action, run, walk & rest).

- With other data like the position of the players on the field, we could have a more detailed game recreation. For example every pass could be more or less dangerous.

Game recreation description

Pre-processing part:

The pre-processing part is computed in extract.py.
The input data are stored in several JSON files, one file constituting the game of one player.
All the JSON files in the directory are loaded thanks to json and os libraries.
It is chosen to return the data in 2 different dictionnaries, one that will be used for the data analysis and one for the game recreation.
For the game recreation, we decide to stop the data acquisition of all the players when it ends for one player.

Processing part

The processing part is computed in functions.py.
The first function is computed for the action analysis by generating 3 characteristics: the duration, the norm and the consecutiveness of the action. The mean value of all gaits is computed for each characteristics. When the consecutiveness is not 0, there is a sequel 0, 1, 2, ... until the highest value. We only keep the highest value when there is such a sequel.

The second function is computed for the game generation. It is chosen to compare for each time step the actions of all players and chose the “most important” one for the game recreation. The order of importance is:

- offensive action (cross, shot or pass)
- defensive action (tackle)
- action without the ball (run, walk, rest or no action)

If 2 players or more are doing the same action the one with the highest norm is chosen.
The game is recreated for each time step and then regrouped when several consecutive actions are either from the same player and the same action or from different players but the same action without the ball. The exceptions are managed by replacing an action (except for ‘rest’ and ‘no action’) longer than 2s or shorter than 0.2s

Post-processing part

The post-processing part is computed in display.py. The action analysis is displayed with matplotlib in bars characteristic by characteristic.
The game recreation is loaded as a dictionnary {gait_label: [norms list], gait_label: [norms list], ...} and saved in the file: game_recreation.json.