TIW 9 TP Patterns

Groupe :

- Maxime Bunel p1914012
- Julien Giraud p1704709
- Diogenes Molinares p2019196
- Jérémy Thomas p1702137

Définitions

- Coup : service ou tir, action représentée par une ligne dans le fichier csv fourni.
- Échange : liste des coups échangés jusqu'à l'obtention d'un point par l'un des deux joueurs.

Normalisation des données

Nous avons choisi de découper les données en deux sous-ensembles:

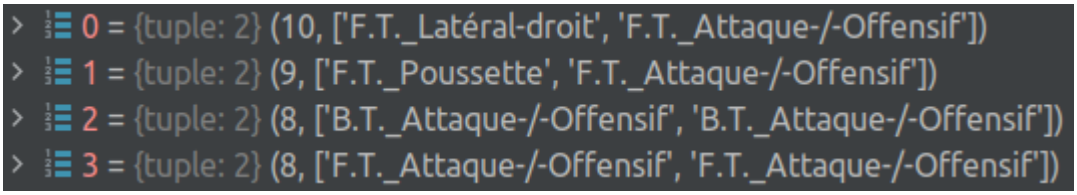
- Le sous-ensemble des échanges gagnants qui contient les coups du joueur qui a gagné le point.
- Le sous-ensemble des échanges perdants qui contient les coups du joueur qui a perdu le point.

Pour déterminer si un coup est gagnant ou perdant, on regarde la différence de score avec l'échange précédent.

Chaque sous-ensemble fournit une liste de coups pour chaque échange du match. Ces listes contiennent une représentation de chaque coup suivant les colonnes sélectionnées.

Pour se conformer à l'algorithme Prefix Span qui compare une liste dont les données sont de même type, nous avons choisi de concaténer les données de chaque colonnes afin d'obtenir une liste de chaîne de caractères.

Par exemple, en sélectionnant les colonnes “Type” et "Coup", on obtient la figure ci-contre pour les 4 premiers échanges gagnants :



Ces données sont à lire de la manière suivante : Le joueur F.T. a gagné 10 points grâce à l'enchainement de coups de type latéral/droit puis attaque/offensif.

Recherche de patterns dans les données

Pour rechercher des patterns dans les données, nous avons utilisé l'algorithme Prefix Span dans le but de trouver les suites d'actions (patterns) les plus fréquentes chez les gagnants et chez les perdants.

Nous avons réalisé un programme pour rechercher ces patterns avec différents paramètres :

- la liste des critères à utiliser parmi les suivants : latéralité, coup utilisé, type de coup et zones de jeu,
- le nombre minimum d'actions par pattern (minlen),
- le nombre de séquences à rechercher (k).

Lancement du code

Notre programme fonctionne sous la forme d'un outil en ligne de commande. Il faut lui fournir les paramètres cités précédemment ainsi que le fichier à traiter. Il affiche le résultat dans le terminal au format JSON, il est possible de l'exporter dans un fichier pour utiliser un outil de visualisation de JSON. Le programme se lance avec Python3

```
$ python3 main.py <file> <minlen> <k> <useLaterabilite> <useCoup> <useZone>
<useType>
```

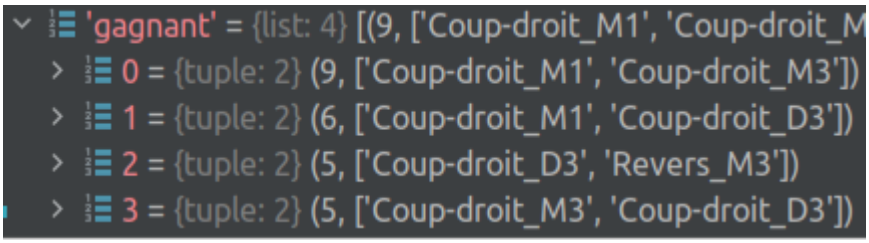
Analyse des résultats

Exemple 1

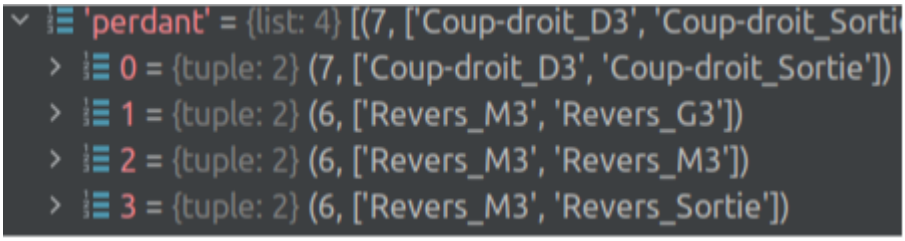
```
$ minlen=2 k=4 useLateralite=True useCoup=False, useZone=True,
useType=False
$ python3 main.py $file $minlen $k $useLaterabilite $useCoup $useZone
$useType > results_example_1.json
```

Si on utilise les colonnes "Latéralité" et "Zone" on obtient les résultats suivants :

Sous-ensemble des coups du point gagnant :



Sous-ensemble des coups du point perdant :



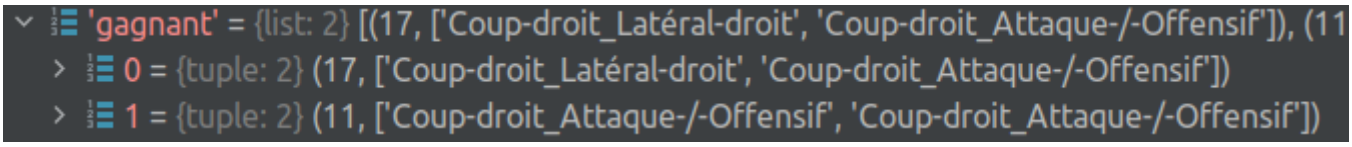
Dans cet exemple, l'enchainement d'un coup droit en M1, puis d'un coup droit en M3 a mené 9 fois au gain d'un point. En revanche, l'enchainement de deux coups droits a mené à la perte du point.

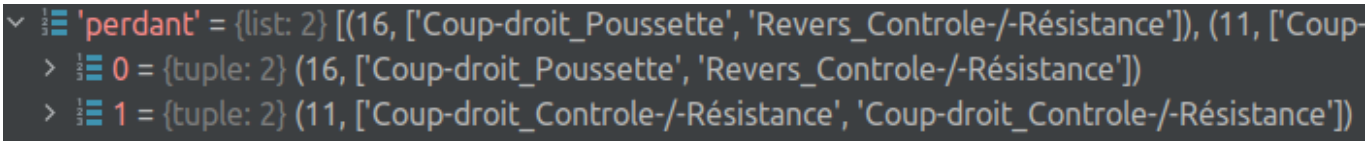
Du sous-ensemble des points gagnants, on peut déduire qu'un coup droit en M1 puis un coup droit en M3/D3 permet de gagner le point, en effet un coup court puis long permet généralement de déstabiliser l'adversaire. On peut également déduire des résultats 2 et 3 qu'un coup D3 puis M3 ou l'inverse permet de gagner le point. En effet, varier un coup de gauche à droite et inversement fatigue l'adversaire.

Du sous ensemble perdant, on remarque globalement qu'un point est perdu lorsque l'on utilise consécutivement plusieurs fois son revers.

Exemple 2

```
$ minlen=2 k=4 useLateralite=True useCoup=False, useZone=False,
useType=True
$ python3 main.py $file $minlen $k $useLaterabilite $useCoup $useZone
$useType > results_example_2.json
```





Pour les points perdants, on remarque que les coups sont généralement défensif ("résistance").

On remarque que l'on a généralement plus de chance de gagner le point lorsque que l'on effectue deux coups droits offensifs ou latéraux.

Conclusion

Nous avons pu déduire des données qu'une stratégie gagnante est d'alterner entre en avant et arrière ou gauche et droite. On voit également que le revers mène plus souvent à la défaite. Ces conclusions semblent logiques étant donnée le jeu du ping-pong.