

Rapport IA : TP3

Antoine LEVAL, Erwan IQUEL

March 30, 2018

Abstract

Ce TP consistait à faire du Data Mining sur les données HearthStone. Vous pouvez retrouver la version finale du TP sur ce dépôt Github : https://github.com/Viinyard/IA_TP3

1 Mis en place

Pour construire le projet il faut au préalable ajouter la librairie fournis dans le répertoire lib : "spmfm.jar" dans votre repository maven, celui-ci n'étant pas disponible sur les repository en ligne.

Placez vous à la racine de projet et lancez la commande suivante :

```
mvn install:install-file -Dfile=spmfm.jar -DgroupId=com.philippe-fournier-viger.spmfm
-DartifactId=spmfm -Dversion=1.0 -Dpackaging=jar -DlocalRepositoryPath=lib/
```

Vous pouvez maintenant construire le projet avec la commande

```
mvn clean package
```

Vous pouvez maintenant tester le programme :

```
ava -jar target/tp3-1.0-SNAPSHOT-jar-with-dependencies.jar -h
usage: [-h] | -m <method> -s <files> [-o <file>] [-r <file>] [-f]
-c,--conf <arg>      minconf, default 50
-f,--fullname        Print the full name of card instead of number
-m,--method <arg>    method to use, must be one of :
                    - A : LCM
                    - B : CloSpan
                    - C : ERMiner
                    - D : RuleGrowth
-o,--output <arg>    output file, default : output.txt
-r,--result <arg>    Result of the computation algorithm, default :
                    result.txt
-s,--source <arg>    source file, required
-t,--minsup <arg>    Minsup Threshold, in percent
```

Les options -m et -s sont requise.

- -m peut avoir comme argument A ou B, respectivement pour la partie A ou la partie B.
- -s est le fichier de donnée dans le dossier data à fournir, possibilité de les chaîner.
- -r est le fichier de sortie du resultat de l'algorithme.
- -o est le fichier de sortie du resultat du traitement.
- -f ou -fullname est à renseigner si vous souhaitez avoir après le traitement par l'algorithme le vrai nom des cartes hearthstone plutot que des indices.

Le process se fait donc en un coup, mais on garde une trace du processus. Par exemple avec le fichier output.txt nous avons toujours la possibilité d'utiliser l'algorithme comme ceci :

```
java -jar spmfm.jar run LCM output.txt result.txt 20%
```

2 Deck Type

Le but de ce traitement est de mettre en évidence les configuration de decks les plus souvent joué par les joueurs.

En observant toute les cartes joué par un joueur durant une partie on peut faire une liste des cartes présentes dans son deck, cette liste n'est dans la plupart des cas pas exhaustive car il ne va pas forcément jouer toute les cartes de son deck au cours de la partie.

Pour observer les cartes les plus souvent jouer dans un même deck il est intéressant de regarder des itemsets assez gros, par exemple :

```
L6
pattern 926: LesserHeal, PowerWord:Shield, WyrmmrestAgent, NorthshireCleric, HolyNova,
          TwilightGuardian, support : 15
L7
pattern 929: TwilightWhelp, LesserHeal, PowerWord:Shield, WyrmmrestAgent,
          NorthshireCleric, HolyNova, TwilightGuardian, support : 14
```

Ce sont des combinaisons qui ont été associé 15 et 14 fois dans un même deck au cours des parties analysées.

Avec un minsup supérieur on peut obtenir

```
L1
pattern 0: TheCoin, support : 186
pattern 1: Dr.Boom, support : 88
pattern 2: PilotedShredder, support : 129
```

Et observer donc les trois cartes les plus présentes dans les decks.

3 Sequence et règles séquentielles

On peut observer avec le CloSpan les cartes les plus souvent joué à la suite, sans faire de différences entre joueur ou adversaire.

```
LesserHeal, LesserHeal, LesserHeal, LesserHeal, #SUP: 31
Fireblast, Fireblast, Fireblast, #SUP: 40
```

Il apparaît par exemple que certaines carte, une ayant l'air de servir de soigneur à en juger par le nom sont souvent joué pleins de fois à la suite, 40 étant un support assez élevé étant donné la diversité des parties.

Mais on ne peut grâce à ces donnée faire la différences entre Joueur A joue tel carte, joueur B riposte avec telle carte, pour cela il faut implémenter d'autres algorithmes comme ERMiner ou RuleGrowth.

Et grâce à ces données, on peut observer quels sont les cartes qui sont directement joué en riposter par le joueur suivant, comme par exemple :

```
java -jar target/tp3-1.0-SNAPSHOT-jar-with-dependencies.jar -s data/all_absolute+.txt
          data/all_absolute-.txt -m D -t 15 -c 30 --fullname
...
Frostbolt, ==> Fireblast, #SUP: 34 #CONF: 0.6181818181818182
```

A en juger par le nom des cartes, Frostbolt doit permettre de geler quelques chose et Fireblast doit permettre d'annuler ou de contrer cette carte par le feu. C'est un combot fréquemment joué à la suite.