IA TP2 – Data Mining sur les données HearthStone

Alexandre Termier, Vincent Drevelle – Option IA M1 Info – 2018/2019

Remerciements à Yann Dauxais et Clément Gautrais pour l'aide à la préparation de ce sujet.

Introduction

Dans ce TP, nous allons analyser des données de partie du célèbre jeu de cartes *HearthStone*, en utilisant des algorithmes de *pattern mining*. Notre but va être d'essayer d'acquérir des connaissances nouvelles et utiles grâce à notre analyse.

Le jeu *HearthStone* (http://eu.battle.net/hearthstone/fr/) est un jeu de cartes en lignes où deux magiciens s'affrontent dans un duel à mort. Chaque magicien dispose d'un *deck* de 30 cartes, d'où il tire 1 carte par tour. Les cartes tirées vont dans sa *main* (limitée à 10 cartes), et peuvent être soit des *sorts*, soit des *créatures*. Suite au paiement d'une ressource indiquée sur la carte, une créature peut être mise en jeu, ou un sort peut être lancé. Les créatures sont posées sur le *plateau de jeu*, où elles pourront attaquer soit les créatures adverses, soit le joueur adverse. Pour des règles plus complètes, référez vous à des sites comme : http://www.millenium.org/hearthstone-heroes-of-warcraft/accueil/Guide/regles-du-jeu-d-hearthstone-regles-du-jeu-de-cartes-pour-les-parties-d-hearthstone-87071?page=1

NB : une compréhension détaillée du jeu n'est pas nécessaire pour ce TP.

Nous avons mis à votre disposition un dataset de parties de jeu. Ce dataset contient pour chaque partie les cartes jouées à chaque tour par chacun des joueurs. Le format est le suivant :

Numéro de séquence/Partie, Type de cartes (M : me, O : Opponent), Tour de jeu

Par exemple un début de fichier pourrait être :

```
1 Begin 0
1 MWebspinner 1
1 OSecretkeeper 1
1 OTheCoin 1
1 ONobleSacrifice 1
1 MHauntedCreeper 2
1 OHauntedCreeper 2
```

Ce qui signifie que :

- Le joueur « Me » a joué Webspinner au tour 1
- Le joueur « Opponent » a joué SecretKeeper, TheCoin et NobleSacrifice au tour 1
- Le joueur « Me » a joué HaunterCreeper au tour 2
- Le joueur « Opponent » a joué HauntedCreeper au tour 2
- …

Nous allons dans ce TP analyser ce dataset à l'aide d'algorithmes de pattern mining. Pour cela vous aurez à :

- Transformer les données pour les mettre dans un format compatible avec l'algorithme de pattern mining utilisé. Pour cela, utilisez le langage que vous maîtrisez le mieux (ex : Java, Python…)
- Utiliser l'algorithme demandé, fourni par la bibliothèque SPMF : http://www.philippe-fournier-viger.com/spmf/. C'est à vous de la télécharger et de l'installer en suivant les instructions fournies sur le site (c'est un jar java).
- Analyser les résultats : cela peut demander des post-traitements, que vous effectuerez là encore avec le langage de votre choix.

Vous pourrez développer les outils de transformation et d'analyse dans le langage de votre choix.

Le TP se déroule sur 1 séance. Le compte-rendu sera à fournir au plus tard avant la dernière séance de TP de l'année. Votre rendu sera envoyé par email à vincent.drevelle.istic+IA@gmail.com et consistera en :

- un rapport expliquant les transformations appliquées aux données, et analysant les résultats obtenus
- le code source des outils de transformation des données et d'analyse des résultats que vous avez développés.

Le sujet comporte 2 parties : **seule la partie A correspond au travail demandé**. La partie B peut être traitée en bonus si vous avez le temps.

A. Decks type

Nous allons chercher à déterminer des sous-ensembles de « decks-type ».

Les joueurs de HearthStone passent beaucoup à chercher la composition de deck qui leur donnera le plus de succès, et de nombreux sites sont dédiés à ce sujet (par exemple : http://www.millenium.org/hearthstone-heroes-of-warcraft/accueil/guides/archetypes-de-decks-types-de-jeu-archetypes-decks-combo-cartes-fabrication-conseil-gestion-101647 ou http://www.hearthpwn.com/decks)

L'analyse de notre dataset de parties peut nous donner des éléments sur des « decks-type », grâce à la recherche d'itemsets.

Le travail à faire est le suivant :

1. Transformez le dataset de parties en un dataset de decks. Pour chaque partie nous pouvons reconstruire deux decks (un par joueur), en construisant l'ensemble des cartes jouées par chaque joueur.
   NB : ce sont des decks partiels (la partie peut se terminer avant que toutes les cartes du decks aient été jouées), ce n'est pas grave : on fait avec ce qu'on observe.
   ➔ Vous avez donc un dataset de transactions, où chaque transaction est un deck (partiel), dont les items sont les cartes.

2. Appliquez l'algorithme LCM de SPMF pour trouver des itemsets fréquents fermés.
    - L'utilisation de cet algorithme est décrite dans l'exemple 15 de la documentation de SPMF : http://www.philippe-fournier-viger.com/spmf/documentation.php#LCM
    - Vous aurez à transformer votre matrice de transaction pour qu'elle soit compatible avec le format d'entrée attendu par SPMF.
        - *Attention : SPMF demande des itemsets numériques, triés en ordre croissant, dont le premier est 1 et surtout pas 0.*
    - C'est à vous de déterminer un seuil de support minimal pertinent. *Notez que cela peut demander plusieurs essais et erreurs…*
3. Présentez une analyse succincte des résultats (min. 10 lignes, max. ½ page), en vous aidant des questions suivantes. Si vous ne connaissez pas le jeu, aidez-vous des sites web présentant des decks types (ou demandez à un joueur de votre entourage ;-))
    - Que représente un itemset fréquent fermé ici ? pourquoi est-ce potentiellement intéressant ?
    - Y a-t-il des itemsets fréquents fermés « triviaux » qui ne nous apprennent pas grand-chose ?
    - Avez-vous trouvé des itemsets fréquents particulièrement intéressants ?

B. Séquences et règles séquentielles *(facultatif, en bonus)*

Après avoir analysé les decks des joueurs pour trouver des connaissances nous aidant à construire notre deck, nous allons analyser plus précisément les parties, pour essayer de trouver des patterns de séquences de jeu, du type :

- (tour i) joueur A joue les cartes C1 et C2 *// attaque*
- (tour i+1) joueur B joue la carte C3 *// contre-attaque*

Chaque partie peut être vue comme une séquence d'actions, où une action est la pose des cartes d'un joueur pendant un tour. Nous considérerons donc que c'est une séquence d'itemsets, où chaque itemset est l'ensemble des cartes posées par un joueur pendant son tour (*NB : nous perdons volontairement l'ordre de pose des cartes pendant un tour, cette information ne nous intéresse pas ici)*.

1. Transformez votre dataset de parties au format dataset de séquences d'itemsets. Vous devez respecter le format d'entrée de l'algorithme CloSpan (http://www.philippe-fournier-viger.com/spmf/documentation.php#clospan)
2. Appliquez l'algorithme CloSpan de SPMF sur ces données.
3. Interprétez les résultats comme en partie A. Outre l'analyse de séquences fréquentes particulièrement pertinentes pour le jeu, il peut par exemple être intéressant de commenter la longueur des séquences fréquentes obtenues par rapport à leur fréquence (arrive t'on à voir des séquences fréquentes à la fois longues et ayant un haut support ?)

Si les séquences fréquentes peuvent vous montrez des « séquences de jeu » intéressantes, elles ne répondent pas à la question : *si le joueur A joue la carte C1, quelle est la carte la plus souvent jouée par le joueur B en riposte ?*
Les règles séquentielles, extension des règles d'association aux séquences, peuvent vous aider pour cette tâche.

4. Lisez le post de blog sur les règles séquentielles : http://data-mining.philippe-fournier-viger.com/introduction-to-sequential-rule-mining/
5. Reprenez vos données précédentes et appliquez-leur un algorithme d'extraction de règles séquentielles (ERMiner ou RuleGrowth).
6. Interprétez les résultats :
    - Que dire des patterns trouvés ? Quels patterns utiles trouvez-vous ? Quid des patterns triviaux ?
    - Comparez ces résultats aux patterns séquentiels précédents.
    - Concluez sur l'utilité d'avoir utilisé les règles séquentielles.