

Rapport de Projet Fouille de données : Awélé “NaiveBonobo”

Table des Matières

Table des Matières

I - Introduction

II - Préparation

1 - Recherche

2 - Idées

3 - Compréhension des sujets et des moyens à notre disposition

III - Mise en oeuvre

1 - Bot finale

IV - Conclusion

V - Références

I - Introduction

L'Awélé est un jeu traditionnel africain. C'est un jeu de société combinatoire abstrait. C'est le plus répandu des jeux de la famille *mancala*, ensemble de jeux africains de type « compter et capturer » dans lesquels on distribue des cailloux, graines ou coquillages dans des coupelles ou des trous, parfois creusés à même le sol. Ce jeu voit s'affronter deux joueurs jouant à tour de rôle, sans information cachée et sans hasard.



Le but du projet étant de faire un moteur d'intelligence artificielle permettant de jouer à l'Awélé grâce à un ou plusieurs algorithmes de fouille de données. cette algorithme permettra d'extraire les connaissances à partir d'une base de données de coups joués pour prendre la décision. Le bot sera à réaliser en langage R.

Ce bot sera confronté à plusieurs autres bots dans un championnat à l'occasion de ce projet.

Concernant les données à disposition:

Nous avons à disposition un jeu d'ensemble de données correspondant à 303 situations de jeu observées au cours de plusieurs parties entre un joueur expérimenté et un joueur novice.

Le fichier awale.R nous permet de voir s'affronter 2 bots.

II - Préparation

1 - Recherche

Nous nous sommes d'abord concentré à comprendre les règles de l'awélé et comment bien y jouer. Nous y avons donc joué grâce au site "<http://s.helan.free.fr/awale/lejeu/jouer/>" où l'on peut y jouer à plusieurs difficultés. De plus, sur ce même site on peut voir les règles du jeu et de très bon conseil pour commencer et s'améliorer.

De même pour avoir d'autres sources de conseils, nous avons consulté ce site qui donne des stratégies adoptées en début, milieu et fin de partie :

<http://www.myriad-online.com/resources/docs/awale/francais/strategy.htm> .

2 - Idées

En abordant ce projet nous avons eu plusieurs idées :

- D'abord testé les différentes techniques de fouille de données (LR, NB, LDA, CDA, SVM, ...) sur différents bots pour voir lequel sera le mieux.
- Se renseigner sur le dernier bot créé par Google AlphaGo qui a vaincu le numéro 1 mondial au jeu de Go (algorithme Monte Carlo avec deep learning, il s'est ensuite entraîné seul avec de l'apprentissage par renforcement). Mais on a vite abandonné.
- Les algorithmes Minimax et Negamax aperçus sur un article parlant du jeu d'awélé.
- Ajouter des données à Awélé.data pour avoir une meilleure prédiction.

3 - Compréhension des sujets et des moyens à notre disposition

Ici, nous devons utiliser des outils de fouilles de données afin de créer un modèle permettant de définir si un couple {état de jeu, coup joué} est pertinent, et surtout à quel point.

Nous possédons une base de donnée de 303 triplés {états de jeu ,coup joué, victoire ou défaite}.

{J1,J2,J3,J4,J5,J6,A1,A2,A3,A4,A5,A6,coup joué, résultat}

	J1	J2	J3	J4	J5	J6	A1	A2	A3	A4	A5	A6	C	R
1	5	1	2	2	8	2	1	0	8	7	6	6	C6	G
2	6	6	1	6	5	0	0	0	0	8	7	7	C3	G
3	1	3	3	0	0	2	19	0	1	0	0	0	C2	G
4	7	6	0	6	1	2	7	7	7	5	0	0	C5	G
5	6	2	4	0	0	4	0	0	3	1	3	0	C6	G
6	4	4	0	5	5	1	0	7	6	6	5	5	C4	G
7	1	2	4	0	3	0	6	2	0	1	1	1	C3	P
8	0	1	0	5	1	9	8	1	7	7	1	0	C6	P
9	10	3	9	0	3	0	0	0	0	3	2	1	C3	G
10	4	4	4	4	4	4	4	4	4	4	4	4	C2	P
11	6	6	6	5	5	0	5	5	5	5	0	0	C2	G
12	0	0	0	0	1	10	2	0	5	10	1	3	C5	P
13	8	8	8	0	1	1	8	0	2	7	1	4	C6	P
14	2	4	4	9	4	1	1	1	1	1	0	10	C5	G
15	5	5	5	5	4	4	4	4	4	4	4	0	C6	P
16	19	1	0	0	1	0	0	3	3	1	0	1	C5	P
17	8	2	1	7	7	2	7	7	0	2	1	2	C5	G
18	0	11	7	6	5	5	4	0	1	1	7	1	C2	P
19	1	1	1	1	1	2	2	3	1	0	5	6	C1	P

Les 12 premières variables nous donnent le nombre de graines dans chaque case du plateau.

La première question que nous nous sommes posées est : “quelles sont les différents algorithmes qui pourraient être les plus pertinents ?”

C’est cette question qui a motivé la lecture des rapports des meilleurs algorithmes des deux années précédentes.

Nous avons ainsi pu accéder à un tableau nous montrant que chaque algorithme a des désavantages par rapport à un autre et qu’aucune stratégie est sans faiblesse.

Nous avons donc décidé de réellement nous intéresser au fonctionnement de chaque algorithme afin de comprendre lequel serait le plus simple à utiliser à son maximum.

Malgré une tentative infructueuse de comprendre les modèles mathématiques cachés derrière chaque technique à l’aide de “Data mining algorithms explained using R”, Pawet Cichosz”, nous avons tout de même acquis une meilleure compréhension de tout cela.

La solution que nous avons choisi fut classification naïve Bayésienne. Cette classification porte bien son nom puisqu'elle est incapable de voir les corrélations entre les données et les prends une par une. Si nous lui donnons cet ensemble de données :
{ fruit, rouge, rond, lisse, utilisé pour faire de la compote ou du cidre"}, il prendra chaque élément séparément pour savoir de quoi nous parlons et risque de ne pas trouver de réponse alors qu'un enfant pourrait nous dire facilement qu'il s'agit d'une pomme.

Pourquoi choisir cet algorithme et pas un autre beaucoup plus puissant capable d'effectuer une tonne de travail seul ? Tout simplement car même si il est incapable de déterminer des corrélations entre les éléments, nous pouvons les lui indiquer. Et avec ce travail pré-machés, nous nous retrouvons avec un algorithme très puissant.

Dans le cas de la pomme par exemple, tout le monde pourrait faire la corrélation entre fruit et rouge pour déterminer qu'il s'agit d'un "fruit de couleur rouge". Nous transformons donc notre information en :

{ fruit de couleur rouge, rond, lisse, utilisé pour faire de la compote ou du cidre}

ou

{ fruit, rouge , rond, lisse, utilisé pour faire de la compote ou du cidre,fruit de couleur rouge}

Ici, nous venons de démultiplier la puissance de notre algorithme et surtout, nous avons entièrement compris pourquoi.

De plus, cet algorithme résiste très bien à la sur-adaptation. C'est à dire qu'il pourra utiliser les connaissances acquises dans d'autre cas que les 303 étudiés.

Dans le cas du projet, le réel travail consiste à lire la table des 303 états et déterminer les corrélations d'élément pouvant mener à la victoire ou à la défaite.

III - Mise en oeuvre

1 - Bot finale

Comme précisé plus haut, nous avons utilisé l'algorithme Naives Bayes et l'avons aidé à déterminer certaines corrélations.

Après de nombreuses lectures du groupes donné, nous en avons tiré plusieurs informations.

- 1) Dans 30 cas sur 36, avoir la première case à 1 et la deuxième à une valeur inférieur à 3 conduit à une défaite
- 2) Dans 18 cas sur 19, avoir la première et la troisième case à 0 conduit à une défaite.
- 3) La somme des 6 premières cases nous donne le nombre de grains de notre coté
- 4) La somme des 6 cases suivantes nous donne le nombre de grains du coté de l'adversaire
- 5) Nous pouvons ainsi déterminer la différence de grain de chaque coté. Une valeur qui semble influencer les chances lorsqu'elle atteint un nombre supérieur à 10 ou inférieur à -10
- 6) Sans réelles statistiques, le nombre de case à zéro de notre coté semble influencer grandement les chances de gagner ou de perdre
- 7) pareil du coté de l'adversaire
- 8) Évidemment, le nombre de graine prises influence les chances de gagner

Nous avons ensuite choisi de modifier le bloc de données d'une autre manière.

De {J1,J2,J3,J4,J5,J6,A1,A2,A3,A4,A5,A6,coup joué, résultat}

Nous passons à l'état de jeu après que le coup soit joué

{J1,J2,J3,J4,J5,J6,A1,A2,A3,A4,A5,A6, résultat}

Nous arrêtons de déterminer si un coup est fonctionnel mais nous déterminons si un état de jeu mène ou non à la victoire.

Ensuite, il ne nous reste plus qu'à ajouter les 8 données précisé plus haut et notre modèle est fini.

Cette méthode est la plus simple et naïve qui soit, mais elle nous semble capable de se battre à arme égales avec des méthodes plus complexes mais peut être moins bien comprise par les créateurs.

Ce bot final a été choisi car il bat tous les autres bots avec moins de données sauf un. Cependant, il est meilleur contre d'autres bots possédants d'autres algorithmes que Naive Bayes donc il semble être le plus intéressant.

IV - Conclusion

Ce projet a été une opportunité de toujours essayé de dépasser les résultats accomplis par nos différents bot. Nous le constatons aujourd'hui, l'importance des techniques de fouilles de données est omniprésente. Que ce soit pour la création de bot (pour ce projet ou celui de Bayes Impact) ou bien pour la prédiction de problèmes de décision.

V - Références

- <http://s.helan.free.fr/awale/conseils/>
- <http://www.myriad-online.com/resources/docs/awale/francais/strategy.htm>
- https://www.scei-concours.fr/tipe/TIPE_2012/sujets_2012/INFORMATIQUE_MP.pdf
- <https://en.wikipedia.org/wiki/AlphaGo>
- Livre : Data mining algorithms explained using R, Pawet Cichosz