

# Machine Learning et Théorie des jeux

Nahas Nicolas

Lambert Antoine

## Sujet choisi :

Minimisation de regret hypothétique profond et Poker



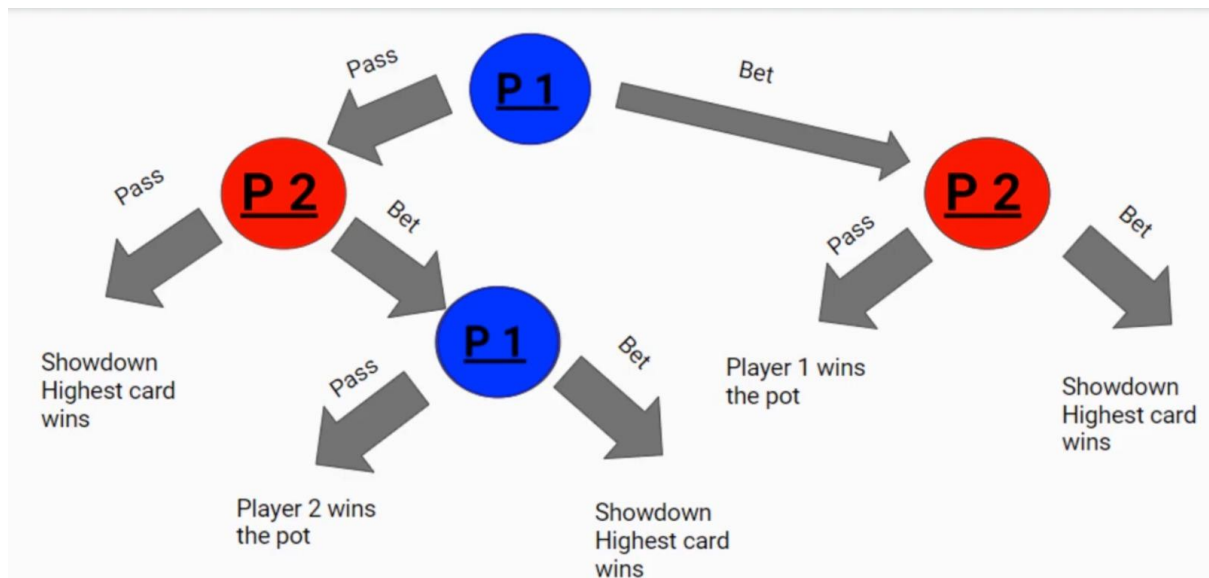
07/04/2021

## Introduction :

Nous avons choisi de travailler sur la minimisation de regret sur un style de poker très simple : le Kuhn. Dans ce jeu on ne joue qu'avec 3 cartes, la plus forte l'emporte, et il n'y a pas de possibilité de ré-enchérir.

Ainsi, le jeu se décompose en trois étapes : c'est au joueur 1 de jouer, en fonction de sa main, il parie ou il passe. Si il parie, alors le joueur 2 a deux choix : parier ou passer. Si il passe, la situation est la même, sauf si le joueur deux choisit de parier, dans ce cas là, le joueur 1 aura à nouveau la possibilité de parier.

Ces étapes peuvent-être résumées avec le schéma suivant :



## **Réalisation d'un code permettant de minimiser le regret hypothétique :**

Ainsi, pour que ce projet nous permette de comprendre le fonctionnement de la minimisation de regret hypothétique dans le cadre du poker, nous avons voulu réaliser un code sans utiliser des librairies facilitant la création de bot.

Notre code est composé de deux fonctions :

### **- La fonction CFR :**

Dans cette fonction, nous allons calculer tour après tour les récompenses associées à une main. Nous allons ensuite calculer le regret, c'est à dire, ce qu'il se passerait en cas de gain. Finalement cela nous permettra par la suite d'implémenter la stratégie de jeu.

A noter que cette fonction a besoin d'un important nombre d'incrémentations pour permettre d'avoir des chiffres dans notre stratégie qui sont exploitables.

calcul regret + strat

### **- La fonction kuhn (fonction de jeu) :**

Ici, nous allons prendre les décisions de jeu en fonction de la carte, du tour et de notre tableau de stratégie. La stratégie étant un tableau prenant en compte le tour, la carte et un pourcentage calculé tour après tour dans la fonction CFR, il nous suffit maintenant d'utiliser ce pourcentage en fonction de la carte pour déterminer l'action à faire. Cette fonction nous retourne le reward réel et hypothétique à partir des cartes et des stratégies des deux joueurs.

Notre repo = <https://github.com/antoinelambert/IAproject>

## Termes clefs de la théorie des jeux appliquée au Poker :

**Équilibre de Nash** : C'est une situation où chaque joueur prévoit correctement le choix des autres et chaque joueur maximise son gain (minimise ses pertes), compte tenu de cette prévision.

**Regret** : En connaissant tous les paramètres, ici par exemple la main adverse, comment on aurait pu faire un meilleur choix

→ Counterfactual reward - actual reward

**Hypothèse de récompense** (Counterfactual reward) : expression de la récompense hypothétique pour les autres décisions.

**Minimisation** : C'est ici la stratégie globale qui minimisera mon regret

## Amélioration de notre algorithme :

En réalité, la minimisation de regret hypothétique au poker devient extrêmement complexe quand on rentre dans des styles de jeu plus classique (texas hold'em). Cela est dû au nombre de combinaisons qui deviennent très conséquentes. D'autre part, l'absence de limite de mise maximale complexifie le choix de l'algorithme.

A titre indicatif une partie de texas holdem limite à seulement deux joueurs présente  $10^{14}$  possibilité de jeu différents. Une partie de texas holdem no limite, toujours à deux joueurs, présente  $10^{160}$  possibilité de partie différente (plus que le nombre d'atome dans l'univers).

Ainsi, les stratégies d'apprentissage sont forcées d'être construites différemment. Les arbres de jeux comme présenté précédemment deviennent trop conséquent et il faut alors se pencher vers les réseaux de neurones et la segmentation : on va s'intéresser à des sections de l'arbre de façon à éviter de traiter toutes les combinaisons & stratégies à chaque décision.

## Sources :

<https://int8.io/counterfactual-regret-minimization-for-poker-ai/>

<https://arxiv.org/pdf/1901.07621.pdf>

<https://www.cmu.edu/news/stories/archives/2019/july/cmu-facebook-ai-beats-poker-pros.htm>

!

<https://science.sciencemag.org/content/359/6374/418>

<http://www.cs.cmu.edu/~sandholm/>

<http://modelai.gettysburg.edu/2013/cfr/cfr.pdf>

<https://poker.cs.ualberta.ca/publications/NIPS07-cfr.pdf>