Distanciel de Méthodes Probabilistes
Jeu de la carte et du bluff : La machine peut-elle apprendre à mentir ?

Adrien BAZOGE, Maëlle BRASSIER

Avril 2019

1 Introduction

Le sujet consiste à simuler une partie de poker entre deux joueurs. Lorsqu'un joueur commence en premier, il est considéré comme joueur A et joueur B lorsqu'il est en second. Nous avons créé un joueur, nommé *ABMB_Player*, qui adopte une stratégie aussi bien en tant que joueur A que joueur B. Ce présent rapport présente l'approche générale de notre stratégie ainsi que l'explication de notre code.

2 Stratégie Joueur A

Le joueur A commençant en premier, il n'a que très peu de stratégie à appliquer face à son adversaire. Son but sera donc de résister au mieux contre un joueur B que nous considérons comme optimum face à lui (selon l'hypothèse de la *Théorie des jeux*). De ce fait, nous étions partis de l'idée que sa marge d'action se restreignait à déterminer quelle somme m miser selon une carte x_1 obtenue. Cependant, il nous est apparu par la suite qu'il était bénéfique de prendre également en compte le comportement du joueur B, c'est-à-dire face à quelle mise il s'est couché.

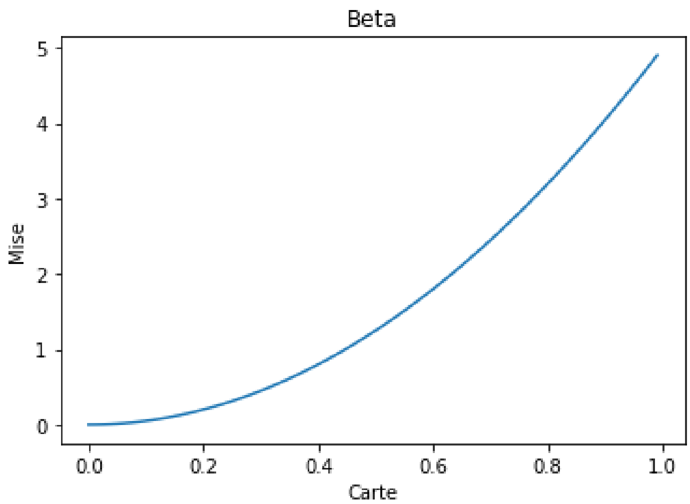
La stratégie en tant que Joueur A se décompose ainsi en deux phases. La première correspond aux 30 premières parties jouées (i.e lorsque le tableau de résultats du joueur A (self.f) comporte moins de 30 lignes). À ce moment là, on calcule l'espérance de gain correspondant à une mise m suivante :

$$\text{Espérance de gain } = ((1+m) \times x_1) - ((1+m) \times (1-x_1))$$

avec m la mise allant de 1 à 5 et x_1 la carte que le joueur A tire. On récupère ensuite la valeur maximale de ces espérances de gain. Si celle-ci est supérieure à -1, alors on misera la valeur y, arrondie supérieure, qui résout l'équation suivante :

$$y = (\sqrt{5} \times x_1)^2$$

Cette dernière peut être représentée par une courbe telle que :



Si l'espérance de gain est inférieure à -1, alors le Joueur A doit se coucher.

La seconde phase de la stratégie intervient lorsque 51 parties sont jouées et dont les résultats sont enregistrés dans *self.f.* Le joueur A va donc apprendre des actions de son adversaire. À l'instar de la phase précédente, nous allons récupérer l'espérance de gain maximale. Si elle est inférieure à -1, nous passons notre tour. Néanmoins, lorsqu'elle est supérieure, nous calculons la probabilité que le joueur B se couche face à une mise m. Cette probabilité correspond à $P(action = reject|m)$. Ce calcul se fait via la portion de code suivante :

```
for i in range(1,6):
    df2 = self.f[self.f['bet']==i]
    nb_occu = (df2['action']=='reject').sum()
    if len(df2) == 0:
      prob[i] = 0
    else:
      prob[i] = nb_occu/len(df2)
maxProbJ = max(prob, key=prob.get)
return min(maxProbJ, valCourbe)
```

Pour chaque mise m possible, on calcule le nombre de fois où le Joueur B a passé son tour face à m.

$$P(action = reject|m) = \frac{\text{\# reject pour mise m}}{\text{\# actions totales pour mise m}}$$

Puis, une fois toutes les probabilités calculées, nous prenons la probabilité maximale, qui renvoie sa clé, à savoir la mise correspondante. Ensuite, nous récupérons une deuxième mise à l'aide de la courbe utilisée dans la première phase de notre stratégie, et nous prenons le minimum entre ces deux mises. Le fait de prendre le minimum entre ces deux mises nous permet de minimiser les pertes mais également de diversifier notre stratégie afin de mieux tromper le joueur B optimum.

3 Stratégie Joueur B

Le Joueur B, quant à lui, va directement s'adapter à la stratégie du Joueur A afin de la contrer. Tout d'abord, si ce dernier a passé son tour, le Joueur B suivra forcément pour remporter les 2€.

Dans le cas inverse, nous allons tout d'abord adopter la même technique que le joueur A pour les 30 premiers rounds joués, en attendant d'avoir suffisamment de données dans *self.s* afin que la deuxième partie de notre stratégie soit efficace. Nous partons du principe que l'adversaire suit la même stratégie que notre propre Joueur A et de ce fait, nous inversons tout simplement l'équation de sorte à obtenir la carte supposée du Joueur A. On obtient donc :

$$x_1 = \sqrt{\frac{y}{5}}$$

avec x_1 la carte (supposée) du Joueur A et y la mise qu'il a placée lors de cette partie. Étant donné que dans la stratégie du joueur A, une mise m couvre un intervalle de cartes possibles, nous prenons ici pour x_1, la carte maximale de l'intervalle correspondant à la mise m. Si notre propre carte x_2 est supérieure à x_1, on mise sinon on passe.

Ensuite, après les 30 premières parties jouées, comme Joueur A change de stratégie, il en est de même pour B. Nous considérons désormais k intervalles de probabilités. En premier lieu, nous calculons :

$$P(x_1 = I_k) = \frac{\text{\# I}_k}{T}$$

$$P(m = bet) = \frac{\text{\# bet}}{T}$$

qui sont respectivement les probabilités que la carte tirée x_1 (de l'adversaire) soit dans l'intervalle I_k et qu'il ait misé une telle mise *bet*, avec T le nombre total de parties jouées en tant que second. Nous calculons ensuite, pour chaque intervalle I_k, la probabilité conditionnelle :

$$P(x_1 = I_k|m = bet) = \frac{\text{\# bet dans } I_k}{\text{\# } x_1 \text{ dans } I_k}$$

qui correspond à la probabilité que la carte de l'adversaire soit dans l'intervalle k sachant qu'il a misé *bet*. Grâce à ces résultats, nous pouvons calculer via le théorème de Bayes :

$$P(m = bet|x_1 = Ik) = \frac{P(x_1 = I_k|m = bet) \times P(x_1 = I_k)}{P(m = bet)}$$

qui est la probabilité qu'il ait misé m sachant que sa carte est dans l'intervalle k. Il s'agit ensuite de trouver l'intervalle de notre propre carte grâce au code suivant :

```
for i in range(self.k):
    if self.card >= self.parts[i][0] and self.card < self.parts[i][-1]+0.01:
        partCard = i+1
```

Enfin, il reste à calculer la probabilité de gain *probGain* qui correspond à la somme de tous les intervalles inférieurs ou égal à notre intervalle.

$$probGain = \sum_{j=1}^{k} P(m = bet | x_1 = I_j)$$

où k est l'intervalle où se trouve notre propre carte x_2

Grâce à elle, nous pouvons calculer l'espérance de gain *esp* trouvé à partir de la carte de l'adversaire prédite que nous comparons au gain d'espérance de notre propre carte *notreEsp*.

$$esp = ((1 + bet) \times probGain) - ((1 + bet) \times (1 - probGain))$$
$$notreEsp = ((1 + bet) \times x_2) - ((1 + bet) \times (1 - x_2))$$

Si le minimum de ces deux valeurs est supérieur à -1 alors nous misons, dans le cas contraire, nous passons. Nous considérons ici le minimum des deux valeurs afin de ne pas prendre une décision uniquement basée sur la carte prédite de l'adversaire.