Probabilités

Raphaël Montaud et Gabriel Misrachi

7 juin 2017

Résumé

Modélisation d'évènements rares appliqué au championnat de football anglais en Python. Ensemble du code disponible sur la page GitHub.

1 Premier League 2015-2016

1.1 Mise en place du modèle et premières simulations

Le modèle utilisé est celui de Bradley-Terry. Chaque équipe est munie d'une force que l'on se donne au début du championnat. Ensuite le résultat de chaque match suit une simple loi de Bernoulli de telle sorte que : $\mathrm{P}(X_{i,j} = 1) = \frac{V_i}{V_i+V_j}$ où les X_{ij} sont les variables représentant les résultats des matchs.

Simuler un championnat revient donc à simuler une matrice de variables aléatoire de Bernoulli. On peut ensuite facilement récupérer le score de chaque équipe et ainsi obtenir le classement. En simulant un grand nombre de championnats, on calcule la probabilité de chaque équipe de gagner. Le lecteur pourra se référer au fichier simulation_standard.py pour tester notre algorithme.

1.2 Evénements rares : décalage en probabilité

La simulation standard de Monte-Carlo ne suffit pas à évaluer les événements rares. Il faut utiliser des méthodes de décalage de probabilité. Pour des expériences de Bernoulli nous disposons du théorème suivant :

Soient $Y_1, \ldots, Y_k$ des variables aléatoires indépendantes de loi de Bernoulli $B(p_1), \ldots, B(p_k)$; et $Z_1, \ldots, Z_k$ des variables aléatoires indépendantes de loi de Bernoulli $B(q_1), \ldots, B(q_k)$. Pour toute fonction mesurable bornée g, on a :

$$\mathrm{E}[g(Y_1, \ldots, Y_k)] = \left(\prod_{i=1}^{k} \frac{1-p_i}{1-q_i} \right) \mathrm{E}\left[g(Z_1, \ldots, Z_k) \prod_{i=1}^{k} \left(\frac{p_i(1-p_i)}{q_i(1-q_i)} \right)^{Z_i} \right]$$

Nous allons ainsi appliquer ce théorème, en prenant comme fonction g la fonction indicatrice de $X \in A$ et ainsi l'espérance correspondra à la probabilité $\mathrm{P}(X \in A)$. Intéressons nous au cas de l'événement $\{Leicester\ gagne\}$. La probabilité obtenue selon la méthode Monte-Carlo standard est très faible et pour avoir une estimation correcte, il nous faudra soit faire un grand nombre de simulations avec la méthode de Monte-Carlo classique, soit faire un décalage de probabilité si l'on souhaite éviter de faire trop de simulations.

Le but du décalage de probabilité est de forcer la réalisation de l'événement $\{Leicester\ gagne\}$ en modifiant les valeurs des $p_{ij} := \mathrm{P}(X_{i,j} = 1)$. Nous allons plutôt modifier les forces des différentes équipes car cela est plus simple à implémenter en Python. Il faut donc modifier la force de Leicester de façon à ce que l'événement ne se réalise ni trop rarement ni trop souvent. Lors des calculs, nous imprimons le taux de réalisation de l'événement de façon à savoir si notre décalage en probabilité est pertinent.

Il y a une simplification des calculs possible quand on parle des événements simples comme $\{Leicester\ gagne\}$. Effectivement, pour provoquer cet événement, on va simplement augmenter

la force de Leicester et ainsi les expériences de Bernoulli entre deux équipes ; autres que Leicester, resteront inchangées (ce qui donne dans la formule des termes qui valent systématiquement 1). Cependant, pour des événements plus complexes comme l'événement $B = \{$Leicester gagne ; Manchester City et Manchester United hors podium ; et Liverpool et Chelsea hors places européennes$\}$, augmenter la force de Leicester ne suffit pas à provoquer l'événement. Il faut donc aussi diminuer les forces des équipes concernées. La simplification n'est alors plus possible. On peut aussi noter que malgré tous les décalages en probabilité, l'événement est extrêmement rare. Ainsi, afin d'obtenir un intervalle de confiance correct, il faut quand même réaliser de très nombreux essais. Le lecteur pourra se référer au fichier evenement_rare.py pour faire des essais. Le résultat obtenu est de $0.41\% \pm 0.025\%$ pour la victoire de Leicester et de $0.0069\% + -0.0024\%$ pour l'événement B. Ces résultats ont été obtenus pour $n = 10000$ pour l'événement $\{Leicester\ gagne\}$ et pour $n = 10^6$ pour l'événement B.
On peut remarquer que le nombre de simulations à faire pour obtenir un intervalle de confiance honnête est élevé (ici un million pour l'événement le plus rare considéré). Si on calcule l'intervalle de confiance pour l'événement $\{Leicester\ gagne\}$ sans échantillonnage préférentiel, on remarque que l'intervalle de confiance est d'ordre comparable à celui avec échantillonnage préférentiel, cependant rien n'indique que nous soyons dans le cadre de validité du théorème centrale limite. Par exemple, pour satisfaire la condition habituellement utilisée $np > 50$ il faudrait 20 millions d'essais pour l'événement B.

1.3 Evénements rares : théorème ergodique et splitting

Cette partie utilise le théorème ergodique et une méthode de splitting pour simuler la probabilité de $\{Leicester\ gagne\}$. Au lieu d'utiliser une part d'histoire de façon continue, on le fait de façon discrète. En fait, on part d'un résultat de championnat, et à partir de ce résultat nous allons resimuler une fraction ρ des matchs de ce championnat. Ainsi, on construit une chaîne de Markov avec un nombre d'états finis. On respecte la propriété de Markov car les probabilités de l'état $n + 1$ ne dépendent que de l'état n. Le splitting se fait sur le score obtenu par Leicester. On applique ensuite le théorème ergodique sur chacun des ensembles, ce qui permet de calculer les probabilités conditionnelles puis la probabilité finale. L'objectif est bien sûr de se rapprocher de cet événement $\{score(Leicester) >= 10\}$ via des probabilités conditionnelles qui doivent être les moins rares possibles.

On simule ici 190 matchs à chaque fois. Donc quand on verra $ro = 0.006$ cela signifie en fait qu'on ne resimule qu'un seul match. Le problème est que le taux de rejet "idéal" pour les méthodes de splitting est d'environ 20%. Admettons que l'on se place dans la situation où Leicester a 9 points. Ses meilleures chances de gagner sont lorsque l'on ne resimule qu'un seul match. Cependant même dans ce cas là, comme Leicester a des scores faibles par rapport aux autres équipes, le taux de rejet sera très probablement élevé. C'est ce que l'on constate en pratique avec des taux autour de 60%. La méthode de splitting n'est donc pas idéale dans ce cas. En effet, les résultats obtenus ne sont pas probants, et les auteurs pensent que le problème vient d'une mauvaise approximation due aux taux de rejet trop élevés. Une autre approche que nous avons développée est de faire du splitting sur le classement de Leicester. Cependant les résultats sont aussi incohérents car les taux de rejet pour cette méthode sont encore plus élevés.

2 Études de théorèmes sur les modèles de Bradley-Terry

2.1 Théorème 1

Le théorème 1 prouve que si l'on vérifie les conditions suivantes sur la fonction quantile Q de la distribution U des forces des équipes :

— Il existe β dans (0, 1/2) et $x_0 > 0$ dans l'intérieur de $suppQ$
tels que $Q^{1/2\beta}$ est convexe sur $[x_0, \infty)$.
— $\mathrm{E}\big[U^2\big] < \infty$.

Alors, la probabilité que le joueur le plus fort gagne tend vers 1 avec le nombre N de joueurs :

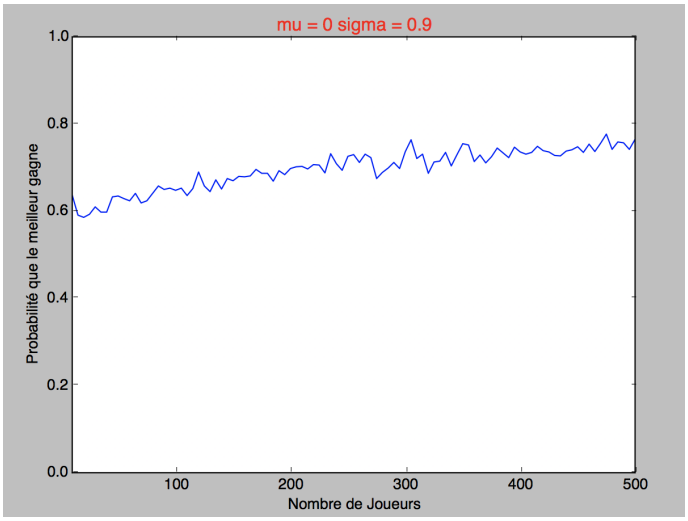
$$P(le\ plus\ fort\ gagne) \xrightarrow[N\to+\infty]{} 1.$$

En pratique, dans le cas continu, on dispose de distributions classiques qui vérifie de telles propriétés. On pense notamment aux distributions exponentielles.
Toutefois, avec un nombre de simulations et de joueurs limités par le temps de calcul, ces distributions ne font pas apparaître la convergence assez rapidement. On va donc illustrer ce théorème avec deux autres distributions :

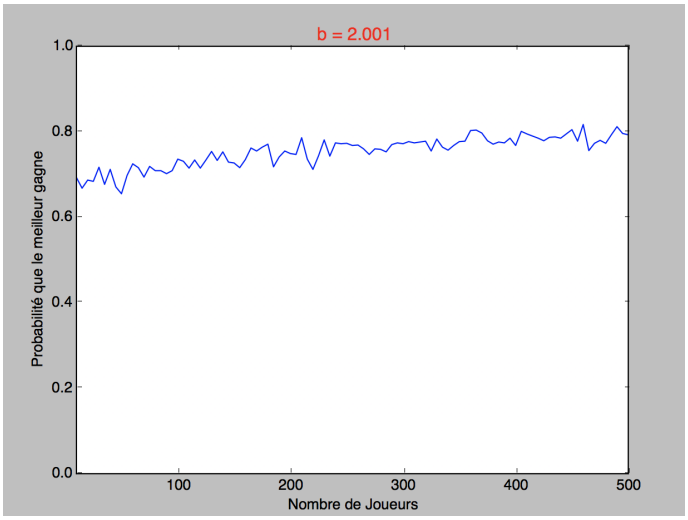
— la première a pour queue de distribution $Q(x) := (x+1)^b$ où $b > 2$.
— la seconde est une distribution U dite log-normale, c'est à dire :

$$X \sim \mathcal{N}(\mu, \sigma),\ U = e^X$$

Cas de la distribution de queue en x^{-b}



Cas de la distribution log-normale



Ces résultats ont été obtenus en simulant mille championnats pour chaque multiple de cinq entre dix et cinq cent. (250 millions de calculs pour la dernière itération) Aussi, la puissance de calcul reste un obstacle majeur pour l'illustration de ce résultat et nous nous sommes contentés de ces maigres résultats. Dans les deux cas, le comportement prédit ne saute pas aux yeux mais l'on décèle bien une tendance croissante de la probabilité que le plus fort gagne avec le nombre de joueurs. On observe de plus que le meilleur joueur a de fortes de chances de gagner sur ces types de distributions. Ce ne sera pas le cas pour toutes les distributions, comme le prouve le théorème 2.

2.2 Théorème 2

Le théorème 2 nous apprend que sous certaines conditions (hypothèse A), on peut déterminer selon la valeur de γ si la portion $\frac{N^\gamma}{N}$ gagne presque sûrement (ou ne gagne pas presque sûrement) quand on fait augmenter le nombre de joueurs. Soit U la loi de distribution des forces des joueurs et Q la fonction de répartition de queue de U. Si $supp(Q)$ est fini, on peut donc le ramener à l'intervalle $[0, 1]$ par homogénéité. Avec ces notations :

Hypothèse A :
Le maximum de $supp(Q)$ est 1 et il existe $\alpha \in [0, 2)$ tel que :

$$logQ(1-u) = \alpha log(u) + o(logu) \quad quand\ u \to 0 \qquad (A)$$

Théorème 2 : sous les conditions de l'hypothèse A
Pour tout $\gamma < 1 - \alpha/2$, on a pour N tendant vers l'infini :

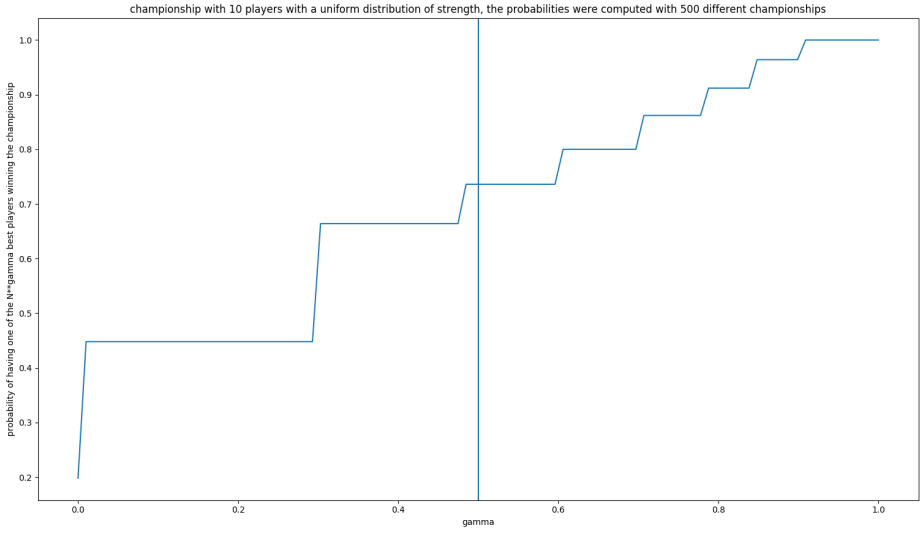
$$\mathrm{P}(aucun\ des\ N^\gamma\ meilleurs\ joueurs\ gagne) \to 1$$

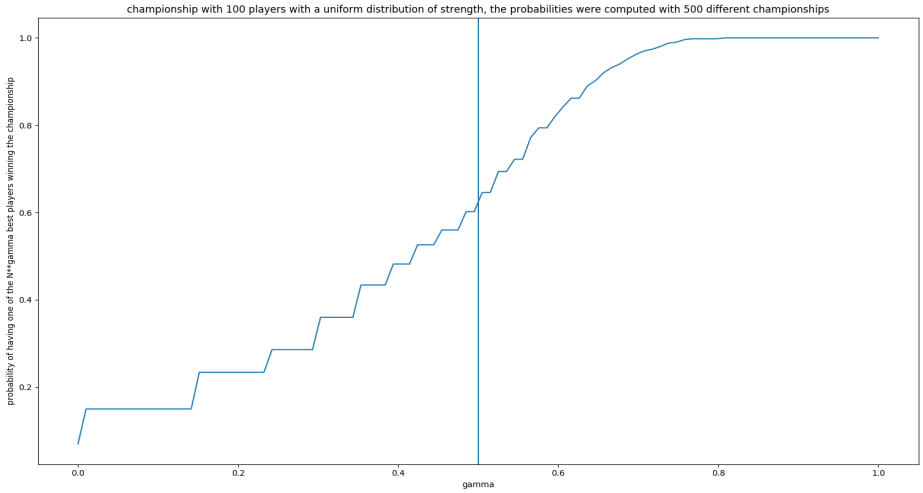
Pour tout $\gamma > 1 - \alpha/2$, on a pour N tendant vers l'infini :

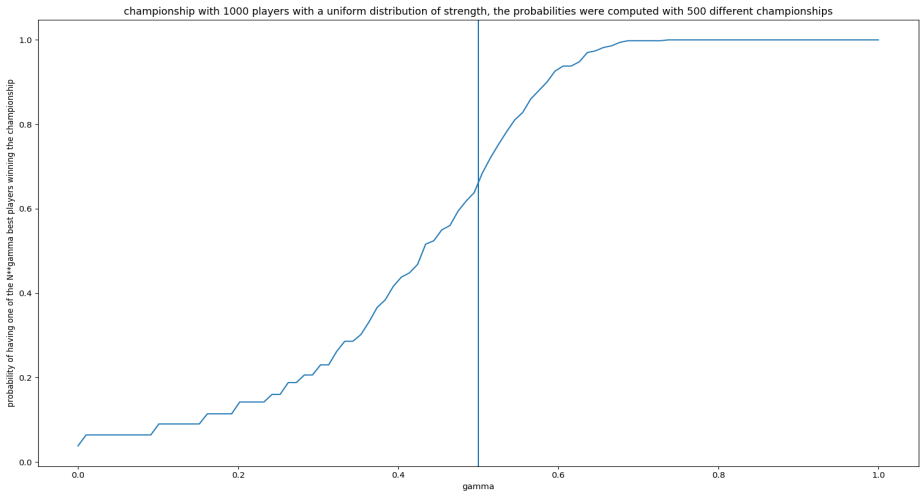
$$\mathrm{P}(un\ des\ N^\gamma\ meilleurs\ joueurs\ gagne) \to 1$$

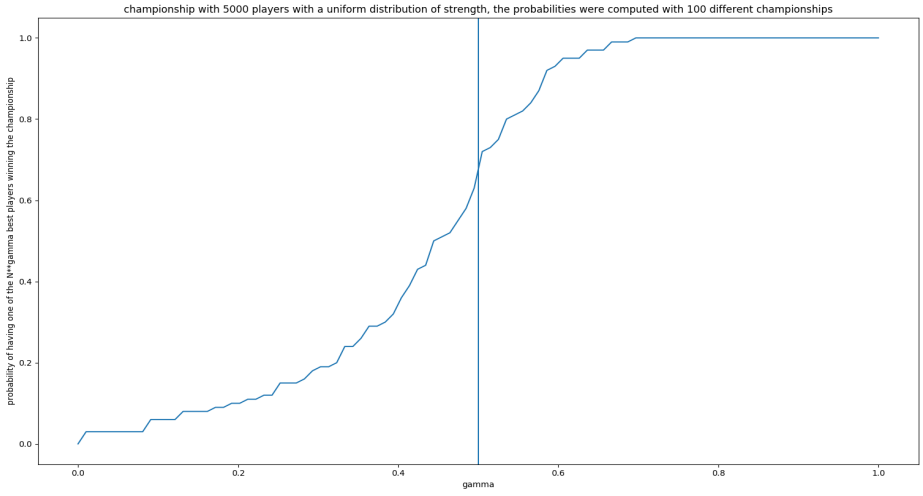
On sait que l'hypothèse A est respectée pour différentes distributions :
— la distribution uniforme avec $\alpha = 1$
— la distribution $arcsin$ avec $\alpha = 1/2$
— les distributions $\beta(a, b)$ avec $\alpha = b$ si $b < 2$

Dans le fichier Theorem2.py, le lecteur pourra trouver une illustration de ce théorème avec deux distributions disponibles : loi β ou uniforme (ces deux distributions respectent bien l'assomption A). On peut illustrer le théorème en faisant des essais pour différentes valeurs du nombre de joueurs et vérifier que la portion de joueurs N^γ finit toujours par gagner (respectivement ne pas gagner). On trace donc le graphe de probabilité en fonction de γ pour des valeurs de N allant vers l'infini. Les graphes présentés ont été faits sans matchs retour. Pour avoir une meilleure illustration du théorème, on pourra envisager de faire des calculs pour de plus grandes valeurs de N en utilisant des méthodes de parallélisation par exemple. Dans le graphe, nous avons ajouté une barre verticale qui montre la rupture entre les valeurs de γ qui vont tendre vers 0 et celles qui vont tendre vers 1 d'après le théorème 2.




championship with 100 players with a uniform distribution of strength, the probabilities were computed with 500 different championships
probability of having one of the N*gamma best players winning the championship
gamma
0.0
0.2
0.4
0.6
0.8
1.0


championship with 1000 players with a uniform distribution of strength, the probabilities were computed with 500 different championships
probability of having one of the N*gamma best players winning the championship
gamma
0.0
0.2
0.4
0.6
0.8
1.0


championship with 5000 players with a uniform distribution of strength, the probabilities were computed with 100 different championships
probability of having one of the N**gamma best players winning the championship
gamma
0.0
0.2
0.4
0.6
0.8
1.0

2.3 Théorème 3 : théorème et illustration

Le théorème 3 permet de savoir quelle force est nécessaire pour un $N+1e$ joueur qui arriverait dans le championnat pour avoir des chances de gagner.

Théorème 3 : sous les conditions de l'hypothèse A

$$Soit\ V_U = \mathrm{E}\left[\frac{U}{(U+1)^2}\right]\ et\ \epsilon_N = \left(\frac{(2-\alpha)(logN)}{NV_U}\right)^{1/2}$$

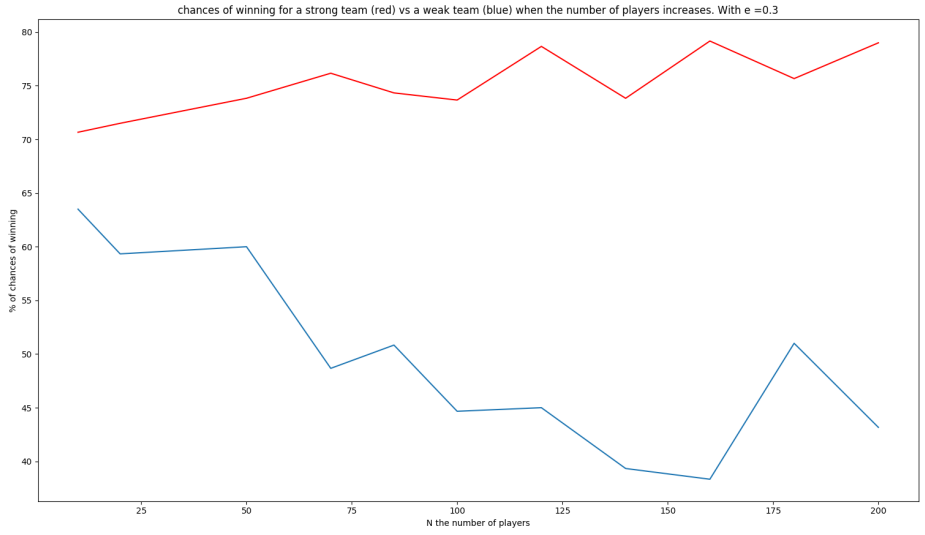
Si x_N représente la force d'un $N+1e$ joueur alors : Si $lim\ inf_{N\to\infty}\frac{x_N-1}{\epsilon_N} > 1$ alors P presque sûrement :

$$\mathrm{P}(le\ N+1e\ joueur\ gagne) \to 1$$

Si $lim\ sup_{N\to\infty}\frac{x_N-1}{\epsilon_N} < 1$ alors P presque sûrement :

$$\mathrm{P}(le\ N+1e\ joueur\ ne\ gagne\ pas) \to 1$$

Le théorème 3 montre donc l'existence d'une valeur limite $1+\epsilon_N$ qui sépare les forces qui vont tendre à perdre et celles qui vont tendre à gagner. Nous prenons donc une ligue où la distribution des forces est uniforme (on respecte ainsi l'assomption A avec $\alpha = 1$). On calcule alors la valeur de ϵ_N et on fait deux essais différents : dans l'une, on rajoute une équipe de force légèrement supérieure à $1+\epsilon_N$ et dans l'autre une équipe légèrement moins forte. On trace alors le taux de victoire pour des valeurs de N (le nombre de joueurs) qui grandissent. D'après le théorème 3, on devrait observer que dans le premier cas, la $N+1e$ équipe tend à gagner tous les championnats et que dans le deuxième cas, la $N+1e$ équipe tend à ne jamais gagner. Le lecteur pourra se référer au fichier Theorem3.py. On peut y régler la valeur du coefficient e qui est tel que le ratio $\frac{v_{N+1}-1}{\epsilon_N}$ tend vers $1+e$ (resp. $1-e$) pour le deuxième cas). Nous avons fait des essais avec différentes valeur de e qui sont toujours concluantes, mais pour illustrer le théorème 3 sans avoir à choisir des très grandes valeurs de N, il faut choisir une valeur de e relativement élevée. Ici nous présentons des résultats pour $e = 0.3$.



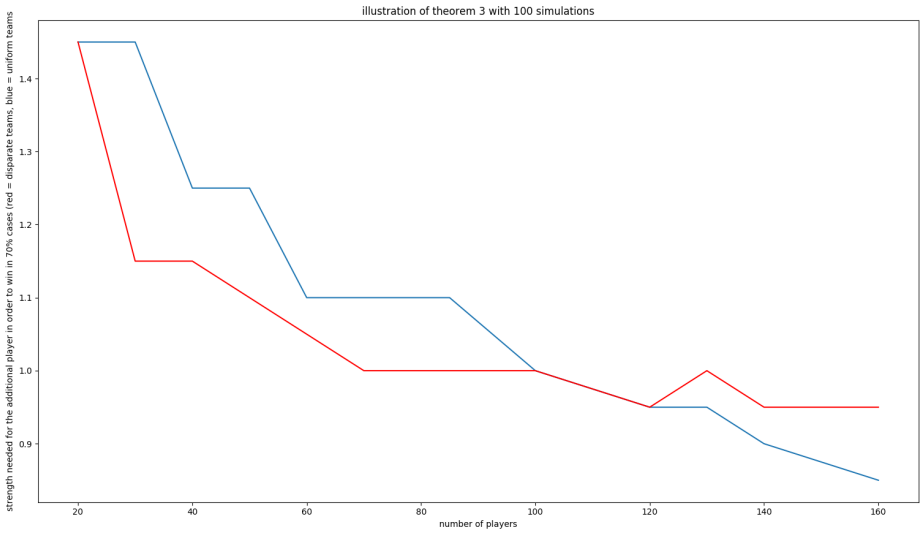
2.4 Théorème 3 : Un exemple concret

Cet exemple reprend une situation qu'on peut retrouver dans la réalité (on peut penser au championnat français). On compare deux situations :

- Une équipe possède une force de 2 tandis que toutes les autres possèdent une force de 1
- Une équipe possède une force de 2, une autre une force de 1 et tout le reste a une force de 0.5.

L'écart entre la force de l'équipe la plus forte et la moyenne est bien plus grand dans le deuxième cas. On s'attend donc à obtenir de meilleurs résultats pour la meilleure équipe dans le cas 2. Quel est le lien avec le théorème 3 ? On prend deux championnats : les championnats un et deux privés du meilleur joueur. On se demande alors quelle force il faut avoir pour qu'un joueur additionnel puisse avoir des chances de gagner. Remarquons d'abord que nos deux distributions respectent l'assomption A avec $\alpha = 0$. Ensuite, un calcul simple montre que $v(U_1) > v(U_2)$ puis que $\epsilon(U_2) > \epsilon(U_1)$ pour tout N. Donc cela signifie, d'après le théorème 3 que la force nécessaire à un joueur supplémentaire pour avoir des bonnes chances de gagner le tournoi est supérieure dans le cas numéro 2. Ceci peut paraître contre-intuitif, mais il s'agit d'un phénomène sensible en fait. Si toutes les équipes sont assez fortes mais toutes au même niveau, l'équipe favorite aura du mal à gagner ses matchs mais elle en gagnera en moyenne plus que les autres. En revanche, si les équipes sont faibles, l'équipe favorite gagnera en moyenne plus de points mais si une équipe se trouve être moins faible que les autres, elle a aussi toutes ses chances de gagner beaucoup de matchs et de passer devant l'équipe favorite.

Pour illustrer cela, nous avons créé un algorithme qui cherche la valeur de la plus petite force nécessaire pour remporter 70% des matchs. Nous l'appliquons aux deux distributions et il devrait apparaître qu'il est plus facile de gagner dans le premier cas que dans le second. Nous exposons ici les résultats obtenus à partir du code du fichier Theorem3_a_concrete_example.py.

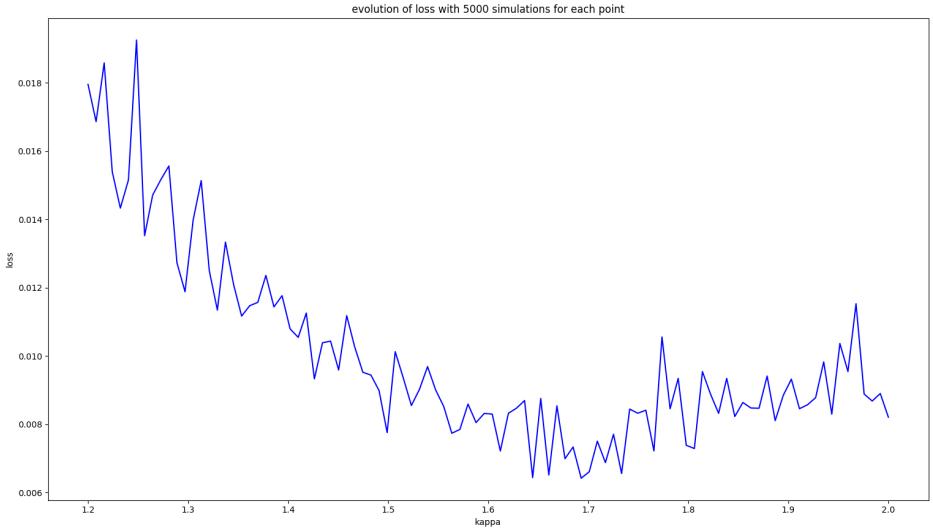


3 Pour aller plus loin

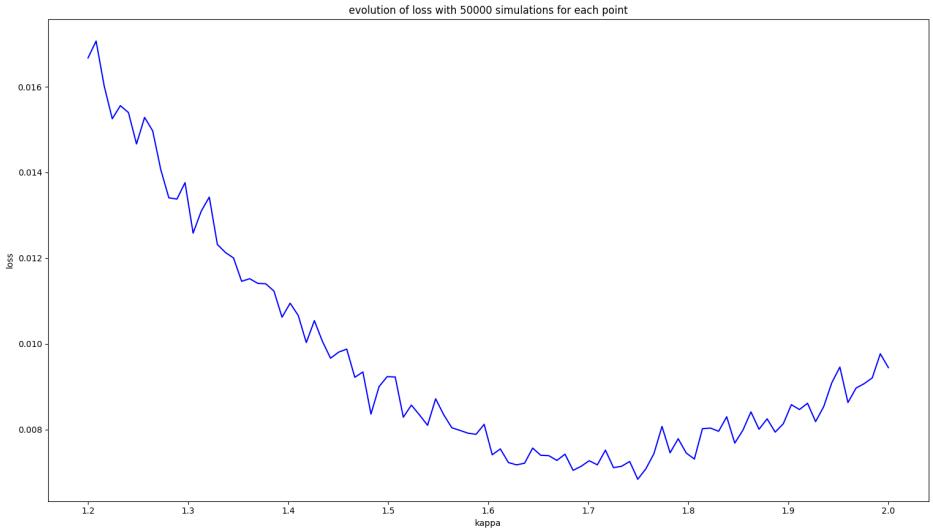
3.1 Estimation des paramètres du modèle en approchant les côtes des bookmakers

Les résultats des championnats de football représentent des enjeux économiques importants. En effet d'après le site d'information eurosport, les bookmakers auraient perdu 30 millions d'euros lors de victoire de Leicester en 2016. Nous avons donc souhaité nous placer dans une situation réelle de recherche de gain sur des paris sportifs. On fait l'hypothèse suivante : on souhaite gagner de l'argent à l'aide d'arbitrages sur des côtes de bookmakers. On se pose comme modèle le modèle avec κ ou θ. Nous partons du principe que les bookmakers sont précis sur les événements assez fréquents comme la victoire de grandes équipes mais qu'ils savent mal estimer les probabilités des événements rares. Nous optimisons donc notre modèle pour coller aux valeurs des bookmakers sur les grandes équipes. Nous pouvons ensuite estimer les probabilités des événements rares grâce à notre modélisation informatique et nous pouvons alors parier sur les événements qui sont mal côtés.

Pour optimiser les paramètres du modèle nous utilisons une méthode de descente de gradient. La difficulté réside ici dans le fait que la fonction à optimiser n'est pas déterministe et est soumises aux aléas. (La fonction théorique que l'on souhaite optimiser est déterministe mais pas notre estimation informatique). De plus cet aléa s'accumule dans le calcul du gradient puisque que l'on doit calculer deux valeurs de la fonction. Voici un premier tracé de la fonction de coût (nous avons choisi une fonction de coût quadratique) pour le modèle utilisant κ :

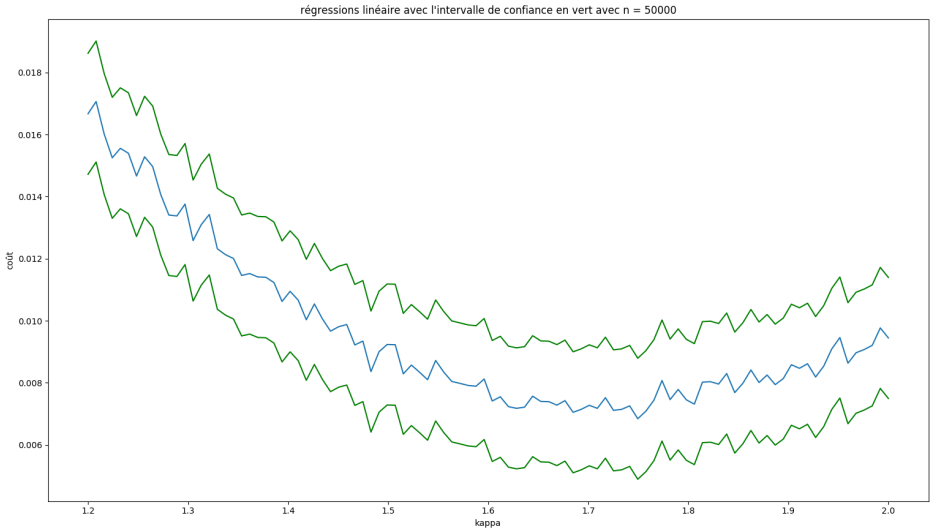


On voit ici que cette fonction qui est censée être convexe oscille et on constate qu'en augmentant le nombre de simulations, la fonction se lisse. C'était attendu puisque d'après le théorème central limite, la fonction simulée tend vers la fonction théorique quand on augmente le nombre de résultats.



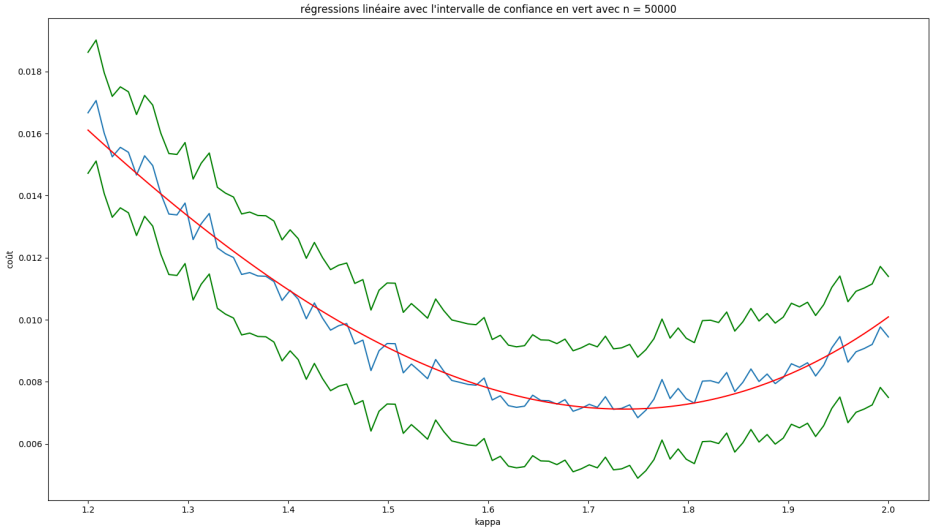
On estime alors la variance du résultat, ce qui nous permet de tracer la fonction avec des barres d'erreur (intervalles de confiance à 95%).

Deux solutions s'offrent alors à nous. Nous pouvons faire une approximation de la fonction réelle en faisant une hypothèse sur sa forme et en minimisant l'écart aux points expérimentaux et



en interdisant de sortir des marges de barre d'erreur. Ou alors nous pouvons faire de la descente de gradient avec des valeurs très élevées de nombre d'essais afin de minimiser l'erreur due à l'aléa.

Développons la première solution. Nous implémentons une méthode de régression linéaire que nous testons avec plusieurs formes de fonctions. Le meilleur résultat est obtenu avec une interpolation par une fonction polynomiale du troisième degré. Voici le résultat obtenu :



Maintenant que nous avons avons interpolé la fonction de coût avec une fonction que nous noterons f, nous pouvons faire une descente de gradient sur f pour trouver son minimum. Nous obtenons $\kappa = 1.736$.

Développons dorénavant la deuxième méthode. Il faut faire une descente de gradient sur la fonction de coût. Fixons nous un nombre maximal de simulations à faire. Une première option est de travailler sur des valeurs assez précises du gradient du coût. Dans ce cas il faut choisir un pas d'apprentissage élevé afin de pouvoir atteindre la solution avant de dépasser le nombre maximal de simulations. Une deuxième option est d'avoir une plus grosse erreur sur le gradient mais un pas

d'apprentissage faible, de sorte que un pas dans la mauvaise direction n'ait pas trop d'impact. On compte alors sur la loi des grands nombres pour faire converger cette "marche aléatoire". Finalement, nous optons pour un entre deux. De toute manière, dans les deux cas nous n'avons pas la puissance de calcul nécessaire pour utiliser cette méthode correctement, la reproductibilité des résultats n'est pas satisfaisante et il semble que nous ayons encore trop de bruit. Nous présentons tout de même le résultat obtenu qui est : $\kappa = 1.690$.

Une fois que nous avons optimisé notre paramètre κ, nous pouvons réinjecter cette valeur de κ dans notre modèle et estimer la probabilité que Leicester gagne le championnat. Ainsi si nous avions fait ces calculs au début du championnat, nous aurions pu estimer les probabilités des événements rares comme la victoire des petites équipes et parier là où la côte est avantageuse d'après notre modèle. Par exemple nous trouvons une probabilité de victoire de Leicester de $0.20\% \pm 0.013\%$. Or sa côte en début de championnat était de 5000 chez certains bookmakers. Ainsi en misant 10 euros, notre espérance de gain était de 90 euros d'après notre modèle. On pourrait alors faire de petits paris sur un grand nombre d'événements rares mal côtés pour répartir le risque et faire de gros gains avec une variance relativement faible. Pour nous donner une idée, supposons que l'on trouve 1000 événements de la sorte et que l'on parie 10 euros sur chaque. Il s'agit alors d'une loi binomiale. L'investissement est de 10 000 euros, l'espérance est de 90 000 euros et la probabilité de tout perdre est de 14%.

4 Conclusion

Après nous être fixés un système probabiliste pour les rencontres sportives, nous avons pu modéliser le déroulement d'un championnat de football. Comme nous l'avons montré, la simulation de Monte-Carlo standard ne suffit pas pour étudier des événements rares sans l'aide de super-ordinateurs. Le recours à la méthode de décalage en probabilité permet d'obtenir des estimations de probabilités pour les événements rares. Ensuite, nous nous sommes penchés sur le comportement de différentes distributions de forces des équipes dans le cas ou le nombre de joueur tend vers l'infini. La théorie montre que certaines distributions sont plus propices à la défaite du meilleur joueur, mais l'expérience montre que certains de ces comportements ne sont sensibles que pour un très grand nombre de joueurs, ce qui nous éloigne du sujet des championnats de football, mais qui peut être valide dans d'autres domaines d'application du modèle de Bradley-Terry comme la comparaison de revues scientifiques. Enfin nous avons montré que les enjeux économiques dans les résultats de football sont importants et nous avons proposé une méthode pour en tirer profit. Après nous être fixé notre modèle probabiliste, nous optimisons nos paramètres pour nous rapprocher de la réalité. Nous pouvons alors utiliser ces paramètres pour estimer des événements rares grâce aux méthodes vues précédemment.

5 Bibliographie

R. Chetrite, R. Diel, M. Lerasle *The number of potential winners in Bradley-Terry model in random environment*, Ann. Appl. Probab., 2016. https ://arxiv.org/abs/1509.07265

S. Escalón. *Le ballon rond à l'épreuve des probabilités*, Le Journal du CNRS, 2017. https ://le-journal.cnrs.fr/articles/ le-ballon-rond-a-lepreuve-des-probabilites.