Rating players for L4D2

Résumé

On cherche à évaluer la performance d'un ensemble de joueurs sur un ensemble de parties. Les joueurs jouent par équipe avec coéquipier changeant, un nombre variable de bots, et toutes les combinaisons ne sont pas jouées. L'objectif est de tenir compte de la force des différents joueurs dans l'évaluation du résultat d'une partie.

Mise en place, notations. On a $N \in \mathbf{N}$ joueurs, ayant effectués $M \in \mathbf{N}$ parties. On note $j_1, \dots, j_N$ les joueurs, $p_1, \dots, p_M$ les parties. La feuille de match de la partie p_m est constitué de : [1]

- $\{\alpha_1(m), \alpha_2(m)\}$ les joueurs de l'équipe A, plus deux bots ;
- $\{\beta_1(m), \beta_2(m)\}$ les joueurs de l'équipe B, plus deux bots ;
- $r(m)$ le résultat de la partie (voir paragraphe, correspondant)

Elo. Le elo doit être un moyen d'évaluer la force d'un joueur. On note e_{j_i} l'elo du joueur j_i. L'algorithme de classement se base sur les deux hypothèses suivantes, que l'on espère grosso-modo vérifiées :

(H1) La force d'une équipe est proportionnelle à la somme des forces de ses joueurs ;

(H2) Le résultat d'un match est proportionnel à la différence entre les forces des deux équipes.

Il devient donc fondamental d'introduire l'élément suivant, concernant une partie opposant $\{\alpha_1, \alpha_2\}$ à $\{\beta_1, \beta_2\}$ (et les bots) :

$$\Delta(\alpha_1, \alpha_2, \beta_1, \beta_2) \;=\; \left(\, e_{\alpha_1} \;+\; e_{\alpha_2} \,\right) \;-\; \left(\, e_{\beta_1} \;+\; e_{\beta_2} \,\right) \,.$$

Sous les hypothèses (H1) et (H2), le résultat de la partie r devient

$$r \;=\; k\Delta, \quad k \text{ une constante} \,.$$

On note que l'on peut librement fixer l'elo "moyen" des joueurs sans changer r ou Δ ; on peut aussi multiplier les elos par une constante bien choisie afin de fixer la constante k. *Dans la suite, on fixe l'elo moyen à* $e = 1500$*, et la constante* $k = 1$. Ces constantes sont purement arbitraires. Rien n'empêche un elo de devenir négatif a priori.

1. On suppose pour l'instant deux équipes de deux joueurs humains, plus deux bots. Les cas avec un nombre différent de joueurs par équipe sera traité dans un paragraphe séparé.

Order Independent Whole History Ranking (OIWHR). L'objectif est de mettre en place un procédé évaluant la performance (elo) des N joueurs, étant donné le résultat des M parties. Il s'agit simplement de trouver le vecteur $E = (e_{j_i})_{i \in 1 \ldots N} \in \mathbb{R}^N$ vérifiant

$$E \;=\; \underset{E' \text{ t.q. } \frac{1}{N}\sum_i e'_{j_i} = 1500}{\arg\min} \; \sum_{m \in 1 \ldots M} \left| r(m) - \Delta_m(E') \right|^2, \qquad (\star)$$

où l'on note $\Delta_m(E) = \Delta(\alpha_1(m), \alpha_2(m), \beta_1(m), \beta_2(m))$.

On cherche donc à trouver l'ensemble des elos, E, minimisant la différence entre le résultat de l'ensemble des parties et le résultat théorique correspondant à E : $(\Delta_m(E))_{i \in 1 \ldots N}$. On veut également un elo moyen $\frac{1}{N}\sum_i e_{j_i} = 1500$, ce qui peut toujours être assuré *a posteriori* par translation.

Quelques remarques et propriétés.

a) Le procédé tient compte de toutes les parties déjà jouées, de manière équivalente, et en particulier indépendamment de l'ordre dans lequel ces parties ont été jouées.

b) À chaque nouvelle partie, les elos de l'ensemble des joueurs sont réactualisés. Le comportement intuitif lorsqu'une nouvelle partie, p_m, est jouée, est le suivant [2]. Imaginons que

$$r(m) > \Delta(\alpha_1^{(m-1)}(m), \alpha_2^{(m-1)}(m), \beta_1^{(m-1)}(m), \beta_2^{(m-1)}(m)),$$

où l'on note $\alpha_1^{(m-1)}(m)$ l'elo de α_1 en tenant compte des $m-1$ premières parties (donc son elo avant le match). Alors l'équipe A a biiiiiien maitrisé l'équipe B. Les membres de l'équipe A, *ainsi que toutes les personnes contre lesquelles ils ont déjà joué* verront leur elo grimper, tandis que leurs anciens partenaires verront leur elo baisser. Inversement les membres de l'équipe B, *ainsi que toutes les personnes contre lesquelles ils ont déjà joué* verront leur elo baisser, tandis que leurs anciens partenaires verront leur elo grimper. Evidemment, une personne peut se retrouver dans plusieurs catégories contradictoires, et le résultat est moins clair.
Notons en particulier que l'on peut parfaitement perdre des points en gagnant une partie, si on ne la gagne pas avec assez de différence.

c) Si (H1) et (H2) sont parfaitement vérifiées, alors le procédé retrouve la force exacte des joueurs, pourvu qu'un nombre suffisant de combinaisons soient jouées (pour avoir unicité du minimum). On peut mesurer le défaut de (H1) et (H2) avec la valeur de $\sum_{m \in 1 \ldots M} |r(m) - \Delta_m(E)|^2$.

d) Le minimum existe toujours. Il peut ne pas être unique ; dans ce cas on choisira le E le plus proche de 1500 (au sens des moindres carrés) parmi les solutions. En particulier, un joueur n'ayant pas joué a un elo virtuel de 1500.

2. à vérifier...

Résolution. On peut résoudre le problème en utilisant des outils d'Algèbre Linéaire. Le problème ($\star$) peut être réécrit comme

$$E \;=\; \underset{E' \text{ t.q. } \frac{1}{N}\sum_i e'_{j_i}=1500}{\arg\min} \left\|AE' - \mathbf{b}\right\|,$$

où $\mathbf{b}$ est le vecteur des résultat de parties, $(r(m))_{m=1\cdots M}$, et A est le tableau de participation aux parties (pour la partie m, on ajoute 1 si la personne a joué dans l'équipe A, -1 si elle a joué dans l'équipe B, 0 sinon—il peut y avoir mutiplicité pour les bots) ; A possède M lignes et $N+N_b$ colonnes, où N_b est le nombre de bots, selon la méthode utilisée.

Le problème ci-dessus est classique, et se résoud à travers le pseudo-inverse [3], A^+. Simplement, $A^+\mathbf{b}$ donne le minimiseur de $||AE'-\mathbf{b}||$, de plus petite norme (s'il y en a plusieurs).

Une subtilité vient du fait que l'on cherche à garantir une moyenne d'elo de 1500, et à minimiser $||E-\mathbf{c}||$ en cas d'égalité (voir point d)), où $\mathbf{c}$ est le vecteur avec 1500 sur chaque composante. Mais on peut vérifier que $\mathbf{c}^\top A^+ = 0$, donc $\boxed{E = \mathbf{c} + A^+\mathbf{b}}$ est exactement la solution que l'on cherche.

Humains *vs* bots. Au dessus, on a supposé que chaque équipe contenait deux humains, et deux bots. Le procédé s'adapte facilement pour des situations différentes, et en particulier pour un nombre différent de bot par équipe. Ceci dit, on a le choix entre :

Introduire un elo de bot, quelque soit la configuration (méthode Daligault). Par exemple, pour une partie opposant trois humains à deux humains, on aura :

$$\Delta(\alpha_1,\alpha_2,\alpha_3,\beta_1,\beta_2) \;=\; \left(\; e_{\alpha_1} \;+\; e_{\alpha_2} \;+\; e_{\alpha_3} \;\right) - \left(\; e_{\beta_1} \;+\; e_{\beta_2} \;+\; e_{\mathrm{bot}} \;\right).$$

Introduire plusieurs elos de bot, selon la configuration (méthode Lefevre), dans l'idée que la force des bots est différente selon leur nombre dans l'équipe. Dans la configuration ci-dessus, cela donne

$$\Delta(\alpha_1,\alpha_2,\alpha_3,\beta_1,\beta_2) \;=\; \left(\; e_{\alpha_1} + e_{\alpha_2} + e_{\alpha_3} \;+ e_{\mathrm{bot1}}\right) - \left(\; e_{\beta_1} + e_{\beta_2} + 2e_{\mathrm{bot2}} \;\right).$$

On peut aussi complexifier en introduisant des bots pour chaque configuration : $e_{\mathrm{2vs3}}, e_{\mathrm{3vs2}}, e_{\mathrm{2vs2}}, e_{\mathrm{bot1vs3}}$, *etc.* Dans tous les cas, les elos des bots évoluent de la même manière que les humains, suivant la même règle.

Exemple. Il y a eu une unique partie jouée, opposant Spil, Fabien, Alfred à Pierre et Jean. Le résultat est $r=-600$.

Selon la méthode Daligault, l'elo de Spil, Fabien, Alfred devient $e=1400$. L'elo de Pierre, Jean et e_{bot} devient $e=1600$.

Selon la méthode Lefevre, l'elo de Spil, Fabien, Alfred, et e_{bot1} est $e=1440$, l'elo de Pierre et Jean est $e=1560$, et l'elo de e_{bot2} est 1620.

3. `http://en.wikipedia.org/wiki/Moore-Penrose_pseudoinverse`

Imaginons désormais que Spil et Duche fassent une partie en un contre un, et que Spil gagne de $r = 1000$. Alors les nouveaux elos, selon la méthode Daligault, sont : Duche, 900 ; Fabien et Alfred, 1300 ; Pierre, Jean et e_{bot}, 1700 ; Spil, 1900.

Dans le cas de la méthode Lefevre, on obtient : Duche, 942.1 ; Fabien, Alfred, et e_{bot1}, 1384.2 ; e_{bot3}, 1500 ; Pierre, Jean, 1615.8 ; e_{bot2}, 1942.1 ; Spil, 3268.4.

Calcul du résultat d'une partie. Les maps ne sont clairement pas équivalentes, certaines pouvant apporter bien plus de points que d'autres. C'est pourquoi, plutôt que de simplement compter la différence de point, on utilisera la formule suivante pour calculer le résultat d'une partie :

$$\text{résultat} = \frac{\text{score équipe A} - \text{score équipe B}}{\text{score équipe A} + \text{score équipe B}} \times 1000.$$

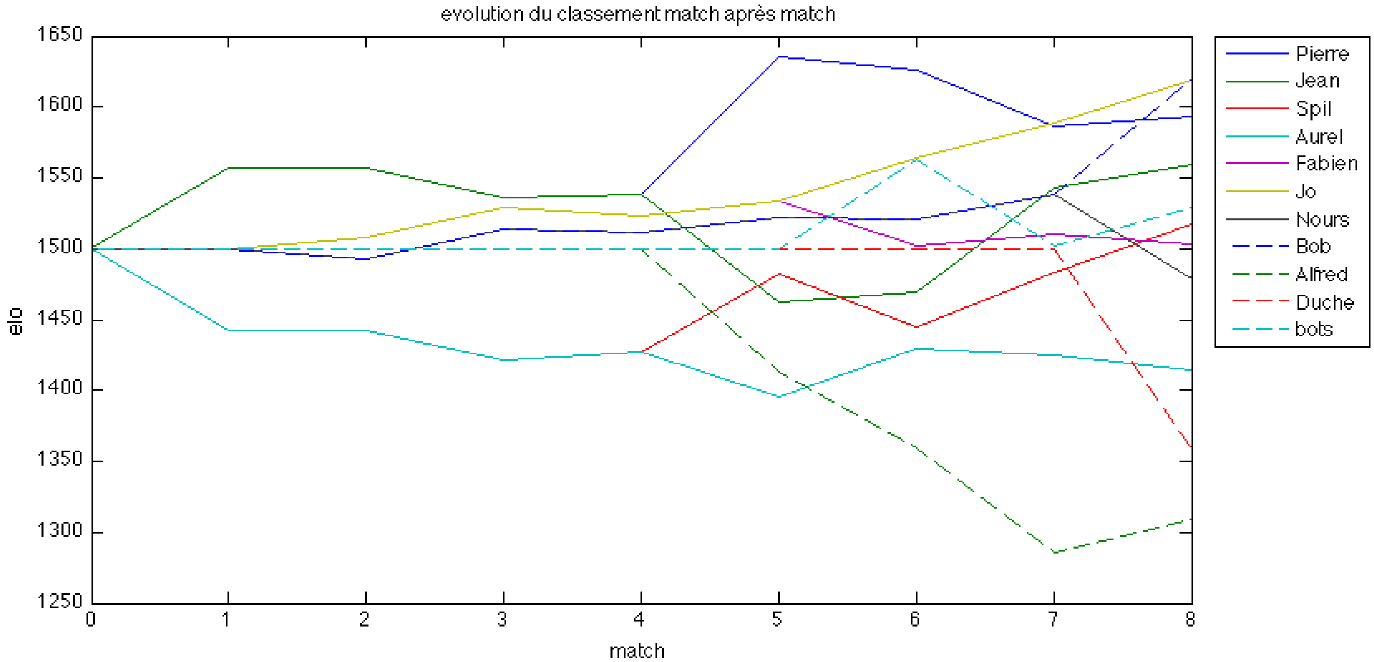
Pour ne pas pénaliser un joueur qui progresse (ni avantager ses adversaires) sur ses premières parties, on peut aussi imaginer mettre un poids sur les parties, décroissant en fontion de leur ancienneté (à discuter).

Classement à ce jour. Avec la feuille de scores suivante :

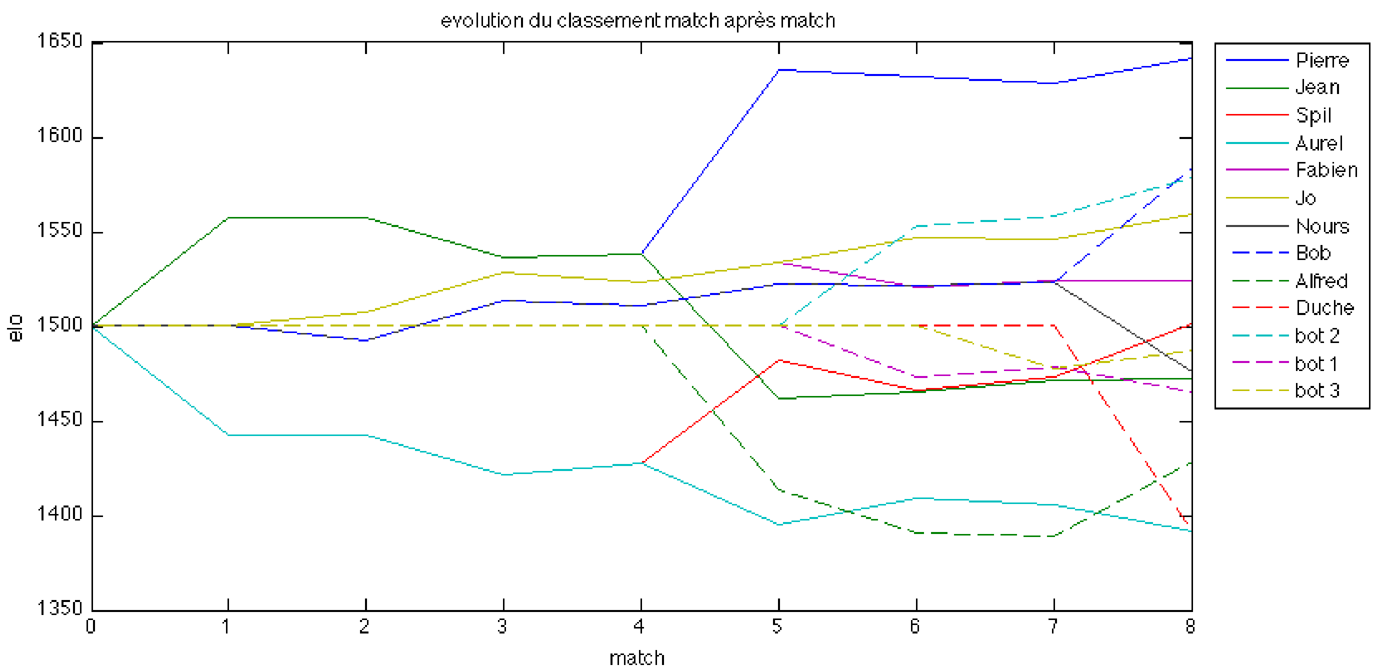
Pierre+Jean vs Spil+Aurel	1471 à 923	(Blood Harvest)
Fabien+Jo vs Nours+Bob	1809 à 1699	(Blood Harvest)
Pierre+Jean vs Nours+Bob	831 à 758	(No Mercy)
Fabien+Jo vs Spil+Aurel	1609 à 1111	(No Mercy)
Spil+Pierre vs Alfred+Jean	1580 à 964	(Dead Air)
Jean+Pierre vs Alfred+Fabien+Spil	1247 à 598	(Dead Center)
Jean+ Spil vs Alfred	1079 à 663	(The Sacrifice ?)
Jean+Alfred vs Duche + Nours	559 à 524	(The Sacrifice ?)

on obtient le classement

méthode Daligault	méthode Lefevre
Alfred : 1308.859	Aurel : 1391.321
Duche : 1357.6377	Duche : 1392.0632
Aurel : 1414.29	Alfred : 1428.1462
Nours : 1478.1772	bot 1 : 1464.9781
Fabien : 1503.4539	Jean : 1472.1345
Spil : 1517.0786	Nours : 1475.8999
bots : 1528.6871	bot 3 : 1487.6164
Jean : 1559.2735	Spil : 1501.0677
Pierre : 1593.1923	Fabien : 1524.1317
Jo : 1618.8113	Jo : 1559.1536
Bob : 1620.5395	bot 2 : 1578.2996
	Bob : 1583.8368
	Pierre : 1641.3514



(a) méthode Daligault



(b) méthode Lefevre

FIGURE 1 – Evolution du classement, match après match