Le problème de la DRH stressée

J Kentzel jkentzel@laposte.net

Partie 1 : la question

La directrice des relations humaines (DRH) d'une entreprise veut embaucher **un** employé. Des candidats se présentent tour à tour pour une audition. Après chaque audition, la DRH, supposée infaillible, attribue une note au candidat, indépendante de l'ordre de passage des candidats. On suppose a priori que toutes les notes sont différentes pour simplifier mais on verra que ça ne change rien.

La DRH a une grosse contrainte : après chaque entretien, elle doit se décider tout de suite, c'est-à-dire que si elle n'embauche pas sur-le-champ un candidat à la fin de son audition, elle ne pourra pas le rappeler plus tard et elle passe à l'audition du candidat suivant. Bien sûr, les candidats se présentent dans un ordre aléatoire et il est impossible a priori de prévoir quand se présentera le meilleur. Comment la DRH peut elle optimiser sa chance, sa probabilité, de recruter **le** meilleur candidat ? En fait, a-t-elle des 'vraies' chances de réussir ?

Remarque 1 : pour qui trouve qu'on est trop loin du réel voir des exemples proches dans la source donnée à la fin.

Conseil : ne pas lire la suite de cet article (je veux dire : ne pas la lire tout de suite !), réfléchir seulement à cette question (et idem avec vos élèves !, en vous assurant qu'ils ont bien compris la question) [entre nous : j'ai l'assez nette impression que je n'aurais aucune idée avec un tel énoncé que je trouve parfaitement déroutant !]

Partie 2 : une réponse en classe de première

Remarque 2 : disons … en théorie, je crois que c'est vraiment mieux en terminale car c'est un peu difficile.

n étant le nombre de candidats, $n > 1$, pour tout i ($\mathbf{1 \leq i < n}$) on s'intéresse à la stratégie notée **Strat (i)** qu'on décrit ainsi : la directrice doit d'abord auditionner i candidats sans en embaucher aucun. C'est la **période d'observation**. (je ne sais pas le prouver mais il semble difficile d'imaginer une stratégie ne débutant pas par une période de ce type)

Ensuite commence la **période de décision** : la première fois que la directrice voit un candidat qui est meilleur que tous ceux qui l'ont précédé, alors elle l'engage.

A vous de trouver le meilleur i (qui dépend évidemment de n) ! C'est encore dur mais on a un 'cadre' et on peut formaliser un peu. On désigne par R l'événement : 'la DRH réussit', ce qui n'aura de sens pour nous qu'avec la notation : pour tout i, $1 \leq i \leq n - 1$, $P_i(R)$ = probabilité (la DRH réussit en appliquant la stratégie Strat(i)).

Informellement on voit que si i est choisi trop grand, le meilleur candidat risque plus d'être 'seulement observé' car $P_i(\bar{R}) > \frac{i}{n}$, et si on prend i trop petit, on risque plus de croiser un 'faux meilleur' avant le vrai…

Exercice : 1) Vérifier que si $n = 2$, le seul i possible est $i = 1$ et $P_1(R)\ ) = \frac{1}{2}$.

2) Montrer que si $n = 3$, le meilleur i est $i = 1$ car $P_1(R)\ ) = \frac{1}{3} + \frac{1}{3} * \frac{1}{2} = \frac{1}{2} > P_2(R) = \frac{1}{3}$.

3) Pour tout k entre 1 et n, on note R_k l'événement : 'La DRH réussit à l'issue de la $k^{ème}$audition'. R est la réunion disjointe des k événements R_k. Bien sûr si $k \leq i$, $P_i(R_k) = 0$ et l'enjeu est calculer les $P_i(R_k)$ pour k tel que $i < k \leq n$.

Montrer que pour $n = 4$, $P_1(R) = P_1(R_2) + P_1(R_3) + P_1(R_4) = \frac{1}{4} + \frac{1}{4} * \frac{1}{2} + \frac{1}{4} * \frac{1}{3} = \frac{11}{24}$ et $P_2(R) = P_2(R_3) + P_2(R_4) = \frac{1}{4} + \frac{1}{4} * \frac{2}{3} = \frac{10}{24}$ donc, pour la troisième fois, Strat(1) est la meilleure stratégie.

On voit que pour tout i tel que $1 \leq i < n$, $P_i(R)$ est la somme des $(n - i)$ nombres non nuls $P_i(R_k)$ où $i < k \leq n$.

Montrer que pour $n = 5$, on obtient $P_2(R_3) = \frac{1}{5}$, $P_2(R_4) = \frac{1}{5} * \frac{2}{3}$ et $P_2(R_5) = \frac{1}{5} * \frac{2}{4}$ d'où $P_2(R) = \frac{26}{60}$. Strat(2) est la meilleure stratégie car $P_1(R)\ ) = \frac{25}{60}$, $P_3(R)\ ) = \frac{21}{60}$ et $P_4(R)\ ) = \frac{12}{60}$. (en passant on a toujours $P_{n-1}(R) = \frac{1}{n}$)

Remarque 3 : n = 4 ou 5, c'est pour ceux qui ont vite fait n = 3. Je ne sais pas si c'est vraiment éclairant pour la suite.

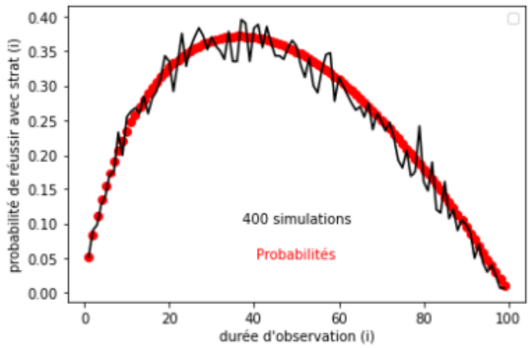
4) Il est (plus que) temps d'arrêter les bricolages. Pour tout k entre 2 et n, on définit les événements

$M_k =$ 'Le $k^{\text{ème}}$ candidat est le meilleur' et $A_k =$ 'Le meilleur des $(k-1)$ premiers était parmi les i premiers'. Montrer que pour $k > i$, $P_i(R_k) = \frac{1}{n} * \frac{i}{k-1}$. En effet $R_k = M_k \cap A_k$ d'où, remplaçant P_i par P pour être plus lisible, $P(R_k) = P(M_k).P_{M_k}(A_k)$. Montrer alors que $P_i(R) = \frac{i}{n}\left(\frac{1}{i} + \frac{1}{i+1} + \frac{1}{i+2} \dots + \frac{1}{n-1}\right)$.

5) Donner une fonction Python prenant n et i en paramètres et renvoyant $P_i(R)$ pour n candidats.

```
def p_i_de_R(n,i):
    a = 0
    for k in range(i+1,n+1):
        a = a+1/(k-1)
    a = a*i/n
    return a

Calc=[p_i_de_R(100,i) for i in range(1,100)]
```



Ci-contre on se limite à n = 100. Pour chacune des 99 valeurs possibles pour i, on a rajouté 400 simulations.

6) Quel conseil pour la DRH s'il y a 100 candidats ?

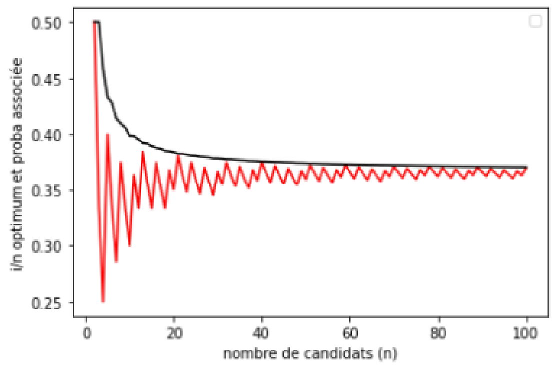
Python dit que la DRH a intérêt à utiliser la stratégie Strat (37). Elle a alors une probabilité proche de 0,37 de réussir.

7) Ces deux 37, c'est une coïncidence liée au choix de n valant 100 ?

Non. On rajoute le code qui suit puis on représente les deux séries (Num, Calc1) et (Num, Calc2). On voit alors la proportion i optimum / n = i_{opt}/n et $P_{i_{opt}}(R)$ pour chaque entier n de 2 à 100. C'est Calc1, c'est-à-dire $\frac{i_{opt}}{n}$, qui est 'dentelée', elle redescend quand on a le même ind = i_{opt} pour des n successifs (exemple vu ci-dessus : ind = i_{opt} = 1 si $2 \le n \le 4$) ; elle monte toujours sinon car si $0 < i < n$, on a $\frac{i}{n} < \frac{i+1}{n+1}$.

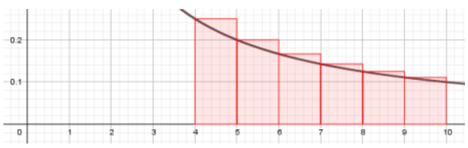
```
def calcul(n):
    max = 0 # proba max de réussir
    ind = 1 # i optimum pour réussir
    for i in range(1,n):
        a = p_i_de_R(n,i)

        if a >max:
            max =a
            ind=i
    return ind/n,max
N=101
Num=[x for x in range(2,N)]
Calc1=[calcul(j)[0] for j in range(2,N)]
Calc2=[calcul(j)[1] for j in range(2,N)]
```



Partie 3 : une réponse en classe de terminale (n 'assez grand')

On suppose dans ce qui suit que i et n, $i < n$, sont 'assez grands'. Voir le dessin ci-dessous avec i = 4 et n = 10, valeurs peu probantes, trop petites, mais lisibles sur un dessin ! Ce qui suit pourrait assez facilement être précisé.



'Comme on le sait', on a alors (i et n 'assez grands')

$\frac{1}{i} + \frac{1}{i+1} + \frac{1}{i+2} \dots + \frac{1}{n-1} \sim \int_i^n \frac{1}{t}\, dt = \mathrm{Ln}\left(\frac{n}{i}\right)$.

Pour obtenir i maximisant $P_i(R)$ pour un n donné, on peut poser $x = \frac{i}{n}$, proportion des candidats qu'il faut seulement observer, et étudier sur]0 ; 1[la fonction f définie par $f(x) = P_i(R) = x\, Ln\left(\frac{1}{x}\right) = -x\, Ln(x)$. . $f'(x) = -1 - Ln(x)$ et le maximum de f est atteint pour $x = \frac{1}{e}$ et vaut … $\frac{1}{e}$. … je me disais bien que ce 0,37 je l'avais déjà croisé !

Remarque 4 : les codes Python et une page Géogébra assez interactive redisant approximativement tout sont ici.
Source : http://images.math.cnrs.fr/Decision.html (c'est une page de l'excellent Etienne Ghys)