Le problème du collectionneur

Le 19 juin 2014
http://www.mp933.fr/ - stephane@gonnord.org

Buts du TP

Le petit François est fan de foot et veut remplir son album Panini « FIFA World Cup 2014 ». Nous allons l'aider à trouver une stratégie optimale. Petits rappels :

- il y a $N = 640$ vignettes différentes, étiquetées de 0 à $N - 1$ (c'est bien fait, n'est-ce pas ?) ;
- elles sont vendues par paquets de 5 (prises au hasard parmi les N) ou à l'unité (et on peut les choisir, mais c'est plus cher) ;
- en paquet, le prix par vignette [1] est $P_p = 0.12$€ alors qu'à l'unité, il vaut $P_u = 0,17$€.

Exercice 1 Créer un dossier pour ce TP, lancer Spyder, gnagnagna...

1 Paquets injectifs

Quelle est la probabilité pour que dans un paquet on trouve deux fois la même vignette ? c'est de l'ordre de 1 à 2%. IRL, je ne sais pas s'il y a une vérification du fait que les 5 étiquettes sont différentes ou non.

Exercice 2 Évaluer numériquement la probabilité théorique pour qu'un paquet contienne (au moins) un double :

$$p = 1 - \frac{(N-1)(N-2)(N-3)(N-4)}{N^4}.$$

Exercice 3 Écrire une fonction sans argument retournant un paquet de 5 vignettes :

```
>>> paquet()
[136, 338, 52, 623, 289]
```

Pour savoir si un paquet contient un double, on peut transformer cette liste en un ensemble (et les doublons sont alors supprimés) et compter le nombre d'éléments de l'ensemble : s'il ne vaut pas 5, c'est qu'il y avait (au moins) un doublon :

```
def differents(t):
    return len(t) == len(set(t))
```

Exercice 4 Tirer au sort 10^5 paquets différents, et compter combien possédaient au moins un double. Est-ce conforme au résultat espéré ?

2 Évolution du nombre d'étiquettes différentes

Si on achète des paquets de 5 vignettes, au début on trouve en général 5 nouvelles vignettes... puis après un certain temps, on commence par tomber sur des vignettes qu'on a déjà... puis cela devient très fréquent (et très frustrant). On va donc évaluer le nombre de vignettes différentes obtenues par un collectionneur au cours du temps.

Pour cela, on tient à jour un tableau contenant N booléens (initialisés à `False`). À chaque fois qu'on tire un numéro, on sait si cette vignette a déjà été vue. Si ce n'est pas le cas, on déclare cette vignette comme vue, et on incrémente le compteur d'étiquettes vues.

1. C'est un peu plus compliqué que cela, mais bon...

```
def album_vide():
    return [False] * nb_vignettes
```

Un algorithme pour calculer une évolution aléatoire du nombre de stickers différents pourrait être :

```
album <- album_vide()
differents <- 0     # le nombre d'étiquettes différentes
evolution <- [0]    # l'évolution au cours du temps de ce nombre
Faire n fois:
    | paquet <- un_paquet()
    | pour x dans paquet:
    |     | si  non(album[x]) # c'est une nouvelle étiquette
    |     | alors:
    |     |     | album[x] <- True
    |     |     | differents <- differents+1
    | placer differents au bout de evolution
retourner evolution
```

Exercice 5 Écrire une fonction prenant en entrée un entier n, et retournant l'évolution du nombre de vignettes différentes après l'achat de $0, 1, ..., n$ paquets de 5 vignettes :

```
>>> graphe_stickers_differents(10)
[0, 5, 10, 15, 19, 24, 29, 34, 39, 44, 48]
```

Dans l'exemple précédent, le collectionneur est tombé sur la première vignette en double au quatrième paquet. Après 10 paquets, il avait 48 *étiquettes différentes.*

On peut maintenant représenter l'évolution :

```
nb_paquets = 600

import matplotlib.pyplot as pypl
pypl.plot([5*k for k in range(nb_paquets+1)],
          graphe_stickers_differents(nb_paquets))
pypl.xlabel(u'Vignettes achetées')
pypl.ylabel(u'Vignettes différentes')
pypl.savefig('stickers_differents.pdf')
pypl.show()
```

Mathématiquement, on connaît l'espérance du nombre de vignettes différentes obtenues après avoir acheté n étiquettes aléatoires : elle est donnée par une suite $(E_n)_{n\in\mathbb{N}}$ telle que $E_0 = 0$, et $E_{n+1} = E_n + \dfrac{N - E_n}{N}$ pour tout $n \in \mathbb{N}$.

Exercice 6 Modifier le graphique précédent en ajoutant le graphe de l'espérance.

3 Une première tactique

Le petit François décide qu'il va acheter un nombre fixe de paquets, disons F, et qu'au delà de ce seuil, il achètera les vignettes restantes à l'unité. Après ces F paquets achetés, le coût total sera donc $F \times P_p \times 5 + (N - d) \times P_u$, avec d le nombre d'étiquettes *différentes* obtenues après les achats de paquets.

Exercice 7 Écrire une fonction prenant en entrée un entier n, simulant l'achat de n paquets de 5 vignettes, et calculant le coût global après achat des vignettes restantes à l'unité. Après chaque paquet, on tiendra à jour les nouvelles vignettes récupérées.
Dans l'exemple suivant, on achète 50 *puis* 200 *paquets de* 5 *vignettes.*

```
>>> achats_fixes(50)
103.27000000000001
>>> achats_fixes(200)
142.1
```

J'ai utilisé l'algorithme suivant :

```
album <- album_vide()
differents <- 0     # le nombre d'étiquettes différentes
Faire F fois:
    | paquet <- un_paquet()
    | ...
retourner F*P_p*5 + (N-differents)*P_u
```

On peut maintenant écrire une fonction évaluant le coût de cette tactique pour un nombre F donné, en réalisant un certain nombre d'expériences :

```
def tactique1(achats, nb_experiences): # nombre de paquets achetés
    prix = sum(achats_fixes(achats) for _ in range(nb_experiences))
    return prix / nb_experiences
```

Exercice 8 Réaliser un graphe représentant (sur 100 expériences) les coûts avec cette tactique, en fonction du nombre $F \in [\![0, 1000]\!]$ choisi.

4 Une autre tactique

Une autre tactique peut consister à acheter des paquets TANT QU'on a pas atteint un certain nombre d'étiquettes différentes, seuil fixé à l'avance.

Exercice 9 Écrire une fonction mettant en place effectivement cette tactique.

```
>>> achats_seuil(100)
104.66
>>> achats_seuil(500)
150.2299999999995
>>> achats_seuil(600)
230.42999999999859
```

J'ai utilisé l'algorithme suivant :

```
album <- album_vide()
cout <- 0
differents <- 0           # le nombre d'étiquettes différentes
Tant que differents < S:  # le seuil
    | paquet <- un_paquet()
    | cout <- cout +5*P_p
    | ...
retourner cout + (N-differents)*P_u
```

Exercice 10 Représenter, comme dans la partie précédente, le coût de cette tactique en fonction du seuil choisi (et toujours sur 100 expériences par seuil).

Exercice 11 La théorie dit que cette deuxième tactique est optimale... si on prend un seuil S tel que

$$P_p \frac{N}{N-S} = P_u.$$

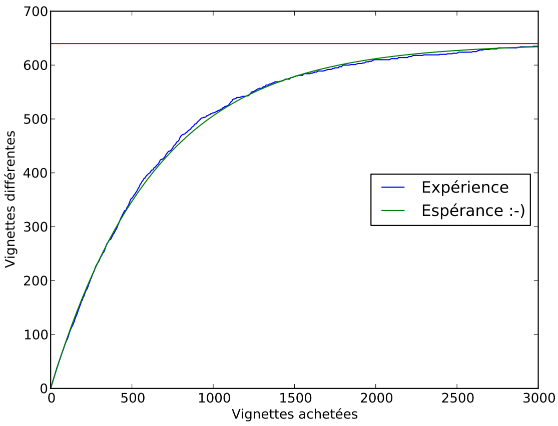
Comparez à vos résultats empiriques !

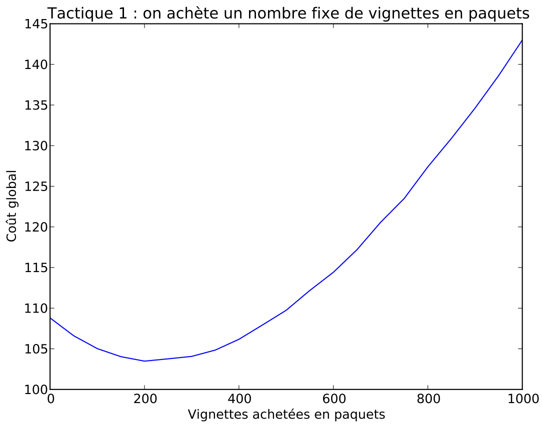
5 Conclusion(s) : IRL

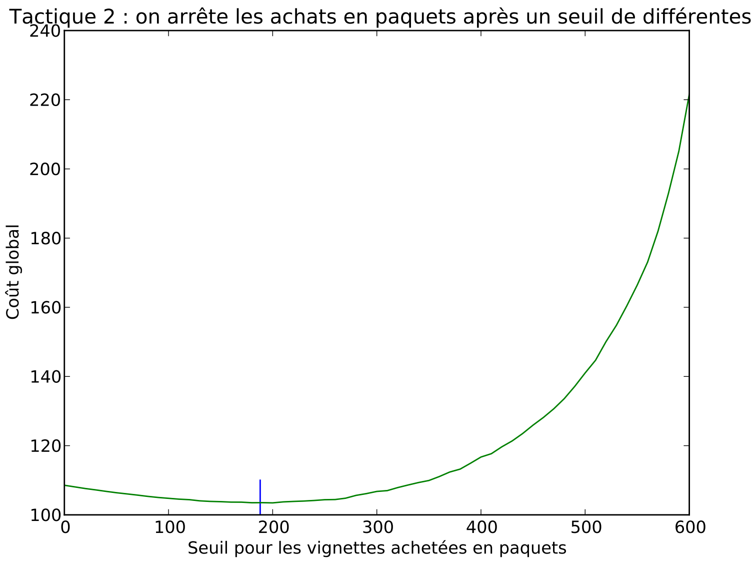
- On se convainc rapidement, à l'usage, que les paquets ne sont pas tout à fait tirés aléatoirement : certaines cartes sont certainement plus difficiles à avoir que les autres, ce qui alimente le marché !
- Sous l'hypothèse (fausse) du hasard complet, la meilleure tactique est la seconde : elle s'adapte à ce qui a déjà été tiré.
- En compliquant le modèle (bref, en le faisant un peu plus coller à la réalité), il y a de quoi faire un TIPE complet : le prix des paquets change si on on prend plus ; le prix à l'unité est en fait un peu supérieur : il y a des frais d'envoi, certains groupes de cartes sont plus rares, etc. On peut aussi faire des statistiques telles que : au boût de combien de temps a-t-on son premier double ? sa première équipe de complète ? etc. À chaque fois, on peut comparer le résultat empirique à la théorie.
- Bon, en pratique, tous ces calculs n'ont aucune valeur : ce qui compte, c'est le plaisir d'ouvrir un nouveau sachet et d'espérer trouver de nouvelles vignettes.

Enfin, à ce qu'il paraît :-)

6 Ce que je trouve







Et à la fin, c'est comme toujours (souvent) le Brésil qui gagne !