Collectionneur de vignettes

3ième année informatique et réseaux TD A
TALEB Nour Eddine
KIZIL Huseyin

ESIREM Dijon

Table des matières

Remerciements

Introduction

Dans le cadre de ce travail, nous explorons une activité populaire : la collection de vignettes. Cette activité consiste à rassembler un ensemble complet de vignettes uniques souvent distribuées aléatoirement dans des paquets de céréales par exemple. Notre étude se concentre sur le temps que peut mettre une personne pour compléter la collection. Combien de paquets de céréales une personne doit-elle acheter pour terminer sa collection de N vignettes différentes ?

Ce sujet nous intéresse pour son lien étroit avec les concepts mathématiques, en particulier la théorie des probabilités. En analysant les probabilités d'obtenir des vignettes déjà possédées ou nouvelles à chaque achat de paquet de céréales, en supposant que la personne achète 1 paquet par semaine, nous cherchons à établir des estimations théoriques pour le nombre moyen de paquets nécessaires pour achever la collection et donc la notion d'ésperances rentrera en jeu.

En outre, nous avons examiné des situations plus complexes, comme les échanges de doublons avec d'autres collectionneurs ou les offres spéciales des fabricants permettant l'acquisition de vignettes spécifiques en échange d'un certain nombre de vignettes (par exemple 10 vignettes contre 1 vignette spécifique). Il est clair que ces éléments vont influencer au niveau du temps de collecte des vignettes et c'est ce que nous allons démontrer durant notre étude.

L'objectif principal de notre étude est donc de développer un modèle mathématique solide pour estimer le temps moyen nécessaire à un collectionneur pour finaliser sa collection de vignettes. Nous analysons également diverses stratégies et conditions qui pourraient affecter ce temps.

Au fil de ce rapport, nous présentons notre approche mathématique, discutons des résultats théoriques obtenus à l'aide des lois de probabilités usuelles et analysons différentes situations pour mieux comprendre le problème du collectionneur de vignettes. Pour cela, nous avons réalisé un programme informatique en C++ et nous avons fait plusieurs simulations à l'aide de RStudio.

Modèle de base

2.1 étude théorique

Lorsque l'on suppose que toutes les vignettes sont équiprobables, c'est-à-dire que chaque type de vignette a une probabilité de $\frac{1}{n}$ d'être obtenu lors de l'achat d'une boîte, on peut étudier la durée nécessaire pour compléter une collection de vignettes.

Soit T la variable aléatoire qui représente le nombre de paquets à acheter (et par extension, le nombre de semaines nécessaires) pour compléter la collection de vignettes.

L'obtention d'une nouvelle vignette dans un paquet k suit une loi géométrique de paramètre $p = \frac{n-(k-1)}{n}$, où n est le nombre total de vignettes à collecter. Ainsi, $T_{n,k}$, une variable aléatoire représentant le nombre de paquets à acheter pour obtenir une nouvelle vignette lorsque vous en avez déjà k différentes, suit également une loi géométrique de paramètre p.

L'espérance de $T_{n,k}$ est donnée par : $E(T_{n,k}) = \frac{1}{p} = \frac{n}{n-(k-1)}$

Par conséquent, la variable aléatoire T qui représente le temps nécessaire pour compléter la collection entière est la somme de toutes les $T_{n,k}$ où k varie de 1 à n,

$$T = \sum_{k=1}^{n} T_{n,k}$$

En utilisant la linéarité de l'espérance, l'espérance de T est :

$$\begin{aligned}
E(T_n) &= \sum_{k=1}^{n} E(T_{n,k}) \\
&= \sum_{k=1}^{n} \frac{n}{n-(k-1)} \\
&= n\left(\frac{1}{n} + \frac{1}{n-1} + \frac{1}{n-2} + \ldots + \frac{1}{2} + \frac{1}{1}\right) \\
&= n \cdot \left(1 + \frac{1}{2} + \ldots + \frac{1}{n-2} + \frac{1}{n-1} + \frac{1}{n}\right) \\
&= n \cdot \sum_{k=1}^{n} \frac{1}{k}
\end{aligned}$$

D'où

$$E(T_n) = n \cdot H_n$$

En utilisant le développement asymptotique de H_n, on obtient :

$$E(T_n) = n \cdot H_n = n \cdot \ln(n) + \gamma \cdot n + \frac{1}{2} + o(1)$$

Où $\gamma \approx 0.5772156649$ est la constante d'Euler-Mascheroni.

REF : https://fr.wikipedia.org/wiki/ProblÃÍme_du_collectionneur_de_vignettes

2.2 Simulations

Les simulations Monte Carlo sont utilisées pour trouver des solutions (approximatives) à des problèmes de probabilité, surtout en inférence statistique lorsque des solutions exactes ne sont pas connues ou sont très difficiles à calculer. Le problème du collectionneur de vignettes, tel que nous le nommons, est particulièrement adapté à cela.

La solution Monte Carlo est simplement un algorithme : nous simulons un certain nombre de paquets de céréales dans lequels des vignettes sont présentes et comptons combien de paquets sont nécessaires pour remplir un une collection séquentiellement. Après avoir répété cette procédure un certain nombre de fois, on calcule simplement la moyenne du nombre de paquets nécessaires pour remplir les albums. Pour ce faire, nous avons codé en language C++ et nous avons réalisé les courbes à l'aide du logiciel R.studio.

L'algorithme doit reproduire les conditions décrites dans le problème : ici, l'album comporte par exemple 500 vignettes, chaque paquet contient d'abord 1 vignette pour le cas simple puis nous verrons avec 5 vignettes distinctes, et les vignettes seront considérées comme réparties uniformément parmi les paquets. Ses étapes sont les suivantes :

1. Nous avons créer un tableau avec les numéros des vignettes que nous possédons. Au départ, sa longueur est de zéro.
2. Nous avons pris k = 1 où k est le nombre de paquets.
3. Nous avons simulé un paquet en prenant un échantillon de taille 1, sans remplacement, à partir d'une urne contenant 500 balles numérotées de 1 à 500.
4. Nous avons vérifié la vignette dans le paquet de l'étape 3 par rapport à celles dans le tableau de l'étape 1 et nous avons inclus toute vignette qui n'est pas encore présente. Nous Supprimons les vignettes que nous possédons déjà.
5. Si le tableau a une longueur de 500, alors on arrête. Sinon, on incrémente k de un et on retourne à l'étape 3.

En éxecutant notre programme, voici ce qu'on obtient :
Dans un premier temps, nous entrons nos différentes valeurs (voir Figure 1).

```
Entrez nombre de vignettes de votre collection (0 pour arreter programme)
Ici >  500
Entrez nombre de simulations (0 pour arreter programme)
Ici >  1000
Entrez nombre de vignettes (0 pour arreter programme)
Ici >  5
```

FIGURE 1 – Entrée valeurs simulation

Voici les résutats obtenus (voir Figure 2 et 3) :

```
-Simulation :
>> Il faut en moyenne 3402.73 semaines pour completer la collection de 500 vignettes sans echange.
```

```
>> Il faut en moyenne 685.576 semaines pour completer la collection de 500 vignettes avec 5 dans la boite de cereales
```

FIGURE 2 – Résultats de la simulation

Soit T la variable aléatoire représentant le nombre de paquets nécessaires pour compléter la collection. Nous nous intéressons à la valeur attendue de T, donc nous répétons simplement la procédure ci-dessus

```
-Valeur theorique :
>> Il faut en moyenne 3396.41 semaines pour completer la collection de 500 vignettes sans echange a l'aide de la formule
 theorique n*somme(1/k).
>> Il faut en moyenne 3396.41 semaines pour completer la collection de 500 vignettes sans echange a l'aide de l'approxim
ation de la forumule theorique.
```

Figure 3 – Résultats théoriques

pour un grand nombre d'albums et prenons la moyenne arithmétique des valeurs finales de k. À titre d'exemple, nous avons exécuté l'algorithme pour 1000 "albums" et avons obtenu 3482,73 comme estimation du nombre attendu de paquets nécessaires pour compléter un album avec 1 vignette dans un paquet. Il convient de souligner que, en tant qu'approximation stochastique, l'algorithme donne généralement une réponse différente à chaque exécution (la réponse 3483 mentionnée précédemment vient de l'arrondi à l'entier supérieur de 3482,73). En comparant cela à la valeur théorique qui de 3397 en prenant l'arrondi supérieur (voir Figure 3), on voit une petite différence.

Et pour la simulation avec 5 vignettes dans un paquet, nécéssairement,le temps de collecte diminue (voir Figure 2).

Stratégies de Collecte

3.1 Collecte de Vignettes avec possibilité d'échange

3.2 Collecte de Vignettes avec ...

3.3 Collecte de Vignettes avec ...

Conclusion

Mes conclusions

Table des figures

Liste des tableaux

Bibliography

[AAC+08] S. Abiteboul, T. Allard, P. Chatalic, G. Gardarin, A. Ghitescu, F. Goasdoué, I. Manolescu, B. Nguyen, M. Ouazara, A. Somani, N. Travers, and G. Vasile. WebContent : Efficient P2P Warehousing of Web Data. In *VLDB*, pages 1428–1431, 2008.

[ACTV10] B. Amann, J. Creus, N. Travers, and D. Vodislav. ROSES et l'agrégation Web avancée. In *BDA*, volume 26, pages 1–6, 2010.

[CAC+12] J. Creus, B. Amann, V. Christophides, N. Travers, and D. Vodislav. Optimisation de grandes collections de requêtes d'agrégation RSS. *ISI*, pages 1–28, 2012.

[CATV10] J. Creus, B. Amann, N. Travers, and D. Vodislav. Un agregateur de flux rss avancé. In *BDA*, 2010.

[CATV11a] J. Creus, B. Amann, N. Travers, and D. Vodislav. Optimizing large collections of continuous content-based RSS aggregation queries. In *BDA*, pages 1–19, 2011.

[CATV11b] J. Creus, B. Amann, N. Travers, and D. Vodislav. RoSeS : A continuous content-based query engine for RSS feeds. In *DEXA*, volume 6861 of *LNCS*, pages 203–218, 2011.

[CATV11c] J. Creus, B. Amann, N. Travers, and D. Vodislav. RoSeS : A continuous query processor for large-scale RSS filtering and agregation. In *CIKM*, pages 2549–2552, 2011.

[CdMR+12a] C. Constantin, C. du Mouza, P. Rigaux, V. Thion, and N. Travers. A Desktop Interface over Distributed Document Repositories. In *EDBT*, pages 104–107, 2012.

[CdMR+12b] C. Constantin, C. du Mouza, P. Rigaux, V. Thion, and N. Travers. Browse Your Content-Based Distributed Repository ! In *BDA*, pages 1–5, 2012.

[DNT07] T.-T. Dang-Ngoc and N. Travers. Tree Graph Views for a Distributed Pervasive Environment. In *NBiS*, pages 406–415, 2007.

[FSRT16a] R. Fournier-S'niehotta, P. Rigaux, and N. Travers. A Digital Score Library Based on MEI. In *(MEC'16) Music Encoding Conference*, pages 1–4, 2016.

[FSRT16b] R. Fournier-S'niehotta, P. Rigaux, and N. Travers. Is There a Data Model in Music Notation ? In *TENOR*, pages 85–91. Anglia Ruskin University, 2016.

[FSRT16c] R. Fournier-S'niehotta, P. Rigaux, and N. Travers. Querying Music Notation. In *TIME*, pages 1–9, Kongens Lyngby, Denmark, October 2016.

[FSRT16d] R. Fournier-S'niehotta, P. Rigaux, and N. Travers. Vers un Traitement Algébrique de la Notation Musicale. In *JIM*, volume 23, pages 61–70, 2016.

[HdMT13] Z. Hmedeh, C. du Mouza, and N. Travers. An Intelligent PubSub Filtering System. In *BDA*, pages 1–5, 2013.

[HdMT15a] Z. Hmedeh, C. du Mouza, and N. Travers. A Real-time Filtering by Novelty and Diversity for Publish/Subscribe Systems. In *SSDBM*, pages 1–4, 2015.

[HdMT15b] Z. Hmedeh, C. du Mouza, and N. Travers. TDV-based Filter for Novelty and Diversity in a Real-time Pub/Sub System. In *IDEAS*, volume 19, pages 136–145, 2015.

[HKC+12a] Z. Hmedeh, H. Kourdounakis, V. Christophides, C. du Mouza, M. Scholl, and N. Travers. Indexes Analysis for Matching Subscriptions in RSS feeds. In *BDA*, pages 1–20, 2012.

[HKC+12b] Z. Hmedeh, H. Kourdounakis, V. Christophides, C. du Mouza, M. Scholl, and N. Travers. Subscription indexes for web syndication systems. In *EDBT*. ACM, 2012.

[HKC+12c] Z. Hmedeh, H. Kourdounakis, V. Christophides, C. du Mouza, M. Scholl, and N. Travers. Subscription Indexes for Web Syndication Systems . In *EDBT*, pages 311–322, 2012.

[HKC+13] Z. Hmedeh, H. Kourdounakis, V. Christophides, C. du Mouza, M. Scholl, and N. Travers. Techniques d'indexation de souscriptions pour la syndication web. *ISI*, 18(4) :33–58, 2013.

[HKC+16] Z. Hmedeh, H. Kourdounakis, V. Christophides, C. du Mouza, M. Scholl, and N. Travers. Content-Based Publish/Subscribe System for Web Syndication. *JCST*, 31(2) :357–378, 2016.

[HTV+11] Z. Hmedeh, N. Travers, N. Vouzoukidou, V. Christophides, C. du Mouza, and M. Scholl. Everything you would like to know about RSS feeds and you are afraid to ask. In *BDA*, pages 1–20, 2011.

[HVT+11] Z. Hmedeh, N. Vouzoukidou, N. Travers, V. Christophides, C. du Mouza, and M. Scholl. Characterizing Web Syndication Behavior and Content. In *WISE*, pages 29–42. Springer Heidelberg, 2011.

[MTX+09] M. Mesiti, M. Traian, Li Xiong, S. Muller, H. Naacke, B. Novikov, G. Raschia, I. Sanz, P. Sens, D. Shaporenkov, and N. Travers. *Proc. EDBT/ICDT Workshops*. ACM, 2009.

[TDN07a] N. Travers and T.-T. Dang-Ngoc. An extensible rule transformation model for XQuery optimization - rules pattern for XQuery tree graph view. In *ICEIS*, pages 351–358, 2007.

[TDN07b] N. Travers and T.-T. Dang-Ngoc. XLive : Integrating Source With XQuery. In *WebIST*, pages 230–233, 2007.

[TDNL07a] N. Travers, T.-T. Dang-Ngoc, and T. Liu. TGV : A Tree Graph View for Modeling Untyped XQuery. In *DASFAA*, pages 1001–1006, 2007.

[TDNL07b] N. Travers, T.-T. Dang-Ngoc, and Tianxiao Liu. Untyped XQuery Canonization. In *AP-Web/WAIM Workshops*, pages 358–371, 2007.

[THV+14] N. Travers, Z. Hmedeh, N. Vouzoukidou, C. du Mouza, V. Christophides, and M. Scholl. RSS feeds behavior analysis, structure and vocabulary. *IJWIS*, 10(3) :291–320, 2014. Top 3 of IJWIS journal papers for 2014.

[Tra12] N. Travers. *Web Data Management*, chapter Putting into Practice : Full-Text Indexing with LUCENE, pages 355–363. Cambridge University Press, 2012.

Abstract

In English

Résumé

En Français