Echantillonnage de motifs séquentiels sous contrainte sur la norme

Lamine Diop[***], Cheikh Talibouya Diop[**], Arnaud Giacometti[*], Dominique Li[*], Arnaud Soulet[*]

[*]Université de Tours, France
{arnaud.giacometti, dominique.li, arnaud.soulet}@univ-tours.fr
[**]Université Gaston Berger de Saint-Louis, Sénégal
{diop.lamine3,cheikh-talibouya.diop}@ugb.edu.sn

Résumé. L'échantillonnage de motifs est une méthode non-exhaustive pour découvrir des motifs pertinents qui assure une bonne interactivité tout en offrant des garanties statistiques fortes grâce à sa nature aléatoire. Curieusement, une telle approche explorée pour les motifs ensemblistes et les sous-graphes ne l'a pas encore été pour les données séquentielles. Dans cet article, nous proposons la première méthode d'échantillonnage de motifs séquentiels. Outre le passage aux séquences, l'originalité de notre approche est d'introduire une contrainte sur la norme pour maîtriser la longueur des motifs tirés et éviter l'écueil de la « longue traîne ». Nous démontrons que notre méthode fondée sur une procédure aléatoire en deux étapes effectue un tirage exact. Malgré le recours à un échantillonnage avec rejet, les expérimentations montrent qu'elle reste performante.

1 Introduction

Les motifs séquentiels ont été introduits par Agrawal et Srikant (1995) il y a plus de 20 ans et leur utilité a été prouvée dans différents domaines de recherche et d'applications comme la fouille d'usage du Web, la fouille de textes, la bioinformatique, la détection de fraudes, etc. Depuis la première publication, de nombreuses méthodes ont optimisé l'extraction des motifs séquentiels (Zaki, 2001; Pei et al., 2001) et ont introduit des variantes (Lo et al., 2008; Gomariz et al., 2013). Malgré toutes ces avancées, l'extraction des motifs séquentiels reste une tâche coûteuse qui génère souvent trop de motifs. Cette limite aussi atteinte par l'extraction des motifs ensemblistes a été contournée par l'échantillonnage de motifs. Une telle approche tire un nombre limité de motifs où la probabilité de tirer un motif est proportionnelle à sa fréquence. Cette approche a l'avantage de contrôler la taille de la sortie et d'apporter une collection de motifs qui reflète l'intégralité de l'espace de recherche. A notre connaissance, une telle approche n'a encore pas été envisagée pour les motifs séquentiels.

Adapter la procédure d'échantillonnage de motifs en deux étapes (Boley et al., 2011) aux données séquentielles n'est pas trivial. D'une part, une limite importante de l'échantillonnage de motifs est d'avoir tendance à retourner des motifs rares correspondant à la longue traîne. En effet, la longue traîne signifie que la très grande majorité des motifs ont une fréquence très faible et elle occulte les motifs les plus fréquents. Ce problème est exacerbé dans le cas

des séquences où le nombre de motifs séquentiels de fréquence 1 explose dans les jeux de données réels. Malgré un tirage proportionnel à la fréquence, l'échantillonnage se concentrerait uniquement sur des séquences très longues et de fréquence 1. Pour éviter cet écueil de la longue traîne, nous choisissons d'introduire une contrainte sur la norme (i.e., sur le nombre d'items) pour contrôler la taille des motifs tirés. D'autre part, le coeur de cette approche requiert de dénombrer pour chaque séquence le nombre de sous-séquences distinctes. Cette tâche n'est pas aisée car une même séquence peut contenir plusieurs occurrences d'une même sous-séquence. A cette fin, nous généraliserons le travail de Egho et al. (2015) afin de dénombrer les sous-séquences en tenant compte de la norme.

Dans cet article, notre objectif est d'échantillonner les motifs séquentiels proportionnellement à la fréquence avec une contrainte sur la norme. Premièrement, dans la section 4, nous proposons une méthode en deux étapes grâce à la généralisation de la formule de dénombrement des sous-séquences de Egho et al. (2015). Nous démontrons que cette méthode effectue un échantillonnage exact. Deuxièmement, dans la section 5, nous expérimentons cette approche sur plusieurs jeux de données réels. Nous montrons que notre approche est suffisamment performante pour retourner des centaines de motifs séquentiels par seconde. Nous montrons également l'apport de la contrainte sur la norme pour mieux maîtriser la qualité des motifs retournés et éviter la malédiction de la longue traîne.

2 Travaux relatifs

Cet état de l'art distingue les méthodes d'échantillonnage de motifs en entrée et en sortie. L'échantillonnage en entrée (Toivonen et al., 1996) consiste à régénérer depuis un échantillon de données tous les motifs qui auraient été extraits depuis le jeu de données complet. L'échantillonnage en sortie (Al Hasan et Zaki, 2009) consiste à générer un échantillon de motifs parmi les motifs qui auraient été extraits depuis le jeu de données complet. Plusieurs approches ont été proposées pour l'échantillonnage en entrée des motifs séquentiels (Raissi et Poncelet, 2007), mais à notre connaissance, cet article propose la première approche d'échantillonnage de motifs séquentiels en sortie. Comme la complexité de l'échantillonnage de motifs est indépendante de la taille du langage, elle est propice aux langages structurés dont la combinatoire est forte. D'ailleurs, des méthodes ont été proposées pour les sous-graphes.

Plusieurs procédures ont été proposées pour l'échantillonnage de motifs. La première famille (Al Hasan et Zaki, 2009) repose sur les méthodes de Monte-Carlo par chaînes de Markov. L'idée est que la loi stationnaire de la marche aléatoire corresponde à la distribution à échantillonner. La limite de telles approches stochastiques est la vitesse de convergence qui peut être lente. La seconde famille (Boley et al., 2011) consiste à tirer une instance du jeu de données, puis à tirer un motif contenu dans cette instance. En choisissant judicieusement les deux distributions de tirage, il est alors possible d'obtenir un tirage exact selon la distribution désirée. Nous avons opté pour une telle approche en deux étapes pour sa rapidité et sa précision. Outre la difficulté de traiter des séquences plutôt que des itemsets, nous avons également ajouté une contrainte sur la norme des motifs. A notre connaissance, seule une approche (Dzyuba et al., 2017) permettrait de traiter à la fois des langages complexes et des contraintes. Fondée sur la satisfaction de contraintes, elle requière de disposer d'un solveur intégrant efficacement des contraintes XOR et elle n'a été utilisée que pour des motifs ensemblistes. Par ailleurs, Dzyuba

et al. (2017) soulignent que leur approche générique rivalisera difficilement avec des approches dédiées à un seul langage et/ou classe de contraintes.

3 Préliminaires

Après avoir rappelé quelques définitions, cette section formalise le problème de l'échantillonnage de motifs séquentiels sous-contraintes sur la norme.

3.1 Définitions

Soit $\mathcal{I}$ un ensemble fini de littéraux nommés *items*. Un *itemset* ou *motif* X est un sous-ensemble non vide de $\mathcal{I}$. Une *séquence* s définie sur $\mathcal{I}$ est une liste ordonnée $s = \langle X_1, \ldots, X_n \rangle$ d'itemsets non-vides $X_i \subseteq \mathcal{I}$ ($1 \leq i \leq n$, $n \in \mathbb{N}$). n est la *taille* de la séquence s noté $|s|$. La *norme* d'une séquence s, notée $\|s\|$, est la somme des cardinalités de ses itemsets, i.e. $\|s\| = \sum_{i=1}^{n} |s_i|$. Par la suite, on note s^l le préfixe $\langle X_1, \ldots, X_l \rangle$ de s ($0 \leq l \leq n$, $l \in \mathbb{N}$), s^0 étant la séquence vide (représentée par $\langle\rangle$) et $s[j] = X_j$ le j-ième itemset de s ($1 \leq j \leq n$, $j \in \mathbb{N}$). Enfin, on note $\mathbb{S}$ l'ensemble universel de toutes les séquences définies sur $\mathcal{I}$, et une base de données séquentielles $\mathcal{S}$ sur $\mathcal{I}$ est un multi-ensemble de séquences définies sur $\mathcal{I}$.

Nous rappelons maintenant les définitions de *sous-séquences* et d'*occurrences* d'une sous-séquence dans une séquence donnée.

Définition 1 (Sous-séquence) *Une séquence $s' = \langle X'_1, \ldots, X'_m \rangle$ est une sous-séquence d'une séquence $s = \langle X_1, \ldots, X_n \rangle$, noté $s' \sqsubseteq s$, s'il existe une séquence d'indices $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ telle que pour tout $j \in [1..m]$, on ait $X'_j \subseteq X_{i_j}$. Etant donnée une séquence s, on note $\phi(s)$ l'ensemble des sous-séquences de s, i.e. $\phi(s) = \{s' \in \mathbb{S} \mid s' \sqsubseteq s\}$, et $\Phi(s)$ la cardinalité de cet ensemble, i.e. $\Phi(s) = |\phi(s)|$.*

Etant donnée une séquence $s = \langle X_1, \ldots, X_n \rangle$, une sous-séquence $s' = \langle X'_1, \ldots, X'_m \rangle$ de s peut apparaître plusieurs fois au sein de s s'il existe plusieurs séquences d'indices $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ telles que pour tout $j \in [1..m]$, on ait $X'_j \subseteq X_{i_j}$. Dans ce cas, on parle d'*occurrences* multiples de la sous-séquence s' au sein de s. La définition suivante précise comment ces différentes occurrences peuvent être représentées.

Définition 2 (Occurrence) *Etant donnée une séquence $s = \langle X_1, \ldots, X_n \rangle$, une liste ordonnée d'itemsets $o = \langle Z_1, \ldots, Z_n \rangle$ de même taille que s est une occurrence d'une sous-séquence $s' = \langle X'_1, \ldots, X'_m \rangle$ de s s'il existe une séquence d'indices $1 \leq i_1 < \cdots < i_m \leq n$ telle que pour tout $j \in \{i_1, \ldots, i_m\}$, on ait $Z_{i_j} = X'_j$, et tout $j \in \{1, \ldots, n\} \setminus \{i_1, \ldots, i_m\}$, on ait $Z_j = \emptyset$. Cette séquence d'indices, appelée signature de o, est unique par définition.*

Exemple 1 *Pour $s = \langle (ab)(cd)(ce) \rangle$, $o_1 = \langle (a)(c)\emptyset \rangle$ et $o_2 = \langle (a)\emptyset(c) \rangle$ sont deux occurrences de $s' = \langle (a)(c) \rangle$ avec pour signature respective $\langle 1, 2 \rangle$ et $\langle 1, 3 \rangle$.*

3.2 Formalisation du problème

Une méthode d'extraction de motifs par échantillonnage a généralement pour objectif de tirer aléatoirement un motif par rapport à une mesure d'intérêt donnée. Dans notre cas, la mesure considérée est la fréquence du motif dans une base de données séquentielles.

Définition 3 (Fréquence) *Etant données une base de données séquentielles $\mathcal{S}$ définie sur $\mathcal{I}$ et une sous-séquence $s \in \mathbb{S}$. La fréquence de s dans $\mathcal{S}$, noté $freq(s, \mathcal{S})$ ou plus simplement $freq(s)$, est définie par : $freq(s, \mathcal{S}) = |\{s' \in \mathcal{S} \mid s \sqsubseteq s'\}|$.*

Notre objectif est de tirer aléatoirement des motifs séquentiels par rapport à la fréquence et sous une contrainte de norme. Etant donnés deux entiers m et M tels que $m \leq M$, on notera $\mathbb{S}_{[m,M]}$ l'ensemble des séquences de $\mathbb{S}$ de norme comprise entre m et M, i.e. $\mathbb{S}_{[m,M]} = \{s \in \mathbb{S} \mid m \leq \|s\| \leq M\}$. Le problème posé peut finalement s'énoncer comme suit :

Etant données une base de données séquentielles $\mathcal{S}$, des normes minimale m et maximale M, notre problème consiste à tirer aléatoirement une sous-séquence $s \in \mathbb{S}_{[m,M]}$ telle que la probabilité de tirage $p(s)$ de s soit égale à la fréquence de s dans $\mathcal{S}$ normalisé par la somme des fréquences des sous-séquences de $\mathcal{S}$ dans $\mathbb{S}_{[m,M]}$, i.e.

$$p(s) = \frac{freq(s, \mathcal{S})}{\sum_{s' \in \mathbb{S}_{[m,M]}} freq(s', \mathcal{S})}.$$

4 Méthode d'échantillonnage en deux étapes sous contrainte

4.1 Aperçu de l'approche

Dans l'approche proposée par Boley et al. (2011), les auteurs montrent comment échantillonner des itemsets proportionnellement à leur support dans une base de données transactionnelles. Nous proposons d'utiliser une solution comparable en deux étapes, mais en ajoutant une contrainte sur la norme des motifs extraits.

Tirage d'une séquence Soient $\mathcal{S}$ une base de données séquentielles et deux entiers m et M tels que $m \leq M$. Dans une première étape (voir lignes 1 et 2 de l'algorithme 1), nous commençons par calculer pour toute séquence $s \in \mathcal{S}$ le nombre $\Phi_{[m,M]}(s)$ de sous-séquences de s de norme comprise entre m et M, i.e. $\Phi_{[m,M]}(s) = |\{s' \sqsubseteq s \mid m \leq \|s\| \leq M\}|$. En se basant sur les travaux de Egho et al. (2015), nous montrons dans la section 4.2 comment calculer un tel nombre de sous-séquences. Ensuite, cette première étape se poursuit par le tirage aléatoire d'une séquence s de $\mathcal{S}$ proportionnellement à son poids $w(s) = \Phi_{[m,M]}(s)$.

Tirage d'un motif séquentiel Dans la deuxième étape, nous commençons par tirer aléatoirement (ligne 3 de l'algorithme 1), la norme k de la sous-séquence de s qui sera finalement retournée. Ce nombre k est tiré proportionnellement au nombre de sous-séquences de s de norme exactement égale à k, i.e. selon la distribution de probabilité $\mathbb{P}_{[m,M]}$ définie pour tout $k \in [m..M]$ par : $\mathbb{P}_{[m,M]}(k) = \frac{\Phi_{[k,k]}(s)}{\Phi_{[m,M]}(s)}$. Finalement, l'algorithme 1 retourne à la ligne 4 une sous-séquence s' de s de norme k selon une distribution uniforme, ce qui signifie que toute sous-séquence s' de s de norme k sera tirée avec la même probabilité $\frac{1}{\Phi_{[k,k]}(s)}$. Nous montrons dans la section 4.3 comment effectuer un tel tirage uniforme grâce à une méthode par rejet. Le problème principal posé est qu'une sous-séquence s' de s peut avoir plusieurs occurrences dans s et qu'il ne faut donc pas tirer avec une probabilité plus élevée des sous-séquences de s ayant un nombre d'occurrences plus important.

Algorithm 1 Echantillonnage de motifs séquentiels sous contraintes de norme

Input: Une base de données séquentielles $\mathcal{S}$, et deux entiers m et M tels que $m \leq M$
Output: Une sous-séquence $s \in \mathbb{S}_{[m,M]}$ tirée aléatoirement, i.e. $s \sim freq(\mathbb{S}_{[m,M]}, \mathcal{S})$
1: Soient les poids w définis par $w(s) = \Phi_{[m,M]}(s)$ pour tout $s \in \mathcal{S}$
2: Tirer une séquence de $\mathcal{S}$ proportionnellement à w : $s \sim w(\mathcal{S})$
3: Tirer un entier k entre m et M selon la distribution $\mathbb{P}_{[m,M]}(k)$
4: **return** Une sous-séquence $s' \sim u(\{s' \sqsubseteq s \mid \|s'\| = k)$ de s où u est la distribution uniforme

4.2 Poids et tirage d'une séquence

Dans cette section, nous montrons comment calculer le nombre de sous-séquences d'une séquence s sous contrainte de norme en généralisant la proposition de Egho et al. (2015). La principale difficulté est de ne pas compter plusieurs fois une même sous-séquence même si elle possède plusieurs occurrences dans s.

Sans contrainte sur la norme Soient une séquence $s = \langle X_1, \ldots, X_n \rangle$ et un itemset Y. Par la suite, nous notons $s \circ Y$ la concaténation de s et Y définie par : $s \circ Y = \langle X_1, \ldots, X_n, Y \rangle$. Intuitivement, si Y est disjoint de tous les itemsets de la séquence s, il est aisé de vérifier que le nombre de sous-séquences distinctes de $s \circ Y$ est égal au nombre de sous-séquences de s multiplié par le nombre de sous-ensembles de Y, i.e. $\Phi(s \circ Y) = \Phi(s) \times 2^{|Y|}$. Si Y n'est pas disjoint des itemsets de S, $\Phi(s \circ Y)$ sera inférieur à $\Phi(s) \times 2^{|Y|}$ et Egho et al. (2015) introduisent un terme correcteur $R(s, Y)$ pour calculer le nombre exact de sous-séquences distinctes. Pour ce faire, ils commencent par introduire un ensemble de positions précisant où les répétitions d'items de l'itemset Y sont localisées dans la séquence s :

Définition 4 (Ensemble de positions - Egho et al. (2015)) *Soient une séquence s et un itemset Y. $L(s, Y) = \{i \in \mathbb{N} \mid i \leq |s| \wedge s[i] \cap Y \neq 0 \wedge (\forall j > i)(s[i] \cap Y \not\subseteq s[j] \cap Y)\}$ est l'ensemble des positions où l'itemset Y a une intersection maximale avec les différents itemsets de s.*

Exemple 2 *Soit la séquence $s = \langle (ab)c(ac) \rangle$. Nous avons $s^1 = \langle (ab) \rangle$, $s[2] = (c)$ et $L(s^1, s[2]) = \emptyset$ car $s[2]$ n'intersecte aucun itemset de s^1. Calculons maintenant $L(s^2, s[3])$. $s[3] = (ac)$ intersecte à la fois le premier itemset $s[1] = (ab)$ de s ($s[1] \cap s[3] = (a)$) et le second itemset $s[2] = (c)$ de s ($s[2] \cap s[3] = (c)$). De plus, ces deux intersections sont disjointes. Par conséquent, nous avons $L(s^2, s[3]) = \{1, 2\}$, ce qui indique qu'en concaténant des sous-ensembles de $s[3]$ à des sous-séquences de s^2 des répétitions de sous-séquences de s^2 pourront être générées du fait que des items de $s[3]$ se retrouvent aux postions 1 et 2 de s^2.*

A partir de cet ensemble de positions, il est possible de calculer le nombre de sous-séquences distinctes d'une séquence s grâce à la formule récursive suivante.

Théorème 1 (Nombre de sous-séquences - Egho et al. (2015)) *Etant donnés une séquence s et un itemset Y, le nombre de sous-séquences distinctes de $s \circ Y$, noté $\Phi(s \circ Y)$, est défini par $\Phi(s \circ Y) = \Phi(s) \times 2^{|Y|} - R(s, Y)$ où $R(s, Y)$ est un terme correcteur :*

$$R(s, Y) = \sum_{\emptyset \subset K \subseteq L(s,Y)} (-1)^{|K|+1} (\Phi(s^{min(K)-1}) \times (2^{|s[K] \cap Y|} - 1))$$

avec $s[K] = \cap_{k \in K} s[k]$ pour toute séquence s et ensemble d'indices K.

L'exemple suivant permet de donner une intuition de la formule récursive introduite précédemment, et en particulier de son terme correcteur.

Exemple 3 *Poursuivons l'exemple 2. L'ensemble $\phi(s^1)$ des sous-séquences de $s^1 = \langle(ab)\rangle$ est défini par $\phi(s^1) = \{\langle\,\rangle, \langle a\rangle, \langle b\rangle, \langle(ab)\rangle\}$. Nous avons donc $\Phi(s^1) = 4$. Comme $L(s^1, s[2]) = 0$ et $R(s^1, s[2]) = 0$, nous avons $\Phi(s^2) = \Phi(s^1)\times|2^{(c)}| = 4\times 2 = 8$. En effet, les sous-séquences de s^2 sont obtenues par simple concaténation de l'itemset vide ou de l'itemset (c) avec une sous-séquence de s^1. Nous détaillons maintenant le calcul de $\Phi(s^3) = \Phi(s^2) \times |2^{(ac)}| - R(s^2, s[3]) = 8 \times 4 - R(s^2, s[3])$ et du terme correcteur $R(s^2, s[3])$. Comme $L(s^2, s[3]) = \{1, 2\}$, nous avons $R(s^2, s[3]) = (-1)^2\Phi(s^0) \times (2^{|(a)|} - 1) + (-1)^2\Phi(s^1) \times (2^{|(c)|} - 1) = 1+4 = 5$. Le premier terme de $R(s^2, s[3])$ permet de ne pas recompter la sous-séquence $\langle a\rangle$ de s^2 en la construisant par concaténation de l'itemset (a) (inclus dans $s[3]$) à la sous-séquence vide de s^0 . Quant au second terme de $R(s^2, s[3])$, il permet de ne pas recompter les sous-séquences $\langle c\rangle, \langle ac\rangle, \langle bc\rangle, \langle(ab)c\rangle$ de s^2 en les construisant par concaténation de l'itemset (c) (inclus dans $s[3]$) aux sous-séquences $\langle\rangle, \langle a\rangle, \langle b\rangle, \langle(ab)\rangle$ de s^1. Nous avons finalement $\Phi(s^3) = 32 - R(s^2, s[3]) = 27$.*

Avec contrainte sur la norme Nous proposons une généralisation du théorème 1 permettant de calculer le nombre de sous-séquences de norme inférieure ou égale à M d'une séquence s.

Théorème 2 (Nombre de sous-séquences de norme bornée) *Etant donnés une séquence s, un itemset Y et un entier $j \leq \|s\|$, le nombre de sous-séquences distinctes de norme inférieure ou égale à j de $s \circ Y$, noté $\Phi_{\leq j}(s \circ Y)$, est défini ci-dessous :*

$$\Phi_{\leq j}(s \circ Y) = \left(\sum_{k=0}^{min\{j,|Y|\}} C^k_{|Y|} \times \Phi_{\leq j-k}(s) \right) - R_{\leq j}(s, Y)$$

où $R_{\leq j}(s, Y)$ est un terme correcteur défini par :

$$R_{\leq j}(s, Y) = \sum_{\emptyset \subset K \subseteq L(s,Y)} (-1)^{|K|+1} \left(\sum_{k=1}^{|s[K]\cap Y|} C^k_{|s[K]\cap Y|} \times \Phi_{\leq j-k}(s^{min(K)-1}) \right)$$

sachant que $R_{\leq j}(s, Y) = 0$ si $L(s, Y) = \emptyset$.

Ce théorème est une généralisation du théorème 1. Notons par exemple que le premier terme $\Phi(s) \times 2^{|Y|}$ de $\Phi(s \circ Y)$ est remplacé par $\sum_{k=0}^{min\{j,|Y|\}} C^k_{|Y|} \times \Phi_{\leq j-k}(s)$ pour calculer $\Phi_{\leq j}(s \circ Y)$. Intuitivement, pour construire une sous-séquence de norme inférieure à j de $s \circ Y$, on peut concaténer tout sous-ensemble de taille k de Y à une sous-séquence de norme inférieure à $j - k$ de s. Ainsi, on est certain d'obtenir une sous-séquence de $s \circ Y$ de norme inférieure à $k + (j - k) = j$, et il faut répéter ce principe pour toute taille possible d'un sous-ensemble de Y. La même intuition explique la généralisation du terme correcteur $R(s, Y)$. En poursuivant l'exemple 3, l'exemple suivant illustre le principe de fonctionnement de la formule du théorème 2.

Exemple 4 *L'ensemble $\phi_{\leq 2}(s^1)$ des sous-séquences de $s^1 = \langle(ab)\rangle$ de norme inférieure à 2 est défini par $\phi_{\leq 2}(s^1) = \{\langle\,\rangle, \langle a\rangle, \langle b\rangle, \langle(ab)\rangle\}$. Nous avons donc $\Phi_{\leq 2}(s^1) = 4$, et on voit aussi aisément que $\Phi_{\leq 1}(s^1) = 3$ (la sous-séquence $\langle(ab)\rangle$ étant de norme strictement supérieure à 1). Comme $L(s^1, s[2]) = 0$, nous avons $R_{\leq 2}(s^1, s[2]) = 0$ et $\Phi_{\leq 2}(s^2) = \sum_{k=0}^{|(c)|} C^k_{|(c)|} \times \Phi_{\leq 2-k}(s^1) = C^0_1 \times \Phi_{\leq 2}(s^1) + C^1_1 \times \Phi_{\leq 1}(s^1) = 4 + 3 = 7$. Le premier terme de la somme correspond aux 4 sous-séquences de s^3 obtenues par concaténation du sous-ensemble vide aux sous-séquences de s^2, alors que le deuxième terme correspond aux 3 sous-séquences de s^3 obtenues par concaténation de l'itemset (c) aux sous-séquences de s^2 de norme inférieure à 1. Détaillons maintenant le calcul de $\Phi_{\leq 2}(s^3) = \sum_{k=0}^{|(ac)|} C^k_{|(ac)|} \times \Phi_{\leq 2-k}(s^2) - R_{\leq 2}(s^2, s[3])$ $= \Phi_{\leq 2}(s^2) + 2 \times \Phi_{\leq 1}(s^2) + \Phi_{\leq 0}(s^2) - R_{\leq 2}(s^2, s[3]) = 7 + 2 \times 4 + 1 - R_{\leq 2}(s^2, s[3])$. Le second terme de $\Phi_{\leq 2}(s^3)$, égal à 2×4, correspond par exemple au nombre de sous-séquences de s^3 pouvant être obtenues par concaténation d'un sous ensemble de taille 1 de (ab) (au nombre de 2) avec une sous-séquence de s^2 de norme inférieure à 1. Pour finir, le calcul du terme correcteur $R_{\leq 2}(s^2, s[3])$ se présente comme suit : $R_{\leq 2}(s^2, s[3]) = (-1)^2 C^1_{|(a)|} \times \Phi_{\leq 1}(s^0) +$ $(-1)^2 C^1_{|(c)|} \times \Phi_{\leq 1}(s^1) = 1 + 3 = 4$. On en déduit ainsi que $\Phi_{\leq 2}(s^3) = 7 + 2 \times 4 + 1 - 4 = 12$.*

La formule présentée au théorème 2 est récursive. Néanmoins, étant données une séquence s et une borne $M \leq \|s\|$, cette récursivité peut facilement être supprimée en calculant ligne par ligne les matrices T et R définies par :

- $T[i][j] = \Phi_{\leq j}(s^i)$ pour $i \in [0..|s|]$ et $j \in [0..M]$. $T[i][j]$ représente le nombre de sous-séquences de norme inférieure ou égal à j de la séquence s^i.
- $R[i][j] = R_{\leq j}(s^{i-1}, s[i])$ pour $i \in [2..|s|]$ et $j \in [0..M]$. Ce terme correcteur représente le terme à soustraire quand on souhaite calculer le nombre de sous-séquences de norme inférieure à j de $s^i = s^{i-1} \circ s[i]$ à partir du nombre de sous-séquences de norme inférieure à j de s^i en y concaténant des sous-ensembles de $s[i]$.

Des exemples de matrices T et R sont données en figure 1 pour la séquence $s = \langle(ab)c(ac)\rangle$, l'exemple 4 illustrant comment calculer $R[3][2] = R_{\leq 2}(s^2, s[3])$ et $T[3][2] = \Phi_{\leq 2}(s^3)$.

T[i][j]	$\leq \mathbf{0}$	$\leq \mathbf{1}$	$\leq \mathbf{2}$	$\leq \mathbf{3}$
$s^{\mathbf{0}} = \langle\rangle$	1	1	1	1
$s^{\mathbf{1}} = \langle(ab)\rangle$	1	3	4	4
$s^{\mathbf{2}} = \langle(ab)c\rangle$	1	4	7	8
$s^{\mathbf{3}} = \langle(ab)c(ac)\rangle$	1	4	12	21

R[i][j]	$\leq \mathbf{0}$	$\leq \mathbf{1}$	$\leq \mathbf{2}$
$s^1, s[\mathbf{2}] = c$	0	0	0
$s^2, s[\mathbf{3}] = (ac)$	2	4	5

FIG. 1: Exemples de matrices T et R

Pour conclure cette section, notons qu'à partir du théorème 2, étant donnés une séquence s et deux entiers m et M tels que $1 \leq m \leq M \leq \|s\|$, il est possible de calculer le nombre de sous-séquences distinctes de s de norme comprise entre m et M. En effet, nous avons $\Phi_{[m,M]}(s) = \Phi_{\leq M}(s) - \Phi_{\leq m-1}(s)$. Dans l'algorithme 1, cette formule permet de calculer à l'étape 1 le poids initial $w(s)$ des séquences s de la base de données séquentielles $\mathcal{S}$.

4.3 Tirage par rejet d'une sous-séquence

Après avoir tiré aléatoirement une séquence $s \in \mathcal{S}$ proportionnellement à son poids $w(s)$ (ligne 2 de l'algorithme 1) et un entier k entre m et M selon la distribution $\mathbb{P}_{[m,M]}(k)$ (ligne

3), l'objectif est maintenant de montrer comment retourner une sous-séquence de norme k tirée uniformément depuis la séquence s (ligne 4). La difficulté est de ne pas favoriser les séquences qui disposent de plusieurs occurrences au sein de la séquence.

Afin de contourner cette difficulté, nous proposons d'utiliser une méthode par rejet, en tirant uniformément une occurrence de la séquence s et en la rejetant si cette occurrence n'est pas la première. Comme chaque séquence dispose d'une unique première occurrence, cette approche garantit un tirage uniforme des motifs séquentiels. Pour commencer, nous formalisons la notion de première occurrence :

Définition 5 (Première occurrence) *Soient o_1 et o_2 deux occurrences d'une sous-séquence s' de s, de signatures respectives $\langle i_1^1, i_2^1, \ldots, i_m^1 \rangle$ et $\langle i_1^2, i_2^2, \ldots, i_m^2 \rangle$. On dit que o_1 précède o_2, noté $o_1 < o_2$, s'il existe un indice $l \in [1..m]$ tel que pour tout $j \in [1..l-1]$, on ait $i_j^1 = i_j^2$, et $i_l^1 < i_l^2$. Enfin, on appelle* première occurrence *de s' dans s sa plus petite occurrence (selon l'ordre défini précédemment).*

Exemple 5 *Dans la continuité de l'exemple 1, comme $\langle 1, 2 \rangle$ et $\langle 1, 3 \rangle$ sont les signatures respectives des deux occurrences $o_1 = \langle (a)(c)\emptyset \rangle$ et $o_2 = \langle (a)\emptyset(c) \rangle$ de la sous-séquence $s' = \langle (a)(c) \rangle$ de $s = \langle (ab)(cd)(ce) \rangle$, et que $\langle 1, 2 \rangle$ précède $\langle 1, 3 \rangle$, nous avons $o_1 < o_2$. Enfin, il est aisé de vérifier que o_1 est la première occurrence de s' dans s, o_1 et o_2 étant les deux seules occurrences de s' dans s.*

En pratique, nous devons surtout vérifier si une occurrence de la sous-séquence $s' \sqsubseteq s$ est la première occurrence de s' au sein de la séquence s :

Propriété 1 *Etant donnée une occurrence o d'une sous-séquence $s' \sqsubseteq s$ de signature $\sigma = \langle i_1, i_2, \ldots, i_m \rangle$, o est la première occurrence de s' si et seulement si pour $i_j \in \sigma$, il n'existe pas $l \in [i_{j-1} + 1..i_j - 1]$ tel que $o[i_j] \subseteq s[l]$ (avec $i_0 = 0$).*

Exemple 6 *Toujours dans la continuité de l'exemple 1, supposons qu'après avoir tiré $k = 2$ items de la séquence $s = \langle (ab)(cd)(ce) \rangle$, à savoir les items aux positions d'index 1 et 5, nous ayons généré l'occurrence $o = \langle (a)\emptyset(c) \rangle$ de signature $\langle 1, 3 \rangle$ de la sous-séquence $s' = \langle (a)(c) \rangle$ de s. Dans ce cas, comme il existe $l = 2$ appartenant à $[1 + 1..3 - 1]$ tel que $o[3] = (c) \subseteq s[2] = (cd)$, o n'est pas une première occurrence de s' et cette occurrence sera rejetée.*

Grâce à la propriété 1, il est finalement aisé de tirer uniformément une sous-séquence de norme k d'une séquence s. En tirant aléatoirement k positions distinctes entre 1 et $\|s\|$ de s, on commence par tirer uniformément une occurrence de norme k d'une sous-séquence de s. Si cette occurrence est une première occurrence, on l'accepte et on la retourne. Sinon on la rejette et on effectue un tirage aléatoire d'une nouvelle occurrence de s. Même si cet algorithme repose sur une technique d'échantillonnage avec rejet, nous montrons dans la section suivante que le nombre moyen de tirages avant acceptation est calculable.

4.4 Analyse de la méthode

La propriété suivante indique que l'algorithme 1 retourne un échantillon exact des motifs séquentiels avec une contrainte sur la norme :

Propriété 2 (Correction) *Soient une base de données séquentielles $\mathcal{S}$, des normes minimale m et maximale M, l'algorithme 1 effectue le tirage d'une sous-séquence de $\mathcal{S}$ de norme comprise entre m et M et proportionnellement à sa fréquence.*

Concernant la complexité, nous pouvons distinguer deux grandes phases dans notre approche : le pré-traitement (où la distribution des motifs séquentiels en fonction de la norme est calculée pour chaque séquence) et le tirage de sous-séquences.

Complexité du pré-traitement Le pré-traitement s'avère coûteux avec une complexité temporelle en $O(|\mathcal{S}| \cdot L \cdot M^2 \cdot 2^P \cdot T^2)$ où L est la longueur maximale d'une séquence, M est la norme maximale des sous-séquences tirées, P est la taille maximale d'un ensemble de positions $L(s^{i-1}, s[i])$ et T est la taille maximale d'un itemset d'une séquence. Néanmoins, $P \leq L$ peut être petit en pratique et ce pré-traitement (ligne 1 de l'algorithme 1) est réalisé une unique fois avant de pouvoir tirer N sous-séquences de $\mathcal{S}$.

Complexité du tirage Le tirage effectif des sous-séquences est moins coûteux. Tout d'abord, le tirage d'une séquence (ligne 2 de l'algorithme 1) se réalise en $O(\ln |\mathcal{S}|)$. Il est plus difficile d'estimer la complexité au pire du tirage d'une sous-séquence car le nombre de rejets n'est pas borné. Néanmoins, une bonne façon de mesurer l'efficacité de l'approche est de calculer le nombre moyen de tirages nécessaires, noté $\mu_{[m,M]}(\mathcal{S})$, pour tirer une sous-séquence de $\mathcal{S}$ de norme comprise entre m et M. Intuitivement, $\mu_{[m,M]}(\mathcal{S})$ dépend à la fois de la probabilité qu'une séquence $s \in \mathcal{S}$ soit tirée et du nombre moyen de tirages nécessaires, noté $\mu_{[m,M]}(s)$, pour tirer une occurrence d'une sous-séquence de s qui soit une première occurrence. La propriété suivante montre comment ces termes peuvent être calculés :

Propriété 3 (Nombre moyen de tirages) *Soient une base de données séquentielles $\mathcal{S}$, des normes minimale m et maximale M, le nombre moyen de tirages pour tirer un motif séquentiel de norme compris entre m et M est défini par :* $\mu_{[m,M]}(\mathcal{S}) = \sum_{s \in \mathcal{S}} \frac{\Phi_{[m,M]}(s)}{\sum_{s' \in \mathcal{S}} \Phi_{[m,M]}(s')} \times \mu_{[m,M]}(s)$ *avec* $\mu_{[m,M]}(s) = \frac{\sum_{k=m}^{M} C^k_{\|s\|}}{\Phi_{[m,M]}(s)}$.

Lorsque le nombre moyen de tirages est proche de 1, cela signifie que le tirage d'un motif séquentiel ne donnera pas lieu à un rejet. Pour une séquence donnée, il n'y a pas de rejet si chaque occurrence est la première occurrence i.e., il n'y a pas de répétition au sein de la séquence. Dans la pratique, le nombre moyen de tirages mesuré sur des jeux de données réels est souvent très faible (voir la section expérimentale suivante). Finalement, la complexité temporelle du tirage d'une occurrence de norme égale à $k \in [m..M]$ d'une séquence s étant dans le pire des cas en $O(M^2)$, la complexité en moyenne du tirage de N sous-séquences d'une base de données $\mathcal{S}$ (après la phase de pré-traitement) est en $O(N \cdot M^2 \cdot \mu_{[m,M]}(\mathcal{S}))$.

5 Expérimentations

L'objectif de cette section expérimentale est d'évaluer la rapidité de notre méthode et d'observer l'impact de la contrainte sur les motifs extraits. Pour cela, nous avons utilisé six jeux de données. `bms` et `sign` sont des jeux de données réels disponibles avec SPMF [1]. Les quatre autres ont été construits avec le générateur de données de IBM également disponible sur le site

1. `www.philippe-fournier-viger.com/spmf`

	Caractéristiques générales				Nombre moyen de tirages pour $M =$				
Jeu de données	$\|\mathcal{S}\|$	$\|\mathcal{I}\|$	$\|S\|_{moy}$	$\|\|S\|\|_{moy}$	3	4	5	6	7
bms	59,601	497	2.5	5.0	1.0	1.0	1.0	1.0	1.0
sign	730	267	52.0	104.0	1.0	1.0	1.0	1.0	1.0
D10K5S2T6I	10,000	6	5.6	15.9	11.4	16.9	23.5	30.8	38.4
D10K6S3T10I	10,000	10	6.0	21.9	10.4	14.4	18.5	22.4	25.7
D100K5S2T6I	100,000	6	4.8	13.3	8.5	11.5	14.9	19.0	23.9
D100K6S2T6I	100,000	6	5.6	16.0	11.1	16.0	21.4	27.0	32.4

TAB. 1: Caractéristiques des benchmarks

de SPMF. Un des intérêts des jeux de données synthétiques est d'avoir des exemples de jeux de données avec un nombre moyen de tirages nécessaires $\mu_{[m,M]}(\mathcal{S})$ supérieur à 1 (grâce à l'ajout de répétitions au sein des séquences). Le tableau 1 présente les caractéristiques générales des jeux de données (partie gauche) et le nombre moyen de tirages nécessaires $\mu_{[m,M]}(\mathcal{S})$ pour extraire un motif avec $m = 1$ et $M \in [3..7]$ (partie droite). Notre méthode est implémentée avec le langage Python. Toutes les expériences sont faites sur un PC avec un processeur AMD 2.5 GHz Quad Core A8-7410, une RAM de 8GB, avec Ubuntu 16.04 LTS 64 bits.

	Prétraitement (s)					Tirage d'un motif (ms)				
	M					M				
Jeu de données	3	4	5	6	7	3	4	5	6	7
bms	11	15	20	21	24	1.1	1.4	1.5	1.7	1.8
sign	10	15	20	24	27	0.6	0.6	0.6	0.7	0.7
D10K5S2T6I	10	15	19	25	30	0.8	1.3	2.5	2.7	4.7
D10K6S3T10I	18	28	38	48	59	0.9	1.2	2.0	2.9	2.5
D100K5S2T6I	71	109	141	165	193	0.6	0.9	1.4	1.7	2.6
D100K6S2T6I	105	155	198	247	271	0.8	1.3	2.2	2.7	3.9

TAB. 2: Temps d'exécution de l'échantillonnage de sous-séquences de norme inférieure à M

Rapidité de l'approche Le tableau 2 indique le temps d'exécution de la méthode en distinguant le temps de prétraitement et le temps moyen pour tirer un motif séquentiel dont la norme est comprise entre 1 et $M \in [3..7]$. On constate que le temps de préparation augmente avec la taille du jeu de données (du nombre de séquences et d'items) et avec la norme maximale. Même pour D100K6S2T6I qui est de grande taille, le temps d'exécution de ce prétraitement (qui peut se faire hors-ligne) est tout à fait raisonnable (moins de 5 min). Concernant la phase de tirage, quel que soit le jeu de données et M, le temps d'exécution est de l'ordre de quelques millisecondes (au plus 4 ms pour D100K6S2T6I avec $M = 7$). Malgré un nombre moyen de tirages nécessaires $\mu_{[m,M]}(\mathcal{S})$ supérieur à 1 (et donc du rejet lors du tirage), les performances sur les jeux de données synthétiques sont bonnes. On observe une hausse du temps d'exécution avec M mais celle-ci reste limitée avec des durées moyennes de tirage inférieures à 4 ms.

Impact de la contrainte La figure 2 montre la répartition de 10 000 motifs séquentiels échantillonnés selon la fréquence avec une contrainte de norme inférieure à 4 ou 7 (en gris) et sans contrainte (en noir) pour les différents jeux de données. Dans tous les cas, la méthode sans contrainte retourne uniquement des motifs de fréquence très faible (en particulier de fréquence unitaire sur les jeux de données réels). A l'inverse, la méthode d'échantillonnage avec contrainte sur la norme retourne des sous-séquences de fréquence significativement plus élevée

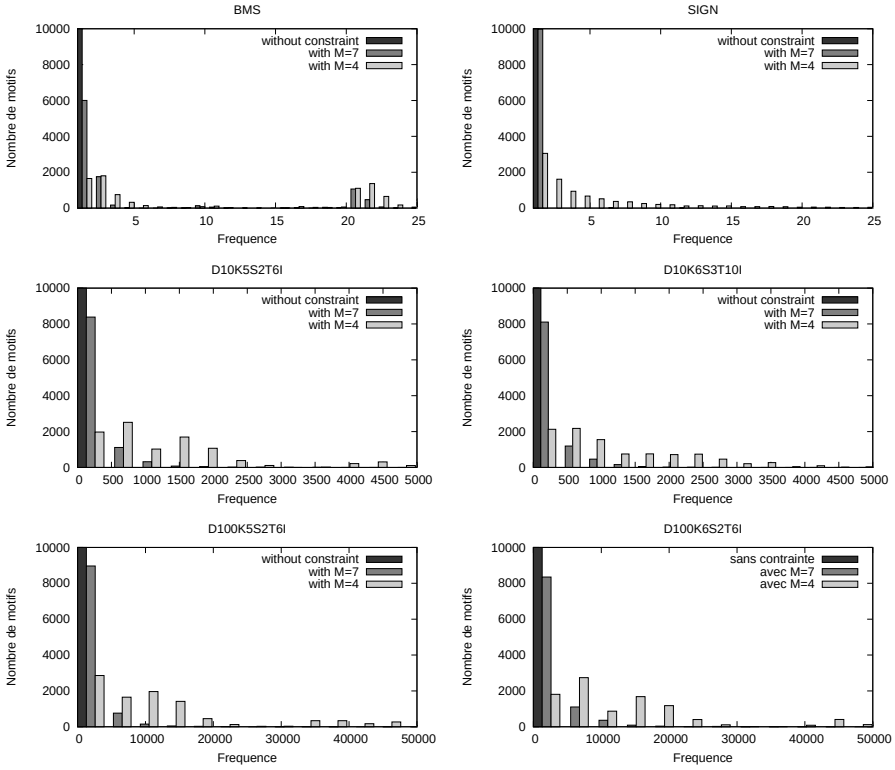

FIG. 2: Répartition de 10 000 motifs séquentiels selon leur fréquence

(de 100 à 1000 fois plus élevées), ce qui démontre l'importance d'introduire des contraintes sur la norme. Notons que pour `sign`, l'effet est peu significatif avec $M = 7$ mais la diminution de la norme maximale parvient à juguler l'explosion des motifs de fréquence peu élevée.

6 Conclusion

Cet article propose la première méthode pour échantillonner en sortie des motifs séquentiels. Elle permet en outre de spécifier un intervalle sur la norme des motifs séquentiels afin de mieux contrôler les motifs retournés. Nous avons démontré que notre approche est exacte et nous avons estimé son efficacité en fonction du nombre de rejets moyen qui se dégrade avec le nombre de répétitions au sein d'une séquence. Néanmoins, la partie expérimentale a montré que l'approche s'avère très performante sur des jeux de données réels où le taux de répétition est très faible. De plus, les expérimentations montrent que l'ajout d'une contrainte sur la norme évite de retourner trop de motifs trop rares. Dans l'immédiat, nous voudrions appliquer l'échantillonnage de motifs séquentiels à la détection de données aberrantes (Giacometti et Soulet, 2016) ou au sein de systèmes interactifs (Giacometti et Soulet, 2017) pour démon-

trer son utilité. Nous souhaiterions aussi étendre notre approche à tout système ensembliste. En effet, le tirage uniforme au sein de structures complexes rendu possible grâce à une forme canonique est envisageable avec d'autres langages structurés.

Remerciements. Lamine Diop est partiellement financé par le CEA-MITIC, Centre d'Excellence Africain en Mathématiques, Informatique et TIC.

Références

Agrawal, R. et R. Srikant (1995). Mining sequential patterns. In *Proc. of ICDE 95*, pp. 3–14.

Al Hasan, M. et M. J. Zaki (2009). Output space sampling for graph patterns. *Proc. of the VLDB Endowment 2*(1), 730–741.

Boley, M., C. Lucchese, D. Paurat, et T. Gärtner (2011). Direct local pattern sampling by efficient two-step random procedures. In *Proc. of the 17th ACM SIGKDD*, pp. 582–590.

Dzyuba, V., M. van Leeuwen, et L. De Raedt (2017). Flexible constrained sampling with guarantees for pattern mining. *Data Mining and Knowledge Discovery 31*(5), 1266–1293.

Egho, E., C. Raïssi, T. Calders, N. Jay, et A. Napoli (2015). On measuring similarity for sequences of itemsets. *Data Mining and Knowledge Discovery 29*(3), 732–764.

Giacometti, A. et A. Soulet (2016). Frequent pattern outlier detection without exhaustive mining. In *Proc. of PAKDD 2016*, pp. 196–207. Springer.

Giacometti, A. et A. Soulet (2017). Interactive pattern sampling for characterizing unlabeled data. In *Proc. of IDA 2017*, pp. 99–111. Springer.

Gomariz, A., M. Campos, et R. M. B. Goethals (2013). ClaSP : An efficient algorithm for mining frequent closed sequences. In *Proc. of PAKDD 2013*, pp. 50–61.

Lo, D., S.-C. Khoo, et J. Li (2008). Mining and ranking generators of sequential patterns. In *Proc. of SDM 2008*, pp. 553–564.

Pei, J., J. Han, B. Mortazavi-Asl, et H. Pinto (2001). PrefixSpan : Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proc. of ICDE 2001*, pp. 215–224.

Raissi, C. et P. Poncelet (2007). Sampling for sequential pattern mining : From static databases to data streams. In *Proc. of ICDM 2007*, pp. 631–636. IEEE.

Toivonen, H. et al. (1996). Sampling large databases for association rules. In *Proc. of VLDB 96*, Volume 96, pp. 134–145.

Zaki, M. J. (2001). SPADE : An efficient algorithm for mining frequent sequences. *Machine Learning 42*(1-2), 31–60.

Summary

Pattern sampling is a method for discovering patterns with strong statistical guarantees. In this paper, we propose the first method for sampling sequential patterns. Beyond addressing the sequential data, the originality of our approach is to constrain the norm of sequential patterns to avoid the long tail issue. We demonstrate that our constrained two-step random procedure performs an exact sampling which is efficient in practice.