Échantillonnage de motifs d'intervalles selon la fréquence

Djawad Bekkoucha*, Abdelkader Ouali*, Bruno Crémilleux*, Patrice Boizumault*

Université Caen Normandie, ENSICAEN, CNRS, Normandie Univ
GREYC UMR6072, F-14000 Caen, France
*prénom.nom@unicaen.fr

Résumé. L'échantillonnage de motifs est une approche récente pour mener à bien un processus de découverte de motifs en limitant le nombre de motifs produits et en favorisant l'interactivité avec l'analyste. Le principe est de tirer aléatoirement un motif proportionnellement à un intérêt donné, par exemple la mesure de fréquence. Dans cet article, nous proposons FIPS, la première méthode d'échantillonnage de motifs d'intervalles à partir de données numériques. FIPS s'appuie sur le principe des approches en plusieurs étapes (Boley et al.). Avec des données numériques, la difficulté majeure est de déterminer le nombre exact de motifs d'intervalles couvrant un objet. Nous montrons formellement que FIPS tire des motifs d'intervalles proportionnellement à la fréquence. Les expérimentations, répondant à plusieurs questions de recherche, montrent la qualité des motifs tirés par FIPS pour la découverte d'information.

1 Introduction

Les spécialistes en science des données sont des acteurs majeurs de la transformation de données pour la mise en évidence d'informations, ou mieux, de connaissances. En pratique, les analystes veulent interagir (visualiser, sélectionner, explorer) non seulement avec les données mais aussi avec les motifs ou modèles supportés par les données. Pour mener à bien de tels processus, il est nécessaire de disposer de méthodes d'extraction de motifs quasi instantanées afin que l'analyste n'attende pas, sinon il décroche du système. L'échantillonnage de motifs est une réponse à ce défi (Dzyuba et al., 2017).

L'échantillonnage de motifs consiste à tirer aléatoirement un motif proportionnellement à une mesure d'intérêt comme par exemple la fréquence. Ainsi un motif $\mathcal{V}_1$ deux fois plus fréquent qu'un motif $\mathcal{V}_2$ aura deux fois plus de chance d'être tiré. Une approche naïve pour effectuer l'échantillonnage de motifs est de générer l'ensemble des motifs puis d'effectuer un tirage pondéré selon la mesure d'intérêt. Une telle méthode échoue en pratique compte-tenu de la taille gigantesque des espaces de recherche même pour des langages de motifs aussi simples que les motifs ensemblistes. Une réponse à ce problème a été apportée par Boley et al. qui ont défini une approche fondée sur deux tirages successifs. Cette stratégie, à condition d'effectuer une décomposition appropriée au problème étudié, assure d'effectuer un tirage exact selon la distribution résultante de la mesure d'intérêt. Les exemples présentés dans Boley et al. concernent des motifs ensemblistes, c'est-à-dire des données décrites par des valeurs booléennes.

Cet article porte sur les données numériques. Celles-ci sont présentes dans de nombreuses applications comme l'analyse de données du transcriptome, les données démographiques ou encore en agronomie pour l'étude du stress thermique auprès de bovins. À notre connaissance, il existe une seule méthode d'échantillonnage portant sur des données numériques (Giacometti et Soulet, 2018). Celle-ci utilise une métrique pour construire des motifs de voisinage dont l'intérêt est caractérisé par une mesure de densité. Les données sont considérées comme continues et la méthode nécessite de fixer une valeur qui définit la taille d'un motif. La recherche de motifs dans les données numériques remonte aux *quantitative rules* (Srikant et Agrawal, 1996). Elle pose le problème de leur discrétisation qui est susceptible d'entraîner une perte d'information. Kaytoue et al. (2011) ont défini les *motifs d'intervalles* qui ont l'avantage de préserver toute l'information originelle. Ce cadre attrayant se heurte cependant au problème du passage à l'échelle du fait du grand nombre de motifs produits, ce qui renforce l'intérêt dans ce contexte de l'échantillonnage de motifs.

Dans cet article, nous présentons FIPS, la première méthode d'échantillonnage de motifs d'intervalles. FIPS, effectue un échantillonnage proportionnellement à la fréquence selon une approche en deux étapes (Boley et al.). La difficulté majeure est de calculer le nombre exact de motifs d'intervalles couvrant un objet. Nous montrons formellement que FIPS tire des motifs d'intervalles proportionnellement à la fréquence. Nous montrons expérimentalement la qualité des motifs obtenus selon plusieurs critères (fréquence, impact du phénomène de la longue traîne, diversité, rapidité et plausibilité).

La section 2 introduit les notations, les définitions et précise le problème étudié. La section 3 situe notre travail dans le domaine de l'échantillonnage de motifs. Nous présentons FIPS à la section 4 et nous évaluons la qualité des motifs obtenus à la section 5.

2 Préliminaires

2.1 Définitions

Base de données numériques. Une *base de données numériques* $\mathcal{N}$ est définie par un ensemble d'objets $\mathcal{G}$ où chaque objet est décrit par un ensemble d'attributs $\mathcal{M}$. Chaque attribut $m \in \mathcal{M}$ dispose d'un ensemble fini $\mathcal{N}_m$ de valeurs possibles dans dans $\mathcal{N}$. Un objet $g \in \mathcal{G}$ est défini par un vecteur de valeurs numériques $\langle v_{g,m} \rangle_{\forall m \in \mathcal{M}}$. Une base de données dans laquelle les valeurs de tous les attributs sont binaires $\mathcal{N}_m = \{0,1\}, \forall m \in \mathcal{M}$, est un cas particulier d'une base de données numériques référée ici par une *base de données binaires*.

Exemple 1. *Le tableau 1 illustre un jeu de données numériques contenant 5 objets* $\mathcal{G} = \{g_1, g_2, g_3, g_4, g_5\}$ *où chaque objet est décrit par 3 attributs* $\mathcal{M} = \{m_1, m_2, m_3\}$.

	m_1	m_2	m_3
g_1	2	8	130
g_2	4	12	102
g_3	3	7	91
g_4	2	9	101
g_5	6	12	110

TAB. 1 – *Exemple d'un jeu de données numériques* $\mathcal{N}$

Motifs d'intervalles. Afin de préserver toute l'information originelle d'une base de données numériques, nous utilisons la notion de motif d'intervalles (Kaytoue et al., 2011) . Un motif d'intervalles est défini par un vecteur d'intervalles $\mathcal{V} = \langle [\underline{w_m}, \overline{w_m}] \rangle_{\forall m \in \mathcal{M}}$ où $\underline{w_m}, \overline{w_m} \in \mathcal{N}_m$. Chaque dimension du vecteur $\mathcal{V}$ correspond à un attribut suivant un ordre canonique sur l'ensemble des attributs $\mathcal{M}$. Nous notons $\mathcal{B}[g] = \langle [v_{g,m}, v_{g,m}] \rangle_{m \in \{1,...,|\mathcal{M}|\}}$ le vecteur des intervalles correspondant à un objet identifié par g. Un objet g est une occurrence du motif d'intervalles $\mathcal{V}$ si chaque intervalle du vecteur $\mathcal{B}[g]$ est inclus dans l'intervalle de $\mathcal{V}$, i.e. $\mathcal{B}[g] \sqsubseteq \mathcal{V} \iff [v_{g,m}, v_{g,m}] \subseteq [\underline{w}_m, \overline{w}_m], \forall m \in \{1,...,|\mathcal{M}|\}$. La couverture de $\mathcal{V}$ dans $\mathcal{N}$ est l'ensemble d'objets $g \in \mathcal{G}$ inclus dans $\mathcal{V}$. La fréquence de $\mathcal{V}$ est le cardinal de sa couverture, c.-à-d. $freq(\mathcal{V}, \mathcal{N}) = |cover(\mathcal{V}, \mathcal{N})|$. Étant donné un seuil de fréquence minimum θ, le motif d'intervalles $\mathcal{V}$ est fréquent si et seulement si $freq(\mathcal{V}, \mathcal{N}) \geq \theta$.

Exemple 2. *Le motif* $\mathcal{V} = \langle [3,4], [7,12], [91,130] \rangle$ *couvre les objets* $\{g_2, g_3\}$ *de la base de données du tableau 1, sa fréquence est ainsi égale à 2.* $\mathcal{B}[g_2] = \langle [4,4],\ [12,12],\ [102,102] \rangle$ *est le vecteur des intervalles identifiés par l'objet* g_2 *et constitue une occurrence de* $\mathcal{V}$.

2.2 Formulation du problème

Soit Ω une population et $f : \Omega \longrightarrow [0,1]$ une mesure. La notation $x \sim f(\Omega)$ signifie que l'élément x est tiré au hasard dans Ω avec une distribution de probabilité $\pi(x) = f(x) / \sum_{x' \in \Omega} f(x')$. Nous nous intéressons ici au langage des motifs d'intervalles $\mathcal{L}$ et à la mesure de fréquence. Étant donné une base de données numériques $\mathcal{N}$ et $k \in \mathbb{N}$, le problème d'échantillonnage de motifs d'intervalles revient à fournir k motifs $\mathcal{V}_1, ..., \mathcal{V}_k$ de $\mathcal{L}$ où chaque motif $\mathcal{V}_i$ est tiré aléatoirement et avec remise, proportionnellement à sa valeur de fréquence $freq(\mathcal{V}_i, \mathcal{N}), i \in \{1,...,k\}$.

3 État de l'art

On distingue trois grandes familles de méthodes d'échantillonnage de motifs : les méthodes stochastiques, les méthodes s'appuyant sur des paradigmes déclaratifs et les méthodes fondées sur plusieurs tirages.

La première famille repose sur les algorithmes de Monte-Carlo par chaînes de Markov, pour lesquels la distribution cible d'échantillonnage correspond à la distribution stationnaire de la marche aléatoire. Hasan et Zaki se sont intéressés aux graphes et ont conçu une approche d'échantillonnage sur les sous-graphes proportionnellement à la fréquence. Il s'agit de la première méthode d'échantillonnage de motifs. Les méthodes de cette famille sont aptes à traiter différents types de motifs et de mesures. Par exemple, Boley et al. (2010) échantillonnent des concepts formels selon des mesures d'intérêt strictement positives et Bendimerad et al. (2020) des tuiles selon un intérêt subjectif. Les approches stochastiques permettent un échantillonage exact, mais souffrent généralement d'une lenteur de convergence.

La deuxième famille s'appuie sur des paradigmes déclaratifs. FLEXICS (Dzyuba et al., 2017) utilise un solveur SAT pour échantillonner des motifs ensemblistes pour une classe de mesures assez générale. Le principe est de diviser aléatoirement l'espace de recherche en différentes cellules en générant de manière récursive des contraintes XOR sur les variables associées aux items décrivant les motifs. Une des limites de ces méthodes réside dans la difficulté

à trouver des modèles pour formuler d'autres langages de motifs, car cela requiert d'adapter l'encodage existant à ces nouveaux langages pour garantir des implémentations efficaces.

La troisième famille regroupe les méthodes en plusieurs étapes où un tirage est effectué à chaque étape. Le principe est de répartir les occurrences de motifs en groupes judicieusement définis afin de pouvoir tirer un groupe proportionnellement à son poids et au final pouvoir tirer uniformément un motif au sein d'un groupe. La difficulté est de trouver une décomposition appropriée en plusieurs étapes afin d'obtenir la distribution souhaitée. Cette approche a été initiée par Boley et al., avec un tirage en deux étapes pour échantillonner des motifs ensemblistes selon la fréquence. Ce principe est retrouvé dans Diop et al. (2020), avec une méthode en 3 étapes, pour échantillonner des motifs séquentiels selon la fréquence et en y ajoutant une contrainte de longueur maximale. Camelin et al. (2024) associent une technique de compression à une approche en plusieurs étapes pour échantillonner des motifs ensemblistes selon la fréquence. La contrainte de fréquence minimale est poussée dans Soulet (2023) en combinaison avec l'échantillonnage de motifs ensemblistes.

À notre connaissance, il existe une unique méthode d'échantillonnage dans des données numériques (Giacometti et Soulet, 2018). Celle-ci suit une approche par étapes. À la différence de FIPS, cette méthode considère un espace continu sur les données, échantillonne suivant une mesure de densité et nécessite de définir la taille d'un motif. Notre travail s'inscrit dans cette classe d'approches en raison de sa rapidité et de sa capacité à suivre la distribution désirée.

4 Méthode d'échantillonnage de motifs d'intervalles

Cette section présente FIPS, une méthode d'échantillonnage de motifs d'intervalles proportionnelle à la fréquence.

4.1 Principes de FIPS

Pour tirer un motif proportionnellement à la fréquence, il est nécessaire de connaître la somme des fréquences de tous les motifs de l'espace des solutions, soit $\sum_{\mathcal{V} \in \mathcal{L}} freq(\mathcal{V})$. Compte tenu de la taille de l'espace des solutions, il n'est pas possible de connaître cette somme en parcourant un tel espace. Effectuer un échantillonnage en plusieurs étapes va permettre de contourner cette difficulté. Avec un échantillonnage à deux étapes (Boley et al.), une fois les occurrences d'un motif réparties de façon appropriée dans différents groupes, le principe est de tirer un groupe proportionnellement à son poids, puis de tirer une occurrence d'un motif au sein du groupe choisi de façon uniforme. L'idée sous-jacente est qu'il est possible d'obtenir la somme des fréquences des motifs composant l'espace des solutions, en connaissant, pour chaque objet, les motifs qui le couvrent. Dans le cas de données ensemblistes, cette tâche est simple car il suffit de tester l'inclusion d'un motif dans un objet. En revanche, pour les données numériques, cette opération est plus complexe car elle nécessite de considérer pour chaque attribut d'un objet tous les intervalles possibles des autres attributs et couvrant l'objet. Pour résoudre ce problème, nous définissons une fonction nommée *NIP* (*Number of Interval Patterns*, cf. section suivante) qui retourne le nombre exact de motifs d'intervalles couvrant un objet. Grâce à cette stratégie, il est possible de calculer la somme des fréquences de tous les motifs (qui est la constante de normalisation Z, cf. section 4.3) en considérant uniquement les

motifs couvrant un objet. La méthode calcule ainsi une distribution de probabilité proportionnelle à la fréquence sans parcourir l'espace des solutions.

4.2 Calcul du nombre de motifs d'intervalles couvrant un objet

Cette section définit la fonction $NIP : g \to \mathbb{N}$ qui calcule, pour un objet g, le nombre exact de motifs d'intervalles le couvrant. Pour une valeur $v_{g,m} \in \mathcal{N}_m$ apparaissant dans un objet $g \in \mathcal{G}$, nous définissons :

— $\mathcal{U}(v_{g,m}) = \{v \in \mathcal{N}_m | \quad v < v_{g,m}\}$, qui représente l'ensemble des valeurs distinctes de l'attribut m formant des bornes inférieures dans les intervalles contenant la valeur $v_{g,m}$.

— $\mathcal{A}(v_{g,m}) = \{v \in \mathcal{N}_m | \quad v > v_{g,m}\}$, qui représente l'ensemble des valeurs distinctes de l'attribut m formant des bornes supérieures dans les intervalles contenant la valeur $v_{g,m}$.

La fonction $NIP(g)$ est définie comme suit :

$$NIP(g) = \prod_{m \in \mathcal{M}} |\mathcal{U}(v_{g,m})| + |\mathcal{A}(v_{g,m})| + |\mathcal{U}(v_{g,m})| \cdot |\mathcal{A}(v_{g,m})| + 1 \tag{1}$$

Pour chaque attribut $m \in \mathcal{M}$, la formule 1 contient plusieurs termes, chacun déterminant le nombre d'intervalles possibles incluant la valeur $v_{g,m}$. Le premier terme, $|\mathcal{U}(v_{g,m})|$, détermine le nombre d'intervalles possibles ayant la valeur $v_{g,m}$ comme borne supérieure. Le deuxième, $|\mathcal{A}(v_{g,m})|$, détermine le nombre d'intervalles possibles ayant la valeur $v_{g,m}$ comme borne inférieure. Le troisième, $|\mathcal{U}(v_{g,m})| \cdot |\mathcal{A}(v_{g,m})|$, quantifie le nombre d'intervalles possibles contenant strictement la valeur $v_{g,m}$ à l'intérieur de l'intervalle. Enfin, le dernier terme représente l'intervalle unique où $v_{g,m}$ est à la fois borne inférieure et borne supérieure. En effectuant le produit des nombres d'intervalles possibles sur l'ensemble des attributs de la base de données, on obtient ainsi le nombre exact de motifs d'intervalles couvrant l'objet g.

Exemple 3. *Considérons la base de données numériques $\mathcal{N}$, l'objet $g_3 \in \mathcal{G}$ et l'attribut $m_1 \in \mathcal{M}$ (cf. table 1). Les intervalles incluant la valeur 3 qui apparaît dans l'attribut m_1 pour l'objet g_3 sont les intervalles où :*

— *la valeur 3 est une borne inférieure :* $[3,4],\ [3,6]$

— *la valeur 3 est une borne supérieure :* $[2,3]$

— *la valeur 3 est strictement incluse :* $[2,4],\ [2,6]$

— *la valeur 3 est à la fois borne inférieure et borne supérieure :* $[3,3]$

Ainsi, le nombre total d'intervalles incluant la valeur 3 pour l'attribut m_1 est 6. En reprenant les termes de la la formule 1, on a : $|\mathcal{U}(3)| + |\mathcal{A}(3)| + |\mathcal{U}(3)| \cdot |\mathcal{A}(3)| + 1 = |\{2\}| + |\{4,6\}| + |\{2\}| \times |\{4,6\}| + 1 = 6$. *Le nombre total de motifs d'intervalles possibles couvrant l'objet g_3 est obtenu en multipliant les nombre d'intervalles de chaque attribut (cf. formule 1) :* $6 \times 4 \times 5 = 120$.

4.3 Algorithme d'échantillonnage

Cette section présente l'algorithme d'échantillonnage FIPS (cf. algorithme 1). Cet algorithme commence par calculer le nombre de motifs d'intervalles couvrant chaque objet de la base de données $\mathcal{N}$ (ligne 3). Ce calcul est effectué via la fonction *NIP* (cf. section 4.2). La ligne 4 correspond à la première étape d'une méthode d'échantillonnage en deux étapes : un objet g est tiré avec une probabilité proportionnelle au nombre de motifs d'intervalles couvrant g. Cette étape permet de biaiser le tirage vers des motifs couvrant un grand nombre

d'objets. Puis, dans la deuxième étape (ligne 5), un motif d'intervalles couvrant au moins l'objet g est tiré uniformément. Plus précisément, pour chaque attribut $m \in \mathcal{M}$, deux valeurs $a_m \in \{e \mid e \in \mathcal{N}_m \wedge e \leq v_{g,m}\}$ et $b_m \in \{e \mid e \in \mathcal{N}_m \wedge e \geq v_{g,m}\}$ sont tirées selon une distribution uniforme dans chaque ensemble, a_m et b_m forment l'intervalle de l'attribut m pour le motif tiré. On obtient un échantillon de k motifs en lançant k fois l'algorithme 1. Comme le montre la proposition 1, FIPS échantillonne les motifs proportionnellement à la fréquence.

Algorithme 1 : Échantillonnage d'un motif d'intervalles selon la fréquence (FIPS)

1 **Entrée :** un jeu de données numériques $\mathcal{N}$;
2 **Sortie :** un motif d'intervalles $\mathcal{V}$ tiré proportionnellement à la fréquence;
3 **Pré-traitement :** Calculer $w(g) = NIP(g)$ pour chaque objet $g \in \mathcal{G}$;
4 **Étape 1 :** Tirer un objet $g \sim w(g)$;
5 **Étape 2 :** Tirer un motif d'intervalles $\mathcal{V}$ uniformément parmi ceux couvrant l'objet g;
6 **Retourner** $\mathcal{V}$;

Proposition 1. *Pour une base de données $\mathcal{N}$, l'algorithme 1 tire un motif d'intervalles noté $\mathcal{V}$ proportionnellement à la fréquence en $O(|\mathcal{G}| \cdot |\mathcal{M}| + (ln|\mathcal{N}| + |\mathcal{M}|))$.*

Preuve. Nous démontrons que l'algorithme 1 effectue un échantillonnage proportionnel à la fréquence. Soit $\mathcal{L}$ l'espace de tous les motifs d'intervalles, une constante de normalisation notée $Z = \sum_{\mathcal{V} \in \mathcal{L}} |cover(\mathcal{V},\mathcal{N})|$ (Z est la somme des fréquences de tous les motifs d'intervalles), un objet $g^* \in \mathcal{G}$ tiré aléatoirement à partir de l'étape 1 de l'algorithme 1 et $\mathcal{V}^*$ un motif d'intervalles tiré par l'étape 2 de cet algorithme.

$$
\begin{aligned}
P[\mathcal{V}^* = \mathcal{V}] &= \sum_{g \in \mathcal{G}} P[\mathcal{V}^* = \mathcal{V} \wedge g^* = g] \\
&= \sum_{g \in cover(\mathcal{V},\mathcal{N})} \frac{1}{NIP(g)} \frac{NIP(g)}{Z} \\
&= \sum_{g \in cover(\mathcal{V},\mathcal{N})} \frac{1}{Z} = \frac{|cover(\mathcal{V},\mathcal{N})|}{Z} = \frac{freq(\mathcal{V},\mathcal{N})}{Z}
\end{aligned}
$$

avec $Z = \sum_{g \in \mathcal{G}} NIP(g)$ (ce qui est égal à $Z = \sum_{\mathcal{V} \in \mathcal{L}} |cover(\mathcal{V},\mathcal{N})|$)

Sur la complexité : $|\mathcal{G}| \cdot |\mathcal{M}|$ est la complexité temporelle du pré-traitement pour calculer les poids $w(g)$ des objets en utilisant la fonction *NIP*. La complexité temporelle du tirage d'un motif est la somme, d'une part, de la complexité temporelle de la recherche dichotomique de l'étape 1 ($\ln|\mathcal{N}|$) et d'autre part, de la complexité temporelle du tirage uniforme des intervalles de l'étape 2 ($|\mathcal{M}|$). Ainsi, l'échantillonnage de k motifs d'intervalles a une complexité temporelle de $k(\ln|\mathcal{N}| + |\mathcal{M}|)$ auquel il faut ajouter le temps du pré-traitement ($|\mathcal{G}| \cdot |\mathcal{M}|$). □

5 Expérimentations

Cette section étudie expérimentalement les performances de FIPS en répondant aux questions suivantes.

1. Quelle est la qualité des motifs échantillonnés par FIPS en terme de fréquence ?
2. Quel est l'impact du phénomène de la longue traîne sur FIPS ?
3. Quelle est la diversité des motifs d'intervalles échantillonnés par FIPS ?
4. Quel est le temps nécessaire à FIPS pour échantillonner des motifs ?
5. Quelle est la pertinence des motifs échantillonnés par FIPS ?

Bases de données sélectionnées. Notre évaluation expérimentale est réalisée sur un ensemble de 13 bases de données numériques. Les bases Glass, Iris, balance-scale, diabetes, sonar et heart proviennent du UCI Machine Learning Repository [1], tandis que les 7 autres sont issues du protocole expérimental de Bekkoucha et al. (2024). Le nombre d'attributs numériques, d'objets et de valeurs distinctes de chaque base de données sont reportés au tableau 2. En raison de la limite du nombre de pages, les résultats des expériences portant sur la fréquence, la plausibilité et le temps CPU sont donnés sur un nombre restreint de jeux de données, les autres résultats sont accessibles via le lien `https://github.com/djawed-bkh/FIP-Sampling`.

	NT	AP	BK	Cancer	CH	Yacht	Iris	LW	Glass	balance-scale	diabetes	sonar	heart
$\#\mathcal{M}$	3	5	5	9	8	7	4	10	9	4	8	60	13
$\#\mathcal{G}$	130	135	96	116	209	308	150	189	214	625	768	208	270
#valeurs distinctes	67	674	313	900	396	322	123	253	939	20	1254	11 256	384

TAB. 2 – *Caractéristiques des bases de données numériques*

Approches comparées. Comme il n'existe pas dans la littérature de méthode d'échantillonnage de motifs d'intervalles à partir de données numériques, nous comparons FIPS à une approche tirant des motifs d'intervalles selon une distribution uniforme. Plus précisément, cette dernière revient à tirer uniformément les bornes des intervalles de chaque attribut de la base de données. Toutefois, on constate expérimentalement (cf. tableau 3) qu'une telle démarche tire un grand nombre de motifs avec une couverture vide. Ces motifs, ne couvrant aucun objet de la base, ne sont pas pertinents pour un analyste. Afin de garantir que tout motif tiré couvre au moins un objet, lors du tirage d'une intervalle d'un motif, au moins une des bornes tirées doit être une valeur présente dans la couverture du motif. Cette technique garantit un chevauchement des intervalles du motif sur au moins un objet de la base. Nous appelons *aléatoire* cette méthode. Les codes binaires et les résultats expérimentaux sont accessibles depuis `https://github.com/djawed-bkh/FIP-Sampling`.

Bases de données	NT	AP	BK	Cancer	CH	Yacht	Iris	LW	Glass	balance-scale	diabetes	sonar	heart
Couvertures vides (%)	8	76	66	98	96	81	56	86	99	0	92	100	96

TAB. 3 – *Pourcentage de motifs ayant une couverture vide en tirant des motifs d'intervalles selon une distribution uniforme et sans contrôle sur la couverture.*

1. https ://archive.ics.uci.edu/

5.1 Fréquences des motifs et l'impact de la longue traîne

La figure 1 affiche les fréquences de 500 motifs échantillonnés avec FIPS et la méthode *aléatoire*, les motifs étant triés par ordre décroissant de fréquence. Pour toutes les bases de données, FIPS tire des motifs de fréquence plus élevée que ceux obtenus avec *aléatoire*. Cela s'explique par l'étape 1 de FIPS, qui favorise le tirage d'objets couverts par un grand nombre de motifs, et donc la fréquence, ce qui est le but recherché.

En échantillonnage de motifs, il est bien connu qu'on a tendance à tirer un grand nombre de motifs inintéressants (i.e. des motifs avec de faibles valeurs pour la mesure considérée). Ce phénomène statistique est notamment discuté par Anderson (2004). Il est caractéristique d'une distribution déséquilibrée de motifs : un petit nombre de motifs très fréquents forme la tête, tandis qu'un grand nombre de motifs de faible fréquence compose la longue traîne. La figure 1 montre que FIPS est nettement moins sensible au phénomène de la longue traîne que la méthode *aléatoire*. Pour les bases *diabetes* et *cancer*, 65% (resp. 68.2%) des motifs issus de FIPS relèvent de la traîne (fréquence inférieure à 1%), contre environ 99% pour *aléatoire*. Pour les bases *balance-scale* et *NT*, FIPS génère très peu de motifs de faible fréquence, rendant la traîne difficile à distinguer. À l'inverse, la méthode *aléatoire* produit 9.4% et 10% de motifs ayant une fréquence inférieure à 1% pour *balance-scale* et *NT* respectivement.

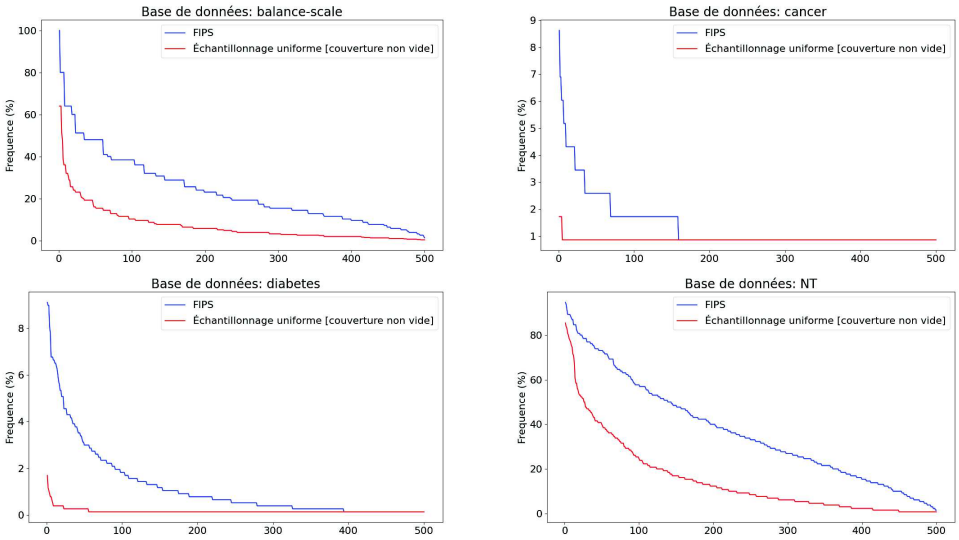

FIG. 1 – *Évaluation de la fréquence pour 500 motifs tirés par* FIPS *et la méthode aléatoire*

5.2 Évaluation de la diversité

La diversité entre motifs est une propriété recherchée en échantillonnage de motifs parce que, dans un processus de fouille interactive, l'analyste préfère des motifs offrant plusieurs facettes de l'information contenue dans les données. Nous utilisons la mesure présentée par Giacometti et Soulet (2018) qui est adaptée aux données numériques et définie comme suit : $diversity(K,\mathcal{N}) = |\{cover(\mathcal{V}_1,\mathcal{N}),\ldots,cover(\mathcal{V}_{|K|},\mathcal{N})\}| \ /|K|$ où $|K|$ est le nombre de motifs

échantillonnés. La figure 2 montre que les motifs échantillonnés par FIPS ont une plus grande diversité que ceux échantillonnés par la méthode *aléatoire* sauf en ce qui concerne la base *balance-scale*. Les motifs tirés par FIPS ont une fréquence plus élevée que ceux tirés par la méthode *aléatoire*, ils possèdent ainsi une couverture plus grande ce qui augmente la probabilité d'une variation de cette couverture lors des autres tirages. L'exception observée sur *balance-scale* pourrait s'expliquer par les caractéristiques spécifiques de cette base. Celle-ci contient un faible nombre de valeurs distinctes et d'attributs (cf. tableau 2) et le tirage biaisé par la fréquence conduit souvent aux mêmes objets et à des motifs d'intervalles similaires.

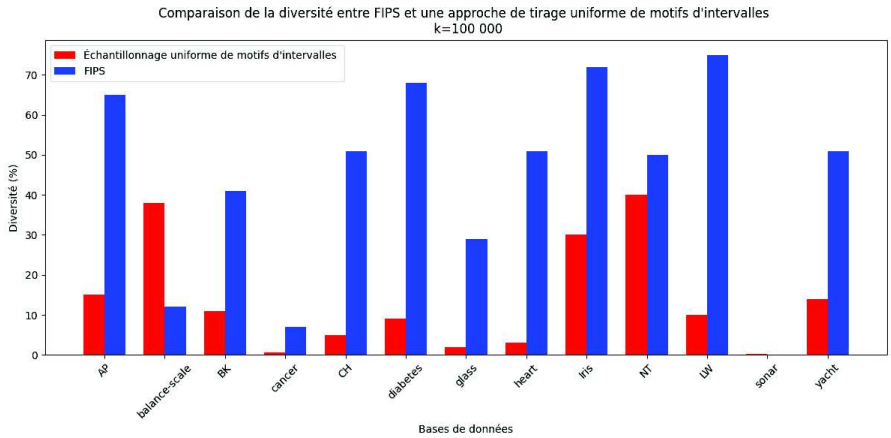

FIG. 2 – *Diversité des méthodes* FIPS *et aléatoire*

5.3 Évaluation du temps CPU

La figure 3 indique le temps CPU [2] nécessaire pour le tirage de chaque motif d'un ensemble de 500 motifs d'intervalles échantillonnés d'une part avec FIPS et d'autre part avec la méthode *aléatoire*. On constate que, pour la majorité des bases de données, FIPS est plus rapide que la méthode *aléatoire*. Cela s'explique par le fait que la méthode *aléatoire* nécessite, pour chaque intervalle du motif, de rechercher des valeurs distinctes menant à des couvertures non vides. La méthode *aléatoire* réalise cette opération en recalculant la couverture courante du motif en train d'être construit pour chaque nouvel intervalle ajouté (soit $|\mathcal{M}|$ fois au total). Cette procédure explique également les oscillations que nous observons pour cette méthode. Si les premiers intervalles tirés sont larges, le temps de calcul de la couverture sera élevé. En revanche, si ces intervalles sont plus restreints, le temps de calcul sera plus faible. Pour certaines bases de données, comme *balance-scale*, on constate que les performances des deux méthodes sont comparables. Cela est dû à la petite taille de cette base et au faible nombre de valeurs distinctes (voir tableau 2), réduisant ainsi le coût de calcul des couvertures courantes.

2. Configuration : Intel core i7, 11ème génération, 3 GHz, 8 coeurs, 30 Go de RAM

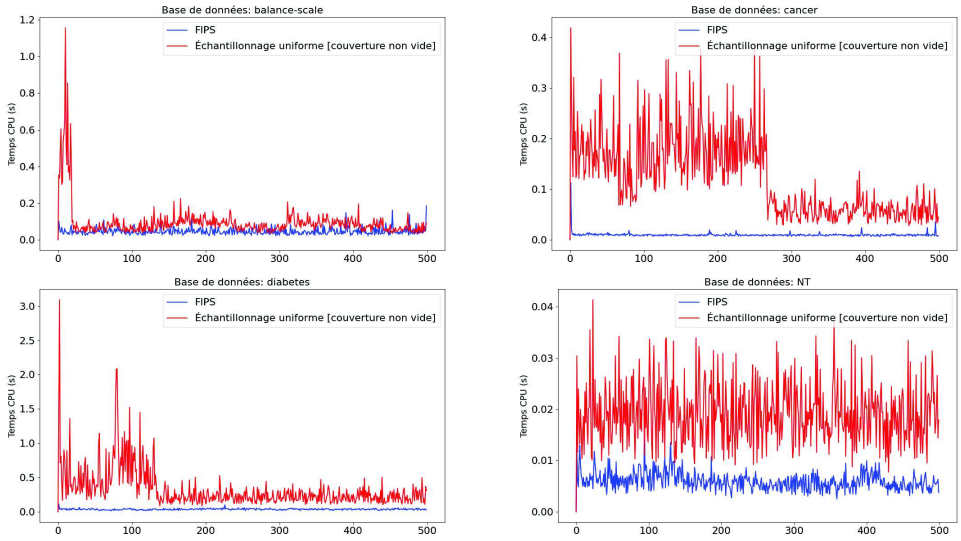

FIG. 3 – *Évolution du temps CPU pour un ensemble de motifs tirés par les méthodes* FIPS *et aléatoire*

5.4 Évaluation de la plausibilité

Nous comparons maintenant la pertinence des motifs tirés par FIPS et la méthode *aléatoire* selon le protocole introduit par Gionis et al. (2006) et adapté aux données numériques dans Giacometti et Soulet (2018) dans le cas de la fréquence. Le principe est d'évaluer la pertinence d'un motif en comparant sa fréquence dans la base de données originale avec celle qu'il possède dans une base de données randomisée notée $\mathcal{N}_{rand}$ où les corrélations sont perturbées. Moins un motif est retrouvé dans la base de données randomisée, plus il est estimé pertinent. Formellement, la plausibilité est définie comme suit : $Plausibility(K,\mathcal{N}) = \sum_{\mathcal{V}_i \in k} \sum_{j=1}^{R} (freq(\mathcal{V}_i, \mathcal{N}) - freq(\mathcal{V}_i, \mathcal{N}^j_{rand})) / \sum_{i=1}^{|K|} (R \times freq(\mathcal{V}_i, \mathcal{N}))$ où R est le nombre de bases randomisées. Nous avons défini des intervalles de fréquence (cf. figure 4) et, pour chaque intervalle et chaque méthode, nous avons sélectionné 10 000 motifs en rejetant ceux dont la fréquence ne respecte pas les seuils définis. La restriction liée aux intervalles de fréquence a pour but de simuler l'intérêt d'un analyste qui ne s'intéresse pas aux motifs très fréquents (souvent trop génériques) ni ceux trop rares (souvent non représentatifs). Le fait de considérer plusieurs intervalles permet de simuler plusieurs valeurs de seuils de fréquence estimés utiles. L'échantillonnage est réalisé dans une limite de temps d'exécution de 5 minutes.

La plausibilité a tendance à diminuer à mesure que la fréquence augmente car plus un motif est fréquent, plus il y a de chance que des occurences de ce motif se retrouvent dans une base randomisée. FIPS ayant un tirage biaisé sur la fréquence, il est donc attendu que les motifs tirés par FIPS aient en moyenne une plausibilité plus faible que ceux tirés par la méthode *aléatoire*. La partie A de la figure 4 confirme ce fait. On constate également que plus les seuils de fréquences augmentent, plus les plausibilités des deux méthodes se rapprochent. Cela s'explique par le fait que plus le seuil de fréquence augmente, moins il y a de motifs

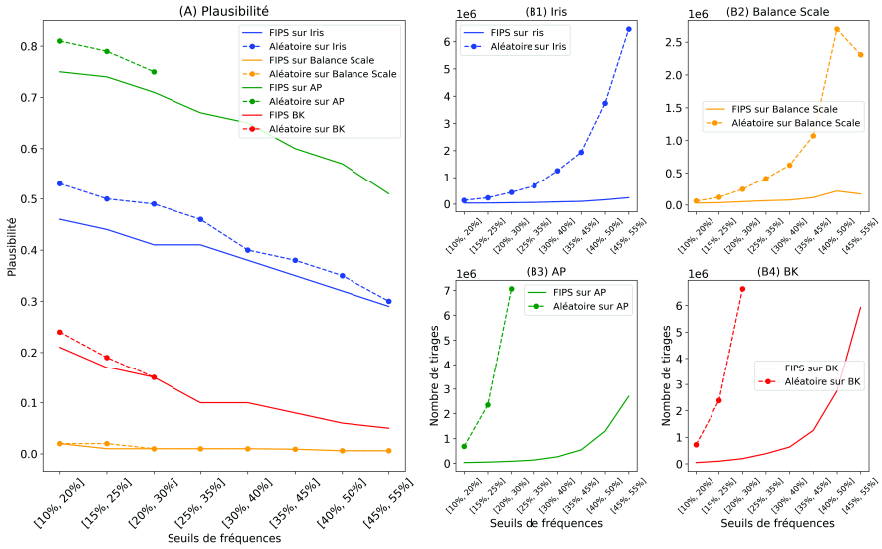

FIG. 4 – *Évaluation de la plausibilité des méthodes* FIPS *et aléatoire (échantillon de 10000 motifs) sur différents seuils de fréquences*

fréquents. Pour les bases *AP* et *BK*, la méthode *aléatoire* ne trouve pas dans le temps imparti le nombre requis de motifs respectant les seuils de fréquence tandis que FIPS y parvient.

Les graphiques B_1, B_2, B_3 et B_4 de la figure 4 montrent que, pour toutes les bases de données, le nombre de tirages nécessaires par la méthode *aléatoire* pour atteindre le nombre de motifs souhaité est largement supérieur au nombre de tirages effectué par FIPS. Pour des seuils de fréquence allant de 10% à 35%, la méthode *aléatoire* nécessite 2 à 11 fois plus de tirages que FIPS sur *iris* et *balance-Scale*. Lorsque les seuils de fréquence augmentent (entre 35 % et 45 %), cette différence est plus marquée, jusqu'à 27 fois plus pour *iris*. L'écart est encore plus prononcé pour les bases *AP* et *BK* : 15 à 80 fois plus de tirages pour les faibles seuils de fréquence et la méthode *aléatoire* ne parvient pas à tirer le nombre requis de motifs pour les seuils de fréquence élevés.

6 Conclusion

Dans cet article, nous avons présenté FIPS, une méthode d'échantillonnage pour les données numériques. Cette méthode repose sur une représentation par intervalles des motifs qui permet de préserver toute l'information originelle. Nous avons prouvé que FIPS échantillonne les motifs proportionnellement à la fréquence. Nous avons montré expérimentalement la qualité des motifs obtenus selon plusieurs critères (fréquence, diversité, rapidité, plausibilité) ainsi que la résistance de FIPS face au phénomène de la longue traîne.

En perspective, il serait intéressant de caractériser l'ensemble des mesures pouvant être traitées par notre approche dans le cadre des motifs d'intervalles. Cela semble assez naturel pour des mesures définies sur un attribut mais nettement plus ambitieux pour d'autres mesures

comme la densité. Il serait également intéressant d'étudier l'intégration de telles techniques d'échantillonnage de motifs dans des applications mettant en œuvre un processus de fouille interactive.

Références

Anderson, C. (2004). The long tail. *Wired magazine*, doi : .

Bekkoucha, D., A. Ouali, P. Boizumault, et B. Crémilleux (2024). Efficiently mining closed interval patterns with constraint programming. In *CPAIOR 2024, Uppsala, Sweden*.

Bendimerad, A., J. Lijffijt, M. Plantevit, C. Robardet, et T. D. Bie (2020). Gibbs sampling subjectively interesting tiles. In *IDA 2020, Germany,*.

Boley, M., T. Gärtner, et H. Grosskreutz (2010). Formal concept sampling for counting and threshold-free local pattern mining. In *SDM 2010, USA*.

Boley, M., C. Lucchese, D. Paurat, et T. Gärtner. Direct local pattern sampling by efficient two-step random procedures. In *SIGKDD 2011, USA*.

Camelin, F., S. Loudni, G. Pesant, et C. Truchet (2024). Échantillonnage d'ensemble de motifs diversifiés par compression locale. In *EGC 2024, Dijon, France*.

Diop, L., C. T. Diop, A. Giacometti, D. Li, et A. Soulet (2020). Sequential pattern sampling with norm-based utility. *Knowl. Inf. Syst.*.

Dzyuba, V., M. van Leeuwen, et L. D. Raedt (2017). Flexible constrained sampling with guarantees for pattern mining. *Data Min. Knowl. Discov.*.

Giacometti, A. et A. Soulet (2018). Dense neighborhood pattern sampling in numerical data. In *SDM 2018 USA*, doi : .

Gionis, A., H. Mannila, T. Mielikäinen, et P. Tsaparas (2006). Assessing data mining results via swap randomization. In *SIGKDD 2006 PA, USA,*.

Hasan, M. A. et M. J. Zaki. Output space sampling for graph patterns. *VLDB 2009 Endow.*.

Kaytoue, M., S. O. Kuznetsov, et A. Napoli (2011). Revisiting numerical pattern mining with formal concept analysis. In *IJCAI 2011, Barcelona, Spain*.

Soulet, A. (2023). Echantillonnage de motifs avec une contrainte de fréquence. In *EGC 2023, Lyon, France*.

Srikant, R. et R. Agrawal (1996). Mining quantitative association rules in large relational tables. In *SIGMOD 1996 , Canada,*.

Summary

In this paper, we introduce FIPS, the first method for sampling interval patterns from numerical data. FIPS is built on the multi-step approach framework (Boley et al.). We formally demonstrate that FIPS samples interval patterns in proportion to their frequency. The experiments, covering various research questions, highlight the quality of the patterns sampled by FIPS for effective knowledge discovery.