Heterogeneous Pattern Sampling according to Frequency

Rayane Lachache, Djawad Bekkoucha, Abdelkader Ouali, Bruno Crémilleux, Thi-Bich-Hanh Dao, Christel Vrain

Heterogeneous Pattern Sampling according to Frequency

Rayane Lachache[1], Djawad Bekkoucha[1,3], Abdelkader Ouali[1], Bruno Crémilleux[1], Thi-Bich-Hanh Dao[2], and Christel Vrain[2]

[1] University of Caen Normandy, CNRS, UMR 6072 G REYC, 14032 Caen, France
firstname.lastname@unicaen.fr
[2] University of Orléans, INSA Centre Val de Loire, LIFO EA 4022, Orléans, France
firstname.lastname@univ-orleans.fr
[3] LISN, CNRS (UMR 9015), Paris Saclay Universit y, Gif-sur-Yvette, France
firstname.lastname@lisn.fr

Abstract. Pattern sampling has emerged as an efficient paradigm for extracting knowledge from large-scale tabular datasets. By stochastically drawing patterns in proportion to a specified interestingness measure, such as frequency, these methods provide an anytime mechanism able to supply users with representative patterns. Existing approaches, however, operate only on homogeneous data structures such as sequential, numerical, or transactional data. In contrast, many real-world applications produce datasets that combine several heterogeneous types of information. In this article, we introduce the first frequency-based sampling method designed specifically for heterogeneous tabular datasets. Our method employs a multi-step decomposition of the sampling process and tackles the central challenge of precisely computing, for each data instance, the exact number of heterogeneous patterns that cover it. We provide a formal proof establishing that the resulting sample is proportional to pattern frequency. A comprehensive experimental study shows the quality of the drawn patterns using several indicators.

Keywords: Pattern Sampling · Data Mining · Heterogeneous tabular Data

1 Introduction

Many works in pattern mining are devoted on a single type of data such as Boolean data, sequential data or graph data [1]. However it is common to cope with heterogeneous data where data come from different type sources. For example, when studying radiologists' behaviour by analysing their sequences of actions over time, it is valuable to enrich this analysis with additional contextual information about the radiologists and their working situation. These pieces of information, which can be categorical, numerical and sequential lead to an heterogeneous tabular database representation. In order to enable the development of user-centered methods that address, in particular, the problem of the huge

number of patterns produced by exhaustive methods, output pattern sampling is an efficient approach that instantly returns patterns thus enabling to produce pattern-based models at any time [2,5]. It consists in randomly selecting a sample of patterns with a probability proportional to an interestingness measure among the patterns that would have been mined from the complete dataset. For example, a pattern X_1 that is twice as frequent as a pattern X_2 will be twice as likely to be selected. A weighted draw based on the interestingness measure on the whole set of patterns is not possible due to the large size of the search space. The first pattern sampling approach was introduced by Hasan and Zaki [2] and it uses a Markov chain Monte Carlo algorithm. It handles a graph pattern language and is biased towards the frequency measure. Boley et al. [5] defined an approach based on two successive sampling steps. It consists in drawing an instance of the dataset and then drawing a pattern contained in this instance. By judiciously selecting the two draw distributions, this strategy *formally* guarantees an exact sampling according to the distribution induced by the interestingness measure. There are a couple of works following this strategy and addressing several pattern languages such as Boolean data [5], sequences [7] or numerical data [11,3]. However, these pattern languages are considered separately. To the best of our knowledge, there is no work addressing pattern sampling in heterogeneous data. In this paper, we propose SEHP, the first method for sampling heterogeneous patterns. SEHP is based on a two-step method [5] and instantly returns heterogeneous patterns sampled proportional to frequency. We formally prove that SEHP draw patterns proportionally to their frequency. The main challenge lies in computing the exact number of heterogeneous patterns that cover a given object in order to find an appropriate two-step decomposition to obtain the desired distribution. A large set of experimental results according to several criteria (frequency, impact of the long-tail phenomenon, diversity of the sampled patterns, and overlap of their coverages) assess the quality of sampled patterns. Section 2 introduces the preliminaries and the problem statement. Section 3 deals with related work. In Section 4, we present our SEHP method, and Section 5 provides a comprehensive set of experiments.

2 Preliminaries

Heterogeneous Database. A heterogeneous database is defined as a pair $\mathcal{H} = (\mathcal{G}, \mathcal{M})$, where $\mathcal{G}$ denotes the set of objects and $\mathcal{M} = \{M_1, \ldots, M_n\}$ denotes the set of attributes describing these objects. The set of attributes $\mathcal{M}$ is made of three disjoint subsets. First, the binary attributes subset, denoted $\mathcal{M}_B = \{M_1, \ldots, M_i\}$ where $i < n$, consists of attributes $b \in \mathcal{M}_B$ that take their values from the domain $\mathcal{H}_B = \{0, 1\}$. Next is the subset of numerical attributes, $\mathcal{M}_N = \{M_{i+1}, \ldots, M_{n-1}\}$ where each attribute $f \in \mathcal{M}_N$ takes its values from a finite domain $\mathcal{H}_N$, which is defined as the set union of all values assigned to $\mathcal{M}_N$. The last subset is the single sequential attribute $\mathcal{M}_S = \{M_n\}$, whose domain is denoted by $\mathcal{H}_S$ and defined over a set of literals $\mathcal{I}$. An itemset s is defined as a subset of $\mathcal{I}$. A sequence $s = \langle s_1, \ldots, s_c \rangle$ is an ordered list of non-empty itemsets,

where $s_i \subseteq \mathcal{I}$ for $1 \leq i \leq c$, and its size is denoted by $|s| = c$. Each object $g \in \mathcal{G}$ is represented by a vector $\langle v_{g,m} \rangle_{m \in \mathcal{M}}$, where each component $v_{g,m}$ belongs to its corresponding domain $\mathcal{H}_m$.

Example 1. Let Table 1 be our running example of a heterogeneous database containing 4 objects $\mathcal{G} = \{g_1, g_2, g_3, g_4\}$, described over three binary attributes $\mathcal{M}_B = \{b_1, b_2, b_3\}$, two numerical attributes $\mathcal{M}_N = \{f_1, f_2\}$, and one sequential attribute $\mathcal{M}_S = \{S\}$.

Table 1. Example of a heterogeneous database H.

	$\mathcal{M}_B$			$\mathcal{M}_N$		$\mathcal{M}_S$
$\mathcal{G}$	b_1	b_2	b_3	f_1	f_2	S
g_1	1	0	1	7	12	$\langle (ab), (c) \rangle$
g_2	0	1	0	4	9	$\langle (ab), (c), (ac) \rangle$
g_3	0	0	1	5	11	$\langle (ab), (cd) \rangle$
g_4	1	0	1	6	15	$\langle (a), (abd), (abc), (bd) \rangle$

Heterogeneous Pattern. A pattern extracted from the dataset $\mathcal{H}$ is called a heterogeneous pattern and is represented as a triple $X_{\mathrm{H}} = \langle X_{\mathrm{B}}, X_{\mathrm{N}}, X_{\mathrm{S}} \rangle$, where each component corresponds to a specific pattern type. X_{B} is a subset of the binary attributes $\mathcal{M}_B$. X_{N} is an interval pattern defined as a vector of intervals $X_{\mathrm{N}} = \langle [a_f, b_f] \rangle_{\forall f \in \mathcal{M}_N}$, i.e. such that $a_f, b_f \in \mathcal{H}_N$. Each interval of the vector corresponds to a numerical attribute, following a canonical order on the set $\mathcal{M}_N$. X_{S} is a sequential pattern extracted over the set of sequences in $\mathcal{H}$. A sequential pattern $X_{\mathrm{S}} = \langle s'_1, \ldots, s'_m \rangle$ is said to be a subsequence of $s = \langle s_1, \ldots, s_c \rangle$, denoted $X_{\mathrm{S}} \preceq s$, if there exists an index sequence $1 \leq i_1 < i_2 < \cdots < i_m \leq c$ such that $s'_j \subseteq s_{i_j}$ for all $j \in [1..m]$, with $m \leq c$.

Projection Operators. A set of projection operators is defined over the disjoint attribute subsets $(\mathcal{M}_B, \mathcal{M}_N, \mathcal{M}_S)$. For an object $g \in \mathcal{G}$, each operator associates the most specific pattern type that describes g according to the corresponding attribute subset.

$$\begin{cases} g[\mathcal{M}_B] = \{b \in \mathcal{M}_B \mid v_{g,b} = 1\}, \\ g[\mathcal{M}_N] = \langle [v_{g,f}, v_{g,f}] \rangle_{f \in \mathcal{M}_N}, \\ g[\mathcal{M}_S] = v_{g,S},\ S \in \mathcal{M}_S. \end{cases}$$

The projection $g[\mathcal{M}_B]$ on the set of binary attributes $\mathcal{M}_B$ returns the finite set of attributes present in g, i.e. such that $v_{g,b} = 1$. The projection $g[\mathcal{M}_N]$ represents the interval vector restricted to the values of the numerical attributes $\mathcal{M}_N$. The projection $g[\mathcal{M}_S]$ returns the sequence $v_{g,S}$ associated with g.

Example 2. Applying the projection operators to g_1 results in the following:

$$g_1[\mathcal{M}_B] = \{b_1, b_3\}, \quad g_1[\mathcal{M}_N] = \langle [7,7], [12,12] \rangle, \quad g_1[\mathcal{M}_S] = \langle (ab), (c) \rangle.$$

Coverage and frequency. An object $g \in \mathcal{G}$ is said to be in the coverage of a heterogeneous pattern X_{H}, denoted by $g \models X_{\mathrm{H}}$, if and only if the following conjunctive conditions are satisfied: $X_{\mathrm{B}} \subseteq g[\mathcal{M}_B] \ \wedge \ X_{\mathrm{N}} \sqsubseteq g[\mathcal{M}_N] \ \wedge \ X_{\mathrm{S}} \preceq g[\mathcal{M}_S]$. More precisely, X_{B} must be a subset of the set $g[\mathcal{M}_B]$, and each interval in $g[\mathcal{M}_N]$ must be included in the corresponding interval of X_{N}. Moreover, the sequential pattern X_{S} must be a subsequence of the sequence associated with g.

Example 3. Let $X_{\mathrm{H}} = \big(\{b_3\}, \langle [6,8],[12,15]\rangle, \langle (a)(c)\rangle\big)$ be a sampled pattern from $\mathcal{H}$. The object g_1 is in the cover of X_{H}, since it satisfies all the three conditions: $(g1 \models X_{\mathrm{H}}) \equiv \{b_3\} \subseteq \{b_1, b_3\} \wedge [6,8] \supseteq ([v_{1,1}, v_{1,1}] = [7,7]) \wedge [12,15] \supseteq ([v_{1,2}, v_{1,2}] = [12,12]) \wedge \langle (a)(c)\rangle \preceq \langle (ab),(c)\rangle$.

The set of all objects in $\mathcal{G}$ that satisfy the heterogeneous pattern X_{H} is defined as its cover in the dataset $\mathcal{H}$: $\mathrm{cover}(X_{\mathrm{H}}, \mathcal{H}) = \{\, g \in \mathcal{G} \mid g \models X_{\mathrm{H}} \,\}$. The cardinality of this set corresponds to the frequency of X_{H} in the dataset $\mathcal{H}$, that is, $\mathrm{freq}(X_{\mathrm{H}}, \mathcal{H}) \ = \ \big|\mathrm{cover}(X_{\mathrm{H}}, \mathcal{H})\big|$.

Problem Statement. Let Ω be a population and $f : \Omega \longrightarrow [0,1]$ a measure. The notation $x \sim f(\Omega)$ states that the element x is drawn randomly from Ω with a probability distribution $\pi(x) = f(x)/\sum_{x' \in \Omega} f(x')$. In our work we focus on the language of heterogeneous patterns $\mathcal{L}_H$ and the frequency measure. Given a heterogeneous database $\mathcal{H}$ and $k \in \mathbb{N}$, the problem of sampling heterogeneous patterns consists on extracting k patterns $X_{\mathrm{H}}^1, \ldots, X_{\mathrm{H}}^k$ from $\mathcal{L}_H$, where each pattern X_{H}^i is randomly drawn, with replacement, with a probability proportional to its frequency value $\mathrm{freq}(X_{\mathrm{H}}^i, \mathcal{H})$, $i \in \{1, \ldots, k\}$.

3 Related work

There are a few works on mining patterns from different type sources. In the context of Relational Concept Analysis, [6] introduced heterogeneous pattern structures as a model to describe objects in a combination of spaces represented in a hybrid formal context such as numerical and Boolean data. Heterogeneity has been studied in multidimensional sequence mining, where sequences are enriched with categorical dimensions [13]. This has been extended to handle multiple levels of granularity by M3SP [14] and it has been further refined by explicitly taking data types and external taxonomies into account [10]. All of these approaches rely on exhaustive pattern extraction over the entire search space and there is no sampling. The literature on pattern output sampling methods can be broadly categorized into three families. The first family is based on stochastic methods such as Markov chain Monte Carlo algorithms. This family was introduced by [2] for sampling graph patterns according to the frequency. In this line of research, Boley et al. [4] handle itemsets and address positive interestingness measures. Despite their accuracy, stochastic approaches often suffer from slow convergence. The second family is based on declarative paradigms. Dzyuba et al. [8] uses a SAT solver to sample itemsets. Several measures are tackled. However, modeling other pattern languages and therefore heterogeneous patterns is challenging as

the encoding needs to be adapted for each language. The last family is composed of multi-step methods. The pioneering work was carried out by Boley at al. [5], using a two-step draw to sample patterns according to the frequency. In short, these methods draw an object of the dataset and then draw a pattern supported by this object. By finding an appropriate decomposition to obtain the desired distribution – which remains challenging – these methods provide formal guarantees on the sampling. This line of research has been extended to other pattern languages such as sequences [7], numerical data [11,3] or richer patterns such as subgroups [12]. However, there is no work on sampling that addresses several pattern languages simultaneously, which is the aim of our work.

4 Two-Step Heterogeneous Pattern Sampling Approach

This section introduces SEHP, our sampling method on heterogeneous patterns (cf. Section 2).

Key Ideas. In order to draw a pattern proportionally to its frequency, it is necessary to compute the sum of the frequencies of all patterns in the solution space, i.e. $Z = \sum_{X \in \mathcal{L}_H} \text{freq}(X, \mathcal{H})$. However, due to the size of this space, it is infeasible to compute this sum by an exhaustive enumeration. By designing an appropriate two-step sampling procedure [5], such an enumeration can be avoided. The key idea is to compute the weight of each object g which is the number of heterogeneous patterns covering g. Indeed, we observe that $\sum_{X \in \mathcal{L}_H} \text{freq}(X, \mathcal{H}) = \sum_{X \in \mathcal{L}_H} \sum_{g \in \mathcal{G}} 1_{g \models X_\text{H}} = \sum_{g \in \mathcal{G}} \sum_{X \in \mathcal{L}_H} 1_{g \models X_\text{H}}$. This equality enables us to reduce the problem to computing the number of patterns that cover the object. Then, knowing the weight of each object makes it possible to efficiently compute Z without exploring the entire pattern space. The principle is then as follows: (1) sample an object proportionally to its weight, and (2) select uniformly a pattern among those that cover it. The main challenge lies in computing the weight of each object. We address this issue by introducing the NHP function.

Number of Heterogeneous Patterns (NHP). The function NHP computes for each object $g \in \mathcal{G}$ the number of heterogeneous patterns covering g. To this end, we first need to determine, for each object, how many sub-patterns of each type cover it. Formally, NHP is defined as follows:

$$\text{NHP}(g) = \text{NI}(g) \times \text{NIP}(g) \times \text{NSP}(g). \tag{I}$$

For each object g, NHP first performs a projection $g[\mathcal{M}_B]$ onto the binary dimension. The function $\text{NI}(g)$ then computes the exact number of attribute-subset pattern X_B covering this projection, as in [5]. A second projection onto the numerical attributes, $g[\mathcal{M}_N]$, allows $\text{NIP}(g)$ to compute the exact number of interval patterns X_N covering g, by reusing the NIP function proposed in [3]. The final projection, $g[\mathcal{M}_S]$, concerns the sequential dimension and enables $\text{NSP}(g)$ to compute the exact number of distinct subsequences X_S over the sequence component of g. This computation follows the procedure described in [9]. By distinct subsequences, we mean that the same subsequence X_S may occur several

times within a single sequence. However, in order to compute Z over $\mathcal{M}_S$, NSP must count each subsequence pattern at most once per sequence, as discussed in [7]. Example 4 illustrates the computation of the number of heterogeneous patterns covering a given object.

Example 4. Consider the object $g_1 \in \mathcal{G}$,

$$g_1[\mathcal{M}_B] = \{b_1, b_3\}, \quad g_1[\mathcal{M}_N] = \langle [7,7], [12,12] \rangle, \quad g_1[\mathcal{M}_S] = \langle (ab), (c) \rangle.$$

(1) *Computation of NI*(g_1). The present binary attributes are b_1 and b_3, hence $\mathrm{NI}(g_1) = 2^{|\{b_1,b_3\}|} = 2^2 = 4$. (2) *Computation of NIP*(g_1). For $f_1 = 7$ with $\mathcal{H}_{f_1} = \{4,5,6,7\}$, let $\mathrm{NIA}(g_1, f_1) = |\{v \in \mathcal{H}_{f_1} \mid v \leq 7\}| \cdot |\{v \in \mathcal{H}_{f_1} \mid v \geq 7\}| = 4 \times 1 = 4$. For $f_2 = 12$ with $\mathcal{H}_{f_2} = \{9, 11, 12, 15\}$, we obtain $\mathrm{NIA}(g_1, f_2) = |\{v \in \mathcal{H}_{f_2} \mid v \leq 12\}| \cdot |\{v \in \mathcal{H}_{f_2} \mid v \geq 12\}| = 3 \times 2 = 6$. Thus, $\mathrm{NIP}(g_1) = 4 \times 6 = 24$. (3) *Computation of NSP*(g_1). The sequence is $\langle (ab), (c) \rangle$. Applying the function NSP yields $\mathrm{NSP}(g_1) = 8$. (4) *Final result.* $\mathrm{NHP}(g_1) = \mathrm{NI}(g_1) \times \mathrm{NIP}(g_1) \times \mathrm{NSP}(g_1) = 4 \times 24 \times 8 = 768$. Hence, the object g_1 is covered by exactly 768 heterogeneous patterns.

NHP complexity. The NHP worst case time complexity is given by :

$$\mathcal{O}\big(|\mathcal{M}_B| + |\mathcal{G}| \cdot |\mathcal{M}_N| + L \cdot 2^L \cdot T\big)$$

The term $|\mathcal{M}_B|$ comes from NI complexity, which scans the number of binary attributes of each object; the term $|\mathcal{G}| \cdot |\mathcal{M}_N|$ corresponds to NIP complexity in the worst case, where $|\mathcal{G}|$ corresponds to the number of distinct values in the worst case of an attribute; the remaining term $L \cdot 2^L \cdot T$ is due to NSP complexity obtained by adapting the complexity analysis reported in [7], where L is the maximum number of itemsets in a sequence and T the maximum itemset size.

SEHP Algorithm. This section presents the sampling algorithm SEHP, described in Algorithm 1. Given a heterogeneous database $\mathcal{H}$, SEHP samples a heterogeneous pattern X_{H} proportionally to its frequency of occurrences in $\mathcal{H}$. In a preprocessing phase, each object $g \in \mathcal{G}$ is assigned a weight $w(g)$, defined as the number of heterogeneous patterns covering g, computed through the NHP function (line 1). The sampling procedure then proceeds in two steps: (i) an object g is drawn with probability proportional to $w(g)$ (line 2). (ii) given the selected g, one heterogeneous pattern X_{H} is drawn uniformly at random among those covering g (line 3). Repeating these two steps independently k times yields a sample of k heterogeneous patterns from $\mathcal{H}$.

Proof. We prove that Algorithm 1 performs sampling proportional to the frequency of patterns. Let $\mathcal{L}_H$ denote the space of all heterogeneous patterns, and let Z be the normalization constant defined as $Z = \sum_{\forall g \in \mathcal{G}} \mathrm{NHP}(g)$. Consider a pattern $X_{\mathrm{H}} \in \mathcal{L}_H$. In Step 1, an object $g \in \mathcal{G}$ is selected with probability proportional to its weight, namely $\pi(g) = \frac{\mathrm{NHP}(g)}{Z}$. In Step 2, if X_{H} covers the object g, then X_{H} is chosen uniformly among the $\mathrm{NHP}(g)$ patterns covering g. In this case, $\pi(X_{\mathrm{H}} \mid g) = \frac{1}{\mathrm{NHP}(g)}$. In the other case, $\pi(X_{\mathrm{H}} \mid$

Algorithm 1 Sampling an heterogeneous pattern proportionally to its frequency (SEHP)

Require: A heterogeneous database $\mathcal{H}$
Ensure: A heterogeneous pattern X_{H} sampled proportionally to its frequency

Preprocessing: For each object $g \in \mathcal{G}$, compute $w(g) = \mathrm{NHP}(g)$.

Step 1: Draw an object $g \sim w(g)$.
Step 2: Uniformly draw a pattern X_{H} among those covering g.
  $X_{\mathrm{B}} \leftarrow$ `DrawUniformItemset`$(\mathcal{M}_B, g[\mathcal{M}_B])$.
  $X_{\mathrm{N}} \leftarrow$ `DrawUniformIntervalPattern`$(\mathcal{M}_N, g[\mathcal{M}_N])$.
  $X_{\mathrm{S}} \leftarrow$ `DrawUniformSubSequence`$(\mathcal{M}_S, g[\mathcal{M}_S])$.
return $X_{\mathrm{H}} \leftarrow \langle X_{\mathrm{B}}, X_{\mathrm{N}}, X_{\mathrm{S}} \rangle$

function DrawUniformItemset($\mathcal{M}_B, g[\mathcal{M}_B]$)
  Initialize $X_{\mathrm{B}} \leftarrow \emptyset$.
  for all present attributes $i \in g[\mathcal{M}_B]$ **do**
    Include i in X_{B} with probability 1/2.
  end for
  return X_{B}
end function

function DrawUniformIntervalPattern($\mathcal{M}_N, g[\mathcal{M}_N]$)
  Initialize $X_{\mathrm{N}} \leftarrow \langle\ \rangle$.
  for all numerical attributes $f \in \mathcal{M}_N$ **do**.
    Let $v_{g,f}$ be the value of feature f from $g[\mathcal{M}_N]$.
    Let $A_f = \{v \in \mathcal{H}_f \mid v \leq v_{g,f}\}$ and $B_f = \{v \in \mathcal{H}_f \mid v \geq v_{g,f}\}$.
    Draw a_f uniformly from A_f and b_f uniformly from B_f.
    Append the interval $[a_f, b_f]$ to X_{N}.
  end for
  return X_{N}
end function

function DrawUniformSubSequence($\mathcal{M}_S, g[\mathcal{M}_S]$)
  Let F be the flatten list of all items in the sequence $g[\mathcal{M}_S]$.
  repeat
    Initialize a set of positions $P \leftarrow \emptyset$.
    for all positions $j \in \{1, \ldots, |F|\}$ **do**
      Include j in P with probability 1/2.
    end for
    $X_{\mathrm{S}} \leftarrow$ Group items at positions in P which belong to the same itemset in $g[\mathcal{M}_S]$.
  until the sub-sequence X_{S} starts at the leftmost occurrence in $g[\mathcal{M}_S]$.
  return X_{S}
end function

$g) = 0$. By combining the two steps, we obtain $\pi(X_{\mathrm{H}}) = \sum_{g \in \mathcal{G}} \pi(g)\, \pi(X_{\mathrm{H}} \mid g) = \sum_{g \in \mathrm{cover}(X_{\mathrm{H}}, \mathcal{H})} \frac{\mathrm{NHP}(g)}{Z} \times \frac{1}{\mathrm{NHP}(g)}$. The terms simplify, leading to $\pi(X_{\mathrm{H}}) = \sum_{g \in \mathrm{cover}(X_{\mathrm{H}}, \mathcal{H})} \frac{1}{Z} = \frac{|\mathrm{cover}(X_{\mathrm{H}}, \mathcal{H})|}{Z}$. Since $|\mathrm{cover}(X_{\mathrm{H}}, \mathcal{H})|$ is precisely the frequency of X_{H} in $\mathcal{H}$, it follows that $\pi(X_{\mathrm{H}}) = \frac{\mathrm{freq}(X_{\mathrm{H}}, \mathcal{H})}{Z}$. Therefore, the probability of drawing a pattern is proportional to its frequency.

5 Experiments

The experimental study aims to evaluate the efficiency of SEHP by addressing the following questions.[4]

1. What is the quality of the heterogeneous patterns sampled with respect to frequency, and how does the long-tail phenomenon affect SEHP?
2. What is the diversity of the patterns sampled by SEHP?
3. To what extent do the sampled pattern's overlap ?
4. What is the efficiency of SEHP in terms of CPU time?

5.1 Synthetic Datasets

In order to assess the performance of our algorithm, we use a configurable generator of heterogeneous tabular datasets that we have developed for the evaluation. Given the number of objects $|\mathcal{G}|$, the generator creates instances $g \in \mathcal{G}$ based on the following parameters. For a given number of binary attributes, a relative density parameter $\rho_B \in [0, 1]$ is used to specify the total number of randomly selected pairs $(g \in \mathcal{G}, b \in \mathcal{M}_B)$ whose value is equal to 1 in the dataset. All the non-selected pairs are equal to zero. For a given number of numerical attributes, a parameter $\rho_N \in [0, 1]$ is used to specify the number of distinct numerical values used in the dataset proportional to the total number of pairs $(g \in \mathcal{G}, f \in \mathcal{M}_N)$. Each pair is assigned to a value drawn uniformly from the given set of distinct values. For the sequential part, the generator considers a set of literals $\mathcal{I}$. Each literal $i \in \mathcal{I}$ is attributed an occurrence rate ρ_i, which is uniformly drawn from the given interval $[\rho_{\min}, \rho_{\max}]$. This rate ρ_i then determines the proportion of objects associated with the literal i. Each itemset in the sequence is constituted by a set of literals which are uniformly drawn from those associated to g. These literals are selected without replacement throughout the sequence generation, ensuring that the selected literal is used exactly once in the current itemset. For each object g, the length of its sequence, in terms of the number of itemsets, is drawn uniformly between two bounds $\eta_{\min}$ and $\eta_{\max}$.

[4] All source code (including the SEHP implementation, the synthetic data generator, and the baseline approach), as well as the full set of experimental results, are available at `https://github.com/lacray/SEHP-Sampling-heterogeneous-pattern-proportionally-to-its-frequency.git`

Table 2. Characteristics of the 10 synthetic datasets used in the experiments.

Dataset	$\lvert\mathcal{G}\rvert$	$\lvert\mathcal{M}_B\rvert$	ρ_B	$\mathcal{M}_N$	ρ_N	$\lvert\mathcal{I}\rvert$	$\lvert X_S\rvert_{max}$	$\lvert X_S\rvert_{mean}$	$\lvert X_S\rvert_{\sigma}$
test1	10000	13	80	18	18000	15	103	40.32	11.65
test2	250	4	80	4	20	4	15	8.88	2.27
test3	500	4	80	4	20	4	16	8.84	2.27
test4	250	9	80	6	15	4	21	12.28	3.25
test5	850	3	35	5	85	7	20	9.28	3.19
test6	1500	2	80	3	18	5	18	8.61	3.26
test7	800	6	20	4	19	9	35	12.21	5.17
test8	800	6	70	8	1216	6	18	8.54	2.84
test9	900	7	70	5	22	20	53	20.45	7.67
test10	750	6	70	6	540	7	30	14.0	4.56

By varying the parameters $(\rho_B, \rho_N, |\mathcal{I}|, \rho_{\min}, \rho_{\max}, \eta_{\min}, \eta_{\max})$, we can generate datasets with different characteristics across the binary, the numerical, and the sequential attributes. Moreover, we do not enforce any explicit similarity structure between heterogeneous objects, which makes the sampling of frequent patterns more challenging. Table 2 summarises all generated datasets and their main characteristics.

5.2 Independent Sampling Approach (I-SEHP)

As there is no existing approach addressing the sampling of heterogeneous patterns, we therefore introduce I-SEHP as an intuitive comparative baseline. Given a heterogeneous database, I-SEHP independently samples one pattern from each component using the corresponding state-of-the-art algorithm, and then aggregates the three resulting sub-patterns to construct a heterogeneous pattern X_H. However, we observe that a substantial proportion of the sampled patterns have an empty coverage, as illustrated in Table 3. This situation arises when no object simultaneously belongs to the coverages of the three sub-patterns. Since

Table 3. Percentage of patterns sampled by I-SEHP having empty coverage (50 000 sampled patterns).

Databases	**test1**	**test2**	**test3**	**test4**	**test5**	**test6**	**test7**	**test8**	**test9**	**test10**
% empty-coverage	99.99	57.95	51.96	65.03	69.56	62.36	67.37	71.12	74.70	77.37

such patterns are of limited practical interest for the user, we incorporate a rejection mechanism to address this issue. More precisely, whenever the coverage of a sampled pattern X_H is empty, the pattern is rejected and the sampling procedure is repeated until a pattern covering at least one object is obtained (cf Algorithm 2). This mechanism preserves frequency-proportional sampling within each component while providing a meaningful and relevant baseline for comparison.

Algorithm 2 I-SEHP

Require: Heterogeneous database $\mathcal{H}$; flag `ALLOWSEmptyCovers` $\in \{\texttt{true}, \texttt{false}\}$
Ensure: A heterogeneous pattern X_{H}

Preprocessing: define weights $w_{\mathcal{D}_B}(g) = \mathrm{NI}(g)$, $w_{\mathcal{D}_N}(g) = \mathrm{NIP}(g)$, $w_{\mathcal{D}_S}(g) = \mathrm{NSP}(g)$.
repeat
  Step 1 — Separate two-step sampling in each sub-database: Sample X_{B} from $\mathcal{D}_B$, X_{N} from $\mathcal{D}_N$, and X_{S} from $\mathcal{D}_S$
  Step 2 — Aggregation: $X_{\mathrm{H}} \leftarrow (X_{\mathrm{B}}, X_{\mathrm{N}}, X_{\mathrm{S}})$
  Coverage: $\mathrm{cover}(X_{\mathrm{H}}, \mathcal{H}) \leftarrow \mathrm{cover}(X_{\mathrm{B}}, \mathcal{D}_B) \cap \mathrm{cover}(X_{\mathrm{N}}, \mathcal{D}_N) \cap \mathrm{cover}(X_{\mathrm{S}}, \mathcal{D}_S)$
  if `ALLOWSEmptyCovers` = `true` **then**
    break ▷ accept the pattern even if $\mathrm{cover}(X_{\mathrm{H}}, \mathcal{H}) = \varnothing$
  end if
until $\mathrm{cover}(X_{\mathrm{H}}, \mathcal{H}) \neq \varnothing$
return X_{H}

5.3 Pattern Frequency and the Impact of the Long Tail

Fig. 1 presents the frequencies of 500 patterns sampled with SEHP and I-SEHP. The patterns are sorted in descending order of frequency. We observe that I-SEHP samples patterns with a higher frequency across all datasets. This is largely a consequence of the sampling procedure's reliance on stochastic independence, where each draw of a sub-pattern is carried using only a subset of attributes. This process, particularly when subjected to rejection, tends to select frequent sub-patterns which leads to increased overlap in their coverage of objects when such overlap exists. It should be noted, however, that the rejection mechanism could be time consuming, as discussed in CPU-time subsection.

In pattern sampling, when an interestingness measure proportional to frequency is applied, it is well known that methods suffer from the curse of the long tail. This phenomenon is characterized by a highly asymmetrical distribution, where a large number of patterns presenting a low interest value form the tail, while a small number of patterns with a high interest value form the head. Assuming the long tail is associated with patterns having a frequency lower than 1%, our two methods do not escape from this curse. More precisely, for the test6 dataset, 15% of I-SEHP sampled patterns are non-tail (vs. 10.2% for SEHP), and on the imbalanced test8 dataset, I-SEHP only manages 0.6% non-tail patterns, whereas SEHP patterns are completely in the long-tail. Furthermore, test4 being a variant of test3 having more attributes but fewer objects, a significant decrease in performance is observed. More precisely, the non-tail rate of SEHP drops from 14.8% on test3 to 2.8% on test4, and the rate of I-SEHP falls from 22.8% to 10.8%, showing that increased dimensionality has an effect on the efficient sampling of frequent pattern, even when the overall object count is reduced. We subsequently evaluated the sampled heterogeneous patterns through their sub-patterns. More precisely, we consider a heterogeneous pattern to be more relevant for the analyst when its sub-patterns do not exhibit a full coverage of the objects in the dataset. Therefore, for each dataset, we report for

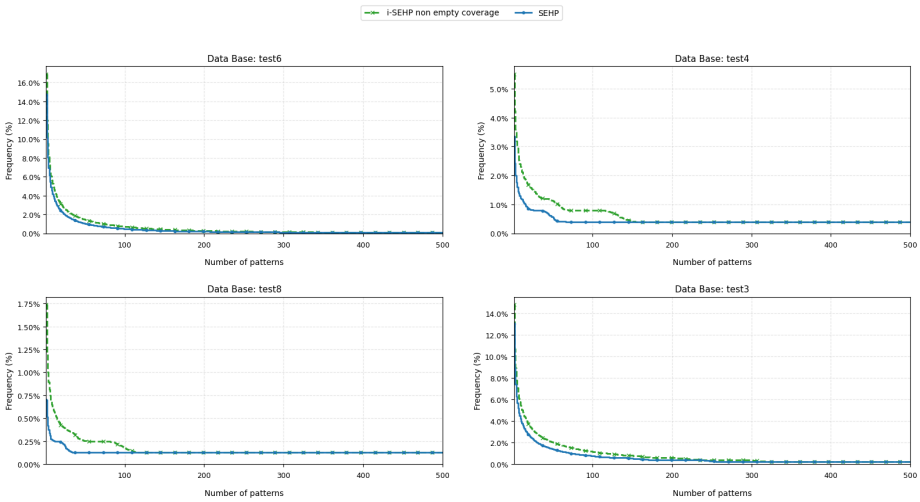

Fig. 1. Frequency evaluation of 500 patterns sampled by SEHP and I-SEHP methods.

the 100 most frequent patterns sampled by both methods the number of sub-patterns having a full coverage of all the objects in the dataset. The threshold of 100 frequent patterns is determined by observing the Area Under the Curve in Fig. 1 which becomes significantly larger starting from the top 100 patterns. The results obtained are illustrated in Fig. 2. We observe that SEHP exhibits slightly better sub-patterns than I-SEHP on relatively balanced datasets such as test3 and test6. However, SEHP yields significantly superior results on the more imbalanced datasets, namely test5 and test7. Extracting respectively 44 and 50 patterns carrying better sub-patterns compared to only 26 and 17 for I-SEHP.

5.4 CPU Time

Fig. 3a shows the cumulative CPU time required for sampling 500 patterns using SEHP and I-SEHP. We observe that SEHP is faster on the majority of datasets, providing the required sample in a near-instantaneous time compared to I-SEHP. This difference in performance is caused by the rejection mechanism employed by I-SEHP to ensure a set of 500 patterns with non-empty coverage, as illustrated in Fig. 3b, which shows the necessary number of rejections. Conversely, SEHP guarantees by design through step 1 a coverage of at least one object. We also note that I-SEHP on the test1 dataset results with 0 pattern drawn, which is due to the fact that the total sampling timeout, fixed to 20 minutes, was reached.

5.5 Diversity

Sampling diverse patterns is critical for the exploration of the solution space in an interactive mining process. We quantify this aspect by using an extension of the

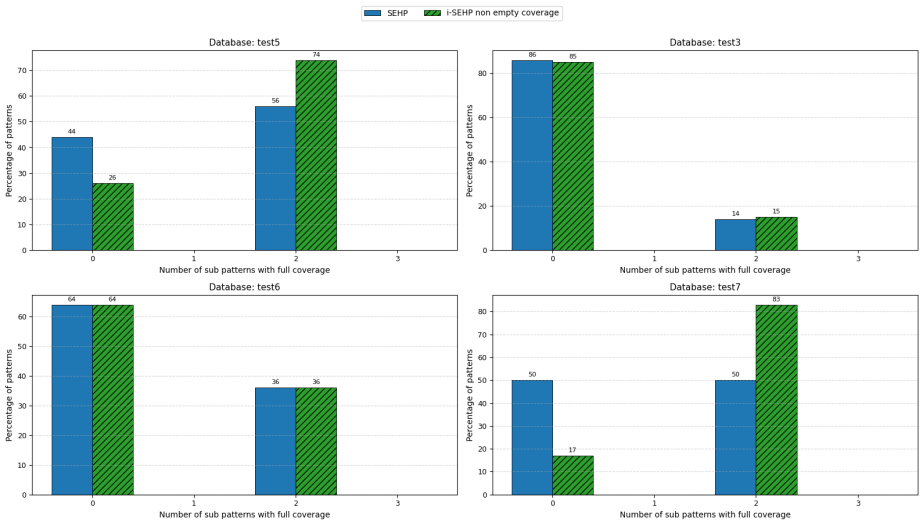

Fig. 2. The number of sub patterns with full coverage among the top 100 frequent sampled heterogeneous Patterns per dataset (SEHP vs. ISEHP)

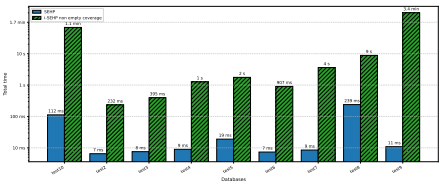

Fig3.a Cumulative CPU time required by SEHP and I-SEHP for sampling 500 patterns across all datasets.

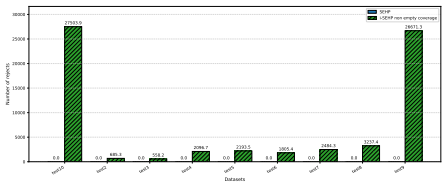

Fig3.b Number of rejections performed by I-SEHP to sample 500 patterns on each dataset.

diversity metric introduced in [11] for heterogeneous patterns. $\text{Diversity}(K, \mathcal{H}) = \frac{|\{\text{cover}(X_{\text{H}}^1, \mathcal{H}), \ldots, \text{cover}(X_{\text{H}}^{|K|}, \mathcal{H})\}|}{|K|}$, where $|K|$ is the total number of sampled patterns, and for each $i \in \{1, \ldots, |K|\}$, $\text{cover}(X_{\text{H}}^i, \mathcal{H})$ is the cover of pattern X_{H}^i in $\mathcal{H}$. Figure 4a reports the diversity obtained over 50,000 patterns. As I-SEHP samples more frequent patterns than SEHP, it leads to a more varied set of covers and, consequently, higher diversity. While both methods achieve a moderate level of diversity across most datasets , they show notably low values for test7 and test9. This is attributed to the characteristics of these datasets, which exhibit a low number of distinct values alongside longer sequence lengths, thereby forcing the sampling of highly similar patterns. We complete the diversity assessment by calculating the mean pairwise Jaccard distance between the covers of the $|K|$ sampled patterns.

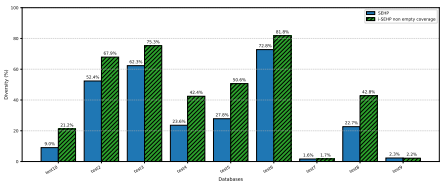

Fig4.a Diversity of 50,000 sampled patterns across all datasets.

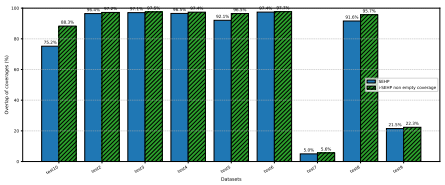

Fig4.b Mean overlap evaluation (SEHP vs. I-SEHP; 50,000 sampled patterns).

$$\text{Diversity}_J(K,\mathcal{H}) = \frac{2}{|K|(|K|-1)} \sum_{1\le i<j\le |K|} \left(1 - \frac{|\ \text{cover}(X^i_{\text{H}},\mathcal{H})\ \cap\ \text{cover}(X^j_{\text{H}},\mathcal{H})\ |}{|\ \text{cover}(X^i_{\text{H}},\mathcal{H})\ \cup\ \text{cover}(X^j_{\text{H}},\mathcal{H})\ |}\right).$$

Fig. 4b reports results over 50,000 sampled patterns, showing that the performances of I-SEHP and SEHP are nearly identical across all datasets, although low values are observed for test7 and test9, confirming that highly similar patterns are drawn from these specific datasets. This low diversity for SEHP is explained by the fact that the same object with a high NHP value is frequently drawn in step 1, leading to similar pattern coverage.

6 Conclusion and Perspectives

In this paper, we introduced SEHP, the first sampling method to tackle heterogeneous patterns. We formally proved that SEHP samples patterns proportionally to their frequency. SEHP instantly returns heterogeneous patterns, enabling tight coupling between the system and the user and improving knowledge discovery. As future work, we plan to address alternative interestingness measures tailored specifically to heterogeneous patterns and to enrich SEHP with constraints to combine the advantages of the worlds of sampling and constraint-based pattern mining [1]. Another direction is to evaluate the usefulness of sampled patterns for designing supervised or unsupervised models. In the context of classification, as discussed in [7], sampled patterns can be used to create features that add information to the original data. It is hoped that classification models built on data enriched by these new features will perform better than models trained on the original data. Finally, we also aim to integrate SEHP into applications implementing interactive pattern mining processes.

Acknowledgement. This work was supported by the French National Research Agency (ANR) through the project FIDD (grant agreement ANR-24-CE23-0711). The second author was supported by the French National Research Agency (ANR) and the Normandie Region through the HAISCoDe project (ANR-20-THIA-0021), and by the DeMik project under the UPS Excellences / Springboard Program (grant SPATTR_2025-12). The work of the fourth author is partly supported the Pandora project ANR-24-CE23-0950.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Aggarwal, C.C., Jiawei, H.: Frequent Pattern Mining. Springer Cham (2014)
2. Al Hasan, M., Zaki, M.J.: Output space sampling for graph patterns. Proceedings of the VLDB Endowment **2**(1), 730–741 (2009)
3. Bekkoucha, D., Diop, L., Ouali, A., Crémilleux, B., Boizumault, P.: Efficiently sampling interval patterns from numerical databases. Data & Knowledge Engineering p. 102566 (2026). https://doi.org/https://doi.org/10.1016/j.datak.2026.102566
4. Boley, M., Gärtner, T., Grosskreutz, H.: Formal concept sampling for counting and threshold-free local pattern mining. In: Proceedings of the 2010 SIAM International Conference on Data Mining. pp. 177–188. SIAM (2010). https://doi.org/10.1137/1.9781611972801.16
5. Boley, M., Lucchese, C., Paurat, D., Gärtner, T.: Direct local pattern sampling by efficient two-step random procedures. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 582–590 (2011). https://doi.org/10.1145/2020408.2020500
6. Codocedo, V., Napoli, A.: A proposition for combining pattern structures and relational concept analysis. In: International Conference on Formal Concept Analysis. pp. 96–111. Springer (2014). https://doi.org/10.1007/978-3-319-07248-7_8
7. Diop, L., Diop, C.T., Giacometti, A., Li, D., Soulet, A.: Sequential pattern sampling with norm-based utility. Knowledge and Information Systems **62**(5), 2029–2065 (2020). https://doi.org/10.1007/s10115-019-01417-3
8. Dzyuba, V., van Leeuwen, M., De Raedt, L.: Flexible constrained sampling with guarantees for pattern mining. Data Mining and Knowledge Discovery **31**(5), 1266–1293 (2017). https://doi.org/10.1007/s10618-017-0501-6
9. Egho, E., Raïssi, C., Calders, T., Jay, N., Napoli, A.: On measuring similarity for sequences of itemsets. Data Mining and Knowledge Discovery **29**(3), 732–764 (2015). https://doi.org/10.1007/s10618-014-0362-1
10. Egho, E., Raïssi, C., Jay, N., Napoli, A.: Mining heterogeneous multidimensional sequential patterns. In: ECAI 2014, pp. 279–284. IOS Press (2014). https://doi.org/10.3233/978-1-61499-419-0-279
11. Giacometti, A., Soulet, A.: Dense neighborhood pattern sampling in numerical data. In: Proceedings of the 2018 SIAM International Conference on Data Mining. pp. 756–764. SIAM (2018). https://doi.org/10.1137/1.9781611975321.85
12. Moens, S., Boley, M.: Instant exceptional model mining using weighted controlled pattern sampling. In: International Symposium on Intelligent Data Analysis. pp. 203–214. Springer (2014). https://doi.org/10.1007/978-3-319-12571-8_18
13. Pinto, H., Han, J., Pei, J., Wang, K., Chen, Q., Dayal, U.: Multi-dimensional sequential pattern mining. In: Proceedings of the tenth international conference on Information and knowledge management. pp. 81–88 (2001). https://doi.org/10.1145/502585.502600
14. Plantevit, M., Laurent, A., Laurent, D., Teisseire, M., Choong, Y.W.: Mining multidimensional and multilevel sequential patterns. ACM Transactions on Knowledge Discovery from Data (TKDD) **4**(1), 1–37 (2010). https://doi.org/10.1145/1644873.1644877