Gibbs sampling subjectively interesting tiles

Anes Bendimerad[1], Jefrey Lijffijt[2], Marc Plantevit[3], Céline Robardet[1], and Tijl De Bie[2]

[1] Univ Lyon, INSA, CNRS UMR 5205, F-69621 France
[2] IDLab, ELIS department, Ghent University, Ghent, Belgium
[3] Univ Lyon, UCBL, CNRS UMR 5205, F-69621 France

Abstract. The local pattern mining literature has long struggled with the so-called pattern explosion problem: the size of the set of patterns found exceeds the size of the original data. This causes computational problems (enumerating a large set of patterns will inevitably take a substantial amount of time) as well as problems for interpretation and usability (trawling through a large set of patterns is often impractical).
Two complementary research lines aim to address this problem. The first aims to develop better measures of interestingness, in order to reduce the number of uninteresting patterns that are returned [6, 10]. The second aims to avoid an exhaustive enumeration of all 'interesting' patterns (where interestingness is quantified in a more traditional way, e.g. frequency), by directly sampling from this set in a way that more 'interesting' patterns are sampled with higher probability [2].
Unfortunately, the first research line does not reduce computational cost, while the second may miss out on the most interesting patterns. In this paper, we combine the best of both worlds for mining interesting tiles [8] from binary databases. Specifically, we propose a new pattern sampling approach based on Gibbs sampling, where the probability of sampling a pattern is proportional to their subjective interestingness [6]—an interestingness measure reported to better represent true interestingness.
The experimental evaluation confirms the theory, but also reveals an important weakness of the proposed approach which we speculate is shared with any other pattern sampling approach. We thus conclude with a broader discussion of this issue, and a forward look.

Keywords: Pattern Mining · Subjective Interestingness · Pattern Sampling · Gibbs Sampling.

1 Introduction

Pattern mining methods aim to select elements from a given language that bring to the user "implicit, previously unknown, and potentially useful information from data" [7]. To meet the challenge of selecting the appropriate patterns for a user, several lines of work have been explored: (1) Many constraints on some measures that assess the quality of a pattern using exclusively the data have been designed [4, 12, 13]; (2) Preference measures have been considered to only

retrieve patterns that are non dominated in the dataset; (3) Active learning systems have been proposed that interact with the user to explicit her interest on the patterns and guide the exploration toward those she is interested in; (4) Subjective interestingness measures [6, 10] have been introduced that aim to take into account the implicit knowledge of a user by modeling her prior knowledge and retrieving the patterns that are unlikely according to the background model.

The shift from threshold-constraints on objective measures toward the use of subjective measures provides an elegant solution to the so-called pattern explosion problem by considerably reducing the output to only truly interesting patterns. Unfortunately, the discovery of subjectively interesting patterns with exact algorithms remains computationally challenging.

In this paper we explore another strategy that is pattern sampling. The aim is to reduce the computational cost while identifying the most important patterns, and allowing for distributed computations. There are two families of local pattern sampling techniques.

The first family uses Metropolis Hastings [9], a Markov Chain Monte Carlo (MCMC) method. It performs a random walk over a transition graph representing the probability of reaching a pattern given the current one. This can be done with the guarantee that the distribution of the considered quality measure is proportional on the sample set to the one of the whole pattern set [1]. However, each iteration of the random walk is accepted only with a probability equal to the acceptance rate α. This can be very small, which may result in a prohibitively slow convergence rate. Moreover, in each iteration the part of the transition graph representing the probability of reaching patterns given the current one, has to be materialized in both directions, further raising the computational cost. Other approaches [5, 11] relax this constraint but lose the guarantee.

Methods in the second family are referred to as direct pattern sampling approaches [2, 3]. A notable example is [2], where a two-step procedure is proposed that samples frequent itemsets without simulating stochastic processes. In a first step, it randomly selects a row according to a first distribution, and from this row, draws a subset of items according to another distribution. The combination of both steps follows the desired distribution. Generalizing this approach to other pattern domains and quality measures appeared to be difficult.

In this paper, we propose a new pattern sampling approach based on Gibbs sampling, where the probability of sampling a pattern is proportional to their Subjective Interestingness (SI) [6]. Gibbs sampling – described in Sec. 3 – is a special case of Metropolis Hastings where the acceptance rate α is always equal to 1. In Sec. 4, we show how the random walk can be simulated without materializing any part of the transition graph, except the currently sampled pattern. While we present this approach particularly for mining tiles in rectangular databases, applying it for other pattern languages can be relatively easily achieved. The experimental evaluation (Sec. 5) confirms the theory, but also reveals a weakness of the proposed approach which we speculate is shared by other direct pattern sampling approaches. We thus conclude with a broader discussion of this issue (Sec. 6), and a forward look (Sec. 7).

2 Problem formulation

2.1 Notation

Input dataset. A dataset $\mathbf{D}$ is a Boolean matrix with m rows and n columns. For $i \in [\![1, m]\!]$ and $j \in [\![1, n]\!]$, $\mathbf{D}(i, j) \in \{0, 1\}$ denotes the value of the cell corresponding to the i-th row and the j-th column. For a given set of rows $I \subseteq [\![1, m]\!]$, we define the support function $supp_C(I)$ that gives all the columns having a value of 1 in all the rows of I, i.e., $supp_C(I) = \{j \in [\![1, n]\!] \mid \forall i \in I : D(i, j) = 1\}$. Similarly, for a set of columns $J \subseteq [\![1, n]\!]$, we define the function $supp_R(J) = \{i \in [\![1, m]\!] \mid \forall j \in J : D(i, j) = 1\}$. Table 1 shows a toy example of a Boolean matrix, where for $I = \{4, 5, 6\}$ we have that $supp_C(I) = \{2, 3, 4\}$.

Table 1. Example of a binary dataset $\mathbf{D}$.

#	**1**	**2**	**3**	**4**	**5**
1	0	1	0	1	0
2	0	1	1	0	0
3	1	0	1	0	1
4	0	1	1	1	0
5	1	1	1	1	1
6	0	1	1	1	0
7	0	1	1	1	1

Pattern language. This paper is concerned with a particular kind of pattern known as a tile [8], denoted $\tau = (I, J)$ and defined as an ordered pair of a set of rows $I \subseteq \{1, ..., m\}$ and a set of columns $J \subseteq \{1, ...n\}$. A tile τ is said to be contained (or present) in $\mathbf{D}$, denoted as $\tau \in \mathbf{D}$, iff $\mathbf{D}(i, j) = 1$ for all $i \in I$ and $j \in J$. The set of all tiles present in the dataset is denoted as T and is defined as: $T = \{(I, J) \mid I \subseteq \{1, ..., m\} \wedge J \subseteq \{1, ...n\} \wedge (I, J) \in \mathbf{D}\}$. In Table 1, the tile $\tau_1 = (\{4, 5, 6, 7\}, \{2, 3, 4\})$ is present in $\mathbf{D}$ ($\tau_1 \in T$), because each of its cells has a value of 1, but $\tau_2 = (\{1, 2\}, \{2, 3\})$ is not present ($\tau_2 \notin T$) since $\mathbf{D}(1, 3) = 0$.

2.2 The interestingness of a tile

In order to assess the quality of a tile τ, we use the framework of subjective interestingness SI proposed in [6]. We briefly recapitulate the definition of this measure for tiles, denoted $\mathrm{SI}(\tau)$ for a tile τ, and refer the reader to [6] for more details. $\mathrm{SI}(\tau)$ measures the quality of a tile τ as the ratio of its subjective information content $\mathrm{IC}(\tau)$ and its description length $\mathrm{DL}(\tau)$:

$$\mathrm{SI}(\tau) = \frac{\mathrm{IC}(\tau)}{\mathrm{DL}(\tau)}.$$

Tiles with large $\mathrm{SI}(\tau)$ thus compress subjective information in a short description. Before introducing IC and DL, we first describe the background model—an important component required to define the subjective information content IC.

Background model. The SI is subjective in a sense that it accounts for prior knowledge of the current data miner. A tile τ is informative for a particular user if this tile is somehow surprising for her, otherwise, it does not bring new information. The most natural way for formalizing this is to use a background distribution representing the data miner's prior expectations, and to compute the

probability $\Pr(\tau \in \mathbf{D})$ of this tile under this distribution. The smaller $\Pr(\tau \in \mathbf{D})$, the more information this pattern contains. Concretely, the background model consists of a value $\Pr(\mathbf{D}(i,j) = 1)$ associated to each cell $\mathbf{D}(i,j)$ of the dataset, and denoted p_{ij}. More precisely, p_{ij} is the probability that $\mathbf{D}(i,j) = 1$ under user prior beliefs. In [6], it is shown how to compute the background model and derive all the values p_{ij} corresponding to a given set of considered user priors. Based on this model, the probability of having a tile $\tau = (I, J)$ in $\mathbf{D}$ is:

$$\Pr(\tau \in \mathbf{D}) = \Pr\left(\bigwedge_{i \in I, j \in J} \mathbf{D}(i,j) = 1\right) = \prod_{i \in I, j \in J} p_{ij}.$$

Information Content IC. This measure aims to quantify the amount of information conveyed to a data miner when she is told about the presence of a tile in the dataset. It is defined for a tile $\tau = (I, J)$ as follows:

$$\mathrm{IC}(\tau) = -\log(\Pr(\tau \in \mathbf{D})) = \sum_{i \in I, j \in J} -\log(p_{ij}).$$

Thus, the smaller $\Pr(\tau \in \mathbf{D})$, the higher $\mathrm{IC}(\tau)$, and the more informative τ. Note that for $\tau_1, \tau_2 \in \mathbf{D}$: $\mathrm{IC}(\tau_1 \cup \tau_2) = \mathrm{IC}(\tau_1) + \mathrm{IC}(\tau_2) - \mathrm{IC}(\tau_1 \cap \tau_2)$.

Description Length DL. This function should quantify how difficult it is for a user to assimilate the pattern. The description length of a tile $\tau = (I, J)$ should thus depend on how many rows and columns it refers to: the larger are $|I|$ and $|J|$, the larger is the description length. Thus, $\mathrm{DL}(\tau)$ can be defined as:

$$\mathrm{DL}(\tau) = a + b \cdot (|I| + |J|),$$

where a and b are two constants that can be handled to give more or less importance to the contributions of $|I|$ and $|J|$ in the description length.

2.3 Problem statement

Given a Boolean dataset $\mathbf{D}$, the goal is to sample a tile τ from the set of all the tiles T present in $\mathbf{D}$, with a probability of sampling P_S proportional to $\mathrm{SI}(\tau)$, that is: $P_S(\tau) = \frac{\mathrm{SI}(\tau)}{\sum_{\tau' \in T} \mathrm{SI}(\tau')}$.

A naïve approach to sample a tile pattern according to this distribution is to generate the list $\{\tau_1, ..., \tau_N\}$ of all the tiles present in $\mathbf{D}$, sample $x \in [0, 1]$ uniformly at random, and return the tile τ_k with $\frac{\sum_{i=1}^{k-1} \mathrm{SI}(\tau_i)}{\sum_i \mathrm{SI}(\tau_i)} \leq x < \frac{\sum_{i=1}^{k} \mathrm{SI}(\tau_i)}{\sum_i \mathrm{SI}(\tau_i)}$. However, the goal behind using sampling approaches is to avoid materializing the pattern space which is generally huge. We want to sample without exhaustively enumerating the set of tiles. In [2], an efficient procedure is proposed to directly sample patterns according to some measures such as the frequency and the area. However, this procedure is limited to only some specific measures. Furthermore,

it is proposed for pattern languages defined on only the column dimension, for example, itemset patterns. In such language, the rows related to an itemset pattern $F \subseteq \{1, ..., n\}$ are uniquely identified and they correspond to all the rows containing the itemset, that are $supp_R(F)$. In our work, we are interested in tiles which are defined by both columns and rows indices. In this case, it is not clear how the direct procedure proposed in [2] can be applied.

For more complex pattern languages, a generic procedure based on Metropolis Hasting algorithm has been proposed in [9], and illustrated for subgraph patterns with some quality measures. While this approach is generic and can be extended relatively easily to different mining tasks, a major drawback of using Metropolis Hasting algorithm is that the random walk procedure contains the acceptance test that needs to be processed in each iteration, and the acceptance rate α can be very small, which makes the convergence rate practically extremely slow. Furthermore, Metropolis Hasting can be computationally expensive, as the part of the transition graph representing the probability of reaching patterns given the current one, has to be materialized.

Interestingly, a very useful MCMC technique is Gibbs sampling, which is a special case of Metropolis-Hasting algorithm. A significant benefit of this approach is that the acceptante rate α is always equal to 1, i.e., the proposal of each sampling iteration is always accepted. In this work, we use Gibbs sampling to draw patterns with a probability distribution that converges to P_S. In what follows, we will first generically present the Gibbs sampling approach, and then we show how we efficiently exploit it for our problem. Unlike Metropolis Hasting, the proposed procedure performs a random walk by materializing in each iteration only the currently sampled pattern.

3 Gibbs sampling

Suppose we have a random variable $X = (X_1, X_2, ..., X_l)$ taking values in some domain Dom. We want to sample a value $x \in Dom$ following the joint distribution $P(X = x)$. Gibbs sampling is suitable when it is hard to sample directly from P but known how to sample just one dimension x_k ($k \in [\![1, l]\!]$) from the conditionnal probability $P(X_k = x_k \mid X_1 = x_1, ..., X_{k-1} = x_{k-1}, X_{k+1} = x_{k+1}, ..., X_l = x_l)$. The idea of Gibbs sampling is to generate samples by sweeping through each variable (or block of variables) to sample from its conditional distribution with the remaining variables fixed to their current values. Algorithm 1 depicts a generic Gibbs Sampler. At the beginning, x is set to its initial values (often values sampled from a prior distribution q). Then, the algorithm performs a random walk of p iterations. In each iteration, we sample $x_1 \sim P(X_1 = x_1^{(i_1)} \mid X_2 = x_2^{(i_1)}, ..., X_l = x_l^{(i_1)})$ (while fixing the other dimensions), then we follow the same procedure to sample x_2, ..., until x_l.

The random walk needs to satisfy some constraints to guarantee that the Gibbs sampling procedure converges to the stationary distribution P. In the case of a finite number of states (a finite space Dom in which X takes values), sufficient conditions for the convergence are irreducibility and aperiodicity:

Algorithm 1: Gibbs sampler

Initialize $x^{(0)} \sim q(x)$
for $k \in [\![1, p]\!]$ **do**
  draw $x_1^{(k)} \sim P\left(X_1 = x_1 \mid X_2 = x_2^{(k-1)}, X_3 = x_3^{(k-1)}, ..., X_l = x_l^{(k-1)}\right)$
  draw $x_2^{(k)} \sim P\left(X_2 = x_2 \mid X_1 = x_1^{(k)}, X_3 = x_3^{(k-1)}, ..., X_l = x_l^{(k-1)}\right)$
  ...
  draw $x_l^{(k)} \sim P\left(X_l = x_l \mid X_1 = x_1^{(k)}, X_2 = x_2^{(k)}, ..., X_{l-1} = x_{l-1}^{(k)}\right)$
return $x^{(p)}$

Irreducibility. A random walk is irreducible if, for any two states $x, y \in Dom$ s.t. $P(x) > 0$ and $P(y) > 0$, we can get from x to y with a probability > 0 in a finite number of steps. I.e. the entire state space is reachable.

Aperiodicity. A random walk is aperiodic if we can return to any state $x \in Dom$ at any time. I.e. revisiting x is not conditioned to some periodicity constraint.

One can also use blocked Gibbs sampling. This consists in growing many variables together and sample from their joint distribution conditioned to the remaining variables, rather than sampling each variable x_i individually. Blocked Gibbs sampling can reduce the problem of slow mixing that can be due to the high number of dimensions used to sample from.

4 Gibbs sampling of tiles with respect to SI

In order to sample a tile $\tau = (I, J)$ with a probability proportional to $\mathrm{SI}(\tau)$, we propose to use Gibbs sampling. The simplest solution is to consider a tile τ as $m + n$ binary random variables $(x_1, ..., x_m, ..., x_{m+n})$, each of them corresponds to a row or a column, and then apply the procedure described in Algorithms 1. In this case, an iteration of Gibbs sampling requires to sample from each column and row separately while fixing all the remaining rows and columns. The drawback of this approach is the high number of variables $(m + n)$ which may lead to a slow mixing time. In order to reduce the number of variables, we propose to split $\tau = (I, J)$ into only two separated blocks of random variables I and J, we then directly sample from each block while fixing the value of the other block. This means that an iteration of the random walk contains only two sampling operations instead of $m+n$ ones. We will explain in more details how this Blocked Gibbs sampling approach can be applied, and how to compute the distributions used to directly sample a block of rows or columns.

Algorithm 2 depicts the main steps of Blocked Gibbs sampling for tiles. We start by initializing $(I, J)^{(0)}$ with a distribution q proportional to the area $(|I| \times |J|)$ following the approach proposed in [2]. This choice is mainly motivated by its linear time complexity of sampling. Then, we need to efficiently sample from $P(\mathbf{I} = I \mid \mathbf{J} = J)$ and $P(\mathbf{J} = J \mid \mathbf{I} = I)$. In the following, we will explain

Algorithm 2: Gibbs-SI

Initialize $(I, J)^{(0)} \sim q(x)$
for $k \in [\![1, p]\!]$ **do**
  draw $I^{(k)} \sim P\left(\mathbf{I} = I \mid \mathbf{J} = J^{(k-1)}\right)$, draw $J^{(k)} \sim P\left(\mathbf{J} = J \mid \mathbf{I} = I^{(k)}\right)$
return $(I, J)^{(p)}$

how to sample I with $P(\mathbf{I} = I|\mathbf{J} = J)$, and since the SI is symmetric w.r.t. rows and columns, the same strategy can be used symetrically to sample a set of columns with $P(\mathbf{J} = J \mid \mathbf{I} = I)$.

Sampling a set of rows I conditionned to columns J. For a specific $J \subseteq \{1, ..., n\}$, the number of tiles (I, J) present in the dataset can be huge, and can go up to 2^m. This means that naïvely generating all these candidate tiles and then sampling from them is not a solution. Thus, to sample a set of rows I conditionned to a fixed set of columns J, we propose an iterative algorithm that builds the sampled I by drawing each $i \in I$ separately, while ensuring that the joint distribution of all the drawings is equal to $P(\mathbf{I} = I|\mathbf{J} = J)$. I is built using two variables: $R_1 \subseteq \{1, ..., m\}$ made of rows that belong to I, and $R_2 \subseteq \{1, ..., m\} \setminus R_1$ that contains candidate rows that can possibly be sampled and added to R_1. Initially, we have $R_1 = \emptyset$ and $R_2 = supp_R(J)$. At each step, we take $i \in R_2$, do a random draw to determine whether i is added to R_1 or not, and remove it from R_2. When $R_2 = \emptyset$, the sampled set of rows I is set equal to R_1. To apply this strategy, all we need is to compute $P\left(i \in \mathbf{I} \mid R_1 \subseteq \mathbf{I} \subseteq R_1 \cup R_2 \wedge \mathbf{J} = J\right)$, the probability of sampling i considering the current sets R_1, R_2 and J:

$$
\begin{aligned}
P\left(i \in \mathbf{I} \mid R_1 \subseteq \mathbf{I} \subseteq R_1 \cup R_2 \wedge \mathbf{J} = J\right) &= \frac{P\left(R_1 \cup \{i\} \subseteq \mathbf{I} \subseteq R_1 \cup R_2 \wedge \mathbf{J} = J\right)}{P\left(R_1 \cup \subseteq \mathbf{I} \subseteq R_1 \cup R_2 \wedge \mathbf{J} = J\right)} \\
&= \frac{\sum_{F \subseteq R_2 \setminus \{i\}} SI(R_1 \cup \{i\} \cup F, J)}{\sum_{F \subseteq R_2} SI(R_1 \cup F, J)} = \frac{\sum_{F \subseteq R_2 \setminus \{i\}} \frac{\mathrm{IC}_{(R_1 \cup \{i\}, J)} + \mathrm{IC}_{(F,J)}}{a + b \cdot (|R_1| + |F| + 1 + |J|)}}{\sum_{F \subseteq R_2} \frac{\mathrm{IC}_{(R_1, D_i)} + \mathrm{IC}_{(F, D_i)}}{a + b \cdot (|R_1| + |F| + |J|)}} \\
&= \frac{\sum_{k=0}^{|R_2|-1} \frac{1}{a + b \cdot (|R_1| + k + 1 + |J|)} \sum_{\substack{F \subseteq R_2 \setminus \{i\} \\ |F| = k}} (\mathrm{IC}(R_1 \cup \{i\}, J) + \mathrm{IC}(F, J))}{\sum_{k=0}^{|R_2|} \frac{1}{a + b \cdot (|R_1| + k + |J|)} \sum_{\substack{F \subseteq R_2 \\ |F| = k}} (\mathrm{IC}(R_1, J) + \mathrm{IC}(F, J))} \\
&= \frac{\sum_{k=0}^{|R_2|-1} \frac{1}{a + b \cdot (|R_1| + k + 1 + |J|)} \left(\binom{|R_2|-1}{k} \cdot \mathrm{IC}(R_1 \cup \{i\}, J) + \binom{|R_2|-2}{k-1} \cdot \mathrm{IC}(R_2 \setminus \{i\}, J) \right)}{\sum_{k=0}^{|R_2|} \frac{1}{a + b \cdot (|R_1| + k + |J|)} \left(\binom{|R_2|}{k} \cdot \mathrm{IC}(R_1, J) + \binom{|R_2|-1}{k-1} \cdot \mathrm{IC}(R_2, J) \right)} \\
&= \frac{\mathrm{IC}(R_1 \cup \{i\}, J) \cdot f(|R_2| - 1, |R_1| + 1) + \mathrm{IC}(R_2 \setminus \{i\}, J) \cdot f(|R_2| - 2, |R_1| + 1)}{\mathrm{IC}(R_1, J) \cdot f(|R_2|, |R_1|) + \mathrm{IC}(R_2, J) \cdot f(|R_2| - 1, |R_1|)},
\end{aligned}
$$

with $f(x, y) = \sum_{k=0}^{x} \frac{\binom{x}{k}}{a + b \cdot (y + k + |J|)}$.

Complexity. Let's compute the complexity of sampling I with a probability $P(\mathbf{I} = I|\mathbf{J} = J)$. Before starting the sampling of rows from R_2, we first compute the value of $\mathrm{IC}(\{i\}, J)$ for each $i \in R_2$ (in $\mathcal{O}(n \cdot m)$). This will allow to compute in $\mathcal{O}(1)$ the values of IC that appear in $P\left(i \in \mathbf{I} \mid R_1 \subseteq \mathbf{I} \subseteq R_1 \cup R_2 \wedge \mathbf{J} = J\right)$, based on the relation $\mathrm{IC}(I_1 \cup I_2, J) = \mathrm{IC}(I_1, J) + \mathrm{IC}(I_2, J)$ for $I_1, I_2 \subseteq [\![1, m]\!]$. In addition to that, sampling each element $i \in R_2$ requires to compute the corresponding values of $f(x, y)$. These values are computed once for the first sampled row $i \in R_2$ with a cost of $\mathcal{O}(m)$, and then they can be updated directly when sampling the next rows, using the following relation:

$$f(x-1, y) = f(x, y) - \frac{1}{a + b \cdot (x + y + |J|)} \cdot f(x-1, y+1).$$

This means that the overall cost of sampling the whole set of rows I with a probability $P(\mathbf{I} = I|\mathbf{J} = J)$ is $\mathcal{O}(n \cdot m)$. Following the same approach, sampling J conditionned to I is done in $\mathcal{O}(n \cdot m)$. As we have p sampling iterations, the worst case complexity of the whole Gibbs sampling procedure of a tile τ is $\mathcal{O}\left(p \cdot n \cdot m\right)$.

Convergence guarantee. In order to guarantee the convergence to the stationary distribution proportional to the SI measure, the Gibbs sampling procedure needs to satisfy some constraints. In our case, the sampling space is finite, as the number of tiles is limited to at most 2^{m+n}. Then, the sampling procedure converges if it satisfies the aperiodicity and the irreducibility constraints. The Gibbs sampling for tiles is indeed aperiodic, as in each iteration it is possible to remain in exactly the same state. We only have to verify if the irreducibility property is satisfied. We can show that, in some cases, the random walk is reducible, we will show how to make Gibbs sampling irreducible in those cases.

Theorem 1. *Let us consider the bipartite graph $G = (U, V, E)$ derived from the dataset $\boldsymbol{D}$, s.t., $U = \{1, .., m\}$, $V = \{1, ..., n\}$, and $E = \{(i, j) \mid i \in [\![1, m]\!] \wedge j \in [\![1, n]\!] \wedge \boldsymbol{D}(i, j) = 1\}$. A tile $\tau = (I, J)$ present in $\boldsymbol{D}$ corresponds to a complete bipartite subgraph $G_\tau = (I, J, E_\tau)$ of G. If the bipartite graph G is connected, then the Gibbs sampling procedure on tiles of $\boldsymbol{D}$ is irreducible.*

Proof. We need to prove that for all pair of tiles $\tau_1 = (I_1, J_1), \tau_2 = (I_2, J_2)$ present in $\mathbf{D}$, the Gibbs sampling procedure can go from τ_1 to τ_2. Let G_{τ_1}, G_{τ_2} be the complete bipartite graphs corresponding to τ_1 and τ_2. As G is connected, there is a path from any vertex of G_{τ_1} to any vertex of G_{τ_2}. The probability that the sampling procedure walks through one of these paths is not 0, as each step of these paths constitutes a tile present in $\mathbf{D}$. After walking on one of these paths, the procedure will find itself on a tile $\tau' \subseteq \tau_2$. Reaching τ_2 from τ' is probable after one iteration by sampling the right rows and then the right columns.

Thus, if the bipartite graph G is connected, the Gibbs sampling procedure converges to a stationary distribution. To make the random walk converge when G is not connected, we can compute the connected components of G, and then apply Gibbs sampling separately in each corresponding subset of the dataset.

Table 2. Dataset characteristics.

dataset	# rows	# columns	avg. |row|
mushrooms	8124	120	24
chess	3196	76	38
kdd	843	6159	65.3

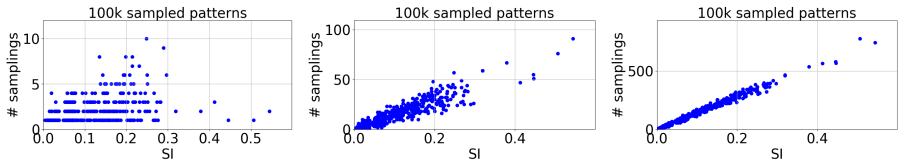

Fig. 1. Distribution of sampled patterns in synthetic data with 10 rows and 10 columns.

5 Experiments

We report our experimental study to evaluate the effectiveness of Gibbs-SI. Java source code is made available[4]. We consider three datasets whose characteristics are given in Table 2. *mushrooms* and *chess* from the UCI repository[5] are commonly used for evaluation purposes. *kdd* contains a set of SIGKDD paper abstracts between 2001 and 2008 downloaded from the ACM website. Each abstract is represented by a row and words correspond to columns, after stop word removal and stemming. For each dataset, the user priors that we represent in the SI background model are the row and column margins. In other terms, we consider that user knows (or, is already informed about) the following statistics: $\sum_j D(i,j)$ for all $i \in I$, and $\sum_i D(i,j)$ for all $j \in J$.

Empirical sampling distribution. First, we want to experimentally evaluate how the Gibbs sampling distribution matches with the desired distribution. We need to run Gibbs-SI in small datasets where the size of T is not huge. Then, we take a sufficiently large number of samples so that the sampling distribution can be created. To this aim, we have synthetically generated a dataset containing 10 rows, 10 columns, and 855 tiles. We run Gibbs-SI with three different numbers of iterations p: $1k$, $10k$, and $100k$, for each case, we keep all the visited tiles, and we study their distribution w.r.t. their SI values. Figure 1 reports the results. For $1k$ sampled patterns, the proportionality between the number of sampling and SI is not clearly established yet. For higher numbers of sampled patterns, a linear relation between the two axis is evident, especially for the case of $100k$ sampled patterns, which represents aroud 100 times the total number of all the tiles in the dataset. The two tiles with the highest SI are sampled the most, and the number of sampling clearly decreases with the SI value.

Characteristics of sampled tiles. To investigate which kind of patterns are sampled by Gibbs-SI, we show in Figure 2 the distribution of sampled tiles w.r.t

[4] http://tiny.cc/g5zmgz [5] https://archive.ics.uci.edu/ml/

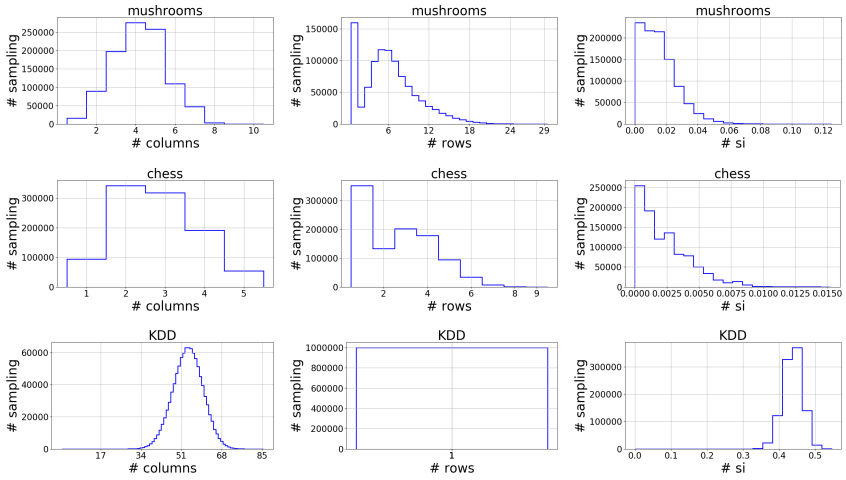

Fig. 2. Distributions of the sampled patterns w.r.t. # rows, # columns and SI.

their number of rows, columns, and their SI, for each of the three datasets given in Table 2. For *mushrooms* and *chess*, Gibbs-SI is able to return patterns with a diverse number of rows and columns. It samples much more patterns with low SI than patterns with high SI values. In fact, even if we are sampling proportionally to SI, the number of tiles in T with poor quality are significantly higher than the ones with high quality values. Thus, the probability of sampling one of low quality patterns is higher than sampling one of the few high quality patterns. For *kdd*, although the number of columns in sampled tiles varies, all the sampled tiles unfortunately cover only one row. In fact, the particularity of this dataset is the existence of some very large transactions (max=180).

Quality of the sampled tiles. In this part of the experiment, we want to study whether the quality of the top sampled tiles is sufficient. As mining exhaustively the best tiles w.r.t. SI is not feasible, we need to find some strategy that identifies high quality tiles. We propose to use LCM [14] to retrieve the closed tiles corresponding to the top 10k frequent closed itemsets. A closed tile $\tau = (I, J)$ is a tile that is present in $\mathbf{D}$ and whose I and J cannot be extended anymore. Although closed tiles are not necessarily the ones with the highest SI, we make the hypothesis that at least some of them have high SI values as they maximize the value of IC function. For each of the the three real world datasets, we compare between the SI of the top closed tiles identified with LCM and the ones identified with Gibbs-SI. In Table 3, we show the SI of the top-1 tile, and the average SI of the top-10 tiles, for each of LCM and Gibbs-SI.

Unfortunately, the scores of tiles retrieved with LCM are substantially larger than the ones of Gibbs-SI, especially for *mushrooms* and *chess*. Importantly, there may exist tiles that are even better than the ones found by LCM. This means that Gibbs-SI fails to identify the top tiles in the dataset. We believe

Table 3. The SI of the top-1 tile, and the average SI of the top-10 tiles, found by LCM and Gibbs-SI in the studied datasets.

	mushrooms		chess		KDD	
	top 1 SI	avg(top 10 SI)	top 1 SI	avg(top 10 SI)	top 1 SI	avg(top 10 SI)
Gibbs sampling	0.12	0.11	0.015	0.014	0.54	0.54
LCM	3.89	3.20	0.40	0.40	0.83	0.70

that this is due to the very large number of low quality tiles which trumps the number of high quality tiles. The probability of sampling a high-quality tile is exceedingly small, necessitating a practically too large sample to identify any.

6 Discussion

Our results show that efficiently sampling from the set of tiles with a sampling probability proportional to the tiles' subjective interestingness is possible. Yet, they also show that if the purpose is to identify some of the most interesting patterns, direct pattern sampling may not be a good strategy. The reason is that the number of tiles with low subjective interestingness is vastly larger that those with high subjective interestingness. This imbalance is not sufficiently offset by the relative differences in their interestingness and thus in their sampling probability. As a result, the number of tiles that need to be sampled in order to sample one of the few top interesting ones is of the same order as the total number of tiles.

To mitigate this, one could attempt to sample from alternative distributions that attribute an even higher probability to the most interesting patterns, e.g. with probabilities proportional to the *square* or other high powers of the subjective interestingness. We speculate, however, that the computational cost of sampling from such more highly peaked distributions will also be larger, undoing the benefit of needing to sample fewer of them. This intuition is supported by the fact that direct sampling schemes according to itemset support are computationally cheaper than according to the square of their support [2].

That said, the use of sampled patterns as features for downstream machine learning tasks, even if these samples do not include the most interesting ones, may still be effective as an alternative to exhaustive pattern mining.

7 Conclusions

Pattern sampling has been proposed as a computationally efficient alternative to exhaustive pattern mining. Yet, existing techniques have been limited in terms of which interestingness measures they could handle efficiently.

In this paper, we introduced an approach based on Gibbs sampling, which is capable of sampling from the set of tiles proportional to their subjective interestingness. Although we present this approach for a specific type of pattern language and quality measure, we can relatively easily follow the same scheme to

apply Gibbs sampling for other pattern mining settings. The empirical evaluation demonstrates effectiveness, yet, it also reveals a potential weakness inherent to pattern sampling: when the number of interesting patterns is vastly outnumbered by the number of non-interesting ones, a large number of samples may be required, even if the samples are drawn with a probability proportional to the interestingness. Investigating our conjecture that this problem affects all approaches for sampling interesting patterns (for sensible measures of interestingness) seems a fruitful avenue for further research.

Acknowledgements. This work was supported by the ERC under the EU's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517, the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, the FWO (project no. G091017N, G0F9816N, 3G042220), and the EU's Horizon 2020 research and innovation programme and the FWO under the Marie Sklodowska-Curie Grant Agreement no. 665501, and by the ACADEMICS grant of the IDEXLYON, project of the Université of Lyon, PIA operated by ANR-16-IDEX-0005.

References

1. Boley, M., Gärtner, T., Grosskreutz, H.: Formal concept sampling for counting and threshold-free local pattern mining. In: Proc. of SDM. pp. 177–188 (2010)
2. Boley, M., Lucchese, C., Paurat, D., Gärtner, T.: Direct local pattern sampling by efficient two-step random procedures. In: Proc. of KDD. pp. 582–590 (2011)
3. Boley, M., Moens, S., Gärtner, T.: Linear space direct pattern sampling using coupling from the past. In: Proc. of KDD. pp. 69–77 (2012)
4. Boulicaut, J., Jeudy, B.: Constraint-based data mining. In: Data Mining and Knowledge Discovery Handbook, pp. 339–354. Springer (2010)
5. Chaoji, V., Hasan, M.A., Salem, S., Besson, J., Zaki, M.J.: ORIGAMI: a novel and effective approach for mining representative orthogonal graph patterns. SADM **1**(2), 67–84 (2008)
6. De Bie, T.: Maximum entropy models and subjective interestingness: an application to tiles in binary databases. DMKD **23**(3), 407–446 (2011)
7. Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C.J.: Knowledge discovery in databases: An overview. AI Mag. **13**(3), 57–70 (1992)
8. Geerts, F., Goethals, B., Mielikäinen, T.: Tiling databases. In: Proc. of DS. pp. 278–289 (2004)
9. Hasan, M.A., Zaki, M.J.: Output space sampling for graph patterns. PVLDB **2**(1), 730–741 (2009)
10. Kontonasios, K.N., Spyropoulou, E., De Bie, T.: Knowledge discovery interestingness measures based on unexpectedness. Wiley IR: DMKD **2**(5), 386–399 (2012)
11. Moens, S., Goethals, B.: Randomly sampling maximal itemsets. In: Proc. of KDD-IDEA. pp. 79–86 (2013)
12. Pei, J., Han, J., Wang, W.: Constraint-based sequential pattern mining: the pattern-growth methods. JIIS **28**(2), 133–160 (2007)
13. Raedt, L.D., Zimmermann, A.: Constraint-based pattern set mining. In: Proc. of SDM. pp. 237–248 (2007)
14. Uno, T., Asai, T., Uchida, Y., Arimura, H.: An efficient algorithm for enumerating closed patterns in transaction databases. In: Proc. of DS. pp. 16–31 (2004)