

Formal Concept Sampling for Counting and Threshold-Free Local Pattern Mining

Mario Boley and Thomas Gärtner and Henrik Grosskreutz
Fraunhofer IAIS
Schloss Birlinghoven
53754 Sankt Augustin, Germany
{mario.boleym, thomas.gaertner, henrik.grosskreutz}@iais.fraunhofer.de

Abstract

We describe a Metropolis-Hastings algorithm for sampling formal concepts, i.e., closed (item-) sets, according to any desired strictly positive distribution. Important applications are (a) estimating the number of all formal concepts as well as (b) discovering any number of interesting, non-redundant, and representative local patterns. Setting (a) can be used for estimating the runtime of algorithms examining all formal concepts. An application of setting (b) is the construction of data mining systems that do not require any user-specified threshold like minimum frequency or confidence.

1 Introduction

We describe a sampling algorithm that combines two directions of data mining research: the well established task of finding closed (item-)sets of a transactional database [3, 10, 24] and the recent trend to sample patterns instead of listing them exhaustively [13, 14, 23]. The advantage of considering closed patterns rather than all patterns is that they represent non-redundant information about the dataset. The advantage of sampling rather than listing is that it allows to efficiently generate exactly the desired number of patterns according to a controlled target distribution. Our algorithm has applications in data mining as well as formal concept analysis.

In data mining a set is called closed if each of its supersets is contained in less transactions than itself. A closed set can hence be seen as a (unique) maximal representer of the set of transactions that it is contained in. In terms of formal concept analysis, closed sets correspond to formal concepts, i.e., they are the columns of maximal all-1-rectangles of a given binary matrix. Focusing on closed sets avoids generating sets that represent the same transactions, i.e., it avoids generating redundant knowledge compared to traditional pattern mining approaches (see [3] and [10]).

The motivation for sampling instead of listing is that it is often infeasible to completely list all (frequent) closed patterns. In this scenario, aborted listing algorithms produce subsets $\mathcal{C}' \subseteq \mathcal{C}$ of the closed patterns \mathcal{C} depending on their internal search order that usually does not reflect any data mining interestingness measure. In contrast, with our sampling approach closed patterns can be generated efficiently according to some controlled target distribution $\pi : \mathcal{C} \rightarrow [0, 1]$. This distribution can be chosen freely according to some interestingness measure $q : \mathcal{C} \rightarrow \mathbb{R}$, i.e., $\pi(\cdot) = q(\cdot)/Z$ with a normalizing constant Z . This approach has the further advantage that it does not rely on a well-chosen value for a threshold parameter (like minimum support), which is often hard to determine in practice. In addition, when using the uniform target distribution, sampling can be used to count the number of closed patterns/formal concepts in polynomial time.

1.1 Outline and Contributions We discuss more detailed motivations and related work in the remainder of this section. Then, after having introduced basic notation and concepts (Section 2), we give our main results:

- For a given transactional database, we define a Markov chain that has the family of closed sets as its state space, allows an efficient computation of a single simulation step, and converges to any desired strictly positive target distribution (in Section 3).
- While the worst-case mixing time of these chains can grow exponentially in the database size, we propose a heuristic polynomially bounded function for the number of simulation steps. This heuristic results in sufficient closeness to the target distribution for a number of benchmark databases.
- We show how concept sampling can be used to build an approximation scheme for counting the

number of all concepts (in Section 4).

- Finally, we demonstrate how to use the sampling approach for generating any number of non-redundant local patterns that are representative for a given distribution reflecting interestingness—without the specification of any threshold parameters (in Section 5).

In a concluding Section we discuss, among other issues, complexity results suggesting that no worst-case polynomial time sampling algorithm for our problem exists.

1.2 Motivation A typical scenario when mining for local patterns of a real-world database is the following. While the complete set of patterns is overwhelmingly large, only a small fraction of these patterns is needed as input for subsequent analysis or modeling steps. For example consider the association rule based outlier detection method LERAD [19]—a two-pass algorithm that first generates a large set of rules and then filters the set until about 50 to 100 “good” rules are left. This two phase approach with a listing/generation step and a subsequent filtering step is used within a wide range of applications and has been formalized to the so-called Lego-framework [18]. Its first step, however, can be problematic, because the computation time of an exhaustive listing algorithm is at least proportional to the number of patterns it produces. Thus, whenever the number of unfiltered patterns is too large the whole approach becomes infeasible.

Dataset	#rows	#cols.	#attr.	size
Election	801	56	227	0.4MB
Questionnaire	11.188	346	1893	19.7MB
Census (30k)	30.000	68	395	8.1MB

Table 1: Databases used in motivational experiment.

We now demonstrate that these scenarios are not mere worst-case constructs but that they can appear in practice already on small to mid-sized databases. Our goal is to discover minimal non-redundant association rules (see Section 2 for precise definitions) in three socio-economic datasets that are summarized in Table 1. Two of them are from internal projects: *Election* contains the result of the 2004 council elections in the city of Cologne along with descriptive attributes for each of 800 polling districts; *Questionnaire* is the result of a socio-economic questionnaire with a total of 11.188 participants. We complement them by the publicly available “US Census 1990” dataset (*Census*) from the UCI machine learning repository [2]. In order to speedup our experiments we generated a sample of 30K rows of this last dataset. For

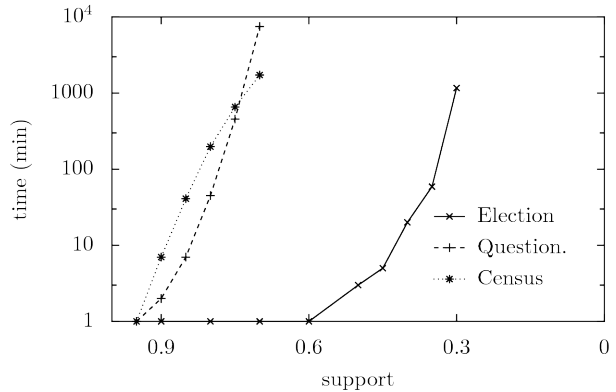


Figure 1: Computation time of exhaustively listing all minimal non-redundant association rules.

all three datasets we converted their nominal columns into binary attributes. For the sake of simplicity we are only interested in exact rules, i.e., rules with a confidence of 1. Figure 1 shows the required computation time for several minimum frequency thresholds using JClose [21] as a representative exhaustive method. Although there are more recent and potentially faster algorithms, all of them exhibit a similar behavior: the well-known exponential increase of the required time with decreasing thresholds. Note that this explosion occurs already very early on the frequency scale. For *Election* the method becomes problematic for thresholds lower than 0.35. The situation for the other two datasets is even worse. For *Questionnaire* the required time for a threshold as high as 0.7 is more than five days!

This demonstrates that even for moderately-sized datasets and moderate minimum support thresholds, finding association rules can be impossible. With a support threshold of 0.7 or higher, however, local pattern discovery becomes in fact *global* pattern discovery which is not desired in many applications [12]. There are even applications where one is interested exclusively in low support patterns (for instance see the study of Chow et al. on detecting privacy leaks using association rules [7]). Altogether, this motivates the development of algorithms that are capable of finding a designated small number of interesting patterns in feasible time. In particular, these patterns must not be restricted to high support patterns but should rather be selected according to application dependent quality measures.

1.3 Related Work and Challenges Exhaustive listing algorithms could be adapted to our task by (a) applying a post-processing step to select a subset of a given complete pattern collection that was previously

listed (see [1, 26] for representative methods), by (b) aborting the pattern enumeration after the desired number of patterns have been generated, or by (c) directly restricting the search to high quality patterns (e.g., [20, 27]). As discussed in Section 1.2, approach (a) can be infeasible, approach (b) usually leads to a non-representative set of patterns that depends on the internal search order of the listing scheme, and approach (c) usually suffers from computational complexity issues. For instance, finding k patterns of maximum frequency having a minimum length [24] or finding a pattern that is infrequent in one database but frequent in another [25] can neither exactly nor approximately be solved in polynomial time (unless $\mathbf{P}=\mathbf{NP}$). This is due to the hard thresholds respectively optimality requirements involved.

Sampling algorithms can circumvent these negative complexity results by replacing hard thresholds with guaranteeing that the probability of a pattern to appear in the output is proportional to its quality. As a positive side-effect, the user is freed of the often troublesome task of finding appropriate thresholds. Al Hasan et al. [13] were the first to propose to circumvent the listing bottleneck by using randomized sampling. Their algorithm aims to generate a representative list of maximal frequent subgraphs of a given graph database by a two-step approach: first a number of maximal frequent subgraphs is randomly generated, and then a representative subset of this sample is selected based on a similarity measure. The sampling phase of this algorithm, however, provides no control over the target distribution. In contrast, the algorithm from [14] essentially simulates a Markov chain with uniform stationary distribution. While the process described there is also used to sample maximal frequent subgraphs, its actual state space is the set of all frequent subgraphs. A similar chain has been used in [5] for the purpose of quickly counting the number of frequent sets without listing them.

There is no straightforward way of adapting these processes to sampling *closed* sets. Using a set sampler within a rejection sampling approach does not lead to an efficient algorithm because the number of sets can grow exponentially in the number of closed sets. Consequently, the probability of successfully drawing a closed set with rejection sampling can be very small; for a small database like “msweb” from the UCI machine learning repository [2] with 285 items and 32K transactions already as small as 10^{-80} . Similarly, mapping a drawn itemset to its smallest closed superset, is not a viable option: the varying number of generators among the closed set would introduce an uncontrollable bias. This motivates the construction of a random walk algorithm that directly uses the closed sets as state space.

2 Background

This section recalls definitions from formal concept analysis (FCA) and closed set mining as well as Markov chains. FCA and closed set mining are strongly related. In this work we prefer the notions of FCA because of its symmetric treatment of items and transactions, which simplifies the subsequent technical discussion. For more information on FCA we refer to Wille’s textbook [9], for Markov chains to Randall’s survey [22] and references therein.

2.1 Formal concept analysis Let X be some set. We denote the power set of X by $\mathcal{P}(X)$. A mapping $\sigma : \mathcal{P}(X) \rightarrow \mathcal{P}(X)$ is called a *closure operator* if it satisfies for all $A \subseteq B \subseteq X$ *extensivity*, i.e., $A \subseteq \sigma(A)$, *monotonicity*, i.e., $\sigma(A) \subseteq \sigma(B)$, and *idempotence*, i.e., $\sigma(A) = \sigma(\sigma(A))$. The set of all closed sets of σ , i.e., its fixpoints, is denoted by $\sigma(\mathcal{P}(X))$.

A (formal) context is a tuple (A, O, \mathcal{D}) with A and O finite sets referred to as *attributes* and *objects*, respectively, and $\mathcal{D} \subseteq A \times O$ a binary relation. In closed set mining, A , O , and \mathcal{D} are referred to as *items*, *transactions*, and *database*, respectively. The maps $O[\cdot] : \mathcal{P}(A) \rightarrow \mathcal{P}(O)$ and $A[\cdot] : \mathcal{P}(O) \rightarrow \mathcal{P}(A)$ are

$$O[X] = \{o \in O : \forall a \in X, (a, o) \in \mathcal{D}\} \text{ and}$$

$$A[Y] = \{a \in A : \forall o \in Y, (a, o) \in \mathcal{D}\} ,$$

respectively. An ordered pair $C = (I, E) \in \mathcal{P}(A) \times \mathcal{P}(O)$ is called a (formal) *concept*, denoted $\langle I, E \rangle$, if $O[I] = E$ and $A[E] = I$. The set I is called the *intent* of C and E is called its *extent*. The set of all concepts for a given context is denoted $\mathcal{C}(A, O, \mathcal{D})$ or just \mathcal{C} when there is no ambiguity. It is partially ordered by the binary relation \preceq defined by $\langle I, E \rangle \preceq \langle I', E' \rangle$ if and only if $I \subseteq I'$ (or equivalently $E \supseteq E'$). For the minimal respectively maximal elements of \mathcal{C} with respect to \preceq we write “ \perp ” for $\langle A[O], O \rangle$ and “ \top ” for $\langle A, O[A] \rangle$. It is well-known that

- a) the maps $A[\cdot]$ and $O[\cdot]$ form an (order-reversing) *Galois connection*, i.e.,

$$X \subseteq A[Y] \iff Y \subseteq O[X] ,$$

- b) their compositions $\phi = A[O[\cdot]]$ and $\psi = O[A[\cdot]]$ form closure operators on $\mathcal{P}(A)$ and $\mathcal{P}(O)$, respectively,
- c) and all concepts $C = \langle I, E \rangle \in \mathcal{C}$ are of the form $C = \langle \phi(X), A[\phi(X)] \rangle$ for some $X \subseteq A$ respectively $C = \langle O[\psi(Y)], \psi(Y) \rangle$ for some $Y \subseteq O$, i.e., the intents of all concepts are the closed sets of ϕ and the extents the closed sets of ψ .

2.2 Markov chains Monte Carlo Methods A (discrete) *Markov chain* on state space Ω is a sequence of discrete random variables $(X_t)_{t \in \mathbb{N}}$ with domain Ω satisfying the Markov condition, i.e., that $\mathbb{P}[X_{t+1} = x | X_1 = x_1, \dots, X_t = x_t]$ is equal to $\mathbb{P}[X_{t+1} = x | X_t = x_t]$ for all $t \in \mathbb{N}$ and $x, x_1, \dots, x_t \in \Omega$ satisfying $\mathbb{P}[X_1 = x_1, \dots, X_t = x_t] > 0$. The *uniform distribution* on Ω is denoted $u(\Omega)$. In this article we only consider time-homogeneous Markov chains on finite state spaces. Given a probability distribution on the initial state, the joined distribution of $(X_t)_{t \in \mathbb{N}}$ is completely specified by the *state transition probabilities* $p(x, y) = \mathbb{P}[X_{t+1} = y | X_t = x]$ of all $x, y \in \Omega$, which do not depend on t . Let $p^t(x, y)$ denote the t -step probability, i.e., $p^t(x, y) = \mathbb{P}[X_{n+t} = y | X_n = x]$. We call a state $y \in \Omega$ *reachable* from a state $x \in \Omega$ if there is a $t \in \mathbb{N}$ such that $p^t(x, y) > 0$. The chain $(X_t)_{t \in \mathbb{N}}$ is called *aperiodic* if for all $x, y \in \Omega$ with x reachable from y there is a $t_0 \in \mathbb{N}$ such that for all $t \geq t_0$ it holds that $p^t(x, y) > 0$, and it is called *irreducible* if any two states are reachable from one another. Finally, $(X_t)_{t \in \mathbb{N}}$ is called *ergodic* if it is irreducible and aperiodic. Any ergodic Markov chain has a unique *limiting stationary distribution* $\pi: \Omega \rightarrow [0, 1]$, i.e., for all states $x, y \in \Omega$ it holds that $\lim_{t \rightarrow \infty} p^t(x, y) = \pi(y)$. Moreover, if there is a distribution $\pi': \Omega \rightarrow [0, 1]$ satisfying the *detailed balance condition*

$$(2.1) \quad \forall x, y \in \Omega, \pi'(x)p(x, y) = \pi'(y)p(y, x)$$

and p is irreducible then π' is a stationary distribution and p is called *time-reversible*.

A Markov chain Monte Carlo method generates an element from Ω according to π by simulating $(X_t)_{t \in \mathbb{N}}$ for some time starting in an arbitrary state $X_0 = x_0 \in \Omega$. Clearly, for that approach to be effective one has to (a) be able to efficiently generate a “neighbor” $y \sim p(x, \cdot)$ for a given state x and (b) know after how many simulation steps t the distribution of X_t is “close enough” to π . The distance from the t -step distribution of a Markov chain with $X_0 = x$ to its stationary distribution can be measured by the *total variation distance* $\|p^t(x, \cdot), \pi\|_{\text{tv}} = 1/2 \sum_{y \in \Omega} |p^t(x, y) - \pi(y)|$. Using this definition the *mixing time* of $(X_t)_{t \in \mathbb{N}}$ is defined by

$$\tau(\epsilon) = \max_{x \in \Omega} \min \{t_0 \in \mathbb{N} : \forall t \geq t_0, \|p^t(x, \cdot), \pi\|_{\text{tv}} \leq \epsilon\}$$

as the minimum number of steps one has to simulate $(X_t)_{t \in \mathbb{N}}$ until the resulting distribution is guaranteed to be ϵ -close to its stationary distribution.

2.3 Closed Local Pattern Mining Finally, we fix notions and notations of closed local pattern mining and their evaluation metrics. This includes scores used in

unsupervised settings, e.g., association rule discovery, as well as scores used in supervised descriptive rule induction tasks such as emerging pattern mining [25], contrast set mining [4], or subgroup discovery [6]. For coherence we define all notions from the perspective of FCA.

Let (A, O, \mathcal{D}) be a context. We consider two types of local patterns: simple sets of attributes $F \subseteq A$, and association rules $X \rightarrow Y$ with $X, Y \subseteq A$. There are many measures in the literature for ranking patterns according to their interestingness.

The *support* of a set $F \subseteq A$ is the relative size of its extent, i.e., $q_{\text{supp}}(F) = |O[F]| / |O|$. A measure that is based on the minimum description length principle is the *area function* [11], i.e., $q_{\text{area}}(F) = |F| |O[F]|$. The *support* of a rule $X \rightarrow Y$ is defined as the support of $X \cup Y$, and its *confidence* is $q_{\text{conf}}(X \rightarrow Y) = |O[X \cup Y]| / |O[X]|$. We sometimes denote the confidence of a rule above its \rightarrow -symbol, i.e., $X \xrightarrow{c} Y$. Rules with a confidence of 1 are called implications or *exact rules*.

In order to define *supervised evaluation measures* one assumes that there are associated binary labels $l(o) \in \{+, -\}$ for all objects $o \in O$. Patterns $F \subseteq A$ can then be ranked according to the distributional unusualness of these labels on the pattern’s extent, respectively according to their support difference between the positive and the negative portion of the objects. A representative measure is the *binomial quality function*

$$q_{\text{bino}}(F) = q_{\text{supp}}(F) \left(\frac{|O^+[F]|}{|O[F]|} - \frac{|O^+|}{|O|} \right)$$

where (A, O^+, \mathcal{D}^+) denotes the sub-context containing only the objects with positive labels, i.e., $O^+ = \{o \in O : l(o) = +\}$ and $\mathcal{D}^+ = \{(a, o) \in \mathcal{D} : o \in O^+\}$.

For both kinds of evaluation measures—unsupervised and supervised—it has been shown that by considering only closed sets, i.e., sets $F \subseteq A$ with $F = \phi(F)$, one can focus on non-redundant rule families. In particular for association rules, a rule $X \xrightarrow{c} Y$ is called *minimal non-redundant* if there is no rule $X' \xrightarrow{c} Y'$ with $X' \subseteq X$, $Y' \supseteq Y$ and the same support. The following result of [3] relates closed sets to exact minimal non-redundant association rules.

PROPOSITION 1. (BASTIDE ET AL.) *All exact minimal non-redundant rules $X \rightarrow Y$ of a context (A, O, \mathcal{D}) are of the form $Y = \phi(Y)$ with X being a minimal set with $\phi(X) = Y$.*

3 Sampling Concepts

We are now ready to describe our concept sampling algorithm. First, we show how the state space, i.e.,

the concept lattice of a given context, can be connected in a way that allows to efficiently select a random element from the neighborhood of some concept. We achieve this by using the closure operators induced by the context. Then, in a second step, we apply to this initial idea the Metropolis-Hastings algorithm and show that the resulting Metropolis process is irreducible. It can therefore be used to sample a concept according to any desired strictly positive distribution.

3.1 Generating Elements In order to construct a stochastic process on the concept lattice of a given context we will exploit its associated closure systems. More specifically, one can “move” from one closed set of a closure operator to another by a single element augmentation and a subsequent closure operation. In this context, we refer to the involved single elements as *generating elements* that can be defined as follows.

DEFINITION 1. (GENERATING ELEMENT) *Let σ be a closure operator on $\mathcal{P}(X)$ and $C, C' \in \sigma(\mathcal{P}(X))$ be closed sets. We say that an element $x \in X$ generates C' from C with respect to σ if $\sigma(C \cup \{x\}) = C'$. The set of all such generating elements is denoted by $G_\sigma(C, C')$.*

These generating elements can be used to define a directed graph on the family of closed sets in which two vertices C, C' are joined by an arc if there is at least one element generating C' from C . This graph is implicitly traversed by several closed set listing algorithms.

DEFINITION 2. (GENERATOR GRAPH) *Let σ be a closure operator on $\mathcal{P}(X)$. The generator graph $\mathbf{G}_\sigma = (\mathbf{C}, \mathbf{E}_\sigma, \mathbf{l}_\sigma)$ of σ is the directed labeled graph on the closed sets $\mathbf{C} = \sigma(\mathcal{P}(X))$ as vertices with edges*

$$\mathbf{E}_\sigma = \{(C, C') \in \mathbf{C} \times \mathbf{C} : G_\sigma(C, C') \neq \emptyset\},$$

and with edge labels $\mathbf{l}_\sigma : \mathbf{E} \rightarrow \mathcal{P}(X)$ equal to the generating elements, i.e., $\mathbf{l}_\sigma(C, C') = G_\sigma(C, C')$.

It is easy to see that \mathbf{G}_ϕ is a) acyclic except for self-loops and b) rooted in $\sigma(\emptyset)$: statement a) follows by observing that $(C, C') \in \mathbf{E}_\sigma$ implies $C \subseteq C'$, and for b) note that for all closed sets $C = \{x_1, \dots, x_k\}$ the sequence x_1, \dots, x_k corresponds to an edge progression (walk) from $\sigma(\emptyset)$ to C due to extensivity and monotonicity of σ . A random walk on the generator graph can be performed by the following procedure: in a current closed set $Y \in \sigma(\mathcal{P}(X))$ draw an element $x \sim u(X)$, and then move to $\sigma(Y \cup \{x\})$. The transition probability from a set Y to a set Z is then directly proportional to the number of generating elements $|G_\sigma(Y, Z)|$ and can be computed efficiently.

Now, if one identifies concepts with their intents respectively with their extents, there are *two* associated

\mathcal{D}	a	b	c	d
1	0	1	1	0
2	1	0	1	0
3	1	1	0	0
4	1	1	0	1
5	0	0	0	1

Table 2: Example context with attributes $\{a, b, c, d\}$ and objects $\{1, 2, 3, 4, 5\}$.

generator graphs to a given context—the one induced by ϕ and the one induced by ψ . For the example context from Table 2 these graphs are drawn in Figure 2. Note that the arc relation \mathbf{G}_ϕ is generally not identical to the transitive reduction of \preceq , which is usually used within diagrams illustrating a concept lattice: the concepts $\langle a, 234 \rangle$ and $\langle abd, 4 \rangle$ are joined by an arc although they are not direct successors with respect to \preceq . The basic idea for our sampling algorithm is to perform a random walk on these generator graphs. Taking just one of them, however, does not result in an irreducible chain. Either \top or \perp would be an absorbing state for such a process, in which any random walk will result with a probability converging to 1 for an increasing number of steps.

A first idea to achieve irreducibility might be to take one of the corresponding generator graphs, say \mathbf{G}_ϕ , add all inverse edges of \mathbf{E}_ϕ as additional possible state transitions, and perform a random walk on the resulting strongly connected graph. Unfortunately, with this approach, drawing a neighbor of a given concept (as required for an efficient chain simulation) is as hard as the general problem of sampling a concept.

PROPOSITION 2. *Given a context (A, O, \mathcal{D}) and a concept $\langle I, E \rangle \in \mathcal{C}$, drawing a predecessor of $\langle I, E \rangle$ in \mathbf{G}_ϕ uniformly at random, i.e., a concept $\langle I', E' \rangle \in \mathcal{C}$ with $\phi(I' \cup a) = I$ for some $a \in A$, is as hard as generating an arbitrary element of \mathcal{C} uniformly at random.*

Proof. A given context (A, O, \mathcal{D}) with concepts \mathcal{C} can be transformed into a context (A', O', \mathcal{D}') with concepts \mathcal{C}' as follows: set $A' = A \cup \{a^*\}$ with $a^* \notin A$, $O' = O \cup \{o^*\}$ with $o^* \notin O$, and

$$\mathcal{D}' = \mathcal{D} \cup \{(a, o^*) : a \in A'\}.$$

Then $\mathcal{C} = \mathcal{C}' \setminus \{\langle A', \{o^*\} \rangle\}$ and for all $\langle I, E \rangle \in \mathcal{C}$, it holds that $\phi'(I \cup \{a^*\}) = A'$, i.e., all $C \in \mathcal{C}$ are predecessors of $\langle A', \{o^*\} \rangle$ in $\mathbf{G}_{\phi'}$. \square

An alternative approach is to use the a random walk on the *union* of both generator graphs as stochastic

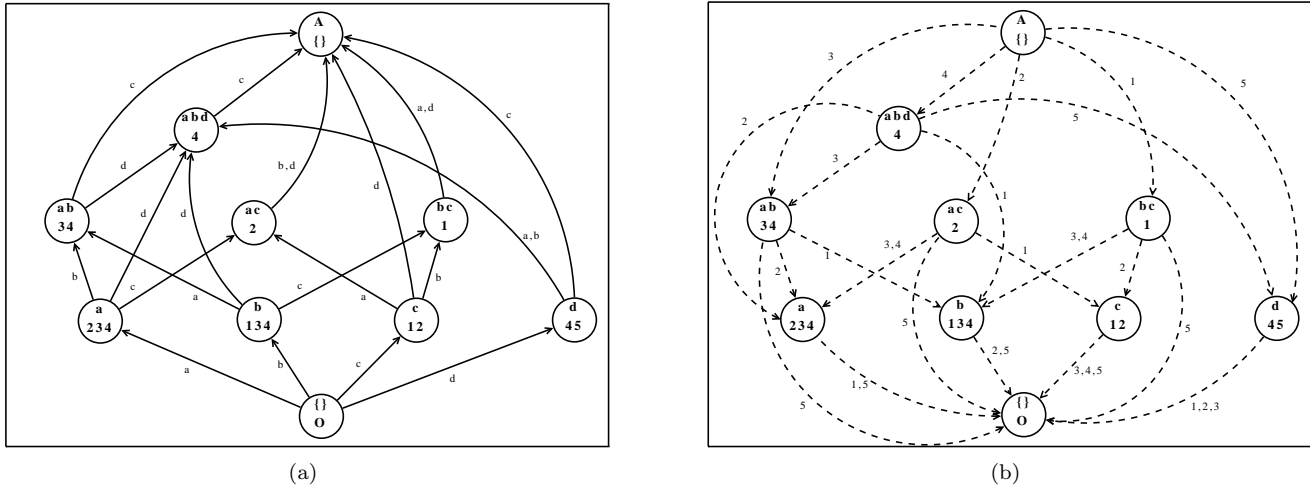


Figure 2: Generator graphs \mathbf{G}_ϕ and \mathbf{G}_ψ (drawn without self-loops) for the context from Table 2.

process. Technically, this can be realized as follows: in a current state flip a fair coin and then, based on the outcome, choose either \mathbf{G}_ϕ or \mathbf{G}_ψ to proceed to the next state as described above. This leads to the state transition probabilities

$$q(C, C') = \begin{cases} |G_\phi(I, I')| / (2|A|), & \text{if } C \prec C' \\ |G_\psi(E, E')| / (2|O|), & \text{if } C \succ C' \\ |I| / (2|A|) + |E| / (2|O|), & \text{if } C = C' \end{cases}.$$

for concepts $C = \langle I, E \rangle$ and $C' = \langle I', E' \rangle$. It is easy to see that the resulting stochastic process is an ergodic Markov chain. Thus, it has a stationary distribution to which it converges. Generally, however, we do not know anything about this distribution. In the next subsection we show how the chain can be modified such that it converges to a distribution that is known and desired.

3.2 Metropolis-Hastings Algorithm Let $\pi : \mathcal{C} \rightarrow [0, 1]$ be the desired target distribution, according to which we would like to sample concepts. For technical reasons that will become clear shortly we require π to be strictly positive, i.e., $\pi(C) > 0$ for all $C \in \mathcal{C}$. In practice this can be achieved easily. If our ergodic preliminary chain would have symmetric state transition probabilities, we could simply use it as *proposal chain* as follows: in a current concept C , propose a successor C' according to q , and then accept the proposal with probability $\pi(C')/\pi(C)$. This is the classic Metropolis algorithm, and for the resulting Markov chain it is easy to check that it satisfies the detailed balance condition for π —granted that q is symmetric as well as ergodic and that π is strictly positive. However, the example

in Figure 2 shows that the union of the two generator graphs corresponding to a given context does not necessarily induce symmetric state transition probabilities. In fact, the resulting process is in general not even time-reversible—see for instance concepts \perp and $\langle bc, 1 \rangle$ in Figure 2, for which we have $q(\perp, \langle bc, 1 \rangle) = 0$ but $q(\langle bc, 1 \rangle, \perp) > 0$. As a solution one can factor the quotient of the proposal probabilities into the acceptance probabilities. The resulting state transitions are

$$p(C, C') = \begin{cases} q(C, C') \min\{\alpha \frac{\pi(C')}{\pi(C)}, 1\}, & \text{if } q(C, C') > 0 \\ 0, & \text{otherwise} \end{cases}$$

where $\alpha = q(C', C)/q(C, C')$. This is the Metropolis-Hastings algorithm [15], and the underlying process is called the *Metropolis-process* of q and π . For the example from Table 2 its state transition probabilities are drawn in Figure 3. It is important to note that, in order to simulate the Metropolis process, one does not need the exact probabilities $\pi(C')$ and $\pi(C)$. As the acceptance probability only depends on their quotient, it is sufficient to have access to an *unnormalized potential* $f : \mathcal{C} \rightarrow \mathbb{R}$ such that there is a constant α with $f(C) = \alpha\pi(C)$ for all $C \in \mathcal{C}$. For instance, for the uniform target distribution one can choose any constant function f . Algorithm 1 is a Markov chain Monte Carlo implementation of the Metropolis process. It takes as input a context, a number of iterations s , and an oracle for the unnormalized potential f . Then, after choosing an initial state X_0 uniformly among \perp and \top , it simulates the process for s steps, and returns the state it has reached by that time, i.e., its realization of X_s .

It is easy to see that the algorithm indeed uses the

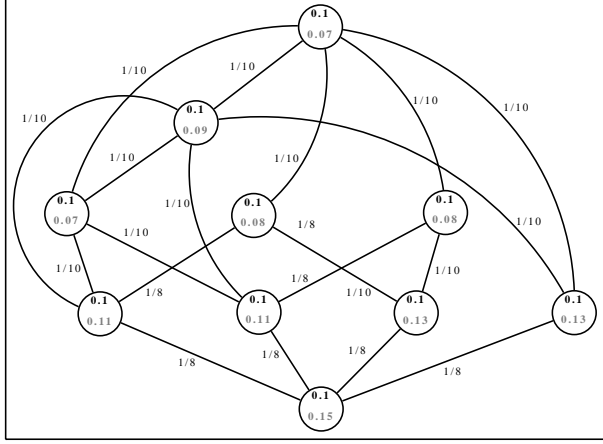


Figure 3: Resulting state transitions for Table 2 and $f(\cdot) \equiv 1$; remaining probabilities are assigned to self-loops; nodes contain stationary probability (black) and apx. probability after five steps with $X_0 = (\emptyset, O)$ (grey).

state transition probabilities p . So far, however, we omitted an important aspect of its correctness: while π satisfies the detailed balance condition for this chain, this comes at the cost of setting $p(C, C') = 0$ for some pairs of concepts that have $q(C, C') > 0$. Thus, the irreducibility of q is not directly implied by the irreducibility of p . It is, however, guaranteed by the closure properties of ϕ and ψ , as shown in the proof of the following summarizing theorem.

Algorithm 1 Metropolis-Hastings Concept Sampling

Input : context (A, O, \mathcal{D}) , number of iterations s ,
oracle of map $f : \mathcal{C} \rightarrow \mathbb{R}_+$

Output : concept $\langle I, E \rangle$

1. **init** $\langle I, E \rangle \sim u(\{\top, \perp\})$ and $i \leftarrow 0$
 2. $i \leftarrow i + 1$
 3. **draw** $d \sim u(\{\text{up}, \text{down}\})$
 4. **if** $d = \text{up}$ **then**
 5. **draw** $a \sim u(A)$
 6. $\langle I', E' \rangle \leftarrow \langle \phi(I \cup \{a\}), O[\phi(I \cup \{a\})] \rangle$
 7. $\alpha \leftarrow (|G_\psi(E', E)| |A|) / (|G_\phi(I, I')| |O|)$
 8. **else**
 9. **draw** $o \sim u(O)$
 10. $\langle I', E' \rangle \leftarrow \langle A[\psi(E \cup \{o\})], \psi(E \cup \{o\}) \rangle$
 11. $\alpha \leftarrow (|G_\phi(I', I)| |O|) / (|G_\psi(E, E')| |A|)$
 12. **draw** $x \sim u([0, 1])$
 13. **if** $x < \alpha f(I') / f(I)$ **then** $\langle I, E \rangle \leftarrow \langle I', E' \rangle$
 14. **if** $i = s$ **then return** $\langle I, E \rangle$ **else goto** 2
-

THEOREM 3. *On input context (A, O, \mathcal{D}) , step number s , and strictly positive function f , Algorithm 1 produces in time $\mathcal{O}(s|\mathcal{D}|)$ a concept $C \in \mathcal{C}(A, O, \mathcal{D})$ according to a distribution p_f^s such that*

$$\lim_{s \rightarrow \infty} \|p_f^s, \pi\|_{tv} = 0$$

where π is the distribution on \mathcal{C} resulting from normalizing f , i.e., $\pi(\cdot) = f(\cdot) / \sum_{C \in \mathcal{C}} f(C)$.

Proof. The time complexity is easy to see. Also, by construction, it can directly be checked that π and p satisfy the detailed balance condition (Eq. 2.1). It remains to show irreducibility, i.e., $p^t(\langle I, E \rangle, \langle I', E' \rangle) > 0$ for some t and all pairs of concepts $\langle I, E \rangle, \langle I', E' \rangle \in \mathcal{C}$. It is sufficient to consider the case $\langle I, E \rangle \preceq \langle I', E' \rangle$: for other states reachability follows then via $\langle I \cap I', A[I \cap I'] \rangle \preceq \langle I, E \rangle, \langle I \cap I', A[I \cap I'] \rangle \preceq \langle I', E' \rangle$, and the transitivity of reachability. Moreover, as the considered state spaces \mathcal{C} are finite, it suffices to consider direct successors $\langle I, E \rangle \preceq \langle I', E' \rangle$. For such concepts it is easy to show that I is a maximal proper subset of I' in $\phi(\mathcal{P}(A))$ and E' is a maximal proper subset of E in $\psi(\mathcal{P}(O))$. Let $a \in I' \setminus I$ and $o \in E \setminus E'$. It follows by the closure operator properties that $\phi(I \cup \{a\}) = I'$ and $\psi(E' \cup \{o\}) = E$. Thus, the proposal probabilities between $\langle I, E \rangle$ and $\langle I', E' \rangle$ are non-zero in both directions, and together with the fact that f is strictly positive, it follows for the transition probability of the Metropolis-process that $p(\langle I, E \rangle, \langle I', E' \rangle) > 0$ as required. \square

Now that we know that our algorithm asymptotically draws samples as desired, it remains to discuss, for how many steps one has to simulate the process until the distribution is “close enough” to the target distribution.

3.3 Number of Iterations Unfortunately, in the worst case the mixing can be infeasibly slow, i.e., require a number of steps that can grow exponentially in the input size. The following statement on the mixing time holds.

PROPOSITION 4. *Let $\epsilon > 0$ be fixed and $\tau_n(\epsilon)$ denote the worst-case mixing time of Algorithm 1 for the uniform target distribution and an input context of size n . Then $\tau_n(\epsilon) \in \Omega(2^{n/2})$.*

Proof. We prove the claim by showing that for all n with an even integral square root there is a context (A, O, \mathcal{D}) with $A = O = \{1, \dots, \sqrt{n}\}$ and $|\mathcal{D}| = n/2 - n$ such that $\tau(\epsilon) \geq 2^{n/2-3} \log(1/(2\epsilon))$. It is well-known (see for instance [22]) that the mixing time of a Markov chain with state space \mathcal{C} and stationary distribution π is lower bounded by

$$(3.2) \quad \tau(\epsilon) \geq 1/(4\Phi) \log(1/\epsilon) ,$$

where Φ is the *conductance* defined as the minimum number $\Phi_{\mathcal{S}}$ over all $\mathcal{S} \subseteq \mathcal{C}$ with $\pi(\mathcal{S}) \leq 1/2$ where $\Phi_{\mathcal{S}}$ is equal to $(1/\pi(\mathcal{S})) \sum_{x \in \mathcal{S}, y \in \mathcal{C} \setminus \mathcal{S}} \pi(x)p(x, y)$. Choose

$$\mathcal{D} = \{(i, j) : i \neq j, (i, j \leq \sqrt{n}/2 \vee i, j > \sqrt{n}/2)\} .$$

Then the set of concepts is a disjoint union $\mathcal{C}(A, O, \mathcal{D}) = \mathcal{C}_1 \cup \mathcal{C}_2$ with

$$\mathcal{C}_1 = \{\langle I, O[I] \rangle : I \subseteq \{1, \dots, \sqrt{n}/2\}\}$$

$$\mathcal{C}_2 = \{\langle I, O[I] \rangle : \emptyset \neq I \subseteq \{\sqrt{n}/2 + 1, \dots, \sqrt{n}\}\} \cup \{\top\}$$

and if π is the uniform distribution, it holds that $\pi(\mathcal{C}_1) = 1/2$. Moreover, the only state pairs that contribute to $\Phi_{\mathcal{C}_1}$ are the \sqrt{n} pairs $\{\perp, C\}$ and $\{C', \top\}$ of the form

$$C \in \{\langle I, E \rangle : I \in \{\{\sqrt{n}/2 + 1\}, \dots, \{\sqrt{n}\}\}\}$$

$$C' \in \{\langle A \setminus \{a\}, E \rangle : a \in \{1, \dots, \sqrt{n}/2\}\} .$$

Consequently, $1/2^{n/2-1} = \Phi_{\mathcal{C}_1} \leq \Phi$ and with (3.2), $\tau(\epsilon) \geq 2^{n/2-3} \log(1/(2\epsilon))$ as required. \square

Thus, even the strictest theoretical worst-case bound on the mixing time for general input contexts would not lead to an efficient algorithm. The proof of the proposition is based on the observation that the chain can have a very small *conductance*, i.e., a relatively large portion of the state space (growing exponentially in n) can only be connected via a linear number of states to the rest of the state space. We assume that real-world datasets do not exhibit this behavior. In our applications we therefore use the following heuristic polynomially bounded function for assigning the number of simulation steps:

$$\text{steps}((A, O, \mathcal{D}), \epsilon) = 4n \ln(n) \ln(\epsilon)$$

where $n = \min\{|A|, |O|\}$. The motivation for this is as follows (see also [5], where a similar heuristic is used). Assume without loss of generality that $|A| < |O|$. As long as only “ $d = \text{up}$ ” is chosen in line 3 of the algorithm the expected number of steps until all elements of A have been drawn at least once is $n \ln(n) + \mathcal{O}(n)$ with a variance that is bounded by $2n^2$ (coupon collector’s problem). It follows that asymptotically after $2n \ln(n)$ all elements have been drawn with probability at least $3/4$. Consequently, as “ $d = \text{up}$ ” is chosen with probability $1/2$, we can multiply with an additional factor of 2 to know that with high probability there was at least one step upward and one step downward for each attribute $a \in A$ —if one assumes a large conductance this suffices to reach every state with appropriate probability. The final factor of $\ln(\epsilon)$ is motivated by the fact that the total variation distance usually decays exponentially in the number of simulation steps.

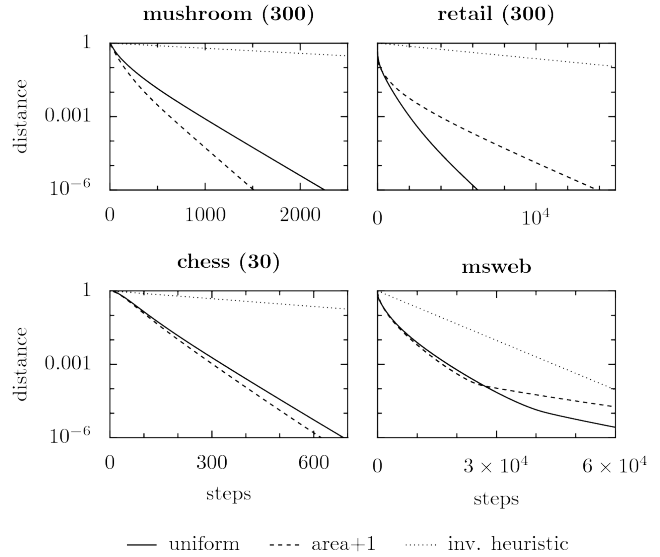


Figure 4: Distribution convergence.

This can also be observed in our experiments with four real-world databases¹ and two target distributions: the uniform distribution and the distribution proportional to the area quality measure $\pi(\cdot) = q_{\text{area}}(\cdot)/Z$. The results are illustrated in Figure 4. Note that the plots differ in their semantics. While for both target distributions a point (x, y) means that the x -step distribution has a total variation distance of y from the target distribution, for the heuristic the inverse of the step-heuristic is shown, i.e., the value of ϵ on the y -axis reflects the desired accuracy that leads to a particular number of steps. Thus, the heuristic was “correct” if its corresponding plot dominates the two total variation distance plots. As we can see, this was the case for both distributions on all four datasets. In fact, the heuristic was rather conservative. Note that the y -axis has a logarithmic scale.

The effect of the target distribution on the mixing time is unclear. On the one hand, deviation from the uniform distribution can create bottlenecks, i.e., low conductance sets, where there have been non before. On the other hand, it can also happen that bottlenecks of the uniform distribution are softened. This can be observed for “mushroom” and “chess”. For that reason we do not reflect the target distribution in our heuristic.

¹Databases are from UCI Machine Learning Repository [2]. In order to explicitly compute the state transition matrices we had to use only a sample of the transactions for three of the databases. The sample size is given in brackets.

4 Concept Counting

In this section we highlight the connection of concept sampling to concept counting. This connection is important because it ties the complexities of these two tasks together. In particular, we show how concept sampling can be used to design a randomized approximation scheme for the number of concepts of a given context, i.e., a randomized algorithm that produces for an input context (A, O, \mathcal{D}) and $\epsilon \in (0, 1/2]$ a number N such that

$$(1 - \epsilon) |(A, O, \mathcal{D})| \leq N \leq (1 + \epsilon) |(A, O, \mathcal{D})|$$

with probability at least $3/4$. Beside theoretical insight, the motivation for this is that such an algorithm can be used to quickly check the feasibility of an potentially exponential time exhaustive listing of all concepts (see also [5] where the same problem is discussed for frequent itemsets).

Let (A, O, \mathcal{D}) be a context and o_1, \dots, o_m some ordering of the elements of O . For $i \in \{0, \dots, m\}$ define the context $\mathbb{C}_i = (A, O_i, \mathcal{D}_i)$ with $O_i = \{o_j : j \leq i\}$, \mathcal{D}_i as the restriction of \mathcal{D} to O_i , ϕ_i as the corresponding attribute closure operator, and $\mathcal{I}_i = \phi_i(\mathcal{P}(A))$ the concept intents. Note that in general a concept intent $I \in \mathcal{I}_{i+1}$ is not an intent of a concept with respect to the context \mathbb{C}_i , i.e., not a fixpoint of ϕ_i . The following two properties, however, hold.

LEMMA 5. *For all contexts (A, O, \mathcal{D}) and all $i \in \{0, \dots, |O|\}$ it holds that (i) $\mathcal{I}_i \subseteq \mathcal{I}_{i+1}$ and (ii) $1/2 \leq |\mathcal{I}_i| / |\mathcal{I}_{i+1}|$.*

Proof. For (i) let $I \in \mathcal{I}_i$. In case $o_{i+1} \notin I[O_{i+1}]$, it is $I[O_{i+1}] = I[O_i]$. Otherwise, per definition we know that all $a \in A$ that satisfy $(a, o) \in \mathcal{D}$ for all $o \in O_i$ are also satisfying $(a, o) \in \mathcal{D}$ for all $o \in O_{i+1}$. Thus, in both cases $\phi_{i+1}(I) = A[O_{i+1}[I]] = A[O_i[I]] = \phi_i(I) = I$ as required.

We prove (ii) by showing that the restriction of the closure operator ϕ_i to $\mathcal{I}_{i+1} \setminus \mathcal{I}_i$ is an injective map into \mathcal{I}_i , i.e., $|\mathcal{I}_{i+1} \setminus \mathcal{I}_i| \leq |\mathcal{I}_i|$. Together with (i) this implies the claim. Assume for a contradiction that there are distinct intents $X, Y \in \mathcal{I}_{i+1} \setminus \mathcal{I}_i$ such that $\phi_i(X) = \phi_i(Y)$. Then $O_i[X] = O_i[Y]$, and, as X and Y are both closed with respect to ϕ_{i+1} but not closed with respect to ϕ_i , it must hold that $o_{i+1} \in O_{i+1}[X] \cap O_{i+1}[Y]$. It follows that $O_{i+1}[X] = O_i[X] \cup \{o_{i+1}\}$ as well as $O_{i+1}[Y] = O_i[Y] \cup \{o_{i+1}\}$. This implies $O_{i+1}[X] = O_{i+1}[Y]$ and in turn $X = Y$ contradicting the assumption that the intents are distinct. \square

Thus, the \mathcal{I}_i define an increasing sequence of closed set families with $|\mathcal{I}_0| = |\{A\}| = 1$ and $|\mathcal{I}_m| = |\mathcal{C}(A, O, \mathcal{D})|$ that allows to express the number of concepts of the

complete context as

$$(4.3) \quad |\mathcal{C}(A, O, \mathcal{D})| = \left(|\mathcal{I}_0| \prod_{i=1}^m |\mathcal{I}_{i-1}| / |\mathcal{I}_i| \right)^{-1}.$$

For $i \in \{1, \dots, m\}$ let $Z_i(I)$ denote the binary random variable on measure space (\mathcal{I}_i, π_i) that takes on value 1 if $I \in \mathcal{I}_{i-1}$ and 0 otherwise. Independent simulations of these random variables can be used to count the number of concepts via Equation 4.3 using the product estimator $Z = \prod_{i=1}^m \bar{Z}_i$ where

$$\bar{Z}_i = (Z_i^{(1)} + \dots + Z_i^{(t)})/t$$

with $Z_i^{(j)}$ independent copies of Z_i . Using standard reasoning (see, e.g., [17]) and Lemma 5 one can show that Z is ϵ -close to $|\mathcal{C}(A, O, \mathcal{D})|$ with probability at least $3/4$ if (i) the total variation distance of the π_i to the uniform distribution on \mathcal{I}_i is not greater than $\epsilon/(12|O|)$ and (ii) $t \geq 12|O|/\epsilon^2$. See Algorithm 2 for a pseudocode simulating this estimator. Observing that the roles of A and O are interchangeable, i.e., the number of concepts of (A, O, \mathcal{D}) is equal to the number of concepts of the “transposed” context, we can conclude:

THEOREM 6. *There is a randomized approximation scheme for the number of concepts $|\mathcal{C}|$ of a given context (A, O, \mathcal{D}) with time complexity $\mathcal{O}(n\epsilon^{-2}T_S(\epsilon/(12n)))$ for accuracy ϵ where $n = \min(|A|, |O|)$ and $T_S(\epsilon')$ is the time required to sample a concept almost uniformly, i.e., according to a distribution with a total variation distance of at most ϵ' from uniform.*

Algorithm 2 Concept Counting

Input : context (A, O, \mathcal{D}) , accuracy $\epsilon \in (0, \frac{1}{2})$
 Require: $\|p_{i, \mathbb{1}}^{\text{steps}(\mathbb{C}_i, \epsilon')}\|_{\text{tv}} \leq \epsilon'$ for all i and ϵ'
 Output: q with $\mathbb{P}[(1-\epsilon)|\mathcal{C}| \leq q \leq (1+\epsilon)|\mathcal{C}|] \geq 3/4$

1. $t \leftarrow 12|O|/\epsilon^2$
 2. **for** $i = 1, \dots, |O|$ **do**
 3. $r_i \leftarrow 0$
 4. **for** $k = 1, \dots, t$ **do**
 5. $\langle I, E \rangle \leftarrow \text{sample}(\mathbb{C}_i, \text{steps}(\mathbb{C}_i, \epsilon/(12|O|)), \mathbb{1})$
 6. **if** $\phi_{i-1}(I) = I$ **then** $r_i \leftarrow r_i + 1$
 7. $r_i \leftarrow r_i/t$
 8. **return** $\prod_{i=1}^{|O|} r_i^{-1}$
-

To evaluate the accuracy of this randomized counting approach, we executed a series of runs on the *chess* dataset and compared the estimate computed by our randomized algorithm with the exact number of concepts. We did not consider the whole dataset but instead used samples of different size. We used an accuracy of $\epsilon = 0.5$. The result is shown in Figure 4. On the

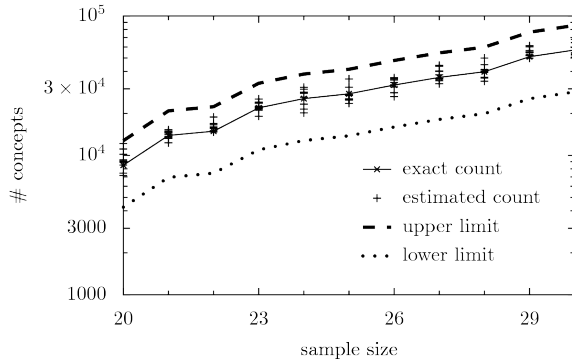


Figure 5: Estimated number of concepts on samples of the 'chess' dataset.

x-axis, we give the size of the sample, while the y-axis shows the exact number of concepts (“exact count”), the upper and lower 0.5 deviation limits (“upper limit” and “lower limit”), as well as the result of the randomized algorithm in a series of 10 runs per sample (“estimated count”). The figure shows that in all randomized runs, the approximated result lies within the given deviation bound. This is a further positive evaluation of the step heuristic developed in Section 3.3.

5 Application to Local Pattern Mining

In this section, we will present exemplary applications of the sampling algorithm to local pattern mining. This revisits the motivations from Section 1.2.

5.1 Association Rule Mining First, we illustrate how the sampling algorithm can be used in the context of *minimal non-redundant association rule discovery* (see Section 2). Again, for the sake of simplicity, we consider exact association rules, i.e., rules with a confidence of 1. Sampling a rule with confidence below 1 can be done, for instance, by sampling the consequence of the rule in a first step; and then sampling the antecedent according to a distribution that is proportional to the confidence and support of the resulting rule. Based on Proposition 1, we can generate exact minimal non-redundant association rules by first sampling a concept $\langle Y, E \rangle$, and then calculating a minimal generator X of the concept’s intent, i.e., a minimal set with $\phi(X) = Y$. For the calculation of the minimal generators, we use a simple greedy algorithm that is essentially equivalent to the greedy algorithm for the set cover problem (see [6] for a discussion). In fact, with this algorithm we generate an approximation to a *shortest* generator and not only a *minimal* generator with respect to set inclusion. Thus, we are aiming for particularly short rule antecedents.

	Election	Questio.	Census
sampling wrt. q_{supp}	2.1 sec.	21 min.	7 min.
sampling wrt. q_{area}	2.2 sec.	21 min.	7 min.
JClose...	59 min.	7500 m.	1729 min.
... with support	0.35	0.7	0.7

Table 3: Time for association rule generation.

With this method we generate association rules for the three datasets *Election*, *Questionnaire*, and *Census* from Section 1.2 (see Table 1) for two different target distributions: the one proportional to the rule’s *support* and the one proportional to their *area* (adding the constant 1 to ensure a strictly positive distribution). That is, we used $f = q_{\text{supp}}$ respectively $f = q_{\text{area}}$ (see Section 2) as parameter for Algorithm 1. As desired closeness to the target distribution we choose $\epsilon = 0.01$. Note that, in contrast to exhaustive listing algorithms, for the sampling method we do not need to define any threshold like minimum support. Table 3 shows the time required to generate a single rule according to q_{supp} and q_{area} . In addition, the table shows the runtime of JClose for exemplary support thresholds. Comparing the runtimes for sampling with that of JClose, we can see that the uniform generation of a rule from the *Election* dataset takes about 1/1609 of the time needed to exhaustively list non-redundant association rules with threshold 0.35. For *Questionnaire* and threshold 0.7, the ratio is 1/340; finally for *Census* with threshold 0.7, the ratio is 1/247. As discussed in Section 1.2, all exhaustive listing algorithms exhibit a similar asymptotic performance behavior. Thus, for methods that outperform JClose on these particular datasets, we would end up with similar ratios for slightly smaller support thresholds. Moreover, it is not uncommon that one is interested in a small set of rules having a substantially lower support than the thresholds considered above. For such cases the ratios would increase dramatically, because the time for exhaustive listing increases exponentially whereas the time for sampling remains constant. Thus, for such applications sampling has the potential to provide a significant speedup. In fact, it is applicable in scenarios where exhaustive listing is hopeless.

5.2 Supervised Descriptive Rule Induction As second application we consider *supervised descriptive rule induction*, i.e., local pattern mining considering labeled data such as emerging pattern mining, contrast set mining, or subgroup discovery (see Section 2). This can be done by choosing the binomial quality q_{bino} as parameter for Algorithm 1. In fact, we can

dataset	#rows	#cols.	#attr.	size	target
sick	3772	27	66	299K	sick
soyb.	683	35	133	199K	br.-spot
lung.	32	56	159	10K	'1'

Table 4: Databases for the supervised descriptive rule induction experiments.

	sick	soyb.	lung.
sampling time	3	1	0.02
listing time	1460	705	69

Table 5: Time (in seconds) of random pattern generation according to q_{bino} compared to exhaustive listing.

q_{bino} from below by a small constant c to ensure a strictly positive target distribution, i.e., we use $f(\cdot) = \max\{q_{\text{bino}}(\cdot), c\}$. With the binomial quality function we have an evaluation metric that has shown to adequately measure interestingness in several applications. Thus, in addition to the computation time, in this experiment we can also evaluate the quality of the sampled patterns.

We consider three labeled datasets from the UCI machine learning repository (listed in Table 4). As baseline method we use exhaustive closed subgroup discovery based on generator graph traversal [6]. This algorithm explores the same state space as the sampling algorithm and outperforms other (non-closed) listing algorithms on the considered datasets. Table 5 gives the time needed to generate a pattern by the randomized algorithm compared to the time for exhaustively listing all patterns with a binomial quality of at least 0.1. This threshold is approximately one third of the binomial quality of the best patterns in all three datasets. As observed in the unsupervised setting, the randomized approach allows to generate a pattern in a fraction of the time needed for the exhaustive computation. In order to evaluate the pattern quality, we compared the best patterns within a set of 100 samples to the globally best patterns (Table 6). Although the quality of the randomly generated patterns do not reach the optimum, we observe the anticipated and desired result: the collected subgroups form a high quality sample relatively to the complete pattern space, from which they are drawn.

6 Discussion

We presented a Metropolis-Hastings algorithm able to effectively generate concepts according to some desired target distribution. In several exemplary applications we demonstrated how this algorithm can be used for

	sick	soybean	lung.
best quality	0.177	0.223	0.336
sampled quality	0.167	0.187	0.263

Table 6: Best quality among all patterns versus best quality among sampled patterns.

knowledge discovery and approximate counting. It is important to note that although we restricted the presentation to closed sets, the approach is not limited to this scenario. In fact it can be applied to all pattern classes having a similar Galois connection between patterns and transactions. Furthermore, the method can be combined with any anti-monotone or monotone constraint. It is for instance easy to see that the algorithm is still correct when the chain is restricted to the set of all frequent concepts.

Although the heuristic polynomial step number leads to sufficient results in our experiments, clearly one would prefer a polynomial sampling algorithm with provable worst-case guarantee. There is, however, some evidence that no such algorithm exists or that at least designing one is a difficult problem: as we have shown in Section 4, an algorithm that can sample concepts uniformly in polynomial time can be used to design a fully polynomial randomized approximation scheme (FPRAS) for counting the number of concepts for a given context. This problem is equivalent to counting the number of maximal bipartite cliques of a given bipartite graph, which is as hard as a complexity class called $\#\text{RHH}_1 \subset \#\text{P}$ (hardness result and complexity class both were introduced in [8]). Despite much interest for such algorithms in several research communities, there is no known FPRAS for any $\#\text{RHH}_1$ -hard problem.

In the absence of feasible a priori bounds on the mixing time, there are several techniques that can be used not only to detect mixing but even to draw *perfect samples*, i.e., generated exactly according to the stationary distribution (see [16] and references therein). The most popular variant of these perfect sampling techniques is coupling from the past (CFTP). It can be applied efficiently if the chain is monotone in the following sense: the state space is partially ordered, contains a global maximal as well as a global minimal element, and if two simulations of the chain that use the same source of random bits are in state x and y with $x \leq y$ then also the successor state of x must be smaller than the successor state of y . Indeed, at first glance it appears that CFTP may be applied to our chain because its state space is partially ordered by \preceq and always contains a global minimal element \perp

as well as a global maximal element \top . Even for the uniform target distribution, however, the chain is not monotone. This can be observed in the example of Figure 2: denote by $\text{succ}_{d,5,0}(\langle I, E \rangle)$ the successor of $\langle I, E \rangle$ when the random bits used by the computation induce the decisions $d = \text{down}$, $o = 5$, and $p = 0$. Then $\langle bc, 1 \rangle \preceq \langle A, \emptyset \rangle$ whereas $\text{succ}_{d,5,0}(\langle bc, 1 \rangle) = \langle bc, 1 \rangle \not\preceq \langle d, 45 \rangle = \text{succ}_{d,5,0}(\langle A, \emptyset \rangle)$. It is an open problem, how to define a Markov chain on the concept lattice that has monotone transition probabilities.

Acknowledgements

Part of this work was supported by the German Science Foundation (DFG) under the reference number ‘GA 1615/1-1’.

References

- [1] Foto N. Afrati, Aristides Gionis, and Heikki Mannila. Approximating a collection of frequent sets. In *KDD*, pages 12–19. ACM, 2004.
- [2] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
- [3] Yves Bastide, Nicolas Pasquier, Rafik Taouil, Gerd Stumme, and Lotfi Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. In *Computational Logic - CL 2000*, pages 972–986. Springer, 2000.
- [4] Stephen D. Bay and Michael J. Pazzani. Detecting group differences: Mining contrast sets. *Data Min. Knowl. Discov.*, 5(3):213–246, 2001.
- [5] Mario Boley and Henrik Grosskreutz. A randomized approach for approximating the number of frequent sets. In *ICDM*, pages 43–52. IEEE Computer Society, 2008.
- [6] Mario Boley and Henrik Grosskreutz. Non-redundant subgroup discovery using a closure system. In *ECML/PKDD (2)*, pages 179–194. Springer, 2009.
- [7] Richard Chow, Philippe Golle, and Jessica Staddon. Detecting privacy leaks using corpus-based association rules. In *KDD 2008*, pages 893–901. ACM, 2008.
- [8] Martin Dyer, Leslie Ann Goldberg, Catherine Greenhill, and Mark Jerrum. The relative complexity of approximate counting problems. *Algorithmica*, 38(3):471–500, 2004.
- [9] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer Verlag, 1999.
- [10] Gemma C. Garriga, Petra Kralj, and Nada Lavrač. Closed sets for labeled data. *J. Mach. Learn. Res.*, 9:559–580, 2008.
- [11] Floris Geerts, Bart Goethals, and Taneli Mielikäinen. Tiling databases. In *DS*, pages 278–289, 2004.
- [12] David Hand. Pattern detection and discovery. In *Pattern Detection and Discovery*, volume 2447 of *LNAI*, pages 1–12. Springer, 2002.
- [13] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, Jérémy Besson, and Mohammed Javeed Zaki. Origami: Mining representative orthogonal graph patterns. In *ICDM*, pages 153–162. IEEE Computer Society, 2007.
- [14] Mohammad Al Hasan and Mohammed Javeed Zaki. Musk: Uniform sampling of k maximal patterns. In *SDM*, pages 650–661. SIAM, 2009.
- [15] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [16] Mark Huber. Perfect simulation with exponential tails on the running time. *Random Struct. Alg.*, 33(1):29–43, 2008.
- [17] Mark Jerrum and Alistair Sinclair. *The Markov chain Monte Carlo method: an approach to approximate counting and integration*, pages 482–520. PWS Publishing Co., Boston, MA, USA, 1997.
- [18] A. Knobbe, B. Crémilleux, J. Fürnkranz, and M. Scholz. From local patterns to global models: the lego approach to data mining. In *Proc. of the ECML PKDD 2008 LEGO Workshop*, 2008.
- [19] Matthew V. Mahoney and Philip K. Chan. Learning rules for anomaly detection of hostile network traffic. In *ICDM*, page 601, Washington, DC, USA, 2003. IEEE Computer Society.
- [20] Shinichi Morishita and Jun Sese. Traversing itemset lattice with statistical metric pruning. In *PODS*, pages 226–236. ACM, 2000.
- [21] Nicolas Pasquier, Rafik Taouil, Yves Bastide, Gerd Stumme, and Lotfi Lakhal. Generating a condensed representation for association rules. *Journal of Intelligent Information Systems*, 24(1):29–60, January 2005.
- [22] Dana Randall. Rapidly mixing markov chains with applications in computer science and physics. *Computing in Science and Engineering*, 8(2):30–41, 2006.
- [23] Leander Schietgat, Fabrizio Costa, Jan Ramon, and Luc De Raedt. Maximum common subgraph mining: A fast and effective approach towards feature generation. In *MLG*, 2009.
- [24] J. Wang, J. Han, Y. Lu, and P. Tzvetkov. Tfp: an efficient algorithm for mining top-k frequent closed itemsets. *Knowledge and Data Engineering, IEEE Transactions on*, 17(5):652–663, May 2005.
- [25] Lusheng Wang, Hao Zhao, Guozhu Dong, and Jianping Li. On the complexity of finding emerging patterns. *Theoretical Computer Science*, 335(1):15–27, 2005.
- [26] Dong Xin, Jiawei Han, Xifeng Yan, and Hong Cheng. Mining compressed frequent-pattern sets. In *VLDB*, pages 709–720. VLDB Endowment, 2005.
- [27] Xifeng Yan, Hong Cheng, Jiawei Han, and Philip S. Yu. Mining significant graph patterns by leap search. In *SIGMOD*, pages 433–444. ACM, 2008.