Direct Local Pattern Sampling by Efficient Two-Step Random Procedures

Mario Boley
Fraunhofer IAIS and
University of Bonn
mario.boley@iais.fhg.de

Claudio Lucchese
I.S.T.I.-C.N.R. Pisa
claudio.lucchese@isti.cnr.it

Daniel Paurat
University of Bonn
daniel.paurat@uni-bonn.de

Thomas Gärtner
Fraunhofer IAIS and
University of Bonn
thomas.gaertner@iais.fhg.de

ABSTRACT

We present several exact and highly scalable local pattern sampling algorithms. They can be used as an alternative to exhaustive local pattern discovery methods (e.g, frequent set mining or optimistic-estimator-based subgroup discovery) and can substantially improve efficiency as well as controllability of pattern discovery processes. While previous sampling approaches mainly rely on the Markov chain Monte Carlo method, our procedures are direct, i.e., non process-simulating, sampling algorithms. The advantages of these direct methods are an almost optimal time complexity per pattern as well as an exactly controlled distribution of the produced patterns. Namely, the proposed algorithms can sample (item-)sets according to frequency, area, squared frequency, and a class discriminativity measure. Experiments demonstrate that these procedures can improve the accuracy of pattern-based models similar to frequent sets and often also lead to substantial gains in terms of scalability.

1. INTRODUCTION

This paper presents simple yet effective procedures for local pattern discovery [20] that attack the task from a different algorithmic angle than the standard search approach—namely, by directly generating individual patterns as the outcome of a random experiment. Local patterns such as association rules [1] or emerging patterns [12] are used in various application contexts from exploratory data analysis where they constitute units of discovered knowledge to predictive model construction where patterns act as binary features [9, 10, 13]. What all applications have in common is that usually only a few patterns can be effectively utilized—either due to the limited attention of a data analyst or because too many features can reduce the comprehensibility and performance of a global model. Standard local pattern discovery algorithms, however, are based on exhaustive search within huge pattern spaces (e.g., frequent set miners [19, 26], or optimistic-estimator-based subgroup and association discovery [17, 23]). Consequently, they tend to either produce a vast amount of output patterns or at least enumerate them internally.

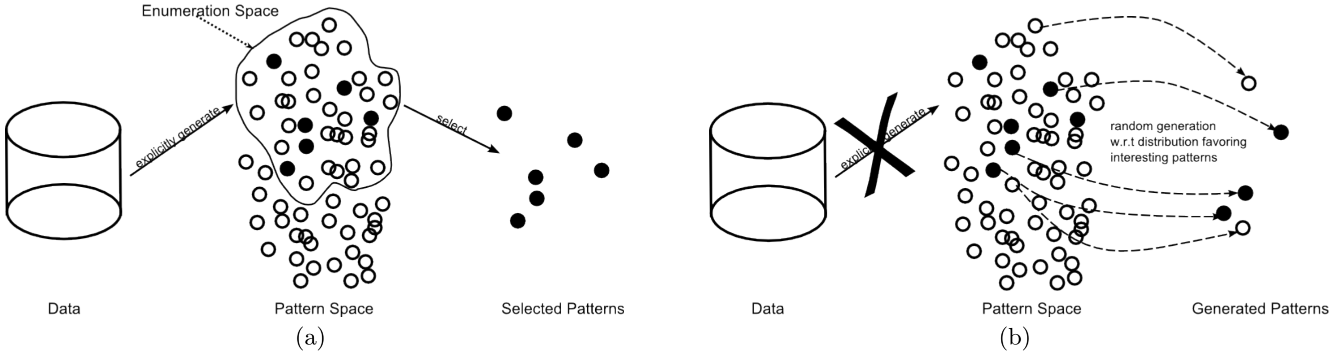
This motivates the invention of algorithms that only sample a representative set of patterns without explicitly searching in the pattern space. Such algorithms exist in the literature [2, 6, 8] but they provide either no control over the distribution of their output or only asymptotic control by simulating a stochastic process on the pattern space using the Markov chain Monte Carlo method (MCMC). In addition to only offering approximate sampling, MCMC methods have a scalability problem: the number of required process simulation steps is often large and, even more critical, individual simulation steps typically involve expensive support counting operations. Hence, these algorithms are often infeasible for large input datasets. Therefore, we present novel pattern generation methods that sample patterns exactly and directly, i.e., without simulating time-consuming stochastic processes. More precisely, given a dataset $\mathcal{D}$ and a number of desired patterns k, the procedures

- produce exactly k patterns each of which is generated exactly according to a distribution proportional to either frequency, squared frequency, area (i.e., frequency times size), or discriminativity (i.e., frequency in positive data portion times negative frequency in negative data portion);
- use time $O(\|\mathcal{D}\| + kn)$ for frequency and area or time $O(\|\mathcal{D}\|^2 + kn)$ for squared frequency and discriminativity, where n denotes the number of items and $\|\mathcal{D}\|$ the size of the dataset, i.e., the sum of all data record sizes[1].

That is, after a linear or quadratic preprocessing phase each pattern is produced in a time linear in the number of items. This time complexity appears to be almost optimal, because only reading the data once requires $O(\|\mathcal{D}\|)$ and just printing k patterns without any further computation requires time $O(kn)$.

KDD'11, August 21–24, 2011, San Diego, California, USA.
Copyright 2011 ACM 978-1-4503-0813-7/11/08 ...$10.00.

[1] This assumes that $\exp(n) > |\mathcal{D}|$; the actual complexities are $O(\|\mathcal{D}\| + k(n + \ln |\mathcal{D}|))$ and $O(\|\mathcal{D}\|^2 + k(n + \ln^2 |\mathcal{D}|))$.



(a)

(b)

Figure 1: (a) *Exhaustive search*: involves complete generation of an enumeration space guaranteed to contain all interesting patterns; however, size of that space usually has no reasonable bound w.r.t. to input size and is hard to predict. (b) *Controlled pattern sampling*: no explicit construction of uncontrolled, potentially huge, part of pattern space; instead random generation of designated number of patterns; no guarantee of finding patterns satisfying hard interestingness threshold, but control over time, output size, and distribution.

After giving some more background on the idea of controlled repeated pattern sampling and reviewing other pattern sampling algorithms, the remainder of this paper is structured as follows. We define formal and notational background (Sec. 2) followed by a detailed description of the sampling procedures (Sec 3). Then we report experimental results showing that sampled patterns are equally useful for pattern-based classification as frequent sets and that pattern sampling can easily outperform exhaustive listing on large datasets (Sec. 4). Finally, we give a summarizing discussion of all results (Sec. 5).

1.1 Pattern Sampling

The data mining literature contains several local pattern discovery algorithms that can efficiently produce large output families. Here, efficiency is defined in an output-sensitive way (see, e.g., [16]), that is for instance amortized polynomial time per pattern, which is a useful notion assuming that the produced pattern collections are the final output. When viewed from a global (application-driven) perspective though, the enumerated patterns are usually only an intermediate result, from which a final—often much smaller—pattern collection is selected. That is, enumeration is only one step within a surrounding local pattern discovery process. This two phase approach, which we refer to as "exhaustive search" is illustrated in Figure 1(a): during the enumeration step part of the implicitly defined pattern space is physically constructed—we refer to that part as "enumeration space"—and then, during the selection step, the most valuable patterns from this enumeration space are collected with respect to some interestingness measure.

An example for this paradigm is listing frequent sets of an input dataset, but subsequently using only those sets that provide rules with a large value of some measure that reflects the primary user interest (e.g., lift). A further example is optimistic-estimator-based pruning for subgroup or association discovery. There the enumeration space is the family of all sets having a large enough optimistic estimate of their interestingness, and the patterns that are selected for the result family are those with a large value of the actual interestingness measure. Note that, in this example, enumeration and selection are algorithmically interweaved, i.e., sets are already selected during the enumeration phase. Many more examples emerge from the LeGo approach to data mining [22] where patterns are selected according to their utility for constructing global models.

Two reasons usually motivate this two phase approach: either there is no efficient algorithm that directly optimizes the measure of primary interest or this measure is not known in advance (this happens in exploratory settings where interestingness can be user-specific). In either case, the enumeration step can constitute a severe bottleneck. Even if enumeration is performed by an amortized polynomial time algorithm, its computation time is essentially unpredictable: the size of the enumeration space cannot be directly controlled and its explicit construction takes time at least proportional to that size. On the other hand, if one enforces a maximum computation time by aborting the execution at a certain point, one ends up with an uncontrolled subset of the enumeration space, which depends on the internal search order of the enumeration algorithm.

In contrast, suppose we can access the pattern space $\mathcal{L}$ by an efficient sampling procedure simulating a distribution $\pi : \mathcal{L} \to [0, 1]$ that is defined proportional to some function that either is our primary interestingness measure or a function that correlates with it. Then, for any designated number of patterns (or corresponding designated computation time) it is possible to efficiently generate a collection of exactly that many patterns that is representative of the distribution π. Consequently, since π has a semantic connection to the underlying notion of interestingness, a meaningful allocation of computational resources is guaranteed for any limited time budget.

Figure 1(b) illustrates this alternative approach, which we want to refer to as "controlled repeated pattern sampling". A potentially positive side-effect of this paradigm is that instead of the usual hard constraints it utilizes parameter-free soft constraints [5]. Hence, the user is freed of the often troublesome task of finding appropriate hard threshold parameters such as a minimum frequency threshold.

1.2 Related Work

In contrast to sampling from the input database (see, e.g., [25, 28]), it is a relatively new development in local pattern discovery to sample from the pattern space. In the context of maximal frequent subgraph mining, Chaoji et al. [8] describes a random process that stops after a certain number of steps that is bounded by the maximum number of edges present in an input graph and produces a maximal frequent subgraph. A similar process is already applied in Gunopoulos et al. [18] within a Las Vegas variant of the Dualize and Advance algorithm. More precisely, it is used for the internal randomization of an algorithm with an otherwise deterministic output (all maximal frequent and minimal infrequent sets of a given input database). When applied for the final pattern discovery, however, this random process has the weakness that it provides no control over the generation probabilities of individual patterns.

Several papers propose to overcome this weakness by applying the MCMC method. Boley and Grosskreutz [7] proposes frequent set sampling to approximate the effect of specific minimum frequency thresholds. The proposed algorithm simulates a simple Glauber dynamic on the frequent set lattice: starting with the empty set, in each subsequent time step a single item is either removed or added to the current set. A similar MCMC method is used in Zaki and Al Hasan [2] for generating a representative set of graph patterns. These MCMC methods provide limited control of the generation probabilities, namely of the infinite limit of the state distribution. The worst-case convergence can, however, be exponentially slow in the size of the input database. For sampling from the family of frequent patterns, this problem appears to be inherent: almost uniform frequent pattern sampling can be used for approximate frequent pattern counting, which one can show to be intractable under reasonable complexity assumptions (see [7]). Similar conclusions can be drawn for enumeration spaces defined by linearly scaled versions of the frequency measure such as the standard optimistic estimator for the binomial test quality function in subgroup discovery [27].

In order to avoid this implication of hard-constraint-based pattern discovery (e.g., using a hard frequency threshold), Boley et al. [6] combines pattern space sampling with soft-constraint-based pattern discovery [5]—resulting in the pattern sampling paradigm described in Section 1.1 above. Still, the underlying method is again MCMC-based, and, despite using a more sophisticated chain defined on the closed set lattice of the input database, it shares the practical weaknesses of this technique. The present paper retains the idea of controlled pattern sampling without hard constraints, but proposes novel pattern generation methods that are exact and direct, i.e., they do not involve MCMC process simulation. Consequently, the resulting pattern discovery processes are efficient not only theoretically but also on a wide range of real-world benchmark datasets.

2. PRELIMINARIES

Before going into technical details, we introduce some basic notions and notation. For a finite set X we denote by $\mathcal{P}(X)$ its power set and by $u(X)$ the uniform probability distribution on X. Moreover, for positive weights $w : X \to \mathbb{R}^+$ let $w(X)$ denote the distribution on X arising from normalizing w, i.e., the distribution described by $x \mapsto w(x)/\sum_{x' \in X} w(x)$—assuming that there is an $x \in X$ with $w(x) > 0$.

A binary **dataset** $\mathcal{D}$ over some finite **ground set** E is a bag (multiset) of sets, called **data records**, $D_1, \ldots, D_m$ each of which is a subset of $E = \{e_, \ldots, e_n\}$. The **size** of $\mathcal{D}$, denoted by $\|\mathcal{D}\|$, is defined as the sum of all its data record sizes $\sum_{D \in \mathcal{D}} = |D|$. Inspired by the application of market basket analysis the elements of E are often referred to as "items". More generally, one can think of E as a set of binary features describing the data records. In particular, a categorical data table can easily be represented as a binary dataset by choosing the ground set as consisting of all attribute/value equality expressions that can be formed from the table. More precisely, a **categorical data table** T consisting of m data row vectors $d_1, \ldots, d_m$ with $d_i = (d_i(1), \ldots, d_i(n))$ can be represented by the dataset $\mathcal{D}_T = \{D_1, \ldots, D_m\}$ with $D_i = \{(j, v) : d_i(j) = v\}$ over ground set

$$E_T = \{(j, d_i(j)) \colon 1 \leq i \leq m, 1 \leq j \leq n\} \ .$$

For a given dataset $\mathcal{D}$ over E, the **pattern space** (or pattern *language*) $\mathcal{L}(\mathcal{D})$ considered in this paper is the power set $\mathcal{P}(E)$ of the features and its elements are interpreted conjunctively. That is, the *local* data portion described by a set $F \subseteq E$, called the **support (set)** of F in $\mathcal{D}$ and denoted $\mathcal{D}[F]$, is defined as the multiset of all data records from $\mathcal{D}$ that contain *all* elements of F, i.e., $\mathcal{D}[F] = \{D \in \mathcal{D} \colon D \supseteq F\}$.

An **interestingness measure** for a pattern language $\mathcal{L}(\cdot)$ is a function

$$q \colon \{(\mathcal{D}, x) \colon \mathcal{D} \text{ a binary dataset}, x \in \mathcal{L}(\mathcal{D})\} \to \mathbb{R} \ .$$

However, often there is a fixed dataset that is clear from the context. In such cases—and if we want to simplify the notation—we just write q as a unary function $q(\cdot) = q(\mathcal{D}, \cdot)$ and omit the first argument. The most basic measures for set patterns are the **support (count)**, i.e., the size of its support set $q_{\text{supp}}(\mathcal{D}, F) = |\mathcal{D}[F]|$ and the **frequency**, i.e., the relative size of its support with respect to the total number of data records $q_{\text{freq}}(\mathcal{D}, F) = |\mathcal{D}[F]| \,/\, |\mathcal{D}|$. For a frequency threshold $t \in [0, 1]$ a set is called ***t*-frequent** (w.r.t. $\mathcal{D}$) if $q_{\text{freq}}(\mathcal{D}, F) \geq t$. A further measure considered here is the **area function** [15] $q_{\text{area}}(\mathcal{D}, F) = |F|\,|\mathcal{D}[F]|$. Intuitively, the area of a set corresponds to the number of 1 entries of the submatrix (of the binary matrix representation of $\mathcal{D}$) consisting of the columns corresponding to F and the rows corresponding to $\mathcal{D}[F]$.

All measures defined so far are unsupervised measures in the sense that they rely on no further information but the dataset itself. In contrast, there are so-called supervised descriptive rule induction techniques that rely on additional information in the form of **class labels** $l(D) \in C = \{c_1, \ldots, c_k\}$ associated to each data record $D \in \mathcal{D}$. For $c \in C$ we denote by $\mathcal{D}_c$ the data portion labeled c, i.e., $\mathcal{D}_c = \{D \in \mathcal{D} \colon l(D) = c\}$. Examples for this setting are emerging pattern mining [12] and contrast set mining [3], where one is interested in patterns having a high support difference between the positive and the negative portion of the data records, or subgroup discovery [27], where one searches for patterns with a high distributional unusualness of these labels on their support set. An important special case is the case of binary labels, i.e., $C = \{\oplus, \ominus\}$. For this case we consider the following **discriminativity measure**

$$q_{\text{disc}}(F) = |\mathcal{D}_{\oplus}[F]|\,|\mathcal{D}_{\ominus} \setminus \mathcal{D}_{\ominus}[F]| \ .$$

A further measure for the discriminative power of a pattern is the **Fisher score** q_{fish}, which is defined for datasets with arbitrary labels C. Intuitively, it measures the relation of the inter-class variance of a feature to its intra-class variances, i.e.,

$$q_{\text{fish}}(F) = \frac{\sum_{c \in C} |\mathcal{D}_c| \left(q_{\text{freq}}(\mathcal{D}_c, F) - q_{\text{freq}}(\mathcal{D}, F)\right)^2}{\sum_{c \in C} \sum_{D \in \mathcal{D}_c} (\delta(D \supseteq F) - q_{\text{freq}}(\mathcal{D}_c, F))^2}$$

where $\delta(D \supseteq F)$ is 1 if $D \supseteq F$ and 0 otherwise. This paper does not present a sampling algorithm for this measure. However, the Fisher score is used for post-processing generated patterns in the context of constructing global classification models.

3. SAMPLING ALGORITHMS

After the introduction of set patterns and interestingness measures, we can now present our sampling procedures. A naive approach for sampling a pattern according to a distribution π is to generate a list $F_1, \ldots, F_N$ of all patterns with $\pi(F) > 0$, draw an $x \in [0, 1]$ uniformly at random, and then return the unique set F_k with $\sum_{i=1}^{k-1} \pi(F_i) \leq x < \sum_{i=1}^{k} \pi(F_i)$. However, the exhaustive enumeration of any non-constant part of the pattern space is precisely what we want to avoid. That is, we are interested in *non-enumerative* sampling algorithms.

Below we give such algorithms for four quality functions: frequency and squared frequency as well as area and discriminativity. The algorithms are inspired by the elementary procedures used in Karp et al. [21] (for estimating the number of satisfying assignments of a DNF formula): in a first step one element of a suitably constructed set of base objects is drawn, and in a second step a sub-object is drawn that is induced by that base object; hence the term "two-step random procedures".

Note that, in contrast to the frequency measures, for the latter two quality functions it is **NP**-hard to find optimal patterns: Finding a set of maximum area for a given input dataset is equivalent to the **NP**-hard problem of computing a biclique with maximum number of edges from a given bipartite graph (see [15]). The same hardness result holds for the discriminativity measure because optimizing area can be linearly reduced to optimizing discriminativity: by setting $\mathcal{D}_\oplus$ to $\mathcal{D}$ and $\mathcal{D}_\ominus$ to $\{E \setminus \{e\} \colon e \in E\}$ we get

$$q_{\text{disc}}(\mathcal{D}_\oplus \cup \mathcal{D}_\ominus, F) = |\mathcal{D}[F]| \left(|E| - |E| + |F|\right) = q_{\text{area}}(\mathcal{D}, F)$$

for all $F \subseteq E$ because with this construction of the negative data portion we have $|\mathcal{D}_\ominus[F]| = |E| - |F|$.

3.1 Frequency and Area

Algorithm 1 Frequency-based Sampling

```
Require: dataset $\mathcal{D}$ over ground set $E$,
Returns: random set $R \sim q_{\text{freq}}(\mathcal{P}(E)) = q_{\text{supp}}(\mathcal{P}(E))$

1. let weights $w$ be defined by $w(D) = 2^{|D|}$ for all $D \in \mathcal{D}$
2. draw $D \sim w(\mathcal{D})$
3. return $R \sim u(\mathcal{P}(D))$
```

We start with sampling according to frequency and area, both of which can be achieved by very similar linear time algorithms. The key insight for frequency-based sampling, i.e., $\pi = q_{\text{freq}}(\mathcal{P}(E))$, is that random experiments are good in reproducing frequent events. Namely, if we look at a pattern that is supported by a random data record we are likely to observe a pattern that is supported by many data records altogether. This intuition leads to a two-step non-enumerative sampling routine (see Algorithm 1), which is as fast as it is simple: First select a data record of the input dataset randomly with a probability proportional to the size of its power set. Then return a uniformly sampled subset of that data record. Using the size of the power set in the first step is important, as otherwise the sampling routine would be biased towards sets occuring in small data records. As noted in Proposition 1 below, the random set resulting from combining both steps follows the desired distribution.

Regarding the computational complexity of the sampling algorithm we can observe that it is indeed efficient: if one has knowledge of the numbers $|D|$ for all data records $D \in \mathcal{D}$ and, moreover, has index access to all data records, a single random set can be produced in time $O(\log |\mathcal{D}| + |E|)$ (the two terms correspond to producing a random number for drawing a data record in step 1 and to drawing one of its subsets in step 2, respectively). Both requirements can be achieved via a single initial pass over the dataset. Thus, we have the following proposition.

PROPOSITION 1. *On input dataset $\mathcal{D}$ over E, a family of k realizations of the random set $\mathbf{R} \sim q_{\text{freq}}(\mathcal{P}(E))$ can be generated in time $O(\|\mathcal{D}\| + k(|E| + \ln |\mathcal{D}|))$.*

PROOF. Let Z be the normalizing constant $\sum_{F \subseteq E} |\mathcal{D}[F]|$ and $\mathbf{D}$ denote the random data record that is drawn in step 2 of the algorithm. For the probability distribution of the returned random set we have

$$\begin{aligned}
\mathbb{P}[\mathbf{R} = R] &= \sum_{D \in \mathcal{D}} \mathbb{P}[\mathbf{R} = R \wedge \mathbf{D} = D] \\
&= \sum_{D \in \mathcal{D}[R]} \frac{1}{2^{|D|}} \frac{2^{|D|}}{Z} \\
&= \frac{|\mathcal{D}[R]|}{Z} = \frac{q_{\text{supp}}(\mathcal{D}, R)}{Z}
\end{aligned}$$

with a normalizing $Z = \sum_{D \in \mathcal{D}} 2^{|D|}$ (which is equal to the desired $\sum_{F \subseteq E} |\mathcal{D}[F]|$). □

Algorithm 2 Area-based Sampling

```
Require: dataset $\mathcal{D}$ over ground set $E$ with $\|\mathcal{D}\| > 0$,
Returns: random set $R \sim q_{\text{area}}(\mathcal{P}(E))$

1. let weights $w$ be defined for all $D \in \mathcal{D}$ by
       $w(D) = |D| \, 2^{|D|-1}$
2. draw $D \sim w(\mathcal{D})$
3. return $R \sim s(\mathcal{P}(D))$ with $s(F) = |F|$
```

Sampling according to area, i.e., $\pi = q_{\text{area}}(\mathcal{P}(E))$, can be achieved via a slight modification of frequency-based sampling: in step two, instead of drawing a subset uniformly from a data record, draw a subset with probability proportional to its size. The latter step can be implemented, e.g., by first drawing a size s with probability proportional to $\binom{|D|}{s}$ and then by uniformly drawing from D a subset of size s. As a side effect, this modification affects the normalization constants and in particular the data record weights of step one. Since for the sum of all subset sizes of a data record D we have

$$\sum_{F \subseteq D} |F| = |D| \, 2^{|D|-1} \ ,$$

the data record weights need to be modified accordingly. The resulting pseudo-code is given in Algorithm 2. Again, after all weights have been computed via an initial pass over the data, an arbitrary number of random sets can be produced in time $O(\log |\mathcal{D}| + |E|)$. Hence, with a similar proof as for Proposition 1 we can conclude:

PROPOSITION 2. *On input dataset $\mathcal{D}$ over E, a family of k realizations of the random set $\boldsymbol{R} \sim q_{\text{area}}(\mathcal{P}(E))$ can be generated in time $O(\|\mathcal{D}\| + k(|E| + \ln |\mathcal{D}|))$.*

It is important to note that area can be replaced by *weighted* area relatively easy without changing the asymptotic complexity—where weighted area is defined as

$$q_{\text{ware}}(F) = \left(\sum_{e \in F} w(e) \right) \left(\sum_{D \in \mathcal{D}[F]} w(D) \right)$$

for a set of positive weights $w \colon (E \cup \mathcal{D}) \to \mathbb{R}_+$. The same holds for weighted frequency. In this paper, however, for the sake of simplicity we only consider the unweighted case.

3.2 Discriminativity and Squared Frequency

Algorithm 3 Discriminativity-based Sampling

```
Require: binary labeled dataset $\mathcal{D}$ over ground set $E$
         such that there is an $F \subseteq E$ with $q_{\text{disc}}(F) > 0$
Returns: random set $R \sim q_{\text{disc}}(\mathcal{P}(E))$

1. let weights $w$ be defined by
       $w(D_\oplus, D_\ominus) = (2^{|D_\oplus \setminus D_\ominus|} - 1) 2^{|D_\oplus \cap D_\ominus|}$
   for all $(D_\oplus, D_\ominus) \in \mathcal{D}_\oplus \times \mathcal{D}_\ominus$
2. draw $(D_\oplus, D_\ominus) \sim w(\mathcal{D}_\oplus \times \mathcal{D}_\ominus)$
3. return $R = (F \cup F')$ with
       $F \sim u(\mathcal{P}(D_\oplus \setminus D_\ominus) \setminus \emptyset)$ and $F' \sim u(\mathcal{P}(D_\oplus \cap D_\ominus))$
```

In order to design a sampling procedure for discriminativity, i.e., $\pi = q_{\text{disc}}(\mathcal{P}(E))$, we can lift the principle of frequency-based sampling to a random experiment that is a little more complicated and has the following intuition: if we look at a pattern that is supported by a random positive data record and *not* supported by a random negative data record, we are likely to observe a pattern that is altogether supported by many positive data records and only few negative data records, i.e., we are likely to observe a pattern with a relatively high discriminativity score. Again, in order to control the resulting distribution, it is necessary to consider a pair of data records $(D_\oplus, D_\ominus)$ with a probability equal to the number of sets $F \subseteq E$ with $F \subseteq D_\oplus$ and $F \not\subseteq D_\ominus$. This implies that the increased expressivity of discriminativity compared to frequency comes at a price: due to the necessity of weight computation for all pairs of positive and negative data records, we end up with a quadratic preprocessing phase. Algorithm 3 contains all the details of the resulting sampling procedure and leads to the following result.

PROPOSITION 3. *Let $\mathcal{D}$ be a binary labeled input dataset over ground set E such that there is a set $F \subseteq E$ with $q_{\text{disc}}(\mathcal{D}, F) > 0$. A family of k realizations of the random set $\boldsymbol{R} \sim q_{\text{disc}}(\mathcal{P}(E))$ can be generated in time $O(\|\mathcal{D}\|^2 + k(|E| + \ln^2 |\mathcal{D}|))$.*

PROOF. Let $\mathbf{R}$ denote the random set returned in step 3 of the algorithm and $\mathbf{D}_\oplus$, $\mathbf{D}_\ominus$ the data records drawn in step 2. Moreover, for $D \in \mathcal{D}_\oplus$ and $D' \in \mathcal{D}_\ominus$ let $\delta(D, D')$ denote the family of all sets $F \subseteq E$ that are supported by D but not supported by D'. We can rewrite this definition as

$$\begin{aligned} &\delta(D, D') \\ =&\{F \subseteq E \colon \ F \subseteq D, F \not\subseteq D'\} \\ =&\{F \cup F' : \emptyset \subset F \subseteq (D \setminus D'), F' \subseteq (D \cap D')\} \ . \end{aligned}$$

This form shows that the weights $w(\cdot,\cdot)$ assigned in step 1 are equivalent to $|\delta(\cdot,\cdot)|$ and, moreover, that $\mathbf{R}$ is a set drawn uniformly from $\delta(\mathbf{D}_\oplus, \mathbf{D}_\ominus)$. With this we can conclude similar to the previous algorithms that

$$\begin{aligned} \mathbb{P}[\mathbf{R} = F] &= \sum_{D \in \mathcal{D}_\oplus} \sum_{D' \in \mathcal{D}_\ominus} \mathbb{P}[\mathbf{R} = F, \mathbf{D}_\oplus = D, \mathbf{D}_\ominus = D'] \\ &= \sum_{D, D' \in \delta^{-1}[F]} \frac{1}{|\delta(D, D')|} \frac{w(D, D')}{Z} \\ &= \frac{1}{Z} \left| \{(D, D') \in \mathcal{D}_\oplus \times \mathcal{D}_\ominus \colon \ D \supseteq F, D' \not\supseteq F\} \right| \\ &= \frac{1}{Z} \left| \mathcal{D}_\oplus[F] \right| \left(|\mathcal{D}_\ominus| - \mathcal{D}_\ominus[F] \right) \end{aligned}$$

with $Z = \sum_{D, D' \in \mathcal{D}_\oplus \times \mathcal{D}_\ominus} |\delta(D, D')| = \sum_{F \subseteq E} q_{\text{disc}}(F)$ as required. □

Algorithm 4 Squared-frequency-based Sampling

```
Require: dataset $\mathcal{D}$ over ground set $E$,
Returns: random set $F \sim q^2_{\text{freq}}(\mathcal{P}(E)) = q^2_{\text{supp}}(\mathcal{P}(E))$

1. let weights $w$ be defined by
       $w(D_1, D_2) = 2^{|D_1 \cap D_2|}$
   for all $(D_1, D_2) \in \mathcal{D} \times \mathcal{D}$
2. draw $(D_1, D_2) \sim w(\mathcal{D} \times \mathcal{D})$
3. return $F \sim u(\mathcal{P}(D_1 \cap D_2))$
```

It is straightforward to see that the approach of drawing two data records can also be used to implement other potentially interesting distributions that can be expressed as the product of two support counts. A basic example is squared frequency. In order to achieve this distribution, one can consider a uniformly[2] drawn subset of two random data records, i.e., a subset of their intersection. The resulting pseudo-code with appropriate pairwise weights is given in Algorithm 4. Closely following the proof of Proposition 3 this algorithm can be used to show another proposition.

PROPOSITION 4. *On input dataset $\mathcal{D}$ over E, a family of k realizations of the random set $\boldsymbol{R} \sim q^2_{\text{freq}}(\mathcal{P}(E))$ can be generated in time $O(\|\mathcal{D}\|^2 + k(|E| + \ln^2 |\mathcal{D}|))$.*

[2] For sampling according to the squared area function, draw a subset with probabilities proportional to its squared size instead of uniformly.

In principle, one can design sampling algorithms for an abitrary power c of the frequency measure by drawing a subset from c random data records. However, the resulting time complexity for computing the weights for each c-tuple of data records gets out of hand quickly.

4. EVALUATION

The sampling procedures presented in the previous section are provably efficient and correct, i.e., their randomized output follows the specified distributions. Beside their practical scalability, it remains to be evaluated how useful these distributions are in the context of local pattern discovery. It is inherently difficult to evaluate pattern discovery methods for exploratory data analysis. There one aims to find *interesting* patterns based on notions of interestingness that are often user-subjective. Hence, a sophisticated experimental design would be required. Here we resort to the other branch of local pattern discovery applications, i.e., pattern-based global model construction, which allows us to simply use Fisher score as primary interestingness measure and accuracy as objective evaluation metric for the overall process.

dataset	class	nm/ct	items	rows	density
autos	7	15/10	135	205	0.190
balance-scale	3	4/0	20	625	0.250
breast-cancer	2	0/9	51	286	0.195
colic	3	7/15	84	366	0.271
credit-a	2	6/9	71	690	0.223
diabetes	2	8/0	40	768	0.225
glass	7	9/0	45	214	0.222
heart-c	5	6/7	49	303	0.285
heart-h	5	6/7	46	294	0.246
heart-statlog	2	13/0	55	270	0.254
hepatitis	2	6/13	55	155	0.344
hypothyroid	4	7/22	78	3772	0.364
iris	3	4/0	20	150	0.250
lymph	4	3/15	57	148	0.333
prim.-tumor	22	0/17	37	339	0.468
sonar	2	60/0	300	208	0.203
tic-tac-toe	2	0/9	27	958	0.370
vehicle	4	18/0	90	846	0.211
zoo	7	1/16	135	101	0.133

Table 1: Benchmark datasets with basic statistics: number of *class*es $|C|$, number of numerical and categorical columns (*nm/ct*), number $|E_T|$ of *items* in corresponding binary dataset, number of *rows*, *density* $|\mathcal{D}_T|\,|E_T|\,/\|\mathcal{D}_T\|$.

In our experiments we use a variety of databases from the UCI machine learning repository [14] listed in Table 1. In order to apply the pattern discovery algorithms, binary datasets are created from these databases by first converting them into categorical datatables using five bucket frequency discretization of all numeric data columns, and then by considering the corresponding binary datasets (using attribute / value pairs of categorical attributes as items; see Section 2). Implementations of the sampling algorithms are available in the software section of `http://www-kd.iai.uni-bonn.de`.

4.1 Predictive Performance

We start with experiments evaluating the sampling algorithms in the context of pattern-based classification. Here one aims to improve classification accuracy by enriching given labeled training data with pattern-based features. We closely follow the framework of Cheng et al. [9]. In a nutshell it consists of three basic steps: extraction of a collection of patterns (which are subsequently considered as features of the data records supporting them), feature selection based on Fisher score and pattern redundancy, and classification, for which we use a linear support vector machine.

In more detail, for an input data table T with corresponding binary dataset $\mathcal{D}_T = \mathcal{D}$ and class labels $C = \{1, \ldots, c\}$ we consider **pattern collections** $\mathcal{C}$, $\mathcal{R}_{\text{frq}}$, $\mathcal{R}_{\text{sfr}}$, $\mathcal{R}_{\text{dcr}}$ that are based on deterministic top-k frequent closed sets (these can be efficiently produced, e.g., by the algorithm of Pietracaprina and Vandin [24]) and sampling according to frequency, squared frequency, and discriminativity, respectively. Area-based sampling is not considered here, because it is not designed to provide good features for classification. Each pattern collection $\mathcal{P}$ is the union of c collections $\mathcal{P}_i$ based on the class labels $i \in C$ where the size of $\mathcal{P}_i$ is proportional to the number of training examples of that class, i.e., $|\mathcal{P}_i| = \alpha\,|\mathcal{D}_i|$. Consequently, the size of the complete collection is $\alpha\,|\mathcal{D}|$, which is linear in the input size. The parameter α is set to 32 independently of the method and the dataset. This setting is a compromise between efficiency and stability of the pattern selection process (described below) as captured by the average Fisher score of finally selected features. See Figure 2 for an illustration on three exemplary datasets.

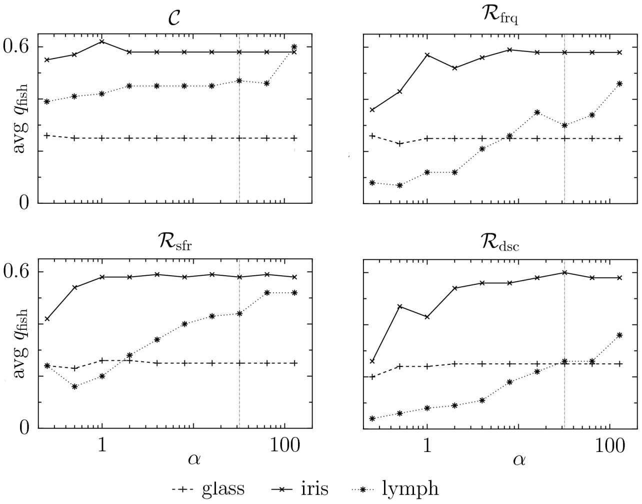


Figure 2: Average Fisher score of collections produced by methods for increasing values of α.

The **feature selection** step for a pattern collection $\mathcal{P}$ is then performed as follows: initialize $\mathcal{P}_0 = \emptyset$ and consider all sets $F_1, \ldots, F_l \in \mathcal{P}$ having $q_{\text{freq}}(\mathcal{D}, \cdot) \geq 0.05$ in descending order of their Fisher score $q_{\text{fish}}(\mathcal{D}, \cdot)$. Select a pattern F_i if the number of previously uncovered data records that it covers is at least 1% of the dataset, i.e.,

$$\left| \mathcal{D}[F_i] \setminus \bigcup_{F \in \mathcal{P}_{i-1}} \mathcal{D}[F] \right| \geq 0.01\,|\mathcal{D}|$$

where $\mathcal{P}_{i-1}$ is the family of sets already selected when considering F_i. The selection process is stopped either after all patterns have been considered or if $\mathcal{P}_i$ covers the dataset completely, i.e., $\bigcup_{F \in \mathcal{P}_i} \mathcal{D}[F] = \mathcal{D}$. This is a very simple instantiation of the framework of Cheng et al. with the condition of 1% coverage improvement acting as binary redundancy measure. A further deviation is that in our case the remaining patterns are not reordered after each selection step. While the original procedure constructs potentially more powerful feature collections, this simplified version is much faster: the time complexity is only linear instead of quadratic in the number of input features. For the final pattern collection $\mathcal{P}'$ the original data table T is then augmented by binary attributes corresponding to the elements of $F_1, \ldots, F_k \in \mathcal{P}'$. That is, the augmented table T' has $n + |\mathcal{P}'|$ columns with rows defined by

$$t'_i(j) = \begin{cases} t_i(j), & \text{if } j \leq n \\ 1, & \text{if } j > n \text{ and } D_i \supseteq F_{j-n} \\ 0, & \text{otherwise} \end{cases}$$

where D_i is the data record of $\mathcal{D}_T$ corresponding the the i-th row of T.

dataset	plain	$\mathcal{C}$	q_{freq}	q^2_{freq}	q_{disc}
autos	76.31	75.79	75.26	74.74	75.79
balance-scale	85.09	85.09	85.09	85.09	85.09
breast-cancer	70.37	**71.48**	**72.59**	**72.22**	**72.96**
colic	65.71	**67.14**	**67.43**	**66.86**	**66.00**
credit-a	85.44	84.56	85.00	85.15	85.00
diabetis	74.40	**75.20**	73.87	73.87	73.87
glass	63.00	**64.00**	**65.50**	**68.00**	**68.50**
heart-c	81.73	79.66	**82.76**	**82.76**	**82.07**
heart-h	82.50	81.79	81.43	81.43	81.79
heart-statlog	81.16	80.39	**83.08**	**83.46**	**82.69**
hepatitis	80.71	80.71	80.71	**85.00**	79.29
hypothyroid	97.55	97.53	97.50	**97.63**	**97.58**
iris	89.29	88.57	**91.43**	**91.43**	**91.43**
lymph	83.08	**85.39**	**85.39**	**83.85**	**84.62**
primary-tumor	40.94	**44.38**	**45.94**	**47.50**	**47.81**
sonar	78.42	77.37	78.42	78.42	78.42
tic-tac-toe	76.28	**92.55**	**96.60**	**96.60**	**94.15**
vehicle	67.95	**70.48**	**69.64**	**69.64**	**70.24**
zoo	91.11	91.11	**93.33**	91.11	**92.22**

Table 2: Accuracy of SVM classification on plain database and with feature enrichment based on frequent sets and sampled pattern collections (bold face indicates outperformed baseline).

The linear SVM of the LIBSVM software is used as **classifier**—wrapped in an optimization layer for its regularization parameter c. That is, the training set is first used to determine the optimal regularization parameter $c \in \{2^i : i = -5, -3, \ldots, 14\}$ using 5-fold cross-validation and then a model is trained with the optimal parameter using the whole training set. The complete workflow is validated using 10-fold cross-validation for all pattern collections simultaneously.

Table 2 contains the **results**. As one can observe, for some of the datasets pattern-based classification provides a substantial accuracy boost compared to the plain SVM baseline ("tic-tac-toe", "primary-tumor", "glass"). For others there is still an improvement but to a lesser extend, or the results are roughly identical. Most importantly, the sampling methods perform stronger than exhaustive top-k mining on a majority of datasets. A Wilcoxon signed ranks test (see [11]) for our $N = 19$ databases reveals that pattern-based classification with each of the random set collections outperforms the plain SVM significantly at the 5%-level (t-values of 42, 28.5, and 45.5 respectively; critical value 46). Moreover, although all random set collections are lying ahead of the frequent set collection on our test databases, it is not significantly outperformed by any of them. We can conclude that pattern-based classification based on all tested sampling algorithms is likely to outperform the plain SVM, and is unlikely to be inferior to standard frequent-pattern-based classification.

4.2 Scalability and Effectivity

Having evaluated the quality of the sampled patterns we now turn to scalability and effectivity studies.[3] The theoretical potential of the direct sampling procedures is already indicated by the guarantees of Propositions 1-4. In particular for frequency and area-based sampling they suggest applicability on larger to large-scale datasets. Below we investigate to what degree this potential can be realized in practice.

The practical advantages of direct pattern sampling over the MCMC pattern sampling algorithms are very clear: while for instance the closed set Markov chain simulation [6] takes seven minutes to sample a closed set from a 30K row subset of the US census dataset, frequency-based direct sampling takes only 0.067 seconds to draw the first sample including preprocessing—afterwards additional samples can be produced in milliseconds.

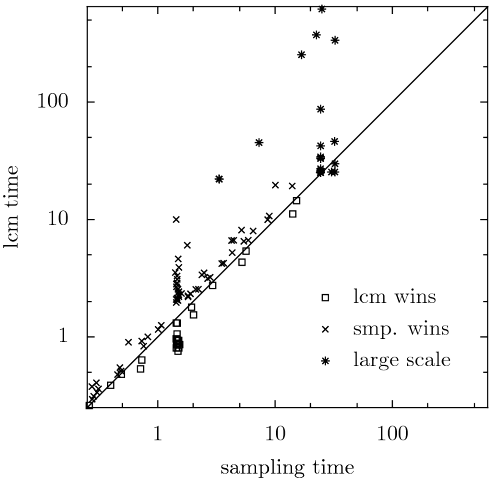


Figure 4: Time of lcm versus time of frequency-based sampling for an identical number of patterns.

For comparing direct sampling to exhaustive search algorithms there is a very large number of possible contenders in the literature, and, in principle, each of them requires an individual comparative study. For this paper we resort to a rather general setting: we compare the computation time of the sampling algorithms with that of the linear closed frequent set mining algorithm (lcm) of Uno et al. [26] *per pattern*. For direct sampling this constitutes a worst-case setting because within the group of exhaustive methods frequent set mining algorithms usually produce the largest output per time unit and lcm is known to be among the fastest of them (winner of the FIMI contest [4]). In addition to the datasets used for the predictive performance study, we also consider several of the larger benchmark datasets of the

[3] Used hardware was a 2.67GHz 2 core CPU with 8GB RAM.

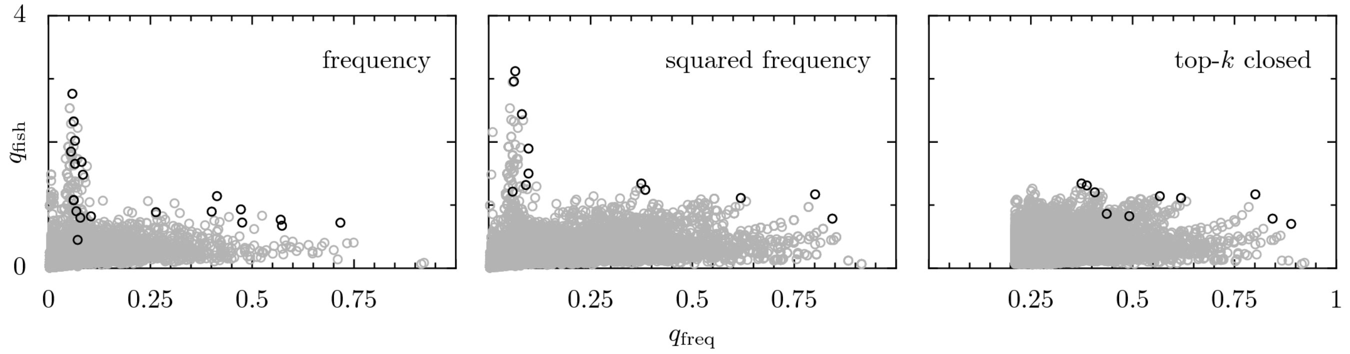


Figure 3: Pattern collections generated for "primary-tumor" by frequency-based sampling, squared-frequency-based sampling, and top-k closed frequent set listing; plotted according to frequency (x-axis) and fisher score (y-axis); highlighted patterns are selected by feature selection procedure.

FIMI workshop, including the 1GB sized "webdocs", and a 500MB random dataset.

The results are presented as log/log scatter-plot in Figure 4. One can observe that for most configurations lcm and the frequency-based sampling generate their patterns in approximately equal time with the majority of wins going to the sampling. However, focusing on the configurations with large-scale datasets (star symbol) reveals that the sampling algorithm can substantially outperform lcm. For "webdocs" this includes a speed-up factor of 10, for the random dataset even one of 25. We can conclude that frequency-based sampling can substantially outperform (closed) frequent set listing on large datasets and it performs equally well with slight advantage on small-scale data.

While the time performance of frequency-based sampling is also representative for area-based sampling, this is not true for the two sampling procedures with quadratic time weight computation phase. For almost all of the relatively small datasets of Tab. 1, this weight computation time is only marginal, i.e., less than 0.2 seconds. For large-scale datasets, however, the quadratic complexity is prohibitive; as already indicated on the 3772 row dataset "hypothyroid" where weight computation takes 13 seconds. After data record weights are computed, the performance is essentially equal to frequency-based sampling.

	$\mathcal{C}$	q_{freq}	q^2_{freq}
diabetes	**0.023** (0.066)	**0.023** (0.066)	**0.023** (0.066)
hypo.	0.016 (0.0190)	0.356 (3.613)	**0.414** (5.420)
lymph	**0.446** (0.580)	0.148 (0.352)	0.372 (0.580)
tumor	1.035 (1.270)	1.552 (3.630)	**1.725** (4.336)

Table 3: Average (maximum) Fisher score of features selected from pattern collections.

The scalability experiments above compared computation times for an equal number of generated patterns. This leads to the question of effectivity, i.e., what can be said about the quality per pattern? To this end, we again use the feature selection procedure from Section 4.1 and consider the quality of features within top-k frequent closed sets and random set families of the same size with respect to our measure of primary interest, the Fisher score; again using the setting of $k = 32\|\mathcal{D}\|$. This time we do not split the datasets according to the label, in order to avoid different values for global and local frequency of patterns (hence, we ignore discriminativity, which inherently requires splits). Table 3 shows the results of four datasets that are representative for different constellations: For datasets that contain high-frequency patterns with high discriminative power, as for instance for "lymph", the top-k paradigm is highly effective. Often, however, the patterns with high Fisher score are of relatively low frequency and are hidden (from the perspective of top-k frequent set listing) underneath a large pattern set of high frequency but low discriminativity. Such a constellation can for instance be observed for "primary-tumor" (see also Fig. 3).

5. CONCLUSION

We introduced four simple direct sampling procedures that generate random set patterns distributed according to frequency, squared frequency, area, and a discriminativity measure for binary labels. All procedures come with tight theoretical performance guarantees. Moreover, we described experimental studies demonstrating that the produced patterns are as useful as frequent pattern collections for pattern-based classification, and that direct sampling can compete with and often even outperform the fastest exhaustive mining algorithms when generating an equal number of patterns.

In the context of pattern-based classification there is a large amount of pattern discovery approaches that range from optimistic-estimator-based best-first-search algorithms [23] to methods interweaving model training and pattern discovery [10, 13]. Although such algorithms typically traverse much less patterns per time unit as lcm, their search is more directed towards high quality patterns. This motivates an in-depth comparative study with such methods potentially leading to more sophisticated usage of the sampling algorithms (e.g., applying it within model training just as the cited approaches do with exhaustive mining).

That said, pattern sampling as a paradigm is in no way restricted to pattern-based classification, and should also be evaluated for other in particular unsupervised model construction tasks as well as for exploratory data analysis. This is likely to motivate further variants of pattern sampling procedures. An example is the introduction of column and row weights to the interestingness measure in order to model subjective interest in certain parts of the input data or to decrease the probability of re-discovering redundant patterns.

Acknowledgements

Part of this work was supported by the German Science Foundation (DFG) under 'GA 1615/1-1' as well as by the European Commission under 'ICT-FP7-LIFT-255951' and 'EU-FP7-250527'.

The authors would like to thank Andrea Pietracaprina and Fabio Vandin for providing the top-k closed frequent set mining software.

6. REFERENCES

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. 1996.

[2] M. Al Hasan and M. J. Zaki. Output space sampling for graph patterns. *PVLDB*, 2(1):730–741, 2009.

[3] S. D. Bay and M. J. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, 2001.

[4] R. Bayardo, B. Goethals, and M. J. Zaki, editors. *Proc. of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, 2004*, volume 126 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[5] S. Bistarelli and F. Bonchi. Soft constraint based pattern mining. *Data and Knowledge Engineering*, 62(1):118–137, 2007.

[6] M. Boley, T. Gärtner, and H. Grosskreutz. Formal concept sampling for counting and threshold-free local pattern mining. In *Proc. of the SIAM Int. Conf. on Data Mining (SDM 2010)*, pages 177–188, 2010.

[7] M. Boley and H. Grosskreutz. Approximating the number of frequent sets in dense data. *Knowledge and Information Systems*, 21(1):65–89, 2009.

[8] V. Chaoji, M. A. Hasan, S. Salem, J. Besson, and M. J. Zaki. Origami: A novel and effective approach for mining representative orthogonal graph patterns. *Statistical Analysis and Data Mining*, 1(2):67–84, 2008.

[9] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *Proc. of the 23rd Int. Conf. on Data Engineering (ICDE 2007)*, pages 716–725, 2007.

[10] H. Cheng, X. Yan, J. Han, and P. S. Yu. Direct discriminative pattern mining for effective classification. In *Proc. of the 24th Int. Conf. on Data Engineering (ICDE 2008)*, pages 169–178, 2008.

[11] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[12] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. 5th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '99)*, pages 43–52. ACM, 1999.

[13] W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. S. Yu, and O. Verscheure. Direct mining of discriminative and essential frequent patterns via model-based search tree. In *Proc. of the 14th Int. Conf. on Knowledge Discovery and Data Mining (KDD '08*, pages 230–238, 2008.

[14] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[15] F. Geerts, B. Goethals, and T. Mielikäinen. Tiling databases. In *Proc. 7th Int. Discovery Science Conf.*, volume 3245 of *LNCS*, pages 278–289. Springer, 2004.

[16] L. A. Goldberg. *Efficient algorithms for listing combinatorial structures*. Cambridge University Press, New York, NY, USA, 1993.

[17] H. Grosskreutz, S. Rüping, and S. Wrobel. Tight optimistic estimates for fast subgroup discovery. In *Proc. of the European Conf. on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD 2008), Part I*, volume 5211 of *LNCS*, pages 440–456, 2008.

[18] D. Gunopulos, H. Mannila, and S. Saluja. Discovering all most specific sentences by randomized algorithms. In *Proc. of 6th Int. Conf. of Database Theory (ICDT '97)*, volume 1186 of *LNCS*, pages 215–229, 1997.

[19] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.*, 8(1):53–87, 2004.

[20] D. J. Hand. Pattern detection and discovery. In *Proc. of the ESF Exploratory Workshop on Pattern Detection and Discovery*, volume 2447 of *LNCS*, pages 1–12. Springer, 2002.

[21] R. M. Karp, M. Luby, and N. Madras. Monte-carlo approximation algorithms for enumeration problems. *J. Algorithms*, 10(3):429–448, 1989.

[22] A. J. Knobbe, B. Crémilleux, J. Fürnkranz, and M. Scholz. From local patterns to global models: the lego approach to data mining. In *From Local Patterns to Global Models: Proceedings of the ECML/PKDD 2008 Workshop (LEGO '08)*, 2008.

[23] S. Morishita and J. Sese. Traversing itemset lattice with statistical metric pruning. In *Proc. of 19th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems (PODS)*, pages 226–236, 2000.

[24] A. Pietracaprina and F. Vandin. Efficient incremental mining of top-k frequent closed itemsets. In *Proc. of 10th International Discovery Science Conference (DS 2007)*, pages 275–280, 2007.

[25] T. Scheffer and S. Wrobel. Finding the most interesting patterns in a database quickly by using sequential sampling. *Journal of Machine Learning Research*, 3:833–862, 2002.

[26] T. Uno, T. Asai, Y. Uchida, and H. Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In *Proc. of the 7th Int. Discovery Science Conf. (DS 2004)*, volume 3245 of *LNCS*, pages 16–31. Springer, 2004.

[27] S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proc. 1st Euro. Symp. on Principles of Data Mining and Knowledge Discovery (PKDD '97)*, volume 1263 of *LNCS*, pages 78–87. Springer, 1997.

[28] M. J. Zaki, S. Parthasarathy, W. Li, and M. Ogihara. Evaluation of sampling for data mining of association rules. In *Proc. of 7th Workshop on Research Issues in Data Engineering (RIDE)*, 1997.