Frequency-Based Views to Pattern Collections

Taneli Mielikäinen
Department of Computer Science
P.O.Box 26 (Teollisuuskatu 23)
FIN-00014 University of Helsinki, Finland
Taneli.Mielikainen@cs.helsinki.fi

Abstract

Finding frequently occurring patterns from data sets is a central computational task in data mining. In this paper we suggest to focus on pattern frequencies. We advocate frequency simplifications as a complementary approach to structural constraints on patterns. As a special case of the frequency simplifications, we consider discretizing the frequencies. We analyze the worst case error of certain discretization functions and give efficient algorithms minimizing several error functions. In addition, we show that the discretizations can be used to find small approximate condensed representations for the frequent patterns.

1 Introduction

Finding interesting patterns from data sets is the fundamental problem in data mining [24, 25, 34]. This problem is known as the *pattern discovery* problem.

The interestingness of a pattern depends (at least) on the data set and the objective of the data analysis. While surprising patterns could be interesting one day, some other day it might be more interesting to find regularities of the data.

Pattern discovery problem can be formulated as follows: Given a data set d from a set $\mathcal{D}$ of possible data sets, a pattern class $\mathcal{P}$, and an *interestingness predicate*

$$q : \mathcal{P} \times \mathcal{D} \to \{0, 1\},$$

find all interesting patterns, i.e., patterns $p \in \mathcal{P}$ such that $q(p, d) = 1$. (The set $\mathcal{D}$ of possible data sets could be e.g. the set of all binary matrices and thus d would be some binary matrix.) Unfortunately defining the interestingness predicate (or even its vague approximations) is highly nontrivial.

In practice the problem of inventing the interestingness predicate has been partly postponed by relaxing the interestingness predicate to an *interestingness measure*. An interestingness measure can be defined as a mapping

$$\phi : \mathcal{P} \times \mathcal{D} \to [0, 1],$$

where the interestingness of a pattern $p \in \mathcal{P}$ is proportional to $\phi(p)$. Several interestingness predicates can be defined using function ϕ, by deciding that a pattern $p \in \mathcal{P}$ is interesting if and only if $\phi(p) \geq \sigma$ for some (fixed) threshold value $\sigma \in [0, 1]$.

Let us now examine some particular instances of patterns and interestingness measures. Classical examples of patterns in data mining are itemsets and associations rules [1]. A data set d for these patterns is a sequence of itemsets. An *itemset* X is a subset of R. The *frequency* (or support) of an itemset $X \subseteq R$ w.r.t. d is

$$fr(X, d) = \frac{|\{Y \in d : X \subseteq Y\}|}{|d|},$$

i.e., the number of sets Y in d containing X. Usually d and R are assumed to be finite. An itemset X is called σ-frequent if $fr(X, d) \geq \sigma$. An *association rule* is an expression $X \Rightarrow Y$ where $X$and Y are subsets of R. The most popular interestingness measure for an association rule $X \Rightarrow Y$ is its *accuracy* (or confidence) which is defined as

$$acc(X \Rightarrow Y, d) = \frac{fr(X \cup Y, d)}{fr(X, d)}.$$

Also several other classes of patterns and measures of interestingness have been studied (see e.g. [4, 6, 9, 13, 14, 15, 27, 32, 33, 36, 37, 43, 44, 47, 48]).

It is not always easy to define an interestingness measure ϕ in such a way that there would be a threshold value σ such that $\phi(p) \geq \sigma$ for almost all interesting patterns $p \in \mathcal{P}$ and for only very few uninteresting ones. One way to augment the interestingness measure is to define additional constraints for the patterns. There has been several studies especially about structural constraints on patterns [5, 26, 40, 41, 45]. In the case of association rules structural constraints could be e.g. constraints between items. Unfortunately, defining the structural constraints may demand considerable amount of domain expertise.

We suggest a complementary approach to structural constraints which can further restrict and sharpen the set of interesting patterns. It is based on an idea of simplifying the interestingness values $\phi(p)$. It can be seen as a natural generalization of the traditional way of characterizing interesting patterns p of the pattern collection $\mathcal{P}$ by a threshold value σ. It can be easily adapted to various pattern classes since it depends only on interestingness values. When a frequency simplification is used as a postprocessing step, it is suitable for efficient interactive mining.

In addition to the simplification of interestingness values in general, we show that the number of interesting patterns can be reduced by discretizing the interestingness values and then pruning the patterns for which the approximate interestingness values can be inferred from the interestingness values of other interesting patterns. Although there might be more powerful ways to compress the set of interesting patterns, the great virtue of discretizations is their conceptual simplicity: it is quite comprehensible how the discretization simplifies the structure of the collection of the interesting patterns.

The rest of this paper is organized as follows. In Section 2 we define the concept of interestingness simplifications. In Section 3 we consider a special case of interestingness simplifications, namely discretization of frequencies. We analyze also the efficiency of some discretization functions and give efficient algorithms for optimal discretizations w.r.t. several loss functions. In Section 4 we show that discretizations can be used to in condensed representations of classes of interesting patterns.

To avoid unnecessary complications, for the rest of this paper we talk about frequencies instead of interestingness values.

2 Frequency-Based Views

A simplification of frequencies is a mapping

$$\psi : [0,1] \to I,$$

where I is the collection of all subintervals of $[0,1]$, i.e.,

$$I = \{[a,b],(a,b],[a,b),(a,b) \subseteq [0,1]\}.$$

For example, the collection of frequent patterns can be defined using frequency simplification:

$$\psi_\sigma(x) = \begin{cases} [0,\sigma) & x < \sigma \\ x & x \geq \sigma \end{cases}$$

There are several immediate applications of frequency simplifications. They can be used, for example, to focus on some particular frequency-based properties of the pattern class, to compress the set of frequent patterns, to speed up the pattern mining algorithms, to hide the confidential information about the data from the pattern user, to correct and indicate errors in data and in frequent patterns, and to evaluate the stability of frequent patterns.

Although the simplifications in general may require a lot of user interaction, we think that defining simple mappings from the interval $[0,1]$ to its subintervals is more tractable than defining structural constraints in the senses of definability and computational complexity. Examples of simple mappings could be replacing the points in an interval by the interval itself, discretizing an interval with a given discretization function, taking logarithms of frequencies, and affine transformations. The frequency simplification problem is further facilitated by the fact that sometimes the simplification can be decomposed into subproblems as it might be possible to simplify the subintervals separately. (Defining the simplifications becomes even simpler as one remembers the fact that for simplifying frequencies of certain finite set, it is not necessary to consider any other frequencies.)

The frequency simplifications have clearly certain limitations as they focus just on frequencies. For example, interesting and uninteresting patterns can have the same frequency. Nevertheless we hope that frequency simplifications could be useful also in the constrained pattern mining as complementary approach to structural constraints. They might also help in the search for good structural constraints.

3 Discretizing Frequencies

Discretization is an important special case of simplifying frequencies. A discretization of frequencies is a mapping f from $[0,1]$ to a subset of $[0,1]$ that preserves the order of points, i.e., if $x, y \in [0,1]$ and $x \leq y$ then $f(x) \leq f(y)$. Points in the range of the discretization f are called the discretization points of f. To slightly simplify the considerations we assume that the frequencies of the patterns are strictly positive.

A good discretization should not introduce too much error. In the next subsections we prove data independent bounds for certain discretization functions and give (data dependent) algorithms for several loss functions. We consider here the error functions absolute error and approximation ratio.

The absolute error for a point $x \in (0,1]$ w.r.t. discretization f is

$$\delta_a(x,f) = |x - f(x)|$$

and the maximum absolute error for a finite set

$P \subset (0,1]$ of points w.r.t. discretization f is

$$\delta_a(P,f) = \max_{x \in P} \delta_a(x,f).$$

Similarly, the approximation ratio for a point $x \in (0,1]$ is

$$\delta_r(x,f) = \frac{f(x)}{x}$$

and the approximation ratio interval for a finite set $P \subset (0,1]$ is

$$\delta_r(P,f) = \left[\min_{x \in P} \delta_r(x,f), \max_{x \in P} \delta_r(x,f)\right].$$

We denote by $\delta(P,w,f)$ an error caused by the discretization f with a point set $P \subset (0,1]$ and a weight function $w : P \to \mathbb{Q}$.

3.1 Data Independent Discretization. In this subsection we show that functions

$$\begin{aligned} f_a^\epsilon(x) &= \epsilon + 2\epsilon \left\lfloor \frac{x}{2\epsilon} \right\rfloor \quad \text{and} \\ f_r^\epsilon(x) &= (1-\epsilon)^{1+2\left\lfloor \frac{\ln x}{2\ln(1-\epsilon)} \right\rfloor} \end{aligned}$$

are the worst case optimal discretization functions w.r.t. absolute error and approximation ratio, respectively. For association rules we bound the maximum absolute error and the interval of approximation ratios.

Theorem 3.1. *Let $P \subset (0,1]$ be a finite set. Then $\delta_a(P, f_a^\epsilon) \leq \epsilon$ and for any other discretization function f with less discretization points, $\delta_a(P,f) > \epsilon$ for some point set P.*

Proof. For any point $x \in (0,1]$ the absolute error $\delta_a(x, f_a^\epsilon)$ is at most ϵ because

$$2\epsilon \left\lfloor \frac{p}{2\epsilon} \right\rfloor \leq p \leq 2\epsilon + 2\epsilon \left\lfloor \frac{p}{2\epsilon} \right\rfloor$$

and

$$f_a^\epsilon(p) = \epsilon + 2\epsilon \left\lfloor \frac{p}{2\epsilon} \right\rfloor.$$

Each discretization point can cover an interval of length at most 2ϵ. Thus at least $\lceil 1/(2\epsilon) \rceil$ discretization points are needed to cover the whole interval $(0,1]$ and f_a^ϵ uses exactly that many discretization points. □

Corollary 3.1. *Absolute error bound ϵ can be replaced with $(2\lceil 1/(2\epsilon) \rceil)^{-1}$ without increasing the number of discretization points.*

Theorem 3.2. *Let $P \subset (0,1]$ be a finite set. Then $\delta_r(P, f_r^\epsilon) \subseteq [(1-\epsilon), (1-\epsilon)^{-1}]$ and for any other discretization function f with less discretization points $\delta_r(P,f) > \epsilon$ for some point set P.*

Proof. Clearly

$$\left\lfloor \frac{\ln x}{2\ln(1-\epsilon)} \right\rfloor \leq \frac{\ln x}{2\ln(1-\epsilon)} \leq 1 + \left\lfloor \frac{\ln x}{2\ln(1-\epsilon)} \right\rfloor$$

and

$$x = (1-\epsilon)^{\frac{\ln x}{\ln(1-\epsilon)}} = (1-\epsilon)^{2\frac{\ln x}{2\ln(1-\epsilon)}}.$$

Thus

$$(1-\epsilon) \leq \frac{(1-\epsilon)^{1+2\left\lfloor \frac{\ln x}{2\ln(1-\epsilon)} \right\rfloor}}{x} \leq \frac{1}{(1-\epsilon)}.$$

The discretization f_r^ϵ is optimal for any interval $[x,1], x \in (0,1]$ as it defines a partition of $[x,1]$ with maximally long intervals. □

Theorem 3.3. *Let f^ϵ be a discretization with maximum absolute error ϵ. The maximum absolute error in the accuracy of an association rule with frequencies discretized with f^ϵ is at most $1/4$.*

Proof. We have

$$\begin{aligned} f^\epsilon(fr(X \cup Y, d)) &\leq f^\epsilon(fr(X,d)) \quad \text{since} \\ fr(X \cup Y, d) &\leq fr(X,d). \end{aligned}$$

Thus the worst case absolute error is

$$\begin{aligned} & \left| \frac{fr(X \cup Y, d) - \epsilon}{fr(X,d) + \epsilon} - \frac{fr(X \cup Y, d)}{fr(X,d)} \right| \\ = & \frac{\epsilon fr(X,d) + \epsilon fr(X \cup Y, d)}{fr(X,d)^2 + \epsilon fr(X,d)} \\ \leq & \frac{2\epsilon fr(X,d)}{fr(X,d)^2 + \epsilon fr(X,d)} \\ = & \frac{2\epsilon}{fr(X,d) + \epsilon} \end{aligned}$$

As $\min f^\epsilon(p,d) = \epsilon$, the smallest possible value for $fr(X,d)$ is 3ϵ. Thus the maximum absolute error is $1/4$. □

Theorem 3.4. *Let f^ϵ be a discretization with maximum absolute error ϵ. The approximation ratio in the accuracy of an association rule with frequencies discretized with f^ϵ is in the interval $[1/2, \infty)$.*

Proof. If we choose

$$\begin{aligned} fr(X \cup Y, d) &= \delta \qquad \text{and} \\ fr(X,d) &= 2\epsilon - \delta, \end{aligned}$$

and δ becomes arbitrary small then the approximation ratio increases unboundedly.

For the lower bound,

$$\begin{aligned} & \frac{fr(X\cup Y,d)-\epsilon}{fr(X,d)+\epsilon}\left(\frac{fr(X\cup Y,d)}{fr(X,d)}\right)^{-1} \\ = \; & \frac{(fr(X\cup Y,d)-\epsilon)\,fr(X,d)}{(fr(X,d)+\epsilon)\,fr(X\cup Y,d)} \\ = \; & \frac{fr(X\cup Y,d)fr(X,d)-\epsilon fr(X,d)}{fr(X\cup Y,d)fr(X,d)+\epsilon fr(X\cup Y,d)} \\ \geq \; & \frac{fr(X,d)^2-\epsilon fr(X,d)}{fr(X,d)^2+\epsilon fr(X,d)} \\ = \; & \frac{fr(X,d)-\epsilon}{fr(X,d)+\epsilon}=\frac{3\epsilon-\epsilon}{3\epsilon+\epsilon}=\frac{1}{2}. \end{aligned}$$

□

Theorem 3.5. *Let f^ϵ be a discretization with maximum absolute error ϵ. The approximation ratio in the accuracy of an association rule with frequencies discretized with f^ϵ is in the interval $[(1-\epsilon)^2,(1-\epsilon)^{-2}]$.*

Proof. By choosing

$$\begin{aligned} f^\epsilon(fr(X\cup Y,d)) &= (1-\epsilon)fr(X\cup Y,d) \quad \text{and} \\ f^\epsilon(fr(X,d)) &= (1-\epsilon)^{-1}fr(X,d) \end{aligned}$$

we get

$$\frac{(1-\epsilon)fr(X\cup Y,d)}{(1-\epsilon)^{-1}fr(X,d)}=(1-\epsilon)^2\frac{fr(X\cup Y,d)}{fr(X,d)}$$

and by choosing

$$\begin{aligned} f^\epsilon(fr(X\cup Y,d)) &= (1-\epsilon)^{-1}fr(X\cup Y,d) \quad \text{and} \\ f^\epsilon(fr(X,d)) &= (1-\epsilon)fr(X,d) \end{aligned}$$

we get

$$\frac{(1-\epsilon)^{-1}fr(X\cup Y,d)}{(1-\epsilon)fr(X,d)}=(1-\epsilon)^{-2}\frac{fr(X\cup Y,d)}{fr(X,d)}.$$

□

Theorem 3.6. *Let f^ϵ be a discretization with approximation ratio between $(1-\epsilon)^{-1}$ and $(1-\epsilon)$. The maximum absolute error in the accuracy of an association rule with frequencies discretized with f^ϵ is at most $1-(1-\epsilon)^2$.*

Proof. Absolute error in the extreme points are

$$\begin{aligned} & \left|(1-\epsilon)^2\frac{fr(X\cup Y,d)}{fr(X,d)}-\frac{fr(X\cup Y,d)}{fr(X,d)}\right| \\ = \; & \left(1-(1-\epsilon)^2\right)\frac{fr(X\cup Y,d)}{fr(X,d)} \end{aligned}$$

and

$$\begin{aligned} & \left|(1-\epsilon)^{-2}\frac{fr(X\cup Y,d)}{fr(X,d)}-\frac{fr(X\cup Y,d)}{fr(X,d)}\right| \\ = \; & \left((1-\epsilon)^{-2}-1\right)\frac{fr(X\cup Y,d)}{fr(X,d)} \\ = \; & \frac{1-(1-\epsilon)^2}{(1-\epsilon)2}\frac{fr(X\cup Y,d)}{fr(X,d)} \end{aligned}$$

of which the first one is larger. The error is maximized by setting $fr(X,d)=fr(X\cup Y,d)=1$. □

3.2 Empirical Loss Minimization.

The computational problem of data-dependent discretization can be formulated as follows:

Input: A finite subset $P\subset(0,1]\cap\mathbb{Q}, |P|=n$, a weight function $w:P\rightarrow\mathbb{Q}$, a value $\epsilon\in\mathbb{Q}$ and an error function $\delta:P\times(P\rightarrow\mathbb{Q})\times(P\rightarrow(0,1])\rightarrow\mathbb{Q}$.

Output: A discretization $f:P\rightarrow D\subset(0,1]$ such that $|D|$ as small as possible and the error $\delta(P,w,f)$ is at most ϵ.

First we minimize the number of discretization points w.r.t. maximum error and the try to solve the problem for more general loss functions.

Maximum error. Discretization that do not exceed the maximum allowed error ϵ can be interpreted as an interval cover for the point set, i.e., a collection of length 2ϵ subintervals of $(0,1]$ that cover all the points in P. The lengths of the intervals depend on the pointwise error function $\delta(p,f), p\in(0,1]$. W.l.o.g., we describe the algorithms for the maximum absolute error.

A simple solution to the problem is to repeatedly choose the minimum uncovered point x and discretize all the points covered by the interval $[p,p+2\epsilon]$ to value $p+\epsilon$.

```
Interval-Cover($P, \epsilon$)
while $P \neq \emptyset$
    do $d \leftarrow \min P$
        $I \leftarrow \{x \in P : d \leq x \leq d + 2\epsilon\}$
        for $x \in I$
            do $f(x) = d + \epsilon$
        $P \leftarrow P \setminus I$
return $f$
```

Theorem 3.7. *Algorithm* Interval-Cover *finds a discretization f such that $\delta_a(P,f)\leq\epsilon$ and for all discretizations g with $|f(P)|>|g(P)|$ holds $\delta_a(P,g)>\epsilon$.*

Proof. $\delta_a(P,f)\leq\epsilon$ because all points in P are covered and the lengths of the intervals are 2ϵ.

Let $x_1,\ldots,x_m$ be the start points of intervals found by the algorithm. As $x_{i+1}-x_i>2\epsilon, 1\leq i<m$, each x_i

needs own discretization point. Thus $|f(P)| \leq |g(P)|$ for all discretizations g such that $\delta_a(P, g) > \epsilon$. □

The straightforward implementation of Interval-Cover runs in time $O(n^2)$: Let ϵ be smaller than $1/(2n)$ and let $P = \{1, 1-1/n, \ldots, 1/n\}$. At each iteration only one point is removed, i.e., there are n iterations and to find minimum we must inspect each points that is not covered. If ϵ is constant, the time complexity of the algorithm is linear.

The worst case time complexity can be reduced to $O(|P| \log |P|)$ by putting the points in P to a heap for which the minimum can be found in constant time and insertions and deletions can be computed in logarithmic time in $|P|$ [30].

If the point set P is sorted then the problem can be solved in linear time in n by the following simple algorithm:

Prefix-Cover(P, ϵ)

```
P ← Sort(P)
d ← −∞
for i ← 1 to |P|
    do if d < P[i] − ε
           then d ← P[i] + ε
       f(P[i]) = d
return f
```

The efficiency of the algorithm Prefix-Cover depends on sorting. Hence it is efficient if the point set P can be sorted efficiently, e.g., when the point set is almost in order. E.g. the levelwise search of frequent patterns produces the frequencies partially in descending order.

It is possible to discretize the points in linear time even if the set P is not sorted with the following algorithm:

1. Discretize the points in P into bins $0, 1, \ldots, \lfloor 1/(2\epsilon) \rfloor$ corresponding to intervals $[0, 2\epsilon), [2\epsilon, 4\epsilon), \ldots, [2\epsilon\lfloor 1/(2\epsilon) \rfloor, 1), [1, 1]$. Put the bins into a set B.

2. Find a minimal nonempty bin i in B. (A nonempty bin i is called minimal if bin $i-1$ is empty.) Find the smallest point x in the bin i, replace the interval corresponding to the bin i by interval $[x, x+\epsilon]$, and move the points in bin $i+1$ in the interval $[x, x+\epsilon]$ to bin i. Remove the bin i from B.

The algorithm can be made to run in linear time in n: Discretization to bins can be computed in time $O(n)$ using a hash table for the set B [30]. A minimal nonempty bin can then be found amortized constant time. This can be done as follows:

Bin-Cover(P, ϵ)

```
B ← ∅
for x ∈ P
    do i ← ⌊x/(2ε)⌋
       B[i] ← B[i] ∪ {x}
for B[i], B[i] ≠ ∅
    do while B[i − 1] ≠ ∅
           do i ← i − 1
       d ← min B[i]
       while B[i] ≠ ∅ ∨ min B[i] > d + 2ε
           do I ← {x ∈ B[i] : d ≤ x ≤ d + 2ε}
              for x ∈ I
                  do f(x) = d + ε
              B[i] ← B[i] \ I
              if B[i] = ∅
                 then i ← i + 1
                 else d ← min B[i]
return f
```

Theorem 3.8. *If $n/(2\epsilon) \leq 1 - c$ for some constant $0 < c < 1$ that is independent of n and ϵ, then there is no deterministic or randomized algorithm that finds for all P and ϵ a discretization f of minimum cardinality $|f(P)|$ and error $\delta_a(P, f)$ at most ϵ by inspecting at most $n - \lceil 1/c \rceil$ points of P.*

Proof. Clearly, all the points in P can be covered with n discretization points. On the other hand n discretization points cover at most fraction $1 - c$ of the interval $(0, 1]$.

Let $x_1, \ldots, x_k$ be the points that are not inspected by the algorithm. If $x_1, \ldots, x_k$ are chosen uniformly from $(0, 1]$, the probability that x_i is not covered, denoted by $P(X_i = 1)$ is at least c. Let X be number of points that are not covered, i.e., $X \geq \sum_{i=1}^{k} X_i$. The expected number of uncovered points is

$$E(X) \geq \sum_{i=1}^{k} E(X_i) = kE(X_1) \geq kc.$$

If we choose $k \geq 1/c$ then $E(X) \geq 1$, i.e., every deterministic or randomized algorithm that inspect at most $n - \lceil 1/c \rceil$ do not cover at least one point on average w.r.t. the uniform distribution over $(0, 1]$ and thus also in worst case. □

Thus we have given asymptotically optimal algorithms for minimizing the number of discretization points within a given maximum discretization error. The algorithms can be transformed to minimize the error instead of the number of discretization points by a simple application of binary search.

Sum of errors. If we would like to minimize the number of discretization points w.r.t. the sum of errors

$$\sum_{x \in P} w(x)\delta(x, f)$$

instead of the maximum error, the algorithms described above would not necessarily find the optimal solution. Fortunately the problem can be solved in time polynomial in n using dynamic programming [20].

Let $P = \{x_1, \ldots, x_n\}$ be sorted and let $P_{i,j}$ denote set $\{x_k : i \leq k \leq j\}$ of consecutive points. The best discretization point and its error for a set $P_{i,j}$ are denoted by $\mu_{i,j}$ and $\varepsilon_{i,j}$, respectively. The minimum sum of errors of discretization of $P_{1,i}$ with k discretization points and the end position of discretization with $k-1$ discretization points that this discretization extends are denoted by Δ_i^k and ω_i^k, respectively. We can write the following recursive formula for the the optimal sum of errors of $P_{1,i}$ with k discretization points:

$$\Delta_i^k = \begin{cases} \Delta_{1,i} & k = 0 \\ \min_{k \leq j \leq i}\{\Delta_{j-1}^{k-1} + \varepsilon_{j,i}\} & 1 \leq k \leq i \end{cases}$$

Discretization with dynamic programming can be splitted into two subtasks:

1. Compute the matrices μ of discretization points and ε of their errors, such that $\mu_{i,j}$ is the discretization point of the $P_{i,j}$ and $\varepsilon_{i,j}$ is its error.

2. Find the optimal discretizations for $P_{1,i}, 1 \leq i \leq n$, with $k, i \leq k \leq n$, discretization points from the matrices μ and ε using dynamic programming.

As a solution to the first task we can compute any matrices $\varepsilon \in \mathbb{Q}^{n \times n}$ and $\mu \in \mathbb{Q}^{n \times n}$. Some of the possible matrices can be computed in reasonable time in n. For example, the matrices ε and μ for weighted absolute error can be computed in time $O(n^3)$ as follows:

```
VALUATE-ABS($P, w$)
for $i \leftarrow 1$ to $|P|, j \leftarrow i$ to $|P|$
    do $\mu_{i,j} \leftarrow$ MEDIAN$_\delta(P_{i,j}, w)$
       $\varepsilon_{i,j} \leftarrow 0$
       for $k \leftarrow i$ to $j$
           do $\varepsilon_{i,j} \leftarrow \varepsilon_{i,j} + w(x_k)|x_k - \mu_{i,j}|$
return $(\varepsilon, \mu)$
```

The discretizations $\mu_{i,j}$ and the errors $\varepsilon_{i,j}$ for each $P_{i,j}, 1 \leq i \leq j \leq n$, can already be informative summaries of the set P (weighted with w). Besides of that, we can extract from ε and μ the matrices Δ of partial sums of errors and ω of the corresponding discretizations using the following algorithm:

```
TABULATOR($P, \varepsilon, \mu$)
for $i \leftarrow 1$ to $|P|$
    do $\Delta_i^0 = \varepsilon_{1,i}$
for $k \leftarrow 1$ to $|P|$
    do for $i \leftarrow k$ to $|P|$
           do $\Delta_i^k \leftarrow \infty$
       for $i \leftarrow k$ to $|P|, j \leftarrow k$ to $i$
           do $\Delta' \leftarrow \Delta_{j-1}^{k-1} + \varepsilon_{j,i}$
              if $\Delta' < \Delta_i^k$
                 then $\Delta_i^k \leftarrow \Delta'$
                      $\omega_i^k \leftarrow j - 1$
return $(\Delta, \omega)$
```

The time complexity of the algorithm TABULATOR is $O(n^3)$. If we are interested only solutions with at most m discretization points, the algorithm can be made to run in time $O(mn^2)$. Also, it can be easily adapted to other kinds of errors. For some error functions the dynamic programming can be implemented with asymptotically better efficiency guarantees [18]. Also in practice there are several ways to speed up the search.

Even as the matrices Δ and ω might be fascinating by themselves, our primary goal is to extract the optimal discretizations from the matrices. The optimal discretizations with k discretization points can be found in time $O(n)$ as follows:

```
FIND-DISCRETIZATION($P, \Delta, \mu, \omega, k$)
$i \leftarrow |P|$
for $l \leftarrow k$ to 1
    do for $j \leftarrow i$ to $\omega_i^l$
           do $f(x_j) \leftarrow \mu_{\omega_i^j, i}$
       $i \leftarrow \omega_i^j$
return $(f, \Delta_{|P|}^k)$
```

The above algorithm can be easily adapted to find discretization with minimum number of discretization points and error less than ϵ in linear time in n.

Hierarchical discretization. Instead of the best discretization with certain number of discretization points, we could search for a hierarchical discretization with several coarseness rates or number of discretization points. In addition to agglomerative and divisive discretizations, it is possible to find hierarchical discretizations that are optimal w.r.t. to some permutation π of $\{1, \ldots, n\}$ in the following sense: The discretization with pi_1 discretization points is has the minimum error. Inductively, the discretization with π_i discretization points has the minimum error of those discretizations that are compatible with already fixed discretization with π_{i-1} discretization point. The time complexity of a straightforward dynamic programming implementation this idea using the TABULATOR is $O(n^4)$. For certain error functions it is possible to construct hierarchical discretizations that are even levelwise close to optimal [16].

Discretization in higher dimensions. If we consider association rules instead of frequent sets, we have two parameters to be discretized: the frequency and the accuracy of the rule. This can be generalized for patterns with d-dimensional vectors of interestingness values. The problem is equivalent to clustering. Thus in general, the problem is NP-hard, but the approximation algorithms for clustering can be applied [10, 19, 17].

4 Condensation with Discretization

Discretization can be used to simplify the collection of the frequent patterns. The high level schema is the following:

1. Discretize the frequencies of the frequent patterns.
2. Find a condensed representation for the collection of the frequent patterns with discretized frequencies.

Condensed representations of frequent patterns are structures from which the set of frequent patterns can be inferred. There has been many studies on condensed representations of frequent patterns [2, 7, 8, 11, 12, 28, 29, 31, 35, 38, 39, 42, 46, 49]

The condensed representations of the frequent patterns rely on regularity properties of the pattern collection and its frequencies. For example, an especially popular condensed representation for frequent sets called *closed frequent sets* is based on the observation that there are exact association rules, i.e., rules $X \Rightarrow Y$ such that $acc(X \Rightarrow Y) = 1$. A frequent set X is *closed* (w.r.t. the data set d) if and only if it has no superset Y such that $fr(X, d) = fr(Y, d)$, i.e., for all supersets Y of X holds $fr(X, d) > fr(Y, d)$.

If the condensed representation of the frequent patterns depends on whether the frequencies of the patterns are equal, the number of discretization points can be used as a crude estimate for the goodness of the discretization.

It might be worth noting that, in addition to simplifying collections of frequent patterns, discretization can be used to make some algorithms for finding condensed representations more efficient.

We studied the usefulness of the discretization for finding a small collection of closed sets that approximate the frequent sets adequately with the following three data sets: the course enrollment data `ilmo` consisting of 3505 rows and 97 attributes, Internet Usage data `internet` consisting of 10104 rows and 10674 attributes, and IPUMS Census data `ipums` consisting of 88443 rows and 39954 attributes. The course enrollment data was collected into the course enrollment sys-

all	closed	maximal	
# of sets	# of sets	# of sets	max error
73413	25618	2114	0.1672

fixed discretization		empirical discretization	
# of sets	max error	# of sets	max error
9597	0.0010	8178	0.0010
3887	0.0050	3212	0.0050
2634	0.0100	2458	0.0100
2251	0.0199	2211	0.0200
2147	0.0399	2139	0.0400
2124	0.0576	2127	0.0600
2122	0.0799	2120	0.0800
2118	0.0963	2118	0.1000

Table 1: `ilmo`, 0.005-frequent sets

all	closed	maximal	
# of sets	# of sets	# of sets	max error
143391	141568	23441	0.4261

fixed discretization		empirical discretization	
# of sets	max error	# of sets	max error
123426	0.0010	123104	0.0010
72211	0.0050	71765	0.0050
54489	0.0100	45944	0.0100
34536	0.0200	31836	0.0200
31587	0.0400	25845	0.0400
26087	0.0600	24399	0.0600
24479	0.0800	23916	0.0800
23960	0.1000	23705	0.1000

Table 2: `internet`, 0.05-frequent sets

all	closed	maximal	
# of sets	# of sets	# of sets	max error
86879	6689	578	0.4000

fixed discretization		empirical discretization	
# of sets	max error	# of sets	max error
3226	0.0010	3242	0.0010
2362	0.0050	2375	0.0050
1776	0.0100	1772	0.0100
1223	0.0200	1225	0.0200
1014	0.0400	841	0.0400
932	0.0600	725	0.0600
711	0.0800	661	0.0800
627	0.1000	627	0.1000

Table 3: `ipums`, 0.2-frequent sets

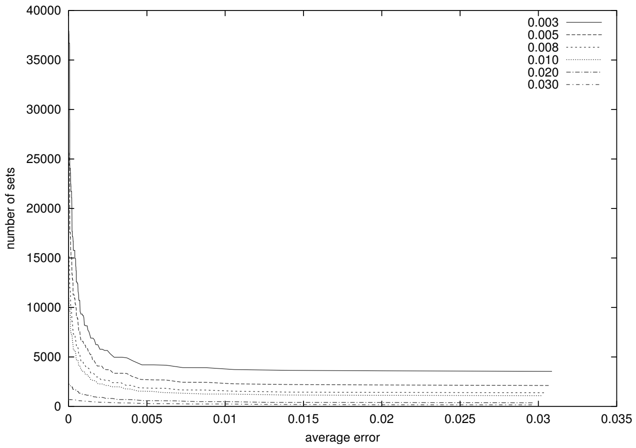

Figure 1: `ilmo`, # of sets vs. average error

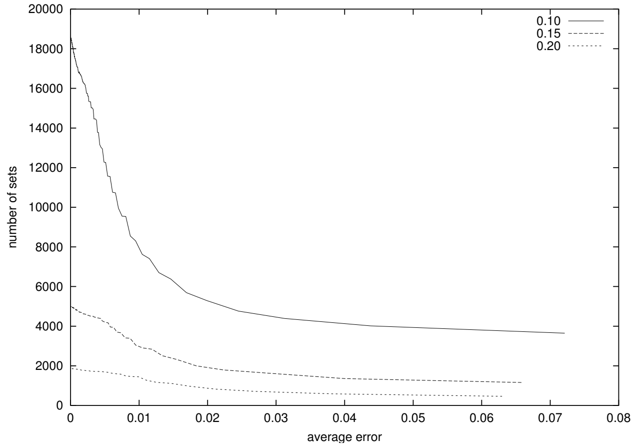

Figure 2: `internet`, # of sets vs. average error

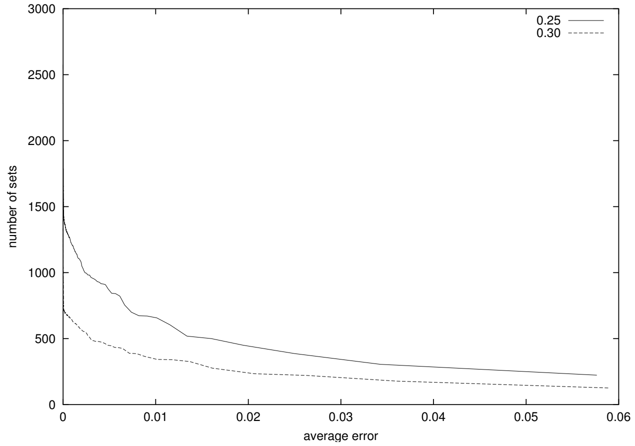

Figure 3: `ipums`, # of sets vs. average error

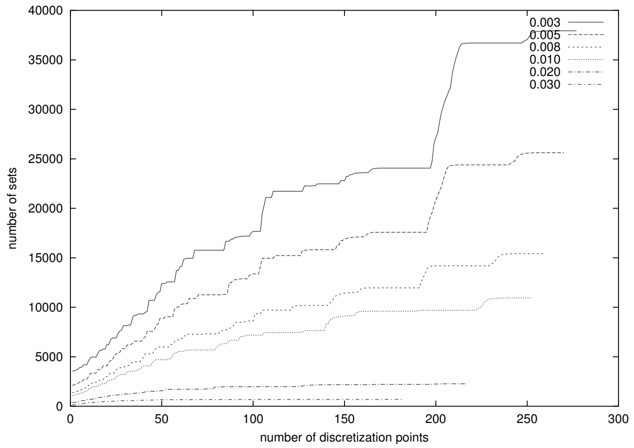

Figure 4: `ilmo`, # of sets vs. # of discretization points

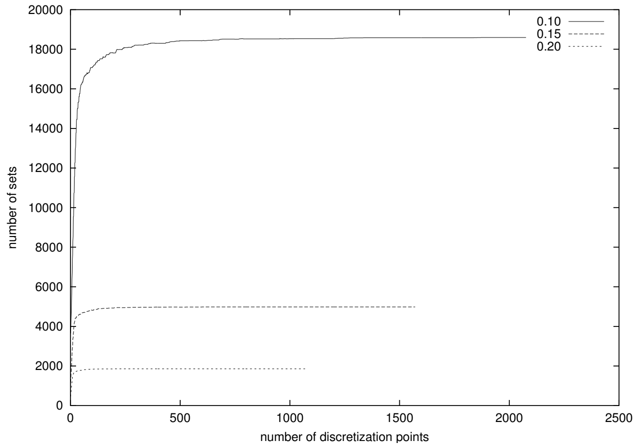

Figure 5: `internet`, # of sets vs. # of discretization points

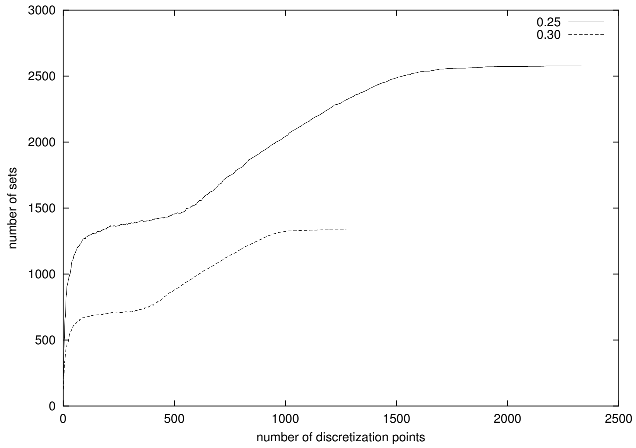

Figure 6: `ipums`, # of sets vs. # of discretization points

tem[1] of Department of Computer Science of University of Helsinki, and the Internet Usage and IPUMS Census data were downloaded from UCI KDD Repository[2]. We computed frequent sets with different minimum frequency thresholds from the data sets using the program `apriori`[3] which is Christian Borgelt's efficient implementation of the Apriori algorithm.

First we discretized the frequencies w.r.t. the maximum absolute error. We were interested especially in the number of closed frequent sets for different error levels. We discretized the frequencies using fixed discretization function f_a^ϵ and algorithm PREFIX-COVER, denoted in the tables 1, 2 and 3 by fixed discretization and empirical discretization, respectively, and then pruned the sets that were not closed w.r.t. the discretization.

Clearly, the maximum number of closed frequent sets for discretized frequencies is the number of closed frequent sets for the original frequencies, i.e., discretization cannot increase the number of closed frequent sets. The minimum number of sets is achieved with one discretization point. Then the closed frequent sets are the frequent sets that have no frequent supersets. They determine the set of frequent sets and are called *maximal frequent sets*. Because of their structural importance, the maximal sets have been studied extensively [3, 6, 21, 22, 23, 36]. The maximum error is obviously minimized by choosing the value of the one discretization point to be the average of the minimum and the maximum frequencies.

We computed the discretizations for several different minimum frequency thresholds and in all cases the number of closed frequent sets clearly decreased. A representative collection of the results of the experiments are shown in Tables 1, 2 and 3. Computing discretization w.r.t. the maximum absolute error were very efficient: more time was consumed even in the detection of the closed frequent sets from all frequent sets.

In addition to minimizing the maximum absolute error, we computed optimal discretizations also w.r.t. average absolute error, using dynamic programming. In particular, we computed the optimal discretizations for each number of discretization points.

The usability of our dynamic programming discretization depends crucially on the number n of different frequencies as its time complexity is $O(n^3)$. Thus we couldn't compute as extensive tests as in the case of maximum absolute error discretization.

For the average error we used uniform distribution over frequent sets. There were differences between the results for different data sets but it is possible to observe that even with quite small number of closed frequent sets and discretization points were able to approximate the frequencies adequately. The number of closed frequent sets vs. the average error are shown in Figures 1, 2 and 3, and the number of closed frequent sets vs. the number of discretization points are shown in Figures 4, 5 and 6.

On the whole, the results are encouraging but more experimentation with different data sets would be useful to better understand the effects of discretization to the structure of the collections of frequent patterns, to detect good heuristic discretization algorithms and to optimize the dynamic programming in practice.

5 Conclusions and Future Work

We have introduced the concept of frequency based views to frequent patterns as a complementary approach to defining structural constraints on patterns. We have given efficient algorithms for discretizing frequencies and demonstrated that the discretization can be used for finding approximate condensed representations of frequent patterns.

There are several open problems related to the frequency simplifications:

- Is it possible (and useful) to generalize frequency simplifications to higher dimensions?
- How should hierarchical frequency simplifications be defined and computed?
- What is the computational complexity of discretization w.r.t. a wider class of loss functions?
- What is the optimal number of discretization points?
- Can the frequency distribution estimated directly from data without actually computing the frequent sets?
- Are there simplification techniques that allow smaller approximate condensed representations of frequent patterns than the discretization?

Furthermore, the condensed representations of pattern collections in general seem to be still an important and relatively open research topic in data mining.

Acknowledgments. I wish to thank Janne Ravantti and Heikki Mannila for helpful comments that significantly improved the manuscript, and Teija Kujala for letting me use the course enrollment data set.

[1] `http://ilmo.cs.helsinki.fi/`

[2] `http://kdd.ics.uci.edu/`

[3] `http://fuzzy.cs.uni-magdeburg.de/~borgelt/apriori/apriori.html`

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, *Fast discovery of association rules. 307-328*, in Advances in Knowledge Discovery and Data Mining, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds., AAAI/MIT Press, 1996, ch. 12, pp. 307–328.

[2] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhai, *Mining frequent patterns with counting inference*, SIGKDD Explorations, 2 (2000), pp. 66–75.

[3] R. J. Bayardo Jr., *Efficiently mining long patterns from databases*, in SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, A. T. Laura M. Haas, ed., ACM, 1998, pp. 85–93.

[4] R. J. Bayardo Jr. and R. Agrawal, *Mining the most interesting rules*, in Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 1999, pp. 145–154.

[5] R. J. Bayardo Jr., R. Agrawal, and D. Gunopulos, *Constraint-based rule mining in large, dense databases*, Data Mining and Knowledge Discovery, 4 (2000), pp. 217–240.

[6] E. Boros, V. Gurvich, L. Khachiyan, and K. Makino, *On the complexity of generating maximal frequent and minimal infrequent sets*, in STACS 2002, H. Alt and A. Ferreira, eds., vol. 2285 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 133–141.

[7] J.-F. Boulicaut and A. Bykowski, *Frequent closures as a concise representation for binary data mining*, in Knowledge Discovery and Data Mining, T. Terano, H. Liu, and A. L. P. Chen, eds., vol. 1805 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2000, pp. 62–73.

[8] J.-F. Boulicaut, A. Bykowski, and C. Rigotti, *Free-sets: a condensed representation of Boolean data for the approximation of frequency queries*, Data Mining and Knowledge Discovery, 7 (2003), pp. 5–22.

[9] S. Brin, R. Motwani, and C. Silverstein, *Beyond market baskets: Generalizing association rules to correlations*, in SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, J. Peckham, ed., ACM, 1997, pp. 265–276.

[10] M. Bădoiu, S. Har-Peled, and P. Indyk, *Approximate clustering via core-sets*, in Proceedings on 34th Annual Symposium on Theory of Computing, ACM, 2002, pp. 250–257.

[11] A. Bykowski and C. Rigotti, *A condensed representation to find frequent patterns*, in Proceedings of the Twenteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, ACM, 2001.

[12] T. Calders and B. Goethals, *Mining all non-derivable frequent itemsets*, in Principles of Data Mining and Knowledge Discovery, T. Elomaa, H. Mannila, and H. Toivonen, eds., vol. 2431 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2002, pp. 74–865.

[13] S. Chakrabarti, S. Sarawagi, and B. Dom, *Mining surprising patterns using temporal description length*, in VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases, A. Gupta, O. Shmueli, and J. Widom, eds., Morgan Kaufmann, 1998, pp. 606–617.

[14] D. Chudova and P. Smyth, *Pattern discovery in sequences under a Markov assumption*, in Proceedings of the Eight International Conference on Knowledge Discovery and Data Mining (KDD-2002), D. Hand, D. Keim, and R. Ng, eds., ACM, 2002.

[15] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. D. Ullman, and C. Yang, *Finding interesting associations without support pruning*, IEEE Transactions on Knowledge and Data Engineering, 13 (2001), pp. 64–78.

[16] S. Dasgupta, *Performance guarantees for hierarchical clustering*, in Computational Learning Theory, J. Kivinen and R. H. Sloan, eds., vol. 2375 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2002, pp. 351–363.

[17] W. F. de la Vega, M. Karpinski, C. Kenyon, and Y. Rabani, *Polynomial time approximation schemes for metric min-sum clustering*, Tech. Rep. 25, Electronic Colloquium on Computational COmplexity, 2002.

[18] T. Elomaa and J. Rousu, *On the computational complexity of optimal multisplitting*, Fundamenta Informaticae, 47 (2001), pp. 35–52.

[19] T. Feder and D. H. Greene, *Optimal algorithms for approximate clustering*, in Proceedings of the twentieth annual ACM Symposium on Theory of Computing, Chicago, Illinois, May 2–4, 1988, ACM, 1988, pp. 434–444.

[20] W. D. Fisher, *On grouping for maximum homogeneity*, Journal of the American Statistical Association, 53 (1958), pp. 789–798.

[21] K. Gouda and M. J. Zaki, *Efficiently mining maximal frequent itemsets*, in Proceedings of the 2001 IEEE International Conference on Data Mining, N. Cercone, T. Y. Lin, and X. Wu, eds., IEEE Computer Society, 2001, pp. 163–170.

[22] D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen, *Data mining, hypergraph transversals, and machine learning*, in Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, ACM, 1997, pp. 209–216.

[23] D. Gunopulos, H. Mannila, and S. Saluja, *Discovering all most specific sentences by randomized algorithms*, in Database Theory - ICDT '97, F. N. Afrati and P. G. Kolaitis, eds., vol. 1186 of Lecture Notes in Computer Science, 1997, pp. 215–229.

[24] D. J. Hand, *Pattern detection and discovery*, in Pattern Detection and Discovery, D. Hand, N. Adams, and R. Bolton, eds., vol. 2447 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2002, pp. 1–12.

[25] D. J. Hand, H. Mannila, and P. Smyth, *Principles*

of Data Mining, MIT Press, 2001.

[26] J. Hipp and U. Güntzler, *Is pushing constraints deeply into the mining algorithms really what we want?*, SIGKDD Explorations, 4 (2002), pp. 50–55.

[27] C. Jermaine, *The computational complexity of high-dimensional correlation search*, in Proceedings of the 2001 IEEE International Conference on Data Mining, N. Cercone, T. Y. Lin, and X. Wu, eds., IEEE Computer Society, 2001, pp. 249–256.

[28] J. Kahan, N. Linial, and A. Samorodnitsky, *Inclusion-exclusion: exact and approximate*, Combinatorica, 16 (1996), pp. 465–477.

[29] D. Kessler and J. Schiff, *Inclusion-exclusion redux*, Electronic Communications in Probability, 7 (2002), pp. 85 – 96.

[30] D. E. Knuth, *Sorting and Seaching*, vol. 3 of The Art of Computer Programming, Addison-Wesley, second ed., 1998.

[31] M. Kryszkiewicz, *Concise representation of frequent patterns based on disjunction-free generators*, in Proceedings of the 2001 IEEE International Conference on Data Mining, N. Cercone, T. Y. Lin, and X. Wu, eds., IEEE Computer Society, 2001, pp. 305–312.

[32] M. Kurakochi and G. Karypis, *Frequent subgraph discovery*, in Proceedings of the 2001 IEEE International Conference on Data Mining, N. Cercone, T. Y. Lin, and X. Wu, eds., IEEE Computer Society, 2001, pp. 313–320.

[33] ———, *Discovering frequent geometric subgraphs*, in Proceedings of the 2002 IEEE International Conference on Data Mining, IEEE Computer Society, 2002.

[34] H. Mannila, *Local and global methods in data mining: Basic techniques and open problems*, in Automata, Languages and Programming, P. Widmayer, F. Triguero, R. Morales, M. Hennessy, S. Eidenbenz, and R. Conejo, eds., vol. 2380 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 57–68.

[35] H. Mannila and H. Toivonen, *Multiple uses of frequent sets and condensed representations*, in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), E. Simoudis, J. Han, and U. M. Fayyad, eds., AAAI Press, 1996, pp. 189–194.

[36] ———, *Levelwise search and borders of theories in knowledge discovery*, Data Mining and Knowledge Discovery, 1 (1997), pp. 241–258.

[37] H. Mannila, H. Toivonen, and A. I. Verkamo, *Discovery of frequent episodes in event sequences*, Data Mining and Knowledge Discovery, 1 (1997), pp. 259–289.

[38] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, *Discovering frequent closed itemsets for association rules*, in Database Theory - ICDT'99, C. Beeri and P. Buneman, eds., vol. 1540 of Lecture Notes in Computer Science, Springer-Verlag, 1999, pp. 398–416.

[39] D. Pavlov, H. Mannila, and P. Smyth, *Beyond independence: probabilistic methods for query approximation on binary transaction data*, IEEE Transactions on Data and Knowledge Engineering, (To appear).

[40] J. Pei and J. Han, *Constrained frequent pattern mining: A pattern-growth view*, SIGKDD Explorations, 4 (2002), pp. 31–39.

[41] J. Pei, J. Han, and L. V. Lashmanan, *Mining frequent itemsets with convertible constraints*, in Proceedings of the 17th International Conference on Data Engineering, IEEE Computer Society, 2001, pp. 433–442.

[42] J. Pei, J. Han, and T. Mao, CLOSET*: An efficient algorithm for mining frequent closed itemsets*, in ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, D. Gunopulos and R. Rastogi, eds., 2000, pp. 21–30.

[43] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal, *Multi-dimensional sequential pattern mining*, in Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, ACM, 2001, pp. 81–88.

[44] P. Smyth and R. M. Goodman, *An information theoretic approach to rule induction from databases*, IEEE Transactions on Knowledge and Data Engineering, 4 (1992), pp. 301–316.

[45] R. Srikant, Q. Vu, and R. Agrawal, *Mining association rules with item constraints*, in Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), D. Heckerman, H. Mannila, and D. Pregibon, eds., AAAI Press, 1997, pp. 67–73.

[46] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal, *Computing iceberg concept lattices with* Titanic, Data & Knowledge Engineering, 42 (2002), pp. 189–222.

[47] P.-N. Tan, V. Kumar, and J. Srivastava, *Selecting the right interestingness measure for association patterns*, in Proceedings of the Eight International Conference on Knowledge Discovery and Data Mining (KDD-2002), D. Hand, D. Keim, and R. Ng, eds., ACM, 2002.

[48] M. J. Zaki, *Efficiently mining frequent trees in a forest*, in Proceedings of the Eight International Conference on Knowledge Discovery and Data Mining (KDD-2002), D. Hand, D. Keim, and R. Ng, eds., ACM, 2002.

[49] M. J. Zaki and C.-J. Hsiao, CHARM*: An efficient algorithms for closed itemset mining*, in Proceedings of the Second SIAM International Conference on Data Mining, R. Grossman, J. Han, V. Kumar, H. Mannila, and R. Motwani, eds., SIAM, 2002.