

Finding All Minimal Infrequent Multi-dimensional Intervals

Khaled M. Elbassioni

Max-Planck-Institut für Informatik, Saarbrücken, Germany;
elbassio@mpi-sb.mpg.de

Abstract. Let \mathcal{D} be a database of transactions on n attributes, where each attribute specifies a (possibly empty) real closed interval $I = [a, b] \subseteq \mathbb{R}$. Given an integer threshold t , a multi-dimensional interval $I = ([a_1, b_1], \dots, [a_n, b_n])$ is called *t-frequent*, if (every component interval of) I is contained in (the corresponding component of) at least t transactions of \mathcal{D} and otherwise, I is said to be *t-infrequent*. We consider the problem of generating all *minimal t-infrequent* multi-dimensional intervals, for a given database \mathcal{D} and threshold t . This problem may arise, for instance, in the generation of association rules for a database of time-dependent transactions. We show that this problem can be solved in quasi-polynomial time. This is established by developing a quasi-polynomial time algorithm for generating maximal independent elements for a set of vectors in the product of lattices of intervals, a result which may be of independent interest. In contrast, the generation problem for *maximal frequent* intervals turns out to be NP-hard.

1 Introduction

Consider a database in which each transaction is associated with a time stamp indicating the start and end times of the transaction. For instance, [15] gives an example of a cellular phone company (or more generally a service provider) which records the time and length for each phone call made by each customer. Then it is useful, for the purpose of both improving the service and making more profit, to determine the intervals of time during which the number of calls exceeds a given threshold (*frequent intervals*), or the intervals of time during which the number of calls lies below some threshold (*infrequent intervals*). Clearly, the property of an interval being infrequent is *monotone*: if an interval I was occupied by less than t customers' phone calls, then the same is true for any interval containing I . Thus we may restrict our attention to *maximal frequent* and *minimal infrequent* intervals. In [15] an algorithm was proposed to enumerate all maximal frequent intervals from a given database.

More generally, one may consider a database of transactions, each of which describes an episode of events appearing over time. For instance, in the above example, we may store in the database the different calls made by each customer in different days. Then an interesting observation, that may be deduced from the database, can take the form "Fewer than 10% of the customers make calls on

Saturday between 1-2 AM, on Sunday between 1-2 AM, and on Monday between 9-10 AM”, or ”At least 60% of the customers who use the service between 5-9 PM in the first 5 days of the month tend also to use the service between 5-9 PM in the last five days”. These examples illustrate the requirement for discovering correlation or *association rules* [1] between occurrences of events over time. As in the case of mining association rules between sets of items in a database (see e.g. [1–3]), a fundamental problem that arises in our case is the generation of frequent and infrequent multi-dimensional intervals (as opposed to frequent and infrequent sets in [1]). As was suggested in [12, 16, 17] for the case of frequent itemsets, it might be much more economical to represent the frequent and infrequent intervals by their boundary, defined as the union of *maximal* frequent and *minimal* infrequent intervals, since typically the number of intervals in such a boundary is much smaller. This motivates us to investigate the complexities of the problems of *jointly* and *separately* generating these two families. It turns out that they exhibit the same behavior as that, discovered in [5], for maximal frequent and minimal infrequent sets. More precisely, let $\mathcal{F}_{\mathcal{D},t}$ and $\mathcal{G}_{\mathcal{D},t}$ denote respectively the families of maximal frequent and minimal infrequent multi-dimensional intervals for a given database \mathcal{D} and an integer threshold t . Then it will be shown that we can generate, in *incremental quasi-polynomial* time ¹, the union $\mathcal{F}_{\mathcal{D},t} \cup \mathcal{G}_{\mathcal{D},t}$ (in some mixed way, and we do not control the order in which the elements of these two families are generated). It will be also illustrated that this result implies that the family of minimal infrequent intervals can also be generated in incremental quasi-polynomial time. Finally, we show also that the problem of incrementally generating the family $\mathcal{F}_{\mathcal{D},t}$ separately is NP-hard in general.

The paper is organized as follows. In the next section, we formally define the problems considered and state our results, and in Section 3, we briefly survey some related work. Following this, Section 4 explains how to view our problem as that of generating maximal frequent/minimal infrequent vectors in the product of lattices of intervals, constructed from the given database \mathcal{D} . In section 5, we reduce the problem of generating minimal infrequent intervals into the so called *dualization* problem in products of lattices of intervals. Finally, In Section 6 we show that this latter problem can be solved in quasi-polynomial time.

2 Problem definition and our results

Let \mathcal{D} be a database of records each of which has n attributes, where each attribute specifies a (possibly empty) real closed interval $I = [a, b] \subseteq \mathbb{R}$, $a, b \in \mathbb{R}$. Denote by \mathbb{B}_n the set of all n -dimensional intervals (or boxes, or hyper-rectangles): $\mathbb{B}_n \stackrel{\text{def}}{=} \{(I_1, \dots, I_n) : I_1, \dots, I_n \text{ are closed intervals of } \mathbb{R}\}$. Henceforth, we shall refer to an n -dimensional interval simply as an interval when it is understood from the context that it has n dimensions. Let us denote by “ \preceq ”

¹ i.e. given a partial list \mathcal{X} of elements that have been already generated, generating a new element requires time $O(k^{\text{poly} \log(k)})$, where $k = n + |\mathcal{D}| + |\mathcal{X}|$.

the precedence relation of the partial order defined on \mathbb{B}_n , that is, given two intervals $I = (I_1, \dots, I_n)$ and (J_1, \dots, J_n) in \mathbb{B}_n , let us say that $I \preceq J$ if and only if $I_i \subseteq J_i$ for all $i = 1, \dots, n$. For $I \in \mathbb{B}_n$, let $S_{\mathcal{D}}(I)$ be the set of transactions of \mathcal{D} that support I , i.e. $S_{\mathcal{D}}(I) \stackrel{\text{def}}{=} \{J \in \mathcal{D} : J \succeq I\}$. Given an integer threshold $0 \leq t \leq |\mathcal{D}|$, an interval I is said to be *t-frequent* if $|S_{\mathcal{D}}(I)| \geq t$ and *maximal t-frequent* if $|S_{\mathcal{D}}(J)| \leq t - 1$ for all $J \succ I$. Similarly an interval I is called *t-infrequent* if $|S_{\mathcal{D}}(I)| \leq t - 1$ and *minimal t-infrequent* if decreasing any interval component of I makes it *t-frequent*. Denote by $\mathcal{F}_{\mathcal{D},t}$ and $\mathcal{G}_{\mathcal{D},t}$ respectively the families of maximal frequent and minimal infrequent multi-dimensional intervals for a given database \mathcal{D} and an integer threshold t , and by $\mathcal{F}_{\mathcal{D},t}^-$ and $\mathcal{G}_{\mathcal{D},t}^+$ the families of *t-frequent* and *t-infrequent* intervals. In this paper, we consider the following problem of incrementally generating all minimal infrequent intervals:

SEP-GEN-($\mathcal{G}_{\mathcal{D},t}, \mathcal{X}$): Given a sublist $\mathcal{X} \subseteq \mathcal{G}_{\mathcal{D},t}$ of minimal *t-infrequent* intervals, either find a new element in $\mathcal{G}_{\mathcal{D},t} \setminus \mathcal{X}$ or declare that the given sublist is complete: $\mathcal{X} = \mathcal{G}_{\mathcal{D},t}$.

Similarly problem *SEP-GEN*-($\mathcal{F}_{\mathcal{D},t}, \mathcal{X}$) of separately generating all maximal *t-frequent* intervals can be defined. We prove the following positive and negative results.

Theorem 1. *Problem SEP-GEN*-($\mathcal{G}_{\mathcal{D},t}, \mathcal{Y}$) can be solved in incremental quasi-polynomial time $k^{O(\log^2 k)}$, where $k = n + |\mathcal{D}| + |\mathcal{Y}|$.

Proposition 1. *There exist instances of problem SEP-GEN*-($\mathcal{F}_{\mathcal{D},t}, \mathcal{X}$) which are NP-hard.

On our way to proving Theorem 1, we also investigate the complexity of the joint generation of minimal infrequent and maximal frequent intervals:

JOINT-GEN($\mathcal{D}, t, \mathcal{X}, \mathcal{Y}$): Given two collections $\mathcal{X} \subseteq \mathcal{F}_{\mathcal{D},t}$ and $\mathcal{Y} \subseteq \mathcal{G}_{\mathcal{D},t}$, either find a new element in $(\mathcal{F}_{\mathcal{D},t} \setminus \mathcal{X}) \cup (\mathcal{G}_{\mathcal{D},t} \setminus \mathcal{Y})$, or declare that these collections are complete: $(\mathcal{X}, \mathcal{Y}) = (\mathcal{F}_{\mathcal{D},t}, \mathcal{G}_{\mathcal{D},t})$.

Theorem 2. *Problem JOINT-GEN*($\mathcal{D}, t, \mathcal{X}, \mathcal{Y}$) can be solved in incremental quasi-polynomial time.

Theorems 1 and 2 indicate that problems *SEP-GEN*-($\mathcal{G}_{\mathcal{D},t}, \mathcal{Y}$) and *JOINT-GEN*($\mathcal{D}, t, \mathcal{X}, \mathcal{Y}$) are, most likely, not NP-hard, since no NP-complete problem is known to be solvable in quasi-polynomial time.

In contrast to these results, we can show that the separate generation problems *SEP-GEN*-($\mathcal{F}_{\mathcal{D},t}^-, \mathcal{X}$) and *SEP-GEN*-($\mathcal{G}_{\mathcal{D},t}^+, \mathcal{X}$) for *t-frequent* and *t-infrequent* intervals can be solved with (amortized) polynomial delay (i.e. the average time required to generate an element of $\mathcal{F}_{\mathcal{D},t}^-$ is bounded by a polynomial in n and $|\mathcal{D}|$). This follows, for instance, from a straightforward generalization of the well-known *A priori* algorithm [3], applied to a product of lattices constructed from the database in a certain way. We omit the proof of the following theorem from this abstract.

Theorem 3. *Given a database \mathcal{D} of transactions each of which is composed of n time intervals, and an integer t , all t -frequent intervals can be computed with amortized delay of $O(n^3|\mathcal{D}|\sum_{i=1}^n|\mathbb{P}_i|)$ per generated interval, and a total number of $O(\sum_{i=1}^n|\mathbb{P}_i|)$ scans of the database, where \mathbb{P}_i is the set of distinct end-points appearing in the i th column of the database. All t -infrequent intervals can be also computed with the same amortized delay.*

We remark that we can also obtain a polynomial delay algorithm for generating $\mathcal{G}_{\mathcal{D},t}^+$ and $\mathcal{F}_{\mathcal{D},t}^-$, but at the cost of increasing the number of scans of the database.

3 Some related work

The problem of enumerating frequent sets arises in the context of mining association rules from binary data, see e.g. [1], mining correlations [6], episodes [18], and many other applications. In [3], an algorithm called *Apriori* was suggested to find all frequent sets from a binary database. Improvements on this algorithm as well as other methods were subsequently proposed, see e.g. [21, 22]. Further work had also considered non-binary databases, for example, databases where items belong to sets of *taxonomies* (or *is-a hierarchies*) [13, 14, 19], and databases with categorical or quantitative attributes [13, 20]. While the Apriori algorithm generates all frequent sets with amortized polynomial delay, it was shown in [5] that the generation of maximal frequent sets is NP-hard. It was also shown in the same paper that the generation of minimal infrequent sets can be solved in incremental quasi-polynomial time. In this paper, we establish similar results for the case of multi-dimensional intervals.

The problem of finding frequent 1-dimensional intervals, in a discrete domain, was considered in [23], where an Apriori-based algorithm was suggested. In [15], an algorithm for finding maximal frequent 1-dimensional intervals, in a continuous domain, was proposed. Another related problem is the generation of empty or sparse boxes in multi-dimensional data, considered in [4, 9]. In this problem, it is required to generate all inclusion-wise maximal hyper-rectangles that contain no point of the database in their interior. A polynomial-time algorithm was presented in [9] to solve the problem in 2-dimensions. This problem was shown to be solvable in quasi-polynomial time in [4] using a similar approach to the one used in this paper. The main difficulty that arises in dealing with frequent intervals is that they may contain some components representing empty intervals, a problem which did not appear in the case of maximal sparse boxes.

4 Embedding the problem into the products of lattices of intervals

4.1 The lattice of intervals

Let $\mathbb{I}_1, \dots, \mathbb{I}_n \subseteq \mathbb{R}^n$ be n sets of real closed intervals. For $i = 1, \dots, n$, let \mathcal{L}_i be the *lattice of intervals* whose elements are all possible intersections and

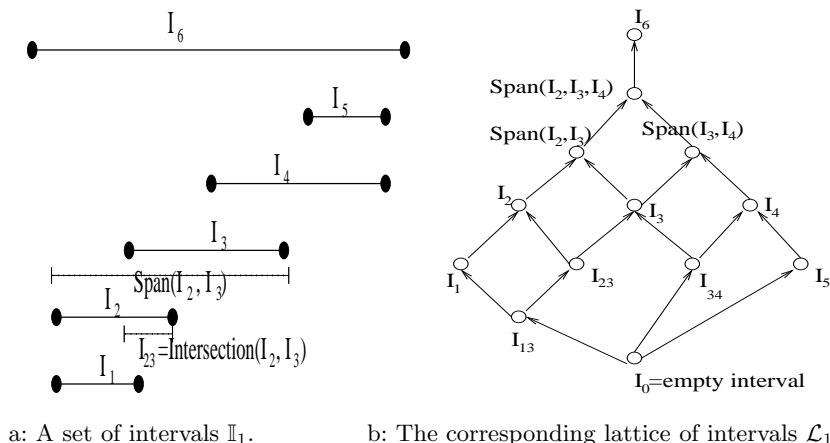


Fig. 1. The lattice of intervals.

spans defined by the intervals in \mathbb{I}_i , and ordered by containment: The meet of any two intervals in \mathcal{L}_i is their *intersection*, and the join is their *span*, i.e., the minimum interval containing both of them (see Figure 1 for an example). Let $\mathcal{L} \stackrel{\text{def}}{=} \mathcal{L}_1 \times \dots \times \mathcal{L}_n$ be the Cartesian product of these n lattices. Throughout we shall denote by \succeq the precedence relation in \mathcal{L} (and also in $\mathcal{L}_1, \dots, \mathcal{L}_n$, i.e. if $p = (p_1, \dots, p_n) \in \mathcal{L}$ and $q = (q_1, \dots, q_n) \in \mathcal{L}$, then $p \preceq q$ in \mathcal{L} if and only if $p_1 \preceq q_1$ in $\mathcal{L}_1, \dots, p_n \preceq q_n$ in \mathcal{L}_n) and use \vee and \wedge to denote the join and meet operators over \mathcal{L} . We shall also denote by $l = (l_1, \dots, l_n)$ and $u = (u_1, \dots, u_n)$ the minimum and maximum elements of \mathcal{L} , respectively. For $x \in \mathcal{L}_i$, denote by x^\perp the set of immediate predecessors of x , i.e.

$$x^\perp = \{y \in \mathcal{L}_i \mid y \prec x, (\nexists z \in \mathcal{L}_i : y \prec z \prec x)\}.$$

Similarly, denote by x^\top the set of immediate successors of x . The following is a simple property satisfied by any lattice of intervals.

Proposition 2. *Let \mathcal{L}_i be a lattice of intervals. Then (i) $|x^\top| \leq 2$ for all $x \neq l_i$ in \mathcal{L}_i , and (ii) $|x^\perp| \leq 2$ for all $x \in \mathcal{L}_i$.*

It is easy to see that $|\mathcal{L}_i| = O(|\mathbb{I}_i|^2)$ and that, if l_i represents the empty interval, then $|l_i^\top| \leq |\mathbb{I}_i|$. Clearly every element in \mathcal{L} represents an n -dimensional interval in \mathbb{B}_n , and the precedence relation in \mathcal{L} corresponds to that in \mathbb{B}_n , i.e. if $p \preceq q$ in \mathcal{L} , then the corresponding intervals $I, J \in \mathbb{B}_n$ satisfy $I \preceq J$. Although \mathcal{L} is a proper subset of \mathbb{B}_n , for our purposes the elements of \mathcal{L} represent the set of all possible *extremal* intervals that are of interest to us, as we shall see in the next subsection.

4.2 Lattices of intervals defined by the database

Given a database of n -dimensional intervals \mathcal{D} , and $i \in [n]$, let $\mathbb{P}_i = \{p_i^1, p_i^2, \dots, p_i^{k_i}\}$ be the set of end-points of intervals appearing in the i th column of \mathcal{D} . Clearly

$k_i \leq 2|\mathcal{D}|$, and assuming that $p_i^1 < p_i^2 < \dots < p_i^{k_i}$, we obtain a set $\mathbb{I}_i = \{[p_i^1, p_i^2], [p_i^2, p_i^3], \dots, [p_i^{k_i-1}, p_i^{k_i}]\}$ of at most $2|\mathcal{D}|$ intervals. Now we let \mathcal{L}_i be the lattice of intervals defined by the set \mathbb{I}_i , for $i = 1, \dots, n$, and let $\mathcal{L} = \mathcal{L}_1 \times \dots \times \mathcal{L}_n$. Clearly, each record in \mathcal{D} appears as an element in \mathcal{L} , i.e. $\mathcal{D} \subseteq \mathcal{L}$. For $x \in \mathcal{L}$, let $S(x) = \{y \in \mathcal{D} \mid y \succeq x\}$. Given an integer threshold t , let us say that an element $x \in \mathcal{L}$ is t -frequent (with respect to \mathcal{D}) if $|S(x)| \geq t$ and maximal t -frequent if $|S(y)| < t$ for all $y \succ x$. Similarly we define t -infrequent and minimal t -infrequent elements of \mathcal{L} .

Now, it is easy to see that the maximal t -frequent elements of \mathcal{L} are in one-to-one correspondence with the maximal t -frequent intervals defined by \mathcal{D} , in the obvious way: if $x = (x_1, \dots, x_n) \in \mathcal{L}$ is a maximal frequent element, then the corresponding interval (I_1, \dots, I_n) (where I_i corresponds to x_i , for $i = 1, \dots, n$) is the corresponding maximal frequent interval. The situation with minimal infrequent intervals is just a bit more complicated: if $x = (x_1, \dots, x_n) \in \mathcal{L}$ is a minimal infrequent element then the corresponding minimal infrequent interval (I_1, \dots, I_n) is computed as follows. For $i = 1, \dots, n$, if $x_i = l_i$ is the minimum element of \mathcal{L}_i , then $I_i = \emptyset$. If x_i represents a point $p_i \in \mathbb{R}$ then $I_i = [p_i, p_i]$. Otherwise, let $[a_i, b_i]$ and $[c_i, d_i]$ be the two intervals corresponding to the two immediate predecessors of x_i , where we assume $a_i < c_i$ (note that $c_i \leq b_i$). If $a_i = b_i$ and $c_i = d_i$ then x_i corresponds to the interval $[a_i, c_i]$ and we have an infinite number of minimal infrequent intervals defined (uniquely) by I_i , namely $I_i = [p_i, p_i]$ for all points p_i in the open interval (a_i, c_i) . Finally, if $a_i < b_i$ and $c_i < d_i$, then $I_i = [c_i - \epsilon, b_i + \epsilon]$ for an infinitesimal constant $\epsilon > 0$. Consequently, in all cases, our problems reduce to finding maximal t -frequent/minimal t -infrequent elements in the lattice product \mathcal{L} .

5 Enumerating minimal infrequent intervals

5.1 Dualization problem in products of lattices of intervals

For a subset $\mathcal{A} \subseteq \mathcal{L}$ of n -dimensional intervals in \mathcal{L} , denote by $\mathcal{A}^+ = \{x \in \mathcal{L} \mid x \succeq a, \text{ for some } a \in \mathcal{A}\}$ and $\mathcal{A}^- = \{x \in \mathcal{L} \mid x \preceq a, \text{ for some } a \in \mathcal{A}\}$. Any element in $\mathcal{L} \setminus \mathcal{A}^+$ is called *independent of \mathcal{A}* . Let $\mathcal{I}(\mathcal{A})$ be the set of all maximal independent elements for \mathcal{A} (also referred to as the *dual of \mathcal{A}*):

$$\mathcal{I}(\mathcal{A}) \stackrel{\text{def}}{=} \{p \in \mathcal{L} \mid p \notin \mathcal{A}^+ \text{ and } (q \in \mathcal{L}, q \succeq p, q \neq p \Rightarrow q \in \mathcal{A}^+)\}.$$

Given $\mathcal{A} \subseteq \mathcal{L}$, we consider the problem of incrementally generating $\mathcal{I}(\mathcal{A})$:

DUAL($\mathcal{L}, \mathcal{A}, \mathcal{B}$): Given subsets $\mathcal{A} \subseteq \mathcal{L}$ and $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$, either find a new element $x \in \mathcal{I}(\mathcal{A}) \setminus \mathcal{B}$, or prove that \mathcal{A} and \mathcal{B} form a dual pair: $\mathcal{B} = \mathcal{I}(\mathcal{A})$.

Clearly, the entire set $\mathcal{I}(\mathcal{A})$ can be generated by initializing $\mathcal{B} = \emptyset$ and iteratively solving the above problem $|\mathcal{I}(\mathcal{A})| + 1$ times. When each lattice $\mathcal{L}_i = \{0, 1\}$, the problem reduces to the well-known hypergraph transversal problem, for which the best-known algorithm [10] runs in time $k^{O(\log k)}$, where $k = |\mathcal{A}| + |\mathcal{B}|$. An

extension of this algorithm, for solving the dualization problem for general lattices was given in [8], and runs in time $\text{poly}(n, \mu(\mathcal{L})) + m^{\gamma(W(\mathcal{L})) \cdot o(\log m)}$, where $m = |\mathcal{A}| + |\mathcal{B}|$, $\gamma(W) = O(W^2 \ln W)$, $\mu = \mu(\mathcal{L}) \stackrel{\text{def}}{=} \max\{|\mathcal{L}_i| : i \in [n]\}$, and $W = W(\mathcal{L}) \stackrel{\text{def}}{=} \max_{i \in [n]} \{W(\mathcal{L}_i)\}$ is the maximum width of the n lattices, i.e. the maximum size of an antichain in the n lattices. Note that for the lattice of intervals \mathcal{L}_i , defined by a set of intervals \mathbb{I}_i , we have $W(\mathcal{L}_i) = O(|\mathbb{I}_i|)$ and $|\mathcal{L}_i| = O(|\mathbb{I}_i|^2)$. Thus, for this special case, the result of [8] gives an *exponential* algorithm in the total number of intervals of $\sum_{i=1}^n |\mathbb{I}_i|$. Here, we shall strengthen this result, in the case of products of lattices of intervals, as follows:

Theorem 4. *Problem $\text{DUAL}(\mathcal{L}, \mathcal{A}, \mathcal{B})$ can be solved in $k^{O(\log^2 k)}$ time, if \mathcal{L} is a product of interval lattices, where $k = |\mathcal{A}| + |\mathcal{B}| + \sum_{i=1}^n |\mathcal{L}_i|$.*

The proof of Theorem 4 will be given in Section 6. In the next section, we show how to use this result to prove Theorems 1 and 2.

5.2 Proof of Theorems 1 and 2

In this section, we argue that the generation problems $\text{JOINT-GEN}(\mathcal{D}, t, \mathcal{X}, \mathcal{Y})$ and $\text{SEP-GEN}(\mathcal{G}_{\mathcal{D}, t}, \mathcal{X})$ reduce in polynomial time to dualization in products of lattices of intervals. For the former problem, the reduction follows from a straightforward generalization of a known result, relating the time complexity of joint generation to that of dualization:

Proposition 3 ([7, 11]). *Problem $\text{JOINT-GEN}(\mathcal{D}, t, \mathcal{X}, \mathcal{Y})$ can be solved in time $\text{poly}(n, |\mathcal{D}|, |\mathcal{X}|, |\mathcal{Y}|) + T_{\text{dual}}$ where T_{dual} denotes the time required to solve problem $\text{DUAL}(\mathcal{L}, \mathcal{A}, \mathcal{B})$.*

For the latter problem, we use Proposition 3 together with a combinatorial Lemma from [5], to show that the family $\mathcal{G}_{\mathcal{D}, t}$ is *uniformly dual-bounded* in the sense that

$$|\mathcal{I}(\mathcal{X}) \cap I(\mathcal{G}_{\mathcal{D}, t})| \leq |\mathcal{D}| |\mathcal{X}|, \quad (1)$$

for any non-empty $\mathcal{X} \subseteq \mathcal{G}_{\mathcal{D}, t}$. Inequality (1) implies that, if we apply joint generation to problem $\text{SEP-GEN}(\mathcal{G}_{\mathcal{D}, t}, \mathcal{X})$, we generate, in addition to the elements of the required family $\mathcal{G}_{\mathcal{D}, t}$, only a polynomial number of unrequired elements belonging to the family $\mathcal{F}_{\mathcal{D}, t} = \mathcal{I}(\mathcal{G}_{\mathcal{D}, t})$. This proves Theorem 1. It remains to show (1), which follows from the following Lemma:

Lemma 1 ([5]). *Let $t \in \mathbb{R}_+$ be a given positive threshold, and $\mathcal{S} \neq \emptyset$ and \mathcal{T} be two families of subsets of a finite set V such that (i) for all $X \in \mathcal{S}$ and $Y \in \mathcal{T}$, we have $|Y| \geq t > |X|$, (ii) for every $Y' \neq Y'' \in \mathcal{T}$ there exists an $X \in \mathcal{S}$ such that $X \supseteq Y' \cap Y''$. Then $|\mathcal{T}| \leq |V| |\mathcal{S}|$.*

To apply the lemma to get (1), let $V = \mathcal{D}$, $\mathcal{S} = \{S(x) : x \in \mathcal{X}\}$ and $\mathcal{T} = \{S(y) : y \in \mathcal{I}(\mathcal{X}) \cap I(\mathcal{G}_{\mathcal{D}, t})\}$, and observe that $|S(y)| \geq t > |S(x)|$ for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y} \stackrel{\text{def}}{=} \mathcal{I}(\mathcal{X}) \cap I(\mathcal{G}_{\mathcal{D}, t})$, since $\mathcal{X} \subseteq \mathcal{G}_{\mathcal{D}, t}$ and $\mathcal{Y} \subseteq \mathcal{F}_{\mathcal{D}, t}$. Furthermore, given two distinct elements $y', y'' \in \mathcal{Y}$, it follows by their maximality in $\mathcal{L} \setminus \mathcal{X}^+$ that $y' \vee y'' \succeq x$, for some $x \in \mathcal{X}$, and thus $S(y') \cap S(y'') = S(y' \vee y'') \subseteq S(x) \in \mathcal{S}$.

6 Dualization algorithm

6.1 Preliminaries

Let $\mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$ where each \mathcal{L}_i is a lattice defined by a set of intervals \mathbb{I}_i . We denote respectively by l_i and u_i the minimum and maximum elements of \mathcal{L}_i . Given two subsets $\mathcal{A} \subseteq \mathcal{L}$, and $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$, we say that \mathcal{B} is *dual to* \mathcal{A} if $\mathcal{B} = \mathcal{I}(\mathcal{A})$. Given any $\mathcal{Q} \subseteq \mathcal{L}$, let us denote by

$$\mathcal{A}(\mathcal{Q}) = \{a \in \mathcal{A} \mid a^+ \cap \mathcal{Q} \neq \emptyset\}, \quad \mathcal{B}(\mathcal{Q}) = \{b \in \mathcal{B} \mid b^- \cap \mathcal{Q} \neq \emptyset\},$$

the subsets of \mathcal{A}, \mathcal{B} whose *ideal* and *filter* respectively intersect \mathcal{Q} .

To solve problem $\text{DUAL}(\mathcal{L}, \mathcal{A}, \mathcal{B})$, we decompose it into a number of smaller subproblems which are solved recursively. In each such subproblem, we start with a sub-lattice $\mathcal{Q} = \mathcal{Q}_1 \times \cdots \times \mathcal{Q}_n \subseteq \mathcal{L}$ (initially $\mathcal{Q} = \mathcal{L}$), and two subsets $\mathcal{A}(\mathcal{Q}) \subseteq \mathcal{A}$ and $\mathcal{B}(\mathcal{Q}) \subseteq \mathcal{B}$, and we want to check whether $\mathcal{A}(\mathcal{Q})$ and $\mathcal{B}(\mathcal{Q})$ are dual in \mathcal{Q} . To estimate the reduction in problem size from one level of the recursion to the next, we measure the change in the "volume" of the problem defined as $v = v(\mathcal{A}, \mathcal{B}, \mathcal{L}) \stackrel{\text{def}}{=} |\mathcal{A}||\mathcal{B}| \sum_{i=1}^n |\mathcal{L}_i|$. Since $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$ is assumed, the following condition holds for the original problem and all subsequent subproblems:

$$a \not\leq b, \quad \text{for all } a \in \mathcal{A}, b \in \mathcal{B}. \quad (2)$$

We stop decomposing a problem when one of the sets \mathcal{A} or \mathcal{B} becomes sufficiently small, in which case the problem is easily solvable in polynomial time.

Let us say that a coordinate $i \in [n]$ is *essential* for an element $a \in \mathcal{A}$ ($b \in \mathcal{B}$), if $a_i \succ l_i$ (respectively, $b_i \prec u_i$). Let us denote by $E(x)$ the set of essential coordinates of a element $x \in \mathcal{A} \cup \mathcal{B}$. The following lemma generalizes a known fact for dual Boolean functions [10].

Lemma 2. *If $\mathcal{A}, \mathcal{B} \subseteq \mathcal{L}$, then either (i) there exists an element $x \in \mathcal{A} \cup \mathcal{B}$ with few essential coordinates: $|E(x)| \leq \log m$, where $m = |\mathcal{A}| + |\mathcal{B}|$, or (ii) an element $x \in \mathcal{L} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$ can be found in polynomial time.*

Lemma 3. *Let \mathcal{A}, \mathcal{B} be a pair of dual subsets of \mathcal{L} with $|\mathcal{A}||\mathcal{B}| \geq 1$. Then there exists a coordinate $i \in [n]$ and a element $z \in \mathcal{L}_i$, such that either:*

- (i) $|\{a \in \mathcal{A} \mid a_i \succeq z\}| \geq 1$ and $|\{b \in \mathcal{B} \mid b_i \not\leq z\}| \geq \frac{|\mathcal{B}|}{\log m}$, or
- (ii) $|\{b \in \mathcal{B} \mid b_i \preceq z\}| \geq 1$ and $|\{a \in \mathcal{A} \mid a_i \not\leq z\}| \geq \frac{|\mathcal{A}|}{\log m}$.

6.2 The algorithm - Proof of Theorem 4

Given subsets $\mathcal{A}, \mathcal{B} \subseteq \mathcal{L}$ that satisfy (2), we proceed as follows:

Step 1. If $\max\{|\mathcal{A}|, |\mathcal{B}|\} \leq 1$, the problem can be solved in $\text{poly}(\sum_{i=1}^n |\mathcal{L}_i|)$ time.

Step 2. For each $k \in [n]$: if $a_k \notin \mathcal{L}_k$ for some $a \in \mathcal{A}$ ($b_k \notin \mathcal{L}_k$ for some $b \in \mathcal{B}$), set $a_k \leftarrow \bigwedge \{x \mid x \in a_k^+ \cap \mathcal{L}_k\}$ (respectively, set $b_k \leftarrow \bigvee \{x \mid x \in b_k^- \cap \mathcal{L}_k\}$).

Step 3. Check if there is an $x \in \mathcal{A} \cup \mathcal{B}$ with at most $\log m$ essential coordinates. If no such x can be found, a new element in $\mathcal{L} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$ can be obtained as described in Lemma 2. Otherwise, we proceed to the next step.

Step 4. If $x = a^\circ \in \mathcal{A}$, find an $i \in E(a^\circ)$, and a $z = a_i^\circ \in \mathcal{L}_i$, for which condition (i) of Lemma 3 is satisfied. Assume without loss of generality that $i = 1$.

In the following steps, we shall decompose \mathcal{L}_1 into two (not necessarily disjoint) sub-lattices \mathcal{L}'_1 and \mathcal{L}''_1 , and let $\mathcal{L}' = \mathcal{L}'_1 \times \mathcal{L}_2 \times \cdots \times \mathcal{L}_n$, and $\mathcal{L}'' = \mathcal{L}''_1 \times \mathcal{L}_2 \times \cdots \times \mathcal{L}_n$ be the sub-lattices of \mathcal{L} induced by this decomposition. It will follow then that \mathcal{A} and \mathcal{B} are dual in \mathcal{L} if and only if

$$\mathcal{A}(\mathcal{L}'), \mathcal{B}(\mathcal{L}') \text{ are dual in } \mathcal{L}', \text{ and } \mathcal{A}(\mathcal{L}''), \mathcal{B}(\mathcal{L}'') \text{ are dual in } \mathcal{L}'', \quad (3)$$

each of which is a dualization problem over the product of lattices of intervals. Note that $\mathcal{A}(\mathcal{L}') = \{a \in \mathcal{A} \mid a_1^+ \cap \mathcal{L}'_1 \neq \emptyset\}$ and $\mathcal{B}(\mathcal{L}') = \{b \in \mathcal{B} \mid b_1^- \cap \mathcal{L}'_1 \neq \emptyset\}$; $\mathcal{A}(\mathcal{L}'')$ and $\mathcal{B}(\mathcal{L}'')$ are defined similarly. Let $\epsilon = 1/\log m$.

Step 4.1. If \mathcal{L}_1 is a total order (chain), then use the following decomposition of \mathcal{L}_1 : $\mathcal{L}'_1 \leftarrow z^+ \cap \mathcal{L}_1$, $\mathcal{L}''_1 \leftarrow \mathcal{L}_1 \setminus \mathcal{L}'_1$. Then $|\mathcal{B}(\mathcal{L}')| \leq (1 - \epsilon)|\mathcal{B}|$ by the selection of a° , and $|\mathcal{A}(\mathcal{L}'')| \leq |\mathcal{A}| - 1$ since $(a^\circ)^+ \cap \mathcal{L}'' = \emptyset$. This reduces the original problem, of volume $v = |\mathcal{A}||\mathcal{B}| \sum_{i=1}^n |\mathcal{L}_i|$ into two subproblems (3) of volumes

$$v' \leq |\mathcal{A}||\mathcal{B}|(1 - \epsilon) \left(\sum_{i=1}^n |\mathcal{L}_i| - 1 \right) \leq (1 - \epsilon)v,$$

$$v'' \leq (|\mathcal{A}| - 1)|\mathcal{B}| \left(\sum_{i=1}^n |\mathcal{L}_i| - 1 \right) \leq v - 1.$$

Step 4.2. Otherwise (\mathcal{L}_1 is not a chain), let w be the *largest* element, with respect to the precedence relation “ \preceq ” on the lattice \mathcal{L}_1 , such that $|w^\perp| = 2$ (see Figure 2-a). Denote respectively by q and y the two immediate predecessors of w . Let $I_q = [a, b]$ and $I_y = [c, d]$ be the two intervals represented by q and y respectively, and assume that $a < c$ (and therefore $b < d$). It is not hard to see that q^- is a lattice of intervals and that $\mathcal{L}_1 \setminus q^-$ is a chain. Now we consider three cases:

- (i) if $z \succ w$, we use the decomposition $\mathcal{L}'_1 \leftarrow z^+ \cap \mathcal{L}_1$, $\mathcal{L}''_1 \leftarrow \mathcal{L}_1 \setminus \mathcal{L}'_1$. Otherwise, the choice of z implies that either cases (ii) or (iii) hold.
- (ii) $|\{b \in \mathcal{B} \mid b_1 \in q^-\}| \geq \frac{\epsilon}{2}|\mathcal{B}|$: in this case, we decompose \mathcal{L}_1 as $\mathcal{L}'_1 \leftarrow \mathcal{L}_1 \cap q^-$, $\mathcal{L}''_1 \leftarrow \mathcal{L}_1 \setminus q^-$.
- (iii) $|\{b \in \mathcal{B} \mid b_1 \in y^-\}| \geq \frac{\epsilon}{2}|\mathcal{B}|$: in this case, we decompose \mathcal{L}_1 as $\mathcal{L}'_1 \leftarrow \mathcal{L}_1 \cap y^-$, $\mathcal{L}''_1 \leftarrow \mathcal{L}_1 \setminus y^-$.

In case (i), we get again that $|\mathcal{B}(\mathcal{L}')| \leq (1 - \epsilon)|\mathcal{B}|$ and $|\mathcal{A}(\mathcal{L}'')| \leq |\mathcal{A}| - 1$, and consequently, the resulting problems are of respective volumes $v' \leq (1 - \epsilon)v$ and $v'' \leq v - 1$. In case (ii), we get $|\mathcal{B}(\mathcal{L}'')| \leq (1 - \epsilon/2)|\mathcal{B}|$ and $|\mathcal{L}'_1| \leq |\mathcal{L}_1| - 1$, and therefore, the resulting two problems have volumes $v' \leq v - 1$ and $v'' \leq (1 - \epsilon/2)v$. Similarly, in case (iii), we get also that $v' \leq v - 1$ and $v'' \leq (1 - \epsilon/2)v$.

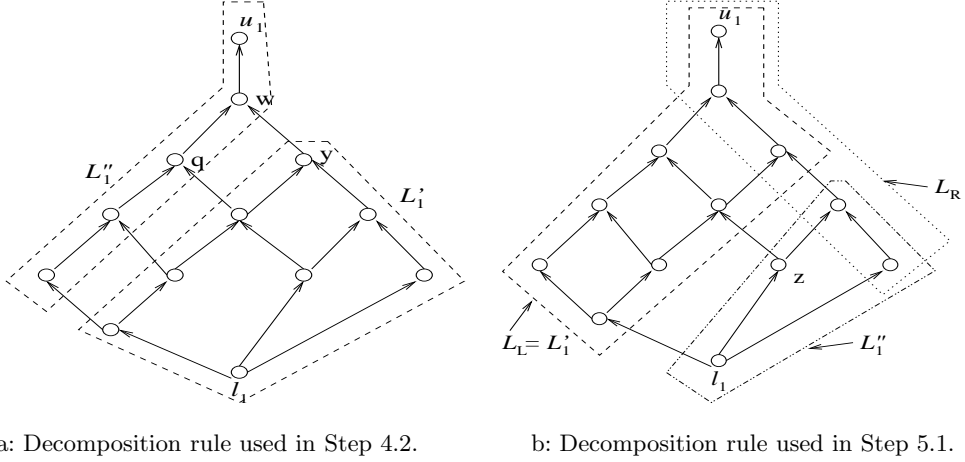


Fig. 2. Decomposing the lattice \mathcal{L}_1 .

Step 5. Now assume that $x = b^o \in \mathcal{B}$, and find an $i \in E(b^o)$, and a $z = b_i^o \in \mathcal{L}_i$, for which condition (ii) of Lemma 3 is satisfied. Assume again, without loss of generality, that $i = 1$.

Step 5.1. If $\min(\mathcal{L}_1) = l_1$ does not represent the empty interval, or $z \succ l_1$, then let $I_z = [a, b]$ be the interval corresponding to z , and let $\mathcal{L}_L \subseteq \mathcal{L}_1$ be the lattice of intervals $I = [c, d]$ for which $c < a$, and likewise, $\mathcal{L}_R \subseteq \mathcal{L}_1$ be the lattice of intervals $I = [e, f]$ for which $f > b$ (see Figure 2-b). Note that these definitions imply that $(\mathcal{L}_L \cup \{l_1\}) \cap z^- = \{l_1\}$, $(\mathcal{L}_R \cup \{l_1\}) \cap z^- = \{l_1\}$, and $\mathcal{L}_L \cup z^- \cup \mathcal{L}_R = \mathcal{L}$. Note also that $\mathcal{L}_L \cup \mathcal{L}_R \neq \emptyset$ since $z \neq u_1 = \max(\mathcal{L}_1)$. By our selection of z , either

$$(i) |\{a \in \mathcal{A} \mid a_1 \in \mathcal{L}_L \setminus \{l_1\}\}| \geq \frac{\epsilon}{2} |\mathcal{A}|, \text{ or } (ii) |\{a \in \mathcal{A} \mid a_1 \in \mathcal{L}_R \setminus \{l_1\}\}| \geq \frac{\epsilon}{2} |\mathcal{A}|.$$

In case (i), we decompose \mathcal{L}_1 as follows: $\mathcal{L}'_1 \leftarrow \mathcal{L}_L$, $\mathcal{L}''_1 \leftarrow (\mathcal{L}_1 \setminus \mathcal{L}'_1) \cup \{l_1\}$. Note that both \mathcal{L}'_1 and \mathcal{L}''_1 are also lattices of intervals, that $|\mathcal{L}'_1| \leq |\mathcal{L}_1| - 1$ since $z \notin \mathcal{L}'_1$, and that $\mathcal{A}(\mathcal{L}'') \leq (1 - \epsilon/2)|\mathcal{A}|$, since $w \not\prec y$ for all $w \in \mathcal{L}'_1 \setminus \{l_1\}$ and $y \in \mathcal{L}''_1 \setminus \{l_1\}$ (indeed, if $I_w = [c, d]$ is the interval corresponding to $w \in \mathcal{L}'_1 \setminus \{l_1\}$ and $I_y = [e, f]$ is the interval corresponding to $y \in \mathcal{L}''_1 \setminus \{l_1\}$, then $c < a$ while $e \geq a$ and thus $I_w \not\subseteq I_y$). Therefore, we get, in this case, two subproblems of volumes $v' \leq v - 1$ and $v'' \leq (1 - \epsilon/2)v$. In case (ii), we let similarly $\mathcal{L}'_1 \leftarrow \mathcal{L}_R$ and $\mathcal{L}''_1 \leftarrow (\mathcal{L}_1 \setminus \mathcal{L}'_1) \cup \{l_1\}$, and we decompose the original problem into two subproblems of volumes $v' \leq v - 1$ and $v'' \leq (1 - \epsilon/2)v$, respectively.

Step 5.2. If $z = l_1$ represents the empty interval, then we let z' be any immediate successor of z , and let \mathcal{L}_L and \mathcal{L}_R be the lattices of intervals as defined in Step 5.1, but with respect to $I_{z'} = [a, b]$ instead of I_z . Note in this case that any interval $[c, d]$ in \mathcal{L}_L either must be strictly to the left of $I_{z'}$, i.e. with $d < a$, or must contain $I_{z'}$. Similarly, any interval $[e, f]$ in \mathcal{L}_R either must be strictly to the right of $I_{z'}$, i.e. with $e > b$, or must contain $I_{z'}$. We consider four cases:

- (i) No interval of \mathbb{I}_1 , corresponding to an element of \mathcal{L}_1 , lies strictly to the left or strictly to the right of $I_{z'}$: the choice of z , in this case, implies that $|\{a \in \mathcal{A} \mid a_1 \succeq z'\}| \geq \epsilon|\mathcal{A}|$. Thus using the decomposition $\mathcal{L}'_1 \leftarrow (z')^+$ and $\mathcal{L}''_1 \leftarrow \{z\}$ results in two subproblems of volumes $v' \leq v-1$ and $v'' \leq (1-\epsilon)v$.
- (ii) No interval lies strictly on the right of $I_{z'}$, but there is at least one that lies strictly to its left: by our choice of z , one of the sets $\{a \in \mathcal{A} : a_1 \in \mathcal{L}_L\}$ or $\{a \in \mathcal{A} \mid a_1 \succeq z'\}$ have size at least $\frac{\epsilon}{2}|\mathcal{A}|$. In the former case we use the decomposition $\mathcal{L}'_1 \leftarrow \mathcal{L}_L$, $\mathcal{L}''_1 \leftarrow \mathcal{L}_1 \setminus \mathcal{L}'_1$, and get two subproblems of volumes $v' \leq v-1$ and $v'' \leq (1-\epsilon)v$. In the latter case, we let \mathcal{L}'_1 be the lattice of intervals lying strictly to the left of $I_{z'}$ and $\mathcal{L}'_1 \leftarrow (z')^+ \cup \{z\}$, and get two subproblems of volumes $v' \leq v-1$ and $v'' \leq (1-\epsilon)v$.
- (iii) No interval lies strictly on the left of $I_{z'}$, but there is at least one that lies strictly to its right: we use a similar decomposition as in case (ii) above.
- (iv) There is at least one interval that lies strictly to the left of $I_{z'}$, and at least one interval strictly to its right: in this case, we know that either $|\{a \in \mathcal{A} \mid a_1 \in \mathcal{L}_L \cup (z')^+\}| \geq \epsilon|\mathcal{A}|/2$ or $|\{a \in \mathcal{A} \mid a_1 \in \mathcal{L}_R \cup (z')^+\}| \geq \epsilon|\mathcal{A}|/2$. In the former case, we use the decomposition $\mathcal{L}'_1 \leftarrow \mathcal{L}_L \cup (z')^+ \cup \{z\}$, $\mathcal{L}''_1 \leftarrow \mathcal{L}_1 \setminus \mathcal{L}'_1 \cup \{z\}$, and in the latter case, we use the decomposition $\mathcal{L}'_1 \leftarrow \mathcal{L}_R \cup (z')^+ \cup \{z\}$, $\mathcal{L}''_1 \leftarrow \mathcal{L}_1 \setminus \mathcal{L}'_1 \cup \{z\}$. In both cases, we get two subproblems of volumes $v' \leq v-1$ and $v'' \leq (1-\epsilon/2)v$.

Thus, in all cases, we apply the algorithm recursively to the resulting subproblems, and obtain the recurrence

$$C(v) \leq 1 + C((1-\epsilon/2)v) + C(v-1),$$

where $C(v)$ is the number of recursive calls required to solve a problem of volume v . Together with $C(v) = 1$, this recurrence evaluates to $C(v) \leq v^{2 \log v/\epsilon}$. Since $v \leq m^2 n \mu$, we get that the running time of the algorithm is $O((m^2 n \mu)^{2 \log m \log(m^2 n \mu)})$.

References

1. R. Agrawal, T. Imielinski and A. Swami, Mining association rules between sets of items in massive databases, in *Proc. the 1993 ACM-SIGMOD Int. Conf. Management of Data*, pp. 207-216.
2. 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* (U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, eds.), pp. 307-328, AAAI Press, Menlo Park, California, 1996.
3. R. Agrawal and R. Srikant, Fast algorithms for mining association rules in large databases, in *Proc. 20th Int. Conf. Very Large Data Bases (VLDB'94)*, pp. 487-499, 1994.
4. E. Boros, K. Elbassioni, V. Gurvich, L. Khachiyan and K. Makino, An Intersection Inequality for Discrete Distributions and Related Generation Problems, in *Automata, Languages and Programming, 30-th International Colloquium, ICALP 2003, Lecture Notes in Computer Science (LNCS) 2719* (2003) pp. 543-555.

5. E. Boros, V. Gurvich, L. Khachiyan and K. Makino, On the complexity of generating maximal frequent and minimal infrequent sets, in *19th Int. Symp. Theoretical Aspects of Computer Science, (STACS)*, March 2002, LNCS 2285, pp. 133–141.
6. S. Brin, R. Motwani, and C. Silverstein, Beyond market basket: Generalizing association rules to correlations, in *Proc. the 1997 ACM-SIGMOD Int. Conf. Management of Data*, pp. 265–276.
7. J. C. Bioch and T. Ibaraki, Complexity of identification and dualization of positive Boolean functions, *Information and Computation* 123 (1995) pp. 50–63.
8. K. Elbassioni, An algorithm for dualization in products of lattices and its applications, in *Proc. 10th Annual European Symposium on Algorithms (ESA 2002)*, LNCS 2461, pp. 424–435, September 2002.
9. J. Edmonds, J. Gryz, D. Liang and R. J. Miller, Mining for empty rectangles in large data sets, in *Proc. 8th Int. Conf. Database Theory (ICDT)*, Jan. 2001, LNCS 1973, pp. 174–188.
10. M. L. Fredman and L. Khachiyan, On the complexity of dualization of monotone disjunctive normal forms, *Journal of Algorithms*, 21 (1996) pp. 618–628.
11. V. Gurvich and L. Khachiyan, On generating the irredundant conjunctive and disjunctive normal forms of monotone Boolean functions, *Discrete Applied Mathematics*, 96-97 (1999) pp. 363–373.
12. D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen, Data mining, hypergraph transversals and machine learning, in *Proc. 16th ACM PODS*, (1997) pp. 12–15.
13. J. Han, Y. Cai and N. Cercone, Data driven discovery of quantitative rules in relational databases, *IEEE Trans. Knowledge and Data Engineering*, Vol. 5 No. 1, (1993) pp. 29-40.
14. J. Han and Y. Fu, Discovery of multiple-level association rules from large databases. In *Proc. 21st Int. Conf. Very Large Data Bases (VLDB'95)*, pp. 420-431, 1995.
15. J.-L. Lin, Mining maximal frequent intervals, in *Proc. 18th Annual ACM Symp. Applied Computing*, pp. 426–431, Melbourne, FL, Sept. 2003.
16. H. Mannila and H. Toivonen, Multiple uses of frequent sets and condensed representations, in *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining* 1996, pp. 189–194.
17. H. Mannila and H. Toivonen, Levelwise search and borders of theories in knowledge discovery, *Data Mining and Knowledge Discovery*, 1(3), 1997, pp. 241–258.
18. H. Mannila, H. Toivonen, and A. I. Verkamo, Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3), 1997, pp. 259–289.
19. R. Srikant and R. Agrawal, Mining generalized association rules. In *Proc. 21st Int. Conf. Very Large Data Bases (VLDB'95)*, pp. 407–419, 1995.
20. R. Srikant and R. Agrawal, Mining quantitative association rules in large relational tables, in *Proc. the 1996 ACM-SIGMOD Int. Conf. Management of Data*, pp. 1–12, 1996.
21. A. Savasere, E. Omiecinski and S. Navathe, An efficient algorithm for mining association rules in large databases, in *Proc. 21st Int. Conf. Very Large Data Bases (VLDB'95)*, pp. 432–444, 1995.
22. H. Toivonen, Sampling large databases for association rules, in *Proc. 22nd Int. Conf. Very Large Data Bases (VLDB'96)*, pp. 134–145, 1996.
23. H.-C. Yu, Efficient data mining for frequent intervals, Master thesis, Department of Information Management, National Taiwan University, Taiwan, July 2002.