Computational Complexity of Finding Highly Co-occurrent Itemsets in Market Basket Databases

Yeon-Dae Kwon† Yasunori Ishihara‡ Shougo Shimizu† Minoru Ito†

{kwon-y,shougo-s,ito}@is.aist-nara.ac.jp
ishihara@ics.es.osaka-u.ac.jp

† Graduate School of Information Science,
Nara Institute of Science and Technology,
8916-5, Takayama, Ikoma, 630-0101, Japan

‡ Department of Informatics and Mathematical Science,
Graduate School of Engineering Science, Osaka University,
1-3, Machikaneyama, Toyonaka, Osaka, 560-8531, Japan

Abstract

Data mining is to analyze all the data in a huge database and to obtain useful information for database users. One of the well-studied problems in data mining is the search for meaningful association rules in a market basket database which contains massive amounts of sales transactions. The problem of mining meaningful association rules is to find all the large itemsets first, and then to construct meaningful association rules from the large itemsets. In our previous work, we have shown that it is NP-complete to decide whether there exists a large itemset with a given size. Also, we have proposed a subclass of databases, called k-sparse databases, for which we can efficiently find all the large itemsets. Intuitively, k-sparsity of a database means that the supports of itemsets of size k or more are sufficiently low in the database.

In this paper, we introduce the notion of (k, c)-sparsity, which is strictly weaker than the k-sparsity in our previous work. The value of c represents a degree of sparsity. Using (k, c)-sparsity, we propose a larger subclass of databases for which we can still efficiently find all the large itemsets. Next, we propose alternative measures to the support. For each measure, an itemset is called highly co-occurrent if the value indicating the correlation among the items exceeds a given threshold. In this paper, we define the highly co-occurrent itemset problem formally as deciding whether there exists a highly co-occurrent itemset with a given size, and show that the problem is NP-complete under whichever measure. Furthermore, based on the notion of (k, c)-sparsity, we propose subclasses of databases for which we can efficiently find all the highly co-occurrent itemsets.

1. Introduction

Recent developments in computer technology have made it possible to analyze all the data in a huge database. *Data mining* is to analyze all the data in a huge database and to obtain useful information for database users. In this paper, we deal with so-called *market basket databases*. A market basket database consists of transactions, where each transaction consists of a set of items. For example, consider a market basket database D_1 shown in Fig. 1. A transaction t_1 indicates that cereal, bacon, eggs, milk, and tea were purchased together by a customer

in a single visit to a store. By examining D_1, we can identify a rule that "if cornflakes is purchased in a transaction, then it is likely that milk will also be purchased in that transaction." Such information is useful for marketing plans such as price management and stock management, also the layout of items.

A set of items is called an *itemset*. An *association rule* is a formula of the form $X \Rightarrow Y$, where X and Y are disjoint itemsets. An intuitive meaning of this formula is that if every item in X is purchased in a transaction, then it is likely that every item in Y will also be purchased. There are two important measures for an association rule introduced by Agrawal et al. [1], called *support* and *confidence*. The *support* of an itemset is the fraction of transactions that contain the itemset. An itemset is called *large* if its support exceeds a given threshold. The *confidence* of a rule $X \Rightarrow Y$ is the fraction of transactions containing X that also contain Y. In addition, we have proposed another measure for an association rule, called *right-hand side* (*rhs*) *size* [7]. For an association rule $X \Rightarrow Y$ to be meaningful, $X \cup Y$ must be large and the confidence of the rule must exceed a given confidence threshold, and also the rule must have a given rhs size.

One of the well-studied problems in data mining is the search for meaningful association rules in a market basket database which contains massive amounts of transactions [1, 3, 8, 9, 11]. The problem of mining meaningful association rules can be decomposed into three subproblems:

1. Find all the large itemsets for a given threshold.
2. Construct rules which exceed the confidence threshold from the large itemsets in step 1. For example, if $\{x, y, z\}$ is a large itemset, then we might check the confidence of $\{x, y\} \Rightarrow \{z\}$, $\{x, z\} \Rightarrow \{y\}$, and $\{y, z\} \Rightarrow \{x\}$.
3. Select rules which have a given rhs size from the rules obtained in step 2.

Having determined the large itemsets, the second and third subproblems are rather straightforward. So a lot of further research has been devoted to speeding up the first subproblem. Although a number of algorithms for computing all the large itemsets have been proposed [1–3, 5, 6, 9, 11], the computational complexity is scarcely discussed. The performances of most of the algorithms are estimated only by empirical evaluation through benchmark tests. On the other hand, in our previous work [7], we have defined the *large itemset problem* formally as deciding whether there exists a large itemset with a given size, and shown that the problem is NP-complete. From this result, it has become clear that finding all the large itemsets (and therefore, all the meaningful association rules) is impossible in polynomial time in the size of a database unless P=NP. Furthermore, we have proposed the notion of k-sparsity of databases [7]. Intuitively, k-sparsity of a database means that the supports of itemsets of size k or more are sufficiently low in the database. Using k-sparsity, we have defined a subclass of databases for which we can efficiently find all the large itemsets.

In this paper, we introduce the notion of (k, c)-*sparsity* of databases, which is strictly weaker than the k-sparsity. The value of c represents a degree of sparsity. Using (k, c)-sparsity, we propose a larger subclass of databases for which we can still efficiently find all the large itemsets.

Several disadvantages of the support-confidence framework have been pointed out in Refs. [4, 5, 10]. For example, the support of an itemset tends to be high if the itemset contains items with high supports, regardless of the correlation among the items. We will explain this in the following example.

Example 1: Consider D_1 shown in Fig. 1. Suppose that the given threshold r is 0.3. Let $Z = \{\text{coffee, eggs}\}$. The number of transactions in D_1 is 6.

t_1	{cereal, bacon, eggs, milk, tea}
t_2	{cornflakes, milk, bread, coffee, eggs}
t_3	{bread, coffee, eggs}
t_4	{cornflakes, milk, bread, coffee}
t_5	{cornflakes, milk, bread, coffee}
t_6	{bread, coffee, eggs}

Fig. 1: A market basket database D_1.

Transactions that contain both coffee and eggs are t_2, t_3, and t_6, and then the support of an itemset Z is $3/6 = 0.5 \geqq r$. Therefore, Z is large. On the other hand, the supports of {coffee} and {eggs} are $5/6$ and $4/6$, respectively. Thus, the support of Z is smaller than the expected value of the support of Z ($5/6 \times 4/6 \approx 0.56$) which is calculated under the assumption that coffee and eggs are purchased independently. That is, it cannot be said that the items in Z have high correlation. □

In this paper, we propose alternative measures to the support, which are defined by the combinations of the aspects such as

- the ratio of the actual value of the support of a given itemset to the expected value of the support of the itemset, based on the assumption of statistical independence,
- the fraction of transactions that do not contain any item in a given itemset,

and so on. Some of these measures are similar to the previous works such as collective strength in Ref. [4] and dependence in Ref. [10].

For each measure, an itemset is called *highly co-occurrent* if the value indicating the correlation among the items exceeds a given threshold. In this paper, we also show that finding all the highly co-occurrent itemsets is still NP-hard under whichever measure, including collective strength. Furthermore, based on the notion of (k, c)-sparsity, we propose subclasses of databases for which we can efficiently find all the highly co-occurrent itemsets.

The rest of this paper is organized as follows. In Sect. 2, we provide preliminary definitions. In Sect. 3, we introduce the notion of (k, c)-sparsity. Then using (k, c)-sparsity, we propose a subclass of databases for which we can efficiently find all the large itemsets. In Sect. 4, we define several alternative measures to the support. Then we show that the problem of finding all the highly co-occurrent itemsets is NP-hard under whichever measure we define. In Sect. 5, we propose subclasses of databases for which we can efficiently find all the highly co-occurrent itemsets. Finally, we discuss our results in Sect. 6.

2. Preliminaries

Let I be a finite set of items. A subset X of I is called an *itemset*. The *size* of X, denoted by $|X|$, is the number of items in X. A market basket database (*MBD*) D is a finite multiset of itemsets; that is, D may contain multiple occurrences of the same itemset. An itemset in D is also called a *transaction* in D. Let $|D|$ denote the number of transactions in D. The *size* of D, denoted by $||D||$, is defined as $|D| \cdot |I|$ (each transaction is supposed to be implemented by a $|I|$-digit binary number).

We say that a transaction t in D *supports* an itemset X if $X \subseteq t$. By $sup_D(X)$, we mean the number of transactions in D that support X. For a given positive integer s ($0 \leqq s \leqq |D|$), called *minimum support number*, we say that an itemset X is *large* in D if $sup_D(X) \geqq s$. The *support rate* $supr_D(X)$ of an itemset X in D is defined as follows:

$$supr_D(X) \triangleq \frac{sup_D(X)}{|D|}.$$

For a given real number r ($0 \leqq r \leqq 1$), called *minimum support rate*, we say that an itemset X is *large* in D if $supr_D(X) \geqq r$. Note that when D is provided, we can use the minimum support number s and the minimum support rate r interchangeably by letting $s = \lfloor r \times |D| \rfloor$.

By finding large itemsets in D, we can identify sets of items that are frequently purchased together.

The large itemset problem is defined as follows [7].

Definition 1 (large itemset problem)**:** Given an MBD D, a minimum support number s (or a minimum support rate r), and a positive integer h, is there a large itemset in D of size at least h? □

This problem is shown to be NP-complete [7]. From this result, finding all the large itemsets is impossible in polynomial time in the size of a given database unless P=NP.

3. (k, c)-sparse Databases

3.1 (k, c)-sparsity

In our previous work [7], we have introduced the notion of k-sparsity of databases, which is defined as follows.

Definition 2 (k-sparsity)**:** A database D is called *k-sparse* if for any itemsets X and Y such that $|X \cup Y| > k$,

$$supr_D(X \cup Y) \leqq supr_D(X) \cdot supr_D(Y).$$

□

Intuitively, k-sparsity of a database means that the supports of itemsets of size k or more are sufficiently low in the database. However, k-sparsity is very strong because the inequality in Definition 2 must be satisfied for *all* combinations of X and Y.

In this section, we introduce the notion of (k, c)-sparsity, which is strictly weaker than k-sparsity, where k is a positive integer and c is a positive real number. The value of c represents a degree of sparsity.

Definition 3 ((k, c)-sparsity)**:** A database D is called *(k, c)-sparse* if for any itemset X such that $|X| > k$, there is some $x \in X$ which satisfies the following inequality:

$$supr_D(X) \leqq c \cdot supr_D(X - \{x\}) \cdot supr_D(\{x\}).$$

□

In this definition, the inequality in Definition 3 must be satisfied only for one $x \in X$. Thus, (k, c)-sparsity is more practical compared to k-sparsity.

t_1	{hot dog, cola}
t_2	{beer}
t_3	{hot dog, popcorn, cola}
t_4	{popcorn, beer}
t_5	{popcorn, cola}
t_6	{hot dog}
t_7	{hot dog, beer}
t_8	{hot dog, popcorn, cola, beer}
t_9	{popcorn, beer}
t_{10}	{cola}
t_{11}	{popcorn, cola}
t_{12}	{hot dog, beer}

Fig. 2: A market basket database D_2.

Example 2: Consider D_2 shown in Fig. 2. Let $k = 2$ and $c = 1$. Let us check whether D_2 is $(2, 1)$-sparse. The only itemset of size 4 which appears in D_2 is $X = \{\text{hot dog, popcorn, cola, beer}\}$, and

$$supr_{D_2}(X) = \frac{1}{12} \leqq supr_{D_2}(X - \{\text{beer}\}) \cdot supr_{D_2}(\{\text{beer}\}) = \frac{1}{6} \cdot \frac{1}{2}.$$

Therefore, X and $x =$ beer satisfy the inequality in Definition 3. It is easy to see that the inequality holds for all the itemsets of size 3. Therefore, D_2 is $(2, 1)$-sparse.

On the other hand, D_2 is not 2-sparse. Let us consider the case that $X \cup Y = \{\text{hot dog, popcorn, cola, beer}\}$. Then, the inequality in Definition 2 is not satisfied for all combinations of X and Y. □

3.2 Class (k, c, M)-Δ

In this section, using (k, c)-sparsity, we propose a subclass of databases, called (k, c, M)-Δ, where M is a positive real number. For a database in (k, c, M)-Δ, we can efficiently find all the large itemsets. Class (k, c, M)-Δ consists of all the (k, c)-sparse databases with the following condition.

Condition 1: For each item $x \in I$,

$$supr_D(\{x\}) \leqq M.$$

□

In addition, we consider the following condition on the minimum support rate.

Condition 2: There exists some r_m $(0 < r_m < 1)$ such that the given minimum support rate r is at least r_m. □

When $cM < 1$, the size of any large itemset in a database in (k, c, M)-Δ is bounded by a constant, which is determined by k, c, M, and r_m.

Lemma 1: Suppose that a database D is in (k, c, M)-Δ with $cM < 1$, and Condition 2 is satisfied. Let X be a large itemset in D. Then, the following inequality holds:

$$|X| \leqq k + \left\lfloor \frac{\log r_m}{\log(cM)} \right\rfloor.$$

Proof: Let $X = \{x_1, \ldots, x_t\}$ where $t > k$. Then, since D is (k, c)-sparse, there is some $x \in X$ such that

$$supr_D(X) \leqq c \cdot supr_D(X - \{x\}) \cdot supr_D(\{x\}).$$

Without loss of generality, let x_t be such x. That is,

$$\begin{aligned} supr_D(X) &= supr_D(\{x_1, \ldots, x_t\}) \\ &\leqq c \cdot supr_D(\{x_1, \ldots, x_{t-1}\}) \cdot supr_D(\{x_t\}). \end{aligned}$$

By repeating the same argument, we can obtain

$$\begin{aligned} supr_D(X) &\leqq c^{t-k} \cdot supr_D(\{x_1, \ldots, x_k\}) \cdot \prod_{i=k+1}^{t} supr_D(\{x_i\}) \\ &\leqq c^{t-k} \cdot \prod_{i=k+1}^{t} supr_D(\{x_i\}) \\ &\leqq c^{t-k} \cdot M^{t-k}. \end{aligned}$$

From Condition 2, $r_m \leqq r \leqq supr_D(X)$. Thus,

$$\begin{aligned} r_m &\leqq c^{t-k} \cdot M^{t-k} \\ \log r_m &\leqq (t-k)\log(cM) \\ t &\leqq k + \left\lfloor \frac{\log r_m}{\log(cM)} \right\rfloor \\ |X| &\leqq k + \left\lfloor \frac{\log r_m}{\log(cM)} \right\rfloor. \end{aligned}$$

□

Let $l = k + \left\lfloor \frac{\log r_m}{\log(cM)} \right\rfloor$. For a given itemset, it can be checked whether the itemset is large in D in $\mathcal{O}(||D||)$ time. Since there are at most $|I|^l$ itemsets of size less than or equal to l, all the large itemsets can be computed in $\mathcal{O}(||D|| \cdot |I|^l)$ time.

Theorem 1: Suppose that a database D is in (k, c, M)-Δ with $cM < 1$, and Condition 2 is satisfied. Then, all the large itemsets in D can be computed in polynomial time in $||D||$. □

4. Computational Complexity of Finding Highly Co-occurrent Itemsets

4.1 Highly Co-occurrent Itemsets

Several disadvantages of the support-confidence framework have been pointed out in Refs. [4, 5, 10]. For example, the support of an itemset tends to be high if the itemset contains items with high supports (see Example 1). In this section, we define alternative measures to the support, which are defined by the combinations of the aspects such as

- the ratio of the actual value of the support of a given itemset to the expected value of the support of the itemset, based on the assumption of statistical independence,
- the fraction of transactions that do not contain any item in a given itemset,

and so on. By $oc_D(X)$, we mean a degree of the correlation among the items in X in D. An itemset X is called *highly co-occurrent* in D if $oc_D(X)$ exceeds a given user-defined threshold, called *minimum co-occurrence*.

In the next section, we provide several formal definitions of $oc_D(X)$. Before proceeding, we introduce the notion of SDI division.

Given an MBD D and an itemset X, the *SDI division* of D with X is to divide D into the three disjoint subsets $D_{\mathbf{S}}(X)$, $D_{\mathbf{D}}(X)$, and $D_{\mathbf{I}}(X)$ which are defined below:

$$\begin{aligned} D_{\mathbf{S}}(X) &\triangleq \{t \mid t \in D \text{ and } X \subseteq t\}, \\ D_{\mathbf{D}}(X) &\triangleq \{t \mid t \in D \text{ and } X \cap t = \emptyset\}, \\ D_{\mathbf{I}}(X) &\triangleq D - (D_{\mathbf{S}}(X) \cup D_{\mathbf{D}}(X)). \end{aligned}$$

Furthermore, we define $V_{\mathbf{S}D}(X)$, $V_{\mathbf{D}D}(X)$, and $V_{\mathbf{I}D}(X)$ as follows:

$$\begin{aligned} V_{\mathbf{S}D}(X) &\triangleq \frac{|D_{\mathbf{S}}(X)|}{|D|}, \\ V_{\mathbf{D}D}(X) &\triangleq \frac{|D_{\mathbf{D}}(X)|}{|D|}, \\ V_{\mathbf{I}D}(X) &\triangleq \frac{|D_{\mathbf{I}}(X)|}{|D|}. \end{aligned}$$

Note that for any itemset X, $V_{\mathbf{S}D}(X) = supr_D(X)$.

Let X be an itemset. For a given transaction, the probability that the itemset X occurs in the transaction is $\prod_{x \in X} V_{\mathbf{S}D}(\{x\})$, which is calculated under the assumption that each item occurs in D independently. The probability that none of the items in X occurs in the transaction is $\prod_{x \in X} V_{\mathbf{D}D}(\{x\})$. Thus the expected fraction of transactions in which at least one of the items in X occurs in the transactions and at least one does not is given by $1 - \prod_{x \in X} V_{\mathbf{S}D}(\{x\}) - \prod_{x \in X} V_{\mathbf{D}D}(\{x\})$. In what follows, we use the following notations:

$$\begin{aligned} E_{\mathbf{S}D}(X) &\triangleq \prod_{x \in X} V_{\mathbf{S}D}(\{x\}), \\ E_{\mathbf{D}D}(X) &\triangleq \prod_{x \in X} V_{\mathbf{D}D}(\{x\}), \\ E_{\mathbf{I}D}(X) &\triangleq 1 - \prod_{x \in X} V_{\mathbf{S}D}(\{x\}) - \prod_{x \in X} V_{\mathbf{D}D}(\{x\}). \end{aligned}$$

We omit a database name D in $V_D(X)$ and $E_D(X)$ if it is clear from the context. For example, we write $V_{\mathbf{S}}(X)$ shortly instead of $V_{\mathbf{S}D}(X)$.

4.2 Definitions of Co-occurrence

4.2.1 Type I

There may be a case that we want to measure the correlation among the items in a given itemset by comparing the actual value to the expected value. Type **I** has the simplest form of the rest of the definitions which consider the expected value.

Definition 4 (type **I**)**:**

$$oc_D(X) \triangleq \frac{V_{\mathbf{S}}(X)}{E_{\mathbf{S}}(X)}.$$

□

t_1	{bread, ham, milk}
t_2	{bread, lettuce, tomato, coffee}
t_3	{eggs, lettuce, milk}
t_4	{cornflakes, milk}
t_5	{bread, lettuce, coffee}
t_6	{bread, eggs}

Fig. 3: A market basket database D_3.

The denominator of this formula is the expected value of the support rate of X under the assumption that each item in X occurs in D independently. When there is no correlation among the items in X, the value of $oc_D(X)$ is equal to 1.

Example 3: Consider D_1 shown in Fig. 1. Suppose that the minimum co-occurrence c is 1.5. Let $X = \{\text{cornflakes, milk}\}$ and $Z = \{\text{coffee, eggs}\}$. Since $V_{\mathbf{S}}(X) = 1/2$ and $E_{\mathbf{S}}(X) = 3/6 \times 4/6 = 1/3$,

$$oc_{D_1}(X) = \frac{1/2}{1/3} = 1.5 \geqq c.$$

Therefore, X is highly co-occurrent in D_1. On the other hand, since $V_{\mathbf{S}}(Z) = 1/2$ and $E_{\mathbf{S}}(Z) = 5/6 \times 4/6 = 5/9$,

$$oc_{D_1}(Z) = \frac{1/2}{5/9} = 0.9 < c.$$

Therefore, Z is not highly co-occurrent in D_1. □

Note that Z is large when the minimum support rate is 0.3 as seen in Example 1. In this definition, Z is not considered to have high correlation because its actual support rate is not sufficiently high compared to the expected value.

4.2.2 Type II

Consider an itemset $X = \{\text{cornflakes, milk}\}$. Then transactions that contain neither cornflakes nor milk can be considered to establish the correlation among cornflakes and milk. We incorporate the fraction of such transactions, that is, $V_{\mathbf{D}}(X)$ into the definition of $oc_D(X)$.

Definition 5 (type **II**)**:**

$$oc_D(X) \triangleq V_{\mathbf{S}}(X) + V_{\mathbf{D}}(X).$$

□

Example 4: Consider D_3 shown in Fig. 3. Suppose that the minimum co-occurrence c is 0.3. Let $X = \{\text{cornflakes, milk}\}$ and $Y = \{\text{bread, milk}\}$. The SDI division with X divides D_3 into $D_{\mathbf{S}}(X) = \{t_4\}$, $D_{\mathbf{D}}(X) = \{t_2, t_5, t_6\}$, and $D_{\mathbf{I}}(X) = \{t_1, t_3\}$. Since $V_{\mathbf{S}}(X) = 1/6$ and $V_{\mathbf{D}}(X) = 1/2$,

$$oc_{D_3}(X) = \frac{1}{6} + \frac{1}{2} \approx 0.67 \geqq c.$$

Therefore, X is highly co-occurrent in D_3. □

In Example 4, let us consider the case that the minimum support rate is 0.3. Then, X is not large because its support rate is less than the minimum support rate, while X is highly co-occurrent in this definition. For database users who want to obtain itemsets like X, this type of definition may be acceptable.

4.2.3 Type III

Type **III** is defined by the combination of type **I** and type **II**.

Definition 6 (type **III**)**:**

$$oc_D(X) \triangleq \frac{V_{\mathbf{S}}(X) + V_{\mathbf{D}}(X)}{E_{\mathbf{S}}(X) + E_{\mathbf{D}}(X)}.$$

□

4.2.4 Type IV

Type **IV** is also defined by the combination of type **I** and **II**, but has the slightly different form from type **III**.

Definition 7 (type **IV**)**:**

$$oc_D(X) \triangleq \frac{V_{\mathbf{S}}(X)}{E_{\mathbf{S}}(X)} \times \frac{V_{\mathbf{D}}(X)}{E_{\mathbf{D}}(X)}.$$

□

The reason why we consider type **IV** is that in the definition of type **III**, when $E_{\mathbf{D}}(X)$ is much larger than $E_{\mathbf{S}}(X)$, $\frac{V_{\mathbf{S}}(X)}{E_{\mathbf{S}}(X)}$ may not be well reflected in the result value of $oc_D(X)$ even if it has very large value. For example, consider the case that $V_{\mathbf{S}}(\{x\}) = V_{\mathbf{S}}(\{y\}) = {}^{10}/_{100}$, $V_{\mathbf{S}}(\{x,y\}) = {}^{10}/_{100}$, and $V_{\mathbf{D}}(\{x,y\}) = {}^{81}/_{100}$. Then, $oc_D(\{x,y\}) = 10$ in type **IV**, while $oc_D(\{x,y\}) \approx 1.11$ in type **III**. Although type **II** may also work in this example, type **IV** considers $V_{\mathbf{D}}(X)$ while type **II** does not. On the other hand, this definition does not work well for an itemset X such that $D_{\mathbf{D}}(X) = \emptyset$ because in that case, $oc_D(X)$ is equal to 0 even if $\frac{V_{\mathbf{S}}(X)}{E_{\mathbf{S}}(X)}$ is large.

4.2.5 Type V

This is an extension of type **III**. Type **V** is the same as *collective strength* [4], which has been proposed as alternative to the support. This can be expressed in our notation as follows.

Definition 8 (type **V**)**:**

$$oc_D(X) \triangleq \frac{V_{\mathbf{S}}(X) + V_{\mathbf{D}}(X)}{E_{\mathbf{S}}(X) + E_{\mathbf{D}}(X)} \times \frac{E_{\mathbf{I}}(X)}{V_{\mathbf{I}}(X)}.$$

□

Since transactions in $D_{\mathbf{I}}(X)$ can be considered to be counterexamples of high correlation among the items in X, the ratio of $V_{\mathbf{I}}(X)$ to $E_{\mathbf{I}}(X)$ is incorporated inversely into the definition of $oc_D(X)$. More details of this formula is described in Ref. [4].

4.2.6 Type VI

This is an extension of type **IV**. Like type **V**, the ratio of $V_{\mathbf{I}}(X)$ to $E_{\mathbf{I}}(X)$ is multiplied inversely.

Definition 9 (type **VI**)**:**

$$oc_D(X) \triangleq \frac{V_{\mathbf{S}}(X)}{E_{\mathbf{S}}(X)} \times \frac{V_{\mathbf{D}}(X)}{E_{\mathbf{D}}(X)} \times \frac{E_{\mathbf{I}}(X)}{V_{\mathbf{I}}(X)}.$$

□

Also, this does not work well for an itemset X such that $D_{\mathbf{D}}(X) = \emptyset$ or $D_{\mathbf{I}}(X) = \emptyset$ from the same reason as stated above.

4.3 NP-Completeness of the Highly Co-occurrent Itemset Problem

In this section, we show that finding all the highly co-occurrent itemsets is NP-hard under whichever measure we define.

Although there are several definitions of $oc_D(X)$, we define the *highly co-occurrent itemset problem* uniformly as follows.

Definition 10 (highly co-occurrent itemset problem)**:** Given an MBD D, a minimum co-occurrence c in fractional representation in binary, and a positive integer l in unary, is there a highly co-occurrent itemset in D of size l? □

It is clear that the highly co-occurrent itemset problem is in NP. Guess an itemset of size l, and then check whether the itemset is highly co-occurrent in D. So, in the rest of this section, we concentrate on proving the NP-hardness of the co-occurrent itemset problem.

By "the problem $\mathcal{X}$", we mean the highly co-occurrent itemset problem which adopts type $\mathcal{X}$ as the definition of $oc_D(X)$.

4.3.1 Type I

We show the NP-hardness of the problem **I** by reducing the large itemset problem to the problem **I**. To make the reduction simple, we suppose that a minimum support number s is given in the large itemset problem. We construct an instance (D', c, l) of the problem **I** from an instance (D, s, h) of the large itemset problem. Here, we can assume $h \geqq 2$ because the large itemset problem is NP-complete even if $h \geqq 2$.

Let $\tilde{\cup}$ be the union operator on multisets (i.e., the union operator which counts multiple occurrences of elements).

Construction method 1:

- Let I be the set of all the items in D. Let $\mathcal{T}_x = \underbrace{\{\{x\}, \ldots, \{x\}\}}_{|D| - |D_{\mathbf{S}}(\{x\})|}$ for each $x \in I$. Let $D_{\mathbf{A}} = \tilde{\cup}_{x \in I} \mathcal{T}_x$. Then we define D' as follows:

  $$D' \triangleq D \,\tilde{\cup}\, D_{\mathbf{A}}.$$

  Note that $|D'|$ is at most $|I| \cdot |D| = ||D||$.

- Let $c = \dfrac{s \cdot |D'|^{h-1}}{|D|^h}$.

- Let $l = h$.

□

Since D' can be constructed in $\mathcal{O}(||D||)$ time and c has at most $h \log |D| + \log s + (h-1) \log ||D||$ digits, the above construction can be done in polynomial time in $||D|| + s + h$, which is the description size of the instance (D, s, h) of the large itemset problem.

Lemma 2: Consider a database D given as an instance of the large itemset problem and a database D' constructed from D. Then, for any item $x \in I$,

$$V_{\mathbf{S}D'}(\{x\}) = \frac{|D|}{|D'|}.$$

Proof: Let x be an item in I. Then,

$$\begin{aligned} |D'_{\mathbf{S}}(\{x\})| &= |D_{\mathbf{S}}(\{x\})| + |\mathcal{T}_x| \\ &= |D_{\mathbf{S}}(\{x\})| + |D| - |D_{\mathbf{S}}(\{x\})| \\ &= |D|. \end{aligned}$$

Thus,

$$V_{\mathbf{S}D'}(\{x\}) = \frac{|D'_{\mathbf{S}}(\{x\})|}{|D'|} = \frac{|D|}{|D'|}.$$

□

Lemma 3: Suppose that (D, s, h) $(h \geqq 2)$ is given as an instance of the large itemset problem. Let (D', c, l) be an instance of the problem **I** constructed from (D, s, h). Then, the following 1 and 2 are equivalent.

1. There is an itemset X in D such that $sup_D(X) \geqq s$ and $|X| \geqq h$.
2. There is an itemset X' in D' such that $oc_{D'}(X') \geqq c$ and $|X'| = l$.

Proof: From Lemma 2, for any itemset X',

$$E_{\mathbf{S}D'}(X') = \prod_{x \in X'} V_{\mathbf{S}D'}(\{x\}) = \left(\frac{|D|}{|D'|}\right)^{|X'|}.$$

$(1 \rightarrow 2)$: Assume that there is an itemset X in D such that $sup_D(X) \geqq s$ and $|X| \geqq h$. If an itemset X is large, then all the subsets of X are also large. Therefore, we can assume that there is an itemset X' such that $sup_D(X') \geqq s$ and $|X'| = h$. Since $|X'| \geqq 2$ and every transaction in $D_{\mathbf{A}}$ consists of just one item, no transaction in $D_{\mathbf{A}}$ supports X'. Thus,

$$|D'_{\mathbf{S}}(X')| = |D_{\mathbf{S}}(X')| \geqq s.$$

By dividing both sides of the above inequality by $|D'|E_{\mathbf{S}D'}(X)$,

$$\begin{aligned} oc_{D'}(X') = \frac{V_{\mathbf{S}D'}(X')}{E_{\mathbf{S}D'}(X')} &= \frac{|D'_{\mathbf{S}}(X')|}{|D'|E_{\mathbf{S}D'}(X')} \\ &\geqq \frac{s}{|D'|E_{\mathbf{S}D'}(X')} \\ &= \frac{s}{|D'| \cdot \left(\frac{|D|}{|D'|}\right)^h} \\ &= \frac{s \cdot |D'|^{h-1}}{|D|^h} \\ &= c. \end{aligned}$$

$(2 \rightarrow 1)$: Assume that there is an itemset X' in D' such that $oc_{D'}(X') \geqq c$ and $|X'| = l\ (= h)$. Then,

$$\begin{aligned} oc_{D'}(X') = \frac{V_{\mathbf{S}D'}(X')}{E_{\mathbf{S}D'}(X')} &\geqq c \\ \frac{V_{\mathbf{S}D'}(X')}{\left(\frac{|D|}{|D'|}\right)^h} &\geqq \frac{s}{|D'| \cdot \left(\frac{|D|}{|D'|}\right)^h} \\ \frac{|D'_{\mathbf{S}}(X')|}{|D'|} &\geqq \frac{s}{|D'|} \\ |D'_{\mathbf{S}}(X')| &\geqq s \end{aligned}$$

Since $|X'| \geqq 2$ and every transaction in $D_{\mathbf{A}}$ consists of just one item, no transaction in $D_{\mathbf{A}}$ supports X'. Thus,

$$|D'_{\mathbf{S}}(X')| = |D_{\mathbf{S}}(X')| = sup_D(X') \geqq s.$$

□

4.3.2 Type II

We show the NP-hardness of the problem **II** by reducing the large itemset problem to the problem **II**. We construct an instance (D', m, u) of the problem **II** from an instance (D, r, h) of the large itemset problem by the following construction method.

Construction method 2:

- Assume that $D = \{t_1, \ldots, t_n\}$. Let $I = \{i_1, \ldots, i_k\}$ be the set of all the items in D. Let $I^* = \{i_{k+1}, \ldots, i_{2k}\}$ be a set of new items, where $I \cap I^* = \emptyset$. Let $t'_j = t_j \cup I^*$ be a transaction for each j $(1 \leqq j \leqq n)$. Then we define D' as follows:

$$D' \triangleq \{t'_1, \ldots, t'_n\}.$$

- Let $m = r$.
- Let $u = h + k$.

□

The above construction can be done in polynomial time in $||D|| + r + h$, which is the description size of the instance (D, r, h) of the large itemset problem.

Lemma 4: Suppose that (D, r, h) is given as an instance of the large itemset problem. Let (D', m, u) be an instance of the problem **II** constructed from (D, r, h). Then, the following 1 and 2 are equivalent.

1. There is an itemset X in D such that $supr_D(X) \geqq r$ and $|X| \geqq h$.
2. There is an itemset X' in D' such that $oc_{D'}(X') \geqq m$ and $|X'| = u$.

Proof: $(1 \rightarrow 2)$: Assume that there is an itemset $X \subseteqq I$ in D such that $supr_D(X) \geqq r$ and $|X| \geqq h$. If an itemset X is large, then all the subsets of X are also large. Therefore, we can assume that there is an itemset X_h such that $supr_D(X_h) \geqq r$ and $|X_h| = h$. Let $X' = X_h \cup I^*$. Then,

$$|X'| = |X_h \cup I^*| = |X_h| + |I^*| = h + k = u.$$

From the construction method of D', it is clear that

- if a transaction t_i supports X_h in D, then a transaction t'_i supports X' in D'. That is, $t'_i \in D'_{\mathbf{S}}(X')$;
- otherwise, there is at least one item in X' that t'_i does not contain, and both of t'_i and X' contain I^*. That is, $t'_i \in D'_{\mathbf{I}}(X')$.

Therefore, $X_h \subseteqq t_i$ if and only if $X' \subseteqq t'_i$ for any i, and hence $|D_{\mathbf{S}}(X_h)| = |D'_{\mathbf{S}}(X')|$. Since any transaction $t'_i \in D'$ and X' contain I^*, $t'_i \cap X' \neq \emptyset$. That is, $D'_{\mathbf{D}}(X') = \emptyset$. Thus,

$$\begin{aligned} oc_{D'}(X') &= V_{\mathbf{S}D'}(X') + V_{\mathbf{D}D'}(X') \\ &= \frac{|D'_{\mathbf{S}}(X')|}{|D'|} + \frac{|D'_{\mathbf{D}}(X')|}{|D'|} \\ &= \frac{|D'_{\mathbf{S}}(X')|}{|D'|} = \frac{|D_{\mathbf{S}}(X_h)|}{|D|} \\ &= supr_D(X_h) \\ &\geqq r = m. \end{aligned}$$

$(2 \rightarrow 1)$: Assume that there is an itemset $X' \subseteqq I \cup I^*$ in D' such that $oc_{D'}(X') \geqq m$ and $|X'| = u$. Let $X = X' \cap I$. Then,

$$|X| \geqq |X'| - |I^*| = u - k = h.$$

From $t'_i = t_i \cup I^*$, $X' \subseteqq t'_i$ if and only if $X \subseteqq t_i$ for any i, and hence $|D'_{\mathbf{S}}(X')| = |D_{\mathbf{S}}(X)|$. Furthermore, since

$$|X'| = u = h + k > k = |I|,$$

X' contains at least one item in I^*.

Since any transaction $t'_i \in D'$ contains all the items in I^*, t'_i and X' contain at least one common item. That is, $D'_{\mathbf{D}}(X') = \emptyset$. Thus,

$$\begin{aligned} supr_D(X) &= \frac{|D_{\mathbf{S}}(X)|}{|D|} = \frac{|D'_{\mathbf{S}}(X')|}{|D'|} \\ &= \frac{|D'_{\mathbf{S}}(X')|}{|D'|} + \frac{|D'_{\mathbf{D}}(X')|}{|D'|} \\ &= oc_{D'}(X') \\ &\geqq m = r. \end{aligned}$$

□

4.3.3 Type III

We show the NP-hardness of the problem **III** by reducing the problem **II** to the problem **III**. We construct an instance (D', c, l) of the problem **III** from an instance (D, m, u) of the problem **II**.

Construction method 3:

- Assume that $D = \{t_1, \ldots, t_n\}$. Let I be the set of all the items in D. Let $t'_i = I - t_i$ for each i $(1 \leqq i \leqq n)$. Let $\bar{D} = \{t'_1, \ldots, t'_n\}$. Then we define D' as follows:
$$D' \triangleq D \tilde{\cup} \bar{D}.$$
- Let $c = m2^{u-1}$.
- Let $l = u$.

□

Since D' can be constructed in $\mathcal{O}(||D||)$ time and c has at most $(u-1) + \log m$ digits, the above construction can be done in polynomial time in $||D|| + \log m + u$, which is the description size of the instance (D, m, u) of the problem **II**.

Lemma 5: Consider a database D given as an instance of the problem **II** and a database D' constructed from D. Then, for any item $x \in I$,

$$V_{\mathbf{S}D'}(\{x\}) = V_{\mathbf{D}D'}(\{x\}) = \frac{1}{2}.$$

Proof: For any item $x \in I$,

$$\begin{aligned} V_{\mathbf{S}D'}(\{x\}) &= \frac{|D_{\mathbf{S}}(\{x\})| + |\bar{D}_{\mathbf{S}}(\{x\})|}{|D'|} \\ &= \frac{|D_{\mathbf{S}}(\{x\})| + |D| - |D_{\mathbf{S}}(\{x\})|}{|D'|} \\ &= \frac{|D|}{|D'|} = \frac{1}{2}. \end{aligned}$$

The proof for $V_{\mathbf{D}D'}(\{x\}) = 1/2$ is similar. □

Lemma 6: Consider a database D given as an instance of the problem **II** and a database D' constructed from D. Then, for any itemset X, the following two equations hold.

$$\begin{aligned} V_{\mathbf{S}D}(X) + V_{\mathbf{D}D}(X) &= V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X) \\ V_{\mathbf{I}D}(X) &= V_{\mathbf{I}D'}(X) \end{aligned}$$

Proof: From the construction method of $\bar{D}$, it is clear that

- if a transaction t_i supports X in D, then the transaction t'_i and X are disjoint in $\bar{D}$;
- if a transaction t_i and X are disjoint in D, then the transaction t'_i supports X in $\bar{D}$; and
- if a transaction t_i contains at least one item (but not all items) in X in D, then the transaction t'_i also contains at least one item (but not all items) in X in $\bar{D}$.

Thus, $|D_{\mathbf{S}}(X)| = |\bar{D}_{\mathbf{D}}(X)|$, $|D_{\mathbf{D}}(X)| = |\bar{D}_{\mathbf{S}}(X)|$, and $|D_{\mathbf{I}}(X)| = |\bar{D}_{\mathbf{I}}(X)|$. Therefore,

$$\begin{aligned} V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X) &= \frac{|D'_{\mathbf{S}}(X)|}{|D'|} + \frac{|D'_{\mathbf{D}}(X)|}{|D'|} \\ &= \frac{|D_{\mathbf{S}}(X)| + |\bar{D}_{\mathbf{S}}(X)| + |D_{\mathbf{D}}(X)| + |\bar{D}_{\mathbf{D}}(X)|}{2|D|} \\ &= \frac{|D_{\mathbf{S}}(X)| + |D_{\mathbf{D}}(X)|}{|D|} \\ &= V_{\mathbf{S}D}(X) + V_{\mathbf{D}D}(X). \end{aligned}$$

Also,

$$\begin{aligned} V_{\mathbf{I}D'}(X) &= \frac{|D'_{\mathbf{I}}(X)|}{|D'|} \\ &= \frac{|D_{\mathbf{I}}(X)| + |\bar{D}_{\mathbf{I}}(X)|}{2|D|} \\ &= \frac{|D_{\mathbf{I}}(X)|}{|D|} = V_{\mathbf{I}D}(X). \end{aligned}$$

□

Lemma 7: Suppose that (D, m, u) is given as an instance of the problem **II**. Let (D', c, l) be an instance of the problem **III** constructed from (D, m, u). Then, the following 1 and 2 are equivalent.

1. There is an itemset X in D such that $oc_D(X) \geqq m$ and $|X| = u$.
2. There is an itemset X in D' such that $oc_{D'}(X) \geqq c$ and $|X| = l$.

Proof: From Lemma 5, for any itemset X,

$$\begin{aligned} & E_{\mathbf{S}D'}(X) + E_{\mathbf{D}D'}(X) \\ &= \prod_{x \in X} V_{\mathbf{S}D'}(\{x\}) + \prod_{x \in X} V_{\mathbf{D}D'}(\{x\}) = \left(\frac{1}{2}\right)^{|X|-1}. \end{aligned}$$

Note that the definitions of $oc_D(X)$ and $oc_{D'}(X)$ are different. $oc_D(X)$ has the definition of type **II**, whereas $oc_{D'}(X)$ has the definition of type **III**.

$(1 \rightarrow 2)$: Assume that there is an itemset X in D such that $oc_D(X) \geqq m$ and $|X| = u$. From Lemma 6,

$$V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X) \geqq m.$$

Thus,

$$\begin{aligned} oc_{D'}(X) = \frac{V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X)}{E_{\mathbf{S}D'}(X) + E_{\mathbf{D}D'}(X)} &\geqq \frac{m}{(\frac{1}{2})^{|X|-1}} \\ &= m \cdot 2^{u-1} \\ &= c. \end{aligned}$$

$(2 \rightarrow 1)$: Assume that there is an itemset X in D' such that $oc_{D'}(X) \geqq c$ and $|X| = l\ (= u)$. Then,

$$\begin{aligned} \frac{V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X)}{E_{\mathbf{S}D'}(X) + E_{\mathbf{D}D'}(X)} &\geqq c \\ (V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X)) \cdot 2^{u-1} &\geqq m \cdot 2^{u-1} \\ V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X) &\geqq m. \end{aligned}$$

Thus, from Lemma 6,

$$V_{\mathbf{S}D}(X) + V_{\mathbf{D}D}(X) = oc_D(X) \geqq m.$$

□

4.3.4 Type IV

We show the NP-hardness of the problem **IV** by reducing the problem **II** to the problem **IV**. We construct an instance (D', c, l) of the problem **IV** from an instance (D, m, u) of the problem **II**, as follows.

Construction method 4:

- Assume that $D = \{t_1, \ldots, t_n\}$. Let I be the set of all the items in D. Let $t'_i = I - t_i$ for each i $(1 \leqq i \leqq n)$. Let $\bar{D} = \{t'_1, \ldots, t'_n\}$. Then, we define D' as follows:

$$D' \triangleq D \,\tilde{\cup}\, \bar{D}.$$

- Let $c = m^2 \cdot 2^{2(u-1)}$.
- Let $l = u$.

□

Since D' can be constructed in $\mathcal{O}(||D||)$ time and c has at most $2\log m + 2(u-1)$ digits, the above construction can be done in polynomial time in $||D|| + \log m + u$, which is the description size of the instance (D, m, u) of the problem **II**.

Lemma 8: Consider a database D given as an instance of the problem **II** and a database D' constructed from D. Then, for any item $x \in I$,

$$V_{\mathbf{S}D'}(\{x\}) = V_{\mathbf{D}D'}(\{x\}) = \frac{1}{2}.$$

Proof: The proof of this lemma is similar to Lemma 5. □

Lemma 9: Consider a database D given as an instance of the problem **II** and a database D' constructed from D. Then, for any itemset X, the following equation holds.

$$V_{\mathbf{S}D'}(X) = V_{\mathbf{D}D'}(X) = \frac{V_{\mathbf{S}D}(X) + V_{\mathbf{D}D}(X)}{2}$$

Proof: The proof of this lemma is similar to Lemma 6. □

Lemma 10: Suppose that (D, m, u) is given as an instance of the large itemset problem. Let (D', c, l) be an instance of the problem **IV** constructed from (D, m, u). Then, the following 1 and 2 are equivalent.

1. There is an itemset X in D such that $oc_D(X) \geqq m$ and $|X| = u$.
2. There is an itemset X in D' such that $oc_{D'}(X) \geqq c$ and $|X| = l$.

Proof: From Lemma 8, for any itemset X,

$$\begin{aligned} E_{\mathbf{S}D'}(X) &= \prod_{x \in X} V_{\mathbf{S}D'}(\{x\}) = E_{\mathbf{D}D'}(X) \\ &= \prod_{x \in X} V_{\mathbf{D}D'}(\{x\}) = \left(\frac{1}{2}\right)^{|X|}. \end{aligned}$$

Note that the definitions of $oc_D(X)$ and $oc_{D'}(X)$ are different. $oc_D(X)$ has the definition of type **II**, whereas $oc_{D'}(X)$ has the definition of type **IV**.

$(1 \rightarrow 2)$: Assume that there is an itemset X in D such that $oc_D(X) \geqq m$ and $|X| = u$. Then, from Lemma 9,

$$\begin{aligned} oc_{D'}(X) &= \frac{V_{\mathbf{S}D'}(X)}{E_{\mathbf{S}D'}(X)} \cdot \frac{V_{\mathbf{D}D'}(X)}{E_{\mathbf{D}D'}(X)} \\ &= \left(\frac{V_{\mathbf{S}D'}(X)}{(\frac{1}{2})^{|X|}}\right)^2 \\ &= \left(\frac{(\frac{V_{\mathbf{S}D}(X) + V_{\mathbf{D}D}(X)}{2})}{(\frac{1}{2})^{|X|}}\right)^2 \\ &\geqq \left(\frac{\frac{m}{2}}{(\frac{1}{2})^u}\right)^2 \\ &= m^2 \cdot 2^{2(u-1)} \\ &= c. \end{aligned}$$

$(2 \rightarrow 1)$: Assume that there is an itemset X in D' such that $oc_{D'}(X) \geqq c$ and $|X| = l\ (= u)$. Then, from Lemma 9,

$$\begin{aligned} oc_{D'}(X) = \frac{V_{\mathbf{S}D'}(X)}{E_{\mathbf{S}D'}(X)} \cdot \frac{V_{\mathbf{D}D'}(X)}{E_{\mathbf{D}D'}(X)} &\geqq c \\ \left(\frac{V_{\mathbf{S}D'}(X)}{(\frac{1}{2})^{|X|}}\right)^2 &\geqq c \\ \left(\frac{(\frac{V_{\mathbf{S}D}(X)+V_{\mathbf{D}D}(X)}{2})}{(\frac{1}{2})^{|X|}}\right)^2 &\geqq c \\ (V_{\mathbf{S}D}(X) + V_{\mathbf{D}D}(X))^2 \cdot 2^{2(u-1)} &\geqq m^2 \cdot 2^{2(u-1)} \\ (oc_D(X))^2 &\geqq m^2 \\ (oc_D(X) + m)(oc_D(X) - m) &\geqq 0 \\ oc_D(X) &\geqq m. \end{aligned}$$

□

4.3.5 Type V

We show the NP-hardness of the problem **V** by reducing the problem **II** to the problem **V**. We construct an instance (D', c, l) of the problem **V** from an instance (D, m, u) of the problem **II**, as follows.

Construction method 5:

- Assume that $D = \{t_1, \ldots, t_n\}$. Let I be the set of all the items in D. Let $t'_i = I - t_i$ for each i $(1 \leqq i \leqq n)$. Let $\bar{D} = \{t'_1, \ldots, t'_n\}$. Then, we define D' as follows:
$$D' \triangleq D \,\tilde{\cup}\, \bar{D}.$$
- Let $c = \dfrac{m}{1-m} \cdot (2^{u-1} - 1)$.
- Let $l = u$.

□

Since D' can be constructed in $\mathcal{O}(||D||)$ time and c has at most $\log(1-m) + \log m + (u-1)$ digits, the above construction can be done in polynomial time in $||D|| + \log m + u$, which is the description size of the instance (D, m, u) of the problem **II**.

Lemma 11: Consider a database D given as an instance of the problem **II** and a database D' constructed from D. Then, for any item $x \in I$,

$$V_{\mathbf{S}D'}(\{x\}) = V_{\mathbf{D}D'}(\{x\}) = \frac{1}{2}.$$

Proof: The proof of this lemma is similar to Lemma 5. □

Lemma 12: Consider a database D given as an instance of the problem **II** and a database D' constructed from D. Then, for any itemset X, the following two equations hold.

$$\begin{aligned} V_{\mathbf{S}D}(X) + V_{\mathbf{D}D}(X) &= V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X) \\ V_{\mathbf{I}D}(X) &= V_{\mathbf{I}D'}(X) \end{aligned}$$

Proof: The proof of this lemma is similar to Lemma 6. □

Lemma 13: Suppose that (D, m, u) is given as an instance of the problem **II**. Let (D', c, l) be an instance of the problem **V** constructed from (D, m, u). Then, the following 1 and 2 are equivalent.

1. There is an itemset X in D such that $oc_D(X) \geqq m$ and $|X| = u$.
2. There is an itemset X in D' such that $oc_{D'}(X) \geqq c$ and $|X| = l$.

Proof: From Lemma 11, for any itemset X,

$$\begin{aligned} & E_{\mathbf{S}D'}(X) + E_{\mathbf{D}D'}(X) \\ = & \prod_{x \in X} V_{\mathbf{S}D'}(\{x\}) + \prod_{x \in X} V_{\mathbf{D}D'}(\{x\}) = \left(\frac{1}{2}\right)^{|X|-1}. \end{aligned}$$

Note that the definitions of $oc_D(X)$ and $oc_{D'}(X)$ are different. $oc_D(X)$ has the definition of type **II**, whereas $oc_{D'}(X)$ has the definition of type **V**.

$(1 \rightarrow 2)$: Assume that there is an itemset X in D such that $oc_D(X) \geqq m$ and $|X| = u$. From Lemma 12,

$$V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X) \geqq m.$$

Then,

$$\begin{aligned} oc_{D'}(X) &= \frac{V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X)}{E_{\mathbf{S}D'}(X) + E_{\mathbf{D}D'}(X)} \cdot \frac{E_{\mathbf{I}D'}(X)}{V_{\mathbf{I}D'}(X)} \\ &= \frac{V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X)}{V_{\mathbf{I}D'}(X)} \cdot \frac{(1 - (\frac{1}{2})^{|X|-1})}{(\frac{1}{2})^{|X|-1}} \\ &\geqq \frac{m}{1 - (V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X))} \cdot (2^{u-1} - 1) \\ &\geqq \frac{m}{1-m} \cdot (2^{u-1} - 1) \\ &= c. \end{aligned}$$

$(2 \rightarrow 1)$: Assume that there is an itemset X in D' such that $oc_{D'}(X) \geqq c$ and $|X| = l\ (= u)$. Then,

$$\begin{aligned} oc_{D'}(X) = \frac{V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X)}{E_{\mathbf{S}D'}(X) + E_{\mathbf{D}D'}(X)} \cdot \frac{E_{\mathbf{I}D'}(X)}{V_{\mathbf{I}D'}(X)} &\geqq c \\ \frac{V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X)}{V_{\mathbf{I}D'}(X)} \cdot (2^{u-1} - 1) &\geqq \frac{m}{1-m} \cdot (2^{u-1} - 1) \\ \frac{V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X)}{1 - (V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X))} &\geqq \frac{m}{1-m}. \end{aligned}$$

Thus, from Lemma 12,

$$\begin{aligned} \frac{V_{\mathbf{S}D}(X) + V_{\mathbf{D}D}(X)}{1 - (V_{\mathbf{S}D}(X) + V_{\mathbf{D}D}(X))} &\geqq \frac{m}{1-m} \\ oc_D(X) \cdot (1-m) &\geqq m \cdot (1 - oc_D(X)) \\ oc_D(X) - m \cdot oc_D(X) + m \cdot oc_D(X) - m &\geqq 0 \\ oc_D(X) &\geqq m. \end{aligned}$$

□

4.3.6 Type VI

We show the NP-hardness of the problem **VI** by reducing the problem **II** to the problem **VI**. We construct an instance (D', c, l) of the problem **VI** from an instance (D, m, u) of the problem **II**, as follows.

Construction method 6:

- Assume that $D = \{t_1, \ldots, t_n\}$. Let I be the set of all the items in D. Let $t'_i = I - t_i$ for each i $(1 \leqq i \leqq n)$. Let $\bar{D} = \{t'_1, \ldots, t'_n\}$. Then, we define D' as follows:
$$D' \triangleq D \,\tilde{\cup}\, \bar{D}.$$
- Let $c = \dfrac{m^2}{1-m} \cdot \left(2^{u-2} - \dfrac{1}{2}\right)$.
- Let $l = u$.

□

Since D' can be constructed in $\mathcal{O}(||D||)$ time and c has at most $\log(1-m) + 2\log m + (u-2)$ digits, the above construction can be done in polynomial time in $||D|| + \log m + u$, which is the description size of the instance (D, m, u) of the problem **II**.

Lemma 14: Consider a database D given as an instance of the problem **II** and a database D' constructed from D. Then, for any item $x \in I$,

$$V_{\mathbf{S}D'}(\{x\}) = V_{\mathbf{D}D'}(\{x\}) = \frac{1}{2}.$$

Proof: The proof of this lemma is similar to Lemma 5. □

Lemma 15: Consider a database D given as an instance of the problem **II** and a database D' constructed from D. Then, for any itemset X, the following equation holds.

$$V_{\mathbf{S}D'}(X) = V_{\mathbf{D}D'}(X) = \frac{V_{\mathbf{S}D}(X) + V_{\mathbf{D}D}(X)}{2}$$

Proof: The proof of this lemma is similar to Lemma 6. □

Lemma 16: Suppose that (D, m, u) is given as an instance of the problem **II**. Let (D', c, l) be an instance of the problem **VI** constructed from (D, m, u). Then, the following 1 and 2 are equivalent.

1. There is an itemset X in D such that $oc_D(X) \geqq m$ and $|X| = u$.
2. There is an itemset X in D' such that $oc_{D'}(X) \geqq c$ and $|X| = l$.

Proof: From Lemma 14, for any itemset X,

$$E_{\mathbf{S}D'}(X) = E_{\mathbf{D}D'}(X) = \prod_{x \in X} V_{\mathbf{S}D'}(\{x\}) = \left(\frac{1}{2}\right)^{|X|}.$$

Note that the definitions of $oc_D(X)$ and $oc_{D'}(X)$ are different. $oc_D(X)$ has the definition of type **II**, whereas $oc_{D'}(X)$ has the definition of type **VI**.

$(1 \rightarrow 2)$: Assume that there is an itemset X in D such that $oc_D(X) \geqq m$ and $|X| = u$. Then, from Lemma 15,

$$\begin{aligned}
oc_{D'}(X) &= \frac{V_{\mathbf{S}D'}(X)}{E_{\mathbf{S}D'}(X)} \cdot \frac{V_{\mathbf{D}D'}(X)}{E_{\mathbf{D}D'}(X)} \cdot \frac{E_{\mathbf{I}D'}(X)}{V_{\mathbf{I}D'}(X)} \\
&= \left(\frac{V_{\mathbf{S}D'}(X)}{(\frac{1}{2})^{|X|}}\right)^2 \cdot \left(\frac{1-(\frac{1}{2})^{|X|-1}}{1-2V_{\mathbf{S}D'}(X)}\right) \\
&= \left(\frac{(\frac{V_{\mathbf{S}D}(X)+V_{\mathbf{D}D}(X)}{2})}{(\frac{1}{2})^{|X|}}\right)^2 \cdot \left(\frac{1-(\frac{1}{2})^{|X|-1}}{1-2(\frac{V_{\mathbf{S}D}(X)+V_{\mathbf{D}D}(X)}{2})}\right) \\
&\geqq \left(\frac{(\frac{m}{2})}{(\frac{1}{2})^{u}}\right)^2 \cdot \left(\frac{1-(\frac{1}{2})^{u-1}}{1-2(\frac{m}{2})}\right) \\
&= \frac{m^2}{1-m} \cdot 2^{u-2}\left(1-\left(\frac{1}{2}\right)^{u-1}\right) \\
&= \frac{m^2}{1-m} \cdot (2^{u-2} - \frac{1}{2}) \\
&= c.
\end{aligned}$$

$(2 \rightarrow 1)$: Assume that there is an itemset X in D' such that $oc_{D'}(X) \geqq c$ and $|X| = l\ (= u)$. Then, from Lemma 15,

$$\begin{aligned}
oc_{D'}(X) = \frac{V_{\mathbf{S}D'}(X)}{E_{\mathbf{S}D'}(X)} \cdot \frac{V_{\mathbf{D}D'}(X)}{E_{\mathbf{D}D'}(X)} \cdot \frac{E_{\mathbf{I}D'}(X)}{V_{\mathbf{I}D'}(X)} &\geqq c \\
\left(\frac{V_{\mathbf{S}D'}(X)}{(\frac{1}{2})^{|X|}}\right)^2 \cdot \frac{1-(\frac{1}{2})^{|X|-1}}{1-2V_{\mathbf{S}D'}(X)} &\geqq c \\
\frac{(V_{\mathbf{S}D}(X)+V_{\mathbf{D}D}(X))^2}{1-2(\frac{V_{\mathbf{S}D}(X)+V_{\mathbf{D}D}(X)}{2})} \cdot (2^{u-2} - \frac{1}{2}) &\geqq \frac{m^2}{1-m} \cdot (2^{u-2} - \frac{1}{2}) \\
\frac{(V_{\mathbf{S}D}(X)+V_{\mathbf{D}D}(X))^2}{1-(V_{\mathbf{S}D}(X)+V_{\mathbf{D}D}(X))} &\geqq \frac{m^2}{1-m} \\
(oc_D(X))^2 \cdot (1-m) &\geqq m^2 \cdot (1-oc_D(X)) \\
(1-m)(oc_D(X))^2 + m^2 oc_D(X) - m^2 &\geqq 0 \\
((1-m)oc_D(X)+m)(oc_D(X)-m) &\geqq 0 \\
oc_D(X) &\geqq m.
\end{aligned}$$

□

5. Subclasses of Databases for which All the Highly Co-occurrent Itemsets can be Computed Efficiently

In this section, we propose subclasses of databases for which we can efficiently find all the highly co-occurrent itemsets. We consider type **I**, type **II**, and type **IV** as the definitions of $oc_D(X)$. Each subclass is defined based on the notion of (k, c)-sparsity.

The following condition on the minimum co-occurrence is assumed throughout this section.

Condition 3: There exists some b_m $(0 < b_m < 1)$ such that the given minimum co-occurrence b is at least b_m. □

5.1 Type I: Class (k, c, ϵ)-Γ

Class (k, c, ϵ)-Γ consists of all the (k, c)-sparse databases which satisfy the following condition.

Condition 4: For each item x,

$$\epsilon \leqq V_{\mathbf{S}}(\{x\}),$$

where ϵ is a positive real number. □

When $c < 1$, the size of any highly co-occurrent itemset in a database in (k, c, ϵ)-Γ is bounded by a constant, which is determined by k, c, ϵ, and b_m.

Lemma 17: Suppose that a database D is in (k, c, ϵ)-Γ with $c < 1$, and Condition 3 is satisfied. Let X be a highly co-occurrent itemset in D. Then, the following inequality holds:

$$|X| \leqq k + \left\lfloor \frac{\log b_m + k \log \epsilon}{\log c} \right\rfloor.$$

Proof: Let $X = \{x_1, \ldots, x_t\}$. Then,

$$oc_D(X) = \frac{V_{\mathbf{S}}(X)}{E_{\mathbf{S}}(X)} = \frac{V_{\mathbf{S}}(\{x_1, \ldots, x_t\})}{\prod_{i=1}^{t} V_{\mathbf{S}}(\{x_i\})}.$$

Suppose that $t > k$. Then, since D is (k, c)-sparse, there is some $x \in X$ such that

$$V_{\mathbf{S}}(X) \leqq c \cdot V_{\mathbf{S}}(X - \{x\}) \cdot V_{\mathbf{S}}(\{x\}).$$

Without loss of generality, let x_t be such x. Thus,

$$\begin{aligned} oc_D(X) &\leqq c \cdot \frac{V_{\mathbf{S}}(\{x_1, \ldots, x_{t-1}\}) \cdot V_{\mathbf{S}}(\{x_t\})}{\prod_{i=1}^{t} V_{\mathbf{S}}(\{x_i\})} \\ &= c \cdot \frac{V_{\mathbf{S}}(\{x_1, \ldots, x_{t-1}\})}{\prod_{i=1}^{t-1} V_{\mathbf{S}}(\{x_i\})}. \end{aligned}$$

By repeating the same argument, we can obtain

$$\begin{aligned} oc_D(X) &\leqq c^{t-k} \cdot \frac{V_{\mathbf{S}}(\{x_1, \ldots, x_k\})}{\prod_{i=1}^{k} V_{\mathbf{S}}(\{x_i\})} \\ &\leqq c^{t-k} \cdot \epsilon^{-k} \cdot V_{\mathbf{S}}(\{x_1, \ldots, x_k\}) \\ &\leqq c^{t-k} \cdot \epsilon^{-k}. \end{aligned}$$

From Condition 3, $b_m \leqq b \leqq oc_D(X)$. Thus,

$$\begin{aligned} b_m &\leqq c^{t-k} \cdot \epsilon^{-k} \\ \log b_m &\leqq (t-k) \log c - k \log \epsilon \\ \log b_m + k \log \epsilon &\leqq (t-k) \log c \\ t &\leqq k + \left\lfloor \frac{\log b_m + k \log \epsilon}{\log c} \right\rfloor \\ |X| &\leqq k + \left\lfloor \frac{\log b_m + k \log \epsilon}{\log c} \right\rfloor. \end{aligned}$$

□

Let $l = k + \left\lfloor \frac{\log b_m + k \log \epsilon}{\log c} \right\rfloor$. For a given itemset, it can be checked whether the itemset is highly co-occurrent in D in $\mathcal{O}(||D||)$ time. Since there are at most $|I|^l$ itemsets of size less than or equal to l, all the highly co-occurrent itemsets can be computed in $\mathcal{O}(||D|| \cdot |I|^l)$ time.

Theorem 2: Suppose that a database D is in (k, c, ϵ)-Γ with $c < 1$, and Condition 3 is satisfied. Then, all the highly co-occurrent itemsets in D can be computed in polynomial time in $||D||$. □

5.2 Type II: Class (k, c, M)-Δ'

Class (k, c, M)-Δ' consists of all the databases which satisfy the following two conditions.

Condition 5: For any itemset X such that $|X| > k$, there is some $x \in X$ which satisfies the following inequality:

$$\begin{aligned} & V_{\mathbf{S}}(X) + V_{\mathbf{D}}(X) \\ & \leqq \quad c \cdot (V_{\mathbf{S}}(X - \{x\}) + V_{\mathbf{D}}(X - \{x\})) \cdot (V_{\mathbf{S}}(\{x\}) + V_{\mathbf{D}}(\{x\})). \end{aligned}$$

□

Condition 6: For each item $x \in I$,

$$V_{\mathbf{S}}(\{x\}) + V_{\mathbf{D}}(\{x\}) \leqq M,$$

where M is a positive real number. □

When $cM < 1$, the size of any highly co-occurrent itemset in a database in (k, c, M)-Δ' is bounded by a constant, which is determined by k, c, M, and b_m.

Lemma 18: Suppose that a database D is in (k, c, M)-Δ' with $cM < 1$, and Condition 3 is satisfied. Let X be a highly co-occurrent itemset in D. Then, the following inequality holds:

$$|X| \leqq k + \left\lfloor \frac{\log b_m}{\log(cM)} \right\rfloor.$$

Proof: Let $X = \{x_1, \ldots, x_t\}$ where $t > k$. Then, since D satisfies Condition 5, there is some $x \in X$ such that

$$V_{\mathbf{S}}(X) + V_{\mathbf{D}}(X) \leqq c \cdot (V_{\mathbf{S}}(X - \{x\}) + V_{\mathbf{D}}(X - \{x\})) \cdot (V_{\mathbf{S}}(\{x\}) + V_{\mathbf{D}}(\{x\})).$$

Without loss of generality, let x_t be such x. That is,

$$\begin{aligned} oc_D(X) \quad &= \quad V_{\mathbf{S}}(X) + V_{\mathbf{D}}(X) \\ &\leqq \quad c \cdot (V_{\mathbf{S}}(X - \{x_t\}) + V_{\mathbf{D}}(X - \{x_t\})) \cdot (V_{\mathbf{S}}(\{x_t\}) + V_{\mathbf{D}}(\{x_t\})). \end{aligned}$$

By repeating the same argument, we can obtain

$$\begin{aligned} & oc_D(X) \\ & \leqq \quad c^{t-k} \cdot (V_{\mathbf{S}}(\{x_1, \ldots, x_k\}) + V_{\mathbf{D}}(\{x_1, \ldots, x_k\})) \cdot \prod_{i=k+1}^{t} (V_{\mathbf{S}}(\{x_i\}) + V_{\mathbf{D}}(\{x_i\})) \\ & \leqq \quad c^{t-k} \cdot \prod_{i=k+1}^{t} (V_{\mathbf{S}}(\{x_i\}) + V_{\mathbf{D}}(\{x_i\})) \\ & \leqq \quad c^{t-k} \cdot M^{t-k}. \end{aligned}$$

From Condition 3, $b_m \leqq b \leqq oc_D(X)$. Thus,

$$\begin{aligned} b_m &\leqq c^{t-k} \cdot M^{t-k} \\ \log b_m &\leqq (t-k)\log(cM) \\ t &\leqq k + \left\lfloor \frac{\log b_m}{\log(cM)} \right\rfloor \\ |X| &\leqq k + \left\lfloor \frac{\log b_m}{\log(cM)} \right\rfloor . \end{aligned}$$

□

Let $l = k + \left\lfloor \frac{\log b_m}{\log(c\,M)} \right\rfloor$. For a given itemset, it can be checked whether the itemset is highly co-occurrent in D in $\mathcal{O}(||D||)$ time. Since there are at most $|I|^l$ itemsets of size less than or equal to l, all the highly co-occurrent itemsets can be computed in $\mathcal{O}(||D|| \cdot |I|^l)$ time.

Theorem 3: Suppose that a database D is in (k, c, M)-Δ' with $cM < 1$, and Condition 3 is satisfied. Then, all the highly co-occurrent itemsets in D can be computed in polynomial time in $||D||$. □

5.3 Type IV: Class (k, c, c', ϵ, M)-Γ'

Class (k, c, c', ϵ, M)-Γ' consists of all the (k, c)-sparse databases which satisfy the following two conditions.

Condition 7: For any itemset X such that $|X| > k$, there is some $x \in X$ which satisfies the following inequality:

$$V_{\mathbf{D}}(X) \leqq c' \cdot V_{\mathbf{D}}(X - \{x\}) \cdot V_{\mathbf{D}}(\{x\}),$$

where c' is a positive real number. □

Condition 8: For each item x,

$$\epsilon \leqq V_{\mathbf{S}}(\{x\}) \leqq M,$$

where ϵ and M are positive real numbers. □

When $cc' < 1$, the size of any highly co-occurrent itemset in a database in (k, c, c', ϵ, M)-Γ' is bounded by a constant, which is determined by k, c, c', ϵ, M, and b_m.

Lemma 19: Suppose that a database D is in (k, c, c', ϵ, M)-Γ' with $cc' < 1$, and Condition 3 is satisfied. Let X be a highly co-occurrent itemset in D. Then, the following inequality holds:

$$|X| \leqq k + \left\lfloor \frac{\log b_m + k\log(\epsilon(1-M))}{\log(cc')} \right\rfloor .$$

Proof: Let $X = \{x_1, \ldots, x_t\}$. Then,

$$\begin{aligned} oc_D(X) &= \frac{V_{\mathbf{S}}(X)}{E_{\mathbf{S}}(X)} \cdot \frac{V_{\mathbf{D}}(X)}{E_{\mathbf{D}}(X)} \\ &= \frac{V_{\mathbf{S}}(\{x_1, \ldots, x_t\})}{\prod_{i=1}^{t} V_{\mathbf{S}}(\{x_i\})} \cdot \frac{V_{\mathbf{D}}(\{x_1, \ldots, x_t\})}{\prod_{i=1}^{t} V_{\mathbf{D}}(\{x_i\})}. \end{aligned}$$

Suppose that $t > k$. Then, since D is (k, c)-sparse, we can obtain the following inequality by the same argument as in Lemma 17.

$$oc_D(X) \leqq c^{(t-k)} \cdot \epsilon^{-k} \cdot \frac{V_{\mathbf{D}}(\{x_1, \ldots, x_t\})}{\prod_{i=1}^{t} V_{\mathbf{D}}(\{x_i\})}$$

Also, since D satisfies Condition 7, and $V_{\mathbf{D}}(\{x\}) = 1 - V_{\mathbf{S}}(\{x\}) \geqq 1 - M$, we can obtain the following inequality from the above.

$$\begin{aligned} oc_D(X) &\leqq c^{(t-k)} \cdot \epsilon^{-k} \cdot c'^{(t-k)} \cdot (1-M)^{-k} \\ &= (cc')^{(t-k)} \cdot (\epsilon(1-M))^{-k} \end{aligned}$$

From Condition 3, $b_m \leqq b \leqq oc_D(X)$. Thus,

$$\begin{aligned} b_m &\leqq (cc')^{(t-k)} \cdot (\epsilon(1-M))^{-k} \\ \log b_m &\leqq (t-k)\log(cc') - k\log(\epsilon(1-M)) \\ \log b_m + k\log(\epsilon(1-M)) &\leqq (t-k)\log(cc') \\ t &\leqq k + \left\lfloor \frac{\log b_m + k\log(\epsilon(1-M))}{\log(cc')} \right\rfloor \\ |X| &\leqq k + \left\lfloor \frac{\log b_m + k\log(\epsilon(1-M))}{\log(cc')} \right\rfloor . \end{aligned}$$

□

Let $l = k + \left\lfloor \frac{\log b_m + k\log(\epsilon(1-M))}{\log(cc')} \right\rfloor$. For a given itemset, it can be checked whether the itemset is highly co-occurrent in D in $\mathcal{O}(||D||)$ time. Since there are at most $|I|^l$ itemsets of size less than or equal to l, all the highly co-occurrent itemsets can be computed in $\mathcal{O}(||D|| \cdot |I|^l)$ time.

Theorem 4: Suppose that a database D is in (k, c, c', ϵ, M)-Γ' with $cc' < 1$, and Condition 3 is satisfied. Then, all the highly co-occurrent itemsets in D can be computed in polynomial time in $||D||$. □

6. Conclusions

In Sect. 3, we have introduced the notion of (k, c)-sparsity of databases. Any database is (k, c)-sparse for some sufficiently high k or c. Thus, the (k, c)-sparsity is a general condition on databases. In fact, the test data in Refs. [1–3, 5, 6, 9, 11] are all (k, c)-sparse for some small k and c unless these algorithms need exponential time of the size of databases. Based on the notion of (k, c)-sparsity, we have proposed a subclass of databases. For a database in that subclass, we can efficiently find all the large itemsets.

In Sect. 4, we have defined alternative measures to the support, called co-occurrence. Of course, no definition of the co-occurrence achieves the *best* quality. In other words, according to the property of the database, database users have a chance to determine which definition of the co-occurrence they should use.

However, we have shown that finding all the highly co-occurrent itemsets is NP-hard under whichever measure we have defined. From these results, it has become clear that finding all the highly co-occurrent itemsets is impossible in polynomial time in the size of a database unless P=NP. It seems that the lack of the monotonicity such as "if an itemset has some property, then all the subsets of the itemset has the same property" makes the problem more difficult than to find all the large itemsets.

In Sect 5, we have proposed subclasses of databases for which we can efficiently find all the highly co-occurrent itemsets. These subclasses are also defined based on the notion of (k, c)-sparsity. To propose weaker conditions on databases is the future work.

Acknowledgment

We would like to thank Research Associate Ryuichi Nakanishi of Nara Institute of Science and Technology for his valuable suggestions and discussions.

References

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," Proc. ACM SIGMOD Int'l Conf. on the Management of Data, pp.207–216, May 1993.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo, "Fast discovery of association rules," Advances in Knowledge Discovery and Data Mining, pp.307–328, AAAI/MIT Press, 1996.

[3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," Proc. 20th Int'l Conf. on Very Large Data Bases, pp.487–499, Sept. 1994.

[4] C.C. Aggarwal and P.S. Yu, "A new framework for itemset generation," Proc. 17th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems, pp.18–24, June 1998.

[5] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," Proc. ACM SIGMOD Int'l Conf. on Management of Data, pp.255–264, May 1997.

[6] D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen, "Data mining, hypergraph transversals, and machine learning," Proc. 16th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems, pp.209–216, May 1997.

[7] Y.D. Kwon, R. Nakanishi, M. Ito, and M. Nakanishi, "Computational complexity of finding meaningful association rules", IEICE Trans. Fundamentals, vol.E82–A, no.9, pp.1945–1952, Sep. 1999.

[8] H. Mannila, H. Toivonen, and A.I. Verkamo, "Efficient algorithms for discovering association rules," Proc. AAAI Workshop in Knowledge Discovery in Databases, pp.144–155, July 1994.

[9] J.S. Park, M.S. Chen, and P.S. Yu, "An effective hash-based algorithm for mining association rules," Proc. ACM SIGMOD Int'l Conf. on Management of Data, pp.175–186, May 1995.

[10] C. Silverstein, S. Brin, and R. Motwani, "Beyond Market Baskets: Generalizing Association Rules to Dependence Rules," Data Mining and Knowledge Discovery, vol.2, pp.39–68, 1998.

[11] A. Savasere, E. Omiecinski, and S. Navathe, "An efficient algorithm for mining association rules in large databases," Proc. 21st Int'l Conf. on Very Large Data Bases, pp.432–444, Sept. 1995.