NAIST-IS-DT9561209

Doctor's Thesis

Computational Complexity of Finding Correlated Items in Market Basket Databases

Yeon-Dae Kwon

February 7, 2000

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

Doctor's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
DOCTOR of ENGINEERING

Yeon-Dae Kwon

Thesis committee: Minoru Ito, Professor
Shunsuke Uemura, Professor
Hiroyuki Seki, Professor

Computational Complexity of Finding Correlated Items in Market Basket Databases[*]

Yeon-Dae Kwon

Abstract

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. 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 sets of correlated items first, and then to construct meaningful association rules from the sets of correlated items. This thesis discusses the issues associated with the problem of mining meaningful association rules.

The notion of *support* has been proposed as a measure which indicates a degree of correlation among the items in a given itemset. An itemset is called *large* if its support exceeds a given threshold. Although a number of algorithms for computing all the large itemsets have been proposed, the computational complexity of them is scarcely discussed. The performances of most of the algorithms are estimated only by empirical evaluation through benchmark tests. This thesis defines the *large itemset problem* formally as deciding whether there exists a large itemset with a given size, and shows the NP-completeness of the problem. 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

[*] Doctor's Thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT9561209, February 7, 2000.

size of a database unless P=NP. Furthermore, this thesis proposes a subclass of databases for which we can efficiently find all the large itemsets.

Also, several disadvantages of the support have been pointed out. For example, the support of an itemset tends to be high if the itemset contains items with high supports, regardless of the relationship among the items. This thesis proposes 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 which do not contain any item in a given itemset,

and so on. For each measure, an itemset is called *highly co-occurrent* if the value indicating the correlation among the items exceeds a given threshold. This thesis defines the *highly co-occurrent itemset problem* formally as deciding whether there exists a highly co-occurrent itemset with a given size, and shows that the problem is NP-complete under whichever measure. Furthermore, this thesis proposes subclasses of databases for which we can efficiently find all the highly co-occurrent itemsets.

Keywords:

data mining, large itemset, highly co-occurrent itemset, meaningful association rule, computational complexity

Acknowledgments

I have been fortunate to have received support and assistance from many individuals. I would especially like to thank Professor Minoru Ito for his invaluable support and discussions throughout the work. I am also grateful to Professors Shunsuke Uemura and Hiroyuki Seki for their invaluable suggestions and discussions on the work.

I would like to thank Assistant Professor Yasunori Ishihara of Osaka University for his valuable comments and continuous support throughout the work. I am also obliged to thank Research Associate Ryuichi Nakanishi for his insightful suggestions on the work. I deeply thank Mr. Shougo Shimizu for his kind support and encouragement throughout the work.

I would like to thank Mr. Nobutaka Suzuki and Mr. Toshiyuki Morita for their kind supports. I would like to thank all the members of Ito Laboratory of Nara Institute of Science and Technology.

I am grateful to the "Heiwa Nakajima Foundation" for its financial support. I am also grateful to Mr. Ishida and Mrs. Hoshino of HNF for their hearty encouragements.

Finally, I sincerely thank my mother and my older siblings for caring very much about me for a long time.

List of Publications

Journal Papers

[1] Y.D. Kwon, R. Nakanishi, M. Ito, and M. Nakanishi, "Computational complexity of finding meaningful association rules," IEICE Transactions on Fundamentals on Electronics, Communications and Computer Sciences, vol. E82-A, no. 9, pp. 1945–1952, Sep. 1999.

[2] Y.D. Kwon, Y. Ishihara, S. Shimizu, and M. Ito, "Computational complexity of finding highly co-occurrent itemsets in market basket databases," IEICE Transactions on Fundamentals on Electronics, Communications and Computer Sciences (submitted).

Workshops

[2] Y.D. Kwon, R. Nakanishi, M. Ito, and M. Nakanishi, "A subclass of databases whose large itemsets can be computed efficiently," IEICE Technical Report, COMP98-9, pp. 1–8, May 1998 (in Japanese).

[3] Y.D. Kwon, R. Nakanishi, M. Ito, and M. Nakanishi, "Computational complexity of the strongly collective itemset problem," IEICE Technical Report, COMP98-86, pp. 25–32, March 1999 (in Japanese).

[4] Y.D. Kwon, Y. Ishihara, S. Shimizu, and M. Ito, "Computational complexity of finding highly co-occurrent itemsets," IEICE Technical Report COMP99-89, March 2000 (to appear).

Contents

List of Figures

Chapter 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 thesis, 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 Figure 1.1. A transaction t_1 indicates that bread, ham, bacon, and milk were purchased together by a customer in a single visit to a store. By examining D_1, we can identify a rule that "if bacon is purchased in a transaction, then it is likely that lettuce and tomato will also be purchased in that transaction." This rule indicates the high correlation among bacon, lettuce, and tomato. 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. For example, the rule stated above can be represented as $\{\text{bacon}\} \Rightarrow \{\text{lettuce, tomato}\}$. 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 *support* of a rule $X \Rightarrow Y$ is the fraction of transactions that contain both X and Y. The *confidence* of a rule $X \Rightarrow Y$ is the fraction of

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

Figure 1.1. A market basket database D_1.

transactions containing X that also contain Y. For the association rule $X \Rightarrow Y$ to hold, $X \cup Y$ must be large and the confidence of the rule must exceed a given confidence threshold. In this thesis, we will refer to measuring an association rule by support and confidence as the *support-confidence framework*.

Example 1.1 Consider D_1 shown in Figure 1.1. Let $X = \{\text{bacon}\}$ and $Y = \{\text{lettuce, tomato}\}$. The number of transactions in D_1 is 6. Transactions which contain both X and Y are t_2, t_4, and t_5, and then the number of transactions which contain both X and Y is 3. Therefore, the support of a rule $X \Rightarrow Y$ is $^{1}/_{2}$. Transactions which contain X are t_1, t_2, t_4, and t_5, and then the number of transactions which contain X is 4. Also, the number of transactions which contain both X and Y is 3. Therefore, the confidence of the rule $X \Rightarrow Y$ is $^{3}/_{4}$. That is, 75% of transactions that purchase {bacon} also purchase {lettuce} and {tomato}. □

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 [1–4, 10, 11, 13–15, 17, 18, 20]. The problem of mining meaningful association rules is to find all the association rules that have support and confidence greater than or equal to certain user-defined thresholds called *minimum support* and *minimum confidence* respectively. This problem can be decomposed into two subproblems:

1. Find all the large itemsets with support greater than or equal to the minimum support.

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}

Figure 1.2. A market basket database D_2.

2. Construct rules with confidence greater than or equal to the minimum confidence from the large itemsets in step 1. Having determined the large itemsets, the second problem is rather straightforward. 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 $\{y, z\} \Rightarrow \{x\}$.

Although a number of algorithms for computing all the large itemsets have been proposed [1, 3, 4, 6, 7, 9, 15, 17, 20], the computational complexity is scarcely discussed. The performances of most of the algorithms are estimated only by empirical evaluation through benchmark tests. This thesis defines the *large itemset problem* formally as deciding whether there exists a large itemset with a given size, and shows 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, this thesis introduces the notion of (k, c)-*sparsity* of databases. Intuitively, (k, c)-sparsity of a database means that the supports of itemsets of size k or more are considerably low in the database. The value of c represents a degree of sparsity. Using (k, c)-sparsity, this thesis proposes a subclass of databases for which we can efficiently find all the large itemsets.

Also, several disadvantages of the support have been pointed out in References [5, 6, 16]. 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 1.2 Consider D_2 shown in Figure 1.2. Suppose that the minimum support is 0.3. The support of an itemset $Z = \{\text{coffee, eggs}\}$ is 0.5, and hence 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. □

This thesis proposes 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 which 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 Reference [5] and dependence in Reference [16].

For each measure, an itemset is called *highly co-occurrent* if the value indicating the correlation among the items exceeds a given threshold. This thesis also shows that finding all the highly co-occurrent itemsets is still NP-hard under whichever measure, including collective strength.

Furthermore, using (k, c)-sparsity, this thesis proposes subclasses of databases for which we can efficiently find all the highly co-occurrent itemsets.

The rest of this thesis is organized as follows. Chapter 2 formally defines the large itemsets problem based on the support-confidence framework and shows the NP-completeness of the problem. Furthermore, we propose a subclass of databases for which we can efficiently find all the large itemsets. Chapter 3 defines 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. Furthermore, we propose subclasses of databases for which we can efficiently find all the highly co-occurrent itemsets. Chapter 4 summarizes this thesis.

Chapter 2

Computational Complexity of Finding All the Large Itemsets

This chapter formally defines the large itemset problem based on the support-confidence framework and shows the NP-completeness of the problem. 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 propose a subclass of databases for which we can efficiently find all the large itemsets.

2.1 Large Itemsets and Meaningful Association Rules

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 $X \subseteq t$ [1]. By $sup_D(X)$, we mean the number of transactions in D that support X. For a given positive integer s $(0 \leq s \leq |D|)$, called *minimum support number*, we say that an itemset X is *large* in D if $sup_D(X) \geq s$. 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 $(0 \leq r \leq 1)$, called *minimum support rate*, we say that an itemset X is *large* in D if $supr_D(X) \geq r$. 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.

Example 2.1 Consider an MBD D_1 shown in Figure 1.1. Suppose that the minimum support rate r is 0.3. Consider two itemsets $X = \{\text{bread, ham, bacon}\}$ and $Y = \{\text{bacon, lettuce, tomato}\}$. X is supported only by t_1, and hence $supr_{D_1}(X) \approx 0.17 < r$. Therefore, X is not large in D_1. On the other hand, Y is supported by t_2, t_4, and t_5, and hence $supr_{D_1}(Y) = 0.5 > r$. Therefore, Y is large in D_1. □

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. The *support rate* $supr_D(X \Rightarrow Y)$ of an association rule $X \Rightarrow Y$ in D is defined as $supr_D(X \cup Y)$. For a given non-negative real number r $(0 \leq r \leq 1)$, called *minimum support rate*, we say that an association rule $X \Rightarrow Y$ is *large* in D if $supr_D(X \Rightarrow Y) \geq r$. The *confidence* $conf_D(X \Rightarrow Y)$ of an association rule $X \Rightarrow Y$ in D is defined as follows [1]:

$$conf_D(X \Rightarrow Y) \triangleq \frac{sup_D(X \cup Y)}{sup_D(X)}.$$

For a given non-negative real number f $(0 \leq f \leq 1)$, called *minimum confidence*, we say that an association rule $X \Rightarrow Y$ is *confident* in D if $conf_D(X \Rightarrow Y) \geq f$.

The support and confidence have been proposed as criteria for mining an association rule by Agrawal et al. [1]. We propose another criterion, called right-hand side size. The *right-hand side (rhs) size* of an association rule $X \Rightarrow Y$ is the number of items in Y, i.e., $|Y|$. For a given positive integer w ($0 \leq w \leq |I|$), called *minimum rhs size*, we say that an association rule $X \Rightarrow Y$ has the rhs size if $|Y| \geq w$.

For a given r, f, and w, we say that an association rule $X \Rightarrow Y$ is *meaningful with respect to* r, f, and w, or simply *meaningful* when r, f, and w are clear from the context, if the rule satisfies all of the following conditions C1, C2, and C3.

C1. $X \Rightarrow Y$ is large.

C2. $X \Rightarrow Y$ is confident.

C3. $X \Rightarrow Y$ has the minimum rhs size.

The intuitive meaning of each condition is as follows. Condition C1 states that X and Y are purchased together frequently. Condition C2 states that if X is purchased, then Y is likely to be purchased together. Condition C3 is required because we want to predict as much information as possible.

Example 2.2 Consider D_1 shown in Figure 1.1. Suppose that the minimum support rate r is 0.3, the minimum confidence f is 0.6, and the minimum rhs size w is 2. Let us consider the following four association rules.

1. ar_1: {bread} $\Rightarrow$ {ham, bacon}
   ar_1 is not large because $supr_{D_1}(ar_1) = 1/6 < r$. That is, ar_1 does not satisfy C1. This indicates that bread, ham, and bacon are rarely purchased together.

2. ar_2: {bacon} $\Rightarrow$ {milk}
   ar_2 is large because $supr_{D_1}(ar_2) = 1/3 \geq r$. That is, ar_2 satisfies C1. However, ar_2 is not confident because $conf_{D_1}(ar_2) = 1/2 < f$. This indicates that when bacon is purchased, the probability that milk is also purchased together is not so high.

3. ar_3: {bacon, lettuce, tomato} $\Rightarrow \emptyset$
   ar_3 is large because $supr_{D_1}(ar_3) = 1/2 \geq r$, and also confident because $conf_{D_1}(ar_3) = 1 \geq f$. That is, ar_3 satisfies C1 and C2. However, it does not have the given rhs size because its rhs size is $0 < w$. That is, even if we can predict that many customers purchase bacon, lettuce, and tomato

together (in such a situation that the store manager sells bacon, lettuce, and tomato at a special bargain sale), there is no other item (in general, less than w items) which can be expected to be purchased together. Thus, ar_3 can be considered to be useless.

4. ar_4: {bacon} ⇒ {lettuce, tomato}
   ar_4 is large because $supr_{D_1}(ar_4) = 1/2 \geq r$, and is also confident because $conf_{D_1}(ar_4) = 3/4 \geq f$, and has the given rhs size because its rhs size is $2 \geq w$. Since ar_4 satisfies all of the conditions C1, C2, and C3, ar_4 is a meaningful association rule. □

If we can obtain a meaningful association rule, then it is useful for the following case, for instance. Consider the case that we have obtained a meaningful association rule {bacon} ⇒ {lettuce, tomato} by analyzing an MBD of a grocery store. Then, when the store manager sells bacon at a special bargain sale, he can avoid being out of stock by having a lot of lettuce and tomato in stock.

In Figure 2.1, we propose an algorithm which computes all the meaningful association rules in a given database.

Theorem 2.1 All the meaningful association rules can be computed using Procedure Find-Mar shown in Figure 2.1.

Proof: (soundness): We show that every association rule $X \Rightarrow Y$ obtained by Procedure Find-Mar shown in Figure 2.1 satisfies all of the conditions C1, C2, and C3. Since every rule $X \Rightarrow Y$ is obtained as a result of Procedure Find-Mar, $X \Rightarrow Y$ must satisfy the *if* statement in step 9. That is, $|Y| \geq w$, and $X \Rightarrow Y$ satisfies C3. Since the *if* statement in step 9 is executed for each association rule in AR obtained in step 7, $X \Rightarrow Y$ must be in AR. That is, $X \Rightarrow Y$ must satisfy the *if* statement in step 5, and $conf_D(X \Rightarrow Y) \geq c$. Hence $X \Rightarrow Y$ satisfies C2. Since the *if* statement in step 5 is executed for each itemset in LI, $X \cup Y$ must be in LI. That is, $supr_D(X \Rightarrow Y) \geq r$. Hence $X \Rightarrow Y$ is large, and satisfies C1. Consequently, every association rule obtained by Procedure Find-Mar is meaningful.

(completeness): We show that all the meaningful association rules can be obtained by Procedure Find-Mar shown in Figure 2.1. Suppose that an association rule

```
procedure FIND-MAR
Input   :  an MBD $D$, a set of items $I$, a minimum support rate $r$,
           a minimum confidence $c$, a minimum rhs size $w$
Output  :  all the meaningful association rules in $D$
begin
Compute the set $LI$ of all the large itemsets
whose support rates are greater than or equal to $r$.
$AR := \emptyset$;
for all $Z \in LI$ do
    for all $X \subset Z$ do
            if $\frac{sup_D(Z)}{sup_D(X)} \geq c$ then
                $Y := Z - X$;
                $AR := AR \cup \{X \Rightarrow Y\}$;
for all $X \Rightarrow Y \in AR$ do
    if $|Y| \geq w$ then
            Output the association rule $X \Rightarrow Y$;
end
```

Figure 2.1. Procedure FIND-MAR.

$X' \Rightarrow Y'$ satisfies all of the conditions C1, C2, and C3. By C1, $X' \Rightarrow Y'$ is large. That is, $supr_D(X' \Rightarrow Y') \geq r$. Since the support rate of $X' \cup Y'$ is at least r, $Z' = X' \cup Y'$ is a large itemset. Thus, Z' is added to LI at step 1. By C2, $X' \Rightarrow Y'$ is confident. That is, $conf_D(X' \Rightarrow Y') \geq c$. When $Z = Z'(= X' \cup Y') \in LI$ and $X = X'$ in step 5, the *if* statement holds. Thus, $X' \Rightarrow Y'$ (i.e., $X \Rightarrow Y$) is added to AR in step 7. By C3, $X' \Rightarrow Y'$ has the minimum rhs size. That is, $|Y'| \geq w$, and $X \Rightarrow Y$ is output at step 10. □

As seen in Procedure FIND-MAR, once we have obtained all the large itemsets, then all the meaningful association rules can be easily constructed from the large itemsets. Therefore, in the following sections, we concentrate on finding all the large itemsets.

2.2 NP-Completeness of the Large Itemset Problem

This section defines the large itemset problem, and shows the NP-completeness of the problem.

Definition 2.1 (large itemset problem)**:** Given an MBD D, a minimum support rate r (or minimum support number s), and a positive integer h called *minimum itemset size*, is there a large itemset in D of size at least h? □

Let us first prove that the large itemset problem is in NP.

Lemma 2.1 The large itemset problem is in NP.

Proof: Note that if an itemset X is large in an MBD D, then every subset of X is also large in D. Thus if there is a large itemset in D of size at least h, then there must be a large itemset in D of size exactly h. We can guess an itemset of size h in NP time. After that, it can be tested in a straightforward way in linear time to $||D||$ whether the itemset is large in D. Hence Lemma 2.1 holds. □

In the following, we show the NP-hardness of the large itemset problem by reducing the well-known clique problem [12] to the large itemset problem. To make the reduction simple, we suppose that a minimum support number s is given in the large itemset problem.

As an instance of the clique problem, let us consider an undirected graph $G = (V, E)$ and a positive integer k. Let $V = \{v_1, \ldots, v_n\}$. From G and k, we construct an instance of the large itemset problem; that is, a set of items I, a minimum support number s, a minimum itemset size h, and an MBD D_G, as follows:

(1) Let $I = \{a_1, \ldots, a_n\} \cup \{\bar{a}_1, \ldots, \bar{a}_n\}$. Intuitively, a_i and $\bar{a}_i$ mean that "v_i is a member of the clique" and "v_i is not a member of the clique," respectively. Thus, a_i is incompatible with $\bar{a}_i$. For example, when $n = 3$, an itemset $\{a_1, \bar{a}_2, a_3\}$ means that "the clique consists of v_1 and v_3."

(2) Let $s = k + (n+1)\binom{n}{2}$.

(3) Let $h = n$.

To define D_G, let $\tilde{\cup}$ denote the union operator to multisets (i.e., the union operator which counts multiple occurrences of elements).

(4) Let $D_G = D_V \tilde{\cup} \underbrace{D_C \tilde{\cup} D_C \tilde{\cup} \cdots \tilde{\cup} D_C}_{n+1}$, where D_V and D_C are sets of transactions which will be defined later.

Before defining D_V and D_C, we provide their intuitive meanings. An itemset X is called *consistent* if X contains exactly one of a_i and $\bar{a}_i$ for each i $(1 \leq i \leq n)$. Note that the size of any consistent itemset is n. Then it will turn out that D_V and D_C have the following properties, respectively.

Property 1 G has a k-clique if and only if there exists a consistent itemset X such that $sup_{D_V}(X) \geq k$. □

Property 2 Let X be an itemset of size n. Then, $sup_{D_C}(X) = \binom{n}{2}$ if X is consistent, and $sup_{D_C}(X) \leq \binom{n}{2} - 1$ otherwise. □

Since $sup_{D_G}(X) = sup_{D_V}(X) + (n+1)sup_{D_C}(X)$ from the definition of D_G, it follows from the above properties that G has a k-clique if and only if there exists an itemset X such that X is consistent and $sup_{D_G}(X) \geq k + (n+1)\binom{n}{2} = s$, that is, X is large in D_G. Hence we will have the desired result so that the large itemset problem is NP-hard (the details will be given in Lemma 2.6). Now let us define D_V and D_C, as follows:

- $D_V \triangleq \{V_1, \ldots, V_n\}$,

  where $V_i = \{a_i\} \cup \{a_j \mid (v_i, v_j) \in E\} \cup \{\bar{a}_j \mid i \neq j \text{ and } 1 \leq j \leq n\}$.

- $D_C \triangleq \{I - \{\alpha_1, \alpha_2\} \mid \alpha_1, \alpha_2 \in I \text{ and } \alpha_1 \neq \alpha_2\} - \{I - \{a_i, \bar{a}_i\} \mid 1 \leq i \leq n\}$

  That is, D_C is the set of itemsets of size $2n-2$ other than $I - \{a_i, \bar{a}_i\}$ with i $(1 \leq i \leq n)$. Then, the size of D_C is $\binom{2n}{2} - n$.

Clearly, we can construct I, s, h, and D_G in polynomial time in the size of G and k.

Example 2.3 Consider the graph $G = (V, E)$ in Figure 2.2. The constructed transactions in D_V are illustrated in Figure 2.3, and the transactions in D_C are in Figure 2.4 (D_A and D_I shown in Figure 2.4 will be defined later). □

To prove Property 1, we provide the following lemma.

Lemma 2.2 For a consistent itemset X, let $A = \{i \mid a_i \in X\}$ and $\bar{A} = \{i \mid \bar{a}_i \in X\}$. Then, $sup_{D_V}(X) = |A|$ if and only if

(i) for every $i \in A$, V_i supports X, and

(ii) there is no $j \in \bar{A}$ such that V_j supports X.

Proof: The *if* part is obvious. We prove the *only if* part. Assume that $sup_{D_V}(X) = |A|$. Let $j \in \bar{A}$. Since $\bar{a}_j$ is in X but not in V_j, V_j does not support X. Thus (ii) holds. Since $sup_{D_V}(X) = |A| = |V| - |\bar{A}|$, the fact (ii) implies that for every $i \in A$, V_i supports X. That is, (i) holds. □

Now, we show Property 1 by the following lemma.

Lemma 2.3 G has a k-clique if and only if there exists a consistent itemset X such that $sup_{D_V}(X) \geq k$.

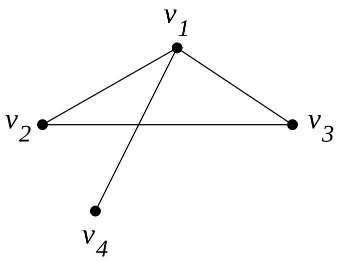

Figure 2.2. A graph G.

	a_1	a_2	a_3	a_4	$\bar{a}_1$	$\bar{a}_2$	$\bar{a}_3$	$\bar{a}_4$
V_1	●	●	●	●	–	●	●	●
V_2	●	●	●	–	●	–	●	●
V_3	●	●	●	–	●	●	–	●
V_4	●	–	–	●	●	●	●	–

$$\left(\begin{array}{lll} \bullet & : & \text{the transaction contains the item} \\ - & : & \text{the transaction does not contain the item} \end{array}\right)$$

Figure 2.3. Transactions in D_V.

		a_1	a_2	a_3	a_4	$\bar{a}_1$	$\bar{a}_2$	$\bar{a}_3$	$\bar{a}_4$	
D_A	$V_{a_1 a_2}$	–	–	●	●	●	●	●	●	D_C
	$V_{a_1 a_3}$	–	●	–	●	●	●	●	●	
	$\vdots$									
	$V_{a_4 \bar{a}_3}$	●	●	●	–	●	●	–	●	
	$V_{a_1 \bar{a}_1}$	–	●	●	●	–	●	●	●	D_I
	$V_{a_2 \bar{a}_2}$	●	–	●	●	●	–	●	●	
	$\vdots$									
	$V_{a_4 \bar{a}_4}$	●	●	●	–	●	●	●	–	

Figure 2.4. Transactions in D_C.

Proof: (*Only if* part): Assume that G has a k-clique. Without loss of generality, let the clique consist of nodes $v_1, \ldots, v_k$. Let $X = \{a_1, \ldots, a_k, \bar{a}_{k+1}, \ldots, \bar{a}_n\}$. Note that X is consistent. Since $v_1, \ldots, v_k$ form a clique in G, it holds that $(v_i, v_j) \in E$ for all distinct i and j such that $1 \leq i, j \leq k$. Thus it follows from the definition of V_i that $\{a_1, \ldots, a_k\} \subseteq V_i$ for every i $(1 \leq i \leq k)$. Furthermore, it also follows that $\{\bar{a}_{k+1}, \ldots, \bar{a}_n\} \subseteq V_i$ for every i $(1 \leq i \leq k)$. Therefore, all of $V_1, \ldots, V_k$ support X. On the other hand, none of $V_{k+1}, \ldots, V_n$ supports X since $\bar{a}_{k+i}$ is in X, but not in V_{k+i} for i $(1 \leq i \leq n-k)$. Thus, $sup_{D_V}(X) = k$.
(*If* part): Assume that there exists a consistent itemset X such that $sup_{D_V}(X) = u \geq k$. Without loss of generality, we assume that X is supported by $V_1, \ldots, V_u$ but not by $V_{u+1}, \ldots, V_n$. From the definition of V_i, $\bar{a}_i \notin V_i$. Since X is consistent, X contains exactly one of a_i and $\bar{a}_i$. From $X \subseteq V_i$ $(1 \leq i \leq u)$, we have $a_i \in X$. Therefore, $\{a_1, \ldots, a_u\} \subseteq V_i$. From the definition of D_V, $(v_i, v_j) \in E$ for all distinct i and j such that $1 \leq i, j \leq u$. Consequently, the nodes $v_1, \ldots, v_u$ form a u-clique in G. Since $u \geq k$, G has a k-clique. □

To prove Property 2, we define D_A and D_I.

$$\begin{aligned} D_A &\triangleq \{I - \{\alpha_1, \alpha_2\} \mid \alpha_1, \alpha_2 \in I \text{ and } \alpha_1 \neq \alpha_2\} \\ D_I &\triangleq \{I - \{a_i, \bar{a}_i\} \mid 1 \leq i \leq n\} \end{aligned}$$

Clearly, $D_C = D_A - D_I$, and $sup_{D_C}(X) = sup_{D_A}(X) - sup_{D_I}(X)$ for all X.

Lemma 2.4 Let X be an itemset of size n. Then it holds that $sup_{D_A}(X) = \binom{n}{2}$.

Proof: Note that D_A is the set of all itemsets Y such that Y is obtained by deleting two distinct items from I. If Y supports X, then the deleted two items must be in $I - X$. Since $|I| = 2n$ and $|X| = n$, it holds that $|I - X| = n$. Thus the number of possible deletions of two items from $I - X$ is exactly $\binom{n}{2}$, and hence Lemma 2.4 holds. □

Next, we show Property 2 by the following lemma.

Lemma 2.5 For an itemset X of size n, $sup_{D_C}(X) = \binom{n}{2}$ if X is consistent, and $sup_{D_C}(X) \leq \binom{n}{2} - 1$ otherwise.

Proof: Since (1) $sup_{D_C}(X) = sup_{D_A}(X) - sup_{D_I}(X)$ and (2) $sup_{D_A}(X) = \binom{n}{2}$ by Lemma 2.4, it suffices to show that $sup_{D_I}(X) = 0$ if and only if X is consistent.
(*If* part): Assume that X is consistent. Then X contains exactly one of a_i and $\bar{a}_i$ for every i $(1 \leq i \leq n)$. Thus it follows from the definition of D_I that no transaction in D_I supports X. That is, $sup_{D_I}(X) = 0$.
(*Only if* part): Assume that X is not consistent. Then, there exists i $(1 \leq i \leq n)$ such that (i) $a_i, \bar{a}_i \in X$ or (ii) $a_i, \bar{a}_i \notin X$. Since $|X| = n$, if $a_i, \bar{a}_i \in X$ hold for some i, then $a_j, \bar{a}_j \notin X$ hold for some j such that $i \neq j$. Thus, without loss of generality, we assume that $a_j, \bar{a}_j \notin X$ for some j. Then, the transaction $I - \{a_j, \bar{a}_j\}$ in D_I supports X. Therefore $sup_{D_I}(X) \geq 1$. □

Lemma 2.6 *G has a k-clique if and only if there exists a large itemset of size n in D_G.*

Proof: (*Only if* part): Assume that G has a k-clique. By Lemma 2.3, there exists a consistent itemset X such that $sup_{D_V}(X) \geq k$. Since X is consistent, $sup_{D_C}(X) = \binom{n}{2}$ by Lemma 2.5. Since $D_G = D_V \tilde{\cup} \underbrace{D_C \tilde{\cup} D_C \tilde{\cup} \cdots \tilde{\cup} D_C}_{n+1}$, it holds that $sup_{D_G}(X) = sup_{D_V}(X) + (n+1)sup_{D_C}(X) \geq k + (n+1)\binom{n}{2} = s$. Thus, X is large in D_G.
(*If* part): Let X be a large itemset of size n in D_G. Assume that X is not consistent. Then, since (1) $sup_{D_V}(X) \leq |D_V| = n$ from the definition, and (2) $sup_{D_C}(X) \leq \binom{n}{2} - 1$ by Lemma 2.5, it follows that $sup_{D_G}(X) = sup_{D_V}(X) + (n+1)sup_{D_C}(X) \leq n + (n+1)(\binom{n}{2} - 1) = (n+1)\binom{n}{2} - 1 < s$. This, however, contradicts that X is large in D_G. Thus X must be consistent. Hence $sup_{D_C}(X) = \binom{n}{2}$ by Lemma 2.5. Furthermore, since X is large in D_G, it holds that $sup_{D_G}(X) \geq s$. From these two facts, $sup_{D_V}(X) = sup_{D_G}(X) - (n+1)sup_{D_C}(X) \geq s - (n+1)\binom{n}{2} = k$. By Lemma 2.3, G has a k-clique. □

NOTE: The proof of NP-completeness of the large itemset problem can be easily shown by a reduction from the k-balanced complete bipartite subgraph (*k-bcbs*) problem [8]. However, the k-bcbs problem is not a well-known problem. Moreover, Reference [8] says that the NP-completeness of the k-bcbs problem can be proved by a reduction from the k-clique problem, but the proof has not been published. This is why we proved the NP-completeness of the large itemset problem by the reduction from the k-clique problem.

Note that the number of transactions in D_G is $|D_V| + (n+1)|D_C| = n + (n+1)(\binom{2n}{2} - n) = 2n^3 - n = \mathcal{O}(n^3)$. From the above discussions, we have the following theorem.

Theorem 2.2 The large itemset problem is NP-complete even if the number of transactions in a given MBD is $\mathcal{O}(|I|^3)$.

NOTE: In the proof of Lemma 2.6, the minimum support rate of the constructed instance (D_G, s, h) is $\dfrac{s}{|D_G|} \approx 1/4$. However, this rate is not essential. When we consider I as a transaction, I supports any itemset. Similarly, $\emptyset$ supports no itemset. For a given minimum support rate r $(0 < r < 1)$ and an instance (D, s, h), by adding the adequate numbers of I and $\emptyset$ to D as transactions, we can reconstruct an instance (D', r, h) preserving the reduction. For example, suppose that r is a rational number b/a, where a and b are positive integers such that $a > b$. Let $x \geq 0$ be the number of I added to D, and $y \geq 0$ be the number of $\emptyset$ added to D. By solving the following equation:

$$\frac{s+x}{|D|+x+y} = r = \frac{b}{a} = \frac{b|D|}{a|D|},$$

we have $x = b|D| - s$ and $y = (a-b-1)|D| + s$. Both x and y are polynomial in $|D|$, s, a, and b.

From Theorem 2.2, we cannot compute the maximum number of the elements of large itemsets in feasible time (i.e., polynomial time). Accordingly in Section 2.3, we propose subclasses of databases of which the maximum number of the elements of large itemsets can be computed efficiently.

As a related problem to the large itemset problem, we will consider the following problem.

Definition 2.2 (large intersection problem)**:** Given a finite family S of finite sets, a positive integer l called *minimum selection number*, and a positive integer c called *minimum intersection size*, are there l distinct sets $s_1, \ldots, s_l$ in S such that $s_1 \cap \cdots \cap s_l$ contains at least c elements? □

Corollary 2.1 The large intersection problem is NP-complete.

Proof: It is easy to see that the problem is in NP. In order to prove NP-hardness of the large intersection problem, we show a reduction from the large itemset problem to the large intersection problem. Let (D, s, h) be an instance of the large itemset problem. Let $D = \{t_1, \ldots, t_m\}$. First, we introduce new distinct m items $c_1, \ldots, c_m$. For each transaction t_i $(1 \leq i \leq m)$ in D, let $t'_i = t_i \cup \{c_i\}$. Let us construct an instance (S, l, c) of the large intersection problem, as follows:

(1) Let $S = \{t'_i \mid 1 \leq i \leq m\}$. Note that S is not a multiset but a set of itemsets because of the unique item c_i $(1 \leq i \leq m)$.

(2) Let $l = s$.

(3) Let $c = h$.

Obviously (S, l, c) can be constructed in polynomial time in the size of (D, s, h). The correctness of the reduction can be easily shown from the fact that for all distinct $i_1, \ldots, i_j$ such that $1 \leq i_1, \ldots, i_j \leq m$, $t_{i_1} \cap \cdots \cap t_{i_j} = t'_{i_1} \cap \cdots \cap t'_{i_j}$. □

2.3 Subclass of Databases for which All the Large Itemsets can be Computed Efficiently

In this section, we introduce the notion of (k, c)-sparsity, where k is a positive integer and c is a positive real number. Intuitively, (k, c)-sparsity of a database means that the supports of itemsets of size k or more are considerably low in the database. The value of c represents a degree of sparsity. Then, using (k, c)-sparsity, we propose a subclass (k, c, M)-Δ of MBDs, where M is a positive real number. For a database in (k, c, M)-Δ, we can efficiently find all the large itemsets.

First, we define the notion of (k, c)-sparsity as follows.

Definition 2.3 ((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:

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

□

Note that the infinite union of (k, c)-sparse MBDs ($1 \leq k < \infty$, $0 < c < \infty$) coincides with the class of all the MBDs.

As an example, consider an MBD of a hot dog stand. In general, the number of items which one customer purchases in a visit to a hot dog stand tends to be smaller than that of items to a supermarket, a grocery shop and so on. Thus, in an MBD of a hot dog stand, for a positive integer k, the support rates of itemsets of size k or more may be considerably low. Such databases are probably (k, c)-sparse.

Example 2.4 First, consider an MBD D_3 shown in Figure 2.5. Let $k = 2$ and $c = 1$. The only itemset of size 4 in D_3 is $X = \{\text{hot dog, popcorn, cola, beer}\}$, and

$$supr_{D_3}(X) = \frac{1}{12} \leq supr_{D_3}(X - \{\text{beer}\}) \cdot supr_{D_3}(\{\text{beer}\}) = \frac{1}{6} \cdot \frac{1}{2}.$$

Therefore, X and x = beer satisfy the inequality in Definition 2.3. It is easy to see that the inequality in Definition 2.3 holds for all the itemsets of size 3. Therefore, D_3 is $(2, 1)$-sparse.

Next, consider an MBD D_2 shown in Figure 1.2. Let $X' = \{\text{bacon, eggs, milk}\}$. Then, the inequality is not satisfied for any $x \in X'$. That is, X' does not satisfy Definition 2.3. Hence, D_2 is not $(2, 1)$-sparse. □

Class (k, c, M)-Δ consists of all the (k, c)-sparse databases with the following condition.

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}

Figure 2.5. A market basket database D_3.

Condition 2.1 For each item $x \in I$,

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

□

When there is no such item that most of customers purchase in a visit, there exists a small M such that for each item x, the support rate of $\{x\}$ is at most M. In such a case, we can assume Condition 2.1 with M.

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

Condition 2.2 There exists some r_m ($0 < r_m < 1$) such that the given minimum support rate r is at least r_m. □

Consider the case that a store manager has a policy that he never sells the items whose support rates are less than r so far. As a result, the store only sells items whose support rates are at least r. Then, in an MBD of the store, the support rate of each item may be relatively large. In such a case, to obtain large itemsets, we need the minimum support rate r which is large to some extent, where we can assume Condition 2.2.

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 2.7 Suppose that a database D is in (k, c, M)-Δ with $cM < 1$ and Condition 2.2 is satisfied. Let X be a large itemset in D. Then, the following inequality holds:

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

Proof: Let D be a database in (k, c, M)-Δ and X be a large itemset in D. 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

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

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

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

By repeating the same argument, we can obtain

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

From Condition 2.2, $r_m \leq r \leq supr_D(X)$. Thus,

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

□

Theorem 2.3 Suppose that a database D is in (k, c, M)-Δ with $cM < 1$ and Condition 2.2 is satisfied. Then, all the large itemsets in D can be computed in polynomial time in $||D||$.

Proof: From Lemma 2.7, it is sufficient to consider only itemsets of size at most $k + \left\lfloor \frac{\log r_m}{\log(cM)} \right\rfloor$. Let $l = k + \left\lfloor \frac{\log r_m}{\log(cM)} \right\rfloor$. There are at most $|I|^l$ itemsets of size less than or equal to l. It can be checked in $\mathcal{O}(||D||)$ time whether a given itemset X is large in D. Therefore, all the large itemsets in D can be computed in $\mathcal{O}(||D|| \cdot |I|^l)$ time. Since l is a constant, all the large itemsets in D can be computed in polynomial time in $||D||$. □

Theorem 2.4 Suppose that a database D is in (k, c, M)-Δ with $cM < 1$ and Condition 2.2 is satisfied. Then, for a given minimum support rate r, a given minimum confidence f, and a minimum rhs size w, all the meaningful association rules $X \Rightarrow Y$ can be computed in polynomial time in $||D||$.

Proof: From Theorem 2.1, all the meaningful association rules can be computed by Procedure FIND-MAR shown in Figure 2.1. From Theorem 2.3, for a database D in (k, c, M)-Δ with $cM < 1$, step 1 can be executed in $\mathcal{O}(||D|| \cdot |I|^l)$ time, where $l = k + \left\lfloor \frac{\log r_m}{\log(cM)} \right\rfloor$. The *for loop* in step 3 is iterated at most $|I|^l$ times, and the *for loop* in step 4 is iterated at most $2^l - 1$ times. Since l is a constant, $2^l - 1$ is also a constant. Thus, step 3 through step 7 can be executed in $\mathcal{O}(|I|^l)$ time. Since the *for loop* in step 8 is iterated at most $|I|^l$ times, step 8 through step 10 can be executed in $\mathcal{O}(|I|^l)$ time. Thus, this algorithm runs in polynomial time in $||D||$. □

2.4 Summary of This Chapter

As one way to find meaningful association rules, a method using large itemsets has been considered. A number of algorithms for computing all the large itemsets have been proposed. However, the large itemset problem is shown to be NP-complete in Section 2.2. In fact, to compute all the large itemsets, these proposed algorithms need exponential time in the size of a given database.

Section 2.3 introduced the notion of (k, c)-sparsity of databases and proposed a subclass of MBDs, called (k, c, M)-Δ, which is defined using (k, c)-sparsity. For a database D in (k, c, M)-Δ, we can find all the meaningful association rules in D in $\mathcal{O}(||D|| \cdot |I|^l)$ time where l is a constant.

Whether a database satisfies Conditions 2.1 and 2.2 considered in Section 2.3 can be tested in $O(||D||)$ time. A polynomial-time algorithm which determines whether a database is (k, c)-sparse for a given positive integer k and a given positive real number c is the future work.

Chapter 3

Computational Complexity of Finding All the Highly Co-occurrent Itemsets

3.1 Highly Co-occurrent Itemsets

Several disadvantages of the support have been pointed out in References [5, 6, 16]. For example, the support of an itemset tends to be high if the itemset contains items with high supports (see Example 1.2). This chapter defines 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 which 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, which is used throughout this chapter.

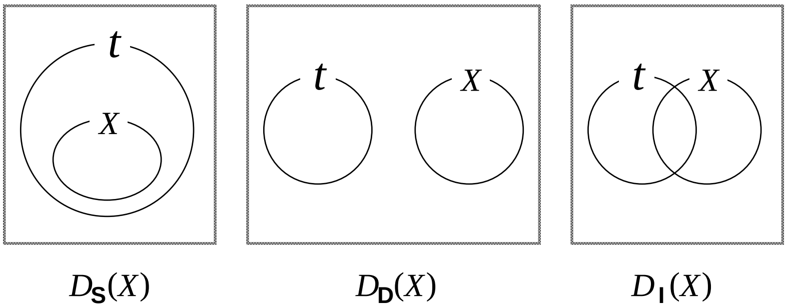

Figure 3.1. SDI division of D with X.

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 (see also Figure 3.1):

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

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, $V_{\mathbf{S}D}(X) = \mathit{supr}_D(X)$.

In SDI division, transactions in $D_{\mathbf{S}}(X)$ or $D_{\mathbf{D}}(X)$ are considered to establish high correlation among the items in X, while transactions in $D_{\mathbf{I}}(X)$ are not.

Let X be an itemset. For a given transaction, the probability that the itemset X occurs in the transaction under the assumption that each item occurs in D independently is $\prod_{x \in X} V_{\mathbf{S}D}(\{x\})$. 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 $V_{\mathbf{S}D}(X)$.

Example 3.1 Consider D_2 shown in Figure 1.2. Let $X = \{\text{cornflakes, milk}\}$. The SDI division with X divides D_2 into $D_{\mathbf{S}}(X) = \{t_2, t_4, t_5\}$, $D_{\mathbf{D}}(X) = \{t_3, t_6\}$, and $D_{\mathbf{I}}(X) = \{t_1\}$. Thus, $V_{\mathbf{S}}(X) = 1/2$, $V_{\mathbf{D}}(X) = 1/3$, and $V_{\mathbf{I}}(X) = 1/6$. Also, $E_{\mathbf{S}}(X) = 3/6 \times 4/6 = 1/3$, $E_{\mathbf{D}}(X) = 3/6 \times 2/6 = 1/6$, and $E_{\mathbf{I}}(X) = 1 - 1/3 - 1/6 = 1/2$. □

3.2 Definitions of Co-occurrence

3.2.1 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 all the definitions which consider the expected value.

Definition 3.1 (type **I**):

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

□

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.

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}

Figure 3.2. A market basket database D_4.

Example 3.2 Consider D_2 shown in Figure 1.2. 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 $E_{\mathbf{S}}(X) = {}^{3}\!/_{6} \times {}^{4}\!/_{6} = {}^{1}\!/_{3}$,

$$oc_{D_2}(X) = \frac{{}^{1}\!/_{2}}{{}^{1}\!/_{3}} = 1.5 \geq c.$$

Therefore, X is highly co-occurrent in D_2. On the other hand, since $V_{\mathbf{S}}(Z) = {}^{1}\!/_{2}$ and $E_{\mathbf{S}}(Z) = {}^{5}\!/_{6} \times {}^{4}\!/_{6} = {}^{5}\!/_{9}$,

$$oc_{D_2}(Z) = \frac{{}^{1}\!/_{2}}{{}^{5}\!/_{9}} = 0.9 < c.$$

Therefore, Z is not highly co-occurrent in D_2. □

Note that Z is large when the minimum support rate is 0.3 as seen in Example 1.2. 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.

3.2.2 Type II

Consider an itemset $X = \{\text{cornflakes, milk}\}$. Then a transaction which contains 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 3.2 (type **II**):

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

□

Example 3.3 Consider D_4 shown in Figure 3.2. 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_4 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 $V_{\mathbf{D}}(X) = 1/2$,

$$oc_{D_4}(X) = \frac{1}{6} + \frac{1}{2} \approx 0.67 \geq c.$$

Therefore, X is highly co-occurrent in D_4. On the other hand, the SDI division with Y divides D_4 into $D_{\mathbf{S}}(Y) = \{t_1\}$, $D_{\mathbf{D}}(Y) = \emptyset$, and $D_{\mathbf{I}}(Y) = \{t_2, t_3, t_4, t_5, t_6\}$. Since $V_{\mathbf{S}}(Y) = 1/6$ and $V_{\mathbf{D}}(Y) = 0$,

$$oc_{D_4}(Y) = \frac{1}{6} \approx 0.17 < c.$$

Therefore, Y is not highly co-occurrent in D_4. □

In Example 3.3, 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.

3.2.3 Type III

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

Definition 3.3 (type **III**):

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

□

Example 3.4 Consider D_2 shown in Figure 1.2. Suppose that the minimum co-occurrence c is 1.5. Let $X = \{\text{cornflakes, milk}\}$ and $W = \{\text{cereal, tea}\}$. Then, using values calculated in Example 3.1,

$$oc_{D_2}(X) = \frac{1/2 + 1/3}{1/3 + 1/6} \approx 1.67 > c.$$

Therefore, X is highly co-occurrent in D_2. On the other hand, the SDI division with W divides D_2 into $D_{\mathbf{S}}(W) = \{t_1\}$ and $D_{\mathbf{D}}(W) = \{t_2, t_3, t_4, t_5, t_6\}$. Since $V_{\mathbf{S}}(W) = {}^1\!/_6$, $V_{\mathbf{D}}(W) = {}^5\!/_6$, and $E_{\mathbf{S}}(W) = {}^1\!/_6 \times {}^1\!/_6 = {}^1\!/_{36}$, $E_{\mathbf{D}}(W) = {}^5\!/_6 \times {}^5\!/_6 = {}^{25}\!/_{36}$,

$$oc_{D_2}(W) = \frac{{}^1\!/_6 + {}^5\!/_6}{{}^1\!/_{36} + {}^{25}\!/_{36}} \approx 1.46 < c.$$

Therefore, W is not highly co-occurrent in D_2. □

3.2.4 Type IV

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

Definition 3.4 (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 **I** may also work in this example, type **IV** considers $V_{\mathbf{D}}(X)$ while type **I** 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.

Example 3.5 Consider D_2 shown in Figure 1.2. Suppose that the minimum co-occurrence c is 1.5. Let $X = \{\text{cornflakes, milk}\}$. Then, using values calculated in Example 3.1,

$$oc_{D_2}(X) = \frac{{}^1\!/_2}{{}^1\!/_3} \times \frac{{}^1\!/_3}{{}^1\!/_6} = 3 \geq c.$$

Therefore, X is highly co-occurrent in D_2. Next, consider $W = \{\text{cereal, tea}\}$, which is not highly co-occurrent in D_2 in type **III**. Then

$$oc_{D_2}(W) = \frac{{}^1\!/_6}{{}^1\!/_{36}} \times \frac{{}^5\!/_6}{{}^{25}\!/_{36}} = 7.2 \geq c,$$

and hence, W is highly co-occurrent in D_2 in this definition. □

3.2.5 Type V

This is an extension of type **III**. Type **V** is the same as *collective strength* [5], which has been proposed as an alternative to the support. This can be expressed in our notation as follows.

Definition 3.5 (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 Reference [5].

Example 3.6 Consider D_2 shown in Figure 1.2. Suppose that the minimum co-occurrence c is 1.5. Let $X = \{\text{cornflakes, milk}\}$ and $Z = \{\text{coffee, eggs}\}$. Then, using values calculated in Example 3.1,

$$oc_{D_2}(X) = \frac{1/2 + 1/3}{1/3 + 1/6} \times \frac{1/2}{1/6} = 5 \geq c.$$

Therefore, X is highly co-occurrent in D_2. On the other hand, the SDI division with Z divides D_2 into $D_{\mathbf{S}}(Z) = \{t_2, t_3, t_6\}$, $D_{\mathbf{D}}(Z) = \emptyset$, and $D_{\mathbf{I}}(Z) = \{t_1, t_4, t_5\}$. Since $V_{\mathbf{S}}(Z) = 1/2$, $V_{\mathbf{D}}(Z) = 0$, $V_{\mathbf{I}}(Z) = 1/2$, and $E_{\mathbf{S}}(X) = 5/6 \times 4/6 = 5/9$, $E_{\mathbf{D}}(X) = 1/6 \times 2/6 = 1/18$, $E_{\mathbf{I}}(X) = 1 - 5/9 - 1/18 = 7/18$,

$$oc_{D_2}(Z) = \frac{1/2 + 0}{5/9 + 1/18} \times \frac{7/18}{1/2} \approx 0.63 < c.$$

Therefore, Z is not highly co-occurrent in D_2. □

3.2.6 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 3.6 (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 $D_{\mathbf{I}}(X) = \emptyset$.

Example 3.7 Consider D_2 shown in Figure 1.2. Suppose that the minimum co-occurrence c is 1.5. Let $X = \{\text{cornflakes, milk}\}$. Then, using values calculated in Example 3.1,

$$oc_{D_2}(X) = \frac{1/2}{1/3} \times \frac{1/3}{1/6} \times \frac{1/2}{1/6} = 9 \geq c.$$

Therefore, X is highly co-occurrent in D_2. □

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

This section shows 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 3.7 (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 under whichever measure. Guess an itemset of size l, and then check whether the itemset is highly co-occurrent in D. So, in the rest of this chapter, 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 $oc_D(X)$.

3.3.1 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 $h \geq 2$ because the large itemset problem is NP-complete even if $h \geq 2$. 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 3.1 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 3.2 Suppose that (D, s, h) $(h \geq 2)$ 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 $sup_D(X) \geq s$ and $|X| \geq h$.

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

Proof: From Lemma 3.1, 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 $sup_D(X) \geq s$ and $|X| \geq h$. 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 $sup_D(X') \geq s$ and $|X'| = h$. Since $|X'| \geq 2$ and every transaction in $D_{\mathbf{A}}$ consists of just one item, no transaction in $D_{\mathbf{A}}$ supports X'. Thus,

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

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

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

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

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

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

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

□

3.3.2 Type II

We show the NP-hardness of the problem **II** by reducing the large itemset problem, which is defined in Definition 2.1, 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, as follows.

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 $(1 \leq j \leq n)$. Then, we define D' as follows:
$$D' \triangleq \{t'_1, \ldots, t'_n\}.$$

- Let $m = r$.

- Let $u = h + k$.

□

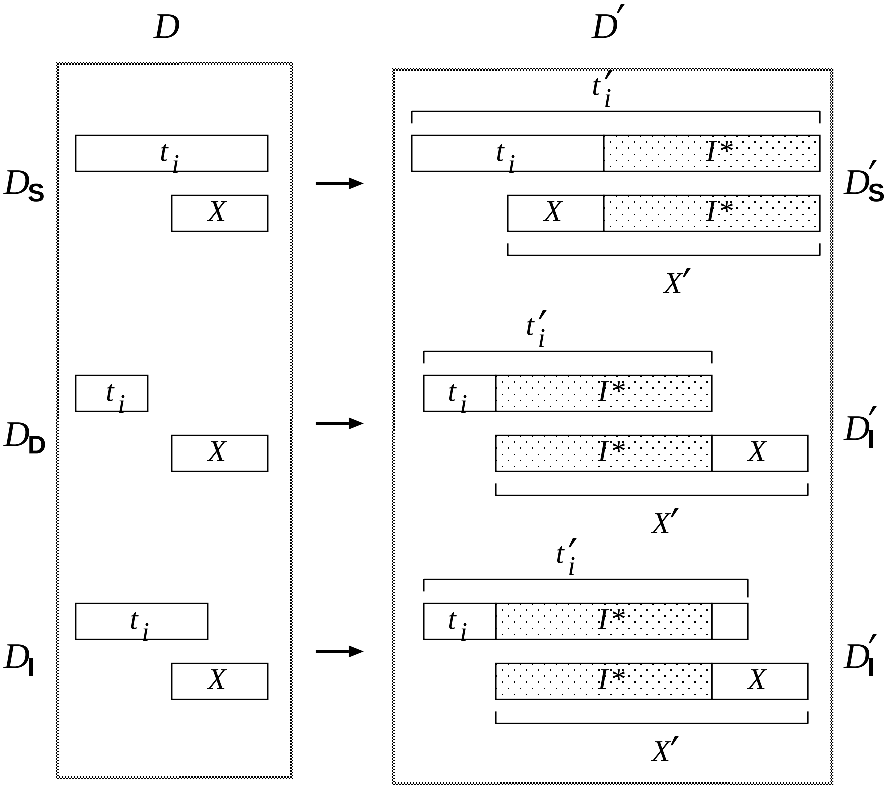

Figure 3.3. Construction method 1 of D'.

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 3.3 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 $supr_D(X) \geq r$ and $|X| \geq h$.

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

Proof: $(1 \rightarrow 2)$: Assume that there is an itemset $X \subseteq I$ in D such that $supr_D(X) \geq r$ and $|X| \geq h$. 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 $supr_D(X_h) \geq r$ 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 the transaction t'_i supports X' in D'. That is, $t'_i \in D'_{\mathbf{S}}(X')$; and

- otherwise, I^* is contained in both of t'_i and X', and there is at least one item in X' which t'_i does not contain. That is, $t'_i \in D'_{\mathbf{I}}(X')$ (see Figure 3.3).

Therefore, $X_h \subseteq t_i$ if and only if $X' \subseteq t'_i$ 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,

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

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

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

From $t'_i = t_i \cup I^*$, $X' \subseteq t'_i$ if and only if $X \subseteq t_i$ 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,

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

□

3.3.3 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**, as follows.

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 $(1 \leq i \leq n)$. 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 3.4 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 3.5 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 3.6 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 $oc_D(X) \geq m$ and $|X| = u$.

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

Proof: From Lemma 3.4, 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 $oc_D(X) \geq m$ and $|X| = u$. From Lemma 3.5,

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

Thus,

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

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

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

Thus, from Lemma 3.5,

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

□

3.3.4 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 ($1 \leq i \leq n$). 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 3.7 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 3.4. □

Lemma 3.8 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 3.5. □

Lemma 3.9 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 $oc_D(X) \geq m$ and $|X| = u$.
2. There is an itemset X in D' such that $oc_{D'}(X) \geq c$ and $|X| = l$.

Proof: From Lemma 3.7, 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 $oc_D(X) \geq m$ and $|X| = u$. Then, from Lemma 3.8,

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

($2 \rightarrow 1$): Assume that there is an itemset X in D' such that $oc_{D'}(X) \geq c$ and $|X| = l$ $(= u)$. Then, from Lemma 3.8,

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

□

3.3.5 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 $(1 \leq i \leq n)$. 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 3.10 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 3.4. □

Lemma 3.11 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.

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

$$V_{\mathbf{I}D}(X) = V_{\mathbf{I}D'}(X)$$

Proof: The proof of this lemma is similar to Lemma 3.5. □

Lemma 3.12 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 $oc_D(X) \geq m$ and $|X| = u$.

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

Proof: From Lemma 3.10, 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 $oc_D(X) \geq m$ and $|X| = u$. From Lemma 3.11,

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

Then,

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

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

$$oc_{D'}(X) = \frac{V_{\mathbf{S}D'}(X) + V_{\mathbf{D}D'}(X)}{E_{\mathbf{S}D'}(X) + E_{\mathbf{D}D'}(X)} \cdot \frac{E_{\mathbf{I}D'}(X)}{V_{\mathbf{I}D'}(X)} \quad \geq \quad c$$

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

Thus, from Lemma 3.11,

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

□

3.3.6 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 $(1 \leq i \leq n)$. Let $\bar{D} = \{t'_1, \ldots, t'_n\}$. Then, we define D' as follows:
$$D' \triangleq D \mathbin{\tilde{\cup}} \bar{D}.$$
- 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 3.13 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 3.4. □

Lemma 3.14 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 3.5. □

Lemma 3.15 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 $oc_D(X) \geq m$ and $|X| = u$.
2. There is an itemset X in D' such that $oc_{D'}(X) \geq c$ and $|X| = l$.

Proof: From Lemma 3.13, 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 $oc_D(X) \geq m$ and $|X| = u$. Then, from Lemma 3.14,

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

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

(2 → 1): Assume that there is an itemset X in D' such that $oc_{D'}(X) \geq c$ and $|X| = l$ $(= u)$. Then, from Lemma 3.14,

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

□

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

This section proposes subclasses of databases for which we can efficiently find all the highly co-occurrent itemsets. In this section, 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, which is defined in Chapter 2.

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

Condition 3.1 There exists some b_m ($0 < b_m < 1$) such that the given minimum co-occurrence b is at least b_m. □

3.4.1 Type I: Class (k, c, ϵ)-Γ

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

Condition 3.2 ((k, c)-sparsity): For any itemset X such that $|X| > k$, there is some $x \in X$ which satisfies the following inequality:

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

□

Condition 3.3 For each item x,

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

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 3.16 Suppose that a database D is in (k, c, ϵ)-Γ with $c < 1$ and Condition 3.1 is satisfied. Let X be a highly co-occurrent itemset in D. Then, the following inequality holds:

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

Proof: Let D be a database in (k,c,ϵ)-Γ and X be a highly co-occurrent itemset in D. Let $X = \{x_1,\ldots,x_t\}$. Then,

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

Suppose that $t > k$. Then, since D satisfies Condition 3.2, there is some $x \in X$ such that

$$V_{\mathsf{S}}(X) \leq c \cdot V_{\mathsf{S}}(X - \{x\}) \cdot V_{\mathsf{S}}(\{x\}).$$

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

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

By repeating the same argument, we can obtain

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

From Condition 3.1, $b_m \leq b \leq oc_D(X)$. Thus,

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

□

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 3.1 Suppose that a database D is in (k, c, ϵ)-Γ with $c < 1$ and Condition 3.1 is satisfied. Then, all the highly co-occurrent itemsets in D can be computed in polynomial time in $||D||$. □

3.4.2 Type II: Class (k, c, M)-Δ'

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

Condition 3.4 For any itemset X such that $|X| > k$, there is some $x \in X$ which satisfies the following inequality:

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

□

Condition 3.5 For each item $x \in I$,

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

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 3.17 Suppose that a database D is in (k, c, M)-Δ' with $cM < 1$ and Condition 3.1 is satisfied. Let X be a highly co-occurrent itemset in D. Then, the following inequality holds:

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

Proof: Let D be a database in (k, c, M)-Δ' and X be a highly co-occurrent itemset in D. Let $X = \{x_1, \ldots, x_t\}$ where $t > k$. Then, since D satisfies Condition 3.4, there is some $x \in X$ such that

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

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

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

By repeating the same argument, we can obtain

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

From Condition 3.1, $b_m \leq b \leq oc_D(X)$. Thus,

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

□

Let $l = k + \left\lfloor \frac{\log b_m}{\log(cM)} \right\rfloor$. 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 3.2 Suppose that a database D is in (k, c, M)-Δ' with $cM < 1$ and Condition 3.1 is satisfied. Then, all the highly co-occurrent itemsets in D can be computed in polynomial time in $||D||$. □

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

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

Condition 3.6 ((k, c)-sparsity)**:** For any itemset X such that $|X| > k$, there is some $x \in X$ which satisfies the following inequality:

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

□

Condition 3.7 For any itemset X such that $|X| > k$, there is some $x \in X$ which satisfies the following inequality:

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

where c' is a positive real number. □

Condition 3.8 For each item x,

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

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 3.18 Suppose that a database D is in (k, c, c', ϵ, M)-Γ' with $cc' < 1$ and Condition 3.1 is satisfied. Let X be a highly co-occurrent itemset in D. Then, the following inequality holds:

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

Proof: Let D be a database in (k, c, c', ϵ, M)-Γ' and X be a highly co-occurrent itemset in D. 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 satisfies Condition 3.6, we can obtain the following inequality by the same argument as in Lemma 3.16.

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

Also, since D satisfies Condition 3.7, and $V_{\mathbf{D}}(\{x\}) = 1 - V_{\mathbf{S}}(\{x\}) \geq 1 - M$, we can obtain the following inequality from the above.

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

From Condition 3.1, $b_m \leq b \leq oc_D(X)$. Thus,

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

□

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 3.3 Suppose that a database D is in (k, c, c', ϵ, M)-Γ' with $cc' < 1$ and Condition 3.1 is satisfied. Then, all the highly co-occurrent itemsets in D can be computed in polynomial time in $||D||$. □

3.5 Summary of This Chapter

In Section 3.2, we have proposed several formal definitions of $oc_D(X)$. Of course, no definition of $oc_D(X)$ achieves the best quality as we have seen in various examples in Section 3.2. In other words, database users have a chance to determine which definition of $oc_D(X)$ they should use, including $supr_D(X)$.

In Section 3.3, we have shown that finding all the highly co-occurrent itemsets is NP-hard under whichever measure we have defined in Section 3.2. 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.

In Section 3.4, we have proposed subclasses of databases for which we can efficiently find all the highly co-occurrent itemsets of type **I**, **II**, and **IV**. These subclasses are defined based on the notion of (k, c)-sparsity introduced in Chapter 2.

As a future work, we intend to propose subclasses of databases for which all the highly co-occurrent itemsets of type **III**, **V**, and **VI** can be computed efficiently.

Chapter 4

Conclusions

In Chapter 2, we have formally defined the large itemset problem based on the support-confidence framework and shown the NP-completeness of the problem. 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.

Also, we have introduced the notion of (k, c)-sparsity of databases. Intuitively, (k, c)-sparsity of a database means that the supports of itemsets of size k or more are considerably low in the database. The value of c represents a degree of sparsity. Any database is (k, c)-sparse for some sufficiently high k or c. Thus, the (k, c)-sparsity is a general condition on 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. Because of the (k, c)-sparsity, the size of a large itemset is bounded by a constant, and so we need not consider any itemset of size greater than the constant. In fact, the test data in References [1, 3, 4, 6, 7, 9, 15, 17, 20] are all (k, c)-sparse for some small k and c unless these algorithms need exponential time in the size of databases.

In Chapter 3, we have defined alternative measures to the support, called co-occurrence. Some of these measures are similar to the previous works such as collective strength in Reference [5] and dependence in Reference [16] in that they all consider the expected value of the support. 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.

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

References

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

[2] R. Agrawal, T. Imielinski, and A. Swami, “Database mining: a performance perspective,” *IEEE Trans. on Knowledge and Data Engineering*, vol.5, pp.914–925, 1993.

[3] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo, “Fast discovery of association rules,” In Fayyad et al, pp.307–328, 1996.

[4] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” In *Proc. of the 20th Int'l Conf. on Very Large Data Bases*, pp.487–499, Sept. 1994.

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

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

[7] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds, “Advances in Knowledge Discovery and Data Mining,” MIT press, 1996.

[8] M.R. Garey and D.S. Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness,” FREEMAN press, 1978.

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

[10] J. Han and Y. Fu, “Discovery of multiple-level association rules from large databases,” In *Proc. of the 21th Int'l Conf. on Very Large Data Bases*, pp.420–431, Sept. 1995.

[11] M. Houtsma and A. Swami, "Set-oriented mining of association rules," In *Proc. of the Int'l Conf. on Data Engineering*, pp.25–34, March 1995.

[12] J.E. Hopcroft and J.D. Ullman, "Introduction to Automata Theory, Language, and Computation," Addison-Wesley, 1979.

[13] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo, "Finding interesting rules from large sets of discovered association rules," In *Proc. of the 3rd Int'l Conf. on Information and Knowledge Management*, pp.401–407, Nov. 1994.

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

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

[16] 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.

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

[18] R. Srikant and R. Agrawal, "Mining generalized association rules," In *Proc. of the 21st Int'l Conf. on Very Large Data Bases*, pp.407–419, Sept. 1995.

[19] R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, pp.1–12, June 1996.

[20] H. Toivonen, "Sampling large databases for association rules," In *Proc. of the 22nd Int'l Conf. on Very Large Data Bases*, pp.134–145, Sept. 1996.