Technical Report CSD-TR No. 030042

Mining Frequent Itemsets in Uncertain Datasets

Richard Muntz and Yi Xia

Computer Science Department
University of California
Los Angeles, CA 90095

{muntz, xiayi}@cs.ucla.edu

August 29, 2003

Abstract

Data in real world are usually noisy or uncertain. However, traditional data mining algorithms ignore the uncertainty in data or take it into consideration in a very limited way. In this paper, we define a relatively generic model for uncertainty in data in which each data item comes with a "tag" that defines the degree of confidence in that value. This is more realistic in many cases where the data items are derived from other evidence or more basic data. Simple examples are face recognition and fingerprint identification where, for example, the raw data itself can influence the degree of confidence in the identification. As an example problem, in this paper we study frequent itemset mining in such uncertain data.

With uncertain data, finding frequent itemsets will not be perfect. There will be false positives (itemsets which are estimated to be frequent but which are not) and false negatives (frequent itemsets which are estimated not to be frequent). We consider several intuitive approaches and propose a new scheme which significantly reduces the number of false positives and false negatives.

1 Introduction & motivation

Data mining is the process of discovering interesting patterns/knowledge from massive data. However, data in the real world are usually imperfect, in the sense that there can be missing, wrong or ambiguous data values. The uncertainty in data, if not taken into account, may lead data mining algorithms to report erroneous patterns or, conversely, fail to report patterns which exist. Our goal is to begin to address the issue of how we might be able to discover true patterns from an uncertain dataset?

There are two issues that have to be addressed: (a) How to model the uncertainty in data, and (b) how to perform data mining algorithms on an uncertain dataset.

Different models have been proposed in the literature to represent uncertainty in data, such as fuzzy set, probability theory([ZCF+97]) and Dempster-Shafer evidence theory([Sha76]). Among them, probability theory is perhaps the most popular and widely accepted model. For the same model, there are different assumptions and representations for the uncertainty in data. For instance, many statistical techniques assume that errors in a dataset follow some distribution. A common assumption is that the error for each instance of an attribute value is an independent and identically distributed random variable; often Gaussian. However, little work has dealt with datasets where each individual data value has a measure of its certainty, though this representation contains more detailed uncertainty information.

Consider association rule mining ([AIS93], [AS94]) as an example. Association rule mining is used to detect correlations among items of a dataset. Traditionally, we use *confidence* and *support* to estimate the correlation among items and the significance of such correlation with regard to the whole population. These quantities are well defined when the underlying dataset is deterministic and the values in it reflect the truth. However, such a dataset may not always be available. For example, consider a dataset derived from a set of pictures taken under certain circumstances. The list of people present in each picture forms a tuple in the dataset. Assume people in all these pictures consist of a social circle. We want to detect the co-occurrence among these people, such as, given that Person X is present, Person Y is likely to be present also. Such a dataset is similar to market basket data where each item's absence or presence in a transaction is indicated by 0 or 1. Unfortunately, with the limitations in image processing techniques, we can not be 100% sure if a person is present in a picture or not. That is, due to the orientation of a face, shadows, partial occlusion, etc, each individual data value may have an independent confidence(probability) associated with it. So the dataset we get is uncertain.

The change in the dataset representation affects the data mining tasks. As we'll see in Section 4, people usually do not deal with such uncertain datasets directly. In fact, they are likely to make an uncertain dataset deterministic before performing any tasks. A problem with such approaches is that the detailed uncertainty information is lost in the transformation to a deterministic dataset and many important patterns can no longer be discovered. This problem can be seen more clearly from our experiments in Section 6.

In this paper, we focus on the problem of frequent itemset mining, which is the first and most important step in association rule mining. We provide a relatively generic model for uncertainty in data and an algorithm, called EST, for discovering frequent itemsets in an uncertain dataset. The experiments show that EST can significantly increase the number of true patterns being discovered, compared to existing approaches, without introducing as many spurious patterns.

The remainder of this paper is organized as follows. Section 2 provides related work; Section 3 formally defines the problem of frequent itemset mining in uncertain datasets. Three existing approaches are provided in Section 4. Section 5 elaborates on our new algorithm EST. The experiments and comparison between EST and other existing approaches are described in Section 6. The paper concludes with a short summary in Section 7. A proof for the main theorem in the paper is provided in Appendix A.

2 Related work

Different types of uncertainty models have been proposed in the database field. [BGMP92], [LLRS97] define models based on probability theory; [MPV94] makes use of fuzzy set theory; while [Lee92] is an instance of Dempster-shafer theory. Among them, [BGMP92] has the most similar uncertainty model to ours. It

uses *stochastic attribute*s to capture non-deterministic properties of database entities. A *stochastic attribute* for each individual entity corresponds to a set of values, with a discrete probability distribution over them, indicating the values' degree of truth.

All the above papers focus on extending relational algebra to handle uncertainty over basic operations, such as *projection*, *selection*, and *join*. Little work has been done on how to deal with aggregate operations, such as *count*. It is even not clear what the meaning is to perform a counting operation over an uncertain dataset, though aggregate operations are the essential building blocks for data mining.

Uncertainty in data mining processes has been considered in a limited way. [YWYH02] mines frequent sequential patterns in a sequence database with noise. The noise is defined by a *compatibility matrix*. Each entry in the matrix corresponds to a pair of items (X, Y) that specifies the conditional probability of X being true given Y being observed. Because of such uncertainty in the data, the *support* measure for a sequential pattern in the traditional algorithms is replaced by a new metric called *match*, which is essentially the expected value for *support*. [KM02] assumes a noisy dataset is generated from a probabilistic model with three components: a generative model of the clean data points, a generative model of the noise values, and a probabilistic model of the corruption process. The goal of [KM02] is to identify and adjust for the noise.

Both [YWYH02] and [KM02] assume the noise in data is controlled by a single parametric statistical model, which is not always true. In fact, the sources of uncertainty could be very diverse and they can not be handled by a single model. The model we propose does not directly characterize the sources of uncertainty, but their effect on each individual data value by a degree of truth. Despite the noise assumed in these papers, their datasets are deterministic. In many situations, the original datasets obtained are uncertain, but they are transformed to deterministic datasets before being presented to others or fed to any mining tasks. (Several simple transformations of this type will be described shortly.) Important uncertainty information could be lost during this process and the mining result may be biased.

[KFW98], [HV02] incorporate uncertainty in data mining based on fuzzy set theory. Fuzzy set theory treats an item's membership to a specific class as a continuous function over $[0, 1]$. It is most appropriate for uncertainty that emerges when a strict demarcation between classes is inappropriate. This kind of uncertainty is beyond the scope of this paper.

A research field that is parallel to our problem is privacy preserving data mining ([AS00], [ESAG02], [RH02], [DZ03]). They deal with randomized datasets. The difference between this work and ours is that the noise in a randomized dataset is injected on purpose and is known to the data mining algorithms; and again, the actual datasets to be mined are deterministic.

3 Problem description

In this section, we define the problem of frequent itemset mining in an uncertain dataset.

3.1 Concept definitions

Definition 3.1 Deterministic dataset: *Let $\mathcal{D}^T$ be a binary dataset with M distinct items $\Omega = \{I_1, I_2, \ldots, I_M\}$. $I_j = 1$ ($j \in [1, M]$) in a tuple $t \in \mathcal{D}^T$ indicates that item I_j is present in t, while $I_j = 0$ means I_j is not present in t. $\mathcal{D}^T$ is called a* **deterministic dataset**. *If t contains $I_j = 1$, we say* **tuple t supports item**

I_j. Similarly, for a set of items $\mathcal{I} \subseteq \Omega$, we say **a tuple** *$t$* **supports** *$\mathcal{I}$ if it supports all of the individual items in $\mathcal{I}$.*

A frequent itemset $\mathcal{I}$ ($\mathcal{I} \subseteq \Omega$) in a *deterministic dataset* is defined as:

Definition 3.2 Frequent itemset: *Itemset $\mathcal{I}$ is* **frequent** *if the number of tuples in $\mathcal{D}^T$ supporting $\mathcal{I}$ is above a user defined threshold. The number of tuples supporting $\mathcal{I}$ is called the* **support** *of $\mathcal{I}$, and is denoted by $S_{\mathcal{I}}$. If $\mathcal{I}$ consists of K distinct items, we call it a* **frequent itemset at level** *K.*

In many situations, a *deterministic dataset* whose values perfectly reflect the truth is not available. Instead, we get a set of uncertain values with different degrees of certitude or confidence. We define a generic uncertainty model as follows:

Definition 3.3 Uncertain dataset: *Let $\mathcal{D}$ be a dataset with M distinct items $\Omega = \{I_1, I_2, \ldots, I_M\}$. For any tuple $t \in \mathcal{D}$, item I_j ($j \in [1, M]$) does not correspond to a single, deterministic value, but a distribution over the pair of values $< 0, 1 >$. (The value 1 and 0 have the same meanings as that in a* **deterministic dataset***.) The probability mass associated with each value we term the* **tag** *of the value. The* **tag** *of a value indicates how likely that value is to be true. $\mathcal{D}$ is called an* **uncertain dataset***.*

Unlike many existing models of uncertainty that assume a common noise distribution (often Gaussian) over the whole dataset or the values of the same attribute, we assume a *tag* is assigned to each value in the dataset individually. In this sense, our model of uncertainty is more generic. An *uncertain dataset* defined in this way preserves the detailed information of uncertainty, of each value and thus could lead to more accurate estimate of the true patterns. Intuitively, the values in which there is greater confidence should be weighted more.

Table 1 gives an example of an *uncertain dataset*. In this example, item X and Y in each row of $\mathcal{D}$ correspond to a pair of values with *tags*. The value pair $\{< 1, 0.3 >, < 0, 0.7 >\}$ for X in the first tuple says that X is present in this tuple with probability 0.3, and absent with probability 0.7. $\mathcal{D}$ can be simplified to $\mathcal{D}_S$ which contains tags for value 1s only without loss of any information. In the following, we will use the simplified form to represent an *uncertain dataset*.

$\mathcal{D}$		$\mathcal{D}_S$	
X	Y	X	Y
$< \mathbf{1}, \mathbf{0.3} >, < \mathbf{0}, \mathbf{0.7} >$	$< 1, 0.8 >, < 0, 0.2 >$	0.3	0.8
$< 1, 0.5 >, < 0, 0.5 >$	$< 1, 0.4 >, < 0, 0.6 >$	0.5	0.4
$< 1, 0.7 >, < 0, 0.3 >$	$< 1, 0.7 >, < 0, 0.3 >$	0.7	0.7
$< 1, 0.9 >, < 0, 0.1 >$	$< 1, 0.9 >, < 0, 0.1 >$	0.9	0.9

Table 1: An example of *uncertain dataset*

3.2 Mining objectives

Suppose we are given an *uncertain dataset* $\mathcal{D}$ corresponding to a true *deterministic dataset* $\mathcal{D}^T$. We want to discover from $\mathcal{D}$ the true *frequent itemsets*, that is, the itemsets that are *frequent* in $\mathcal{D}^T$.

Based on the information in $\mathcal{D}$, we may not be able to discover the exact set of *frequent itemsets* in $\mathcal{D}^T$, due to the inherent uncertainty in $\mathcal{D}$. An *infrequent itemset* in $\mathcal{D}^T$ that is discovered to be *frequent* by some mining algorithm applied to $\mathcal{D}$ is called a *false positive itemset*. Symmetrically, a *frequent itemset* in $\mathcal{D}^T$ that is found to be *infrequent* in $\mathcal{D}$ is called a *false negative itemset*. A good mining algorithm should produce both a small number of *false negative itemsets* and a small number of *false positive itemsets*.

To simplify the problem, we make the following two assumptions concerning *uncertain dataset*s:

- The tags associated with values of different items in the same tuple are assigned independently given the true values of these items.
- Values for different items in an *uncertain dataset* are independent given the tags.

4 Existing algorithms

To mine frequent itemsets in an uncertain dataset $\mathcal{D}$, a straightforward approach is to convert $\mathcal{D}$ to a deterministic dataset $\mathcal{D}_{det}$ first, then apply a traditional mining algorithm on $\mathcal{D}_{det}$. Generally speaking, there are two common ways to generate a deterministic dataset. For a tag tag_X associated with value $X = 1$ in $\mathcal{D}$, one way is to output

$$X = \begin{cases} 1, & \text{if } tag_X \geq threshold; \\ 0, & \text{otherwise.} \end{cases}$$

Another way is to output

$$X = \begin{cases} 1, & \text{with probability } tag_X; \\ 0, & \text{with probability } 1 - tag_X. \end{cases}$$

We call the mining algorithm based on the first transformation DET1, and that based on the second transformation DET2. $\mathcal{D}_{det1}$ and $\mathcal{D}_{det2}$ in Table 2 give an example of the deterministic datasets generated from the uncertain dataset $\mathcal{D}$ by DET1 and DET2. Here the threshold for DET1 is set to 0.5.

$\mathcal{D}$	
X	Y
0.6	0.9
0.6	0.9
0.6	0.9
0.6	0.9
0.6	0.9
0.6	0.9
0.6	0.9
0.6	0.9
0.6	0.9
0.6	0.9

$\mathcal{D}_{det1}$	
X	Y
1	1
1	1
1	1
1	1
1	1
1	1
1	1
1	1
1	1
1	1

$\mathcal{D}_{det2}$	
X	Y
1	1
1	1
0	1
0	1
1	1
0	1
1	1
1	0
1	1
0	1

Table 2: The deterministic datasets derived from an uncertain dataset

Both DET1 and DET2 discard the detailed uncertainty information in original data and this will affect the quality of the mined result.

A third algorithm EXP uses a similar idea as that in [YWYH02]. It extends the traditional mining algorithm to an uncertain dataset by replacing the *support* measure with the expected *support* value. For example, itemset $\{XY\}$ in $\mathcal{D}$ (Table 2) has expected support value $E[S_{XY}] = 0.6 * 0.9 * 10 = 5.4$. Itemset $\{XY\}$ is output as a frequent itemset by algorithm EXP if $E[S_{XY}]$ is above the support threshold.

All three approaches discussed thus far can be easily implemented based on existing frequent itemset mining algorithms. However, a common problem with them is that they often output either a large number of *false positive* or a large number of *false negative* itemsets, as we will see in the experiments in Section 6. Another problem they all have is that they have difficulty in recognizing the frequent itemsets at high level.

5 Algorithm EST: using ESTimated *support* to discover frequent itemsets in an uncertain dataset

Let $\mathcal{D}^T$ denote a deterministic dataset representing "ground truth", i.e., the true contents of a set of transactions. Let $\mathcal{D}$ denote an uncertaint dataset representing the same ground truth. Figure 1 (a) illustrates the relationship of $\mathcal{D}^T$ and $\mathcal{D}$; figure 1 (b) illustrates how we model the relationship of $\mathcal{D}^T$ and $\mathcal{D}$.

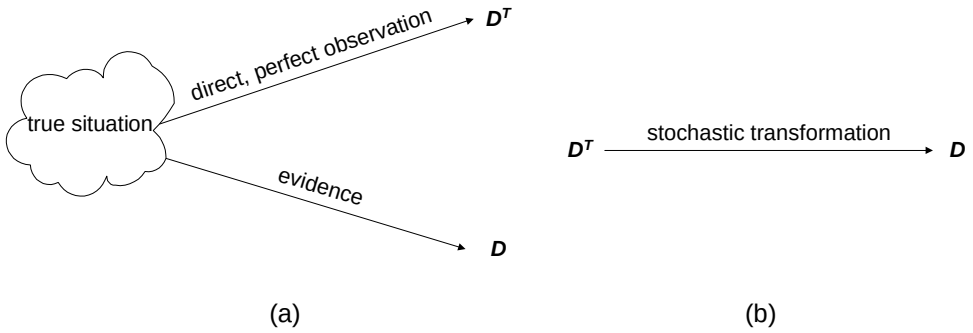

Figure 1: The relationship between "ground truth" and $\mathcal{D}$

Let $S_{\mathcal{I}}$ be the support of itemset $\mathcal{I}$ in $\mathcal{D}^T$. The basic idea of algorithm EST is to develop a good *support* estimator $e(S_{\mathcal{I}})$ for $\mathcal{I}$. Using $e(S_{\mathcal{I}})$, itemsets that are likely to be frequent in $\mathcal{D}^T$ can be discovered according to the following criterion $\mathcal{C}$:

$$\mathcal{C}: \quad \mathcal{I} \text{ is } \begin{cases} \textit{frequent}, & \text{if } e(S_{\mathcal{I}}) \geq \textit{threshold}; \\ \textit{infrequent}, & \text{otherwise.} \end{cases}$$

In the following, we first give a different view of the uncertain dataset $\mathcal{D}$, then talk about how to derive the estimator $e(S_{\mathcal{I}})$.

5.1 A different view of the uncertain dataset $\mathcal{D}$

An uncertain dataset $\mathcal{D}$ with a set of items Ω can be viewed as generated through the following thought experiment: Let $\mathcal{D}^T$ be the true deterministic dataset behind $\mathcal{D}$. We use a recognition tool to identify the values for each item X in $\mathcal{D}^T$ independently. Due to some sources of uncertainty in $\mathcal{D}^T$ and the limitations of the tool, the tool outputs for each value of X a pair of tagged values: $\{< 1, tag_X >, < 0, 1 - tag_X >\}$. As we mentioned in Section 3.1, only tags for value 1s are preserved in $\mathcal{D}$. So for each tag_X in $\mathcal{D}$, it may come from a true value 1 or 0. Given $\mathcal{D}^T$ and $\mathcal{D}$, we can compute the proportion of instances of $X = 1$ in $\mathcal{D}^T$ that have been assigned a tag value tag_X in $\mathcal{D}$, as well as the proportion of instances of $X = 0$ in $\mathcal{D}^T$ that have been assigned a tag value tag_X. We call these tag proportions the *tag distributions* for item X, and denote them by

$$f_X(tag_X) = P(X \text{ is assigned } tag_X \text{ in } \mathcal{D} | X = 1 \text{ in } \mathcal{D}^T),$$

and

$$g_X(tag_X) = P(X \text{ is assigned } tag_X \text{ in } \mathcal{D} | X = 0 \text{ in } \mathcal{D}^T).$$

The *tag distributions* for all items in $\mathcal{D}^T$ is denoted by

$$\mathbb{T} = \{f_X(tag_X), g_X(tag_X) | X \in \Omega\}.$$

$\mathbb{T}$ defines a mapping from $\mathcal{D}^T$ to a space Ψ of uncertain datasets. Each uncertain dataset in Ψ is generated by assigning tags to values in $\mathcal{D}^T$ following the distributions $\mathbb{T}$. $\mathcal{D}$ can be viewed then as a sample point in Ψ. Figure 2 illustrates this process.

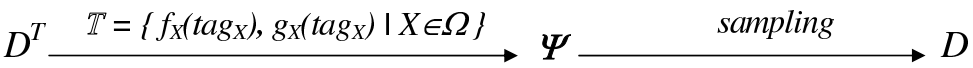

Figure 2: A different view of the uncertain dataset $\mathcal{D}$

5.2 An unbiased *support* estimator $e(S_\mathcal{I})$

Let $S_\mathcal{I}$ be the support of itemset $\mathcal{I} = \{I_1, I_2, \ldots, I_K\}$ in $\mathcal{D}^T$. The expected value of $S_\mathcal{I}$ conditioned on the given uncertain dataset $\mathcal{D}$ is denoted by $\widetilde{S_\mathcal{I}} = E[S_\mathcal{I}|D]$. From Figure 2, we can see that $\mathcal{D}$ is a sample point in space Ψ defined by the true dataset $\mathcal{D}^T$ and the tag distributions $\mathbb{T}$. Given $\mathcal{D}^T$ and $\mathbb{T}$, we can compute the expected value of $\widetilde{S_\mathcal{I}}$, denoted by $E[\widetilde{S_\mathcal{I}}]$.

Theorem 1. *Let $S_\mathcal{I}$ be the support of itemset $\mathcal{I} = \{I_1, I_2, \ldots, I_K\}$ in $\mathcal{D}^T$, $\widetilde{S_\mathcal{I}}$ be the expected value of $S_\mathcal{I}$ conditioned on $\mathcal{D}$. The expected value of $\widetilde{S_\mathcal{I}}$ conditioned on the true dataset $\mathcal{D}^T$ and tag distributions $\mathbb{T}$ has the following form:*

$$E[\widetilde{S_\mathcal{I}}] = A(\mathbb{T}) * S_\mathcal{I} + B(\mathbb{T}, \{S_f | f \subset \mathcal{I}\}). \quad (1)$$

Here, $A(\mathbb{T})$ is a function of $\mathbb{T}$; $B(\mathbb{T}, \{S_f | f \subset \mathcal{I}\})$ is a function of $\mathbb{T}$ and the true support for each of $\mathcal{I}$'s true subsets. Let $\{tag_i | i = 1, 2, \ldots, M_{I_j}\}$ be the M_{I_j} distinct tag values item I_j can take, and $F_{I_j} =$

$\sum_i[tag_i * f_{I_j}(tag_i)]$, $G_{I_j} = \sum_i[tag_i * g_{I_j}(tag_i)]$, *then,*

$$A(\mathbb{T}) = \prod_{I_j \in \mathcal{I}} (F_{I_j} - G_{I_j}); \quad (2)$$

$$B(\mathbb{T}, \{S_f | f \subset \mathcal{I}\}) = \sum_{f \subset \mathcal{I}} \{S_f * \prod_{I_j \in f} (F_{I_j} - G_{I_j}) * \prod_{I_j \in \mathcal{I} \setminus f} G_{I_j}\} \quad (3)$$

■

PROOF. See Appendix A. ■

Intuitively, $B(\mathbb{T}, \{S_f | f \subset \mathcal{I}\})$ represents the contribution of tuples in $\mathcal{D}^T$ that actually only partially support $\mathcal{I}$ but, due to the uncertainty, contribute a non-zero value to $E[\widetilde{S_\mathcal{I}}]$. $A(\mathcal{I})$ represents the degree to which a tuple fully supporting $\mathcal{I}$ in $\mathcal{D}^T$ is affected by the sources of uncertainty. In the extreme situation where there is no uncertainty, the tag distributions will be $f_{I_j}(1) = 1$, and $g_{I_j}(0) = 1$ for each $I_j \in \mathcal{I}$. $\mathcal{D}$ then becomes equal to $\mathcal{D}^T$. According to formula (1), we have

$$\begin{aligned} E[\widetilde{S_\mathcal{I}}] &= A(\mathbb{T}) * S_\mathcal{I} + B(\mathbb{T}, \{S_f | f \subset \mathcal{I}\}) \\ &= 1 * S_\mathcal{I} + 0 \\ &= S_\mathcal{I}. \end{aligned}$$

By a simple transformation of formula (1), we get

$$S_\mathcal{I} = E[\frac{\widetilde{S_\mathcal{I}} - B(\mathbb{T}, \{S_f | f \subset \mathcal{I}\})}{A(\mathbb{T})}]. \quad (4)$$

So, $\frac{\widetilde{S_\mathcal{I}} - B(\mathbb{T}, \{S_f | f \subset \mathcal{I}\})}{A(\mathbb{T})}$ is an unbiased estimator of the true support $S_\mathcal{I}$. We denote it by:

$$e(S_\mathcal{I}) = \frac{\widetilde{S_\mathcal{I}} - B(\mathbb{T}, \{S_f | f \subset \mathcal{I}\})}{A(\mathbb{T})}. \quad (5)$$

By replacing each S_f in (5) with its unbiased estimate $e(S_f)$, we get:

$$e(S_\mathcal{I}) = \frac{\widetilde{S_\mathcal{I}} - B(\mathbb{T}, \{e(S_f) | f \subset \mathcal{I}\})}{A(\mathbb{T})}. \quad (6)$$

Equation (6) is a recursive expression. Given the uncertain dataset $\mathcal{D}$, $\widetilde{S_\mathcal{I}}$ and $\{\widetilde{S_f} | f \subset \mathcal{I}\}$ are known. If we have the information about $\mathbb{T}$, we can recursively compute the unbiased support estimate for all $\mathcal{I}$'s subsets $\{e(S_f) | f \subset \mathcal{I}\}$, and finally, we obtain $e(S_\mathcal{I})$.

5.3 Deriving the tag distributions $\mathbb{T}$ from the uncertain dataset $\mathcal{D}$

In the previous discussion, the tag distributions $\mathbb{T}$ are given. In real situations, they are not provided. However, if we assume that the tag associated with a value in $\mathcal{D}$ really reflects the value's probability of being true, the following constraint holds for any item X in Ω:

Definition 5.1 Tag constraint: *The probability of $X = 1$ in the true dataset $\mathcal{D}^T$ given X being assigned tag value tag_X in the uncertain dataset $\mathcal{D}$ is tag_X. That is:*

$$\begin{aligned} tag_X &\cong P(X = 1 \text{ in } \mathcal{D}^T | X \text{ is assigned } tag_X \text{ in } \mathcal{D}) \\ &= \frac{f_X(tag_X)S_X}{f_X(tag_X)S_X + g_X(tag_X)(N - S_X)}. \end{aligned} \tag{7}$$

Here, N is the total number of tuples in $\mathcal{D}^T$ (or $\mathcal{D}$). For example, suppose $\mathcal{D}$ has 10 tuples with $tag_X = 0.9$. If the recognition tool that generated $\mathcal{D}$ has a perfect estimate about its errors, the expected number out of the 10 tuples to have $X = 1$ in $\mathcal{D}^T$ is 9.

Note*: the tag constraint (7) is expected to hold under the assumption that the tool is perfect in assigning tag values and in the limit as the number of tuples becomes large. However, it is only an approximation. For a tag value close to 0 or 1, the tag constraint is expected to be a tight approximation, while for a tag value close to 0.5, the approximation is expected to be loose.*

Suppose there are m distinct tag values for item X, to estimate the tag distributions for X, we have $2m+1$ unknown values. They are:

- S_X: the support of X in the true dataset $\mathcal{D}^T$;
- $\{f_X(tag_i)|i = 1, 2, \ldots, m\}$: the percentage of $X = 1$ in $\mathcal{D}^T$ that is assigned tag value tag_i in $\mathcal{D}$;
- $\{g_X(tag_i)|i = 1, 2, \ldots, m\}$: the percentage of $X = 0$ in $\mathcal{D}^T$ that is assigned tag value tag_i in $\mathcal{D}$.

We have a total $2m + 2$ constraints:

- $2m$ soft constraints derived from the *tag constraints*:

$$f_X(tag_i)S_X = tag_i N_{tag_i}, \quad i = 1, 2, \ldots, m; \tag{8}$$

$$f_X(tag_i)S_X + g_X(tag_i)(N - S_X) = N_{tag_i}, \quad i = 1, 2, \ldots, m. \tag{9}$$

  Here, N is the total number of tuples in $\mathcal{D}$; N_{tag_i} is the number of tuples in $\mathcal{D}$ that contain X with tag value tag_i.

- 2 strict constraints:

$$\sum_i f_X(tag_i) = 1; \tag{10}$$

$$\sum_i g_X(tag_i) = 1. \tag{11}$$

From equations (8), (9), (10) and (11), we get:

$$S_X = \sum_i tag_i N_{tag_i}; \tag{12}$$

$$f_X(tag_i) = \frac{tag_i N_{tag_i}}{S_X}; \tag{13}$$

$$g_X(tag_i) = \frac{(1 - tag_i)N_{tag_i}}{N - S_X}. \tag{14}$$

5.4 Level-wise mining of frequent itemsets in the uncertain dataset $\mathcal{D}$

With the uncertain dataset $\mathcal{D}$ known and the tag distributions $\mathbb{T}$ computed according to the development in Section 5.3, we are able to discover the frequent itemsets level by level, using formula (6). The estimated support value for itemsets at lower levels can be used to estimate the support for their super-itemsets. Algorithm 1 gives the complete procedure for EST.

Algorithm 1 EST

Input: the uncertain dataset $\mathcal{D}$ with a set of distinct items Ω; the *support* threshold t_{sup};
Output: $\mathbb{F} = \{\mathcal{I} | \mathcal{I} \subseteq \Omega, \text{ and } e(\mathcal{I}) \geq t_{sup}\}$.

derive the tag distributions $\mathbb{T}$ from $\mathcal{D}$;
$k \leftarrow 1$;
$\mathbb{F} \leftarrow \emptyset$;
$\mathbb{C}^k \leftarrow \Omega$; /* $\mathbb{C}^k$ is the set of candidate itemsets at level k */
while $\mathbb{C}^k \neq \emptyset$ **do**
  $\mathbb{F}^k \leftarrow \emptyset$; /* $\mathbb{F}^k$ is the set of frequent itemsets at level k */
  for all $f \in \mathbb{C}^k$ **do**
    compute $e(S_f)$ according to formula (6);
    if $e(S_f) \geq t_{sup}$ **then**
      add f to $\mathbb{F}^k$;
    end if
  end for
  $\mathbb{F} \leftarrow \mathbb{F} \cup \mathbb{F}^k$;
  generate $\mathbb{C}^{k+1}$ from $\mathbb{F}^k$; /* $\mathbb{C}^{k+1}$ is generated by the way the Apriori algorithm uses in [AS94] */
  $k \leftarrow k + 1$;
end while
return the set of *frequent itemsets* $\mathbb{F}$;

Note: *in Algorithm EST, the candidate itemsets are generated using what the traditional Apriori algorithm [AS94] ueses. However, to discover the itemsets with estimated support values above the support threshold, the a priori property does not hold. So the itemsets discovered by EST are a subset of those actually qualifying under the criterion.*

6 Experiments

In our experiments, we compare algorithm EST with algorithms DET1, DET2 and EXP defined in Section 4.

An uncertain dataset $\mathcal{D}$ is generated in the following way: A true dataset $\mathcal{D}^T$ with a set of items Ω is first generated by the IBM Almaden generator [AS94] with parameters T=10, I=4, D=10K, and N=0.1K.[1] $\mathcal{D}^T$ contains 10,000 tuples and 100 distinct items. Then for each item $X \in \Omega$, its values in $\mathcal{D}^T$ are assigned tags according to the tag distributions $f_X(tag_X), g_X(tag_X)$. Without loss of generality, let $f_X(tag_X)$ for each item $X \in \Omega$ be the same. Depending on the support value of X in $\mathcal{D}^T$, a corresponding tag distribution $g_X(tag_X)$ is determined so that the tag constraint for X is satisfied on $\mathcal{D}$.

To study how the quality of an uncertain dataset affect the mining result, we use three different $f_X(tag_X)$s (see Table 3). They correspond to an uncertain dataset with good, medium and bad quality respectively. Based on each $f_X(tag_X)$, 100 uncertain datasets are generated from $\mathcal{D}^T$. The total number of frequent itemsets mined from $\mathcal{D}^T$ under the support threshold 100 (1% of $10,000$) is $13,243$. The average mining results over each 100 datasets by the four approaches are shown in Table 4. Here, **Num** represents the number of frequent itemsets (Num) that are discovered, **FPs** and **FNs** represent the number of false positive itemsets and false negative itemsets respectively.

Good		Medium		Bad	
tag_X	$f_X(tag_X)$	tag_X	$f_X(tag_X)$	tag_X	$f_X(tag_X)$
0.7	0.1	0.1	0.1	0.3	0.4
0.8	0.4	0.6	0.4	0.4	0.4
0.9	0.4	0.8	0.4	0.9	0.1
1	0.1	1	0.1	1	0.1

Table 3: three tag distributions $f_X(tag_X)$ for item X

The result indicates that DET1 has a very large number of false positive or false negative itemsets, so we will not consider it further. DET2 and EXP have quite similar number of false positive and false negative itemsets. However, neither of them discovered more than half of the true frequent itemsets.

In the following, we study the results obtained from one uncertain dataset with medium quality. Table 5 lists the number of frequent itemsets mined by DET2, EXP and EST, the FPs and FNs at each level. The results indicate that DET2 and EXP have difficulty discovering frequent itemsets at level 6 or higher.

Figure 3 shows the true support value distributions for the false positive and false negative itemsets under methods EXP and EST. We can see that most of the false positive and false negative itemsets in EST have true support values around the support threshold 100.

One might consider an itemset with support value within $\pm 5\%$ of the support threshold is a "marginal" frequent/infrequent itemset and not consider these as errors even if misclassified as frequent or not since they are so close to the threshold. Table 6 shows the number of FNs and FPs remaining when those itemsets with support within $\pm 5\%$ of the threshold are removed from the FP and FN sets in Table 5. For the EST

[1] The parameter's notation follows the naming convention in [AS94]: T represents the average tuple size; I represents the average size of the maximal potentially large itemsets; D represents the number of tuples; and N represents the number of distinct items.

quality	method	Num		FPs		FNs	
		mean	sd	mean	sd	mean	sd
	result on $\mathcal{D}^T$	13243	0	-	-	-	-
Good	DET1	27205.61	171.10	13962.61	171.10	0	0
	DET2	8124.59	94.12	377.01	22.18	5495.42	81.93
	EXP	7939.24	38.39	307.41	13.00	5611.17	33.66
	EST	13203.68	116.44	572.33	68.79	611.65	65.09
Medium	DET1	29906.84	290.84	17448.51	277.36	784.67	37.35
	DET2	5935.18	79.03	702.8	29.50	8010.62	56.33
	EXP	5783.6	26.45	624.72	11.28	8084.12	20.63
	EST	13218.34	236.25	1504	141.60	1528.66	125.39
Bad	DET1	58.12	1.45	0	0	13184.88	1.45
	DET2	5156.58	74.93	991.58	37.45	9078	43.23
	EXP	5034.17	15.22	917.7	9.02	9126.53	10.04
	EST	12426.89	480.21	3869.76	313.56	4685.87	202.95

Table 4: Results of DET1, DET2, EXP and EST under uncertain datasets with different quality

algorithm, approximately 1/4 of the FP and 1/2 of the FN itemsets in Table 5 are in this marginal category.

level	true number	EXP			DET2			EST		
		FPs	FNs	Num	FPs	FNs	Num	FPs	FNs	Num
1	70	0	0	70	0	0	70	0	0	70
2	1177	119	36	1260	120	36	1261	23	29	1171
3	4210	453	1111	3552	538	1109	3639	321	339	4192
4	4228	45	3382	891	82	3253	1057	699	471	4456
5	2389	0	2357	32	0	2341	48	342	330	2401
6	937	0	937	0	0	936	1	122	191	868
7	209	0	209	0	0	209	0	21	39	191
8	22	0	22	0	0	22	0	2	1	23
9	1	0	1	0	0	1	0	0	0	1
All	13243	617	8055	5805	740	7907	6076	1530	1400	13373

Table 5: Comparison of DET2, EXP and EST at each level

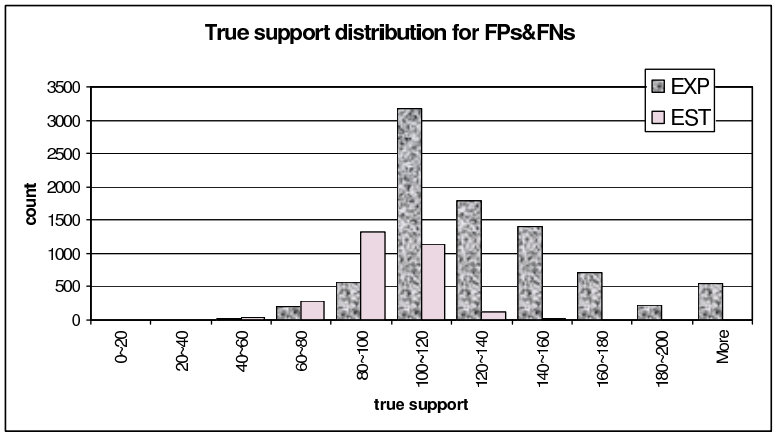

Figure 3: True support distribution for FPs&FNs under method EXP and EST

level	true number	EXP		DET2		EST	
		FPs	FNs	FPs	FNs	FPs	FNs
1	70	0	0	0	0	0	0
2	1177	94	29	94	27	9	17
3	4210	387	914	467	923	227	182
4	4228	34	2978	64	2859	508	262
5	2389	0	2060	0	2044	272	188
6	937	0	789	0	788	80	101
7	209	0	174	0	174	14	15
8	22	0	21	0	21	1	1
9	1	0	1	0	1	0	0
All	13243	515	6966	625	6837	1111	766

Table 6: FPs&FNs after the removal of the marginals

7 Conclusion and future work

In this paper, we define a generic model of uncertainty for binary datasets, and provide an algorithm EST to discover frequent itemsets under this model. Compared with existing approaches, EST makes use of a support estimator that can greatly reduce the number of false positive and false negative itemsets. In the future, we plan to study the variance of the estimator. Furthermore, since the estimator is based on the assumption that the tag constraints are satisfied, we plan to study how sensitive the estimator will be if the tag constraints are violated to some degree.

The uncertainty model in this paper is defined on a binary dataset, it could be extended to dataset with categorical or numeric values. We plan to study how the data mining algorithms should be adapted to such datasets.

Acknowledgment This work was supported in part by NSF grants IIS-0086116, ANI-0085773 and EAR-9817773.

References

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 1993.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.

[AS00] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.

[BGMP92] Daniel Barbará, Hector Garcia-Molina, and Daryl Porter. The management of probabilistic data. *TKDE*, 4(5):487–502, 1992.

[DZ03] Wenliang Du and Zhijun Zhan. Using randomized response techniques for privacy-preserving data mining. In *Proc. of 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, 2003.

[ESAG02] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proc. of 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, 2002.

[HV02] M. Halkidi and M. Vazirgiannis. Managing uncertainty and quality in the classification process. In *Proceedings of SETN Conference*, April 2002.

[KFW98] Chan Man Kuok, Ada Wai-Chee Fu, and Man Hon Wong. Mining fuzzy association rules in databases. *SIGMOD Record 27*, pages 41–46, 1998.

[KM02] Jeremy Martin Kubica and Andrew Moore. Probabilistic noise identification and data cleaning. Technical Report CMU-RI-TR-02-26, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, October 2002.

[Lee92] Suk Kyoon Lee. An extended relational database model for uncertain and imprecise information. In *Proc. 8th Int. Conf. Very Large Data Bases, VLDB*, 1992.

[LLRS97] Laks V. S. Lakshmanan, Nicola Leone, Robert Ross, and V. S. Subrahmanian. Probview: A flexible probabilistic database system. *TODS*, 22(3):419–469, 1997.

[MPV94] Juan Miguel Medina, Olga Pons, and Maria-Ampora Vila. GEFRED: A generalized model of fuzzy relational databases. *Information Sciences*, 76(1-2):87–109, 1994.

[RH02] Shariq Rizvi and Jayant R. Haritsa. Maintaining data privacy in association rule mining. In *Proc. 28th Int. Conf. Very Large Data Bases, VLDB*, 2002.

[Sha76] G. Shafter. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[YWYH02] Jiong Yang, Wei Wang, Philip Yu, and Jiawei Han. Mining long sequential patterns in a noisy environment. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 406–417, 2002.

[ZCF+97] Carlo Zaniolo, Stefano Ceri, Christos Faloutsos, Richard T. Snodgrass, V.S. Subrahmanian, and Roberto Zicari. *Advanced Database Systems*, chapter Part V: Uncertainty in Databases and Knowledge Bases, pages 315–411. Morgan Kaufmann Publishers, 1997.

Appendix

A Proof of Theorem 1

Assume itemset $\mathcal{I} = \{I_1, I_2, \ldots, I_K\}$. Each item I_j in $\mathcal{I}$ has tag distributions $f_{I_j}(\alpha), g_{I_j}(\alpha)$. Let the number of distinct tag values item I_j can take on be M_{I_j}. Then, itemset $\mathcal{I}$ could take on a total of $J = \prod_{j=1}^{K} M_{I_j}$ distinct combinations of tag values. We denote the ith tag combination by

$$\boldsymbol{\alpha}_i =< \alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iK} >, \quad i \in \{1, 2, \ldots, J\}.$$

The true dataset $\mathcal{D}^T$ can be viewed as composed of 2^K disjoint blocks: $\mathcal{D}^T = \bigcup_{f \subseteq \mathcal{I}} B_f$. All the tuples in block B_f support f and no other subset of $\mathcal{I}$ that contains f. For example, for $\mathcal{I} = \{X, Y\}$, $\mathcal{D}^T$ can be partitioned into 4 blocks: $B_\emptyset, B_X, B_Y$ and B_{XY}. B_X contains tuples that support X but not XY. Correspondingly, the uncertain dataset $\mathcal{D}$ derived from $\mathcal{D}^T$ can also be partitioned into 2^K blocks: $\mathcal{D} = \bigcup_{f \subseteq \mathcal{I}} B'_f$, where B'_f is generated from B_f. We denote the number of tuples in B_f by $\mathbb{N}_f$, $\mathbb{N}_f = |B_f| = |B'_f|$.

Now let's look at block B_f. An indicator function $r(I_j)$ is defined in B_f for item $I_j \in \mathcal{I}$, such that:

$$r(I_j) = \begin{cases} 1, & \text{if } I_j \in f; \\ 0, & \text{otherwise.} \end{cases}$$

The probability of $\mathcal{I}$ having the ith tag combination $\boldsymbol{\alpha}_i$ in B_f is:

$$p_{\boldsymbol{\alpha}_i} = p(\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iK}) = \prod_{j=1}^{K} [f_{I_j}(\alpha_{ij})]^{r(I_j)} [g_{I_j}(\alpha_{ij})]^{1-r(I_j)}.$$

Let $N_{\boldsymbol{\alpha}_i}$ be the number of tuples in B'_f where $\mathcal{I}$ takes the ith tag combination, then $\sum_{i=1}^{J} N_{\boldsymbol{\alpha}_i} = \mathbb{N}_f$. Actually, $N_{\boldsymbol{\alpha}_1}, N_{\boldsymbol{\alpha}_2}, \ldots, N_{\boldsymbol{\alpha}_J}$ follow a multinomial distribution with parameters $p_{\boldsymbol{\alpha}_1}, p_{\boldsymbol{\alpha}_2}, \ldots, p_{\boldsymbol{\alpha}_J}$. That is,

$$p(N_{\boldsymbol{\alpha}_1} = n_1, N_{\boldsymbol{\alpha}_2} = n_2, \ldots, N_{\boldsymbol{\alpha}_J} = n_J) = \mathbb{N}_f! \prod_{i=1}^{J} \frac{p_{\boldsymbol{\alpha}_i}^{n_i}}{n_i!}.$$

Now consider a tuple t' in B'_f where itemset $\mathcal{I}$ has the ith tag combination $\boldsymbol{\alpha}_i$. Since α_{ij} represents the probability of item I_j being present in t —— the corresponding tuple of t' in B_f, the probability of itemset $\mathcal{I}$ being present in t (i.e., all items in $\mathcal{I}$ are present in t) is:

$$\delta_{\boldsymbol{\alpha}_i} = \prod_{j=1}^{K} \alpha_{ij}.$$

Let X_t be a binary variable, indicating whether itemset $\mathcal{I}$ is present in t. Then,

$$X_t = \begin{cases} 1, & \text{with probability } \delta_{\boldsymbol{\alpha}_i}; \\ 0, & \text{with probability } 1 - \delta_{\boldsymbol{\alpha}_i}. \end{cases}$$

The z-transform for X_t is:

$$\mathcal{G}_{X_t}(z) = 1 - \delta_{\boldsymbol{\alpha}_i} + \delta_{\boldsymbol{\alpha}_i} z = 1 - \delta_{\boldsymbol{\alpha}_i}(1 - z).$$

If there are $N_{\boldsymbol{\alpha}_i}$ tuples in B'_f where $\mathcal{I}$ takes the ith tag combination $\boldsymbol{\alpha}_i$, then the z-transform for $\sum X_t$ over those tuples is:

$$[\mathcal{G}_{X_t}(z)]^{N_{\boldsymbol{\alpha}_i}}.$$

In the following, we simplify the notation $N_{\boldsymbol{\alpha}_i}$ as N_i; $\delta_{\boldsymbol{\alpha}_i}$ as δ_i, and $p_{\boldsymbol{\alpha}_i}$ as p_i.

Given N_i for $i = 1, 2, \ldots, J$ in B'_f, the z-transform for $\sum_{t \in B_f} X_t$ is:

$$\prod_{i=1}^{J} [1 - \delta_i(1 - z)]^{N_i}.$$

Unconditioned on $N_1, \ldots, N_J$, the z-transform for $\sum_{t \in B_f} X_t$ is:

$$\begin{aligned}
\mathcal{G}(z) &= \sum_{n_1 + n_2 + \ldots + n_J = \mathbb{N}_f} \{p(N_1 = n_1, \ldots, N_J = n_J) \prod_{i=1}^{J} [1 - \delta_i(1 - z)]^{n_i}\} \\
&= \sum_{n_1 + n_2 + \ldots + n_J = \mathbb{N}_f} \{[\mathbb{N}_f! \prod_{i=1}^{J} \frac{p_i^{n_i}}{n_i!}] * \prod_{i=1}^{J} [1 - \delta_i(1 - z)]^{n_i}\} \\
&= \sum_{n_1 + n_2 + \ldots + n_J = \mathbb{N}_f} \{\mathbb{N}_f! \prod_{i=1}^{J} \frac{[p_i(1 - \delta_i(1 - z))]^{n_i}}{n_i!}\} \\
&= [\sum_{i=1}^{J} p_i(1 - \delta_i(1 - z))]^{\mathbb{N}_f}.
\end{aligned}$$

According to the z-transform, the mean of $\sum_{t\in B_f} X_t$ can be computed as follows:

$$\begin{aligned}
m_{B_f} &= \frac{\partial \mathcal{G}}{\partial z}\Big|_{z=1} \\
&= \mathbb{N}_f[\sum_{i=1}^{J} p_i(1-\delta_i(1-z))]^{\mathbb{N}_f-1}[\sum_{i=1}^{J} p_i\delta_i]\Big|_{z=1} \\
&= \mathbb{N}_f \sum_{i=1}^{J} p_i\delta_i \\
&= \mathbb{N}_f \sum_{i=1}^{J} p_{\boldsymbol{\alpha}_i}\delta_{\boldsymbol{\alpha}_i} \\
&= \mathbb{N}_f \sum_{i=1}^{J} \{ \prod_{j=1}^{K} \{[f_{I_j}(\alpha_{ij})]^{r(I_j)}[g_{I_j}(\alpha_{ij})]^{1-r(I_j)}\} \prod_{j=1}^{K} \alpha_{ij}\} \\
&= \mathbb{N}_f \sum_{i=1}^{J} \{ \prod_{j=1}^{K} \{[f_{I_j}(\alpha_{ij})]^{r(I_j)}[g_{I_j}(\alpha_{ij})]^{1-r(I_j)}\alpha_{ij}\}\} \\
&= \mathbb{N}_f \prod_{j=1}^{K} \{ \sum_{l=1}^{M_{I_j}} \{[f_{I_j}(\alpha_l)]^{r(I_j)}[g_{I_j}(\alpha_l)]^{1-r(I_j)}\alpha_l\}\}.
\end{aligned}$$

Let $F_{I_j} = \sum_{l=1}^{M_{I_j}} f_{I_j}(\alpha_l)\alpha_l$, $G_{I_j} = \sum_{l=1}^{M_{I_j}} g_{I_j}(\alpha_l)\alpha_l$. We get,

$$m_{B_f} = \mathbb{N}_f \prod_{j=1}^{K} [F_{I_j}^{r(I_j)} G_{I_j}^{1-r(I_j)}]. \tag{15}$$

Since the tuples are independent of each other, we have,

$$E[\widetilde{S_{\mathcal{I}}}] = \sum_{f\subseteq\mathcal{I}} m_{B_f}. \tag{16}$$

According to the Principle of Inclusion/Exclusion, we have

$$\mathbb{N}_f = S_f - \sum_{f\subset\omega\subseteq\mathcal{I},|\omega|=|f|+1} S_\omega + \ldots + (-1)^i \sum_{f\subset\omega\subseteq\mathcal{I},|\omega|=|f|+i} S_\omega + \ldots + (-1)^{|\mathcal{I}|-|f|} S_{\mathcal{I}}. \tag{17}$$

Here, S_f, S_ω are the true support for itemset f and ω in $\mathcal{D}^T$ respectively. Replacing $\mathbb{N}_f$ in equation (13) with (12) and (14), we get

$$E[\widetilde{S_{\mathcal{I}}}] = S_{\mathcal{I}} * \prod_{I_i\in\mathcal{I}} (F_{I_i} - G_{I_i}) + \sum_{f\subset\mathcal{I}} \{S_f * \prod_{I_i\in f} (F_{I_i} - G_{I_i}) * \prod_{I_i\in\mathcal{I}\setminus f} G_{I_i}\}. \tag{18}$$

Following is an example:

For itemset $\mathcal{I} = \{X, Y\}$, from equation (13), we get

$$E[\widetilde{S_{XY}}] = \mathbb{N}_{XY} F_X F_Y + \mathbb{N}_Y G_X F_Y + \mathbb{N}_X F_X G_Y + \mathbb{N}_\emptyset G_X G_Y. \quad (19)$$

Since

$$\mathbb{N}_Y = S_Y - S_{XY}; \quad (20)$$
$$\mathbb{N}_X = S_X - S_{XY}; \quad (21)$$
$$\mathbb{N}_\emptyset = S_\emptyset - S_X - S_Y + S_{XY}; \quad (22)$$

Replacing $\mathbb{N}_Y, \mathbb{N}_X$ and $\mathbb{N}_\emptyset$ in Equation (16) with (17), (18), (19), we get:

$$E[\widetilde{S_{XY}}] = (F_X - G_X)(F_Y - G_Y)S_{XY} + (F_X - G_X)G_Y S_X + G_X(F_Y - G_Y)S_Y + G_X G_Y S_\emptyset. \quad (23)$$