Mining Probabilistic Generalized Frequent Itemsets in Uncertain Databases

Erich A. Peterson
Myeloma Institute for Research and Therapy
University of Arkansas for Medical Sciences
contact@erichpeterson.com

Peiyi Tang
Department of Computer Science
University of Arkansas at Little Rock
pxtang@ualr.edu

ABSTRACT

Researchers have recently defined and presented the theoretical concepts and an algorithm necessary for mining so-called probabilistic frequent itemsets in uncertain databases—based on possible world semantics. Further, there exist algorithms for mining so-called generalized itemsets in certain databases, where a taxonomy exists relating concrete items to abstract (generalized) items not in the database. Currently, no research has been done in formulating a theory and algorithm for mining generalized itemsets from uncertain databases. Using probability theory and possible world semantics, we formulate a method for calculating the probability a generalized item will occur within an uncertain transaction.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; G.3 [**Probability and Statistics**]: Distribution functions

Keywords

Probabilistic generalized frequent itemsets, existential probability of generalized itemsets, uncertain databases

1. INTRODUCTION

Data mining *generalized frequent itemsets* (GFIs) and generalized association rules was first proposed by Srikant et al. [6]. The methods employed in what will sometimes be referred to as generalized itemset mining in this paper, take much inspiration from the algorithms and concepts of traditional frequent itemset and association rules mining [1, 2, 5][1]. Generalized itemset mining differs from traditional itemset mining, through the addition of a taxonomy G to an itemset database T. This taxonomy forms an is-a relationship among items, and is represented as a directed acyclical graph. An example taxonomy used throughout this paper is shown in Figure 1. In G one sees that apple and banana is-a fruit, and fruit is-a produce. Further, within G there exists an ancestor/descendant relationship among all items within G. For example, an item a is said to be a descendant of a', if and only if, there exists a path from a' to a. Conversely, a' is called an ancestor of a. An arbitrary ancestor of an item a is denoted as $\hat{a}$. With the addition of a taxonomy, one can develop new problems and discover knowledge previously not easily discernible. One such problem, is to find "rules that span different levels of the taxonomy." [6] This problem is especially important as items at lower levels of the taxonomy may not have the minimum support necessary to qualify as frequent; however, items higher up in the taxonomy may. For example, "department stores or supermarkets typically have hundreds of thousands of items, [and] the support for rules involving only leaf items (typically UPC or SKU codes) tends to be extremely small." [6] Another data mining problem, which has received a fair amount of more recent research, is that of mining for so-called *probabilistic frequent itemsets* (PFIs). First proposed by Bernecker et al. [4], this work concerns itself with mining of frequent itemsets in *uncertain* databases; that is, databases in which each item in the database T has an associated existential probability—denoting the probability of its occurrence within a transaction. In this work, an arbitrary itemset is said to be probabilistically frequent if its *frequentness probability*, defined as the probability the item will have a support greater than or equal to a user-defined minimum support threshold $minsup$, is greater than or equal to a user-defined confidence threshold τ. Thus, to find the frequentness probability of a particular itemset, a discrete probability distribution function is calculated, which describes the probability of the itemset's support being a certain value, and is based on possible world semantics.

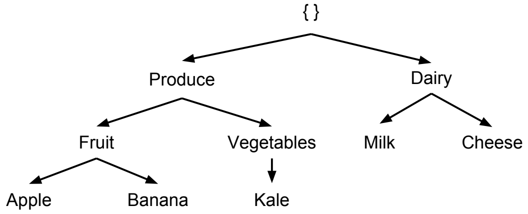


Figure 1: Example Taxonomy G

Given an uncertain database T and a taxonomy G, no method currently exists to mine for generalized frequent itemsets in uncertain databases. The two salient problems that need to be solved to formulate such a method are: 1) A method needs to be formulated to calculate the probability of a generalized item occurring within a particular transaction according to possible world semantics; 2) An algorithm is needed which will enumerate candidate itemsets

[1] Although generalized itemset mining was discovered after traditional itemset mining, as the name implies, generalized itemset mining constitutes a general case of the special case of itemset mining.

ACMSE'13, April 4-6, 2013, Savannah, GA, USA.
Copyright 2013 ACM 978-1-4503-1901-0/13/04 ...$15.00.

(including generalized items) in an efficient manner. In this paper, we formulate a method for calculating the probability of a particular generalized item occurring within a particular transaction. This method can be used to extend the original uncertain database, to then mine for so-called *probabilistic generalized frequent itemsets* (PGFIs). We further present an algorithm for mining these concepts from uncertain databases, and provide an experimental evaluation of the algorithm.

The rest of the paper is laid out as follows: section 2 provides preliminary concepts necessary for understanding the rest of the paper (including concepts such as generalized frequent itemset mining and the uncertain data model used); section 3 disseminates the new PGFI concepts and algorithm `PGFIM` for probabilistic generalized frequent itemset mining; section 4 provides the reader with an experimental evaluation of the newly formulated algorithm; and finally, section 5 provides a conclusion.

2. PRELIMINARIES

2.1 Generalized Frequent Itemsets

The taxonomy G is represented as a weakly connected directed acyclical graph; formally, as the pair $G = (A, E)$, where A and E both sets, and $E \subseteq A \times A$. The elements of A are vertices, which represent distinct items; and the elements of E represent directed edges between elements of $A = \{a_1, a_2, \ldots, a_m\}$. If a directed edge between a vertex a_i and a_j exists, a_j is called a child of a_i, and conversely, a_i is called the parent of a_j. Further, G has the constraint that it is weakly connected, that is, ignoring the direction of the edges within G, a path must exist from all vertices to all other vertices. Further, it must also be acyclical, i.e., there is no path from a node to itself. A taxonomy G contains vertices representing both items found in the databases (leaf items), and those which are generalized or abstracted (non-leaf) items. A vertex $a \in A$ represents a non-leaf item, if and only if, a has at least one child. Also, a vertex a represents a leaf item, if and only if, a has no children. The set of vertices representing items are partitioned into the two disjoint sets: $A_L = \{a \mid a \text{ is a leaf item in } G\}$ and $A_{NL} = \{a \mid a \text{ is a non-leaf item in } G\}$. Also, let the taxonomy G define the relationships (e.g., ancestor and descendant) among each of the items $a \in A$. If a path in G exists from some item a' via directed edges to another item a, we say a' is an ancestor of a; and conversely, that a is a descendant of a'. Let the set $D(a) = \{x \mid x \in A_L \wedge \ x \text{ is a descendant of } a\}$ denote the set of all descendant leaf items in G of an arbitrary non-leaf item $a \in A_{NL}$.

A generalized itemset I is defined to be any subset of A, i.e., $I \subseteq A$. Normally, only leaf items are found in the actual database, i.e., elements of A_L (alluded to above), and in this paper this is assumed to be the case. That is, given a database $T = \{t_1, t_2, \ldots, t_n\}$, each t_j is a subset of only A_L. For itemsets which contain only leaf items, the set inclusion and subset (i.e., $\in$ and $\subseteq$, respectively) notation meanings are intuitive. However, given the presence of non-leaf (generalized) item, the previously mentioned notions must be redefined.

DEFINITION 1 (GENERALIZED ITEM SET INCLUSION). *An item $a \in A$ is an element of a set S with respect to taxonomy G, denoted as $a \in_G S$, if and only if, $a \in S$, or there exists an $a' \in D(a)$ such that $a' \in S$.*

DEFINITION 2 (GENERALIZED ITEMSET SUBSET-SUPERSET). *An itemset $I \subseteq A$ is a subset of a set S with respect to taxonomy G, denoted as $I \subseteq_G S$, if and only if, for each item $a \in I$, $a \in_G S$.*

TID	Itemset
t_1	$apple, banana, kale$
t_2	$apple, banana, cheese$
t_3	$apple, milk, cheese$

Figure 2: Example Database

Further, we say that the *support* of an arbitrary *generalized itemset* $I \subseteq A$ over the database T, denoted $Sup_T(I)$, is the the number of transactions in T that contains I with respect to taxonomy G, and is defined formally as:

$$Sup_T(I) = |\{t \mid I \subseteq_G t \wedge t \in T)\}| \qquad (1)$$

For example, if given the example itemset database in Figure 2 and the taxonomy in Figure 1, the support of the itemset $\{fruit, cheese\}$ is 2, because $\{fruit, cheese\} \subseteq_G t_2$, $\{fruit, cheese\} \subseteq_G t_3$, but $\{fruit, cheese\} \nsubseteq_G t_1$. If the support of a generalized itemset $I \subseteq A$ is greater than or equal to some user-defined threshold $minsup$, then I is considered *frequent*. Thus, the definition of a *generalized frequent itemset* (GFI) is given below.

DEFINITION 3. *Given a taxonomy G and an itemset database $T = \{t_1, t_2, \ldots, t_n\}$, where each $t_j \subseteq A_L$, a generalized itemset $I \subseteq A$ is considered a* generalized frequent itemset (GFI)*, if and only if, $Sup_T(I) \geq minsup$, where $minsup \in [0, n]$ is a user-defined threshold.*

Thus, the problem statement for mining GFIs, is to discover all (generalized) itemsets $I \subseteq A$, whose support is greater than or equal $minsup$.

2.2 Uncertain Data Model

The major difference between traditional frequent itemset mining and probabilistic frequent itemset mining, is that in the traditional case, one knows with certitude whether or not an item occurs within a particular transaction or not; whereas, in the probabilistic case, one may only have the probability of an item occurring within a particular transaction. Let $T = \{t_1, t_2, \ldots t_n\}$ be an *uncertain database*, in which each transaction t_j consists of a set of pairs $(a_i, Pr(a_i \in t_j))$, where $a_i \in A$ and $Pr(a_i \in t_j)$, denotes the probability of item a_i occurring in transaction t_j. The probability of a_i occurring within transaction t_j, is between 0 and 1. An *uncertain item* a_i within a transaction t_j, is one which $Pr(a_i \in t_j) \in (0, 1)$. If $Pr(a_i \in t_j) = 1$ then the item a_i is known to occur with certitude and it is a certain item in t_j[2]; otherwise a_i is called an uncertain item. Let each transaction t be partitioned into two disjoint sets—one containing all uncertain items $u_1 u_2 \cdots u_{L_t}$, i.e., $Pr(u_i \in t) \in (0, 1)$, and the other which contains all certain items $c_1 c_2 \cdots c_{N_t}$, i.e., $Pr(c_i \in t) = 1$. Figure 3 shows an example uncertain database, where t_1 has two uncertain items, $apple$ and $banana$ with existential probabilities 0.89 and 0.99, respectively, and one certain item $kale$. Transactions t_2 and t_3 both have three uncertain items and no certain items. A *possible world* of transaction t is a concrete instance of t which contains all certain items $c_1 c_2 \cdots c_{N_t}$ and either contains each uncertain item u_i $(1 \leq i \leq L_t)$ or not. Therefore, a possible world can be derived from an L_t-bit binary string $v = v_1 v_2 \cdots v_{L_t} \in \{0, 1\}^{L_t}$ through a bijection from the set of L_t-bit binary strings to the set of possible worlds of t, denoted as $W(t)$, $\phi : \{0, 1\}^{L_t} \to W(t)$, defined as:

$$\phi(v) = \phi(v_1 v_2 \cdots v_{L_t}) = b_1 b_2 \cdots b_{L_t} c_1 c_2 \cdots c_{N_t}$$

[2] Items having a probability of zero occurring within a particular transaction are not shown in the uncertain database—as they certainly do not occur.

TID	Uncertain Itemset
t_1	$(apple, 0.89), (banana, 0.99), (kale, 1.0)$
t_2	$(apple, 0.4), (banana, 0.45), (cheese, 0.12)$
t_3	$(apple, 0.9), (milk, 0.95), (cheese, 0.20)$

Figure 3: Example Uncertain Database

where

$$b_i = \begin{cases} u_i, & \text{if } v_i = 1 \\ \epsilon, & \text{if } v_i = 0 \end{cases}$$

for $i = 1, \ldots, L_t$ and ϵ is the empty symbol. Since there are 2^{L_t} L_t-bit binary strings in $\{0,1\}^{L_t}$, we must have 2^{L_t} possible worlds in $W(t)$. Let a possible world in $W(t)$ be denoted by $w(t) = \phi(v)$. The probability of possible world $w(t) = \phi(v)$, denoted by $Pr(w(t))$, is calculated by:

$$\begin{aligned} Pr(w(t)) &= Pr(\phi(v), t) \\ &= Pr(\phi(v_1 v_2 \cdots v_{L_t}), t) \\ &= \prod_{v_i=1} Pr(u_i \in t) \cdot \prod_{v_i=0} (1 - Pr(u_i \in t)) \end{aligned} \tag{2}$$

if independence among the uncertain items if assumed. From (2), we have

$$\sum_{v \in \{0,1\}^{L_t}} Pr(\phi(v), t) = 1$$

In other words, (2) defines a probability distribution for all possible worlds of $W(t)$.

In a traditional (certain) itemset database, the probability of itemset I occurring in transaction t is 1 if $I \subseteq t$, or 0 otherwise. In contrast, for an uncertain itemset database, the probability of I occurring in t, denoted as $Pr(I \subseteq t)$, is the sum of the probabilities of all the possible worlds $w \in W(t)$ which contain I. Let $W_I(t)$ be the set of possible worlds, which contain I; that is, $W_I(t) = \{w \mid w \in W(t) \wedge I \subseteq w\}$. Further, recall that the function ϕ is a bijection, and therefore, its inverse ϕ^{-1} is a mapping from $W(t)$ to $\{0,1\}^{L_t}$; that is: $\phi^{-1} : W(t) \rightarrow \{0,1\}^{L_t}$. Let S be a subset of $W(t)$. We use $\phi^{-1}(S)$ to denote the range of function ϕ^{-1} from S; that is: $\phi^{-1}(S) = \{\phi^{-1}(w) \mid w \in S\}$. Thus, $\phi^{-1}(W_I(t))$ is the set of all L_t-bit binary strings, whose corresponding possible worlds contain I, that is, for all $v \in \phi^{-1}(W_I(t))$, $I \subseteq \phi(v)$. Finally, let $\phi(v)_i$ be the i-th element of $\phi(v)$, $1 \leq i \leq L_t + N_t$, and v_i be the i-th element of v, $i \leq i \leq L_t$. With that, one may calculate $Pr(I \subseteq t)$ as:

$$\begin{aligned} & Pr(I \subseteq t) \\ &= \sum_{v \in \phi^{-1}(W_I(t))} Pr(\phi(v), t) \\ &= \sum_{v \in \phi^{-1}(W_I(t))} \left(\prod_{v_i=1} Pr(u_i \in t) \cdot \prod_{v_i=0} (1 - Pr(u_i \in t)) \right) \\ &= \prod_{\phi(v)_i \in I} Pr(u_i \in t) \cdot \prod_{\phi(v)_i \notin I} (Pr(u_i \in t) + (1 - Pr(u_i \in t)) \\ &= \prod_{\phi(v)_i \in I} Pr(u_i \in t) \\ &= \prod_{u_i \in I} Pr(u_i \in t) \end{aligned} \tag{3}$$

Since $Pr(c_i \in t) = 1$ for a certain item in t, one can calculate $Pr(I \subseteq t)$ simply as:

$$Pr(I \subseteq t) = \prod_{a \in I} Pr(a \in t) \tag{4}$$

The set of all possible worlds W induced by *all* transactions in the uncertain database $T = \{t_1, \ldots, t_n\}$ is the Cartesian product of $W(t_j), j = 1, \ldots, n$ as follows:

$$W = W(t_1) \times W(t_2) \times \cdots \times W(t_n) \tag{5}$$

A simplified example uncertain database is shown in Figure 4(a), and the possible worlds of each transaction is shown in Figure 4(b). (Each transaction of each possible world in $W(t_j)$ is enclosed by $\langle\ \rangle$.). If the assumption of independence between the transactions in T is valid, the probability of a possible world $w = (w(t_1), w(t_2), \ldots, w(t_n)) \in W$ can be calculated as follows:

TID	Uncertain Itemset
t_1	$(banana, 0.99)$
t_2	$(banana, 0.45), (kale : 0.4)$

(a) Simplified Example

	Possible Worlds
$W(t_1)$	$\langle\ \rangle$, $\langle banana \rangle$
$W(t_2)$	$\langle\ \rangle$, $\langle banana \rangle$, $\langle kale \rangle$, $\langle banana, kale \rangle$

(b) Possible Worlds

Figure 4: Simple Example of Possible Worlds

$$Pr(w) = \prod_{i=1}^{n} Pr(w(t_i)) \tag{6}$$

where $Pr(w(t))$ is calculated by (2).

In a traditional (certain) itemset database $T = \{t_1, \ldots, t_n\}$, the support of itemset I in transaction t, denoted $Sup_t(I)$, is 1 if $I \subseteq t$ or 0 otherwise. The support of I over the entire database T, denoted $Sup_T(I)$ is:

$$Sup_T(I) = Sup_{t_1}(I) + Sup_{t_2}(I) + \cdots + Sup_{t_n}(I) \tag{7}$$

Notice (7) is equal to the number of transactions that contain I, i.e., $Sup_T(I) = |\{t \mid I \subseteq t \wedge t \in T\}|$.

However, in an *uncertain* database T, the support of I in transaction t_j, $Sup_{t_j}(I)$, is no longer a concrete 0 or 1. Instead, it is a random variable X_j^I following a Bernoulli distribution with parameter p_j, where $p_j = Pr(X_j^I = 1) = Pr(I \subseteq t_j)$ (calculated by (4)) and $1 - p_j = Pr(X_j^I = 0) = Pr(I \nsubseteq t_j)$ (or the probability of success and failure, respectively). Therefore, in an uncertain database T, the support of I over the entire database T is a random variable $X^I = \sum_{j=1}^{n} X_j^I$ (according to (7)). The random variable X^I follows the Poisson binomial distribution with parameters $p_j = Pr(I \subseteq t_j), j = 1, \ldots, n$, if the assumption of independence between transactions is made.

The probability that $X^I = i$, $(0 \leq i \leq n)$, is:

$$Pr(X^I = i) = \sum_{S \subseteq T, |S|=i} \left(\prod_{t_j \in S} p_j \cdot \prod_{t_j \in T-S} (1 - p_j) \right) \tag{8}$$

Equation (8) is true, because the set $S \subseteq T$ has exactly i transactions, and the probability of an arbitrary possible world $w \in W$ in which all i transactions in S contain I and the rest (n - i) do not, is equal to $\prod_{t_j \in S} p_j \cdot \prod_{t_j \in T-S} (1 - p_j)$. Given $minsup$ and a confidence threshold $\tau \in [0,1]$, an itemset I is said to be *frequent* with confidence τ, if and only if, $Pr(X^I \geq minsup) \geq \tau$. One can calculate $Pr(X^I \geq minsup)$ using the following formula:

$$Pr(X^I \geq minsup) \\ = \sum_{S \subseteq T, |S| \geq minsup} \left(\prod_{t_j \in S} p_j \cdot \prod_{t \in T-S} (1-p_j) \right) \quad (9)$$

DEFINITION 4. *Given an uncertain database T and an itemset I, I is a* probabilistic frequent itemset (PFI) *with confidence τ, if and only if $Pr(X^I \geq minsup) \geq \tau$, where $minsup \in [0, n]$ and $\tau \in [0, 1]$ are user-defined thresholds. [4]*

Finally, the problem of mining probabilistic frequent itemsets, is to discover all itemsets $I \subseteq A$ such that $Pr(X^I \geq minsup) \geq \tau$, where $minsup$ and τ are user-defined thresholds.

3. MINING GENERALIZED PROBABILISTIC FREQUENT ITEMSETS

In section 2.1, the problem statement and theory behind the mining of arbitrary generalized frequent itemsets $I \subseteq A = A_L \cup A_{NL}$—given a taxonomy G and an itemset database T, in which each transaction t is a subset of A_L—was disseminated. Further, in section 2.2, the problem statement and theory behind the mining of probabilistic frequent itemsets from uncertain databases was presented. In that domain, no taxonomy exists. Instead each transaction t is a set of pairs $(a_i, Pr(a_i \in t)$, where $a_i \in A$ and the $Pr(a_i \in t)$ denotes the probability of item a_i appearing in transaction t. So far, no research has been done to formulate a method for mining generalized itemsets from uncertain databases.

DEFINITION 5. *Given a taxonomy $G = (A, E)$, $A = A_L \cup A_{NL}$, and an uncertain database $T = \{t_1, \ldots, t_n\}$, which only contains items in A_L, an itemset $I \subseteq A$ is considered a* probabilistic generalized frequent itemset *(PGFI) with confidence τ, if and only if, $Pr(X^I \geq minsup) \geq \tau$, where $minsup \in [0, n]$ and $\tau \in [0, 1]$ are user-defined thresholds.*

Thus, the problem statement for probabilistic generalized frequent itemset mining, is to discover all itemsets $I \subseteq A$, such that $Pr(X^I \geq minsup) \geq \tau$, where $minsup$ and τ are user-defined thresholds.

There are two major problems which need solving in order to successfully mine for PGFIs: 1) a way to calculate the existential probability of a generalized itemset occurring within an uncertain transaction; 2) a way to efficient calculate the aforementioned probability, and to enumerate possible generalized itemset candidates. The first problem is solved in section 3.1, and the second in section 3.2.

3.1 Calculating Existential Probabilities for Generalized Itemsets

The first major problem mining for PGFIs, is to formulate a method for calculating the probability that a generalized item, $g \in A_{NL}$, occurs within a particular transaction t, denoted as $Pr(g \in_G t)$. This should be the sum of the probabilities of all possible worlds $\phi(v)$ that contains at least one leaf item that is a descendant of g. Recall that $D(g)$ is the set of leaf nodes in A_L that are descendants of g. In other words,

$$Pr(g \in_G t) = \sum_{\phi(v) \cap D(g) \neq \emptyset} Pr(\phi(v), t) \quad (10)$$

Using (10) one is able to calculate the existential probability of an arbitrary non-leaf (generalized) item g occurring within a transaction t. However, (10) requires the enumeration of all possible worlds, and is therefore infeasible for all but trivial databases. Thus, similar to (3), we re-write the (10) as:

$$\begin{aligned} Pr(g \in_G t) &= \sum_{\phi(v) \cap D(g) \neq \emptyset} Pr(\phi(v), t) \\ &= 1 - \sum_{\phi(v) \cap D(g) = \emptyset} Pr(\phi(v), t) \\ &= 1 - \sum_{D(g) \subseteq \overline{\phi(v)}} Pr(\phi(v), t) \\ &= 1 - \prod_{x \in D(g)} (1 - Pr(x \in t)) \\ &\cdot \prod_{y \notin D(g)} (Pr(y \in t) + (1 - Pr(y \in t))) \\ &= 1 - \prod_{x \in D(g)} (1 - Pr(x \in t)) \end{aligned} \quad (11)$$

where $\overline{\phi(v)}$ is the complement of $\phi(v)$, i.e., $\overline{\phi(v)} = A_L(t) - \phi(v)$. Here $A_L(t)$ is the set of items in t, $u_1 \cdots u_{L_t} c_1 \cdots c_{N_t}$, and possible world $\phi(v)$ is a subset of A_L. If $x \in D(g)$ is a certain item c_j, we have $Pr(x \in t) = 1$, then $Pr(g \in t) = 1$. This is consistent with the understanding that if a leaf descendant of a generalized item g occurs in t with certitude, then g also occurs in t with certitude. When calculating the probability of a generalized item, one may choose to do so in an ad-hoc manner, or once there is a need to calculate the probability of the item, it can be done for every transaction in T and the corresponding probability added to each transaction—creating an extended database. In this way, future need for the probability can be simply "looked-up".

3.2 Efficient Enumeration & Probability Calculation

To discover all PGFIs, two salient questions must be addresses: 1) what type of enumeration scheme will be used to guide the mining process? 2) how will the probability distribution of X^I for an arbitrary $I \subseteq A$ will be calculated?

To simplify discussion of the enumeration technique used, the taxonomy shown in Figure 1 has been recreated in Figure 5, where each item has been abbreviated using the first letter of the corresponding item, and the generalized item $Dairy$ (and its descendants) have been pruned. In [7], Sriphaew et al. present the SET algorithm. That algorithm uses an efficient technique to enumerate candidate generalized itemsets in a *certain* itemset database. In this paper, we choose to use its fundamental enumeration scheme to determine which candidate generalized itemsets to enumerate and in which order. The SET algorithm's enumeration technique is performed in a top-down fashion, which uses both the subset-superset relationship, defined in Definition 2, over the items in A and the parent-child relationships defined by the taxonomy G. In simple terms, the search space is enumerated in such a fashion,

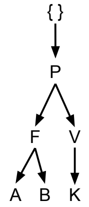


Figure 5: Simplified Taxonomy G

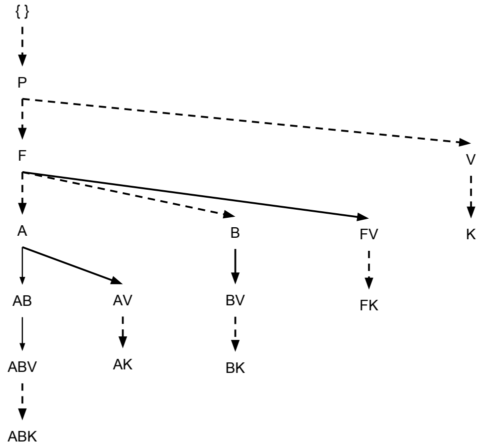


Figure 6: Example SET Enumeration

as to ensure a candidate is only generated if all of its subsets are also frequent; and thus, eliminating the gratuitous enumeration of candidates that could not possibly be frequent. This is done using the downward closure property of generalized and non-generalized itemsets. Under Definition 2, the generalized item P in Figure 5 is the smallest subset of all other possible generalized itemsets, with respect to G, because P or a descendant of P must be in any generalized itemset. Conversely, ABK is the largest superset. Thus, when we say the SET algorithm is performed in a top-down manner, we mean enumeration starts with the most general (smallest subset) candidates, to the most concrete (largest superset). Thus, the following subset-superset relationship exists, with respect to G, between P, ABK, and any arbitrary generalized itemset I: $P \subseteq_G I \subseteq_G ABK$. All candidate generalized itemsets enumerated by the SET algorithm using the simplified taxonomy in Figure 5, are shown in Figure 6. The reader is encouraged to see [7] for full details. Next, one must answer how the probability distribution of X^I for an arbitrary $I \subseteq A$ is calculated. Recall that X^I follows a Poisson binomial distribution. In [4], Bernecker et al. disseminated a method that uses a dynamic programming approach to calculate $Pr(X^I \geq minsup)$ in $O(|T|)$ time. To do so, calculating $Pr(X^I \geq i)$ is recast into the problem of calculating $Pr(X^I \geq i, j)$, which is the probability of X^I being greater than or equal i in the first j transactions of T.

Thus, the recursive equation used to drive the dynamic programming algorithm is (the reader is encouraged to see [4] for full details):

$$\begin{aligned} Pr(X^I \geq i, j) &= Pr(X^I \geq i-1, j-1) \cdot Pr(I \subseteq t_j) \\ &+ Pr(X^I \geq i, j-1) \cdot (1 - Pr(I \subseteq t_j)) \end{aligned}$$

$$\text{where } Pr(X^I \geq 0, j) = 1 \, \forall . 0 \leq j \leq |T|$$

$$Pr(X^I \geq i, j) = 0 \, \forall . i > j$$

4. EXPERIMENTAL EVALUATION

In order to perform an experimental evaluation of the PGFIM algorithm, two types of input must be provided: a taxonomy G and an uncertain transaction databases T. In the next two subsections we tackle how each in turn is generated.

4.1 Taxonomy Generation

Previous research has tended to use small experimental taxonomies, which usually have a depth of between 1 and 5. Further, they do not take into account the possibility of an item having more than one direct parent, which is feasible given that the taxonomy is a generic connected directed acyclical graph. The scheme which this paper devises allows for a rich diversity of possible experimental taxonomies to be generated using parameters V and E, the number of non-leaf nodes (excluding the empty root) and the number of uniformly distributed random edges connecting those nodes, respectively. Once the the number of non-leaf nodes and random edges have been created, a root node is created to connect all weakly connected components of the graph. Lastly, $|A_L|$ nodes are created (representing the items of the databases) are equally and randomly distributed as children among the nodes of the graph which have no descendants. By adjusting these parameters, a rich diversity of taxonomies can be generated. The smaller the value of V, the fewer internal (generalized) item nodes there will be, and the lower the depth of the taxonomy—if E is kept constant. If V is constant, a larger E results in a more connected (fewer weakly connected components) and possibly deeper graph. V should be limited by the number of leaf nodes, $|A_L|$, because $|A_L|$ leaf nodes will be evenly distributed and connected to the nodes without children in the generated graph. In this paper, we limit V to be less than or equal to $|A_L|$. Once V is fixed, we use E to control the connectedness and depth of the graph generated. Increasing E causes the graph to become more connected and its depth to increase. [3]

4.2 Dataset Generation

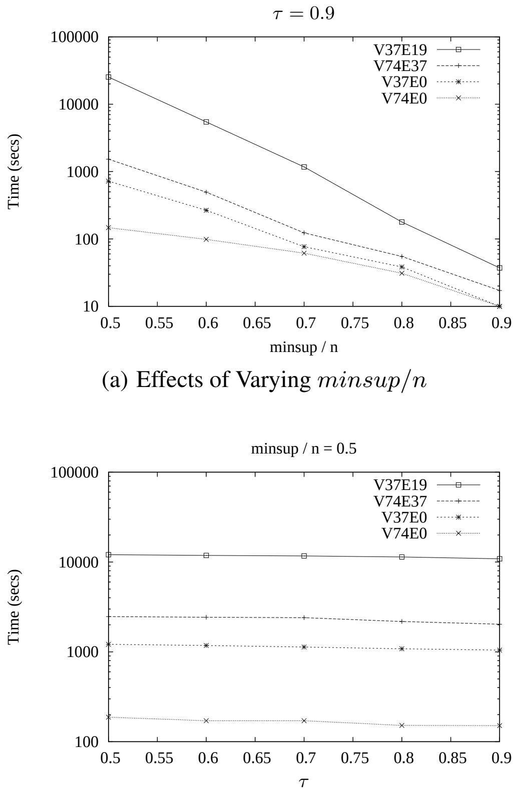
All datasets used in this evaluation were taken from the Frequent Itemset Mining Dataset Repository fimi.ua.ac.be/data/. However, since the datasets are exact or certain, transforming them into uncertain datasets was required. The procedure used to perform this transformation was done as follows: for each certain item in a transaction, the item is copied to the new uncertain dataset; a random probability $p \in (0, 1]$ is then chosen from the beta distribution with parameters $\alpha = 5$ and $\beta = 1$ for this item; finally, p is assigned to the item with probability $1/2$ and $1 - p$ is assigned with probability $1/2$. This method of transforming a certain itemset database into an uncertain one, is different from other methods, in which the probability p is drawn from a uniform distribution [4, 3], or from a normal distribution [8]. We believe drawing probabilities from the beta distribution gives a possibly better representation of a real-world dataset, in which items are close either to existing or not, rather than being uniformly random (uniform distribution), or "ho-hum" average (normal distribution).

4.3 Algorithm Performance

All experiments were carried out on a Intel Core 2 Quad-Core desktop computer, with 4GB of RAM, running Mac OS X v10.6; further, all code was written in C/C++ using Apple's LVM v3.0 compiler. All code and datasets used can be downloaded from www.erichpeterson.com/publications/. Experiments where carried out on both the Chess and Mushroom datasets. All execution times include the time needed to extend the database to include the probability of a generalized item occurring. When the probability of a generalized item occurring within a particular transaction is needed, the database is extended to include the existential probability for all transactions in the database. For each of the datasets, we evaluate the algorithm's performance in time when varying $minsup/n$ and τ. The results for the Chess datasets can be seen in Figure 7 and

[3] Further details can be found in this paper's tech report at www.erichpeterson.com/publications/.

for the Mushroom dataset in Figure 8. For each data point in Figures 7(a) and 8(a), five random taxonomies were generated and the average of the five were taken. In Figure 7(a), one sees time in seconds as a function of $minsup/n$ ($0.5 \leq minsup/n \leq 0.9$) and $\tau = 0.9$—for each of the taxonomies, i.e., V37E19, V74E37, V37E0, and V74E0. The range of parameters tested were chosen to cover a majority (above 50%) of frequency values, and to perform the experiments in a reasonable amount of time. Also, in Figure 7(b), one sees time as a function of τ ($0.5 \leq \tau \leq 0.9$) and $minsup/n = 0.5$—for each of the taxonomies. Notice the effect of τ is virtually nil.



(a) Effects of Varying $minsup/n$

(b) Effects of Varying τ

Figure 7: Chess Dataset

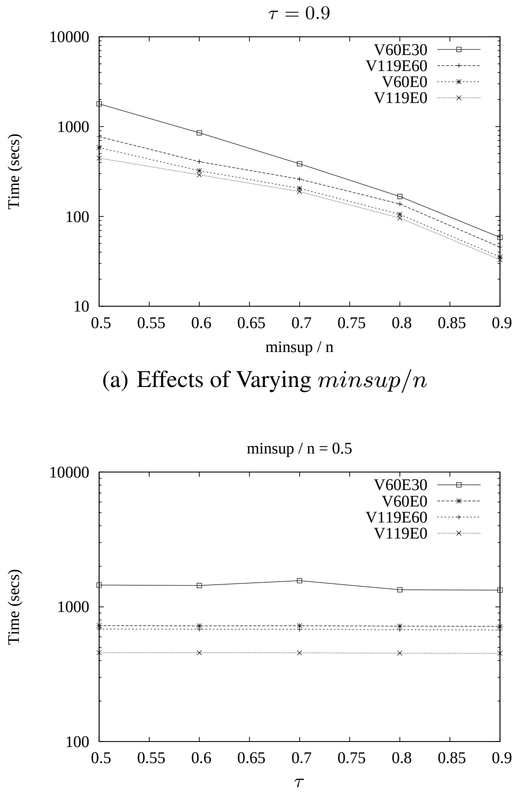
In Figure 8(a) and Figure 8(b) one sees the effect of varying $minsup/n$ and τ, respectively, for the taxonomies V37E19, V74E37, V37E0, and V74E0.

5. CONCLUSION

In this paper, we have disseminated the new concept of a *probabilistic generalized frequent itemset* (PGFI)—rooted in probabilistic mathematics and possible world semantics. Further, an algorithm to mine for such concepts was presented. Lastly, an experimental evaluation of the new algorithm—named PGFIM—was shown.

6. REFERENCES

[1] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *Proc. ACM SIGMOD Int. Conf. on Management of Data*, June 1993.



(a) Effects of Varying $minsup/n$

(b) Effects of Varying τ

Figure 8: Mushroom Dataset

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. 20th Int. Conf. Very Large Data Bases*, Jan. 1994.

[3] T. Bernecker, H.-P. Kriegel, M. Renz, and F. Verhein. Probabilistic Frequent Pattern Growth for Itemset Mining in Uncertain Databases (Technical Report). *arXiv.org*, cs.DB, Aug. 2010.

[4] T. Bernecker, H.-P. Kriegel, M. Renz, F. Verhein, and A. Zuefle. Probabilistic frequent itemset mining in uncertain databases. *Proc. 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 119–127, June 2009.

[5] H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. *Knowledge Discovery in Databases*, pages 181–192, 1994.

[6] R. Srikant and R. Agrawal. Mining Generalized Association Rules. *Research Report*, 1995.

[7] K. Sriphaew and T. Theeramunkong. A New Method for Finding Generalized Frequent Itemsets in Generalized Association Rule Mining. *Proc. 7th Int. Symposium on Computers and Communications*, July 2002.

[8] L. Wang, R. Cheng, S. D. Lee, and D. Cheung. Accelerating probabilistic frequent itemset mining: a model-based approach. In *Proc. 19th ACM Int. Conf. on Information and Knowledge Management*, pages 429–438, Oct. 2010.