Mining Probabilistic Association Rules from Uncertain Databases with Pruning

Erich A. Peterson
Department of Computer Science
University of Arkansas at Little Rock
Little Rock, AR 72204
Email: contact@erichpeterson.com

Liang Zhang
Department of Biological Statistics
and Computational Biology
Cornell University
Ithaca, NY 14850
Email: lz278@cornell.edu

Peiyi Tang
Department of Computer Science
University of Arkansas at Little Rock
Little Rock, AR 72204
Email: pxtang@ualr.edu

***Abstract*—In this paper, we rigorously define the problem of mining probabilistic association rules from uncertain databases. We further analyze the probability distribution space of a candidate probabilistic association rule, and propose an efficient mining algorithm with pruning to find all probabilistic association rules from uncertain databases.**

I. Introduction

In recent years, a great deal of interest in mining probabilistic frequent itemsets (p-FIs) from uncertain databases has been generated [1]–[6], [8]. Since the landmark work of Bernecker et al. [3], almost all the research in mining p-FIs uses a probabilistic model based on possible world semantics. While frequent itemsets are useful, the most common purpose for mining frequent itemsets is then to find association rules from them. While generating association rules from frequent itemsets is straightforward for certain databases [7], to find probabilistic association rules from probabilistic frequent itemsets is not trivial.

In this paper, we give a rigorous definition of a probabilistic association rule (p-AR), based on possible world semantics, using two ratios: (1) a frequency ratio ξ and (2) a confidence ratio η, both between 0 and 1. Using two ratios instead of one (confidence ratio), as in [8], we found that if $\eta < \xi$, $X \Rightarrow Y$ is always a probabilistic association rule with respect to a minimum probability threshold `minprob`, given that $X \cup Y$ is a probabilistic frequent itemset with respect to ξ and `minprob`. This observation offers the means of an extremely useful pruning technique when performing probabilistic association rule mining. In other words, as long as $\eta < \xi$ and Z is a probabilistic frequent itemset with respect to ξ, $X \Rightarrow Z - X$ is guaranteed to be a probabilistic association rule for any $X \subset Z$.

We also derive the recursive equation for the probability distribution of a p-AR and show a dynamic programming algorithm for mining p-ARs. We further introduce additional pruning techniques in the mining algorithm based on the value of η which reduces the computation required to determine whether a candidate is a probabilistic association rule.

The rest of the paper is organized as follows: Section II introduces the uncertain database model used in this paper, probabilistic support based on possible world semantics, and the probabilistic frequent itemset. This section also includes our definition of a probabilistic association rule using two ratios. Section III analyzes the probability distribution of association rules and presents a theorem of pruning when $\eta < \xi$. Section IV presents the dynamic programming algorithm to determine whether a candidate is a probabilistic association rule and an algorithm for enumerating candidates, both with pruning. Section V describes the results of our preliminary evaluation of our mining algorithm. Section VI provides a discussion of related and future work. Section VII concludes the paper.

II. Preliminaries

A. Uncertain Database Model

An *uncertain database* of size n denoted as T is a set of n transactions, $T = \{t_1, \ldots, t_n\}$. A transaction t_j $(1 \leq j \leq n)$ is a set of items, each of which has a non-zero probability $0 < p \leq 1$ that it exists in the transaction t_j. That is, $p = Pr(x \in t_j)$ for any item x contained in t_j $(x \in t_j)$. The items in a transaction with an existential probability between 0 and 1 are called *uncertain items*, whereas the itemset with existential probability 1 are called *certain items*.

When defining probabilistic frequent itemsets and association rules from uncertain databases, the principles and theories of possible world semantics must be used. Under *possible world semantics*, a *possible world* is a certain database instantiated from the uncertain database by including or not including every uncertain item. If the assumption of independence between the transactions in T , as well as the items in each transaction holds, the probability of each possible world w can be calculated as follows:

$$Pr(w) = \prod_{t \in T(w)} \left(\prod_{x \in t} Pr(x \in t') \cdot \prod_{x \notin t} (1 - Pr(x \in t'))\right) \quad (1)$$

where $T(w)$ is the database containing all certain transactions of possible world w, t is a transaction in $T(w)$, t' is the corresponding transaction in the uncertain database T from which t is instantiated, and x is an uncertain item of t'.

In the traditional certain database model, an itemset is either contained in a transaction or not. In the uncertain database model, we cannot say itemset X is contained in transaction t_j with certainty if X contains at least one uncertain item, even though $X \subseteq t_j$. What we have is a probability that X is contained in transaction t_j, $Pr(X \subseteq t_j)$. According to possible world semantics, this is the marginal probability that X is contained in t_j as follows

$$Pr(X \subseteq t_j) = \sum_{w \in W, X \subseteq t_j(w)} Pr(w) \quad (2)$$

where W is the set of all possible worlds and $t_j(w)$ is the j-th transaction in the certain database of possible world w.

From (1) and (2), this marginal probability can be expressed as follows [3]:

$$Pr(X \subseteq t_j) = \prod_{x \in X} Pr(x \in t_j) \quad (3)$$

The probability that t_j does not contain X thus is

$$Pr(X \not\subseteq t_j) = 1 - Pr(X \subseteq t_j) \quad (4)$$

B. Probabilistic Support and Frequent Itemsets

Given an uncertain database of size n, T, and an itemset X, the *probabilistic support* of X in T, denoted as $Sup(X,T)$, is a random variable that can have values $0, 1, \ldots, n$. According to the possible world semantics, the probability that $Sup(X,T) = i$, $0 \leq i \leq n$, denoted as $p_i(X)$, is the marginal probability

$$p_i(X) = \sum_{w \in W, Sup(X,T(w))=i} Pr(w) \quad (5)$$

where $Sup(X, T(w))$ is the support of X in the certain database $T(w)$ of possible world w, which is the number of transactions in $T(w)$ that contain X. $p_i(X)$ for $i = 0, \ldots, n$ is the probability mass function (pmf) of the support of X in T and we have $\sum_{i=0}^{n} p_i(X) = 1$.

Let S be a set of i transactions from T. The probability that every transaction in S contains itemset X and every transaction in $T - S$ does not contain X is

$$\prod_{t \in S} Pr(X \subseteq t) \prod_{t \in T-S} (1 - Pr(X \subseteq t))$$

This is obtained by summing the probability of each possible world in which every transaction in S contains itemset X and every transaction in $T - S$ does not contain X, using (1), (3) and (4). Since there are many different subsets $S \subseteq T$ with size i, the probability of $Sup(X,T) = i$, $p_i(X)$, is the summation of the probabilities above over different subsets S [3]:

$$p_i(X) = \sum_{\substack{S \subseteq T, \\ \|S\| = i}} \prod_{t \in S} Pr(X \subseteq t) \prod_{t \in T-S} (1 - Pr(X \subseteq t)) \quad (6)$$

Given a *minimum support ratio* $\xi \in (0,1)$, an itemset X is a *probabilistic frequent itemset* (*p-FI*) if the probability that $Sup(X,T)$ is greater than or equal to $\xi\|T\|$ (i.e. ξn) is above a *minimum probability threshold* denoted as `minprob`. Since $p_i(X)$ is the *pmf* of $Sup(X,T)$, the probability that $Sup(X,T)$ is greater than or equal to ξn is $\sum_{i=\lceil \xi n \rceil}^{n} p_i(X)$. Hence, we say that X is a frequent itemset if and only if $\sum_{i=\lceil \xi n \rceil}^{n} p_i(X) \geq$ `minprob`. Computing the pmf $p_i(X)$ using (6) is expensive. Bernecker et al. proposed a dynamic programming algorithm to calculate $p_i(X)$ to mine probabilistic frequent itemsets efficiently in [3].

C. Probabilistic Association Rules (p-ARs)

Given an uncertain database T of size n, a *minimum frequency* $\xi \in (0,1)$, a *minimum confidence* $\eta \in (0,1)$, and a minimum probability threshold `minprob`, $X \Rightarrow Y$ is *a probabilistic association rule* (p-AR) if and only if $X \cap Y = \emptyset$, and the probability that $Sup(X \cup Y, T) \geq \xi n$ and $Sup(X \cup Y, T) \geq \eta Sup(X,T)$ is greater or equal to `minprob`.

According to possible worlds semantics, the probability that $Sup(X \cup Y, T) \geq \xi n$ and $Sup(X \cup Y, T) \geq \eta Sup(X,T)$, denoted as $P(X, Y, \xi, \eta, T)$, is the sum of each possible world w's probability of occurring satisfying $Sup(X \cup Y, T(w)) \geq \xi n$ and $Sup(X \cup Y, T(w)) \geq \eta Sup(X, T(w))$, where $T(w)$ is the certain database of small world w. That is,

$$P(X, Y, \xi, \eta, T) = \sum_{\substack{w \in W, \\ Sup(X \cup Y, T(w)) \geq \xi n, \\ Sup(X \cup Y, T(w)) \geq \eta Sup(X, T(w))}} Pr(w) \quad (7)$$

The problem of mining probabilistic association rules, is to find all the probabilistic association rules from an uncertain base T, given the thresholds: minimum frequency ξ, minimum confidence η, and minimum probability `minprob`.

III. Probability Distribution for Association Rules

Calculating $P(X, Y, \xi, \eta, T)$ using (7) is infeasible, as the number of possible worlds are too numerous. Thus, we need to find another way to calculate it.

Let us consider the probability that a transaction t_j contains both X and Y:

$$\begin{aligned} Pr(X \subseteq t_j \wedge Y \subseteq t_j) &= Pr(X \cup Y \subseteq t_j) \\ &= \prod_{z \in X \cup Y} Pr(z \in t_j) \\ &= \prod_{x \in X} Pr(x \in t_j) \prod_{y \in Y} Pr(y \in t_j) \\ &= Pr(X \subseteq t_j) Pr(Y \subseteq t_j) \end{aligned} \quad (8)$$

The third equality above is due to $X \cap Y = \emptyset$ and the second and the fourth follow (3).

The probability that a transaction t_j contains X and but not Y is, thus,

$$Pr(X \subseteq t_j \wedge Y \not\subseteq t_j) = Pr(X \subseteq t)(1 - Pr(Y \subseteq t)) \quad (9)$$

because $Pr(X \subseteq t_j \wedge Y \not\subseteq t_j) + Pr(X \subseteq t_j \wedge Y \subseteq t_j) = Pr(X \subseteq t_j)$.

Consider the possible worlds w where j transactions contain both X and Y and another i transactions contain X but not Y. These $j + i$ transactions are the transactions containing X and the rest $n-(j+i)$ transactions do not contain X. Thus, the support of X and $X \cup Y$ in theses possible worlds are $j+i$ and j, respectively. Let S_1 and S_2 be the sets of the transactions that contain X but not Y, and X and Y, respectively and we have $\|S_1\| = i$ and $\|S_2\| = j$. Let S_1 and S_2 also denote the same sets of transactions in the uncertain database. Then, the probability that the i transactions of S_1 contain X but not Y,

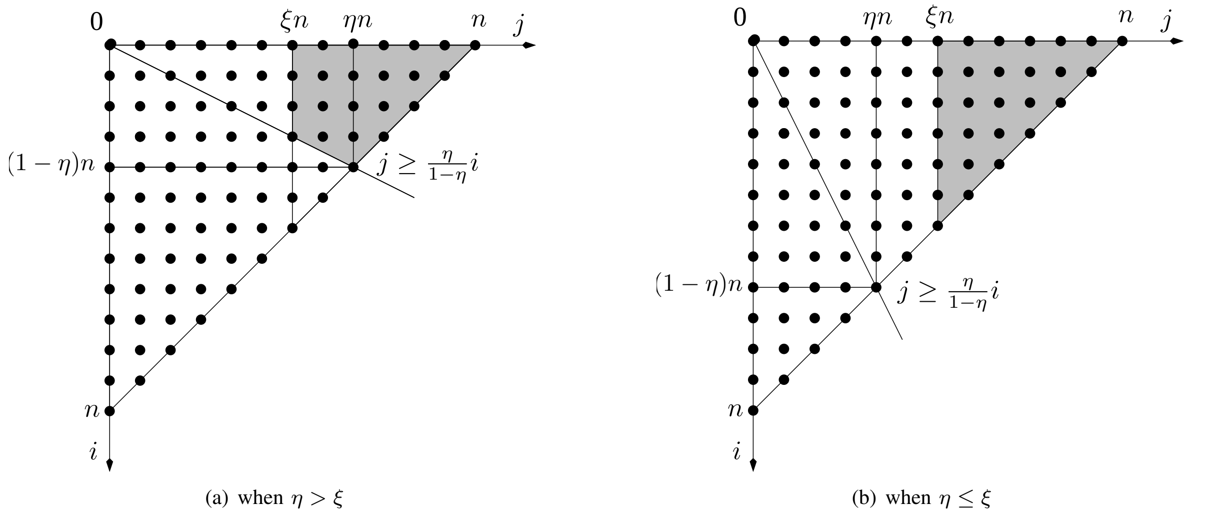

Fig. 1. Probability Distribution of $p_{i,j}(X, Y)$

the j transactions of S_2 contains both X and Y, and the rest of $n-(j+i)$ transactions do not contain X is as follows:

$$\prod_{t \in S_1} Pr(X \subseteq t)(1 - Pr(Y \subseteq t)) \cdot \prod_{t \in S_2} Pr(X \subseteq t)Pr(Y \subseteq t) \cdot \prod_{t \in T - S_s - S_1} (1 - Pr(X \subseteq t))$$

according to (9), (8) and (4).

Further, the probability that there are j transactions in T containing both X and Y and i transactions containing X but not Y, or equivalently the $Sup(X \cup Y, T) = j$ and $Sup(X, T) - Sup(X \cup Y, T) = i$, denoted as $p_{i,j}(X, Y)$, is given as follows:

$$p_{i,j}(X,Y) = \sum_{\substack{S_1 \subseteq T, \\ \|S_1\| = i, \\ S_2 \subseteq T, \\ \|S_2\| = j}} \prod_{t \in S_1} Pr(X \subseteq t)(1 - Pr(Y \subseteq t)) \cdot \prod_{t \in S_2} Pr(X \subseteq t)Pr(Y \subseteq t) \cdot \prod_{t \in T - S_s - S_1} (1 - Pr(X \subseteq t)) \quad (10)$$

Obviously, $j + i \leq n$ and $0 \leq i, j \leq n$. $p_{i,j}(X, Y)$ is actually the pmf of the number of transactions containing X but not Y, and both X and Y, or equivalently $Sup(X \cup Y, T) = j$ and $Sup(X, T) = j + i$. Also, we have

$$\sum_{0 \leq i,j \leq n, j+i \leq n} p_{i,j}(X, Y) = 1.$$

Therefore, the probability that $Sup(X \cup Y, T) \geq \xi n$ and $Sup(X \cup Y, T) \geq \eta Sup(X, T)$, (denoted as $P(X, Y, \xi, \eta, T)$) is the sum of those $p_{i,j}(X, Y)$ satisfying both $j \geq \xi n$ and $j \geq \eta(j + i)$. Also, $j \geq \eta(j + i)$ is equivalent to $j \geq \frac{\eta}{1-\eta} i$. Thus, we have

$$P(X, Y, \xi, \eta, T) = \sum_{\substack{\xi n \leq j \leq n, j + i \leq n \\ j \geq \frac{\eta}{1-\eta} i, i \geq 0}} p_{i,j}(X, Y) \quad (11)$$

Figure 1(a) shows the points where $p_{i,j}(X, Y)$ is defined. The shaded area is the intersection of the two triangles (1) $j \geq \xi n, i \geq 0$ and $j + i \leq n$ and (2) $j \geq \frac{\eta}{1-\eta} i, i \geq 0$ and $j + i \leq n$. The summation of the $p_{i,j}(X, Y)$ in the shaded area gives $P(X, Y, \xi, \eta, T)$ as shown in (11). $X \Rightarrow Y$ is a probabilistic association rule if and only if this $P(X, Y, \xi, \eta, T)$ is greater or equal to threshold `minprob`.

Notice that the intersection of lines $j \leq \frac{\eta}{1-\eta} i$ and $j + i = n$ is $(i, j) = ((1 - \eta)n, \eta n)$. Thus, when $\eta \leq \xi$, the intersection of the two triangles is the first triangle $j \geq \xi n, i \geq 0$ and $j + i \leq n$ itself as illustrated in Figure 1(b). The summation of the $p_{i,j}(X, Y)$ in this triangle is the probability that $j = Sup(X \cup Y, T) \geq \xi n$. If $X \cup Y$ is a p-FI with respect to ξ defined in Section II-B and $\eta \leq \xi$, the summation of $p_{i,j}(X, Y)$ in this triangle is greater than or equal to `minprob`. Thus, $P(X, Y, \xi, \eta, T) \geq$ `minprob` and $X \Rightarrow Y$ is a p-AR with respect to ξ and η. We summarize this finding in the following theorem.

Theorem 1: If itemset Z is a probabilistic frequent itemset with respect to minimum frequency ξ and minimum probability threshold `minprob` and minimum confidence η is less than or equal to ξ, $X \Rightarrow Y$ is a probabilistic association rule with respect to ξ, η and `minprob` for any $X \subset Z$ and $Y = Z - X$.

Proof: If $Z = X \cup Y$ is a probabilistic frequent itemset with respect to ξ and `minprob`, then the probability that $Sup(X \cup Y, T) \geq \xi n$, where $n = \|T\|$, is greater than or equal to `minprob`. The probability that $Sup(X \cup Y, T) \geq \xi n$ is equal to $\sum_{\substack{i \geq 0, j+i \leq n, \\ \xi n \leq j \leq n}} p_{i,j}(X, Y)$. We thus have

$$\sum_{\substack{i \geq 0, j+i \leq n, \\ \xi n \leq j \leq n}} p_{i,j}(X, Y) \geq \texttt{minprob}.$$

We now show that $j \geq \xi n$, $j + i \leq n$ and $\xi \geq \eta$ implies $j \geq \frac{\eta}{1-\eta} i$. First we have $i \leq n - j \leq n - \xi n = (1 - \xi)n$.

Since $j \geq \xi n$ and $\xi \geq \eta$, we have

$$\frac{j}{i} \geq \frac{\xi n}{(1-\xi)n} = \frac{\xi}{1-\xi} \geq \frac{\eta}{1-\eta}$$

which gives $j \geq \frac{\eta}{1-\eta} i$. Since $j \geq \xi n$, $j+i \leq n$ and $\xi \geq \eta$ implies $j \geq \frac{\eta}{1-\eta} i$, we have

$$\sum_{\substack{i \geq 0, j+i \leq n, \\ \xi n \leq j \leq n, \\ j \geq \frac{\eta}{1-\eta} i}} p_{i,j}(X,Y) = \sum_{\substack{i \geq 0, j+i \leq n, \\ \xi n \leq j \leq n}} p_{i,j}(X,Y) \geq \texttt{minprob}.$$

Thus, $X \Rightarrow Y$ is a probabilistic association rule with respect to ξ, η and minprob. ■

Theorem 1 tells us that if $\eta \leq \xi$ and Z is a p-FI, we can simply enumerate all the subset X of Z and every $X \Rightarrow Z-X$ is a p-AR; thus, we need only mine p-ARs when $\eta > \xi$.

IV. Mining Probabilistic Association Rules

A. Dynamic Programming Algorithm

The complexity of computing $p_{i,j}(X,Y)$ directly by (10) is exponential with n. We can use dynamic programming method to reduce it to $O(n^3)$.

Let T_k be an uncertain database made of the first k transactions $\{t_1, \ldots, t_k\}$ from T. That is, $T_0 = \emptyset$ and $T_k = T_{k-1} \cup \{t_k\}$ for $k = 1, \ldots, n$. Let the probability that there are j transactions in T_k containing both X and Y and i transactions containing X but not Y, or equivalently $Sup(X \cup Y, T_k) = j$ and $Sup(X, T_k) - Sup(X \cup Y, T_k) = i$, be denoted as $p_{i,j}^{(k)}(X,Y)$. Obviously, $i+j \leq k$ and $0 \leq i,j \leq k$ in $p_{i,j}^{(k)}(X,Y)$ and we have

$$\sum_{0 \leq i,j \leq k, j+i \leq k} p_{i,j}^{(k)}(X,Y) = 1. \quad (12)$$

That is, all the $p_{i,j}^{(k)}(X,Y)$ outside of the triangle of $0 \leq i,j \leq k, j+i \leq k$ are zeros. Notice that $p_{i,j}^{(n)}(X,Y)$ is the $p_{i,j}(X,Y)$ given in (10).

Since $T_k = T_{k-1} \cup \{t_k\}$, $p_{i,j}^{(k)}(X,Y)$ can be calculated from $p_{i,j}^{(k-1)}(X,Y)$ using the following recursion:

$$\begin{aligned} p_{i,j}^{(k)}(X,Y) &= p_{i-1,j}^{(k-1)}(X,Y) Pr(X \subseteq t_k)(1 - Pr(Y \subseteq t_k)) \\ & \quad p_{i,j-1}^{(k-1)}(X,Y) Pr(X \subseteq t_k) Pr(Y \subseteq t_k) + \\ & \quad p_{i,j}^{(k-1)}(X,Y)(1 - Pr(X \subseteq t_k)) \end{aligned} \quad (13)$$

with $p_{-1,j}^{(k)}(X,Y) = 0$ and $p_{i,-1}^{(k)}(X,Y) = 0$ due to (12). This is because transaction t_k either contains both X and Y, or contains X but not Y, or does not contain X at all. The recursive equation (13) above is derived from the Law of Total Probability and the fact that transaction t_k is independent from all the transactions in T_{k-1}. In particular, when the case is that T_k has j transactions containing both X and Y and i transactions containing X but not Y, only one of the following situations will be true: (1) t_j does not contain X and T_{k-1} has j transactions containing both X and Y and i transactions containing X but not Y, (2) t_j contains both X and Y and T_{k-1} has $j-1$ transactions containing both X and Y and i transactions containing X but not Y, or (3) t_j contains X but not Y and T_{k-1} has j transactions containing both X and Y and $i-1$ transactions containing X but not Y.

Note that the values of $p_{i,j}^{(k-1)}(X,Y)$ are only used to calculate the values of $p_{i,j}^{(k)}(X,Y)$. Once used, they do not need to exist. Thus, we can use two 2-dimensional data arrays $(f_0[i,j])$ and $(f_1[i,j])$ to store values of $p_{i,j}^{(k)}(X,Y)$ for even and odd k, respectively. The following triple-nested loop will calculate and leave $p_{i,j}^{(n)}(X,Y)$ in $f_{n \bmod 2}[i,j]$:

```
Initialize array $f_0$ with $f_0[i,j] = \begin{cases} 1 & \text{if } i=j=0 \\ 0 & \text{otherwise} \end{cases}$ ;
Initialize array $f_1$ with zero for all elements;
for $k = 1, n$
  for $i = 0, k$
    for $j = 0, k-i$
      tmp = $f_{k-1 \bmod 2}[i,j](1 - p_k^X)$;
      if $(i > 0)$ tmp += $f_{(k-1) \bmod 2}[i-1,j] p_k^X (1 - p_k^Y)$;
      if $(j > 0)$ tmp += $f_{(k-1) \bmod 2}[i,j-1] p_k^X p_k^Y$;
      $f_{k \bmod 2}[i,j]$ = tmp;
```

where p_k^X and p_k^Y are the shorthands for $Pr(X \subseteq t_k)$ and $Pr(Y \subseteq t_k)$, respectively. The nested loop above is the dynamic programming method to calculate $p_{i,j}^{(k)}(X,Y)$ for $k = 1, \ldots, n$ using (13). The **if** statements in the loop body are to deal with boundary conditions for $p_{i,j}^{(k)}(X,Y)$, where $i = 0$ or $j = 0$ which require values $p_{-1,j}^{(k-1)}(X,Y)$ and $p_{i,-1}^{(k-1)}(X,Y)$. These values should be all zeros, because neither of i and j can be negative.

Iteration k ($k = 1, \ldots, n$) of the outer-most loop calculates each $p_{i,j}^{(k)}(X,Y)$ and stores them in $f_{k \bmod 2}[i,j]$ in the triangle delineated by $0 \leq i,j \leq k, j+i \leq k$. It uses the $p_{i,j}^{(k-1)}(X,Y)$ values calculated by the previous iteration $k-1$ and stores them in $f_{(k-1) \bmod 2}[i,j]$ in the triangle delineated by $0 \leq i,j \leq k-1, j+i \leq k-1$ (when $k > 1$), as well as the initial values stored in the $f_{(k-1) \bmod 2}[i,j]$ on the diagonal of $j+i = k$. The initial values on the diagonal $j+i = k$ of $f_{(k-1) \bmod 2}[i,j]$ are $p_{i,j}^{(k-1)}(X,Y)$ with $j+i = k > k-1$. They should be zeros, because $j+i$ cannot exceed $k-1$ in $p_{i,j}^{(k-1)}(X,Y)$. (Also see (12)). The initial value of in $f_0[0,0]$ is $p_{0,0}^{(0)}(X,Y)$. According to (12), this value should be 1. Thus, the array f_0 should be initialized as follows

$$f_0[i,j] = \begin{cases} 1 & \text{if } i=j=0 \\ 0 & \text{otherwise} \end{cases}$$

and f_1 should be initialized with all zero.

B. Pruning

To determine whether $X \Rightarrow Y$ is a p-AR with respect to ξ, η ,and minprob (and $\eta > \xi$)—we can either sum values of $p_{i,j}(X,Y)$ in the shaded area of Figure 1(a) and see if the sum is greater than or equal to minprob, or sum the values of $p_{i,j}(X,Y)$ in the un-shaded area to see if the sum is less than or equal to $1 - \texttt{minprob}$. The intersection of the lines $j+i = n$ and $j = \frac{\eta}{1-\eta} i$ is $(i,j) = ((1-\eta)n, \eta n)$ as shown in Figure 1(a). To compute the sum of the points for each $p_{i,j}(X,Y)$ in the shaded area, we only need to calculate the values for each point $p_{i,j}(X,Y)$ in the trapezoid bounded by $i=0$, $i=(1-\eta)n$, $j=0$, and $j=n-i$. If we want to calculate the sum of of the points for each $p_{i,j}(X,Y)$ in the unshaded area, we only need to to calculate the values for each point $p_{i,j}(X,Y)$ in the trapezoid bounded by $j=0$, $j=\eta n$, $i=0$, and $i=n-j$. We want to chose the smaller of the possible trapezoids to speed computation. The areas of trapezoids for computing the shaded and unshaded areas are $\frac{(1-\eta^2)n^2}{2}$ and $\frac{(2\eta-\eta^2)n^2}{2}$, respectively. Thus, if $2\eta \geq 1$, i.e. $\eta \geq 0.5$, we chose to calculate the trapezoid for the shaded area. Otherwise, we calculate the trapezoid for the unshaded area. The dynamic programming algorithm with pruning to determine whether $X \Rightarrow Y$ is a p-AR is shown in Figure 2. It is assumed that function p-AR$(X,Y,\xi,\eta,\texttt{minprob})$ is called only when $X \cap Y = \emptyset$ and $X \cup Y$ is a p-FI with respect to ξ and `minprob`.

C. Finding Association Rules

For $X \Rightarrow Y$ with $X \cap Y = \emptyset$ to be a p-AR with respect to ξ, η and `minprob`, $X \cup Y$ has to be a p-FI with respect to ξ and `minprob`. We use the apriori set enumeration [7] and Bernecker's dynamic programming algorithm [3] to find probabilistic frequent itemsets (p-FIs) Z with respect to ξ and `minprob`.

For each p-FI Z, we use an algorithm to enumerate the candidates X, Y with $X \cup Y = Z$ and $X \cap Y = \emptyset$. Figure 3 shows the enumeration tree of p-AR candidates X, Y for p-FI $Z = abcd$. At each iteration of the enumeration, an item is moved from antecedent X which is less than all the items in consequent Y and add it to Y, to be a child of X, Y. Formally, let $X = x_1 \cdots x_n$ and $Y = y_1 \cdots y_m$ and items in X and Y are all ordered according to the total order of the items, i.e. $x_i < x_j$ and $y_i < y_j$ for $i < j$. Then X, Y has only those children $x_1 \cdots x_{i-1}x_{i+1} \cdots x_n, x_i y_1 \cdots y_m$ such that $x_i < y_1$. Notice that the consequents in each of the descendants of X, Y are supersets of Y. According to the anti-monotonicity principle proved by [8], if $X \Rightarrow Y$ is not a p-AR, none of the descendants of X, Y would be a p-AR and, thus, the entire subtree formed by node X, Y can be pruned out.

V. Performance Evaluation

Here some preliminary tests are disseminated, whose results were culled from an implementation of the p-AR mining algorithm described above.

The uncertain database we used was converted from the certain database `chess` from the Frequent Itemset Mining Dataset Repository fimi.ua.ac.be/data/. The transformation of the `chess` database was performed as follows: for each item in each transaction, we assign a random probability p between 0 and 1 drawn from Beta distribution with $\alpha = 10$ and $\beta = 1$, to be used as the item's existential probability. This procedure was run five times to produce five random datasets. Each experiment (data point in the performance evaluation) was run on all five datasets and the results of each averaged. To reduce the amount of time needed to run each experiment, the total number of unique items was reduced to 14, only the first 100 transactions were used, and only rules $X \Rightarrow Y$ of

```
boolean p-AR($X, Y, \xi, \eta, \texttt{minprob}$)
{
if ($\eta \leq \xi$) return true;
Initialize array $f_0$ with $f_0[i,j] = \begin{cases} 1 & \text{if } i=j=0 \\ 0 & \text{otherwise} \end{cases}$ ;
Initialize array $f_1$ with zero for all elements;
float sum = 0;
if ($\eta \geq 0.5$)
  for $k = 1, n$
    for $i = 0, \min(k, \lfloor (1-\eta)n \rfloor)$
      for $j = 0, k - i$
        tmp = $f_{k-1 \bmod 2}[i,j](1 - p_k^X)$;
        if ($i > 0$) tmp =+ $f_{(k-1) \bmod 2}[i-1,j]p_k^X(1-p_k^Y)$;
        if ($j > 0$) tmp =+ $f_{(k-1) \bmod 2}[i,j-1]p_k^X p_k^Y$;
        $f_{k \bmod 2}[i,j]$ = tmp;
        if ($k = n \wedge j \geq \lceil \eta n \rceil \wedge j \geq \lceil \frac{\eta}{1-\eta} i \rceil$)
          sum += $f_{k \bmod 2}[i,j]$;
          if (sum $\geq$ minprob) return true;
        end if
      end for
    end for
  end for
  return false;
else
  for $k = 1, n$
    for $i = 0, n$
      for $j = 0, \min(k - i, \lfloor \eta n \rfloor)$
        tmp = $f_{k-1 \bmod 2}[i,j](1 - p_k^X)$;
        if ($i > 0$) tmp =+ $f_{(k-1) \bmod 2}[i-1,j]p_k^X(1-p_k^Y)$;
        if ($j > 0$) tmp =+ $f_{(k-1) \bmod 2}[i,j-1]p_k^X p_k^Y$;
        $f_{k \bmod 2}[i,j]$ = tmp;
        if ($k = n \wedge (j < \lfloor \eta n \rfloor \vee j < \lfloor \frac{\eta}{1-\eta} i \rfloor)$)
          sum += $f_{k \bmod 2}[i,j]$;
          if (sum $> 1 -$ minprob) return false;
        end if
      end for
    end for
  end for
  return true;
end if
```

Fig. 2. Dynamic Programming Algorithm

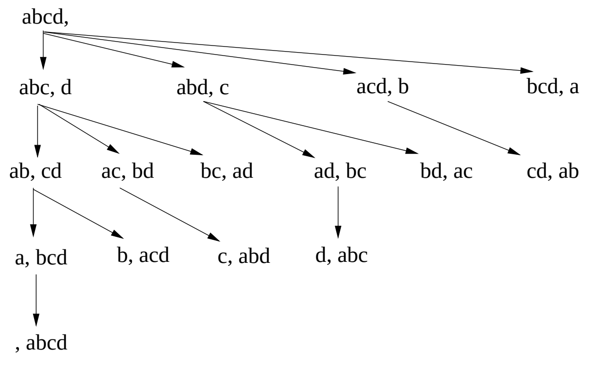

Fig. 3. Enumeration of AR Candidates

TABLE I. AVERAGE TIME IN P-AR() (MS)

	ξ								
η	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	0.0000402	0.0000352	0.0000317	0.0000353	0.0000373	0.0000000	0.0000000	0.0000000	0.0000000
0.15	3.821358	0.0000311	0.0000346	0.0000228	0.0000208	0.0001322	0.0000000	0.0000000	0.0000000
0.2	4.82638	0.0000273	0.0000349	0.0000337	0.0000101	0.0000693	0.0000000	0.0000000	0.0000000
0.25	5.710972	5.706706	0.0000331	0.0000374	0.0000374	0.0000000	0.0000000	0.0000000	0.0000000
0.3	6.513988	6.571548	0.0000304	0.0000254	0.0000517	0.0002080	0.0000000	0.0000000	0.0000000
0.35	7.338718	7.369756	7.40545	0.0000293	0.0000402	0.0000000	0.0000000	0.0000000	0.0000000
0.4	8.107516	8.129858	8.151792	0.0000373	0.0000423	0.0000691	0.0000000	0.0000000	0.0000000
0.45	8.777348	8.816462	8.831548	8.914608	0.0000165	0.0000691	0.0000000	0.0000000	0.0000000
0.5	2.102258	2.124626	2.139828	2.181686	0.0000379	0.0000000	0.0000000	0.0000000	0.0000000
0.55	1.941636	1.972038	1.988556	2.030988	2.075702	0.0000000	0.0000000	0.0000000	0.0000000
0.6	1.827738	1.869998	1.87882	1.92489	1.970294	0.0000694	0.0000000	0.0000000	0.0000000
0.65	1.683634	1.707508	1.73654	1.783594	1.821012	1.856634	0.0000000	0.0000000	0.0000000
0.7	1.532532	1.546806	1.572222	1.61223	1.670418	1.695872	0.302456	0.0000000	0.0000000
0.75	1.361404	1.377852	1.387406	1.432224	1.472064	1.495822	1.464326	0.0000000	0.0000000
0.8	1.12049	1.137	1.142428	1.171142	1.215424	1.244568	1.2089	0.0000000	0.0000000
0.85	0.9382024	0.9464922	0.9661912	0.9943356	1.012592	1.047416	0.9889136	0.877778	0.0000000
0.9	0.6398418	0.644898	0.6557288	0.6697286	0.6936324	0.7244038	0.6548648	0.5444444	0.0000000
0.95	0.4179482	0.4182372	0.4276158	0.4408202	0.4587366	0.4733338	0.4721832	0.4111112	0.0000000

total length 6 (length of X plus the length of Y) or less were considered as possible p-ARs. The aforementioned restrictions were necessary because mining probabilistic association rules is extremely time-expensive: to find all probabilistic frequent itemsets is already exponential. Then, to enumerate allocation rule candidates for each frequent itemset as shown in Figure 3 is exponential with respect to the length of the frequent itemset.

Bernecker et al.'s dynamic programming algorithm [3] is used to find probabilistic frequent itemset and for each probabilistic frequent itemset found, we follow the enumeration in Figure 3 with apriori pruning and call function p-AR() in Figure 2 to determine whether a candidate is a p-AR. We measured two times in our analysis: (1) the total time of mining all probabilistic association rules and (2) the average time of running function p-AR() in Figure 2. The tests are run on a Dell Precision T7500 machine with Intel Xeon CPU (E5620) 2.4 GHz w/ 4 Cores and 48GB RAM. Again, each experiment was the average of the results culled from running each of the five random datasets previously mentioned.

In all experiments we fix $\texttt{minprob} = 0.9$, and vary ξ (from 0.1 to 0.9 in 0.1 increments) and η (from 0.1 to 0.95 in 0.05 increments). In Table I the average time in milliseconds needed to determine whether a candidate association rule is an uncertain association rule (i.e., the average time taken to execute the function p-AR()) for various values of η and ξ. In the table, one sees the drastic effect of pruning according to the value of η with respect to ξ. In particular, one sees that when $\eta < \xi$, that the time is virtually zero, as the function p-AR() simply returns `true`. One also sees the effect of the differing areas of the trapezoid being computed when $\eta > \xi$. For $\xi = 0.1, 0.2, 0.3$ and 0.4, times all peeked at $\eta = 0.45$. We see that the time is decreasing as η decreases from 0.45 to ξ. This is because the trapezoid covering the unshaded area, which is computed when $\eta < 0.5$, is decreasing as η decreases. The time is also decreasing as η increases from 0.5 to 0.95. This is because the trapezoid covering the shaded area, which is computed when $\eta \geq 0.5$, is decreasing as η increases. We also see the time is decreasing as η increases from ξ to 0.95 for $\xi = 0.5, \ldots, 0.9$ for the same reason.

Table II shows the total average time of mining probabilistic frequent itemsets (p-FIs) *and* probabilistic association rules

TABLE II. AVERAGE TOTAL EXECUTION TIME (SECS)

	ξ								
η	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	16.2	12.2	10	4	1	0	0	0	0
0.15	2332.8	12.2	10	4	1	0	0	0	0
0.2	2813.6	12	10	4	1	0	0	0	0
0.25	3024.2	3006	10	4	1	0	0	0	0
0.3	2525	2529	10	4	1	0	0	0	0
0.35	1847.2	1835	1806.8	4	1	0	0	0	0
0.4	1338.6	1320.8	1283.6	4	1	0	0	0	0
0.45	674.4	653.8	611.4	546.4	1	0	0	0	0
0.5	135.4	128.4	117	97.6	1	0	0	0	0
0.55	75.8	68.6	57.8	38.4	25	0	0	0	0
0.6	60	53	42.6	23.6	10.6	0	0	0	0
0.65	53.8	46.8	37.2	19.2	6.8	2.2	0	0	0
0.7	49.8	43	34.4	17.4	6	1.6	0	0	0
0.75	45.8	39.6	31	15.6	5	1	0	0	0
0.8	40.8	34.6	27.4	13.4	4	1	0	0	0
0.85	36.2	31	24.6	12	4	1	0	0	0
0.9	29.8	24.8	19.8	9.2	3	0	0	0	0
0.95	24.8	20	16.4	7.4	2	0	0	0	0

(p-ARs) in seconds. One can see that when η is held constant, the total mining time decreases over increasing values of ξ, because for lower values of ξ far more p-FIs are generated. When ξ is held constant, we see again that for values of $\xi > \eta$, the total time is small because of the near zero time needed to return `true` from p-AR(); however, we see that as η increases past ξ, total time decreases as the number of candidate p-ARs decreases. For example, for $\xi = 0.1$, the total mining time decreases as η increases from 0.25 to 0.95. This is because with a higher η fewer candidate p-ARs are generated. The low total time for $\eta = 0.1$ is because the running time of function p-AR() is very near zero. Similarly for $\xi = 0.4$, we see that the total time decreases as η increases from 0.45 to 0.95. The total times for $0.1 \leq \eta \leq 0.45$ are low; again, because of the near zero time needed to return true from p-AR().

VI. RELATED AND FUTURE WORK

Most work in mining uncertain data based on possible world semantics are centered around finding probabilistic frequent itemsets [3], [6], [8] or probabilistic frequent closed itemsets [4], [5]. In [8], Sun et al. presented an algorithm for mining probabilistic association rules. The algorithm disseminated in [8], uses parameter `minsup`, an integer between 0 and the size of database n for the minimum support, and `minconf` (η in our paper), a real number between 0 and 1 for the minimum confidence, to define a probabilistic association rule, whereas we use two ratios ξ (minimum frequency) and η (minimum confidence), between 0 and 1. As a consequence, we are able to find the mathematical relationship between the two parameters (Theorem 1) by analyzing the probability distribution space, and use it to prune the p-AR search space significantly when $\eta \leq \xi$. We further prune the search space by choosing the smaller area of the probability distribution space depending on the value of η, whereas [8] always calculates the probability distribution in the shaded area no matter how small η is.

The association rule probability distribution in [8] is calculated from an itemset's support probability distribution, using convolution through a Fast Fourier Transformation. In this paper, we use a two-dimensional dynamic programming approach to calculate a candidate association rule's probability distribution directly. The theoretical time complexity of the algorithm in [8] is $O(n^2 \log n)$, where ours is $O(n^3)$ and $O(1)$, when $\eta > \xi$ and $\eta \leq \xi$, respectively. Even when $\eta > \xi$, since our algorithm has deep pruning with η close to 1 or 0, it is our belief that our algorithm will perform well, particularly when η is close to 1 or 0. One future work, is to compare the performance between the two algorithms using practical experiments—using databases of different sizes and using varying thresholds.

VII. Conclusion

In this paper, we defined probabilistic association rules (p-ARs) using two minimum ratios, minimum frequency ξ and minimum confidence η. We throughly analyzed the probability distribution space for a candidate association rule, and proposed an efficient dynamic programming algorithm with pruning for mining probabilistic association rules.

References

[1] CK Chui, B Kao, and E Hung. Mining frequent itemsets from uncertain data. *Advances in Knowledge Discovery and Data Mining*, pages 47–58, 2007.

[2] C Chui and B Kao. A decremental approach for mining frequent itemsets from uncertain data. *PAKDD'08, Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining*, pages 64–75, Jan 2008.

[3] Thomas Bernecker, Hans-Peter Kriegel, Matthias Renz, Florian Verhein, and Andreas Zuefle. Probabilistic frequent itemset mining in uncertain databases. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 119–128, New York, NY, USA, 2009. ACM.

[4] Peiyi Tang and Erich A. Peterson. Mining probabilistic frequent closed itemsets in uncertain databases. In *Proceedings of the 49-th Annual Association for Computing Machinery Southeast Conference (ACMSE'11)*, pages 86–91, Kennesaw, GA, USA, March 2011.

[5] Yongxin Tong, Lei Chen, and Bolin Ding. Discovering threshold-based frequent closed itemsets over probabilistic data. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 270–281, 2012.

[6] Liang Wang, D.W.-L. Cheung, R. Cheng, Sau-Dan Lee, and X.S. Yang. Efficient mining of frequent item sets on large uncertain databases. *Knowledge and Data Engineering, IEEE Transactions on*, 24(12):2170–2183, 2012.

[7] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th Internatioanl Conference on Very Large Data Bases*, pages 487–499, 1994.

[8] Liwen Sun, Reynold Cheng, David W. Cheung, and Jiefeng Cheng. Mining uncertain data with probabilistic guarantees. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 273–282, New York, NY, USA, 2010. ACM.