以派翠網路為基礎的資料摘要
SUMMARIZING DATA COLLECTION SUPPORT USING FUZZY PETRI NET

謝明興
真理大學資訊管理學系
shiehms@ms21.hinet.net

黃詠淮　林栩傑
真理大學資訊管理學系
yonghuai@ms28.hinet.net

曾定章
中央大學資訊工程研究所
tsengdc@ip.csie.ncu.edu.tw

Abstract — With the growth of electronic commerce (EC) on the Internet and the increase of interests in software agents, researches for the automated summarizing collections are proceeding rapidly. A common view about mining is that it is an exercise of clustering customers, markets, products, and other objects of interest in useful ways from large amounts of data. Database summarization is a task that reduces a large number of actual database tuples into a relatively small number of generalized descriptions. Summary discovery provides the user with comprehensive information for grasping the essence of a focused portion from a large amount of information in a database. In this paper, we propose a summary discovery process based on the fuzzy Petri net model to form more generalized tuples. We present a general type of fuzzy Petri net as a representation form of a database summary including fuzzy concepts. By virtue of fuzzy Petri net model where fuzzy set hierarchies in the actual domain are natually expressed, the discovery process yields more accurate database summaries. The proposed method is more efficient due to the concurrent process of summary discovery technique.

Key Words—Data mining, fuzzy Petri net, fuzzy set hierarchy, summary discovery.

1. INTRODUCTION

EC is rapidly growing area on the Internet. A customer can contact with many vendors worldwide even to buy pieces of products owing to the reduced search and transaction costs than before. But the explosive increase and the heterogeneous and distributed state of product or service information made difficult not only for user to search but for mail-order house to provide. Nowadays, mining of database has grown more attentive in database communities due to its wide applicability in retail business to improving marketing strategy. As pointed out in [9], the progress in bar-code technology has made it possible for retail organizations to collect and store massive amounts of sales data. A record in such data typically consists of the transaction date and the items bought in the transaction. It is noted that analysis of past transaction data can provide valuable information on customer buying behavior, and thus improve the quality of business decisions (such as what to put on sale, which merchandises to be placed on shelves together, how to customize marketing programs, to name a few).

One of the most important data-mining problems is database summarization problem. A database summary is one of the major important types of knowledge to be discovered. Specifically, give a database of sales transaction, a market manager would like to discover all summaries among items such that the presence of some items that is belonged to one group in a transaction will imply the presence of some item of another group in the same transaction.

Definitely, a task of database summarization is to reduce a large number of actual database tuples into a relatively small amount of generalized descriptions. Take a transaction table as an example, a transaction table with attribute scheme (*SaleItem, Customer*) might contains thousands of transaction tuples such as <fresh milk, Peter> and <guava juice, Jane>. To give a more general description about classes of customer, these shopping records could be reduced into a more generalized tuples like *<Juice, Marketer>*. This delivers an assertion that *Marketers* have shopped some kinds of *juice* during the past period.

Among several requirements for effective summary techniques, we concentrate on the following ones. First, it must be possible to represent database summaries automatically. Secondly, it must be allowed to utilize fuzzy knowledge, since actual domain knowledge tends toward including fuzziness essentially.

In [11], notions of linguistic summaries with fuzzy terms were proposed to evaluate validity measures based on fuzzy set theory. However, conjuctive summaries, where multiple attributes are included, are not considered. Han *et al.* proposes an attribute-oriented induction method to extract database summaries [4]. In its attribute-oriented induction method, each attribute value of a tuple is substituted with a more generalized description. After one pass of the substitution, equivalent classes of generalized tuples are identified and each class is then regarded as a candidate summary. This bottom-up procedure is repeated until satisfactory or qualified summaries are obtained. In [6], Lee and Kim propose an interactive top-down summary discovery process that utilizes fuzzy ISA hierarchies as domain knowledge. In their paper, a generalized tuple as a representational form of a database summary including fuzzy concepts is proposed. However, it would not be achieved efficiently since each tuple is disposed without concurrency.

The rest of this paper is organized as follows: A generalized tuple as a representation form of a database summary is delivered in Section 2. Fuzzy set hierarchies used as domain knowledge is proposed in Section 3. In Section 4, domain knowledge representation using fuzzy Petri net to discovery generalized tuples based on given fuzzy domain knowledge is presented. In Section 5, a summary discovery process based on fuzzy Petri net is proposed. Finally, conclusions are discussed in Section 6.

2. REPRESENTATION OF DATABASE SUMMARIES

Hereupon, a generalized tuple as a representation form of a database summary is defined. Also, we explicate the way to evaluate the validity of a generalized tuple concerning a given database. We suppose that all attributes appear in a single table to avoid unnecessary complexity of the presentation. However, this work can be applied to any other data models where a database tuple can be regarded as a series of attribute values.

There are many domain concepts that are too complicated for precise descriptions to be obtained in practice. Therefore, fuzziness (or approximation) must be introduced in order to obtain a reasonable, yet trackable, model. It is more natural to express such domain concepts in terms of fuzzy sets. Consequently, a vector of fuzzy sets is used to availingly represent a database summary.

A fuzzy set f on a domain D is defined by its membership function $\mu_f(x)$, where $\mu_f(x) \rightarrow [0, 1]$, is the membership function of the fuzzy set f; $\mu_f(x)$ indicates the degree of membership of x in f. Since $\mu_f(x)$ represents the degree to which an element x belongs to a fuzzy set f, a conventional set is assumed to be a special case of a fuzzy set whose membership degrees are either one or zero.

An example of a generalized tuple concerning an attribute scheme (Item, Customer) is <juice, writer>. It implies an assertion that "The item is juice and the customer is writer who bought it". Definition of a generalized tuple is as follows: [6]

Definition 1: A *generalized tuple* is defined as an m-ary tuple $\langle f_1, f_2, \ldots, f_m \rangle$ of fuzzy sets on an attribute scheme $(A_1, A_2, \ldots, A_m)$. The definition is interpreted as an assertion that "each tuple has attribute value $f_1, f_2, \ldots, f_m$ for attributes $A_1, A_2, \ldots, A_m$, respectively. Given two different generalized tuples $g_1 = \langle f_{11}, f_{12}, \ldots, f_{1m} \rangle$ and $g_2 = \langle f_{21}, f_{22}, \ldots, f_{2m} \rangle$, on the same attribute scheme, g_2 is called a generalization of g_1, iff $\forall k, f_{1k} \subseteq f_{2k}$.

As pointed out in [2], a given database and a set of possible generalized tuples are regarded as an *instance space* and a *pattern space*, respectively. A summarization process is to choose valid generalized tuples from a pattern space of a given instance space. Consequently, the support degree of a generalized tuple is determined by the sum of database correspondence tuples. This notion is formulated as the support degree as follows: [6]

Definition 2: The *support degree* of a generalized tuple $g = \langle f_1, f_2, \ldots, f_m \rangle$ on an attribute scheme $(A_1, A_2, \ldots, A_m)$ is defined as follows:

$$SD(g \mid C) = SD(g) = \left[\sum_{i=1}^{|C|} Min\left[\mu_{f_1}(r_i A_1), \mu_{f_2}(r_i A_2), \ldots, \mu_{f_m}(r_i A_m)\right] \right] \Big/ |C|$$

where $\mu_{fj}(r_i A_j)$ denotes the membership degree of an attribute A_j of a tuple r_i concerning a fuzzy set f_j, and $|C|$ denotes the cardinality of the database C. We'll denote $SD(g \mid C)$ as $SD(g)$ for simplicity.

Let us look at an example of support degree computation. Suppose a generalized tuple g on an attribute scheme (Item, Customer) is <juice, writer>, where fuzzy sets <juice> and <writer> and a data view C is given as Table 1, support degree of <juice, ->, <-, writer> and <juice, writer> is computed as follows. The first tuple #1 support <juice, ->, <-, writer> and <juice, writer> as strong as 0.4, 1.0 and 0.4, respectively, since its first attribute and second attribute value, apple milk and Mary, belongs to fuzzy sets, <juice, -> and <-, writer>, to the degrees, 0.4 and 1.0, respectively. Analogically, the rest tuples suppost <juice, ->, <-, writer> and <juice, writer> are similar. As result, we can say that <juice, -> is supported by 0.4 + 1 + 1 + 0.2 + 0.1 = 2.7 tuples out of a total of five tuples, i.e., 54% of the database table. Similarily, the generalized tuple g, i.e., <juice, writer>, is supported by 0.4 + 0 + 0.8 + 0.2 + 0.1 = 1.5 tuples out of a total of five tuples, i.e., 30% of the database table.

Table 1: The Support Strength of Example Data Tuples

Tuple	Item	Customer	Juice	Writer
#1	Apple milk	Mary	0.4	1.0
#2	Graph juice	Lisa	1.0	0.0
#3	Watered lemon juice	Peter	1.0	0.8
#4	Pink lady	Tom	0.2	0.9
#5	Beer	Norman	0.1	0.3
Total			2.7	3.0

Tuple	Tuple #*i* supports <Juice, Writer> as strong as
#1	Min(0.4, 1.0) = 0.4
#2	Min(1.0, 0.0) = 0.0
#3	Min(1.0, 0.8) = 0.8
#4	Min(0.2, 0.9) = 0.2
#5	Min(0.1, 0.3) = 0.1
Total	1.5

3. FUZZY DOMAIN KNOWLEDGE

Fuzzy set hierarchies are too flexible of a structure to represent data hierarchies. A fuzzy set hierarchy [6] is defined a partial ordered set, $(\Phi, \subseteq)$ where Φ is a set of fuzzy sets defined on the domain D. The binary relation $\subseteq$ is the set inclusion relationship between two fuzzy sets. A fuzzy set f_i is a direct subset of another fuzzy set f_j if $f_i \subseteq f_j$ and there is no other fuzzy set f_k such as $f_i \subseteq f_k \subseteq f_j$. A fuzzy set hierarchy is shown as in Fig. 1.

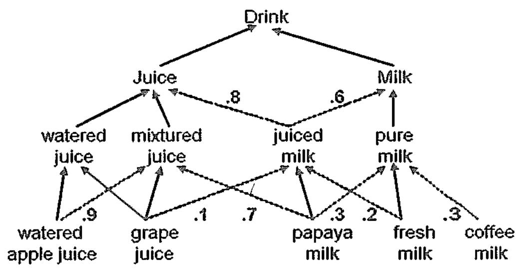

Fig. 1: A fuzzy set hierarchy on attribute drink.

As pointed out in [5], the elements of a fuzzy set can themselves be fuzzy sets. Ordinary fuzzy sets whose elements are atomic values are called level-1 fuzzy sets. Fuzzy sets whose elements are level - (k-1) fuzzy sets are called level-k fuzzy sets. Fig. 1 shows an example fuzzy set hierarchy on drink that could be used in transaction analysis. Note that a fuzzy set f in a fuzzy set hierarchy is a level-k fuzzy set, if the maximal path length from f to terminal nodes is k-1. In Fig. 1, terminal node such as grape juice is more specialized than higher level fuzzy set, i.e., mixtured juice, since an arrow from level-1 fuzzy set, say grape juice, to a higher level fuzzy set, mixtured juice. The meaning of a fuzzy set arrow from level-(k-i) to level-(k-j) can be interpreted as level-(k-i) fuzzy set is a partially specified concept of level-(k-j), where $i > j$. Table 2 depicts some level-k fuzzy sets obtained from Fig. 1.

Table 2: Level-k Fuzzy Sets

set level	membership function	level
Juice	{1.0/watered juice, 1.0/mixtured juice, 0.8/juiced milk}	3
Milk	{0.8/juiced milk, 1.0/pure milk}	3
watered juice	{1.0/watered apple juice, 1.0/grape juice}	2
mixtured juice	{0.9/watered apple juice, 1.0/grape juice, 0.7/papaya milk}	2
juiced milk	{0.1/grape juice, 1.0/papaya milk, 0.2/fresh milk}	2
pure milk	{0.3/papaya milk, 1.0/fresh milk, 0.3/coffee milk}	2
watered apple juice	{1.0/watered apple juice}	1
grape juice	{1.0/grape juice}	1
papaya milk	{1.0/papaya milk}	1
fresh milk	{1.0/fresh milk}	1
coffee milk	{1.0/coffee milk}	1

If two fuzzy sets have the same levels, we can directly determine the inclusion relationship between them. If they have different levels, the inclusion relationship between them can not be directly determined, since the domains are different. To overcome such obstacles, the level of a fuzzy set can be either upgraded or downgraded by some fuzzy set-theoretic treatments. In the transformation procedure, t-norm operator, $\otimes$, and t-conorm operator, $\oplus$, were used to obtain disjunctive combinations of membership degrees. In this paper, we use Min and Max for t-norm and t-conorm operators to adjust different level of fuzzy sets to the same through support fuzzification. But, there are also several alternatives such as Dombi, Dubois-Prade and Yager class for t-norm operators [12].

Herein, a level-3 fuzzy set Juice is downgraded to level-1 fuzzy sets in Fig. 1 is illustrated as follows:

Juice = {1.0/watered juice, 1.0/mixtured juice, 0.8/juiced milk}

= {$\oplus(\otimes(1.0,\ 1.0),\ \otimes(0.9,\ 1.0))$ / watered apple juice,

$\oplus(\otimes(1.0,\ 1.0),\ \otimes(0.9,\ 1.0),\ \otimes(0.1,\ 0.80))$ / grape juice,

$\oplus(\otimes(0.7,\ 1.0),\ \otimes(1.0,\ 0.8))$ / papaya milk,

$\otimes(0.2,\ 0.8)$ / fresh milk}

= {1.0/watered apple juice, 1.0/grape juice, 0.8/papaya milk, 0.2/fresh milk}

To visualize the implication of support fuzzification, Let's consider the reason why the membership degree of the terminal node papaya milk is determined as 0.8. Since an arrow from papaya milk to its antecedent mixtured juice, juiced milk and pure milk, respectively. The next step is to scan and select fuzzy sets from these candidates that is also a member of Juice. As a result, mixtured juice and juiced milk are both candidates. Since mixtured juice is a member of Juice, and papaya milk is a member of mixtured juice, papaya milk is also regarded as a member of Juice. By this transitivity, the membership degree of papaya milk to Juice is determined as $\otimes(\mu_{Juice}$(mixtured juice), $\mu_{mixtured\ juice}$(papaya milk)) = $\otimes(0.7,\ 1.0) = 0.7$. Meantime, the alternative transitivity that juiced milk is a member of Juice, and papaya milk is a member of juiced milk, also implies that papaya milk is regarded as a member of Juice. Following the latter transitivity, the membership degree of papaya milk to Juice is determined as = $\otimes(\mu_{Juice}$(juiced milk), $\mu_{juiced\ milk}$(papaya milk)) = $\otimes(1.0,\ 0.8) = 0.8$. Note that as far as either of such two transitivity relationships exists, the membership degree of papaya milk to Juice holds. Thence the membership degree of papaya milk to Juice is concluded as $\oplus(\otimes(0.7,\ 1.0),\ \otimes(1.0,\ 0.8)) = 0.8$. As a result of this transformation, we have a collection of level-1 fuzzy sets as shown in Table 3.

Table 3: Level-1 Fuzzy Sets Obtained Through Support Fuzzication From Level-k Fuzzy Sets.

value	watered juice	Mixtured juice	juiced milk	pure milk	Juice	Milk
watered apple juice	1.0	0.9	0.0	0.0	1.0	0.0
grape juice	1.0	1.0	0.1	0.0	1.0	0.1
papaya milk	0.0	0.7	1.0	0.3	0.8	0.6
fresh milk	0.0	0.0	0.2	1.0	0.2	1.0
coffee milk	0.0	0.0	0.0	0.3	0.0	0.3

4. DOMAIN KNOWLEDGE REPRESENTATION USING FUZZY PETRI NET

Now we present a process to discover generalized tuples based on given fuzzy domain knowledge. In our summary discovery process looks for generalized tuples in a bottom-up and concurrent manner.

Knowledge which is concerning Petri net theory and applications has been developed since the creation of Petri net by C. A. Petri in 1962. The properties, concepts and techniques of Petri nets have been developed to analyze the flow of information which occurred concurently but with constraints on the concurrence. An ordinary Petri net is described as a bipartite directed graph which is composed of several types of components: places, transitions, tokens and directed arcs. The places and their token population represent a system state and the transitions represent potential events, firing, which lead to a new state. The arcs define the state transition possibilities. Pictorially, places are expressed as circles and transitions are expressed as bars. A place is an input place if it has a directed arc connecting to the transition; a place is an output place if it has a directed arc connecting from the transition.

A Petri net represents a system when a meaning or interpretation is assigned to the various entities in the net. However, it may be difficult to represent data in precise form. To overcome these situations, fuzzy production rules have been used for knowledge representation. A fuzzy production rule is a rule that describes the fuzzy relation between two propositions. Let $R = \{ R_k \mid k = 1, 2, \ldots, n\}$ denote a set of fuzzy production rules. The general formulation of the ith fuzzy production is as follows: [1]

$$R_i : \text{IF } d_j \text{ THEN } d_k \text{ (CF} = \mu_i)$$

where

1) d_j and d_k are propositions associated with fuzzy values between zero and one.

2) μ_k is the certainty factor, $0 \le \mu_k \le 1$.

A fuzzy Petri net model is used to represent fuzzy production rule of a rule-based system [1]. The reader is referred to [1], [10] for tutorials on fuzzy Petri nets.

For representing level-k fuzzy sets, each place can potentially hold either none or one token. If a place contains a token, then the net is called a marked fuzzy Petri net. The token value in a place p_k is denoted by $\alpha(p_k)$, where $\alpha(p_k) \in [0, 1]$, and it indicates that the degree of truth of premise. Usually, a place with respect to a level-1 fuzzy set contains either 0 or 1 of token value.

In Fig. 2, the premise part "The item value is coffee milk", associated with fuzzy value = 1.0, and the transition t_1, the membership degree to pure milk associated with fuzzy value = 0.3, and then the membership degree of offee milk to pure milk associated with fuzzy value = 1.0 × 0.3 = 0.3.

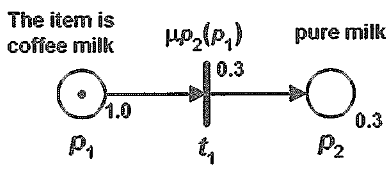

Fig. 2: Representation of a fuzzy Petri net.

A transition t_k is enabled if $\forall p_j \in I(t_k)$, $\alpha(p_j) \ge \lambda$, where λ is a threshold value and $\lambda \in [0, 1]$. A transition t_k is fired to result in removing token from its input places and depositing one token into each of its output places.

If a transition has more than one input places or output places, then the modeled rule is called a composite fuzzy production rule. If the premise protion or consequence portion of a fuzzy production rule contains "and" or "or" connectors, then it is called a composite fuzzy production rule.

The composite fuzzy production rule can be distinguished into the following rule-types applied to compute the membership degree.

Type 1: IF d_j THEN d_k (CF = μ_i). $\alpha(d_k) = \alpha(d_j) \times (\mu_i)$.

Type 2: IF d_{j1} AND… AND d_{jn} THEN d_k (CF = μ_i). $\alpha(d_k) = \otimes((\alpha(d_{j1}), \ldots, \alpha(d_{jn}))\times(\mu_i))$.

Type 3: IF d_{j1} OR … OR d_{jn} THEN d_k (CF = μ_i). $\alpha(d_k) = \oplus(\alpha(d_{j1})\times(\mu_1), \ldots, \alpha(d_{jn})\times(\mu_n))$.

Type 4: (Concurrent) IF d_{j1} THEN d_{k1} AND …AND d_{kn} (CF = μ_i). $\alpha(d_{km}) = \alpha(d_{jm}) \times (\mu_i)$, $m = 1, 2, \ldots, n$.

Type 5: (Exclusion) IF d_{j1} THEN d_{k1} OR … OR d_{kn} (CF = μ_i). $\alpha(d_{km}) = \alpha(d_{jm}) \times (\mu_i)$, $m = 1, 2, \ldots, n$.

5. A SUMMARY DISCOVERY PROCESS

Suppose that we have a collection of transaction records of a large database C whose attributes are Item and Customer as shown in Table 4. Two fuzzy set hierarchies on Item and Customer are also given as shown in Fig. 3.

Table 4: An Example Collection of Customer Transaction Log

Item	Customer	Item	Customer	Item	Customer
A_{31}	B_{31}	A_{31}	B_{31}	A_{31}	B_{36}
A_{33}	B_{31}	A_{33}	B_{33}	A_{33}	B_{37}
A_{31}	B_{32}	A_{32}	B_{31}	A_{31}	B_{35}
A_{31}	B_{31}	A_{31}	B_{31}	A_{31}	B_{34}
A_{32}	B_{34}	A_{31}	B_{31}	A_{32}	B_{34}
A_{34}	B_{34}	A_{34}	B_{31}	A_{32}	B_{34}
A_{31}	B_{31}	A_{31}	B_{31}	A_{31}	B_{31}
A_{33}	B_{31}	A_{31}	B_{33}	A_{35}	B_{31}
A_{31}	B_{31}	A_{32}	B_{33}	A_{36}	B_{31}
A_{32}	B_{33}	A_{33}	B_{31}	A_{32}	B_{37}
A_{33}	B_{32}	A_{33}	B_{31}	A_{33}	B_{32}
A_{31}	B_{31}	A_{31}	B_{31}	A_{31}	B_{31}
A_{32}	B_{31}	A_{31}	B_{33}	A_{31}	B_{34}
A_{33}	B_{32}	A_{31}	B_{35}	A_{35}	B_{35}
A_{31}	B_{34}	A_{36}	B_{33}	A_{32}	B_{31}
A_{32}	B_{37}	A_{33}	B_{31}	A_{32}	B_{34}
A_{33}	B_{36}	A_{36}	B_{31}	A_{33}	B_{31}
A_{31}	B_{31}	A_{35}	B_{31}	A_{31}	B_{37}

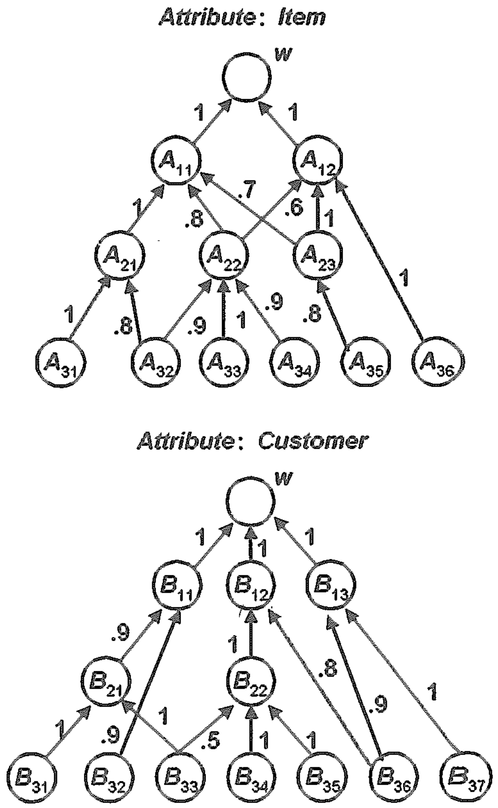

Fig. 3: Fuzzy set hierarchies for attribute Item and Customer.

Table 5 depicts some level-k fuzzy sets obtained from Fig. 3. To respresent level-k fuzzy sets on the same domain in the relational form, semantic relations are then elicited from Table 5 as shown in Table 6 [7]. If the domain of an arbitary attribute is a continous interval, a semantic relation partitions the domain into disjoint subintervals and assigns a representative membership degree to each subinterval [7].

Table 5: Level –k Fuzzy Sets

set label	Membership function	level
A_{11}	{1.0/A_{21}, 0.8/A_{22}, 0.7/A_{23}}	3
A_{12}	{0.6/A_{22}, 1.0/A_{23}, 1.0/A_{36}}	3
A_{21}	{1.0/A_{31}, 0.8/A_{32}}	2
A_{22}	{0.9/A_{32}, 1.0/A_{33}, 0.9/A_{34}}	2
A_{23}	{0.8/A_{35}}	2
A_{31}	{1.0/A_{31}}	1
A_{32}	{1.0/A_{32}}	1
A_{33}	{1.0/A_{33}}	1
A_{34}	{1.0/A_{34}}	1
A_{35}	{1.0/A_{35}}	1
A_{36}	{1.0/A_{36}}	1

Set label	Membership function	level
B_{11}	{0.9/B_{21}, 0.9/B_{32}}	3
B_{12}	{1.0/B_{22}, 0.8/B_{36}}	3
B_{13}	{0.9/B_{36}, 1.0/B_{37}}	3
B_{21}	{1.0/B_{31}, 1.0/B_{33}}	2
B_{22}	{0.5/B_{33}, 1.0/B_{34}, 1.0/B_{35}}	2
B_{31}	{1.0/B_{31}}	1
B_{32}	{1.0/B_{32}}	1
B_{33}	{1.0/B_{33}}	1
B_{34}	{1.0/B_{34}}	1
B_{35}	{1.0/B_{35}}	1
B_{36}	{1.0/B_{36}}	1
B_{37}	{1.0/B_{37}}	1

Table 6: Semantic Relation Representation Fuzzy Sets in the Fuzzy Set Hierarchies

For *Attribute Item*

Item	A_{21}	A_{22}	A_{23}	A_{11}	A_{12}	w
A_{31}	1.0	0.0	0.0	1.0	0.0	1.0
A_{32}	0.8	0.9	0.0	0.8	0.5	1.0
A_{33}	0.0	1.0	0.0	0.7	0.5	1.0
A_{34}	0.0	0.9	0.0	0.7	0.5	1.0
A_{35}	0.0	0.0	0.8	0.4	0.6	1.0
A_{36}	0.0	0.0	0.0	0.0	1.0	1.0

For *Attribute Customer*

Item	B_{21}	B_{22}	B_{11}	B_{12}	B_{13}	w
B_{31}	1.0	0.0	0.9	0.0	0.0	1.0
B_{32}	0.0	0.0	0.9	0.0	0.0	1.0
B_{33}	1.0	0.5	0.9	0.5	0.0	1.0
B_{34}	0.0	1.0	0.0	0.7	0.0	1.0
B_{35}	0.0	1.0	0.0	0.7	0.0	1.0
B_{36}	0.0	0.0	0.0	0.0	0.7	1.0
B_{37}	0.0	0.0	0.0	0.0	0.9	1.0

Fig. 4 shows the fuzzy Petri net representation for the summary discovery. By virtue of bottom-up summary discovery process, in the first instance, a transaction record <A_{31}, B_{31}> is accessed. The budget of accumulation frequency and support degree (SD) of a fuzzy set A_{31} and B_{31} runs to the degrees, 1 and 0.0185, respectively, since the number of tuples are enumberated to be fifty four in transaction file.

Meantime, each transition t from max transition t_m to t_1 is trigered. By the transitivity, transition t_1 is fired, the budget of accumulation frequency of level-1 fuzzy set, i.e., A_{31}, to level-2 fuzzy set, i.e., A_{21}, is obtained to the original order, 0, plus the token value $1.0 \times \mu(t_1) = 1.0$. Note that the token value from input place A_{31} of transition t_1 multiplied by $\mu(t_1)$ is greater than the one of output place A_{21}. In such instance, the original token value of output place of selected transition t_i is substituted for minimum token value of its input place multiplied by $\mu(t_1)$.

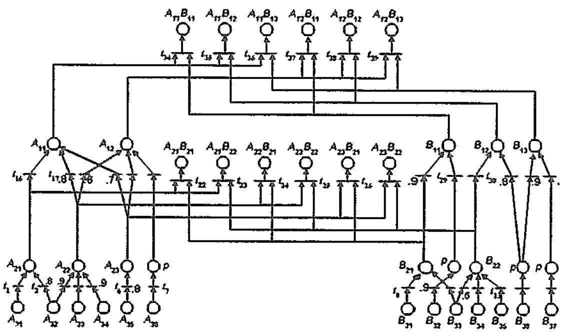

Fig. 4: The fuzzy Petri net representation for the summary discovery.

In Fig. 5, after completely accessing data records from transaction file, we can conclude that the identity trade of customer, B_{11}, who shopped class of item, A_{11}, most frequent. In particular, the identity trade of customer, B_{21}, had oftentimes shopped class of item, A_{21}.

Then, the total number of disk access is the number of transitions, i.e., specialized phases, multiplied by the number of disk accesses to read database tuples in C. Let us denote the number of transitions is a constant. As a result, the complexity of the summary discovery is $O(n)$, where n is the size of total tuples in C. Thus, we claim that the cost of our summary discovery process increases linearly along with the number of database tuples.

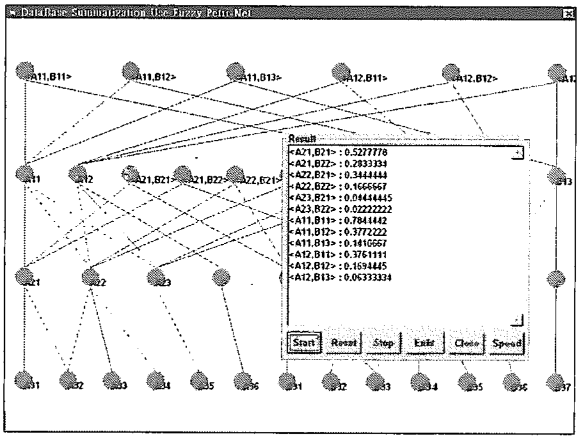

Fig. 5: After accessing data records from transaction file completely, we can conclude that the identity trade of customer, B_{11}, who shopped class of item, A_{11}, most frequent.

6. CONCLUSION

In the Internet base electronic market, in customer profiling, characteristics of good customers are identified with the goals of predicting who will become one and helping marketing departments targer new prospects. Data mining can find patterns in a customer database that can be applied to a prospect database so that customer acquistion can be targeted appropriately.

In this paper, we have presented data mining technique, a summary discovery process based on the fuzzy Petri net model to form more generalized tuples. The proposed method is more flexible than the one presented in [4] by reason of the capability to deal with fuzzy domain knowledge. The proposed method is more efficient than the one presented in [6] due to the concurrent process of summary discovery technique.

We have defined a general type of fuzzy Petri net as a representation form of a database summary including fuzzy concepts. By virtue of fuzzy Petri net model where fuzzy set hierarchies in the actual domain are natually expressed, the discovery process yields more accurate database summaries. Fuzzy set hierarchies makes it possibile to decrease unnecessary hypothesis derivations without missing any potentially generalized tuples.

Reference

[1] S. M. Chen, J. S. Ke, and J. F. Chang, "Knowledge representation using fuzzy petri nets," *IEEE Trans. Knowledge and Data Engineering*. vol. 2, no. 3, 311-319, 1990.

[2] P. Cohen and E. Feigenbaum, *The handbook of Artificial Intelligence*. vol. 3, Los Altos, CA: Kaufmann, 411-418, 1982.

[3] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus, "Knowledge discovery in databases: An overview," *Knowledge Discovery in Databases*. Menlo Park, CA: AAAI Press, 1-27, 1991.

[4] J. Han, Y. Cai, and N. Cercone, "Knowledge discovery in databases: An attribute-oriented approach," *Proc. 18th VLDB Conf.*, 547-559, 1992.

[5] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*. Eaglewood Cliffs, NJ: Prentice-Hall, 260-265, 1988.

[6] D. H. Lee and M. H. Kim, "Database summarization using fuzzy ISA hierarchies," *IEEE Trans. Systems, Man, and Cybernetics—Part B: Cybernetics*. vol. 27, no. 1, 68-77, 1997.

[7] D. H. Lee and M. H. Kim, "Accommodating subjective vagueness through a fuzzy extension to the relational data model," *Inf. Syst.* vol. 18, no. 6, 363-374, 1993.

[8]C. Looney, "Fuzzy Petri nets for rule-based decisionmaking," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 18, no. 1, 178-183, 1988.

[9] J. S. Park, M-S Chen and P. S. Yu, "Using a hash-based method with transaction trimming for mining association rules," *IEEE Trans. Knowledge and Data Engineering*. vol. 9, no. 5, 813-825, 1997.

[10] S-K Yu, "Knowledge representation and reasonong using fuzzy Pr/T net-systems," *Fuzzy Sets and Systems*. vol. 1, 317-322, 1993.

[11] R. Yager, "On linguistic summaries of data. *Knowledge Discovery in Databases*," Menlo Park, CA: AAAI Press, 347-363, 1991.

[12] L-X Wang. *A Course in Fuzzzy Systems and Control*. Upper Saddle River, NJ: Prentice-Hall, 37-42, 1997.