Association Rule Mining by Agents

Xiaowei Yan, Chengqi Zhang, Shichao Zhang, and John Debenham
Faculty of Information Technology
University of Technology Sydney
Sydney, NSW 2007, Australia
{xyan, chengqi, zhangsc, debenham}@it.uts.edu.au

Abstract—**This paper describes a distributed system of intelligent agents, MARA (Mining Association Rule Agents), for performing association rules mining over databases on behalf of a community of users. MARA can not only identify association rules in databases, but also refine the mined rules by the information sharing among users with similar interests automatically. MARA provides agents which can collect outer information from such as Web, journals, and news medium. And then the collecting information is also used to refine the mined association rules for the purpose of mining very small databases. For example, nuclear power plants and earthquake bureaus have only some very small databases. Apparently, the data in small databases may not be large enough to form any meaningful patterns. We explore a functional method for these databases. We evaluate the proposed techniques in the system, and our experimental results demonstrate that the system is efficient and promising.**

Keywords: Data mining, KDD, multi-agent, information gathering.

I. Introduction

Due to the potential power of multi-agent, there has been an increasing interest in multi-agent technology in recent years. In particular, cooperative and mobile agent systems received enormous attention. The application areas based on agents include, but are not limited to, distributed computing, software engineering, electronic commerce, system design, robots, and intelligent system.

Also, agent techniques are recently applied into data mining for dealing with very large databases [5], [10]. Indeed, the pressure of enhancing corporate profitability has caused companies to spend more energy in identifying diverse opportunities such as sales and investments. So huge amounts of data including inner and outer information are collected in their databases for decision-support purpose. A short list of examples is probably enough to place the current situation into perspective [10]:

- NASA's Earth Observing System (EOS) of orbiting satellites and other spaceborne instruments send one terabyte of data to receiving stations every day.
- The World Wide Web is estimated to have at least 450,000 hosts, 1 million sites and as many as 80 million pages (as of March 1998).
- By the year 2000 a typical Fortune 500 company is projected to possess more than 400 trillion characters in their electronic databases requiring 400 terabytes of mass storage.

Hence, today we are overwhelmed with data. So we must take techniques to manage, mine, analyze, process, and make them well-kept so as to efficiently operate and apply the data. There are many models proposed to discover useful patterns from large scale databases [1], [5], [10].

However, previous algorithms have been focused on the inner data for a given application. This work is dealt with not only the inner data for a given application, but also its outer related data. It is particularly useful to the organizations with very small databases (simply written as SDs). For example, new companies, nuclear power plants and earthquake bureaus have some very small databases. Apparently, these companies/organizations also expect to apply data mining techniques to extract practical patterns in their small databases so as to make their decisions. However, the data in these databases such as the accident database of a nuclear power plants and the earthquake database in an earthquake bureau, may not be large enough to form any useful patterns. The current mining techniques cannot work well in these small databases. To explore a valuable method for these applications, we present a new mining model in this paper, which is based on agents.

The rest of this paper is organized as follows. We begin with introducing the architecture of MARA in Section *II*. In Section *III* we presents the watching agent. Section *IV* shows the data mining agent. Section *V* advocates a synthesizing agent. Section *VI* proposes a fusion agent. And we simply conclude this paper in the last section.

II. Architecture of MARA

This section discusses the facilities which MARA agents offer the users in data mining. The architecture of MARA is drawn in Figure 1.

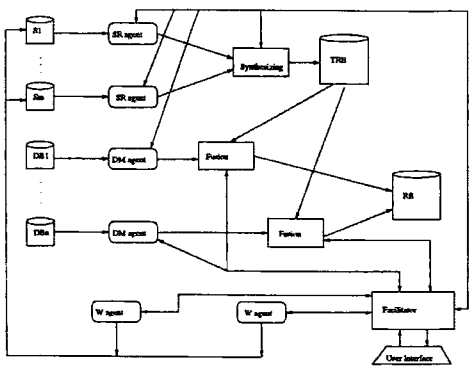

Fig. 1. The MARA architecture

In the figure, DB_i $(1 \leq i \leq n)$ denotes a database, SB_i $(1 \leq i \leq m)$ denotes a source collected by a W agent, TRB denotes a temporary rule base, RB denotes association rule base, DM agent stands for a data mining agent, W agent stands for watcher responsible for collecting outer information from various medium, SR agent stands for selection and representation responsible for getting needed rules from sources. The facilitator is responsible for managing the agents and interfacing users. The synthesizing is an agent responsible for analyzing and synthesizing the rules collected from different sources. The fusion is an agent responsible for synthesizing and refining the mined rules by the collected rules. The user interface is also an agent responsible for inputting requirements to the system and outputting the mined rules to users. These agents share their information through the facilitator in the system. The functions of agents will be presented in following sections.

The work procedure of MARA is as follows. (1) A user sends a "mining" request to MARA via the user interface to the facilitator. The facilitator firstly calls DM agents to mine the databases given by the user. Secondly, the DM agents pass the facilitator the mining rules. Finally, the facilitator forwards the user the mined results by the user interface. (2) When a user needs high-confidence results, MARA can support the applications. A user sends a "mining" request to the facilitator by the user interface. The facilitator firstly calls DM agents to mine the databases given by the user. The DM agents pass the facilitator the mining rules. Secondly, the facilitator calls W agents to gather information from sources, which is related the mined rules. Thirdly, the collected information is selected and represented into the form of rules by SR agents. Fourthly, the facilitator calls the synthesizing to analyze and synthesize the collected rules. Fifthly, the facilitator calls the fusion to refine the mined rules by synthesizing the collected rules. Finally, the facilitator passes the user the mined results by the user interface.

III. Watching Agent

The watching agent (W agent) is responsible for collecting outer information from various medium. It is described as follows.

To mine databases, we may use outer data and knowledge collected from such as emails, Web, journals, papers, and newspapers. To discover useful patterns in databases, we can first mine them, and then synthesize the mined association rules and the collected information. So we suggest a way to gather information from varied media in this section.

To gather useful information related to a given database, we can let knowledge watchers to collect scientific, technical, and economic information from such as journals, newspapers, and the Web. And then the information is represented as the same as what we want. Generally, we can collect the needed information from the following four transmissible mediae.

Firstly, the vast amount of information available on the WWW (World Wide Web) has great potential to improve the quality of decisions [6], [7]. This means that we can collect the related information in WWW for enhancing the mined results in very small databases. However, data in WWW is apparently with rough, structureless, dynamical, changeful, uncertain, and huge. And the large number of information sources and their different levels of accessibility, reliability and associated costs present a complex information gathering coordination problem [6], [7]. On the other hand, the gathered information must be transformed into the representation that we want. The information from WWW is generally free but time-consuming.

Secondly, emails are currently a novel and prevailing way to quickly and effectively share and exchange information. The information from email is controllable. And the representation of the information can be of the form that we want. The information may need to be paid.

Thirdly, news mediae such as TV, radio, magazines and newspapers are also an important way to get the related information. We often hear/read news as "Because A, then B in some place at a certain time", or "B was happened in some place at a certain time and the causes of accident are investigating". Hence, the representation of the information can easily be transformed into the form that we want. Much information from WWW is free and rapid.

Finally, academic forums such as books, journals, conferences, tutoring, seminars, and academic magazines are commonly a main way to obtain theoretical information. The information is generally matured, explained, and detailed. But some of them are conjectures and need to be proven. The information must be paid.

However, the gathered information would be analyzed, tested, synthesized, and refined before they are applied because they may contain noise or they are unfit for other places/time. So the outer information is taken as an interpreted knowledge to enhance the patterns mined in the given small database in this paper.

It is not our main goal in this paper to construct satisfying models for collecting information from variety media. For WWW, there are many nice methods for information gathering proposed in current literature [4], [6], [7], [8]. For simplicity, we only illustrate how to gather useful information from the Web by some tools offered in the Web.

Individuals and organizations can take advantage of remarkable possibilities of access to information and knowledge that the Internet provides. Web technologies such as HTTP and HTML have dramatically changed enterprise information management. Information search engines such as Yahoo, Alta Vista, Excite, and so on have offered easier way to get information that you need. Moreover, an intranet relying on Internet technology and protocols enables intra-organizational communication and internal information sharing through the corporate internal network. For example, a multinational corporation can benefit from intranets and the Internet to gather, manage, distribute, and share knowledge, inside and outside the corporation.

Generally, a company can exploit the Internet and in-

tranet features in several ways. It can use internal HTML or XML pages or external URLs containing organizational dataset, making it accessible throughout the company. More proactive methods of creating and revising corporate dataset include integrating messages exchanged through email in the corporate dataset, extracting information from the external Web sources for technological or strategic intelligence, and using computer-supported cooperative work tools to support complex-system collaborative design or collaborative software development. The wide variety of organizational choices involves several actors with different roles [8]:

- human knowledge sources (such as experts, specialists, or operators), whose knowledge must be made explicit or who have written documents that others will access through the organizational dataset;
- knowledge engineers, who acquire and model knowledge;
- knowledge watchers, who gather, filter, analyze, and distribute knowledge elements from the external world (from external information Web sources, for example);
- organizational dataset developers, who concretely build, organize, annotate, maintain, and evolve the corporate dataset;
- a team of validating experts (for example, a reference team), who validate the knowledge elements before their insertion in the organizational dataset;
- corporate dataset users, who must easily access and reuse the elements in the dataset;
- organizational dataset manages, who supervise the organizational project on the dataset.

Wherever knowledge is collected, it would generally be represented to the form what we need in applications. In this paper, all collected knowledge are represented as rules.

IV. Data Mining Agent

The data mining agent (*DM* agent) is responsible for discovering databases. It is described as follows.

Identifying Itemsets of Interest

Generally, large itemsets are interested in discovering associations in databases. In previous work, the main time is taken in identifying large itemsets due to the fact that the mined databases are commonly huge. However, we only deal with small databases in this paper. Consequently, we can use any one of proposed algorithms of recognizing large itemsets from databases in current technical articles. The idea is to statistics and dig up all itemsets in a given database, which each itemset is greater than or equal to the minimum support (*minsupp*). So we only give an example to show how to get all large itemsets as follows.

Example 1: A transaction database TD with 10 transactions in Table 1 is obtained from a grocery store. Let $A = bread$, $B = coffee$, $C = tea$, $D = sugar$, $E = beer$, $F = butter$. Assume $minsupp = 0.3$. The supports of single large items are shown in Table 2 and other itemsets are listed in Table 3.

Table 1: Transaction database TD

Transaction ID	Items
T_1	A, B, D
T_2	A, B, C, D
T_3	B, D
T_4	B, C, D, E
T_5	A, C, E
T_6	B, D, F
T_7	A, E, F
T_8	C, F
T_9	B, C, F
T_{10}	A, B, C, D, F

Table 2: Supports of single large items

Item	Number of Transactions	Support $p(X)$
A	5	0.5
B	7	0.7
C	6	0.6
D	6	0.6
E	3	0.3
F	5	0.5

Table 3: Supports of large itemsets

Itemset	Support $p(X)$	Itemset	Support $p(X)$
A, B	0.3	A, C	0.3
A, D	0.3	B, C	0.4
B, D	0.6	B, F	0.3
C, D	0.3	C, F	0.3
A, B, D	0.3	B, C, D	0.3

Finding out Association rules

There are many proposed methods of measuring the uncertainty of association rules. We advocate to apply subjective Bayesian method [3] to capture the uncertainty of association rules in our agents.

Duda, Hart and Nilsson proposed a "subjective Bayesian method for rule-based inference systems" [3] which aims at combining some of the advantages of both formal and informal methods for uncertain reasoning.

Assuming $\Omega_X = X \cup \overline{X}$ and $\Omega_H = H \cup \overline{H}$ and given the simple expert rule "if X then H", according to the rule of conditional probability we derive

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

for $P(X) > 0$. If X is known to be true then the premise of the rule will match the knowledge base and the above formula applies to compute the a posteriori probability $P(H|X)$ on the basis of the probabilities $P(H)$, $P(X) = 1$, and the conditional probability $P(X|H) = 1$, i.e., $P(H|X) = P(H)$.

Dividing the above formula by the corresponding formula for $P(\overline{H}|X)$ leads to the following from of Bayes' formula

$$\frac{P(H|X)}{P(\overline{H}|X)} = \frac{P(X|H)}{P(X|\overline{H})} * \frac{P(H)}{P(\overline{H})}$$

which applies to compute the so-called posterior odds on hypothesis H under the condition that the premise

matches, i.e., that X is known to be true. Notice that a possible uncertainty concerning the premises' truth is not considered at this stage of derivation. Let

$$O(H) = \frac{P(H)}{P(\overline{H})} = \frac{P(H)}{1 - P(H)}$$

$$O(H|X) = \frac{P(H|X)}{P(\overline{H}|X)} = \frac{P(H|X)}{1 - P(H|X)}$$

$$\lambda = \frac{P(X|H)}{P(X|\overline{H})},$$

then we have

$$O(H|X) = \lambda * O(H).$$

Inversely, we have

$$P(H|X) = \frac{O(H|X)}{1 + (H|X)}.$$

By analogy we can characterize the update process for hypothesis H if premise X is known to be false:

$$O(H|\overline{X}) = \gamma * O(H),$$

where

$$\gamma = \frac{P(\overline{X}|H)}{P(\overline{X}|\overline{H})} = \frac{1 - P(X|H)}{1 - P(X|\overline{H})},$$

Consequently, the expert rules will have the following structure:

$$X \longrightarrow H, \qquad (\lambda, \gamma)$$

In this model, if $\lambda \gg 1$, then it supports that H occurs when X occurs.

This model can directly be applied to measure the uncertainties of association rules. In this model, an association rule is a relationship of the form $A \rightarrow B$, where A and B are sets of items and $A \cup B = \emptyset$. Each association rule has a support factor *supp* and a 2-tuple (λ, γ). *supp* is the ratio of the number of transactions in a database that contain the itemset $A \cup B$ to the total number of transactions in the database, (λ, γ) is the same as the above definition.

In the same reasons, for an association rule $A \rightarrow B$, support, λ and γ must be greater than or equal to some user specified minimum support (*minsupp*), minimum λ_{min} and γ_{min} thresholds, respectively.

We now demonstrate how to apply this model to measure association rules with the database in Example 1. For simplicity, we still take $P(X) = |X|/n$.

Example 2: For itemset $B \cup D$, $P(B) = 0.7$, $P(D) = 0.6$ and $P(B \cup D) = 0.5$, then

$$P(B|D) = \frac{P(B \cup D)}{P(D)} = \frac{0.6}{0.6} = 1,$$

$$P(B|\overline{D}) = \frac{P(B \cup \overline{D})}{P(\overline{D})} = \frac{0.1}{0.4} = 0.25.$$

So,

$$\lambda = \frac{P(B|D)}{P(B|\overline{D})} = \frac{1}{0.25} = 4,$$

$$\gamma = \frac{1 - P(B|D)}{1 - P(B|\overline{D})} = \frac{1 - 1}{1 - 0.25} = 0.$$

Hence,

$$supp = 60\%, \lambda = 4, \gamma = 0.$$

The other itemsets is listed in the following table.

Table 4: Some association rules, λ and γ

Association Rule $(X \rightarrow Y)$	$P(X\|Y)$	$P(X\|\overline{Y})$	λ	γ
$A \rightarrow B$	0.429	0.667	0.643	1.715
$A \rightarrow C$	0.5	0.5	1	1
$A \rightarrow D$	0.5	0.5	1	1
$B \rightarrow C$	0.667	0.75	0.889	1.332
$B \rightarrow D$	1	0.25	4	0
$B \rightarrow F$	0.6	0.8	0.75	2
$C \rightarrow D$	0.333	0.333	1	1
$C \rightarrow F$	0.6	0.6	1	1
$A \wedge B \rightarrow D$	0.5	0	∞	0.5
$B \wedge C \rightarrow D$	0.5	0.25	2	0.667

Let $minsupp = 30\%$, $\lambda_{min} = 3.5$ and $\gamma_{min} = 0.6$ then $B \rightarrow D$, $A \wedge B \rightarrow D$ and $A \wedge B \wedge C \rightarrow D$ can be extracted as rules. If $minsupp = 50\%$, $\lambda_{min} = 3.5$ and $\gamma_{min} = 0.3$ then only $B \rightarrow D$ can be discovered as a rule.

V. Synthesizing Agent

The synthesizing agent is responsible for synthesizing the collected outer information from various medium. For simplicity, the collected information has been represented in the form of rules. And the synthesizing is described as follows.

Generally, the number of rules may be very large when they are collected from unknown data sources. Consider a rule $X \rightarrow Y$, it has different supports $s_1, s_2, \cdots, s_m$, and confidences $c_1, c_2, \cdots, c_m$ in the gathered association rules. We can use one of the following aggregation operators to roughly aggregate this rule.

(1) Maximum aggregation operator

$$a \oplus b = Max\{a, b\}$$

(2) Average aggregation operator

$$a \oplus b = \frac{1}{2}(a + b)$$

Example 3: Suppose we have the following rules from different unknown data sources.

$A \wedge B \rightarrow C$ with $supp = 0.4, conf = 0.72$

$A \rightarrow D$ with $supp = 0.3, conf = 0.64$;

$A \rightarrow D$ with $supp = 0.36, conf = 0.7$;

$A \wedge B \rightarrow C$ with $supp = 0.5, conf = 0.82$;

$A \rightarrow D$ with $supp = 0.25, conf = 0.62$;

For rule $A \wedge B \rightarrow C$, according to the maximum aggregation operator we have

$$supp = Max\{0.4, 0.5\} = 0.5,$$
$$conf = Max\{0.72, 0.82\} = 0.82.$$

According to the average aggregation operator we have

$$supp = \frac{1}{2}(0.4 + 0.5) = 0.45,$$
$$conf = \frac{1}{2}(0.72 + 0.82) = 0.77.$$

To construct a better method, we now use clustering to obtain Normal Distribution intervals among the supports and confidences of a gathered rule when they exist.

A. Normal Distribution

Suppose a rule $X \rightarrow Y$ has the following supports and confidences in the gathered association rules:

$s_1, c_1,$
$s_2, c_2,$
$\cdots$
$s_n, c_n,$

If these confidences are irregularly distributed, we can apply one of the above models to aggregate them, but the aggregation is rather rough. However, if these confidences are in a normal distribution, we can take an interval as the confidence and a corresponding interval as the support. In other words, for $0 \leq a \leq b \leq 1$, let m be the number of confidences belonging to interval $[a, b]$. If $m/n \geq \lambda$, then these confidences are in a normal distribution, where $0 < \lambda \leq 1$ is a threshold given by human experts. This means that $[a, b]$ can be taken as the confidence of rule $A \rightarrow B$. For the corresponding supports, we can estimate an interval as the support of the rule. In other words, suppose we have a random variable $X \sim N(\mu, \sigma^2)$ and we need the probability

$$P\{a \leq X \leq b\} = \frac{1}{\sigma\sqrt{2\pi}} \int_a^b e^{-(x-\mu)^2/2\sigma^2} dx$$

to satisfy $P\{a \leq X \leq b\} \geq \lambda$ and, $|b - a| \leq \alpha$, where X is valued from $c_1, c_2, \cdots, c_n$, and α is a threshold given by domain experts.

For $c_1, c_2, \cdots, c_n$, let $c_{i,j} = 1 - |c_i - c_j|$ be the *closeness* value between c_i and c_j, the closeness value between any two confidences be given below.

Table 5: The distance table

	c_1	c_2	$\cdots$	c_n
c_1	$c_{1,1}$	$c_{1,2}$	$\cdots$	$c_{1,n}$
c_2	$c_{2,1}$	$c_{2,2}$	$\cdots$	$c_{2,n}$
$\cdots$	$\cdots$	$\cdots$		$\cdots$
c_n	$c_{n,1}$	$c_{n,2}$	$\cdots$	$c_{n,n}$

We can use clustering technology to obtain this normal $[a, b]$. To determine the relationship between confidences, a closeness degree measure is required. The measure calculates the closeness degree between two confidences by closeness values. We define a simple closeness degree measure as follows:

$$Close(c_i, c_j) = \sum (c_{k,i} * c_{k,j})$$

where "k" is summed across the set of all confidences. In effect the formula takes the two columns of the two confidences being analyzed, multiplying and accumulating the values in each row. The results can be placed in a resultant "n" by "n" matrix, called a *confidence-confidence matrix.* This simple formula is reflexive so that the generated matrix is symmetric.

For example, let $\lambda = 0.7$, $\alpha = 0.08$, $minconf = 0.65$, an aggregated rule $X \rightarrow Y$ with confidences $c_1 = 0.7$, $c_2 = 0.72$, $c_3 = 0.68$, $c_4 = 0.5$, $c_5 = 0.71$, $c_6 = 0.69$, $c_7 = 0.7$, and $c_8 = 0.91$, and the closeness value between any two confidences is given below.

Table 6: The distance relation table

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
c_1	1	0.98	0.98	0.8	0.99	0.99	1	0.79
c_2	0.98	1	0.96	0.78	0.99	0.97	0.98	0.81
c_3	0.98	0.96	1	0.82	0.97	0.99	0.98	0.77
c_4	0.8	0.78	0.82	1	0.79	0.81	0.8	0.59
c_5	0.99	0.99	0.97	0.79	1	0.98	0.99	0.8
c_6	0.99	0.97	0.99	0.81	0.98	1	0.99	0.78
c_7	1	0.98	0.98	0.8	0.99	0.99	1	0.79
c_8	0.79	0.81	0.77	0.59	0.8	0.78	0.79	1

Its confidence-confidence matrix is shown as follows.

Table 7: Confidence-Confidence matrix

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
c_1		7.0459	7.0855	6.0181	7.125	7.1252	7.1451	5.9546
c_2	7.0459		7.0247	5.9609	7.0664	7.0646	7.0851	5.9164
c_3	7.0855	7.0247		5.9793	7.0648	7.067	7.0936	5.898
c_4	6.0181	5.9609	5.9793		5.9974	6.0068	6.0181	4.971
c_5	7.125	7.0664	7.0648	5.9974		7.1047	7.125	5.9435
c_6	7.141	7.0646	7.067	6.0068	7.1047		7.1252	5.9341
c_7	7.1451	7.0851	7.0936	6.0181	7.125	7.1252		5.9546
c_8	5.9546	5.9164	5.898	4.971	5.9435	5.9341	5.9546	

There are no values on the diagonal since that represents the auto-correlation of a confidence to itself. Assume that 6.9 is the threshold that determines if two confidences are considered close enough to each other to be in the same class. This produces a new binary matrix called the *confidence relationship matrix* as follows.

Table 8: Confidence closeness relationship matrix

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
c_1		1	1	0	1	1	1	0
c_2	1		1	0	1	1	1	0
c_3	1	1		0	1	1	1	0
c_4	0	0	0		0	0	0	0
c_5	1	1	1	0		1	1	0
c_6	1	1	1	0	1		1	0
c_7	1	1	1	0	1	1		0
c_8	0	0	0	0	0	0	0	

Cliques require all confidences in a cluster to be within the threshold of all other confidences. The methodology to create the clusters using cliques is described in Procedure 1 as follows.

Procedure 1: Cluster

Input: c_i: confidence, λ: threshold value;

Output: *Class*: class set of closeness confidences;

(1) **let** i=1;
(2) **select** c_i and place it in a new class;
(3) $r = k = i + 1$;
(4) **validate** if c_k is within the threshold of all terms within the current class;
(5) **if** not, let $k = k + 1$;
(6) **if** $k > n$ (number of confidences) **then**
$r = r + 1$;
if $r = m$ **then go to** (7) **else**
$k = r$;
create a new class with c_i in it;
go to (4);
(7) **if** the current class only has c_i in it and there are other classes with c_i in them **then**
delete the current class;
else $i = i + 1$;
(8) **if** $i = n + 1$ **then go to** (9)
else go to (2);
(9) **eliminate** any classes that duplicate or are elements of other classes.

Applying the above procedure to the above example in this section, the following classes are created:
Class 1: c_1, c_2, c_3, c_5, c_6, c_7
Class 2: c_4
Class 3: c_8

For Class 1, $a = 0.68$, $b = 0.72$. Hence,

$$|b - a| = |0.72 - 0.68| = 0.04 < \alpha = 0.08,$$

$$P\{a \le X \le b\} = 6/8 = 0.75 > \lambda = 0.7,$$

and

$$b > a > minconf = 0.65.$$

For Class 2, $a = 0.5$, $b = 0.5$. Hence,

$$|b - a| = |0.5 - 0.5| = 0 < \alpha = 0.08,$$

$$P\{a \le X \le b\} = 1/8 = 0.125 < \lambda = 0.7,$$

and

$$b = a < minconf = 0.65.$$

For Class 3, $a = 0.91$, $b = 0.91$. Hence,

$$|b - a| = |0.91 - 0.91| = 0 < \alpha = 0.08,$$

$$P\{a \le X \le b\} = 1/8 = 0.125 < \lambda = 0.7,$$

and

$$b = a > minconf = 0.65.$$

Therefore, $[0.68, 0.72]$ can be taken as the interval of the confidence of rule $A \rightarrow B$.

We can also aggregate the corresponding support of a rule into an interval in the same way. For simplicity, we can also take the minimum of supports corresponding to a class as its support.

B. Algorithm Design

Let $A \rightarrow B$ be a gathered rule, $s_1, c_1, s_2, c_2, \cdots, s_n, c_n$ the supports and confidences of the rule, *minsupp* and *minconf* the threshold values given by the user and λ and α the threshold values given by domain experts. Our aggregation algorithm for association rules from different unknown data sources is designed as follows.

Procedure 2: RelativeAggregation
Input: $A \rightarrow B$: rule;
$s_1, s_2, \cdots, s_n$: the supports of the rule;
$c_1, c_2, \cdots, c_n$: the confidences of the rule;
minsupp, *minconf*, λ, α: threshold values;
Output: $A \rightarrow B$: aggregated association rule;
(1) **for** the confidences of $A \rightarrow B$ **do**
call Cluster;
(2) **for** each class C **do**
begin
let $a \leftarrow$ the minimum of values in C;
let $b \leftarrow$ the maximum of values in C;
let $d_C \leftarrow |b - a|$;
let $P_C\{a \le X \le b\} \leftarrow |C|/n$;
end;
(3) **for** all classes **do**
if there is a class C satisfying $d_C \le \alpha$, $P_C \ge \lambda$ and $a \ge minconf$ **then**
begin
let *supp* $\leftarrow$ the minimum of supports corresponding to C;
output $A \rightarrow B$ as a valid rule
with support *supp* and confidence interval $[a, b]$;
end;
(4) **if** there are no classes satisfying the conditions **then**
begin
let $supp \leftarrow \frac{1}{n}(s_1 + s_2 + \cdots + s_n)$;
let $conf \leftarrow \frac{1}{n}(c_1 + c_2 + \cdots + c_n)$;
if $supp \ge minsupp$ and $conf \ge minconf$ **then**
output $A \rightarrow B$ as a valid rule
with support *supp* and confidence *conf*;
end;

The *RelativeAggregation* procedure above synthesize gathered association rules from unknown data sources into two kinds of rules: one is that the supports and confidences of each rule are in normal distribution and they are clustered into intervals; the other is that the supports and confidences of each rule are not in normal distribution and they are roughly synthesized points. Step (1) clusters the confidences of a rule. Step (2) solves the bounded values. Step (3) checks if there is a clustered class that satisfies the given threshold values. The supports and confidences of each rule in normal distribution are evaluated in this step. Otherwise, the rules are synthesized in Step (4).

VI. Fusion Agent

The fusion agent is responsible for refining the mined rules by synthesizing the collected rules. The collected rules have been synthesized by the synthesizing agent. Refining the mined rules is a procedure of weighting as follows.

The mined association rules in databases may not be trusted due to the fact that the rules have too low amount

of information to being taken as knowledge in common-sense. On the other hand, the synthesized collected rules are fused by a lot of information. And the high-rank rules are generally believed in common-sense. Therefore, if a mined association rule with higher confidence matches a high-rank rule synthesized in the above method, then we can certainly extract this rule as a valid rule in the database. To catch this idea, we use weighting as follows.

Let SD and D be the given database and the synthesized source respectively, and R_1 and R_2 the set of rules in SD and D respectively. For a given rule $X \rightarrow Y$, suppose w_1 and w_2 are the weights of SD and D respectively, the weighting is defined as follows.

$$\begin{aligned} s_w(X \cup Y) &= w_1 * s_1(X \cup Y) + w_2 * s_2(X \cup Y), \\ c_w(X \rightarrow Y) &= w_1 * c_1(X \rightarrow Y) + w_2 * c_2(X \rightarrow Y). \end{aligned}$$

Certainly, we can determine the above weights w_1 and w_2 by applications, experts, users and so on. Now design the algorithm of mining databases as follows.

Algorithm 1: MiningDB

Input: SD: database; S_i: the set of the collected rules $(1 \leq i \leq m)$,

$minsupp$, $minconf$: threshold values; w_1, w_2: weights;

Output: $X \rightarrow Y$: valid association rules;

(1) **mine** SD in subjective Bayesian method;
let $R_1 \leftarrow$ the association rules in SD;
(2) **collect** $\{S_1, S_2, \cdots, S_m\}$ in such as the Web, journals, and papers;
(3) **call** RelativeAggregation(S_i);
let $R_2 \leftarrow \{S_1, S_2, \cdots, S_m\}$;
(4) **for** each rule $X \rightarrow Y \in R_1$ **do**
let $s_w \leftarrow w_1 * s_1 + w_2 * s_2$;
let $c_w \leftarrow w_1 * c_1 + w_2 * c_2$;
(5) **rank** all rules in R_1;
(6) **output** the high-rank rules in R_1;
(7) **end all**.

The *MiningDB* algorithm above generates high-rank and valid rules in a given database, where each ranked rule is with a high support and confidence. Step (1) is to generate all possible association rules in the given database SD by subjective Bayesian method. And the association rules are saved in R_1. Step (2) is to collect rules in $S_1, S_2, \cdots, S_m$ from such as Web, journals, and news medium. Step (3) is to aggregate the rules in $S_1, S_2, \cdots, S_m$ and S_m into the set S by procedure *RelativeAggregation*. And the synthesized rules in S are saved in R_2. Step (4) is to enhance the rules in R_1 by weighting. Note that if a rule $A \rightarrow B \in R_1$ and $A \rightarrow B \notin R_2$, then the rule $A \rightarrow B$ would be labelled and not be presented in the ranked results. Step (5) is to rank the synthesized association rules. And the high-rank rules are output in Step (6).

Apparently, we can also mine large scale databases with using *MiningDB*. And the mined results are certainly fused more information than previous models of mining large scale databases.

VII. Conclusions

As have seen, to discover association rules in large scale databases has received much attention recently [1], [2], [9], [11], [12], [13]. We presented a new mining model in this paper, which is based on agents. The main contributions in this paper are as follows.

- The architecture of a data mining model based on agents is built.
- Proposed to consider the knowledge inside and outside organizations when a database is mined. But previous mining models consider only the data in databases.
- Advocated to refine the mined rules by the collected rules.

References

[1] R. Agrawal, T. Imielinski, and A. Swami, Mining association rules between sets of items in large databases. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1993:207-216.

[2] S. Brin, R. Motwani and C. Silverstein, Beyond market baskets: Generalizing association rules to correlations. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1997: 265-276.

[3] R. Duda, P. Hart and N. Nilsson, Subjective Bayesian methods for rule-based inference systems. *Proc. National Computer Conference*, AFIPS, Vol. 45, 1976: 1075-1082.

[4] O. Etzioni, S. Hanks, T. Jiang, R. Karp, O. Madani, and O. Waarts: Efficient Information Gathering on the Internet. *Proceedings of FOCS*, 1996: 234-243

[5] H. Kargupta, B. Park, D. Hershberger, and E. Johnson, Collective Data Mining: A New Perspective Toward Distributed Data Mining. *Advances in Distributed and Parallel Knowledge Discovery*, Eds: Hillol Kargupta and Philip Chan. MIT/AAAI Press, 1999.

[6] V. Lesser, B. Horling, F. Klassner, A. Raja, T. Wagner, and S. Zhang. A Next Generation Information Gathering Agent. In *Proceedings of the 4th International Conference on Information Systems, Analysis, and Synthesis; in conjunction with the World Multiconference on Systemics, Cybernetics, and Informatics (SCI'98)*, Orlando, FL, July 1998.

[7] V. Lesser, B. Horling, F. Klassner, A. Raja, T. Wagner, and S. Zhang, BIG: An Agent for Resource-Bounded Information Gathering and Decision Making. In *Artificial Intelligence Journal, Special Issue on Internet Information Agents*, Vol. 118, 1-2(2000): 197-244.

[8] P. Martin and P. Eklund, Knowledge retrieval and the World Wide Web, *IEEE Intelligent Systems & Their Applications*, Vol. 15, 3(2000): 18-24.

[9] J. Park, M. Chen, and P. Yu, Using a Hash-based Method with Transaction Trimming for Mining Association Rules. *IEEE Trans. Knowledge and Data Eng.*, Vol. 9, 5(1997): 813-824.

[10] A. Prodromidis, P. Chan, and S. Stolfo, Meta-learning in distributed data mining systems: Issues and approaches, In *Advances in Distributed and Parallel Knowledge Discovery*, H. Kargupta and P. Chan (editors), AAAI/MIT Press, 2000.

[11] T. Shintani and M. Kitsuregawa, Parallel mining algorithms for generalized association rules with classification hierarchy. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1998: 25-36.

[12] Shichao Zhang, Aggregation and maintenance for databases mining, *Intelligent Data Analysis: an international journal*, Vol. 3(6) 1999: 475-490.

[13] Shichao Zhang and Xindong Wu, Large Scale Data Mining Based on Data Partitioning, Applied Artificial Intelligence, forthcoming.