Mining Fuzzy Association Rules in a Bank-Account Database

Wai-Ho Au and Keith C. C. Chan

***Abstract*—This paper describes how we applied a fuzzy technique to a data-mining task involving a large database that was provided by an international bank with offices in Hong Kong. The database contains the demographic data of over 320,000 customers and their banking transactions, which were collected over a six-month period. By mining the database, the bank would like to be able to discover interesting patterns in the data. The bank expected that the hidden patterns would reveal different characteristics about different customers so that they could better serve and retain them. To help the bank achieve its goal, we developed a fuzzy technique, called Fuzzy Association Rule Mining II (FARM II), which can mine fuzzy association rules. FARM II is able to handle both relational and transactional data. It can also handle fuzzy data. The former type of data allows FARM II to discover multidimensional association rules, whereas the latter data allows some of the patterns to be more easily revealed and expressed. To effectively uncover the hidden associations in the bank-account database, FARM II performs several steps. First, it combines the relational and transactional data together by performing data transformations. Second, it identifies fuzzy attributes and performs fuzzification so that linguistic terms can be used to represent the uncovered patterns. Third, it makes use of an efficient rule-search process that is guided by an objective interestingness measure. This measure is defined in terms of fuzzy confidence and support measures, which reflect the differences in the actual and the expected degrees to which a customer is characterized by different linguistic terms. These steps are described in detail in this paper. With FARM II, fuzzy association rules were obtained that were judged by experts from the bank to be very useful. In particular, they discovered that they had identified some interesting characteristics about the customers who had once used the bank's loan services but then decided later to cease using them. The bank translated what they discovered into actionable items by offering some incentives to retain their existing customers.**

***Index Terms*—Customer relationship management, data mining, fuzzy association rules, rule interestingness measures, transformation functions.**

I. Introduction

WIDESPREAD deregulation, diversification, and globalization have stimulated a dramatic rise in the competition between companies all over the world. To maintain profitability, many companies consider effective customer relationship management (CRM) to be one of the critical factors for success. The central objective of CRM is to maximize the lifetime value of a customer to a company [19]. It has been shown in recent studies (e.g., [12], [20], and [22]) that: 1) existing customers are more profitable than new customers; 2) it costs much more to attract a new customer than it does to retain an existing customer; and 3) retained customers are good candidates for cross selling. It is for these reasons that many companies consider customer retention to be one of their most important business activities.

More than 150 international banks, which are headquartered all over the world, have offices set up in Hong Kong. Due to relaxed interest rate controls, the banks in Hong Kong (local or international) have faced fierce competition from each other. To better serve and retain customers, the loans department of a major international bank, with many branches in Hong Kong, decided recently to look at the use of data mining techniques. The bank's aim was to try to discover hidden patterns in its databases so that it could better understand its customers and design new products to ensure that they are willing to stay with the bank. For the purpose of data mining, the bank decided to look at its bank-account database, which contained data on over 320 000 customers that have used or were using its loan services. More specifically, the bank wanted to look at both the demographic data of the customers and their banking transactions over a period covering the last three months. With these data, the goal was to discover interesting patterns in the data that could provide clues on what incentives it could offer to increase the retention of its customers.

The problem of mining *association rules* was introduced to reveal interesting patterns in data [1]. The mining of association rules was originally defined for transactional data. This was later extended to also handle relational data containing categorical and quantitative data [23]. In its most general form, an association rule is defined for the attributes of a database relation, T. It is an implication of the form $X \Rightarrow Y$, where X and Y are conjunctions of certain conditions. A condition is either $A_i = a_i$, where a_i is a value in the domain of the attribute A_i if A_i is categorical, or $a_i \in [l_i, u_i]$, where l_i and u_i are bounding values in the domain of the attribute A_i if A_i is quantitative. The association rule $X \Rightarrow Y$ holds in T with a certain *support*, which is defined as the percentage of tuples that have the characteristics satisfying X and Y and a certain *confidence*, which is defined as the percentage of tuples that have the characteristics satisfying Y given that they also satisfy X. An associative relationship is usually considered to be interesting if its support and confidence values are greater than some user-specified minimum [1], [2], [18], [21], [23].

An example of an association rule is

$$\begin{aligned}&\text{Marital Status} = \text{Single} \wedge \text{Age} \in [35, 45]\\ &\wedge \text{Account Balance} \in [1000, 2500]\\ &\Rightarrow \text{Loan Balance} \in [10000, 15\,000]\end{aligned}$$

Manuscript received December 31, 2000; revised September 19, 2002. This work was supported in part by The Hong Kong Polytechnic University under Grant A-P209 and Grant G-V918.

The authors are with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: cswhau@comp.polyu.edu.hk; cskcchan@comp.polyu.edu.hk).

Digital Object Identifier 10.1109/TFUZZ.2003.809901

which describes a person who is single, aged between 35 and 45 and with an account balance that is between \$1 000 and \$2 500, as someone who is likely to use a loan that is between \$10 000 and \$15 000. An association rule defined over market basket data has a special form. The antecedent and consequent are conjunctions involving Boolean attributes that take on the value of 1. An example of an association rule that is defined over market basket data is

$$\text{Pizza} = 1 \wedge \text{Chicken Wings} = 1 \Rightarrow \text{Coke} = 1 \wedge \text{Salad} = 1.$$

This rule states that a customer who buys pizza and chicken wings also buys coke and salad.

Although the existing algorithms for mining association rules (e.g., [23]) can be used to identify interesting characteristics of different types of bank customers, they require the domains of the quantitative attributes to be discretized into intervals. These intervals are often difficult to define. In addition, if too much data lies on the boundaries of the intervals, then this could result in very different discoveries in the data that could be both misleading and meaningless. In addition to the need for discretization, there is a requirement for users to provide the thresholds for minimum support and confidence and this also makes the existing techniques difficult to use (e.g., [1], [2], [18], [21], and [23]). If the thresholds are set too high, a user may miss some useful rules, but if the thresholds are set too low, the user may be overwhelmed by too many irrelevant rules [11].

To handle the problems that were given to us by the banking officials, we developed a fuzzy technique for data mining that is called the Fuzzy Association Rule Mining II (FARM II). FARM II employs *linguistic terms* to represent the revealed regularities and exceptions. This linguistic representation is especially useful when the discovered rules are presented to human experts for examination because of its affinity with human knowledge representation. Since our interpretation of the linguistic terms is based on fuzzy-set theory, the association rules that are expressed in these terms are referred to hereinafter as *fuzzy association rules* [3]–[6].

An example of a fuzzy association rule is given as follows:

$$\begin{aligned} &\text{Marital Status} = \text{Single} \wedge \text{Age} = \text{Middle} \\ &\wedge \text{Account Balance} = \text{Small} \\ &\Rightarrow \text{Loan Balance} = \text{Moderate} \end{aligned}$$

where $Single$ is a crisp value, $Middle$ is a linguistic term that is represented by the fuzzy set, $\int_{30}^{40} (1/10)(x-30)/x + \int_{40}^{50} (1/10)(50-x)/x$, $Small$ is a linguistic term that is represented by the fuzzy set, $\int_{0}^{20\,000} 1/x + \int_{20\,000}^{30\,000} (1/10\,000)(30000-x)/x$ and $Moderate$ is a linguistic term that is represented by the fuzzy set

$$\int_{10\,000}^{20\,000} \frac{\frac{1}{10\,000}(x-10\,000)}{x} + \int_{20\,000}^{30\,000} \frac{\frac{1}{10\,000}(30000-x)}{x}.$$

This rule states that a middle-aged person who is single and has a small balance in his/her bank account is likely to use a loan for a moderate amount. When this rule is compared to the association rule involving discrete intervals, the fuzzy association rule is easier for human users to comprehend. In addition to the linguistic representation, the use of fuzzy techniques hides the boundaries of the adjacent intervals of the quantitative attributes. This makes FARM II resilient to noise in the data, such as inaccuracies in the physical measurements of real-life entities. Furthermore, the fact that 0.5 is the fuzziest degree of membership of an element in a fuzzy set provides a new means for FARM II to deal with missing values in databases. Using defuzzification techniques, FARM II allows quantitative values to be inferred when fuzzy association rules are applied to as yet unseen records.

To avoid the need for user-specified thresholds, FARM II utilizes an objective interestingness measure, which is defined in terms of a fuzzy support and confidence measure [3]–[7] that reflects the actual and expected degrees to which a tuple is characterized by different linguistic terms. Unlike other data-mining algorithms (e.g., [1], [2], [18], [21], and [23]), the use of this interestingness measure has the advantage that it does not require any user-specified thresholds.

In addition to dealing with fuzzy data and using an objective interestingness measure, the technique also needs to deal with the problem that is created by the fact that there is more than one database relation. In such a case, the concept of a *universal relation* needs to be used. A universal relation is an imaginary relation that can be used to represent the data that is constructed by logically joining all of the separate tables of a relational database [24]. The use of a universal relation, therefore, makes it possible for the existing data-mining systems [16] to deal with both transactional and relational data. Unfortunately, the construction of universal relations will very likely lead to the introduction of redundant information, which will mislead the rule-discovery process of many data-mining algorithms.

Existing data-mining algorithms (e.g., [1], [2], [18], [21], and [23]) can be made more powerful if they can overcome such a problem. They can also be further improved if they can discover rules that involve attributes that were not originally contained in a database. The ability to do so is essential to the mining of interesting patterns in many different application areas. For example, rules regarding consumers' buying habits at Christmas cannot be discovered if a new attribute of "holiday" has not been considered.

Taking into consideration the need to address these issues, FARM II is equipped with some transformation functions that can be used to deal with both transactional and relational data and the different types of attributes in the databases of a database system so as to construct new relations. To discover the interesting fuzzy association rules that are hidden in these transformed relations, FARM II makes use of an efficient rule-search process that is guided by an objective interestingness measure. This measure is defined in terms of fuzzy confidence and support measures that reflect the differences in the actual and expected degrees to which a tuple is characterized by different linguistic terms.

The rest of this paper is organized as follows. In Section II, we provide a description of how the existing algorithms can be used for the mining of association rules and how fuzzy techniques can be applied to the data-mining process. In Section III,

we describe the bank-account database that was provided by the bank. We then introduce a formalism to handle the union of relational and transactional data in Section IV. The details of FARM II are given in Section V. In this same section, we also present the definition of the linguistic terms and an interestingness measure that can be used for finding the interesting associations that are hidden in databases. In Section VI, we discuss the fuzzy association rules that were discovered by FARM II in the bank-account database. Finally, in Section VII, we conclude this paper with a summary.

II. Related Work

To discover association rules, existing data-mining algorithms [23] require the domains of quantitative attributes to be discretized into intervals. The idea has been proposed in [23] to use *equidepth partitioning* for optimizing a *partial completeness* measure so that the intervals are neither too big nor too small with respect to the set of association rules that are discovered by their data-mining algorithm.

Regardless of how the values of the quantitative attributes are discretized, the intervals might not be concise and meaningful enough for human users to easily obtain nontrivial knowledge from the discovered association rules. *Linguistic summaries*, which were introduced in [25], express knowledge using a linguistic representation that is natural for human users to comprehend. An example of a linguistic summary is the statement, "about half of the people in the database are middle aged." Unfortunately, no algorithm was proposed for generating the linguistic summaries in [25]. Recently, the use of an algorithm for mining association rules for the purpose of linguistic summaries has been studied in [14]. This technique extends AprioriTid [2], which is a well-known algorithm for mining association rules, to handle linguistic terms (fuzzy values). An attribute is replaced by a set of artificial attributes (items) so that a tuple supports a specific item to a certain degree, which is in the range 0 to 1. Given two user-specified thresholds, $threshold_1$ and $threshold_2$, an item or an itemset (i.e., a combination of items) is considered interesting if its *fuzzy support* is greater than $threshold_1$ and it is also less than $threshold_2$. Although this technique is very useful, many users may not be able to set the thresholds appropriately.

In addition to the linguistic summaries, an interactive process for the discovery of top-down summaries, which utilizes *fuzzy is-a hierarchies* as domain knowledge, has been described in [15]. This technique is aimed at discovering a set of *generalized tuples*, such as ⟨technical writer, documentation⟩. In contrast to association rules, which involve implications between different attributes, the generalized tuples only provide summarization on different attributes. The idea of implication has not been taken into consideration and hence these techniques are not developed for the task of rule discovery.

Furthermore, the applicability of fuzzy modeling techniques to data mining has been discussed in [13]. Given a relational table, X and a context variable, A, the *context-sensitive fuzzy clustering* method is aimed at revealing the structure in X in the context of A. Since this method can only manipulate quantitative attributes, the values of any categorical attributes are first encoded into numeric values. The context-sensitive fuzzy clustering method is then applied to the encoded data to induce clusters in the context of A. Although the encoding technique allows this method to deal with categorical attributes, the distances between the encoded numeric values, which do not possess any meaning in the original categorical attributes, are used to induce the clusters. Therefore, the associations that are concerned with these attributes, which are discovered by the context-sensitive fuzzy clustering method, may be misleading.

In addition to the use of intervals to represent the revealed associations that are concerned with quantitative attributes, many existing algorithms (e.g., [1], [2], [18], [21], and [23]) are based on using support and confidence measures to discover association rules. Given an association rule $X \Rightarrow Y$, its support $\sup(X \Rightarrow Y)$ and confidence $\mathrm{conf}(X \Rightarrow Y)$ are defined as

$$\sup(X \Rightarrow Y) = \Pr(X \wedge Y)$$
$$\mathrm{conf}(X \Rightarrow Y) = \Pr(Y|X) = \frac{\Pr(X \wedge Y)}{\Pr(X)}.$$

Data-mining algorithms, such as [1], [2], [18], [21], and [23] are aimed at finding association rules with support and confidence values that are greater than a user-specified minimum support and minimum confidence. Such an approach has a weakness in that many users do not have any idea what values to use for the thresholds. If thresholds are set too high, a user may miss some useful rules, but if they are set too low, the user may be overwhelmed by many irrelevant rules [11].

III. Bank-Account Database

The bank-account database was provided by a bank in Hong Kong. The bank does not want to be identified in our paper because customer attrition rates are confidential. The bank-account database is stored in an Oracle database, which is one of the most popular *relational database management systems* [9]. It is composed of three relations, namely, CUSTOMER, ACCOUNT, and TRANSACTION. Of these relations, CUSTOMER and ACCOUNT contain relational data, whereas TRANSACTION contains transactional data. Specifically, the bank maintains a tuple in CUSTOMER for each customer (e.g., sex, age, marital status, etc.), a tuple in ACCOUNT for each account owned by a customer (e.g., account type, loan amount limit, etc.) and a tuple in TRANSACTION for each transaction made by a customer on one of his/her accounts (e.g., cash deposit, cash withdrawal, etc.). A customer can have one or more accounts and an account can have one or more transactions. Accordingly, a tuple in CUSTOMER is associated with one or more tuples in ACCOUNT and a tuple in ACCOUNT is associated with one or more tuples in TRANSACTION.

Fig. 1 shows the schema of the bank-account database. Since each relation in the bank-account database contains many attributes, we only show a subset of these attributes in Fig. 1.

It is important to note that a relation in a relational database may contain relational data or transactional data. The entity that a relation represents is what makes it either relational or transactional. In a relation that contains transactional data, each tuple (transaction record) represents a business transaction. Specifically, a transaction record represents a debit or credit transaction

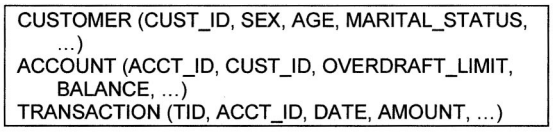

Fig. 1. Schema of the bank-account database.

TABLE I
SUMMARY OF THE BANK-ACCOUNT DATABASE

Relation	No. of Attributes	No. of Tuples
CUSTOMER	48	320,000
ACCOUNT	42	558,431
TRANSACTION	37	1,746,996

in the bank-account database. A transaction record, therefore, has to store the account involved in the transaction, the date of the transaction, the amount of the transaction, etc.

In the bank-account database, CUSTOMER contained data for 320 000 customers. Each customer had opened one or more bank accounts for the purpose of using loan services, such as a mortgage loan, a tax payment loan, etc. In this data, 99.5% of all customers were from Hong Kong and the remaining 0.5% of customers were from other countries (for example, Singapore, Taiwan, France, the United States, etc.). The total loan balance of all customers in the bank-account database was H.K. $11.8 billion in November 1999.

The bank-account database was extracted from the time interval of September 1999 through to November 1999. The task was to reveal the interesting associative relationships in the data so as to better serve and retain customers. These relationships are represented in the form of fuzzy association rules. Table I gives a summary of the bank-account database.

IV. HANDLING OF RELATIONAL AND TRANSACTIONAL DATA

Together with a domain expert from the bank, we have identified 102 variables, which are associated with each customer, which might affect his/her satisfaction concerning the loan services. Some of these variables can be extracted directly from the bank-account database whereas some of them are not contained in the original data and they are produced by the transformation functions. To handle the union of both relational and transactional data, we have defined a set of *transformation functions* to operate on the relations of CUSTOMER, ACCOUNT, and TRANSACTION. The application of these transformation functions to the bank-account database results in a set of *transformed data*. To manage the data-mining process effectively, the transformed data is stored in a relation in the Oracle database. We refer to this relation as the *transformed relation*. The use of transformation functions to handle the union of relational and transactional data has been described informally in [6]. More formally, we define the problem formalism.

Let $A_{i1}, A_{i2}, \ldots, A_{iK_i}$, for $i = 1, 2, \ldots, I$, be the attributes of the real-world entities represented by the relational tables, R_i, $i = 1, 2, \ldots, I$, respectively. Let the domain of A_{ik}, $k = 1, 2, \ldots, K_i$, be represented by $\mathrm{dom}(A_{ik}) = \{a_{ik}^{(1)}, a_{ik}^{(2)}, \ldots, a_{ik}^{(m_{ik})}\}$, $i = 1, 2, \ldots, I$, $k = 1, 2, \ldots, K_i$. In other words, $R_i \subseteq \mathrm{dom}(A_{i1}) \times \mathrm{dom}(A_{i2}) \times \cdots \times \mathrm{dom}(A_{iK_i})$. For any R_i, we use $\mathcal{A}_{R_i}$ to denote the set of attributes of R_i, that is, $\mathcal{A}_{R_i} = \{A_{i1}, A_{i2}, \ldots, A_{iK_i}\}$. The primary key of R_i, which is composed of one or more attributes and is associated with each tuple in a relation, is represented by $\mathcal{K}_i \subseteq \{A_{i1}, A_{i2}, \ldots, A_{iK_i}\}$.

For a database system, a set of transaction records can be denoted by T_j, $j = 1, 2, \ldots, J$, where each T_j is characterized by a set of attributes, which are denoted by $A_{j1}, A_{j2}, \ldots, A_{jL_j}$ and has a unique transaction identifier TID_j. In other words, $T_j \subseteq TID_j \times \mathrm{dom}(A_{j1}) \times \mathrm{dom}(A_{j2}) \times \cdots \times \mathrm{dom}(A_{jL_j})$.

The definition of the transaction records, which is used here, follows the idea presented in [23]. It is a generalization of the definition of the transactions used in many of the existing algorithms for mining association rules (e.g., [1], [2], [18], and [21]). In these algorithms, a transaction, t, is typically defined as $\langle TID, J' \rangle$, where TID is the transaction identifier of t, $J' \subseteq J$ and $J = \{item_1, \ldots, item_n\}$ is a set of items. To store transactions of this kind in a relational database, one can define a relation, T (TID, $A_1, A_2, \ldots, A_n$), where TID is a transaction identifier. For any $t \in T$, $t[A_k] = 1$ if t contains $item_k$; otherwise, $t[A_k] = 0$, for $k = 1, 2, \ldots n$. This is a special case of the definition of the transaction records used in this paper. In addition to handling items, our definition can also handle categorical and quantitative attributes. This allows richer semantics to be captured in the transaction records as compared to the definition that is only concerned with items (e.g., [1], [2], [18], and [21]).

In a database system, there are some one-to-many relationships between the records in R_i, $i = 1, 2, \ldots I$ and those in T_j, $j = 1, 2, \ldots, J$. For example, the bank-account database contains a set of relational tables (i.e., CUSTOMER and ACCOUNT) that contain background information about each customer and a transactional table (i.e., TRANSACTION) that contains details of each transaction made by a customer. The relational data are related to the transactional data by some one-to-many relationships in such a way that we can find $\mathcal{K}_i$, which is the primary key of R_i, in $\{A_{j1}, A_{j2}, \ldots, A_{jL_j}\}$, which can be used as a foreign key to provide a reference to the corresponding tuple in R_i, $i = 1, 2, \ldots, I$.

Given R_i and T_j, to deal with both relational and transactional data and to consider additional attributes that were not originally in the database, we propose the concept of using transformation functions that are defined on the original attributes in R_i and T_j. Let $f_1, f_2, \ldots, f_p$ be a set of transformation functions, where

$$f_p : A_{p1} \times A_{p2} \times \cdots \times A_{pr_p} \rightarrow A'_p \qquad p = 1, 2, \ldots, r_P$$

where $r_p \geq 1$ and

$$A_{pu} \in \left(\bigcup_{i=1}^{I} \mathcal{A}_{R_i} \right) \bigcup \left(\bigcup_{j=1}^{J} \mathcal{A}_{T_j} \right), \qquad u = 1, 2, \ldots, r_p.$$

We can construct a new relation R' that contains both the original attributes in R_i and T_j and the transformed attributes that are obtained by applying appropriate transformation functions. Let R' be composed of attributes, $A'_1, A'_2, \ldots, A'_n$, that

is, $R' \subseteq \mathrm{dom}(A'_1) \times \mathrm{dom}(A'_2) \times \cdots \times \mathrm{dom}(A'_n)$, where A'_u, $u = 1, 2, \ldots, n$, can be any attribute in R_i, $i = 1, 2, \ldots, I$, or $T_j, j = 1, 2, \ldots, J$, or any transformed attribute. In other words

$$A'_u \in \left(\bigcup_{i=1}^{I} \mathcal{A}_{R_i}\right) \bigcup \left(\bigcup_{j=1}^{J} \mathcal{A}_{T_j}\right) \bigcup \left(\bigcup_{p=1}^{P} f_p\left(A_{p1}, \ldots, A_{pr_p}\right)\right).$$

Instead of performing data mining on the original R_i and T_j, we perform data mining on R'.

Given a database, different kinds of transformation functions can be performed. They include *logical*, *arithmetic*, *substring*, and *discretization* functions. Depending on the type of attribute, one or more of these functions can be applied to the attribute. We provide the definitions of each type of transformation function in the following sections.

A. Logical Functions

The logical functions are composed of a combination of logical operators, such as NOT, AND, OR, etc. A logical function can take one or more attributes as aguments. Let $f_1, f_2, \ldots, f_n$ be a set of functions so that

$$f_j(a_1, a_2, \ldots, a_r) = (a_1 = c_{j1}) \oplus (a_2 = c_{j2}) \otimes \cdots \Theta (a_r = c_{jr}) \quad j = 1, 2, \ldots, n$$

where $a_i \in \mathrm{dom}(A_i)$, $c_{ji} \in \mathrm{dom}(A_i)$

$$A_i \in \left(\bigcup_{i=1}^{I} \mathcal{A}_{R_i}\right) \bigcup \left(\bigcup_{j=1}^{J} \mathcal{A}_{T_j}\right) \quad i = 1, 2, \ldots, r$$

and

$$\oplus, \otimes, \ldots, \Theta \in \{\text{AND, OR, NOT XOR, NAND, NOR}\}.$$

A generic way of utilizing these functions is to construct a logical function, f, defined in terms of $f_1, f_2, \ldots, f_n$, as follows:

$$f(a_1, a_2, \ldots, a_r) = \begin{cases} 1 & \text{if } f_1(a_1, a_2, \ldots, a_r) = \text{true} \\ 2 & \text{else if } f_2(a_1, a_2, \ldots, a_r) = \text{true} \\ \cdot & \\ \cdot & \\ \cdot & \\ n & \text{else if } f_n(a_1, a_2, \ldots, a_r) = \text{true} \end{cases}$$

where $a_i \in \mathrm{dom}(A_i)$, $A_i \in \left(\bigcup_{i=1}^{I} \mathcal{A}_{R_i}\right) \bigcup \left(\bigcup_{j=1}^{J} \mathcal{A}_{T_j}\right)$, $i = 1, 2, \ldots, r$.

In the case where none of $f_1, f_2, \ldots, f_n$ are evaluated as being true, the logical function, f, produces an unknown value as its output. Furthermore, if the value of any attribute, A_i, $i = 1, 2, \ldots, r$, of a tuple is unknown, the logical function, f, also produces an unknown value as its output.

B. Arithmetic Functions

The arithmetic functions can involve addition, subtraction, multiplication and division. An arithmetic function takes a set of attributes as its argument and produces an attribute that has a type of real or integer. Let $f_1, f_2, \ldots, f_r$ be operations in relational algebra, each of which produces an integer or a real number. The arithmetic function f is defined as follows:

$$f(a_1, a_2, \ldots, a_r) = f_1(a_1) \oplus f_2(a_2) \otimes \cdots \Theta f_r(a_r)$$

where

$$a_i \in \mathrm{dom}(A_i),\ A_i \in \left(\bigcup_{i=1}^{I} \mathcal{A}_{R_i}\right) \bigcup \left(\bigcup_{j=1}^{J} \mathcal{A}_{T_j}\right) \quad i = 1, 2, \ldots, r$$

and $\oplus, \otimes \cdots \Theta \in \{+, -, \times, \div\}$.

In the case where the value of any attribute, A_i, $i = 1, 2, \ldots, r$, of a tuple is unknown, the arithmetic function, f, produces an unknown value as its output.

C. Substring Functions

The substring functions extract a specific portion of a given attribute. Let the given attribute, A, be a string of s characters. For any $a \in \mathrm{dom}(A)$, we use $a[i]$ to denote the i-th character of a. The substring function, f, is defined as follows:

$$f(a) = a[l]a[l+1] \ldots a[u]$$

where

$$a \in \mathrm{dom}(A),\ A \in \left(\bigcup_{i=1}^{I} \mathcal{A}_{R_i}\right) \bigcup \left(\bigcup_{j=1}^{J} \mathcal{A}_{T_j}\right)$$

and

$$1 \le l \le u \le s.$$

In the case where the value of an attribute A of a tuple is unknown, the substring function f produces an unknown value as its output.

D. Discretization Functions

The discretization functions discretize the domain of any numeric attribute into a finite number of intervals. Let f be the discretization function that creates r intervals. We use u_i to denote the upper limit of the ith interval, for $i = 1, 2, \ldots, r-1$. Then, f is defined as follows:

$$f(a) = \begin{cases} 1, & \text{if } a \le u_1 \\ 2, & \text{if } u_1 < a \le u_2 \\ \cdot & \\ \cdot & \\ \cdot & \\ r-1, & \text{if } u_{r-2} < a \le u_{r-1} \\ r, & \text{if } a > u_{r-1} \end{cases}$$

where

$$a \in \mathrm{dom}(A),\ A \in \left(\bigcup_{i=1}^{I} \mathcal{A}_{R_i}\right) \bigcup \left(\bigcup_{j=1}^{J} \mathcal{A}_{T_j}\right).$$

In the case where the value of an attribute A of a tuple is unknown, the discretization function, f, produces an unknown value as its output.

The boundaries of the intervals can be specified by users or determined automatically by using various algorithms (e.g., [8]). One of the commonly used algorithms involves discretizing the attribute into equal intervals. Another popular algorithm involves discretizing the attribute into intervals in such a way that the number of tuples in each interval is the same. As a result, each tuple has an equal probability of lying in any interval.

E. Transformation Functions Defined Over the Bank-Account Database

In this section, we describe how we can construct a transformed relation, R(T_ACCT_TYPE, T_AMOUNT, T_NATIONALITY, ...), using the transformation functions. To obtain the transformed relation, we (including a domain expert from the bank) have defined 102 transformation functions in total. From the 102 transformation functions, in this section, we present three of them as an illustration. Consider the attribute ACCOUNT[ACCT_ID]. The first digit of this attribute denotes the type of account. Let us suppose that it is a personal account if this digit is 1 and that it is a corporate account if this digit is 2. There exists a transformation function, f_1, defined as

$$f1(s) = \text{first\_digit\_of}(s)$$

where first_digit_of(s) returns the first digit of string s. The *transformed attribute* T_ACCT_TYPE is produced by applying f_1(ACCOUNT[ACCT_ID]) to every tuple in ACCOUNT, which is an example of the substring functions that are defined in Section IV-C.

To compute the average amount in the customers' accounts, we make use of another transformation function, f_2, which is defined as follows:

$$f_2(id) = \frac{\sum_{t \in \sigma_{\text{CUST\_ID}=id}(\text{ACCOUNT})} t[\text{AMOUNT}]}{|\sigma_{\text{CUST\_ID}=id}(\text{ACCOUNT})|}$$

where σ denotes the SELECT operation from *relational algebra* and $|S|$ denotes the cardinality of set S. The function, f_2, is an example of the arithmetic functions that are defined in Section IV-B. The transformed attribute, T_AMOUNT, is produced by applying the function f_2(CUSTOMER[CUST_ID]) to every tuple in CUSTOMER.

The nationality of the customers can be grouped into different geographical regions for the purpose of discovering more meaningful rules. Such a grouping is performed by a transformation function, f_3, which is defined as the equation shown at the bottom of the page.

This function f_3 is an example of the logical functions that are defined in Section IV-A. The transformed attribute, T_NATIONALITY, is produced by applying the function f_3(CUSTOMER[NATIONALITY]) to every tuple in CUSTOMER.

By applying the transformation functions to the bank-account database, we have obtained the required transformed relation. There are 102 attributes in the transformed relation. Among the 102 transformed attributes, six are categorical and 96 are quantitative. Instead of performing data mining on the original data, we discover interesting associations from the transformed data.

V. FARM II for Mining Fuzzy Association Rules

In this section, we describe a novel algorithm, called FARM II, which makes use of linguistic terms to represent the regularities and exceptions that are discovered in databases. Furthermore, FARM II employs an objective interestingness measure to identify the interesting associations among the attributes of the database. The definition of the linguistic variables and the linguistic terms is presented in Section V-A. In Section V-B, we describe how the interesting associations can be identified. The formation of the fuzzy association rules to represent the interesting associations is described in Section V-C. In this same section, a confidence measure is defined to provide a means for representing the uncertainty that is associated with the fuzzy association rules. In Section V-D, we provide the details of FARM II. In Section V-E, we describe how the previously unknown values can be inferred using the fuzzy association rules.

A. Linguistic Variables and Linguistic Terms

Given a transformed relation, R, each tuple, t, in R consists of a set of attributes, $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$, where $A_1, A_2, \ldots, A_n$ can be quantitative or categorical. For any tuple, $t \in R$, $t[A_i]$ denotes the value a_i in t for attribute $A_i \in \mathcal{A}$. Let $\mathcal{L} = \{L_1, L_2, \ldots, L_n\}$ be a set of linguistic variables such that $L_i \in \mathcal{L}$ represents $A_i \in \mathcal{A}$.

For any quantitative attribute, $A_i \in \mathcal{A}$, let $\text{dom}(A_i) = [l_i, u_i] \subseteq \Re$ denote the domain of the attribute. A_i is represented by a linguistic variable, L_i, whose value is a linguistic term in $T(L_i) = \{l_{ij} | j = 1, 2, \ldots, s_i\}$ where l_{ij} is a linguistic term characterized by a fuzzy set, F_{ij}, that is defined on $\text{dom}(A_i)$ and whose membership function is $\mu_{F_{ij}}$ so that

$$\mu_{F_{ij}} : \text{dom}(A_i) \rightarrow [0, 1]$$

The fuzzy sets F_{ij}, $j = 1, 2, \ldots, s_i$, are then represented by

$$F_{ij} = \begin{cases} \sum_{\text{dom}(A_i)} \frac{\mu_{F_{ij}}(a_i)}{a_i}, & \text{if } A_i \text{ is discrete} \\ \int_{\text{dom}(A_i)} \frac{\mu_{F_{ij}}(a_i)}{a_i}, & \text{if } A_i \text{ is continuous} \end{cases} \quad (1)$$

where $a_i \in \text{dom}(A_i)$. The degree of compatibility of $a_i \in \text{dom}(A_i)$ with linguistic term l_{ij} is given by $\mu_{F_{ij}}(a_i)$.

$$f_3(n) = \begin{cases} \text{Asian}, & \text{if } n = \text{Chinese or Japanese or } \ldots \text{ or Korean} \\ \text{European}, & \text{else if } n = \text{UK or French or } \ldots \text{ or German} \\ \text{North American}, & \text{else if } n = \text{US or Canadian.} \end{cases}$$

For any categorical attribute, $A_i \in \mathcal{A}$, let $\mathrm{dom}(A_i) = \{a_{i1}, a_{i2}, \ldots, a_{im_j}\}$ denote the domain of A_i. A_i is represented by linguistic variable L_i whose value is a linguistic term in $T(L_i) = \{l_{ij} | j = 1, 2, \ldots, m_i\}$ where l_{ij} is a linguistic term characterized by a fuzzy set, F_{ij}, so that

$$F_{ij} = \sum_{\mathrm{dom}(A_i)} \frac{\mu_{F_{ij}}(a_i)}{a_i} \tag{2}$$

where $a_i \in \mathrm{dom}(A_i)$. The degree of compatibility of $a_i \in \mathrm{dom}(A_i)$ with linguistic term l_{ij} is given by $\mu_{F_{ij}}(a_i)$.

In addition to handling categorical and quantitative attributes in a uniform fashion, the use of linguistic terms to represent categorical attributes also allows the fuzzy nature of some real-world entities to be easily captured. Interested readers are referred to [17] and [26] for the details of the linguistic variables, linguistic terms, fuzzy sets and membership functions.

Using the aforementioned technique, the original attributes, $\mathcal{A}$, are represented by a set of linguistic variables, $\mathcal{L} = \{L_i | i = 1, 2, \ldots, n\}$. These linguistic variables are associated with a set of linguistic terms, $l = \{l_{ij} | i = 1, 2, \ldots, n, j = 1, 2, \ldots, s_i\}$. These linguistic terms are, in turn, characterized by a set of fuzzy sets, $\mathcal{F} = \{F_{ij} | i = 1, 2, \ldots, n, j = 1, 2, \ldots, s_i\}$. Given a tuple $t \in R$ and a linguistic term $l_{ij} \in l$, which is characterized by a fuzzy set $F_{ij} \in \mathcal{F}$, the degree of membership of the values in t with respect to F_{ij} is given by $\mu_{F_{ij}}(t[A_i])$. The degree to which t is characterized by l_{ij}, $\lambda_{l_{ij}}(t)$, is defined as follows:

$$\lambda_{l_{ij}}(t) = \mu_{F_{ij}}(t[A_i]). \tag{3}$$

If $\lambda_{l_{ij}}(t) = 1$, t is completely characterized by the linguistic term l_{ij}. If $\lambda_{l_{ij}}(t) = 0$, t is undoubtedly not characterized by the linguistic term l_{ij}. If $0 < \lambda_{l_{ij}}(t) < 1$, t is partially characterized by the linguistic term l_{ij}. In the case where $t[A_i]$ is unknown, $\lambda_{l_{ij}}(t) = 0.5$, which indicates that there is no information available concerning whether t is or is not characterized by the linguistic term l_{ij}.

It is important to note that t can also be characterized by more than one linguistic term. Let φ be a subset of integers so that $\varphi = \{i_1, i_2, \ldots, i_h\}$ where $\varphi \subseteq \{1, 2, \ldots, n\}$ and $|\varphi| = h \geq 1$. We also suppose that $\mathcal{A}\varphi$ is a subset of $\mathcal{A}$ so that $\mathcal{A}\varphi = \{A_i | i \in \varphi\}$. Given any $\mathcal{A}\varphi$, it is associated with a set of linguistic terms, $T(L\varphi) = \{l\varphi_j | j = 1, 2, \ldots, s_\varphi = \prod_{i \in \varphi} s_i\}$ where $l_{\varphi j}$ is represented by a fuzzy set, $F_{\varphi j}$, so that $F_{\varphi j} = F_{i_1 j_1} \cap F_{i_2 j_2} \cap \cdots \cap F_{i_h j_h}$, $i_k \in \varphi$, $j_k \in s_{i_k}$. The degree to which t is characterized by the term $l_{\varphi j}$, $\lambda_{l_{\varphi j}}(t)$, is defined as follows:

$$\lambda_{l_{\varphi j}}(t) = \min\Big(\mu_{F_{i_1 j_1}}(t[A_{i_1}]) \; \mu_{F_{i_2 j_2}}(t[A_{i_2}]), \ldots, \mu_{F_{i_h j_h}}(t[A_{i_h}])\Big). \tag{4}$$

Based on the linguistic terms, we can apply FARM II to discover the fuzzy association rules, which are represented in a manner that is natural for human users to understand.

B. Identification of Interesting Associations Between Linguistic Terms

The *fuzzy support* of a linguistic term, $l_{\varphi k}$, is represented by $f\mathrm{sup}(l_{\varphi k})$ and it is defined as follows:

$$f\mathrm{sup}(l_{\varphi k}) = \frac{\sum_{t \in R} \lambda_{l_{\varphi k}}(t)}{\sum_{t \in R} \sum_{j=1}^{s_\varphi} \lambda_{l_{\varphi j}}(t)}. \tag{5}$$

The fuzzy support of the linguistic term $l_{\varphi k}$, $fsup(l_{\varphi k})$, can be considered as being the probability that a tuple is characterized by $l_{\varphi k}$.

In the rest of this paper, the association between a linguistic term, $l_{\varphi k}$ and another linguistic term, l_{pq}, is expressed as $l_{\varphi k} \rightarrow l_{pq}$. The fuzzy support for the association $l_{\varphi_k} \rightarrow l_{pq}$, $f\mathrm{sup}(l_{\varphi k} \rightarrow l_{pq})$, is given by

$$f\mathrm{sup}(l_{\varphi k} \rightarrow l_{pq}) = \frac{\sum_{t \in R} \min\left(\lambda_{l_{\varphi k}}(t), \lambda_{l_{pq}}(t)\right)}{\sum_{t \in R} \sum_{j=1}^{s_\varphi} \sum_{u=1}^{s_p} \min\left(\lambda_{l_{\varphi j}}(t), \lambda_{l_{pu}}(t)\right)}. \tag{6}$$

The *fuzzy confidence* of the association $l_{\varphi k} \rightarrow l_{pq}$, is represented by $fconf(l_{\varphi k} \rightarrow l_{pq})$ and this is calculated by

$$fconf(l_{\varphi k} \rightarrow l_{pq}) = \frac{f\mathrm{sup}(l_{\varphi k} \rightarrow l_{pq})}{f\mathrm{sup}(l_{\varphi k})}. \tag{7}$$

Intuitively, the fuzzy support for $l_{\varphi k} \rightarrow l_{pq}$, $f\mathrm{sup}(l_{\varphi k} \rightarrow l_{pq})$, can be considered as being the probability that a tuple is characterized by $l_{\varphi k}$ and l_{pq} whereas the fuzzy confidence of $l_{\varphi k} \rightarrow l_{pq}$, $fconf(l_{\varphi k} \rightarrow l_{pq})$, can be considered as being the probability that a tuple is characterized by l_{pq} given that it is also characterized by $l_{\varphi k}$.

To decide whether an association, $l_{\varphi k} \rightarrow l_{pq}$, is interesting, we determine whether the difference between $fconf(l_{\varphi k} \rightarrow l_{pq})$ and $f\mathrm{sup}(l_{pq})$ is significant. The significance of the difference can be objectively evaluated using an objective interestingness measure, $d(l_{\varphi k} \rightarrow l_{pq})$. This is defined in terms of fuzzy confidence and support measures [3]–[7] that reflect the differences in the actual and expected degrees to which a tuple is characterized by different linguistic terms. The objective interestingness measure, $d(l_{\varphi k} \rightarrow l_{pq})$, is defined as follows:

$$d(l_{\varphi k} \rightarrow l_{pq}) = \frac{z(l_{\varphi k} \rightarrow l_{pq})}{\sqrt{\gamma(l_{\varphi k} \rightarrow l_{pq})}} \tag{8}$$

where

$$z(l_{\varphi k} \rightarrow l_{pq}) = \frac{f\mathrm{sup}(l_{\varphi k} \rightarrow l_{pq}) - e(l_{\varphi k} \rightarrow l_{pq})}{\sqrt{e(l_{\varphi k} \rightarrow l_{pq})}} \tag{9}$$

$$e(l_{\varphi k} \rightarrow l_{pq}) = f\mathrm{sup}(l_{\varphi k}) \times f\mathrm{sup}(l_{pq}) \times \sum_{t \in R} \sum_{j=1}^{s_\varphi} \sum_{u=1}^{s_p} \min\left(\lambda_{l_{\varphi j}}(t), \lambda_{l_{pu}}(t)\right) \tag{10}$$

and

$$\gamma(l_{\varphi k} \rightarrow l_{pq}) = (1 - f\mathrm{sup}(l_{\varphi k}))(1 - f\mathrm{sup}(l_{pq})). \tag{11}$$

If $d(l_{\varphi k} \rightarrow l_{pq}) > 1.96$ (i.e., the 95th percentile of the normal distribution), we can conclude that the discrepancy between $f\text{conf}(l_{\varphi k} \rightarrow l_{pq})$ and $f\text{sup}(l_{pq})$ is significantly different and, hence, $l_{\varphi k} \rightarrow l_{pq}$ is interesting. Specifically, if this condition is satisfied, the presence of $l_{\varphi k}$ implies the presence of l_{pq}. In other words, it is more *likely* for a tuple to be characterized by both $l_{\varphi k}$ and l_{pq}.

C. Formation of Fuzzy Association Rules

A first-order fuzzy association rule can be defined as a rule involving one linguistic term in its antecedent. A second-order fuzzy association rule can be defined as a rule involving two linguistic terms in its antecedent. A third-order fuzzy association rule can be defined as a rule involving three linguistic terms in its antecedent and so on for other higher orders. Given that $l_{\varphi k} \rightarrow l_{pq}$ is interesting, we can form the following fuzzy association rule:

$$l_{\varphi k} \Rightarrow l_{pq}\,[w\,(l_{\varphi k} \Rightarrow l_{pq})]$$

where

$$w\,(l_{\varphi k} \Rightarrow l_{pq}) = \log \frac{f\text{sup}\,(l_{\varphi k} \rightarrow l_{pq})}{f\text{sup}\left(\bigcup_{j \neq q} l_{\varphi k} \rightarrow l_{pj}\right)}. \tag{12}$$

This last term is a confidence measure that represents the uncertainty associated with $l_{\varphi k} \Rightarrow l_{pq}$. Intuitively, $w(l_{\varphi k} \Rightarrow l_{pq})$ can be interpreted as being a measure of the difference in the gain in information when a tuple that is characterized by $l_{\varphi k}$ is also characterized by l_{pq} as opposed to being characterized by other linguistic terms.

Since $l_{\varphi k}$ is defined by a set of linguistic terms, $l_{i_1 j_1}$, $l_{i_2 j_2}, \ldots, l_{i_h j_h} \in T(L_\varphi)$, we have a high-order fuzzy association rule

$$\begin{aligned} L_{i_1} &= l_{i_1 j_1} \wedge L_{i_2} = l_{i_2 j_2} \wedge \cdots \wedge L_{i_h} = l_{i_h j_h} \\ \Rightarrow L_p &= l_{pq}\,[w\,(l_{\varphi k} \Rightarrow l_{pq})] \end{aligned}$$

where $i_1, i_2, \ldots, i_h \in \varphi$.

D. FARM II in Detail

To discover the high-order fuzzy association rules, FARM II makes use of a heuristic in which the association between $l_{\varphi' k}$ where $\varphi' = \varphi_1 \cup \varphi_2$ and l_{pq} is considered to be more likely to be interesting if the association between $l_{\varphi_1 k}$ and l_{pq} and the association between $l_{\varphi_2 k}$ and l_{pq} are interesting. Based on such a heuristic, FARM II evaluates the interestingness of the associations between different combinations of linguistic terms only in lower order association rules. This approach can effectively prevent an exhaustive search for the interesting associations involving all combinations of the linguistic terms.

FARM II starts the data-mining process by finding a set of first-order fuzzy association rules using the objective interestingness measure (introduced in Section V-B). After these rules are discovered, they are stored in R_1. The rules in R_1 are then used to generate second-order rules, which are, in turn, stored in R_2. The rules in R_2 are then used to generate third-order rules, which are stored in R_3 and so on for fourth and higher orders. FARM II iterates until no higher-order association rule is found. The details of the algorithm are given in Fig. 2.

```
R1 ← {lik ⇒ lpq [w(lik ⇒ lpq)] | i ≠ p
    and d(lik → lpq) > 1.96};
h ← 2;
while |Rh-1| ≠ φ do
begin
    C ← {each linguistic term in the antecedent of r
        | r ∈ Rh-1};
    forall lφk composed of h linguistic terms in C do
    begin
        forall lpq, q = 1, 2, …, sp, do
        begin
            if d(lφk → lpq) > 1.96 then
                Rh ← Rh ∪ {lφk ⇒ lpq [w(Lφk ⇒ Lpq)]};
        end
    end
    h ← h + 1;
end
R = ⋃h Rh ;
```

Fig. 2. Algorithm of FARM II.

FARM II employs the objective interestingness measure (described in Section V-B) to determine whether relationship $l_{\varphi k} \rightarrow l_{pq}$ is interesting. If $l_{\varphi k} \rightarrow l_{pq}$ is identified as being interesting, a rule is then generated, $l_{\varphi k} \Rightarrow l_{pq}$, whose uncertainty is represented by the confidence measure that is defined in Section V-C. All generated rules are stored in $\mathcal{R}$, which is used later for inference or for human users to examine.

E. Inferring Previously Unknown Values Using Fuzzy Association Rules

Using the discovered fuzzy association rules, FARM II is able to predict the values of some of the characteristics of previously unseen records. The results can be quantitative or categorical, depending on the nature of the attributes whose values are to be predicted. Unlike other classification techniques, which classify records into distinct classes, FARM II allows quantitative values to be inferred from fuzzy association rules.

Given a tuple $t \in \text{dom}(A_1) \times \cdots \times \text{dom}(A_p) \times \cdots \times \text{dom}(A_n)$, let t be characterized by n attribute values, $\alpha_1, \ldots, \alpha_p, \ldots, \alpha_n$, where α_p is the value that is to be predicted. Let l_p be a linguistic term with a domain of $T(L_p)$. The value of α_p is determined according to l_p. To predict the correct value of α_p, FARM II searches the discovered rules in the transformed data. If some attribute value, say $\alpha_j, j \neq p$, of t is characterized by the linguistic term in the antecedent of a rule that implies l_{pq}, then it can be considered as providing some confidence that the value of l_p should be assigned to l_{pq}. By repeating this procedure, that is, by matching each attribute value of t against the rules, FARM II can determine the value of l_p by computing the total confidence measure.

Each of the attributes of t may or may not provide a contribution to the total confidence measure and those that do may support the assignment of different values. Therefore, the different contributions to the total confidence measure are measured quantitatively and then combined for comparison in order to find the most suitable value of l_p. For any combination of the attribute values, α_φ, $p \notin \varphi$, of t, it is characterized by a linguistic term, $l_{\varphi k}$, to a degree of compatibility, $\lambda_{l_{\varphi k}}(t)$, for each

$k \in \{1, 2, \ldots, s_\varphi\}$. Given the rules that imply the assignment of l_{pq}, $l_{\varphi k} \Rightarrow l_{pq}[w(l_{\varphi k} \Rightarrow l_{pq})]$, for all $k \in \zeta \subseteq \{1, 2, \ldots, s_\varphi\}$, the confidence provided by α_φ for such an assignment is given by

$$w_{\mathcal{L}_{pq}\alpha_\varphi} = \sum_{k \in \zeta} w(\mathcal{L}_{\varphi k} \Rightarrow \mathcal{L}_{pq}) \times \lambda_{\varphi k}(t). \tag{13}$$

Suppose that, among the $n-1$ attribute values excluding α_p, only some combinations of them, $\alpha_{[1]}, \ldots, \alpha_{[j]}, \ldots, \alpha_{[\beta]}$, where $\alpha_{[j]} = \{\alpha_i | i \in \{1, 2, \ldots, n\} - \{p\}\}$, are found to match one or more rules. Then, the total confidence measure for assigning the value of l_p to l_{pq} is given by

$$w_q = \sum_{j=1}^{\beta} w_{l_{pq}\alpha_{[j]}}. \tag{14}$$

In the case where A_p is categorical, l_p is assigned to l_{pc} if

$$w_c > w_g, g = 1, 2, \ldots, s'_p \text{ and } g \neq c \tag{15}$$

where $s'_p (\leq s_p)$ denotes the number of linguistic terms that are implied by the rules and α_p is, therefore, assigned to $a_{pc} \in \mathrm{dom}(A_p)$.

If A_p is quantitative, a new method is used to assign an appropriate value to α_p. Given the linguistic terms, $l_{p1}, l_{p2}, \ldots, l_{ps_p}$ and their total confidence measures $w_{p1}, w_{p2}, \ldots, w_{ps_p}$, let $\mu'_{F_{pu}}(a_p)$ be the weighted degree of membership of $a_p \in \mathrm{dom}(A_p)$ to the fuzzy set F_{pu}, $u \in \{1, 2, \ldots, s_p\}$. The value of $\mu'_{F_{pu}}(a_p)$ is given by

$$\mu'_{F_{pu}}(a_p) = w_u \cdot \mu_{F_{pu}}(a_p) \tag{16}$$

where $a_p \in \mathrm{dom}(A_p)$ and $u = 1, 2, \ldots, s_p$. The predicted value, α, is then defined as

$$\alpha = \frac{\int_{\mathrm{dom}(A_p)} \mu'_{F_{p1} \cup F_{p2} \cup \cdots \cup F_{ps_p}}(a_p) \cdot a_p da_p}{\int_{\mathrm{dom}(A_p)} \mu'_{F_{p1} \cup F_{p2} \cup \cdots \cup F_{ps_p}}(a_p)\, da_p} \tag{17}$$

where $\mu'_{X \cup Y}(a) = \max(\mu'_X(a), \mu'_Y(a))$ for any fuzzy sets X and Y. This prediction α provides an appropriate value for α_p.

VI. Fuzzy Association Rules Discovered in the Bank-Account Database

Instead of applying FARM II to the three original relations in the bank-account database, we performed data mining on the transformed relation (discussed in Section IV). In consultation with the banking officials, we defined appropriate linguistic terms for each attribute in the transformed relation. As an example, two linguistic terms *Small* and *Large* were defined for the attribute called *Loan Balance*. The definitions of these linguistic terms are given in Fig. 3.

As another illustration, let us consider the attribute called *Customer Age*. Four linguistic terms *Young*, *Youth*, *Middle Aged*, and *Elderly* were defined for *Customer Age* (see Fig. 4).

Using the linguistic terms that were defined by the domain expert, we applied FARM II to the transformed relation. From the discovered fuzzy association rules, we selected 200 rules randomly and presented them to the banking officials whom we consulted on the definition of the linguistic terms. The rules were evaluated according to how useful and how unexpected they were, as judged by the domain expert. The domain expert

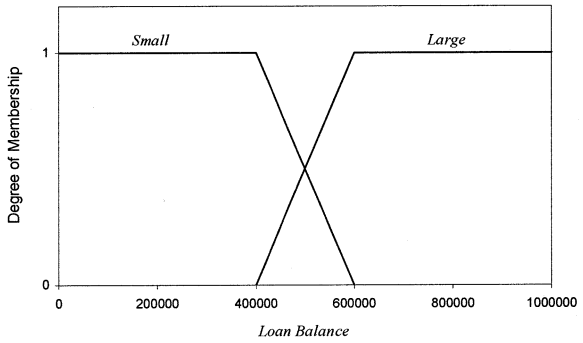

Fig. 3. Definitions of the linguistic terms for the attribute called *Loan Balance*.

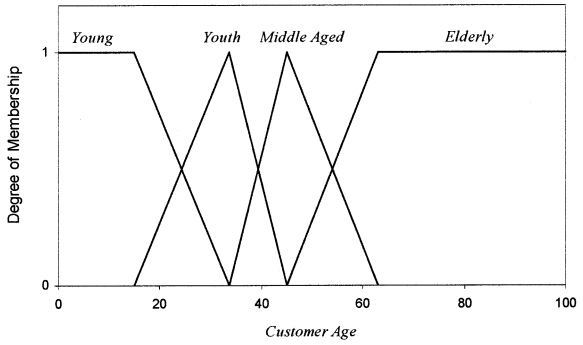

Fig. 4. Definitions of linguistic terms for the attribute called *Customer Age*.

TABLE II
Classification of the Fuzzy Association Rules Discovered in the Bank-Account Database

	No. of Rules	Percentages
Very useful	51	25.5%
Useful	132	66.0%
Less useful	17	8.5%

classified the rules into three categories: *very useful*, *useful*, and *less useful*. The result of the classification of these rules is summarized in Table II.

Among the 200 rules, the domain expert found 91.5% of them to be either useful or very useful. We expect that the evaluation of the remaining rules will follow a similar distribution because the 200 evaluated rules were selected randomly. This evaluation is quite high for an automated data-mining tool. The reasons for this are likely to be that our interestingness measure can effectively reveal the interesting associations that are hidden in the data and that the fuzzy association rules, which employ linguistic terms to represent the underlying relationships, are more natural for human users to understand.

In the rest of this section, we show some of the discovered fuzzy association rules, which have been identified as very useful by the domain expert. The following rule, regarding the affect that the annual income of a customer and the number of accounts that he/she holds has on the length of the customer relationship, was found to be very useful

$$\begin{aligned} \text{Annual Income} &= \text{Very Large} \wedge \text{No. of Accounts} \\ &= \text{Very Small} \Rightarrow \text{Relationship Length} \\ &= \text{Very Short } [0.71] \end{aligned}$$

where *Relationship Length* is produced by an arithmetic function $f_{\text{Relationship Length}}$ which is defined as follows:

$$f_{\text{Relationship Length}}(id) = \text{SYSDATE} - \pi_{\text{MEMBER\_SINCE}}\left(\sigma_{\text{CUST\_ID}=id}(\text{CUSTOMER})\right)$$

where π is the PROJECT operation in relational algebra and SYSDATE returns the current date in Oracle.

This rule states that a customer who has a very large annual income and who holds a very small number of accounts will have a very short relationship with the bank. The length of the relationship that the bank has with a customer is important because the bank has a greater opportunity to cross-sell its products and services to a customer if he/she stays with the bank for a longer time. The domain expert found this rule to be useful because it identifies the characteristics of customers who are more likely to have a short-tem relationship with the bank. By providing incentives to these customers, the bank can lengthen the relationships with them and increase its cross-selling opportunities (and hence we hope also improve its profitability). It is important to note that this rule only involves the attributes in the relational data.

The following fuzzy association rule, regarding the factors affecting the transaction costs, was also found to be very useful.

$$\begin{aligned}&\text{Sales Cost (Direct)} = \text{Large}\\ &\wedge \text{Sales Cost (Branch)} = \text{Very Large}\\ &\Rightarrow \text{ATM Transaction Cost} = \text{Very Large}\\ &\wedge \text{Branch Transaction Cost} = \text{Very Large}\ [5.38].\end{aligned}$$

This rule describes the costs of ATM transactions and branches as being very large if the cost of direct sales is large and the cost of branch sales is also very large. The rule identifies the factors that affect the costs of ATM transactions and branches. Based on this rule, the domain expert suggested that the bank could provide better control of the costs of direct and branch sales so that the costs of ATM transactions and branches could be reduced. It is also important to note that this rule only involves the attributes in the transactional data.

Let us consider the fuzzy association rules that involve attributes that are in both the relational and transactional data.

$$\begin{aligned}&\text{Customer Sex} = \text{Female} \Rightarrow \text{Loan Balance} = \text{Small}\ [1.23]\\ &\text{Customer Sex} = \text{Male} \Rightarrow \text{Loan Balance} = \text{Large}\ [0.67]\end{aligned}$$

where *Loan Balance* is produced by an arithmetic function, $f_{\text{Loan Balance}}$, which is defined as follows:

$$f_{\text{Loan Balance}}(id) = \frac{\sum_{t \in \sigma_{\text{CUST\_ID}=id}(\text{ACCOUNT})} t\,[\text{LOAN\_BALANCE}]}{\left|\sigma_{\text{CUST\_ID}=id}(\text{ACCOUNT})\right|}.$$

The former rule states that female customers are more likely to use small loans whereas the latter rule describes male customers as being more likely to use large loans. It is important to note that these rules are concerned with how the demographics of a customer affect his/her transactions. Specifically, they describe the associative relationships between a customer's gender, which is contained in the relational data and his/her total loan balances, which are contained in the transactional data. These rules cannot be discovered unless both relational and transactional data are considered together.

In addition to these rules, let us also consider the following fuzzy association rule:

$$\begin{aligned}&\text{Customer Sex} = \text{Female} \wedge \text{Marital Status}\\ &= \text{Widowed} \Rightarrow \text{Loan Balance} = \text{large}\ [3.62].\end{aligned}$$

This rule states that female customers who are widowed are more likely to use large loans. As discussed above, a female customer is expected to make use of only small loans. However, the fact that these women are widowed, means that they tend to use large loans. Similar to the rules discussed above, this rule associates the demographics (i.e., gender and marital status) of a customer with his/her transactions (i.e., loan balances). This rule can only be revealed if relational and transactional data are considered together.

A. Customer Retention

On the basis of the fuzzy association rules concerning the loan balance, the domain expert revealed that customers who use small loans could easily settle the loans as compared to those with larger loans. Because of this, customers who use small loans are more likely to stop using the loan services and cease to be a customer. Based on the rules concerning a small loan balance, the bank was able to identify the characteristics of customers that may cease being customers. The bank can retain more of its customers in the future by offering incentives to the customers that have the same characteristics. In this way, FARM II can be used for customer retention or to help reduce the customer attrition rate.

Let us consider the fuzzy association rules concerning the affect of the gender of a customer on his/her loan balance. Specifically, they state that female customers are more likely to use small loans whereas male customers tend to use large loans. Based on these rules, the domain expert also revealed that female customers usually have a significant amount of savings and it is probably because of this reason that they tend to use small loans. This characteristic means that female customers tend to find it easier to settle loans and hence they are more likely to cease using the loan services as compared to male customers. The attrition of customers is therefore related to gender. This finding was very useful to the domain expert because customers who are likely to cease using the loan services could be identified using these rules. To reduce the attrition rate, the domain expert suggested that incentives, such as lower interest rates, could be offered to female customers.

Let us also consider the fuzzy association rule that states that female customers who are widowed are more likely to use large loans. From other rules, we have revealed that female customers are more likely to cease using the loan services. However, the fact that these women are widowed, means that they tend to continue using the loan services. The domain expert found this rule especially useful because it identified a new niche market for promoting the bank's loan services.

VII. Conclusion

In this paper, we presented a novel algorithm, called FARM II, for mining fuzzy association rules. Unlike other data-mining

algorithms, which discover association rules based on support and confidence measures, FARM II employs an objective interestingness measure to identify interesting associations between linguistic terms without using any user-supplied threshold. Furthermore, FARM II uses a confidence measure to represent the uncertainty that is associated with fuzzy association rules. To handle both relational and transactional data in the bank-account database, we proposed the concept of using transformation functions and then introduced a formal approach for this problem. Depending on the type of attribute, we can apply different types of transformation functions to the attributes. The types of transformations include logical, arithmetic, substring and discretization functions. The use of transformation functions results in a transformed relation. Instead of performing data mining on the original data, we applied FARM II to the transformed data. Among the discovered fuzzy association rules, we selected 200 rules randomly and presented them to a domain expert from the bank. The domain expert confirmed that she could understand the fuzzy association rules without any difficulty although it was nontrivial for her to explain the basis for some of the rules. In particular, the domain expert found that 91.5% of these randomly selected rules were useful or very useful. The reasons for this are likely to be that our interestingness measure can effectively reveal the interesting associations that are hidden in the data and that the fuzzy association rules, which employ linguistic terms to represent the underlying relationships, are more natural for human users to understand.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACM SIGMOD Int. Conf. Management Data*, Washington, DC, 1993, pp. 207–216.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th Int. Conf. Very Large Data Bases*, Santiago, Chile, 1994, pp. 487–499.

[3] W.-H Au and K. C. C. Chan, "An effective algorithm for discovering fuzzy rules in relational databases," in *Proc. 7th IEEE Int. Conf. Fuzzy Systems*, Anchorage, AK, 1998, pp. 1314–1319.

[4] ——, "FARM: A data mining system for discovering fuzzy association rules," in *Proc. 8th IEEE Int. Conf. Fuzzy Systems*, Seoul, Korea, 1999, pp. 1217–1222.

[5] K. C. C. Chan and W.-H. Au, "Mining fuzzy association rules," in *Proc. 6th Int. Conf. Information Knowledge Management*, Las Vegas, NV, 1997, pp. 209–215.

[6] ——, "Mining fuzzy association rules in a database containing relational and transactional data," in *Data Mining and Computational Intelligence*, A. Kandel, M. Last, and H. Bunke, Eds. New York: Physica-Verlag, 2001, pp. 95–114.

[7] K. C. C. Chan and A. K. C. Wong, "APACS: A system for the automatic analysis and classification of conceptual patterns," *Comput. Intell.*, vol. 6, no. 3, pp. 119–131, 1990.

[8] J. Y. Ching, A. K. C. Wong, and K. C. C. Chan, "Class-dependent discretization for inductive learning from continuous and mixed-mode data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 641–651, July 1995.

[9] C. J. Date, *An Introduction to Database Systems*, 7th ed. Reading, MA: Addison-Wesley, 2000.

[10] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery: An overview," in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Cambridge, MA: MIT Press, 1996, pp. 1–34.

[11] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. San Francisco, CA: Morgan Kaufmann, 2000.

[12] J. L. Heskett, T. O. Jones, G. Loveman, and W. E. Sasser, "Putting the service profit chain to work," *Harvard Business Rev.*, pp. 164–174, Mar./Apr. 1994.

[13] K. Hirota and W. Pedrycz, "Fuzzy computing for data mining," *Proc. IEEE*, vol. 87, pp. 1575–1600, Sept. 1999.

[14] J. Kacprzyk and S. Zadrozny, "On linguistic approaches in flexible querying and mining of association rules," in *Flexible Query Answering Systems: Recent Advances*, H. L. Larsen, J. Kacprzyk, S. Zadrozny, T. Andreasen, and H. Christiansen, Eds. Heidelberg, Germany: Physica-Verlag, 2001, pp. 475–484.

[15] D. H. Lee and M. H. Kim, "Database summarization using fuzzy ISA hierarchies," *IEEE Trans. Systems, Man, Cybernetics B*, vol. 27, pp. 671–680, Apr. 1997.

[16] C. J. Matheus, P. K. Chan, and G. Piatetsky-Shapiro, "Systems for knowledge discovery in databases," *IEEE Trans. Knowledge Data Eng.*, vol. 5, pp. 903–913, Dec. 1993.

[17] M. Mendel, "Fuzzy logic systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, pp. 345–377, Mar. 1995.

[18] J. S. Park, M.-S. Chen, and P. S. Yu, "An effective hash-based algorithm for mining association rules," in *Proc. ACM SIGMOD Int. Conf. Management Data*, San Jose, CA, 1995, pp. 175–186.

[19] J. Peppard, "Customer relationship management (CRM) in financial services," *Eur. Manage. J.*, vol. 18, no. 3, pp. 312–327, 2000.

[20] F. Reichheld, *The Loyalty Effect: The Hidden Force Behind Growth, Profits and Lasting Value*. Boston, MA: Harvard Business School Press, 1996.

[21] A. Savasere, E. Omiecinski, and S. Navathe, "An efficient algorithm for mining association rules in large databases," in *Proc. 21st Int. Conf. Very Large Data Bases*, Zurich, Switzerland, 1995, pp. 432–444.

[22] M. Schwaiger and H. Locarek-Junge, "Realising customer retention potentials by electronic banking," *Electron. Markets*, vol. 8, no. 4, pp. 23–26, 1998.

[23] R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," in *Proc. ACM SIGMOD Int. Conf. Management Data*, Montreal, QC, Canada, June 1996, pp. 1–12.

[24] J. D. Ullman, *Principles of Database and Knowledge-Base Systems*. Rockville, MD: Computer Science, 1988, vol. 1.

[25] R. R. Yager, "On linguistic summaries of data," in *Knowledge Discovery in Databases*, G. Piatetsky-Shapiro and W. J. Frawley, Eds. Cambridge, MA: MIT Press, 1991, pp. 347–363.

[26] J. Yen, "Fuzzy logic—A modern perspective," *IEEE Trans. Knowledge Data Eng.*, vol. 11, pp. 159–165, Jan. 1999.

[27] L. Zadeh, "Fuzzy sets," *Inform. Control*, vol. 8, pp. 338–353, 1965.

Wai-Ho Au received the B.A. degree (with First Class Honors) in computing studies and the M.Phil. degree in computing from The Hong Kong Polytechnic University, Hong Kong, in 1995 and 1998, respectively. He is currently working towards the Ph.D. degree in the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

He has been in charge of several large-scale software development projects, including a system integration project for an international airport, a data warehouse project for a utility company, and an intelligent home system for a high-tech startup. He is now a Manager of Software Development in the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests include data mining, data warehousing, fuzzy computing, and evolutionary algorithms.

Keith C. C. Chan received the B.Math. (Hons.) degree in computer science and statistics and the M.A.Sc. and Ph.D. degrees in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 1983, 1985, and 1989, respectively.

He has a number of years of academic and industrial experience in software development and management. In 1989, he joined the IBM Canada Laboratory, Toronto, ON, Canada, where he was involved in the development of image and multimedia software as well as software development tools. In 1993, he joined the Department of Electrical and Computer Engineering, Ryerson Polytechnic University, Toronto, ON, Canada as an Associate Professor. He joined the Hong Kong Polytechnic University, Hong Kong, in 1994, and is currently the Head of the Department of Computing. He is an Adjunct Professor of the Institute of Software, The Chinese Academy of Sciences, Beijing, China. He is active in consulting, and has served as Consultant to government agencies as well as large and small-to-medium sized enterprises in Hong Kong, China, Singapore, Malaysia, Italy, and Canada. His research interests are in data mining and machine learning, computational intelligence, and software engineering.