# Mining Fuzzy Sequential Patterns from Quantitative Data

**Tzung-Pei Hong[†], Chan-Sheng Kuo[‡], and Sheng-Chai Chi[‡]**
**[†]Department of Information Management**
**[‡]Graduate School of Management Science**
**I-Shou University**
**Kaohsiung, 84008, Taiwan, R.O.C.**
**e-mail: tphong@csa500.isu.edu.tw**

## ABSTRACT

Data mining is the process of extracting desirable knowledge or interesting patterns from existing databases for specific purposes. Most of the conventional data mining algorithms can identify the relation among transactions with binary values. Temporal transactions with quantitative values are however commonly seen in real world applications. This paper thus attempts to propose a new data-mining algorithm, which takes the advantages of fuzzy sets theory, to enhance the capability of exploring interesting sequential patterns from the databases with quantitative values. The proposed algorithm integrates the concepts of fuzzy sets and the AprioriAll algorithm to find interesting sequential patterns and fuzzy association rules from transaction data.

## 1. INTRODUCTION

As useful databases have become public and pervasive, the technology of data mining is urgently requested and developed in the recent years. Data mining is the process to extract desirable knowledge with interesting patterns for a certain purpose from the existing databases. Due to the importance of data mining, many researchers in database and machine learning fields are primarily interested in this new research topic because it offers opportunities to discover useful information and important relevant patterns in large databases, thus helping decision-makers easily analyze the data and make good decisions regarding the domains concerned.

Data-mining is most commonly used in attempts to induce association rules from transaction data. Most previous studies have only shown, however, how binary valued transaction data may be handled. Transaction data in real-world applications usually consist of quantitative values, so designing a sophisticated data-mining algorithm able to deal with various types of data presents a challenge to workers in this research field. Among the topics of data mining, finding useful sequential patterns is very interesting. It is concerned with inter-transaction patterns, which are ordered lists of items, instead of unordered sets of items.

Fuzzy set theory is being used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning [9]. The theory has been applied in fields such as manufacturing, engineering, diagnosis, economics, among others [6, 9, 10, 12].

The proposed algorithm in this paper integrates the concepts of fuzzy sets and the AprioriAll algorithm [5] to find interesting sequential patterns and fuzzy association rules from transaction data. It transforms quantitative values in transactions into linguistic terms, then filters them to find association rules by modifying the AprioriAll mining algorithm [5]. The proposed mining algorithm can predict the next plausible consequence underlying the generation of a given sequence of items. For example, the mined fuzzy rules can predict what products and quantities will be bought next for a customer, if a sequence of products and quantities have been bought by him. The rules mined out thus exhibit the sequential quantitative regularity in large databases and can be used to provide some suggestions to appropriate supervisors.

## 2. REVIEW OF AGRAWAL ET AL.'S DATA-MINING ALGORITHMS

The goal of data mining is to discover important associations among items such that the presence of some items in a transaction will imply the presence of some other items. To achieve this purpose, Agrawal and his co-workers proposed several mining algorithms based on the concept of large itemsets to find association rules in transaction data [1, 2, 3, 4]. They divided the mining process into two phases. In the first phase, candidate itemsets were generated and counted by scanning the transaction data. If the number of an itemset appearing in the transactions was larger than a pre-defined threshold value (called minimum support), the itemset was considered a large itemset. Itemsets containing only one item were processed first. Large itemsets containing only single items were then combined to form candidate itemsets containing two items. This process was repeated until all large itemsets had been found. In the second phase, association rules were induced from the large itemsets found in the first phase. All possible association combinations for each large itemset were formed, and those with calculated confidence values larger than a predefined threshold (called minimum confidence) were output as association rules.

In addition to proposing methods for mining association rules from transactions of binary values, Agrawal et al. also proposed a method [11] for mining association rules from those with quantitative and categorical attributes. Their proposed method first determines the number of partitions for each quantitative attribute, and then maps all possible values of each attribute into a set of consecutive integers. It then finds large itemsets whose support values are greater than the user-specified minimum-support levels. These large itemsets are then processed to generate association rules, and rules of interest to users are output.

Agrawal and Srikant also proposed several mining algorithms

0-7803-5731-0/99/$10.00 ©1999 IEEE

for finding sequential patterns from transaction data. Five phases, including sort, litemset, transformation, sequence, maximal, are included in their algorithms.

In this paper, we use fuzzy set concepts to mine interesting sequential patterns from transactions with quantitative attributes. The mined rules are expressed in linguistic terms, which are more natural and understandable for human beings.

## 3. THE FUZZY DATA-MINING ALGORITHM FOR SEQUENTIAL PATTERNS

In this paper, the fuzzy concepts are used in the AprioriAll data-mining algorithm [5] to discover interesting sequential patterns and fuzzy association rules from quantitative values. Notation used in this paper is first stated as follows.

*n: the total number of transaction data;*
*m: the total number of attributes;*
$D^{(i)}$ *: the i-th transaction datum,* $1 \le i \le n$*;*
$A_j$ *: the j-th attribute,* $1 \le j \le m$*;*
$|A_j|$ *: the number of fuzzy regions for* $A_j$*;*
$R_{jk}$ *: the k-th fuzzy region of* $A_j$*,* $1 \le k \le |A_j|$*;*
$v_j^{(i)}$ *: the quantitative value of* $A_j$ *for* $D^{(i)}$*;*
$f_j^{(i)}$ *: the fuzzy set converted from* $v_j^{(i)}$*;*
$f_{jk}^{(i)}$ *: the membership value of* $v_j^{(i)}$ *in Region* $R_{jk}$*;*
$count_{jk}$ *: the summation of* $f_{jk}^{(i)}$ *for i=1 to n;*
$\alpha$ *: the predefined minimum support level;*
$\lambda$ *: the predefined minimum confidence value;*
$C_r$*: the set of candidate itemsets with r attributes (items);*
$L_r$ *: the set of large itemsets with r attributes (items).*

The proposed fuzzy mining algorithm first transforms each quantitative value into a fuzzy set with linguistic terms using membership functions. It then calculates the scalar cardinality of each linguistic term on all the transaction data. The linguistic terms with their scalar cardinalities larger than the minimum support value are kept as the large one-item set. Different permutations of patterns with two items are then formed from the large one-item set. Fuzzy operations are used to calculate the scalar cardinalities of these patterns. The patterns with their scalar cardinalities larger than the minimum support value are thus kept as the large two-item set. The same procedure is repeated to find large sets of other numbers of items. The mining process based on fuzzy counts is then performed to find fuzzy association rules from these sequential patterns. The detail of the proposed mining algorithm is described as follows.

***The Fuzzy Data Mining Algorithm for sequential patterns:***

INPUT: A body of $n$ transaction data, each with $m$ attribute values, a set of membership functions, a predefine minimum support value $\alpha$, and a predefined confidence value $\lambda$.

OUTPUT: A set of fuzzy association rules.

STEP 1: Transform the quantitative value $v_j^{(i)}$ of each transaction datum $D^{(i)}$, $i$=1 to $n$, for each attribute $A_j$, $j$=1 to $m$, into a fuzzy set $f_j^{(i)}$ represented as $\left( \frac{f_{j_1}^{(i)}}{R_{j_1}} + \frac{f_{j_2}^{(i)}}{R_{j_2}} + .... + \frac{f_{j_l}^{(i)}}{R_{j_l}} \right)$ using the given membership functions, where $R_{jk}$ is the $k$-th fuzzy region of attribute $A_j$, $f_{jk}^{(i)}$ is $v_j^{(i)}$'s fuzzy membership value in region $R_{jk}$, and $l$ $(=|A_j|)$ is the number of fuzzy regions for $A_j$.

STEP 2: Calculate the scalar cardinality of each attribute region $R_{jk}$ in the transaction data:

$$count_{jk} = \sum_{i=1}^{n} f_{jk}^{(i)} .$$

STEP 3: For each $R_{jk}$, $1 \le j \le m$ and $1 \le k \le |A_j|$, check whether its $count_{jk}$ is larger than or equal to the predefined minimum support value $\alpha$. If $R_{jk}$ satisfies the above condition, put it in the set of large 1-itemsets ($L_1$). That is:

$$L_1 = \{ R_{jk} \mid count_{jk} \ge \alpha,\ 1 \le j \le m \text{ and } 1 \le k \le |A_j| \}.$$

STEP 4: Set $r$=1, where $r$ is used to represent the number of items kept in the current large itemsets.

STEP 5: Generate the candidate set $C_{r+1}$ from $L_r$ except that two regions belonging to the same attribute cannot simultaneously exist in an itemset in $C_{r+1}$. Restated, the algorithm first joins $L_r$ and $L_r$ under the condition that $r$-1 items in the two itemsets are the same and with the same order of sequences, and the other one is different. Different permutations represent different conditions. The algorithm then keeps in $C_{r+1}$ the itemsets which have all their sub-itemsets of $r$ items existing in $L_r$ and do not have two items $R_{jp}$ and $R_{jq}$ $(p \ne q)$.

STEP 6: Do the following substeps for each newly formed ($r$+1)-itemset $s$ with items $(s_1, s_2, ..., s_{r+1})$ in $C_{r+1}$:

(a) Calculate the fuzzy value of each transaction data $D^{(i)}$ in $s$ as $f_s^{(i)} = f_{s_1}^{(i)} \Lambda f_{s_2}^{(i)} \Lambda ... \Lambda f_{s_{r+1}}^{(i)}$, where $f_{s_j}^{(i)}$ is the membership value of $D^{(i)}$ in region $s_j$ and $(s_1, s_2, ..., s_{r+1})$ is a subsequence of $D^{(i)}$. If the minimum operator is used for the intersection, then:

$$f_s^{(i)} = \underset{j=1}{\overset{r+1}{Min}} f_{s_j}^{(i)} .$$

If more than one sequence exists, find their maximum fuzzy value as the final result.

(b) Calculate the scalar cardinality of $s$ on the transactions as:

$$count_s = \sum_{i=1}^{n} f_s^{(i)} .$$

(c) If $count_s$ is larger than or equal to the predefined minimum support value $\alpha$, put $s$ in $L_{r+1}$.

STEP 7: IF $L_{r+1}$ is null, then do the next step; otherwise, set $r$=$r$+1 and repeat STEPs 5 to 7.

STEP 8: Construct the association rules for all large $q$-itemset $s$ with items $(s_1, s_2, ..., s_q)$, $q \ge 2$, using the following substeps:

(a) Form the possible association rule as follows:

$$s_1 \Lambda s_2 \Lambda ... \Lambda s_{q-1} \rightarrow s_q .$$

(b) Calculate the confidence values of all association rules using:

$$\frac{\sum_{i=1}^{n} f_s^{(i)}}{\sum_{i=1}^{n} ( f_{s_1}^{(i)} \wedge f_{s_2}^{(i)} \wedge \ \ldots \ \wedge \ f_{s_{q-1}}^{(i)} )} .$$

STEP 9: Output the rules with confidence values larger than or equal to the predefined confidence threshold $\lambda$.

After STEP 9, the rules output can serve as meta-knowledge concerning the given transactions.

## 4. AN EXAMPLE

In this section, an example is given to illustrate the proposed fuzzy data-mining algorithm. This is a simple example to show how the proposed algorithm can be used to generate interesting sequential patterns for customers' purchase behavior according to historical data with customers' purchase quantities. The data set, including 10 transactions, is shown in Table 1.

*Table 1: The data set used in the example*

| Customer ID | Transaction Time | (Product, Quantity) |
|---|---|---|
| 1 | May 5 '99 | (B,2) |
| 1 | May 17 '99 | (E,3) |
| 1 | May 28 '99 | (B,4) |
| 2 | May 13 '99 | (A,4) |
| 2 | May 19 '99 | (B,2) |
| 2 | May 21 '99 | (D,4) |
| 3 | May 7 '99 | (A,3) |
| 3 | May 15 '99 | (D,7) |
| 3 | May 18 '99 | (A,7) |
| 4 | May 11 '99 | (B,3) |
| 4 | May 16 '99 | (C,8) |
| 4 | May 23 '99 | (E,1) |
| 5 | May 3 '99 | (A,5) |
| 5 | May 13 '99 | (D,9) |
| 5 | May 17 '99 | (B,2) |
| 5 | May 29 '99 | (E,9) |
| 6 | May 19 '99 | (D,6) |
| 6 | May 25 '99 | (D,2) |
| 7 | May 4 '99 | (D,9) |
| 7 | May 12 '99 | (B,1) |
| 7 | May 16 '99 | (C,6) |
| 7 | May 31 '99 | (E,10) |
| 8 | May 1 '99 | (A,6) |
| 8 | May 20 '99 | (C,1) |
| 8 | May 27 '99 | (E,7) |
| 9 | May 13 '99 | (D,8) |
| 9 | May 17 '99 | (B,1) |
| 9 | May 28 '99 | (C,7) |
| 10 | May 7 '99 | (C,9) |
| 10 | May 8 '99 | (D,9) |
| 10 | May 19 '99 | (E,5) |
| 10 | May 23 '99 | (E,9) |

Here we assume that each customer buys only one product each time. The proposed method can be easily extended for a customer to buy several products each time. There are five kinds of products in this example. The data set in Table 1 is then represented for each customer according to the occuring time. Results are shown in Table 2.

Assume the fuzzy membership functions for the product quantities are shown in Figure 1.

*Table 2: The data set represented according to occuring time*

| Transaction No. | Items bought and Quantities |
|---|---|
| 1 | (B,2),(E,3),(B,4) |
| 2 | (A,4),(B,2),(D,4) |
| 3 | (A,3),(D,7),(A,7) |
| 4 | (B,3),(C,8),(E,1) |
| 5 | (A,5),(D,9),(B,2),(E,9) |
| 6 | (D,6)(D,2) |
| 7 | (D,9),(B,1),(C,6),(E,10) |
| 8 | (A,6),(C,1),(E,7) |
| 9 | (D,8),(B,1),(C,7) |
| 10 | (C,9),(D,9),(E,5),(E,9) |

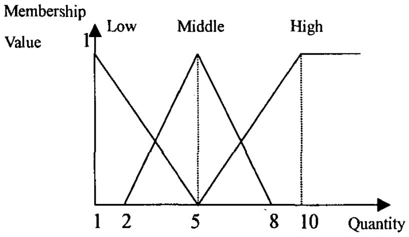

Figure 1: The membership functions used in this example

In this example, the quantities purchased are divided into three fuzzy regions: *Low*, *Middle* and *High*. Thus, three fuzzy membership values are produced for each transaction according to the predefined membership functions. For the transaction data in Table 1, the proposed mining algorithm proceeds as follows.

STEP 1: Transform the quantitative values of each transaction datum into fuzzy sets. Take the *B* product bought by customer 1 as an example. The quantity "2" is converted into a fuzzy set $(\frac{0.8}{Low} + \frac{0.0}{Middle} + \frac{0.0}{High})$ using the given membership functions. This step is repeated for the other products and customers, and the results are shown in Table 3.

*Table 3: The fuzzy sets transformed from the data in Table 2*

| Customer | Fuzzy quantities of products bought |
|---|---|
| 1 | (0.8/B.Low),(0.5/E.Low+0.3/E.Middle),<br>(0.3/B.Low+0.7/B.Middle) |
| 2 | (0.3/A.Low+0.7/A.Middle),(0.8/B.Low),<br>(0.3/D.Low+0.7/D.Middle) |
| 3 | (0.5/A.Low+0.3/A.Middle),(0.3/D.Middle+0.4/D.High),<br>(0.3/A.Middle +0.4/A.High) |
| 4 | (0.5/B.Low+0.3/B.Middle),(0.6/C.High),(1.0/E.Low) |
| 5 | (1.0/A.Middle),(0.8/D.High),(0.8/B.Low),(0.8/E.High) |
| 6 | (0.7/D.Middle+0.2/D.High),(0.8/D.Low) |
| 7 | (0.8/D.High),(1.0/B.Low),(0.7/C.Middle+0.2/C.High),<br>(1.0/E.High) |
| 8 | (0.7/A.Middle+0.2/A.High),(1.0/C.Low),(0.3/E.Middle<br>+0.4/E.High) |
| 9 | (0.6/D.High),( 1.0/B.Low),( 0.3/C.Middle +0.4/C.High) |
| 10 | (0.8/C.High),(0.8/D.High),(1.0/E.Middle),(0.8/E.High) |

Each linguistic term such as *B.Low* is then thought of as an item in the mining process.

STEP 2: Calculate the scalar cardinality of each item in the transactions as the *count* value. Take the region *A.Low* as an example. Its scalar cardinality = (0.3 + 0.5) = 0.8. This step is repeated for the other regions, and the results are shown in Table

4.

*Table 4: The set of candidate 1-itemsets $C_1$ for this example*

| Itemset | Support |
|---|---|
| A.Low | 0.8 |
| A.Middle | 2.7 |
| A.High | 0.6 |
| B.Low | 4.9 |
| B.Middle | 1.0 |
| ... | ... |
| E.High | 3.0 |

STEP 3: For each item, check whether its count is larger than or equal to the predefined minimum support value $\alpha$ . Assume $\alpha$ is set at 2 in this example. Since the count values of A.Middle, B.Low, C.High, D.High and E.High are larger than 2, these items are put in $L_1$ (Table 5).

*Table 5: The set of large 1-itemsets $L_1$ for this example*

| Itemset | Support |
|---|---|
| A.Middle | 2.7 |
| B.Low | 4.9 |
| C.High | 2.0 |
| D.High | 3.6 |
| E.High | 3.0 |

STEP 4: Set $r$=1.

STEP 5: Generate the candidate set $C_{r+1}$ from $L_r$. $C_2$ is first generated from $L_1$ as follows: (A.Middle, B.Low), (B.Low, A.Middle), (A.Middle, C.High), (C.High, A.Middle), ..., (E.High, D.High). Note that the same itemsets with different sequences are thought of as different.

STEP 6: Do the following substeps for each newly formed candidate itemset.

(a) Calculate the fuzzy membership value of the candidate itemset in each transaction datum. Here, the minimum operator is used for the intersection. Take (B.Low, D.High) as an examples. Its membership value for Customer 5 is calculated as: *min*(0.8, 0.0)=0.0 since no items of D.High appearing after B.Low. Note that the membership value for (D.High, B.Low) is 0.8. The results for the other customers are shown in Table 6 and Table 7.

*Table 6: The membership values for (B.Low, D.High)*

| Customer | B.Low | D.High | (B.Low, D.High) |
|---|---|---|---|
| 1 | 0.8 | 0.0 | 0.0 |
| 2 | 0.8 | 0.0 | 0.0 |
| 3 | 0.0 | 0.4 | 0.0 |
| 4 | 0.5 | 0.0 | 0.0 |
| 5 | 0.8 | 0.0 | 0.0 |
| 6 | 0.0 | 0.2 | 0.0 |
| 7 | 1.0 | 0.0 | 0.0 |
| 8 | 0.0 | 0.0 | 0.0 |
| 9 | 1.0 | 0.0 | 0.0 |
| 10 | 0.0 | 0.8 | 0.0 |

The results for the other 2-itemsets can be derived in similar fashion.

*Table 7: The membership values for (D.High, B.Low)*

| Customer | D.High | B.Low | (D.High, B.Low) |
|---|---|---|---|
| 1 | 0.0 | 0.8 | 0.0 |
| 2 | 0.0 | 0.8 | 0.0 |
| 3 | 0.4 | 0.0 | 0.0 |
| 4 | 0.0 | 0.5 | 0.0 |
| 5 | 0.8 | 0.8 | 0.8 |
| 6 | 0.2 | 0.0 | 0.0 |
| 7 | 0.8 | 1.0 | 0.8 |
| 8 | 0.0 | 0.0 | 0.0 |
| 9 | 0.6 | 1.0 | 0.6 |
| 10 | 0.8 | 0.0 | 0.0 |

(b) Calculate the scalar cardinality (count) of each candidate 2-itemset in the transaction data. Results for this example are shown in Table 8.

Table 8: The fuzzy counts of the itemsets in $C_2$

| Itemset | Support |
|---|---|
| (A.Middle, B.Low) | 1.5 |
| (B.Low, A.Middle) | 0.0 |
| (A.Middle, C.High) | 0.0 |
| (C.High, A.Middle) | 0.0 |
| (A.Middle, D.High) | 1.1 |
| (D.High, A.Middle) | 0.3 |
| ... | ... |
| (E.High, D.High) | 0.0 |

(c) Check whether these counts are larger than or equal to the predefined minimum support value 2. Two itemsets, including (D.High, B.Low), (D.High, E.High) , are thus kept in $L_2$ (Table 9).

*Table 9: The itemsets and their fuzzy counts in $L_2$*

| Itemset | Support |
|---|---|
| D.High B.Low | 2.2 |
| D.High E.High | 2.4 |

STEP 7: IF $L_{r+1}$ is null, then do the next step; otherwise, set $r$=$r$+1 and repeat STEPs 5 to 7. Since $L_2$ is not null in the example above, $r$=$r$+1=2. STEPs 5 to 7 are then repeated to find $L_3$. $C_3$ is first generated from $L_2$, and two itemsets (D.High, B.Low, E.High) and (D.High, E. High, B.Low) are formed. Since all their counts are smaller than 2, they are not put in $L_3$. $L_3$ is thus an empty set. STEP 8 then begins.

STEP 8: Construct the association rules for each large itemset using the following substeps.

(a) Form all possible association rules. The following two association rules are possible:

If D.High then B.Low, and
If D.High then E.High.

(b) Calculate the confidence factors for the above association rules. Assume the given confidence threshold $\lambda$ is 0.6. Take the second association rule as an example. The fuzzy count of (D.High, E.High) is calculated as shown in Table 10.

Table 10: The fuzzy count of (D.High, E.High)

| Customer | D.High | E.High | (D.High, E.High) |
|---|---|---|---|
| 1 | 0.0 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 | 0.0 |
| 3 | 0.4 | 0.0 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 |
| 5 | 0.8 | 0.8 | 0.8 |
| 6 | 0.2 | 0.0 | 0.0 |
| 7 | 0.8 | 1.0 | 0.8 |
| 8 | 0.0 | 0.4 | 0.0 |
| 9 | 0.6 | 0.0 | 0.0 |
| 10 | 0.8 | 0.8 | 0.8 |
| count | 3.6 | 3.0 | 2.4 |

The confidence factor for the association rule "*If D = High, then E = High* " is then:

$$\frac{\sum_{i=1}^{10} (D.High \cap E.High)}{\sum_{i=1}^{10} (D.High)} = \frac{2.4}{3.6} = 0.67$$

In the same way, the confidence factor for the association rule "If D=High, then B.Low" is 0.61.

STEP 9: Check whether the confidence factors of the above association rules are larger than or equal to the predefined confidence threshold $\lambda$. Since the confidence $\lambda$ was set at 0.6 in this example, the following one rule is thus output to users:

If a customer purchases high quantity of product D, then he will next purchases high quantity of product E, with a confidence factor of 0.67.

If a customer purchases high quantity of product D, then he will next purchases low quantity of product B, with a confidence factor of 0.61.

The two rules above are thus output as meta-knowledge concerning the given transactions.

## 5. CONCLUSION

In this paper, we have proposed a generalized data-mining algorithm, which can process transaction data with quantitative values and discover interesting sequential patterns among them. The rules can thus predict what products and quantities will be bought next for a customer and can be used to provide some suggestions to appropriate supervisors.

Although the proposed method works well in data mining for quantitative values, it is just a beginning. There is still much work to be done in this field. Our method assumes that the membership functions are known in advance. In [7, 8], we also proposed some fuzzy learning methods to automatically derive the membership functions. In the future, we will attempt to dynamically adjust the membership functions in the proposed mining algorithm to avoid the bottleneck of membership function acquisition. We will also attempt to design specific data-mining models for various problem domains.

## REFERENCES

[1] R. Agrawal, T. Imielinksi and A. Swami, "Mining Association Rules Between Sets of Items in Large Database," *The 1993 ACM SIGMOD Conference,* Washington DC, USA, 1993.

[2] R. Agrawal, T. Imielinksi and A. Swami, "Database Mining: A Performance Perspective," *IEEE Transactions on Knowledge and Data Engineering,* Vol. 5, No. 6, 1993, pp. 914-925.

[3] R. Agrawal, R. Srikant and Q. Vu, "Mining Association Rules with Item Constraints," *The Third International Conference on Knowledge Discovery in Databases and Data Mining,* Newport Beach, California, August 1997.

[4] R. Agrawal and R. Srikant, "Fast Algorithm for Mining Association Rules," *The International Conference on Very Large Data Bases*, 1994, pp. 487-499.

[5] R. Agrawal and R. Srikant, *Mining Sequential Patterns*, Research Report RJ 9910, IBM Almaden Research Center, San Jose, California, 1994.

[6] I. Graham and P. L. Jones, *Expert Systems – Knowledge, Uncertainty and Decision,* Chapman and Computing, Boston, 1988, pp.117-158.

[7] T. P. Hong and J. B. Chen, "Finding Relevant Attributes and Membership Functions," *Fuzzy Sets and Systems,* Vol.103, No. 3, 1999, pp. 389-404.

[8] T. P. Hong and C. Y. Lee, "Induction of Fuzzy Rules and Membership Functions from Training Examples," *Fuzzy Sets and Systems,* Vol. 84, 1996, pp. 33-47.

[9] A. Kandel, *Fuzzy Expert Systems*, CRC Press, Boca Raton, 1992, pp. 8-19.

[10] E. H. Mamdani, "Applications of Fuzzy Algorithms for Control of Simple Dynamic Plants, " *IEEE Proceedings*, 1974, pp. 1585-1588.

[11] R. Srikant and R. Agrawal, "Mining Quantitative Association Rules in Large Relational Tables," *The 1996 ACM SIGMOD International Conference on Management of Data*, Monreal, Canada, June 1996, pp. 1-12.

[12] L. A. Zadeh, "Fuzzy Logic," *IEEE Computer*, 1988, pp.83-93.