Journal of Network Intelligence ©2018 ISSN 2414-8105(Online)
Taiwan Ubiquitous Information Volume **3**, Number **2**, May **2018**

Generalizing Concept-Drift Patterns for Fuzzy Association Rules

Tzung-Pei Hong

Department of Computer Science and Information Engineering
National University of Kaohsiung
700 University Rd., Kaohsiung City, 811, Taiwan
Department of Computer Science and Engineering
National Sun Yat-sen University
70 Lienhai Rd., Kaohsiung City, 804, Taiwan
tphong@nuk.edu.tw

Jimmy Ming-Tai Wu*

College of Computer Science and Engineering
Shandong University of Science and Technology
579 Qianwangang Rd., Huangdao, Qingdao, Shandong, 266590, China
School of Humanities and Social Sciences
Harbin Institute of Technology-Shenzhen
HIT Campus, The University Town of Shenzhen, Xili, Shenzhen, Guangdong, 518055, China
*Corresponding Author: wmt@wmt35.idv.tw

Yan-Kang Li

Department of Computer Science and Information Engineering
National University of Kaohsiung
700 University Rd., Kaohsiung City, 811, Taiwan
m1025506@mail.nuk.edu.tw

Chun-Hao Chen

Department of Computer Science and Information Engineering
Tamkang University
No.151, Yingzhuan Rd., New Taipei City, 251, Taiwan
chchen@mail.tku.edu.tw

Received Feb. 2018, Revised April 2018

Abstract. *In recent years, concept drift has become a popular research topic in data mining because of its wide range of practical applications. Concept drift discusses the significant change of concepts along with time or location movement. It can thus be used to derive the purchasing behavior of customers from a database at different times or locations. In the past, most of the research focuses on concept drift from traditional transaction databases. However, quantities of purchased items usually exist in databases and can provide more information in concept drift. In this paper, we integrate fuzzy data mining with concept drift and generalize the similarity measures for achieving the purpose. These measures are used for defining fuzzy concept-drift patterns derived from quantitative transaction databases at different time. The proposed approach can thus fully utilize the results from fuzzy mining and provide effective information about customers behavior change to managers for tuning the business strategy.*
Keywords: Concept drift, Data mining, Fuzzy set, Fuzzy association rules, Membership functions.

1. **Introduction.** Data storage and processing is now more convenient than before because of the booming development of information technologies. It has a significant impact on daily life as well as on business. For example, if policy-makers can obtain information and knowledge from databases effectively and quickly, they are likely to make good decisions. The sizes of databases, however, get bigger and bigger in these years. Therefore, getting useful and valuable information from large databases for decision-making becomes difficult. Research dealing with information storage and knowledge mining is thus an important and challenging task.

Data mining techniques have been applied to various practical applications for finding useful rules or patterns, a survey paper [1] about sequential pattern mining was proposed by Philippe et al. to introduce some previous researches. When quantitative databases are processed, it is natural and informative to use fuzzy sets to represent quantities into linguistic terms. Thus, Kuok et al. applied the fuzzy set theory to traditional data mining [2]. The main reason is that the fuzzy set theory can be easily used in various applications due to its simplicity and similarity to human reasoning. According to the key steps of the approach, the quantitative values in transactions were first converted into linguistic terms through membership functions, and then the count of a fuzzy itemset in a transaction could be calculated by the product of the fuzzy values of the fuzzy terms in the itemset of that transaction. Finally, the fuzzy association rules, which satisfied the user-specified minimum fuzzy confidence threshold, could be derived from a set of fuzzy frequent itemsets with high fuzzy frequency. Different from the calculation function in the above study, Hong et al. proposed a fuzzy mining algorithm to find fuzzy association rules by transforming quantitative data into fuzzy values and using the standard fuzzy operators to find the fuzzy value of a fuzzy itemset [3]. Because of the success of fuzzy mining, many extended approaches were proposed as well [4][5][6].

Along with the strict competition of business in these days, understanding and adapting to the evolvement of customer behavior turns out to be an important aspect of enterprise survival in the continuously changing environment. Good companies have to know what is changing and how it has been changed in order to provide right products and services to satisfy the varying market needs. Due to the above reason, concept drift has thus become a popular research topic in data mining [7][8][9]. Concept drift discusses the significant change of concepts along with the time process or location movement. It can be used to derive the purchasing behavior of customers from a database at different times or locations.

In the past, most of the research focuses on concept drift from traditional transaction databases. However, quantities of purchased items usually exist in databases and can provide more information in concept drift. Therefore, in this paper, we adopt fuzzy sets to handle concept drift with quantitative databases. We propose an algorithm to mine fuzzy concept-drift patterns from two quantitative databases at different times or locations. Fuzzy association rules with fuzzy values are first mined by a fuzzy mining algorithm [3][6]. A concept-drift judging algorithm then compares the fuzzy rules to find several kinds of fuzzy concept-drift patterns.

The rest of this paper is organized as follows. The review of some related researches is given in Section 2. Generating fuzzy association rules and defining their similarity are stated in Section 3. Generalizing concept-drift patterns for fuzzy association rules is explained in Section 4. The proposed algorithm for finding fuzzy concept-drift patterns is stated in Section 5. The conclusion and future work are given in Section 6.

2. **Review of Related Works.** In this section, we review some related researches about this paper. They are concept drift and fuzzy data mining.

2.1. **Concept Drift.** In recent years, the field of concept drift is very popular. Tsymbal defined the concept drift as finding patterns which changed over time in unexpected ways [10]. For example, assume at time t there is an association rule "if buying milk, then buying bread", and at time $t+k$, there is another rule "if buying milk, then buying apple". The latter rule differs from the former one in the consequent part along the time. This change is a type of concept-drift patterns.

Based on the concept-drift patterns, the traditional method on data mining has been used in various research areas. Au et al. proposed an online algorithm for changing detection in frequent pattern mining [11]. Hora et al. then proposed an extracting system for specific rules by mining systematic changes over source code history [12]. Cheng et al. utilized the group information at different time points for consensus sequence mining [13]. Their aim was to find the change in consensus sequence at different times for understanding the changes in group preference moving closer to the authentic idea of that group.

As to concept drift in the association rule mining, Song et al. defined three types of concept-drift patterns [14]. They were emerging patterns, unexpected change, and added/perished patterns. The different types of concept-drift patterns indicated the different meaning of the concept drift of association rules. An evaluative function was also designed to calculate the degrees of the concept drift. If the degree of the concept drift between two rules was bigger than a predefined threshold, concept-drift patterns were generated.

Assume there are two rules: $r_i^t : A \to B$ with support $= a$ and $r_i^{t+k} : C \to D$ with support $= b$, where r_i^t is the i-th rule of the rule set RS^t at time t, r_i^{t+k} is the j-th rule of the rule set RS^{t+k} and A, B, C, D are itemsets. The definitions of the three patterns are given below [14].

Definition 2.1. ***(Emerging Patterns)*** *If a rule r^k is an emerging pattern, then the following two conditions must be satisfied: (1) Both the conditional and the consequent parts of the two rules r_i^t and r_i^{t+k} are the same. That is, $A = C$ and $B = D$; (2) The supports of rules r_i^t and $r_i^{t_k}$ are different. That is $sup\,(A \to B) \neq sup\,(C \to D)$.*

For example, assume there are the following two rules: r_i^t : Bread $=$ *high* $\to$ Milk $=$ *Large* (support $= 0.2$), and r_i^{t+k} : Bread $=$ *high* $\to$ Milk $=$ *Large* (support $= 0.5$). In this case, the two rules have the same rule contents but different support values. The difference of the support values for the two rules is 0.3. If we set the minimum threshold at 0.2, then r_i^{t+k} is the emerging pattern to r_i^t.

Definition 2.2. ***(Unexpected Change)*** *If a rule r_k is an unexpected change, then the following two conditionals must be satisfied: (1) The conditional parts of the two rules r_i^t and r_i^{t+k} are the same. That is $A = C$; (2) The consequent parts of the two rules r_i^t and r_i^{t+k} are different. That is $B \neq D$.*

For example, assume there are the following two rules: r_i^t : Bread $=$ *high* $\to$ Milk $=$ *Large*, and r_i^{t+k} : Bread $=$ *high* $\to$ Milk $=$ *Low*. In this case, rule r_i^{t+k} is an unexpected change with respect to r_i^t since the conditional parts of r_i^t and r_i^{t+k} are the same, but the consequent parts of the two rules are different.

Definition 2.3. ***(Added/Perished Rules)*** *If a rule r_i^{t+k} is an added rule, then the conditional part C and the consequent part D of r_i^{t+k} are different from those of any r_i^t in RS^t. On the contrary, if a rule r_i^t is a perished rule, then the conditional part A and the consequent part B of r_i^t are different from those of any r_i^{t+k} in RS^{t+k}.*

For example, assume there are the following two rules: r_i^t : Bread $= high \to$ Milk $= Large$, and r_i^{t+k} : Vegetable $= high \to$ Apple $= High$. In this case, r_i^{t+k} is an added rule with respect to RS^t if the conditional part and the consequent part of r_i^t is different from those of all the rules (not only r_i^t) in RS^t.

2.2. **Fuzzy Data Mining.** The goal of data mining is to discover 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 [15]. Han et al. then proposed the Frequent-Pattern-tree (FP-tree) structure for efficiently mining association rules without generation of candidate itemsets [16]. Some other improved techniques were proposed based on the two.

The fuzzy set theory has recently been used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning [17][18][19][20]. When quantitative databases are processed, it is natural and informative to use fuzzy sets to represent quantities into linguistic terms. Several fuzzy data mining algorithms for inducing rules from a given set of data have thus been designed and used with good results for specific domains. For example, Hong et al. proposed a fuzzy mining algorithm to mine fuzzy rules from quantitative transaction data [3]. It used fuzzy membership functions to derive linguistic terms from quantitative data and applied the fuzzy minimum operator in the fuzzy set theory to evaluate the counts of fuzzy itemsets in a set of transactions. They also proposed an apriori-based mining algorithm to find fuzzy association rules efficiently. In addition, Hong et al. investigated the trade-off problem between the number of fuzzy rules and computation time [21]. Besides, Hong et al. also proposed a fuzzy weighted data mining approach based on the support-confidence framework to extract weighted association rules with linguistic terms from quantitative transactions [4].

Zheng et al. proposed a novel optimized fuzzy association rule mining method to mine association rules from quantitative data [20]. Wang et al. then proposed a data mining algorithm for extracting fuzzy knowledge from transactions stored as quantitative values [22]. Because of the success of fuzzy mining, many extend approaches are widely proposed.

3. **Generating Fuzzy Rules and Finding Rule Similarity.** In this section, we present the steps of generating fuzzy association rules and calculating the similarity of two fuzzy rules.

3.1. **Membership Functions.** Membership functions play a role in converting commodity items into fuzzy terms close to human semantics. Figure 1 shows an example of membership functions that apple was purchased in a transaction from a store. Figure 1 consists of three membership functions, representing low, medium and high according to the purchased amount. For example, if we buy five apples, the amount is low with a 0.4 fuzzy value, is middle with a 0.6 fuzzy value, and is high with a zero fuzzy value.

In this example, each membership function is designed as a triangle with two parameters, the center and the span. It can be given or learned from automatic methods [23]. Other types of membership functions can also be used in the proposed approach below.

3.2. **Generating Fuzzy Assocation Rules.** After the membership functions are set, we can then use our previous fuzzy mining algorithm [3][6] to generate fuzzy association rules. First, the transactions are first transformed into a fuzzy transactions using the given membership functions. Table 1 shows an example of a transaction database, in which a symbol in a transaction represents an item and a number denotes the purchased quantity of an item.

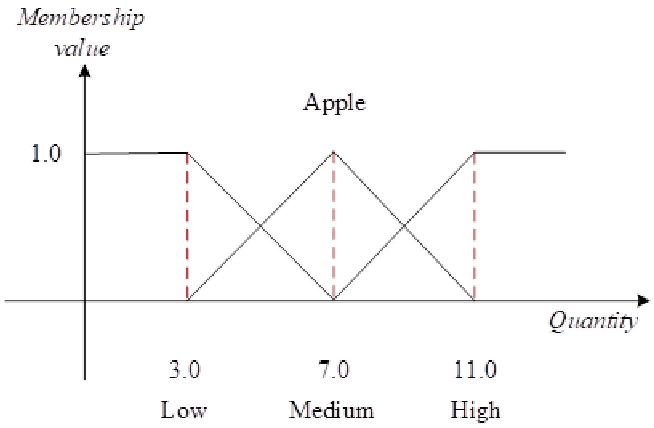

FIGURE 1. An example of membership functions.

TABLE 1. An example of a transaction database.

ID	Transaction Items
1	$(A, 3)(C, 6)(E, 9)$
2	$(B, 4)(C, 7)(D, 10)$
3	$(B, 2)(C, 5)(E, 8)$
4	$(C, 1)(E, 14)$

After the conversion with a given set of membership functions, we can get the fuzzy values of different linguistic terms of each item. The original transaction database can thus be converted into a database with fuzzy linguistic terms. The converted results for Table 1 are shown in Table 2.

TABLE 2. The fuzzy transactions converted from the transactions in Table 1.

TID	Fuzzy Transactions
1	$\left(\frac{1}{A.Low}\right)\left(\frac{0.25}{C.Low}+\frac{0.75}{C.Middle}\right)\left(\frac{0.5}{E.Middle}+\frac{0.5}{E.High}\right)$
2	$\left(\frac{0.75}{B.Low}+\frac{0.25}{B.Middle}\right)\left(\frac{1}{C.Middle}\right)\left(\frac{0.25}{D.Middle}+\frac{0.75}{D.High}\right)$
3	$\left(\frac{1}{B.Low}\right)\left(\frac{0.5}{C.Low}+\frac{0.5}{C.Middle}\right)\left(\frac{0.75}{E.Middle}+\frac{0.25}{E.High}\right)$
4	$\left(\frac{1}{C.Low}\right)\left(\frac{1}{E.High}\right)$

Then the cardinality of each fuzzy region (linguistic terms) in the fuzzy transaction is calculated by adding up the fuzzy values of the fuzzy region in all the transactions. For example, the cardinality of the fuzzy region *C.Middle* is 0.75+1+0.5, which is 2.25. After all the cardinalities of the fuzzy regions are found, they are checked with the threshold $n \times \alpha$, where n is the transaction number and α is the minimum threshold. If the cardinality of a fuzzy region is larger than or equal to the minimum count $n \times \alpha$, then it is a frequent (or called large) fuzzy region.

Next, the frequent fuzzy itemsets with more than one fuzzy region can be derived in a way similar to the Apriori or the FP-Tree approach except the fuzzy operation needs to be used to calculate the cardinalities of the fuzzy itemsets. For example, the cardinality of (*B.Middle*, *C.Middle*) in the second transaction in Table 2 is calculated as $min(0.25, 1)$, which is 0.25. In addition, the fuzzy terms from the same original item cannot be combined together. For example, *B.Low* and *B.Middle* cannot be combined and formed as a fuzzy itemset. After the frequent fuzzy itemsets are generated, the fuzzy association rules are obtained based on their confidence measures calculated from the conditional probability of the fuzzy itemsets in the rules.

3.3. **Calculating the Similarity of Fuzzy Rules.** In order to calculate fuzzy concept-drift patterns, we need to calculate the similarity of two fuzzy rules. We modify the formulas from [14] to calculate the matching similarity. First, the premise similarity (abbreviated as *ps*) of the premise (conditional) part in two fuzzy association rules is calculated as follows:

$$ps_{ij} = \begin{cases} \frac{\varrho_{ij} \times \sum_{S \in A_{ij}} x_{ijs}}{|A_{ij}|} & ,\text{if } |A_{ij}| \neq 0 \\ 0 & ,\text{if } |A_{ij}| = 0 \end{cases} \quad (1)$$

The notation in this formula is briefly explained as follows:

ps_{ij}: The degree of premise similarity between two rules r_i^t and r_j^{t+k} , where r_i^t and r_j^{t+k} are obtained from time t and $t+k$, respectively;

$|A_{ij}|$: The number of common items (not including linguistic value) in the premise parts of two rules r_i^t and r_j^{t+k};

ϱ_{ij}: The ratio of common items (not including linguistic value) in the premise parts of two rules r_i^t and r_j^{t+k}; It is calculated by the following:

$$\varrho_{ij} = \frac{|A_{ij}|}{max\left(\left|X_i^t\right|,\left|X_j^{t+k}\right|\right)},$$

where $|X_i^t|$ and $\left|X_j^{t+k}\right|$ are the numbers of items in the premise parts of r_i^t and r_j^{t+k}, respectively.

x_{ijs}: The linguistic-term match degree of the s-th matching item in A_{ij}; It can be calculated by considering fuzzy matching as follows:

$$x_{ijs} = 1 - \left[\frac{interval\_distance_{ijs}}{n_s - 1}\right]^{\beta},$$

where n_s is the number of membership functions of the s-th item in A_{ij}, $interval\_distance_{ijs}$ is the number of intervals between the linguistic values of the s-th item in the two rules r_i^t and r_j^{t+k}, and β is a parameter controlling the effect of different linguistic values. For example, if an attribute has only three linguistic terms: *High*, *Middle* and *Low*, then $n_s = 3$. The value of *interval_distance* between *High* and *Middle* is 1 and between *High* and *Low* is 2.

After the premise similarity of two fuzzy rules is defined, the consequent similarity (cs) of the consequent parts of two fuzzy association rules is designed as follows:

$$cs_{ij} = c_{ij} \times \left(1 - \left[\frac{interval\_distance_{ij}}{n-1}\right]^{\beta}\right),$$

where $c_{ij} = 1$ if the consequent items (not including linguistic values) of the two rules r_i^t and r_j^{t+k} are the same, and $c_{ij} = 0$ otherwise. Besides, n is the number of membership functions of the common consequent item.

The similarity s_{ij} of two rules r_i^t and r_j^{t+k} can be found by using the fuzzy intersection operator on ps_{ij} and cs_{ij}. If the minimum operator is used as the fuzzy intersection operator, then the following holds:

$$s_{ij} = min\left(ps_{ij}, cs_{ij}\right).$$

Note that the following product operator can also be used as the fuzzy intersection operator:

$$s_{ij} = ps_{ij} \times cs_{ij}.$$

Table 3 shows an example to calculate the similarity of two rules. The numbers in the rules represent the support values. Note the support value in the consequent part of a rule represents the support of a whole rule, which will be explained later.

Table 3. An example for calculating the similarity of two fuzzy association rules.

Database	Fuzzy Association Rule
D^t	$(A.Low, B.High, 0.5) \to (D.High, 0.4)$
D^{t+k}	$(A.Low, B.Middle, C.Low, 0.45) \to (D.Low, 0.3)$

Assume the number of membership functions for each item is 3. The premise similarity of two fuzzy rules is first calculated. In this example, $|A_{ij}|$ is 2 since there are two common terms, A and B, in the two rules. l_{ij} is 2/3 since the maximum number of items in the conditional parts of the two rules is 3. Let the parameter β is 1. Then x_{ijl} is 1 - 0/(3-1), which is 1, since the first matched item A has the same value (*Low*) in both the rules. x_{ij2} is 1 - 1/(3-1), which is 0.5, since the second matched item B has the value of *High* in one rule and has the value of *Middle* in the other rule. The premise similarity is thus calculated as follows:

$$ps_{ij} = \frac{\varrho_{ij} \times \sum_{S \in A_{ij}} x_{ijs}}{|A_{ij}|} = \frac{2/3 \times (1 + 0.5)}{2} = 0.5.$$

Next, the consequent similarity cs_{ij} of two fuzzy rules is calculated. In the example, c_{ij} is 1 since the consequent items in the two rules are the same (item D). Since the interval between *D.High* and *D.Low* is 2, cs_{ij} is then calculated as 1(1-2/2), which is 0. If we use the fuzzy intersection operator to calculate the similarity, then the similarity of the rules is $min(0.5, 0)$, which is 0.

4. **Finding Fuzzy Concept-drift Patterns.** In this section, we present the concept of the fuzzy concept-drift patterns for fuzzy association rules. We mainly generalize the original concept-drift patterns in [14] to quantitative transactions. The following different concept-drift patterns of fuzzy association rules are considered. The first one is the fuzzy emergent patterns in which both the conditional and the consequent parts between two fuzzy association rules from two different databases are similar but the fuzzy support values are quite different. The second one is the unexpected change of fuzzy association rules. It considers two fuzzy association rules in different databases with similar change of the conditional parts, but their consequent parts are quite different. The third one also considers the unexpected change of fuzzy association rules, but it handles the two rules with similar consequent parts but quite different conditional parts. The last one considers the added and perished fuzzy rules in which both the conditional and conditional parts are quite different. They are described below.

4.1. **The Fuzzy Emerging Patterns.** In fuzzy emerging patterns, both the conditional and the consequent parts between two fuzzy association rules from two different databases are similar, but their fuzzy support values are quite different. Since the confidence for an association rule $A \to B$ is calculated by dividing the support of $A\&B$ over the support of A, thus the change of both $A\&B$ and A is important. Therefore, we consider the following three kinds of fuzzy support change for fuzzy emerging patterns. The first case is that the fuzzy support values of the conditional parts between two fuzzy association rules are similar but the fuzzy support values of the consequent parts (including the fuzzy regions in the conditional parts) are different. Here, the fuzzy support value of the consequent part means the fuzzy support value of the whole rule because it is calculated under the consideration that the conditional part of the rule exists. The second case is that the fuzzy support values of the conditional parts are quite different, but the fuzzy support values of the consequent parts are similar. The third case is that the fuzzy support values of both the conditional parts and the consequent parts are quite different. Two fuzzy rules with similar support values in both the conditional and consequent parts are not considered since they do not change significantly and are thus not emerging patterns.

For each rule r_i^t in a database at time t, we will find the rule r_j^{t+k} with the maximum similarity to it in another database $t+k$. Similarly, for each rule r_j^{t+k} in a database at time $t+k$, we will find the rule r_i^t with the maximum similarity to it in another database t. If the s_{ij} value is equal to or larger than a predefined threshold value T, then the two rules are similar in structure. The supports of the two rules are then checked for finding emergent patterns according to the three cases mentioned above.

If the support growth ratio of the conditional part of the rule r_j^{t+k} at time $t+k$, when compared to the rule r_i^t at time t, is larger than or equal to a given growth threshold, and the support growth ratio of the consequent part is smaller than the threshold, then it is the first case of an emerging pattern. Below is an example to show this case. Assume two fuzzy association rules from two databases at different time are shown in Table 4.

TABLE 4. The first case of an emerging pattern for two fuzzy association rules.

Database	Fuzzy Association Rule
D^t	$(A.Low, B.High, 0.5) \rightarrow (D.High, 0.5)$
D^{t+k}	$(A.Low, B.High, 0.6) \rightarrow (D.High, 0.5)$

In Table 4, both the premise similarity and the conditional similarity between the two association rules are very similar (actually the same), and thus we further judge their fuzzy support growth ratio. The fuzzy support growth ratio at the premise part of the rule at time $t+k$ is (0.6-0.5)/0.5, which is 20%; The fuzzy support growth ratio at the consequent part of the rule at time $t+k$ is (0.5-0.5)/0.5, which is 0%. If we set the growth threshold at 10%, then the example is the first case of the emerging pattern.

For the second case of an emerging pattern, the support growth ratio of the conditional part of the r_j^{t+k} at time $t+k$ is smaller than a given growth threshold, and the support growth ratio of the consequent part is larger than or equal to the threshold. The two fuzzy association rules in Table 5 show this case. The premise similarity and the consequent similarity of the two rules is 0.75 and 1. The similarity between the two association rules is then $min(0.75, 1)$, which is 0.75. If the similarity threshold is set at 0.6, then the two rules are similar, and thus we judge their fuzzy support change for emerging patterns.

TABLE 5. The second case of an emerging pattern for two fuzzy association rules.

Database	Fuzzy Association Rule
D^t	$(A.Low, B.Middle, 0.7) \rightarrow (D.High, 0.5)$
D^{t+k}	$(A.Low, B.High, 0.7) \rightarrow (D.High, 0.6)$

Both the premise similarity and the conditional similarity between the two association rules are very similar (actually the same), thus we judge their fuzzy support change. The fuzzy support growth ratio at the premise part of the rule at time $t+k$ is (0.7-0.7)/0.7, which is 0%; The fuzzy support growth ratio at the consequent part of the rule at time $t+k$ is (0.6-0.5)/0.5, which is 20%. If we set the growth threshold at 10%, then the example is the second case of the emerging pattern.

At last, if the support growth ratios of both the conditional and the consequent parts of a rule r_j^{t+k} is larger than or equal to a given growth threshold, then it is the third case of an emerging pattern. Table 6 shows the third case of an emerging pattern for two fuzzy association rules when the growth threshold is set at 10%.

Table 6. The third case of an emerging pattern for two fuzzy association rules.

Database	Fuzzy Association Rule
D^t	$(A.Low, B.High, 0.6) \rightarrow (D.High, 0.5)$
D^{t+k}	$(A.Low, B.High, 0.8) \rightarrow (D.High, 0.7)$

4.2. **The Unexpected Change for Fuzzy Association Rules.** There are two kinds of fuzzy concept drift patterns for unexpected change. The first one is that the premise similarity between two fuzzy association rules is large, but their consequent similarity is small. The second one is the contrary. That is, the consequent similarity between two fuzzy association rules is large, but their premise similarity is small. It can be judged by the similarity threshold.

Below is an example to show the first case. In Table 7, there are two fuzzy association rules. Assume there are three membership functions for each item. The premise similarity of the two rules is 0.66, and the consequent similarity is 0. If the similarity threshold is set at 0.6, then the premise parts of the two rules are similar, but the consequent part is very different. Thus, the example is an unexpected consequent change.

Table 7. The unexpected consequent change of two fuzzy association rules.

Database	Fuzzy Association Rule
D^t	$(A.Low, B.High, C.Middle, 0.6) \rightarrow (D.High, 0.5)$
D^{t+k}	$(A.Low, B.Middle, C.Low, 0.5) \rightarrow (D.Low, 0.3)$

Another example is shown in Table 8. In this case, since the items D and E in the consequent parts are different, their similarity is 0. In a classification problem, this case is thought of as the unexpected consequent change because there is only a consequent item for the same or similar premise. In an association-rule mining problem, this case may often happen because the same premise may derive different items. In addition to the pair (r_i^t, r_j^{t+k}) in Table 8, if we can find at least one more rule at time t which is similar to r_j^{t+k} in both the premise part and the consequent part, or if we can find at least one more rule at time $t+k$ which is similar to r_i^t in both the premise part and the consequent part, then the pair (r_i^t, r_j^{t+k}) is not an unexpected consequent change; otherwise, (r_i^t, r_j^{t+k}) is regarded as an unexpected consequent change.

Table 8. Different consequent items of two fuzzy association rules.

Database	Fuzzy Association Rule
D^t	$(A.Low, B.High, C.Middle, 0.6) \rightarrow (D.High, 0.5)$
D^{t+k}	$(A.Low, B.Middle, C.Low, 0.5) \rightarrow (E.High, 0.3)$

For example in Table 9, in which one more rule is in D^t, the new rule is similar to r_j^{t+k} in both the premise and consequent parts. Thus, the original pair (r_i^t, r_j^{t+k}) is not regarded as unexpected consequent change. However, if no rule in D^t is similar to r_j^{t+k} in the premise and consequent parts, then r_j^{t+k} is the unexpected consequent change.

Table 9. An example of not unexpected consequent change.

Database	Fuzzy Association Rule
D^t	$(A.Low, B.High, C.Middle, 0.6) \rightarrow (D.High, 0.5)$
D^t	$(A.Middle, B.Middle, C.Low, 0.6) \rightarrow (D.Low, 0.5)$
D^{t+k}	$(A.Low, B.Middle, C.Low, 0.5) \rightarrow (D.Low, 0.3)$

The other case is that the premise similarity between two fuzzy association rules is small, but their consequent similarity is large. An example in Table 10 shows this case.

Table 10. An example of unexpected premise change for two fuzzy rules.

Database	Fuzzy Association Rule
D^t	$(A.Middle, B.Middle, C.Low, 0.6) \rightarrow (D.Low, 0.5)$
D^{t+k}	$(A.Low, B.Middle, C.High, 0.5) \rightarrow (D.Low, 0.4)$

Assume there are three membership functions for each item. The premise similarity of the two rules is 0.5, and the consequent similarity is 1. If the similarity threshold is set at 0.6, then the premise parts of the two rules are different, but the consequent part is similar. Thus, the example is an unexpected premise change.

4.3. **The Added and Perished Fuzzy Association Rules.** These kinds of fuzzy rules are defined here based on the non-fuzzy rules in [14]. If a fuzzy rule r_i^{t+k} at time $t+k$ is an added fuzzy rule, then the premise part and the consequent part of r_i^{t+k} are different from those of any fuzzy rule r_i^t at time t. On the contrary, if a fuzzy rule r_i^t at time t is a perished rule, then the premise part and the consequent part of r_i^t are different from those of any fuzzy rule r_i^{t+k} at time $t+k$.

For example in Table 11, both D^t and D^{t+k} contain only two rules. When the similarity threshold is set at 0.6, the first fuzzy rule in D^t is not a perished fuzzy rule because it is similar to the first rule in D^{t+k} in both the premise and consequent parts. Their premise similarity is 0.75 and their consequent similarity is 1. The rule will be checked as an emergency pattern according to the support change, rather than a perished fuzzy rule.

The second rule in D^t is a perished fuzzy rule because it is not similar to any rule in D^{t+k} in both the premise and consequent parts. The first rule in D^{t+k} is not an added fuzzy rule because it is similar to the first rule in D^t in both the premise and consequent parts. The second rule in D^{t+k} is an added fuzzy rule because it is not similar to any rule in D^t in both the premise and consequent parts.

Table 11. An example of added and perished fuzzy association rules.

Database	Fuzzy Association Rule
D^t	$(A.Low, E.Middle, 0.6) \rightarrow (D.Low, 0.5)$
D^t	$(A.Middle, B.Middle, C.Low, 0.6) \rightarrow (D.High, 0.5)$
D^{t+k}	$(A.Low, E.Low, 0.7) \rightarrow (D.Low, 0.6)$
D^{t+k}	$(B.Low, F.Low, 0.5) \rightarrow (D.Middle, 0.3)$

5. **The Proposed Algorithm.** In this section, the proposed approach to find fuzzy concept-drift patterns is summarized in Algorithm 1.

Algorithm 1 Mining Fuzzy Concept-Drift Patterns

Input: Two quantitative transaction databases D^t and D^{t+k}, where D^t consists of n quantitative transactions at time t and D^{t+k} consists of m quantitative transactions at time $t+k$.

Output: The fuzzy concept-drift patterns.

1: **Step 1:** Find fuzzy frequent itemsets L^t with fuzzy supports and fuzzy association rules from D^t based on the given membership functions by a fuzzy mining approach.
2: **Step 2:** Find fuzzy frequent itemsets L^{t+k} with fuzzy supports and fuzzy association rules from D^{t+k} based on the given membership functions by a fuzzy mining approach.
3: **Step 3:** For each pair of fuzzy rules r_i^t and r_j^{t+k}, where r_i^t is in R^t and r_j^{t+k} is in R^{t+k}, calculate its premise similarity ps_{ij}, consequent similarity cs_{ij} and rule similarity s_{ij}.
4: **Step 4:** For each rule r_i^t, find the rule in R^{t+k} with the maximum rule similarity, and for each rule r_j^{t+k}, find the rule in R^t with the maximum rule similarity. If the maximum rule similarity is larger than or equal to the similarity threshold, calculate the support growth ratios of the premise and the consequent parts, and decide whether it is an emerging pattern and which kind if it is.
5: **Step 5:** For each pair of fuzzy rules r_i^t and r_j^{t+k}, if ps_{ij} is larger than or equal to the similarity threshold and cs_{ij} is smaller than the threshold T_c, then it is the unexpected consequent change; If ps_{ij} is smaller than the similarity threshold and cs_{ij} is larger than the threshold, then it is the unexpected premise change.
6: **Step 6:** For each rule r_i^t, if its ps_{ij} and cs_{ij} are both smaller than the similarity threshold for any rule r_i^{t+k} in R^{t+k}, then r_i^t is a perished fuzzy rule.
7: **Step 7:** For each rule r_j^{t+k}, if its ps_{ij} and cs_{ij} are both smaller than the similarity threshold for any rule r_i^t in R^t, then r_j^{t+k} is an added fuzzy rule.
8: **Step 8:** Output the fuzzy concept-drift patterns to users.

Note that in Step 5 of the algorithm, some additional checking may be needed for unexpected consequent change according to the discussion in Section 4.2.

6. **Conclusion.** In this paper, we generalize the traditional concept-drift mining approach by fuzzy sets and integrate it with our fuzzy data mining mechanism, which considers not only items but also quantities and linguistic terms. We modify the calculation of premise similarity, consequent similarity and rule similarity by considering the relationship of fuzzy regions of items. We use these similarity measures to decide different kinds of fuzzy concept-drift patterns and give some examples to illustrate them. An algorithm is also stated step by step to effectively get the patterns. The patterns can reflect the implicit knowledge change of transaction databases at different times. In the future, we would like to apply the proposed approach to other practical applications, such as observing the change of customers behavior in different years or different seasons. In addition, we will study how to use good data structures to design efficient implementation for decreasing the computational time.

REFERENCES

[1] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 54–77, 2017.
[2] C. M. Kuok, A. Fu, and M. H. Wong, "Mining fuzzy association rules in databases," *ACM Sigmod Record*, vol. 27, no. 1, pp. 41–46, 1998.

[3] T.-P. Hong, C.-S. Kuo, and S.-C. Chi, “Mining association rules from quantitative data,” *Intelligent Data Analysis*, vol. 3, no. 5, pp. 363–376, 1999.

[4] T.-P. Hong, M.-J. Chiang, and S.-L. Wang, “Fuzzy weighted data mining from quantitative transactions with linguistic minimum supports and confidences,” *International Journal of Fuzzy Systems*, vol. 8, no. 4, pp. 173–182, 2006.

[5] H. Jin, J. Sun, H. Chen, and Z. Han, “A fuzzy data mining based intrusion detection model,” in *The International Workshop on Future Trends of Distributed Computing Systems*, pp. 191–197, IEEE, 2004.

[6] C.-W. Lin, T.-P. Hong, and W.-H. Lu, “An efficient tree-based fuzzy data mining approach,” *International Journal of Fuzzy Systems*, vol. 12, no. 2, pp. 150–157, 2010.

[7] E. Apeh and B. Gabrys, “Change mining of customer profiles based on transactional data,” in *The International Conference on Data Mining Workshops*, pp. 560–567, IEEE, 2011.

[8] P. K. Bala, “Mining changes in purchase behavior in retail sale with products as conditional part,” in *The International Advance Computing Conference*, pp. 78–81, IEEE, 2010.

[9] C. Wang and Y. Li, “Mining changes of E-shopper purchase Behavior in B2c,” in *The International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 2, pp. 240–244, IEEE, 2008.

[10] A. Tsymbal, “The problem of concept drift: definitions and related work,” *Computer Science Department, Trinity College Dublin*, vol. 106, no. 2, 2004.

[11] W.-H. Au and K. C. Chan, “Fuzzy data mining for discovering changes in association rules over time,” in *The International Conference on Fuzzy Systems*, vol. 2, pp. 890–895, IEEE, 2002.

[12] A. Hora, N. Anquetil, S. Ducasse, and M. T. Valente, “Mining system specific rules from change patterns,” in *The Working Conference on Reverse Engineering*, pp. 331–340, IEEE, 2013.

[13] L.-C. Cheng and M.-T. Lai, “Mining the change of consensus from group ranking decisions,” in *The International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 3, pp. 1459–1463, IEEE, 2011.

[14] H. S. Song, J. kyeong Kim, and S. H. Kim, “Mining the change of customer behavior in an internet shopping mall,” *Expert Systems with Applications*, vol. 21, no. 3, pp. 157–168, 2001.

[15] R. Agrawal, T. Imieliski, and A. Swami, “Mining association rules between sets of items in large databases,” in *ACM Sigmod Record*, vol. 22, pp. 207–216, ACM, 1993.

[16] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” in *ACM Sigmod Record*, vol. 29, pp. 1–12, ACM, 2000.

[17] A. Mangalampalli and V. Pudi, “Fuzzy association rule mining algorithm for fast and efficient performance on very large datasets,” in *The International Conference on Fuzzy Systems*, pp. 1163–1168, IEEE, 2009.

[18] K. Noori and K. Jenab, “Fuzzy reliability-based traction control model for intelligent transportation systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 1, pp. 229–234, 2013.

[19] B.-Y. Wang and S.-M. Zhang, “A mining algorithm for fuzzy weighted association rules,” in *The International Conference on Machine Learning and Cybernetics*, vol. 4, pp. 2495–2499, IEEE, 2003.

[20] H. Zheng, J. He, G. Huang, and Y. Zhang, “Optimized fuzzy association rule mining for quantitative data,” in *The International Conference on Fuzzy Systems*, pp. 396–403, IEEE, 2014.

[21] T.-P. Hong, C.-S. Kuo, and S.-C. Chi, “Trade-off between computation time and number of rules for fuzzy mining from quantitative data,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 05, pp. 587–604, 2001.

[22] S.-L. Wang, C.-Y. Kuo, and T.-P. Hong, “Mining fuzzy similar sequential patterns from quantitative data,” in *The International Conference on Systems, Man and Cybernetics*, vol. 7, pp. 5–pp, IEEE, 2002.

[23] T.-P. Hong, C.-H. Chen, and V. S. Tseng, “Genetic-fuzzy data mining techniques,” in *Encyclopedia of Complexity and Systems Science*, pp. 4145–4160, Springer, 2009.