Temporal fuzzy association rule mining with 2-tuple linguistic representation

MATTHEWS, Stephen G., GONGORA, Mario A., HOPGOOD, Adrian A. and AHMADI, Samad

Citation:

Copyright and re-use policy

Temporal Fuzzy Association Rule Mining with 2-tuple Linguistic Representation

Stephen G. Matthews*, Mario A. Gongora*, Adrian A. Hopgood† and Samad Ahmadi*
*Centre for Computational Intelligence, De Montfort University, UK
Email: sgm@dmu.ac.uk, mgongora@dmu.ac.uk, sahmadi@dmu.ac.uk
†Sheffield Business School, Sheffield Hallam University, UK
Email: a.hopgood@shu.ac.uk

Abstract—**This paper reports on an approach that contributes towards the problem of discovering fuzzy association rules that exhibit a temporal pattern. The novel application of the 2-tuple linguistic representation identifies fuzzy association rules in a temporal context, whilst maintaining the interpretability of linguistic terms. Iterative Rule Learning (IRL) with a Genetic Algorithm (GA) simultaneously induces rules and tunes the membership functions. The discovered rules were compared with those from a traditional method of discovering fuzzy association rules and results demonstrate how the traditional method can loose information because rules occur at the intersection of membership function boundaries. New information can be mined from the proposed approach by improving upon rules discovered with the traditional method and by discovering new rules.**

I. INTRODUCTION

Knowledge discovery in databases is the process of acquiring useful information from databases [1]. Data mining is one step of this process that seeks to discover knowledge that is accurate, comprehensible and interesting [2]. There are two tasks of data mining that are distinguished by the use of the information discovered: predictive for classification or prediction tasks, and descriptive for providing information about patterns and relationships present in data.

Association rule mining is a descriptive data mining task that identifies significant correlations between items in transactional data [3], which is often referred to as market basket analysis. Fuzzy sets [4] are used to model quantitative attributes with fuzzy association rule mining [5]. An example of a typical fuzzy association rule is 20% of customers matched the rule below.

IF *quantity of pizza is high* THEN *quantity of beer is high*

Applications of association rule mining are found in domains such as business, bioinformatics, environmental monitoring and network security, to mention a few. This paper focuses on an extension of fuzzy association rules where the rules occur more frequently in a temporal period of a dataset, e.g., the rule above may occur frequently over the weekend.

This paper uses fuzzy sets to represent numeric values with linguistic labels [6] so they are more comprehensible and interpretable. This is common in the area of association rule mining [5], [7] but also more generally in predictive tasks, e.g., [8], [9], [10]. For mining fuzzy association rules in the traditional manner, the following procedure is often used.

1) Define linguistic labels and membership function parameters.
2) Mine the rules using the linguistic labels.

It is this two stage procedure that presents an interesting problem because some temporal rules can be lost as a result of the first stage. The traditional method assumes that the membership functions are static, meaning they do not change between when the first and last transactions occurred, and so they hold across the entire dataset. However, different membership functions to those that were defined before the mining process may yield more significant rules in some temporal periods of the dataset. For example, the membership function drawn with a dashed line in Figure 1 was not found before the mining process, but, it appears more frequently in transactions in a temporal period of a dataset.
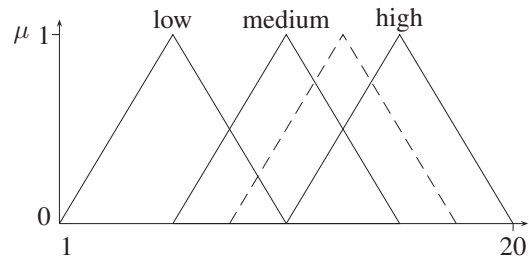

Fig. 1. Example of membership function occurring (dashed line) on intersection of two adjacent membership functions in a temporal period of a dataset.

Figure 1 shows how some rules may not be represented fully because membership functions may lie on the intersection of membership function boundaries. Although traditional methods do find temporal patterns of fuzzy association rules, they may not discover all significant patterns because of this problem. This paper addresses the problem of how to define these membership functions in a temporal context and how to discover rules associated with them.

Previous work [11], [12] has tackled this problem with methods that focus more on accuracy. Other work [13] has used the 2-tuple linguistic representation has previously been used to achieve good accuracy without a significant loss in interpretability. In this paper, the 2-tuple linguistic representation [14] maintains interpretability of knowledge and investigates its use within a temporal context to find rules.

This paper is presented as follows. Section II provides an overview of preliminaries of association rule mining. The novel concept of our approach is described in Section III. In Section IV the experimental methodology and results are discussed, and conclusions are drawn in Section V.

II. ASSOCIATION RULE MINING

Association rule mining is an exploratory and descriptive rule induction process of identifying significant correlations between items in Boolean transaction datasets [3]. Association rules are expressed as an implication of the form $X \Rightarrow Y$ where the consequent and antecedent are sets of Boolean items where $X \cap Y = \emptyset$.

A dataset contains a set of N transactions $T = \{t_1, t_2, ..., t_N\}$ where each transaction comprises a subset of items, from M items $I = \{i_1, i_2, ..., i_M\}$, referred to as an itemset. To extract association rules from datasets the support-confidence framework was introduced with the Apriori algorithm in [15]. The support count for an itemset, $\sigma(X)$, is defined as

$$\sigma(X) = |\{t_i | X \subseteq t_i, t_i \in T\}|. \tag{1}$$

The support measure determines the strength of a relationship by evaluating how often the rule occurs and is defined as

$$s(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{N}. \tag{2}$$

The confidence measure determines how frequently the items in the consequent occur in transactions containing the antecedent and is defined as

$$c(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}. \tag{3}$$

These measures have minimum thresholds that are used to extract rules from the dataset with a deterministic method, such as the Apriori algorithm. The background behind the two extensions is now discussed.

A. Fuzzy Association Rules

Fuzzy sets are used to model the quantities of items in an association rule, e.g., *large* amount of pizza. A linguistic representation describes quantities of an item in a way that is more interpretable and comprehensible for humans [9].

Using quantitative information of items in association rules was first realised by [16] where the quantitative values were discretised into intervals with uniform partitions. Fuzzy association rules were introduced with the F-APACS algorithm [5] to express quantitative attributes with linguistic labels in a way that is more natural to human reasoning and to overcome issues with discovering rules because of the crisp boundaries of attribute intervals. This represented quantitative attribute values of rules with linguistic labels modelled by fuzzy sets to enhance the interpretability and to better handle inaccuracies in physical measurements.

In [17] it was recognised that preprocessing the data to define attribute intervals can loose information because the generation of rules is limited to the crisp boundaries of discretised intervals and this does not allow for a concise representation of some rules. Over the past decade, computational intelligence has been used to overcome this problem where tasks focus on searching for optimal intervals, inducing rules and also modelling quantities with fuzzy sets. The synergy of evolutionary computation and fuzzy sets has become popular for such tasks [18]. Evolutionary algorithms are suitable for association rule mining because they can search complex spaces and they address difficult optimisation problems, which has led to much recent interest in this data mining problem [19].

Some methods for defining membership functions for fuzzy association rules are clustering [20], expert knowledge [5] and GAs[7], [21]. These approaches define the membership functions first before exhaustively searching for fuzzy association rules. A GA is a metaheuristic global search method based on the principles of natural selection and genetics [22]. GAs have proven to be effective search methods for a variety of problems using fuzzy rules [19]. Simultaneously evolving both membership functions and fuzzy rules is a common approach in fuzzy rule-based systems (FRBSs) [19], particularly for FRBS controllers [23], approximate interval based association rules [17] and approximate fuzzy association rules [11], [12]. In these works the purpose of simultaneously evolving both the definition of membership functions and induction of rules leans more towards improving accuracy. Previous approaches for discovering temporal patterns in fuzzy association rules [11], [12] use an approximate approach because the focus on accuracy allows the discovery of hidden temporal patterns. In this paper a GA simultaneously induces rules and tunes membership functions where the main contribution is that the interpretability is maintained. Similar works that simultaneously evolve parts of rules are those that select rules from a rule base (not rule induction) and also tune membership functions. Some of the many [24] approaches for maintaining interpretability of a FRBS include the use of linguistic hedges [9] and the 2-tuple linguistic representation [8].

A review of fuzzy association rule mining with evolutionary algorithms can be found in [25].

B. Temporal Association Rules

The term temporal association rules can cover a broad area of temporal data mining [26]. This paper focuses on association rules where the frequency of their occurrence (i.e., support) changes throughout a temporal dataset.

Methods based on the support-confidence framework do not capture rules that fall below the minimum support threshold. However, some of these rules may have a relatively high support over a short period of time so these are known as temporal association rules. The *lifespan* property [27] is an example that measures support from when the items were available to when they stopped being available, or taken off the shelf. This captures an element of the dynamic nature of a dataset.

A step towards analysing areas of the dataset where rule support changes throughout an item's lifespan of its availability is cyclic association rule mining [28]. The dataset is partitioned to a desired time granularity and rules are induced from each partition. Support values of association rules in the partitions are represented as binary sequences and pattern matching identifies cyclical patterns. These are fully periodic rules because they repeatedly occur at regular intervals. Partially periodic rules [29] relax the regularity found in fully periodic rules so the cyclic behaviour is found in only some segments of the dataset and is not always repeated regularly. Calendar-based schemas can be used [30] to define the temporal intervals so the method is less restrictive and reduces the requirement of prior knowledge. These works illustrate the types of temporal patterns that can potentially be extracted with our proposed method.

III. LATERAL DISPLACEMENT OF MEMBERSHIP FUNCTIONS

The aim of this paper is to find fuzzy association rules that have a temporal pattern whilst maintaining the interpretability of the linguistic labels. Traditional methods define the membership functions before the mining process and this restricts the temporal patterns that can be discovered. Some fuzzy association rules can have stronger temporal patterns in a temporal period of the dataset because different membership functions are used. These temporal patterns can be discovered by simultaneously inducing rules and tuning the membership functions with a GA. The 2-tuple linguistic representation is used to tune the membership functions within the context of a temporal period. This captures temporal patterns that can occur on the intersections of membership function boundaries. In this section, the 2-tuple linguistic representation is introduced and then the GA for mining temporal fuzzy association rules is presented.

A. The 2-tuple linguistic representation

A 2-tuple fuzzy set is a linguistic representation based on a symbolic translation of a fuzzy set [14]. A symbolic translation is the lateral displacement of the fuzzy set within the interval $[-0.5, 0.5)$ that expresses the domain of a term when it is displaced between two linguistic labels. It is a fuzzy set that maintains its shape whilst it is shifted left or right from its original membership function. A 2-tuple fuzzy set is defined as

$$(s_j, \alpha_j), s_j \in S, \alpha_j \in [-0.5, 0.5), \qquad (4)$$

where S represents a set of linguistic labels and α is the lateral displacement of a linguistic label. Figure 2 is an example of 3 membership functions, where s_1 (grey) is laterally displaced (light grey) to give a 2-tuple membership function, $(s_1, -0.3)$.

The 2-tuple linguistic representation was proposed by [14] for computing words. The computational methods for computing with words can produce a loss of information and the 2-tuple linguistic representation is used to overcome this limitation [14]. Since then, the 2-tuple linguistic representation
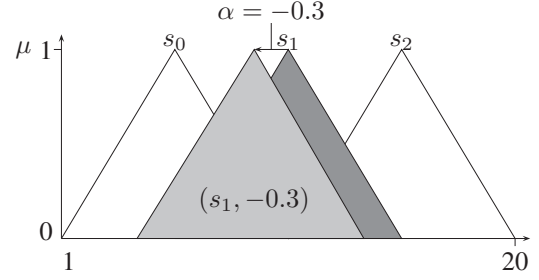

Fig. 2. Example of 2-tuple membership function, $(s_1, -0.3)$.

has been used for FRBSs in control and regression problems, which are demonstrated in [8]. The 2-tuple linguistic representation has been used as a postprocessing step for tuning linguistic rules to improve accuracy whilst maintaining interpretability of rules [8]. This was later applied to fuzzy association rules in [13] to improve rule quality. The initial fuzzy sets were uniformly partitioned and a GA learnt the lateral displacement.

It is crucial that the meaning of linguistic labels is maintained because this is a descriptive data mining process where interpretability is important. Approximate fuzzy models (typically for regression, control and classification) focus on accuracy and tuning interpretability [19, p. 19], but this work focuses on interpretability and tuning accuracy. With a linguistic representation, particularly for Mamdani FRBSs, there is a lack of flexibility of input and out spaces [21, p. 16]. In this paper the 2-tuple linguistic representation overcomes this by allowing flexibility within a temporal context. The interpretability of linguistic labels is maintained and the accuracy is tuned to temporal periods of the data set using the 2-tuple linguistic representation.

B. Evolving rules and lateral displacements

In this paper the GA finds fuzzy association rules that exhibit temporal patterns. The GA is well suited to this problem because the combination of temporal and fuzzy association rules presents a challenging problem with a complex search space. The GA serves two purposes in this approach; it simultaneously tunes the lateral displacements of membership functions and also finds fuzzy association rules. The GA works by tuning 2-tuple membership functions of a rule in a random temporal partition of the dataset so the rule is specific to that temporal partition. The result is that new rules can be induced with higher temporal support for that partition.

The GA is based on CHC [31], an algorithm whose key differences from a traditional GA (e.g., [22]) is that it uses no mutation and has a restart approach. The CHC algorithm is chosen because it has slow convergence and can run for a long time to achieve higher quality solutions [8]. Rules are coded with the Michigan method of representing one rule with one chromosome and an IRL approach is incorporated into CHC.

Chromosome

A chromosome, C, has mixed types and is defined as

$$C = (e_l, e_u, i_1, s_1, \alpha_1, a_1, \ldots, i_k, s_k, \alpha_k, a_k),$$

where the lower temporal endpoint is e_l, the upper endpoint is e_u, i is the item (e.g., pizza), s is the linguistic label for that item (e.g., lots), α is the lateral displacement of that linguistic label, a determines the antecedent or consequent part and k is the number of items in a rule. All parts are randomly initialised within their bounds. Minimum temporal support [11] restricts the length of the endpoints, otherwise the GA evolves towards the smallest temporal period.

The linguistic labels are modelled with symmetric triangular fuzzy sets so the lateral displacement can be encoded with one parameter. This approach has the advantage of reducing the search space by removing other parameters of a membership function from the chromosome. This is particularly important because it counteracts the increase in space arising from the need to simultaneously search for rules, tune the membership functions and search the temporal space. Not all items are represented in the chromosome because there can be a large number of items to consider in real-world market basket applications. So, items are randomly selected to appear in chromosomes during initialisation and restarts.

Fitness Evaluation

Fitness of a chromosome is the addition of temporal fuzzy support (modified from [27] to include fuzzy sets) and confidence, and is defined as

$$\text{Fitness}(C) = \left(\frac{\sum_{j=e_l}^{e_u} \text{FuzzySupport}(C_X^{(j)} \cap C_Y^{(j)})}{e_u - e_l} \right) \quad (5)$$
$$+ \left(\frac{\sum_{j=e_l}^{e_u} \text{FuzzySupport}(C_X^{(j)} \cap C_Y^{(j)})}{\sum_{j=e_l}^{e_u} \text{FuzzySupport}(C_X^{(j)})} \right),$$

where C is a chromosome, X is the antecedent fuzzy itemset, Y is the consequent fuzzy itemset and j is a dataset transaction between the e_l lower endpoint and the e_u upper endpoint. The FuzzySupport [32] uses minimum for the intersection operation.

Selection

Selection combines both the offspring population and the parent population that then compete to form the next population. The key difference from other GAs is that the competition occurs across generations rather than competing only amongst the offspring population. Elitist selection is applied.

Restart

CHC is particularly good at maintaining diversity and so mutation is not used [31]. Instead mutation is introduced in the form of a restart operator only when the population has converged (not termination criteria). When a population is restarted each individual is reinitialised, except the best individual, this is just copied, and the algorithm continues. The best individual is used as a template for creating the other individuals. CHC does this by flipping a percentage of bits in a binary representation, this was referred to as

divergence rate [31]. CHC uses a binary representation but here we use a mixed representation of interval data types (lateral displacement) and nominal data types (item, attribute, time). Bits should not be flipped for nominal types because there is no order amongst elements. So in this paper divergence rate is redefined as a threshold for determining the probability that a gene is reinitialised.

Population convergence is measured by the number of generations where no new individuals are introduced. An incest prevention mechanism uses a difference threshold that is decremented by 5% at every generation when there is no new individual and once it drops below 0% the population is restarted. This incest prevention mechanism is linked with the crossover operator. Crossover is only performed on two individuals when the difference in genotypes is above the difference threshold. The purpose is so that only very different chromosomes are crossed over. In the original CHC algorithm the Hamming distance is used on bit strings. As in [8], [13], the Gray Code is used for genes that are interval data types (lateral displacement and endpoints), allowing the Hamming distance to be used. For coherence, only lateral displacements of the same item–attribute pair are compared and if they do not match then the maximum difference in Gray Code is assigned for each chromosome (e.g., 0s for a chromosome and 1s for the other). The representation used in this paper is mixed where the combination of item, fuzzy label and antecedent/consequent clauses are nominal, so the Jaccard distance is used to provide a measure of dissimilarity. The measures are normalised and aggregated with the arithmetic mean, and this is then used with the difference threshold.

Reproduction

Only the crossover operator is used in CHC. The chromosome has mixed data types and so crossover is a hybrid method of the parent centric BLX-α (PCBLX-α) operator [33] for genes with continuous data and a method of swapping genes for discrete data. The steps involved in crossover are presented here.

STEP 1: Crossover the endpoints by randomly swapping both lower and upper endpoints, and perform the following substeps.

STEP 1.1: If endpoints are the same. Add/subtract minimum temporal support to/from the lower/upper.
STEP 1.2: If upper endpoint subtracted lower endpoint is greater than the minimum temporal support.
STEP 1.3: If the lower endpoint is larger than the lower endpoint.

STEP 2: For each part of a rule (combination of item, attribute, lateral displacement and antecedent/consequent clause) in the chromosome, perform the following substeps.

STEP 2.1: If the both items and attributes are the same then copy items and attributes into the offspring, crossover the lateral displacement using PCBLX-α and crossover the consequent flag by random swap.

STEP 2.2: If the items are the same but the attributes are different then copy the items into the offspring, crossover the attributes by randomly swapping the attributes and their corresponding lateral displacement values, and crossover the consequent flag by random swap.

STEP 2.3: If the items are different then crossover the items by randomly swapping the items, and their corresponding attributes and lateral displacements.

Iterative Rule Learning

IRL is a process where one rule is chosen from one run of a GA. The GA is run many times to produce a set of rules. This is an approach first used in [10] where a GA continues to extract classification rules when there are still examples labelled "uncovered". Other methods penalise previously covered rules (e.g., [17]). In this paper IRL is performed by maintaining a set of rules evolved from each run of the GA, this final rule set contains all discovered rules and is considered the final result of this data mining method. Chromosomes are penalised in the fitness function if the candidate rule matches a rule from the rule set. A match is determined by comparing clauses of a rule where the item, attribute and antecedent/consequent flag are considered to be a single clause that are compared. The lateral displacements of two clauses are then compared and they are considered the same if the difference in absolute values of lateral displacements is less than 0.5. For example, the absolute difference of the lateral displacements -0.45 and 0.25 is 0.7 so the fuzzy sets are considered different. Candidate chromosomes that have previously been discovered, and are present in the rule set, are penalised by setting their fitness to 0. This penalisation method helps to guide search away from previously discovered rules so that the final rule set is diverse.

IV. EXPERIMENTS

An experiment was run to demonstrate improvements over traditional methods and how extra knowledge can be gained with the method proposed in this paper. The experiment was conducted with software modified from the KEEL tool (Knowledge Extraction based on Evolutionary Learning) [34].

A. Dataset

The dataset was produced from the IBM synthetic dataset generator [15] and can be considered as a benchmark because it is used in many works such as [15], [7], [28], [30], [11], [12]. This is a market basket dataset that consists of the items and quantities of items sold in every shopping basket. Quantitative values were assigned randomly to items in a similar manner as [7] and the parameters used are: 10,000 transactions, 64 items, with quantities in the range 1–20.

B. Methodology

A comparison was performed between a traditional approach to mining fuzzy association rules and the approach proposed in this paper. The purpose was to discover new descriptive knowledge by:

- an improvement in temporal fuzzy support of existing rules discovered by the traditional approach.
- discovering new rules that were lost with the traditional approach but then discovered with our proposed approach.

The CHC algorithm has already been described in Section III and so the traditional algorithm will be discussed here with details of its configuration and parameters.

The FuzzyApriori algorithm [32] is an extension to the classical Apriori algorithm [15] that mines fuzzy association rules. FuzzyApriori uses a breadth-first search to find all fuzzy association rules that are above user specified thresholds, minimum support and minimum confidence. This is the traditional method that is compared with the CHC algorithm. FuzzyApriori only discovers fuzzy association rules and not rules that are temporal. To find temporal fuzzy association rules with FuzzyApriori an exhaustive search of dataset partitions is conducted. The dataset is partitioned according to a temporal granularity and FuzzyApriori is applied to each partition separately. This is similar to the approach in [28], except our method searches for single temporal patterns in one partition only and not across many partitions as is the case for cyclic rules. The time granularity is the same as the minimum temporal support so the partitions used in FuzzyApriori are directly comparable with the lower and upper endpoints evolved with CHC. Partitions of the dataset are created by enumerating all partition sizes, of granularity equal to minimum temporal support, and enumerating all possible starting positions. This ensures every temporal period is covered at this level of granularity. The rules discovered largely depend on how the dataset is partitioned, and so in practise, various levels of granularity would be used to gain information relevant to an application.

Uniform discretisation was applied to the dataset to produce membership functions for 3 linguistic labels before running both the FuzzyApriori and CHC algorithm. All methods of discretisation evaluate the entire dataset to produce fuzzy labels so they suffer from the same problem when analysing temporal patterns. For this reason, other methods of discretisation are not considered.

Thresholds for minimum temporal support and minimum confidence were set at 0.025 and 5% respectively for the FuzzyApriori algorithm. The same level of minimum temporal support was also used in CHC. The results of the FuzzyApriori showed that $99.\dot{9}\%$ of rules produced were of length 2, so IRL was limited to only evolve rules of that length. The GAs population size was 50, divergence rate was 0.35, factor for PCBLX-α was 1.0, temporal granularity was 100, CHC was limited to 50,000 fitness evaluations and IRL ran for 10,000 iterations for rules of length 2.

C. Results

Some general results are presented here and then the improvement in temporal fuzzy support and discovery of new rules is discussed.

In Table I, the FuzzyApriori algorithm produced more rules because it is an exhaustive search, whilst the IRL approach was limited to 10,000 rules. The average temporal fuzzy support is lower for CHC so the rules produced have less temporal fuzzy support on average. Yet, the confidence is considerably higher, which is consistent with the results in [13].

TABLE I
RESULTS OF CHC AND FUZZYAPRIORI

Measure	CHC	FuzzyApriori
Number of Rules	10000	90325
Average temporal fuzzy support	0.025	0.031
Average confidence (%)	99.986	24.187
Mode of dataset partitions	100	100

Figure 3 provides more information on the temporal fuzzy support measures of the rules discovered from each method. The minimum temporal support threshold in FuzzyApriori effectively sets the minimum in the boxplot, where as CHC does not have this threshold and so has the ability to produce low temporal support rules. Lowering the minimum temporal support threshold in FuzzyApriori would only produce low support rules because the exhaustive search method has discovered the rules already reported. FuzzyApriori has a more prominent right skew than CHC because of the downward closure property, which is a key part to the Apriori algorithm. The difference in confidence values between the two methods is considerable (Figure 4). Although FuzzyApriori discovers rules with higher temporal fuzzy support further analysis is required to ascertain whether CHC is improving existing rules and discovering new rules.
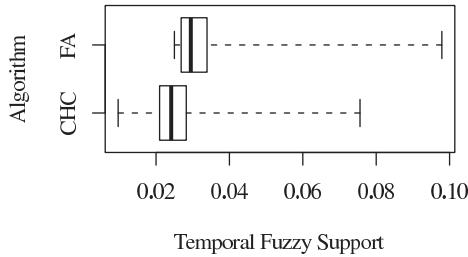

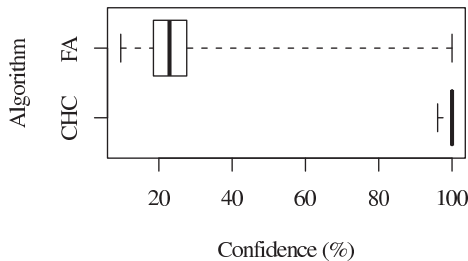Fig. 3. Boxplot of temporal fuzzy support for CHC and FuzzyApriori (FA).

Fig. 4. Boxplot of confidence for CHC and FuzzyApriori (FA).

1) Improved Rules: Rules that were present in the results of both approaches were identified and analysed to show

how many were improved. Rules were determined to be present in both approaches if they had the same temporal period and item/linguistic label, but the lateral displacement was not checked because FuzzyApriori does not use the 2-tuple linguistic representation. Table II shows the percentage of rules that were found in both approaches and whether these rules were an improvement in temporal fuzzy support. There were 21.27% of rules from CHC that were also discovered in FuzzyApriori. The CHC approach improved the temporal fuzzy support for 10.78% rules that were discovered in FuzzyApriori. This demonstrates how the CHC method can improve upon existing temporal fuzzy association rules because the temporal fuzzy support has increased. Although, nearly the same amount (10.49%) where found to degrade the temporal fuzzy support and these rules have no benefit.

TABLE II
RULES FOUND IN CHC AND FUZZYAPRIORI

	Temporal Fuzzy Support		
	Decrease(%)	Increase(%)	Total(%)
CHC and FuzzyApriori	10.49	10.78	21.27
Only CHC	4.26	74.47	78.73

2) New Rules: This analysis shows new rules that were discovered with our proposed approach. From Table II, the remaining 78.73% of rules found with CHC were new rules not discovered with the traditional approach, FuzzyApriori. From all rules, 74.47% had an increase in temporal fuzzy support and this appears to show that these rules discovered with CHC are new. The rules are considered to be new because an exhaustive search (FuzzyApriori approach) did not discover them, although, that is not to say they do not exist. These rules have been discarded by the FuzzyApriori approach because they fall below the minimum temporal support and minimum confidence thresholds. It is the proposed approach of using the 2-tuple linguistic representation that is able to find these rules if they have a strong temporal fuzzy support.

Further analysis in Table III reveals how the minimum thresholds contributed towards removing rules in the FuzzyApriori approach. There were 77.71% rules that were not discovered because they fell below the minimum temporal support threshold. There were also 1.02% rules that were not discovered because they fell below the minimum confidence threshold[1]. Since the minimum temporal support and minimum confidence thresholds are determined by the user as a levels of significance for rules, we are only interested in the rules that have evolved a temporal fuzzy support that is now above the minimum temporal support threshold.

Table IV analyses the rules that were not found with the FuzzyApriori approach and have evolved a temporal fuzzy support above the minimum threshold. The data in Table IV is the same as Table III except that it reports only on those rules that are now above the minimum temporal support. These are considered to be the final result of the mining

[1]Some of these rules have temporal fuzzy support above the threshold.

	Temporal Fuzzy Support		
	Decrease(%)	Increase(%)	Total(%)
Below min. temporal support	3.73	73.98	77.71
Below min. confidence	0.53	0.49	1.02
			78.73

process because they are now significant (i.e., above the minimum temporal support that is a user defined level of significance). A total of 25.38% rules were not discovered with the FuzzyApriori approach because they fell below a minimum threshold: 24.65% below the minimum temporal support and 0.73% below the minimum confidence.

For discovering new rules, the data in Table IV is of most interest because this represents rules that were lost from the traditional approach. Figure 5 shows a boxplot of the improvements in these rules and it can be seen that the central tendency is left skewed suggesting there is generally a large improvement.

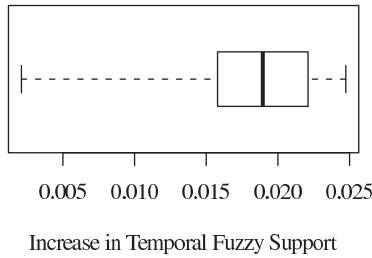|	Temporal Fuzzy Support		
	Decrease(%)	Increase(%)	Total(%)
Below min. temporal support	0	24.65	24.65
Below min. confidence	0.23	0.50	0.73
			25.38

Fig. 5. Boxplot of improvements for rules not discovered in FuzzyApriori and have a final temporal fuzzy support above the minimum threshold.

Figure 6 shows a comparison of the temporal fuzzy support of the improved rules with all the rules produced from CHC and FuzzyApriori (i.e., compared with the data in Figure 3). The improved rules are a subset of all rules that have been filtered according to the rules that were improved above the minimum temporal support threshold. The central tendency and interquartile range is higher for the improved rules than first appeared when initially analysing the rules in Figure 3. These improved rules represent rules that were lost with FuzzyApriori and the improved rules are generally higher than all the rules discovered with CHC. This means that although many rules can be discovered with CHC, the quality of the

rules is high for rules that are genuine improvements. Figure 6 provides an accurate representation of the improved temporal fuzzy support from our proposed approach because it focuses on the rules were originally below the minimum temporal support.
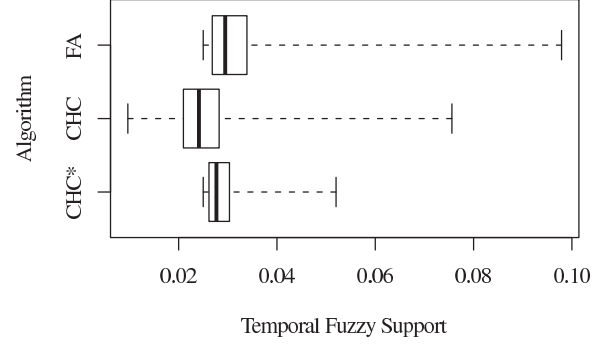

Fig. 6. Boxplot of temporal fuzzy support for CHC* (only improvements below minimum temporal support), CHC and FuzzyApriori (FA).

3) Rules near Intersections: Temporal patterns in fuzzy association rules can be lost because the patterns occur close to the intersection of membership functions. An example of a temporal fuzzy association rule discovered from the CHC method in the 24.65% of rules found to be initially below the minimum temporal support in Table IV is shown below.

Endpoints: 9300–9400
Rule: IF *quantity of Item38* is (*medium*, -0.422)
      THEN *quantity of Item12* is (*medium*, 0.315)

This rule demonstrates high α values because the temporal patterns occur near to the intersection of triangular membership function. Figure 7 visually shows where the membership function are located. It can be seen that both membership function lie near to the intersection and this rule was lost because of a low minimum temporal support threshold.

V. CONCLUSION

A novel approach for revealing hidden temporal patterns of fuzzy association rules is presented. A new application of the 2-tuple linguistic representation is used in a temporal context to maintain interpretability and tune the membership functions with a GA. The problem requires the simultaneous discovery of both rules and membership functions because the search space is complex due to the temporal aspect.

From analysing the rules discovered from the GA and comparing with rules produced from an exhaustive search of rules and dataset partitions (traditional approach), it has been demonstrated that new information can be learnt from two perspectives.

- Improving existing rules discovered with a traditional approach.
- Discovering new rules that would otherwise be lost under the minimum temporal support threshold in a traditional approach
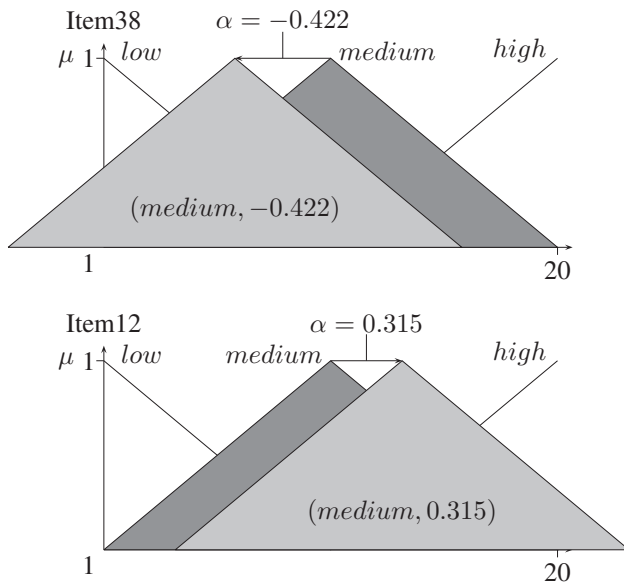
Fig. 7. Fuzzy sets for example rule (IF *Item38* is (*medium*, -0.422) THEN *Item12* is (*medium*, 0.315)).

Future work will focus on a real-world application and investigate the proposed method in more detail.

ACKNOWLEDGMENT

REFERENCES

[1] G. Piatetsky-Shapiro, "Knowledge discovery in real databases: A report on the IJCAI-89 workshop," *AI Magazine*, vol. 11, no. 4, pp. 68–70, 1990.

[2] A. A. Freitas, *Data mining and knowledge discovery with evolutionary algorithms*. Springer-Verlag, 2002.

[3] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of ACM SIGMOD*, Washington, DC, USA, 1993, pp. 207–216.

[4] L. A. Zadeh, "Fuzzy sets," *Information Control*, vol. 8, pp. 338–353, 1965.

[5] K. C. C. Chan and W.-H. Au, "Mining fuzzy association rules," in *Proceedings of the Sixth International Conference on Information and Knowledge Management*, 1997, pp. 209–215.

[6] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning. Parts I, II, III," *Information Sciences*, vol. 8–9, pp. 199–249, 301–357, 43–80, 1975.

[7] T.-P. Hong, C.-H. Chen, Y.-C. Lee, and Y.-L. Wu, "Genetic-fuzzy data mining with divide-and-conquer strategy," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 252–265, 2008.

[8] R. Alcalá, J. Alcalá-Fdez, and F. Herrera, "A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 4, pp. 616–635, 2007.

[9] J. Casillas, O. Cordón, M. del Jesus, and F. Herrera, "Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 1, pp. 13–29, 2005.

[10] G. Venturini, "SIA: A supervised inductive algorithm with genetic search for learning attributes based concepts," in *Proceedings of the ECML*, vol. 667. Springer Berlin / Heidelberg, 1993, pp. 280–296.

[11] S. G. Matthews, M. A. Gongora, and A. A. Hopgood, "Evolving temporal fuzzy itemsets from quantitative data with a multi-objective evolutionary algorithm," in *The IEEE 5th International Workshop on GEFS*. Paris: IEEE, 2011, pp. 9–16.

[12] ——, "Evolving temporal fuzzy association rules from quantitative data with a multi-objective evolutionary algorithm," in *Hybrid Artificial Intelligent Systems (Proceedings of HAIS 2011)*, ser. Lecture Notes in Computer Science. Springer Berlin, 2011, vol. 6678, pp. 198–205.

[13] J. Alcalá-Fdez, R. Alcalá, M. J. Gacto, and F. Herrera, "Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms," *Fuzzy Sets and Systems*, vol. 160, no. 7, pp. 905–921, 2009.

[14] F. Herrera and L. Martinez, "A 2-tuple fuzzy linguistic representation model for computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 6, pp. 746–752, 2000.

[15] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago, Chile, 1994, pp. 487–499.

[16] R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," in *Proceedings of the 1996 ACM SIGMOD*, Montreal, Quebec, Canada, 1996, pp. 1–12.

[17] J. Mata, J. L. Alvarez, and J. C. Riquelme, "An evolutionary algorithm to discover numeric association rules," in *Proceedings of the 2002 ACM Symp. on Applied Comp.* New York, USA: ACM, 2002, pp. 590–594.

[18] H. Ishibuchi, I. Kuwajima, and Y. Nojima, "Relation between pareto-optimal fuzzy rules and pareto-optimal fuzzy rule sets," in *IEEE Symposium on Computational Intelligence in MCDM*, 2007, pp. 42–49.

[19] F. Herrera, "Genetic fuzzy systems: taxonomy, current research trends and prospects," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 27–46, 2008.

[20] M. Kaya and R. Alhajj, "Facilitating fuzzy association rules mining by using multi-objective genetic algorithms for automated clustering," in *Proceedings of the Third IEEE ICDM*, 2003, pp. 561–564.

[21] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*, ser. Advances in Fuzzy Systems - Applications and Theory. World Scientific, 2001.

[22] J. H. Holland, *Adaptation in natural and artificial systems*. MIT Press, 1992.

[23] A. Homaifar and E. McCormick, "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 2, pp. 129–139, 1995.

[24] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, *Interpretability issues in fuzzy modeling*, ser. Studies in fuzziness and soft computing. Springer, 2003.

[25] M. J. del Jesus, J. A. Gámez, P. González, and J. M. Puerta, "On the discovery of association rules by means of evolutionary algorithms," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 5, pp. 397–415, 2011.

[26] T. Mitsa, *Temporal Data Mining*. CRC Press Online, 2010.

[27] J. M. Ale and G. H. Rossi, "An approach to discovering temporal association rules," in *Proceedings of the 2000 ACM Symposium on Applied computing (SAC '00)*, Como, Italy, 2000, pp. 294–300.

[28] B. Özden, S. Ramaswamy, and A. Silberschatz, "Cyclic association rules," in *Proceedings of the Fourteenth International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 1998, pp. 412–421.

[29] J. Han, W. Gong, and Y. Yin, "Mining segment-wise periodic patterns in time-related databases," in *Proceedings of the Fourth International Conference on KDDM*, New York, NY, USA, 1998, pp. 214–218.

[30] Y. Li, P. Ning, X. S. Wang, and S. Jajodia, "Discovering calendar-based temporal association rules," *Data & Knowledge Engineering*, vol. 44, no. 2, pp. 193–218, 2003.

[31] L. J. Eshelman, "The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination," in *Foundations of Genetic Algorithms*, vol. 1. Morgan Kaufmann, 1991, pp. 265–283.

[32] 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 & Knowledge-Based Systems*, vol. 9, no. 5, pp. 587–604, 2001.

[33] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina, "Real-coded memetic algorithms with crossover hill-climbing," *Evolutionary Computation*, vol. 12, no. 3, pp. 273–302, 2004.

[34] J. Alcalá-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms to data mining problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318, 2009.