# Evolving temporal fuzzy itemsets from quantitative data with a multi-objective evolutionary algorithm

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



**Published version**

MATTHEWS, Stephen G., GONGORA, Mario A. and HOPGOOD, Adrian A. (2011). Evolving temporal fuzzy itemsets from quantitative data with a multi-objective evolutionary algorithm. In: IEEE 5th International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS), 2011. IEEE Xplore, 9-16.

# Evolving Temporal Fuzzy Itemsets from Quantitative Data with a Multi-Objective Evolutionary Algorithm

Stephen G. Matthews, Mario A. Gongora and Adrian A. Hopgood
Centre for Computational Intelligence, De Montfort University, Leicester, UK
Email: {sgm, mgongora, aah}@dmu.ac.uk

***Abstract*—We present a novel method for mining itemsets that are both quantitative and temporal, for association rule mining, using multi-objective evolutionary search and optimisation. This method successfully identifies temporal itemsets that occur more frequently in areas of a dataset with specific quantitative values represented with fuzzy sets. Current approaches preprocess data which can often lead to a loss of information. The novelty of this research lies in exploring the composition of quantitative and temporal fuzzy itemsets and the approach of using a multi-objective evolutionary algorithm. This preliminary work presents the problem, a novel approach and promising results that will lead to future work. Results show the ability of NSGA-II to evolve target itemsets that have been augmented into synthetic datasets. Itemsets with different levels of support have been augmented to demonstrate this approach with varying difficulties.**

## I. Introduction

Data Mining is the process of obtaining high level knowledge by automatically discovering information from data in the form of rules and patterns. Data mining seeks to discover knowledge that is accurate, comprehensible and interesting [1]. Association rule mining is a well established method of data mining that identifies significant correlations between items in transactional data [2]. An example association rule for the classical market basket problem of a supermarket would look something like "40% of customers who purchase bread and milk also purchase cheese". The novel aspect of this paper is the extension of the classical problem by exploring the composition of two variants of association rule mining which are commonly found in real-world data.

Classical association rule mining assumes the dataset to be static where discovered rules hold across the entire period of the dataset. In many cases this does not reflect real-world data. Often there can be a temporal pattern behind the occurrence of association rules. The scope of these patterns is far reaching, many systems producing time series data have underlying processes/systems that are dynamic. For example, association rules may occur more frequently:

- In the days leading up to a large sports event.
- When an unforeseen event occurs, such as hurricanes (e.g., [3]) and network intrusions.
- During a temporary change of state of a measurable object i.e. scientific experiments.

Discovering and adapting to changes with well-informed information is important in many domains. Within business it is critical for success, whilst for scientific applications it can enhance insight and understanding. Association rules that incorporate temporal information have greater descriptive and inferential power [4], and can offer an additional element of interestingness [1].

Association rule mining uses data with Boolean attributes to represent the occurrence of items. However, many real-world applications provide a richer source of information with quantitative (e.g. height, pressure) or categorical attributes (e.g. type of species, fruit). Quantitative association rule mining [5] discovers rules that express associations between intervals of item attributes. This type of data is a challenge because of the large range of distinct values for each attribute. A common approach is to discretise attributes, with a method such as partitioning (e.g. [5]) or clustering (e.g. [6]), to produce new attributes representing interval values that can then be applied to classical association rule mining methods. However, preprocessing can lead to a loss of information. Evolutionary computing has been used to remove the requirement for prior discertisation, and so relying upon prior knowledge, which can be detrimental to accuracy and overall result. The synergy of evolutionary computing and fuzzy sets has become popular for data mining tasks [7] such as classification and association rule mining.

The composition of temporal association rule mining and quantitative association rule mining provides in-depth and interesting information. In this work the combination of association rule mining problems is treated as a multi-objective optimisation problem to jointly tackle criteria for quantitative and temporal tasks. The aim is to extract temporal itemsets from quantitative data using fuzzy sets, that can later be used to generate association rules. The use of fuzzy sets [8] allows a linguistic interpretation, a smoother transition between boundaries and provides an ability to better handle uncertainty. The itemset space, temporal space and quantitative space are simultaneously searched and optimised. This paper extends our previous work in [9] by including a quantitative element. The temporal itemsets sought are those that occur more frequently over an interval of the dataset, which are seen as an area of greater itemset density. Applications benefiting from this composition of association rule mining tasks are identifying red tide caused by Noctiluca scintillans [6], mining changes in customer behaviour [10] and network behaviour

anomaly detection, to name a few.

This paper is organised as follows. An overview of related work covering association rule mining, including the quantitative and temporal variants, is discussed in Section 2. In Section 3 the multi-objective evolutionary algorithm (MOEA) for mining temporal fuzzy itemsets from quantitative data is presented. An experiment to analyse the efficacy is presented and discussed with results in Section 4 and we conclude our work in Section 5.

## 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 [2] used for data analysis and interpretation. 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 $D = \{d_1, d_2, ..., d_N\}$ where each transaction comprises a subset of items, referred to as an itemset, from $M$ items $I = \{i_1, i_2, ..., i_M\}$. To extract association rules from datasets the support-confidence framework was introduced with the Apriori algorithm in [11]. Support determines the strength of a relationship by measuring how often the rule occurs in a dataset.

$$s(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \tag{1}$$

Confidence determines how frequently the items in the consequent occur in transactions containing the antecedent, which measures the reliability of the inference.

$$c(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \tag{2}$$

These measures have minimum thresholds which are used by a deterministic method to extract rules from the dataset.

### A. Quantitative Association Rule Mining

Classical quantitative association rule mining uses statistical methods such as equal depth (support) discretisation [5] and clustering [12]. For example, age is numeric, it could be partitioned into new items such as *young*, *middle-aged* and *old*. A disadvantage of these classical methods is they require preprocessing of the data which can lose information [13]. Rule generation is then limited to the crisp boundaries of the discretised values, potentially allowing for other rules to be hidden. Over the years soft computing techniques have been used to overcome this issue by optimising the intervals for quantitative data and inducing rules. In [13] a genetic algorithm was used to evolve attribute intervals with each attribute directly represented in the chromosome, which differs from the variable number of attributes used in this approach. 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.

Fuzzy association rules [14] deal with the inaccuracies in physical measurements and better handle unnatural boundaries found in crisp partitions. They provide a linguistic interpretation of numerical values, which is of importance when interfacing with experts. Evolving fuzzy association rules [15] enhances the interpretability of quantitative association rules.

There are two common approaches to mining quantitative association rules. One approach is to tune the membership functions and then use a deterministic method to induce rules afterwards, as seen in [16]. This typically aims to tune the membership functions to produce maximum support for 1-itemsets before exhaustively mining the rules. Another approach is to extract the association rules as well as define the attribute intervals [13] or membership functions [15]. In [17], an alternative approach to tuning the fuzzy sets includes combining clustering with a MOEA. Although the association rules are identified as well as fuzzy sets tuned, all the dataset attributes are directly represented in the chromosome.

### B. Temporal Association Rule Mining

A key issue of classical methods, based on the support-confidence framework, is that temporal patterns with low support values can escape below the minimum support threshold. For example, consider a product item in a supermarket, it may be available for sale only during a particular seasonal period, such as British asparagus during summer. Its support since it was introduced is high but its support across the entire dataset is low. This rule may not be discovered with classical association rule mining algorithms if its support across the entire dataset drops below the threshold. Assuming that the minimum support is sufficiently low for the asparagus rule in summer to be discovered, further analysis is then required to ascertain any temporal pattern. The *lifespan* property was introduced in [18] as an extension on the Apriori algorithm to incorporate temporal information. This is a measure of support that is relative to the lifespan of the itemset defined by a time interval, known as temporal support, corresponding to the first and last occurrences of the itemset. But this does not consider datasets where the frequency of rules may be skewed towards particular areas whilst still occurring throughout the entire dataset.

A step towards analysing areas of a dataset where rules occur more frequently is found in cyclic association rule mining [19]. Cyclic rules are induced from user-defined partitions of regular periods throughout a dataset. Support values of association rules in user-defined 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 [20] relax the regularity found in fully periodic so the cyclic behaviour is found in only some segments of the dataset and is not always repeated regularly. Defining the temporal intervals with calendar-based schemas is less restrictive and reduces the requirement of prior knowledge [21]. These works illustrate the types of temporal patterns that can be potentially by extracted with our proposed method.

Our previous work [9] has demonstrated the efficacy of mining association rules that occur more frequently over single areas of a Boolean dataset. Iterative Rule Learning evolved temporal itemsets based on the temporal support metric used in [18] by simultaneously searching the itemset space and temporal space. This paper extends our previous work by including an additional search space and employing a more capable evolutionary computing method.

## III. Multi-Objective Evolutionary Search and Optimisation

The aim of this evolutionary method is to extract a set of fuzzy itemsets, leading to fuzzy association rules, from areas of the dataset where they occur more frequently. This is treated as a multi-objective problem, which is defined as the optimisation (minimisation/maximisation) of two or more functions, whilst satisfying optional constraints [22]. A MOEA finds optimal solutions which are compromises between objectives, these solutions are said to have trade-offs. These trade-offs are often managed with the concept of Pareto Optimality. A solution is said to be Pareto optimal when no change in the solution will improve one objective without degrading another objective.

For association rule mining a Pareto based MOEA is capable of producing multiple rules from a single run because a set of maximally-spread Pareto-optimal solutions is maintained with crowding distance. This is desirable when the cardinality of the optimal set may be more than one, for instance in the case of multiple temporal patterns. This is an improvement of our previous work [9] which required numerous runs of the algorithm with Iterative Rule Learning to idenitfy multiple temporal patterns. This is a challenging task that involves simultaneously searching the itemset space, the temporal space and the quantitative space, which together form a multi-dimensional search space. Previous MOEAs for association rule mining have focused on Boolean data [23] and quantitative datasets [15], but not the composition of temporal and quantitative, which is a significant step in problem dimensionality.

From the plethora of MOEAs, NSGA-II [24] is selected for its popularity, computational speed and ability to maintain a diverse set of solutions, which is suitable for extracting multiple patterns. NSGA-II is based on a non-dominated sorting approach and uses elitism. Previous works on association rule mining have used NSGA-II for Subgroup Discovery [25], a closely related area, and motif sequence discovery [26], a different form of temporal mining.

### A. Representation

The Michigan approach of representing a single solution with a chromosome is adopted. Normally an association rule is encoded in the representation so the direct output of the algorithm is a rule. In this research only the itemset is encoded, without distinguishing antecedent from consequent, but containing the same items as would be found in a rule. Only the support of the association rules/itemsets is used to identify temporal patterns. The confidence is not calculated to generate association rules from itemsets because the support is considerably more influential in identifying temporal patterns, as seen in [18] and [19]. The generation of association rules occurs after the algorithm has executed, similar to that of [13]. The Apriori algorithm uses the downward closure property to generate candidate itemsets, but the method presented here uses evolutionary computing to generate itemsets. For this reason only $k$ length association rules are produced from itemsets.

A mixed coding scheme is used to represent the temporal interval and fuzzy itemsets as

$$C = (t_0, t_1, i_0, a_0, b_0, c_0, \ldots, i_k, a_k, b_k, c_k) \tag{3}$$

where the temporal interval is defined with $t_0$ and $t_1$ as integers. The items are integers denoted with $i$ and the basic parameters of the triangular membership functions are real numbers indicated with $a$, $b$ and $c$ for itemsets with $k$ distinct items. The number of items, $k$, is limited to 4 for this study. The membership function parameters are limited to a granularity of 0.05, as seen in [16].

### B. Objectives

The fitness objectives are designed to search the multi-dimensional space based on temporal support, fuzzy itemset support and membership function widths. The following objectives are minimised to zero.

*1) Temporal Support:* The temporal support objective guides the MOEA to find itemsets that occur more frequently in areas of the dataset. Modified from [18] and used in our previous work [9], this is redefined as a minimisation function

$$ts(X, l_X) \quad = \quad 1 - \frac{\sigma(X)}{l_X} \tag{4}$$

where $l$ is a time interval i.e. $l_X = t_1 - t_0$ where $t_0$ is the lower endpoint and $t_1$ is the upper endpoint and $\sigma(X)$ denotes support of itemset $X$. A minimum temporal support is used to prevent solutions evolving towards the smallest time interval of length 1.

*2) Fuzzy Itemset Support:* This objective optimises the membership function parameters of matching itemsets. The quantitative values are modelled with fuzzy sets and the objective's optimal solution is one where the fuzzy sets support the quantitative values associated with the itemset to the highest degree of membership. Fuzzy itemset support, $fis$, is the sum of the degrees of memberships, $sum(\mu(x^{(i)}))$, for a chromosome itemset, $x^{(i)}$, in the $i$th transaction.

$$fis = (k \cdot (t_1 - t_0)) - \sum_{i=t_0}^{t_1} sum(\mu(x^{(i)})) \tag{5}$$

$$sum(\mu(x^{(i)})) = \sum_{j=0}^{k} \begin{cases} \mu(x_j^{(i)}), & \text{dataset item matches gene item} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$\mu(x_j^{(i)}) = \begin{cases} \frac{x_j^{(i)}-a}{b-a}, & \text{if } a \leq x_j^{(i)} < b \\ \frac{c-x_j^{(i)}}{c-b}, & \text{if } b \leq x_j^{(i)} \leq c \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Equation 5 subtracts the sum of the actual degrees of memberships from the maximum possible sum if all items in a transaction match those in the chromosome. Equation 6 performs the summation of actual degrees of memberships for chromosome items matching dataset transaction items. Equation 7 is the triangular memerbship function.

*3) Membership Function Widths:* The aim of this objective is to prevent the membership function parameters evolving to cover the entire range of values i.e. the feet of the membership function ($a$ and $c$) nearing the limits of the attribute values. Without this objective solutions evolve to cover the entire range of attribute values because this yields higher support values as it includes more items.

$$mf\_widths = \begin{cases} \sum_{j=0}^{k} c_j - a_j, & \text{if } c_j - a_j > 0 \\ nitems, & \text{otherwise} \end{cases} \quad (8)$$

### C. Initialisation

The initial population is randomly generated using the Mersenne Twister pseudorandom number generator. The itemset is randomly generated without item duplication. The number of items in the dataset (e.g. inventory) must be greater than the itemset size otherwise this will result in chromosomes where the only difference is the ordering of items. The membership function parameters are randomly generated within the limits of the dataset's attribute range and reordered, in ascending order, if required.

The lower and upper endpoints are generated within proximity to the first and last transactions. The endpoint range defines the range for creating and also mutating endpoints. Defining initial time endpoints near the boundaries of the dataset initialises the algorithm with solutions having large temporal coverage of the dataset. Without the endpoint range random sampling of time intervals occurs. This may lead to some potentially strong itemsets being lost, for example, British asparagus being assigned a time interval over the winter months would produce 0% temporal support, assuming the item is not present. This initial large temporal coverage, combined with the mutation operator mentioned below, provides more opportunity for solutions with great potential that initially may be weak.

### D. Genetic Operators

The crossover operator is based on uniform crossover and consists of three methods for operating on different sections of the chromosome which have different constraints. The section of a chromosome containing the lower and upper endpoints are crossed over with uniform crossover and the feasibility of offspring is ensured by satisfying the minimum temporal support constraint on endpoints $t_0$ and $t_1$, shown here as

$$t_1 - t_0 >= min\_temp\_sup \quad (9)$$

For the itemsets found in the next section of a chromosome, uniform crossover is adapted to ensure that only feasible solutions are produced, i.e. combinations of integers without duplicates. The method for crossing over only the itemsets is taken from [9] and is presented in Figure 1. The advantage of this method is that the ordering of items remains unless a duplicate is present in the itemset. A summary of each stage of the crossover is briefly described here.

**Stage 1**
Merge the chromosomes from two selected parents into an intermediate array so that no two items from the same parent are adjacent.

**Stage 2**
Check each item in the array for duplicate values against the remaining items. If a duplicate is found the duplicate item is swapped with the next item. The result is that all duplicate items are now adjacent and the items can now be selected from the intermediate array to form offspring.

**Stage 3**
Select items from the intermediate array by iterating over every even index value. A random integer (0 or 1) is added to the index and the indexed item is added to the offspring. If a 0 is generated, it is checked for duplicates with the preceding item and if a duplicate is found it adds 1 to the index otherwise it adds 0.

The cross over of membership functions depends on whether the parents have the same itemset. If two parents have the same

Fig. 1. Algorithm for performing crossover on itemsets

**Require:** $Parent1.length \equiv Parent2.length$
{Stage 1}
**for** $i = 0$ to $Parent1.length - 1$ **do**
  $Auxiliary[2i] = Parent1[i]$
  $Auxiliary[2i + 1] = Parent2[i]$
**end for**
{Stage 2}
**for** $i = 0$ to $Auxiliary.length - 1$ **do**
  **for** $j = i + 2$ to $Auxiliary.length - 1$ **do**
    **if** $Auxiliary[i] \equiv Auxiliary[j]$ **then**
      exchange $Auxiliary[j]$ with $Auxiliary[i + 1]$
    **end if**
  **end for**
**end for**
{Stage 3}
**for** $i = 0$ to $Parent1.length - 1$ **do**
  **if** $i > 1$ and $Auxiliary[2i - 1] \equiv Auxiliary[2i]$ **then**
    $Child[i] = Auxiliary[2i + 1]$
  **else**
    $Child[i] = Auxiliary[2i+$RANDOM(0,1)$]$
  **end if**
**end for**

itemset then uniform crossover is applied to those membership function parameters. Otherwise, the membership function parameters are copied across. This prevents crossing over membership parameters from different items, which would be more exploratory (i.e. mutation) than exploitative. The membership function parameters are reordered if they are out of sequence as a result of uniform crossover.

To produce a mutated individual, a randomly chosen gene is replaced with a randomly created value. Mutated genes forming the itemset part of chromosomes are randomly generated with an identical process to that used during initialisation. For genes forming the time interval endpoints, the values are generated within the endpoint range ($epr$) where the midpoint is the value of the current gene ($g$), such that the mutated value is a member of the set $\{-epr/2, \ldots, g, \ldots, epr/2\}$. This reduces the effect of randomly sampling the dataset. The endpoint range is decremented every generation until reaching 10, to allow further mutations. Reducing the range of mutation reduces the magnitude of mutation with the aim of fine tuning solutions towards the end of evolution.

## IV. Experimental Study

### A. Methodology

The IBM Quest Synthetic Data Generator [11] [1] has been used to generate a dataset for experimentation. The generator produces datasets that replicate transactions. A synthetic dataset is chosen rather than a real-world dataset so that a controlled experiment can be conducted to validate this approach before progressing to real-world datasets. The data generator has been extended to include quantitative values for item attributes. A similar method to [16] is used to randomly generate quantitative values. Individual temporal itemsets that exhibit relatively low, medium and high support are identified and used as target solutions.

A dataset has been produced with the following features: 1000 transactions, 50 items, an average transaction size of 10 and a maximum size for quantitative values of 20. There is no guarantee that the generated dataset contains any temporal patterns so, to include temporal information, the method from [9] was used to augment temporal patterns into datasets with the following process:

1) Run Apriori algorithm on dataset to produce frequent itemsets.
2) Select a frequent itemset with desired level of support.
3) Insert the itemset as a transaction near to the centre of the dataset. Transactions are constructed exclusively from the entire frequent itemset with no additional items so no unexpected correlations between items are introduced.
4) Crop datasets to same number of transactions.

Figure 2 is a histogram of a dataset augmented with a high support itemset that illustrates how the itemset frequency of the dataset has been modified. The peak represents the target itemset that the MOEA seeks to identify, along with corresponding membership functions for quantitative values.

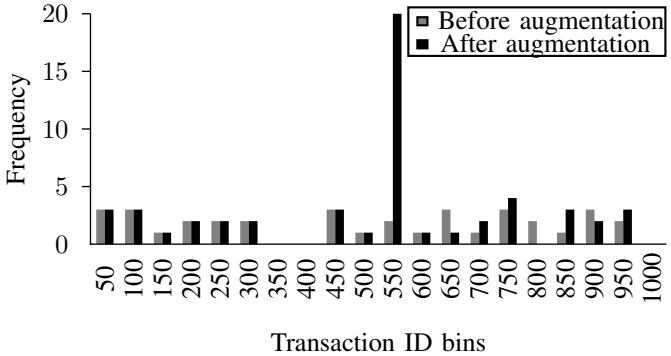



Fig. 2. Histogram of high support itemset {8, 12, 21, 45}

TABLE I
Results of augmenting varying itemset supports

| Itemset | Pre.-Aug. Support | Discovered target in runs |
|---|---|---|
| 24(12), 31(16), 32(10), 38(14) | 0.2% | 7/50 |
| 12(3), 31(11), 41(9), 48(15) | 1.5% | 17/50 |
| 8(9), 12(14), 21(6), 45(19) | 3.5% | 50/50 |

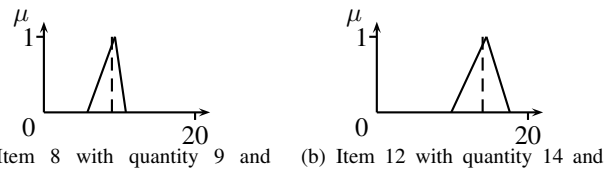
The parameters of the MOEA were determined through trial and error to achieve best results. The parameters were set as a population size of 1000, a crossover probability of 0.5 and a mutation probability of 0.4. The algorithm was limited to 200 generations as test showed that no further improvements were discovered beyond this. Minimum temporal support is set at 30 and the endpoint interval is set at 100.

### B. Results

Various levels of itemset support were selected and augmented with the same dataset to investigate the efficacy of our approach. These *low* (0.2%), *medium* (1.5%) and *high* (3.5%) itemset support datasets are shown in Table I with their corresponding support measures. The algorithm was run 50 times with different random seeds on each dataset. The number of times the algorithm found the augmented itemset within a correct time interval was recorded in Table I. From these results it can be seen that the algorithm successfully evolved the high support itemset for every run of the algorithm, it was not as successful with the medium support itemset and less successful for the low support itemset. A low support itemset occurs infrequently and so also produces low temporal support, hence a it is a weaker individual that struggles to survive through the evolution process. Despite varying levels of performance with each dataset, the target temporal pattern has been discovered for each.

Table II shows some of the chromosomes from a population of a run. A 0.5% threshold on the temporal support was taken, where all results matched the target itemset and corresponding time endpoints. These chromosomes, and those in Figure 7, demonstrate how the objectives conflict for fuzzy itemset

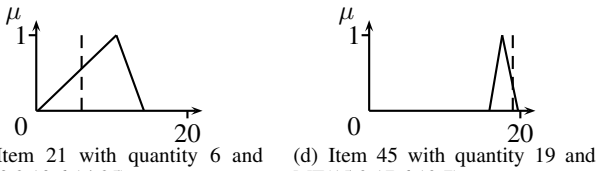
[1]Original source no longer available, instead see http://www.cs.nmsu.edu/~cgiannel/assoc_gen.html, Last accessed 23rd October 2010.




(a) Item 8 with quantity 9 and MF(5.75,9.4,10.85)




(b) Item 12 with quantity 14 and MF(9.85,14.5,17.6)




(c) Item 21 with quantity 6 and MF(0.2,10.6,14.25)




(d) Item 45 with quantity 19 and MF(15.9,17.6,19.7)

Fig. 3. Example derived membership functions (solid lines) for target quantities (dashed lines) of items with fuzzy itemset support fitness 163.18 and function width fitness of 7.68

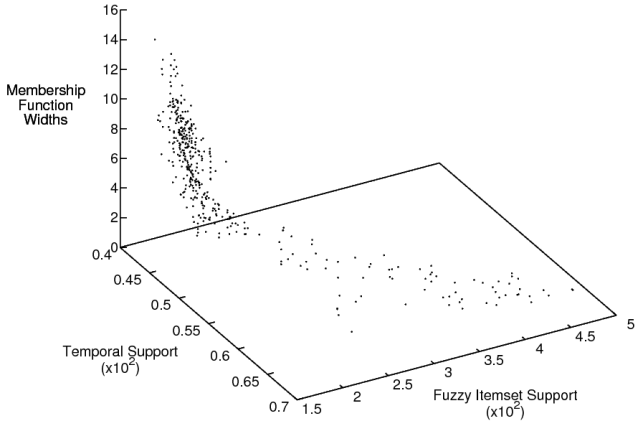



Fig. 4. All objectives for best solutions in a final population

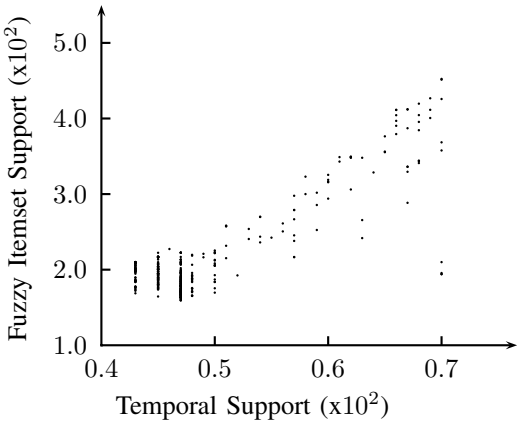



Fig. 5. Objectives 1 and 2 for best solutions in a final population

support and membership function widths. As the fuzzy itemset support grows it tends to widen the shape of the membership function to gain more coverage of quantitative values.

The quality of the derived triangular membership functions can be seen in Figure 3, which shows a dashed vertical line representing the target item against the representative membership function. The membership functions were chosen from a final population of a run with the high support itemset. Most peaks of the triangular membership functions are close to the value of the target quantitative values, with the exception of item 21 with quantity 6 (Figure 3(c)). Here the membership function is much wider and its central point (parameter b) is not close to the quantity. From inspecting more examples of the solutions in Table II, it can be seen that this individual repeatedly evolved incorrect membership function parameters.

Figures 5, 6 and 7 show each objective plotted against one other for a proportion of a final population's best solutions from a high support itemset. The temporal support objective in Figures 5 and 6 clearly show discontinuous regions where many solutions have settled, predominately on vertical lines with temporal support fitness 0.43, 0.45 and 0.47. Each region contains solutions with the same itemset and the same time interval, the vertical height of the region comes from the variation in one of the other objectives. The presence and height of each discontinuous region demonstrates NSGA-II's ability to maintain diversity.

The evolved endpoints, as seen from examples in Table II, are shorter than the actual time interval of the itemset augmented into the dataset. This is due to the minimum temporal support being lower than the augmented time interval. This is a difficult parameter to set because in a real-world application we would not know the length of the itemset.

For the purposes of evaluating our approach, a target temporal pattern has been augmented into the dataset and so the desired result is known a priori. For real-world applications where the temporal patterns are genuinely not known then the Pareto front of the final population is used to identify results. All objectives have been plotted in Figure 4. The Pareto front is found on the outer edge of points that near towards the miniumum fitness of objectives. These solutions are a trade-off between the fitness of the objectives, which can be used to obtain the most interesting temporal patterns for a particular real-world application. The selected indivdual for this evaluation is depicted in Figure 3 with time endpoints of 502 and 532, and itemset (membership parameters) of 8 (5.75,9.4,10.85), 12 (9.85,14.5,17.6), 21 (0.2,10.6,14.25), 45 (15.9,17.6,19.7).

## V. Conclusion

In this paper, we have proposed the use of a multi-objective evolutionary algorithm for extracting fuzzy itemsets, leading to association rules, from quantitative data that form temporal patterns. The temporal patterns sought are those that occur more frequently in areas of the dataset. NSGA-II is applied to mining temporal fuzzy itemsets and is shown to maintain diversity within the population. The advantages of the proposed approach is that it does not exhaustively search the various spaces, it requires no discretisation and yields

TABLE II
EXAMPLE CHROMOSOMES FROM A FINAL POPULATION WITH FITNESS OBJECTIVES

| Chromosome | | | Objectives | | |
|---|---|---|---|---|---|
| Start | End | Item (Membership Function Parameters) | 1 | 2 | 3 |
| 502 | 532 | 8 (5.75,9.4,10.85) 12 (9.85,14.5,17.6) 21 (0.2,10.6,14.25) 45 (15.9,17.6,19.7) | 0.47 | 163.18 | 7.68 |
| 502 | 532 | 8 (5.2,9.05,10.65) 12 (2.7,14.5,14.55) 21 (1.75,13.7,19.9) 45 (14.55,19.1,19.7) | 0.47 | 159.25 | 10.15 |
| 502 | 532 | 8 (6.2,9.2,15.65) 12 (0.8,4.5,14.75) 21 (2.45,10.6,16.3) 45 (3.2,9.4,19.45) | 0.47 | 159.8 | 13.38 |
| 502 | 532 | 8 (8.4,13,19.3) 12 (8.55,9.2,14.45) 21 (15.9,18.3,18.45) 45 (16.45,18.05,19.7) | 0.47 | 185 | 5.65 |
| 501 | 532 | 8 (11.8,13.7,14.7) 12 (10.3,19.35,19.95) 21(0.15,0.95,13.9) 45 (7.7,18.35,19.85) | 0.45 | 195.09 | 9.61 |
| 501 | 531 | 8 (2.05,2.15,5.6) 12 (2.8,3.6,7.2) 21 (11.55,15.65,17.15) 45 (2.7,5.65,18.85) | 0.43 | 209.14 | 7.43 |
| 502 | 535 | 8 (8.4,9.6,11.8) 12 (5.95,13.1,16.45) 21 (19.65,19.65,19.75) 45 (17.45,17.95,18.15) | 0.48 | 218.52 | 3.68 |
| 501 | 536 | 8 (12.25,17.95,19.75) 12 (4.7,6.8,9.1) 21 (16.2,18.05,18.4) 45 (7.2,12.25,19.4) | 0.46 | 227.34 | 6.58 |

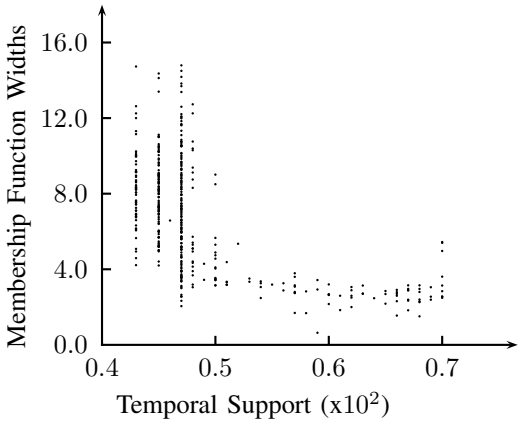



Fig. 6. Objectives 1 and 3 for best solutions in a final population

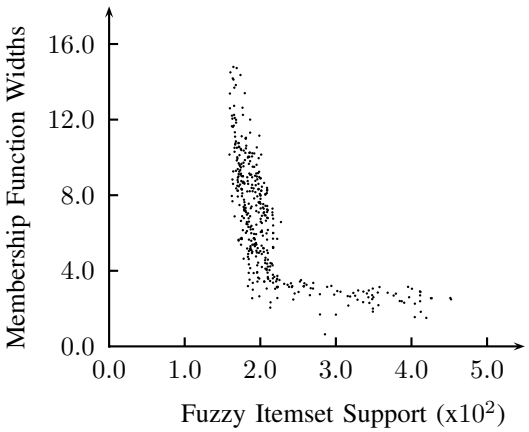



Fig. 7. Objectives 2 and 3 for best solutions in a final population

numerous diverse itemsets, which are potentially different temporal patterns.

Our approach is capable of evolving target solutions that have been augmented into datasets with varying levels of difficulty. The more challenging datasets are those where the itemsets forming the temporal patterns occur very infrequently throughout the dataset. A key aim of future work will be to enhance the algorithm's robustness when using more challenging datasets, perhaps by providing a good starting point in the initial population or a completely different approach. The few experiments conducted for this paper have shown that the derived membership functions do not always correctly identify the target quantitative values. This suggests a second stage of fine tuning the final population may be required.

We have evaluated our methodology with a synthetic dataset augmented with temporal pattens so future plans include using real-world datasets. Depending on these datasets, the scalability of this approach may need to be analysed. The novelty of this paper is in both the problem of extracting temporal patterns from quantitative data and also the use of a MOEA for temporal pattern mining. So, to demonstrate the suitability of our method we will compare against statistical data mining algorithms and methodologies, such as a temporal based Apriori algorithm and discretisation, but also other MOEAs.

## ACKNOWLEDGEMENTS

Supported by an Engineering and Physical Sciences Research Council Doctoral Training Account.

## REFERENCES

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

[2] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD International Conference on Management of Data*, vol. 22, no. 2, pp. 207–216, 1993.

[3] D. Leonard, "After katrina: Crisis management, the only lifeline was the wal-mart," *FORTUNE Magazine*, October 2005.

[4] S. Laxman and P. S. Sastry, "A survey of temporal data mining," *Sādhanā*, vol. 31, no. 2, pp. 173–198, 2006.

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

[6] L. Zhai, X. Tang, L. Li, and W. Jiang, "Temporal association rule mining based on t-apriori algorithm," in *Proceedings of the International Symposium on Spatio-temporal Modeling, Spatial Reasoning, Analysis, Data Mining and Data Fusion (STM '05)*, Peking University, China, 2005.

[7] H. Ishibuchi, "Multiobjective genetic fuzzy systems: Review and future research directions," in *Proceedings of IEEE International Fuzzy Systems Conference (FUZZ-IEEE 2007)*, 2007, pp. 1–6.

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

[9] S. G. Matthews, M. A. Gongora, and A. A. Hopgood, "Evolving temporal association rules with genetic algorithms," in *Research and Development in Intelligent Systems XXVII*, M. Bramer, M. Petridis, and A. Hopgood, Eds. Springer London, 2010, pp. 107–120.

[10] 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.

[11] 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.

[12] R. J. Miller and Y. Yang, "Association rules over interval data," *ACM SIGMOD Record*, vol. 26, no. 2, pp. 452–461, 1997.

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

[14] 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.

[15] M. Kaya, "Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 10, no. 7, pp. 578–586, 2006.

[16] 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.

[17] R. Alhajj and M. Kaya, "Multi-objective genetic algorithms based automated clustering for fuzzy association rules mining," *Journal of Intelligent Information Systems*, vol. 31, no. 3, pp. 243–264, 2008.

[18] 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.

[19] 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.

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

[21] 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.

[22] C. A. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. Springer, 2007.

[23] A. Ghosh and B. Nath, "Multi-objective rule mining using genetic algorithms," *Information Sciences*, vol. 163, no. 1–3, pp. 123–133, 2004.

[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[25] C. Carmona, P. Gonzalez, M. del Jesus, and F. Herrera, "NMEEF-SD: Non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 5, pp. 958–970, 2010.

[26] M. Kaya, "Mogamod: Multi-objective genetic algorithm for motif discovery," *Expert Systems with Applications*, vol. 36, no. 2, Part 1, pp. 1039–1047, 2009.