International Journal of Approximate Reasoning 93 (2018) 463–486

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Adaptive fuzzy partitions for evolving association rules in big data stream

Elena Ruiz [a,b,*], Jorge Casillas [a,b]

[a] Department of Computer Science and Artificial Intelligence, University of Granada, E-18071, Granada, Spain
[b] CITIC-UGR (Research Center on Information and Communication Technology), University of Granada, E-18071, Granada, Spain

A R T I C L E I N F O

Article history:
Received 12 February 2017
Received in revised form 8 August 2017
Accepted 29 November 2017
Available online 2 December 2017

Keywords:
Association rules
Data stream mining
Electroencephalography
Genetic fuzzy systems
Incremental learning
Real-time systems

A B S T R A C T

The amount of data being generated in industrial and scientific applications is constantly increasing. These are often generated as a chronologically ordered unlabeled data flow which exceeds usual storage and processing capacities. Association stream mining is an appealing field which models complex environments online by finding relationships among the attributes without presupposing any *a priori* structure. The discovered relationships are continuously adapted to the dynamics of the problem in a pure online way, being able to deal with both categorical and continuous attributes. This paper presents a new advanced version, Fuzzy-CSar-AFP, of an online genetic fuzzy system designed to obtain interesting fuzzy association rules from data streams. It is capable of managing partitions of different granularity for the variables, which allows the algorithm to adapt to the precision requirements of each variable in the rule. It can also work with data streams without needing to know the domains of the attributes as it includes a mechanism which updates them in real-time. Fuzzy-CSar-AFP performance is validated in an original real-world Psychophysiology problem where associations between different electroencephalogram signals in subjects which are put through different stimuli are analyzed.

1. Introduction

We live in the age of data where all our actions are recorded (or they could be) and sometimes they also are stored and processed. The amounts of data being generated and recorded have been increasing significantly over the past few years. Obviously, a huge part of the information generated lacks of relevance or interest. Picking out what information is relevant, synthesizing it and extracting knowledge from this information is, each time, a more critical aspect in the current society. Frequently, the information is simply stored so that it might be consulted later but in other cases it is also used to obtain models (data mining) that simplify the complex reality such information contains. Many times these data are employed in prediction tasks. However, other times interest resides in supervising systems in order to prevent situations, understand certain processes, help in decision-making... Data sources which continuously produce chronologically ordered information that exceeds usual storage and processing capacities are becoming increasingly common.

To address this kind of problem, it is possible to manage data streams, infinitely structured record sequences that arrive continuously [1]. The key characteristic of these systems is that the data produced by these flows are not stored in a permanent way but they are processed on-the-fly. Each datum is analyzed, processed and finally forgotten, making man-

* Corresponding author.
E-mail addresses: eruiz@decsai.ugr.es (E. Ruiz), casillas@decsai.ugr.es (J. Casillas).

https://doi.org/10.1016/j.ijar.2017.11.014

aging huge amounts of data in real-time possible, even with reduced storage and processing capacities. Thus, data stream techniques are also a good option for managing big data problems. Analyzing data as a stream, rather than permanently storing data and immediately processing datum once it is received, adds flexibility and scalability benefits, whether the data is real-time or historical. Regardless, data are born as a stream (e.g. a stream of transactions of a sale system) and it might be easier to stream mine big data than would be to batch it up and then deal with it in batches. Real-world applications are becoming more numerous every day, being increasingly frequent in telecommunications, vehicle traffic, commerce, finances, physiology, robotic or social networks, among others.

The main learning problems studied in data stream mining [1,2] are: (1) classification: the algorithms studied attempt to maximize the percentage of success but also react and adapt to changing situations — *concept drift* [3,4] —, (2) clustering [5] and (3) frequent patterns.

In the last years, most of the specialized literature in this area has been focused on classification (and concept drift [6]); despite of its lack of applicability in real-world cases. This lack of applicability often results in experimentation based exclusively on benchmarks and synthetic data [7–11].

In real world problems with very high arrival rates and immense volumes of data is very rare to find data which are completely labeled and structured. Moreover, when we find one case (e.g., web activity), it seems to make no sense to focus on predictive models which aspire to be, in the best case, hardly as accurate as the entity (human, community, machine...) that has been labeling each data first. It would be more realistic to generate descriptive models with good interpretability which enables system monitoring. Supervised learning for regression tasks makes more sense in real-world problems as the continuous output can be generated by a natural phenomenon or other similar sources [12].

In general, unsupervised learning is more directly applicable to real-world problems, so incremental clustering has also experienced a significant development. Nonetheless, the knowledge which is discovered (segmentation) is often insufficient to be able to help in decision making tasks. Hence, discovering frequent patterns and association rules is considered as a very good way to address many data stream problems where the purpose is to supervise or monitor (not predict) using independent, significant, readable, and simple models. Among these approaches, recently proposed Fuzzy-CSar [13] is a steady-state genetic algorithm designed to discover interesting association rules in data streams. Its learning process is performed to manage huge amounts of data and to adapt its knowledge to concept drifts. Furthermore, thanks to the use of fuzzy logic, it can deal with both categorical and continuous variables.

The aim of this paper is to propose an advanced version of the Fuzzy-CSar algorithm. Furthermore, this new algorithm is applied in a new challenging real-world problem. Most importantly, two new improvements are proposed: (1) new representation and new genetic operators in order to allow the use of fuzzy partitions with different granularities (number of linguistic terms), and (2) a mechanism to update the range of each attribute in an online way (this mechanism makes unnecessary to know a priori the domain of each attribute), which is a demanding feature in real-world data stream environments. The results of this new algorithm, henceforth referred to as Fuzzy-CSar-AFP (Fuzzy-CSar with Adaptive Fuzzy Partitions), are validated in a real-world Psychophysiology problem where associations between different signals from electroencephalogram electrodes are analyzed online while the subjects are subjected to undergo different activities and stimuli.

The remainder of this paper is organized as follows: Section 2 provides a short introduction to the association rule mining (ARM) field, describes the difficulties of learning from online environments, and briefly discusses the behavior and main characteristics of Fuzzy-CSar. Section 3 discusses the proposed characteristics of Fuzzy-CSar-AFP. In Section 4, we describe the real-world data streams that have been used to validate the functionality of Fuzzy-CSar-AFP. In Section 5, we explain the experiments conducted, also indicating the configuration of the algorithm. Section 6 discusses and analyses the results of the experiments conducted. In this section, two different types of analysis are performed: firstly, we validate the Fuzzy-CSar-AFP comparing it in various ways with Fuzzy-CSar and Fuzzy-Apriori [14]. Secondly, we study the relationships discovered between the data attributes with the help of several visualization tools originally proposed in this paper. Finally, Section 7 summarizes and concludes the paper.

2. Association rules in data streams and Fuzzy-CSar

2.1. Association rules in data streams

The need for extracting relevant information from massive amounts of data, usually as a continuous, high speed and time-changing data stream has greatly increased in the industrial and scientific environments [15,1]. Traditional data mining algorithms were not designed for handling these types of data so they are not able to obtain really meaningful models in this type of environments [16]. There are several challenges that must be faced in any learning problem in data streams [17]: (1) obtaining a fast reaction to concept changes, (2) data can only be handled once, (3) storage limitations, and (4) varying noise levels.

In most real-world problems, an unsupervised learning approach is more interesting than a supervised one. More concretely, the discovery of associations in data streams, focused on discovering associations between variables with the use of association rule production in a completely online process, is particularly interesting due to: (1) the demand of interpretability of the patterns discovered in data, (2) the need for discovering patterns while they are happening, and (3) the high and continuous volumes of data to be processed, which demand scalable learners.

Association stream mining is a new area and consequently, to the best of our knowledge, there are no fully operational approaches which fulfill all the requirements mentioned. The most similar approaches for knowledge extraction under these conditions focus mainly on the identification of frequent items (*online frequent pattern mining*), without rule generation [18] or relegating rule generation to a subsequent offline process [1]. This offline phase makes these algorithms mostly unpractical when handling association stream real-world problems, where the rules discovered have to be present immediately and adapt to the changing dynamics of the continuous flow of data. Some few approaches try to palliate this drawback by using sliding window [19,20], so the causality relationship contained in the rule is extracted by analyzing the discovered frequent items in this data window. This approach, however, has the disadvantage of requiring store data (that can be unrealistic) and determining the window size, which is revealed as crucial for a good performance. Furthermore, most of these algorithms can only work exclusively with problems that only contain categorical features [21–24].

In order to deal with continuous features, quantitative association rules are needed. Traditionally, two different strategies were explored to work with quantitative association rules: (1) to discretize the features and then deal with them in a purely qualitative way [25–27], and (2) to use an interval-based representation [28,29]. Later, fuzzy modeling was introduced to improve the legibility of models from domains with imprecision and to avoid rigid and unnatural boundaries produced by interval-based representations [30–34,20]. Fuzzy limits fit better with our normal use of language. Despite all, most of these proposals only focus on identifying frequent variables and not on the final rules extracted in a pure online process.

2.2. Fuzzy-CSar

Fuzzy-CSar [13] is an online genetic fuzzy system designed for mining fuzzy association rules from data which contain both quantitative and qualitative attributes by means of combining genetic algorithms and the apportionment of credit mechanisms online. Fuzzy-CSar actualizes a population of fuzzy association rules in an incremental mode and is able to quickly adapt to the presence of concept changes in the data.

A population of individuals is maintained, where each one is represented by a fuzzy association rule and a group of parameters which evaluate the quality of the rules. The fuzzy association rule consists of an antecedent and a consequent. The antecedent is allowed to have an arbitrary number of attributes while the consequent consists of only one attribute that must not be present in the antecedent of the same rule. Each variable can be represented by a disjunction of linguistic terms (labels) for a greater generalization. The group of quality parameters is formed by eight parameters: (1) the support *supp*, (2) the confidence *conf*, (3) the lift *lif*, (4) the accuracy *acc*, (5) the fitness *F*, (6) the experience *exp*, (7) the numerosity *num*, which represents the number of copies of the individual in the population, and (8) the average size of the action sets in which the individual has participated *as*.

At each iteration of the learning process of Fuzzy-CSar, an example arrives and the algorithm performs a sequence of steps in order to update the parameters of the individuals (association rules) and to discover new relevant rules. To discover these new rules, a niche based, steady-state genetic algorithm is applied. Furthermore, crossover operators, different types of mutation and covering are applied to the rules, with certain probabilities. After checking for subsumption, the new individuals are introduced into the population. In Algorithm 1 we can see a high-level description of the learning phase of Fuzzy-CSar. For a more complete and detailed explanation on the architecture and features of Fuzzy-CSar we refer the reader to [13].

Algorithm 1: A high-level description of the Fuzzy-CSar learning algorithm (from [13]).

```
procedure TrainFuzzy-CSar( trainingSample e_t, Population [P] at time t )
Data: e_t has the form {x_i}_{i=1}^ℓ
Result: Population [P] at time t+1
begin
    e'_t ← granulation(e_t)
    generate [M] out of [P] using e'_t
    if |[M]| < θ_mna then
        generate θ_mna − |[M]| matching individuals using e'_t and updating [P]
    end
    group individuals in [M] by their antecedent forming distinct [A]_i
    select [A] probabilistically
    subsume individuals in [A]
    update individuals in [M] // Therefore, all [A]_i are updated
    if the average time of [A] since last GA > θ_GA then
        perform a genetic event in [A] considering e'_t and updating [P]
    end
end
```

3. Fuzzy-CSar-AFP: adaptive fuzzy partitions for association stream mining

In this paper we propose an advanced version of Fuzzy-CSar, Fuzzy-CSar-AFP, that has been designed in order to provide the system with greater flexibility to enable it to automatically adapt to the singularities of each problem. This adaptation capability allows the algorithm to deal with complex real-world data stream problems.

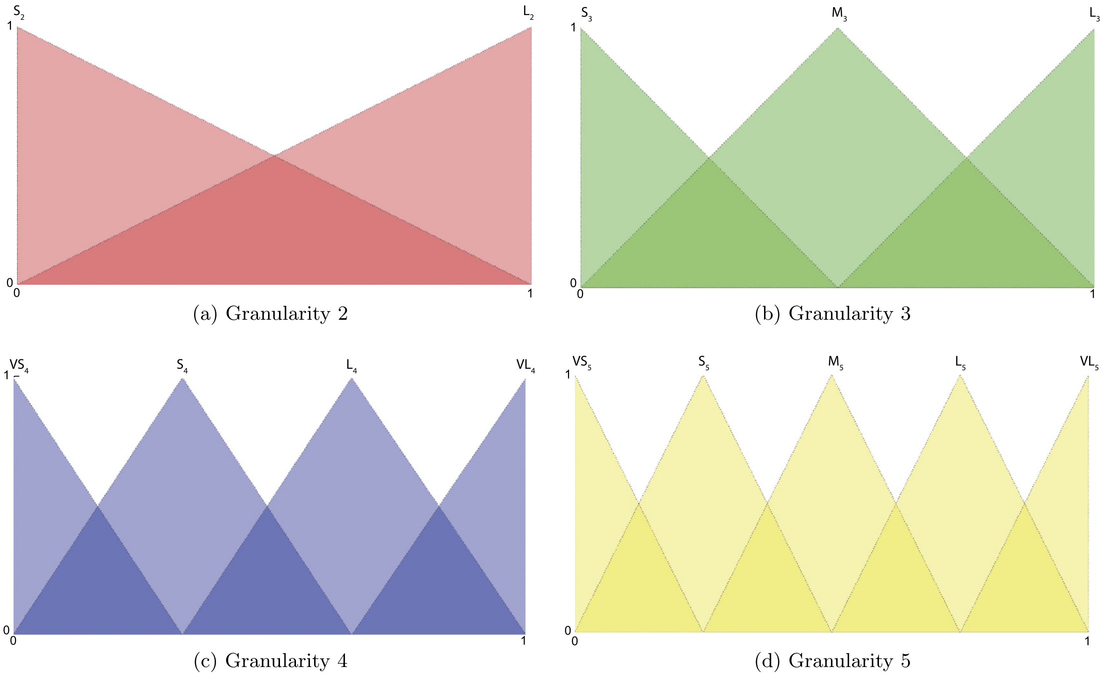


Fig. 1. Illustration of fuzzy partitions with four different granularities (from 2 to 5) where *VS* stands for *very small*, *S* for *small*, *M* for *medium*, *L* for *large*, and *VL* for *very large*.

We propose two main axes for the algorithm's improvement:

- The use of a new representation for the attributes which form the data stream, which allow fuzzy partitions of different granularity and fuzzy set shapes to be managed, thus adapting to the precision requirements of each variable in each rule. The genetic operators are also newly designed for this representation.
- In real world problems where the data are received as continuous flows of information (data stream) it is very likely that the attributes domain will not be known a priori, as this can evolve along the stream or simply not be bound to a static interval. Because of this, in the proposed algorithm we have included a method by which it is not necessary to fix the working range of each variable, the minimum and maximum values for each attribute are updated on-the-fly as the data are being processed.

These improvements, as well as the components of the learning architecture of Fuzzy-CSar whose comprehension is necessary to understand the modifications, are detailed in the following.

3.1. Representation

We propose a new way of encoding the fuzzy association rules in order to allocate patterns with different granularities per attribute. This mechanism allows the algorithm to precisely adapt the generality degree of each fuzzy association rule for an optimum covering of data. We consider fuzzy partitions from 2 to 5 fuzzy sets, though it can be easily adapted to other values. The advantage is twofold, as richer or poorer vocabulary can be used depending on performance demands and, at the same time, different support ranges and cores can be used to better locate the fuzzy set where required. This approach allows the system to also evolve not only symbolic structure of the fuzzy rules but also the fuzzy set parameters, which until now, is a field which has been poorly explored in the data stream, to the best of our knowledge.

In Fig. 1 each of the four possible granularities used in our experiments (from 2 to 5 fuzzy sets) are represented independently, while in Fig. 2 we can observe how the fuzzy partitions of all these four granularities overlap in order to better visualize the different fuzzy set parameters applied by our method.

The idea of maintaining different granularities per attribute was originally proposed for classification tasks in [35] and has been widely applied by the same authors since then. In classification, this approach may suffer from an important lack of interpretability as the expert expects to understand the classification policy by viewing the fuzzy rule set as a whole. However, in unsupervised tasks by association rules, the target is to provide the expert with independent patterns that explain individual association relationships among different attributes, so the fact of using different granularities of the same attribute in different rules is compatible with expert's understanding.

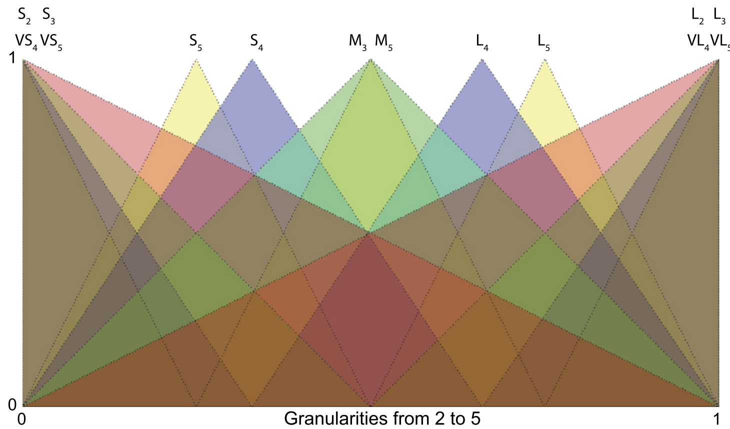


Fig. 2. Illustration of the overlapping with the four different granularities used in this paper. It is possible to appreciate how the fuzzy partitions overlap each other and how some of the cores of the partitions from different granularities match (even the complete partitions do not).

3.2. Support, confidence, and fitness

In the Association Rules Mining field there are several measures for evaluating the quality of rules. We have mentioned that in Fuzzy-CSar and Fuzzy-CSar-AFP each individual from the population has 8 parameters, we are going to explain how both algorithms compute two of them, the two most meaningful for this paper and traditionally most commonly used: Support and Confidence. For information on the computation form of other parameters we refer the reader to [13].

To properly explain the calculation of these measures we are going to formally define Association Rules Mining field as follows [36]: Let $I = i_1, i_2, \ldots, i_l$ be a set of l binary features (items). Let $Tr = tr_1, tr_2, \ldots, tr_N$ be a set of N transactions where each transaction tr_j contains a binary vector indicating in each position if a particular item is present or not. An itemset X ($X \subset I$) is associated with a support which is a measure of its relevance in T and is computed as $sup(X) = |X(T)|/|T|$, where $X(T)$ is the set of variables in the antecedent of the rule. An itemset is considered to be a *frequent itemset* if its support gets over a user-defined threshold (*minsup*). An *association rule* R is an implication of the form $X \to Y$, where both X and Y are *frequent item sets* and $X \cap Y = \emptyset$. The most traditional qualitative measures for association rules are two: their support and their confidence:

$$supp(X \to Y) = \frac{supp(X \cup Y)}{|T|} \tag{1}$$

$$conf(X \to Y) = \frac{supp(X \cup Y)}{supp(X)}. \tag{2}$$

Where support indicates the frequency of occurring patterns and confidence evaluates the strength of the implication denoted in the association rule.

Traditionally, the process of Association Rule Mining consists of two phases: (1) finding all the frequent itemsets in T and (2) extracting rules from these frequent itemsets.

Real-world problems are *flooded* with imprecision and uncertainty that make difficult to traditional machine learning algorithms to find useful patterns from data. The use of fuzzy logic in Association Rules Mining allows the creation of highly legible models from both qualitative and quantitative data. Let $I = \{i_1, i_2, \ldots i_\ell\}$ be a set of ℓ features, and let $A \subset I$, $C \subset I$ and $A \cap C = \emptyset$. A fuzzy association rule is, like before, an implication of the form $X \to Y$ in which:

$$X = \bigwedge_{i_i \in A} \mu_{\widetilde{A}}(i_i) \quad \text{and} \quad Y = \bigwedge_{i_j \in C} \mu_{\widetilde{C}}(i_j), \tag{3}$$

where $\mu_{\widetilde{C}}(i_j)$ is the membership degree of features in the consequent and $\mu_{\widetilde{A}}(i_i)$ is the membership degree of features in the antecedent. In this situation, support is extended by using the product T-norm and confidence is extended by using the *Dienes implication* [31]:

$$supp(X \to Y) = \frac{1}{|T|} \sum \mu_{\widetilde{A}}(X) \cdot \mu_{\widetilde{C}}(Y) \tag{4}$$

$$conf(X \to Y) = \frac{\sum(\mu_{\widetilde{A}}(X) \cdot \max\{1 - \mu_{\widetilde{A}}(X), \mu_{\widetilde{C}}(Y)\})}{\sum \mu_{\widetilde{A}}(X)} \tag{5}$$

where $\mu_{\widetilde{A}}(X)$ is the membership degree of the antecedent part of the rule and $\mu_{\widetilde{C}}(Y)$ is the membership degree of the consequent part of the rule.

3.3. Covering operator

The aim of this operator is to generate new fuzzy association rules when the current pool does not properly match the received sample data e. Indeed, the system creates the *match set* $[M]$ with all the individuals in the population that matches e with a degree greater than 0 in both the antecedent and the consequent of the rule. If $[M]$ contains less than θ_{mna} individuals θ_{mna} is a configuration parameter, the covering operator is applied until there are θ_{mna} individuals in $[M]$.

The covering operator creates a new individual that matches e with the maximum degree: for each variable e_i the operator randomly decides with a certain probability if the variable is in the antecedent of the rule. Then a variable which has not been selected for the antecedent is chosen to be in the consequent.

In the proposed Fuzzy-CSar-AFP, due to the new use of fuzzy partitions with different granularities, when the covering operator generates a new individual for each of the variables in the rule, a specific granularity is randomly chosen where two fuzzy labels are the minimum possible value and the maximum number of fuzzy labels is specified in a configuration parameter. Once the granularity of every variable in the rule has been established, each one of these variables is initialized with the linguistic label that maximizes the matching degree with the corresponding sample value. As in [13], individual's parameters are all set to initial values ($sup = exp = 0$, $conf = lif = acc = num = 1$, $F = 0.01$, and $as =$ the size of the actual action set).

3.4. Association set subsumption

During the learning iteration of Fuzzy-CSar, individuals in $[M]$ are organized into various *association set candidates* $[A]_i$ grouping individuals by their antecedent. Each $[A]_i$ has a probability of being selected that is proportional to the average confidence of the individuals that belong to it. The selected $[A]$ experiments a *subsumption* process. The purpose of this mechanism is to reduce the number of different rules that express similar knowledge.

Each rule in $[A]$ is checked for subsumption with each other rule of the set. A rule r_i is a candidate subsumer of r_j if: (1) r_i has a similar confidence to r_j and it is experienced enough, and (2) r_i is more general than r_j. In Fuzzy-CSar, a rule r_i was considered more general than r_j if all the variables of r_i are also defined in r_j and r_i has, at least, the same linguistic terms than r_j for each one of its variables.

In the proposed Fuzzy-CSar-AFP, the same variable can have fuzzy partitions with different granularity in r_i than in r_j. Thus, r_i will be more general than r_j if all the variables of r_i are also defined in r_j; for each one of its variables the number of fuzzy partitions in r_i is lower or equal than the one in r_j, and between the cores of the linguistic terms of r_i are included the cores of all the linguistic terms of r_j. Each time a rule r_i subsumes a rule r_j, the numerosity of r_i is increased and r_j, the subsumed one, is removed from the population.

3.5. Discovery component

As we have mentioned before, Fuzzy-CSar uses a steady-state, niche-based, incremental genetic algorithm to discover new rules. This genetic algorithm is applied to the selected $[A]$. In the previous version of Fuzzy-CSar, there were three different types of mutation: (1) mutation of the antecedent variables, which randomly decides if a new antecedent variable has to be added to the rule or one of the antecedent variables has to be removed from it; (2) mutation of the consequent variable, which selects one of the variables that appears in the antecedent and exchanges it with the variable of the consequent; and (3) mutation of the linguistic terms of the variables, which selects one of the variables of the rule and mutates its value in one of three possible modes: shift, contraction, or expansion.

In Fuzzy-CSar-AFP we incorporate a fourth type of mutation, mutation of the granularity of the variables, which selects one of the variables of the rule and changes the number of fuzzy partitions. The linguistic terms of the variable and its granularity determine the possible options for the new granularity value and which are going to be the new linguistic terms of the variables. In general terms, mutating the granularity of the variable to the next value up or down is allowed but if the cores of all the linguistic terms of the variables match with values of cores on two granularity levels up or down, mutation to this level will be also allowed. When mutating from n to $n + 1$ partitions, the new label of the variable is expected to be more specific than the original one so it is allowed to move only to a fuzzy set whose support is contained by the support of the original fuzzy set (the closest one, which can be more than one if they are equally distant). If the mutation occurs from n to $n - 1$ partitions, the new label of the variable is expected to be more general than the original one so the operator move to a fuzzy set whose support contains the original one (the closest to it). Taking into account these guidelines, Fuzzy-CSar-AFP considers an adjacency matrix indicating the possible mutations to be made. In Table 1 we can observe the content of the adjacency matrix used in the experiments discussed in this paper (possible granularities from 2 to 5) indicating with 1 which mutations are possible and with 0 which ones are not, considering rows as original linguistic terms and columns as mutated ones. This matrix contains only one exception to the specified guidelines: we do not allow to mutate from M_3 to S_2 or L_2 because we consider M_3 is very different to S_2 and L_2 despite having the same support. Using this matrix, a new granularity is randomly selected to apply mutation to the variable and the original linguistic terms of the variable are transform to their corresponding ones in the new fuzzy partition. In Fig. 3 we represent three mutation situations, two of them ((a), (b)) are permitted mutations and the other one (c) is not.

Table 1
Adjacency matrix used in granularity mutation (rows as original linguistic terms and columns as mutated ones).

	S_2	L_2	S_3	M_3	L_3	VS_4	S_4	L_4	VL_4	VS_5	S_5	M_5	L_5	VL_5
S_2	0	0	1	0	0	0	0	0	0	0	0	0	0	0
L_2	0	0	0	0	1	0	0	0	0	0	0	0	0	0
S_3	1	0	0	0	0	1	0	0	0	0	0	0	0	0
M_3	0	0	0	0	0	0	1	1	0	0	0	1	0	0
L_3	0	1	0	0	0	0	0	0	1	0	0	0	0	0
VS_4	0	0	1	0	0	0	0	0	0	1	0	0	0	0
S_4	0	0	0	1	0	0	0	0	0	0	1	0	0	0
L_4	0	0	0	1	0	0	0	0	0	0	0	0	1	0
VL_4	0	0	0	0	1	0	0	0	0	0	0	0	0	1
VS_5	0	0	0	0	0	1	0	0	0	0	0	0	0	0
S_5	0	0	0	0	0	0	1	0	0	0	0	0	0	0
M_5	0	0	0	1	0	0	0	0	0	0	0	0	0	0
L_5	0	0	0	0	0	0	0	1	0	0	0	0	0	0
VL_5	0	0	0	0	0	0	0	0	1	0	0	0	0	0

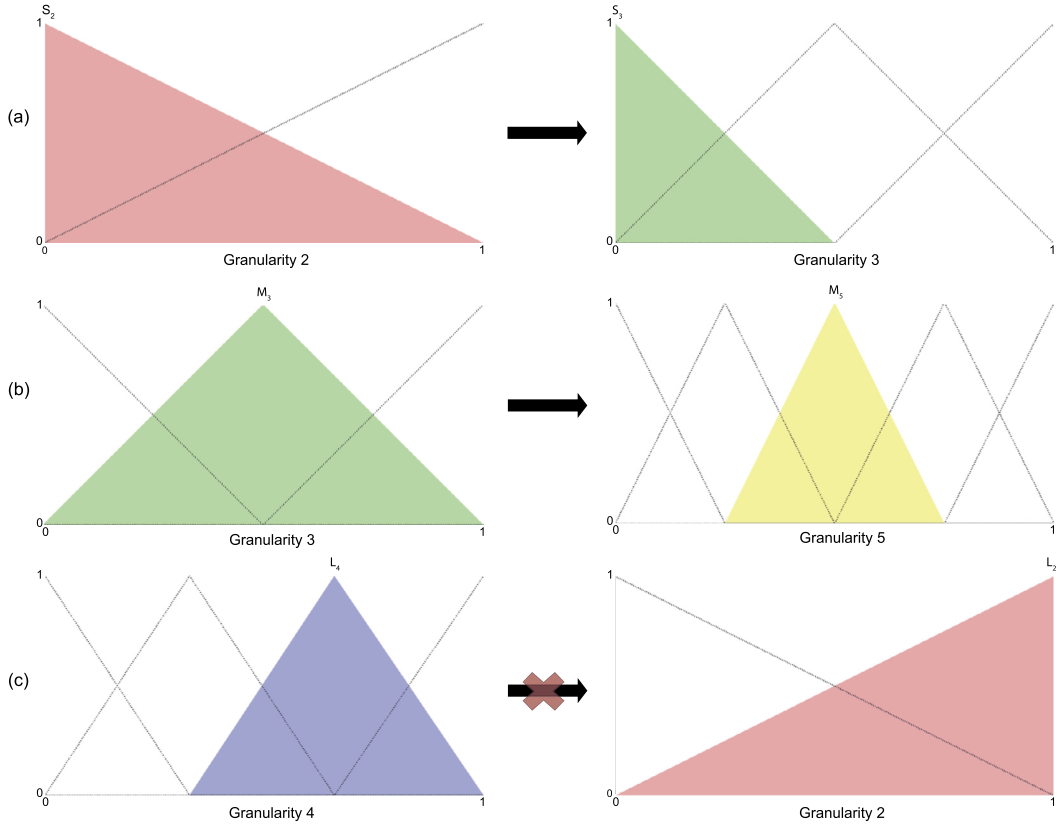


Fig. 3. Three different examples of granularity mutation, two allowed ((*a*) and (*b*)) and one not allowed ((*c*)), following the mutation conditions described in this paper according to starting linguistic terms and granularity.

3.6. Attribute domain update

The aim of this mechanism is to update online the minimum and maximum values of each attribute so it is not necessary to know these values a priori. Each time a new datum is received, the minimum and maximum values of each attribute are recalculated. To do so, a mean (μ) and a standard deviation (σ) incremental computation is considered [37] as starting point. We have redesigned this computation so the latest data values have a higher influence and the domain fits better to the real values evolution. By this mechanism, the minimum and maximum values might be updated every time a sample is received, and thus having the advantage of not needing to store any sliding window from the data stream.

For each attribute, the minimum and maximum values at a certain time stamp are figured as described in Algorithm 2, where β is a parameter to control the percentage of samples covered by the range and α to control the influence of the oldest values compared to the most recent ones. To maintain the online character of Fuzzy-CSar-AFP the algorithm employed to update the minimum and maximum values of the attributes must inspect each input data only once. For such an online algorithm, a recurrence relation is required between quantities from which the required statistics can be calculated in a numerically stable fashion. There are several formulas that can be used to update μ and σ for a new element. However, many of these formulas suffer from numerical instability. In addition to the necessity of update minimum and maximum values inspecting each input value only once, we also want to get that these minimum and maximum values evolve quickly enough to adapt them to the changing nature of the input data stream. To achieve this goal, we modify the original mean and variance incremental algorithm [37] (modifying the computation of *temp* and M_2 in Algorithm 2 including α in them) to increase the influence of the most recent input values, decreasing the influence of the oldest ones. A traditional weighted approach was not the best option since a priori there are not examples with different weights, instead the weight of the examples decreases as they are getting older. For this aim, it is employed a decay factor, α, which penalizes the oldest examples. If $\alpha = 1$ there is no penalization. The lower α is, the higher the relevance of the most recent data until reaching the case of $\alpha = 0$ when μ matches the last value received and σ is always zero. In our experimentation, we use $\beta = 2.5$ so about 98% of the samples are included in the range, and $\alpha = 10^{\log_{10}(0.5)/(sps \cdot s)}$ being $sps = 256$ (samples per second, i.e., Hz) and $s = 40$ (i.e., after 40 seconds the decay factor α will be faded out to 0.5).

Algorithm 2: A description of the incremental algorithm used to calculate minimum and maximum values of a certain variable.

```
procedure Streaming-Mean-Variance( sum at time t, μ at time t, M₂ at time t, n at time t, α, β, x )
Data: n is the number of samples processed at time t
x is the value of a new data for a certain attribute
μ and σ are the stream mean and standard deviation
sum, M₂, α and β are real values
Result: min and max at time t + 1

begin
    temp ← sum · α + 1
    diff ← x − μ
    R ← diff / temp
    μ ← μ + R
    M₂ ← M₂ · α + (sum · α · diff · R)
    sum ← temp
    σ² ← (M₂ · n) / (sum · (n − 1))
    (min, max) ← (μ − β · σ², μ + β · σ²)
end
```

4. Some background on the problem of exploring networks instead of single physiological signals

The physiological adaptation to an ever-changing internal and external environment is the result of a complex interaction between physiological systems who have shown to exhibit non-stationary, intermittent, scale-invariant and nonlinear behaviors. Moreover, physiological dynamics are in constant flux, responding to changes in the underlying control mechanisms caused by different physiological states or pathologic conditions. Here we employ the novel methodology of association stream mining in order to dynamically obtain association rules that explain in an online fashion the relationships between signals derived from the recording of electrophysiological brain activity at distinct electrode sites.

4.1. Method

4.1.1. Participants

The database consists of physiological recordings from 50 young adult participants, divided into two groups of 25 participants per group based on fitness level. Since the consequences of a sedentary lifestyle reach far beyond the development of chronic diseases as they also directly influence brain plasticity and function [38]. Numerous studies have repeatedly shown that exercise improves learning and memory, counteracts the mental decline that comes with age, and facilitates functional recovery from brain injury, disease, and depression [39]. The average age of the high-fit (trained – *TRA*) group was 22 years (age range: 21–24 years old) and of the low-fit (sedentary – *SED*) group 23 years (age range: 22–24 years old). The two subjects compared here were randomly selected as representative from their corresponding groups.

4.1.2. EEG recordings

Continuous EEG data were recorded using a BioSemi Active Two system (Biosemi, Amsterdam, Netherlands) and were digitized at a sample rate of 1024 Hz with 24-bit A/D conversion and subsequently resampled at 256 Hz. The 64 active scalp Ag/AgCl electrodes were arranged according to the international standard 1020 system for electrode placement using a nylon head cap.

4.1.3. Behavioral task

The behavioral task was designed to measure vigilance by recording participants' reaction times (RT) to visual stimuli in a computer screen. Participants were instructed to respond as fast as they could once they had detected the presentation of the stimuli. They had to respond with their dominant hand by pressing the space bar on the keyboard. The task comprised a single block of 60 minutes of total duration.

4.2. What makes this problem challenging in data stream mining

The explained problem on EEG analysis has some properties that makes it an ideal benchmark for association stream mining by Fuzzy-CSar-AFP: (1) the variables (electrode signals) are continuous (which justifies the use of fuzzy logic) and the variation interval unknown (so the proposed attribute domain update mechanism makes sense here); (2) the problem does not have any dependent or output variable, so it needs to be addressed by unsupervised learning; (3) the rate of incoming data is very high (256 per second), which justifies the need of processing data on-the-fly instead of storing them or using sliding window; and (4) there is interest from experts regarding the analysis of relationships among variables as a complementary study to their conventional time series approach.

5. Experimental methodology

5.1. Addressing the difficulty of evaluating association stream

Unlike other types of problems such as supervised learning problems where there are standard measures and mechanisms to evaluate the goodness of the results of algorithms and to establish comparisons between them; in the association rules field there is no standard way to fairly establish comparisons between algorithms. Association rules discovery is an unsupervised learning problem so we do not know what the perfect association rules might be. There are several parameters for individual association rule evaluation like support, confidence, lift, etc., but even with them it is hard to reliably compare different rules sets resulting from two different algorithms or experiments. These issues increase when we talk about association streams, we have to deal with the additional difficulty of evaluating the ability of the algorithm to adapt to concept drifts. Again unlike in supervised and clustering stream fields, for association stream mining, there is no formal way to quantitatively evaluate what happens with the learned model when a concept drift occurs. To evaluate and better understand the results we have designed a set of original graphs and visualizations which would help interpret the results in two different ways: (1) comparison between the performance of Fuzzy-CSar-AFP, Fuzzy-CSar and Fuzzy-Apriori; and (2) visual tools to represent the associations discovered by Fuzzy-CSar-AFP in the data. This set of graphics and visualizations are detailed in the following.

5.1.1. Attribute domain evolution

For each attribute we represent a plot with three functions: (1) the evolution of the maintained minimum value of the attribute (represented in red), (2) the evolution of the real values of this attribute in all the data that have been processed (represented in blue), and (3) the evolution of the maintained maximum value of the attribute (represented in red). With these plots we get a clearer picture of the data stream received and check how well the domains maintained by the algorithm fit to the real data. This type of plot is shown in Section 6.1.

5.1.2. Number of rules: minimum confidence vs. minimum support

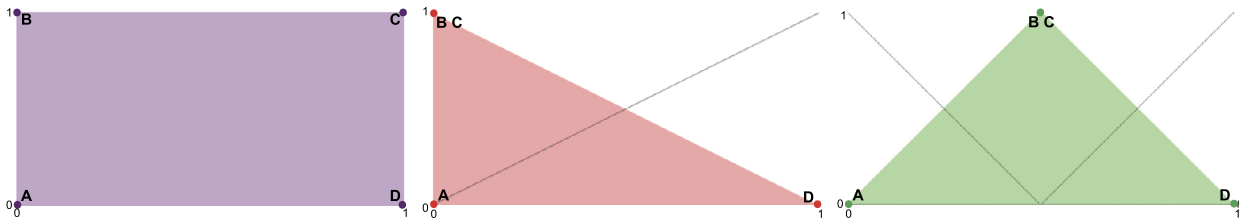
Given a minimum confidence value c, the amount of fuzzy association rules with support equal to or higher than the one corresponding to abscissa value is counted. Several curves corresponding to several confidence thresholds and algorithms are plotted. These graphs are traditionally used to analyze the results of association rules algorithms as they represent the quality of the rules obtained. The correspondent figures are shown in further sections.

5.1.3. Evolution of the amount of good rules

It is an analysis of the amount of fuzzy association rules from each algorithm which pass certain minimum support and confidence values at each moment of the experiment along with the total amount of rules (unfiltered rules). Several curves are plotted for the compared algorithms. This graph complements the one explained in the previous section by showing the dynamical behavior in the data stream and we can find it in Section 6.1.

5.1.4. Multidimensional Scaling to analyze dispersion of association rules

We consider the previous plots concerning the amount of rules obtained according to different quality criteria are not enough for a real comparison between two approaches as an algorithm can find many rules which are very similar to each other (e.g., with slight differences on the variables used in the antecedent), which in practice is not so useful for expert decision making. On the contrary, an algorithm that generates association rules with more diversity is preferable. However, this has not been thoroughly analyzed in the specialized literature until now although some visualizations have been proposed [40,41].



(a) Not used (b) Used with Granularity 2 (c) Used with Granularity 3

Fig. 4. Three examples of variables representation used to calculate distances between rules where (*a*) shows the representation used when the variable is not used in the rule; (*b*) is an example of the representation used when the variable is used in the rule with granularity 2, and (*c*) is an example of the representation used when the variable is used in the rule with granularity 3.

Here we have proposed a new method with the aim of finding a way to represent how diverse the rules obtained by the algorithm are. If the rules are represented as points in a two-dimensional coordinate system, then we could easily represent three groups of points (each group with a different color) representing the rules obtained by three different algorithms, and so it would be pretty simple to visually compare which rules group spreads their points more evenly and which has the majority of its points (association rules) concentrated in certain small zones of the plot.

In order to represent these plots it is necessary to carry out a preprocessing phase that can be summed up in four steps: (1) we filter the rules using minimum thresholds of support and confidence (0.02 and 0.75, respectively, in our experiment); (2) we group all the obtained association rules from the three algorithms which have overcome the mentioned thresholds into a single set; (3) a distance matrix containing the distances between every pair of association rules is calculated; (4) MultiDimensional Scaling (MDS) [42] is applied to such distance matrix obtaining as output 2D points that represent the original association rules, which are shown in a scatter plot. So that the interpretation might be clearer, we have separated the rules according to the variable used as consequent and we have generated a plot for each case.

In order to use MDS, we need to define a distance function between fuzzy association rules as follows. Each fuzzy set is considered trapezoidal-shaped and represented by four values A, B, C and D. In the antecedent section we check which attributes appear in the rule and which do not. The attributes that do not appear in the antecedent of the rule are represented as in Fig. 4 (*a*), where $A = B =$ *minimum value of the attribute* and $C = D =$ *maximum value of the attribute*. If the attribute is present in the antecedent of the rule it will be associated with a granularity value and a linguistic term so it would be represented similarly to the examples of Fig. 4 (*b*) and Fig. 4 (*c*). We repeat the same operation for the consequent of the rule. Once this is finished, the distance between the fuzzy sets used in variable v in two different rules r_i and r_j is:

$$\mathrm{d}(FS_i^v, FS_j^v) = \frac{|A_i^v - A_j^v| + |B_i^v - B_j^v| + |C_i^v - C_j^v| + |D_i^v - D_j^v|}{4} \tag{6}$$

This distance for variable v in two different rules is normalized by the maximum possible distance between two rules for such variable, i.e., in the case of a maximum granularity of five fuzzy sets the distance between VS_5 and VL_5:

$$\bar{\mathrm{d}}(FS_i^v, FS_j^v) = \frac{\mathrm{d}(FS_i^v, FS_j^v)}{\mathrm{d}(VS_5^v, VL_5^v)} \tag{7}$$

Therefore, if we have n attributes $v_1, v_2, ..., v_n$ (in both antecedent and consequent parts), the distance between two rules r_i and r_j is:

$$d(r_i, r_j) = \sqrt{\sum_{k=1}^{n} \bar{\mathrm{d}}(FS_i^{v_k}, FS_j^{v_k})^2} \tag{8}$$

Finally, we build the entire distance matrix needed in MDS by computing the distance for every pair of fuzzy association rules in the set. The corresponding figures are shown in Section 6.1.

In addition to this graphic representation we have also designed a quantitative measure to numerically assess how sparse each group of rules is and quantify the difference between groups. To calculate this measure of diversity in association rules set, denoted as δ_{mds}, we start from the set of 2D points obtained as MDS output (each point represents an original fuzzy association rule) and complete a three-step process: (1) we separate the 2D points according to the source algorithm into three groups; (2) we calculate the Euclidean distance between every pair of points; and, finally, (3) obtain δ_{mds} as the mean of all these Euclidean distances in each group.

5.1.5. Streamgraphs

The streamgraph represents, for association rules with a specific variable in their consequent, how important each of the other variables are as antecedents of these rules at the very moment of the data stream.

To construct this graph we have a counter for each variable which is different from the one which has been selected to be in the consequent and for each time in which we are going to read the rules population. Then, for each of these times, we take all the rules whose consequents have the specified attribute and count the number of variables in the antecedent n. We add $1/n$ to each antecedent variable counter. For example, if a rule has three variables as antecedent, the area of each one of these variables is increased by 0.3333. If the rule has only one variable as antecedent, its area is increased by 1. Once this phase is completed, the graph is generated.

In the graph we can distinguish a color strip for each one of the antecedent variables and the width of each one is going to be proportional to the importance of this variable in the antecedent of the rules in a certain time. The sum of width from all the color strips is equal to the number of rules with the selected attribute in the consequent. The graph is displaced around a central axis to give an impression of fluidity. The correspondent figures are shown in Section 6.2.

5.1.6. Dependency wheels

The objective of this last visualization is to help understand in a certain moment how the associations between variables are distributed by showing which variables it seems to depend or which it seems to influence. To construct this graph we use the same width procedure as for streamgraph: the relevance of a variable that appears as antecedent is conditioned to the number of antecedents present in this same rule.

While the outer ring represents the different variables of the problem, the inner ring distinguishes when the variable appears as antecedent (dashed sectors) or consequent (dotted sectors). Therefore, the arcs always connect antecedent and consequent sectors. The width of these sectors for each variable is proportional to the number of rules in which this variable appears. Likewise, the width of the arcs represent the importance of these connections. The color of each arc corresponds to the one assigned to the antecedent variable of such link. In Section 6.2 dependency wheels obtained from the results of Fuzzy-CSar-AFP are shown.

5.2. Experimental setup

As it is described in [13], Fuzzy-CSar has several configuration parameters which enable it to adjust the behavior of the system and get models of maximal quality. In our experiments we respect almost all the configuration parameters used in [13] which were obtained experimentally following the recommendations found in [29]. However, in this paper we propose a challenging real-world data stream problem where 256 samples are analyzed every second during 60 minutes per subject (i.e., about a million of samples are processed in each experiment). Therefore, to deal with this high rate of in-coming data, the values of some configuration parameters have been changed. The configuration parameters which are different in our experiments are those related to experience thresholds: $\theta_{exp} = 10000$ (a rule that has been updated for about 40 seconds is experimented enough as to consider the performance estimation reliable), $\theta_{del} = \theta_{sub} = 4000$ (after about 15 seconds the rules can be deleted or subsumed), and the population size was set to 5000 individuals. The allowed fuzzy partition granularity is set to five fuzzy sets in Fuzzy-CSar, and between two and five in Fuzzy-CSar-AFP. The same algorithm configuration is applied to all the experiments.

As we have pointed before, Fuzzy-CSar-AFP is also analyzed in comparison with Fuzzy-Apriori [14], which integrates the Apriori algorithm and fuzzy sets concepts to discover interesting fuzzy association rules among quantitative values. Because Fuzzy-Apriori is not a data stream algorithm the dataset is partitioned into 30 subsets, each one containing 30720 samples (data registered during two minutes of experiment) and Fuzzy-Apriori is applied on each subset. So we can obtain results from Fuzzy-Apriori for different stretches from the original dataset, and not only at the end. The same analysis is performed for two different options with aim of better analysis the impact of adaptive fuzzy partitions regardless of whether the attribute's domains evolve or not: (1) one in which the minimum and maximum values used by the algorithms for each one of the input attributes evolve as described in Section 3.6, and (2) another one in which the three algorithms use the absolute domains of each input attribute. All results shown in Section 6 have been obtained following this procedure.

6. Results

The data streams described in Section 4 are used in several experiments in order to corroborate our hypothesis considering Fuzzy-CSar-AFP better able to adapt to the peculiarities of each variable, and so obtaining better and more heterogeneous rules that could help discover useful associations between the forming features of the data. In this paper we widely compare Fuzzy-CSar-AFP with Fuzzy-Apriori [14] and Fuzzy-CSar. For a further analysis of the performance of Fuzz-CSar's architecture to deal with concept drift and a comparative in static data we refer the interested reader to [13]. For this comparison we employ two different versions of the mentioned data streams. The full data streams with their 64 variables are used to assess the time-performing of the algorithms. However, to analyze and compare the sets of rules obtained by the different algorithms we select EEG data from six scalp locations [43] in order to enable a more conscious analysis. These six selected scalp locations are: frontal left (Fp1), frontal right (Fp2), occipital left (O1), occipital right (O2), central left (C3) and central right (C4).

As we have mentioned above, the experiments performed can be grouped into two groups depending on whether the algorithms (Fuzzy-Apriori, Fuzzy-CSar and Fuzzy-CSar-AFP) employ the absolute domain of each attribute or the evolving ones (Section 3.6). The comparisons are performed separately in these two groups of experiments due to the difficulty to

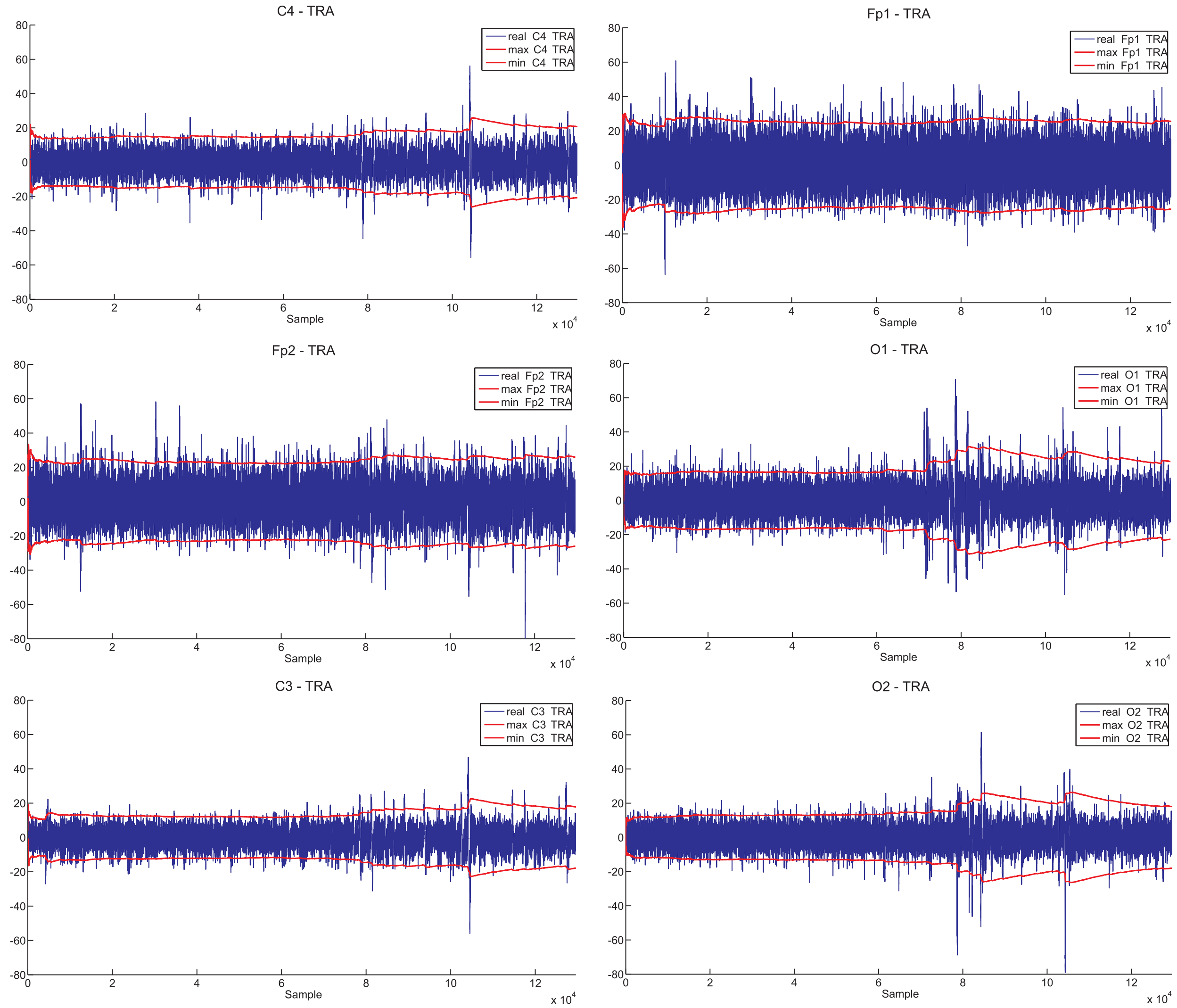


Fig. 5. Evolution of real value (blue), algorithm maximum value (red) and algorithm minimum value (red) for each input attribute as data of trained subject (*TRA*) are processed. Because of the high sampling rate and to reduce the weight of the images, only 1 in 20 data is represented in these plots. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

carry out a fair comparison (specially in terms of dispersion) between algorithms which use different attribute's domains. In any case, we must remember that the principal aim of updating attribute's domains is to make the algorithm useful in real-world problems where it is very common not to know the range of values of the attributes.

6.1. Analysis of Fuzzy-CSar-AFP performance and comparative results

6.1.1. Domains update

In Section 3.6 we have explained how each time a new datum is processed by the algorithm, the range of each variable (defined by its minimum and maximum values) is updated including the value of the variable in the new datum (if it is present). In Fig. 5 and Fig. 6 we can see the evolution of the minimum and maximum values during the algorithm application for every attribute for two different subjects. It is possible to observe how both minimum and maximum values are updated no matter if it is because of a value bigger or lower than the majority of the data. In the mentioned figures we can see how, even when the fitting is not completely perfect, it is pretty good and in any case much better than the option of having to process the data and using the absolute domains of the attributes. The figures also show the advantage of the proposed domain updating mechanism to filter noise and peaks, i.e., using the absolute maximum and minimum value to limit the domain of the attributes there are several fuzzy partitions (labels) which are not going to be used hardly ever. For example, we can analyze the case of C4 for *TRA* where if the algorithm takes the absolute limits some fuzzy partitions cover values that only appear in one particular point of the process, irrelevant compared with the full dataset.

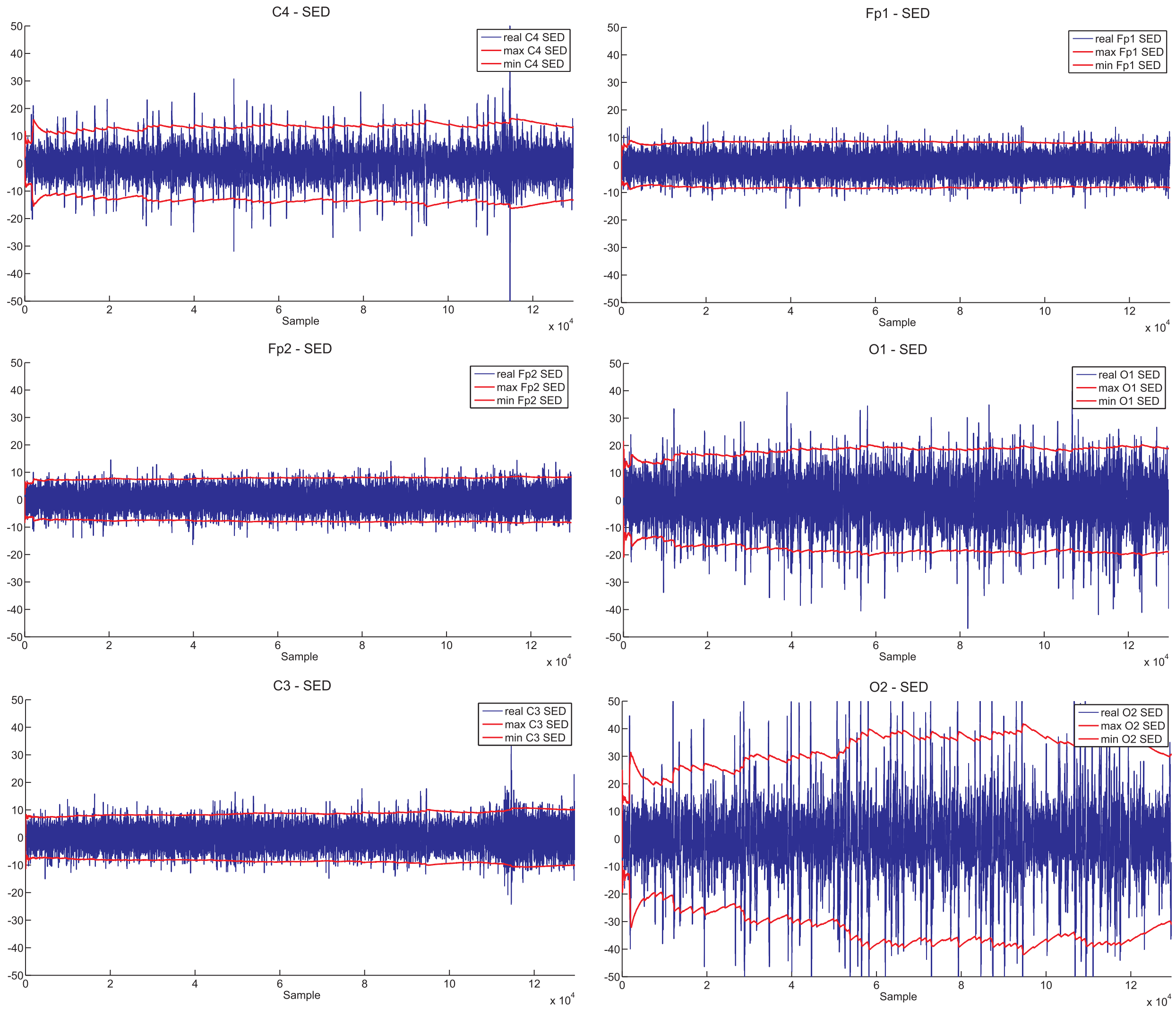


Fig. 6. Evolution of real value (blue), algorithm maximum value (red) and algorithm minimum value (red) for each input attribute as data of not trained subject (*SED*) are processed. Because of the high sampling rate and to reduce the weight of the images, only 1 in 20 data is represented in these plots. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

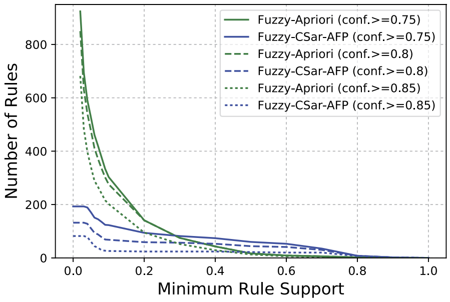
Table 2
Number of rules (#*R*) obtained by Fuzzy-Apriori for the different confidence thresholds and the lowest minimum support values represented in Fig. 7 for both trained (*TRA*) and sedentary (*SED*) subjects when evolving domains.

min. supp.	0			0.01			0.02		
min. conf.	0.75	0.8	0.85	0.75	0.8	0.85	0.75	0.8	0.85
TRA #R	317514	274020	212082	2608	794	30	846	42	0
SED #R	156852	135884	104812	1282	334	17	395	23	1

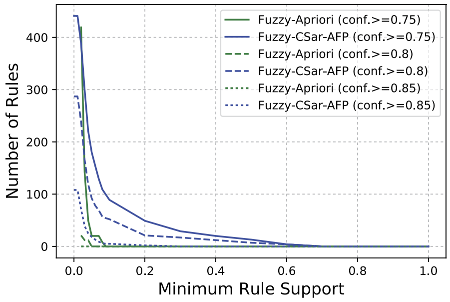
6.1.2. *Number of rules with minimum confidence vs. minimum support*

In Fig. 7 it is possible to observe the number of rules with certain confidence thresholds and different minimum support values (from 0.0 to 1.0). Note that the graphs shown in Fig. 7 do not show the number of rules generated by Fuzzy-Apriori for minimum support values lower than 0.02 because it is too high and would avoid to correctly appreciate the whole plot. In any case, this number falls quickly as the support rises slightly, as shown in Table 2.

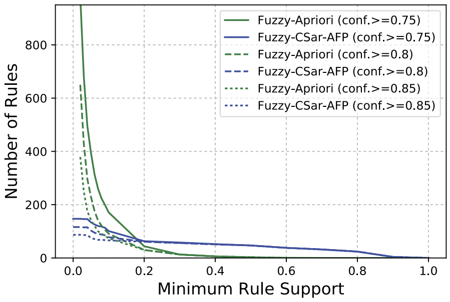
As the minimum support value increases slightly the number of rules from Fuzzy-Apriori decreases dramatically and gets lower than the number of rules from Fuzzy-CSar-AFP in both *TRA* and *SED* data flows, and in both absolute and evolving attribute's domains (Fig. 7). In the case of evolving domains, this phenomenon is much more noticeable. In this case, if we focus on support values between 0.0 and 0.1 our attention is drawn to the fact that the number of Fuzzy-Apriori's rules starts descending quickly, then seems to stabilize for a moment and finally descends sharply again. Before the minimum support



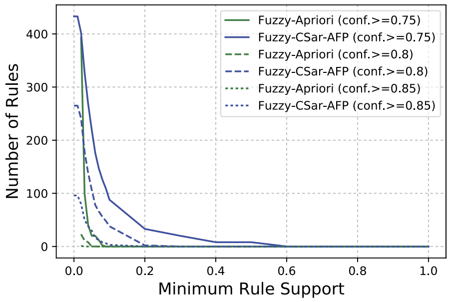
(a) TRA subject, absolute domains



(b) TRA subject, evolving domains



(c) SED subject, absolute domains



(d) SED subject, evolving domains

Fig. 7. Amount of rules from Fuzzy-CSar-AFP (blue) and Fuzzy-Apriori (green) which get over three different confidence thresholds ($conf \geq 0.75$, $conf \geq 0.8$ and $conf \geq 0.85$) and minimum support thresholds from 0.0 to 1.0. The results are shown for both the experiments with absolute domains (left) and evolving attribute's domains (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

reaches 0.1 the number of rules from Fuzzy-Apriori falls to zero for both subjects and for the three possible confidence thresholds. This evolution in the amount of rules discovered by Fuzzy-Apriori is overriding to the choice of rules quality thresholds for other comparatives and analysis shown in this paper. Unlike Fuzzy-Apriori, Fuzzy-CSar-AFP draws a smoother curve and continues generating rules even for quite high support minimum values, also in the case of absolute attribute's domains.

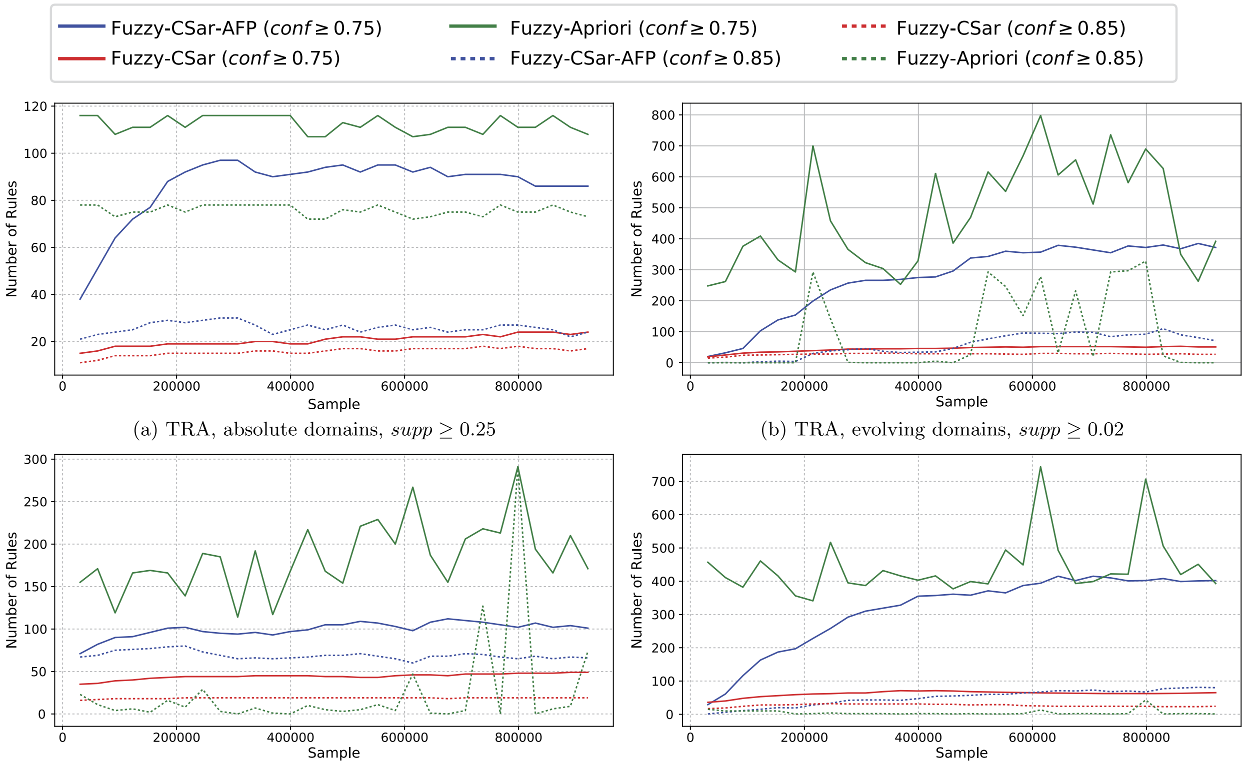
6.1.3. *Evolution of the amount of good rules*

In Fig. 8 we can observe the number of rules which beat minimum support and confidence values as the number of samples processed by the algorithms increases. Each algorithm is identified by a different color. It is important to remember that Fuzzy-Apriori is not a data stream algorithm and due to that it has been executed on data windows. Each window represents data recording for two minutes (30 720 samples).

With basis on Fig. 7 we have selected minimum support thresholds for which the number of filtered rules from Fuzzy-Apriori and Fuzzy-CSar-AFP tends to be similar and not too low. The minimum supports are 0.02 for evolving domains, 0.25 for absolute domains in trained subject (*TRA*) and 0.10 for absolute domains in sedentary subject (*SED*). These are the support filters applied to obtain the plots shown in Fig. 8 combined with two different confidence thresholds (0.75 and 0.85).

In Fig. 8 we can observe how for 0.75 as minimum confidence threshold and the selected minimum support the number of filtered rules generated by Fuzzy-Apriori and by Fuzzy-CSar-AFP maintain similar values for the whole experiment. If we focus on the comparative between Fuzzy-CSar and Fuzzy-CSar-AFP, we appreciate how when the number of processed samples starts to increase the amount of different rules obtained by Fuzzy-CSar-AFP is always higher.

Since Fuzzy-Apriori is not an incremental algorithm the trend of the evolution on the number of different rules is not the same as for Fuzzy-CSar and Fuzzy-CSar-AFP, and there are noticeable differences among the four graphics. First, in the case of absolute domains, we find some differences between trained and sedentary subject. For the trained subject the number of filtered rules of Fuzzy-Apriori is always higher than the number of filter rules of Fuzzy-CSar-AFP for both 0.75 and 0.85 confidence thresholds. However, for the sedentary subject when the minimum confidence increases to 0.85 the number of rules from Fuzzy-Apriori descends a lot getting lower than the number of rules generated by Fuzzy-CSar-AFP. In the case of evolving domains, we also find some differences between both subjects. For the trained subject and 0.75 as confidence threshold the amount of filtered rules from Fuzzy-Apriori is always bigger than the number from Fuzzy-CSar-AFP. However, if we raise the confidence threshold up to 0.85, Fuzzy-Apriori only beats Fuzzy-CSar-AFP in a few cases. For the sedentary subject the results are quite different. When we choose 0.75 as minimum confidence and the amount of processed samples is enough, the number of filtered rules from Fuzzy-Apriori and Fuzzy-CSar-AFP keep quite close. But when the minimum confidence comes to 0.85 the number of filtered rules from Fuzzy-Apriori falls significantly to be bellow both Fuzzy-CSar-AFP



(a) TRA, absolute domains, $supp \geq 0.25$

(b) TRA, evolving domains, $supp \geq 0.02$

(c) SED, absolute domains, $supp \geq 0.10$

(d) SED, evolving domains, $supp \geq 0.02$

Fig. 8. Representation of how the amount of different rules evolves as the data are processed in both subject cases. Normal lines represent number of rules generated by the algorithms whose confidence is equal or greater than 0.75 while dotted lines represent rules which get over a confidence threshold of 0.85. The results are shown for both the experiments with absolute domains (left) and evolving domains (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and Fuzzy-CSar throughout the experiment. In summary, we can say that the number of rules from Fuzzy-CSar-AFP gets over the number of rules from Fuzzy-CSar. However, the relationship between the number of different rules obtained by Fuzzy-CSar-AFP and Fuzzy-Apriori is not so clear even though we can observe that in most cases the difference between the confidence thresholds (0.75 and 0.85) is most strongly seen in Fuzzy-Apriori. Anyway, this analysis is going to be expanded throughout this section.

6.1.4. Multidimensional Scaling to analyze dispersion of association rules

As we could ascertain in Section 6.1.2, the number of rules discovered by Fuzzy-Apriori decreases very rapidly as the minimum support value moves away from zero, describing a different evolution from that of the rules discovered by Fuzzy-CSar-AFP, giving up generating rules for supports lower than 0.1 in the case of evolving domains of attributes. To allow a more fair comparison in terms of dispersion between these two algorithms we decided to choose minimum support and confidence thresholds for which both algorithms obtain a very close number of rules. Therefore, all the figures, tables and results commented in this section refer to rules filtered according to a support threshold of 0.02 in experiments corresponding to evolving domains, meanwhile a support threshold of 0.10 is used in case of absolute domains and sedentary subject, and a support threshold of 0.25 in case of absolute domains and trained subject. Confidence threshold of 0.75 is used in all the cases.

Figs. 9 and 10 contain a scatter plot for each possible attribute in the consequent in which each point matches to an original association rule (as explained in Section 5.1.4) discovered by Fuzzy-Apriori (green), Fuzzy-CSar (red) or Fuzzy-CSar-AFP (blue) at the end of the experiment, i.e., when 60 minutes of data have been processed (921 600 samples). Tables 3 and 4 contain δ_{mds} values and number of rules computed for the same cases.

In Tables 3 and 4 along with Figs. 9 and 10 we can quantitatively and visually analyze how sparse the rules of each algorithm are distributed. Visually, we can observe that for both trained (Fig. 9) and sedentary (Fig. 10) subjects the rules obtained by Fuzzy-CSar-AFP (blue points) use to be more distanced from each other and quite sparse, covering a wide area and having rules for each possible attribute in the consequent. If we check Tables 3 and 4, δ_{mds} values confirm that in most of the analyzed cases Fuzzy-CSar-AFP is the algorithm whose rules are distributed more widely, with a higher mean distance between each other.

Table 3
Comparison between the number (#R) and dispersion (δ_{mds}) of the rules obtained by Fuzzy-Apriori, Fuzzy-CSar and Fuzzy-CSar-AFP for the *TRA* dataset. The rules included in this table have overcome support and confidence thresholds ($conf \geq 0.75$ and different minimum supports). The results are shown for both the experiments with both absolute and evolving domains.

		Fp2		C4		O2		O1		C3		Fp1	
		δ_{mds}	#R	δ_{mds}	#R	δ_{mds}	#R	δ_{mds}	#R	δ_{mds}	#R	δ_{mds}	#R
Absolute	F-Apriori	0.144	35	0.0	0	0.152	36	0.0	0	0.148	37	0.0	0
	FCSar	0.161	7	0.0	0	0.156	8	0.0	0	0.162	9	0.0	0
	FCSar-AFP	0.095	4	0.132	19	0.107	8	0.124	16	0.106	12	0.134	27
Evolving	F-Apriori	0.180	64	0.184	75	0.182	71	0.186	62	0.183	58	0.175	62
	FCSar	0.163	12	0.155	10	0.150	9	0.079	2	0.170	12	0.153	6
	FCSar-AFP	0.183	62	0.183	60	0.190	53	0.174	54	0.190	56	0.207	38

Table 4
Comparison between the number (#R) and dispersion (δ_{mds}) of the rules obtained by Fuzzy-Apriori, Fuzzy-CSar and Fuzzy-CSar-AFP for the *SED* dataset. The rules included in this table have overcome support and confidence thresholds ($conf \geq 0.75$ and different minimum supports). The results are shown for both the experiments with both absolute and evolving domains.

		Fp2		C4		O2		O1		C3		Fp1	
		δ_{mds}	#R	δ_{mds}	#R	δ_{mds}	#R	δ_{mds}	#R	δ_{mds}	#R	δ_{mds}	#R
Absolute	F-Apriori	0.147	28	0.150	74	0.137	40	0.144	7	0.0	0	0.153	22
	FCSar	0.122	7	0.0	0	0.137	4	0.160	11	0.179	15	0.156	12
	FCSar-AFP	0.159	19	0.151	23	0.159	15	0.187	8	0.126	16	0.162	20
Evolving	F-Apriori	0.171	45	0.187	68	0.171	76	0.168	67	0.194	54	0.193	59
	FCSar	0.089	10	0.240	6	0.126	6	0.135	11	0.137	19	0.140	13
	FCSar-AFP	0.179	80	0.165	54	0.203	53	0.203	64	0.181	52	0.178	52

It is specially worth highlighting those cases in which Fuzzy-Apriori obtains more rules but, however, Fuzzy-CSar-AFP gets a higher δ_{mds} value, such as O2 for the four cases. In these examples, Fuzzy-Apriori obtains more rules than Fuzzy-CSar-AFP but δ_{mds} is clearly greater for Fuzzy-CSar-AFP, i.e., Fuzzy-CSar-AFP manages to discover with less association rules a more diverse and widespread knowledge. An extremely low number of rules is neither desirable even if δ_{mds} is higher. The aim is to find a balance between the number of quality rules and how sparse they are.

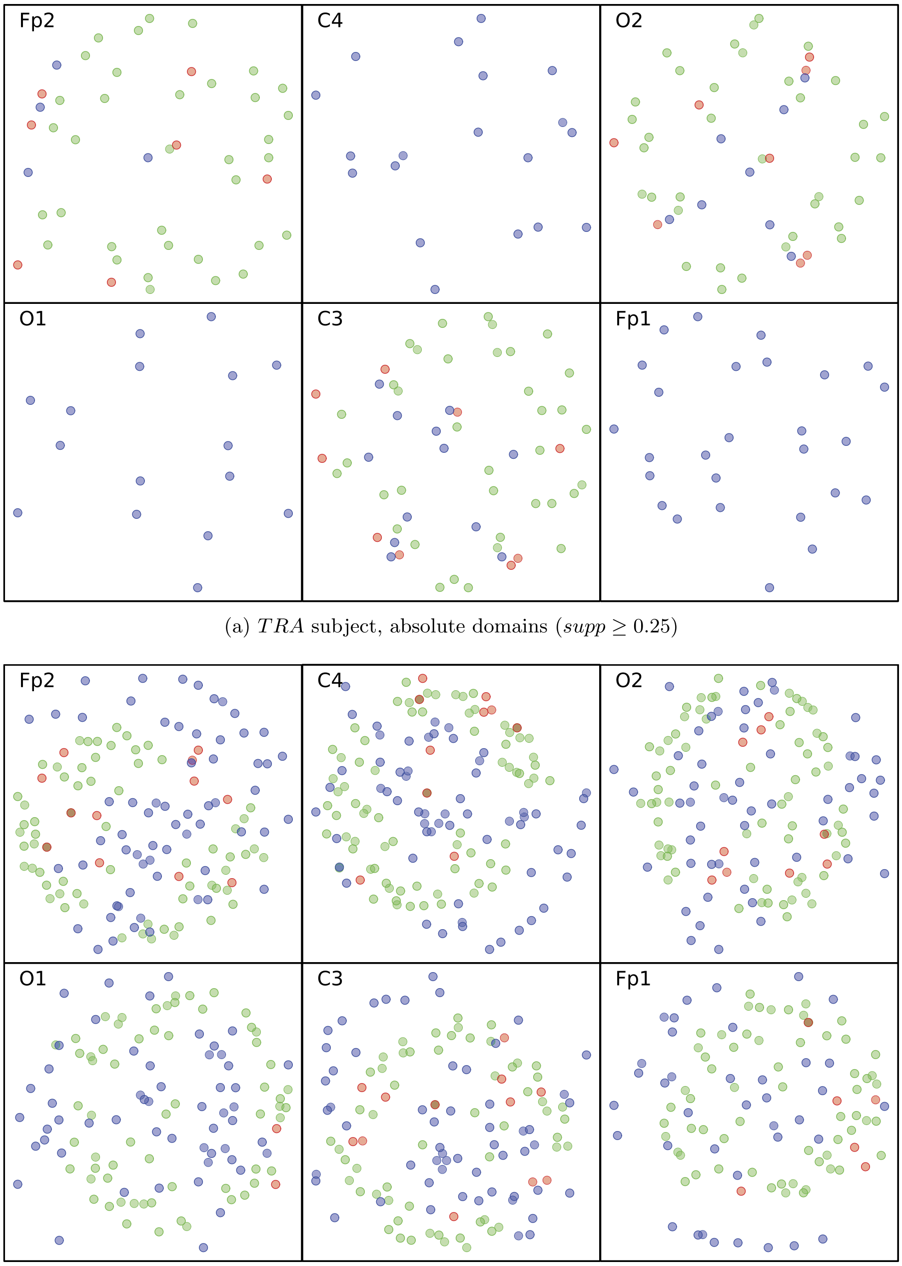
This analysis is focused only in a specific moment of the process, the final one, when all the samples have been processed. In order to have a more global vision of the evolution of δ_{mds} throughout the full process, in Fig. 11 we represent δ_{mds} at 30 different points of the experiment. δ_{mds} value for each algorithm's rules is computed everytime two new minutes of data (30720 new samples) have been processed by the algorithms. The association rules are filtered by the same quality parameters as in the previous figures and tables ($conf \geq 0.75$ and $supp \geq 0.02$).

In the plots contained in Fig. 11 we can appreciate that Fuzzy-CSar-AFP obtains very good results of δ_{mds} for hardly the entire duration of the experiments in all cases. If we focus on the comparative between Fuzzy-CSar-AFP and Fuzzy-Apriori, Fuzzy-CSar-AFP equals or improves Fuzzy-Apriori in all experiments (both subjects and both domains). The improvement of Fuzzy-CSar-AFP over Fuzzy-Apriori is more noticeable in the case of absolute domains. But even in the case of evolving domains Fuzzy-CSar-AFP beats Fuzzy-Apriori in most of the occasions. Furthermore, it is important to recall that Fuzzy-Apriori is not really an incremental online algorithm but we apply it on different windows of data. If we would need to get a real time update of the state of the population of rules each time a new data is received, this would be possible with Fuzzy-CSar or Fuzzy-CSar-AFP but not with Fuzzy-Apriori. When comparing Fuzzy-CSar with Fuzzy-CSar-AFP we can clearly notice that Fuzzy-CSar-AFP gets better results for three out of four experiments, and with a smaller difference also gets better results in the remaining one. Furthermore, if we also check Fig. 8, 0.75 as confidence thresholds we observe that the number of Fuzzy-CSar's rules after applying the filters is lower. So we can conclude that Fuzzy-CSar-AFP tends to extract a sufficiently high number of quality rules that are less similar to each other during the whole experiment than the ones obtained by other algorithms in most cases, i.e., these rules represent different knowledge rather than a lot of rules expressing the same information. As we have explained before, an algorithm that generates rules with more diversity is preferable.

6.1.5. Analysis of efficiency

In order to check and compare the efficiency of the algorithms, two different types of experiments have been performed. Firstly, the computation time spent by each algorithm in processing every two new minutes of data (30 720 sampling) is registered. Since the minimum support value specified as input argument of Fuzzy-Apriori determines its number of frequent itemsets and, therefore, influences its execution time, we carry out several tests varying the support threshold. Fuzzy-CSar and Fuzzy-CSar-AFP do not need minimum support as input parameter. The results of these tests are shown in Fig. 12. The experiments are performed in a Intel® Core™ i7-4790 3.60 GHz, RAM 16 GB DDR3 (1600 MHz) computer.

In both figures, we can observe how the support threshold is a key factor in the execution time of Fuzzy-Apriori. Relating these results with Fig. 7, as the number of rules obtained by Fuzzy-Apriori quickly decreases if the minimum support value increases a little, its execution time follows the same trend. Comparing the performances of Fuzzy-CSar-AFP and Fuzzy-CSar, the latter gets better efficiency because of it maintains a significantly lower number of rules (see Fig. 8). In other words, in

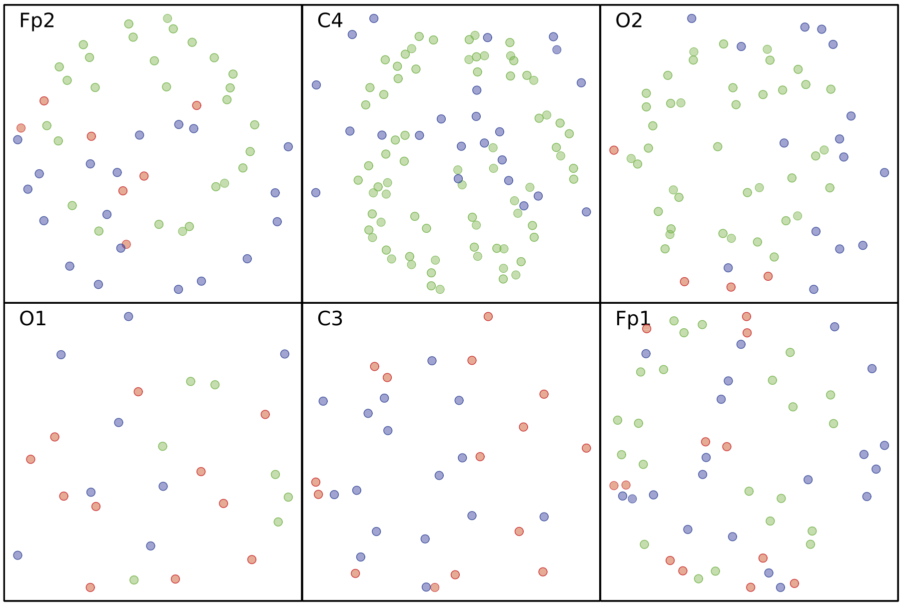


(b) TRA subject, evolving domains ($supp \geq 0.02$)

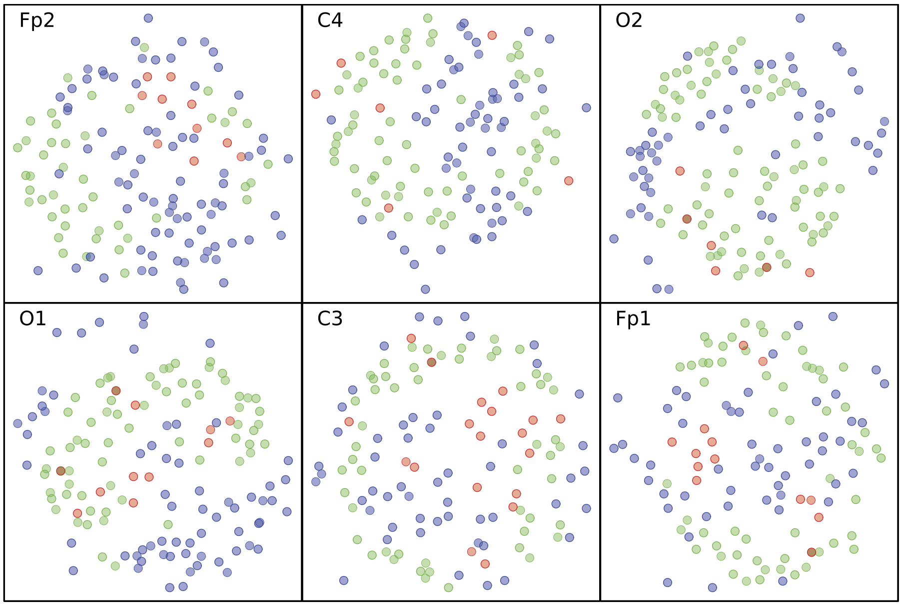
Fig. 9. Two-dimensional representations of how the rules are distributed in Fuzzy-CSar-AFP (blue points), Fuzzy-CSar (red) and Fuzzy-Apriori (green) for trained subject (*TRA*) data. The results are shown for both the experiments with absolute domains (up) and evolving domains (down). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

exchange for getting many more quality rules, Fuzzy-CSar-AFP needs to evolve a wider pool of rules that increases processing time. In addition, we can observe how in the first subsets both Fuzzy-CSar and Fuzzy-CSar-AFP register lower time values due to the incremental and online character of these algorithms. Initially, the population of association rules maintained by these algorithms during the whole process is empty, then it begins to grow up and, finally, to stabilize, as well as the execution time of the algorithm.

Another key factor in the time efficiency of these algorithms is the number of attributes forming the input data. In order to test the influence of this factor on these three algorithms, we test them on the full original EEG dataset (before feature selection) including its 64 attributes. The average execution times per sample registered along with those recorded using the 6-attribute dataset are shown in Table 5. The values shown in this table are the average results after three runs. In this table we can appreciate how Fuzzy-CSar-AFP scales much better than Fuzzy-Apriori. With the 6-attribute dataset, both



(a) SED subject, absolute domains ($supp \geq 0.10$)

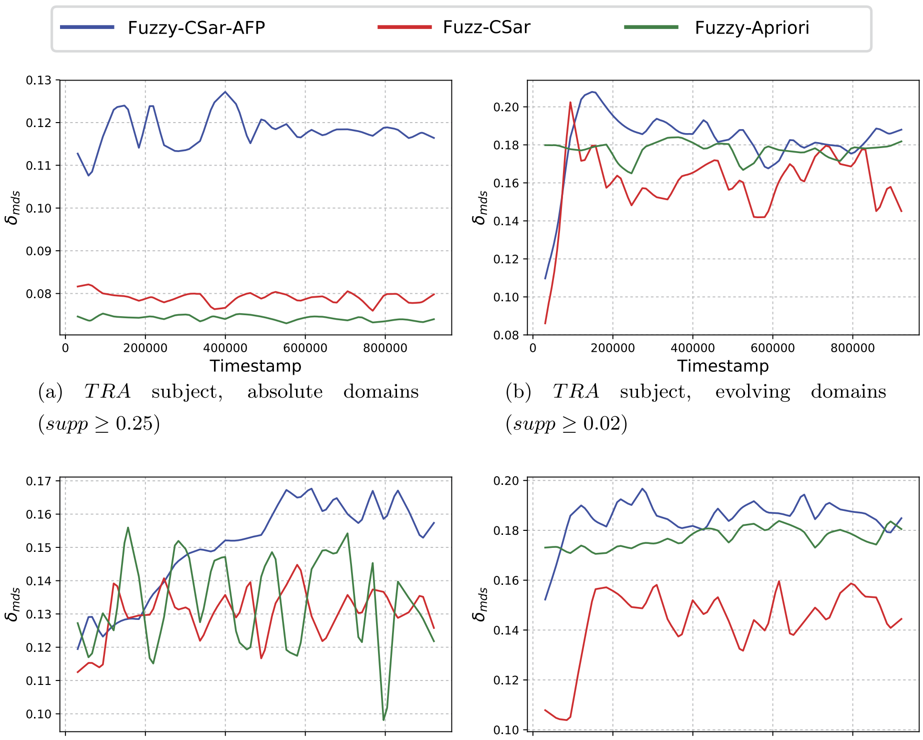


(b) SED subject, evolving domains ($supp \geq 0.02$)

Fig. 10. Two-dimensional representations of how the rules are distributed in Fuzzy-CSar-AFP (blue points), Fuzzy-CSar (red) and Fuzzy-Apriori (green) for sedentary subject (*SED*) data. The results are shown for both the experiments with absolute domains (up) and evolving domains (down). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fuzzy-CSar-AFP and Fuzzy-Apriori can manage a sampling frequency of 1000 Hz. However, with the 64-attribute dataset, Fuzzy-CSar-AFP can manage 200 Hz while Fuzzy-Apriori needs about a second and a half to process each sample. That is, for a sampling frequency of 200 Hz, Fuzzy-Apriori may spend more than 44 hours to process the data recorded during 10 minutes of experiment, which is completely unfeasible in a real-world data stream problem. Fuzzy-CSar-AFP, on the contrary, is able to process the data on-the-fly.

In Table 5 we can also discover another interesting fact: the time registered by Fuzzy-CSar is lower than the time registered by Fuzzy-CSar-AFP for the case of the 6-attribute dataset but higher for the 64-attribute dataset. This happens because Fuzzy-CSar obtains a lower amount of different rules than Fuzzy-CSar-AFP for the 6-attribute dataset (as we showed above) but gets more different rules than Fuzzy-CSar-AFP for the 64-attribute dataset (1919 vs. 1547). However, these higher number of rules does not imply more quality. Indeed, Fuzzy-CSar-AFP obtains more high-quality rules (with minimum



(a) TRA subject, absolute domains ($supp \geq 0.25$)

(b) TRA subject, evolving domains ($supp \geq 0.02$)

(c) SED subject, absolute domains ($supp \geq 0.10$)

(d) SED subject, evolving domains ($supp \geq 0.02$)

Fig. 11. Comparison between the evolution of δ_{mds} for the association rules (filtered according to $conf \geq 0.75$ and different support thresholds) obtained by Fuzzy-CSar-AFP (blue), Fuzzy-CSar (red) and Fuzzy-Apriori (green) for trained *TRA* (up) and sedentary *SED* (down) subjects. The results are shown for both absolute domains (left) and evolving domains (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Average execution time (in seconds) per sample of Fuzzy-Apriori, Fuzzy-CSar and Fuzzy-CSar-AFP applied on 6-electrodes and 64-electrodes dataset.

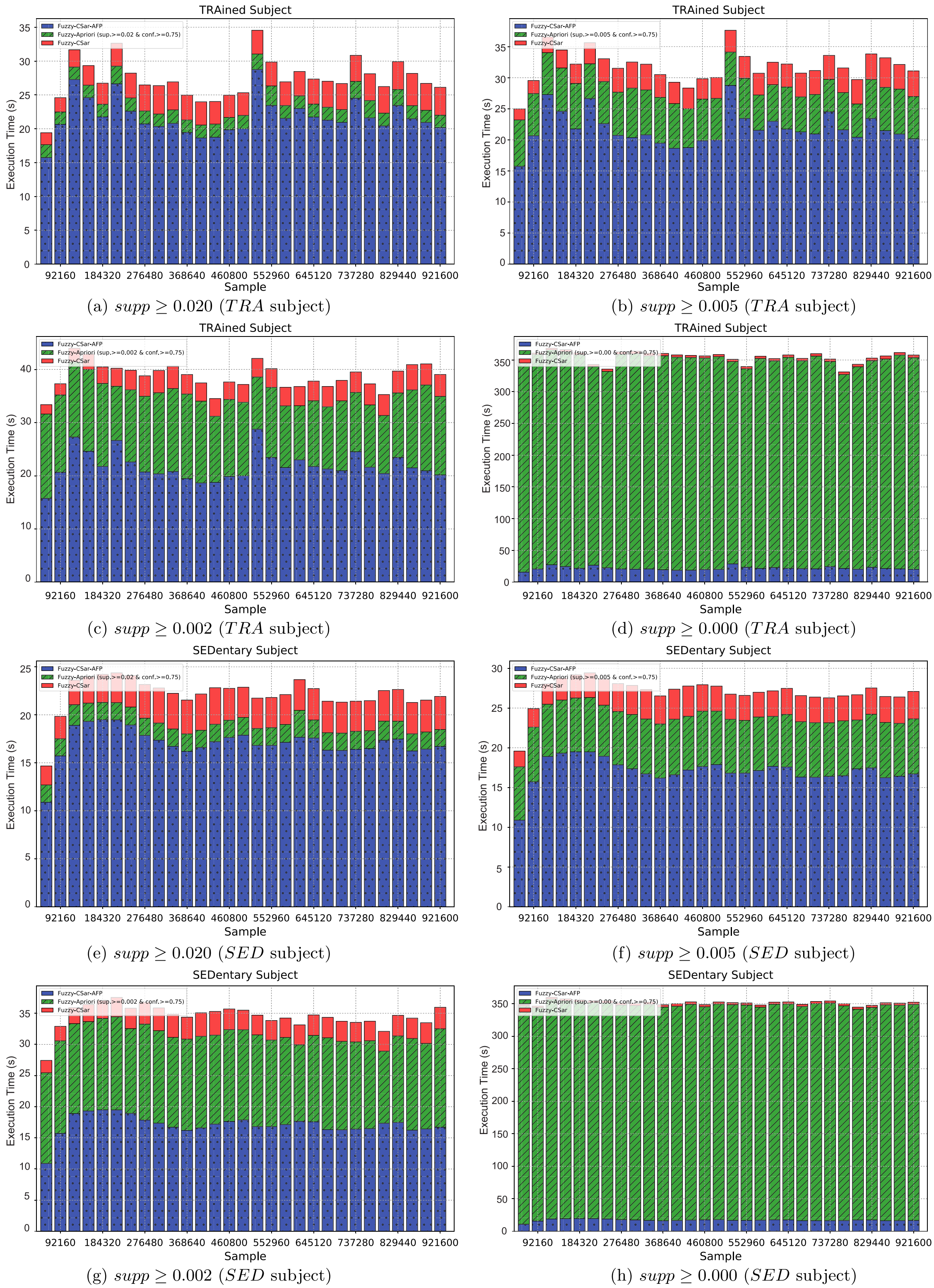
	6-attribute dataset		64-attribute dataset	
	TRA	SED	TRA	SED
Fuzzy-Apriori	0.000022	0.000018	1.337197	1.265868
Fuzzy-CSar	0.000203	0.000166	0.007194	0.007203
Fuzzy-CSar-AFP	0.001092	0.001057	0.004616	0.005056

confidence of 0.85) than Fuzzy-CSar. The 64-attribute dataset is a very sparse problem so Fuzzy-CSar does not enhance quality properly, thus generating a high number of different rules but with poorer quality.

6.2. Interpretation of obtained results in psychophysiology

The association streams in both subjects (Figs. 14 and 13) revealed a stable increase in the number of association rules throughout the experimental session. This is an expected behavior for biological signals that fluctuate in response to variations in common underlying physiological sources. The result confirms the sensitivity of the method when detecting natural associations between signals from electrodes at distinct scalp locations. In addition, the method shows robust to mild physiological changes induced by mental fatigue during a behavioral task with low cognitive demands. This robustness of the algorithm to the factor of time and to modest cognitive effort is important for further experimentation with more demanding conditions marked by pronounced physiological changes (e.g. sleep vs. awake and distinct sleep stages, tasks with significant cognitive load, etc.).

Although the algorithm proved robust to changes induced by the length and the particular behavioral demands of the task, it also proved capable of tracking the evolution of specific association rules throughout the experimental session. Differences in the dynamical pattern of concrete association rules between subjects seem like a promising methodological tool for detecting minute which would be impossible to assess with current averaging or pair-wise correlation techniques.



(a) $supp \geq 0.020$ (TRA subject)

(b) $supp \geq 0.005$ (TRA subject)

(c) $supp \geq 0.002$ (TRA subject)

(d) $supp \geq 0.000$ (TRA subject)

(e) $supp \geq 0.020$ (SED subject)

(f) $supp \geq 0.005$ (SED subject)

(g) $supp \geq 0.002$ (SED subject)

(h) $supp \geq 0.000$ (SED subject)

Fig. 12. Execution times spent by Fuzzy-CSar-AFP (dotted blue bars), Fuzzy-CSar (red bars) and Fuzzy-Apriori (dashed green bars) to process each new subset (30720 sampling, 2 minutes of data recording) varying the minimum support value used by Fuzzy-Apriori. A minimum confidence value of 0.75 is used by Fuzzy-Apriori. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

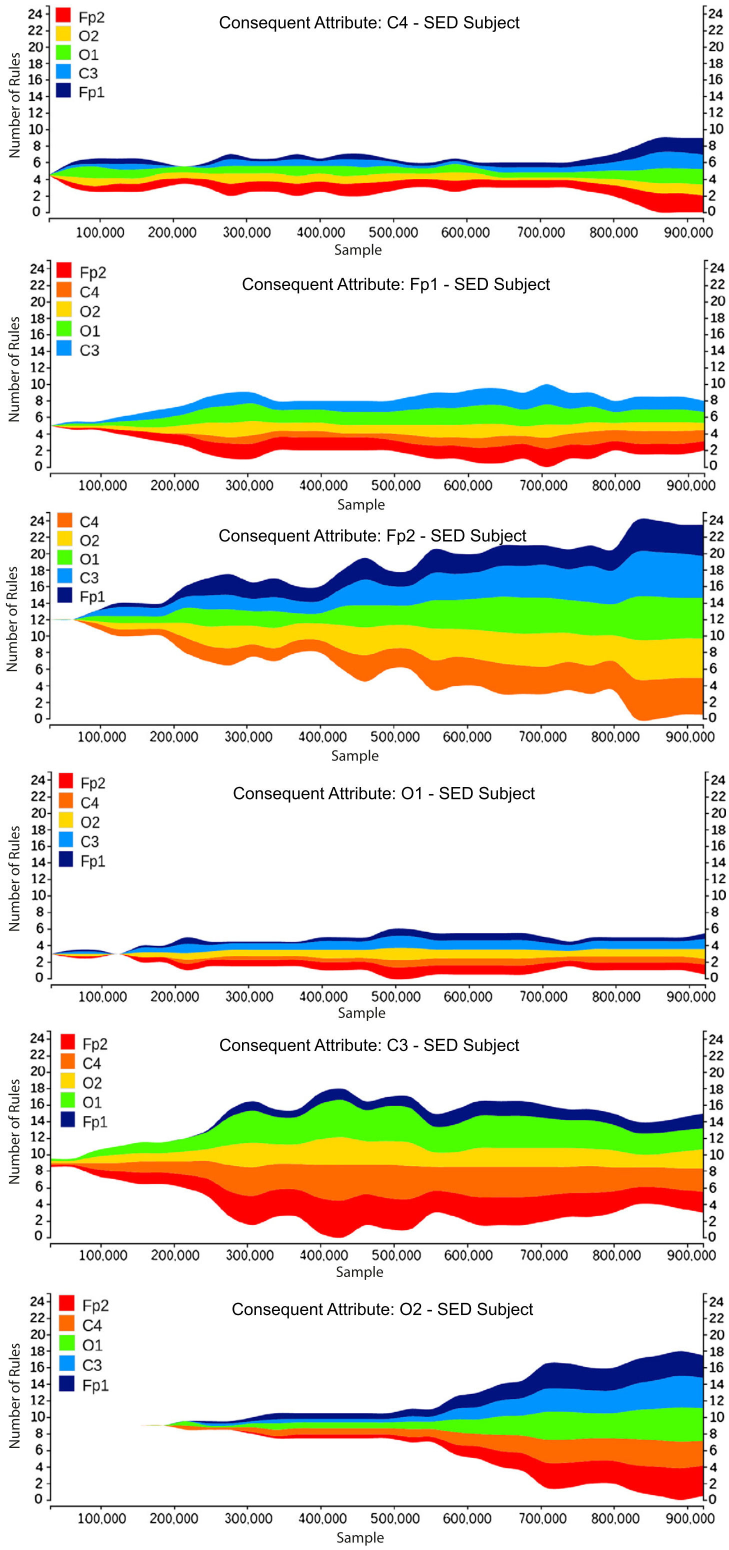


Fig. 13. Stream graphs for *SED* (sedentary subject) *supp* ≥ 0.02 and *conf* ≥ 0.8.

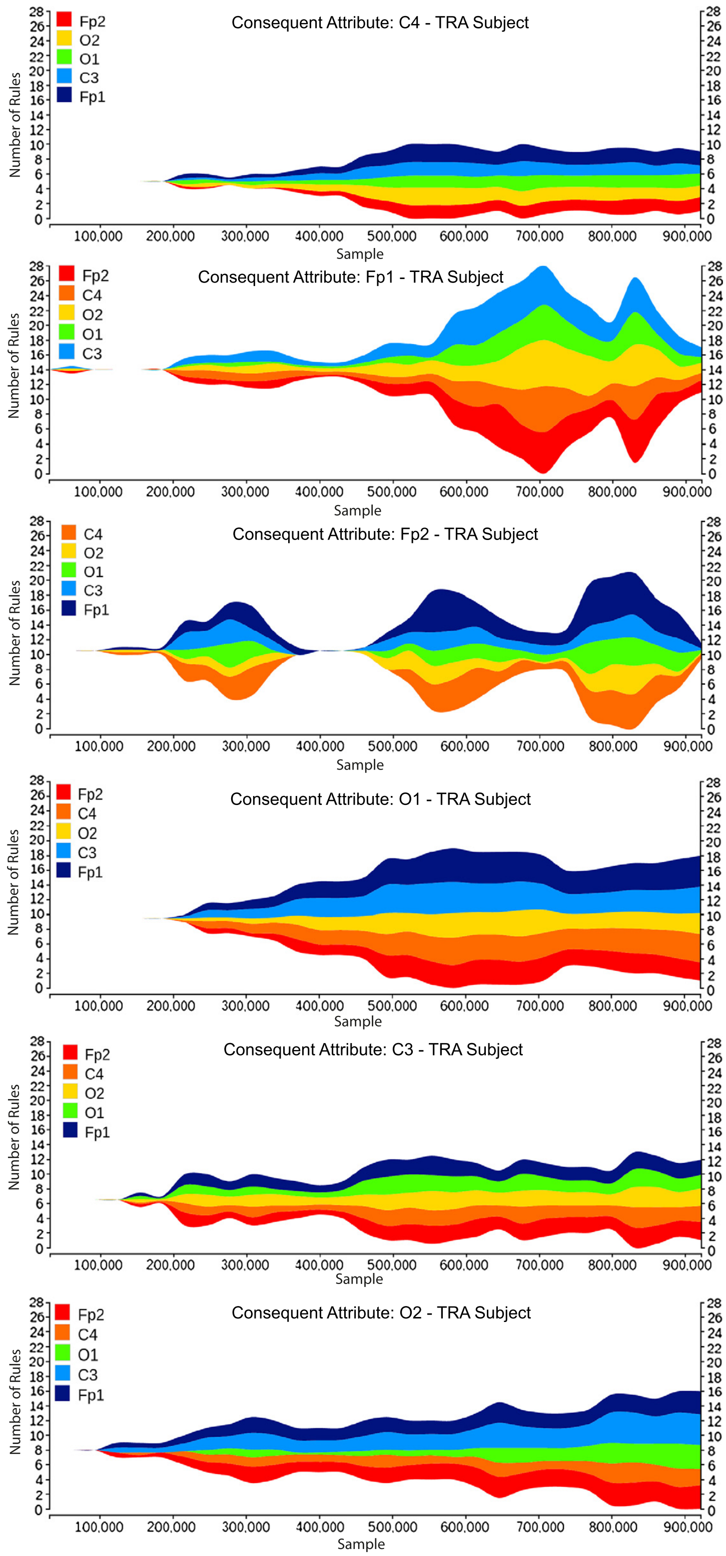
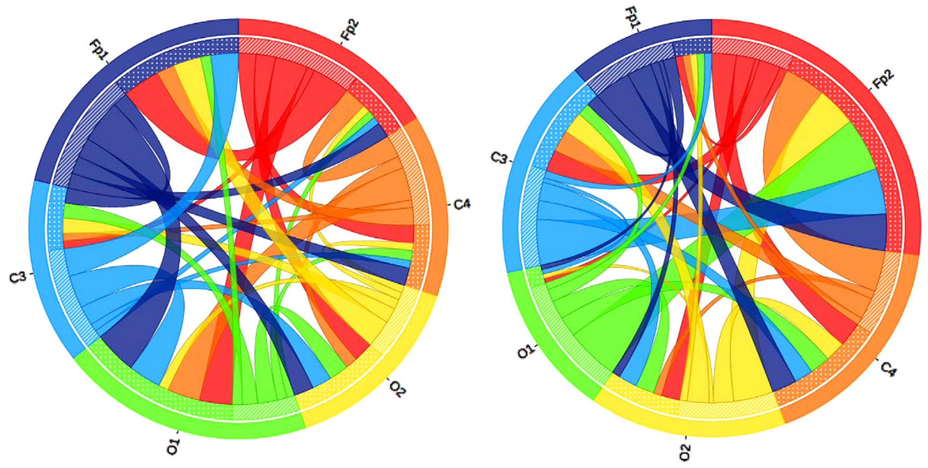


Fig. 14. Stream graphs for *TRA* (trained subject) *supp* ≥ 0.02 and *conf* ≥ 0.8.



(a) TRA Subject (b) SED Subject

Fig. 15. Dependency wheels for (a) *TRA* and (b) *SED* subjects at the end of the sampling with $supp \geq 0.05$ and $conf \geq 0.85$.

Fig. 15 shows the dependency wheels of very good fuzzy association rules ($supp \geq 0.05$ and $conf \geq 0.85$) obtained for both *TRA* and *SED* subjects at the end of the experiment (921 600 sampling, i.e., after 60 minutes). At this level of performance, we can observe differences between the fuzzy association rules in these two cases. We can perform a zone analysis of these two graphics and highlight certain contrasts between them. Looking at the frontal electrodes (Fp1 and Fp2) in *TRA* is Fp1 the frontal electrode with greater presence as consequent (dotted sectors) and Fp2 appears mainly as antecedent (dashed sectors). This behavior is completely reversed in *SED*. On the one hand, Fp2 has a much larger total occurrence in *SED* and most of it as consequent. On the other hand, Fp1 loses relevance, chiefly, its consequent sector. In the central electrodes (C3 and C4) there are not so significant differences between subjects, even though the relevance of C3 as consequent (dotted sectors) is clearly bigger in *TRA*. Finally, looking at the last electrodes (O1 and O2) we appreciate again significant variances between subjects, specially in O1 that has a pretty important consequent sector in *TRA* while in *SED* its consequent sector almost does not exist. Instead of focusing on each individual electrode, we also can pay attention to the relationships and dependencies established between them. Many of these relations also change depending on the subject but we can also distinguish certain associations which seem steady and more independent to subject change such as those built between Fp1 and Fp2, or Fp2 and C4.

7. Conclusion

In this paper we go into the problem of knowledge discovery from streams of unlabeled data formed by continuous attributes in real-world environments to monitor the system using online maintenance and updating a population of association rules, which is commonly known as association stream mining. The association stream mining field presents the following main features: (1) the data arrives as a continuous high-rate information flow; (2) each data is processed only once: the data is received, processed and forgotten; (3) the concept to be learnt may change or drift over time and the algorithm should adapt; (4) no *a priori* structure or distribution is assumed; and (5) there are restrictions in memory usage.

This paper provides novelty in multiple ways: (1) new mechanisms which adapt membership functions and fuzzy partitions are proposed so that the algorithm might be endowed with more flexibility to fit the features of each variable; (2) a new methodology to evaluate and compare results from association stream mining algorithms and to analyze data stream is used; (3) a new problem in association stream mining with real application by helping psychophysiologists in knowledge discovery is addressed; and (4) complex real-world data streams are tackled.

Furthermore, the better behavior and performance of Fuzzy-CSar-AFP as compared with Fuzzy-CSar and Fuzzy-Apriori are evidenced in Section 6. In this section we show how the quality rules generated by Fuzzy-CSar-AFP are distributed in a more spacious way. This more spaced distribution means that the rules are more relevant as they represent different knowledge.

As further work, we continue working on future advanced versions of Fuzzy-CSar-AFP; e.g. reducing the number of configuration parameters by making them self-adaptive or improving its time efficiency even more. We plan to work with different real-world applications and other electroencephalogram problems.

Acknowledgements

We would like to express our gratitude to Pandelis Perakakis—CIMCYC (Research Center on Mind, Brain, and Behaviour) and the Department of Psychology, University of Granada (e-mail: peraka@ugr.es)—for sharing his experience and data with us.

This research has been supported by MINECO/FEDER under the Spanish National Research Project TIN2014-57251-P, and the BBVA Foundation project 75/2016 BigDaPTOOLS.

References

[1] J. Gama, Knowledge Discovery from Data Streams, CRC Press, 2010.
[2] M. Sayed-Mouchaweh, E. Lughofer, Learning in Non-Stationary Environments: Methods and Applications, Springer Science & Business Media, 2012.
[3] M. Scholz, R. Klinkenberg, Boosting classifiers for drifting concepts, Intell. Data Anal. 11 (1) (2007) 3–28.
[4] A. Orriols-Puig, J. Casillas, Fuzzy knowledge representation study for incremental learning in data streams and classification problems, Soft Comput. 15 (12) (2011) 2389–2414.
[5] S. Guha, A. Meyerson, N. Mishra, R. Motwani, L. O'Callaghan, Clustering data streams: theory and practice, IEEE Trans. Knowl. Data Eng. 15 (3) (2003) 515–528.
[6] L.L. Minku, X. Yao, DDD: a new ensemble approach for dealing with concept drift, IEEE Trans. Knowl. Data Eng. 24 (4) (2012) 619–633.
[7] W.N. Street, Y. Kim, A streaming ensemble algorithm (SEA) for large-scale classification, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2001, pp. 377–382.
[8] S. Wang, L.L. Minku, X. Yao, Resampling-based ensemble methods for online class imbalance learning, IEEE Trans. Knowl. Data Eng. 27 (5) (2015) 1356–1368.
[9] Y. Sun, K. Tang, L. Minku, S. Wang, X. Yao, Online ensemble learning of data streams with gradually evolved classes, IEEE Trans. Knowl. Data Eng. 28 (6) (2016) 1532–1545.
[10] B. Krawczyk, M. Wozniak, Weighted Naïve Bayes classifier with forgetting for drifting data streams, in: 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2015, pp. 2147–2152.
[11] B. Krawczyk, M. Wozniak, Reacting to different types of concept drift with adaptive and incremental one-class classifiers, in: 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF), IEEE, 2015, pp. 30–35.
[12] M. Moshtaghi, J.C. Bezdek, C. Leckie, S. Karunasekera, M. Palaniswami, Evolving fuzzy rules for anomaly detection in data streams, IEEE Trans. Fuzzy Syst. 23 (3) (2015) 688–700.
[13] A. Sancho-Asensio, A. Orriols-Puig, J. Casillas, Evolving association streams, Inf. Sci. 334 (2016) 250–272.
[14] T.-P. Hong, C.-S. Kuo, S.-C. Chi, Mining association rules from quantitative data, Intell. Data Anal. 3 (5) (1999) 363–376.
[15] P. Angelov, Autonomous Learning Systems: From Data Streams to Knowledge in Real-Time, John Wiley & Sons, 2012.
[16] M. Núñez, R. Fidalgo, R. Morales, Learning in environments with unknown dynamics: towards more robust concept learners, J. Mach. Learn. Res. 8 (2007) 2595–2628.
[17] E. Lughofer, P. Angelov, Handling drifts and shifts in on-line data streams with evolving fuzzy systems, Appl. Soft Comput. 11 (2) (2011) 2057–2068.
[18] Z. ur Rehman, M. Shahbaz, M. Shaheen, A. Guergachi, FPS-tree algorithm to find top-k closed itemsets in data streams, Arab. J. Sci. Eng. 40 (12) (2015) 3507–3521.
[19] M.M. Rashid, I. Gondal, J. Kamruzzaman, Mining associated patterns from wireless sensor networks, IEEE Trans. Comput. 64 (7) (2015) 1998–2011.
[20] A. Moustafa, B. Abuelnasr, M.S. Abougabal, Efficient mining fuzzy association rules from ubiquitous data streams, Alex. Eng. J. 54 (2) (2015) 163–174.
[21] H.-F. Li, M.-K. Shan, S.-Y. Lee, DSM-FI: an efficient algorithm for mining frequent itemsets in data streams, Knowl. Inf. Syst. 17 (1) (2008) 79–97.
[22] J. Cheng, Y. Ke, W. Ng, Maintaining frequent closed itemsets over a sliding window, J. Intell. Inf. Syst. 31 (3) (2008) 191–215.
[23] E.T. Wang, A.L. Chen, Mining frequent itemsets over distributed data streams by continuously maintaining a global synopsis, Data Min. Knowl. Discov. 23 (2) (2011) 252–299.
[24] Z. Farzanyar, M. Kangavari, N. Cercone, Max-FISM: mining (recently) maximal frequent itemsets over data streams using the sliding window model, Comput. Math. Appl. 64 (6) (2012) 1706–1718.
[25] R. Srikant, R. Agrawal, Mining quantitative association rules in large relational tables, SIGMOD Rec. 25 (2) (1996) 1–12.
[26] R.J. Miller, Y. Yang, Association rules over interval data, SIGMOD Rec. 26 (2) (1997) 452–461.
[27] K. Wang, S.H.W. Tay, B. Liu, Interestingness-based interval merger for numeric association rules, in: KDD, 1998, pp. 121–128.
[28] J. Mata, J.-L. Alvarez, J.-C. Riquelme, Discovering numeric association rules via evolutionary algorithm, in: Advances in Knowledge Discovery and Data Mining, Springer, 2002, pp. 40–51.
[29] A. Orriols-Puig, J. Casillas, E. Bernadó-Mansilla, First approach toward on-line evolution of association rules with learning classifier systems, in: Proceedings of the 10th Annual Conference Companion on Genetic and Evolutionary Computation, ACM, 2008, pp. 2031–2038.
[30] T.-P. Hong, C.-S. Kuo, S.-C. Chi, Trade-off between computation time and number of rules for fuzzy mining from quantitative data, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 9 (5) (2001) 587–604.
[31] D. Dubois, E. Hüllermeier, H. Prade, A systematic approach to the assessment of fuzzy association rules, Data Min. Knowl. Discov. 13 (2) (2006) 167–192.
[32] C.K.-S. Leung, B. Hao, Mining of frequent itemsets from streams of uncertain data, in: IEEE 25th International Conference on Data Engineering, 2009, ICDE'09, IEEE, 2009, pp. 1663–1670.
[33] P. Chen, H. Su, L. Guo, Y. Qu, Mining fuzzy association rules in data streams, in: 2010 2nd International Conference on Computer Engineering and Technology, vol. 4, ICCET, IEEE, 2010, pp. V4–153.
[34] B. Chandra, S. Bhaskar, A novel approach of finding frequent itemsets in high speed data streams, in: 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), vol. 1, IEEE, 2011, pp. 40–44.
[35] H. Ishibuchi, T. Yamamoto, Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining, Fuzzy Sets Syst. 141 (1) (2004) 59–88.
[36] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, SIGMOD Rec. 22 (2) (1993) 207–216.
[37] B. Welford, Note on a method for calculating corrected sums of squares and products, Technometrics 4 (3) (1962) 419–420.
[38] M.W. Voss, C. Vivar, A.F. Kramer, H. van Praag, Bridging animal and human models of exercise-induced brain plasticity, Trends Cogn. Sci. 17 (10) (2013) 525–544.
[39] S. Vaynman, F. Gomez-Pinilla, Revenge of the "sit": how lifestyle impacts neuronal and cognitive health through molecular systems that interface energy metabolism with neuronal plasticity, J. Neurosci. Res. 84 (4) (2006) 699–715.
[40] S. Yamada, T. Funayama, Y. Yamamoto, Visualization of relations of stores by using association rule mining, in: 2015 13th International Conference on ICT and Knowledge Engineering (ICT & Knowledge Engineering 2015), IEEE, 2015, pp. 11–14.
[41] D.G. Trevisan, N. Sanchez-Pi, L. Marti, A.C.B. Garcia, Big data visualization for occupational health and security problem in oil and gas industry, in: Human Interface and the Management of Information, Information and Knowledge Design, Springer, 2015, pp. 46–54.
[42] J.J. Meulman, The integration of multidimensional scaling and multivariate analysis with optimal transformations, Psychometrika 57 (4) (1992) 539–565.
[43] R.P. Bartsch, K.K. Liu, A. Bashan, P.C. Ivanov, Network physiology: how organ systems dynamically interact, PLoS ONE 10 (11) (2015) e0142143.