Fuzzy Optimization and Decision Making (2021) 20:429–449
https://doi.org/10.1007/s10700-021-09350-3

Incremental maintenance of discovered fuzzy association rules

A. Pérez-Alonso[1] · I. J. Blanco[2] · J. M. Serrano[3] · L. M. González-González[4]

Accepted: 29 January 2021 / Published online: 31 March 2021

Abstract

Fuzzy association rules (FARs) are a recognized model to study existing relations among data, commonly stored in data repositories. In real-world applications, transactions are continuously processed with upcoming new data, rendering the discovered rules information inexact or obsolete in a short time. Incremental mining methods arise to avoid re-runs of those algorithms from scratch by re-using information that is systematically maintained. These methods are useful for extracting knowledge in dynamic environments. However, executing the algorithms only to maintain previously discovered information creates inefficiencies in real-time decision support systems. In this paper, two active algorithms are proposed for incremental maintenance of previously discovered FARs, inspired by efficient methods for change computation. The application of a generic form of measures in these algorithms allows the maintenance of a wide number of metrics simultaneously. We also propose to compute data operations in real-time, in order to create a reduced relevant instance set. The algorithms presented do not discover new knowledge; they are just created to efficiently maintain valuable information previously extracted, ready for decision making. Experimental results on education data and repository data sets show that our methods achieve a good

✉ A. Pérez-Alonso
alain.perez@usm.cl

I. J. Blanco
iblanco@ugr.es

J. M. Serrano
jschica@ujaen.es

L. M. González-González
luisagon@uclv.edu.cu

1 Department of Electronics and Informatics, Universidad Técnica Federico Santa María, 4030000 Concepción, Chile

2 Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain

3 Department of Computer Science, University of Jaén, 23071 Jaén, Spain

4 Department of Computer Science, University "Marta Abreu" of Las Villas, 54830 Santa Clara, Cuba

performance. In fact, they can significantly improve traditional mining, incremental mining, and a naïve approach.

Keywords Fuzzy association rules · Incremental maintenance · Real-time decision support systems · Active databases

1 Introduction

Association rules are one of the best studied models for knowledge discovery in the data-mining research field. They represent associations or dependencies among attributes' values in a data repository (Agrawal et al. 1993). Finding association rules from quantitative attributes introduce several problems such as the increase of algorithm complexity (Delgado et al. 2003). Fuzzy Association Rules (FARs) present a model that certainly helps to solve this problem by mapping crisp data to fuzzy data, in order to reduce the granularity (Delgado et al. 2003). This reduction, by means of sets of linguistic labels, also improves the semantic content of the rules, becoming more comprehensible for humans (Delgado et al. 2014).

Usually, fuzzy association rule mining algorithms are run for large portions of data, resulting in a very expensive process for traditional methods. The knowledge discovered by those algorithms is specific for the current stage of the repository in which they were run. In real-world applications, data is not static because new information is commonly introduced, and old one is deleted or modified. These continuous changes can render the measures of rules inexact and eventually obsolete. This becomes a problem if up-to-date measures are needed just-in-time by Real-Time Decision Support Systems (RTDSS) (Sauter 2014). Example applications can be found in the field of data streams like web click stream data, sensor network data, and network traffic data (Tan et al. 2010; Lee and Guu 2003). Recently, emerging research field in big data offers similar issues in association with velocity and volume (Wu et al. 2014).

At this time, many research efforts are being made to improve the performance of FARs-mining algorithms (Hong and Lin 2010; Papadimitriou and Mavroudi 2005). These efforts try to reduce the problem of updating FARs to find the new set of fuzzy large itemsets and share an intermediate maintenance form (fuzzy frequent itemsets) for this goal. In this work, the update problem is oriented to directly maintain the measures of previously discovered rules, covering the decision-makers' needs for just-in-time data information (Sauter 2014). This approach can also be helpful to refine rules discovery at post-mining stage (Boettcher et al. 2009). Our perspective is neither to remove previously discovered rules nor to add new ones, but to efficiently updates initial mined rules measures allowing real-time decision-making. A splitted form of rule is defined, enabling rules direct maintenance in a wide range of metrics (Greco et al. 2012; Lenca et al. 2008), and simultaneously, maintaining this metrics in an efficient way. We handle the maintenance problem from a change computation point of view. The process of change computation deals with modifications induced by data operations; it is an important field in active database systems (Urpí and Olivé 1994). Additionally, the materialized view maintenance and the integrity constraint checking are significant fields of active systems that provide multiple methods (Cabot

and Teniente 2005; Jain and Gosain 2012). Two algorithms, inspired in these methods, are proposed in this paper. One of them is intended for rules immediate maintenance and the other one for rules deferred maintenance. Both algorithms reuse previous results incrementally to avoid measures calculations from scratch. So far, there is no other method to maintain FARs from this perspective.

Experimental results were obtained from active relational databases with real educational data and repository datasets. They show that the proposed algorithms achieve good performance and improve classical mining, incremental mining and a naïve approach significantly. The proposed algorithms have been implemented and compared in two of the most-used open source database management systems.

The main contribution of this paper is twofold. First, we propose two efficient algorithms to maintain the previously discovered rules, a very well addressed research from Cheung et al. (1996). Second, from a deferred perspective, we propose to consider the interactions between data operations in real-time, in order to create a reduced relevant instance set. A common characteristic of the proposed algorithms is the efficient maintenance of existing rules, keeping their up-to-date measures available. It is made without having access to the database itself, making our approaches self-maintainable (Colby et al. 1996; Jain and Gosain 2012).

The remainder of this paper is organized as follows. First, Sect. 2 defines basic concepts and describes the current research problem. Section 3 presents a change computation scope with integrity constraints and materialized views for FARs maintenance. Section 4 describes the proposed algorithms, including an initial naïve approach. In Sect. 5 we briefly review related work and compare it with our approach. Section 6 presents the experimental results of the reviewed and proposed methods for the performance evaluation. Finally, Sect. 7 concludes this paper summarizing the results from our work.

2 Preliminary concepts and problem statement

This section defines preliminary concepts used throughout this paper and describes the research problem. A measure indicator for fuzzy rules interestingness and a relational database context for fuzzy association rules are presented too.

2.1 Association rules definition

Association Rules (ARs) can formally be represented as implications of itemsets (sets of items) in transactional databases (Agrawal et al. 1993). Let $It = \{it_1, it_2, \ldots, it_m\}$ be a non-empty set of m distinct items. Let T be the transaction scheme that contains a set of items such as $It \subseteq T$. An AR is an implication of the form $X \Rightarrow Y$ where $X, Y \subset It$ such that $X \neq \emptyset$, $Y \neq \emptyset$ and $X \cap Y = \emptyset$. In this statement X and Y are called rule itemsets and they are the antecedent and consequent of the rule, respectively.

There are two important classical parameters to measure using the association rules: support and confidence. Support is defined as the fraction of records that contain $X \cup$

Y in all records. Confidence is the fraction of the transactions that contains $X \cup Y$ in records that contain X.

Alternative metrics are very well established (Lenca et al. 2008) and they solve some drawbacks associated with the original indicators. An example is the certainty factor (CF) (Berzal et al. 2002) shown in Eq. (1) that is a confidence alternative. The certainty factor takes values in $[-1, 1]$. It is positive when the dependence between X and Y is positive, 0 when they are independent, and a negative value represents negative dependence.

$$CF(X \Rightarrow Y) = \begin{cases} \dfrac{Conf(X \Rightarrow Y) - Supp(Y)}{1 - Supp(Y)}, & Conf(X \Rightarrow Y) > Supp(Y) \\ \dfrac{Conf(X \Rightarrow Y) - Supp(Y)}{Supp(Y)}, & Conf(X \Rightarrow Y) < Supp(Y) \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Association rules were first studied in market basket data, where each basket is a transaction containing the set of items bought by a client. In a relational database context, it is usual to consider that items are pairs $<attribute, value>$ and transactions are tuples in a relation. For example, the item $<Z, z_0>$ belongs to a transaction associated to a tuple t if $t[Z] = z_0$.

A typical issue arising from this context is the granularity problem (Delgado et al. 2003). Attributes described with high granularity (frequently on attributes with real domain) provide a large number of items. This granularity increases exponentially the complexity of the search and can be solved by clustering the value of the domain. A set of clusters is then considered to be the new domain of the attribute. In this scenario, the set of items associated to the attribute is the set of pairs $<attribute, cluster>$. The attributes with numerical domains are called quantitative, and the task of finding rules that relate items on the form $<attribute, interval>$ to other items is called the quantitative association rules problem.

Clustering solutions have some inherent problems. One of such problems is related to the meaning of the clusters because, in many situations, concepts are imprecise and cannot be suited by intervals. Another problem is related to the boundaries. The boundaries between every pair of consecutive intervals should not be sharp; it is not clear that a border value of one interval does not belong to the next interval. This has much in common with the problem of small variation in the boundaries.

2.2 Fuzzy association rules definition

The fuzzy-set theory (Zadeh 1965) provides an important tool to solve the problems just discussed before. Performing a fuzzy clustering of the domains allows to obtain good representations of imprecise concepts. In this approach, items are considered as $<attribute, label>$ where $label$ has an internal representation as a fuzzy set over the domain of the $attribute$. These items are called fuzzy items, and rules that associate them are called fuzzy association rules.

In the framework presented by Delgado et al. (2003), the model for association rules is extended in order to manage fuzzy values in databases. The approach is based on the definition of fuzzy transactions as fuzzy subsets of items. Let It be a set of items and T' be a set of fuzzy transactions, where each fuzzy transaction is a fuzzy subset of It. Let $\tilde{\tau} \in T'$ be a fuzzy transaction, $\tilde{\tau}(it_k)$ is the membership degree of item it_k in $\tilde{\tau}$. We will note that $\tilde{\tau}(It_0)$ is the degree of inclusion of $It_0 \subseteq It$ in $\tilde{\tau}$, defined as

$$\tilde{\tau}(It_0) = \min_{i \in It_0} \tilde{\tau}(i) \tag{2}$$

It follows that the set of transactions where a given item appears is a fuzzy set.

A fuzzy association rule is an implication of the form $A \Rightarrow C$ such that $A, C \subseteq It$, $A \cap C = \emptyset$ and $A, C \neq \emptyset$. In this statement A and C are two crisp subsets called antecedent and consequent of the rule, respectively.

In order to measure the interest and accuracy of a fuzzy association rule, we must be used approximate tools. In Delgado et al. (2003), a semantic approach is proposed based upon the evaluation of quantified sentences (see Zadeh 1983). Let U be the fuzzy coherent quantifier $U_M(x) = x$:

- The support of an itemset $\tilde{\Gamma}_{It_0}$ is equal to the result of evaluating the quantified sentence "U of T' are $\tilde{\Gamma}_{It_0}$", where $\tilde{\Gamma}_{It_0}$ is a fuzzy set on T' defined as

$$\tilde{\Gamma}_{It_0}(\tilde{\tau}) = \tilde{\tau}(It_0) \tag{3}$$

- The support of the fuzzy association rule $A \Rightarrow C$ in the FT-set T', $Supp(A \Rightarrow C)$, is the evaluation of the quantified sentence "U of T' are $\tilde{\Gamma}_{A \cup C}$" = "$U$ of T' are $(\tilde{\Gamma}_A \cap \tilde{\Gamma}_C)$".
- The confidence of the fuzzy association rule $A \Rightarrow C$ in the FT-set T', $Conf(A \Rightarrow C)$, is the evaluation of the quantified sentence "U of $\tilde{\Gamma}_A$ are $\tilde{\Gamma}_C$".

As seen in Delgado et al. (2003), the proposed method is a generalization of the ordinary association rule assessment framework in the crisp case.

In relational databases, let $R = \{At_1, \ldots, At_m\}$ be a set of attributes, and let $Lab(At_k)$=$\{L_1^{At_k}, \ldots, L_n^{At_k}\}$ be the set of linguistic labels defined over $Dom(At_k)$ $\forall At_k \in R$, where $L_j^{At_k} : Dom(At_k) \rightarrow [0, 1]$. To our purposes, we can represent a given item as a pair $<At_k, L_j^{At_k}>$. Every instance r of R is associated to the FT-set $T_L'^r$ and each tuple $t \in r$ is associated to a unique fuzzy transaction $\tilde{\tau}_L^t \in T_L'^r$ such that $\tilde{\tau}_L^t(< At_k, L_j^{At_k} >) = L_j^{At_k}(t[At_k])$.

2.3 Fuzzy association rules maintenance problem

Real-world systems must handle new and old information coming from the universe of discourse. In this process, the information can be affected by many data operations. A common approach in active databases known as event-condition-action rules defines these data operations at event occurrences that can be primitive or composite. Primitive type, called the primitive structural event (PSE), is a single low-level event. A

composite type is a combination of multiple primitive or composite structural events (CSE). Those events produce a database transition (DB_{s0}, DB_{s1}) where DB_{s0} is the initial state and DB_{s1} is the final state.

Three types of PSE are considered (also called data operations): inserting a tuple Δt^{+}, deleting a tuple Δt^{-} and updating a tuple Δt^{-+}. The inserted and deleted tuples are denoted by t^{+} and t^{-} respectively. An update event can be seen like an independent event in which t_0^{-+} is the tuple before the update event corresponding to DB_{s0} state, and t_1^{-+} the tuple after update in the DB_{s1} state.

Let us consider an initial database state DB_i and a composite structural event over this state that produces a transition (DB_i, DB_f). Let $FRS_i = \{(FAR_1, FRM_1^i), (FAR_2, FRM_2^i), \ldots, (FAR_n, FRM_n^i)\}$ be a set of mined FARs and their measure value, discovered in DB_i state. This paper is focused on efficiently finding the new measures of previously discovered fuzzy association rules in the final state DB_f. In other words, the fuzzy association rules maintenance problem can be reduced to finding a fuzzy rule set $FRS_f = \{(FAR_1, FRM_1^f), (FAR_2, FRM_2^f), \ldots, (FAR_n, FRM_n^f)\}$.

3 Fuzzy association rules maintenance under the change computation scope

Change computation is the capability to compute data operations efficiently (Urpí and Olivé 1992, 1994). Their methods have been accepted in active databases, materialized views maintenance and integrity constraint checking (Urpí and Olivé 1994). All these methods share similar characteristics like definition of modifications to be monitored, computation of the changes and reaction to defined changes. Materialized views, integrity constraints, and FARs share the capacity of reflecting data information, but for different purposes. In this section, we present FARs under the change computation scope allowing and formalizing the use of those methods in fuzzy association rules maintenance problem.

3.1 Fuzzy association rules and materialized views

Views define derived data, which can be materialized in database systems. The process of keeping these views up-to-date in database transitions is called materialized view maintenance (Colby et al. 1996; Jain and Gosain 2012). Let a view V be defined by query Q and materialized in MV. Any correct materialization of V in a database state DB_s must return the same data as Q: $MV(DB_s) = Q(DB_s)$. Specifically, the materialization of V must be equivalent to its querying (Colby et al. 1996): $MV \equiv Q$. An important aspect to materialize a view is the speed of its querying, a desirable quality when the response time is critical. If tm_1 is the time for retrieving $MV(DB_s)$ and tm_2 the time for retrieving $Q(DB_s)$ then, in general terms, $tm_1 \ll tm_2$.

Just like a view FARs define some data information, but in a particular way because they expose attributes correlated in an implication form. A fuzzy association rule of the form $A \Rightarrow C$ establishes, with some measure, that when A occurs so does C. Items in such rule are represented by linguistic labels as a fuzzy set over the

domain of the attribute. An example query Q of such rule is a query that obtain the cardinality of the fuzzy set according to Eq. (3). Such query can be represented using the Structured Query Language (SQL) in a relation scheme R, as follows in query (4). Here, $\{At_1, \ldots, At_j\} \subseteq R$ is the set of attributes involved in $A \cup C$.

$$|\tilde{\Gamma}_{A\cup C}| = SELECT\ sum(least(L_1^{At_1}(At_1), \ldots, L_{n_1}^{At_1}(At_1), \ldots, L_1^{At_j}(At_j), \ldots, L_{n_j}^{At_j}(At_j)))$$
$$FROM\ R; \tag{4}$$

By materializing similar queries through aggregate operators, FARs measures can be directly maintained on the fuzzy rule base.

3.2 Fuzzy association rules and integrity constraints

Integrity is a mandatory property for a relational database, and it is associated with two components: validity and completeness. Validity represents the truth of data, completeness represents the totality of relevant data, and integrity constraints are conditions that guarantee its satisfaction at any time.

Each state of a database must satisfy all integrity constraints. Let ICS be a set of integrity constraints in denial form and F a boolean function that evaluates if any constraint is violated or not in a database state. A consistent database state of DB_s guarantees that all integrity constraints in ICS evaluated with F must return false: $\forall ic \in ICS\ (F(ic, DB_s) = false)$. A correct database transition must have a final consistent state. Integrity constraints checking methods are aimed at holding the database integrity.

FARs only represent data knowledge and never deny any database state. Nevertheless, they have an inherent restriction: they measure thresholds. The measure evaluates the level of interest in the rules and also establishes a minimum acceptable value that defines FARs existence. Let FRS_s be a set of mined FARs in a database state DB_s, G a function that evaluates the measure of a FAR and δ the minimum threshold. A correct FRS set guarantees that all FAR evaluated with G must return a value equal to or greater than δ: $\forall far \in FRS\ (G(far, DB_s) \geq \delta)$. The FARs incremental maintenance goal is not to maintain the measure threshold, but it can be part of the efficient evaluation of G.

4 Fuzzy association rules maintenance proposals

Many research activities propose measures of rules quality with different properties, and their number is overwhelming (Greco et al. 2012; Lenca et al. 2008). Existing measures are usually defined by counting a total number of records that satisfy some condition. These conditions are generally associated with the antecedent, consequent, rule examples and counterexamples among others (Greco et al. 2012; Lenca et al. 2008).

In our proposals, a fuzzy measure is considered the fuzzy version for measures associated to a fuzzy association rule and is defined as a set of k distinct fuzzy measure-parts $FMP = \{FMp_1, FMp_2, \ldots, FMp_k\}$. Each item of this set represents a different part

of the measure formula. Fuzzy measure-parts must be atomic, it means that they cannot be divided into smaller items and still bring the same measure value. For example, confidence can be split into two parts: sum of antecedent and sum of (antecedent ∪ consequent). On the other hand, the certainty factor shown in Eq. (1) needs three parts: sum of antecedent, sum of consequent and sum of (antecedent ∪ consequent). In this way, it is possible to efficiently maintain several metrics at the same time because metrics shares some fuzzy measure-parts. For example, following Lenca et al. (2008) it is possible with only five distinct measure-parts to maintain 20 interestingness measures simultaneously. The final measure of a fuzzy association rule is a formula over *FMP* parts.

Three methods are considered for direct rule maintenance. The first one consists in a naïve approach. This initial strategy is later enhanced from a change computation perspective in an immediate and a deferred way thus becoming the second and the third methods. The improvements are oriented towards two essential points: rules to be maintained and data instances to be analyzed.

4.1 Naïve approach

One way in which a naïve strategy may arise is via database queries. By this form, each item of *FMP* is obtained following query (4) as a query over all data: FMp_1=Qp_1, FMp_2=Qp_2, . . ., FMp_k=Qp_k. As a result, previous information of fuzzy measure-parts is not used to recalculate new values. Fuzzy rule base is updated from scratch after each primitive or composite structural event takes place.

These queries are *sum* aggregate queries. Each one represents the *sum* of the involved fuzzy item. Fuzzy measure-parts take real positive values FMp_1, FMp_2, …, $FMp_k \in \mathbb{R}^+$.

The naïve approach involves maintaining all rules after each data operation and querying the entire data. Additionally, many irrelevant instances are analyzed, which results in a waste of time. The simplicity of implementation and avoiding extra manipulation as a reaction to database operations are, in fact, the only attractive aspects of the naïve approach. However, this is irrelevant when the stored data is large and *sum* queries become highly inefficient.

4.2 Immediate incremental maintenance method

An immediate approach is oriented to updates the fuzzy rule base immediately after the event takes place, in an active fashion. This approach verifies the specific rules that must be updated and it computes only the changes made by a primitive structural event. It means that just one record can be checked at a time.

Incremental view maintenance algorithms offer multiple solutions according to query operators. Specifically, a counting algorithm for view maintenance (Gupta and Mumick 1995) provides an interesting perspective for maintaining FARs. The main difference is that we sum the membership degree of attributes in linguistic labels. Algorithm 1 presents the proposed immediate incremental maintenance where a fuzzy rule measure is updated for data operations.

Algorithm 1: Immediate incremental maintenance for an FAR.

Input: A composite structural event *CSE* that modifies the attributes related in list *MAt*, fuzzy measure-parts *FMP* of $X \Rightarrow Y$, and the set of attribute linguistic label L in FAR.
Output: An updated fuzzy measure-parts *FRM*.
Method:
foreach $PSE \in CSE$ **do**
  if $(PSE = \Delta t^{-+})$ **then** /* update event */
    if $(MAt \cap \{X \cup Y\} \neq \emptyset)$ **then** /* FAR condition */
      forall FMp_j of *FMP* **do**
        if $(MAt \cap \{\textit{involved attributes in } FMp_j\} \neq \emptyset)$ **then**
          update FMp_j, increment with L_j of t_1^{-+};
          update FMp_j, decrement with L_j of t_0^{-+};
        end
      end
  else if $(PSE = \Delta t^{+})$ **then** /* insert event */
    forall FMp_j of *FMP* **do**
      update FMp_j, increment with L_j of t^{+};
    end
  else if $(PSE = \Delta t^{-})$ **then** /* delete event */
    forall FMp_j of *FMP* **do**
      update FMp_i, decrement with L_j of t^{-};
    end
  end
end

Fuzzy rules must be checked in updates events. It is only when the values of the attributes are changed or their unknown value changes, i.e., we are interested in a set *MAt* of attributes such that $MAt = \{At \in R | t_0^{-+}[At] \neq t_1^{-+}[At]\}$. In an insert or delete event, all schema attributes are affected and always change measures of rules (except when they have unknown status). Not all fuzzy measure-parts of *FMP* need to be recalculated either. For example, if an update operation modifies only the antecedent attributes of a rule, then it is not necessary in certainty factor measure (1) to recalculate the sum part of the consequent.

4.3 Deferred incremental maintenance method

A deferred approach efficiently maintains a fuzzy rule base up-to-date but not for each data operation like immediate approach. This method computes modified instances in a data transition and updates fuzzy rule base for these relevant instances. Principal differences of immediate and deferred maintenance approaches are illustrated in Fig. 1.

Typically, fuzzy incremental mining algorithms consider different types of operations but do not consider the interactions between such operations (Hong et al. 2012; Lin et al. 2014). We propose an algorithm specifically for these interactions; in the best scenario it reduces the number of operations significantly and in the worst case, it only maintains the same original number.

The deferred problem in our proposal is divided into two subproblems. The first subproblem consists of computing the relevant instances affected in a database transition.

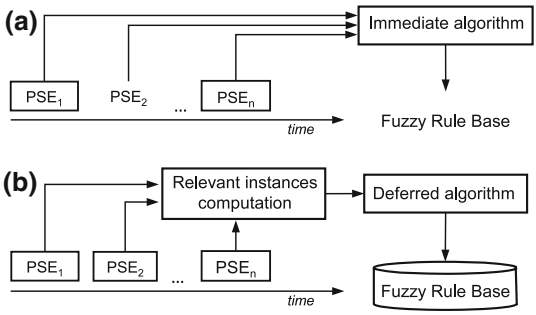

Fig. 1 Immediate incremental maintenance approach (**a**), and deferred incremental maintenance approach (**b**)

In this step, a relevant operation set at real-time is built, after each primitive structural event takes place. A different approach would be to scan the original operation set to reduce their number. The second one is related to incrementally updating the fuzzy rule base with those relevant instances. Integrity constraint checking satisfactorily handles these subproblems in its field (Cabot and Teniente 2005).

Relevant instances computation in a transition must consider the relationships among primitive structural events. These interactions are controlled by net effect policy (Cabot and Teniente 2005). For example, if a tuple is inserted and deleted in the same database transition, then these events do not cause any variation on the final database state and its measures of rules. We consider the following policies for structural event interactions in which each tuple is identified by its primary key value:

- If a tuple is inserted and later deleted, then it does not register.
- If a tuple is inserted and later updated, then it registers as inserted.
- If a tuple is several times updated, then it registers as one update.
- If a tuple is updated and later deleted, then it registers as deleted.
- If a tuple is deleted and later inserted, then it registers as updated.

Usually, non-modeled update events such as deletions followed by insertions are registered by an auxiliary relation (Cabot and Teniente 2005). In this deferred approach a different consideration is followed: each database relation related to any rule has only two auxiliary relations. These auxiliary relations register the insert, update and delete events. Their relation schemas are copies with different names of the base relation scheme, and they must store the newest and oldest record values. Insert and delete auxiliary relations store $t^+ \cup t_1^{-+}$ and $t^- \cup t_0^{-+}$ tuples respectively, according to net effect considerations. These structural event interactions are applied over relations at real-time by the active Algorithm 2.

This active process adds a minimum activity over regular data operations, just the necessary ones to store relevant instances and to apply net effect policy. The behavior of the algorithm is similar when considering only insertion and deletion events, but note the benefits of using update occurrences when a record is already inserted or modified.

The fuzzy rule base is updated only with these instances, by incrementing previous measures information. These updates could be made automatically on fuzzy rules base access or scheduled. Algorithm 3 is presented in order to update an *FMP*.

Algorithm 2: Compute relevant instances that may modify FARs.

Input: A composite structural event *CSE*, I and D their auxiliary relations of base relation.
Output: Auxiliary relations I and D updated for a *CSE*.
Method:
foreach $PSE \in CSE$ **do**
  if ($PSE = \Delta t^{-+}$) **then** /* update event */
    if ($\{u \in I \mid u = t_0^{-+}\} = \emptyset$) **then**
      insert into I values t_1^{-+};
      insert into D values t_0^{-+};
    else
      update $u \in I$ set t_1^{-+} where $u = t_0^{-+}$;
    end
  else if ($PSE = \Delta t^{+}$) **then** /* insert event */
    insert into I values t^{+};
  else if ($PSE = \Delta t^{-}$) **then** /* delete event */
    if ($\{u \in I \mid u = t_0^{-}\} = \emptyset$) **then** insert into D values t^{-};
    else delete $u \in I$ where $u = t^{-}$;
  end
end

Algorithm 3: FARs deferred actualization for relevant instances.

Input: I, D auxiliary relations of Algorithm 2 output, fuzzy measure-parts *FMP*, and the set of attribute linguistic labels L in FAR.
Output: An updated fuzzy measure-parts *FMP*.
Method:
forall FMp_j of *FMP* **do**
  update FMp_j, increment with L_j of I;
  update FMp_j, decrement with L_j of D;
end
truncate I;
truncate D;

Note that all tuples in auxiliary relation I implies an increment in the fuzzy measure value and all tuples in auxiliary relation D implies a decrement. The increment/decrement with L_j is related to the use of linguistic labels in query (4), but instead of access to the entire base relation, access is only needed to the auxiliary relation I/D. The fuzzy rule base refreshing without accessing the base relation is a special feature in a large amount of data. Moreover, it entails the benefits of having only two auxiliary relations instead of more. Heuristically, the time for querying auxiliary relations is still much lower than querying the whole data like naïve approach. In rare occasions, this is not the case. For example, if the entire database is deleted, then querying it is very fast and querying auxiliary relation D is highly inefficient.

Table 1 Age and salary of six people (r_0: R)

ID	Age	Salary
1	26	900
2	52	2200
3	37	1400
4	65	2700
5	70	3000
6	21	500

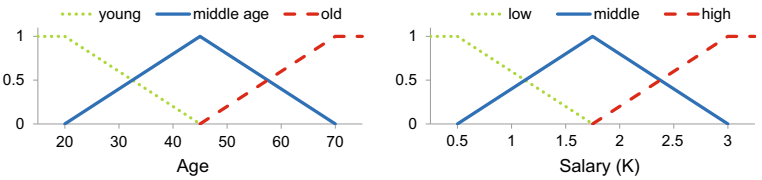

Fig. 2 Fuzzy labels for Age attribute (left) and Salary attribute (right) of Table 1 relation

Table 2 Fuzzy transactions for linguistic labels of Table 1 relation

	$\tilde{\tau}_L^{t1}$	$\tilde{\tau}_L^{t2}$	$\tilde{\tau}_L^{t3}$	$\tilde{\tau}_L^{t4}$	$\tilde{\tau}_L^{t5}$	$\tilde{\tau}_L^{t6}$
<Age,young>	0.76	0	0.32	0	0	0.96
<Age,middle age>	0.24	0.72	0.68	0.2	0	0.04
<Age,old>	0	0.28	0	0.8	1	0
<Salary,low>	0.68	0	0.28	0	0	1
<Salary,middle>	0.32	0.64	0.72	0.24	0	0
<Salary,high>	0	0.36	0	0.76	1	0

4.4 Algorithms examples

To illustrate proposed algorithms behavior, consider the age and salary information about six people in the relation shown in Table 1. This relation and the structural events that modified it share immediate and deferred algorithm examples.

In order to obtain more semantic information about attributes and to diminish the granularity involved in their domain, these people attributes have been fuzzified using the triangular membership functions shown in Fig. 2. For the age attribute, three labels are defined in their domain and three labels for salary attribute domain too. Labels are: Lab(*Age*)={*young*,*middle age*,*old*} and Lab(*Salary*)={*low*,*middle*,*high*}.

Each tuple $t \in r_0$ is associated with a unique fuzzy transaction. In this case, a fuzzy transaction can contain more than one item corresponding to different labels of the same attribute, because it is possible for a single value in the table to match more than one label to a certain degree. In our example, Table 2 contains all fuzzy transactions for defined linguistic labels.

Three structural events modifying the relation r_0 show the behavior of proposed algorithms in a common way. For all primitive structural events, the FAR <Age,old> $\Rightarrow$ <Salary,high> is incrementally maintained. Such maintenance is made through

Fig. 3 Immediate algorithm example for FAR

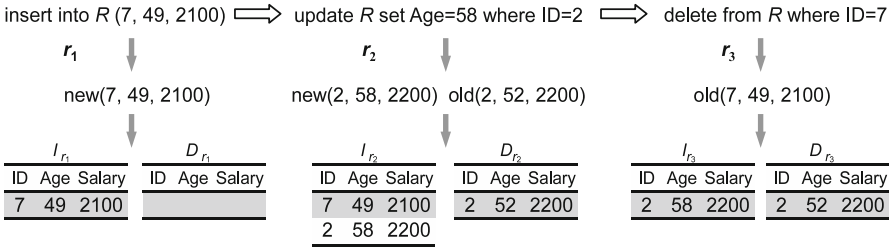

Fig. 4 Example of compute relevant instances that may modify a FAR

four fuzzy measure-parts: sum of antecedent, sum of consequent, sum of (antecedent ∪ consequent) and count of records. Those fuzzy measure-parts are able to simultaneously maintain the certainty factor (1) and lift quality measures (Lenca et al. 2008). An example of this variations on *FMP* parts after each PSE takes place is shown on Fig. 3. Each *FMP* part is adjusted according to Eq. (3), with the cardinalities of the fuzzy sets on Table 2:

$$|\tilde{\Gamma}^{r_0}_{\{<Age,old>\}}| = |\{0.28/\tilde{\tau}^{t_2}_L + 0.8/\tilde{\tau}^{t_4}_L + 1/\tilde{\tau}^{t_5}_L\}| = 2.08$$
$$|\tilde{\Gamma}^{r_0}_{\{<Salary,high>\}}| = |\{0.36/\tilde{\tau}^{t_2}_L + 0.76/\tilde{\tau}^{t_4}_L + 1/\tilde{\tau}^{t_5}_L\}| = 2.12$$
$$|\tilde{\Gamma}^{r_0}_{\{<Age,old>,<Salary,high>\}}| = |\{0.28/\tilde{\tau}^{t_2}_L + 0.76/\tilde{\tau}^{t_4}_L + 1/\tilde{\tau}^{t_5}_L\}| = 2.04$$

In Figs. 4 and 5 the same FAR maintenance is presented but in a deferred fashion. For the first subproblem of this proposal, illustrated in Fig. 4, the transitions of auxiliary relations in each r_i relation are available. This step computes the relevant instances that may modify FAR_1 according to net effect policies. Specifically, in this example the original three structural events are reduced to two relevant instances.

The FAR_1 rule is updated only with these instances in a second step, by incrementing previous rules information. To illustrate our deferred proposal for FARs maintenance let us look at Fig. 5.

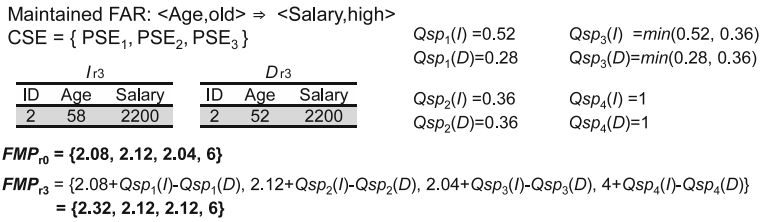

Fig. 5 Example of FMP deferred actualization for Fig. 4 relevant instances

5 Related work and comparison with our approach

Recently, the fuzzy-set theory (Zadeh 1965) has been used more and more frequently in the data mining field (Cadenas and Verdegay 2009). This theory solves typical crisp mining issues like the granularity problem (Delgado et al. 2003). Several fuzzy mining algorithms based on the Apriori algorithm (Agrawal et al. 1993) have been proposed at this time (Hong et al. 2001). These Apriori-like algorithms generate candidate fuzzy itemsets level-by-level, which might cause multiple scans of the database and high computational costs. In order to avoid re-scanning the whole data and breaking Apriori bottlenecks, many algorithms have been proposed by using fuzzy tree-structures like the fuzzy frequent-pattern tree (fuzzy FP-tree) structure (Lin et al. 2010; Papadimitriou and Mavroudi 2005). The fuzzy FP-tree is used to compress a database into a tree structure which stores only large fuzzy items. A very well studied fuzzy FP-tree is the multiple fuzzy-term FP (MFFP) tree (Hong and Lin 2010). After the fuzzy FP-tree is constructed, the desired fuzzy frequent itemsets can be derived by the corresponding MFFP-growth algorithm (Hong and Lin 2010).

In real-world applications, data repositories are not static. Generally, data will increase with time. Traditional batch mining algorithms solve this problem by re-scanning the whole data when new transactions are inserted, deleted or modified. This is clearly inefficient because all previous mined information is wasted. The incremental mining research field defines this issue as an update problem and reduces it to find the new set of fuzzy large itemsets incrementally. Extended fuzzy FP-tree algorithms are being designed to efficiently handle this problem like the incremental multiple fuzzy frequent pattern tree (incMFFP-tree) (Hong et al. 2012). The incMFFP-tree structure specifically handles newly inserted transactions. These incremental proposals improve the MFFP structure in different ways but maintain the execution of similar fuzzy FP-growth algorithms in a second step.

Incremental mining techniques can be very useful not only to update all the rules, but also to maintain indirectly the metrics of a set of interesting rules. Unlike incremental mining methods, we handle the update problem by maintaining the measures of previously discovered fuzzy association rules, as an extension of our work in association rules and approximate dependencies (Pérez-Alonso et al. 2017a, b). That does not lead to maintain fuzzy itemsets information, instead, existed rules measures are directly updated in an incremental way. After the process of rules extraction, we keep only the discovered FARs and do not remove any rule or add new ones, allowing to avoid the still expensive incremental mining algorithm. In Fig. 6 three scenarios

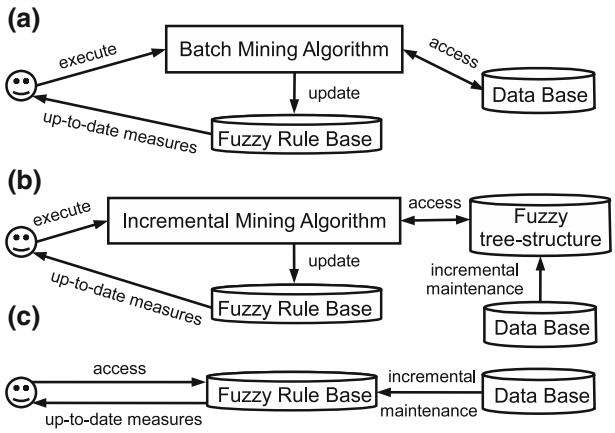

Fig. 6 Batch mining method (**a**), incremental mining method (**b**), and incremental maintenance proposal (**c**) for up-to-date measures

illustrate when a system decision-makers needs the measures of previously discovered FARs just-in-time. That includes the batch mining method, the incremental mining method, and our proposal for FARs.

As it can be appreciated in Fig. 6, discovering new knowledge is out of our scope, instead, our main objective is to efficiently maintain up-to-date measures of rules.

6 Experimentation results

The experiments have been designed to observe the different behaviors of the proposed algorithms in order to consider their implementation in real applications. The experiments also compare the proposed algorithms with those reported in the literature. These are being performed on real data and real structural events obtained from SWAD, a web system for education support at the University of Granada (Cañas et al. 2007). The studied data set consists of information about students' courses containing nine attributes over 5K instances. The most relevant attributes are the gender of the student (*gend*), the average of questions answered in all student exams (*avg_aqst*), the sum of visits to signature web files (*sum_vfiles*), the count of clicks in the platform (*cnt_clicks*), the count of downloads files (*cnt_dfiles*), and the average of all exam scores (*avg_score*).

Results compare the performance of the proposed algorithms and the naïve approach in order to maintain seven FARs related in Table 3. These rules were discovered using the KEEL data-mining software tool. The maintenance is implemented using the certainty factor metric (1) in two open source database management systems: PostgreSQL and MySQL. Both management systems have equivalent results but, due to space limitations only PostgreSQL results are included. The experiments were carried out on a dedicated GNU/Linux server with eight processors i7-2600 at 3.4 GHz and 15 GB of main memory.

Table 3 FARs used in experiments

Antecedent		Consequent	Supp.	Conf.	CF
pc_pcontestadas='middle'	$\Rightarrow$	*avg_score='middle'*	0.22	0.81	0.64
cnt_files='low'	$\Rightarrow$	*sum_vfiles='low'*	0.67	0.92	0.67
pc_pcontestadas='middle'	$\Rightarrow$	*cnt_clicks='low'*	0.25	0.92	0.58
gend='male' $\wedge$ *avg_score='high'*	$\Rightarrow$	*pc_pcontestadas='high'*	0.24	0.85	0.59
cnt_clicks='middle'	$\Rightarrow$	*avg_score='middle'*	0.16	0.83	0.67
pc_pcontestadas='high'	$\Rightarrow$	*avg_score='high'*	0.14	0.89	0.77
sum_vfiles='middle' $\wedge$ *cnt_files='middle'*	$\Rightarrow$	*avg_score='middle'*	0.13	0.85	0.71

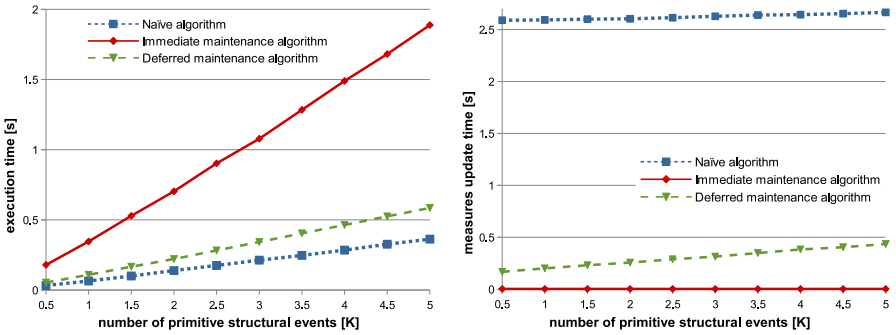

Fig. 7 Comparison of execution times (left) and measures update times (right) for different *PSE*

The experiments have been designed to observe two approaches behavior: active process execution time and measures update time after this process. The former presents the time needed for processing different numbers of primitive structural events on studied data set. Here, naïve approach represents the normal behavior of the database when no active algorithm is executed. The latter presents the time needed for the rule measures updates, after the same primitive structural events takes place. The total execution time for fuzzy rules base updating is the sum of both behaviors. The primitive structural events contain database insert, update, and delete operations extracted from real database transitions. In Fig. 7 these results for FARs maintenance are presented.

It can be noticed from Fig. 7 (left) the overhead time added to the active sections of our proposals for FARs maintenance. This overhead is higher in the immediate approach as expected. Otherwise, measures updating times for the naïve and immediate approaches (right) do not depend on the structural event quantity and remain almost constant, just the deferred proposal increases measure updates time as expected. This deferred proposal behavior is due to the increment of auxiliary relations and is not troubling since it corresponds with a single measure update time. For constantly measured access, auxiliary relations are always truncated.

All these experiments take a close look to our proposed behavior in a small data set. In order to observe the scalability of those algorithms, we obtained the total time needed to update a fuzzy rule base for different synthetically generated database sizes. The total execution time is calculated as the sum of executing 5K primitive structural events and updating measures of rules. Results for maintaining the FARs are illustrated in Fig. 8. In this way, Fig. 8 shows the sum of the last point in Fig. 7 for different sizes of the database.

Note that our proposals keep almost the same total execution time, an important consequence of a self-maintainable characteristic. This allows keeping the lowest total execution time in very large or big databases. Evidently, the naïve approach presents a low performance when a database grows up in size. The experiments for this result only consider a single measure update, multiple updates increase the difference between the naïve and proposed methods.

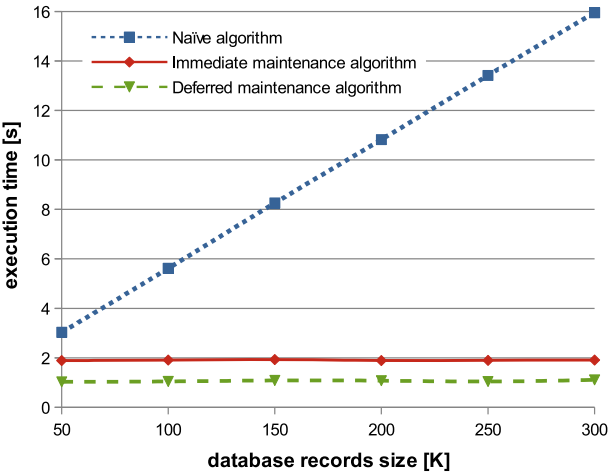

Fig. 8 Proposals comparison of total execution time for FARs maintenance in different database records size

The performance of the proposed algorithm was also compared with traditional and incremental algorithms for FARs maintenance. In Fig. 9 a total execution time for proposed algorithms, batch mining, and incremental mining methods is presented for different data sets. These data sets were obtained from the UCI Machine Learning Repository: the Diabetes 130-US hospitals for years 1999–2008 (diabetes), the Color Texture and the Color Moments parts of Corel Image Features. Details about these data sets can be found on the UCI Machine Learning website. For the diabetes data set, nine attributes were selected. Seven FARs were extracted using KEEL data-mining software tool from each data set in order to be incrementally maintained by proposed algorithms.

Our proposal shows the total time of executing 5K data operations plus updates measures of rules in order to maintain the fuzzy rule base up-to-date. Batch and incremental mining methods reflect the mining execution time for the same goal. The fuzzy Apriori algorithm (Hong et al. 2001) stands for batch mining methods. For incremental mining methods, we only consider the fuzzy FP-growth (Hong and Lin 2010) execution time and reject the fuzzy FP-tree built time, assuming that it was incrementally maintained. This approach is referred to as incremental fuzzy FP-growth. For fuzzy Apriori and fuzzy FP-growth algorithms, three fuzzy regions were defined for numeric attributes. The minimum support threshold was set at 10% and minimum confidence threshold at 80%. Both mining algorithm experiments were created using the KEEL data mining software tool.

It is obvious to conclude from Figs. 8 and 9 that the proposed algorithms are faster than the other algorithms. The above result times are accepted specially in real-world applications where decision-makers need to make decisions using an existing rule's information as soon as possible.

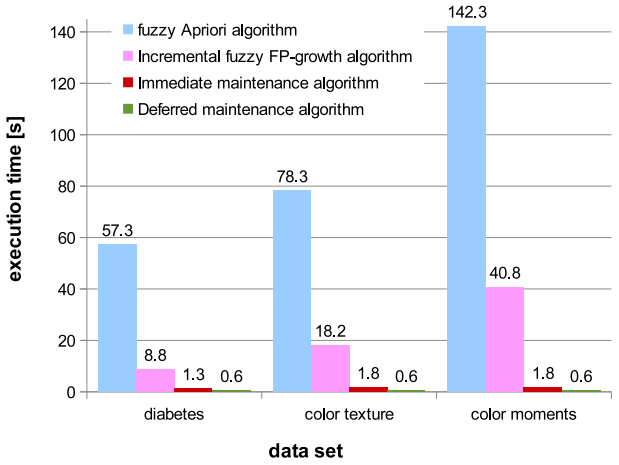

Fig. 9 Related and proposed algorithm comparison of total execution time for FARs maintenance in different data sets

7 Conclusions

In real-world applications, records are commonly inserted, updated or deleted outdating the previously extracted knowledge as inexact and invalid. In some scenarios, it is necessary to re-run traditional mining or incremental mining algorithms only for updating previously discovered FARs. It is possible, from another perspective, to maintain the known rules incrementally by computing data changes efficiently.

In this article, two algorithms have been proposed specifically for maintaining the previously discovered FARs, ready for decision support. These algorithms operate over a generic form of measures, allowing the maintenance of a wide range of rule metrics in an efficient way. We also propose to consider the interactions between data operations at real-time in order to create a reduced relevant instance set. Experimental results with real data and operations show that our proposals achieve a better performance against the batch mining, incremental mining, and a naïve approach. These improvements increase as the database size gets bigger, making it suitable in very large databases or big data systems.

There are still some interesting research issues related to the contributions of this paper that can be applied to other areas, specifically, to incrementally maintain other types of rules, to consider interactions between data operations in existing incremental mining algorithms, and to explore the memory usage of the proposed algorithms in different implementations.

Acknowledgements The authors would like to thank the members of the Iberoamerican Association of Postgraduate Universities (AUIP) for their international academic mobility program. We also thank our colleagues from the IDBIS Research Group (Intelligent DataBases and Information Systems) who provided insight and expertise that greatly assisted the research. We are grateful to all people who have contributed with their suggestions for improving the final version of the manuscript.

References

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Record*, *22*(2), 207–216.

Berzal, F., Blanco, I., Sánchez, D., & Vila, M. A. (2002). Measuring the accuracy and interest of association rules: A new framework. *Intelligent Data Analysis*, *6*(3), 221–235.

Boettcher, M., Ruß, G., Nauck, D., & Kruse, R. (2009). From change mining to relevance feedback: A unified view on assessing rule interestingness. In *Post-mining of association rules: Techniques for effective knowledge extraction* (pp. 12–37).

Cabot, J., & Teniente, E. (2005). Computing the relevant instances that may violate an OCL constraint. In O. Pastor & J. Falcão e Cunha (Eds.), *Advanced information systems engineering. Lecture notes in computer science* (Vol. 3520, pp. 48–62). Berlin: Springer.

Cadenas, J. M., & Verdegay, J. L. (2009). Towards a new strategy for solving fuzzy optimization problems. *Fuzzy Optimization and Decision Making*, *8*(3), 231–244.

Cañas, A., Calandria, D., Ortigosa, E., Ros, E., & Díaz, A. (2007). Swad: Web system for education support. In B. Fernández-Manjón, J. Sánchez-Pérez, J. Gómez-Pulido, M. Vega-Rodríguez, & J. Bravo-Rodríguez (Eds.), *Computers and Education* (pp. 133–142). Berlin: Springer.

Cheung, D., Han, J., Ng, V., & Wong, C. Y. (1996). Maintenance of discovered association rules in large databases: an incremental updating technique. In *1996. Proceedings of the 12th international conference on data engineering* (pp. 106–114).

Colby, L. S., Griffin, T., Libkin, L., Mumick, I. S., & Trickey, H. (1996). Algorithms for deferred view maintenance. *SIGMOD Record*, *25*(2), 469–480.

Delgado, M., Marin, N., Sánchez, D., & Vila, M. A. (2003). Fuzzy association rules: General model and applications. *IEEE Transactions on Fuzzy Systems*, *11*(2), 214–225.

Delgado, M., Ruiz, M. D., Sánchez, D., & Vila, M. A. (2014). Fuzzy quantification: A state of the art. *Fuzzy Sets and Systems*, *242*, 1–30. Theme: Quantifiers and Logic.

Greco, S., Słowiński, R., & Szczęch, I. (2012). Properties of rule interestingness measures and alternative approaches to normalization of measures. *Information Sciences*, *216*, 1–16.

Gupta, A., Mumick, I. S., et al. (1995). Maintenance of materialized views: Problems, techniques, and applications. *IEEE Data Engineering Bulletin*, *18*(2), 3–18.

Hong, T., & Lin, C. (2010). Tsung-Ching Lin: Mining complete fuzzy frequent itemsets by tree structures. In *2010 IEEE international conference on systems man and cybernetics (SMC)* (pp. 563–567).

Hong, T. P., Kuo, C. S., & Chi, S. C. (2001). Trade-off between computation time and number of rules for fuzzy mining from quantitative data. *International Journal of Uncertainty and Fuzziness and Knowledge-Based Systems*, *9*(5), 587–604.

Hong, T. P., Lin, C. W., Lin, T. C., & Wang, S. L. (2012). Incremental multiple fuzzy frequent pattern tree. In *2012 IEEE international conference on fuzzy systems (FUZZ-IEEE)* (pp. 1–5).

Jain, H., & Gosain, A. (2012). A comprehensive study of view maintenance approaches in data warehousing evolution. *SIGSOFT Software Engineering Notes*, *37*(5), 1–8.

Lee, H. C., & Guu, S. M. (2003). On the optimal three-tier multimedia streaming services. *Fuzzy Optimization and Decision Making*, *2*(1), 31–39.

Lenca, P., Meyer, P., Vaillant, B., & Lallich, S. (2008). On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. *European Journal of Operational Research*, *184*(2), 610–626.

Lin, C. W., Hong, T. P., & Lu, W. H. (2010). Linguistic data mining with fuzzy FP-trees. *Expert Systems with Applications*, *37*(6), 4560–4567.

Lin, C. W., Wu, T. Y., Lin, G., & Hong, T. P. (2014). Maintenance algorithm for updating the discovered multiple fuzzy frequent itemsets for transaction deletion. In *2014 international conference on machine learning and cybernetics (ICMLC)* (Vol. 2, pp. 475–480).

Papadimitriou, S., & Mavroudi, S. (2005). The fuzzy frequent pattern tree. In *Proceedings of the 9th WSEAS international conference on computers, ICCOMP'05* (pp. 3:1–3:7). Stevens Point, Wisconsin: World Scientific and Engineering Academy and Society (WSEAS).

Pérez-Alonso, A., Blanco, I. J., Serrano, J. M., & González-González, L. M. (2017a). Drims: A software tool to incrementally maintain previous discovered rules. In H. Christiansen, H. Jaudoin, P. Chountas, T. Andreasen, & H Legind Larsen (Eds.), *Flexible query answering systems* (pp. 174–185). Cham: Springer International Publishing.

Pérez-Alonso, A., Medina, I. J. B., González-González, L. M., & Serrano Chica, J. M. (2017b). Incremental maintenance of discovered association rules and approximate dependencies. *Intelligent Data Analysis*, *21*(1), 117–133.

Sauter, V. (2014). *Decision support systems for business intelligence*. London: Wiley.

Tan, J., Bu, Y., & Zhao, H (2010). Incremental maintenance of association rules over data streams. In *2010 2nd international conference on networking and digital society (ICNDS)* (Vol. 2, pp. 444–447).

Urpí, T., & Olivé, A. (1992). A method for change computation in deductive databases. In *Proceedings of the 18th international conference on very large data bases, VLDB'92* (pp. 225–237). San Francisco, CA: Morgan Kaufmann Publishers Inc.

Urpí, T., & Olivé, A. (1994). Semantic change computation optimization in active databases. In *1994. Active database systems. Proceedings 4th international workshop on research issues in data engineering* (pp. 19–27).

Wu, X., Zhu, X., Wu, G. Q., & Ding, W. (2014). Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering*, *26*(1), 97–107.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353.

Zadeh, L. A. (1983). A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with Applications*, *9*(1), 149–184.