STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LXIII**, Number 1, 2018
DOI: 10.24193/subbi.2018.1.07

AN ADAPTIVE GRADUAL RELATIONAL ASSOCIATION RULES MINING APPROACH

DIANA-LUCIA MIHOLCA

Abstract. This paper focuses on adaptive Gradual Relational Association Rules mining. Gradual Relational Association Rules capture gradual generic relations among data features. We propose *AGRARM*, an algorithm for mining the interesting Gradual Relational Association Rules characterizing a data set that has been extended with a number of new attributes, through adapting the set of interesting rules mined before extension, so as to preserve the completeness. We aim, through *AGRARM*, to make the mining process more efficient than resuming the mining algorithm on the enlarged data. We have experimentally evaluated *AGRARM* versus mining from scratch on three publicly available data sets. The obtained reduction in mining time highlights *AGRARM*'s efficiency, thus confirming the potential of our proposal.

1. Introduction

Data mining is widely applied in various domains, such as medicine [5], bioinformatics [6] or software engineering [10] [9] [14], to discover relevant patterns in large data sets.

Association Rules (ARs) *mining* [4] is a data mining procedure for identifying frequent associations in data. Classical association rules capture frequent co-occurrences of attribute values, while ignoring any possible frequent relation between attribute values.

Ordinal Association Rules (OARs) [3] customize *Association Rules* (ARs) [1] so as to express ordinal relations among numeric attributes that characterize a data set. But different informative relations, that are not ordinal, may exist between the attribute values. OARs fail to capture them.

Received by the editors: April 2, 2018.

2010 *Mathematics Subject Classification.* 68P15, 68T05, 62B86.

1998 *CR Categories and Descriptors.* H.2.8[**Database management**]: Database Applications – *Data Mining*; I.2.6[**Computing Methodologies**]: Artificial Intelligence – *Learning*.

Key words and phrases. data mining, Gradual Relational Association Rules, adaptive algorithm.

Consequently, *Relational Association Rules* (RARs) [18] [2] generalize *Ordinal Association Rules* so as to capture relations that may not be ordinal, between not necessary numeric attributes. Compared to the classical *Association Rules*, RARs express more powerful rules which may lead to valuable data mining results.

Subsequently, *Adaptive Relational Association Rule Mining* (*ARARM*) [8] has been proposed as a method for adapting the set of all interesting RARs discovered within a data set before extending its features set, so as to obtain all interesting RARs within the extended data set.

There are situations when the degree to which a relation between two attributes is satisfied is relevant. So, RARs have been further extended to *Gradual Relational Association Rules* (or GRARs) [7] which, through the use of *fuzzy* relations instead of *boolean* relations, are also aware of the degree to which the relations are satisfied.

For discovering all the interesting Gradual Relational Association Rules that describe a data set, *Gradual Relational Association Rules Miner* (*GRANUM*) [7] has been proposed. *GRANUM* mines a known set of objects that are measured against a known set of features and discovers all interesting GRARs characterizing the data set. But there are also situations where the data is horizontally dynamic, in the sense that the feature set characterizing its objects evolves (i.e. new attributes are added). Clearly, for obtaining, in such a setting, the interesting GRARs, the mining algorithm can be re-applied, from scratch, every time the feature set changes (i.e. one or more new attributes are added). But this could be inefficient and unworthy especially if the attribute set is only very slightly expanded, for instance by adding just one new attribute.

Consequently, we propose, in the current paper, an alternative to resuming the *GRANUM* mining algorithm when the data set is enlarged with a number of new attributes. We propose, therefore, *Adaptive Gradual Relational Association Rules Miner* (*AGRARM*), which is an algorithm that adapts the set of all interesting GRARs mined before extension so as to obtain all interesting GRARs that characterize the extended data. *AGRARM* is the equivalent of *ARARM* [8], but for mining GRARs instead of RARs, within a dynamic data set.

The remaining of this paper is structured as follows. We start by giving, in Section 2, a background on *Gradual Relational Association Rules*. The proposed *Adaptive Gradual Relational Association Rules Miner* (*AGRARM*) is presented in Section 3. In Section 4, we detail the experiments performed in order to evaluate *AGRARM* against *GRANUM* applied from scratch and we discuss the results obtained. A comparison to related approaches is also

given in Section 4. Finally, the conclusions and directions for further work are stated in Section 5.

2. Background on Gradual Relational Association Rules

We briefly present in the following the concept of *Gradual Relational Association Rules* [7].

Gradual Relational Association Rules (*GRARs*) generalize *Relational Association Rules* (*RARs*) [18] by using *fuzzy relations* instead of crisp relations and thus enhancing them with gradualness. The gradual rules are able to express additional semantically relevant characteristics of data and have been proven to be more noise-tolerant [7].

Let $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ be a set of *instances* (entities, records or objects). Each instance e_i in $\mathcal{E}$ consists of a sequence of values for m *attributes* (or features), $\mathcal{A} = (a_1, \ldots, a_m)$. Each attribute a_j takes values from a non-empty and non-fuzzy domain D_i, which also contains a *null* (or empty) value. If we denote by $\Phi(e_i, a_j)$ the value of the instance e_i for the attribute a_j, an instance will be $e_i = (\Phi(e_i, a_1), \Phi(e_i, a_2),$
$\Phi(e_i, a_3), ...\Phi(e_i, a_m))$.

A fuzzy binary relation $\mathcal{G}$ between two attribute domains D_i and D_j is defined as follows:

$$\mathcal{G} = \{< (v_1, v_2),\ \mu_R(v_1, v_2) >: v_1 \in D_i, v_2 \in D_j\}$$

$\mu_R : D_i \times D_j \to [0, 1]$ is a *membership* function which associates to each pair $(v_1, v_2), v_1 \in D_i, v_2 \in D_j$ the *membership degree* $\mu_R(v_1, v_2)$ which numerically expresses the degree to which the relation $\mathcal{G}$ is satisfied.

We denote by $\mathcal{F}$ the set of all fuzzy binary relations which can be defined between any two crisp attribute domains.

Definition 2.1. *A **Gradual Relational Association Rule**, gRule, is a sequence* $(a_{i_1}\ \mathcal{G}_1\ a_{i_2}\ \mathcal{G}_2\ a_{i_3} \ldots \mathcal{G}_{\ell-1}\ a_{i_\ell})$*, where* $\{a_{i_1}, a_{i_2}, a_{i_3}, \ldots, a_{i_\ell}\} \subseteq \mathcal{A} = \{a_1, \ldots, a_m\}$*,* $a_{i_j} \neq a_{i_k}$*,* $j, k = 1..\ell$ *and* $\mathcal{G}_j \in \mathcal{F}$ *is a binary fuzzy relation over* $D_{i_j} \times D_{i_{j+1}}$ [7].

*The **membership degree** of the gradual relational association rule gRule for data instance* $e \in \mathcal{E}$ *is defined as* $\mu_{gRule}(e) = \min\{\mu_{R_j}(\Phi(e, a_{i_j}), \Phi(e, a_{i_{j+1}}))$*,* $j = 1, 2, \ldots, \ell - 1\}$ *and expresses the magnitude to which the rule is satisfied.*

a) *If* $a_{i_1}, a_{i_2}, a_{i_3}, \ldots, a_{i_\ell}$ *are non-missing in p instances from the data set then we call* $\frac{p}{n}$ *the **support** of the rule.*

b) *If we denote by* $\mathcal{E}' \subseteq \mathcal{E}$ *the set of instances where* $a_{i_1}, a_{i_2}, a_{i_3}, \ldots, a_{i_\ell}$ *are non-missing and* $\mu_{gRule}(e) > 0$ *for each instance e from* $\mathcal{E}'$*, then we call* $\frac{|\mathcal{E}'|}{n}$ *the **confidence** of the rule.*

c) *Using the notation from b), we call* $\frac{\sum_{e \in \mathcal{E}'} \mu_{gRule}(e)}{n}$ *the rule's* ***membership***.

The number l of attributes in a rule gives the rule *length*.

When introducing the concept of *Gradual Relational Association Rules* in the literature [7], we kept the definition of *interestingness* previously proposed for non-gradual *Relational Association Rules*. In accordance with this, a rule is *interesting* if its support and confidence are greater or equal to given thresholds. In a later work [14], we suggested that we could customize *interestingness* by including an additional minimum threshold condition for *membership*. So, the current work is in accordance with the definition for *interestingness* customized as follows:

Definition 2.2. *We call a GRAR* interesting *if its support s, confidence c and membership m are greater than or equal to given thresholds, i.e. $s \geq s_{min}$, $c \geq c_{min}$ and $m \geq m_{min}$.*

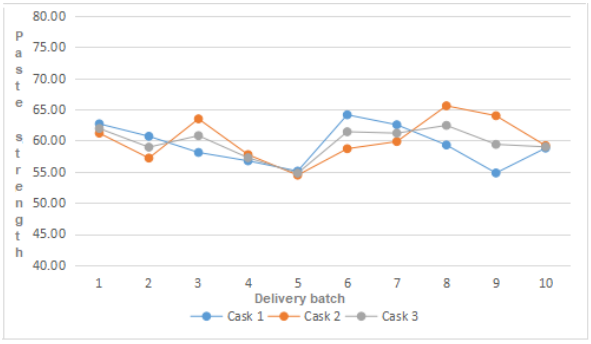
Definition 2.3. *The* inverse *of binary fuzzy relation $\mathcal{G} = \{< (x, y), \mu_{\mathcal{G}}(x, y) >: x \in X, y \in Y\}$ will be denoted in the following by $\mathcal{G}^{-1}$ and is defined as $\mathcal{G}^{-1} = \{< (x, y), 1 - \mu_{\mathcal{G}}(x, y) >: x \in X, y \in Y\}$.*

GRANUM [7] has been proposed as an *Apriori* mining algorithm for discovering all *interesting* GRARs within a data set. For more details about *GRANUM* and *GRARs* in general, we refer the reader to [7].

2.1. **Example.** We exemplify in the following the previously presented concept of *Gradual Relational Association Rules*. Therefore, we mine a small real data set taken from [12] and depicted in Figure 1. The data consist of the results obtained by testing chemical pastes as described in the following. The chemical paste product is delivered in batches of casks. Immediately after the arrival of a batch, the material from three randomly selected casks is analyzed, errors arising from both the sampling and the analysis. The data instances correspond to ten delivery batches chosen at random, while the data attributes are given by the average of the percentage paste strengths obtained by two analyzes of the contents of the three selected casks.

We propose to compare the paste strengths obtained by analyzing the contents of the three randomly selected casks. Since there are errors in data, we opt for GRARs [7] instead of non-gradual RARs.

Having $\mathcal{F} = \{\approx (\textit{approximately equal}), \lesssim (\textit{fuzzy less}) \textit{and} \gtrsim (\textit{fuzzy greater})\}$ as the set of gradual relations and setting the minimum support, confidence and membership thresholds at $s_{min} = 1$, $c_{min} = 1$ and $m_{min} = 0.9$, the



Cask 1	Cask 2	Cask 3
62.70	61.20	61.95
60.70	57.20	58.95
58.10	63.50	60.80
56.75	57.75	57.25
55.10	54.45	54.78
64.15	58.70	61.43
62.55	59.85	61.20
59.30	65.60	62.45
54.80	64.00	59.40
58.80	59.20	59.00

FIGURE 1. *Strength of chemical pastes* data set

GRANUM mining algorithm will discover as interesting rules the rules given in Table 1.

Rule	Length	Support	Confidence	Membership
Cask 1 ≈ Cask 2	2	1.0	1.0	0.935
Cask 1 ≈ Cask 3	2	1.0	1.0	0.982
Cask 2 ≈ Cask 3	2	1.0	1.0	0.983
Cask 1 ≈ Cask 2 ≈ Cask 3	3	1.0	1.0	0.935

TABLE 1. Interesting rules on data set from Table 1 for $s_{min} = 1$, $c_{min} = 1$ and $m_{min} = 0.9$

Interpreting the obtained GRARs, we can conclude that the results of the analyzes performed for the three selected casks are approximately equal (since *Cask 1* ≈ *Cask 2* ≈ *Cask 3* with a rather large membership degree of 0.935). Furthermore, we deduce that the strengths of the material from the third selected cask differ in almost equal extents from the strengths obtained for the other two casks (since *Cask 1* ≈ *Cask 3* with membership 0.982 and *Cask 2* ≈ *Cask 3* with membership 0.983), while these two are not as close to each other (since *Cask 1* ≈ *Cask 2* with a smaller membership of 0.935). These conclusions are confirmed by analyzing the graphical data representation from Figure 1.

3. METHODOLOGY

We introduce in the current section *AGRARM*, the ***A****daptive* ***G****radual* ***R****elational* ***A****ssociation* ***R****ules* ***M****ining* method we propose for mining all interesting *GRARs* in a dynamic data set whose feature set is extended with one or more new features.

Let $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ be a data set. Each entity is initially defined by the values for m features (attributes or characteristics), $\mathcal{A} = (a_1, \ldots, a_m)$, thus being a m-dimensional sequence: $e_i = (e_i^1, \ldots, e_i^m)$. Subsequently, $\mathcal{A}$ is extended with $s \geq 1$ new features, thus obtaining an extended feature set $\mathcal{A}^{ext} = (a_1, \ldots, a_m, a_{m+1}, \ldots,$
$a_{m+s})$ and an afferent extended data set $\mathcal{E}^{ext} = \{e_1^{ext}, e_2^{ext}, \ldots, e_n^{ext}\}$. Each extended data instance $e_i^{ext} \in \mathcal{E}^{ext}$ is therefore given by the values for the $m + s$ attributes that describe the extended data set $\mathcal{E}^{ext}$: $e_i^{ext} = (e_i^{ext,\ 1}$, $e_i^{ext,\ 2}, \ldots, e_i^{ext,\ m+s})$.

In this context, the problem we are approaching is to find the set $\mathcal{GR}ules^{ext}$ of all interesting $GRARs$ that occur in the extended data set $\mathcal{E}^{ext}$, starting from the set $\mathcal{GR}ules$ of all interesting $GRARs$ in the non-extended data set $\mathcal{E}$. The motivation is that we expect a better time performance through completing the rules already mined on the data before extension than by applying the mining process from scratch on the extended data.

So, we further present $AGRARM$ (***A**daptive **G**radual **R**elational **A**ssociation **R**ule **M**iner*), a complete algorithm that, starting from $\mathcal{GR}ules$ and considering the newly added features, adapts the rule set so as to obtain $\mathcal{GR}ules^{ext}$.

`Function AGRARM`($\mathcal{E}$, $\mathcal{E}^{ext}$, $\mathcal{F}$, $\mathcal{GR}ules$, c_{min}, s_{min}, m_{min})

Input: $\mathcal{E}$ - the initial non-extended set of m-dimensional entities,
$\mathcal{E}^{ext}$ - the final extended set of $m+s$-dimensional entities,
$\mathcal{F}$ - the set of fuzzy binary relations used in the mining process,
$\mathcal{GR}ules$ - the set of all interesting $GRARs$ mined on the non-extended data set $\mathcal{E}$,
c_{min}, s_{min} and m_{min} - the minimum thresholds for support, confidence and membership, respectively

Output: $\mathcal{GR}ules^{ext}$ - the set of all interesting $GRARs$ that characterize $\mathcal{E}^{ext}$, the extended data set

$AdaptiveRules \leftarrow$ *the binary* $(2-length)$ *rules from* $\mathcal{GR}ules$
$Cand \leftarrow \{\ (a_{i_1}\ \mathcal{G}\ a_{i_2}) \mid a_{i_1}, a_{i_2} \in A,\ i_1 = 1 \ldots m+s, i_2 = m+1 \ldots m+s,\ i_1 < i_2,\ \mathcal{G} \in \mathcal{F}\ \}$
`Foreach` $gRule$ `in` $Cand$ `do`
  `If` $IsInteresting(gRule, \mathcal{E}^{ext}, c_{min}, s_{min}, m_{min})$ `then`
    $AdaptiveRules \leftarrow AdaptiveRules \cup \{gRule\}$
  `EndIf`
`EndFor`
$\mathcal{GR}ules^{ext} \leftarrow AdaptiveRules$
$l \leftarrow 3$
$complete \leftarrow false$
`While` $(\neg complete)$ `do`
  $Cand \leftarrow GenCandidates(AdaptiveRules)$
  $AdaptiveRules \leftarrow l-length$ *rules from* $\mathcal{GR}ules$

```
    Foreach gRule in Cand do
      If IsInteresting(gRule, E^ext, c_min, s_min, m_min) then
        AdaptiveRules ← AdaptiveRules ∪ {gRule}
      EndIf
    EndFor
    If AdaptiveRules = ∅ then
      complete ← true
    else
      l ← l + 1
      GRules^ext ← GRules^ext ∪ AdaptiveRules
    EndIf
  EndWhile
  AGRARM ← GRules^ext
EndFunction
```

The *AGRARM* algorithm discovers all interesting *GRARs* through an iterative process. At each iteration, the length-level generation of rules is followed by the verification of their *interestingness*. As we mentioned in Section 2 , the *interestingness* of a *GRARs* is a property that is tested in relation to given support, confidence and membership minimum thresholds. We give in the following the function that checks if a candidate *GRAR* is or is not interesting at the level of the extended data set $\mathcal{E}^{ext}$.

```
Function IsInteresting(gRule, E^ext, c_min, s_min, m_min)
```

Input: $\mathcal{E}^{ext}$ - the final extended set of m+s-dimensional entities,
$gRule$ - the gradual relational association rule whose *interestingness* on $\mathcal{E}^{ext}$ is verified
c_{min}, s_{min} and m_{min} - the minimum thresholds for support, confidence and membership, respectively

Output: $true$ - if $gRule$ is interesting on $\mathcal{E}^{ext}$ (i.e. it satisfies c_{min}, s_{min}, and m_{min} minimum thresholds) or
$false$ - otherwise

$n \leftarrow |\mathcal{E}^{ext}|$
$requiredSuppport \leftarrow \lceil n \cdot s_{min} \rceil$
$requiredConfidence \leftarrow \lceil n \cdot c_{min} \rceil$
$requiredMembership \leftarrow \lceil n \cdot m_{min} \rceil$
$support \leftarrow 0$
$confidence \leftarrow 0$
$membership \leftarrow 0$
$remainingEntities \leftarrow n$

```
Foreach instance in E^ext do
  UpdateSuppConfM(gRule, instance, support, confidence, membership)
  remainingEntities ← remainingEntities − 1
  If (support + remainingEntities < requiredSupport)
    or (confidence + remainingEntities < requiredConfidence)
    or (membership + remainingEntities < requiredMembership) then
```

$IsInteresting \leftarrow false$
EndIf
If $(support \geq requiredSupport)$ *and* $(confidence \geq requiredConfidence)$ *and* $(membership \geq requiredMembership)$ then
$IsInteresting \leftarrow true$
EndIf
EndFor
$IsInteresting \leftarrow false$
EndFunction

The method that follows presents the update of support, confidence and membership of a $GRAR$ when considering a current data instance.

Subalgorithm $UpdateSuppConfM(gRule, instance, supp, conf, membership)$
Input: $gRule$ - the gradual relational association rule whose support, confidence and membership will be updated considering the *instance* data entity
$instance$ - the data instance on which the rule $gRule$ is evaluated so as to update the $supp$, $conf$ and $membership$ values
$supp$, $conf$ and $membership$ - the current support, confidence and membership for $gRule$ which are required to be updated through also considering *instance*.
Output: $supp'$, $conf'$ and $membership'$ - the support, confidence and membership of $gRule$ are updated as a result of evaluating $gRule$ on *instance*
If @$instance$ *has* $non-missing$ *values for all attributes in* $gRule$ then
$supp \leftarrow supp + 1$
$m \leftarrow min(@$ *the memberships of the fuzzy relations in gRule on the instance data entity*$)$
If $m > 0$ then
$conf \leftarrow conf + 1$
EndIf
$membership \leftarrow membership + m$
EndIf
EndSubalgorithm

So, $AGRARM$, the proposed method, starts by performing an initial pass over the extended data set $\mathcal{E}^{ext}$ so as to identify the interesting binary rules in addition to the 2-length rules from $GRules$. In every subsequent iteration, the set of interesting rules of length $k > 2$ will be mined. This set will obviously include the k-length rules from the set $\mathcal{G}Rules$. But there is an alternative to obtain a k-length interesting rule. The alternative consists in generating a new candidate rule by joining two $(k-1)$-length rules from $\mathcal{G}Rules^{ext}$ such that at least one of the two rules contains at least one newly added attribute. The candidate rules generation is followed by the verification of minimum support, confidence and membership compliance. At the end of each iteration, all the k-length interesting rules will be included in the set $\mathcal{G}Rules^{ext}$. The mining process stops when no new interesting rules have been discovered in the latest iteration.

We present in the following the method of generating candidate rules.

```
Function GenCandidates(GRules_k)
Input:   GRules_k - the interesting GRARs of length k
Output:  GRules_{k+1} - the candidate GRARs of length k+1 which were obtained
    through joining pairs of rules in GRules_k
    GRules_{k+1} ← ∅
    n ← |GRules_k|
    For i ← 1 to n − 1 do
      For j ← i + 1 to n do
        gRule_i ← the i − th rule from GRules_k
        gRule_j ← the j − th rule from GRules_k
        If @gRule_i or gRule_j contain at least one newly added attribute (i.e.
          in the set {a_{m+1}, a_{m+2}, ..., a_{m+s}}) then
          If @gRule_i matches for join with gRule_j in one of the cases (1)
            −(4) from Figure 2 then
            resultingRule ← @ the rule obtained by joining gRule_i and gRule_j
            GRules_{k+1} ← GRules_{k+1} ∪ {resultingRule}
          EndIf
        EndIf
      EndFor
    EndFor
    GenCandidates ← GRules_{k+1}
  EndFunction
```

In Figure 2, we present the four rules according to which *GenCandidates* proposes new candidate rules.

4. Results and discussion

We present in the following the experiments we performed in order to comparatively evaluate *AGRARM* against *GRANUM* applied from scratch, the comparison being performed in the context in which the data of interest is extended with a number of new attributes.

In these comparative experiments, we considered three different data sets, various possibilities of extending their attribute sets and multiple values for the minimum support, confidence and membership thresholds.

The three data sets we have considered in our experiments are publicly available in $tera - PROMISE$ repository [17]. They are *Tomcat*, *Ar* and $JM1$. The *Tomcat* data set consists of the values for 20 Chidamber and Kemerer (CK) software metrics, computed for the 858 classes in Apache Tomcat software, version 6.0. The *Ar* data set is composed of 29 static code attributes (McCabe, Halstead and LOC software measures), for 745 modules in Ar, which is an embedded software implemented in C. The third data set, $JM1$, consists of 7782 instances, corresponding to modules in $JM1$ software, each being characterized by 21 attributes (5 different lines of code measures, 3 *McCabe* metrics, 4 base *Halstead* measures, 8 derived *Halstead* measures and a

$$
\begin{aligned}
&gRule_1 \equiv (a^1\mathcal{G}^1 a_{i_1}\mathcal{G}_1 a_{i_2}\ldots\mathcal{G}_{k-3}a_{i_{k-2}}),\\
&gRule_2 \equiv (a_{i_1}\mathcal{G}_1 a_{i_2}\ldots\mathcal{G}_{k-3}a_{i_{k-2}}\mathcal{G}^2 a^2), \qquad (1)\\
\Rightarrow &resultingRule \equiv (a^1\mathcal{G}^1 a_{i_1}\mathcal{G}_1 a_{i_2}\ldots\mathcal{G}_{k-3}a_{i_{k-2}}\mathcal{G}^2 a^2),\\
&\text{or}\\
&gRule_1 \equiv (a_{i_1}\mathcal{G}_1 a_{i_2}\ldots\mathcal{G}_{k-3}a_{i_{k-2}}\mathcal{G}^1 a^1),\\
&gRule_2 \equiv (a^2\mathcal{G}^2 a_{i_1}\mathcal{G}_1 a_{i_2}\ldots\mathcal{G}_{k-3}a_{i_{k-2}}), \qquad (2)\\
\Rightarrow &resultingRule \equiv (a^2\mathcal{G}^2 a_{i_1}\mathcal{G}_1 a_{i_2}\ldots\mathcal{G}_{k-3}a_{i_{k-2}}\mathcal{G}^1 a^1),\\
&\text{or}\\
&gRule_1 \equiv (a^1\mathcal{G}^1 a_{i_1}\mathcal{G}_1 a_{i_2}\ldots\mathcal{G}_{k-3}a_{i_{k-2}}),\\
&gRule_2 \equiv (a^2\mathcal{G}^2 a_{i_{k-2}}\mathcal{G}_{k-3}^{-1}\ldots a_{i_2}\mathcal{G}_1^{-1}a_{i_1}), \qquad (3)\\
\Rightarrow &resultingRule \equiv (a^1\mathcal{G}^1 a_{i_1}\mathcal{G}_1 a_{i_2}\ldots\mathcal{G}_{k-3}a_{i_{k-2}}(\mathcal{G}^2)^{-1}a^2),\\
&\text{or}\\
&gRule_1 \equiv (a_{i_1}\mathcal{G}_1 a_{i_2}\ldots\mathcal{G}_{k-3}a_{i_{k-2}}\mathcal{G}^1 a^1),\\
&gRule_2 \equiv (a_{i_{k-2}}\mathcal{G}_{k-3}^{-1}\ldots a_{i_2}{\mathcal{G}_1}^{-1}a_{i_1}\mathcal{G}^2 a^2), \qquad (4)\\
\Rightarrow &resultingRule \equiv (a^2(\mathcal{G}^2)^{-1}a_{i_1}\mathcal{G}_1 a_{i_2}\ldots\mathcal{G}_{k-3}a_{i_{k-2}}\mathcal{G}^1 a^1).
\end{aligned}
$$

FIGURE 2. The joining rules considered by the candidate generation process in the *AGRARM* algorithm

branch-count). We mention that, prior to the mining phase, the data have been preprocessed in the sense that the values have been scaled using the *Min-Max* scaling method.

In each of the experiments, the interesting *GRARs* on the extended $(m + s)$-dimensional instances have been mined in the following two ways: (1) by applying *GRANUM* from scratch on the extended data and (2) by applying *AGRARM* so as to adapt the rules mined before extension. Certainly, the interesting *GRARs* mined were the same regardless of the mining method applied (i.e. (1) or (2)). But we will compare the time required by the two methods in order to test our expectation that *AGRARM* is faster than *GRANUM* applied from scratch, at least if the data set is expanded with a relatively small number of attributes.

We considered, in the mining processes, the following set of fuzzy binary relations: $\mathcal{F} = \{\approx$ (*approximately equal*), $\lesssim$ ($fuzzy\ less$), $\gtrsim$ ($fuzzy\ greater$), $\sim\ll$ ($fuzzy\ much$
less), $\sim\gg$ ($fuzzy\ much\ greater$)$\}$. The $\approx$ relation has been defined using the asymmetric Gaussian membership function, while the rest of the fuzzy relations have been defined through S-shaped membership functions, which have been parameterized, of course, so that the following inequalities occur: $\lesssim (x, y) \geq \sim\ll (x, y)$ and $\gtrsim (x, y) \geq \sim\gg (x, y)$.

We mention that the experiments have been carried out on a PC with an Intel Core i7 Processor at 2.40 GHz, with 8 GB of RAM.

We depict in Table 2 the results obtained by applying *AGRARM* versus *GRANUM* from scratch on *Tomcat* data set, when considering the minimum support threshold

m	s	Rules on $\mathcal{E}$	Time *GRANUM* (ms)	Time *AGRARM* (ms)	Time reduction
2	18	0	273.67	269.66	0.014
3	17	0	275.08	275	0.0002
4	16	0	274.14	273.8	0.001
5	15	0	274.55	272.58	0.007
6	14	9	275.42	264.03	0.041
7	13	9	274.68	261.28	0.048
8	12	9	274.49	259.55	0.054
9	11	32	274.12	237.93	0.132
10	10	32	274.04	232.9	0.150
11	9	32	274.99	230.38	0.162
12	8	32	273.77	226.75	0.172
13	7	32	273.97	223.12	0.186
14	6	32	273.83	219.78	0.197
15	5	99	274.87	119.13	0.567
16	4	100	275.9	115.27	0.582
17	3	117	275.52	88.92	0.677
18	2	171	276.52	14.29	0.948
19	1	171	275.5	8.22	0.970

TABLE 2. Experimental results obtained on *Tomcat* data set for $s_{min} = 1$, $c_{min} = 0.97$ and $m_{min} = 0.5$

$s_{min} = 1$, the minimum confidence threshold $c_{min} = 0.97$ and the minimum membership threshold $m_{min} = 0.5$. Here, m gives the number of initial attributes, while s gives the number of newly added attributes.

In Table 2, we give, on the first column, the number m of initial attributes, on the second column, the number s of newly added attributes, on the third column, the number of interesting *GRAR*s mined before extension, on the fourth and fifth columns the mining time for *GRANUM* and *AGRARM*, respectively, and, on the last column, the time reduction obtained by applying *AGRARM* to the detriment of *GRANUM* applied from scratch. The reduction in mining time has been computed as the ratio between the gained time (i.e. the difference between the time required by *GRANUM* and the time required by *AGRARM*) and the time consumed through resuming the mining process (i.e. applying *GRANUM* from scratch).

We observe from the table that the time reduction becomes significant when the newly added attributes count no more than one third of the number of initial attributes. For instance, when $\frac{s}{m} = \frac{1}{3}$ (i.e. $s = 5$ and $m = 15$), the mining time is reduced by more than 56%. The most substantial reduction, namely 97%, is obtained when the data set is extended with only one attribute.

Figures 3 and 4 illustrate how the time reduction evolves, depending on the number s of new attributes, for additional case studies on the *Tomcat* data set.

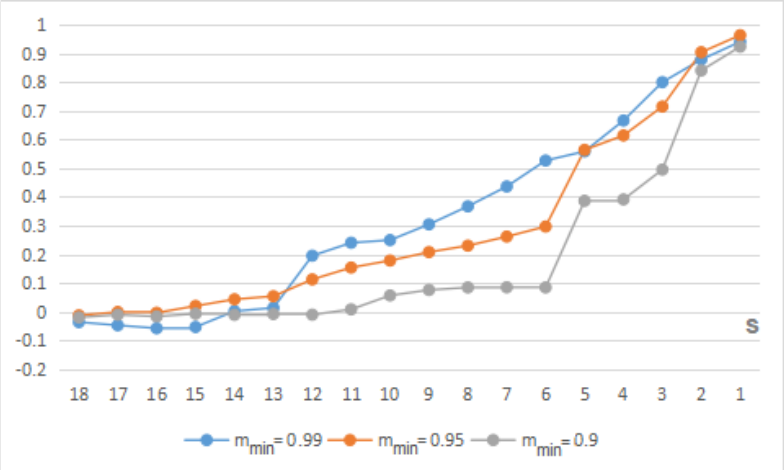


FIGURE 3. The reduction in total mining time when applying *AGRARM* on Tomcat and considering $s_{min} = 1$, $c_{min} = 0$ and $m_{min} \in \{0.99,\ 9.95,\ 0.9\}$

The results illustrated in Figure 3 have been obtained by imposing, besides the condition of a minimum support threshold $s_{min} = 1$, minimum membership thresholds, thus renouncing at also using a minimum confidence threshold to condition the *interestingness* of a *GRAR* (i.e. c_{min} has been set as 0). We successively initialized the minimum membership threshold with the following values: 0.99, 0.95 and 0.9.

In Figure 4 we give the reductions obtained by considering the minimum support threshold $s_{min} = 1$ and by varying both the minimum confidence and membership thresholds. We successively considered $c_{min} = 0.99$ and $m_{min} = 0.95$, $c_{min} = 0.95$ and $m_{min} = 0.9$ and, as a third setting, $c_{min} = 0.97$ and $s_{min} = 0.5$.

From both figures we can deduce that the time required by *AGRARM* decreases as the number s of newly added attributes decreases. Consequently, the adaptive algorithm we propose proves to be significantly more efficient than *GRANUM* applied from scratch when s is relatively small.

In order to strengthen the finding according to which *AGRARM* really makes the mining process more efficient when data is enlarged with relatively few new attributes, we comparatively tested it on two more data sets.

We present in Figure 5 the time reductions obtained on *Ar* data set when considering $s_{min} = 1$ and various values for c_{min} and m_{min}.

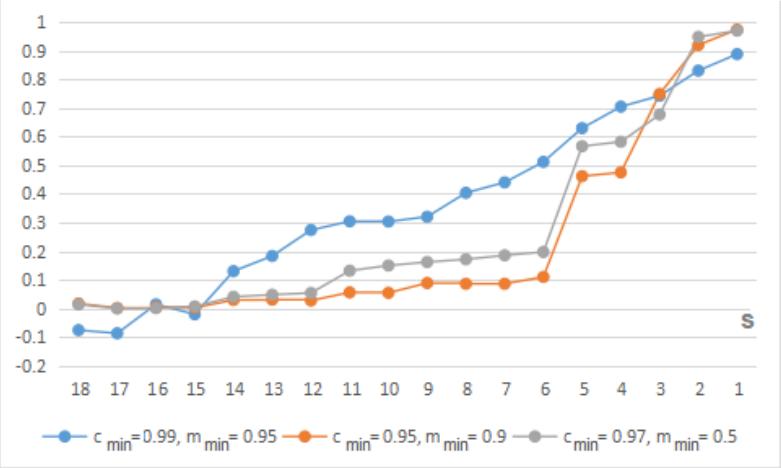


FIGURE 4. The reduction in total mining time when applying *AGRARM* on *Tomcat* and considering $s_{min} = 1$ and $(c_{min}, s_{min}) \in \{(0.99, 0.95),\ (0.95, 0.9),\ (0.97, 0.5)\}$

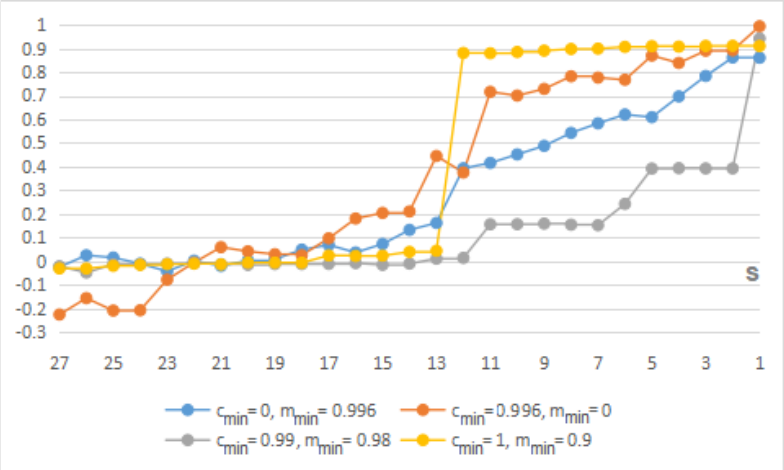


FIGURE 5. The reduction in total mining time when applying *AGRARM* on *Ar* and considering $s_{min} = 1$ and $(c_{min}, s_{min}) \in \{(0, 0.996),\ (0.996, 0),\ (0.99, 0.98),\ (1, 0.9)\}$

Figure 6 illustrates how the total mining time is reduced when applying, on *JM1* data set, *AGRARM* instead of *GRANUM* from scratch. In the experiments performed on *JM1* we also set the minimum support threshold, s_{min}, to 1, while varying the values for the minimum confidence and membership.

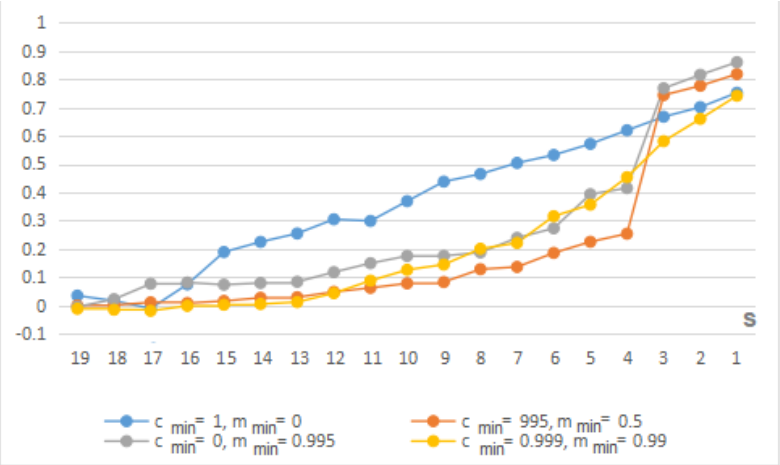


FIGURE 6. The reduction in total mining time when applying $AGRARM$ on $JM1$ and considering $s_{min} = 1$ and $(c_{min}, s_{min}) \in \{(1,0),\ (0.995,0.5),\ (0,0.995),\ (0.999,0.99)\}$

As we can see in Figures 5 and 6, the reduction in total mining time becomes substantial when the newly added attributes are relatively few. Consequently, the results of the experiments performed on Ar and $JM1$ also confirm the effectiveness of $AGRARM$, the algorithm we propose for adapting the interesting $GRARs$ mined before extension, so as to avoid applying $GRANUM$ from scratch on the extended data.

4.1. **Comparison to related work.** $AGRARM$, the adaptive mining approach introduced in Section 3, is new in the data mining literature. The existing approaches consider *non-relational* Association Rules and their adaptability refers to other aspects, except for $ARARM$, which handles *non-gradual* Relational Association Rules.

$AGRARM$ is an adaptation of $ARARM$ [8] so as to additionally consider the degree to which the rules are satisfied. This implies that the rules $AGRARM$ discovers as interesting are also filtered according to a given minimum *membership* threshold (see Function *isInteresting* in Section 3) in addition to support and confidence minimum thresholds.

Apart from $ARARM$, the perspectives of the other incremental mining approaches are quite different. Still, we will briefly present several recent approaches that are somehow related to our approach, since they focus on mining dynamic data. They are incremental in the sense that the dynamics of data refers to adding new instances and not new features to the existing instances.

Nath et al. [15] provides a survey on association rule mining, insisting on the situation in which the data set is not static. The authors have highlighted the important issues and challenges of mining dynamic data, including: the multiple passes over the

data set, the high number of generated candidates and the incremental behaviour of the data set.

Dhanabhakyam and Punithavalli [11] have proposed an efficient Market Basket Analysis mining method, called *Adaptive Association Rule Mining with Faster Rule Generation Algorithm* ($FRG - AARM$). The adaptability of the method refers to regulating the minimum support threshold during mining so as to attain a suitable number of rules.

Ogunde et al. [16] have introduced an *Adaptive Incremental Mining Algorithm* ($AIMA$). $AIMA$ has been designed to adapt the existing rules to the changes in the distributed databases, by mining, with the help of mobile agents, only the incremental database updates, in order to improve the response time and communication overhead.

A different incremental data mining algorithm has been proposed by Chang et al. [4]. The proposed method is based on FP-Growth and uses the concept of heap tree for incrementally updating the frequent itemsets.

A similar approach has been proposed by Yu-Dong et al [19]. The incremental association rule mining algorithm is called $PVSIFP - Growth$. The authors have incorporated in their proposal the *Improved FP-Growth* ($VSIFP - Growth$) and parallel computing based on $MapReduce$. $PVSIFP - Growth$ can discover association rules when both database increase or decrease and minimum support changes.

Li et al. [13] have proposed a *three-way decision update pattern approach* ($TDUP$) combined with a synchronization mechanism for efficiently updating and maintaining the frequent itemsets. It is based on using an additional support-based measure, so as to classify all possible itemsets into positive, boundary, and negative regions.

So, the existing adaptive approaches either rely on adapting the mining parameters [11] so as the discovered rules to be relevant, or aim the adaptation of the rules in the case of a dynamic data set, but which is extended vertically, not horizontally (i.e. by adding new data instances to it rather than adding new attributes to the existing instances) [19, 4, 13].

5. Conclusions and further work

We have proposed in the current paper $AGRARM$, a complete approach for adaptively uncovering the interesting Gradual Relational Association Rules within a dynamic data set that is extended by adding new features to it. Multiple experiments have been performed in order to comparatively evaluate $AGRARM$'s time performance. The evaluation results confirm that $AGRARM$ provided the interesting GRARs within the enlarged data more rapidly than resuming the mining algorithm $GRANUM$, i.e. applying it from scratch on the updated data set.

A first direction of further work is to further improve the efficiency of the adaptive mining process. To this effect, we aim to study possible algorithmic improvements of $AGRARM$ (like trying to generate a new candidate rule only from relevant pairs of rules, i.e. when at least one rule in the pair contains at least one newly added attribute) and also to develop a distributed version of it. We also plan to apply

AGRARM in concrete data mining tasks including incremental software defect prediction.

As an additional direction for further work, we plan to propose an adaptive-incremental approach for discovering interesting Gradual Relational Associations Rules within a dynamic data set to which both new features and new objects are added.

References

[1] Abdelhamid Boudane, Said Jabbour, Lakhdar Sais, and Yakoub Salhi. A SAT-based approach for mining association rules. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2472–2478. AAAI Press, 2016.

[2] Alina Câmpan, Gabriela Şerban, and Andrian Marcus. Relational association rules and error detection. *Studia Universitatis Babes-Bolyai Informatica*, LI(1):31–36, 2006.

[3] Alina Campan, Gabriela Şerban, Traian Marius Truta, and Andrian Marcus. An algorithm for the discovery of arbitrary length ordinal association rules. In *DMIN*, pages 107–113, 2006.

[4] H. Y. Chang, J. C. Lin, M. L. Cheng, and S. C. Huang. A novel incremental data mining algorithm based on FP-growth for big data. In *2016 International Conference on Networking and Network Applications (NaNA)*, pages 375–378, July 2016.

[5] Gabriela Şerban, Istvan Gergely Czibula, and Alina Câmpan. Medical diagnosis prediction using relational association rules. In *Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics (ICTAMI'07)*, pages 339–352, 2008.

[6] Gabriela Czibula, Maria-Iuliana Bocicor, and Istvan Gergely Czibula. Promoter sequences prediction using relational association rule mining. *Evolutionary Bioinformatics*, 8:181–196, 04 2012.

[7] Gabriela Czibula, Istvan Gergely Czibula, and Diana-Lucia Miholca. Enhancing relational association rules with gradualness. *International Journal of Innovative Computing, Communication and Control*, 13(1):289–305, 2017.

[8] Gabriela Czibula, Istvan Gergely Czibula, Adela-Maria Sîrbu, and Ioan-Gabriel Mircea. A novel approach to adaptive relational association rule mining. *Appl. Soft Comput.*, 36(C):519–533, November 2015.

[9] Gabriela Czibula, Zsuzsanna Marian, and István Gergely Czibula. Software defect prediction using relational association rule mining. *Inf. Sci.*, 264:260–278, 2014.

[10] Gabriela Czibula, Zsuzsanna Marian, and Istvan Gergely Czibula. Detecting software design defects using relational association rule mining. *Knowledge and Information Systems*, 42(3):545–577, Mar 2015.

[11] M. Dhanabhakyam and Punithavalli M. An efficient market basket analysis based on adaptive association rule mining with faster rule generation algorithm. *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA)*, 1(3), 2013.

[12] David J. Hand, Fergus Daly, K. McConway, D. Lunn, and E. Ostrowski. *A Handbook of Small Data Sets*, volume 1. CRC Press, 1993.

[13] Yao Li, Zhi-Heng Zhang, Wen-Bin Chen, and Fan Min. TDUP: an approach to incremental mining of frequent itemsets with three-way-decision pattern updating. *International Journal of Machine Learning and Cybernetics*, 8(2):441–453, Apr 2017.

[14] Diana-Lucia Miholca, Gabriela Czibula, and Istvan Gergely Czibula. A novel approach for software defect prediction through hybridizing gradual relational association rules with artificial neural networks. *Information Sciences*, 441:152 – 170, 2018.
[15] B. Nath, D. K. Bhattacharyya, and A. Ghosh. Incremental association rule mining: A survey. *Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(3):157–169, 2013.
[16] Adewale O. Ogunde, Olusegun Folorunso, and Adesina S. Sodiya. The design of an adaptive incremental association rule mining system. In *Proceedings of the World Congress on Engineering 2015 - Volume I*, London, UK, 2015.
[17] J. Sayyad Shirabad and T.J. Menzies. The PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada, 2005.
[18] Gabriela Serban, Alina Câmpan, and Istvan Gergely Czibula. A programming interface for finding relational association rules. *IJCCC*, I(S.):439–444, June 2006.
[19] Guo Yu-Dong, Li Sheng-Lin, Li Yong-Zhi, Wang Zhao-Xia, and Zeng Li. Large-scale dataset incremental association rules mining model and optimization algorithm. *International Journal of Database Theory and Application*, 9(4):195–208, 2016.

Department of Computer Science, Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Kogălniceanu 1, Cluj-Napoca, 400084, Romania
Email address: diana@cs.ubbcluj.ro