CBGP: Classification Based on Gradual Patterns

Yeow Wei Choong, Lisa Di Jorio, Anne Laurent, Dominique Laurent, Maguelonne Teisseire

CBGP: Classification Based on Gradual Patterns

Yeow Wei Choong
HELP University College
Dept. of IT
Kuala Lumpur - Malaysia
choongyw@help.edu.my

Lisa Di Jorio, Anne Laurent
LIRMM
Univ Montpellier 2 - CNRS
Montpellier - France
{dijorio,laurent}@lirmm.fr

Dominique Laurent
ETIS
Univ Cergy Pontoise - CNRS
Cergy Pontoise - France
dlaurent@u-cergy.fr

Maguelonne Teisseire
CEMAGREF
UMR Tetis
Montpellier - France
teisseire@teledetection.fr

***Abstract*—In this paper, we address the issue of mining gradual classification rules. In general, gradual patterns refer to regularities such as "The older a person, the higher his salary". Such patterns are extensively and successfully used in command-based systems, especially in fuzzy command applications. However, in such applications, gradual patterns are supposed to be known and/or provided by an expert, which is not always realistic in practice.**

In this work, we aim at mining from a given training dataset such gradual patterns for the generation of gradual classification rules. Gradual classification rules thus refer to rules where the antecedent is a gradual pattern and the conclusion is a class value.

***Keywords*-data mining; classification; gradual rules**

I. Introduction

Classification is a key problem in many real world applications that aim at determining the class of objects knowing their values over a given set of attributes. For instance, considering gene analysis applications, determining whether a tumor is malignant or not based on observed gene expressions is a key issue.

In this framework, many methods have been defined, *e.g.,* decision trees, naive bayes, neural networks, SVMs, (see [9], [17]), some of these methods being more explicative/understandable for the user than other ones. In particular, decision tree-based approaches, such as C4.5 ([18]), provide the user with a set of rules (so-called *classification rules*) that explicitly show why an example has been labeled with a particular class.

In this paper, we address the problem of computing classification rules when traditional approaches fail, because considering attribute values as such is not appropriate. In such a case, we look for *trends* shown by *sets* of values over the same attribute, such as *the higher/lower the value of attribute A, the higher/lower the value of attribute B.*

For example, in gene analysis, it could be the case that the values of gene expressions do not allow for any prediction regarding the malignancy of a tumor, whereas *sets* of such values contain trends like *the more gene $G1$ is expressed, the less gene $G2$ is expressed*, when the tumor being considered is malignant. Therefore, this type of gradual rule can be used to determine whether a given new tumor is malingnant.

For this purpose, we consider gradual patterns to describe trends in a given training dataset, thus leading to a novel and appealing rule-based classification method, extending existing methods such as CBA (Classification Based on Association rules) [14] or SPaC (Sequential Patterns for Classification) [11].

The paper is organized as follows: Section II introduces a running example considered throughout the paper to illustrate our approach. In Section III, we review previous work on gradual rules and rule-based classification methods. In Section IV, we introduce the basic definitions related to gradual frequent patterns and gradual classification rules. In Section V, we report on the experiments we have conducted to assess our method, and Section VI concludes the paper and provides further research directions.

II. Running Example

To illustrate our approach, we consider the (toy) example described in Table I, referring to fruit quality measurement as a mean to determine their provenance.

In this case, we assume that fruits, with properties size (S), weight (W) and sugar rate (SR), are delivered in box containers that have to be dispatched according to their provenance. Assuming that this provenance is ignored, we try and predict it, knowing that two provenances (*i.e.,* two classes) are possible, namely $North$ and $South.$

As argued next, it turns that in Table I, the size-, weight- and sugar rate values of fruits, considered one by one, do not allow for the prediction of the provenance, meaning that traditional classification approaches fail to predict fruit provenance based on their size, weight and sugar rate.

On the other hand, it will be seen shortly that, considering gradual patterns (*i.e., tendencies* of the size-, weight- and sugar rate values) allows to characterize the provenance.

As mentioned above, traditional classification methods do not apply in this example as no generality can be extracted tuple by tuple. Indeed, under Weka environment:

- Running $C4.5$ gives:

```
J48 pruned tree
------------------
: S (9.0/4.0)
```

Id	Size	Weight	Sugar Rate	Provenance
t_1	6	6	5.3	$North$
t_2	10	12	5.1	$North$
t_3	14	4	4.9	$North$
t_4	23	10	4.9	$North$
t_5	6	8	5.0	$South$
t_6	14	9	4.9	$South$
t_7	18	9	5.2	$South$
t_8	23	10	5.3	$South$
t_9	28	13	5.5	$South$

Table I
TRAINING SET: FRUIT PROVENANCES

Id	Size	Weight	Sugar Rate
n_1	4	7	5.3
n_2	18	14	5.0
n_3	22	6	4.9
n_4	16	9	6.1

Table II
TEST SET: NEW FRUITS

This means that the classifier predicts all fruit provenances be equal to South as the corresponding class contains most examples. The accuracy of this model (evaluated using cross validation) is very low, that is 11.11% (4 exceptions over 9 examples as reported in the brackets), meaning that this model fails to predict the provenance in 88.89% of the cases.

- Similarly, running a Naive Bayes method, a multilayer perceptron method and the KNN method gives a prediction accuracy of, respectively, 33%, 22.22% and 33.3333%.

Thus, clearly, whatever the method, attribute values as such can not determine fruit provenance, due to the fact that all fruits have almost the same size, weight and sugar rate (implying that standard classification methods cannot avoid over-fitting).

However, we argue that the way these attributes co-variate can help determining fruit provenance. Indeed, it can be seen from Table I that in general:

- For fruits from North, *the higher the size, the lower the sugar rate* (referring to the first four tuples); but no correlation involving the weight can be found.
- For fruits from South, *the higher the size, the higher the weight, the higher the sugar rate* (referring to the last five rows except tuple t_2).

In our apporach, we formalize these tendencies through the following two gradual classification rules:

- $\{(S, \geq), (SR, \leq)\} \Rightarrow North$, and
- $\{(S, \geq), (W, \geq), (SR, \geq)\} \Rightarrow South$

Consider now a new set of fruits, known to have the same provenance, but which has to be determined. The measurements of these fruits, reported in Table II, show that the pattern *the higher the size, the lower the sugar rate* holds. As this gradual pattern holds in Table I for northern fruits and not for southern fruits, the fruits described in Table II are considered as northern fruits, which can *not* be found using any traditional classification approach, as shown above.

III. RELATED WORK

A. Gradual Rules

Gradual patterns and gradual rules have been studied for many years in the framework of control, command and recommendation. More recently, data mining algorithms have been studied in order to automatically mine such patterns [3], [4], [5], [7], [10].

The approach in [10] uses statistical analysis and linear regression in order to extract gradual rules. In [3], the authors formalize four kinds of gradual rules in the form *The more/less X is in A, then the more/less Y is in B*, and propose an Apriori-based algorithm to extract such rules. However, frequency is computed from pairs of objects, increasing the complexity of the algorithm. Despite a good theoretical study, the algorithm is limited to the extraction of gradual rules of length 3.

The approach in [7] is the first attempt to formalize gradual sequential patterns. This extension of itemsets allows for the combination of gradual temporality ("*the more quickly*") and gradual list of itemsets. The extraction is done by the algorithm GRaSP, based on generalized sequential patterns [16] to extract gradual temporal correlations.

In [4] and [5], two methods to mine gradual patterns are proposed. The difference between these approaches lies in the computation of the support: whereas, in [4], a heuristic is used and an approximate support value is computed, in [5], the correct support value is computed. It is important to note in this respect that, in the current paper, the method of [5] is used for the computation of frequent gradual patterns. We refer to Section IV for the corresponding formal definitions.

B. Classification Based on Rules

Classification has been extensively studied [17]. In this paper, we focus on classification based on rules as this is the method we aim at extending.

In [14] the authors propose a text categorization method based on association rules, called CBA. The method consists of two steps: (i) rule generation (CBA-RG), based on the well-known Apriori algorithm, and (ii) building the classifier (CBA-CB), based on generated rules.

In the first step, each assignment of a text to a category is represented by a pair $\rho = \langle condset, C_i \rangle$, called a ruleitem, and where $condset$ is a set of items and C_i is a class label. Each ruleitem ρ should be thought of as a rule $condset \rightarrow C_i$, whose support and confidence are defined as follows:

$$supp(\rho) = \frac{\text{\#texts from } C_i \text{ matching condset}}{\text{\#texts in } D}$$

$$conf(\rho) = \frac{\text{\#texts in } C_i \text{ matching condset}}{\text{\#texts in } D \text{ matching condset}}$$

Ruleitems whose support is above a given threshold are said to be frequent, and the set of class association rules (CARs) consists of all frequent ruleitems whose confidence is greater than a minimum confidence threshold. Once all CARs are generated, they are totally ordered according to the following precedence relation: If r_i and r_j are two CARs, $r_i \prec r_j$ (precedes) if:

- either $conf(r_i) > conf(r_j)$
- or $conf(r_i) = conf(r_j)$ and $supp(r_i) > supp(r_j)$
- or $conf(r_i) = conf(r_j)$ and $supp(r_i) = supp(r_j)$ and r_i has been generated before r_j.

Let R be the set of CARs and D the training data. The basic idea of the algorithm is to choose a set of high precedence rules in R to cover D. The categorizer is represented by an ordered list of rules $r_i \in R$, leading to pairs such as $\langle (r_1, r_2, ..., r_k), C_i \rangle$, where C_i is a target category.

Each rule is then tested on D. If a rule does not improve the accuracy of the classifier, this rule and the following ones are discarded from the list.

Once the categorizer has been built, each ordered rule $r_i \in R$ is tested on every new text to classify. As soon as the condset part of a rule is supported by the text, the text is then assigned to the target class of the rule.

Many other research has been done using association rules for classification. In [12], the authors replace the confidence by the intensity of implication. In [13], the authors integrate the CBA method with other methods such as decision trees, Naive Bayes, RIPPER, etc. to increase the classification score. In [2], the authors investigate a method for association rule classification called L^3. Contrary to CBA-CB which takes only one rule into account, the authors propose to determine the category of a text by considering several rules, using a majority voting.

In [1], association rules are used for partial classification. Partial classification means that the classifier does not cover all cases. In particular, this work is interesting when dealing with missing values.

CAEP [6], LB, ADT [19], CMAR [15] are other existing classification systems using association rules. LB and CAEP are based on rule aggregation rather than rule selection. ARC-CB [20] proposes a solution for multi-classification (*i.e.,* a text is associated to one or more classes).

To end the section, we emphasize that none of these classification approaches have considered using gradual patterns, as we do in this paper.

IV. CBGP: Classification Based on Gradual Patterns

In this section, we detail our proposal for Classification Based on Gradual Patterns. We recall that our approach is composed of two parts: first we compute the gradual patterns representing each class value (thus constructing the gradual pattern-based classifier), and then we define how to determine the class values of new examples. We recall that the notions of gradual itemsets and their support are borrowed from [5]; these definitions are stated next.

A. Basic Definitions

In this work, we assume that we are given a database DB that consists of a single table whose tuples are defined on the attribute set $\mathcal{I}$. In this context, gradual patterns are defined to be subsets of $\mathcal{I}$ whose elements are associated with an ordering, meant to take into account increasing or decreasing variations.

In the definitions given below, using standard notation from the relational database model, for each tuple t in DB and each attribute I in $\mathcal{I}$, $t[I]$ denotes the value of t over attribute I.

Definition 1: (Gradual Itemset) Given a table DB over the attribute set $\mathcal{I}$, a gradual item is a pair (I, θ) where I is an attribute in $\mathcal{I}$ and θ a comparison operator in $\{\geq, \leq\}$.

A gradual itemset $g = \{(I_1, \theta_1), ..., (I_k, \theta_k)\}$ is a set of gradual items of cardinality greater than or equal to 2.

We point out that gradual itemsets are assumed to be sets of cardinality greater than or equal to 2, because sorting tuples according to one attribute is always possible, which is not the case when considering more than one attribute.

The support of a gradual itemset in a database DB is defined as follows.

Definition 2: (Support of a Gradual Itemset) Let DB be a database and $g = \{(I_1, \theta_1), ..., (I_k, \theta_k)\}$ a gradual itemset. The cardinality of g in DB, denoted by $\lambda(g, DB)$, is the length of the longest list $l = \langle t_1, \ldots, t_n \rangle$ of tuples in DB such that, for every $p = 1, \ldots, n-1$ and every $j = 1, \ldots, k$, the comparison $t_p[I_j]\, \theta_j\, t_{p+1}[I_j]$ holds.

The support of g in DB, denoted by $supp(g, DB)$, is the ratio of $\lambda(g, DB)$ over the cardinality of DB, which we denote by $|DB|$. That is, $supp(q, DB) = \frac{\lambda(g,DB)}{|DB|}$.

In order to compute $\lambda(g, DB)$, [5] proposes to consider the graph where nodes are the tuples from DB and where there exists a vertex between two nodes if the corresponding tuples can be ordered according to g.

Example 1: Referring back to our running example, since we aim at building rules to determine fruit provenance, the table DB to be considered for the definition of gradual itemsets is the projection of Table I over the attributes S, W and SR. In other words, we consider that $\mathcal{I} = \{S, W, SR\}$.

Figure 1 shows the ordering of the tuples of DB, according to the gradual itemset $g = \{(S, \geq), (SR, \leq)\}$, whose intuitive meaning is *the higher the size, the lower the sugar rate*. As in this graph, the length of the longest totally ordered list of tuples is 5, and as DB contains 9 tuples, we have $supp(g, DB) = \frac{5}{9}$.

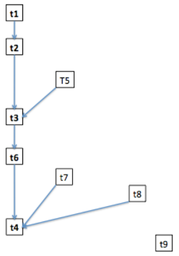

Figure 1. Graph of $g = \{(S, \geq), (SR, \leq)\}$ as computed on Table I

Consider now the tuples in DB whose Provenance value is $North$, *i.e.,* the first four tuples in DB. Denoting this set by $DB(North)$, we have $supp(g, DB(North)) = 1$.

B. Building the GP-Based Classifier

Considering a training database, *i.e.,* a set of tuples representing examples whose class is known, we aim at building a classification model based on frequent gradual itemsets, as done in CBA-like methods.

The two step processing works as follows: During the first step, frequent gradual itemsets are computed for each class. Then, during the second step, for every class, the corresponding gradual rules are defined based on the frequent gradual itemsets previously computed. The quality of these gradual rules is assessed, based on the notion of *discriminance of a given rule* that measures to which extent a gradual rule is representative of the class it has been extracted from.

More formally, let $\mathcal{C}$ be a set of class values, and $\mathcal{I}$ be a set of attributes, we aim at building a set of gradual classification rules in the form $g \Rightarrow c$ where g is a gradual itemset and $c \in \mathcal{C}$.

The training database, denoted by DB_T, is defined over the attribute set $\mathcal{I} \cup \{\mathcal{C}\}$. Intuitively, every tuple in DB_T is characterized by its values over attributes in $\mathcal{I}$ and its class value, stored as an attribute value.

Table I shows an example of such a training database, in which the set $\mathcal{C}$ of class values is $\{North, South\}$, seen as the domain of the attribute showing the provenance.

The table DB_T is partitioned according to the values in $\mathcal{C}$, and for every c in $\mathcal{C}$, we denote by $DB_T(c)$ the set of all restrictions over $\mathcal{I}$ of all tuples in DB_T where the class value is c. That is, $DB_T(c) = \{t[\mathcal{I}] \mid t \in DB_T \text{ and } t[\mathcal{C}] = c\}$.

Given an itemset g over $\mathcal{I}$, the *support of g in class c* is the support of g in $DB_T(c)$, that is, according to Definition 2, $supp(g, DB_T(c))$, which we denote by $supp(g, c)$ for the sake of simplification.

Given a minimum support threshold $minsup$, for every class value in $\mathcal{C}$, all gradual itemsets g such that $supp(g, c) \geq minsup$ are computed using the method of [5]. These gradual patterns are then considered in order to define whether they are discriminant for the class. The *discriminance measure* is defined as follows.

Definition 3: (Discriminance of a Gradual Itemset) Let g be a gradual itemset and c be a class value in $\mathcal{C}$. Denoting by $\Gamma(g)$ the set of all class values γ such that $supp(g, \gamma) \geq minsup$, the discriminance of g regarding c is defined as:

$$disc(g, c) = \frac{supp(g, c)}{\sum\limits_{\gamma \in \Gamma(g)} supp(g, \gamma)}.$$

Given a support threshold $minsup$ and a discriminance threshold $mindisc$, the problem of building a set of gradual classification rules amounts to computing, for every class value c in $\mathcal{C}$, the set of all rules of the form $g \Rightarrow c$ such that $supp(g, c) \geq minsup$ and $disc(g, c) \geq mindisc$.

In what follows, we denote by $\mathcal{G}_c$ the set of all gradual itemsets g such that $supp(g, c) \geq minsup$ and $disc(g, c) \geq mindisc$.

Example 2: Referring back to our running example, consider $minsup = 0.75$ and $mindisc = 0.25$.

It can be seen that for the class value $North$, $g = \{(S, \geq), (SR, \leq)\}$ is the only gradual itemset such that $supp(g, North) \geq minsup$. Therefore, the only rule to be considered in this case is $\{(S, \geq), (SR, \leq)\} \Rightarrow North$, and by Definition 3, $disc(g, North) = 1$. Thus $\mathcal{G}_{North} = \{g\}$.

For the class value $South$, the frequent gradual itemsets are $g_1 = \{(S, \geq), (SR, \geq)\}$, $g_2 = \{(S, \geq), (W, \geq)\}$, $g_3 = \{(W, \geq), (SR, \geq)\}$ and $g_4 = \{(S, \geq), (W, \geq), (SR, \geq)\}$, with respective supports in $DB_T(South)$ $\frac{4}{5}$, 1, $\frac{4}{5}$ and $\frac{4}{5}$. Thus, $\Gamma(South) = \frac{17}{5}$, and so, $disc(g_2, South) = \frac{5}{17} \geq 0.25$ and for $i = 1, 3$ or 4, $disc(g_i, South) = \frac{4}{17} < 0.25$. Therefore, $\mathcal{G}_{South} = \{g_2\}$.

Gradual classification rules are used for the classification of new examples, as shown below.

C. Classification of New Examples

Considering the classification rules constructed in the previous step, we address the issue of labelling new examples where the class values are unknown. We consider in this respect that we are provided with:

1) either a set of tuples known to be of the same class,
2) or a single tuple.

Notice that, in the second case above, a set of tuples not known to be of the same class can be considered. In this case, the tuples are processed one by one.

1) Set of Tuples of the Same Class: Let DB_{New} be a set of tuples, known to be of the same class that has to be determined. In this case, we first extract all frequent gradual itemsets from DB_{New}, considering the same $minsup$ threshold as above. Denoting this set by $\mathcal{G}_{New}$, in order to compute which class is the most appropriate, we compare

$\mathcal{G}_{New}$ to every $\mathcal{G}_c$. The class c_{New} issued by our method is such that:

$$c_{New} = argmax_{c \in \mathcal{C}} \left(\sum_{g \in \mathcal{G}_{New}} (disc(g, c)) \right)$$

where $disc(g, c) = 0$ if $g \notin \mathcal{G}_c$

Example 3: Let us consider the set of fruits as described in Table II, where the common provenance is ignored. It is easy to see that $\mathcal{G}_{New}$ contains the only gradual itemset $g = \{(S, \geq), (SR, \leq)\}$ with support 0.75. Since $g \in \mathcal{G}_{North}$ and $g \notin \mathcal{G}_{South}$, we obtain $c_{New} = North$, meaning that the fruits are considered from the North.

2) Single Tuple: We now assume that we are provided with a single tuple t defined over $\mathcal{I}$. Computing the class of t is achieved as follows: for every class value c and every g in $\mathcal{G}_c$, we compute to which extent t complies with g, as described in Algorithm 1. The criterion introduced to this end, called *relevance of* c, is denoted by $rel(c)$.

```
Data: $t$ a tuple defined over $\mathcal{I}$ ;
Result: $c_{New}$ ;
foreach $c \in \mathcal{C}$ do
    $rel(c) \leftarrow 0$ ;
    foreach $g \in \mathcal{G}_c$ do
        $s \leftarrow supp(g, DB_T(c) \cup \{t\})$ ;
        if $s \geq supp(g, DB_T(c))$ then
            $rel(c) \leftarrow rel(c) + disc(g, c)$
        end
    end
end
$c_{New} \leftarrow argmax_{c \in \mathcal{C}}(rel(c))$ ;
```

Algorithm 1: Classifying a new tuple

It should be noticed that computing $supp(g, DB_T(c) \cup \{t\})$ can be performed incrementally provided that the ordering graphs are stored when constructing the frequent gradual itemsets of $\mathcal{G}_c$. In this case, the computation simply amounts to inserting the new tuple within the graph. It is easy to see that, if the insertion results in a longer maximal ordered list then $s \geq supp(g, DB_T(c))$, else the test fails.

Example 4: Referring back to our running example, for the thresholds $minsup = 0.75$ and $mindisc = 0.25$, let $n = (12, 14, 5.0)$ (over attributes S, W and SR) be a new tuple to be classified.

As seen in Example 2, for the class value $North$, $\mathcal{G}_{North} = \{g\}$ where $g = \{(S, \geq), (SR, \leq)\}$ and $disc(g, North) = 1$. Since $supp(g, DB_T(North) \cup \{n\}) = 1$ (by inserting n between t_2 and t_3), we obtain that $rel(North) = 1$.

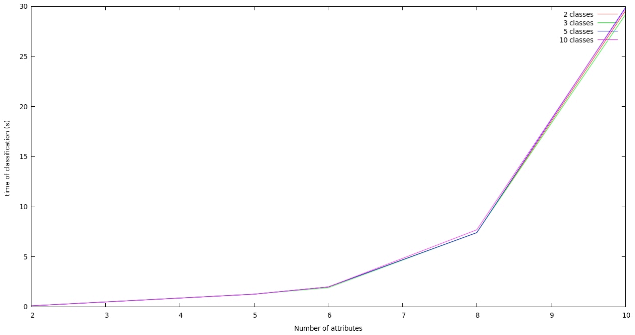

Figure 2. Runtime for Classifying New Tuples With Respect to the Number of Attributes

For the class value $South$, it has been seen in Example 2 that $\mathcal{G}_{South} = \{g_2\}$ where $g_2 = \{(S, \geq), (W, \geq)\}$, $supp(g_2, South) = 1$ and $disc(g_2, South) = \frac{5}{17}$.

Moreover, $supp(g_2, DB_T(South) \cup \{n\}) = \frac{5}{6}$ (since n cannot be inserted in $DB_T(South)$ so as to preserve the ordering according to g_2). Thus, $supp(g_2, DB_T(South) \cup \{n\}) < supp(g_2, DB_T(South))$, showing that $rel(South) = 0$. Hence, $c_{New} = North$.

V. Experiments

In this section, we report on preliminar experiments conducted to assess our approach on synthetic data sets. We show that the runtime for classification new examples is not prohibitive, assuming that gradual rules have been mined.

To this end, the following criteria have been considered, in the case where a set of 500 new examples known to be of the same class value is considered: the size of the database, the number of class values, and the number of attributes.

Figure 2 reports on the runtime for classifying a set of 500 new tuples regarding the number of attributes. We notice that the number of underlying class values in $\mathcal{C}$ has a very low impact on runtime, showing that the runtime is constant whatever the number of class values.

On the other hand, it can be seen that, for up to 6 attributes, runtime is far below 5 seconds, and that, for 6 up to 8 attributes, runtime does not exeed 10 seconds. It should be noticed in this respect that, as reported in [3], most existing approaches for mining gradual rules encounter difficulties when handling more than 6 attributes.

Figure 3 reports on the runtime regarding the size of the training database DB_T, when classifying 500 new examples. In particular, this figure shows that for less than 10 attributes in $\mathcal{I}$, the increase of runtime is not significant (since less than 10 seconds for up to 5000 tuples in DB_T), whereas for 10 attributes (and more) this increase is more important.

Clearly, more experiments are needed in order to better assess our approach. In particular, we plan to consider real data sets collected from satellite pictures for the classification of land areas, according to the type of farming. Indeed,

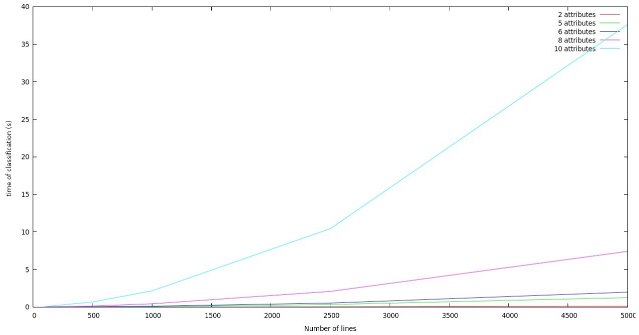

Figure 3. Runtime for Classifying New Tuples With Respect to the Size of the Database

it has been identified that standard classification methods do not perform satisfactorily in this context, and it is expected that gradual classification rules can give relevant results.

VI. Conclusion

In this paper, we have proposed a novel classification method based on gradual rules. In this context, we have provided an approach for computing gradual rules and for assigning a class value to new examples. The experiments reported in this paper show that our approach is tractable.

Further work include conducting more experiments to better assess the performance and the relevance of our approach. We also intend to investigate other measures to determine to what extent a new tuple t complies with a gradual pattern g. The heuristic presented in [4] will be investigated in this respect, as well as the computation of the number of attributes that prevent a tuple t to be inserted according to the ordering specified by a given gradual itemset. We also intend to study how to efficiently maintain the discriminence updated when new examples are inserted in the training database.

Acknowledgment

The authors acknowledge Fabien Millet, a Master Student at the Polytech'Montpellier Engineering School, University of Montpellier 2 for his work in implementing and testing the methods and algorithms presented in this paper.

References

[1] K. Ali, S. Manganaris and R. Srikant. Partial Classification Using Association Rules. In *Knowledge Discovery and Data Mining*, 1997.

[2] E. Baralis and P. Garza. Majority Classification by Means of Association Rules. In Proc. *European Conf. on Principles and Practice of Knowledge Discovery in Databases* (PKDD), 2003.

[3] F. Berzal, J.C. Cubero, D. Sanchez, M.A. Vila and J.M. Serrano. An Alternative Approach to Discover Gradual Dependencies. In *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* (IJUFKS), 15(5), 2007.

[4] L.D. Jorio, A. Laurent and M. Teisseire. Fast Extraction of Gradual Association Rules: A heuristic based method. In Proc. *Int. Conf. on Soft Computing as Transdisciplinary Science and Technology* (CSTST), 2008.

[5] L.D. Jorio, A. Laurent and M. Teisseire. Mining Frequent Gradual Itemsets from Large Databases. In Proc. *Intelligent Data Analysis Conf.* (IDA), LNCS, Springer-Verlag, 2009.

[6] G. Dong, X. Zhang, L. Wong and J. Li. CAEP: Classification by Aggregating emerging patterns. In Proc. *Int. Conf. on Discovery Science* (DS'99), LNAI 1721, Springer-Verlag, 1999.

[7] C. Fiot, F. Masseglia, A. Laurent and M. Teisseire. Gradual Trends in Fuzzy Sequential Patterns. In Proc. *Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems* (IPMU), 2008.

[8] S. Galichet, D. Dubois and H. Prade. Categorizing Classes of Signals by Means of Fuzzy Gradual Rules. In Proc. *Int. Joint Conf. on Artificial Intelligence* (IJCAI), 2003.

[9] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufmann Pub., 2000.

[10] Eyke Hüllermeier. Association Rules for Expressing Gradual Dependencies. In Proc. *European Conf. on Principles of Data Mining and Knowledge Discovery* (PKDD), 2002.

[11] S. Jaillet, A. Laurent, M. Teisseire. Sequential Patterns for Text Categorization. in *Int. Journal of Intelligent Data Analysis*, 10(3), 2006.

[12] D. Janssens, G. Wets, T. Brijs, K. Vanhoof and G. Chen. Adapting the CBA-Algorithm by Means of Intensity of Implication. In Proc. *Int. Conf. on Fuzzy Information Processing Theories and Applications*, 2003.

[13] B. Liu, Y. Ma and C.K. Wong. Classification Using Association Rules: Weaknesses and Enhancements. In *Data mining for scientific application*, 2001.

[14] B. Liu, W. Hsu and Y. Ma. Integrating Classification and Association Rule Mining. In *Knowledge Discovery and Data Mining*, 1998.

[15] W. Li, J. Han and J. Pei. CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. In Proc. *Int. Conf. on Data Mining* (ICDM), 2001.

[16] F. Masseglia, P. Poncelet and M. Teisseire. Pre-processing Time Constraints for Efficiently Mining Generalized Sequential Patterns. In Proc. *Int. Symposium on Temporal Representation and Reasoning* (TIME '04), 2004.

[17] Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.

[18] J. Quinlan. *C4.5 - Programs for Machine Learning.* Morgan Kaufman, 1993.

[19] K. Wang, S. Zhou and Y. He. Growing Decision Trees on Support-Less Association Rules. In *Knowledge Discovery and Data Mining*, 2000.

[20] M.L. Antonie and O.R. Zaiane. Text Document Categorization by Term Association. In Proc. *Int. Conf. on Data Mining* (ICDM), 2002.