# Fast Extraction of Gradual Association Rules: A Heuristic Based Method

Lisa Di Jorio, Anne Laurent, Maguelonne Teisseire

# Fast Extraction of Gradual Association Rules: A Heuristic Based Method

Lisa Di Jorio*
LIRMM – CNRS
161 rue Ada
34392 Montpellier Cedex 5 - France
dijorio@lirmm.fr

Anne Laurent†
LIRMM – CNRS
161 rue Ada
34392 Montpellier Cedex 5 - France
laurent@lirmm.fr

Maguelonne Teisseire†
LIRMM – CNRS
161 rue Ada
34392 Montpellier Cedex 5 - France
teisseire@lirmm.fr

## ABSTRACT

Even if they have proven to be relevant on traditional transactional databases, data mining tools are still inefficient on some kinds of databases. In particular, databases containing discrete values or having a value for each item, like gene expression data, are especially challenging. On such data, existing approaches either transform the data to classical binary attributes, or use discretisation, including fuzzy partition to deal with the data. However, binary mapping of such databases drives to a loss of information and extracted knowledge is not exploitable for end-users. Thus, powerful tools designed for this kind of data are needed. On the other hand, existing fuzzy approaches hardly take gradual notions into account, or are not scalable enougth to tackle the problem.

In this paper, we thus propose a heuristic in order to extract tendencies, in the form of gradual association rules. A gradual rule can be read as "*The more X and the less Y, then the more V and the less W*". Instead of using fuzzy sets, we apply our method directly on valued data and we propose an efficient heuristic, thus reducing combinatorial complexity and scalability. Experiments on synthetic datasets show the interest of our method.

## Categories and Subject Descriptors

H.2.8 [**Data mining**]: Miscellaneous

## Keywords

Gradual Rules, Data Mining, Trends

## 1. INTRODUCTION

Data mining aims at helping users to extract frequent patterns from large datasets. Many kinds of schemas have been proposed, such as the well known association rules [1], providing confidence and frequency information. Association rules can be written as "$X \Rightarrow Y$" (Freq%, Conf%) where $X$ and $Y$ are disjoint sets of attributes. "*Freq*" measures the number of occurrences of $X \cup Y$ in the entire database, and "*Conf*" is the probability to obtain $Y$ when $X$ occurs.

These first methods were originally designed to fit binary attributes. However, with the evolution of storage tools, most of the databases do not only contain binary attributes, but rather discrete values, such as quantity values (in a supermarket, for example) or observation measures (for example, sensor readings).

Thus, new challenges are raised: *how to integrate this kind of attributes? How can we represent them without loosing information?* Fuzzy logic plays an important role to resolve quantity and uncertainty problems. As these methods are successful in data mining, new works taking new structures into account have raised. These last years, we have seen the apparition of proposals dealing with the notion of "*gradual values*"[4]. Most of them plug gradual approaches into data mining algorithms, in order to extract "*gradual association rules*" . We consider here gradual rules in the form "*the more / the less*", such as "*the higher the age, the higher the pay*" or "*the older a subject, the less his memory*".

These powerful structures can be applied in a wide range of domains. Among them are the marketing datasets ("*the more increase of champaign sales, the more increase of caviar sales*"), sensor readings, and medical databases (gene databases, symptom databases, etc.) where attributes are often quantitative (as for gene expressions database).

In this paper, we are interested in automatically and efficiently extracting gradual association rules. We propose an heuristic, based on local set optimization. The rest of this paper is organized as follows: first, we describe existing work on gradual association rule extraction. In Section 3, we present our definition of gradual. Then Section 4 shows some experiments. Finally, we conclude after a brief discussion.

## 2. RELATED WORK

As mentioned previously, fuzzy logic plays an important role in quantitative data mining. In set theory, an item belongs to a given set or to its complement. Such a system cannot deal with quantitative values, as we will only consider a presence of an item for a given object.

In fuzzy logic, an item can gradually belong to several sets, according to a membership function. Semantically, the

*Phd Student
†Assistant Professor


*CSTST 2008* October 27-31, 2008, Cergy-Pontoise, France

membership degree denotes the idea of "*more or less*". For example, instead of being only cheap or expensive, a product can be considered as mainly cheap and a little bit expensive. For instance, the object $o_2$ of the second row from table 1 is mainly considered as a cheap object, but its prize is a little bit expensive. Thus, one can define fuzzy sets on the domain of a given item. Continuing the prize example, we introduce two fuzzy sets: in Figure 2, a product is considered as totally cheap up to 30 Euro, and starting to be expensive above 30 Euro. From 40 Euro, we consider the prize as expensive. Then, each item value of the database can be transformed as a membership degree to each corresponding fuzzy set, as shown in Table 1.

| Obj | Pr. | Ch. | Exp. |
|---|---|---|---|
| $o_1$ | 20 | 1 | 0 |
| $o_2$ | 33 | 0.75 | 0.25 |
| $o_3$ | 35 | 0.5 | 0.5 |
| $o_4$ | 60 | 0 | 1 |

Figure 1: Database Sample

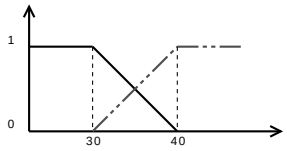



Figure 2: Fuzzy Sets

In the general case, fuzzy association rule extraction is done through an extension of classical rules extraction algorithms. The main difference lies in the frequency definition: the frequency of an itemset $XY$ is defined on the logical conjunction between $X$ and $Y$, which can be expressed through a t-norm operator. A t-norm expresses the membership degree of $X$ and $Y$ together in a given fuzzy set. In a fuzzy extraction rule process, minimum operator ($min(X, Y)$) and Lukasievez operator ($max(X+Y-1, 0)$) are commonly used. A fuzzy association rule is then of the form ($[X, A] \Rightarrow [Y, B]$), where A and B are fuzzy sets defined on X and Y domain respectively. This rule can be read as "*X is in A implies Y is in B*".

Fuzzy sets are usually defined by the user. Thus, fuzzy association rules are an interface between the user and the database, as extracted knowledge will be based on user-understandable fuzzy sets. Moreover, fuzzy rules have a great expression power, as they give a linguistic sense to attribute quantities. Therefore, fuzzy rule extraction has been widely studied since these last years. But fuzzy theory does not restrict itself to a membership degree meaning "*X is in A*". Indeed, fuzzy logic allows to integrate linguistic modifiers, like "*almost*" or "*more or less*". More recently, with the apparition of performants algorithms, a particular attention has been given to gradual expression extraction, using fuzzy set.

According to Zadeh, the "*transition from non-membership to membership is gradual rather than abrupt*". Noticing that gradualness is still missing part from fuzzy theory, [5] introduces a formalization of the so-called "*gradual element*". [5] shows some possible applications of such elements, mainly in fuzzy logic theory, like fuzzy cardinality or defuzzyfication. However, gradual dependencies between fuzzy sets are not really evoked in this paper.

Some works explore more deeply the notion of gradual rules. Rescher-Gaines implication is employed in order to measure gradualness ($A(X)$ is the membership degree of $X$ in $A$):

$$X \rightarrow_{RG} Y = \begin{cases} 1 & if\ A(X) \leq B(Y) \\ 0 & else \end{cases}$$

However, this kind of implication is too restrictive: value of $A(X)$ is ruled by $B(Y)$ value, giving thus 1 if $B(Y)$ increases. By binarizing values, Rescher-Gaines does not really measure a variation of $X$ value and $Y$ value. Moreover, it is a challenging issue to combine more than two items (see [8] for a complete study) in the premise and the conclusion using a Rescher-Gaines implication. To overcome this problem, [3] proposes to mine rules having only one item in the conclusion. The problem of managing several attributes is resolved by using a t-norm operator in the premise of the implication. This approach raises two kinds of problems. Firstly, a study of extracted rules shows that not all the rules are coherent on their semantic interpretation. For example, in some case, two rules are contradictory. Secondly it is not possible to combine increasing and decreasing variations (for example "*when age increases and performance decreases, then the number of fired persons increases*").

[7] uses statistical analysis in order to extract gradual rules. In the non fuzzy case, association can be represented by the mean of *contingency tables*. [7] adapt these one to the fuzzy context by the mean of a *contingency diagram*. Then, linear regression is used in order to derive gradual rules. Afterwards, a quality measure keeps the more interesting rules. This approach brings a new point of view, but cannot be directly adapted to a classical algorithm, as linear regression could quickly become a bottleneck. [7] offers a good formalization and notices that an extraction of positive and negative trends could result in a redundancy information.

Starting from this last observation, [2] formalizes four kinds of gradual rules of the form "*The more / less X is in A, then the more / less Y is in B*", and proposes an Apriori-based [1] algorithm to extract them. 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.

Finally, [6] is the first to formalize gradual sequential patterns. This extension of association rules 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 [9] to extract gradual temporal correlations.

All these approaches extract gradual rules from quantitative databases using fuzzy membership degree. In this paper, we simply use order relation directly on the values instead of membership degrees. Moreover, this method overcomes the problem of Rescher-Gaines conjunction, and extracts more relevant rules, as premise of the rule will not be restricted on the conclusion. Thus, we are able to plug new definitions to classical algorithms in order to be scalable. Defining increasing and decreasing items, allows us to combine two kinds of items, and to extract gradual rules of length n.

# 3. OUR APPROACH

## 3.1 Definitions

In this paper, we consider gradual rules like "*when X varies, then Y varies*". We consider a database $\mathcal{DB}$ containing a set of objects $\mathcal{O}$ and a set of items $\mathcal{I}$. Each row represents a transaction $t$ for a corresponding object, and $t[i]$ denotes the value associated to the item $i$. A sample database is displayed on Table 1 with a set of eight persons

with their age, salary and number of cars. For example, the person described by object $o_1$ is 22 years old, earns 1,200 Euro a month, and has one car. From this kind of database, we wish to extract rules like "*The older the person, the higher the salary*".

Our objective is to use a classical algorithm for association rules extraction. There are two main paradigms to extract association rules: *pattern-growth* approach, and *generate and prune* approach. Their efficiency is similar, even if pattern-growth approach has been empirically proved to be more efficient than generate and prune. In our case, this approach can be used only if gradual items and gradual itemsets are clearly defined. So, gradualness for a given item $i$ denotes two possible variations on its domain of values:

- The value increases. In this case, we have a gradual item that can be interpreted by "*the more i*". We note it $i^+$, and use the $\geq$ operator to extract it.
- The value decreases. In this case, we have a gradual item that can be interpreted by "*the less i*". We note it $i^-$, and use the $\leq$ operator to extract it.

*Definition 1.* (gradual item) Let $i \in \mathcal{I}$ be an item and $* \in \{\geq, \leq\}$ a comparison operator. Then a gradual item $i^*$ is defined as an item $i$ associated to an operator $*$.

*Definition 2.* (gradual itemset) A gradual itemset is an non-empty set of gradual items. A $k$-itemset is a gradual itemset of length $k$, i.e. containing $k$ gradual items.

Note that operators $\{\geq, \leq\}$ are used, including the case when two values are equal. Thus ordered values are directly compared. In [2] a strict inequality is considered. In a classical way, frequency of an item is the number of transactions containing this item. In a gradual context, we have to compare each $t[i]$ and to select the ones respecting an increasing (or decreasing) variation. Thus, gradual mining automatically leads to an object comparison. There are some ways to achieve this, including a two-by-two comparison. Therefore, to find objects supporting a gradual itemset, [2] projects the database in a database of pairs. Thus, there is no loss of information due to equality. For example, let us consider the values for item "Car" in Table 1, and consider objects $o_6, o_7, o_8$. When looking for $Car^+$, [2] will construct six pairs: $\{(o_6, o_7), (o_6, o_8), (o_7, o_6), (o_7, o_8), (o_8, o_6), (o_8, o_7)\}$, and will only keep $\{(o_6, o_7), (o_6, o_8)\}$ as pairs respecting the increasing variation.

However, projecting the database into a pair database can be too memory consuming and will not allow for mining large datasets, as it leads to handle a database with $|\mathcal{O}|.(|\mathcal{O}|-1)$ objects. Consequently, we propose as an alternative the use of an ordered dataset.

*Definition 3.* Let $(i_1^{*_1} i_2^{*_2} ... i_n^{*_n})$ be a gradual itemset where $*_1 ... *_n \in \{+, -\}$. Let $G^{\mathcal{D}}$ be the transaction set ordered first on $i_1^{*_1}$, then on $i_2^{*_2}$... then on $i_n^{*_n}$. A transaction $t$ supports $(i_1^{*_1} i_2^{*_2} ... i_n^{*_n})$ if:

$$\forall t_j, t_k, j \neq k \begin{cases} t_j[i_1] *_1 t_k[i_1] \wedge \ ...t_j[i_n] *_n t_k[i_n] \ if \ t_j > t_k \\ \\ t_j[i_1]\neg *_1 t_k[i_1] \wedge \ ...t_j[i_n]\neg *_n t_k[i_n] \ else \end{cases}$$

Definition 3 allows to extract transactions which are gradual on an itemset. Note that we can construct more than one $G^{\mathcal{D}}$.

| Object | Age (A) | Salary (S) | Car (C) |
|---|---|---|---|
| $o_1$ | 22 | 1200 | 1 |
| $o_2$ | 28 | 1850 | 1 |
| $o_3$ | 24 | 1200 | 0 |
| $o_4$ | 35 | 2200 | 1 |
| $o_5$ | 38 | 2000 | 1 |
| $o_6$ | 44 | 3400 | 1 |
| $o_7$ | 52 | 3400 | 2 |
| $o_8$ | 41 | 5000 | 2 |

Table 1: A database $\mathcal{DB}$

| O | A | S |
|---|---|---|
| $o_1$ | 22 | 1200 |
| $o_3$ | 24 | 1200 |
| $o_2$ | 28 | 1850 |
| $o_5$ | 38 | 2000 |
| $o_8$ | 41 | 5000 |

Table 2: $A^+S^+$

| O | A | S |
|---|---|---|
| $o_1$ | 22 | 1200 |
| $o_3$ | 24 | 1200 |
| $o_2$ | 28 | 1850 |
| $o_4$ | 35 | 2200 |
| $o_6$ | 44 | 3400 |
| $o_7$ | 52 | 3400 |

Table 3: Other $A^+S^+$

For example, starting from the database shown on Table 1, we are looking for objects that support the gradual itemset $A^+S^+$. Clearly, keeping $o_4$ will not allow to keep $o_5$ as $t_{o_4}[P] > t_{o_5}[P]$. So, we can create two gradual sets: one containing $o_4$ and excluding $o_5$, and one keeping $o_5$. The same kind of contradiction is found for $o_8$. Among all possible $G^{\mathcal{D}}$, some contain more objects than others. These ones are thus considered as the more *representative* of the considered gradual itemset. Then frequency is defined by:

*Definition 4.* Let $s = (i_1^{*_1} i_2^{*_2} ... i_n^{*_n})$ be a gradual itemset and $G_s^{\mathcal{D}}$ be the set of all possibles $G^{\mathcal{D}}$ for $s$. The frequency of $s$ is given by:

$$Freq(s) = \frac{max(|G_s^{i\mathcal{D}}|)}{|\mathcal{O}|}$$

where $G_s^{i\mathcal{D}} \subset G_s^{\mathcal{D}}$.

As an illustration, let us calculate $Freq(A^+S^+)$. Among all the $G^{\mathcal{D}}$, one of the maximal is $\{o_1, o_2, o_3, o_5, o_6, o_7\}$. Then $Freq(A^+S^+) = \frac{6}{8} = 0.75$. It can be read as "*the more the age increases, the more the salary increases*". Note that the conclusion is not a consequence of the premise, i.e. an increasing age will not induce an increasing salary. At this stage, we are only talking about gradual itemsets, and not about gradual rules including causality. A gradual association rule is defined as follows:

*Definition 5.* Let $s_1$ and $s_2$ be two gradual itemsets such as $s_1 \cap s_2 = \emptyset$. A gradual association rule is of the form $R : s_1 \Rightarrow s_2$ with two associated measures:

- **frequency** is the frequency of all the gradual items: $Freq(R) = Freq(s_1 \cup s_2)$
- **confidence** measures the probability to have $s_2$ having $s_1$: $Conf(R) = \frac{Freq(s_1 \cup s_2)}{Freq(s_1)}$

All measures associated to a gradual rule are computed when considering the best way to organise and order the data in the best $G_s^{i\mathcal{D}}$. This maximal set is the core of the

algorithm. Finding classical association rules is done by growing the set of frequent itemsets. Our intuition is that gradual itemsets extraction can be done in a similar way. It is possible to use gradual $k$-itemsets to construct gradual $(k+1)$-itemsets. To apply this, we need to handle two challenges. Fistly, we have to find the a maximal $G^{\mathcal{D}}$ in order to compute the more representative frequency. Secondly, the join operation between two $G^{\mathcal{D}}$ have to be formally defined. However, this is not a trivial task: we have seen that we have to choose which element will be discarded from the original set. *Which one is the best?* In the following section, a heuristic based on maximal sets is proposed as a first solution.

## 3.2 Finding the Best Candidates

Our proposition is based on the following observation: some elements are conflicting with others, and keeping them leads to discard the others. So, we can easily make a list of the ones discarding more other objects. Therefore, we propose to keep a list of *conflicting set*, and to base our choices on this list. From it, we will be able to generate the maximal local $G^{\mathcal{D}}$.

### 3.2.1 2-itemset case

For the sake of simplicity, we first explain our method for gradual 2-itemsets, and then generalize it to $n$-itemsets. We define a conflicting set for a 2-itemset as:

*Definition 6.* Let $i_1^{*_1} i_2^{*_2}$ be a 2-itemset, and $\mathcal{O}$ a set of objects from $\mathcal{DB}$ ordered on $i_1$ according to $*_1$ and then on $i_2$ according to $*_2$. For an object $o_i \in \mathcal{O}$, we keep all objects discarded in a conflicting set, called $\mathcal{C}_i$. Namely, $\forall o_j \in \mathcal{C}_i, t_{o_i}[i_2] \neg *_2 t_{o_j}[i_2]$.

It is easy to see that an empty set $\mathcal{C}_i$ will mean that $o_i$ can participate to the frequency of the associated gradual 2-itemset, as it does not contradict the operator $*_2$. On the opposite, the bigger a $\mathcal{C}_i$ is, the more objects we will have to discard if we want to keep $o_i$. In other words, the conservation of such an object brings us to discard $|\mathcal{C}_i|$ other objects. In order to construct a representative set of objects associated to a gradual itemset, we first delete the ones having the maximal $\mathcal{C}_i$. Note that our structure is symmetric: if $o_i \in \mathcal{C}_j$ then $o_j \in \mathcal{C}_i$. In the rest of this paper, we call $\mathcal{C}$ the set containing all the conflicting sets for a gradual $n$-itemset.

On a first step, we keep all the objects having an empty conflict set: $t_0 = G^{\mathcal{D}} \leftarrow f_{emp}(\mathcal{C})$ (where $f_{emp}$ returns all objects having an empty conflicting set). Then, we discard the object having the biggest conflicting set using $f_{max}$ function: $t_1 = \mathcal{O} \setminus f_{max}(\mathcal{C})$. Deleting an object from the candidate set will delete it from the conflicting sets it was in. Actually, due to the symmetry of the structure, we only have to follow each object contained in the deleted $\mathcal{C}_i$. Thus, these two steps can be summarized into $t_{01} = G^{\mathcal{D}} \leftarrow f_{emp}(\mathcal{O} \setminus f_{max}(\mathcal{C}))$. Then, we repeat our process until obtaining only empty conflicting sets. This leads to a recursive formula:

*Proposition 1.* The recursive function $t_n = G^{\mathcal{D}} \leftarrow t_{n-1}$ with $t_0 = G^{\mathcal{D}} \leftarrow f_{emp}(\mathcal{O} \setminus f_{max}(\mathcal{C}))$ computes a maximal local representative set.

PROOF. Let us say that $|G^{\mathcal{D}}| = n$. Suppose that there is another representative set $\mathcal{F}$ such that $|\mathcal{F}| = m | m > n$. This means that there is an object $o_i$ in $\mathcal{F}$ and not in $G^{\mathcal{D}}$. Then $\mathcal{C}_i = \emptyset$. But, by construction, if $\mathcal{C}_i = \emptyset$, then $o_i \in G^{\mathcal{D}}$. It is thus impossible that $o_i \notin G^{\mathcal{D}}$. □

Let us illustrate Proposition 1 by calculating a representative $G^{\mathcal{D}}$ for $(A^+S^+)$. Ordering database from Table 1 on $A^+$ and then on $S^+$ gives the database shown on Table 4. We have calculated, for all the objects, the corresponding conflicting set, which can be viewed on the third column. For example, we can see that conserving $o_8$ means deleting $o_6$ and $o_7$, and symmetrically keeping $o_6$ and $o_7$ means discarding $o_8$.

| Object | A | S | $\mathcal{C}_i$ |
|---|---|---|---|
| $o_1$ | 22 | 1200 | $\emptyset$ |
| $o_3$ | 24 | 1200 | $\emptyset$ |
| $o_2$ | 28 | 1850 | $\emptyset$ |
| $o_4$ | 35 | 2200 | $\{o_5\}$ |
| $o_5$ | 38 | 2000 | $\{o_4\}$ |
| $o_8$ | 41 | 5000 | $\{o_6, o_7\}$ |
| $o_6$ | 44 | 3400 | $\{o_8\}$ |
| $o_7$ | 52 | 3400 | $\{o_8\}$ |

Table 4: Sorted $\mathcal{O}$ on $A^+$ then on $S^+$

In this example, $o_8$ is the object having the maximal conflicting set. During the first step, the operation $G^{\mathcal{D}} \leftarrow f_{emp}(\mathcal{O} \setminus o_8) \equiv G^{\mathcal{D}} \leftarrow \{o_1, o_3, o_2, o_6, o_7\}$ is done. Table 5 shows this first operation: discarding $o_8$ updates $o_6$'s and $o_7$'s conflicting sets. These sets become empty and can be added to the representative set.

| Object | A | S | $\mathcal{O}_{\mid i}$ |
|---|---|---|---|
| $o_1$ | 22 | 1200 | $\emptyset$ |
| $o_3$ | 24 | 1200 | $\emptyset$ |
| $o_2$ | 28 | 1850 | $\emptyset$ |
| $o_4$ | 35 | 2200 | $\{o_5\}$ |
| $o_5$ | 38 | 2000 | $\{o_4\}$ |
| ~~$o_8$~~ | ~~41~~ | ~~5000~~ | ~~$\{o_6, o_7\}$~~ |
| $o_6$ | 44 | 3400 | ~~$\{o_8\}$~~ $= \emptyset$ |
| $o_7$ | 52 | 3400 | ~~$\{o_8\}$~~ $= \emptyset$ |

Table 5: Operation $t_{01}$

Note that on the following step, $o_4$ or $o_5$ can be equally discarded as they are excluding each other. The final cardinality of the representative set will be the same, but we will discuss later about the consequences of this choice. Here, we discard the first one, thus obtaining $G^{\mathcal{D}} = \{o_1, o_3, o_2, o_5, o_6, o_7\}$ as a result.

### 3.2.2 n-itemset case

Using a *generate and prune* algorithm, it is easy to extend a gradual 2-itemset extraction to the general case. Actually, in a such algorithm, itemsets are generated level by level by the mean of an intersection between level $n$ and level $n-1$. In our case, a simple intersection between two representative sets cannot be performed. It can lead to an incorrect result, due to the gradual aspect of the method. However, a level-wise method brings us a great advantage: we can order objects starting from the second level, and keep this order level by level. In other words, the order found for a gradual $n$-itemset is the same for an $(n+1)$-itemset. Thus, we gain on the sort operation, which can be time-consuming.

*Definition 7.* Let $i_1^{*_1} ... i_n^{*_n}$ be a $n$-itemset, and $\mathcal{O}$ a set of objects from $\mathcal{DB}$ ordered on $i_1$ according to $*_1$ and then on $i_2$

according to $*_2$... and then on $i_n$ according to $*_n$. For an object $o_j \in \mathcal{O}$, we keep all objects discarded in two conflicting sets: one concerning item $i_{n-1}$ called $\mathcal{C}_{i_{n-1}}$ and one concerning $i_n$ called $\mathcal{C}^{i_n}$. So, $\forall o_k \in \mathcal{C}^{i_{n-1}}, t_{o_j}[i_{n-1}] \neg *_{n-1} t_{o_k}[i_{n-1}]$ and $\forall o_k \in \mathcal{C}_{i_n}, t_{o_j}[i_n] \neg *_n t_{o_j}[i_n]$.

The method is the same as before, except that we manage two conflicting sets to find objects having the maximal one. Our joining algorithm is given by Algorithm 1. It implements the recursive function given in proposition 1. The "While" loop makes the recursion, and $G^{\mathcal{D}}$ is constructed into the "if" condition. Function $f_{cnf} : \mathcal{O} \rightarrow \mathcal{C}$ associates a conflict set to an object.

**Algorithm 1**: n-SupportCount

**Data**: A g-itemset $s = (i_1^{*_1}...i_n^{*_n})$,
Set of objects $\mathcal{O}$ sorted according $n-1$ items,
Conflictual sets $\mathcal{C}^n$ and $\mathcal{C}^{n-1}$
**Result**: Representative $G^{\mathcal{D}}$ for $s$

$G^{\mathcal{D}} \longleftarrow \emptyset$
**while** $\mathcal{O} \neq \emptyset$ **do**
  $o = f_{max}(\mathcal{C}^{i_n}, \mathcal{C}^{i_{n-1}})$
  $\mathcal{O} \longleftarrow \mathcal{O} \setminus \{o\}$
  **foreach** $o_j \in \mathcal{O}$ **do**
    $f_{cnf}(o_j, \mathcal{C}^{i_n}) \longleftarrow f_{cnf}(o_j, \mathcal{C}^{i_n}) \setminus \{o\}$
    $f_{cnf}(o_j, \mathcal{C}_{i^{n-1}}) \longleftarrow f_{cnf}(o_j, \mathcal{C}^{i_{n-1}}) \setminus \{o\}$
    **if** $f_{cnf}(o_j, \mathcal{C}_{i_n}) = \emptyset$ *and* $f_{cnf}(o_j, \mathcal{C}^{i_{n-1}}) = \emptyset$ **then**
      $G^{\mathcal{D}} \longleftarrow G^{\mathcal{D}} + \{o_j\}$
      $\mathcal{O} \longleftarrow \mathcal{O} \setminus o_j$
    **end**
  **end**
**end**
**return** $\mathcal{O}_R$

## 3.3 Interesting Properties

Our proposition raises some interesting properties discussed in this section. First of all, we found a common property with [2] concerning the negation of an itemset. Order relations such as $\{\geq, \leq\}$ have a negation (or complementary) defined as $c$. Here $c(\geq) = \leq$ and $c(\leq) = \geq$. So, the negation of an itemset will be defined as follow:

*Definition 8.* Let $s = (i_1^{*_1}...i_n^{*_n})$ be an itemset. Then the negation of $s$, noted $c(s)$, is $(i_1^{c(*_1)}...i_n^{c(*_n)})$.

We thus have:

*Proposition 2.* (negative g-itemset) Let $s = (i_1^{*_1}...i_n^{*_n})$ be a g-itemset. If a set of objects $G^{\mathcal{D}}$ respects this g-itemset, then it respects $c(s) = (i_1^{c(*_1)}...i_n^{c(*_n)})$.

PROOF. $\forall o, p \in \mathcal{O}, o * p \Leftrightarrow p\ c(*)\ o$. This implies immediately that every object from $G^{\mathcal{D}}$ respects its complementary. □

*Corollary 1.* $Freq(s) = Freq(c(s))$

This means that only half of the gradual itemsets can be generated, as all the other part will be deduced from them. This leads to an important time and memory optimization.

In our proposition, gradualness is expressed through a total order relation. Thus, whatever the 1-g-itemset considered, every object of the database will participate to its representative set, as every object is comparable. So, the frequency of a 1-g-itemset will always be 1 (100%). A 1-g-itemset does not bring a great expressive power (having only that "*A increases*" for 100% of the database is not useful: we know that every person age's can be ordered). Moreover, as our proposition is based on an **object-to-object** comparison, there is no semantic explanation of a 1-g-itemset. So, we will start the generation of representative sets from the second level (i.e., from 2-g-itemsets).

The confidence is based on frequencies of g-itemsets. We know that $\forall i \in \mathcal{I}, Freq(i^+) = Freq(i^-) = 1$. However, for a rule deduced from a 2-g-itemset:

- $Conf(i_1^{*_1} \Rightarrow i_2^{*_2}) = \frac{Freq(i_1^{*_1} i_2^{*_2})}{Freq(i_1^{*_1})}$
- $Conf(i_2^{*_2} \Rightarrow i_1^{*_1}) = \frac{Freq(i_1^{*_1} i_2^{*_2})}{Freq(i_2^{*_2})}$

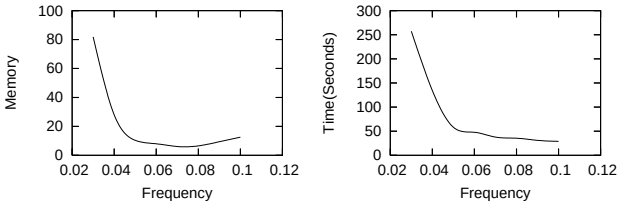
As $Freq(i_1^{*_1}) = Freq(i_2^{*_2})$, we obtain $Conf(i_1^{*_1} \Rightarrow i_2^{*_2}) = Conf(i_2^{*_2} \Rightarrow i_1^{*_1}) = Freq(i_1^{*_1} i_2^{*_2})$. Thus, it is impossible to establish the most significant implication of the rule for a rule of length 2. We start the gradual association rule generation from the third level.

# 4. EXPERIMENTS

Our approach has been implemented in C++ as C++ allows a deep memory management.

We ran our algorithm on synthetic datasets, in order to measure memory and execution performances. We used the IBM Synthetic Data Generation Code for Associations and Sequential Patterns[1] in order to generate synthetic datasets. However, IBM Generator was designed for association rules, and therefore generates datasets in a presence or absence form. So, we used a simple random in order to assign a numerical value to a given item. Zero values mean "*this item is not present in this transaction*". As we use equality, zero values can participate in the frequency computation. However, as we consider them as abscence values, they are thus ignored by the program.

IBM Generator allows to choose a good number of important parameters, among them the number of transactions and their average size. Intuitively, as g-itemset calculation is based on the value from one transaction to another for the same itemset, if we want to generate some gradual rules, we need to generate databases with transaction having most of the set $\mathcal{I}$ of items. This kind of bases can be clearly compared to gene expression databases.



(a) memory (b) time

Figure 3: 1000 transaction and 100 items performances

Tests are very promising: for databases containing about 1,000 large transactions, the execution takes some seconds,

[1] www.almaden.ibm.com/software/projects/hdb/resources.shtml

and has a good reaction to a very low support (0.005%) as show Figures 3a and 3b.

## 5. DISCUSSION

One of the drawbacks of the proposed here approach is that, as we use a heuristic, it could be the case that some rules are not extracted. In fact, each time we have more than one maximal conflicting set, we choose one of them. There are several manners to do this choice (choosing the first one, choosing by random, etc.). However, whatever this choice, the frequency returned is always lower or equal to the real one, due to the level-wise aspect of our algorithm. For example, considering the database from Table 6, constructing g-itemset $(A^+B^+)$ will discard $\{o_x, o_y, o_z\}$, as they are contradictory with all the others. However, for $(A^+B^+C^+D^+E^+)$, $\{o_x, o_y, o_z\}$ is the best set. Our method will choose the other solution and find in the end $\{o_1, o_4\}$.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| $o_1$ | 3 | 1 | 3 | 3 | 1 |
| $o_2$ | 4 | 2 | 1 | 4 | 2 |
| $o_3$ | 5 | 3 | 4 | 1 | 3 |
| $o_4$ | 6 | 4 | 3 | 5 | 4 |
| $o_5$ | 7 | 1 | 5 | 2 | 5 |
| $o_6$ | 8 | 1 | 6 | 6 | 1 |
| $o_x$ | 1 | 20 | 10 | 15 | 10 |
| $o_y$ | 2 | 30 | 20 | 40 | 20 |
| $o_z$ | 2.5 | 40 | 30 | 50 | 30 |

Table 6: A Problematic Database

However, in such case, *how to choose the one to discard?* It is important to highlight that if discarding $o_i$ instead of $o_j$ seems to be best to improve the frequency of $(i_1, ...i_n)$, it may be the worst solution for $(i_1, ...i_{n+3})$'s. But while generating $i_{n-1}$, we cannot predict the best decision for the $i_{n+x}$ level. Thus, exhaustive extraction of gradual itemset is a challenging task.

In another hand as we are using total order relation, it is possible to use restriction properties. Indeed, equality does not directly determine wether an object participates to $s_1 = (i_1^+ i_2^+)$ or to $s_2 = (i_1^+ i_2^-)$, but restricted order can clearly identify to which g-itemset this object belongs. Thus, it is possible to adapt the inclusion-exclusion principle and build at the same time representative object sets for $s_1$ and $s_2$.

Integrating the equality relation could make some g-itemset "*non-gradual*". A typical example is $(A^+C^+)$ from Table 1 which will generate the following representative set: $\{o_1, o_2, o_4, o_5, o_6\}$. However, $t_{o_1}[C] = ... = t_{o_6}[C]$, meaning that even if the age increases, the number of cars does not evolve. To overcome this problem, we could introduce a quality measure. The simplest one would be the percentage of common values for an item. Statistical "measures" such as covariance or entropy could be used too. However, it will be necessary to adapt the former to a multi-variable context. Note that these "measures" do not have an anti-monotonicity property, due to the introduction of a mean. Thus, we will not be able to use them as a prune constraint. At this time, we have not done tests on this point. This is let as a future work.

## 6. CONCLUSION AND FUTURE WORK

In this paper, we address the problem of mining for gradual rules, including rules combining different kinds of variation (increasing and decreasing). This kind of rules is useful and can be applied in many domains, such as bioinformatics, medecine or marketing... However, it requires intensive calculation as many combinations have to be checked. We propose here to use a heuristic-based approach to tackle this challenging problem. Experiments reported here empirically show that our approach is time efficient and scalable.

However, by using a heuristic, we may loose frequent gradual rules as the frequency given by the algorithm may be too low compared to the real value. We are thus currently working on a new complete approach extracting all the rules. We are planning to compare the two approaches in term of time performance, and to study how many gradual rules are discarded when using the proposed here heuristic, compared to the complete extraction. Besides, we will test our approach on real databases, particularly on gene expression databases.

Eventually, our approach will be extended to sequential patterns.

## 7. REFERENCES


[1] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *20th International Conference on Very Large Data Bases, (VLDB'94)*, pages 487–499, 1994.

[2] F. Berzal, J. Cubero, D. Sanchez, M. Vila, and J. Serrano. An alternative approach to discover gradual dependencies. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS)*, 15(5):559–570, Oct. 2007.

[3] P. Bosc, O. Pivert, and L. Ughetto. On data summaries based on gradual rules. In *Proceedings of the 6th International Conference on Computational Intelligence, Theory and Applications*, pages 512–521, London, UK, 1999. Springer-Verlag.

[4] D. Dubois and H. Prade. Gradual inference rules in approximate reasoning. *Inf. Sci.*, 61(1-2):103–122, 1992.

[5] D. Dubois and H. Prade. Gradual elements in a fuzzy set. *Soft Comput.*, 12(2):165–175, 2008.

[6] C. Fiot, F. Masseglia, A. Laurent, and M. Teisseire. Gradual trends in fuzzy sequential patterns. In *12th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, 2008.

[7] E. Hüllermeier. Association rules for expressing gradual dependencies. In *PKDD '02: Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 200–211, London, UK, 2002. Springer-Verlag.

[8] H. Jones, D. Dubois, S. Guillaume, and B. Charnomordic. A practical inference method with several implicative gradual rules and a fuzzy input: one and two dimensions. *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*, pages 1–6, July 2007.

[9] F. Masseglia, P. Poncelet, and M. Teisseire. Pre-processing time constraints for efficiently mining generalized sequential patterns. In *11th International Symposium on Temporal Representation and Reasoning (TIME '04)*, pages 87–95, 2004.