Parallel Approaches for Mining Fuzzy Orderings based Gradual Patterns

Malaquias Quintero Flores*, Federico Del Razo†, Anne Laurent*, Nicolas Sicard‡
* *University Montpellier 2 CNRS - LIRMM, Montpellier, France*
Email: {quinterofl, laurent}@lirmm.fr
† *Toluca Institute of Technology, Toluca, Mexico*
Email: delrazo@ittoluca.edu.mx
‡ *EFREI - AllianSTIC, Paris, France*
Email: nicolas.sicard@efrei.fr

***Abstract*—Mining gradual patterns invokes a number of iterations for generating, adjusting, measuring, and comparing gradual tendencies between numeric attributes of imprecise or uncertain databases. Gradual tendencies are complex correlations of the form {*The hight/lower X, the hight/lower Y*}. Automatic extraction of such gradual patterns involves huge amounts of processing time, load balance, and high memory consumption. When managing large databases, taking this into account is challenging. In this paper, we show a framework and an algorithm based on rank correlation and fuzzy orderings for mining gradual patterns from imprecise or uncertain data. We also present an approach to improve performance of the algorithm using the parallel programming model of OpenMP and the Yale Sparse Matrix Format to reduce memory consumption. Through an experimental study, we show the performance of our approach with respect to the number of attributes of the databases and the number of cores available.**

***Keywords*-Scaling fuzzy models; Fuzzy orderings; Fuzzy gradual dependencies; OpenMP; Parallel programming.**

I. INTRODUCTION

Data mining is often defined as the formulation, analysis, and implementation of an induction process proceeding from specific data to general patterns that facilitates the non-trivial extraction of implicit, unknown, and potentially useful information [20]. Data mining techniques are employed in traditional scientific discovery disciplines, such as biological, medical, biomedical, chemical, physical, social research, and other knowledge industries.

Mining gradual patterns (MGP) is an important task for knowledge discovery (KDD) and data mining (DM). In MGP, the goal is to find in numeric databases interesting and complex correlations of the form $\{X\{\geq|\leq\}, Y\{\geq|\leq\}\}$, interpreted as {*The hight/lower X, the hight/lower Y*} and named *gradual dependencies*.

Gradual dependencies express a relation (*tendency* or *correlation*) among the variation of the values of attributes of a *gradual pattern*. For example, in the database shown in Table I containing data about fruit characteristics, such as A_1*:Size*, A_2*:Weight* and A_3*:Sugar Rate*, the gradual pattern {*the higher the weight*, *the hight the sugar rate*}, means that as the *weight* of a fruit increases, its *sugar rate* tends to increases. Accordingly {*the lower the weight*, *the lower the sugar rate*}, means that as the *weight* of a fruit decreases, its *sugar rate* tends to decreases.

Table I
A SMALL DATABASE OF SOME FRUIT CHARACTERISTICS

Id	$A_1 : Size$	$A_2 : Weight$	$A_3 : SugarRate$
t_0	6	6	5.3
t_1	10	12	5.1
t_2	14	4	4.9
t_3	23	10	4.9
t_4	6	8	5.0
t_5	14	9	4.9
t_6	18	9	5.2
t_7	23	10	5.3
t_8	28	13	5.5

This new way to analyse the degree in which a pattern is present in a transaction, called *gradual pattern* or *gradual dependency*, was first proposed in [10], thereafter studied in [1][13][16], and more recently in [5][12][18].

There exist some algorithms for MGP that focus on finding gradual patterns from precise and certain data [12][22]. Unfortunately, this is not always so, there are situations where databases contain imprecise and uncertain data (e.g., meteorological data, river pollution data, indicator economic data, biological data, and so on) due to human errors, instrument errors, recording errors, noisy data, variables with imprecise and uncertain behaviour, and so on [2][6][22].

In this paper, we present a framework and an algorithm based on rank correlation and fuzzy orderings for MGP (*fuzzy*-MGP) [18] from imprecise or uncertain data. We also present an approach to improve performance of *fuzzy*-MGP algorithm using the parallel programming model of OpenMP [19][21] and a dedicated technique for handling sparse matrices to reduce memory consumption.

The outline of the paper is as follows: In Section II, we present the definitions of gradual dependency, fuzzy orderings, fuzzy γ rank correlation and related works. In

Section III, we explain our parallel fuzzy orderings for fuzzy gradual pattern mining algorithm (*parallel fuzzy*-MGP). Section IV presents experiments and main results. Lastly, in Section V, we present our conclusions and perspectives of research.

II. Definitions and Related Works

A. Gradual dependency: Some Background

In the context of fuzzy data mining, the concept of *gradual dependency* was first proposed to perform a regression analysis between the change of the presence of fuzzy items in a transaction [10].

In [1], a gradual dependence is defined as a rule of the form $(v_1, X, A) \rightarrow (v_2, Y, B)$ holds in D *iff* $\forall (x,y),(x^{'},y^{'}) \in D$, meaning that $A(x)v_1A(x^{'})$ implies $B(y)v_2B(y^{'})$, where X, Y are two attributes of database D containing pairs of values $(x,y) \in X \times Y$, two fuzzy sets A, B defined on the domains of X, Y respectively, and $v_1, v_2 \in \{<,>\}$, represent two variations *the less* ($<$) and *the more* ($>$).

In [16], Molina et al. propose a particular definition of fuzzy gradual dependence from an specific approach to fuzzy association rules, where gradual dependencies represent tendencies in the variation of the degree of fulfilment of properties in a set of objects, and also introduce the notion of degree of variation associated to a pair of objects.

In [13], Laurent et al. present an approach for extracting gradual itemsets, where combines the interpretation of gradual dependency of Berzal in [1], with a rank correlation measure and the concept of binary matrices for representing the sets of concordant couples.

Koh and Hüllermeier [12] present a framework for mining fuzzy gradual dependencies, in which the strength of association between itemsets is measured in terms of a fuzzy rank correlation coefficient.

B. Gradual Pattern: Concepts and Definitions

Given a DB (database), constituted of *n* data record (transactions) **T**=$\{t_1, t_2, ..., t_n\}$, described by *m* numeric attributes **A**=$\{A_1, A_2, ..., A_m\}$, each record t_i is represented as a vector with *m* values, $t_i = \{A_1(t_i), A_2(t_i), ..., A_m(t_i)\}$, were $A_2(t_i)$ is the value of t_i for the attribute A_2, similarly for each record t_j.

A *GP*(*gradual pattern*) defines a relation of simultaneous variation between values of the attributes of two or more *gradual items*, different approaches have been defined according to the interpretation of the concept of gradual dependency, a *GP* is a combination of two or more *gradual items* of the form *GP*=$\{A_1{\geq}A_2{\geq}A_3{\leq}\}$ interpreted as $\{$*The hight* A_1*, the hight* A_2*, the lower* $A_3\}$, where the size (*k*) of a *GP* is defined as the number of *gradual items* contained in the *GP*, such that *k*$\in\{$*2, 3, 4,..., m*$\}$.

A *gradual item* is defined as the variation **v** associated to the values of a attribute $A_l{\in}$*DB* denoted as A_l**v**, where **v** can be ascending ($\geq$|+) if the attribute values increase, descending ($\leq$|$-$) if the attribute values decrease, i.e., $\{A_l{\geq}\} \simeq \{A_l(t_i){<}A_l(t_j)\}$ and $\{A_l{\leq}\} \simeq \{A_l(t_i){>}A_l(t_j)\}$ for *i=1, 2, ..., n*, for *j=i+1, ..., n*, *i*$\neq$*j* and *l*$\in\{$*1,2, ..., k*$\}$.

A *concordant couple (cc)* is an index pair *cc(Ii,Ij)* defined in (1), where the records (t_i, t_j) satisfy all the variations **v** expressed by the involved *gradual items* in a given *GP* of size *k*, e.g., let *GP*=$\{A_1{\geq}A_2{\geq}A_3{\leq}\}$ with size *k=3*, an index pair *cc(Ii,Ij)* is a *concordant couple* if $((A_1(t_i){<}A_1(t_j)\ implies\ A_2(t_i){<}A_2(t_j))\ implies\ A_3(t_i){>}A_3(t_j))$, where *Ii* is defined in (2) and *Ij* in (3).

$$cc(Ii, Ij) = \begin{cases} 1 & \text{if } cc(Ii,Ij)\ is\ concordant\ couple \\ \\ 0 & \text{in otherwise.} \end{cases} \tag{1}$$

$$\begin{cases} Ii = ((A_1(t_i), A_2(t_i)), (A_1(t_i), A_3(t_i)))\ or \\ \quad = ((A_1(t_i), A_2(t_i)), A_3(t_i)) \end{cases} \tag{2}$$

$$\begin{cases} Ij = ((A_1(t_j), A_2(t_j)), (A_1(t_j), A_3(t_j)))\ or \\ \quad = ((A_1(t_j), A_2(t_j)), A_3(t_j)) \end{cases} \tag{3}$$

In *MGP* a *minimal support* is used to choose interesting *GP* from *gradual items* which frequently occur together. A *GP* is an interesting pattern if *support(GP)* is greater than or equal to the user-predefined minimal support named *minimum threshold*.

In order to compute the support of a GP, different approaches have been defined according to the interpretation of the concept of gradual dependency, such as based on *fuzzy implication interpretation* [4][7][9], *regression analysis* [10], *induced rankings correlation* [1][13], *ranking-compliant data subsets* [15], *strengthening quality criteria* [5], and so on. An interesting analysis of such interpretations is considered in [5][13].

In the framework of the interpretation of gradual dependency based on *induced rankings correlation* and *concordant couple* concept, the *support* of a *GP* is computed as

$$support(GP) = \frac{\sum_{i=1}^{n} \sum_{j \neq i} cc\,(Ii, Ij)}{\frac{n(n-1)}{2}} \tag{4}$$

C. Fuzzy Orderings-Based $\tilde{\gamma}$ Rank Correlation Measure

Fuzzy orderings is a concept that have been introduced with the aim to model human-like decisions by taking the graduality of human thinking and reasoning into account. Within the framework of a process of making decisions *fuzzy orderings* allow express and evaluate preferences among a set of available alternatives [2][24].

A fuzzy relation $L : X^2 \rightarrow$[0,1] is called *fuzzy ordering* with respect to a t-norm *T* and a *T*-equivalence $E : X^2 \rightarrow$[0,1], for brevity *T-E-ordering*, if and only if the following three axioms are fulfilled for all *x, y, z* ∈ *X*:

(i) *E*-Reflexivity: $E(x,y) \leq L(x,y)$
(ii) *T-E*-Antisymmetry: $T(L(x,y),L(y,x)) \leq E(x,y)$
(iii) *T*-Transitivity: $T(L(x,y),L(y,z)) \leq L(x,z)$.

The result of combining fuzzy orderings and *gamma rank correlation measure* is a robust rank correlation coefficient ideally suited for measuring rank correlation for numerical data perturbed by noise [3]. This innovative correlation coefficient is known as *fuzzy ordering-based rank correlation measure* $\tilde{\gamma}$. Its formal definition is:

$$\tilde{\gamma} = \frac{CT - DT}{CT + DT} \quad (5)$$

$$CT = \sum_{i=1}^{n} \sum_{j \neq i} \tilde{C}(i,j) \quad (6)$$

$$DT = \sum_{i=1}^{n} \sum_{j \neq i} \tilde{D}(i,j) \quad (7)$$

$$\tilde{C}(i,j) = \top(R_X(x_i,x_j), R_Y(y_i,y_j)) \quad (8)$$

$$\tilde{D}(i,j) = \top(R_X(x_i,x_j), R_Y(y_j,y_i)) \quad (9)$$

$$R_X(x_i,x_j) = 1 - L_X(x_j,x_i) \quad (10)$$

$$L_X(x1,x2) = min(1, max(0, 1 - \frac{(x1-x2)}{r})) \quad (11)$$

$$R_Y(y_i,y_j) = 1 - L_Y(y_j,y_i) \quad (12)$$

$$L_Y(y1,y2) = min(1, max(0, 1 - \frac{(y1-y2)}{r})) \quad (13)$$

$$\top(a,b) = max(1, a+b-1) \quad (14)$$

$$\{(x_i,y_i)_{i=1}^{n} | x_i \in X \ and \ y_i \in Y\} \quad (15)$$

where $\tilde{C}$*(i, j)* is the degree to which *(i, j)* is a concordant pair and $\tilde{D}$*(i, j)* is the degree to which *(i, j)* is a discordant pair, for *i*=(x_i, y_i) and *j*=(x_j, y_j), such that i=1, 2,...,n, j=1, 2,...,n, $i \neq j$ and $n \neq CT + DT$. Given $n \geq 2$ pairs of numeric observations, defined in (15)

$R_X(x_i,x_j)$ is a strict $T_L - E_X - ordering$ on *X* defined in (10), $R_Y(y_i,y_j)$ is a strict $T_L - E_Y - ordering$ on *Y* defined in (12), $L_X(x_i,x_j)$ is a strongly complete $T_L - E_r - ordering$ on *X* defined in (11), and $L_Y(y_i,y_j)$ is a strongly complete $T_L - E_r - ordering$ on *Y* defined in (13) (assume *r>0*).

$\top$ is a $\top-$*equivalence relation* and denotes a Lukasiewicz *t-norm* defined in (14) for a=$R_X(x_i,x_j)$ and b =$R_Y(y_i,y_j)$. For more information we recommend consulting [2][3][12].

D. Problem statement

An important challenge in gradual pattern mining is to analyze the correlation between the variation of numerical attribute values perturbed by noise, and to consider when such a small difference between two values is meaningful. In this context, we present a fuzzy orderings-based framework for mining fuzzy gradual patterns, where we propose to compute the support of a *GP*, as in (16) based on compute of the degree to which *(Ii, Ij)* is a concordant pair as is defined in (8) and according to the definition of *concordant couple* given in (1), (2), and (3).

$$fuzzsupport(GP) = \frac{\sum_{i=1}^{n} \sum_{j \neq i} \tilde{C}(Ii,Ij)}{n(n-1)} \quad (16)$$

Huge amounts of processing time, load balance and high memory consumption are important problems observed in gradual pattern mining algorithms. We addressed these problems making use of a parallel programming model and an efficient technique to reduce memory consumption.

E. Related Work

Recently, in [14] and [15], Laurent et al. have presented PGP-mc a multicore parallel approach for mining gradual patterns where the evaluation of the correlation and support is based on conflict sets and precedence graph approaches. PGP-mc was implemented using the g++ 3.4.6 and 4.3.2 with POSIX threads, on two different workstations: i) COYOTE machine, with 8 AMD Opteron 852 processors (each with 4 cores), 64 GB of RAM with Linux Centos 5.1 And ii) IDKONN machine, with 4 Intel Xeon 7460 processors (each with 6 cores), 64 GB of RAM with Linux Debian 5.0.2. The experiments were led on synthetic databases automatically generated by a tool based on adapted version of IBM Synthetic Data Generation Code for Associations and Sequential Patterns.

 ISBN: 978-1-61208-217-2

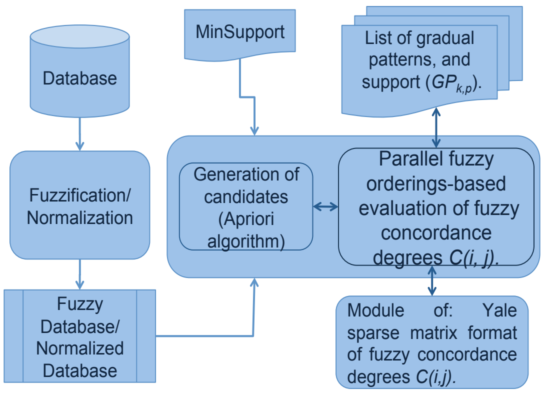

Figure 1. Parallel fuzzy orderings-based extraction of fuzzy frequent gradual dependencies: General structure.

An efficient parallel mining of closed frequent gradual patterns, named PGLCM, has been proposed by Do et al. [8]. This approach is based on the principle of the LCM algorithm for mining closed frequent patterns, an adaptation of LCM named GLCM in order to mine closed frequent gradual patterns, and parallelization of the GLCM algorithm named PGLCM based on the Melinda parallelism environment. The comparative experiment is based on synthetic databases produced with the same modified version of IBM Synthetic Data Generator for Association and Sequential Patterns. All the experiment have been conduced on a 4-socket server with 4 Intel Xeon 7460 with 6 cores each and 64 GB of RAM.

III. Parallel Fuzzy orderings for fuzzy gradual pattern mining

In this section, we present an approach to improve the performance of our method of automatic extraction of frequent gradual patterns on the basis of fuzzy orderings [18] and the idea of storing the fuzzy concordance degrees $C(i,j)$ in sparse matrices.

Figure 1 shows the general structure of our optimization approach which has two purposes: A) Reduce memory consumption and B) Improve execution time via parallelization.

A. Memory consumption

In order to reduce memory consumption, we represented and stored each matrix of concordance degrees $C(i,j)$ according to the *Yale Sparse Matrix Format* [23], such as only non-zero coefficients are retained. Because we generate candidates from the frequent $k-$patterns, only matrices of the (*k-1*)$-$level frequent gradual patterns are kept in memory while being used to generate the matrices of the $(k)-$level gradual pattern candidates, e.g., Figure 2 illustrates the extraction of gradual patterns, from the data set of Table I, with a *minimum threshold*= 0.15, for level *k=2* and *k=3*, where, if *support* of a *gradual pattern* (*GP*) is less than *minimum threshold* then the *GP* is pruned and its matrix of

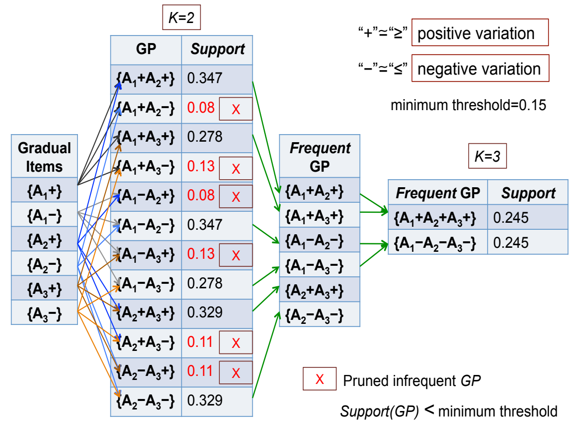

Figure 2. Extraction of gradual patterns from data set of Table I for level *k=2* and *k=3*.

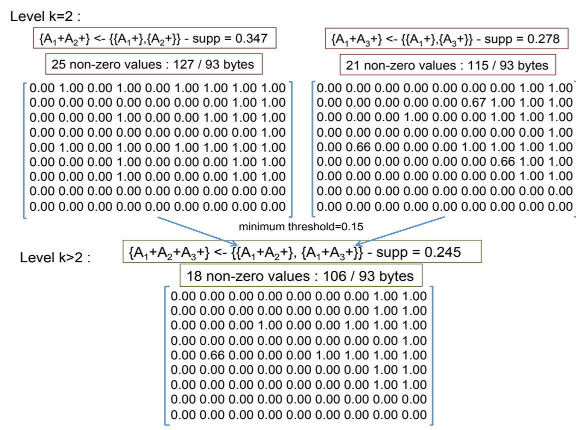

Figure 3. Examples of matrices of fuzzy concordance degrees *C(i,j)* of three gradual patters of Figure 2 with ascending variation.

Compressed matrix of fuzzy concordance degrees *C(i,j)*:

Gradual pattern $GP_{3,1}$= $\{A_1+A_2+A_3+\} \longleftarrow \{\{A_1+A_2+\},\{A_1+A_3+\}\}$

18 non-zero values : 106 / 93 bytes

	0	1	2	3	4	5	6	7	8
IA	0	2	4	8	9	14	16	null	null

	0	1	2	3	4	5	6	7	8	9	10
JA	7	8	7	8	3	6	7	8	8	1	5
A	1.0	1.0	1.0	1.0	1.0	1.0	0.72	1.0	1.0	0.66	1.0

	11	12	13	14	15	16	17
JA	6	7	8	7	8	7	8
A	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Figure 4. Example of representation of a matrix of fuzzy concordance degrees *C(i,j)* in the *Yale Sparse Matrix Format*.

 ISBN: 978-1-61208-217-2

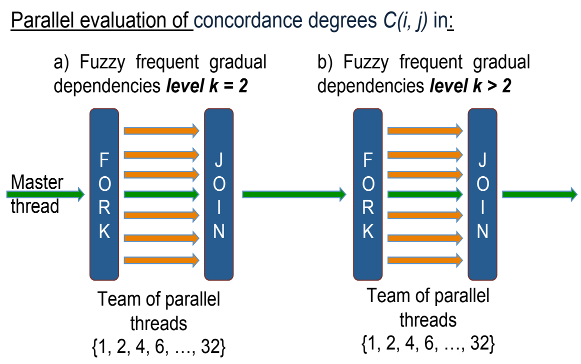

Figure 5. Parallel evaluation of fuzzy concordance degrees in: a) Gradual dependency of level $k = 2$, b) Gradual dependency of level $k > 2$.

fuzzy concordance degrees *C(i,j)* is removed.

Figure 3 shows the matrices of fuzzy concordance degrees *C(i,j)* of the gradual patters $GP_{2,1}$=$\{A_1+A_2+\}$, $GP_{2,3}$=$\{A_1+A_3+\}$ of level *k=2*, and $GP_{3,1}$=$\{A_1+A_2+A_3+\}$ of level *k=3*. Notation "+" represents an ascending variation "$\geq$" and notation "$-$" represents an descending variation "$\leq$". Figure 4 shows the representation of the matrix of fuzzy concordance degrees *C(i,j)* in the *Yale Sparse Matrix Format* for the gradual patter $GP_{3,1}$.

B. Parallelization of algorithm

Load balance, huge amounts of processing time and high memory consumption are important problems observed in gradual pattern mining algorithms. We addressed these problems using the shared memory architecture API of OpenMP, which is ideally suited for multi-core architectures [19][21].

Figure 5 gives an overall view of our approach to parallelize the automatic extraction of frequent gradual patterns of level *k=2* and the automatic extraction of frequent gradual patterns of level *k>2*.

Algorithm 1 shows the pseudocode of the master thread, where DB is a database, ***m*** is the number of attributes in DB, ***n*** is the number of records in DB, $\{A_{\mathbf{m}}\}$ represents the identifiers of the ***m*** attributes.

The variable named $minSupp$ is the *minimum threshold*, ***v***←{+} denotes a positive (ascending) variation in each index pair of an attribute, ***v***←{-} denotes a negative (descending) variation in each index pair of an attribute, *ListGIs* represents the list of *gradual items* level *k=1*, $\mathcal{F}_{k=2}$ is the set of gradual patterns/dependencies of level *k=2*, $\mathcal{F}_k$ is the set of gradual patterns/dependencies of level *k>2*, $\mathcal{F}_{FGD}$ are all fuzzy frequent gradual dependencies and *Nt* is the number of threads of each parallel region.

Algorithm 1: Main Thread for Fuzzy Gradual Dependency Mining

Data: Database(DB), # Attributes(**m**), # Records(**n**),
Data: Id. Attributes$\{A_{\mathbf{m}}\}$, $minSupp$, # Threads (Nt).
Result: Fuzzy Frequent Gradual Dependencies $\mathcal{F}_{FGD}$

```
F_FGD ← ∅;  v ← {+, −};
ListGIs = build_gItems({A_m} × v);
k ← 2;
/* Parallel extraction of fuzzy frequent gradual
dependencies level k = 2*/;
F_{k=2} = GradualPatterns(ListGIs, Ds, minSupp);
F_FGD ← F_FGD ∪ {F_{k=2}};
k++;
repeat
    /* Parallel extraction of fuzzy frequent gradual
    dependencies level k > 2*/;
    F_k=GradualPatterns(F_{k−1}, level(k), minSupp);
    F_FGP ← F_FGP ∪ {F_k};
    Delet(F_{k−1}.M);
    k++;
until F_FGP does not grow any more;
```

The pseudo-code of the Algorithm 2 corresponds to the parallel extraction of frequent gradual dependencies of level $k = 2$, where $Sdc_{k=2}$ is the set of gradual patterns of size $k = 2$, Gdc is a pattern gradual candidate $\in Sdc_{k=2}$, $Gdc_q.M$ represents the matrix of fuzzy concordance degrees $C(i,j)$ *computed by fuzzy orderings*.

The pseudo-code of the Algorithm 3 corresponds to the parallel extraction of frequent gradual dependencies of level $k > 2$, where $\{\mathcal{F}_{k^1}\}$ is the set of frequent gradual patterns of level $(k-1)$, $C_{k,q}.M$ represents the matrix of fuzzy concordance degrees $C(i,j)$ of a gradual pattern candidate of level k (*computed* by $T-norm$) and $\mathcal{F}_k$ is the set of frequent gradual patterns of level k.

IV. Experiments and Main Results

A. Experiments

We present an experimental study of the scaling capacities of our approach on several cores, for the database C500A100 with 500 records and 100 attributes, and database C500A150 with 500 records and 150 attributes, which were used in [8][14] and produced with the IBM Synthetic Data Generator for Association and Sequential Patterns.

Our experiments were performed on a workstation with up to 32 processing cores, named COYOTE, with 8 AMD Opteron 852 processors (each with 4 cores), 64 GB of RAM with Linux Centos 5.1, GCC OpenMP 3.1.

 ISBN: 978-1-61208-217-2

Algorithm 2: Parallel Fuzzy Orderings for Gradual Pattern Mining: Level $k = 2$

Data: List_of_gradual_Items($List\_gIs$), minSupp
Data: Attribute values (aValues), # Threads (Nt)
Result: Fuzzy Frequent Gradual Dependencies $(\mathcal{F}_{k=2})$
$\mathcal{F}_{k=2} \leftarrow \emptyset$; $q \leftarrow 1$;
/*Each thread computes the ***support*** of its fuzzy frequent gradual dependency of level $k = 2$*/;
for *all*$(thread\ in(1, 2, 4, 6, 8, ..., Nt = 32))$ **do**
  $Sdc_{k=2} \leftarrow GenCand(\{List\_gIs\}, level(k = 2))$;
  $Gdc \leftarrow FirstxCandidate \in Sdc_{k=2}$;
  foreach $Gdc \in Sdc_{k=2}$ **do**
    $Gdc_q.M = FuzzOrderings(Gdc, aValues)$;
    $Support(Gdc_q) = EvalSupport(Gdc_q.M)$;
    / minSupp stands for a user-specified minimum support value */*;
    if $Support(Gdc_q) \geq minSupp$ **then**
      $Critical\ section$:
      $>>>$ $\mathcal{F}_{k=2} \leftarrow \mathcal{F}_{k=2} \cup \{Gdc_q\}$;
    $q + +$;
    $Gdc_q \leftarrow NextCandidate \in Sdc_{k=2}$;

Algorithm 3: Parallel Fuzzy Orderings for Gradual Pattern Mining: Level $k > 2$

Data: Fuzzy Frequent Gradual Dependency$\mathcal{F}_{k-1}$,
Data: minSupp, Level*(k>2)*, # Threads (Nt).
Result: Fuzzy Frequent Gradual Dependencies $(\mathcal{F}_k)$
$\mathcal{F}_k \leftarrow \emptyset$; $q \leftarrow 1$; $k^1 \leftarrow k - 1$;
/*Each thread computes the ***support*** of its fuzzy frequent gradual dependency of level $k > 2$*/;
for *all*$(thread\ in(1, 2, 4, 6, 8, ..., Nt = 32))$ **do**
  $F_{k^1,a} \leftarrow GenFirstFather1(\{\mathcal{F}_{k^1}\})$;
  $F_{k^1,b} \leftarrow GenFirstFather2(\{\mathcal{F}_{k^1}\})$;
  foreach $\{F_{k^1,a}, F_{k^1,b}\} \in \{\mathcal{F}_{k^1}\}$ **do**
    $C_{k,q}.M = T\!-\!norm(\{F_{k^1,a}.M\}, \{F_{k^1,b}.M\})$;
    $Support(C_{k,q}.M) = EvalSupport(C_{k,q}.M)$;
    / minSupp stands for a user-specified minimum support value */*;
    if $Support(C_{k,q}.M) \geq minSupp$ **then**
      $Critical\ section$:
      $>>>$ $\mathcal{F}_k \leftarrow \mathcal{F}_k \cup \{C_{k,q}\}$;
    $F_{k^1,a} \leftarrow GenNextFather1(\{\mathcal{F}_{k^1}\})$;
    $F_{k^1,b} \leftarrow GenNextFather2(\{\mathcal{F}_{k^1}\})$;
    $q + +$;

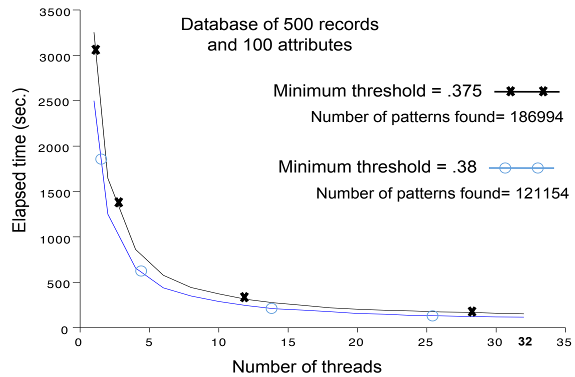

Figure 6. Threads vs. elapsed time with a database of 500x100 and minSupp=.375 and .38, using uncompressed binary matrices of concordance degrees.

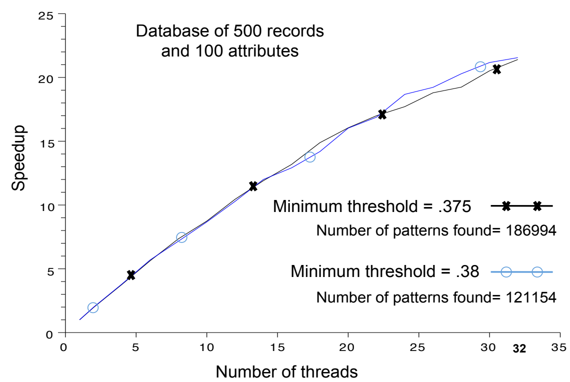

Figure 7. Speedup with a database of 500x100 and minSupp=.375 and .38, using uncompressed binary matrices of concordance degrees.

B. Main Results

The first experiment involves a database with 500 lines and 100 attributes, Figures 6 and 7 depict the execution time and speed-up related to: 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, and 32 threads, on the test database, from which were found 186994 frequent gradual patterns for a minimum threshold of 0.375, and 121154 frequent gradual patterns for a minimum threshold of 0.38. In the first case we reach a memory consumption of 36.2% using uncompressed binary matrices of concordance degrees and of 14.4% using compressed matrices of concordance degrees (Yale Sparse Matrix Format). While for the second case we obtained a memory consumption of 24.7% with uncompressed binary matrices of concordance degrees and of 10.3% with compressed matrices of concordance degrees. Within this experimental framework, Figures 8 and 9 illustrate the execution time and speed-up of our approach using compressed matrices of concordance degrees.

The second experiment involves a database with 500 lines and 150 attributes, from which were found 100834 frequent

 ISBN: 978-1-61208-217-2

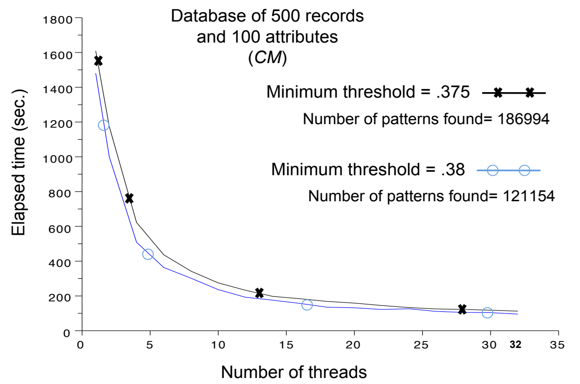

Figure 8. Threads vs. elapsed time with a database of 500x100 and minSupp=.375 and .38, using compressed matrices of concordance degrees.

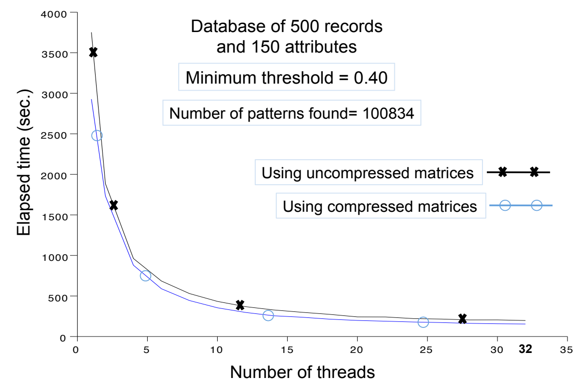

Figure 10. Threads vs. elapsed time with a database of 500x150 and minSupp=0.40, using uncompressed and compressed matrices of concordance degrees.

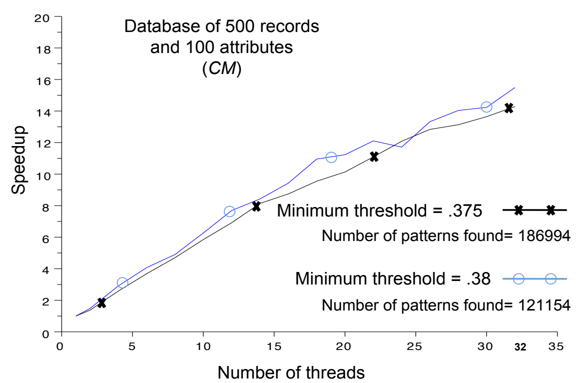

Figure 9. Speedup with a database of 500x100 and minSupp=.375 and .38, using compressed matrices of concordance degrees.

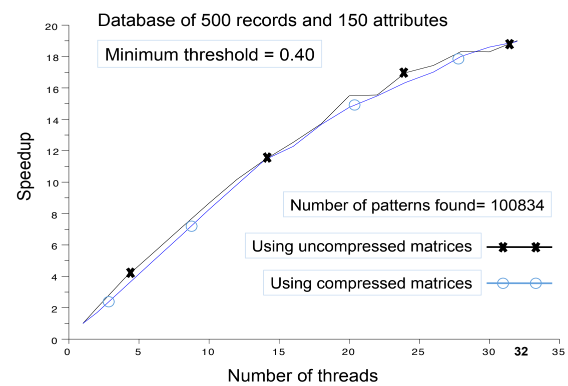

Figure 11. Speedup with a database of 500x150 and minSupp=0.40, using uncompressed and compressed matrices of concordance degrees.

gradual patterns for a minimum threshold of 0.40, where we reach a memory consumption of 24.5% using uncompressed binary matrices of concordance degrees and of 10.8% using compressed matrices of concordance degrees (Yale Sparse Matrix Format). Within this experimental framework, Figures 10 and 11 depict the execution time and speed-up of our approach, related to: 1, 2, 4, 6, . . . , to 32 threads.

V. Conclusion and Future Work

In this paper, we presented the parallelization of our fuzzy gradual pattern mining named *fuzzy orderings for fuzzy gradual pattern mining*. Our approach consists in parallelizing the extraction of gradual dependencies of level $k = 2$ and in parallelizing the extraction of gradual dependencies of level $k > 2$. The programming parallel model used was the shared memory with the API of OpenMP - C++.

In order to reduce memory consumption, each matrix of concordance degrees $C(i, j)$ is represented and stored according to the *Yale Sparse Matrix Format*, such as only non-zero coefficients are retained. Because we generate candidates from the frequent $k-$itemsets, only matrices of the $(k-1)-$level frequent gradual patterns are kept in memory while being used to generate the matrices of the $(k)-$level gradual pattern candidates.

The experimental work reported in this document was conducted on two databases, the first with 500 records and 100 attributes and the second with 500 records and 150 attributes.

In general, the performance of our *Parallel Fuzzy Gradual Dependency Mining* is significantly improved with the use of compressed matrices, mainly with databases containing a large number of attributes. We recommend testing our algorithm with databases with a larger number of records and real world databases.

References

[1] F. Berzal, J.C. Cubero, D. Sanchez, J. M. Serrano, and M. A. Vila, *An alternative approach to discover gradual dependencies,* Int. Journal of Uncertainty, Fuzziness and Knowledge-based Systems, vol. 15, no. 5, pp. 559-570, (2007).

[2] U. Bodenhofer and F. Klawonn, *Towards robust rank correlation measures for numerical observations on the basis of fuzzy orderings,* in 5th Conference of the European Society for Fuzzy Logic and Technology, pp. 321 - 327, Ostrava, Czech Republic, University of Ostrava, Institute for Research and Applications of Fuzzy Modeling, (Septembre, 2007).

[3] U. Bodenhofer and F. Klawonn, *Roboust rank correlation coefficients on the basis of fuzzy orderings: Initial steps,* in Mathware & Soft Computing 15, 5-20 (2008).

[4] B. Bouchon-Meunier and S. Desprès. *Acquisition numérique / sysmbolique de connaissance graduelles,* in 3èmes Journées Nationales du PRC Intelligence Artificielle. Hermès, pp. 127-138 (1990).

[5] B. Bouchon-Meunier, A. Laurent, M.-J. Lesot, and M. Rifqi, *Strengthening fuzzy gradual rules through "all the more" clauses,* in Proc. WCCI 2010 IEEE World Congress on Computatinal Inteligence, FUZZ-IEEE 2010, pp. 2940-2946 (2010).

[6] T. Calders, B. Goethals, and S. Jaroszewicz, *Mining Rank-Correlated Sets of Numerical Attributes,* in Proc. KDD'06 ACM, pp. 96-105, (2006).

[7] D. Dubois and H. Prade, *Gradual inference rules in approximate reasoning,* Information Sciences, vol. 61, no. 1-2, pp. 103-122 (1992).

[8] T. D. T. Do, A. Laurent, and A. Termier, *PGLCM: Efficient Parallel Mining of Closed Frequent Gradual Itemsets,* in Proc. International Conference on Data Mining (ICDM), pp. 138-147 (2010).

[9] E. Hüllermeier, *Implication-based fuzzy association rules,* in Proceedings PKDD'01, pp. 241-252, (2001).

[10] E. Hüllermeier, *Association rules for expressing gradual dependencies,* in Proceedings PKDD 2002 Lecture Notes in Computer Science 2431, pp. 200-211 (2002).

[11] E. Hüllermeier, *Fuzzy Sets in Machine Learning and Data Mining,* in Applied Soft Computing Journal, vol. 11, pp. 1493-1505, (2011).

[12] H-W. Koh and E. Hullermeier, *Mining gradual dependencies based on fuzzy rank correlation,* in Combining Soft Computing and Statistical Methods in Data Analysis. Advances in Intelligent and Soft Computing, Vol. 77, pp. 379-386. Springer Heidelberg (2010).

[13] A. Laurent, M.-J. Lesot, and M. Rifqi, *GRAANK: Exploiting rank correlations for extracting gradual itemsets,* in FQAS'09, LNAI 5822, Springer Berlin Heidelberg, pp. 382-393, (2009).

[14] A. Laurent, B. Negrevergne, N. Sicard, A. Termier, *PGP-mc: Towards a multicore parallel approach for mining gradual patterns,* in: Proc. DASFAA, pp. 78-84, (2010).

[15] A. Laurent, B. Negrevergne, N. Sicard, A. Termier, *Efficient parallel mining of gradual patterns on multi-core processors,* in AKDM-2, Advances in Knowledge Discovery and Management. Vol. 2. Springer, (2010).

[16] C. Molina, J. M. Serrano, D. Sanchez, and M. A. Vila, *Mining gradual dependencies with variation strength,* in Mathware & Soft Computing, Volume 15, pp. 75 - 93 (2008).

[17] P. L. Nancy, and C. Hao-en, *Fuzzy Correlation Rules Mining,* in Proceedings of the 6th WSEAS International Conference on Applied Computer Science, Hangzhou, China, pp 13-18, April 15-17 (2007).

[18] M. Quintero, A. Laurent, P. Poncelet, *Fuzzy Ordering for Fuzzy Gradual Patterns,* in FQAS 2011, LNAI 7022, Springer-Verlag Berlin Heidelberg, pp. 330-341, (2011).

[19] T. Rauber, G. Rnger, *Parallel Programming: for Multicore and Cluster Systems,* Springer-Verlag Berlin Heidelberg (2010)

[20] V. Stankovski, *Special section: Data mining in grid computing environments,* in CienceDirect Elsevier, Future Generation Computer Systems, vol. 23, pp. 31-33, (2007).

[21] R. Van der Pas, *An Overview of OpenMP,* In the OpenMP API specification for parallel programming, http://openmp.org/wp/resources/#Tutorials, [retrieved: May, 2012].

[22] D.-H. Weng, and Y.-L. Chen, *Mining fuzzy association rules from uncertain data,* in Knowl Inf Syst, vol. 23, Springer, pp. 129-152, (2010).

[23] D. Wilhelm, *Sparse Matrix Formats,* in ECE 250 Algorithms and Data Structures, Department of Electrical and Computer Engineering, University of Waterloo, Ontario, Canada, [retrieved: May, 2012].

[24] L. A. Zadeh, *Similarity Relations and Fuzzy Orderings,* in Information Sciences. Volume 3, Issue 2, pp. 177 - 200, (April, 1971).