Fuzzy-Temporal Gradual Patterns

Dickson Odhiambo Owuor, Anne Laurent, Joseph Onderi Orero

Fuzzy-Temporal Gradual Patterns

Dickson Owuor
LIRMM Univ Montpellier, CNRS
Montpellier, France
doowuor@lirmm.fr

Anne Laurent
LIRMM Univ Montpellier, CNRS
Montpellier, France
laurent@lirmm.fr

Joseph Orero
FIT, Strathmore University
Nairobi, Kenya
jorero@strathmore.edu

***Abstract*—Gradual patterns allow for retrieval of correlations between attributes through rules such as "the more the exercise, the less the stress". However, it may be the case that there is a lag between changes in some attributes and their impact on others ones, current methods do not take this into account. In this paper, we extend existing methods to handle these situations in order to retrieve patterns such as: "the more the exercise increases, the more the stress decreases 1 month later". We also extend our gradual rules to include fuzzy temporal constraints such as "the more the exercise increases, the more the stress decreases almost 1 month later". For this kinds of patterns, we designed three algorithms that were implemented and tested on real data.**

***Index Terms*—fuzzy membership, gradual patterns, temporal tendencies**

I. Introduction

Mining gradual patterns enables the testing of data crossings that can detect relevant correlations between the attributes of a data-set. One of the methods for extracting gradual patterns is to apply gradual rules in the form of "the more/less A1, ..., the more/less An" that correlates n attributes A1, A2, ..., An. For instance the "the greater the number of exercise activities, the lower the level of stress" [1], [2]. Example 1 is an instance applying gradual rules in the data-set shown in Table I.

Example 1. we consider a data-set containing the types of physical exercises that a person performed together with the stress levels reading for different dates. It is important to note that the values in Table I are arbitrary. Each tuple in the data-set correspond to a daily record of a person.

Correlation between exercise and stress levels			
id	**date** (day/month)	**activity** (exercise)	**stress levels**
r1	01/06	swim	4
r2	02/06	jog,swim	2
r3	03/06	walk,jog	3
r4	04/06	walk	5
r5	05/06	walk,jog,swim	1

Table I: Sample data-set D_1

A key point to realize is that deriving the support for a gradual pattern involves at least two or more records because the patterns are built on the increasing or decreasing nature of an attribute. In the case of pattern $\{(exercise,\uparrow),(stress,\downarrow)\}_{sup=3}$, the support is 3 because we can order records <r1, r2, r5> successively to match the gradual pattern.

Because of the complexity deriving support, the efficiency of gradual patterns relies on the antimonocity property which states that *no frequent pattern containing n attributes can be built over a pattern containing a subset of these n attributes* [3]. For example if the pattern "the greater the A, the greater B" is not relevant, then it is impossible for the pattern "the greater the A, the greater the B, the greater the C" to be relevant.

In comparison to association rules, extracting gradual patterns allows for discovery of more meaningful correlations between attributes of a data-set beyond finding frequent related item-sets. However, it may be the case that the value an attribute causes a ripple effect on other attribute with respect to time. For instance in Table I, it may be the case that exercise causes stress to reduce a few days (or weeks) later and not on the same day.

In order to for us to extract a pattern that correlates attributes with a time lag, we need to extend the existing methods for extracting gradual patterns in order to capture the temporal aspects of such data-sets.

II. Preliminary Definitions

We recall below some definitions taken from literature that describe gradual pattern mining.

Definition 1. *Gradual Item. A gradual item is a pair (i,v) where i is an item and v is a variation $v \in \{\uparrow,\downarrow\}$. $\uparrow$ stands for an increasing variation while $\downarrow$ stands for a decreasing variation.*

Example 2. $(exercise,\uparrow)$ is a gradual item that can be interpreted as "the more the exercise".

Definition 2. *Gradual Pattern (also known as Gradual item-set). A gradual pattern is a set of gradual items, denoted by $GP = \{(i_1,v_1),...,(i_n,v_n)\}$. The set of all gradual patterns that can be defined by GP.*

Example 3. $\{(jogging,\uparrow),(walking,\uparrow),(stress,\downarrow)\}$ is a gradual item-set that can be interpreted as "the more the jogging, the more the walking, the less the stress".

It is important to note that gradual pattern mining aims at extracting the frequent patterns, in contrast to the classical data mining framework that aims to extract frequent item-sets through techniques such as association rules.

Definition 3. *Given a threshold of a minimum support σ, a gradual pattern GP is said to be frequent if $_{supp}(GP) \geq \sigma$.*

There is a need to describe what *frequent* means in the context of gradual patterns. The principle idea that the support is based on, is that of counting the proportion of tuples in a data-set that respects the gradual pattern [1]. For instance in Table I, we see that for records r1 and r2 the number of exercise activities increase while the stress level decrease simultaneously, since 'swim' $<$ 'jog, swim' and $4 > 2$.

One support proposed in [4] is based on the length of the longest path of exercises that can be built on this pattern. While [5] and [6] consider the number of tuples that are concordant by exploiting the Kendall's τ rank correlation.

Definition 4. *The support of gradual pattern GP is given by the following formula: $_{supp}(GP) = \frac{max_{L \in l}(|L|)}{|R|}$, where L is the set of rows that when ordered, match the gradual pattern GP and R is the set of all rows in the data-set D.*

In order to determine the longest path, a precedence graph is built for the pattern considered as shown in Figure 1. The precedence graph shown can also be represented in a binary matrix, which allows to optimize the computations. Let us consider the graph for the pattern $\{(exercise, \uparrow), (stress, \downarrow)\}$, there is one long path in this instance: $<$ r1, r2, r5 $>$. Therefore the support is equal to $\frac{3}{5}$.

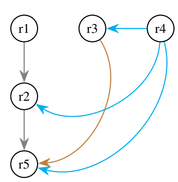


Figure 1: Precedence graph for $\{(exercise, \uparrow), (stress, \downarrow)\}$

Similarly, the technique of gradual ranking based on Kendall's τ can be applied on the data-set in Table I, in order to compute concordant pairs for the pattern $\{(exercise, \uparrow), (stress, \downarrow)\}$. The possible number of ordering pairs is given by the formula: $\frac{n(n-1)}{2}$ where n is the number of tuples.

In our case, the possible ordering pairs are 10: [r1,r2], [r1,r3], [r1,r4], [r1,r5], [r2,r3], [r2,r4], [r2,r5], [r3,r4], [r3,r5] and [r4,r5]. Using these orderings, the concordant pairs for pattern $\{(exercise, \uparrow), (stress, \downarrow)\}$ are 8: [r1,r2], [r1,r2], [r1,r5], [r2,r5], [r3,r5], [r4,r2], [r4,r3] and [r4,r5]. Therefore, the support is $\frac{8}{10}$

III. Related Work

According to [7], temporal data mining concerns the analysis of events ordered by one or more dimensions of time. Further, they distinguished the field by two main areas: one involves discovering similar patterns within the same or among different time sequences; the other involves discovering causal relationships among temporally-oriented events.

The first area, also known as *trend analysis* has been a field of active research for a long time. Srikant and Agrawal are among the first contributors, they tried to solve the problem of 'absence of time constraint' in an algorithm that they had introduced earlier 'AprioriAll' [8], for *discovering sequential patterns* [9], [10].

For instance, *a shop does not care if someone bought 'bread', followed by 'bread and jam' three weeks later; they may want to specify that a customer should support a sequential pattern only if adjacent elements occur within a specified interval, say 3 days. (So for a customer to support this pattern, the customer should have bought 'bread and jam' within 3 days of buying 'bread').*

So, [9] proposed a new algorithm known as 'Generalized Sequential Pattern' (GSP) that allowed users to set a *time gap* that was used to generate candidates for the frequent pattern. GSP was 5 times faster than AprioriAll since it counted less candidates. However, [11] proposed a more efficient algorithm than GSP known as 'Graph for Time Constraint' (GTC). GTC handled time constraints prior to and separately from the counting step of the data sequence, thus making it perform faster.

The latter area, discovering causal relationships can easily be conceptualized by a gradual pattern that correlates the causal effect among gradual items. As an illustration, [12] proposed algorithms that could mine evolution patterns and gradual trends such as: *An increasing number of purchases of jam during a short period is frequently followed by a purchase of bream a few days later.* The overall principle entailed converting a quantitative database into a variation database, which was converted into a membership degree database (also known as the *trend database*) which is mined for evolution patterns.

Similarly, [13] proposed an a novel algorithm called, GSTD that combined two concepts: gradual patterns and spatio-temporal pattern to extract gradual-spatio-temporal patterns. This algorithm can be used to mine frequent moving objects such as: *the more time is going on, the more objects are moving from east to west.* They achieved this by defining a gradual-spatio-temporal rule that provided notations for direction (positive or negative) and time duration and object set variation.

IV. Proposition

In our work, we aim at extending gradual patterns in such a way that they can include the temporal correlations between attributes. For instance "the more the exercise increases, the more the stress decreases **2 weeks later**" denoted as: $\{(exercise, \uparrow), (stress, \downarrow)_{+=2weeks}\}$. We also intend to extend our gradual rules to include fuzzy constraints such as "the more the exercise increases, the more the stress decreases **almost 2 weeks later**" denoted as: $\{(exercise, \uparrow), (stress, \downarrow)_{+\simeq 2weeks}\}$.

Definition 5. *Time Lag. A time lag is the amount of time that elapses before or after the changes in a one gradual item affects the changes in another gradual item. Time lag is denoted as* $\alpha\beta t$ *where* α *is an operator* $\alpha \in \{+, -\}$ *and '*$+$*' implies after/later and '*$-$*' implies before/earlier;* β *is an operator*
$\beta \in \{=, \simeq\}$ *and '*$=$*' implies equal to and '*$\simeq$*' implies almost;* t *is the value of time lag and is given by the formula:* $t = Medial_{M \in m}$*, where* M *is a medial of sequence* m*, and* $m = (c_i r_1 - c_i r_{1+k}), ..., (c_i r_n - c_i r_{n+k})$ *where* c *is the column for time/date,* r *is a single tuple/row,* $i = 1, 2, ..., I$*,* $n = 1, 2, ..., N$*, and* $k = 1, 2, ..., K$*.*

Remark 1. In definition 5, we consider the *medial* value of the sequence as the approximation for *time lag* when the largest proportion of members are split around it.

Definition 6. *Temporal Gradual Item. A temporal gradual item is made up of two parts: a gradual item and a time lag, denoted by* $(i, v)_{\alpha\beta t}$ *where* (i, v) *is a gradual item and* $\alpha\beta t$ *is a time lag where* $\beta \in \{=\}$ *so that '*$+ = t$*' implies a time lag of* t *later and, '*$- = t$*' implies a time lag of* t *earlier.*

Definition 7. *Fuzzy-Temporal Gradual Item. A fuzzy-temporal gradual item is a temporal gradual item with a fuzzy time lag* $\alpha\beta t$ *where* $\beta \in \{\simeq\}$ *so that '*$+ \simeq t$*' implies a time lag of almost* t *later and, '*$- \simeq t$*' implies a time lag of almost* t *earlier.*

Example 4. $(stress, \downarrow)_{+=2weeks}$ is a temporal gradual item interpreted as the "the less the stress 2 weeks later".

Example 5. $(exercise, \uparrow)_{-\simeq 1week}$ is a fuzzy-temporal gradual item that can be interpreted as the "the more the exercise almost 1 week earlier".

Definition 8. *Temporal Gradual Pattern. A temporal gradual pattern consists of one reference gradual item-set together with a set of temporal gradual items, denoted by* $TGP = \{(i_1, v_1), (i_2, v_2)_{\alpha\beta t_2}, ..., (i_n, v_n)_{\alpha\beta t_n}\}$*.*

Definition 9. *Fuzzy-Temporal Gradual Pattern. A fuzzy-temporal gradual pattern consists of one reference gradual item and a set of fuzzy-temporal gradual items, denoted by* $TGP_f = \{(i_1, v_1), (i_2, v_2)_{\alpha\beta t_2}, ..., (i_n, v_n)_{\alpha\beta t_n}\}$*.*

Remark 2. In order for definitions 7 and 8 to be relevant, there must be one *reference gradual item*. A *reference gradual item* is the anchor gradual item selected by a user and, from which other temporal gradual items in the item-set are varied with respect to time.

Example 6. $\{(jogging, \uparrow), (walking, \uparrow)_{-=1week}, (stress, \downarrow)_{+\simeq 2weeks}\}$ is a fuzzy temporal gradual item-set that can be interpreted as "the more the jogging, the more the walking 1 week earlier, the less the stress almost 2 weeks later".

In the first place, our goal is to transform data-sets into a temporal format that allows for extraction of gradual patterns with the corresponding time information. We intend to achieve this by modifying an existing gradual pattern mining.

However, it is very critical to highlight here that temporal correlations between attributes introduce more possible combinations of *gradual patterns* within a given data-set. For instance, the notation $\{(A, \uparrow)_{+/-=time}, (B, \downarrow)_{+/-=time}\}$ introduces 4 possible pattern combinations if attributes A and B are used as reference item-sets interchangeably and *time* is held to a constant value.

V. Data Transformation

In this section, we will demonstrate how a typical data-set can be transformed into a temporal format in order to allow for extraction of *temporal gradual patterns*. The raw (or non-transformed) data in the data-set should be chronologically ordered with respect to time.

Definition 10. *The representativity of a temporal gradual pattern* TGP *is given by the formula:* ${}_{rep}(TGP) = \frac{|N|}{|R|}$*, where* N *is the set of all rows in the transformed data-set* D' *and* R *is the set of all rows in the original data-set* D*.*

Definition 11. *Given a threshold of minimum representativity* δ*, a temporal gradual pattern* TGP *is said to be relevant if* ${}_{rep}(TGP) \geq \delta$*.*

Remark 3. Definition 10 and 11 also hold for *fuzzy-temporal gradual patterns* which is denoted as TGP_f. In Section V-A, we propose an algorithm for transforming data based on the *representativity threshold* set by the user.

Example 7. we consider a data-set containing the number of hours a person spent performing physical exercises together with the stress levels after irregular number of days. It is important to note that the values in Table II are arbitrary.

id	**date** (day/month)	**exercise** (hours)	**stress levels**
r1	01/06	1	4
r2	04/06	2	2
r3	05/06	3	3
r4	10/06	1	2
r5	12/06	3	3

Table II: A sample data-set D_3

A. The Data Transformation Algorithm

In this section, we propose an algorithm calculates the number of possible transformations based on the *representativity threshold*, and extracts the temporal gradual patterns for each transformation. $T-GRAANK$ is the proposed algorithm for mining temporal gradual patterns, see Section VII-A.

The main goals of the algorithm are: to calculate the *time lags* for each transformation, and to generate a new data-set table that includes columns that have been restructured (*step-wise*) while excluding *the Time column*. The *time lags* in the *Time column* are separately processed by a fuzzy modality, see Section VI.

Algorithm 1: Mining temporal gradual patterns

```
Input  : D− data set, refColumn− reference column, minSup− minimum support,
         minRep− minimum representativity
Output: F− set of Frequent Gradual Patterns, T_l− corresponding approximated time lag

sMax ← maximum number of steps w.r.t minRep;
rMax ← totalRows (D);
for s ← 1 to sMax do
    for i ← 0 to (rMax − s) do
        d ← Cell_[i+s] − Cell_[i];                    /* time is in the 1st column */
        tempRow.append (refColumn);
        cMax ← totalCols (Row_[i]);
        for j ← 1 to cMax do
            ;                                          /* excluding 1st column */
            if Column_[j] is same as refColumn then
                skip;
            else
                tempRow.append (Cell_[j][i+s]); /* Cell_[col][row] in D */
            end if
        end for
        D'.append (tempRow);
        T_d.append (d);
    end for
    F, T_l ← T-GRAANK(D', T_d, minSup);
    display F, T_l;
end for
return D', T_d
```

Let us transform Table II (using Algorithm 1) so that we compare the hours of exercise in r_n with the corresponding stress level in r_{n+1}, as illustrated in Table III.

id	days *lag* $(r_n - r_{n+1})$	exercise (r_n)	stress (r_{n+1})
t1	3	1	2
t2	1	2	3
t3	5	3	2
t4	2	1	3
t5	-	-	-

Table III: Transformed data-set D'_3: transformation: r_{n+1}

In the first place, we determine the longest path that match gradual pattern $\{(exercise, \uparrow), (stress, \downarrow)\}$ for transformation r_{n+1}. Using the path <t2, t3>, the support is $\frac{2}{4}$. It is important to note that the rows that do not have values for stress level are removed from the computation.

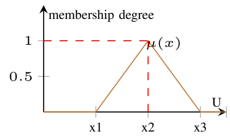
We observe that transformation r_{n+1} represents 4 out of 5 tuples. On the negative side, there is a decrease in the *representativity* of the data as we progress our transformations to larger *time gaps*. However, *representativity* has a less significant effect on large data-sets because of their great number of tuples.

Next, we determine the *time lag* between transformation r_n and transformation r_{n+1} and at this point we observe that the *'day lags'* in Table III vary. In order to approximate the most relevant *time lag*, we apply fuzzy logic which is described in the section that follows.

VI. Building The Fuzzy Modality

First thing to remember is that there exists a great number of membership modalities that one can build functions from (for instance triangular, trapezoidal, Gaussian among others), and it is very difficult to determine which one will fit the data-set perfectly. However, it is enough to pick modalities that span the whole universe and remain scalable [2], [14], [15], [16].

We recommend a triangular membership function (described in Figure 2), because we are interested in approximating the value *medial time lag*. Such that when this *value* is taken as the center membership function, the function should include a majority of the members without extending its boundaries.



$$\mu(x) = \begin{cases} \frac{x - x1}{x2 - x1} & \text{if } x1 \leq x < x2 \\ \frac{x - x3}{x2 - x3} & \text{if } x2 \leq x < x2 \\ 0 & Otherwise \end{cases}$$

the values must satisfy the condition: $x1 < x2 < x3$

Figure 2: A triangular membership function

The *TRUE* center of a distribution is established when the largest proportion of members are closely spaced around it [17]. In light of this, we initially can take the median as the center and slide the membership function left or right until we find the value that represents the *TRUE* center of the distribution, see proposed Algorithm 2.

Algorithm 2: Slide, re-calculate membership function

```
Input  : selTs− selected time-lags, allTs− all time-lags, minSup− minimum support
Output: q2− approximated medial value, sup− support

q1 ← quartile (1, allTs), q2 ← quartile (2, allTs), and q3 ← quartile (3, allTs);
boundaries ← append (q1, q2, q3);
left, right ← False, slice ← (0.1 ∗ q2), and sup ← 0;
while sup < minpSup do
    memberships ← fuzzTrimf (selTs, boundaries);
    sup ← countAverage (memberships);
    if sup >= minSup then
        return q2, sup;
    else
        if left == False then
            center ← minimum (selTs);
            if center <= q2 then
                q1 ← (q1 − slice), q2 ← (q2 − slice), q3 ← (q3 − slice);
                boundaries ← append (q1, q2, q3);
            else
                left ← True;
            end if
        else if right == False then
            center ← maximum (selTs);
            if center >= q2 then
                q1 ← (q1 + slice), q2 ← (q2 + slice), q3 ← (q3 + slice);
                boundaries ← append (q1, q2, q3);
            else
                right ← True;
            end if
        else
            return False, False;
        end if
    end if
end while
```

In our modality, the triangular membership function will initially have the median *time lag* as the center and minimum and maximum *time lags* as the extremes so that it spans the entire universe of data-set. Figure 3 shows the membership function for the transformed data-set D'_3 in Table III.

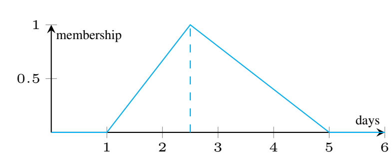


Figure 3: Membership function for r_{n+1}

When the membership function in Figure 3 is applied to the data-set in Table III, we generate Table IV. We observe that the membership degree support of '$\simeq 2.5$' for path <t2, t3> is $\frac{0}{2}$, in this case the support is less than half.

id	$\simeq 2.5days$ ($\in$)	exercise (r_n)	stress (r_{n+1})
t1	1	1	2
t2	0	2	3
t3	0	3	2
t4	.5	1	3
t5	-	-	-

Table IV: Transformed data-set $D_3^{\prime 1}$

As can be seen, the problem may be that the membership function in Figure 3, is either be too narrow or is pivoted on a wrong median value. We shy away from widening the function since increases the size of the universe. We recommend sliding the median then re-calculating the membership degrees.

For instance, we slide the membership function for transformation r_{n+1} to the left as shown in Figure 4. We observe that the support for '$\simeq 1.5$' for path <t2, t3> is $\frac{1}{2}$.

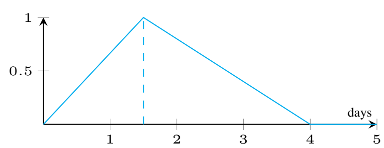


Figure 4: Modified membership function for r_{n+1}

The fuzzy-temporal gradual pattern $\{(exercise, \uparrow), (stress, \downarrow)_{+\simeq 1.5days}\}$ has a support of $\frac{2}{4}$, a *representativity* of $\frac{4}{5}$ and the *time lag*: '$\simeq 1.5$' has a support of $\frac{1}{2}$.

VII. Experiments

This section seeks to show that our proposed approach allows to discover new temporal knowledge in gradual patterns that previously could not be discovered using existing gradual pattern mining techniques.

A. The Proposed T-GRAANK Approach

The proposed algorithm known as T-GRAANK (denotes Temporal GRAANK) modifies the GRAANK algorithm proposed by [5] in order to extend its functionality to mining gradual patterns with temporal tendencies. More precisely, the algorithm works as illustrated in Algorithm 3:

It is important to mention that since the proposed algorithm is based on GRAANK, it inherits all the good that comes with it. For instance, the algorithm benefits from the computational efficiency and low computational complexity since it is also based on binary matrices.

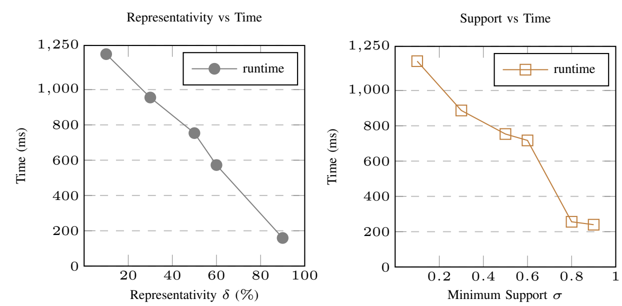
On one hand, the proposed algorithm seems to be more computationally intensive than the original GRAANK algorithm proposed by [5] because it executes 4 additional functions. On the other hand, the increase in computations can be justified by the fact that new knowledge about *time lag* is extracted which was not possible previously.

Algorithm 3: The $T - GRAANK$ algorithm

```
Input : D' − transformed data-set, T_d − time differences, minSup − minimum support
Result: F − set of Frequent gradual patterns, T_l − corresponding approximated time lags
foreach Attribute A in D' do
    G ← build concordance matrices A↑ and A↓;
end foreach
G' ← APRIORIgen (G);   /* generates frequent gradual item-set candidates */
foreach Candidate C in G' do
    sup ← calculateSupport (C);
    if sup < minSup then
        discard C;
    else
        pos_indices ← concordantPositions (C_pairs);
        time_lags ← timeDifferences (pos_indices, T_d);
        boundaries ← buildTriMembership (T_d);
        t_lag ← fuzzyFunc (time_lags, boundaries);
        F.append (C);
        T_l.append (t_lag);
    end if
end foreach
return F, T_l;
```

B. Short Performance Analysis

The runtime performances of our algorithm for temporal gradual pattern mining shown in Figure 5 were obtained from the execution of dummy data containing 50 tuples and 2 gradual items. The runtime values were generated by a python code that recorded the start-time and stop-time.



(a) (b)

Figure 5: (a) Temporal gradual pattern mining runtime according to *minRep*, (b) temporal gradual pattern mining runtime according to *minSup*

In Figure 5 (a), as the *representativity* threshold is decreased, the number of data-set transformations (or number transformed data-sets) to be mined increase hence the increase in *runtime*. In Figure 5 (b), as minimum support threshold is decreased, the number of possible gradual patterns increase which in turn increases the number of scans in the data-set.

C. Results for Temporal Gradual Patterns

In order to test the efficacy of the $T - GRAANK$, we performed two separate tasks related to weather and compared their results. The aim was to confirm the conclusions of [18], that the NDVI (Normalized Difference Vegetation Index) is a sensitive indicator of the inter-annual variability of rainfall in the East African region.

In the first task, we retrieved the historical rainfall distribution amounts for 4 towns in Kenya (October-December 2013 and 2015) from Kenya Meteorological Service weather report in [19], [20], shown in Table V. Here, we selected two observable patterns: $\{(MAK, \downarrow), (WAJ, \uparrow)\}_{+=2years}$ and $\{(ELD, \uparrow), (NRB, \uparrow)\}_{+=2years}$, see also http://www.meteo.go.ke/index.php?q=archive.

town	**amount**	
	2013	2015
MAK	104	75
WAJ	49	69
ELD	174	200
NRB	44	223

Table V: Rainfall distribution in Kenya

In the second task, we first generated NDVI data (for year 2013 and 2015) from LANDSAT 7 satellite images over Kenya through a novel tool known as *data-cube*. The *data-cube* is a great tool for the expanded use of satellite data in an Open Source framework, see also https://www.opendatacube.org.

Lastly in the second task, we applied our approach on the NDVI data and we obtained the results shown in Table VI. It can be seen that the patterns built by our algorithm match the selected patterns in Table V; except, the *time lag* is slightly less for pattern $\{WAJ+, MAK-\}$.

Ref. Item	**Pattern : Sup**	**Time : Sup**	**Rep**
NRB	$\{ELD+, NRB+\} : 0.666$	$\approx +1.999yrs : 1.0$	50%
	$\{WAJ+, NRB+, MAK+\} : 0.666$	$\approx +1.999yrs : 1.0$	50%
WAJ	$\{ELD+, WAJ+\} : 0.600$	$\approx +1.223yrs : 0.5$	62.5%
	$\{WAJ+, MAK-\} : 0.600$	$\approx +1.747yrs : 0.5$	62.5%

Table VI: NDVI Temporal Pattern Results

We note here that it is difficult to get clear satellite images after short intervals due to cloud coverage; therefore, the *data-cube* tool runs an algorithm that re-creates the image based on previous images. It may be for this reason that the time approximation for pattern $\{WAJ+, MAK-\}$ is slightly less than 2 years.

VIII. Conclusion

In this paper, we propose an approach for extending the existing GRAANK algorithm in order to extract fuzzy temporal gradual patterns. This approach integrates two main areas: fuzzy logic and mining gradual patterns (with temporal tendencies). We provide formal definitions for temporal and fuzzy temporal patterns based on existing formal definitions for gradual patterns. Further, we demonstrate a step-wise transformation of the data-set and recommend a technique for approximating the medial *time lag* using a fuzzy membership function.

Apart from extensive experimentation, including both computation efficiency and semantics (relevance of the extracted patterns), further works include scaling and optimizing the technique in order to allow for multi-level temporal gradual pattern extraction. Moreover, we aim at improving the fuzzy modality in order to increase its accuracy in perfectly fitting the distribution of *time lags*.

Acknowledgment. The authors would like to thank the members of staff at the Co-operation and Cultural Service, Embassy of France in Kenya and Campus France (Montpellier) for their involvement in creating the opportunity for this work to be produced.

References

[1] Y. S. Aryadinata, Y. Lin, C. Barcellos, A. Laurent, and T. Libourel, "Mining Epidemiological Dengue Fever Data from Brazil: A Gradual Pattern Based Geographical Information System," *Communications in Computer and Information Science*, vol. 443 CCIS, pp. 414–423, 2014.

[2] S. Ayouni, S. B. Yahia, A. Laurent, and P. Poncelet, "Fuzzy gradual patterns: What fuzzy modality for what result?" *Proceedings of the 2010 International Conference of Soft Computing and Pattern Recognition, SoCPaR 2010*, pp. 224–230, 2010.

[3] Y. S. Aryadinata, A. Laurent, and M. Sala, "M2LGP : Mining Multiple Level Gradual Patterns," vol. 7, no. 3, pp. 353–360, 2013.

[4] L. Di-Jorio, A. Laurent, and M. Teisseire, "Mining frequent gradual itemsets from large databases." Berlin, Heidelberg: Springer-Verlag, 2009, pp. 297–308.

[5] A. Laurent, M.-J. Lesot, and M. Rifqi, "Graank: Exploiting rank correlations for extracting gradual itemsets," in *Proceedings of the 8th International Conference on Flexible Query Answering Systems*, ser. FQAS '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 382–393.

[6] F. Berzal, J. C. Cubero, D. Sanchez, M. A. Vila, and J. M. Serrano, "An alternative approach to discover gradual dependencies," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 15, no. 05, pp. 559–570, 2007.

[7] J. F. Roddick and M. Spiliopoulou, "A bibliography of temporal, spatial and spatio-temporal data mining research," *SIGKDD Explor. Newsl.*, vol. 1, no. 1, pp. 34–38, Jun. 1999.

[8] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering*, ser. ICDE '95. Washington, DC, USA: IEEE Computer Society, 1995, pp. 3–14. [Online]. Available: http://dl.acm.org/citation.cfm?id=645480.655281

[9] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Advances in Database Technology — EDBT '96*, P. Apers, M. Bouzeghoub, and G. Gardarin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 1–17.

[10] C. Chand, A. Thakkar, and A. Ganatra, "Sequential pattern mining : Survey and current research challenges," 2012.

[11] F. Masseglia, P. Poncelet, and M. Teisseire, "Efficient mining of sequential patterns with time constraints: Reducing the combinations," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 2677 – 2690, 2009.

[12] C. Fiot, F. Masseglia, A. Laurent, and M. Teisseire, "Evolution patterns and gradual trends," *International Journal of Intelligent Systems*, vol. 24, no. 10, pp. 1013–1038, oct 2009.

[13] N. H. Phan, P. Poncelet, and M. Teisseire, "Moving Objects: Combining Gradual Rules and Spatio-Temporal Patterns," no. March, 2013.

[14] S. Nath Mandal, J. Choudhury, and S. Bhadra Chaudhuri, "In Search of Suitable Fuzzy Membership Function in Prediction of Time Series Data," *International Journal of Computer Science Issues*, vol. 9, pp. 293–302, 2012.

[15] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.

[16] S. Schockaert, M. D. Cock, and E. E. Kerre, "Fuzzifying allen's temporal interval relations," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 2, pp. 517–533, April 2008.

[17] D. Montgomery and G. Runger, *Applied Statistics and Probability for Engineers*, 3rd ed. John Wiley and Sons, Inc, 2003.

[18] M. L. Davenport and S. E. Nicholson, "On the relation between rainfall and the normalized difference vegetation index for diverse vegetation types in east africa," *International Journal of Remote Sensing*, vol. 14, no. 12, pp. 2369–2389, 1993.

[19] K. M. S. Archive1, "Review of weather in kenya during the october-december 2013," Press Release, Nairobi, Kenya, Mar 2014, *The Kenya Meteorological Service*, pages 5-6. Ref No: KMD/FCST/5-2014/SO/01.

[20] K. M. S. Archive2, "Review of weather in kenya during the october-december 2015," Press Release, Nairobi, Kenya, Mar 2016, *The Kenya Meteorological Service*, pages 5-6. Ref No: KMD/FCST/5-2016/SO/01.