A fuzzy approach for the model of sliding window. An application to behaviour patterns mining

María Ros, Miguel Delgado, Amparo M. Vila[1]

1.Department of Computer Science and A.I. University of Granada, España
Email: {marosiz, mdelgado, vila}@decsai.ugr.es

Abstract— *Many traditional environment applications base their operations on using sensor. For instance, the Tagged World uses the information from sensors to identify user behaviour and to provide services. We present a method to limit an user behaviour through external user knowledge: a Temporal Window. However, an user behaviour has random imprecision by definition. Thus, we have adapted the method to manage the imprecision, Fuzzy Temporal Window, which includes Fuzzy Logic concepts. As an application, we present a method to extract behaviour patterns on time defined by a Fuzzy Temporal Window.*

Keywords— Behaviour, Fuzzy Windows, Quantified Sentences, Tagged World, Frequent Itemsets

1 Introduction

In the last years, Smart Computers have suffered a great evolution that is transforming the world. *Ubiquitous Computing*[1] uses them to make user daily life easier. Computers are included on daily life without disturbing normal user activities. Different devices are added in traditional environment from common PDA or Mobile Phone to sensors. Nowadays, the last trend in *Ubiquitous Computing* is to provide the environment with different mechanisms to get information.

Sensors are devices that are able to know what is happening in the environment. There are many kind of sensors from temperature sensor to mechanical one. We also find RFID sensors [2] that identify every object in the environment with an unique number by means of radio frequency. Placing RFID tags in an environment we get to construct an intelligent space in which everyone can enjoy services. An intelligent space obtains user position, user behaviour, and environment information around a user and so on by a sensor network [3]. Other example about these spaces can be found in [4]. We find an intelligent space that provides services in a variety of scenes, but hardly ever, without reasoning or inference. The development of these projects is the Tagged World concept.

In [5, 3], the Tagged World project is presented. This project is developed in the University of Ritsumeikan, Japan and consists on providing appropriate personalized services for each user, to make their life easier and safer by recognizing and reasoning the human behaviour. They use a wearable computer as a Pocket Assistant that compares an access log with patterns to recognize human activities. The system is based on a Bayesian network and obtains results with a probability value for every extracted behaviour.

Other alternative is proposed by Philipose and et al. [6]. They proposed a system to infer Activities of Daily Living in a Tagged World. They present a new paradigm for ADL inferencing leverage radio-frequency-identification technology, data mining and a probabilistic inference engine to recognize ADLs based on objects that people use. As sensor, they use RFID technology with other sensor streams to fill in the gaps. The system represents activities as linear sequences of activity stages, and annotates each stage with the involved objects and the probability of their involvement.

In [7][8] is proposed a system to identify correct behaviour using Data Mining Techniques. The system is divided into two main parts: inductive learning mechanism, which produces a behaviour database and a reasoning system for the recognition of sequences that uses this database. The first stage uses Frequent Itemsets, while the second one does Regular Grammar.

This paper is organized in six main sections. In Section 2 we present the formal problem to solve. In Section 3, Fuzzy Temporal Window concept is explained. Section 4 presents the method that manage problem uncertainty. In section 5, we show some empirical results, comparing crisp and fuzzy models. Finally, the conclusions and future works are reported in section 6.

2 Formal Problem

This section defines the formal problem: to obtain sequence patterns to identify user behaviour that are defined on a specific domain and context. Thus, we have to define some basic concepts which are the objectives of the system.

DEFINITION **2.1 (Action)** *An action, a, is an activity that happens using a specific object.*

However, it results very interesting to know when an action happens on time.

DEFINITION **2.2 (Action on time)** *An action on time, a, is an activity that happens over a specific object in a known time and it is denoted as a pattern $a = (h, l)$, where h is a fact and l is a temporal label that defines the time of the action.*

These definitions give us a basic element to work, but our aim is to find different actions that make a behaviour up.

DEFINITION **2.3 (Behaviour)** *Let $A = \{a_1, ..., a_n\}$ be set of possible user actions in some situations or domains. An user behaviour is a finite set of actions:*

$$\beta = \{\alpha_1, \alpha_2, , ...\alpha_{p(\beta)}\}$$

with $\alpha_j \in A \forall j$, and where α_j is performed before α_k iif $j \leq k$.

ISBN: 978-989-95079-6-8

In this definition, we do not pay attention on the time, although, in general, a behaviour happens in a specific moment on time.

DEFINITION **2.4 (Behaviour on time)** *Let $A = \{a_1, ..., a_n\}$ be set of possible user actions in some situations or domains. Let τ be the temporal line, then A user behaviour on time is a finite set of pair such as:*

$$b_I = \{(\alpha_1, t_1), ...(\alpha_{n(b)}, t_{n(b)})\}$$

where $\alpha_j \in A \forall j$, where α_j is performed before α_k iif $j \le k$, $t_j \in [0, \tau]$ $\forall t_i < t_j$ if $i \ge j$, with $I = [t_1, t_{n(b)}]$

The System is based on a database about user actions. This database has been obtained from user observation, i.e., from actions that user has done. In rest of this document, we name this database as Observation Data Base (ODB).

The representation of ODB is a transactional database T, where every row is an observation over an user. Normally, we study his activity in a whole day; and every column is a possible action from A set, $A = \{a_1, ..., a_n\}$.

3 Fuzzy Temporal Windows. Extract behaviour patterns tool.

In previous section, we have presented behaviour sequence pattern problem. In [7][8] we study behaviours in a crisp way, however the human behaviour is not crisp. Human activity has random imprecision by definition.

3.1 Motivations

In [7][8] we present a process to obtain correct sequence patterns of actions from user behaviour. This method extracts from a particular ODB common actions. A general ODB contains actions that a user has realized for a whole day. For this reason, we do not know where a behaviour starts or ends. The proposed method needs an expert to indicate when a behaviour is performed. This proposal is not applicable in a real system, since every person has his/her specific activities and habits.

Therefore, we have to design a mechanism to identify the interval of the ODB that we have to study. In this point, we use a knowledge about the behaviour :generally, there exist some behaviours that are realized by user at the same time every day.

EXAMPLE **3.1** *Luis leaves home at 8:30 o'clock.*

We use this knowledge to situate an interval on the temporal line. The result is a subset from particular ODB to the studied behaviour.

In spite of this fact, nobody usually does an action at the exactly moment every day. They usually do actions roughly at the same moment. This raises a new problem: how we fix the interval's ranges to detect a specific activity. Ranges should consider random imprecision of the situations.

EXAMPLE **3.2** *Let us suppose that* Luis often leaves home at 8:30 o'clock*, then we could control actions that happen about 8:30 (from 8:20 to 8:40, for instance).*

We name the interval defined over temporal line as *Temporal Window*. It permits to get a subset from each ODB tuple (See figure 1). In this subset, we know that actions have not

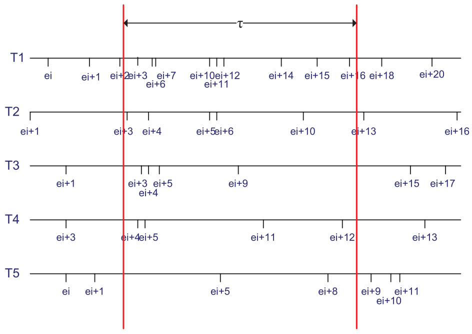

Figure 1: Temporal Window over a ODB

the same degree of importance, since actions are about the specific time are more important than actions far from the interval centre. So, we could assign a degree of importance at each action, in function of its situation in the interval. Then, we can define the interval as a fuzzy set and name it as *Fuzzy Temporal Window* (See figure 2).

3.2 Model Formulation

In this section we present a formal representation to *Temporal Window* and *Fuzzy Temporal Window*.

DEFINITION **3.1 (Temporal Window, W)** *Let I be a interval from the temporal line τ, ODB the Observation Data Base and $t \in ODB$ a tuple from ODB. Let i_j be an action $i_j = (h_j, l_j)$, then a Temporal Window, W, for a specific behaviour, which happens in interval I of τ, is defined as a subset of t where*

$$\forall i_j \in W(t) \text{ then } i_j \in t \text{ and } l_j \in I$$

DEFINITION **3.2 (Fuzzy Temporal Window, FW)** *Let I be a interval from the temporal line τ, ODB the Observation Data Base and $t \in ODB$ a tuple from ODB. Let i_j be an action $i_j = (h_j, l_j)$ and a Temporal Window, W, to a specific behaviour. Let fs be a fuzzy set over τ, then it defines a* Fuzzy Temporal Window*, FW, as a Temporal Window where*

$$\forall i_j \in W(t)\ \mu_{FW}(i_j) = \mu_{fs}(l_j)$$

If we apply a Fuzzy Temporal Window over the ODB, we obtain an image where each item has its membership degree. As we have represented an ODB as a Transactional Database, we obtain a Transactional Database image too.

DEFINITION **3.3 (Fuzzy ODB applying W)** *Let ODB be an Observation DataBase and W a Fuzzy Temporal Window, it defines $\tilde{ODB}$ as a Fuzzy Observation Data Base constructs as W(ODB) such as*

$$\forall t \in ODB, W(t) \in \tilde{ODB}$$

DEFINITION **3.4 (Fuzzy Transactional Data Base)** *Let T be a Transactional Data Base and W a Fuzzy Temporal Window, it defines $\tilde{T}$ as a Fuzzy Transactional Data Base constructs as W(t) $\forall t \in T$.*

We represent the Fuzzy Transactional DataBase as a table where for every row and column we have membership degree corresponding to the Fuzzy Temporal Window.

ISBN: 978-989-95079-6-8

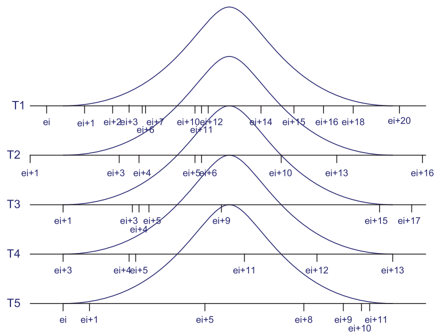

Figure 2: Fuzzy Temporal Window over a ODB

3.3 Extending the Fuzzy Temporal Windows.

Until this point, we establish a Fuzzy Temporal Window with previous behaviour knowledge. We suppose this knowledge is completely true, although this statement always is not correct.

EXAMPLE **3.3** *Let us suppose that we want to check the behaviour* `Luis leaves home at 8:30 o'clock`. *However, we know that Luis sometimes arrives late at work. So, we can affirm that* `Luis leaves home at time almost always`.

In the example 3.3, we give a quantified adjective to actions of a behaviour to express certain. We employ a semantic approach based on evaluation of quantified sentences [9]. A quantified sentence is an expression of the form "Q of F are G", where F and G are two fuzzy subsets of a finite set X, and Q is a relative fuzzy quantifier. Some examples could be (The most times, Luis leaves home, 8:30), (Almost never, Luis leaves home, on time), (Almost all times, Luis leaves home, 8:30).

This knowledge is used to expand the Fuzzy Temporal Window size. We use the method presented in [10]. This method transforms the used function to define a specific behaviour according with the knowledge expressed by a quantified sentence. The proposed process obtains a new window F' in two steps:

1. Firstly, they truncate a fuzzy number, our Fuzzy Temporal Window, using the certainty degree α associated to the fuzzy value A. After this operation, we obtain a non normalized fuzzy set A^α.

2. Secondly, they normalize the fuzzy set. The authors assume that uncertainty is being translated into imprecision under the condition of the amount of information provided by the fuzzy number remains equal before and after normalization process.

The transformation function is defined as:

DEFINITION **3.5** *Let $A \in \tau$ be a fuzzy number such that $A = \{(m_1, m_2, a, b), \alpha_A\}$, where m_1, m_2, a, b are the values that defines a trapezoidal fuzzy number and α_A is the height of A. Let $\alpha \in (0, 1]$. We will denote $\triangle(\alpha_A, \alpha) = \triangle$ and define*

$$T_\alpha(A) = \left\{ \left(m_1, m_2, a + \frac{\triangle}{k}, b + \frac{\triangle}{k} \right), \alpha \right\} \quad (1)$$

for those α in which the transformation makes sense.

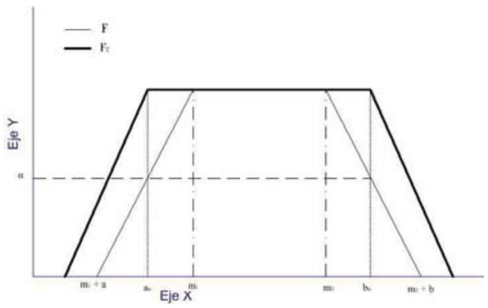

Figure 3: Transformations over a Trapezoidal Fuzzy Temporal Windows

To use the definition 3.5, we need to define two basic parameters:

- k.- scale parameter
- α.- value for doing the transformation

In figure 3 we represent the followed process over a trapezoidal window.

In this point, we introduced the quantified sentences over a Fuzzy Temporal Windows, using their evaluation to know α value. When we have this value, we transform A set in its α version (A^α), which is the basis of the final transformation A^T. Example 3.4 shows the followed process. We use a trapezoidal Fuzzy Temporal Window for the sake of clarity.

EXAMPLE **3.4** *Let us suppose that we want to check the behaviour* `Luis hardly ever leaves home at 8:30 o'clock`. *Thus, our Fuzzy Temporal Windows, represented as a lr-number as $F = (m_1, m_2, a, b)$ where m_1, m_2 is the higher points of a trapezoidal fuzzy number, and a, b are the time to establish the lower points.*

$$F = (8:25, 8:35, 0:05, 0:05) \quad (2)$$

Firstly, we have to evaluate a quantified sentence to find out the α-value. There exists a lot of ways to evaluate it. Here we have selected the basic way, but not the best: Zadeh. The expression of the Zadeh Cardinal is $\alpha_A = \frac{P(A)}{|X|}$ where $P(A) = sum_{x \in X} A(x)$ and $|X|$ is the X set cardinal.

So, $\alpha_A = \frac{15}{20} = 0.75$. The evaluation of the quantified sentence is $Z_Q(A) = Q(\alpha_A) = 0.53$.

Next, we do the transformation of our windows using α value.

$$F = \{(8:25, 8:35, 0:05, 0:05), 1\} \rightarrow \quad (3)$$

$$F^\alpha = \left\{ \left(\begin{array}{c} 8:25 - 0:05(1-\alpha), 8:35- \\ 0:05(1-\alpha), 0:05\alpha, 0:05\alpha \end{array} \right), \alpha \right\} \rightarrow \quad (4)$$

$$F^T = \left\{ \left(\begin{array}{c} 8:25 - 0:05(1-\alpha), 8:35 - 0:05 \\ (1-\alpha), 0:05\alpha - \frac{1-\alpha}{k\alpha}, 0:05\alpha - \frac{1-\alpha}{k\alpha} \end{array} \right), 1 \right\} \quad (5)$$

Replacing α value with $Z_Q(A)$, i.e., $\alpha = 0.53$ we expand our Fuzzy Temporal Windows to

$$F = (8:23, 8:37, 0:08, 0:08) \quad (6)$$

ISBN: 978-989-95079-6-8

4 Fuzzy method to extract behaviour patterns. Obtaining sequence patterns by alpha-cuts

Once we have information, we should get common actions and valid sequences for every behaviour. However, we can not apply the method explained in [7] [8] directly, because we have to manage membership degree corresponded to the Fuzzy Temporal Window.

Here, we cannot present our Crisp model but we have to clarify some basic concepts of it that are need to can understand the process.

4.1 Previous concepts

In this section, we want to specify some important concepts: common actions and valid sequences.

1. Common actions: We define the common actions as a sequence of events that occurs more often in the observed knowledge. We can identify the common actions with the concept of Frequent Itemset: every frequent itemset corresponds to a particular common behaviour [8]. However, an itemset is a set and, so, it has not order relationship among its elements. However, a basic condition for a behaviour is the existence of this relationship. Since, we have to control which sequences of these itemsets are valid.

2. Valid/Correct sequences We define the valid sequences as the permutations of the set whose order relationship among their elements appear in the knowledge of the *ODB*. We use the permutation concept and an designed algorithm to determine if a sequence is valid or not, since not all permutation will be accepted as a pattern sequence.

4.2 Changes to manage the random imprecision

In this subsection, we present the different changes to manage the random imprecision. Thus, the objective is to develop a system that obtains stimulation from user who realizes some actions sequentially. The problem consists on extracting the sequence patterns to specific behaviour when we have a Fuzzy ODB $\tilde{ODB}$, represented as $\tilde{T}$, and a Fuzzy Temporal Window W defined from user knowledge.

Firstly, we have to transform a fuzzy problem to a crisp problem to apply the method explained in [7] [8]. We could use α-cut set [11]. By the α-cut, we would have a new crisp image of $\tilde{T}$, T^α where every value is in $\{0,1\}$. Then, we apply classical method to T^α, obtaining frequent itemsets and sequence patterns to specific α value (I^α and P^α, respectively). After extracting all sequence patterns, we create a fuzzy set **P** by sequence patterns which have obtained to every α-value applying the Representation Theorem.

As we use frequent itemsets, we need to represent extracted frequent itemsets as an unique fuzzy set. Thus, we have to ensure *consistent restriction* between every α frequent itemset I^{α_i}. For sequence patterns representation is the same. The proof of these statements can be studied in [8].

4.3 An illustrative example

In this example we want to show the operation of the method. We start from ODB which is collected from the touched object by the user in the daily activity[1].

Let W a Fuzzy Temporal Window defines by the following fuzzy set over temporal line τ:

$$W = \left\{ \begin{array}{c} \frac{0}{8:20}, \frac{0,2}{8:21}, \frac{0,4}{8:22}, \frac{0,6}{8:23}, \frac{0,8}{8:24}, \frac{1}{8:25}, \frac{1}{8:26} \\ \frac{1}{8:27}, \frac{1}{8:28}, \frac{1}{8:29}, \frac{1}{8:30}, \frac{1}{8:31}, \frac{1}{8:32}, \frac{1}{8:33}, \frac{1}{8:34} \\ \frac{1}{8:35}, \frac{0,8}{8:36}, \frac{0,6}{8:37}, \frac{0,4}{8:38}, \frac{0,2}{8:39}, \frac{0}{8:40} \end{array} \right\} \quad (7)$$

Then, we apply W over T and obtain a Fuzzy T $\tilde{T}$ where every time of actions is replaced with the membership degree value in W. We apply the α-cut for α values α_1 = 0.4, α_2 = 0.6, α_3 = 0.8, α_4 = 1.0. We obtain for each α values the T^α, the result of apply the α-cut in $\tilde{T}$. After these operations, we have transformed the fuzzy problem to a crisp problem. Now, we extract frequent itemsets and sequence patterns with the model explained in [7] and [8].

We have executed the Apriori Algorithm with two support values: *minsup = 0.8 and minsup = 0.9*. The results are showed in tables 1 and table 2.

Since we have the common actions to specific behaviour, the next stage consists of obtaining the valid sequences patterns and the final representation as a fuzzy set applying the Identity Principle. The valid patterns are showed in table 3 and table 4. And the final pattern representation in equation 8 to minsup = 0.8 and equation 9 to minsup = 0.9.

Table 1: Frequent itemset to minsup = 0.8

I^{α_1}	{Shoes, Bag, Keys, ODoor, MobilePhone} {Keys, Keys2, ODoor, ODoor2, MobilePhone}
I^{α_2}	{Shoes, Bag, Keys, ODoor, MobilePhone} {Keys, Keys2, ODoor, ODoor2, MobilePhone}
I^{α_3}	{Shoes, Bag, Keys, ODoor, MobilePhone}
I^{α_4}	{Shoes, Bag, Keys, ODoor, MobilePhone}

Table 2: Frequent itemset to minsup = 0.9

I^{α_1}	{Keys, ODoor, MobilePhone}
I^{α_2}	{Keys, ODoor, MobilePhone}
I^{α_3}	{Keys, ODoor}
I^{α_4}	{ODoor}

Table 3: Valid patterns to minsup = 0.8

P^{α_1}	{Shoes, Bag, Keys, MobilePhone, ODoor} {Shoes, Keys, MobilePhone, Bag, ODoor} {Shoes, MobilePhone, Bag, Keys, ODoor} {Keys, MobilePhone, ODoor, Keys2, ODoor2} {MobilePhone, Keys, ODoor, Keys2, ODoor2}	p_1 p_2 p_3 p_4 p_5
P^{α_2}	IDEM	
P^{α_3}	{Shoes, Bag, Keys, MobilePhone, ODoor} {Shoes, Keys, MobilePhone, Bag, ODoor} {Shoes, MobilePhone, Bag, Keys, ODoor}	p_1 p_2 p_3
P^{α_4}	IDEM	

$$\tilde{P} = \left\{ \frac{p_1}{\alpha_4}, \frac{p_2}{\alpha_4}, \frac{p_3}{\alpha_4}, \frac{p_4}{\alpha_2}, \frac{p_5}{\alpha_2} \right\} = \left\{ \frac{p_1}{1}, \frac{p_2}{0.4}, \frac{p_3}{1}, \frac{p_4}{1}, \frac{p_5}{0.6} \right\} \quad (8)$$

[1]In this paper, we do not show the Data Base due to the lack of space.

ISBN: 978-989-95079-6-8

Table 4: Valid patterns to minsup = 0.9

P^{α_1}	{Keys, MobilePhone, ODoor}	p_1
	{MobilePhone, Keys, ODoor}	p_2
P^{α_2}	IDEM	
P^{α_3}	{Keys, ODoor}	p_3
P^{α_4}	{ODoor}	p_4

$$\tilde{P} = \left\{ \frac{p_1}{\alpha_2}, \frac{p_2}{\alpha_2}, \frac{p_3}{\alpha_3}, \frac{p_4}{\alpha_4} \right\} = \left\{ \frac{p_1}{0.6}, \frac{p_2}{0.6}, \frac{p_3}{0.8}, \frac{p_4}{1} \right\} \tag{9}$$

5 Performance Evaluation

In this section, our objective is to evaluate fuzzy model versus crisp model presented in [7][8]. In other hand, we present advantages of modifying the windows using quantified sentences. To do this proof, we used a database that represents daily activities of the user. To be precise, we focus study on the analysis of Leave Home behaviour, to simplify the process. We study two databases. These databases try to force the temporal model, thus they imply different temporal relations between their items. In the first one, the interval between two consecutive actions is always the same. In the second database, we break this temporal relation.

With these experiments, we want to control the fuzzy window influence over the database.

As well as, we need a goodness measure to analyze obtained outcomes. We design a measure to evaluate the method that works with a sequence of words. So, we have to design a way to transform these sequences in a numerical way. We have to design this measure because it tries to pay attention all important aspects: correction, rubbish, alpha, minsup, time, etc.

Therefore, we have to define a goodness measure that makes the result evaluation easier:

$$c = \alpha_1 correction + \alpha_2 alpha + \alpha_3 \frac{1}{time} \tag{10}$$

where,

- correction.- similarity measure of every pattern to the correct pattern. We define a function that associates a numeric value with sequences that a higher value indicates greater similarity.
- alpha.- mean of the alpha-values used for the α-cut for every pattern.
- time.- amount of millisecond that the method inverts in obtaining results.

Goodness measure is inversely proportion with time raises to the power of minus one and directly proportion with the correction and confidence measure. So, if the absolute value of the correction gets bigger, the absolute value of goodness measure gets bigger too.

For the experiments, we use a specific values for α. We have chosen this values because we believe they are rougher to the reality. We use: α_1 and α_2 equals 0.3 and α_3 equals 0.4.

5.1 Crisp model vs Fuzzy model

We were evaluating our system with three different fuzzy distribution: a trapezoidal distribution, a the Gauss bell shape with mean 40 minutes and a the Gauss bell shape with mean 60 minutes. We have used some distributions more, however we believe these distributions show the most interesting results. In figure 4 and 5, two graphics show goodness measure for every window. In addition, we show results of crisp model to compare the methods. If we observe results, we extract some conclusions which are convenient to study:

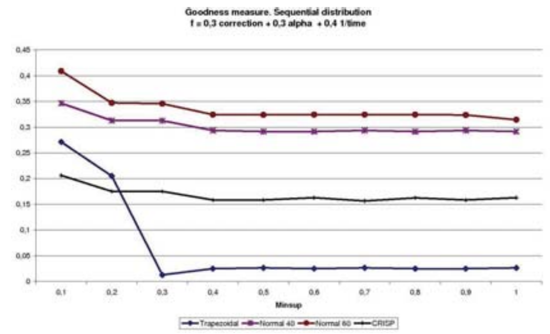

Figure 4: Crisp vs Fuzzy. Database a

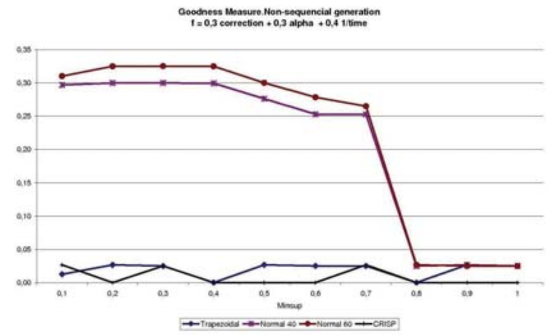

Figure 5: Crisp vs Fuzzy. Database b

- In general, we conclude that the Fuzzy model obtains better results than the Crisp one. The Fuzzy method includes a way to provide importance to some actions that are key but are irregular. On the other hand, the Crisp method consider all actions at the same importance, and can lose this key actions. In addition, the Crisp method has to study all the data, while the Fuzzy method can reduce the size of transactions and can do the elements become frequent, minimizing the computational cost. This reduction could be very interesting when we have a forgetful user, because we would have a database with many mistakes that we would have to avoid. Moreover, this size reduction of the database makes it easier to find if the order of the sequences are valid.
- Among the Fuzzy Temporal Windows, the best results are obtained with the Gauss bell shape. It relaxes the degree of membership of their elements to the fuzzy window. When we perform the cut with a specific α, we include more items to study. Instead, in the trapezoidal shape the α-cuts are more abrupt, then more elements are lost.
- For the second database, the Crisp method provide good results: if we study all possible data, you always get

ISBN: 978-989-95079-6-8

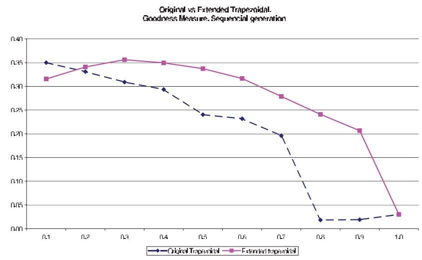

Figure 6: Original vs Extended Trapezoidal. Database a

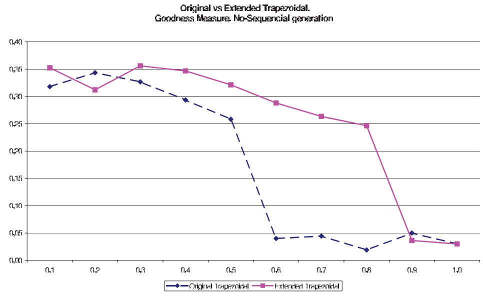

Figure 7: Original vs Extended Trapezoidal. Database b

key actions. However, with any Fuzzy Temporal Window with a Gauss bell shape we can extract almost the same results than the Crisp model minimizing the computational cost. Other alternative is to extend the variance of the fuzzy window, as we will see in the next subsection.

- Thus, the outcomes obtained with the fuzzy model are better than using crisp model.

5.2 *Analysis of process of extending the Fuzzy Temporal Window*

Once we have checked our fuzzy method, we analysed the process of extending the Fuzzy Temporal Window from a knowledge expressed as a quantified sentence. We are going to continue with example 3.4 that we proposed on section 3.3. We applied original Trapezoidal Temporal Window over two databases used in the previous section, and the new window obtained after expanding the window. Goodness measure is represented on figures 6 and 7: From these graphics, we observe that sometimes we extract results with the second window where the first window do not. This event happens because with the second window we have expanded the amount of data we studied to obtain the results. Although, always, we studied less data than the Crisp Method. Therefore, we adjusted the window to get better results than original window, without studying the whole interval.

6 Conclusions

In this paper, we have presented a method to detect user behaviour. As we have indicated, this problem is random imprecision by definition, because we do not know the interval of the time in which the action happen. Imprecision appears due of a behaviour is not static on a context or domain and is specific for every user. Thus, we have designed a method which uses fuzzy logic to limit events in a specific behaviour using a Fuzzy Temporal Window. With a Window, we can assign a membership degree for every event for the studied behaviour. In addition, we extend the Fuzzy Temporal Window concept, to adjust the interval using the knowledge from user we have. This adjustment is made with quantified sentences. With this adjustment we solve the possibility of a person changes his habits.

System's output is the fuzzy pattern sequences which define the behaviour. To manage non-random imprecision, we use the α-cuts and the Decomposition Theorem to obtain a fuzzy result applying the crisp method.

We have demonstrated that the fuzzy method is better than crisp method and that the fuzzy method obtains better outcomes than crisp one. As well as we conclude that results depend on the type of window we use. In addition, we have studied that outcomes are independent from the database.

In future work, we want to distinguish between three problems of our system:

- Non-random imprecision
- Uncertainty, because it could exists an error in the sensor reading. There are many reasons to provoke mistakes in sensors reading: the sensors or the reader could fail, the environment affects to the sensors, sensors damage, etc.
- Uncertainty in the Fuzzy Temporal Window, because the behaviour does no always happen exactly equal.

Our objective consists on extracting patterns that could control these kinds of problems, obtaining an adjusted pattern.

References

[1] M. Weiser. The computer for the twenty-first century. *Scientific American*, 1:94, 1994.

[2] Roy Want. Rfid: A key to automating everything. *Scientific American Magazine*, -:10, Enero 2004.

[3] Hiroyuki Yamahara et al. An individual behavioral pattern to provide ubiquitous service in intelligent space. volume 6, pages 562–569, 2007.

[4] T. Mori et al. Sensing room: Distributed sensor environment from measurement of human daily behaviour. *Intelligent Robots and Systems, 2005. (IROS 2005).IEEE/RSJ International Conference*, 2:1703 – 1709, 2-6 Aug. 2005.

[5] Kyohei Koyamaet al. Embedded action detector to enhance freedom from care. In *Proc. of 11th International Conference on Human-Computer Interaction*, page 8, Las Vegas, July 2005.

[6] Matthai Philipose et al. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50–57, 2004.

[7] María Ros et al. Un sistema para identificacin de comportamientos correctos en un mundo marcado. Primeras Jornadas Cientficas sobre RFID, 21-23 Noviembre 2007.

[8] María Ros et al. Correct behavior identification system in a tagged world. *Expert systems with applications*, 36:9899–9906, 2009.

[9] L A Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computing and Mathematics with Applications*, (9):149–184, 1983.

[10] A. Gonzalez et al. Dealing with uncertainty and imprecision by means of fuzzy numbers. *International Journal of Approximate Reasoning*, 21:233–256(24), August 1999.

[11] Hung T. Nguyen and Elbert A. Walker. *A first Course in Fuzzy Logic*. Chapman & Hall/CRC Taoylr & Frances Group, 2007.

ISBN: 978-989-95079-6-8