Gradual Trends in Fuzzy Sequential Patterns

Céline Fiot, Florent Masseglia
INRIA Sophia-Antipolis
2004 route des Lucioles - BP 93
06902 Sophia Antipolis - France
{celine.fiot,florent.masseglia}@sophia.inria.fr

Anne Laurent, Maguelonne Teisseire
LIRMM, Univ. of Montpellier II - CNRS
161 rue Ada
34392 Montpellier - France
{laurent,teisseire}@lirmm.fr

Abstract

Fuzzy sequential pattern mining is a relevant approach when dealing with temporally annotated numerical data since it allows discovering frequent sequences embedded in the records. However, such patterns, in their current form, do not allow extracting another kind of knowledge that is typical of sequential data: temporal tendencies. Thanks to a relevant use of fuzzy sequential patterns, we propose the GraSP algorithm that discovers gradual trends in sequences. Our proposal is validated through experiments on web access logs.

Keywords: Sequential Patterns, Data Mining, Sequence Database, Trends, Gradual Rules.

1 Introduction

As databases available from most of industrial or biological areas often contain timestamped or ordered records, sequence mining has become an important data mining task. One specific approach, sequential pattern mining, aims to discover frequent sequences within sequence databases, i.e. the mined datasets are sets of timestamped records, each of them constituted of a set of values. Consider the knowledge extracted from a commercial website. With sequential patterns, one could extract that *10% of users first request the page registration.php and later help.html.*

Most of the time, databases do not only contain binary attributes, but also numeric attributes, such as prices, quantities,... Therefore generalizations of *symbolic* sequential patterns have been proposed to handle such information [1, 2, 3]. These *fuzzy sequential patterns* contain additional pieces of knowledge compared to crisp ones. For instance, the previous pattern could be more explicit: *10% of users first request the page registration.php many times and later few request help.html.*

In this paper we propose to incorporate graduality within fuzzy sequential patterns and express temporal tendency, in order to mine rules like *"the more the increase of x in A, the more the increase of y in B"*, defining *gradual sequential patterns*. Gradual dependencies are rules that represent a relation among the variations in the degree of fulfillment of imprecise properties by different objects [6, 7]. An example of gradual dependence is *the higher the number of simultaneous connections to a website, the slower response*, meaning that observing web logs, when the number of simultaneous connections increases, the time response of this website also increases.

Within a context of web mining such patterns could be for instance that *the more the number of requests to registration.php fulfills the property "high number of requests" at time t the more the number of requests to help.html page fulfills the property "high number of requests" a few seconds later.* This knowledge would be explicit for the end-user (is the registration form easy to fill-in?). However the problem of gradual sequential pattern mining is not a mere extension of gradual dependencies, since the temporal aspects introduce a large number of combinations. Our goal is to

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 456–463
Torremolinos (Málaga), June 22–27, 2008

show that handling time and extract gradual sequential patterns can be done efficiently and leads to relevant knowledge.

In this paper, we describe our proposal which relies on the following points. First we propose the definition of gradual sequential patterns. Secondly we introduce our method for extracting gradual sequential patterns by means of fuzzy sequential patterns. Last, we focus on the algorithmic tricks that we use in GRASP to efficiently mine sequence databases for gradual sequential patterns.

In the next section, we define the fundamental concepts associated with gradual rules and fuzzy sequential patterns. In section 3 we present our proposal, first defining what gradual sequential patterns are, then detailing how to implement their discovery. We propose some experiments on web access logs in section 4, showing the benefits of such pattern discovery. We conclude in section 5 on the prospects opened by our work.

2 About Sequential Patterns and Graduality

Sequential patterns are often introduced as an extension of association rules in [8]. Initially proposed in [9], they highlight correlations between database records as well as their temporal relationships. Some generalization were proposed to use fuzzy set theory to handle numeric attributes. In this paper we use fuzzy sequential patterns to mine gradual sequences defined from the concepts of gradual rules.

2.1 Sequential Patterns

Sequential patterns are based on the idea of *maximal frequent sequences*.
Let $\mathcal{R}$ be a set of object records where each record R consists of three information elements: an object-id, a record timestamp and a set of attributes/items in the record. Let $I = \{i_1, i_2, ..., i_m\}$ be a set of items or attributes. An *itemset* is a non-empty set of attributes i_k, denoted by $(i_1 i_2 \ldots i_k)$. It is a non-ordered representation. A *sequence* s is a non-empty ordered list of itemsets s_p, denoted by $\langle s_1 s_2 ... s_p \rangle$. A *n-sequence* is a sequence of n items (or of size n).

Example 1 *Consider a website. Objects are users, and records are the URLs requested made by each user during one session. Timestamps are the sessions. If an user requests URLs encoded by e, a, k, u, and f according to the sequence $s = \langle (e)\ (a\ k)\ (u)\ (f) \rangle$, then all items of the sequence were requested separately, except URLs a and k which were accessed at the same session. In this example, s is a 5-sequence.*

A sequence $S = \langle s_1\ s_2 ... s_p \rangle$ is a *subsequence* of another one $S' = \langle\ s'_1\ s'_2\ ... s'_m \rangle$ if there are integers $l_1 < l_2 < ... < l_p$ such that $s_1 \subseteq s'_{l_1},\ s_2 \subseteq s'_{l_2}, ...,\ s_p \subseteq s'_{l_p}$.

Example 2 *The sequence $s' = \langle (a)\ (f) \rangle$ is a subsequence of s because $(a) \subseteq (a\ k)$ and $(f) \subseteq (f)$; $\langle (a)\ (k) \rangle$ is not a subsequence of s.*

All records from the same object o are grouped together and sorted in increasing order of their timestamp, constituting a *data sequence*. An object *supports* a sequence s if it is included within the data sequence of this object (s is a subsequence of the data sequence). The *frequency* of a sequence ($freq(s)$) is defined as the percentage of objects supporting s in the whole set of objects $\mathcal{O}$. A minimum frequency value ($minFreq$) is specified by the user and the sequence is said to be frequent if the condition $freq(s) \geq minFreq$ holds. A sequence that may be frequent is a *candidate sequence*. Given a database of object records, the problem of sequential pattern mining is to find all maximal sequences of which the frequency is greater than a specified threshold ($minFreq$) [9]. Each of these sequences represents a *sequential pattern*, also called a maximal frequent sequence.

Several extensions were proposed to handle numerical and quantitative values [1, 2, 3], to generalize sequential patterns with respect to various temporal parameters (time-interval between events of a sequence, grouping several records into a single itemset...) [9, 10], or even to deal with missing values [11].

2.2 Gradual Rules and Dependencies

Gradual correlations in the fulfillment of imprecise properties may be represented using several formalism such as gradual dependencies and gradual rules.

Gradual dependenciess describe the implication that may exist between two fuzzy properties. Several proposals have been formulated in order to discover such rules within relational databases. [4] and [5] propose to use the various fuzzy implications to extract different kinds of gradual dependencies. [5] describes some tracks for designing a mining algorithm.

Gradual rules describe a correlation among the variation in the degree of fulfillment of imprecise properties by different objects [6, 7]. They express a tendency. For instance, a gradual dependency could be *the higher the number of connections, the lower the download rate*, meaning that as the number of connections increases, download rate tends to decrease. Formally a gradual rule can be expressed by a sentence *the more (or less) X is A, the more (or less) Y is B.*
[6] gives the very first complete formalism of gradual rules, expressing the relations and correlations between attributes using statistical formulæ and tools, such as contingency diagrams and linear regressions. [7] describes a framework that discovers gradual rules on a preprocessed database using a fuzzy association rule mining algorithm. In this paper, we adopt a similar principle and propose to mine for *gradual sequential patterns* using fuzzy sequential patterns.

2.3 Fuzzy Sequential Patterns

In order to allow for handling numerical or quantitative information several works proposed to partition each numerical attribute into several fuzzy sets. The quantitative database is thus converted into a membership degree database, which is then mined for fuzzy sequential patterns [1, 2, 3].

The item and itemset concepts have been redefined relative to classical sequential patterns. A **fuzzy item** is the association of one item and one corresponding fuzzy set. It is denoted by $[x, a]$ where x is the item (also called attribute) and a is the associated fuzzy set.

Example 3 $[URL\_A, lot]$ *is a fuzzy item where lot is a fuzzy set defined by a membership function on the quantity universe of the possible number of requests of the* URL_A *during one session.*

A **fuzzy itemset** is a set of fuzzy items. It can be denoted as a pair of sets (set of items, set of fuzzy sets associated to each item) or as a list of fuzzy items. We use the following notation: (X, A), where X is a set of items and A is a set of corresponding fuzzy sets. One fuzzy itemset only contains one fuzzy item related to one single attribute.

Example 4 $([URL\_A, lot][URL\_B, few])$ *is a fuzzy itemset and can also be denoted by* $((URL\_A, URL\_B)(lot, few))$.

Last a **g-k-sequence** $S = \langle s_1 \cdots s_g \rangle$ is a sequence constituted by g fuzzy itemsets $s = (X, A)$ grouping together k fuzzy items $[x, a]$.

Example 5 *The sequence* $\langle([URL\_B, lot]$ $[URL\_A, lot])$ $([URL\_C, few])\rangle$ *groups together 3 fuzzy items into 2 itemsets. It is a fuzzy* 2-3-*sequence.*

In the next sections of this article, we use the following notations: let $\mathcal{O}$ represent the set of objects and $\mathcal{R}_o$ the set of records for one object o. Let $\mathcal{I}$ be the set of attributes or items and $\varrho[i]$ the value of attribute i in record ϱ. Each attribute i is divided into fuzzy sets. Then one record in a fuzzy sequence database consists of the membership degrees of each attribute to each fuzzy set, e.g $r(x, a) = \mu_a(x)$ represents the membership degree of item/attribute x to the fuzzy set a in record r.

The frequency of a fuzzy sequence S is then computed by the formula:

$$FFreq(S) = \frac{\sum_{o \in \mathcal{O}} \varphi(S, o)}{|\mathcal{O}|}$$

where $\varphi(S, o)$ gives the degree to which S is included into the object o data sequence.

This degree is computed by considering the best appearance – i.e. the appearance with the highest degree – of the ordered list of itemsets of S. It is computed by:

$$\varphi(S,o) = \underline{\perp}_{\varsigma \subseteq \zeta_o | S=\varsigma=\langle s_1 \ldots s_i \ldots s_k \rangle} \overline{\top}_{s_1 \ldots s_k} (\overline{\top}_{j \in s_i} \mu(j))$$

where k is the number of itemsets in S, ζ_o is the set of sequences included in the data sequence of object o and $\overline{\top}$ and $\underline{\perp}$ are the t-norm and t-conorm operators generalized to n-ary cases. Practically we use the Zadeh t-norm and t-conorm, min and max.

3 Gradual Trends in Sequential Patterns

The objective of this work is to discover temporal relations among the variation in the degree of fulfillment of imprecise properties by different objects. For instance a gradual sequential pattern could be that *considering mail server breakdowns, the more the number of received e-mails is "high" and the more the average size of received e-mails is also "high" at time t, the higher the number of time delivery errors becomes later.*

Moreover we propose to use the fuzzy data sequence formalism to discover an additional information, expressing in the previous example what is typically *"later"* for this pattern, *e.g. few minutes, half an hour, ...*

In order to extract gradual sequential patterns we thus propose to process the original fuzzy sequence database into a gradual sequence database that will then be mined using our algorithm GraSP (subsection 3.4).

3.1 Overall Principle

The global process can be described by figure 1. In addition to the definition of gradual sequential patterns, our contribution corresponds to steps 2 and 3.

First the database is converted into a *membership degree database* (μdb on Figure 1, e.g. Table 1), such as for fuzzy sequential pattern mining, using predefined fuzzy sets – automatically or from expert knowledge designed.

Then this membership degree database is converted into a *time-related variation degree database* (step 2, leading to νdb in Figure 1). This step of the process is detailed in subsection 3.3. This dataset is the one mined for gradual sequential patterns (step 3 in Figure 1) as it is described in subsection 3.4.

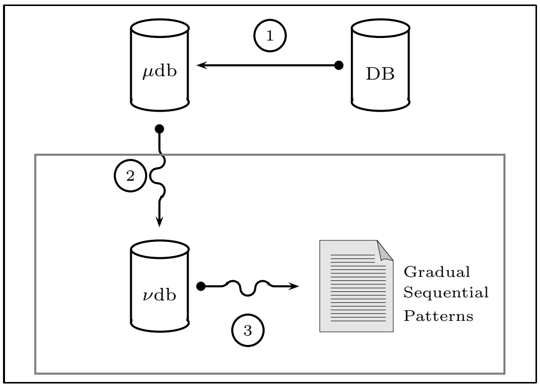

Figure 1: Overall principle of our approach

Example 6 *Let* $\langle([x,4])([x,3][y,5][z,8])([x,2][y,4][z,10])([y,6][z,7])\rangle$ *be a sequence characterizing the number of connections to URL x, y and z during successive sessions of one IP o. Table 1 describes a membership degree database (μdb) drawned from this sequence. It contains four ordered fuzzy records for the IP o. These records may for instance contain the fuzzification of the number of accesses to three URL X, Y and Z during successive session of an identified IP. This dataset μdb then gives the membership degrees for three fuzzy items, consisting of attribute X (resp. Y, Z) associated with quantitative property A (resp. B, C), for instance "very few", "few" and "lot".*

Table 1: A membership degree data sequence.

	date	**x,a**	**y,b**	**z,c**	
d1	1	0.1			r^1
d2	3	0.2	0.3	0.4	r^2
d3	4	0.4	0.5	0.7	r^3
d4	5		0.2	0.3	r^4

Then Table 3 gives the time-related variation degrees associated to gradual items obtained from Table 1, after step 2 of the global process (described in subsection 3.3).

3.2 Definitions

A *gradual item* should denote a variation – increase or decrease – of the membership degree of an attribute to a property (i.e. one of its associated fuzzy sets). Then a gradual item is defined as a fuzzy item $[x, a]$, associated to an operator $\blacklozenge$, with $\blacklozenge \in \{<, >\}$. For instance, the *gradual item* $[x, a, >]$ means the more x is a. While building the gradual sequence database, each gradual item will be associated with a membership degree. This degree describes the strength of the variation expressed by the $\blacklozenge$ operator [7].

A *gradual itemset* can then be defined as a non-ordered, non empty set of gradual items and a *gradual sequence* is an ordered list of gradual itemsets. A gradual itemset will be denoted by parenthesis (e.g. $([x, a, \blacklozenge_1]\ [y, b, \blacklozenge_2])$) and a sequence by angles (e.g. $\langle([x, a, \blacklozenge_1]\ [y, b, \blacklozenge_2])\ ([z, c, \blacklozenge_3])\rangle$).

Thus a gradual sequence database will be a specific fuzzy sequence database in which the fuzzy items represent variation strength in fulfillment degree of attribute properties. So mining gradual sequential patterns should be as "simple" as mining fuzzy sequential patterns. Unfortunately, due to the temporal aspect of the data and the combinatory problems it causes, this naive solution is not feasible and we need an efficient approach.

In the forthcoming subsection, we show how the variation degree database is built and in subsection 3.4 we detail our algorithm for gradual sequential pattern mining.

3.3 Building Gradual Records

Each item $[x, a, \blacklozenge]$ in the variation degree database νdb represents the evolution of $\mu_a(x)$ between two successive records r^1 and r^2, related to a same object o in the membership database. Then each record in νdb will be made by the combination of two records of the initial μdb dataset. More specifically, for each ordered pair of records r^i and r^j of one data sequence – r^j is later than r^i, but not necessarily consecutive – such that $r^i(\mu_a(x^i))$ and $r^j(\mu_a(x^j))$ are strictly greater than 0, we create a *gradual record* $gr_{r^i r^j}$ containing the gradual item $[x, a, \blacklozenge]$ iff $\mu_a(x^i) \blacklozenge \mu_a(x^j)$.

Example 7 *From records r^1 and r^2 in Tab. 1, a gradual record is created, containing item $[x, a, >]$, since $\mu_a(x^2) = 0.2 > \mu_a(x^1) = 0.1$.*

In order to keep the temporal relationships originally existing within the sequence database, for each object o, gradual records are chronollogically ordered by the timestamps of r and r'. Moreover, to keep the variation strength we use the definition of variation degree $gr_{rr'}([x, a, \blacklozenge])$ given by [7] :

$$gr_{rr'}([x, a, \blacklozenge]) = \mu_\blacklozenge(\mu_a(x), \mu_a(x'))$$

$$\text{where } \mu_\blacklozenge(u, v) = \begin{cases} |u - v| & \text{if } u \blacklozenge v \\ 0 & \text{else} \end{cases}$$

Example 8 *From Table 1 records r^1 and r^2, the gradual item in Example 7 is associated to the variation degree $gr_{r^1 r^2}([x, a, >]) = 0.1$. Table 2 is the variation degree database obtained from Table 1.*

Table 2: Variation degree data sequence obtained from Table 1.

		x,a,>	**y,b,>**	**y,b,<**	**z,c,>**	**z,c,<**	
d1	d2	0.1					r^1
d1	d3	0.3					r^2
d2	d3	0.2	0.2		0.3		r^3
d2	d4			0.1	0.1		r^4
d3	d4			0.3		0.4	r^5

Having this set of records for each data sequence o in $\mathcal{O}$, we could then go to the mining step. However one information, still available in a sequence database, would be lost. We are indeed in a context of temporally annotated records. So comparing two records to generate a gradual item with a variation degree could lead to an additional entry in the gradual dataset: timestamps of both records may also be compared thus creating an additional fuzzy item in $gr_{rr'}$ describing duration between r and r'. This duration item is expressed by a linguistic variable computed from the difference between timestamps of r and r'.

Example 9 *Considering the fuzzy sets on Figure 2 for duration, the final time-related variation degree database obtained from Table 1 is described by Table 3.*

Table 3: Time-related variation degree data sequence obtained from Table 1.

		x,a	y,b,>		z,c,>		duration	
		>	>	<	>	<	sh.	lg
d1	d2	0.1					0.75	0.25
d1	d3	0.3					0.25	0.75
d2	d3	0.2	0.2		0.3		1	
d2	d4			0.1	0.1		0.75	0.25
d3	d4			0.3		0.4	1	

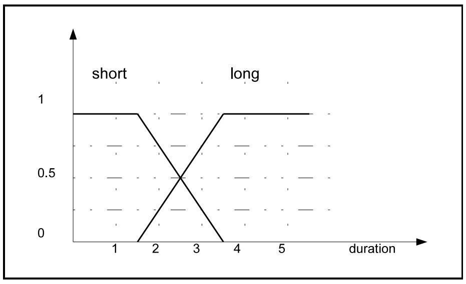

Figure 2: Duration linguistic variables

Finally, the time-related variation degree database consists of a set $\mathcal{GR}$ of gradual records $gr_{r^1r^2}$, each consisting of:

- one object-id o, corresponding to the object-id of r^1 and r^2,
- r^1 timestamp, denoted by $t(r^1)$,
- r^2 timestamp, denoted by $t(r^2)$,
- a set of gradual items $[x, a, \blacklozenge]$, such that
$$\forall [x, a]/r^1(\mu_a(x^1)) \neq 0, r^2(\mu_a(x^2)) \neq 0,$$
$$gr_{r^1r^2}([x, a, \blacklozenge]) = \mu_\blacklozenge(\mu_a(x^1), \mu_a(x^2))$$
$$\text{with} \blacklozenge \in \{<, >\}$$
- fuzzy duration items
$$gr_{r^1r^2}(\delta) = \mu_d(t(r^2) - t(r^1))$$

3.4 Algorithmic Tricks

Our aim is to extract gradual sequential patterns from the time-related variation degree database obtained after step 2.

Since each gradual record $gr_{r^1r^2}$ has been built from two records of the initial dataset, it contains two timestamps $t(r^1)$ and $t(r^2)$ that describe a duration. Since we want to discover sequences we need to define an order and/or constraints to be satisfied between gradual records, taking into account the original timestamps $t(r^1)$ and $t(r^2)$.

Actually several gradual records can have the same starting timestamp $t(r^1)$ and they cannot belong to the same sequence. But they correspond to the same object in the dataset. In the same way, one record covering the time-period from $t(r^1)$ to $t(r^3)$ does not preceed or follow a record including r^2 that happened between r^1 and r^3 in the original μdb.

Example 10 *Figure 3 represents each gradual record of Table 3.*

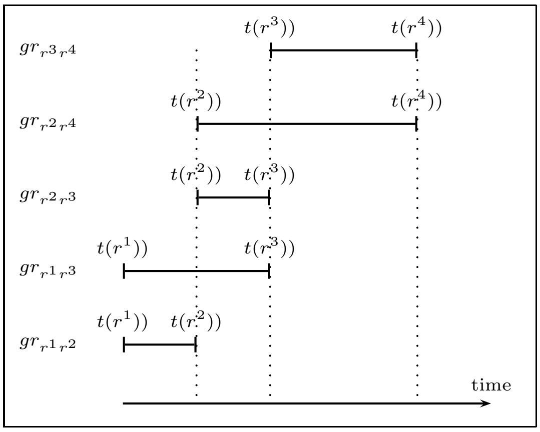

Figure 3: Overlapping gradual records

The second gradual record overlaps the first and third ones and the fourth record overlaps the third and fifth ones.

To take the possible overlaps of itemsets into account, the sequence database νdb should be parsed with lots of forward and backward phases during examination of data sequences. To avoid such expensive parsing of the data, we propose an algorithm that skips overlapping itemsets for one candidate sequence. Our approach uses a graph structure to represent allowed sequences in a data sequence, such as the sequence graphs developped in [10] to handle time constraints. Vertices in the sequence graph are gradual records and edges represent sequences. Thus each gradual data sequence of the final dataset νdb is then preprocessed into a sequence graph in which the gradual sequential patterns are mined for.

The global mining algorithm, GraSP, Fig. 4, can be described as follows. It is a generate-and-prune approach, that uses the frequent sequences of size k to generate candidate sequences of size k+1. The frequency of these (k + 1)-sequences is calculated using the TotallyFuzzy algorithm from [3]. Then, the al-

---

GRASP - **Input:** $minFreq$, νDB
Ouput: F, frequent gradual sequences

$F_0 \leftarrow \varnothing$; $k \leftarrow 1$;
$F_1 \leftarrow \{\{< i >\}/i \in \mathcal{I} \& freq(i) > minFreq\}$;
For each gradual data sequence $gS \in \nu DB$ **do**
  $graphDB \leftarrow$ createGraph(gS) ;
End For
While ($Candidate(k) \neq \varnothing$) **do**
  For each sequence graph $g \in graphDB$ **do**
    [countFreq is a version of the TotallyFuzzy]
    [algorithm adapted to sequence graph parsing]
    countFreq($Candidate(k), minFreq, g$) ;
  End For
  $F_k \leftarrow \{s \in Candidate(k)/freq(s) > minFreq\}$;
  $Candidate(k+1) \leftarrow$ generate(F_k) ;
  k++;
End While
return $F \leftarrow \bigcup_{j=0}^{k} F_k$

Figure 4: GRASP

gorithm searches for the candidate sequences within the sequence graphs.

Example 11 *Figure 5 represents the sequence graph obtained from the time-related variation degree sequence given by Table 3.*

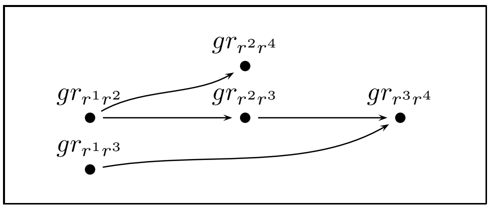

Figure 5: Sequence graph for the data sequence in Table 3.

From the data sequences in Table 3 we can build four longest sequences to mine gradual sequential patterns: $\langle(gr_{r^1r^2})(gr_{r^2r^3})(gr_{r^3r^4})\rangle$, $\langle(gr_{r^1r^3})(gr_{r^3r^4})\rangle$ *and* $\langle(gr_{r^1r^2})(gr_{r^2r^4})\rangle$.

The soundness and completeness of such approaches, based on sequence graphs, have been proved. So at the end of the process we obtain all the gradual sequential patterns contained in the time-related variation degree dataset.

4 Experiments

The aim of these experiments is to apply gradual sequential patterns for web usage analysis.

4.1 Data

In our case, records contain the number of access to one page, the same half-day by one user. For example, record *"1500 5067 10 6"* means that *"visitor 1500"* on half-day *5067* visited *6* times the URL coded by *10.* This dataset contains 27209 web pages visited by 79756 different IPs during 16 days (32 half-days). The translation into the formalism given in sections 2 and 3 is given by Tab. 4.

Table 4: Data sequences for web usage mining

Object	↔	IP
Timestamp	↔	halfday
Fuzzy items	↔	properties of # of accesses to each web page
Gradual items	↔	variation of degree of the properties of accesses to each web page
Duration	↔	time period between two accesses to one web page

As detailed in section 3.1, quantities are converted into three fuzzy sets membership degrees, using the principle given in [3].Then the membership degree database is converted into the time-related variation degree database, mined for gradual sequential patterns.

4.2 Results

The runtime performances of our algorithm are similar to fuzzy sequential pattern algorithms, Figure 6.

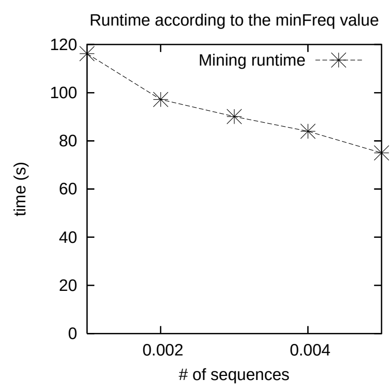

Figure 6: Runtime according to $minFreq$.

Regarding the qualitative analysis, discovered patterns are relevant. One typical pattern we discovered on the log file of INRIA Sophia is related to the Koala project and expresses the following temporal tendency *"The lower the number of connections to page KBM becomes, the more the connections to KOML fulfills the "high number" property after a short period and during a long period, then followed by an increasing high number of connections to DJAVA"*.

5 Conclusion and Future Work

In this paper we have developed an approach for mining gradual trends in sequential patterns based on fuzzy sequential pattern mining. The purpose of this technique is to discover time-related tendencies, showing how the variation of one (or several properties) is linked to other that previously happened. This work extends some concepts that were initially proposed for gradual rule discovery. Within the context of time-related databases, we need to define some specific process for handling temporal information. Time indeed leads to an explosion of runtime and space complexity, thus making existing algorithms for gradual rules mining inappropriate. Therefore we implemented an efficient algorithm that we tested on web access logs. We thus discovered relevant knowledge such as *the more the number of requests to registration.php fulfills the property "high number of requests" at time t the more the number of requests to help.html page fulfills the property "high number of requests" a few seconds later.* This temporal relation among web page browsing could then be used to improve website architecture and quality of services. Now we are working on defining time-related gradual rules based on these sequential patterns. The objective of this work is to find the causal relationships that may exist between several sequential events.

References

[1] T. Hong, K. Lin, and S. Wang, "Mining Fuzzy Sequential Patterns from Multiple-Items Transactions," in *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, 2001.

[2] R.-S. Chen, G.-H. Tzeng, C.-C. Chen, and Y.-C. Hu, "Discovery of Fuzzy Sequential Patterns for Fuzzy Partitions in Quantitative Attributes," in *ACS/IEEE Int. Conf. on Computer Systems and Applications*, 2001.

[3] C. Fiot, A. Laurent, and M. Teisseire, "From crispness to fuzziness: Three algorithms for soft sequential pattern mining," *IEEE Transactions on Fuzzy Systems*, to appear.

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

[5] P. Bosc and O. Pivert, "On some fuzzy extensions of association rules," in *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, 2001.

[6] E. Hüllermeier, "Association rules for expressing gradual dependencies," in *6th European Conference on Principles of Data Mining and Knowledge Discovery*, 2002.

[7] C. Molina, J. M. Serrano, D. Sanchez, and M. A. Vila, "Measuring variation strength in gradual dependencies," in *5th European Society for Fuzzy Logic and Technology Conference*, 2007.

[8] R. Agrawal, T. Imielinski, and A. N. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD International Conference on Management of Data*, 1993.

[9] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *5th International Conference on Extending Database Technology*, 1996.

[10] F. Masseglia, P. Poncelet, and M. Teisseire, "Pre-Processing Time Constraints for Efficiently Mining Generalized Sequential Patterns," in *11th International Symposium on Temporal Representation and Reasoning*, 2004.

[11] C. Fiot, A. Laurent, and M. Teisseire, "Approximate sequential patterns for incomplete sequence database mining," in *16th IEEE International Conference on Fuzzy Systems (FuzzIEEE'07)*, 2007.