Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Data Mining (CIDM 2007)

Mining Association Rules in Temporal Sequences

Khellaf BOUANDAS and Aomar OSMANI
LIPN UMR CNRS 7030
Université de Paris 13
99, avenue J.-B. Clément 93430 Villetaneuse, France
{Khellaf.Bouandas, Aomar.Osmani}@lipn.univ-paris13.fr

***Abstract*—Mining association rules is an important technique for discovering meaningful patterns in datasets. Temporal association rule mining can be decomposed into two phases: finding temporal frequent patterns and finding temporal rules construction. Till date, a large number of algorithms have been proposed in the area of mining association rules. However, most of these algorithms consider patterns as a collection of point primitives and their three basic relations $(<, =, >)$.**
Several applications consider patterns with duration and need to reason about intervals and their thirteen possible relationships. In this paper we investigate properties of temporal sequences represented as a collection of intervals. We present a simple framework for temporal sequence and describe DATTES (Discovering pATterns in TEmporal Sequences), an innovative algorithm using interval properties to mine temporal patterns. The framework can be used to mine temporal association rules. According to some interval algebra properties, this paper introduces a new confidence evaluation function for mining temporal rules. Experiments on real dataset (Human face identification problem) show the effectiveness and the performances of this approach.

I. INTRODUCTION

Association rule mining (ARM) was introduced in [1]. The problem of mining temporal association rules (TAR) is considered as the generation of the ARM when the duration of pattern is considered. In this paper, we assume that the reader knows the basic assumptions and terminologies of ARM. However, it is noteworthy at this point that the work of TAR can be decomposed into two phases, i.e. (i) frequent temporal pattern generation: Find frequent temporal pattern that exceed the given minimum support,(ii) temporal rules construction: from the frequent temporal patterns generated in step 1, generate all temporal association rules having confidence higher than the given minimum confidence.
The second phase is straightforward and less-expensive, researchers have generally focused on the first phase only. We can think of the problem of mining Frequent Temporal Patterns (FTP) as a generalization of Frequent Pattern (FP) mining to temporal datasets. There exists a vast body of work in which many innovative algorithms were presented for solving the same problem, also under different user-provided constraints [2], [6], [12], [13]. A number of studies, such as [2], regard a sequence as an order among primitives and do not consider the duration of each primitive. These studies aim to find patterns (itemsets) occurring with a given minimum support within a dataset D, which corresponds to collections of point primitives. A pattern is frequent if its support, the occurrence number of pattern in dataset, is greater than (or equal to) a given threshold. In [6], while the duration constraint of pattern is considered, the author proposes a notion of temporal pattern very similar to ours. Thus, an input dataset D for the FTP problem is instead composed of a collection of interval primitives. The attributes related to the temporal information present in this type of datasets need to be treated differently from other kinds of attributes. A temporal pattern is frequent if its temporal support is greater than (or equal to) a given threshold.
This paper is organized as follows: Section 2 develops the problem statement and the suitable notions needed to model the problem of temporal sequence mining. Section 3 addresses the FTP problem. This section is started by a new evaluation function for the temporal support, DATTES-Gen, the generator function, and the algorithm details. Section 4 describes the temporal association rules mining. Section 5 shows some results of the proposed algorithm and comparative results. The section 6 concludes the paper.

II. TEMPORAL SEQUENCES

The temporal sequence can be considered as subsections of the temporal data, an observed data sequence which is ordered in time. We confine our interest to temporal association rules mining from temporal sequences and not on how to obtain these temporal sequences which is called the discretization techniques. Thus, we assume that this transformation has been done along with other steps of preprocessing (see [7], [8], [9]). Than, Figure 1 illustrates a simple intuitive transformation of f and h, two observed temporal data: we are interested, for example, when f (or h) "goes up", labeled f^+ (or h^+) and when f (or h) "goes down", labeled f^- (or h^-).
As mentioned below, the input dataset for the FTP problem consider the duration of each primitive (item). Let us consider n primitives $D = \{s_i | i = 1 \dots n\}$. For each $s_i \in D$, $[b_i, e_i]$ is an interval, with begin and end points, in which the primitive holds. An interval primitive can be represented as a triplet (b, s, e) such that s and $[b, e]$ are the primitive identifier and the occurrence interval of the interval primitive respectively. Assume for every e_i and e_j in D, $[b_i, e_i] \cap [b_j, e_j] = \emptyset$ if s_i and s_j did not refer the same primitive. In other wise,i.e. s_i and s_j refer the same primitive and $[b_i, e_i] \cap [b_j, e_j] \neq \emptyset$, we replace these two primitives by this one $(min(b_i, b_j), s_i, max(e_i, e_j))$. This property is called the maximality assumption of interval primitives. A temporal sequence S of n interval primitives is

defined as $(b_1, s_1, e_1), \ldots, (b_i, s_i, e_i), \ldots, (b_n, s_n, e_n)$, where $b_i < e_i$ and $b_i \leq b_{i+1}$. Therefore, we simplify the temporal sequence as $S = (s_1, \ldots, s_i, \ldots, s_n)$.

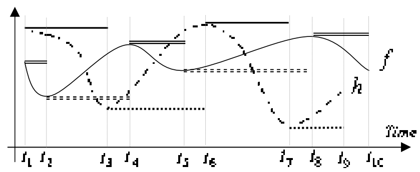

Fig. 1. A visual intuition of distinguish trend of f and h, two temporal data.

Since all primitives may potentially be defined, we need to sweep these intervals sequence; this can be achieving by the using of the sliding window [10]: the length of the sliding window, denoted w, is must be specified in advance.

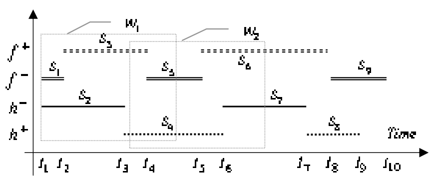

Fig. 2. The temporal sequence and two sliding windows: Note that f and h are converted respectively into a five-symbol ("$f^- f^+ f^- f^+ f^-$") and a four-symbol string ("$h^- h^+ h^- h^+$").

Example 1: Figure 2 shows nine states $(s_1, .., s_9)$, obtained from the discretization of f and h, and the sliding window. The f's behavior: "f goes up", denoted f^+, is expressed by two interval primitives (t_2, s_3, t_4) and (t_5, s_6, t_8), "f goes down", denoted f^-, is expressed by three interval primitives: (t_1, s_1, t_2), (t_4, s_5, t_5) and (t_8, s_9, t_{10}).

Definition 1 (Temporal Pattern): A temporal pattern of dimension k, denoted by $k - TP$, is defined as a pair (S, R) where S represents $(s_i, s_{i+1}, ..., s_{i+k-1})$, a sequence of interval primitives, and $R \in I^{k \times k}$, a matrix of qualitative relationships between these k primitives, is given as follows:

$$R = \begin{pmatrix} eq & r_i^{i+1} & \ldots & r_i^{i+k-2} & r_i^{i+k-1} \\ r_{i+1}^{i} & eq & \ldots & r_{i+1}^{i+k-2} & r_{i+1}^{i+k-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{i+k-1}^{i} & r_{i+k-1}^{i+1} & \ldots & r_{i+k-1}^{i+k-2} & eq \end{pmatrix}$$

where r_i^j denote the temporal relation between s_i and s_j. This relation is expressed by Allen's interval logic [4]. Possible basic relations between two intervals are $\{before, meets, overlaps, contains, starts, finishes, equals\}$, and their opposite relations. The abbreviate set of these relations, labeled I, is defined as follows $\{b, m, o, c, s, f, eq, b^{-1}, m^{-1}, o^{-1}, c^{-1}, s^{-1}, f^{-1}\}$.

Remark 1: If we consider the opposite relation r_j^i, we can optimize the matrix representation by the linear representation of R as follows:

$$R = ((r_i^{i+1}), (r_{i+1}^{i+2}, r_i^{i+2}), \ldots, (r_{i+j-1}^{i+j}, r_{i+j-2}^{i+j}, \ldots, r_i^{i+j}), \ldots, (r_{i+k-2}^{i+k-1}, r_{i+k-3}^{i+k-1}, \ldots, r_i^{i+k-1})).$$

Example 2: If we consider the temporal sequence in Figure 2, we can generate the following $k - TP$:

$1 - TP : (s_i)$ for $i \in \{1..9\}$
$2 - TP : (s_2, s_3; (o))$ and $(s_4, s_5; (c))$
$3 - TP : (s_3, s_4, s_5; (o)(c, m))$ and $(s_4, s_5, s_6; (c)(m, o))$
$4 - TP : (s_1, s_2, s_3, s_4; (s)(m, b)(o, m, b))$ and $(s_3, s_4, s_5, s_6; (o)(c, m)(m, o, b))$

Definition 2 (Temporal Instance): Let us consider two $k - TP$s $Q = (S_Q, R_Q)$ and $P = (S_P, R_P)$ such that: $R_P = R_Q$, all P ends points intervals have fixed real values and $P \neq Q$ then P is said instance sequence of Q and Q is said model of interval sequences.

The set of all instances of the pattern Q is denoted I_Q. We denote by $I_P^d \subseteq I_P$ the set of instances of Q respecting the maximality assumption of interval primitives.

Example 3: In Figure 2, if we consider P = "$h^- overlaps\ f^+$" and Q = "$f^- before\ h^+$": $I_P = \{(s_2, s_3)\}$, $I_Q = \{(s_1, s_4), (s_1, s_8), (s_5, s_8)\}$ and $I_Q^d = \{(s_1, s_4), (s_5, s_8)\} \subseteq I_Q$

Definition 3 (Prefix and Suffix Pattern): Given P and Q respectively the n-TP and the k-TP. Q is the:

1) k-prefix of P iff $s_Q = \{(s_P)_i | (s_P)_i \in s_P \wedge 1 \leq i \leq k\}$
2) k-suffix of P iff $s_Q = \{(s_P)_i | (s_P)_i \in s_P \wedge n - k < i \leq n\}$

Example 4: If we consider Figure 2, the 2-TP represented by (s_1, s_2) is the *2-prefix* of the 5-TP represented by $(s_1, s_2, s_3, s_4, s_5)$ and the 3-TP represented by (s_3, s_4, s_5) is the *3-suffix* of the 4-TP represented by (s_1, s_3, s_4, s_5).

Definition 4 (Precedence Relation): Let P and Q be two $k - TP$s. We say P precedes Q, noted $P \preceq Q$,iff $\exists i = max\{1, min\{j | (s_P)_j \neq (s_Q)_j\}\} : (s_P)_i \leq (s_Q)_i$.

Theorem 1: The relation $\preceq$ defines a partial order on $TP_k(S)$, the space of all temporal patterns of dimension k.

Proof: Let us consider $M, P, Q \in TP_k(S)$. The precedence relation is:

1) *reflexive: $P \preceq P$ for i=1.*
2) *transitive: $M \preceq P \wedge P \preceq Q \Rightarrow M \preceq Q$.*
   $M \preceq P \Leftrightarrow \exists i = max\{1, min\{j/(s_M)_j \neq (s_P)_j\}\} : (s_M)_i \leq (s_P)_i$
   $P \preceq Q \Leftrightarrow \exists r = max\{1, min\{j/(s_P)_j \neq (s_Q)_j\}\} : (s_P)_r \leq (s_Q)_r$
   For $l = min(i, r) : (s_M)_l \leq (s_P)_l \leq (s_Q)_l$, i.e. $M \preceq Q$.
3) *anti-symmetric: $P \preceq Q \wedge Q \preceq P \Rightarrow P = Q$.*
   $P \preceq Q \Leftrightarrow \exists i = max\{1, min\{j/(s_P)_j \neq (s_Q)_j\}\} : (s_P)_i \leq (s_Q)_i$,
   $Q \preceq P \Leftrightarrow \exists r = max\{1, min\{j/(s_Q)_j \neq (s_P)_j\}\} : (s_Q)_r \leq (s_P)_r$,
   Therefore $i = r$ and $(s_P)_i \leq (s_Q)_i \wedge (s_Q)_r \leq (s_P)_r$
   Then $\forall i \leq dim(P) : (s_P)_i = (s_Q)_i$, i.e. $P = Q$.

Definition 5 (Adjacent Pattern): Let P and Q be two $(k+1)$-TP. We say P is adjacent to Q, denoted by $P \| Q$, iff $P \preceq Q$ and the k-suffix of P is the same as the k-prefix of Q.

III. Frequent Temporal Pattern

The problem of recognizing interesting temporal patterns with temporal constraints is discussed in [6]. The major differences between our contribution and the Höppner's work is the construction of the pattern space: In contrast to [6], where author proposes a simple adaptation of Apriori algorithm and where all possible combinations are explored, we propose a new exploring relation, labeled adjacent relation. This main component of DATTES exploits the topological neighborhood property of interval relationships in order to discover the compact set of frequent temporal patterns in polynomial time.

A. Temporal Support

As one can see the previous works, the definition of support is based on the occurrence number within the sliding window of this pattern. Unlike others, our intuitive definition of the temporal support of a temporal pattern is based on the union of the disjoined instances of the temporal pattern, i.e.:

Definition 6 (Temporal Support): The temporal support of a temporal pattern P, denoted $T_{supp}(P)$, is defined as the optimum time in which its disjoined temporal instances can be observed within the sliding window, i.e.:

$T_{supp}(P) = \sum card(\cup_{Q \in I_P} O_Q)$

where w is the width of the sliding window, and O_Q, the time in which the temporal instance of Q can be observed within the sliding window, labeled the observability interval of Q [6], is given as:

$$O_Q = \begin{cases} [e_{(s_Q)_1}, b_{(s_Q)_1} + w] \; if \; dim(Q) = 1 \\ O_M \cap [e_{(s_Q)_k}, b_{(s_Q)_k} + w] \; otherwise \end{cases}$$

where M is the $(k-1)-$prefix of Q.

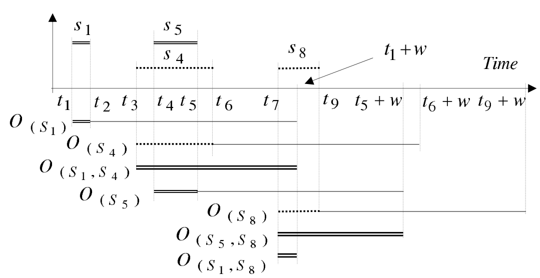

Fig. 3. The observability interval of the temporal pattern "f^- $before$ h^+".

Example 5: Let us consider Q, the temporal pattern given in Example 3, and we take $[t_1, t_7]$ as the sliding window ($w = t_7 - t_1$), the pattern Q has three instances (s_1, s_4), (s_1, s_8) and (s_5, s_8) (see Figure 3). Their respective observability intervals are $[t_3, t_2 + w]$, $[t_7, t_2 + w]$ and $[t_7, t_5 + w]$. If we consider just the disjoined instances of Q, we can confirm that the observation of (s_1, s_4) is sufficient to observe (s_1, s_8) (because $O_{(s_1,s_8)} = [t_7, t_2 + w] \subseteq [t_5, t_2 + w] = O_{(s_1,s_4)}$). Than, let us confine our attention just to (s_1, s_4) and (s_5, s_8), the two instances not disjoined of Q. We can confirm that the observation of Q is over $[t_3, t_2 + w] \cup [t_7, t_5 + w] = [t_3, t_5 + w]$. To declare P, a given $k - TP$, frequent or not we must to consider this temporal pattern within $S = (s_1, \ldots, s_n)$. Thus, P is considered frequent iff the ratio of the temporal support of P and the duration of S is greater than or equal σ, a given threshold, i.e.:

$$T_{supp}(P)/D(S) \geq \sigma$$

where $D(S)$, the duration of S is given as follows:

$$D(S) = card([min\{b(s_i)|s_i \in S\}, max\{e(s_i)|s_i \in S\}])$$

B. Candidate Generation Method

The most works for mining frequent pattern explore all possible combinations between primitives. Instead, in our work we use exploit the adjacent relation for a better selection of the patterns two by two. Therefore, the candidate generation method, denoted DATTES-Gen, generates C, a single candidate $(k+1) - TP$, from P and Q, two adjacent $k - TP$. These two adjacent patterns are represented respectively by $S_P = (s_i, \ldots, s_{i+k-1})$ and $S_Q = (s_{i+1}, \ldots, s_{i+k})$, and their temporal matrix are:

$$R_P = \begin{pmatrix} eq & r_i^{i+1} & \ldots & r_i^{i+k-1} \\ r_{i+1}^i & eq & \ldots & r_{i+1}^{i+k-1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{i+k-1}^i & r_{i+k-1}^{i+1} & \ldots & eq \end{pmatrix}$$

$$R_Q = \begin{pmatrix} eq & \ldots & r_{i+1}^{i+k-1} & r_{i+1}^{i+k} \\ \vdots & \ddots & \vdots & \vdots \\ r_{i+k-1}^{i+1} & \ldots & eq & r_{i+k-1}^{i+k} \\ r_{i+k}^{i+1} & \ldots & r_{i+k}^{i+k-1} & eq \end{pmatrix}$$

The candidate pattern $C = (S_C, R_C)$ maintains P as prefix pattern: DATTES-Gen adds s_{i+k}, the last state of S_Q, to S_P, to achieve $S_C = (s_i, s_{i+1}, \ldots, s_{i+k-1}, s_{i+k})$. The main idea to create R_C is to maintain R_P as prefix matrix and to add a new line and a new column. DATTES-Gen computes only r and its converse ir. r denotes the temporal relation between s_i and s_{i+k}, respectively the first interval primitive of P and the last one of Q. R_C is given as follows:

$$R_C = \begin{pmatrix} eq & r_i^{i+1} & \ldots & r_i^{i+k-1} & r \\ r_{i+1}^i & eq & \ldots & r_{i+1}^{i+k-1} & r_i^{i+k} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{i+k-1}^i & r_{i+k-1}^{i+1} & \ldots & eq & r_{i+k-1}^{i+k} \\ ir & r_{i+k}^{i+1} & \ldots & r_{i+k}^{i+k-1} & eq \end{pmatrix}$$

Example 6: Let us consider $(s_1, s_2; (s))$ and $(s_2, s_3; (o))$, two adjacent $2 - TP$s represented in Figure 2. The $3 - TP$ generated by the DATTES-Gen is $(s_1, s_2, s_3; (s)(o, m))$.

C. Algorithm Details

Now, we present DATTES, our algorithm for discovering the frequent patterns in temporal sequences. DATTES starts with the generation of the $1 - TP$, the elementary patterns, which always have the observability intervals like temporal support (steps 1 to 3). The second step uses the results of the first one to generate the $2 - TP$. Step by step, new patterns of higher dimension respecting the adjacent relation are generated (step 8), and the temporal support (step 15). At each iteration k, DATTES creates a new subset of candidate $(k+1) - TP$ from adjacent $k - TP$ (step 10) discovered during the previous iteration by the using DATTES-Gen. Step 9 guaranties that the two adjacent hold in the same sliding window. After the generation of the candidate pattern, DATTES computes the observability interval of this candidate (step 11). It takes the observability interval of the second joined pattern and updates the right extremity. To consider the candidate pattern frequent, DATTES computes the ratio between the support of the candidate pattern and the temporal support of S, the temporal sequence (loop 14 to 16). The process is repeated until no more frequent temporal pattern can be found (loop 5 to 17). The fact that the temporal support of a temporal pattern is always greater or equal to the temporal support of any of its subpatterns [3].

DATTES algorithm

Input: $S = (s_1, s_2, \ldots, s_n)$ the temporal sequence,
  w the sliding windows width,
  σ the temporal support threshold.
Output: Max_F the optimal set of TP
1. $F_1 = \{((s_i); (R_i)) | s_i \in S \wedge R_i(1, 1) = eq\}$;
2. For $(i = 1$ to $|F_1|)$
3. $\quad O_{F_1[i]} = [b_i, e_i + w]$;
4. $k = 2$; $T_{supp}(S) = (e_n - b_1) + w$;
5. While $(F_{k-1} \neq \emptyset)$
6. $\quad F_k = \emptyset$;$i = 1$;
7. $\quad$ While $(i < |F_{k-1}|)$
8. $\quad\quad$ If $((F_{k-1}[i] \| F_{k-1}[i + 1])$ and
9. $\quad\quad\quad (O_{F_{k-1}[i]} \cap O_{F_{k-1}[i+1]} \neq \emptyset))$;
10. $\quad\quad\quad C =$DATTES-Gen$(F_{k-1}[i], F_{k-1}[i + 1])$;
11. $\quad\quad\quad O_C = [b_{F_{k-1}[i+1]}, min(e_{F_{k-1}[i]}, e_{F_{k-1}[i+1]})]$;
12. $\quad\quad\quad C_k = C_k \cup \{C\}$;
13. $\quad i = i + 1$;
14. $\quad$ For $(j = 1$ to $|C_k|)$
15. $\quad\quad if(T_{supp}(C_k[j])/D(S) \geq \sigma)$
16. $\quad\quad\quad F_k = F_k \cup \{C_k[j]\}$;
17. $k = k + 1$;
18. $Max_F = \cup_{i=1}^{i=k-1} F_i$;

EndDATTES

The number of all generated patterns is given by the following theorem.

Theorem 2: Given a temporal sequence of dimension n. DATTES generates at least $n(n + 1)/2$ frequent patterns independently with the sliding window width.

Proof: Let n be the dimension of the given intervals sequence S. We note S_i the interesting $i-TP$ subset. Initially, DATTES considers the n elementary patterns interesting (steps 1 to 3) i.e. $S_1 = n$. The generator function (step 10) guarantees that, at each level k, the number of the candidate $(k+1)$-patterns is at least the number of interesting k-pattern truncated by one, i.e. $dim(F_{k+1}) \leq dim(S_k) - 1$. The step 9 confirms that the number of interesting patterns is at least the number of candidate patterns, i.e.: $dim(S_k) \leq (F_k)$. The number of temporal patterns generated by DATTES, noted z, is computed as follow:

$$
\begin{aligned}
z =& dim(S_1) + \ldots + dim(S_j) + \ldots + dim(S_n) \\
&\leq dim(F_1) + \ldots + dim(F_j) + \ldots + dim(F_n) \\
&\leq (n - 1 + 1) + \ldots + (n - j + 1) + \ldots + (n - n + 1) \\
&\leq \textstyle\sum_{k=1}^{n} k = n(n + 1)/2
\end{aligned}
$$

IV. Mining Temporal Rules

Having defined the compact set of frequent temporal patterns, we are now interested to generate the temporal association rules $X \Rightarrow Y$. The problem was first defined in the context of the market basket data to identify customer buying habits [1]. For a temporal sequence, we want to discover similar association rules, which reflect the relationships between the frequent temporal patterns. In such association rules, X and Y are frequent temporal patterns and X is the $k-$prefix of Y with $k \in \{1 \ldots dim(s_Y)\}$(i.e. $X \subseteq Y$). Such a temporal rule reveals that the subsequence in the temporal sequence, containing pattern X tend to contain pattern Y, and the probability, measured as the fractions of the temporal support of the pattern X also of Y, is called the confidence of the temporal rule, i.e.:

$$TAR_{conf}(X \Rightarrow Y) = T_{supp}(Y)/T_{supp}(X)$$

For an association rule to hold, the temporal confidence of the temporal rule should satisfy a user-specified minimum confidence. Enumeration of all possible temporal rules can be done efficiently by using techniques described in [3].

V. Evaluation and Discussion

In order to show how DATTES performs, when it is run to generate frequent patterns, we have chosen to compare it with Apriori-Like [6]. The considered application contains real dataset (Human face identification problem proposed in [5]). We use our proposed framework in order to detect individual facial movements and their characteristics in men and in women. Behavioral results indicate quantitative differences between men and women.

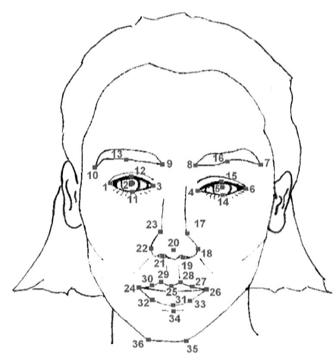

Fig. 4. Position of the face points

A. Dataset Definition

The basic dataset concern 10 persons (5 women and 5 men) volunteered to participate in the study. Subjects were received by an experimenter and then left alone in a room where they followed instructions given by a laptop screen. The task consisted in looking at pictures and saying whether it was ambiguous/normal or not. Some subjects were set aside because of particular situations (important movements of the body or the head, wearing glasses or a beard, etc.). With a view to standardizing the dataset the experimenter chose 3 sequences (a, b and c) of 3 seconds centered on an easy to locate verbal answer from the subjects. Thus 3 sequences with a similar context are available per subject. Figure 4 indicates 36 face points involved in the facial movements that were easy to identify [11]. The sequences were sampled at 13 images per second, and an operator recorded the 36 face point coordinates. This selection was repeated at least twice and the mean position was retained to reduce errors due to tiredness of the operator. The coordinates of the points were relative to the subjects face.

B. FTP Space Evolution

For the sake of curiosity, we have compared the number of patterns generated by DATTES and Apriori-Like proposed in [6]. For the experiments, we have chosen two subjects Man_4^b and $Woman_4^b$ (the second sequence of fourth subject) and we have varied the window width w and fixed the minimal threshold support σ to 2% of w of the sequence duration.

0	1	2	5	10	0	1	2	5	10
26	26	26	26	26	24	24	24	24	24
17	20	21	23	25	11	17	18	22	23
13	15	17	19	24	4	11	14	17	22
10	13	15	16	23	1	4	9	14	19
6	11	13	14	21		1	3	12	17
3	8	11	12	19			2	10	15
1	5	9	10	17			1	8	13
	4	6	8	15				4	11
	2	5	6	13				1	9
		4	5	9					7
		1	4	7					6
			3	6					5
			2	4					3
			1	3					1
				1					
76	104	128	149	213	40	57	71	112	175

TABLE I
FTP SPACE EVOLUTION FOR DATTES ALGORITHM.

TABLE I and TABLE II show the evolution of the temporal pattern space with the variation of the width of sliding window respectively for DATTES algorithm and for Apriori-Like algorithm. The first line denotes the window width. The first and the last five columns concerns respectively two subjects $Woman_4^b$ and Man_4^b. The next 15th lines correspond to the step number. They contain the number of FTP generated at each step. Note that, the last lines denotes the number of distinct primitive in the temporal sequence. TABLE I shows a simple decreasing trend for the evolution of the patterns number. Therefore, the number of candidate patterns, generated at each step, is bounded by the number of the frequent temporal patterns at the preceding iteration.

In contrast, TABLE II illustrates the combinatorial explosion of the candidate patterns generation for Apriori-Like algorithm: It explores all possible combination between patterns, which is very expensive. The colossal difference in pattern space between DATTES and Apriori-Like is expressed by the last line in TABLE I and the last line in TABLE II. These two lines summarize the evolution of the two algorithms, and give a significant comparison. The two algorithms are initialized by the same number of elementary patterns. The difference between the two algorithms for becomes colossal when the width of the sliding window increases and becomes vital for heavy sliding window ($w > 5$) which justify the time complexity and the combinatorial explosion of Apriori-Like algorithm. This difference can be case justified by the next theorem.

Theorem 3: Given n the primitives number in the sequence S. The number of temporal patterns extracted from the FTP space generated by DATTES is smaller than $\sum_{k=3}^{n}(n-k+1)(2^{k-2}-1)$.

Proof: Let $((s_P)_1, (s_P)_2, \ldots, (s_P)_k))$ be the primitives series of P, a frequent $k-TP$ generated by DATTES at level $k > 2$. The extraction process maintains the two primitives bordering, $(s_P)_1$ and $(s_P)_k$, and takes all possible ordering combinations of the $k-2$ remainder intervals. Than, we have 2^{k-2} possible combinations. The number of extracted pattern is the number of possible combination truncated by the last greatest extracted pattern because it refers to the concerned P i.e. $2^{k-2}-1$ for each level of DATTES. In consequence, DATTES generates at least $(n-k+1)$ frequent $k-TP$. We extract at least $(n-k+1)(2^{k-2}-1)$ complementary patterns. Thus, the quantity of complementary patterns is bounded by the sum of the complementary patterns at level k with $2 < k \leq n$ and it expresses by $\sum_{k=3}^{n}(n-k+1)(2^{k-2}-1)$.

0	1	2	5	10	0	1	2	5	10
26	26	26	26	26	24	24	24	24	24
57	81	104	124	189	20	35	47	88	151
95	189	302	428	780	10	27	54	192	570
100	305	604	1092	2202	2	11	40	271	1512
68	344	854	2093	4580		2	21	247	2953
30	273	868	3066	7272			7	145	4341
8	149	632	3461	8971			1	53	4854
1	53	322	3011	8654				11	4134
	11	109	2003	6513				1	2663
	1	22	1001	3784					1276
		2	364	1663					441
			91	534					104
			14	118					15
			1	16					1
				1					
385	1432	3845	16775	45303	56	99	194	1032	23039

TABLE II
FTP SPACE EVOLUTION FOR APRIORI-LIKE ALGORITHM.

C. Temporal Rule Generation

As mentioned above, we are interested to detect individual facial movements and their characteristics in men and in women. Thus, we aim to extract the maximum FTP for each subject (men or women) i.e. as each subject has 3 sequences (a, b and c), we try to mine FTP space these sequences. The next step is the construction of the temporal rule space for each subject. In this step we are interested on the X, $k-TP$, is considered as the $k-$prefix of Y, $(k+1)-TP$. After, the intersection of these three temporal rule spaces generates the rule space for each subject. Finally, we mine the rule model for each gender by the unification of the rule space for each gender: $Rule\_male$ for male and $Rule\_female$ for female. For the experiments, we have varied the sliding window width w and fixed the minimal threshold support σ and the minimal threshold confidence θ respectively to 2% and 25%.

TABLE III shows the temporal rule space for each gender model. This illustrates a quantitative comparison between the female model the male model. The inspection of the rule spaces indicates that male produce more temporal rule than female. It's due to the fact that the gap between the primitives constituting the interval sequences is large between the primitives generated by the women subject that those between the men subject.

To evaluate a qualitative comparison between the two gender model, we have varied w (see the first line in TABLE III) and fixed σ and θ respectively to 2% and 1%. In this special case we are attracted by the rules concerned the male model $\{\ (25) \rightarrow (25,28;(e)), (2) \rightarrow (2,3;(e))\ \}$, the rule concerned the female model $\{(14) \rightarrow (14,15;(e)), (17) \rightarrow (17,18;(e)), (29) \rightarrow (29,30;(e)), (4) \rightarrow (4,5;(e)), (11) \rightarrow (11,12;(e)), (3) \rightarrow (3,4;(e)), (4,5;(e)) \rightarrow (4,5,6;(e)(e,e))\}$ and the shared rules between these two models $\{(5) \rightarrow (5,6;(e))\}$. Note that the confidence of each temporal rule growth with the window width.

A deeper analysis of the rules values emphasized that male model (respectively female model) is characterized by more structured movements of points 2,3,25 and 28 (respectively 3, 4, 5, 6, 11, 12, 14, 15, 17, 18, 29 and 30). We notice that the top of the face produces more information on the subject gender. If we consider the vertical distribution of the points face, $2/3$ rules are from the high part of the face whereas if we consider the horizontal distribution, $5/9$ rules are from the left part of the face. To conclude, $4/9$ generated rules are expressed from the high left part of the face.

w	0	2	10	20	40
$(25) \rightarrow (25,28;(e))$	1	1	1	1	1
$(2) \rightarrow (2,3;(e))$	0,5	0,6	0.85	0,92	0,95
$(5) \rightarrow (5,6;(e))$	1	1	1	1	1
$(14) \rightarrow (14,15;(e))$	5	0.5	0.5	0.5	0.53
$(17) \rightarrow (17,18;(e))$	0,33	0.33	0.38	0.45	0.61
$(29) \rightarrow (29,30;(e))$	0,75	0.7	0.25	0.47	0.67
$(4) \rightarrow (4,5;(e))$	0,5	0.5	0.5	0.5	0.05
$(11) \rightarrow (11,12;(e))$	0,5	0.5	0.5	0.5	0.63
$(3) \rightarrow (3,4;(e))$	0,33	0.33	0.33	0.36	0.53
$(4,5;(e)) \rightarrow (4,5,6;(e)(e,e))$	0,5	0.5	0.5	0.5	1.0

TABLE III
TEMPORAL RULES SPACE EVOLUTION.

VI. CONCLUSION

In this paper, we investigated an approach for temporal association rules mining. This approach, based on a new framework, treats the temporal sequences as an ordered collection of intervals and matrix relationships between them. The main component of our proposed framework is DATTES, a new algorithm for discovering the optimal set of frequent temporal patterns in sequences, and a new method for temporal rule construction. In more that DATTES is based on the anti-monotone Apriori property and explores some properties of interval algebra relations (ex. symmetric property), our proposed algorithm exploits a new support evaluation function. Compared to the generalization of Apriori algorithm for interval sequences, the number of temporal patterns generated by DATTES is polynomial according to the dimension of the patterns and the width of the sliding window.

The temporal rules construction method, based on the precedence relation, takes the ratio between the temporal support of the conclusion rule and the temporal support of the premise rule as the confidence evaluation function. This function favors the temporal rules that the gap between the composite primitives.

Using the new framework, it is possible to extract all other frequent patterns and all other possible temporal rules respecting both the adjacent relation and the prefix relation. The extraction of the complementary patterns and rules is done without explicit checking of Apriori property.

REFERENCES

[1] R. Agrawal, T. Imielinski and A. Swami: *Mining Association Rules Between Sets of Items in Large Datasets*. In *Pro. of the ACM SIGMOD Conference on Management of Data*, 1993.

[2] R. Agrawal and R. Srikant: *Mining Sequentiel Motifs*. In *11th Int'l Conf. on Data Engineering* 1995.

[3] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and I. Verkamo: *Fast Discovery of Association Rules*. In *Advances in Knowledge Discovery and Data Mining, 1996.*

[4] J. F. Allen: *Maintaining Knowledge about Temporal Intervals*. In *Comm. of the ACM*, 1983.

[5] S. Guérif, Y. Bennani and C. Baudoin. *Connectionist and Ethological Approaches for Discovering Salient Facial Movement Features in Human Gender Recognition*. In *the 28th ITI'06*, 2006.

[6] F. Höppner: *Discovery of Temporal Patterns Learning Rules about the Qualitative Behaviour of Time Series*. In *Proc. of the 5th European Conf. on Principles and Practice of KDD*, 2001.

[7] E. Keogh: *Exact indexing of dynamic time warping.* In *Proc. of VLDB02*, 2002.

[8] A. Lerner, D. Shasha, Z. Wang, X. Zhao and Y. Zhu: *Fast Algorithms for Time Series with applications to Finance, Physics, Music, Biology, and other Suspects*. In *ACM SIGMOD'04*, 2004.

[9] J. Lin, E. Keogh, S. Lonardi, J.P. Lankford and D.M. Nystrom: *Visually Mining and Monitoring Massive Time Series.*In *KDD04*, 2004.

[10] H. Mannila, H. Toivonen and I. A. Verkamo: *Discovery of Frequent Episods in Event Sequence*. In *Data Mining and Knowledge Discovery*, 1997.

[11] C-H. J. Tzou, P. Giovanoli, M. Ploner and M. Frey: *Are there ethnic differences of facial movements between Europeans ans Asians? British Journal of Plastic Surgery* , 2005.

[12] J. Wang, W. Hsu and M. L. Lee: *A Framwork for Mining Topological Patterns in Spatio-Temporal Databases*. In *CIKM'05*, 2005.

[13] M.J. Zaki and K. Gouda: *GenMax: an Efficient Algorithm for Mining Maximal Frequent Itemsets*. In *Data Mining and Knowledge Discovery*, 2005.