Temporal Feature Extraction from Temporal Information Systems *

Piotr Synak

Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland

Abstract. We present a framework for new feature extraction from temporal information systems. Such systems provide information about behaviour of object(s) in time and state of an object is described by some attributes. We propose a modification of Apriori algorithm for frequent episode detection across many series. The episodes are built from events being temporal patterns found in temporal information system.

1 Introduction

Temporal data analysis is a topic of high interest in many real world applications like weather forecasting or web log processing. There are several kinds of temporal data, from classical time series, through temporal sequences to temporal information systems. Each type of data, as well as field of application, arises new problems, which in many cases are very hard to solve. Therefore, there is no any universal method that could be applied to temporal data of any kind.

In the paper we focus on, so called, temporal information systems, which describe objects changing in time in terms of attributes. The attributes can be both of numeric or symbolic type. In our approach we search for temporal patterns in series related to investigated objects. Some specific configurations of discovered patterns may appear to be characteristic to, e.g., particular groups or classes of objects. Thus, we search across many objects for kind of episodes described in a language of temporal patterns. We propose an algorithm for such episodes detection, which is a modification of Apriori algorithm. Episodes specific to groups or classes of objects can leverage the clustering or classification algorithms.

We present a complete framework describing all the basic stages of analysis, from temporal information system processing to new feature extraction. The framework is highly parameterised, so tuning parameters is an important step of the presented method. We present some results of experiments related to musical instrument sound analysis. They seem to be very satisfactory.

2 Temporal data

There are many kinds of temporal data being investigated in real applications. The most commonly known are *time series* data, which are results of some observations usually ordered in time, i.e. $X = \{x_t : t = 1, \ldots, n\}$, where x_t is a numerical value observed in time t. There are several examples of time

* Supported by Polish National Committee for Scientific Research (KBN) grant No. $8T11C02519$ and PJIIT grant No.$PJ/AI/03/2002$.

$\mathbb{A}$	*User_id*	*Page*	*Duration*
t_1	1	*a.htm*	5
t_2	3	*b.htm*	4
t_3	1	*b.htm*	4
t_4	2	*a.htm*	6
t_5	3	*a.htm*	1
t_6	1	*c.htm*	1
t_7	1	*d.htm*	2
t_8	2	*d.htm*	5
t_9	3	*c.htm*	3
t_{10}	3	*d.htm*	5

Fig. 1. A temporal multiple information system – each $User\_id$ represents one series.

series: data describing seismic activity, stock market indicator's changes or EEG signals. The following are examples of problems related to time series analysis: forecasting – for a given sequence predict the future values, or predict occurrence of some special events (e.g. earthquakes); clustering – for given sequences group them into clusters of similar series; classification – for a given sequence classify it to the right class. Time series are widely studied in the literature [7, 5, 10].

Temporal sequences are another class of temporal data. There is given an alphabet E of event types. An event is a pair (e, t), where $e \in E$ and t is time index. Events can be of numerical or symbolic type and time is discrete. Temporal sequence is then sequence of events. There are several examples of temporal sequences, e.g. logs of telecommunication network monitors or web logs. In the first case event types are different notification types in the network, in the second – different pages of given web site. There can be several problems formulated for this kind of data including searching for frequent episodes, i.e. collections of events occurring frequently in data [11, 12].

In several cases temporal data are described by many features of both numerical and symbolic types. *Temporal information system* is a system, which describes behaviour of one object in time. It is a pair $\mathbb{A} = (\{x_1, x_2, \ldots, x_n\}, A)$ of universe ordered in time and set of attributes A. An example is a system describing behaviour of stock market in time. There can be several economic and market indicators used to describe the situation on a market. One row of such system describes situation at given time index. In this case we can formulate e.g. classification problem: given values of indicators classify the the situation described by them [20].

Temporal information system is a special case of *temporal multiple information system* that describes behaviour in time of many objects. We can consider e.g. several users visiting a web site. Each request can be described by several attributes including referrer, elapsed time, etc. In this case the system describes several series (user sessions) in terms of attributes. The problems considered in this case include clustering – find sessions with similar behaviour and prediction – find characteristic patterns that precede given action taken by the user.

3 Temporal templates

In this section we present the notion of temporal templates, i.e. temporal patterns that can be generated from temporal information systems. Temporal templates

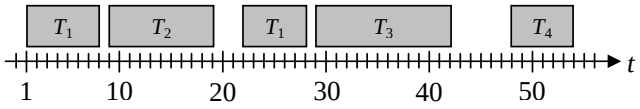

Fig. 2. An example of temporal templates found for one series.

can be of numeric or symbolic type. They are built by using expressions $(a \in V)$, where a is an attribute and V is a set of values.

Let $\mathbb{A} = (\{x_1, x_2, \ldots, x_n\}, A)$ be a temporal information system. By *descriptor* over $\mathbb{A}$ we understand any expression of form $(a \in V)$, where $a \in A$ and V is a subset of V_a – domain of attribute a. *Temporal template* is then a set of expressions involving any subset $B \subseteq A$: $\mathbf{T} = (T, t_s, t_e)$, where $T = \{(a \in V) : a \in B, V \subseteq V_a\}$ and $1 \leq t_s \leq t_e \leq n$. The set of descriptors T we call a *template* (or *generalised template*). Temporal templates are intensively studied in the literature ([1], [15], [20]). To outline the intuition which is in behind, let us understand temporal template as strong regularity occurring in time.

There can be defined several properties of temporal templates. By *width* of $\mathbf{T} = (T, t_s, t_e)$ we mean $t_e - t_s + 1$. *Support* of $\mathbf{T}$ is the number of objects from interval $[t_s, t_e]$ that match all of the descriptors from T. The *precision* of template T in $\mathbb{A}$ we define by

$$S_{\mathbb{A}}(T) = \sum_{(a \in V) \in T} s^T_{\mathbb{A}}((a \in V)), \tag{1}$$

where $s^T_{\mathbb{A}}((a \in V))$ is a measure of how much a descriptor $(a \in V)$ is specific (the exact definition can be found in [20]). If template T consists of many descriptors that describe the whole domain of attributes, it is a general template (not precise). If T consists of one-value descriptors only, it is very specific (precise) and then $S_{\mathbb{A}}(T)$ gives the number of descriptors of T (length of T). Upon to the investigated problem we can also define *quality* of temporal template as some function of width, support, number of descriptors and precision.

Let us outline an algorithm for temporal templates generation [20]. It uses time window of given size, which is moved across information system. In each window there is generated an optimal template, which is compared to that from previous time window. In this way we can compute time of validity $[t_s, t_e]$ of found template. The algorithm results with a sequence of temporal templates found for the whole temporal information system. However, there can be a time interval without any satisfactory (in terms of quality function) template at all, i.e. without any strong regularity.

4 Temporal feature extraction

In this section we present a step-by-step schema of temporal feature extraction process from temporal multiple information systems. This schema consists of many parameters that are about to be tuned up while applying to the real problem. There are five main steps. In the first step the input information system is pre-processed. Then for each sequence there are computed temporal patterns

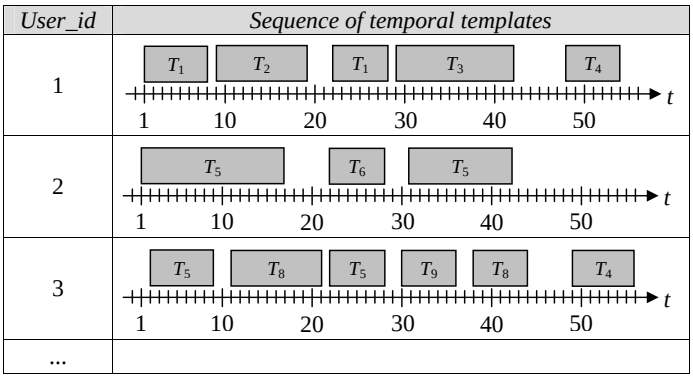

Fig. 3. A sequence of temporal templates found for each series ($User\_id$).

(templates). Next, found patterns need to be processed in order to, e.g., decrease the total amount of different patterns, so the extracted information is more general. In the fourth step we search for collections of patterns (episodes) appearing in sequences frequently. Finally, we generate new attributes from found frequent episodes.

4.1 Data pre-processing

Information stored in data can be of numerical and/or symbolic type. In case when there are too many different values no strong regularity may occur at all. Thus, the pre-processing step is needed to discretise (numerical case) or group (symbolic case) stored information. In the simplest case real-value attributes can be discretised by using a uniform scale quantisation (the number of intervals is a parameter). More sophisticated methods can be found, e.g., in [14].

4.2 Temporal patterns extraction

We propose to search for temporal patterns in series – their collections may be specific to some groups of investigated objects, what can be very helpful in solving clustering or classification problems. Because temporal multiple information systems consists of many series, i.e. objects changing in time, we propose to search for temporal templates in each series separately. The mentioned algorithm for temporal templates generation results with a sequence of templates for each series, i.e. with a set of sequences of temporal templates. Let us observe that this set may consist of relatively large number of different (in terms of descriptors only) templates. In the pessimistic case there can be no repeating templates at all, and then it makes no sense to search for combinations of templates occurring across sequences frequently. Therefore, we propose to search for template representatives and use them only for encoding of sequences of templates. Such template representatives are computed from the set of all templates by taking optimal ones with respect to some quality measures. In the simplest case we can compute frequencies of templates, i.e., number of different sequences containing given template. We can use the information about decision class where given sequence belongs to, i.e. we can generate most frequent templates for each class

separately. The following measures are examples based on theories of machine and statistical learning, as well as rough sets. Let $\mathbf{T} = (T, t_s, t_e)$ be a temporal template. Measure

$$BayesDist(T) = \sum_{v_d \in V_d} |P(T/v_d) - P(T)| \tag{2}$$

describes the impact of decision classes onto probability of T. By $P(T)$ we mean simply the prior probability that elements of the universe satisfy T, estimated directly from data. By $P(T/v_d)$ we mean the same probability, but derived from the decision class corresponding to $v_d \in V_d$. If we regard T as the left side of an inexact decision rule $T \Rightarrow d = v_d$, then $P(T/v_d)$ describes its sensitivity [13]. Quantities $|P(T/v_d) - P(T)|$ express a kind of degree of information we gain about T given knowledge about membership of the analysed objects to particular decision classes. According to the Bayesian reasoning principles [6], $BayesDist(T)$ provides the degree of information we gain about decision probabilistic distribution, given additional knowledge about satisfaction of T. The second exemplary measure

$$RoughDisc(T) = P(T)(1 - P(T)) - \sum_{v_d \in V_d} P(T, v_d)(P(v_d) - P(T, v_d)) \tag{3}$$

is an adaptation of one of the rough set measures used, e.g., in [14] and [21] to express the number of pairs of objects belonging to different decision classes, being discerned by a specified condition. Normalisation, understood as division by the number of all possible pairs, results with (3), where $P(T)$ is understood as before, $P(v_d)$ denotes the normalised cardinality of the decision class corresponding to $v_d \in V_d$ and $P(T, v_d)$ is the joint probability of satisfying conditions T and $d = v_d$.

4.3 Temporal patterns processing

We use template representatives to replace all the templates in found sequences of templates. Any template is substituted with the closest representative. In this way the total number of different templates is limited to the number of representatives, which is a parameter. There can be defined several measures of closeness between templates. In general the closeness can be defined as follows:

$$cl(T_1, T_2) = \sum_{a \in A} cl_a(T_1(a), T_2(a)) \tag{4}$$

where $T_i(a)$, $i = 1, 2$, is the descriptor of template T_i corresponding to attribute a (or empty set if there is no such descriptor), and cl_a is closeness function between descriptors related to a. Let us note that attributes can be of both numerical or symbolic type and closeness can be defined in different ways for different attributes. For example, if a is a real attribute and descriptors of temporal templates are described by intervals, we can have the following definition:

$$cl_a((a \in v_1), (a \in v_2)) = \frac{|v_1 \cap v_2|}{|v_1 \cup v_2|} \tag{5}$$

where $|.|$ is the length of interval. In case when a is numerical, but templates are built from one-value descriptors we can define closeness by

$$cl_a((a = v_1), (a = v_2)) = 1 - \frac{|v_1 - v_2|}{|V_a|} \tag{6}$$

where $|V_a|$ is the length of domain of attribute a. For symbolic attributes we can define the closeness by

$$cl_a((a \in v_1), (a \in v_2)) = \frac{card(v_1 \cap v_2)}{card(v_1 \cup v_2)}. \tag{7}$$

4.4 Frequent episode generation

Till now, for each series we have generated a sequence of temporal templates, each of them being replaced by the closest representative. We can expect that some combinations of representative templates may be specific to particular decision classes (one or more). Let us reformulate this problem in terms of events. We can treat each template representative as an *event*. Then any sequence of templates is a sequence of events and any collection of templates is an *episode*. Thus, from sequences of templates we can discover frequent episodes – collections of templates occurring together [12, 20]. We say that an episode *occurs* in a sequence of templates if each element (template) of the episode exists in the sequence and the order of occurrence is preserved. For example, an episode AAC occurs in a sequence $B\underline{A}C\underline{A}B\underline{B}\underline{C}$. Definition of episode occurrence depends on the particular application. We can modify the above definition by not preserving order of occurrence of events in episodes. In this way we obtain parallel episodes while in the former case we have serial episodes.

We propose an algorithm, based on Apriori method [2, 12], which discovers frequently occurring episodes with respect to some occurrence measure Occ. The main difference, comparing, e.g., to Winepi algorithm [12], is that here we are looking for episodes across many series of events. The input to the algorithm is a set of template sequences and the frequency threshold τ.

Detection of episodes occurring frequently
1. $\mathcal{F}_1 = \{1\text{-sequences occurring frequently}\}$
2. for $(l = 2;\ \mathcal{F}_{l-1} \neq \emptyset;\ l{+}{+})$ {
3. $\quad \mathcal{C}_l = GenCandidates(\mathcal{F}_1, \mathcal{F}_{l-1}, l)$
4. $\quad \mathcal{F}_l = \{c \in \mathcal{C}_l : Occ(c) \geq \tau\}$ }
5. return $\bigcup_l \mathcal{F}_l$

At first, we check which templates occur frequently enough in all sequences. That forms the set $\mathcal{F}_1$ of frequent episodes of length one. We can use several measures of occurrence. The fundamental one is the number of occurrences, however, by being "frequent" we can also understand frequent occurrence in one class and rare occurrence in another classes. Thus, we can adapt measures (2), (3) to the definition of function Occ. Next, we recursively create a set of candidates $\mathcal{C}_l$ by

combining frequent templates ($\mathcal{F}_1$) with frequently occurring episodes of size $l-1$ ($\mathcal{F}_{l-1}$). The last step is to verify the set of candidates $\mathcal{C}_l$ and eliminate infrequent episodes. The algorithm stops if there are no more candidates.

4.5 New attributes generation

Frequent episodes can be used for new feature generation. They carry information about combinations of temporal patterns specific to particular classes. Such a new feature can be used as additional attribute in an information system describing investigated series. New attributes may support process of series clustering or, in decision case, series classification.

We propose two definitions of new attributes based on occurrence of frequent episodes. Let us assume that in the training step we have computed frequent episodes for all series in the training sample. There can be several episodes out of them that occur in a testing object (series). The value of such a new attribute is simply the most frequent episode (in terms of function Occ). In the second case we take the longest episode occurring in a testing object. If there are several such episodes we choose the most frequent one (up to Occ).

Let us note that in both cases the attributes are of symbolic type. One can consider also numeric attributes generated from several properties of episodes.

5 Results of experiments

The presented method requires evaluation of many parameters. The most important ones are: window size (when generating temporal templates), number of representative templates, quality and frequency measure, and frequency threshold. We have tested the method on musical instrument sound data [16]. The investigated problem is to classify samples of sound to one out of 18 classes (11 instruments + different articulations). There are 667 different samples in the database and for each sample we generated 16 spectral attributes (see [22] for details). In the experiments we built a classifier, based on decision rules generation by using rough set theory. For evaluation we used standard CV-5 method.

After tuning the parameters we have obtained an attribute with average prediction quality higher than 40%. The result seems to be significant – in [22] several spectral attributes give about 15%. It is also comparable to temporal attributes presented in [22], however here the result is for one single attribute.

6 Conclusions

We have presented a framework for supporting analysis of, so called, temporal multiple information systems, i.e., information systems describing temporal behaviour of objects. We have shown how notion of temporal templates can be used to temporal patterns extraction from sequences. We have also proposed modification of Apriori algorithm for frequent episodes detection in many series. The presented approach can be used for new temporal features extraction and supporting clustering or classification problems.

Further research is needed to establish a broader family of valuable episode based descriptors. Mainly, investigation of numerical features based on frequent episodes seems to be of high importance.

References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pp. 307–328, 1996. AAAI Press.
2. R. Agrawal and R. Srikant. Mining sequential patterns. In P. S. Yu and A. S. P. Chen (eds.), *XIX Int. Conf. on Data Engineering*, pp. 3–14, 1995. IEEE Press.
3. J. Bazan, H. S. Nguyen, S. H. Nguyen, P. Synak, and J. Wróblewski. Rough set algorithms in classification problems. L. Polkowski, T. Y. Lin, S. Tsumoto (eds.), *Studies in Fuzziness and Soft Computing*, vol. 56, pp. 49–88. Springer, 2000.
4. J. Bazan, A. Skowron, and P. Synak. Market data analysis: A rough set approach. ICS Research Reports 6, Warsaw Univeristy of Technology, Warsaw, Poland, 1994.
5. B. Bollobas, G. Das, D. Gunopulos, and H. Mannila. Time-series similarity problems and well-separated geometric sets. In *Symposium on Computational Geometry*, pp. 454–456, 1997.
6. Box, G., and Tiao, G. C. Bayesian Inference in Statistical Analysis. Wiley, 1992.
7. P.J. Brockwell and R. Davis. *Introduction to Time-Series and Forecasting.* Springer-Verlag, Berlin, Germany, 1996.
8. G. Das, D. Gunopulos, and H. Mannila. Finding similar time series. In *Principles of Data Mining and Knowledge Discovery*, pp. 88–100, 1997.
9. V. Guralnik and J. Srivastava. Event detection from time series data. In *Fifth ACM SIGKDD Int. Conf. on KDD*, pp. 33–42, 1999. ACM Press.
10. E. Keogh and M. Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro (eds.), *IV Int. Conf. on KDD*, pp. 239–241, 1998. ACM Press.
11. H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in sequences. *First Int. Conf. on KDD*, pp. 210–215, 1995. AAAI Press.
12. H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, 1997.
13. Mitchell, T. Machine Learning. Mc Graw Hill, 1998.
14. H. S. Nguyen. *Discretization of Real Value Attributes, Boolean Reasoning Approach.* PhD thesis, Warsaw University, Poland, 1997.
15. S. H. Nguyen. *Regularity Analysis And Its Applications In Data Mining.* PhD thesis, Warsaw University, Poland, 2000.
16. Opolko, F., Wapnick, J.: MUMS – McGill University Master Samples. CD's (1987).
17. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data, Series D: System Theory, Knowledge Engineering and Problem Solving*, vol. 9. Kluwer, 1991.
18. L. Polkowski and A. Skowron (eds.), *Rough Sets in Knowledge Discovery 1: Methodology and Applications, Studies in Fuzziness and Soft Computing*, vol. 18. Physica-Verlag, 1998.
19. Richard J. Povinelli. Identifying temporal patterns for characterization and prediction of financial time series events. In J. F. Roddick and K. Hornsby (eds.), LNCS 2007, pp. 46–61, 2000. Springer-Verlag.
20. P. Synak. Temporal templates and analysis of time related data. In W. Ziarko and Y. Yao (eds.), LNAI 2005, pp. 420–427, 2000. Springer-Verlag.
21. D. Ślęzak Approximate decision reducts. Ph.D. thesis, Institute of Mathematics, Warsaw University, 2001.
22. A. Wieczorkowska, J. Wróblewski, P. Synak, and D. Ślęzak. Application of temporal descriptors to musical instrument sound recognition. *Journal of Intelligent Information Systems*, 21(1), 2003.