Finding Informative Rules in Interval Sequences*

Frank Höppner and Frank Klawonn

Department of Electrical Engineering and Computer Science
University of Applied Sciences, Emden
Constantiaplatz 4
D-26723 Emden, Germany
frank.hoeppner@ieee.org

Abstract. Observing a binary feature over a period of time yields a sequence of observation intervals. To ease the access to continuous features (like time series), they are often broken down into attributed intervals, such that the attribute describes the series' behaviour within the segment (e.g. increasing, high-value, highly convex, etc.). In both cases, we obtain a sequence of interval data, in which temporal patterns and rules can be identified. A temporal pattern is defined as a set of labeled intervals together with their interval relationships described in terms of Allen's interval logic. In this paper, we consider the evaluation of such rules in order to find the most informative rules. We discuss rule semantics and outline deficiencies of the previously used rule evaluation. We apply the J-measure to rules with a modified semantics in order to better cope with different lengths of the temporal patterns. We also consider the problem of specializing temporal rules by additional attributes of the state intervals.

1 Introduction

Most of the data analysis methods assume static data, that is, they do not consider time explicitly. The value of attributes is provided for a single point in time, like "patient A has disease B". If we observe the attributes over a period of time, we have to attach a time interval in which the attribute holds, for example "patient A has had disease B from 1st to 7th of July". It may happen that patient A gets disease B a second time, therefore sequences of labeled state intervals (state sequences) can be viewed as a natural generalization of static attributes to time-varying domains. Compared to static data, little work has been done to analyse interval data.

Even in continuous domains discretization into intervals can be helpful. As an example, the problem of finding common characteristics of multiple time series or different parts of the same series requires a notion of similarity. If a process is subject to variation in time (translation or dilation), those measures used traditionally for estimating similarity (e.g. pointwise Euclidean norm) will fail

* This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) under grant no. Kl 648/1.

in providing useful hints about the time series similarity in terms of the cognitive perception of a human. This problem has been addressed by many authors in the literature [1, 4, 6]. In [8] we have used qualitative descriptions to divide up the time series in small segments (like increasing, high-value, convexly decreasing, etc.), each of it easy to grasp and understand by the human. Then, matching of time series reduces to the identification of patterns in interval sequences.

Motivated by association rule mining [2], and more specific the discovery of frequent episodes in event sequences [10], we have proposed a method to discover frequent temporal patterns from a single state sequence in [8]. From the patterns rules can be formed that identify dependencies between such patterns. In this paper, we reconsider rule semantics and the problem of rule evaluation. We use the J-measure [11] to rank rules by their information content. And we discuss how to specialize rules by incorporating additional information about the state intervals to further improve the rules.

The outline of the paper is as follows: In Sect. 2 we define our notion of a state sequence and temporal patterns. We briefly summarize the process of mining frequent patterns in Sect. 3. We discuss the problems in measuring how often a pattern occurs in a sequence in section 4 and concentrate on rule evaluation in section 5. Section 6 deals with specializing rules to increase their usefuleness (information content).

2 Temporal Patterns in State Sequences

Let $\mathcal{S}$ denote the set of all possible trends, properties, or states that we want to distinguish. A state $s \in \mathcal{S}$ holds during a period of time $[b, f)$ where b and f denote the *initial point* in time when we enter the state and the *final point* in time when the state no longer holds. A state sequence on $\mathcal{S}$ is a series of triples defining state intervals

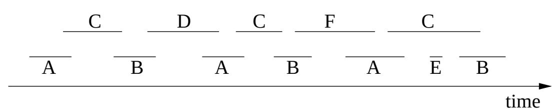
$$(b_1, s_1, f_1), (b_2, s_2, f_2), (b_3, s_3, f_3), (b_4, s_4, f_4), ...$$

where $b_i \leq b_{i+1}$ and $b_i < f_i$ holds. We do not require that one state interval has ended before another state interval starts. This enables us to mix up several state sequences (possibly obtained from different sources) into a single state sequences.

We use Allen's temporal interval logic [3] to describe the relation between state intervals. For any pair of intervals we have a set $\mathcal{I}$ of 13 possible relationships. For example, we say "A meets B" if interval A terminates at the same point in time at which B starts. The inverse relationship is "B is-met-by A". Given n state intervals (b_i, s_i, f_i), $1 \leq i \leq n$, we can capture their relative positions to each other by an $n \times n$ matrix R whose elements $R[i, j]$ describe the relationship between state interval i and j. As an example, consider the state sequence in Fig. 1. Obviously state A is always followed by B. And the lag between A and B is covered by state C. Below the state interval sequence both of these patterns are written as a matrix of interval relations. Formally, a temporal pattern P of size n is defined by a pair (s, R), where $s : \{1, .., n\} \to \mathcal{S}$ maps index

i to the corresponding state, and $R \in \mathcal{I}^{n \times n}$ denotes the relationship between $[b_i, f_i)$ and $[b_j, f_j)$[1]. By $\dim(P)$ we denote the dimension (number n of intervals) of the pattern P. If $\dim(P) = k$, we say that P is a k-pattern. Of course, many sets of state intervals map to the same temporal pattern. We say that the set of intervals $\{(b_i, s_i, f_i) \mid 1 \leq i \leq n\}$ is an *instance* of its temporal pattern (s, R). If we remove some states (and the corresponding relationships) from a pattern, we obtain a *subpattern* of the original pattern.

interval sequence:



temporal relations:

	A	B
A	=	b
B	a	=

	A	B	C
A	=	b	o
B	a	=	io
C	io	o	=

(abbreviations: a=after, b=before, o=overlaps, io=is−overlapped−by)

Fig. 1. Example for state interval patterns expressed as temporal relationships.

3 Pattern Discovery

In this section we briefly review the process of pattern discovery and rule generation, which is on a coarse view roughly the same with many kinds of patterns. For a more detailed treatment, see [2, 10].

As already mentioned, we intend to search for frequent temporal patterns. The support of a pattern denotes how often a pattern occurs. Postponing the exact definition of support for the moment, a pattern is called *frequent*, if its support exceeds a threshold supp_{min}. To find all frequent patterns we start in a first database pass with the estimation of the support of every single state (also called candidate 1-patterns). After the kth run, we remove all candidates that have missed the minimum support and create out of the remaining frequent k-patterns a set of candidate $(k+1)$-patterns whose support will be estimated in the next pass. This procedure is repeated until no more frequent patterns can be found. The fact that the support of a pattern is always less than or equal to the support of any of its subpatterns guarantees that we do not miss any frequent patterns.

After having determined all frequent patterns, we can construct rules $X \mapsto Y$ from every pair (X, Y) of frequent patterns where X is a subpattern of Y. If the confidence of the rule $\text{conf}(A \to B) = \frac{\text{supp}(A \to B)}{\text{supp}(A)}$ is greater than a minimal confidence, the rule is printed as an interesting candidate for an important rule.

The space of temporal patterns is even larger than the space of itemsets in association rule mining [2] or episodes in event sequences [10], since for every pair of objects (intervals) we have to maintain a number of possible relationships.

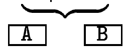
[1] To determine the interval relationships we assume closed intervals $[b_i, f_i]$

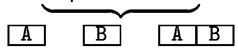
Thus, efficient pruning techniques are a must to overcome the combinatorial explosion of possible patterns. We refer the reader to [8] for the details of the frequent pattern discovery process.

4 Counting Temporal Patterns in State Series

What is a suitable definition of support in the context of temporal patterns? Perhaps the most intuitive definition is the following: The support of a temporal pattern is the number of temporal patterns in the state series. Let us examine this definition in the context of the following example:

$$\text{if}\quad \underbrace{\begin{array}{c|cc} & A & B\\ \hline A & = & b\\ B & a & = \end{array}} \quad\text{then}\quad \underbrace{\begin{array}{c|cccc} & A & B & A & B\\ \hline A & = & b & b & b\\ B & a & = & b & b\\ A & a & a & = & m\\ B & a & a & im & = \end{array}}\quad \text{with probability } p \qquad (1)$$





We call the pattern in the premise the *premise pattern* P and the pattern in the conclusion the *rule pattern* R. The rule pattern also comprises the premise pattern. If we remove the premise from the rule pattern we obtain the *conclusion pattern* C. In (1), the pattern is depicted below the relation matrix (a=after, b=before, m=meets, im=is-met-by).

How often does the pattern in the conclusion occur in the state series in Fig. 2(a)? We can easily find 3 occurrences as shown in Fig. 2(b). The remaining (unused) states do not form a 4^{th} pattern. How often does the premise pattern occur? By pairing states (1,4), (2,6), (3,7), etc. we obtain a total number of 7. So we have $p = \frac{3}{7}$. This may correspond to our intuitive understanding of the rule, but we can improve p to $\frac{4}{7}$ when using the rule pattern assignment in Fig. 2(c). The latter assignment is perhaps less intuitive than the first, because the pattern's extension in time has increased. But now we have a state series that is assembled completely out of rule patterns, there is no superfluous state. Then, would it not be more natural to have a rule probability near 1 instead of $\frac{4}{7}$?

The purpose of the example is to alert the reader that the rule semantics is not that clear as might be expected. Furthermore, determining the maximum number of pattern occurrences is a complex task and does not necessarily correspond to our intuitive counting. We therefore define the total time in which (one or more) instances of the patterns can be observed in a sliding window of width w as the support of the pattern. If we divide the support of a pattern by the length of the state sequence plus the window width w we obtain the relative frequency p of the pattern: If we randomly select a window position we can observe the pattern with probability p.

We note in passing, that the rule probability $p = \frac{3}{7}$ is obtained by using the concept of minimal occurrences [10], as used by Mannila et al. for the discovery of frequent episodes in event sequences. An instance of a pattern P in a time interval

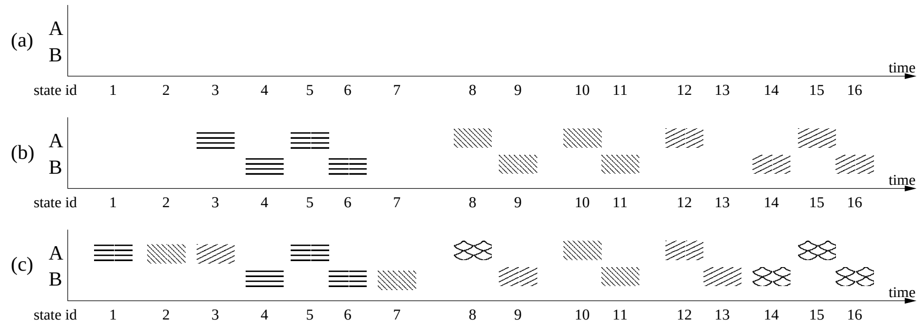


Fig. 2. Counting the occurrences of temporal patterns. (States with different labels (A and B) are drawn on different levels. Note that the pattern of interest (1) requires a *meets* relation in the conclusion.)

$[t_0, t_3]$ is a minimal occurrence, if there is no $[t_1, t_2] \subset [t_0, t_3]$ such that there is also an instance of P within $[t_1, t_2]$. We do not follow this idea, since we consider the rule discovery to be less robust when using minimal occurrences. Consider a pattern "A before B", where the length of the intervals is characteristic for the pattern. If the interval sequence is noisy, that is, there may be additional short B intervals in the gap of the original pattern, the minimal occurrence of A and noisy B would prevent the detection of A and original B. Rule specialization as we will discuss in Sect. 6 would not have a chance to recover the original pattern. Such a situation can easily occur in an automatically generated state sequence which describes the local trend of a time series, where noise in the time series will cause noise in the trend sequence.

5 Rule Evaluation

Let us consider the case when two patterns perfectly correlate in a state sequence. Using again our example rule (1), let us assume that whenever we observe "A before B", we find another two states A and B such that in combination they form the rule pattern in (1). Usually, the support and confidence value of the rule are used to decide about its usefulness [2]. If a sequence consists of rule patterns only, we should expect a confidence value near 1, however, this is not necessarily the case.

5.1 Modified Rule Semantics

There are two possible reasons for a low rule probability (or confidence). The greater the (temporal) extent of the pattern, the lesser the probability of observing the pattern in the sliding window. Consequently, the confidence of a rule decreases as the extent of the rule pattern increases. Secondly, if there are more premise patterns and less rule patterns, rule confidence also decreases. The latter is what we usually associate with rule confidence, whereas the first seems a bit counterintuitive. To reduce the effect of pattern extension, we define a different rule semantics: *Given a randomly selected sliding window that contains an instance of the premise pattern, then with probability p this window overlaps a sliding window that contains the rule pattern.* Loosely speaking, the effect of this redefinition is an increase in the support of the rule, since we substitute "number of windows that contain rule pattern" by "number of windows that contain the premise and overlap a window with a rule pattern".

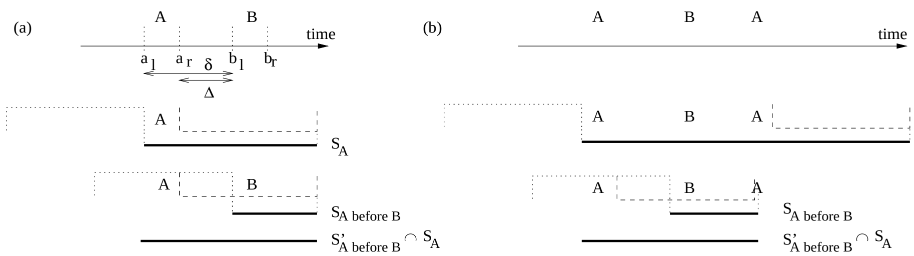


Fig. 3. Support sets of "A" and "A before B", determined by the sliding window positions when the pattern is observed for the first time (dotted window position) and for the last time (dashed window position).

Figure 3(a) illustrates the problem. We consider the premise pattern $P =$"A", the conclusion pattern $C =$"B", and the rule pattern $R =$"A before B", w denotes the window width. For any pattern Q, let S_Q be the support set of a pattern Q, that is, a set of sliding window positions for which Q is observable. Then we have $\text{supp}(Q) = \text{card}(S_Q)$ (cardinality). In the example we have $\text{supp}(P) = \text{card}([a_l, a_r + w])$ and $\text{supp}(C) = \text{card}([b_l, b_r + w])$. Here $\text{supp}(R) = \text{supp}(P \cap C)$ holds and hence $\text{supp}(R) = \text{card}(S_A \cap S_B) = \text{card}([b_l, a_r + w])$. Thus, defining $\Delta := b_l - a_r < w$ and denoting the length of A by l_A, the rule confidence is

$$\text{conf}(A \to A \text{ before } B) = \frac{\text{card}(S_A \cap S_B)}{\text{card}(S_A)} = \frac{a_r + w - b_l}{a_r + w - a_l} = \frac{w - \Delta}{w + l_A}$$

Obviously, as the gap (Δ) between A and B increases, the confidence approaches zero. Now for any pattern Q, let $S'_Q := S_Q \cup \{t - w \,|\, t \in S_P\}$. S'_Q can be interpreted as the support of a pseudo-pattern "pattern Q is visible or will be visible within time w". If we now replace the cardinality of "windows that contain rule patterns" by the cardinality of "windows that contain premise and overlap a window that contains rule patterns" as required by the new semantics, we obtain[2]

$$\text{conf}(A \to A \text{ before } B) = \frac{\text{card}(S_A \cap S'_B)}{\text{card}(S_A)} = \frac{a_r + w - a_l}{a_r + w - a_l} = 1$$

[2] Here we have $S_R = S_A \cap S_B$ and therefore $S'_R \cap S_P = S'_A \cap S'_B \cap S_A = S'_B \cap S_A$.

Thus, as long as we can see A and the beginning of B within the sliding window, we obtain a confidence value of 1, no matter how far A and B are apart. Cases where no conclusion pattern occurs are not affected by this modification (see Fig. 3(b)). Thus, this modification helps to recover the usual semantics of confidence values.

The sets S_P have been determined while searching for frequent patterns anyway [8], they can be handled easily as sorted lists of intervals. Therefore the operations discussed above can be implemented efficiently without looking at the data again: We replace every interval $[l, r] \in S_Q$ by $[l - w, r]$ to obtain S'_Q. In general, rule confidence is then given by

$$\text{conf}(P \to R) = \frac{\text{card}(S_P \cap S'_R)}{\text{card}(S_P)}$$

5.2 Information Content of a Rule

Usually one obtains a large number of frequent patterns and thus a large number of rules. Considerable efforts have been undertaken in the literature to make the vast amount of rules more amenable. We use the J-measure [11] to rank the rule by their information content. It is considered as one of the most promising measures for rule evaluation [5], however, it is still not widely used. Given a rule "if $Y = y$ then $X = x$" on random variables X and Y, the J-measure compares the a priori distribution of X with the a posteriori distribution of X given that $Y = y$. In the context of a rule, we are only interested in two cases, given that $Y = y$, either the rule was right ($X = x$) or not ($X = \bar{x}$), that is, we only consider the distribution of X over $\{x, \bar{x}\}$. Then, the relative information

$$j(X|Y = y) = \sum_{z \in \{x, \bar{x}\}} Pr(X = z|Y = y) \log_2 \left(\frac{Pr(X = z|Y = y)}{Pr(X = z)} \right)$$

yields the *instantaneous* information that $Y = y$ provides about X (j is also known as the Kullbach-Leibler distance or cross-entropy). When applying the rule multiple times, on average we have the information $J(X|Y = y) = Pr(Y = y) \cdot j(X|Y = y)$. The value of J is bounded by ≈ 0.53 bit.

In our context, the random variable Y indicates whether the premise occurred in the sliding window $\mathcal{W}$ or not. The probability $Pr(P \in \mathcal{W})$ when choosing a sliding window position at random is $\text{supp}(y)/T$ where T is the support of the whole sequence. The random variable X indicates whether the rule pattern has occurred. The a priori probability for $R \in \mathcal{W}$ is $\text{supp}(S_R)/T$, the a posteriori probability is given by $\text{supp}(S_R)/\text{supp}(S_P) = \text{conf}(P \to R)$. When using the modified rule semantics, we have to replace S_R by $S'_R \cap S_P$.

5.3 From Rules to Correlations

We have investigated rules $P \to R$ so far, what about $C \to R$ (where C is the conclusion pattern that is determined uniquely by P and R)? If $P \to R$ and

$C \rightarrow R$ hold, then we have a correlation or equivalence $P \leftrightarrow_R C$, that is, the premise is an indication for the conclusion and also vice versa. We can easily extend the rule evaluation to consider correlations. Then, Y denotes a random variable that indicates whether the conclusion has been found in the sliding window (or in a window overlapped by it, if we use modified rule semantics), thus $Pr(C \in \mathcal{W}) = \text{card}(S_C)$ (or $\text{card}(S'_C)$). The random variable X is left unchanged. So we obtain two J-values for $P \rightarrow R$ and $C \rightarrow R$; if one of them is much higher than the other, we can print the rule $P \rightarrow R$ or $C \rightarrow R$, if both values are similar we print $P \leftrightarrow_R C$.

6 Rule Specialisation

The rule evaluation considers only the interval relationships of temporal patterns, but often there is additional information available for each interval. For example, we have not yet evaluated the length of the intervals, or the size of a gap between two states, etc. These lengths are always available when dealing with interval data, but there might be additional information attached to the intervals. For instance, if the intervals denotes ingredients in a chemical process, an additional attribute might denote the intensity or dose of the admixture. A rule that seems interesting to an expert might not have reached the desired confidence value or information content, unless this additional information is incorporated into the rule. For instance, the desired product quality might be achieved only if admixture D has been supplemented to the process at a dose greater than x. In this section we consider the problem of improving rule confidence using such additional state information.

Given a rule $P \rightarrow R$ with temporal patterns P and R and a real-valued attribute a attached to one of the states used in R. Besides some notational differences, we do not make a distinction between attributes of states that occur in the premise (e.g. "if $A \wedge \text{length}(A) < 3$ then A before B") or in the conclusion (e.g. "if A then A before $B \wedge \text{length}(B) > 1$"). Potentially it is possible to improve the information content of a rule in both cases. For notational convenience, however, let us consider the first case, where we examine an attribute of a state in the premise. Now, we run once through the database and store for each instance i of the rule pattern a triple $(a_i, I_i^{(P)}, I_i^{(R)})$, where a_i denotes the value of the attribute, $I_i^{(P)}$ the support interval of the premise instance, and $I_i^{(R)}$ the support interval of the rule instance. For all instances i of the premise pattern that cannot be completed to a rule pattern we store $(a_i, I_i^{(P)}, I_i^{(R)} = \emptyset)$. In contrast to the frequent pattern mining process, now we are not satisfied if we know that there is an occurrence of the rule pattern in the sliding window, but this time we are interested in *all* occurrences with all possible state combinations. This is computationally more expensive, therefore only selected rules should be considered (e.g., best 100 rules without specialisation).

Next, we have to find a threshold α such that the J-value of either $P \wedge (a > \alpha) \rightarrow R$ or $P \wedge (a < \alpha) \rightarrow R$ is maximized. This can be accomplished by sweeping α once through $[\min_i a_i, \max_i a_i]$ and calculating the J-value each time. When J

becomes maximal, we have found the α value that yields the most informative rule. Having done this for all available attributes, we specialise the rule with the most informative attribute. Then, we can refine the specialised rule again, or use the bounds on the J-value [11] to stop when no improvement is possible.

Sweeping through the range of possible attribute values is done incrementally. Let us assume that the indices are chosen such that $a_{i+1} > a_i$, that is, we sort by attribute values. Furthermore, without loss of generality we assume that for no i we have $a_{i+1} = a_i$. If there are i and j with $a_i = a_j$, we substitute $(a_i, I_i^{(P)}, I_i^{(R)})$ and $(a_j, I_j^{(P)}, I_j^{(R)})$ by $(a_i, I_j^{(P)} \cup I_i^{(P)}, I_j^{(R)} \cup I_i^{(R)})$. Now, we run once through the indices and set $\alpha = \frac{a_i + a_{i+1}}{2}$. We start with empty sets for the support of the rule pattern S_R and premise pattern S_P. After the incrementation of i and α, we incrementally update S_P to $S_P \cup I_i^{(P)}$ and S_R to $S_R \cup I_i^{(R)}$. Given the support sets S_P and S_R we can now calculate the J-value for this α. If we want to check for correlations rather than just rules, we additionally maintain the support of the conclusion pattern S_C.

7 Example

We have applied our technique to various real data sets (weather data, music) that would require more detailed background information in order to understand the results. Due to the lack of space, we consider only a small artificial example. We have generated a test data set where we have randomly toggled three states A, B, and C at discrete time points in $\{1, 2, ..., 9000\}$ with probability 0.2, yielding a sequence with 2838 states. Whenever we have encountered a situation where only A is active during the sequence generation, we generate with probability 0.3 a 4-pattern A meets B, B before C, and C overlaps a second B instance. The length and gaps in the pattern were chosen randomly out of $\{1, 2, 3\}$. We have executed the pattern discovery ($\text{supp}_{min} = 2\%$) and rule generation process several times, using the old and new rule semantics and different sliding window widths (8,10,12). We consider the artificially embedded pattern and any subpattern consisting of at least 3 states as interesting. As expected, using the old rule semantics the confidence value is not very helpful in finding interesting rules. Most of the top-ranking rules were not interesting. Among the top 10 rules, we have found 1/2/3 interesting patterns for $w = 8/10/12$, they all had 2-3 states in the premise and 1 in the conclusion pattern. The J-measure yields much better results, even when using the old semantics. When using the modified semantics, we obtain higher confidence values and J-values. The top 5 rules rated by J-values were identical, regardless of the window width, among them all 3 possible rules with 4 states. It is interesting to note that the rule $A \rightarrow BCB$ ranges among the top 5 although its confidence value is still clearly below 0.5 (and the best interesting rule has a confidence of 0.96).

In a second dataset, we have created the described pattern whenever the length of the A interval is greater or equal to 5. For this dataset the rule "$AB \rightarrow CB$" obtained conf $= 0.75$ and $J = 0.26$ bit. We have searched for a threshold α to specialize the rule with "length$(A) \geq \alpha$". When comparing the rules with different α values we obtain a single rule where J becomes maximal with the correct value ($\alpha = 5$). The confidence increases to 0.85 and the information content to 0.34 bit. The confidence value for $\alpha = 5$ represents only a local maximum, beyond $\alpha = 8$ confidence increases monotonically with α. The run time for the last example was slightly above 1 minute on a Mobile Pentium II, 64 MB Linux computer (40s pattern discovery (see also [8]); 28s rule evaluation (naive implementation); 2s rule specialization).

8 Conclusion

Compared to static data, the development of attributes over time is much more difficult to understand by a human. Rather than explaining the temporal process as a whole, which is usually very difficult or even impossible, finding local rules or correlations between temporal patterns can help a human to understand interdependencies and to develop a mental model of the data [9]. In this paper, we have discussed means to find out the most promising temporal rules, which have been generated out of a set of frequent patterns in a state sequence [8]. In combination, modified rule semantics, J-measure, and rule specialisation are much better suited to rank informative rules than support and confidence values alone.

References

[1] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *Proc. of the 4th Int. Conf. on Foundations of Data Organizations and Algorithms*, pages 69–84, Chicago, 1993.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In [7], chapter 12, pages 307–328. MIT Press, 1996.

[3] J. F. Allen. Maintaining knowledge about temporal intervals. *Comm. ACM*, 26(11):832–843, 1983.

[4] D. J. Berndt and J. Clifford. Finding patterns in time series: A dynamic programming approach. In [7], chapter 9, pages 229–248. MIT Press, 1996.

[5] M. Berthold and D. J. Hand, editors. *Intelligent Data Analysis.* Springer, 1999.

[6] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proc. of ACM SIGMOD Int. Conf. on Data Management*, May 1994.

[7] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining.* MIT Press, 1996.

[8] F. Höppner. Discovery of temporal patterns – learning rules about the qualitative behaviour of time series. In *Proc. of the 5th Europ. Conf. on Principles of Data Mining and Knowl. Discovery*, number 2168 in LNAI, pages 192–203, Freiburg, Germany, Sept. 2001. Springer.

[9] K. B. Konstantinov and T. Yoshida. Real-time qualitative analysis of the temporal shapes of (bio)process variables. *AIChE Journal: Chemical Engineering Research and Development*, 38(11):1703–1715, Nov. 1992.

[10] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. Technical Report 15, University of Helsinki, Finland, Feb. 1997.

[11] P. Smyth and R. M. Goodman. Rule induction using information theory. In *Knowledge Discovery in Databases*, chapter 9, pages 159–176. MIT Press, 1991.

Published on 4th International Symposium on Intelligent Data Analysis, Lissabon, Portugal, Sept. 2001. Lecture Notes in Computer Science, vol. 2189, pages 123-132, © Springer.