Learning Dependencies in Multivariate Time Series

Frank Höppner[1]

Abstract.

This paper sketches an approach to learn interdependencies between multiple time series. At the beginning the time series are segmented and thereby transformed into sequences of labeled intervals. The labels denote qualitative aspects of the signal in the respective intervals. Then, from the sequence of labeled intervals, we discover rules where premise and conclusion consist of temporal patterns. The temporal patterns are sets of intervals where Allen's interval logic is used to capture their temporal relationships. Rules are specialized with respect to numerical attributes like the length of the intervals or the slope of the signal within the interval. Finally, we obtain rules like "when signal A decreases while signal B increases with slope greather than 2 then signal C will decrease". Since humans use a similar syntax when discussing such aspects, the proposed methodology may support a human in learning dependencies in multivariate time series.

1 INTRODUCTION

Humans prefer reasoning on a more abstract, symbolic level over digging deeply in large tables of numerical values. This is especially true when dealing with multivariate time series that were measured over a long period of time, which is the kind of data that we will consider in this paper. If an expert of the field is available, thanks to her or his background knowledge she or he *knows* the kind of patterns that are important and worth looking for. The occurrence or absence of such patterns is then the basis for further decision making. Typical application areas are computer aided monitoring and control of technical systems and medical diagnosis and surveillance.

Now, let us assume that we have no expert at hand, either because there are rather few of them or we deal with new systems or observations no-one is experienced with. What would we do in such a case? Probably we would scan the readings by eye in order to identify frequently repeating patterns (probably corresponding to normal situations) as well as seldom patterns (probably indicating some abnormal situation). Having found a vocabulary of patterns, next we could try to find dependencies about them, say, a specific pattern in time series *A* preceedes another pattern in time series *B*. Such kind of information can then be analyzed in greater detail to see if there is some causal relationship. The more we get used to the patterns, the more we are able to further differentiate the patterns (e.g. to discriminate situations), that is, we start to look at details like the steepness of the curve or the duration of some features.

In this paper, we propose a way to support a human in this task. It provides assistance in focussing more quickly on the "interesting things" and uses a notion of patterns a human is already familiar with. The methodology can be sketched in three steps. Starting with an initial set of labels or descriptions (addressing qualitative features like "increasing" or "decreasing") we extract segments (intervals in time) in the time series that are conform to these labels (section 2). Thereby the sequences of numerical values will be transformed into sequences of labeled intervals. This more abstract representation has the advantage that we can easily match similar segments at different locations (by their label) and do not have to use computationally expensive methods like dynamic time warping [21, 24]. Next, the interval sequence will be used to induce multivariate temporal patterns, which will be formalized via Allen's interval logic [2]. From the patterns and their frequency we can derive rules about pattern dependencies (section 3). Finally, once we have identified interesting rules, we may want to further specialize the qualitative symbols with quantitative values (section 4) and restart the process.

2 QUALITATIVE DESCRIPTIONS

Starting from numerical time series, the first task is to abstract from the raw signal to a more condensed, qualitative description. The kind of patterns an expert would look for strongly depends on the application area, of course. However, since we have been teached in school to sketch a function by analyzing the zeroes in the first and second derivative, we can find descriptions like "linearly increasing segment" or "convexly decreasing segment" quite often when listening to the explanations of an expert – and this is independent of the domain. Therefore, we want to use the zero-crossings in the first few derivatives to segment the time series into increasing/decreasing and convex/concave parts.

Then, dealing with noise becomes one of the most important problems, because noise introduces many zero-crossings and thus fragments the time series into many tiny segments rather than a few long ones that correspond to the perceptually salient features. But to support a human in the analysis of time series, the extracted features have to be similar to those features a human would identify in the data. Although the abstraction is "only" a preprocessing step, the usefuleness of the subsequent findings strongly depends on the appropriateness of the abstraction and thus the correct handling of noise. To make things even more complicated, our assumption in knowledge discovery (KDD) is that we do not have the necessary knowledge to finetune the parameters of a time series abstraction method nor can we assume that these parameters do not change over time (especially when examining very long time series). This makes noise elemination extremly difficult.

Many different methods have been used for time series abstraction in the literature, for instance, piecewise linear approximation is a very popular and efficient technique [14, 6, 10], because the piecewise linear representation is easy to understand and easy to

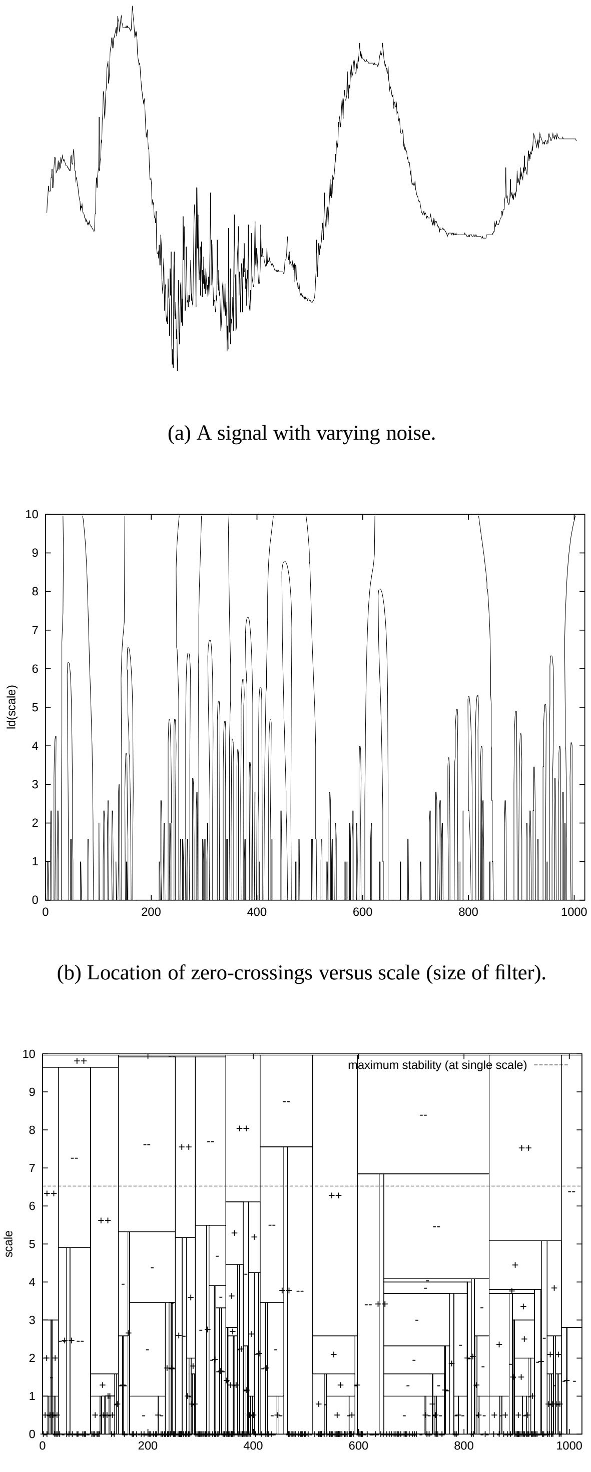
[1] University of Applied Sciences, Dept. of Computer Science, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel, Germany, email: frank.hoeppner@ieee.org This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) under grant no. Kl 648.

process. Local trend information (increasing, steady, decreasing, extrema) may be extracted, and from a sequence of increasing segments we may try to locate the position of inflection points. However, these approximation methods, as well as many other, require either the a priori specification of the number of segments or a maximum error bound for each segment (regardless whether uniform or least-squares approximation is used). Such parameters are difficult to fix a priori or even contradict our assumption that the noise ratio may change over time. Other approaches try to attack the problem of noise by imposing constraints on the smoothness of the approximating curve. But usually there is a number of distinct physical processes that collectively influence the appearance of a signal. Among them there may be system failures that manifest in a similar way as noise does (sharp peaks). Then, imposing smoothness constraints leads to a blurring of such peaks. While it may be useful to have some parameters to finetune the approximation to an experts understanding in some applications [18], here we are in need of a method that does not require such parameters or assumptions.

We use scale-space filtering [25, 16] to derive our symbolic description, a technique that is well-known in image analysis but less frequently used in 1D-signal analysis and almost unknown to the KDD community. Instead of uniform or least-squares approximation, kernel smoothing is applied to cancel noise. The problem of an error threshold selection can be reformulated as a variation of a *scale* parameter, that is, smoothing the signal with a filter of varying size. As before, for different sizes (scales) we get different smoothed signals and thus obtain different qualitative descriptions. Intuitively, the greater the scale gets, the smoother becomes the signal. So we have a scale parameter to fix and thus have not gained any advantage so far. But what if we analyse *multiple* curves at *different* degrees of smoothing *simultaneously*? Recall that we are in need of a procedure that helps us in distinguishing noise from important features in the time series: If we observe the number of smoothing operations that are necessary to remove an extremum, we can conclude about the perceptual salience of the feature: If a feature disappears quickly after little smoothing, it was probably noise, but if it persists against a large number of smoothing operations, it seems to be a perceptually salient feature. This idea has been formulated by Witkin [25], who considers the signal at multiple scales and observes at which scale zero-crossings vanish. From the scale interval during which a feature is observable we can derive a measure of feature stability. Since we are interested in a robust method, we should not take features into account that can be observed in a very limited range of scales only, but should choose those features that survive over a broad range of scales. Witkin has shown empirically, that feature stability and its perceptual salience are related. This stability criterion is therefore well-suited to handle the problem of noise and perform signal abstraction in a way that is close to the human perception.

2.1 Compensating Dislocation

As already indicated, we want to select features that persist over a broad range of scales to build up our symbolic signal description. It is well known, that smoothing dislocates features. (This is not a particular problem of kernel smoothing, but also occurs with uniform or least-squares approximation.) As long as we consider a signal at a single scale only, we may want to ignore these dislocations. However, when comparing signals at different scales, this dislocation may become a problem: Consider a case where the length of a linearly decreasing segment has some discriminative power, and that due to varying noise levels different scales are used to extract such



(a) A signal with varying noise.

(b) Location of zero-crossings versus scale (size of filter).

(c) Interval Tree of Scales.

Figure 1. Multiresolution analysis of a signal.

segments. Then, the start and end point of the segment will be dislocated by different degrees. The dislocation hinders the detection of such dependencies, or makes it impossible in the worst case.

Figure 1(a) shows a time series and figure 1(b) its zero-crossing contours, that is, the location of the zero crossings as the scale parameter varies. If no dislocation would take place, all these lines were perfectly straight. In scale-space filtering, a unified description is produced from the scale-space image by treating each contour as a single physical event rather than a set of unrelated events. This allows us to compensate the temporal distortion of the zero-crossings at coarser scales: we use a coarser representation to identify important features (since they are more robust against smoothing), and use coarse-to-fine tracking to localize the features exactly in the original time series. Having identified the true locations, we can propagate them from the zero scale up to higher scales, thereby turning all contours into straight lines, as shown in figure 1(c).

In our argumentation we assumed that iterative smoothing can only remove zero-crossings but does not introduce new ones: This corresponds to the fact that all lines in figure 1(b) can be traced down to scale 0, that is, a zero-crossing in the original curve. If smoothing would introduce new zero-crossings somewhere, we would obtain lines that do not correspond to features in the original signal. This is not desirable, because we do not want to argue about phenomena that are apparently not present in our data. The assumption we have made corresponds to what we intuitively expect from a smoothing operation, but it is not the guaranteed if we perform kernel smoothing with arbitrary filters. To illustrate that smoothing with some filter coefficients may be counterintuitive, let us consider the example in figure 2, which shows a short time series on the left and the result after applying three different smoothing filters on the right [16]. In the non-constant part of the curve we have 3 extrema in the original curve (min - max - min). For the uniform filter with three coefficients (case a), the number of extrema has increased from 3 to 5 (min - max - min - max - min), whereas other filters (case b and c) behave as we have expected. Result (a) contradicts our intuition: our objective was to make the curve smoother (and thus simpler), but we obtained a curve with even more extrema. It has been shown in [3] that for the continuous case the Gaussian kernel is the only kernel that guarantees that no zero crossings will be introduced in the smoothed signal. For the discrete case, Lindeberg [16] characterizes the *scale-space kernels* that guarantee this property. Only when using such a kernel, we can track down all zero-crossings in an arbitraty scale to a zero-crossing in the original signal, as shown in figure 1(b).

However, we are not interested in the zero-crossings themselves, but the intervals bounded by them. We use the fact that zero-crossings always disappear pairwise (we cannot have two minima without a maximum in between), as it is shown schematically in figure 3. As the scale decreases, the interval between two zero-crossing either persists, or is subdivided by two new zero-crossings into three subintervals. Thus we can construct a tree describing the successive partitioning of the signal into finer subintervals as new zero-crossings appear at finer scales. The resulting (ternary) tree is called *interval tree of scales* [25] and is shown in figure 1(c). The "$+$" and "$-$" signs in the rectangles indicate whether the segment between the zero-crossings represents an increasing or decreasing segment. The rectangles tesselate the time-scale plane completely.

2.2 Stable Features

Since we are interested in the robustness of the extracted features, we want to use the scale-space lifetime over which an time interval persists as a measure of robustness or significance. For the continuous case the scale-space lifetime (stability) is defined by $\log(s_D) - \log(s_A)$, where s_A denotes the scale where the feature appears and s_D where it disappears. (For discrete signals some compensation is required [15].) Once we have a numerical measure for segment stability, we can seek for a single scale which maximizes the mean stability (maximum stability line in figure 1(c), local maximum when starting from coarsest scale). However, once we have done a multiscale analysis we do not have to restrict ourselves to features that appear in a single scale. Witkin proposes to descend the tree from the top to the bottom, as long as the mean lifetime of the offsprings is larger than the lifetime of any of the parents. The latter criterion gives signal descriptions that correspond very well to the human perception of the time series. The stable features according to this criterion are indicated by double signs in the figure ("$++$" and "$--$"). From the interal tree of scale we can see, for instance, that the the peak before the second major hill is perceptually more important than any of the noisy peaks before (tall rectangle near $t = 425$).

Thus, we use the interval tree of scales to convert a numerical time series into a stable symbolic description consisting of labeled intervals. The labels adress increasing/decreasing behaviour if the first

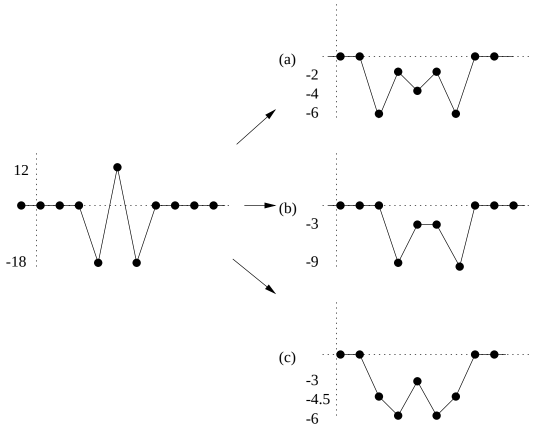


Figure 2. Application of various filters to the signal on the left (filter coefficients from top (a) to bottom (c): $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, $(\frac{1}{2}, \frac{1}{2})$, $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$).

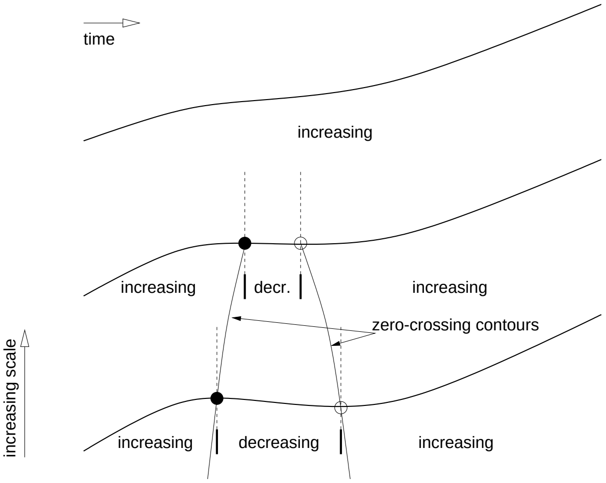


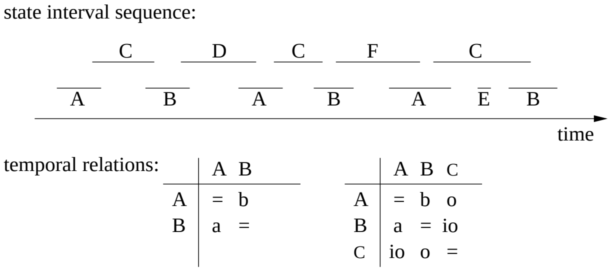
Figure 3. Iterative smoothing (increasing the scale) makes zero-crossings disappear pairwise.

derivative is used, or concave/convex behaviour in case of the second derivative, or both. The description is *stable* in the sense that small changes in the scale parameter or noise level do not change the symbolic description. No thresholds were necessary. The time consuming kernel smoothing may be replaced by efficient wavelet analysis [4], as we will discuss in section 4.

3 INDUCTION OF QUALITATIVE RULES

Most of the popular rule induction algorithms in machine learning (like C4.5, AQ11, etc.) are static, that is, they assume that the variables do not change over time. If consecutive measurements are embedded in a vector, these algorithms can be used to learn rules that reflect temporal dependencies [11, 13]. A certain attribute used in an induced rule may then address the value of a variable 5 minutes ago, for instance. However, temporal processes are often subject to dilatation in time, that is, in similar situations a discriminating peak may occur a bit earlier or later than in the past. If we simply check for a value at a specific point in time, there is no way to cope with such effects. Furthermore, values measured at a single point in time are more sensitive to noise compared to stable, local trend information (signal plus Gaussian noise can easily exceed some threshold found by a rule inducer, however, it is less likely that Gaussian noise turns an increasing segment (of considerable length) into a decreasing one.)

Here, we consider labeled interval sequences as a natural generalization of static attributes to time-varying domains: whenever the attribute changes, a new interval with the appropriate label is introduced. We assume that such a sequence is given or is derived as explained in section 2. There is not much literature about the analysis of such interval sequences, only recently some work has been done [12, 20, 7]. Not surprisingly, since we have a typical knowledge discovery application all these papers are motivated by the discovery of association rules [1, 17]. Among these, [7] deals explicitly with self-similarity in long sequences (observation of one system for a long time), while the others approaches compare many different short sequences (observation of many systems for a short time). There are also differences in the expressiveness of the used temporal patterns.



	A	B
A	=	b
B	a	=

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

Figure 4. Temporal patterns in labelled interval sequences.

Following [7], a temporal pattern is considered as a set of labels, together with a matrix of interval relationships. Figure 4 gives an example of an interval sequence and the patterns "A before B" and "A before B and the gap is covered by C". To be considered interesting, such a pattern has to have a limited temporal extension, that is, it must be observable in a sliding window of a certain width. The underlying assumption is that events have to be neighboured in time in order to affect each other. As the window is sliding along the interval sequence, we denote the set of window positions in which a pattern P was observed by $S \subset \mathbb{R}$. We integrate over the characteristic function χ_S to obtain its support supp(P) ($\chi_S(t) = 1 \Leftrightarrow t \in S$, 0 otherwise, supp$(P) = \int_t \chi_S(t)dt$). Figure 5 shows an example pattern (left) and the part of an interval sequence that can be seen through the sliding window at a fixed position. Is the pattern on the left observable in the window? At first glance, it seems so, because the first three intervals are contained and overlap each other, and the fourth interval has just entered the window. On second thought, however, since we are not yet sure about the relative position of the endpoints of the last two black intervals in the window, the pattern on the left cannot be observed yet: we may have "C overlaps B" or "C contains B" for the last two intervals! As soon as B or C ends we can decide the true relationship and observe the pattern. For further details, see [7, 9].

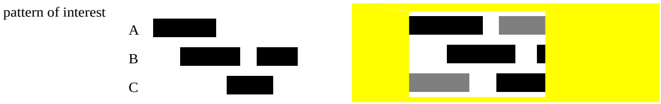


Figure 5. Notion of support.

We say that a pattern Q is a subpattern of P ($Q \sqsubseteq P$) if Q can be obtained from P by removing intervals. Obviously, we have

$$Q \sqsubseteq P \Rightarrow \text{supp}(Q) \geq \text{supp}(P) \qquad (1)$$

For any $Q \sqsubseteq P$ we can write down a temporal rule, for instance, from Q ="A overlaps C" and P ="A before B, A overlaps C, C overlaps B" we obtain a rule: Whenever A overlaps C, there will be a B after A, which is overlapped by C with probability p. The confidence p of this rule is given by conf$(P \rightarrow Q) = \text{supp}(P)/\text{supp}(Q)$. Confidence alone is not a good indicator for valuable rules, for instance, if P and Q occur only once, we obtain a confidence of 1, but of course we would never induce a rule from a single observation. Therefore, we are only interested in those rules that can be observed in at least supp$_{\text{min}}$ percent of the sliding window positions. Such patterns are called *frequent patterns*. For rule induction, we are only interested in rules with supp$(Q) >$ supp$_{\text{min}}$.

Condition (1) is the fundamental requirement to enumerate the space of temporal patterns incrementally in the fashion of association rule mining [1]: Starting from the 1-patterns, consisting only of a single interval, we remove those patterns P that do not reach the minimum support supp$_{\text{min}}$. From (1) we know that, for any k-pattern to be frequent, every subpattern has to be frequent, especially all k subpatterns of size $k-1$. Thus, a set of candidate 2-patterns is build from the frequent 1-patterns, for which the support values are estimated in another scan of the sequence. This procedure is repeated until no more candidates are created (see [1] for details).

When estimating the support of the candidate patterns, we have to check for each candidate and each sliding window position, whether the candidate is currently observable or not. The space of temporal patterns increases faster than 7^k with the number k of intervals in the pattern, therefore efficient pruning techniques are necessary to keep the number of candidates as small as possible. Making use of the fact that the content of the sliding window changes slowly, the number of these tests can be reduced reasonably [7, 9]. (By the way, the high complexity usually found when using the Allen algebra refers to constraint satisfaction problems, where a (partially filled) matrix of temporal relationships is given and a set of intervals that respects these relationships has to be found. Here, we have the reverse situation and start with the intervals.)

For interesting rules, thresholds on minimum support and minimum confidence have to be fulfilled before a rule is induced. Still, one usually obtains a large number of frequent patterns and thus a large number of rules. While originally the confidence value was suggested to rank interesting rules in association rule mining, it has been observed by many authors that confidence is not very well suited for this task. Instead, we use the J-measure [22] to rank rules $P \to Q$ by their information content ($Pr(P \in \mathcal{W})$ is the probability of observing P in an arbitrary sliding window $\mathcal{W}$):

$$\begin{aligned} J(P \to Q) &= Pr(P \in \mathcal{W}) \cdot j(P \to Q) \qquad \text{with} \\ j(P \to Q) &= Pr(Q \in \mathcal{W} | P \in \mathcal{W}) \log_2 \frac{Pr(Q \in \mathcal{W} | P \in \mathcal{W})}{Pr(Q \in \mathcal{W})} \\ &+ Pr(Q \notin \mathcal{W} | P \in \mathcal{W}) \log_2 \frac{Pr(Q \notin \mathcal{W} | P \in \mathcal{W})}{Pr(Q \notin \mathcal{W})} \end{aligned}$$

The j-term is the Kullback-Leibler distance (or relative entropy) of the a priori distribution of the rule pattern Q and the a posteriori distribution of Q given that P has been observed. 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. Using this measure, much better results in rule ranking have been reported in [8].

At the end of the discovery process, we have obtained a set of rules, ranked by their information content. These rules use qualitative attributes of the original time series and their temporal relationships. They provide hints for potential dependencies between different time series (if the symbols in a rule stem from different time series).

4 QUANTITATIVE CONSTRAINTS

Once we have found some potentially interesting qualitative rules, we may want to further specialize the rules in order to obtain a higher confidence value or information content. This can be done in two ways, either we can refine the labels for say increasing segments into linearly increasing, exponentially increasing, logarithmically increasing, etc. From an algorithmic point of view, this does not add any further difficulties to the approach, we can restart the rule induction process after having generated the new intervals. Or we can attribute the labels with numerical values (like slope in case of linear segments). Then, by requiring that the slope of a linearly increasing segment has to be larger or smaller than a certain threshold, the rule may become more discriminating and informative.

Before we consider the problem of rule refinement, let us first consider the problem of estimating the quantitative attributes. As we have discussed in section 2.1, we select the interval representation at different scales, that is, different amount of smoothing has been applied. Any extracted feature will depend on the degree of smoothing, since it determines the degree of dislocation and blurring that has occurred. If we do not compensate these effects, it will be difficult or even impossible to detect the correct quantitative dependencies.

In [4] a solution to this problem has been proposed. There, the iterative smoothing is replaced by more efficient wavelet analysis. Having selected a qualitative signal representation in section 2, we are now interested in signal reconstruction such that the synthesized signal is conform to the qualitative description. The (wavelet) interval tree of scales tessellates the time/scale plane, and can also be seen as a tessellation of the wavelet coefficients. By using only those wavelet coefficients for the reconstruction that correspond to the stable features in our selected representation, only the undesired unstable features are removed from the original signal. It is this reconstruction where we extract slope information from.

Now that we have attributed intervals, we may select a rule and see if additional constraints like "length$(A) < \alpha$" or "slope$(B) > \sigma$" increase the information content of a rule. This can be done in a similar way as rule inducers like C4.5 do this (see e.g. [19]), we just have to replace "the number of training instances" that satisfy a constraint by the "support of the temporal pattern". For a specific rule, we have to collect all instances of the premise and rule pattern and sort these instances by the value of the attribute that is used for specialization. In figure 6 we have four instances (for $\alpha = 1, 3, 4, 7$), collected in decreasing order from top to bottom. The first line shows the support of the premise pattern with $\alpha = 7$. Together with the support of the rule pattern (which is not shown in the figure), the J-value for $\alpha > 5.5$ can be calculated[2] (which is in this case, since we have no instances with $\alpha > 7$, the same as the J-value of $\alpha = 7$). The next value is 4, to calculate the J-value of a rule with condition $\alpha > 3.5$ we unite the support sets we have so far with the support set of all pattern instances with $\alpha = 4$ (line $\alpha = 4$ in figure 6). Again, the J-value is calculated and compared with the J-value we obtained before for $\alpha > 5.5$. Thus, while we sweep once through the possible values of the selected attribute, we incrementally increase the support of the temporal patterns, and for each value we then calculate the J-value of the specialized rule and select the specialization that maximizes the J-value in a greedy fashion. More details can be found in [8].

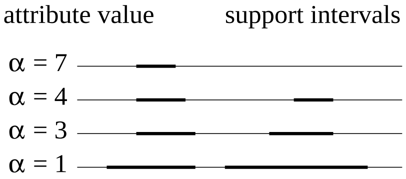


Figure 6. Specializing rules.

The specializations found may then be used to restart the rule discovery process. For instance, if the slope of increasing segments in signal A was of importance for the examined rule, additional intervals "slope$(A) > \sigma$" could be extracted and added to the input sequence of labeled intervals. Since the threshold σ has proven to be helpful, it is not unlikely that new rules will be discovered which also share the same or a similar constraint. This incremental induction of new intervals/labels can be used if no background knowledge about thresholds is available, or if one puts the existing thresholds in question.

5 APPLICATION

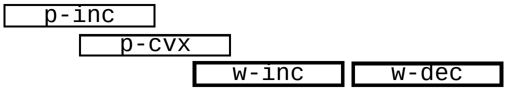
We have applied the sketched method to the analysis of air pressure and wind strength data that has been measured hourly over approximately 7.5 years. This application has been selected because (a) it is well known that local differences in air pressure are the cause for wind, therefore the method should be able to find some relationships between these two variables and (b) such data is readily available. Of course, since weather phenomena have been examined for a very long time, we do not expect to discover new knowledge in this area, but want to show that the discovered knowledge is valid and easily interpretable.

Intervals have been extracted from the scale of maximum mean stability when sweeping from the finest to the coarsest scale (the

[2] 5.5 is the midpoint between $\alpha = 4$ and $\alpha = 7$

first local maximum has been chosen). We have only generated those rules that make predictions into the future. For the following examples, in the pictorial rule presentation we use a prefix `w/p` for windstrength/airpressure, respectively, and a suffix `inc/dec` for increasing/decreasing segments. Intervals that belong to the premise are drawn with thin lines, those that belong to the conclusion with thicker lines.

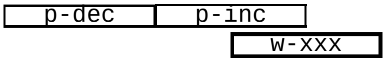
At the end of the rule induction, the top-ranking rules are dominated by rules that refer to a single variable only. This is due to the way in which we have generated the interval sequences, the deterministic alternation between increasing and decreasing segments is reflected by these rules. We select all rules that refer to air pressure in the premise and (at least partially) to the wind strength in the conclusion. For the interpretation of the rules we have to consider the temporal relationships carefully. For instance, we may find a rule `p-inc p-cvx` → `w-inc` as well as a rule `p-inc p-cvx` → `w-dec`. In its static interpretation it seems that the sequence `p-inc p-cvx` is not very well suited to distinguish an increase from an decrease in wind strength. However, from the temporal relationships we will see that the `p-cvx` segment *overlaps* the `w-inc` interval in the conclusion, whereas the `w-dec` segment follows always *after* the `p-cvx` segment:



Given the temporal relationships, the first rule appears more valuable, since the interval in the conclusion is closely connected to those in the premise (via an *overlaps* rather than a *before* relationship).

The `w-inc` variant of the rule has an initial information content of 0.08 bit, which can be increased to 0.12 bit by requiring a curvature[3] ≤ -0.16 and a length of 19 to 46 hours for the convex segment. Therefore, the corresponding air pressure curve has a peak (`p-inc` will be followed by `p-dec`), which is restricted to not being too flat (condition on the curvature). On the average, in the following `w-inc` segment the wind strength will increase by 2.5 ± 1 m/s every hour. We can also find dual rules where `inc/dec/cvx` is replaced by `dec/inc/ccv`, resp.

Similar situations can be expressed by using `p-inc` and `p-dec` labels alone. For instance, a rule



can be found with `w-inc` but also with `w-dec` for `w-xxx`. However, specialization yielded that a segment of increasing windstrength is overlapped by a peak in the air pressure curve if the slope of the `p-inc` segment is greater than 1.07. This is an alternative constraint to avoid flat peaks. The condition on the slope of the segment corresponds pretty well to the thresholds of 1.0 mentioned in [23].

The whole process of interval extraction, rule generation and refinement took a few minutes on an AMD Athlon 1200 MHz computer. Figure 7 shows the number of frequent patterns ($|F_k|$) and candidate patterns ($|C_k|$) for each value of k, together with the runtimes for support estimation (t_{SE}) and candidate generation (t_{CG}). Only 8 labels were used for the 3 variables and the time series have been converted into approximately 10000 intervals. The window width was 72 hours and minimum support was 1%. The frequent pattern enumeration is linear in the sequence length, whereas the specialization is in the worst case quadratic in the number of pattern instances. This increased complexity is, however, not a major drawback, because specialization is carried out for much fewer rules than the frequent pattern discovery process.

k	$\lvert F_k\rvert$	$\lvert F_k\rvert/\lvert C_k\rvert$	$\lvert C_k\rvert$	t_{SE}	t_{CG}
1	8	100%	8	0.29	0
2	210	56.5%	372	1.34	0.01
3	3127	74.2%	4213	6.58	0.83
4	9561	86.8%	11010	12.97	6.79
5	5343	99.3%	5378	11.64	3.61
6	792	100%	792	4.21	0.52
7	29	100%	29	0.87	0.03
8	0	0%	0	0.43	0
$\sum$	19070	87.4%	2.1%	50.12	

Figure 7. Frequent pattern enumeration for weather data.

6 CONCLUSIONS

We have discussed an inductive approach to learn dependencies between multiple time series in a fashion that is close to the way a human would perform this task. When describing phenomena in time series, humans use a syntax that is similar to our notion of temporal patterns. Therefore the proposed methodology may support a human in learning from temporal data. It requires only very few parameters (in the pattern discovery stage: min. support, min. confidence and window width), whose exact settings seem not to be critical.

There is room for a number of straightforward improvements, for instance, the rules may be examined with respect to optimal interval relationships. Arbitrary relations like "A and B do not intersect" can be composed out of the simple relationships in Allen's interval logic (here: A before B or A after B). Since the support for the basic relationships has already been calculated, an optimal combination can be found in a similar way as discussed for quantitative refinement in section 4.

REFERENCES

[1] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo, 'Fast discovery of association rules', in [5], chapter 12, 307–328, MIT Press, (1996).

[2] James F. Allen, 'Maintaining knowledge about temporal intervals', *Comm. ACM*, **26**(11), 832–843, (1983).

[3] Jean Babaud, Andrew P. Witkin, Michel Baudin, and Richard O. Duda, 'Uniqueness of the Gaussian kernel for scale-space filtering', *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **8**(1), 26–33, (January 1986).

[4] Bhavik R. Bakshi and George Stephanopoulos, 'Reasoning in time: Modelling, analysis, and pattern recognition of temporal process trends', in *Advances in Chemical Engineering*, volume 22, 485–548, Academic Press, Inc., (1995).

[5] *Advances in Knowledge Discovery and Data Mining*, eds., Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, MIT Press, 1996.

[6] Michael T. Goodrich, 'Efficient piecewise-linear function approximation using the uniform metric', in *Proc. of the 10th Symp. on Computational Geometry (SCG)*, pp. 322–331, (1994).

[7] Frank 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, pp. 192–203, Freiburg, Germany, (September 2001). Springer.

[8] Frank Höppner and Frank Klawonn, 'Finding informative rules in interval sequences', in *Proc. of the 4th Int. Symp. on Intelligent Data Analysis*, volume 2189 of *LNCS*, pp. 123–132, Lissabon, Portugal, (September 2001). Springer.

[3] Here, we refer to mean of the second derivative.

[9] Frank Höppner and Frank Klawonn, ‘Learning rules about the development of variables over time’, in *Intelligent Systems: Technology and Applications*, ed., Cornelius T. Leondes, volume IV, chapter 9, 201–228, CRC Press, (2003). To appear.

[10] Hiroshi Imai and Masao Iri, ‘An optimal algorithm for approximating a piecewise linear function’, *Journal of Information Processing*, **9**(3), 159–162, (1986).

[11] Mohammed Waleed Kadous, ‘Learning comprehensible descriptions of multivariate time series’, in *Proc. of the 16th Int. Conf. on Machine Learning*, pp. 454–463, (1999).

[12] Po-Shan Kam and Ada Wai-Chee Fu, ‘Discovering temporal patterns for interval-based events’, in *Proc. of the 2nd Int. Conf. on Data Warehousing and Knowl. Discovery*, volume 1874 of *LNCS*, pp. 317–326. Springer, (2000).

[13] Kamran Karimi and Howard J. Hamilton, ‘Finding temporal relations: Causal bayesian networks vs. C4.5’, in *Proc. of the 12th Int. Symp. on Methodologies for Intelligent Systems*, pp. 266–273, Charlotte, NC, USA, (2000).

[14] Eamonn J. Keogh and Padhraic Smyth, ‘A probabilistic approach to fast pattern matching in time series databases’, in *Proc. of the 3rd Int. Conf. on Knowl. Discovery and Data Mining*, pp. 20–24, (1997).

[15] Tony Lindeberg, ‘Effective scale: A natural unit for measuring scale-space lifetime’, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **15**(10), 1068–1074, (October 1993).

[16] Tony Lindeberg, *Scale-Space Theory in Computer Vision*, Int. Series in Engineering and Computer Science, Robotics: Vision, Manipulation and Sensors, Kluwer Academic Publishers, Dordrecht, 1994.

[17] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo, ‘Discovery of frequent episodes in event sequences’, Technical Report 15, University of Helsinki, Finland, (February 1997).

[18] Silvia Miksch, Andreas Seyfang, Werner Horn, and Christian Popow, ‘Abstracting steady qualitative descriptions over time from noisy, high-frequency data’, in *Proc. of the Joint Europ. Conf. on Artificial Intelligence in Medicine and Medical Decision Making*, number 1620 in LNAI, pp. 281–290. Springer, (1999).

[19] Tom M. Mitchell, *Machine Learning*, McGraw Hill, 1997.

[20] Chris P. Rainsford and John F. Roddick, ‘Adding temporal semantics to association rules’, in *Proc. of 10th Europ. Conf. on Machine Learning*, eds., J. Zytkow and J. Rauch, volume 1704 of *LNAI*, pp. 504–509. Springer, (1999).

[21] David Sankoff and Joseph B. Kruskal, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison Wesley, 1983.

[22] Padhraic Smyth and Rodney M. Goodman, ‘Rule induction using information theory’, in *Knowledge Discovery in Databases*, chapter 9, 159–176, MIT Press, (1991).

[23] Sprecher Energie, Linz, *MeteoLiner Kurz-Information*, 1990. User manual for electronic barometer.

[24] H. J. L. M. Vullings, M. H. G. Verhaegen, and H. B. Verbruggen, ‘ECG segmentation using time-warping’, in *Proc. of the 2nd Int. Symp. on Intelligent Data Analysis*, number 1280 in LNCS, pp. 275–285. Springer, (1997).

[25] Andrew P. Witkin, ‘Scale space filtering’, in *Proc. of the 8th Int. Joint Conf. on Artifial Intelligence*, pp. 1019–1022, Karlsruhe, Germany, (1983).

Presented at the workshop on “Knowledge Discovery from Temporal- and Spatio-Temporal Data”, European Conference on Artificial Intelligence (ECAI), Lyon, France, July 2002.