9

Extracting Knowledge from Sensor Signals for Case-Based Reasoning with Longitudinal Time Series Data

P. Funk and N. Xiong

Department of Computer Science and Electronics
Mälardalen University
SE-72123 Västerås, Sweden
{peter.funk, ning.xiong@mdh.se}

Summary. In many industrial and medical diagnosis problems it is essential to investigate time series measurements collected to recognize existing or potential faults/diseases. Today this is usually done manually by humans. However the lengthy and complex nature of signals in practice often makes it a tedious and hard task to analyze and interpret available data properly even by experts with rich experiences. The incorporation of intelligent data analysis method such as case-based reasoning is showing strong benefit in offering decision support to technicians and clinicians for more reliable and efficient judgments.

This chapter addresses a general framework enabling more compact and efficient representation of practical time series cases capturing the most important characteristics while ignoring irrelevant trivialities. Our aim is to extract a set of qualitative, interpretable features from original, and usually real-valued time series data. These features should on one hand convey significant information to human experts enabling potential discoveries/findings and on the other hand facilitate much simplified case indexing and similarity matching in case-based reasoning. The road map to achieve this goal consists of two subsequent stages. In the first stage it is tasked to transform the time series of real numbers into a symbolic series by temporal abstraction or symbolic approximation. A few different methods are available at this stage and they are introduced in this chapter. Then in the second stage we use knowledge discovery method to identify key sequences from the transformed symbolic series in terms of their cooccurrences with certain classes. Such key sequences are valuable in providing concise and important features to characterize dynamic properties of the original time series signals. Four alternative ways to index time series cases using discovered key sequences are discussed in this chapter.

9.1 Introduction

Case-based reasoning (CBR) [1] has been widely recognized as a powerful learning methodology for circumstances where generalized domain knowledge is not available or hard to obtain. Based on the tenet that similar problems

P. Funk and N. Xiong: *Extracting Knowledge from Sensor Signals for Case-Based Reasoning with Longitudinal Time Series Data*, Studies in Computational Intelligence (SCI) **73**, 247–284 (2008)
www.springerlink.com

have similar solutions, CBR attempts to solve new problems by retrieving previous similar cases for which solutions are already known. Usually condition parts of cases are represented as vectors of selected attribute values when making similarity matching between a query case and previous ones in the case base. Proper case index capturing truly relevant features has shown to be one of the crucial factors for the success of case retrieval.

Tackling time series cases is attaining increasing importance in applying case-based reasoning to various real-world problems. As long as processes in the underlying domain are inherently dynamic, cases should be constructed to reflect the phenomena that were evolving overtime rather than be depicted as snapshots at a given time instant. Unlike static cases described by time independent attributes, time series cases contain profiles of time-varying variables wherein pieces of data are associated with a times tamp and are valid only for a specific interval in the case duration. Temporal aspect of time series data has to be taken into account in the tasks of case indexing and case retrieval. Abstraction and representation of temporal knowledge in CBR systems were discussed in [7, 22, 42].

Signal analysis techniques have been applied to extract relevant features from time series signals such as sequential sensor readings. The most common methods used in applications are discrete Fourier transform (DFT) and wavelet analysis, see [9,35,36,51]. Both aims to capture significant characteristics of original signals by providing frequency related information. However, as noted in [11], such traditional analytical tools are only competent on signals with relatively simple dynamics, they fail to characterize patterns of more complex dynamics such as bifurcations and chaotic oscillations.

Another concern with signal analysis is the large number of coefficients produced during signal transformation such that feature selection is entailed to reduce the number of inputs to build similarity measures for case matching and retrieval. The issue of dimensionality reduction becomes particularly critical when measurements are gathered within a very long time span. Longitudinal time series signals are prevalent in circumstances such as patient monitoring for medical health care or condition-based maintenance of industrial equipments, where subjects monitored are expected to possibly change their behavior patterns during the long period of observation. Later in Sect. 9.2, we shall show that, using traditional signal processing methods, it is hard to acquire a moderate number of features as concise representation of original signals while retaining their time-varying properties.

This chapter suggests a general framework fostering compact and efficient representation of lengthy time series cases capturing important temporal behaviors while ignoring irrelevant trivialities. The aim is to extract a set of qualitative, interpretable features from original and usually real-valued time series data. These features should on one hand convey significant information to human experts enabling potential discoveries/findings and on the other hand facilitate much simplified case indexing and similarity matching in case-based reasoning. The road map to achieve this goal consists of two subsequent stages. In the first stage the time series of real numbers is transformed into a symbolic series by temporal abstraction or symbolic approximation. A few different methods are available to be utilized at this stage. Then, in the second stage, we use knowledge discovery method to identify key sequences from the transformed symbolic series in terms of their cooccurrences with certain classes. Such key sequences are valuable in providing important features to characterize dynamic properties of sensor signals, thus leading to concise index of longitudinal time series as well as reduced input dimensionality of similarity measures.

The remainder of this chapter is organized as follows. Section 9.2 gives a brief overview and outlines our general framework to handle longitudinal time series for efficient and compact case index. Approaches to transforming series of sensor measurements into symbolic ones are introduced in Sect. 9.3, followed by a knowledge discovery method presented in Sect. 9.4 to identify key sequences from symbolic series transformed. Subsequently, in Sect. 9.5, we discuss the utilities of key sequences discovered in case-based reasoning, e.g., case indexing, measures for similarity. Relevance to related works is discussed in Sect. 9.6. Finally Sect. 9.7 ends this chapter with concluding remarks.

9.2 Classification Based on Sensor Signals

Categories of subjects can be recognized by observing their relevant variables during their operation. Using sensor technology it is possible to measure the values of such variables and also record the profiles of their evolution with the time. We can then process and analyze the collected sensor recordings to find out hidden symptoms. These symptoms give us basis to reason about the class the subject belongs to or make prediction about a potential failure, that it is likely to emerge in the near future. A general road map for this purpose is illustrated in Fig. 9.1, which includes signal filtering, feature extraction, and pattern classifier as its functional components.

Signal filtering is used to purify original sensor readings by removing noises contained in the signals such that more reliable classification results will be

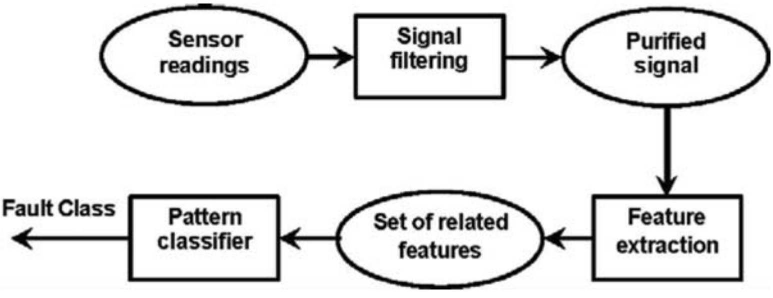

Fig. 9.1. Classification based on sensor signals

warranted. Usually there are two kinds of noises included in the received signals; one is measurement noise due to intrinsic imprecision of sensors and the other is external noise caused by disturbance from surroundings and which is added to the sensor data recorded. Signal recovery from external background noise has been achieved by applying signal processing methods like wavelet analysis and time domain averaging [29,30]. Reduction of measurement errors is outside the scope of this chapter, but interested readers can refer to sensor fusion systems in which Bayesian-based filtering approaches such as Kalman filtering [4] and particle filtering [15] merit to be used to acquire more accurate estimates of states for subjects.

Feature extraction is purported to identify characteristics of sensor signals as useful symptoms for further analysis. This stage is crucial in many industrial and medical domains where the process in consideration is dynamic such that measured variables generally change with the time. This means that it is not possible to depict sensor observations with static single values. Instead we need to identify a collection of features to characterize the evolution of a time-varying variable. The set of extracted features is desired to have a moderate size to reduce the input dimensionality for the pattern classifier (Fig. 9.1). On the other hand, features extracted also ought to be adequate to accommodate temporal information or transitional patterns of signals to be analyzed.

Regarding pattern classifier a number of methods might be considered. Expert systems were developed in support of gathering, representing, and utilizing human expert knowledge for problem solving but they suffer from the knowledge acquisition bottleneck. Regression functions distinguish objects by defining linear boundaries between classes using a moderate number of attributes as function variables. For problems with nonlinear boundaries artificial neural networks would be a suitable approach because they are capable of realizing arbitrary nonlinear mappings between input and output units. Nevertheless the functions of neural networks are rather like a black box, they hardly provide any reasons and arguments for decisions recommended by them. Comparatively, CBR is more transparent by making decisions according to similar cases retrieved such that human users are given reference information to understand, verify, and occasionally also modify the suggested results. The explanatory issue is quite important in many medical and industrial applications where AI systems serve as decision support and every decision made has to be well justified before taking into effect. This motivates us to adopt the methodology of CBR to classify time series signals and we narrow down to case-based classifier in the remaining of this chapter.

9.2.1 Conventional Methods for Feature Extraction

As mentioned before, the measurements from a dynamic system constitute time-varying data streams that are not suitable for immediate usage. Hence we need to “dig out” representative features hidden in the signals prior to classification. The features extracted are delivered to the case-based classifier as an index of the query case. Currently features extracted with traditional methods fall into two categories, namely statistical features and frequency-based features.

Statistical features are extracted from the profile of signal values with respect to calculated statistics as overall generalization. Typical features of this kind can be peak value, start time, overshoot, rising time, mean value, integral, standard deviation, etc. In practice what features to derive for case indexing is commonly ad hoc and domain dependent. An example of using statistical features for case-based circuit diagnosis was illustrated in [39]. However extracting statistical features has a weakness of converting dynamic data streams into static values, thus losing information of temporal relation between data.

Frequency-based features characterize sensor signals by groups of quantities related to a diversity of frequencies. As numerous signal transforms are available to yield frequency spectra, we seem to have more solid basis for extracting features based on frequency than for deriving features based on statistics. The two most common signal transform methods to this end are Fourier transform and wavelet analysis. We shall introduce them briefly in the following and also indicate their limitations facing longitudinal signals with substantial variations.

The Discrete Fourier Transform (DFT)

The DFT transforms a series of sampled values from a signal into spectral information about the signal. Let $x^*(t) = x(nT) = x(n)$ be a sampled function taking samples at times t = 0, T, ..., nT, ..., (N − 1)T, and T is the sampling period (the time between two consecutive samples). The components of DFT for the sampled signal x*(t) are given by a complex summation [47] as follows

$$X(k) = \sum_{n=0}^{N-1} x(nT) \exp\left[-jknT2\pi/(NT)\right] = \sum_{n=0}^{N-1} x(n) \exp\left[-jkn\Omega_0\right] \qquad (9.1)$$

where

$$k = 0, 1, 2, N-1 \qquad \text{and} \qquad \Omega_0 = 2\pi/N$$

If we substitute $\exp\left[-jkn\Omega_0\right]$ in (9.1) with Euler identity, the DFT components can be equivalently written as

$$X(k) = \sum_{n=0}^{N-1} x(n)\cos(kn\Omega_0) - j\sum_{n=0}^{N-1} x(n)\sin(kn\Omega_0) \qquad (9.2)$$

Note that X(k) shows the signal characteristics at the frequency k/(N-1)T. Thus the frequency spacing between two neighboring components in the DFT spectrum is

$$\nabla f = \frac{1}{(N-1)T} Hz \tag{9.3}$$

In general, the DFT component X(k) is a complex consisting of a real and an imaginary part. We adopt the magnitude of X(k)

$$|X(k)| = \sqrt{\left(\sum_{n=0}^{N-1} x(n)\cos(kn\Omega_0)\right)^2 + \left(\sum_{n=0}^{N-1} x(n)\sin(kn\Omega_0)\right)^2} \tag{9.4}$$

as the feature value corresponding to the frequency k/(N-1)T when doing Fourier transforms for feature extraction.

Two limitations of using DFT for feature extraction have to be pointed out here. First, it is clear that, for every k = 0, 1, ...N − 1, there is a DFT term X(k) such that the number of features equals the number of sampled data N. This would lead to an explosion in the number of features extracted when dealing with longitudinal signals common in practice. The sampling period of a signal must be kept below an upper bound according the Shanon theorem in order to avoid distortion of the original signal.

The second limitation with DFT is that the features extracted from the magnitudes of X(k) cannot guarantee to distinguish different orders of patterns within a signal. To show this point more clearly, let us consider two sampled signals with six sampling periods as follows:

$$x_1^*(t) = \left[\underbrace{3,\ 3,\ 3,}_{\text{mode } A}\ |\ \underbrace{5,\ 5,\ 5}_{\text{mode } B}\right]$$

$$x_2^*(t) = \left[\underbrace{5,\ 5,\ 5,}_{\text{mode } B}\ |\ \underbrace{3,\ 3,\ 3}_{\text{mode } A}\right]$$

Obviously the two signals differ only in the order of appearances of values (x_1^* takes the value of 3 in the first three sampling instances followed by 5 later, whereas x_2^* appears in the opposite order). The DFT terms of these two signals are calculated according to (9.2) with the results given by

$$X_1(0) = 24, \quad X_1(1) = -2.000 + 3.4641j\,, \quad X_1(2) = 0$$
$$X_1(3) = -2, \quad X_1(4) = 0, \quad X_1(5) = -2.000 - 3.4641j$$

$$X_2(0) = 24, \quad X_2(1) = 2.000 - 3.4641j\,, \quad X_2(2) = 0$$
$$X_2(3) = 2, \quad X_2(4) = 0, \quad X_2(5) = 2.000 + 3.4641j$$

It is clearly seen from the above that the magnitudes of X_1(k) and X_2(k) are identical for any k = $0, 1,$ 2, ..., 5. As a consequence the signals x_1^* and x_2^* cannot be distinguished by using the features extracted by DFT. This example, although simple, implies a potential problem for signals with varying modes in the whole duration, because temporal information of mode A followed by B or mode B followed by A might be completely indifferent to the DFT features.

Wavelet Transform

Wavelet transform (WT) is a relatively recently introduced signal analysis method [5, 19], which aims to provide a means of analyzing local behavior of signals. In this sense it is fundamentally different from global transforms such as the Fourier transform. The basic principle underlying WT is to represent a signal, x(t), of interest as a weighted sum of wavelets and scaling function by

$$x(t) = A_1\varphi(t) + A_2\psi(t) + \sum_{\substack{n\in+Z\\ m\in Z}} A_{n,m}\psi(2^{-n}t-m) \tag{9.5}$$

where $\psi(t)$ is the mother wavelet function and $\varphi(t)$ denotes the scaling function. Principally any function with positive and negative areas canceling out can be adopted as a wavelet. In other words the only condition imposed on a wavelet function is that it satisfies

$$\int\limits_{-\infty}^{\infty} \psi(t)dt = 0 \tag{9.6}$$

In practice a very frequently used wavelet function is the Haar function which is defined as

$$\psi(t) = \begin{cases} 1 & if \quad 0 \le t < 0.5 \\ -1 & if \quad 0.5 \le t < 1 \\ 0 & \qquad otherwise \end{cases} \tag{9.7}$$

Dilations and translations of the mother wavelet function (9.7) create child wavelets functions as expressed by

$$\psi_{s,l}(t) = 2^{-\frac{s}{2}}\psi(2^{-s}t-l) \tag{9.8}$$

where parameters s and l are integers according to which the mother wavelet function $\psi(t)$ is scaled and dilated. The child wavelets constitute an orthonormal basis of the Haar system. Using this orthonormal basis, time series x can now be formulated as a linear combination of the Haar wavelets:

$$x = x^0 + \sum_{s=1}^{\log_2 N}\sum_{l=0}^{\frac{N}{2^s}-1} c_{s,l}\Psi_{s,l}(t) \tag{9.9}$$

Here N is a power of 2 representing the number of data points in the time series. By x^0 we denote the coarsest approximation of the signal. The coefficients $\mathrm{c}_{\mathrm{s},l}$ are considered as features obtained from wavelet transform. The WT features can be derived by a procedure of averaging and differencing applied on a finite signal, as illustrated in the following example.

Assume a finite time series x = [2, 5, 8, 9, 7, 4, −1, 1]. We now want to express the signal into the form of (9.9) using Haar basis. This can be achieved with three steps below.

Step 1: Perform averaging and differencing at the level corresponding to s = 1 such that

$$x = [2+5, 8+9, 7+4, -1+1, 2-5, 8-9, 7-4, -1-1]/$$
$$\sqrt{2} = [7, 17, 11, 0, -3, -1, 3, -2]\sqrt{2}$$

Here we obtain the WT features in the highest frequency subband

$$WF(1) = [c_{1,0}, c_{1,1}, c_{1,2}, c_{1,3}] = [-3, -1, 3, -2]/\sqrt{2}$$

, which reflects the changing rates of the signal within every two sampling periods in the time dimension.

Step 2: Perform averaging and differencing at the level corresponding to s=2 such that

$$x = \left[\frac{7+17}{\sqrt{2}}, \frac{11+0}{\sqrt{2}}, \frac{7-17}{\sqrt{2}}, \frac{11-0}{\sqrt{2}}, -3, -1, 3, -2\right]$$
$$\Big/\sqrt{2} = \left[\frac{24}{\sqrt{2}}, \frac{11}{\sqrt{2}}, \frac{-10}{\sqrt{2}}, \frac{11}{\sqrt{2}}, -3, -1, 3, -2\right]\Big/\sqrt{2}$$

Here we obtain the WT features in the medium frequency subband

$$WF(2) = [c_{2,0}, c_{2,1}] = \left[\frac{-10}{\sqrt{2}}, \frac{11}{\sqrt{2}}\right]\Big/\sqrt{2} = [-5.00, 5.50]$$

, which reflects the changing rates of the signal within every four sampling periods in the time dimension.

Step 3: Perform averaging and differencing at the level corresponding to s=3 such that

$$\mathrm{x} = \left[\frac{24+11}{(\sqrt{2})^2}, \frac{24-11}{(\sqrt{2})^2}, \frac{-10}{\sqrt{2}}, \frac{11}{\sqrt{2}}, -3, -1, 3, -2\right]$$
$$\Big/\sqrt{2} = \left[\frac{35}{2}, \frac{13}{2}, \frac{-10}{\sqrt{2}}, \frac{11}{\sqrt{2}}, -3, -1,\ 3, -2\right]$$

$$\Big/\sqrt{2} \approx [12.4, 4.60, -5.00, 5.50, -2.12, -0.707, 2.12, -1.41]$$

Here we obtain the WT features in the lowest frequency subband

$$WF(3) = [c_{3,0}] = \left[\frac{13}{2}\right]\Big/\sqrt{2} = [4.60]$$

, which reflects the changing rate of the signal in the whole duration of eight sampling periods.

With the above example we understand that the WT coefficients can be divided into different frequency subbands, each of which reflects how fast the signal increases or decreases its values in the corresponding frequency. The total number of coefficients is equal to $N/2 + N/4 + \Lambda + 1 = N - 1$, which is almost the same as the length of the time series N. In order to get a reduced number of features for case indexing, common practices so far are to choose a dominant coefficient as representative of the subband [36] or to derive statistic values from each frequency subband [50]. Such methods work well with relatively short time series exhibiting simple dynamics. However, considering that WT coefficients themselves constitute a dynamic time series in a frequency subband, how to extract complete and compact information to characterize lengthy, time-varying subbands is still an unresolved issue.

9.2.2 The Proposal of Hybridizing Symbolization and Knowledge Discovery

As was noted in Sect. 9.2.1, traditional methods for feature extraction suffer from some drawbacks, such as undesired large number of features as well as the risk of loss of temporal relationship, when they are applied to complex, longitudinal series of measurements. The reason for this can be attributed to the primary representation of time series based on which features are derived. The data streams utilized are data rich but poor in information content. They only record measurements at every sampling point whereas contain no generalized descriptions of how the data in series evolve with the time. Pure signal processing and mathematical manipulations do not suffice to ensure the derivation of concise and complete dynamic information from primary sampling point-based data records.

The solution we propose in this chapter is to convert the sampling point-based representation of the time series into an interval-based representation. An interval consists of a set of consecutive sampling points and thus encompasses multiple sampling periods in the time dimension. Then data within an interval have to be generalized and aggregated into one symbolic value; the symbolization is conducted via discretization of the range for possible values of the signal. By doing this, the primary time series is now transformed into a symbolic series associated with intervals. Symbolization of primary numerical (usually real valued) time series signals brings the following merits:

1. Symbolic series are shorter in length and more intensive in information content (every symbol is considered as a generalization of the signal behavior in the associated interval), while much of the important temporal information is still retained.
2. Symbolic series facilitate higher computational efficiency; require less computational resource and memory space.

3. Symbolic data are more robust, less sensitive to measurement noises, and also enhance human understanding.
4. With symbolic time series data, it is relatively easier to apply data mining and knowledge discovery methods, algorithms, and tools to find novel, interesting knowledge, and patterns [12], which would in turn help better indexing and characterization of time series cases.

After a numerical signal has been converted into symbolic series, we have to focus on transitions of symbols in it rather than single symbolic values for interpreting and characterizing the series. This is supported by the fact that behaviors in dynamic processes are reflected from transitional patterns over time and occurrences of certain sequences are believed to be significant evidences to identify properties of sequential records. For instance, in medical domains, sequence of symptoms of patients are crucial for diagnosis by physicians, and frequently conditional changes with patients are more important than their static states within single time intervals. Deciding key sequences for case characterization is domain dependent. We need knowledge acquisition and discovery to find knowledge about key sequences when it is not known in advance.

The process of knowledge discovery for key sequences is highlighted in Fig. 9.2. It first entails converting the original database of numerical time series into a database of symbolic ones by means of symbolization. Subsequently the symbolic database with classified series is delivered as input to the knowledge discovery module, which then searches for qualified sequences in the space of all possible sequences. All competent sequences are to be picked up into the library of key sequences as final results.

Once the knowledge about key sequences has been made available, they are utilized as reference to capture important contents in a time series of query. This is shown in Fig. 9.3. The symbolic series transformed from the numerical one is checked thoroughly to detect any occurrences of key sequences stored in

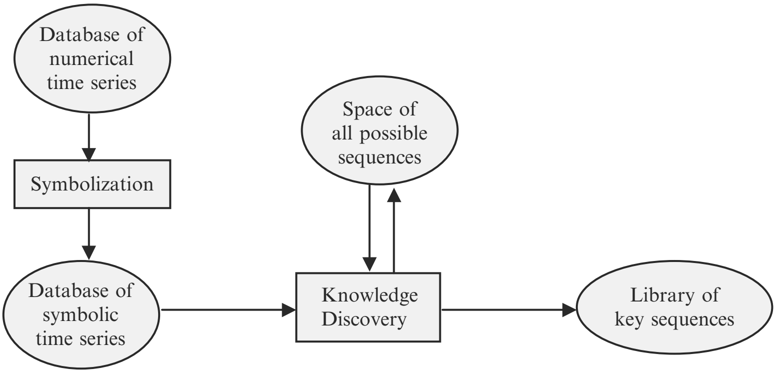
Database of numerical time series
Symbolization
Space of all possible sequences
Database of symbolic time series
Knowledge Discovery
Library of key sequences

Fig. 9.2. Knowledge discovery to find key sequences

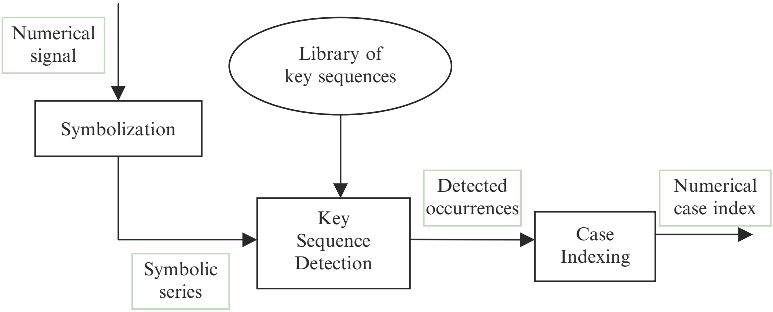

Fig. 9.3. Detection of key sequences in a time series

the key sequence library. Then the information derived about whether a key sequence has occurred and with how many times is made use of for building a numerical case index. This case index is concise since it only considers appearances of key sequences while ignoring other trivial randomness. Various ways to construct such case indexes will be addressed with details in Sect. 9.5.

At this point we can summarize the road map of our suggested solution as

$$Numericaltimeseries \longrightarrow Symbolictimeseries \longrightarrow Numericalcaseindex$$

Symbolization is performed as an intermediate stage to create symbolic expressions of cases which are more abstract and information intensive to facilitate knowledge discovery. Key sequences found by knowledge discovery capture the most significant transitions in a signal to identify its property. Finally case indexing in terms of key sequences creates compact descriptions of cases by quantifying influences of key sequences occurrences into numerical values.

9.3 Transformation into Symbolic Time Series

As noted before, symbolization of original numerical time series is a prestep for performing knowledge discovery to find key sequences. Fortunately many previous studies have shown that it is feasible to do so. Various approaches to making such transformation have been reported in the literature and they will be briefly outlined in this section.

9.3.1 Defining Symbols

Defining appropriate symbols is the first task to conduct during symbolization of signals. It involves partitioning the range of possible values of measurements into a set of regions. Each region corresponds to a specific symbol and each measurement value is thus uniquely mapped into the symbol of the region in which it falls in. The number of regions (symbols) reflects the level of resolution for the information that is retained. A low number of regions implies coarse discrimination of measurement details yet reduced problem space as well as improved efficiency in computation. On the other hand, increased number of regions preserves deeper information details whereas causes higher sensitivity to measurement noise at the same time. There are hence tradeoffs between different criteria to account for when making decisions about the number of regions.

After fixing the number of regions, we have to select locations of these partitions to reach satisfactory results. Sensitivity of the results to the way of locating regions also has to be evaluated. In [10] authors proposed a theoretical approach to choosing optimal locations of partitions for noise-free, deterministic processes. However, selecting theoretically optimal partitions is hardly possible for practical sensor measurements. The reason is that the sensor data are expected to be produced from practical, uncertain processes with hidden dynamics and unknown characteristics of noise, such that a universally strict optimization method does not exist. In practice problem dependent ad hoc techniques are widely employed to determine suitable ways to partition sensor measurements.

In some cases partitioning can be conducted according to the context of the problem provided that the underlying physics betrays a natural choice for granulation. This means the situations where systems in consideration involve dynamics with natural borderlines dividing system states into distinct physical areas. For example, in neurobiological and chemical systems, there is often an excitability threshold above which oscillations will be activated [8,24]. Natural partitioning based on problem context gives a means of accommodating physical knowledge and makes meaningful results easy for human understanding.

In most of other cases traditional methods are to use data mean, midpoint, or median, equal-sized regions, or regions with equal probability to divide the whole range of sensor measurements. In [48] binary symbols corresponding to regions separated by sample median were adopted for reconstruction of dynamics of nonlinear models in light of existing heavy noise. Equal-sized partitions were developed by [18] in dealing with EEG signals. Kim and his colleagues [27] used combinations of sample mean and standard deviations to define regions when analyzing heart-rate dynamics. Finally symbols with equal probability were addressed in [14] by dividing the whole data range into regions in which observation values have identical likelihood to fall.

9.3.2 Symbolic Approximation

The method of symbolic approximation was proposed in [43] with the aim to convert a primary real-valued time series into a condensed symbolic sequence of much shorter length. In doing this the whole duration of the signal is divided into equally sized intervals, i.e., each interval encompasses the same amount of sampling periods. The data in each interval is averaged into a mean value, thus creating a sequence of real numbers summarizing signal behaviors in consecutive time intervals. We term this sequence as PAA (piecewise aggregate approximation) representation of the original signal. Then the PAA sequence is further transformed into a symbolic form by mapping the real numbers in it into corresponding symbols.

To be more concrete, PAA is performed to convert a real-valued time series (primary sensor signal) $C = [c_1, c_2, \ldots, c_n]$ into a shorter sequence $\tilde{C} = [\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_w]\,(w < n)$. The element $\tilde{c}_i$ in $\tilde{C}$ represents the mean value of the data collected during the ith interval, thus it is given by

$$\tilde{c}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j \tag{9.10}$$

where w is the total number of time intervals. PAA creates a substantial data reduction by aggregating signal values within intervals into single values. The PAA sequence, as an intermediate representation, is visualized in Fig. 9.4 where the original signal is approximated by pieces of horizontal segments reflecting the signal's average levels during respective time intervals.

After the PAA sequence $\tilde{C}$ is obtained, it has to be transformed into a symbolic sequence $\hat{C} = [\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_w]$ via symbolization. This entails discretization of the range of signal values into a set of nonoverlapping regions. According to [43] it is desirable to define regions (symbols) with equiprobability [3, 31]. The equal probability of symbols demands that the ordered list of breakpoints $B = \beta_1, \beta_2, \ldots, \beta_{r-1}$ separating regions be defined in such a way that

$$\int_{\beta_i}^{\beta_{i+1}} p(x)dx = \frac{1}{r} \tag{9.11}$$
$$\beta_0 = -\infty, \qquad \beta_r = \infty$$

holds for any $i \in \{0, 1, \ldots, r-1\}$, where r is the number of symbols and by p(x) we denote the probability density function of measured values.

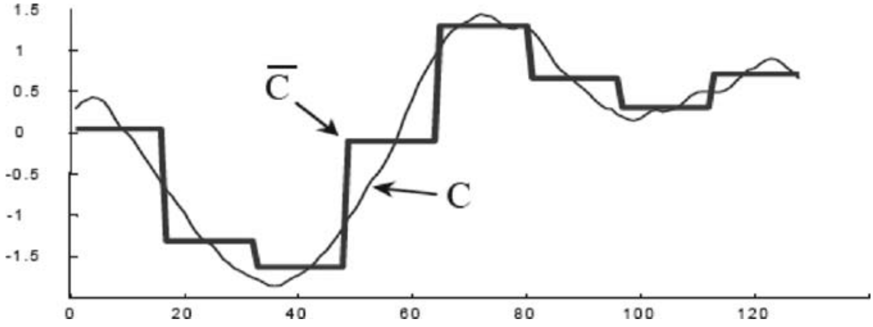

Fig. 9.4. PPA sequence approximating an original signal [43]

Table 9.1. The breakpoints when measurements subject to Gaussian distribution N(0,1)

Symbol number	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9
3	−0.43	0.43							
4	−0.67	0	0.67						
5	−0.84	−0.25	0.25	0.84					
6	−0.97	−0.43	0	0.43	0.97				
7	−1.07	−0.57	−0.18	0.18	0.57	1.07			
8	−1.15	−0.67	−0.32	0	0.32	0.67	1.15		
9	−1.22	−0.76	−0.43	−0.14	0.14	0.43	0.76	1.22	
10	−1.28	−0.84	−0.52	−0.25	0	0.25	0.52	0.84	1.28

Specifically, for sensor measurements subject to the Gaussian distribution N(0,1), the values of these breakpoints are given in Table 9.1 where the number of symbols ranges from 3 to 10.

Once the breakpoints for separation of regions are fixed, the symbolic sequence $\hat{C} = [\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_w]$ can be obtained from the intermediate PAA sequence $\tilde{C} = [\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_w]$ by using the following rule:

$$\hat{c}_i = alpha_j \qquad iff \quad \beta_{j-1} \leq \bar{c}_i \leq \beta_j \tag{9.12}$$

where $alpha_j$ denotes the symbol assigned to the jth region bounded by $\beta_{\mathrm{j}-1}$ and β_{j}.

9.3.3 Temporal Abstraction

Temporal abstraction is an artificial intelligence technique first proposed by [45] to solve data interpretation tasks. The goal is to derive high level generalization from time-stamped representations of time series to evolve toward interval-based representations. Basically this is achieved by aggregating adjacent events exhibiting a common behavior overtime into a generalized concept. Through temporal abstraction, large amounts of temporal data in primary, longitudinal signals can be compressed into compact, abstract, and more meaningful descriptions in the form of series of symbolic values.

Basic temporal abstraction seems sufficient to derive symbolic time series data in the context of this chapter. The ontology for basic temporal abstraction is depicted in Fig. 9.5 which includes state abstraction and trend abstraction. The former focuses on the measured values themselves to extract intervals associated to qualitative concepts such as low, normal, and high, while the latter considers differences between two neighboring records to detect specific patterns like increase, decrease, and stationarity in the series. If differences between consecutive measurements are treated as data for a new

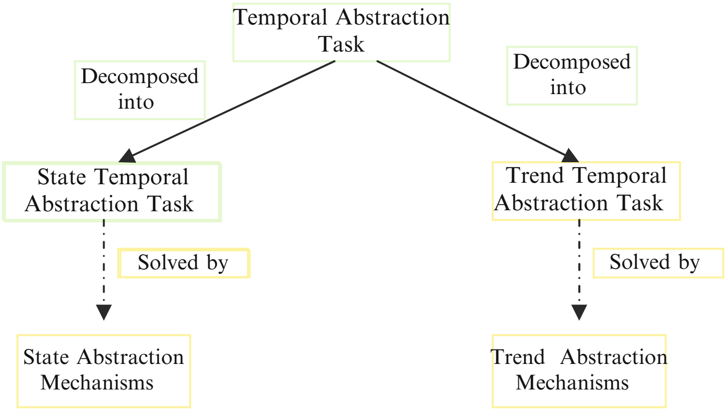
Temporal Abstraction Task
Decomposed into
Decomposed into
State Temporal Abstraction Task
Trend Temporal Abstraction Task
Solved by
Solved by
State Abstraction Mechanisms
Trend Abstraction Mechanisms

Fig. 9.5. Ontology of basic temporal abstraction

series, trend abstraction is equivalent to applying state abstraction to the secondary series of differences derived from the original series of measured values.

The regions associated with qualitative concepts are to be defined beforehand through discretization of the range of measured values for state abstraction or the range of difference values for trend abstraction. The essential thing to do in abstraction is merging adjacent entities falling in the same region into a cluster and summarizing behaviors in this cluster with a concept (symbol) corresponding to the region. Then, arranging concepts of clusters according to the order of their appearances produces a required symbolic series. This new series is compact, abstract, and contains more meaningful information than the primary one.

Temporal abstraction was applied successfully to intelligent analysis of longitudinal data series gathered from monitoring of chronic patients, as reported in [6]. This work, for instance, analyzed and abstracted body temperature profiles in terms of the concepts of low, normal, high, and very high. At the same time the trend for temperature changes were identified as stationarity, increase, or decrease. A simple example given in [6] to illustrate abstractions of body temperatures is shown in Fig. 9.6.

9.3.4 Phase-Based Pattern Identification

In some cases a longitudinal signal consists of a series of phases and every phase has its physical significance to identify its property (pattern) alone. This motivates us to divide the whole signal profile into pieces of subsignals and each of which corresponds to a phase. Since subsignals are assumed to be relatively short and simple, conventional signal processing methods like Fourier or wavelet transforms can be applied to them for extracting features and classifying their patterns. Further, arranging patterns of subsignals in order of their appearances creates a symbolic series as compact and abstract

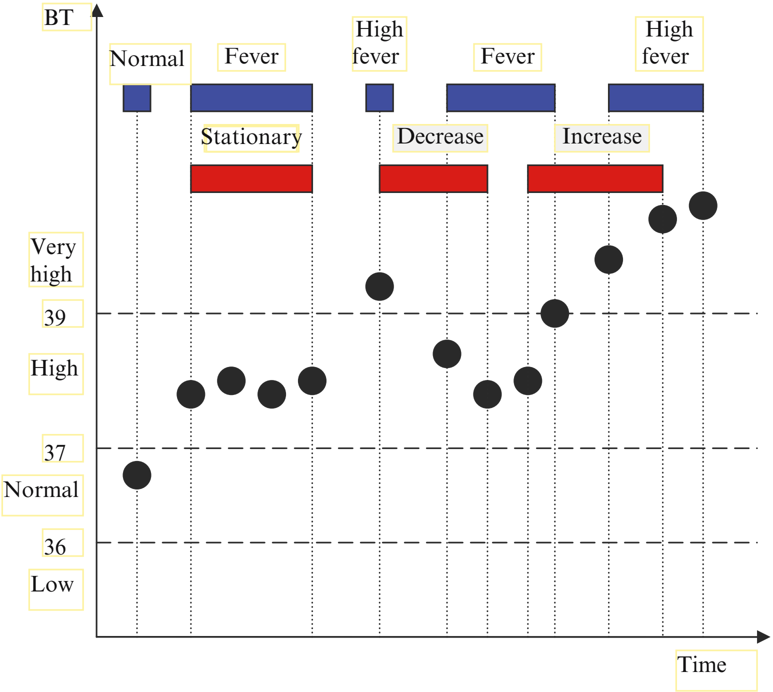

Fig. 9.6. A simple example for temporal abstractions on body temperatures [6]

representation for the overall signal. Each symbol in this series signifies the pattern of a subsignal corresponding to a phase. We refer the method stated above as *phase-based pattern identification.*

This method is used in one of our AI projects related to stress medicine [35], where RSA signals obtained from patients are employed to classify their stress levels. A patient is usually tested through a series of 40–80 breathing cycles (including inhalation and exhalation). Every respiration cycle lasts on average 5–15 s and corresponds to either a normal breathing pattern or one of the dysfunctional patterns. The patterns of breathing (also called RSA patterns) are identified from RSA measurements in the respective respiration periods. Further patterns from consecutive breathing cycles constitute a symbolic time series, which is to be investigated to find information reflecting stress levels of patients.

An overview of the stress medicine project is depicted in Fig. 9.7. First the RSA signal measured during the whole test period is decomposed into a collection of subsignals. By subsignal i in Fig. 9.7 we denote the phase of the signal recorded for the ith breathing cycle. Each subsignal i is delivered to the block "signal classifier," where wavelet analysis and CBR are applied to decide upon its pattern [35]. The identified patterns are then composed into a symbolic series in terms of their appearance order for classifying categories concerning stress levels.

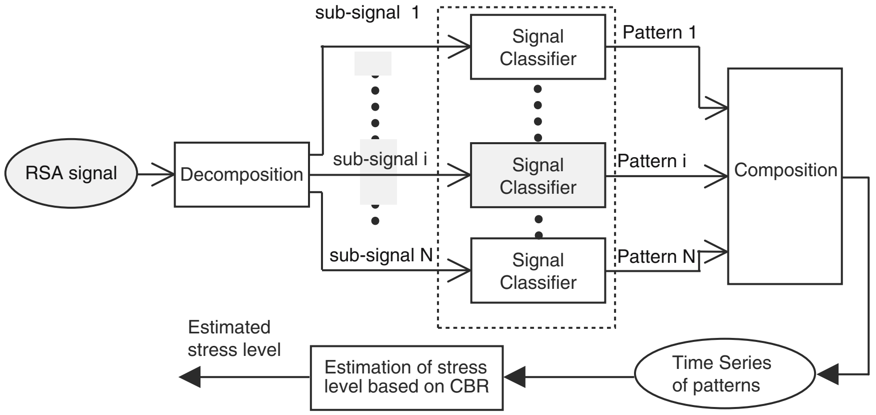

Fig. 9.7. An overview of the stress medicine project

9.4 Knowledge Discovery on Symbolic Time Series Representations

Once a numerical signal has been converted into a symbolic series, it is crucial to get awareness of which parts in the series make sense while others are ignorable. This entails discovering knowledge about key sequences for an underlying domain. Key sequences help better interpret and characterize time series at hand to capture real nature of dynamic systems

Indeed the value of knowledge about key sequences has been made obvious in many application scenarios. For instance, in health monitoring of engineering equipments, original sensor readings can be converted into discrete symbols [41], and some critical changes in series of measurements like swell, sag, impulsive transients, might be signs indicating a present or potential anomaly. In telecommunications, useful information can be obtained from sequences of alarms produced by switches for analysis and prediction of network faults. In defense, sequences of deployments/actions of enemies would possibly betray their tactical intentions. Finally, in a medical scenario, a data sequence of symptoms exhibited on a patient may help to forecast a disease that follows the emerging symptoms.

9.4.1 Problem Statements

To clearly present the proposed knowledge discovery approach, we now give descriptions of the various terms and concepts that are related. We start from formal definitions of symbolic time series, sequences, and time series databases, and then we precisely formulate the problem this section aims to tackle.

Definition 1. *A symbolic time series is a series of symbols signifying events that occurred sequentially overtime, $X = [x(1), x(2), \ldots, x(i), \ldots, x(w)]$, where i indexes an time segment corresponding to symbolic value $x(i)$.*

In a general sense, a symbolic series in Definition 1 can be either a conversion from a numerical signal or a recording of discrete events that happened sequentially in nature. In the remaining of this chapter we refer time series to only symbolic ones given no special notes.

Moreover, every time series has an inherent class. The previous time series data are assumed to have been classified and they are stored in a database together with their associated classes to facilitate data mining. A definition of time series database in the context of classification is given as follows:

Definition 2. *A time series database is a set of pairs $\{(X_i, Z_i)\}_{i=1}^{K}$, where X_i denotes a time series, Z_i the class assigned to X_i, and K is the number of time series cases in the database.*

With a time series database at hand, the data mining process involves analyzing sequences that are included in the database. A sequence of a time series is formally described in Definition 3.

Definition 3. *A sequence S of a time series $X = [x(1), x(2), \ldots, x(w)]$ is a list consisting of elements taken from contiguous positions of X, i.e., $S = [x(k), x(k+1), \ldots, x(k+m-1)]$ with $m \leq w$ and $1 \leq k \leq w - m + 1$.*

Usually there is a very large amount of sequences included in the time series database. But only a part of them that carry useful information for estimating classes are in line with our interest. Such sequences are referred to as indicative sequences and defined in the following:

Definition 4. *A sequence is regarded as indicative given a time series database provided that:*

(1) *It appears in a sufficient amount of time series profiles of the database*
(2) *The discriminating power of it, assessed upon the database, is above a specified threshold*

A measure for discriminating power together with the arguments that lie behind this definition will be elaborated in Sect. 9.4.2. The intuitive explanation is that an indicative sequence is such a one that, on one hand, appears frequently in the database, and on the other hand, exhibits high cooccurrence with a certain class.

Obviously, if a sequence is indicative, another sequence that contains it as subsequence may also be indicative for predicting the class. However, if these both are indicative of the same class, the second sequence is considered as redundant with respect to the first one because it conveys no more information. Redundant sequences can be easily recognized by checking possible inclusion between sequences encountered. The goal here is to find sequences that are not only indicative but also nonredundant and independent of each other.

Having given necessary notions and clarifications we can now formally define our problem to be addressed as follows:

Given a time series database consisting of time series profiles and associated classes, find a set of indicative sequences $\{S_1,\ S_2,\ldots,S_p\}$ *that satisfy the following two criteria*:

(1) For any $i, j \in \{1,\ 2,\ldots,p\}$ *neither* $S_i \subseteq S_j$ *nor* $S_j \subseteq S_i$ *if* S_i *and* S_j *are indicative of a same class*

(2) For any sequence S that is indicative, $S \in \{S_1,\ S_2,\ldots,S_p\}$ *if S is not redundant with respect to* S_j *for any* $j \in \{1, 2, \ldots, p\}$

The first criterion above requests compactness of the set of sequences $\{S_1,\ S_2,\ldots,S_p\}$ in the sense that no sequence in it is redundant by having a subsequence indicative of the same class as it. A sequence that is both indicative and nonredundant is called a key sequence. The second criterion further requires that no single key sequence shall be lost, which signifies a demand for completeness of the set of key sequences to be discovered.

9.4.2 Evaluation of Single Sequences

This section aims to evaluate individual sequences to decide whether one sequence can be regarded as indicative. The main thread is to assess the discriminating power of sequences in terms of their cooccurrence relationship with possible time series classes. In addition we also illustrate the importance of sequence appearing frequencies in the case base for ensuring reliable assessments of the discriminating power.

We assume that given a sequence S there are a set of probable consequent classes $\{C_1,\ C_2,\ldots,C_k\}$. The strength of the cooccurrence between sequence S and class $C_i (i = 1 \ldots k)$ can be measured by the probability, $p(C_i|S)$, of C_i conditioned upon S. Sequence S is considered as discriminative in predicting outcomes as long as it has a strong cooccurrence with either of the possible classes. The discriminating power of S is defined as the maximum of the strengths of its relations with probable consequent classes. Formally this definition of discriminating power PD is expressed as:

$$PD(S) = \max_{i=1\ldots k} P(C_i|S) \tag{9.13}$$

In addition we say that the class yielding the maximum strength of the cooccurrences, i.e., $C = \underset{i=1\ldots k}{\arg} \max P(C_i|S)$, is the class that sequence S is indicative of.

The conditional probabilities in (9.13) can be derived according to the Bayesian theorem as:

$$P(C_i|S) = \frac{P(S|C_i)P(C_i)}{P(S)} \tag{9.14}$$

As the probability $P(S)$ is generally obtainable by

$$P(S) = P(S|C_i)P(C_i) + P(S|\overline{C}_i)P(\overline{C}_i) \tag{9.15}$$

(9.14) for conditional probability assessment can be rewritten as

$$P(C_i|S) = \frac{P(S|C_i)P(C_i)}{P(S|C_i)P(C_i) + P(S|\overline{C}_i)P(\overline{C}_i)} \tag{9.16}$$

Our aim here is to yield the conditional probability $P(C_i|S)$ in terms of (9.16). As $P(C_i)$ is a priori probability of occurrence of C_i which can be acquired from domain knowledge or approximated by experiences with randomly selected samples, the only things that remain to be resolved are the probabilities of S in (time series) cases having class C_i and in cases not belonging to class C_i, respectively. Fortunately such probability values can be easily estimated by resorting to the given case base. For instance we use the appearance frequency of sequence S in class C_i cases as an approximation of $P(S|C_i)$, thus we have:

$$P(S|C_i) \approx \tfrac{N(C_i,S)}{N(C_i)} \tag{9.17}$$

where $N(C_i)$ denotes the number of cases having class C_i in the case base and $N(C_i,\ S)$ is the number of cases having both class C_i and sequence S. Likewise the probability $P(S|\overline{C}_i)$ is approximated by

$$P(S|\overline{C}_i) \approx \tfrac{N(\overline{C}_i,S)}{N(\overline{C}_i)} \tag{9.18}$$

with $N(\overline{C}_i)$ denoting the number of cases not having class C_i and $N(\overline{C}_i, S)$ being the number of cases containing sequence S but not belonging to class C_i.

The denominator in (9.16) has to stay enough above zero to enable reliable probability assessment using the estimates in (9.17) and (9.18). Hence it is crucial to acquire an adequate amount of time series cases containing S in the case base. The more such cases available the more reliably the probability assessment could be derived. For this reason we refer the quantity $N(S) = N(C_i, S) + N(\overline{C}_i, S)$ as evaluation base of sequence S in this chapter.

At this point we realize that two requirements have to be satisfied for believing a sequence to be indicative of a certain class. Firstly the sequence has to possess an adequate evaluation base by appearing in a sufficient amount of time series cases. Obviously a sequence that occurred randomly in few occasions is not convincing and can hardly be deemed significant. Secondly, the conditional probability of that class under the sequence must be dominatingly high, signifying a strong discriminating power. These explain why indicative sequence is defined by the demands on its appearance frequency and discriminating power in Definition 4.

In real applications two minimum thresholds need to be specified for the evaluation base and discriminating power, respectively, to judge sequences as indicative or not. The values of these thresholds are domain dependent and are to be decided by human experts in the related area. The threshold for discriminating power may reflect the minimum probability value that suffices to predict a potential outcome in a specific scenario. The threshold for the evaluation base indicates the minimum amount of samples required to fairly approximate the conditional probabilities of interest. This threshold value can be estimated in terms of the distribution of cases in classes in the case library as well as their prior probabilities. It is shown in the following.

Let δ > 0 be the smallest distance for the denominator in (9.16) to remain sufficiently away from zero, we demand

$$\frac{N(C_i,S)}{N(C_i)}P(C_i)+\frac{N(\overline{C}_i,S)}{N(\overline{C}_i)}P(\overline{C}_i)\geq\delta \tag{9.19}$$

Further the above relation has to hold for every class $\mathrm{C_i}$ to ensure reliable assessments of conditional probabilities for all the classes given sequence S. Next the lower bound for the left side of inequality (9.19) is yielded by

$$\frac{N(C_i,S)}{N(C_i)}P(C)+\frac{N(\overline{C}_i,S)}{N(\overline{C}_i)}P(\overline{C}_i)\geq\frac{N(C_i,S)P(C_i)+N(\overline{C}_i,S)P(\overline{C}_i)}{\max[N(C_i),N(\overline{C}_i)]}$$

$$\geq\frac{[N(C_i,S)+N(\overline{C}_i,S)]\bullet\min[P(C_i),P(\overline{C}_i)]}{\max[N(C_i),N(\overline{C}_i)]}=\frac{\min[P(C_i),P(\overline{C}_i)]}{\max[N(C_i),N(\overline{C}_i)]}N(S) \tag{9.20}$$

Since this lower bound not being less than δ is a sufficient condition for satisfaction of inequality (9.19), we simply impose constraints on the evaluation base N(S) as given by

$$N(S)\geq\frac{\max[N(C_i),N(\overline{C}_i)]}{\min[P(C_i),P(\overline{C}_i)]}\bullet\delta \qquad \forall i \tag{9.21}$$

Herewith it is clearly seen that the threshold value for the evaluation base can be defined as the minimum number of N(S) that satisfies all the constraints in (9.21) for every class $\mathrm{C_i}$. Finally only those sequences that pass thresholds for both discriminating power and evaluation base are evaluated as indicative ones.

9.4.3 Searching for Key Sequences

With the evaluation of sequences being established, we now turn to exploration of qualified sequences in the problem space. The goal is to locate all key sequences that are nonredundant and indicative. We first detail a sequence

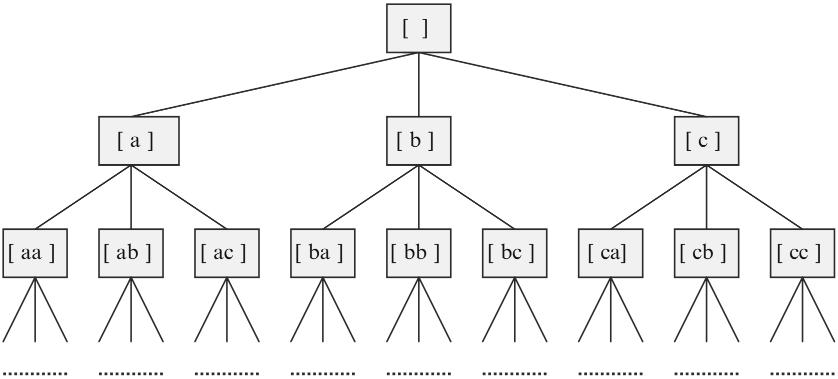

Fig. 9.8. The state space for sequences with three symbols

search algorithm for this purpose in this section and then we demonstrate simulation results on a synthetic case base with the proposed algorithm in Sect. 9.4.4

Discovery of key sequences can be considered as a search problem in a state space in which each state represents a sequence of symbols. Connection between two states signifies an operator between them for transition, i.e., addition or removal of a single symbol in time sequences. The state space for a scenario with three symbols *a, b, c* is illustrated in Fig. 9.8, where an arc connects two states if one can be created by extending the sequence of the other with a following symbol.

A systematic exploration in the state space is entailed for finding a complete set of key sequences. We start from a null sequence and generate new sequences by adding a single symbol to parent nodes for expansion. The child sequences are evaluated according to evaluation bases and discriminating powers. The results of evaluation determine the way to treat each child node in one of the following three situations:

(a) If the evaluation base of the sequence is under a threshold required for conveying reliable probability assessment, terminate expansion at this node. The reason is that the child nodes will have even smaller evaluation bases by appearing in fewer cases than their parent node.
(b) If the evaluation base and discriminating power are both above their respective thresholds, do the redundancy checking for the sequence against the list of key sequences already identified. The sequence is redundant if at least one known key sequence constitutes its subsequence while both remaining indicative of the same consequent. Otherwise the sequence is considered nonredundant and hence is stored into the list of key sequences together with the consequent it indicates. After that this node is further expanded with the hope of finding, among its children, qualified sequences that might be indicative of other consequents.

(c) If the evaluation base is above its threshold whereas the discriminating power still not reaching the threshold, continue to expand this node with the hope of finding qualified sequences among its children.

The expansion of nonterminate nodes is proceeded in a level-by-level fashion. A level in the search space consists of nodes for sequences of the same length and only when all nodes at a current level have been visited does the algorithm move on to the next level of sequences having one more symbol. This order of treating nodes is very beneficial for redundancy checking because a redundant sequence will always be encountered later than its subsequences including the key one(s) during the search procedure.

From a general structure, the proposed sequence search algorithm is a little similar to the traditional breadth-first procedure. However, there are still substantial differences between both. The features distinguishing our search algorithm are (1) it does not attempt to expand every node encountered and criteria are established to decide whether exploration needs to be proceeded at any given state and (2) it presumes multiple goals in the search space and thus the search procedure is not terminated when a single key sequence is found. Instead the search continues on other prospective nodes until none of the nodes in the latest level needs to be expanded. A formal description of the proposed search algorithm is given as follows:

Algorithm for finding a complete set of key sequences

1. Initialize the *Open* list with an empty sequence.
2. Initialize the Key_List to be an empty list.
3. Remove the most left node t from the Open list.
4. Generate all child nodes of t
5. For each child node, $C(t)$, of the parent node t
   a) Evaluate $C(t)$ according to its discriminating power and evaluation base;
   b) If the evaluation base and discriminating power are both above their respective thresholds, do the redundancy checking for $C(t)$ against the sequences in the *Key_list*. Store $C(t)$ into the *Key_list* if it is judged as not redundant. Finally put $C(t)$ on the right of the *Open* list.
   c) If the evaluation base of $C(t)$ is above its threshold but the discriminating power is not satisfying, put $C(t)$ on the right of the *Open* list.
6. If the *Open* list is not empty go to step 3, otherwise return the *Key_list* and terminate the search.

Finally it bears mentioning that finding key sequences in our context differs from those [2, 13, 46] in the literature of sequence mining. Usually the goal in sequence mining is merely to find all legal sequential patterns with their frequencies of appearances above a user-specified threshold. Here we have to consider the cause-outcome effect for classification purpose. Only those nonredundant sequences that are not only frequent but also possess strong discriminating power will be selected as the results of search.

Table 9.2. Sequences discovered on a synthetic case base

Sequence discovered	Discriminating power%	Evaluation base	Dominating consequent
[a d c]	76.70	103	Class 1
[b c a]	78.22	101	Class 1
[d e b]	73.39	124	Class 2
[e a e]	83.18	107	Class 3

9.4.4 Simulation Results

To verify the feasibility of the mechanism addressed above we now present the simulation results on a synthetic case base. A case in this case base is depicted by a time series of 20 symbols and one diagnosis class as the outcome. A symbol in a time series belongs to {a, b, c, d, e} and a diagnosis class is either 1, 2, or 3. The four key sequences assumed are [a d c], [b c a], [d e b], and [e a e]. The first two sequences were supposed to have strong cooccurrences with class 1 and the third and fourth exhibit strong cooccurrences with classes 2 and 3, respectively. Each time series in the case base was created in such a way that both sequences [a d c] and [b c a] had a chance of 80% of being reproduced once in the time series cases of class 1 while sequences [d e b] and [e a e] were added into class 2 and class 3 cases, respectively, with a probability of 90%. After stochastic reproduction of these key sequences, the remaining symbols in the time series of all cases were generated randomly. The whole case base consists of 100 instances for each class. Presuming such time series cases to be randomly selected samples from a certain domain, a priori probability of each class is believed to be one-third.

The sequence search algorithm was applied to this case base to find key sequences and potential cooccurrences hidden in the data. The threshold for the discriminating power was set at 70% to ensure an adequate strength of the relationships discovered. We also specified 50 as the threshold of the evaluation base for reliable assessment of probabilities. The sequences found in our test are shown in Table 9.2 below.

As seen from Table 9.2 we detected all the four key sequences previously assumed. They were recognized to potentially cause the respective consequents with the probabilities ranging from 73.39 to 83.18%. These relationships with a degree of uncertainty are due to the many randomly generated symbols in the case base such that any sequence of symbols is more or less probable to appear in time series of any class. But such nondeterministic property is prevalent in many real-world domains.

9.5 Utility of Key Sequences in Case-Based Classification

The key sequences discovered help us better focus on the most important dynamic patterns while ignoring trivial randomness in examining a time series. They are treated as significant features in capturing dynamic system behaviors. Rather than enumerating what happened in every consecutive time segment, we can now more concisely represent a time series case in terms of occurrences of key sequences in it. Let $\{S_1, S_2, \ldots, S_p\}$ be the set of key sequences. We have to search for every $S_i (i = 1 \ldots P)$ in a time series X to detect all possible appearances. Then case index for X can be established according to the results of key sequence detection. In the following four alternate ways to index X based on key sequences are suggested.

9.5.1 Naive Case Index

A naive means of indexing a time series case X is to depict it by a vector of binary numbers each of which corresponds to a key sequence. A number in the vector is unity if the corresponding sequence is detected in X and zero otherwise. This means that, by the naive method, the index of X is given by

$$Id_1(X \mid S_1, \ldots, S_P) = [b_1, b_2, \ldots, b_P] \tag{9.22}$$

where

$$b_i = \begin{cases} 1 \; if & S_i \;\; is \; subsequence \quad of \quad X \\ 0 & otherwise \end{cases} \tag{9.23}$$

This index has the merit of imposing low demand in computation. It also enables the similarity between two cases to be calculated as the proportion of the positions where their indexing vectors have identical values. Suppose two time series cases X_1 and X_2 which are indexed by binary vectors $[b_{11}, \ldots, b_{1P}]$ and $[b_{21}, \ldots, b_{2P}]$, respectively, the similarity between them is simply defined as

$$Sim_1(X_1, X_2) = 1 - \frac{1}{P} \sum\nolimits_{j=1}^{P} |b_{ij} - b_{2j}| \tag{9.24}$$

9.5.2 Case Index Using Sequence Appearance Numbers

With a binary structure the case index in Sect. 9.5.1 carries a little limited content and would be usable only in relatively simple circumstances. A main reason is that the index cannot reflect how many times a key sequence has appeared in a series of consideration. To incorporate that information, an alternate way is to directly employ the numbers of appearances of single key sequences in describing time series cases. By doing this we acquire the second method of indexing time series X by an integer vector as

$$Id_2(X \mid S_1, \ldots, S_P) = [f_1, f_2, \ldots, f_P] \tag{9.25}$$

where f_i denotes the number of occurrences of sequence S_i in series X.

Further, considering the case index in (9.25) as a state vector, we use the cosine matching function [44] as the similarity measure between two time series cases X_1 and X_2. Thus we have

$$Sim_2(X_1, X_2) = \frac{\sum_{j=1}^{P} f_{1j} f_{2j}}{\sqrt{\sum_{j=1}^{P} f_{1j}^2}\sqrt{\sum_{j=1}^{P} f_{2j}^2}} \tag{9.26}$$

with $f_{1\mathrm{j}}$, $f_{2\mathrm{j}}$ denoting the numbers of occurrences of key sequence S_j in X_1 and X_2, respectively.

9.5.3 Case Index in Terms of Discriminating Power

Although the case index in (9.25) can distinguish two cases having a same key sequence but with different numbers of appearances, it still might not be an optimal representation to capture the exact nature of the problem. Recall that the value of a key sequence is conveying a degree of confidence in the sense of discriminating power for predicting a potential consequent, a time series X would be more precisely characterized by the discriminating powers of the appearances of single key sequences. Intuitively two times of occurrences of a key sequence would give a stronger discriminating power than occurring just once, but not twice in the quantity of the strength. From view of this we suggest indexing X as a vector of real numbers, representing discriminating powers for the appearances of single key sequences, as follows:

$$Id_3(X \mid S_1, \ldots, S_P) = [g_1, g_2, \ldots, g_P] \tag{9.27}$$

with

$$g_i = \begin{cases} DP(f_i * S_i) & if \quad f_i \geq 1 \\ 0 & if \quad f_i = 0 \end{cases} \tag{9.28}$$

By $DP(f_i * S_i)$ we denote the discriminating power by sequence S_i appearing f_i times in X.

Let C be the class that the key sequence S_i is indicative of. We define the discriminating power $DP(f_i * S_i)$ as the probability for class C given f_i appearances of sequence S_i. This probability can be obtained by applying the Bayes theorem in a sequential procedure. Assuming a two class problem without loss of generality, this procedure is depicted here by a series of equations as follows:

$$P(C|S_i) = \frac{P(S_i|C)P(C)}{P(S_i|C)P(C) + P(S_i|\overline{C})P(\overline{C})} \tag{9.29}$$

$$P(C|2 * S_i) = \frac{P(S_i|C)P(C|S_i)}{P(S_i|C)P(C|S_i) + P(S_i|\overline{C})P(\overline{C}|S_i)} \tag{9.30}$$

$$P(C|t * S_i) = \frac{P(S_i|C)P\left(C|(t-1) * S_i\right)}{P(S_i|C)P\left(C|(t-1) * S_i\right) + P(S_i|\overline{C})P\left(\overline{C}|(t-1) * S_i\right)} \tag{9.31}$$

$$DP(f_i * S_i) = P(C|f_i * S_i) = \frac{P(S_i|C)P\left(C|(f_i-1)*S_i\right)}{P(S_i|C)P\left(C|(f_i-1)*S_i\right) + P(S_i|\overline{C})P\left(\overline{C}|(f_i-1)*S_i\right)} \tag{9.32}$$

where the probabilities $P(S_i|C)$ and $P(S_i|\overline{C})$ can be estimated according to (9.17) and (9.18), respectively. The probability updated in (9.29) represents the probability for class C given one appearance of S_i, which is further updated in (9.30) by the second appearance of S_i producing a higher probability considering both occurrences. Generally, the probability $P(C|t*S_i)$ is yielded by updating the prior probability $P\left(C|(t-1)*S_i\right)$ with one more occurrence of S_i in (9.31). Finally we obtain the ultimate probability assessment incorporating all appearances, i.e., the required discriminating power, by (9.32).

We now give a concrete example to illustrate how a case index can be built in terms of occurrences of key sequences. Suppose a two class situation in which three key sequences S_1, S_2, and S_3 are discovered. Sequence S_1 appears twice in time series X and S_2 appears once while S_3 is not detected. S_1 and S_2 are both indicative of a certain class C. The a priori probability for class C is 50% and the probabilities of sequences S_1, S_2 in situations of class C and its complementary are shown below:

$$\begin{aligned} P(S_1\,|C\,) &= 0.56 \qquad & P(S_1\,|\overline{C}\,) &= 0.24 \\ P(S_2\,|C\,) &= 0.80 \qquad & P(S_2\,|\overline{C}\,) &= 0.40 \end{aligned}$$

With all the information assumed above, the discriminating powers for the appearances of S_1 and S_2 are calculated in the following:

1. Calculate the probability for C with the first appearance of S_1 by

$$P(C\,|S_1) = \frac{P(S_1\,|C)P(C)}{P(S_1\,|C)P(C) + P(S_1\,|\overline{C})P(\overline{C})} = \frac{0.56\cdot 0.5}{0.56\cdot 0.5 + 0.24\cdot 0.5} = 0.70$$

2. Refine the probability $P(C|S_1)$ with the second appearance of S_1, producing the discriminating power for the appearances of S_1

$$\begin{aligned} DP(2*S_1) = P(C\,|2*S_1) &= \frac{P(S_1\,|C)P(C\,|S_1\,)}{P(S_1\,|C)P(C\,|S_1\,) + P(S_1\,|\overline{C})P(\overline{C}\,|S_1\,)} \\ &= \frac{0.56\cdot 0.70}{0.56\cdot 0.70 + 0.24\cdot 0.30} = 0.8448 \end{aligned}$$

   It is clearly seen here that the power of discrimination is increased from 0.70 to 0.8448 due to the key sequence occurring for the second time.

3. Derive the discriminating power for the occurrence of S_2 by calculating the conditional probability for C upon S_2 as

$$\begin{aligned} DP(1*S_2) = P(C\,|S_2) &= \frac{P(S_2\,|C)P(C)}{P(S_2\,|C)P(C) + P(S_2\,|\overline{C})P(\overline{C})} \\ &= \frac{0.80\cdot 0.50}{0.80\cdot 0.50 + 0.40\cdot 0.50} = 0.6667 \end{aligned}$$

Moreover, because S_3 is not detected in X, there is no discriminating power for it. Hence we construct the index for this time series case as:

$$Id_3(X\,|S_1\;,S_2,S_3) = [0.8448,\; 0.6667,\; 0]$$

Finally, with this case indexing scheme, we use the cosine function again as the similarity measure for case retrieval. So the similarity between two time series X_1 and X_2 is given by

$$Sim_3(X_1,X_2) = \frac{\sum_{j=1}^{P} g_{1j}g_{2j}}{\sqrt{\sum_{j=1}^{P} g_{1j}^2}\sqrt{\sum_{j=1}^{P} g_{2j}^2}} \tag{9.33}$$

where g_{1j} and g_{2j} denote the jth elements in the case indexes (9.27) for X_1 and X_2, respectively.

9.5.4 Case Indexing with Key Sequence Union

In Sect. 9.5.3 cases are indexed according to the discriminating powers of occurrences of single key sequences. Such work could be extended by regarding the key sequences that are indicative of a common class as a collective union. This view motivates us to group occurrences of key sequences in time series X into a set of clusters. For every class C_i there is a cluster V_i corresponding to it. V_i is a collection of events for occurrences of those key sequences that are indicative of class $\mathrm{C_i}$. The discriminating power of cluster $\mathrm{V_i}$ is defined as the probability of class $\mathrm{C_i}$ in light of the events included in the cluster. Hence we write

$$DP(V_i) = \begin{cases} P\left(C_i\,|\{e_j\,|e_j \in V_i\}\right) & if \quad V_i \neq \emptyset \\ 0 & if \quad V_i = \emptyset \end{cases} \tag{9.34}$$

Further, the discriminating powers of clusters of events representing key sequences occurrences are utilized to index a time series case. Hence the index for time series X is given by

$$Id_4(X \mid S_1, \ldots, S_P) = [DP(V_1), DP(V_2), \ldots, DP(V_K)] \tag{9.35}$$

where K denotes the number of classes of interest.

It is clear that the case index in the form of (9.35) is highly concise. It reduces the length of index vector to the number of classes. This is achieved by calculating the discriminating power for a union of key sequences that are consistent. Consequently every component in the vector of (9.35) contains rich information by fusion of occurrences from multiple key sequences. This proposed case index is valuable for further dimensionality reduction particularly under the circumstances when the number of key sequences discovered is still quite large.

Let $V_i = \{e_1, e_2, \Lambda, e_T\}$ be a cluster of events of key sequences occurrences corresponding to class $\mathrm{C_i}$. We now want to obtain the discriminating power of cluster $\mathrm{V_i}$ by calculating the conditional probability $P(C_i|e_1, e_2, \Lambda, e_T)$. This probability is yielded by exploiting the events e_j as evidences for probability updating in separate steps. At every step we use a single event to revise prior probabilities according to the Bayes theorem and these updated probability estimates are then propagated as prior beliefs to the next step. The procedure of probability updating using events in cluster $\mathrm{V_i}$ is depicted by a series of equations as follows:

$$P(C_i|e_1) = \frac{P(e_1|C_i)P(C_i)}{P(e_1|C_i)P(C_i) + P(e_1|\overline{C}_i)P(\overline{C}_i)} \tag{9.36}$$

$$P(C_i|e_1, e_2) = \frac{P(e_2|C_i)P(C_i\,|e_1\,)}{P(e_2|C_i)P(C_i\,|e_1\,) + P(e_2|\overline{C}_i)P(\overline{C}_i\,|e_1\,)} \tag{9.37}$$

$$P(C_i|e_1, \ldots, e_i) = \frac{P(e_i|C_i)P(C_i\,|e_1\,, \ldots, e_{i-1})}{P(e_i|C_i)P(C_i\,|e_1, \ldots, e_{i-1}) + P(e_i|\overline{C}_i)P(\overline{C}_i\,|e_1, \ldots e_i\,)} \tag{9.38}$$

$$p(C_i|e_1, \ldots, e_T) = \frac{P(e_T|C_i)P(C_i\,|e_1\,, \ldots, e_{T-1})}{P(e_T|C_i)P(C_i\,|e_1, \ldots, e_{T-1}) + P(e_T|\overline{C}_i)P(\overline{C}_i\,|e_1, \ldots\ e_{T-1}\,)} \tag{9.39}$$

where the probabilities $P(e_i|C_i)$ and $P(e_i|\overline{C}_i)$ for $i \in \{1, \ldots, T\}$ can be estimated according to (9.17) and (9.18), respectively, as e_i is considered as the occurrence of a sequence. The probability updated in (9.36) represents the probability for class C_i given event e_1, which is further updated in (9.37) by event e_2 producing a more refined belief considering both e_1 and e_2. Generally the probability $P(C|e_1, \ldots, e_i)$ is yielded by updating the prior probability $P(C|e_1, \ldots, e_{i-1})$ with a new event e_i in (9.38). Finally we obtain the ultimate probability assessment incorporating all available events in (9.39).

At this stage one may question the order in which single events from a cluster are used to refine probability assessments. This seems a fundamental issue and involves allocation of events to different steps of a sequential procedure. Fortunately our study has clarified that the order of events used in probability updating is completely indifferent. The final probability value remains constant as long as each piece of event is assigned to a distinct step. The claims as such are formally based on the following theorems.

Lemma 1. *Let $\{e_1, \ldots, e_T\}$ be a cluster of events representing appearances of certain key sequences in a time series X. The probability for class C given the cluster is not affected if two adjacent events exchange their positions in the order of events used for probability refinements. This means that the relation $P(C|e_1, \ldots e_i, e_{i+1}, \ldots, e_T\} = P(C|e_1, \ldots e_{i+1}, e_i, \ldots, e_T\}$ holds for $i \in \{1, \ldots T-1\}$.*

Proof. For proof of the lemma with the statement that $P(C|e_1,\ldots,e_{i-1},e_i, e_{i+1},\ldots,e_T\} = P(C|e_1,\ldots,e_{i-1},e_{i+1},e_i,\ldots,e_T\}$, we only need to establish the relation for $P(C|e_1,\ldots,e_{i-1},e_i,e_{i+1}\} = P(C|e_1,\ldots,e_{i-1},e_{i+1},e_i)$, which is equivalent to the lemma.

We start to consider the probability $P(C|e_1,\ldots e_i,e_{i+1}\}$ which is acquired by updating the prior belief $P(C|e_1,\ldots,e_i)$ with a new evidence e_{i+1}, hence it can be written as

$$P(C|e_1,\ldots,e_i,e_{i+1}) = \frac{P(e_{i+1}|C)P(C|e_1,\ldots,e_i)}{P(e_{i+1}|C)P(C|e_1,\ldots,e_i) + P(e_{i+1}|\overline{C})P(\overline{C}|e_1,\ldots,e_i)} \tag{9.40}$$

Further the probability $P(C|e_1,\ldots,e_i)$ is formulated by taking $P(C|e_1,\ldots,e_{i-1})$ as its prior estimate such that

$$P(C|e_1,\ldots,e_i) = \frac{P(e_i|C)P(C|e_1,\ldots,e_{i-1})}{P(e_i|e_1,\ldots,e_{i-1})} \tag{9.41}$$

Likewise we obtain

$$P(\overline{C}|e_1,\ldots,e_i) = \frac{P(e_i|\overline{C})P(\overline{C}|e_1,\ldots,e_{i-1})}{P(e_i|e_1,\ldots,e_{i-1})} \tag{9.42}$$

Combining (9.41) and (9.42) into (9.40) gives rise to a transformed formulation as

$$\begin{aligned}&P(C|e_1,\ldots,e_i,e_{i+1})\\ &= \frac{P(e_{i+1}|C)P(e_i|C)P(C|e_1,\ldots,e_{i-1})}{P(e_{i+1}|C)P(e_i|C)P(C|e_1,\ldots,e_{i-1}) + P(e_{i+1}|\overline{C})P(e_i|\overline{C})P(\overline{C}|e_1,\ldots,e_{i-1})}\end{aligned} \tag{9.43}$$

Next we express the conditional probabilities $P(e_{i+1}|C)$, $P(e_{i+1}|\overline{C})$, $P(e_i|C)$, $P(e_i|\overline{C})$ with their Bayes forms by

$$P(e_{i+1}|C) = \frac{P(C|e_{i+1})P(e_{i+1})}{P(C)} \tag{9.44}$$

$$P(e_{i+1}|\overline{C}) = \frac{P(\overline{C}|e_{i+1})P(e_{i+1})}{P(\overline{C})} \tag{9.45}$$

$$P(e_i|C) = \frac{P(C|e_i)P(e_i)}{P(C)} \tag{9.46}$$

$$P(e_i|\overline{C}) = \frac{P(\overline{C}|e_i)P(e_i)}{P(\overline{C})} \tag{9.47}$$

where *P(C)* and $P(\overline{C})$ denote the initial probability estimates for class C and its complementary without any events about key sequences appearances. Using the Bayes forms from (9.44) to (9.47), (9.43) is finally rewritten as

$$P(C|e_1,\ldots,e_i,e_{i+1}) \\ = \frac{P^2(\overline{C})P(C|e_{i+1})|P(C|e_i)P(C|e_1,\ldots,e_{i-1})}{P^2(\overline{C})P(C|e_{i+1})|P(C|e_i)P(C|e_1,\ldots,e_{i-1}) + P^2(C)P(\overline{C}|e_{i+1})|P(\overline{C}|e_i)P(\overline{C}|e_1,\ldots,e_{i-1})} \tag{9.48}$$

Clearly we see from (9.48) that the order between e_i and e_{i+1} has no effect at all on the probability $P(C|e_1,\ldots,e_i,e_{i+1})$ assessed. It follows that

$$P(C|e_1,\ldots,e_{i-1},e_i,e_{i+1}) = P(C|e_1,\ldots,e_{i-1},e_{i+1},e_i) \tag{9.49}$$

and here from the lemma is proved.

With the lemma justified by the proof above, we further contemplate the implication of it. This leads to a corollary presented below. □

Corollary 1. *Let $\{e_1,\ldots,e_T\}$ be a cluster of events representing appearances of certain key sequences in a time series X. The probability for X in class C given the cluster is independent of the order according to which single events e_1, $e_{2,\ldots,}$ e_T, are used in probability refinements.*

The proof of Corollary 1 *is obvious. According to the lemma, an element in a given order of events can be moved to an arbitrary position by repeatedly exchanging its position with an adjacent one while not affecting the final probability assessments. As this can be done to every piece of event, we enable transitions to any orders of events without altering the estimated value of the probability.*

This corollary is important in providing theoretic arguments allowing for an arbitrary order of sequences to be used in probability fusion based on the Bayes theorem. The connotation is that when a key sequence occurred in the time series does not matter for the case index. Instead only the numbers of appearances of key sequences affect the likelihoods of classes given respective occurrence clusters, which are included as components in the case index vector.

Now let us study an illustrative example to better understand how the above sequential procedure works in derivation of required probabilities using clusters of events as evidences. Consider a time series X with two probable classes. Suppose that four key sequences S_1, S_2, S_3, *and* S_4 *are detected in X, and S_1, S_2 are indicative of class C while* S_3 *and* S_4 *are indicative of the complementary of C. The a priori probability of class C is 50%, and the probabilities of sequences S_1, S_2,* S_3, *and* S_4 *in situations of class C and its complementary are shown below:*

$$\begin{array}{ll} P(S_1|C) = 0.56 & P(S_1|\overline{C}) = 0.24 \\ P(S_2|C) = 0.80 & P(S_2|\overline{C}) = 0.40 \\ P(S_3|C) = 0.35 & P(S_3|\tilde{C}) = 0.62 \\ P(S_4|C) = 0.18 & P(S_4|\tilde{C}) = 0.30 \end{array}$$

Further we assume that sequence S_1 appears twice in X and S_2, S_3, S_4 *appear once, hence the clusters of key sequence occurrences for X are notated as*

$V_1(X) = \{S_1, S_1, S_2\}$ and $V_2(X) = \{S_3, S_4\}$. With these three occurrences detected, the probability of class C is yielded in the following three steps:

Step A1: Update the a priori probability $P(C)$ with the first appearance of S_1 by

$$P(C|S_1) = \frac{P(S_1|C)P(C)}{P(S_1|C)P(C) + P(S_1|\overline{C})P(\overline{C})} = \frac{0.56 \cdot 0.5}{0.56 \cdot 0.5 + 0.24 \cdot 0.5} = 0.70$$

Step A2: Refine the probability updated in step A1 with the second appearance of S_1, thus we have

$$\begin{aligned} P(C|S_1, S_1) &= \frac{P(S_1|C)P(C|S_1)}{P(S_1|C)P(C|S_1) + P(S_1|\overline{C})P(\overline{C}|S_1)} \\ &= \frac{0.56 \cdot 0.70}{0.56 \cdot 0.70 + 0.24 \cdot 0.30} = 0.8448 \end{aligned}$$

Step A3: Refine the probability updated in step A2 with the occurrence of S_2, and we acquire the final probability assessment taking into account all events by

$$\begin{aligned} P(C|S_1, S_1, S_2) &= \frac{P(S_2|C)P(C|S_1, S_1)}{P(S_2|C)P(C|S_1, S_1) + P(S_2|\overline{C})P(\overline{C}|S_1, S_1)} \\ &= \frac{0.80 \cdot 0.8448}{0.80 \cdot 0.8448 + 0.40 \cdot 0.1552} = 0.9159 \end{aligned}$$

Likewise we calculate the probability $P(\overline{C}|S_3, S_4)$ with two steps as follows:

Step B1: Update the prior probability $P(\overline{C})$ with occurrence of S$_3$

$$P(\overline{C}|S_3) = \frac{P(S_3|\overline{C})P(\overline{C})}{P(S_3|C)P(C) + P(S_3|\overline{C})P(\overline{C})} = \frac{0.62 \cdot 0.5}{0.35 \cdot 0.5 + 0.62 \cdot 0.5} = 0.6392$$

Step B2: Refine the probability updated in step B1 with appearance of S_4

$$\begin{aligned} P(\overline{C}|S_3, S_4) &= \frac{P(S_4|\overline{C})P(\overline{C}|S_3)}{P(S_4|C)P(C|S_3) + P(S_4|\overline{C})P(\overline{C}|S_3))} \\ &= \frac{0.30 \cdot 0.6392}{0.18 \cdot 0.3608 + 0.30 \cdot 0.6392} = 0.7470 \end{aligned}$$

Finally, with the required probabilities at hand, we can establish the case index for the time series X as follows

$$\begin{aligned} &Id_4(X|S_1, S_2, S_3, S_4) = [DP(V_1), DP(V_2)] \\ &= \left[P(C|S_1, S_1, S_2), P(\overline{C}|S_3, S_4)\right] = [0.9159, 0.7470] \end{aligned}$$

For similarity assessment, we first calculate the dissimilarity between two time series X_1 and X_2 as the average of the differences in discriminating powers over all key sequences clusters

$$Dis_4(X_1, X_2) = \frac{1}{K}\sum\nolimits_{j=1}^{K} |DP(V_{1j}) - DP(V_{2j})| \tag{9.50}$$

where V_{1j} and V_{2j} denote the jth clusters of key sequences corresponding to class $\mathrm{C_j}$, for X_1 and X_2, respectively. Since the concept of dissimilarity is opposite to that of similarity, the degree of similarity between X_1 and X_2 is simply defined as unity subtracted by the dissimilarity value

$$Sim_4(X_1, X_2) = 1 - Dis_4(X_1, X_2) \tag{9.51}$$

9.6 Relation to Relevant Works

Representation and retrieval of sequential sensor measurements as time series cases have received increasing research efforts during the recent years. The primary idea is to convert time-varying profiles into somehow simplified and shorter vectors that still preserve distances between original signals. Fourier transform and wavelet transform are two commonly used methods for such a conversion, and their usages for retrieving similar cases to support clinical decisions and industrial diagnoses have been shown in [33,35,36], respectively.

A more general framework for tackling cases in time dependent domain was proposed by [34], in which temporal knowledge embedded in cases are represented at two levels: case level and history level. The case level is tasked to depict single cases with features varying within case durations, while consecution of cases occurrences have to be captured in the history level to reflect the evolution of the system as a whole. It was also recommended by the authors that, at both of the two levels, the methodology of temporal abstraction [6,45] could be exploited to derive series of qualitative states or behaviors, which facilitate easy interpretation as well as pattern matching for case retrieval.

This chapter would be a valuable supplementary to the framework by Montani and Portinale in the sense that our key sequence discovery approach can be beneficially applied to the series of symbols abstracted from original numerical time series. The point of departure is that, in many practical circumstances, significant transitional patterns in history are more worthy of attentions than the states or behaviors themselves associated with single episodes. It follows that the key sequences discovered will offer us useful knowledge to focus on what are really important in case characterization. Moreover, as the number of key sequences is usually is much smaller than the number of elements in the series, indexing cases in terms of key sequences exhibits a further dimensionality reduction from series obtained via temporal abstraction.

It is worthy noting that the knowledge discovery treated here distinguishes itself from traditional learning included in a CBR cycle. The retain step in CBR typically stores a new case in the library or modifies some existing cases and may contain a number of substeps [1]. Learning therein is therefore case specific with knowledge stemming directly from newly solved cases. Contrarily, in our approach, learning is treated as a background task separated from the retain step and the whole case library is the input to the knowledge discovery process. Some relevant works combining knowledge discovery and CBR systems include: genetic-based knowledge acquisition for case indexing and matching [23], incremental learning to organize a case base [38], exploitation of background knowledge in text classification [53], and analysis of pros and cons for explanations in CBR systems [32].

Finding sequential patterns was widely addressed in the literature of sequence mining [2,13,46], where the goal was merely to find all legal sequential patterns with adequate frequencies of appearances. Identifying key sequences in our context differs from those in sequence mining in that we have to consider the cause-outcome effect for classification purpose. Only those nonredundant sequences that are not only frequent but also indicative in predicting outcomes will be selected.

Finally, time series data mining have gained increasing attention recently. Three embedding methods were proposed by [16] to transform time series data into a vector space for classification purpose. Keogh and his colleagues addressed the issue of dimensionality reduction for indexing large time series databases [25] and also for fast search in these databases [26]. In [52] a family of three unsupervised methods was suggested to identify optimal and valid features given multivariate time series data. Similarity mining in time series was tackled by [21] and various methods for efficient retrieval of similar time sequences were discussed in [9, 17, 37, 51]. Algorithms for mining association rules were handled in [28,40,49] to model and predict time series behaviors in dynamic systems, and the application of association mining to disclose stock prices relations in time series was presented in [20].

9.7 Conclusion

This chapter suggests a novel hybrid methodology combining data symbolization and knowledge discovery for analysis and interpretation of complex, longitudinal signals prevalent in medical and industrial domains. Data symbolization is tasked to transform primary numerical (usually real valued) series of measurements into shorter, more abstract series of symbolic data. The process of knowledge discovery is then applied to the case base of converted symbolic series to find key sequences, which would in turn help better characterizing and indexing primary numerical sensor signals into a concise case structure.

The knowledge discovery approach proposed utilizes the whole case library as available resources and is able to find from the problem space all qualified sequences that are nonredundant and indicative. An indicative sequence exhibits a high cooccurrence with a certain class and is hence valuable in offering discriminative strength for prediction and classification. A sequence that is both indicative and nonredundant is termed as a key sequence.

It is shown that the key sequences discovered are highly usable to characterize time series cases in case-based reasoning. The idea is to transform an original (lengthy) time series into a more concise case structure in terms of the occurrences of key sequences detected. Four alternate ways to develop case indexes based on knowledge about key sequences are suggested. The performance and applicability of these four case indexing methods are being tested in practical case studies related to our medical and industrial projects.

References

1. A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations and systems approaches. AI Communications, 7:39–59, 1994.
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the 11th International Conference on Data Engineering. (1995) 3–14
3. A. Apostolico, M.E. Bock, and S. Lonardi. Monotony of surprise and large-scale quest for unusual words. In: Proceedings of the 6^{th} International conference on Research in Computational Molecular Biology, Washington, DC, pp. 22–31, 2002.
4. Bar-Shalom, Y. and X. Li, Estimation and Tracking: Principles, Techniques, and Software, Artech House, Boston, 1993.
5. E. Beckenstein, G. Bachman and L. Narici. Fourier and Wavelet Analysis, Springer, 2000.
6. R. Bellazzi, C. Larizza, ans A. Riva. Temporal abstractions for interpreting diabetic patients monitoring data. Intelligent Data Analysis, 2: 97–122, 1998.
7. I. Bichindaritz and E. Conlon. Temporal knowledge representation and organization for case-based reasoning. In Proc. TIME-96, IEEE Computer Society Press, Washington, DC, 1996, pp. 152–159.
8. H.A. Braun et al. Low-Dimensional Dynamics in Sensory Biology 2: Facial Cold Receptors of the Rat. J. of Comp. Neuroscience 7(1), pp. 17–32, 1999.
9. Chan, K.P., Fu, A.W.: Efficient time series matching by wavelets. In: Proceedings of the International Conference on Data Engineering. (1999) 126–133
10. J.P. Crutchfield and K. Young. Inferring statistical complexity. Phys. Rev. Lett., Vol. 63, No. 2, pp. 105–108, 1989.
11. Daw, C.S., Finney, C.E.A.: A review of symbolic analysis of experimental data. Review of Scientific Instruments, 74(2): 915–930, 2003.
12. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery. In: Advances in Knowledge Discovery and Data Mining. MIT Press (1996) 1–36
13. Garofalakis, M.N., Rajeev, R., Shim, K.: SPIRIT: Sequential sequential pattern mining with regular expression constraints. In: Proceedings of the 25th International Conference on Very Large Databases. (1999) 223–234
14. J. Godelle and C. Letellier. Symbolic sequence statistical analysis for free liquid jets. Phys. Rev. E 62, Issue 6, pp. 7973–7981, 2000.
15. Gordon, N., A. Marrs and D. Salmond, Sequential Analysis of Nonlinear Dynamic Systems Using Particles and Mixtures, in: Nonlinear and Nonstationary Signal Processing, W. Fitzgerald, R. Smith, A. Walden, and P. Young, ed., Chapter 2, Cambridge University Press, Cambridge, 2001.

16. Hayashi, A., Mizuhara, Y., Suematsu, N.: Embedding time series data for classification. In: Perner, P., Imiya, A. (eds.): Proceedings of the IAPR International Conference on Machine Learning and Data Mining in Pattern Recognition. Leipzig (2005) 356–365
17. Hetland, M.L.: A survey of recent methods for efficient retrieval of similar time sequences. In: Last, M., Kandel, A., Bunke, H. (eds.): Data Mining in Time Series Databases. World Scientific (2004)
18. L.M. Hively, V.A. Protopopescu, and P.C. Gailey. Timely detection of dynamical change in scalp EEG signals. Chaos, Vol. 10, Issue 4, pp. 864–875, 2000.
19. M. Holschneider. Wavelet – An Analysis Tool. Oxford Science publications, 1995.
20. Huang, C.F., Chen, Y.C., Chen, A.P.: An association mining method for time series and its application in the stock prices of TFT-LCD industry. In: Perner, P. (ed.): Proceedings of the 4th Industrial Conference on Data Mining. Leipzig (2004)
21. Huhtala, Y., Kärkkäinen, J., Toivonen, H.: Mining for similarities in aligned time series using wavelets. In: Data Mining and Knowledge Discovery: Theory, Tools, and Technology. SPIE Proceedings Series, Vol. 3695. Orlando, FL (1999) 150–160
22. M.D. Jaere, A. Aamodt, and P. Skalle. Representing temporal knowledge for case-based prediction. In S. Craw and A. Preece, editors, Proceeding of the European Conference on Case-Based Reasoning, 2002, pp. 174–188.
23. J. Jarmulak, S. Craw, and R. Rowe. Genetic algorithms to optimize CBR retrieval. In E. Blanzieri and L. Portinale, editors, Proceedings of the European Conference on Case-Based Reasoning, pages 136–147. Springer, 2000.
24. S. Kadar, J. Wang, and K. Showalter. Noise-supported travelling waves in subexcitable media. Nature 391, pp. 770–772, 1998.
25. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Locally adaptive dimensionality reduction for indexing large time series databases. In: Proceedings of ACM SIGMOD Conference on Management of Data. Santa Barbara, CA (2001) 151–162
26. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Journal of Knowledge and Information Systems (2001)
27. J.-S. Kim et al. Decreased entropy of symbolic heart rate dynamics during daily activity as a predictor of positive head-up tilt test in patients with alleged neurocardiogenic syncope. Phys. Med. Biol. 45, pp. 3403–3412, 2000.
28. Last, M., Klein, Y., Kandel, A.: Knowledge discovery in time series databases. IEEE Trans. Systems, Man, and Cybernetics — Part B: Cybernetics 31 (2001) 160–169
29. Lee, S.K., White P.R.: The Enhancement of Impulse Noise And Vibration Signals For Fault Detection in Rotating and Reciprocating Machinery, Journal of Sound and Vibration 217 (1998), 485–505.
30. Lin, J.: Feature Extraction of Machine Sound Using Wavelet and Its Application in Fault Diagnosis, NDT&E International 34 (2001), 25–30.
31. S. Lonardi. Global detectors of unusual words: Design, implementation, and applications to pattern discovery in biosequences. Ph.D thesis, Department of Computer Sciences, Purdue University, 2001.
32. D. McSherry. Explaining the Pros andv Cons of conclusions in CBR. In P. Funk and P.A.G. Calero, editors, Proceedings of the European Conference on Case-Based Reasoning, pages 317–330. Springer, 2004.

33. S. Montani, et al. Case-based retrieval to support the treatment of end stage renal failure patients, Artificial Intelligence in Press.
34. S. Montani and L. Portinale. Case based representation and retrieval with time dependent features. In Proceedings of the International Conference on Case-Based Reasoning, pages 353–367, Springer, 2005.
35. Nilsson, M., Funk, P.: A Case-Based Classification of Respiratory Sinus Arrhythmia. In P. Funk and P.A.G. Calero, editors, Proceedings of the European Conference on Case-Based Reasoning, pages 673–685. Springer, 2004.
36. E. Olsson, P. Funk, and N. Xiong. Fault diagnosis in industry using sensor readings and case-based reasoning. Journal of Intelligent & Fuzzy Systems, 15:41–46, 2004.
37. Park, S., Chu, W.W., Yoon, J., Hsu, C.: Efficient search for similar subsequences of different lengths in sequence databases. In: Proceedings of the International Conference on Data Engineering. (2000) 23–32
38. P. Perner. Incremental learning of retrieval knowledge in a case-based reasoning system. In K. D. Ashley and D. G. Bridge, editors, Proceedings of the International Conference on Case-Based Reasoning, pages 422–436. Springer, 2003.
39. Pous, C., Colomer, J., and Melendez, J.: Extending a fault dictionary towards a case based reasoning system for linear electronic analog circuits diagnosis. In: Proceedings of the 7$^{\text{th}}$ European Conference on Case-Based Reasoning, Madrid, 2004, pp 748–762.
40. Pray, K.A., Ruiz, C.: Mining expressive temporal associations from complex data. In: Perner, P., Imiya, A. (eds.): Proceedings of the IAPR International Conference on Machine Learning and Data Mining in Pattern Recognition. Leipzig (2005) 384–394
41. Ray, A.: Symbolic dynamic analysis of complex systems for anomaly detection. Signal Processing 84 (2004) 1115–1130
42. R. Schmidt, B. Heindl, B. Pollwein, and L. Gierl. Abstraction of data and time for multiparametric time course prognoses. In: Advances of Case-Based Reasoning, LNAI 1168, Springer-Verlag, Berlin, 1996, pp. 377–391.
43. J. Lin, E. Keogh, S. Lonardi et al. A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pages 2–11, 2003
44. G. Salton. Automatic information organization and retrieval. New York: McGraw-Hill, 1968.
45. Y. Shahar. A framework for knowledge-based temporal abstractions. Artificial Intelligence, 90:79–133, 1997.
46. Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: Proceedings of the 5th International Conference on Extending Database Technology. (1996) 3–17
47. S.D. Stearns. Digital Signal Processing with Examples in Matlab. CRC Press, Florida, 2003
48. X.Z. Tang, E.R. Tracy, and R. Brown. Symbol statistics and spatio-temporal systems. Physica D, Vol. 102, Issue 3–4, pp. 253–261, 1997.
49. Tung, A.K.H., Lu, H., Han, J., Feng, L.: Breaking the barrier of transactions: Mining inter-transaction association rules. In: Proceedings of ACM Conference on Knowledge Discovery and Data Mining. (1999) 297–301

50. G. Tzanetakis, G. Essl, and P. Cook. Audio Analysis using the Discrete Wavelet Transform. In Proceedings of the WSES International Conference on Acoustics and Music: Theory and Applications (AMTA 2001) Skiathos, Greece, 2001.
51. Wu, Y., Agrawal, D., Abbadi, A. EI: A comparison of DFT and DWT based similarity search in time series databases. In: Proceedings of the 9th ACM CIKM Conference on Information and Knowledge Management. McLean, VA (2000) 488–495
52. Yoon, H., Yang, K., Shahabi, C.: Feature subset selection and feature ranking for multivariate time series. IEEE Trans. Knowledge and Data Engineering 17 (2005) 1186–1198
53. S. Zelikovitz and H. Hirsh. Integrating background knowledge into nearest-neighbor text classification. In S. Craw and A. Preece, editors, Proceedings of the European Conference on Case-Based Reasoning, pages 1–5. Springer, 2002.