Part III

Knowledge extraction from time series

Chapter 10

Supervised learning with temporal data

In this chapter, we first give our motivation for studying automatic learning in the context of temporal data. In particular, we discuss the practical relevance of the topic and argue intuitively why automatic learning has some specific problems in this context. We also position this problem as a particular case of the more general problem of handling complex structured data in automatic learning. Then, we formally define the problem of supervised learning with temporal data, particularize it to time series classification, and explain more in detail why automatic learning in this context is difficult. Before concluding, we provide a description of datasets used in the sequel and give a short guide to related literature on time series classification.

10.1 Motivation

The basic supervised learning problem is defined in Chapter 2 as the problem of learning a function of several input attributes which predicts as well as possible the value of an output attribute. Both input and output attributes were defined as random variables taking their values either on the real axis or in a discrete set of symbolic values. Most mature supervised learning algorithms are actually designed for solving such problems. However, in recent applications of supervised learning, algorithms are facing much more complex data (as input attributes but also at the output of the model) which structure does not fit easily to the simple attribute-value framework used in the first part of this thesis. For example, the field of text mining studies methods that extract information from texts and, in particular, techniques to classify documents automatically according to some user's preferences [Coh96]. In bioinformatics [BB98], automatic learning algorithms must be able to handle data in the form of DNA sequences, for example in order to locate or classify genes. In network security, researchers try to build intrusion detection algorithms which aim at identifying abnormal behaviors of users on the basis of a temporal sequence of discrete system events triggered by the user. In this case, the desired output of the model is thus a time varying classification of the user as a normal user or as an intruder [LS98]. Many other applications of automatic learning are also intended to handle images. For example, automatic learning is used in [BAS+98] to detect volcanoes on Venus from satellite images. In [Sen99], the goal is to detect and follow faces in a video sequence, and the desired output of a model is the sequence of positions of different faces in the successive video frames. In all these examples, the input information which must be processed, and in many cases also the output information provided by the model, are clearly more complex than a single numerical or symbolic value.

From the automatic learning point of view, these problems share some common characteristics that we want to highlight. At the basic level, data corresponding to these problems is represented by a large number of low-level attributes (characters in the case of text, pixel values in the case of images, instantaneous values in the case of time series, elementary alleles in the case of genetics, ...) structured in a uni-,bi- or tri-dimensional space. For example, in the case of digital image processing, a 1024×1024 high resolution grey level image would be represented by 1024×1024 low-level numerical attributes. In general, in these applications the original low-level attributes independently give almost no information about the output variable; rather it is the way they interact among each other which defines the global properties on the basis of which classification or prediction models could be formulated easily. For example, in an e-mail filtering application, it is clear that nothing interesting can be said from the fact that an "a" appears in position 10 in a text. Rather, it is the way the characters are combined to form words, and words to form sentences which is important in terms of global interpretation. The high number of low-level attributes is responsible for the main difficulties found when applying automatic learning to these problems. First, if automatic learning algorithms are applied directly on the basis of the low-level representation they need to build very complex models combining in some fashion the large number of low-level attributes in order to extract information from them. As a result, these methods will almost surely suffer from a very high variance.[1] Also, these problems often correspond to very large datasets (especially if the variance is to be kept low) and for computational efficiency reasons, some techniques can not be applied to these problems. Another difficulty is to produce interpretable rules in such cases. Indeed, even supposing that it would be possible to build an accurate decision tree or rule using the low-level attributes (say pixels) to classify objects (say images) the resulting model is likely to be very intricate and essentially impossible to interprete.

Practically, these difficulties can nevertheless be overcome by introducing a priori information about the problem. For example, in a document classification problem, we know from our knowledge of the language that a text can be segmented into words and that it is a good idea to characterize different texts on the basis of the frequency of occurrence of some well chosen words. From this prior knowledge, the goal of supervised learning can thus be reduced to the simpler problem of the choice of the words to combine to give an accurate classification rule. As suggested by this example, we can apply the standard automatic learning methods to these problems, provided that we are able to change the representation of objects by defining new attributes (or *features*) which can be computed from the low-level ones and are more directly related to the output variable. This is quite frustrating, since the most difficult part of the job is in this way carried out during the feature extraction step, which in practice relies highly on human expertise (a priori knowledge about the problem and about the automatic learning method used) and is difficult to formalize. Also, in a growing number of applications such a priori information is not available, or there is not enough time (or money) available to let a human expert work out the synthetic features. In such domains, what is missing is general approaches which would be able to directly use the low-level attributes and without requiring other information would be able to infer the models. Such methods could either explicitly combine an automatic feature generation mechanism with a general purpose automatic learning method, or directly use the low-level features to formulate the models. In this latter case, we would need automatic learning methods which variance does not increase significantly with the number of attributes.

[1]Because of the high number of low-level attributes, it is easy to find models which perfectly separate learning samples but which in terms of generalization are meaningless.

The preceding discussion gives some ideas and motivations around current topics in automatic learning from complex structured data. In our research, we have considered the particular problem where input data are time series and the output attribute is a constant symbolic classification variable. We call this problem the time series classification problem. The possible application domains for time series classification are numerous. In speech recognition, an acoustic speech signal must be mapped to a set of words. In a medical application, we are interested in detecting heart diseases from electrocardiograms (ecg) of patients. Intrusion detection systems aim at classifying user event sequences as normal or not. In all these application fields (especially speech recognition), there exist working systems but these systems make many a priori hypotheses about the application domain and tend to be of low flexibility. In this thesis, we are interested in designing general, possibly more flexible approaches to handle such problems. Note that we are aware that our goal is quite ambitious, and also that our contribution is only a modest first step towards this goal.

Before focusing on the time series classification problem, we will in the remaining of this chapter define the framework and notation for supervised learning with temporal data, in general, and discuss several typical temporal classification problems so as to circumscribe more precisely the scope of what we call time series classification. Then, in Section 10.3, we discuss more in depth the peculiarities of this problem that could prevent naive approaches from working. Note that (part of) this discussion can more generally be applied to other problems such as image and text classification. The last section of this chapter describes the datasets that will be used throughout the experiments of the other two chapters of this part of the thesis. Note that two of these test problems have been specifically designed to make appear the peculiarities discussed in Section 10.3.

10.2 Problem statement

In this section, we first generalize the framework given in Chapter 2 to handle temporal data in supervised learning. Then, we propose a classification of the problems according to the exact nature of the output attribute. The particular problem which will be tackled in this part of the thesis is the problem of time series classification. This problem and our simplifying hypothesis are described in Section 10.2.3.

10.2.1 General framework for temporal supervised learning

When dealing with temporal data, an object from the universe $\mathcal{U}$ denotes a dynamic system trajectory (or episode, scenario)[2]. In order to describe such a trajectory, the notion of attribute must be extended by allowing time functions as attribute values. A temporal attribute is thus a function of object and time, defined on $\mathcal{U} \times [0, +\infty[$[3]:

$$A(.,.) : \mathcal{U} \times [0, +\infty[\mapsto \mathcal{A}, \tag{10.1}$$

where $\mathcal{A}$ is the codomain of the function $A(o, .)$. In what follows, we denote by $A(o, t)$ the value of the attribute A at time t for the object o. $A(., t)$ for each $t \in [0, +\infty[$ is thus a random variable and the time function $A(., .)$ denotes what is called a random or stochastic process. In what follows, we will sometimes drop the o variable from the notation and use respectively $A(t)$ and $A(.)$ to denote these two random values. Of course,

[2]In what follows, we will indifferently use the terms object and scenario to denote an element of $\mathcal{U}$.

[3]We assume that time is continuous, and without loss of generality, that the starting time of a scenario is always 0.

non temporal attributes can still be mixed with temporal ones to describe scenarios; to simplify our notation, they will be considered as a particular case of a temporal attribute where the function $A(o,.)$ is constant for any o. In a supervised learning problem, a particular attribute denoted by Y is the output variable and input attributes are denoted by a vector $\underline{A}$, which is now a vector of random processes.

As in Chapter 2, mainly two types of temporal attribute values may be considered:

- Symbolic values: the codomain of the time function is a finite set $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$ of symbolic values. This kind of temporal attribute models a sequence of discrete events (for example, the evolution of the on/off status of some switch or the sequence of discrete actions performed by a user on a computer).

- Numerical values: the codomain of the time function is the real axis $\mathbb{R}$. A numerical temporal attribute may be used to model the continuous evolution of some measurement on the system.

A learning sample is a sample of N objects randomly drawn from $\mathcal{U}$ and described by their attributes. Each object of the learning set is further defined on a finite interval of time $[0, t_f(o)]$ during which the values of its attributes have been observed. Note that, in general, observation times may be different from one scenario to another. Denoting by $A_{[0,t']}(o,.)$ the restriction of the attribute A to the interval $[0, t']$ and by $\underline{A}_{[0,t']}(o,.)$, the same restriction applied to the vector of input attributes, we have:

$$ls = ((\underline{A}_{[0,t_f(o_1)]}(o_1,.), Y_{[0,t_f(o_1)]}(o_1,.)), \ldots, (\underline{A}_{[0,t_f(o_N)]}(o_N,.), Y_{[0,t_f(o_N)]}(o_N,.))). \quad (10.2)$$

The goal of supervised learning is to induce from a learning set ls a function of object and time $f(.,.)$ which approximates as well as possible the output attribute $Y(.,.)$ for objects of $\mathcal{U}$. Of course, the model $f(o,t)$ must satisfy the following property:

$$f(o,t) = g(\underline{A}(o,.), t), \quad (10.3)$$

i.e. compute a prediction at time t for an the object o which is a function of its input attributes only.

Discussion

The preceding formalization has been made in a rather general way so as to cover a broad spectrum of temporal learning problems.

In particular, we have assumed that the time axis is continuous and that temporal attribute values may change continuously through time, because in some applications it is difficult to fix a priori a sampling period (in particular for the output of the models produced by automatic learning). In terms of computer representation, even if the functions are defined on the continuous time axis, they will actually be represented by the combination of a finite number of parameters and some implicit interpolation or function generation mechanism; these parameters could be temporal samples, or expansion coefficients of some set of basis functions, for example.

The output produced by automatic learning is a function of time, in the general case, and this function is not necessarily generated in a causal way from the temporal attribute values. Further restrictions can be introduced when the framework is particularized to some practical (on-line) applications (see below).

10.2.2 Classification of temporal supervised learning problems

Depending of the type of the output attribute Y and the form of the function f, very different problems may be fitted to our general framework. A classification of these problems is given below.

Constant vs temporal output

Besides the distinction between symbolic or numerical output variable, there is a more important distinction in the presence of temporal data:

- $Y(o,t) = Y(o,t'), \forall t, t'$, i.e. $Y(o,t)$ is constant over time. It is the classical supervised problem but now with temporal attributes. The output variable may describe in some way the state of the system at the end or at the beginning of the scenario which should be determined from its observation during some finite period of time. For example, in speech recognition, the goal is to determine the word which has been pronounced from the observation of the acoustic voice signals.

- $Y(o,t)$ is a time function. When $Y(o,t)$ takes its values on the real axis, one common application is time series prediction. In this case, there is only one temporal attribute A and the output is the time shifted version of this attribute ($Y(o,t) = A(o, t+\triangle t)$, where $\triangle t$ defines the time horizon for the prediction) which has to be approximated at time t from only present and past attribute values. One possible application is the prediction of the evolution of prices in a financial market. When $Y(o,t)$ is discrete, we are facing a segmentation and labeling problem. Indeed, the time interval has to be divided into segments where the classification is constant and each segment has to be labeled with the correct class. One common application is again speech recognition, where it is sometimes needed to segment a phrase before labeling each of its constituting words. Note that segmentation is an application where it is difficult to predict in advance at what times the output may change, so it fits better in a continuous time setting than in a discrete time setting.

Off-line vs on-line model

Another distinction between problems appears when it comes to apply the model in practice. Sometimes, the model must be applied on-line, i.e. be able to receive measurements on the system in real-time and process this information to give a prediction (e.g. forecasted output) at each time step. In such cases, the output at time t must take into account only past and present attribute values, i.e. be of the form (causality):

$$f(o,t) = g(\underline{A}_{[0,t]}(o,.), t). \tag{10.4}$$

In other situations, the model is used off-line to analyze the data (which is supposed to be complete) and then, the desired model can have the more general form:

$$f(o,t) = g(\underline{A}_{I(t)}(o,.), t), \tag{10.5}$$

where $I(t)$ is a subinterval of $[0, +\infty[$ and $\underline{A}_{I(t)}(o,.)$ is the restriction of the attribute vector to this interval. The difference is not only esthetic. Indeed, in the first case, some interesting techniques simplifying the problem are not applicable. For example, normalization of the length in time of the signals and classical Fourier transformation do not make any sense with partially observed signals.

A practical model is sometimes a combination of these two kinds of models. For example, in a real-time speech recognition system, a first model may segment the time series on-line into words and then each of these segments are processed by an off-line model which gives the output word corresponding to the data in this segment.

Early detection problems

In some cases, even though the desired output variable is constant over time, it makes sense to use an automatic learning method which will produce a time varying output. Indeed, for a non temporal output variable, an on-line temporal model could actually produce predictions which may evolve with time as the available information about the system behavior increases. For example, consider an intrusion detection system and suppose that objects represent the sessions of an FTP sever, in terms of sequences of system calls generated by the server. The goal would be to build an algorithm observing a stream of such events (for a given session) and if the session corresponds to an attempt to intrude into the system provide an alarm. In this application, objects are sessions and for the purpose of supervised learning we would use a subset of normal sessions and a subset of intrusion attempts. The objective of automatic learning would be to build a causal function of the input stream of system events, which as early as possible would switch from "normal" to "alarm", so as to cancel the session before the intruder has started taking advantage of his intrusion. Notice that in this application it would be very difficult to determine in advance how many system events would be necessary to generate an alarm (in the case of intrusion); also the time required to detect intrusions in a reliable fashion would most probably vary in practice from one intrusion attempt to another.

The preceding example highlights some specific features of the "early detection problem". In general, the early detection problem corresponds to applications where one seeks to induce a monitoring system that can trigger emergency procedures of a system. Other examples are patient monitoring in a medical application [KCC01] or anticipation of power systems black-out [GW98]. One way to handle this problem by automatic learning is to use a constant (over time) desired output, a causal temporal model, and to incorporate into the search procedure used by the automatic learning method a preference with respect to models which carry out the detection as early as possible. Note that in these problems, accuracy and earliness of detection are usually competing objectives, because in many detection problems the "undesired" behavior tends to become sooner or later obvious to detect. For example, in the work reported in [GW98], we have designed a method based on temporal decision trees for the early on-line detection of instability in electric power systems; in this application the objective was to detect (or predict) instability when it is still time to prevent it, i.e. before its undesired consequences start to show up. In this problem, instability sooner or later translates into a monotone and strong decrease of voltage levels in the power system, and, without bias towards early detection, the automatic learning method would have produced useless models detecting instability when it is already far too late to react.

10.2.3 Time series classification

From the above analysis, it is clear that the introduction of the time dimension in the supervised learning problem increases very much the complexity and the diversity of the encountered problems. In this thesis, we focus on the particular problem of temporal signal classification. Every scenario, o, is classified into one class, $C(o) \in \{c_1, \dots, c_m\}$ and the goal of learning is to find a function $f(o)$ which is as close as possible to the true classification $C(o)$. This function should depend only on attribute values, i.e. $f(o) =$

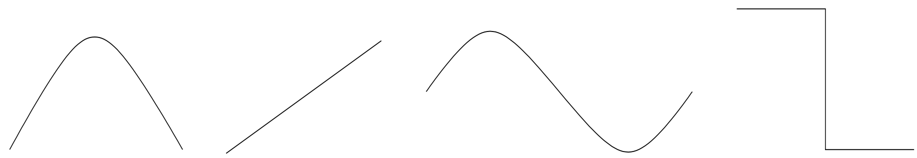

Figure 10.1: Four examples of patterns

$g(\underline{A}(o,.))$, and it should not depend on absolute time values. A consequence of this latter property is that the model should be able to classify every scenario whatever its duration of observation.

Furthermore, we restrict ourselves to numerical attributes and we assume a discrete time representation of such attributes:

$$A(.,.) : \mathcal{U} \times \{t_0, t_1, t_2, \ldots\} \mapsto \mathcal{A} \subset I\!\!R. \tag{10.6}$$

Each scenario is described by a sequence of vectors: $(\underline{A}(o,t_0), \underline{A}(o,t_1), ..., \underline{A}(o,t_{n(o)-1}))$, where the number of time-points $n(o)$ is possibly object dependent. We further assume that the discretization step of time is constant over time and object, i.e. $t_i = i. \triangle t, \forall i$. This is not a too strong assumption since continuous signals usually are sampled for representation in computer memory and, if the discretization step is not constant, it can be changed by resampling the time signal at the lowest time scale. Usually, in the literature, the term "time series" is devoted to such discrete time and constant discretization step representation of temporal signals.

As the models we will develop here are independent of absolute time values, time instants can most of the time be forgotten. So, in what follows, we will denote by $(a_0, a_1, \ldots, a_{N-1})$ a vector of instantaneous values defining a time series, without explicit reference to the time instants t_i, and we will identify elements of $I\!\!R^N$, time series of N time steps, and temporal signals defined on the bounded interval $\{t_0, t_1, \ldots, t_{N-1}\}$. Also, although we are essentially dealing with the time series representation, we will however sometimes use the term temporal signal to denote a time function corresponding to an attribute.[4]

10.3 Peculiarities of time series classification problems

A large diversity of problems can be imagined with temporal data in terms of the goal of learning. Even restricting ourselves to time series classification, there are still many different possible behaviors that can characterize scenarios of one class with respect to other classes and it is probably too ambitious to aim at building a single learning algorithm which would be able to handle effectively every such kind of behavior. Therefore, we will build up our reasoning here on a class of problems for which we suppose that it is possible to discriminate between classes by extracting some elementary more or less local *patterns* from the temporal signals corresponding to attributes and by combining these patterns into decision rules able to express temporal relationships among them (from a simple temporal ordering of the patterns to a strict temporal constraint on their relative start times).

More precisely, we define a pattern as a limited support signal that satisfies some properties specified a priori (say in the form of a dictionary of elementary signals and a

[4]Moreover, we will (non-orthodoxly) mix sequence, vector, and signal notation where we find it appropriate.

pattern generation mechanism). A few examples of such patterns are given in Figure 10.1: a local maximum, an increasing trend, a cycle, a decreasing step... These patterns play a similar role to the words in the context of text mining, and we assume that it is possible to discriminate between different classes of signals by expressing rules or models of moderate complexity formulated on the basis of a small number of such local patterns.

Note that there are many problems that do not satisfy this locality requirement. For example, a system that discriminates scenarios on the basis of the integration of a temporal attribute (or, in the discrete case, by counting the number of occurrences of a symbol) is impossible to model with only local characteristics. Nevertheless, many of these problems may be solved by introducing new temporal attributes which are functions of the original ones and by detecting some local patterns among these latter. For example, the integration or counting problems may be solved by introducing a new attribute which computes the integration of new functional attribute

$$a'(o,t) = \int_0^t a(o,\tau)d\tau$$

and which would present discriminative local characteristics for the problem under consideration.

What we do not propose at this stage is a technique that would be able to automatically build such functional attributes. Rather, we rely on the user for an a priori good choice of attributes.

Even with this further restriction, the problem of temporal signal classification may be difficult to solve directly with existing standard learning algorithms. What we call here a direct (or naive) approach consists in sampling temporal signals at different moments and using these newly extracted non temporal attributes to induce a model without taking into account their temporal ordering. The limits of this naive approach will be evaluated in the next chapter but, before looking at these results, let us analyze what would prevent such techniques to yield good results with temporal data or equivalently what makes classification with temporal data a challenging task (see also [Kad98] for a similar discussion).

- **Variable number of features.** We have seen that scenarios may be described on different intervals of time ($t_f(o)$ and $n(o)$ are in general variable from one object to another) and thus sampling may result in a different number of features from one scenario to another, which is impossible to tackle with most learning algorithms. Some solutions must be imagined (e.g. truncation of larger signals or normalization of time before sampling).

- **Large number of features.** The number of non temporal attributes extracted by sampling is equal to the number of temporal attributes times the number of time instants. If the dynamics of the system are fast, the sampling period must be small in order not to loose information. Hence the number of potential attributes may be high, thereby increasing variance and overfitting.

- **Different time or amplitude scale.** Discriminative patterns may differ in amplitude and duration but still represent the same local properties in the signal. For example, it is clear that the three signals of Figure 10.2 represent similar behaviors (a maximum immediately followed by a minimum) but the second one is obtained from the first one by a linear transformation of amplitude and time scale while the third one is obtained by a non linear transformation of these two axes. Thus, these signals may correspond to very different attribute vectors (in terms of the euclidian

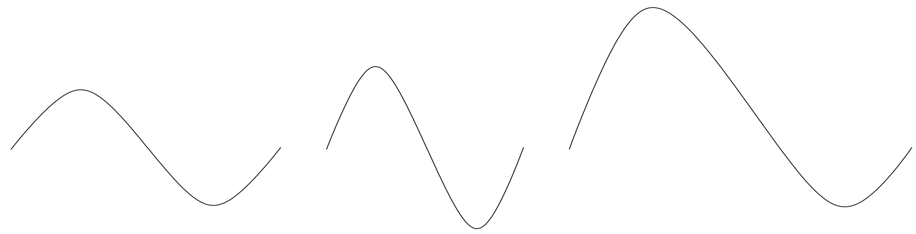

Figure 10.2: Different transformations of the same pattern

distance) when discretized although they may characterize scenarios of the same class.

- **Shift invariant properties.** If we are interested in an on-line model, the absolute location in time of a pattern in a signal should not be relevant for classification (only the relative distance between different patterns may be important). The representation of such shift invariant properties is difficult with the naive approach.

- **Temporal constraints.** Pattern detections may need to be relied by strong temporal constraints for classification purpose. These relationships may also be difficult to model using traditional learning algorithms.

Because of their universal approximation properties, learning algorithms like decision trees and neural networks are theoretically able to model the last three properties of temporal problems. However, this is at the price of very complex models and thus only possible if the learning set is large and if good care is taken to reduce the variance. Before proposing real temporal approaches, the next chapter is devoted to approaches based on sampling which do not really take into account the temporal ordering of attributes.

Note that the characteristics presented here are not the monopoly of temporal problems. For example, image classification models are certainly facing the same problems, further amplified in this domain by the additional dimension.

10.4 Test problems

While there are very large repositories of data sets which can be used to validate classical algorithms (for example [Bay99]), there is not such a large choice of datasets in the temporal domain, and hence many researchers rely on artificial problems. In this thesis, we use six problems for the validation of our algorithms among which four are artificial. These datasets are summarized in Table 10.1 in terms of the number of temporal attributes, number of classes, number of time points (minimum and maximum if it is variable) and the total number of scenarios. Notice that the literature on time series classification via automatic learning is for the time being much more scarce than the literature on standard supervised learning, so we have also incorporated in our datasets some which have been used by other researchers, so as to allow us to compare some of our results with those obtained by others.

The first four problems are the artificial ones, specifically designed to present properties typical of temporal problems. With respect to the enumeration of the previous section, CBF is mainly characterized by amplitude and time scale invariant properties. CBF-tr is a variant of this problem which emphasizes the shift invariant property (already present in the CBF problem). In Two-pat, the combination of several patterns is necessary for classification. These problems are described below in quantitative terms. The exact way datasets are generated is reported to Appendix B.

Table 10.1: Temporal data sets summary

Data set	Nb. attributes	Nb. classes	Nb. instants	$\#DB$
CBF	1	3	128	796
CBF-tr	1	3	128	5000
Two-pat	1	4	128	5000
CC	6	1	60	600
Auslan	8	10	32-101	200
JV	12	9	7-29	640

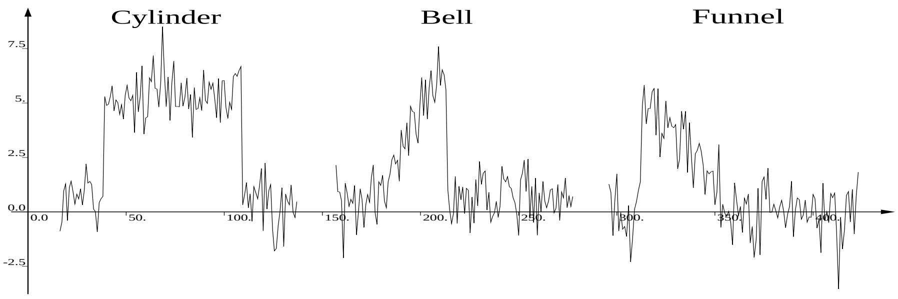

Figure 10.3: An example of each class from the CBF dataset

Cylinder-Bell-Funnel (CBF)

This problem was first introduced in [Sai94] and then used in [Man97, Kad99, AR00] for validation of time series classification systems. There is only one temporal attribute and the goal is to separate three classes of objects: cylinder(c), bell(b), and funnel(f). Figure 10.3 shows an example of each class. Each class is characterized by a specific pattern. The cylinder[5] class is characterized by a plateau, the bell class by an increasing (linear) ramp followed by a sharp decrease, the funnel class by a sharp increase followed by a decreasing ramp. To catch some typical properties of the time domain, the start time, the duration, and the amplitude of the pattern are randomized from one scenario to another. A random additive independent Gaussian noise is added to each time point. Note that because of this noise, the residual error in this problem is not necessarily null. As in [Man97], we generate 266 examples for each class.

CBF-translated

Although the start times of patterns are randomized in the CBF problem, the pattern is still always more or less centered in the time interval. In order to enable us to study the robustness of some of our algorithms to cope with shift-invariant properties, we have generated another version of the same dataset by emphasizing the temporal shifting of patterns. In this variant, the durations of patterns are less variable than in CBF but their start time is much more randomized. Figure 10.4 shows several examples of the bell class. This problem should be harder for non temporal approaches than the original CBF problem.

As this dataset will be used for bias/variance computation, we draw a large dataset of 5000 cases which will be divided into a pool set of size 4000 and a test set of size 1000.

[5] To our knowledge, the names of the class do not refer to some application; they probably are related to the geometrical shape of the signals.

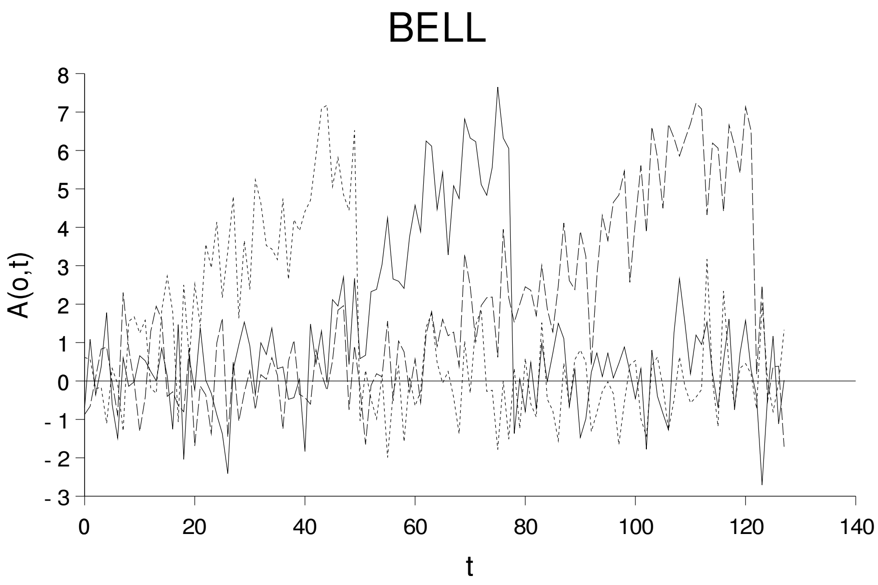

Figure 10.4: Three instances of the bell class in CBF-tr problem

Two pattern (Two-pat)

The CBF and CBF-tr problems present some typical properties of the temporal domain but the classification of temporal signals is based on the detection on single simple patterns. In some problems however, it is necessary to combine the occurrence of several patterns for classification. To validate our algorithms with respect to such cases, we designed a simple artificial problem where a class is characterized by the occurrence of two patterns in a definite order. We will call this problem the Two-pat problem.

The problem is characterized by a single temporal attribute and four classes of signals. Figure 10.5 shows one element of each class. Two different patterns are used to define classes: an upward step (which goes from -5 to 5) and a downward step (which goes from 5 to -5). Class UU corresponds to the succession of two upward steps, UD to the succession of an upward step and a downward step, etc. By defining classes in this way, it is necessary to detect the strict succession of patterns to distinguish elements of the class DU from elements of the class UD. The positions and durations of the patterns are randomized (but in such a way that they never overlap). Around patterns, the signal is characterized by a independent Gaussian noise (of mean 0 and unit standard deviation). Note that in this problem, because there is no noise on the pattern and because the noise level is rather small with respect to pattern levels, the residual error is negligible. However, we expect this problem to be especially difficult to handle with classical (non-temporal) learning algorithms.

Like CBF-tr, this problem will be used for bias/variance computation and hence a large dataset of 5000 scenarios is generated.

Control chart (CC)

This data base was proposed in [AM99] to validate clustering techniques and used in [AR00] for classification problems. Objects are described by one temporal attribute and classified into one of six possible classes: normal, cyclic, increasing, decreasing, downward and upward. Figure 10.6 shows two scenarios of each class.. What makes this problem challenging for clustering or classification algorithms in the temporal domain is that

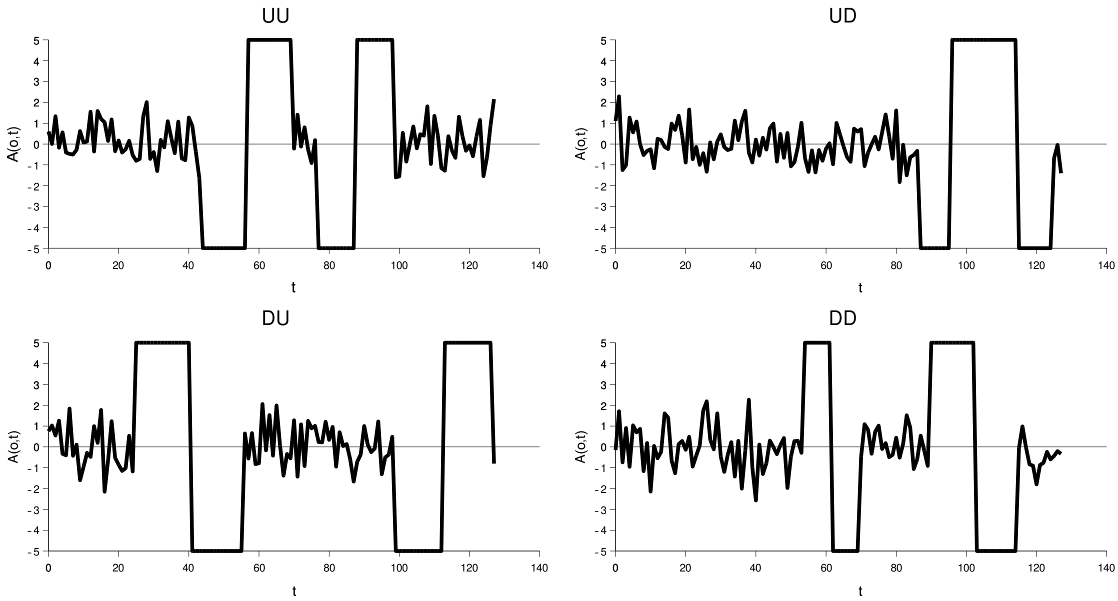

Figure 10.5: An example of each class from the Two-pat dataset

some pairs of classes are difficult to separate (cyclic and normal, upward and increasing, downward and decreasing).

The dataset was obtained from the UCI KDD Archive [Bay99] and contains 100 objects of each class. Each time series is defined by 60 time points.

Japanese vowels (JV)

This dataset is also available in the UCI KDD Archive and was built by Kudo et al. [KTS99] to validate their multidimensional curve classification system. The dataset records 640 time series corresponding to the successive utterance of two Japanese vowels by nine male speakers. Each object is described by 12 temporal attributes corresponding to 12 LPC cepstrum[6] transform coefficients. Each signal is represented in memory by between 7 to 29 time points. The goal of machine learning is to identify the correct speaker from this description.

Australian sign language recognition (Auslan)

This dataset was generated by Kadous [Kad99] to validate a time series classification system and used by [AR00]. He recorded instances of the Australian sign language (the language of the Australian deaf community, called Auslan) using an instrumented glove. There are ten classes corresponding to ten different words. Each example is described by 8 temporal numerical attributes: the x, y, and z positions of the hand, wrist roll, thumb, fore, middle, and ring finger bend. The number of time points in each example is variable and ranges between 32 and 101.

10.5 Existing approaches for time series classification

The problem of time series classification has been approached by many researchers from a very broad spectrum of origin: statistics, machine learning but mainly engineering commu-

[6] This is a special transform often used in speech recognition.

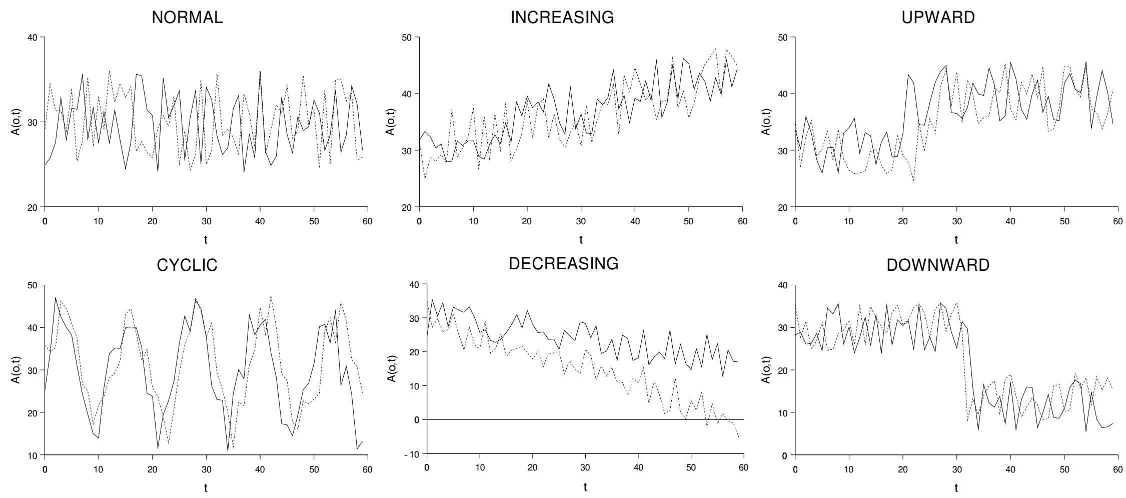

Figure 10.6: Two examples of each class from the CC dataset

nity. However, most of the literature in this domain is applicative and proposes techniques which do not generalize directly to other domains or which need a lot of preprocessing to be applied. Because of this very large spectrum of techniques, the literature is also difficult to appraise. So, the brief review given below is at best superficial and certainly incomplete. The goal of this discussion is more to give some idea of the different approaches that exist but that we have not deepened in our search for a general method. Approaches more directly related to our work will be discussed at the end of each chapter.

A very large spectrum of methods for time series classification originate from the speech recognition community. Another application domain for time series classification which presents similar problems and hence solutions is handwriting recognition (where time series record the time varying positions of the hand or the pencil when writing). In these fields, the approach generally adopted is to model the probability density function of the attributes in each class. It is assumed that the signal in each class has been generated by some stochastic process. A model for this stochastic process is learned for each class and then to classify a new scenario, the probability that it has been generated by the model corresponding to each class is computed and the one which has the most probably generated the signal is chosen. Generally, a parametric form is chosen for the model and the parameters of this model are learned from the data. There exist many different techniques to model stochastic processes. The most widely used models in speech recognition are certainly hidden Markov models (see [Rab89] for a tutorial). In such models, it is assumed that the signal at each time step is generated from a distribution conditional of a hidden state random variable which value evolves in time following a first order Markov hypothesis (the state at time t directly depends only on the state at time $t-1$). Much of the recent works in this domain tries to circumvent the limitation of the hypotheses of a simple hidden Markov model (for recent development in HMM-based models see [Ben99]). To model stochastic processes, another method is to take advantage of the many predictive models which have been proposed for time series prediction. These models give a probability distribution on the next value of a time series given the previous values and hence are also models of a stochastic process (for a review of predictive models and their use for speech recognition see [Gal98]). Generally, the design of models for stochastic processes uses a lot of a priori information (see [Ben99]). When the model is complex (contains a lot of parameters), this information is even necessary to keep learning algorithms efficient and avoid the problem of overfitting.

Although natural for problems like speech recognition, this approach is not economic since it supposes to model the whole process in each class. On the other hand, discriminative approaches focus directly on a modeling of the classification frontier. The usual approach for time series classification is to extract a set of features and use existing learning algorithms to solve the classification problem. Since using the initial low-level attributes corresponding to each time point is often not appropriate (see the discussion of Section 10.3 and experiments of the next chapter), some techniques may be used to reduce the dimensionality of the problems by selecting good attributes for classification purpose. The simplest way is to use prior information about the problem to manually define useful attributes for classification. This approach is for example adopted in [SLCM98] for a robot learning problem. There exist however general feature extraction techniques such as principal component analysis (PCA) which can be used for time series classification. For feature extraction, one can also take advantage of the many techniques of time series analysis which compute global characteristics of a time series in terms of various statistics and parameters of predictive models (see [BJ76, WG94]). For a review of feature extraction and selection for time series classification, see for example [Has00] and the references therein. For time ordered data, a common approach is also to apply some transform like Fourier transform [Shu82] or wavelet transform [Sai94] and use the transformed signal for classification. According to the particular application domain, one or several of these feature extraction techniques may be useful for classification and often it requires a lot of trials and errors to find the appropriate feature subsets. In this phase, one can also use automatic feature selection techniques (like in [KJ97]) but these techniques do not tell us which initial (large) set of features to choose.

With respect to all this literature, the problem of time series classification has received little specific attention from the machine learning community. Few people have tried to design learning algorithms to solve the time series classification problem. There exists nevertheless some recent works [Man97, Kad99, KTS99, AR00, Ols01]. With respect to the approaches described above, the main difference is logically the focus on the induction of interpretable models and the use of as little as possible knowledge about the application problem. The way to satisfy these objectives is generally to automatically extract features which are used by a standard automatic learning algorithm. So as to yield a general approach, these features are typically automatically adapted in some way to the data. As they are more strongly related to our work, these approaches will be discussed individually in the related work section of Chapter 12.

10.6 Conclusion

In this chapter, we have introduced the time series classification problem and the underlying hypothesis of our approach in this thesis. Our goal in this thesis is to design general approaches, which do not rely on any a priori knowledge about the application domain. In the next two chapters, we tackle this objective in two ways:

- Chapter 11 is devoted to techniques which use only low-level attributes while making very few hypotheses about the ordering of these values. We first apply in this context the variance reduction techniques discussed in the second part of the thesis. Then, we propose a new technique to reduce bias and variance in the context of time series data. The goal of these experiments is to show that the time series classification problem is indeed difficult and its solution needs specific algorithms.

- Chapter 12, on the other hand, proposes to extend the set of candidate tests used in decision tree induction so as to make possible the explicit detection of temporal patterns in time series. The resulting algorithm gives interpretable rules but the price to pay is a decrease of accuracy with respect to dedicated variance reduction techniques.

Although they already provide interesting results, both these approaches only constitute first steps in the goal of building accurate and interpretable methods for time series classification. The main goals of the research presented in this part of the thesis is to open up the domain and give interesting directions for future research. In our conclusion we will give some idea about how we hope to extend our results to the treatment of other complex structured data in machine learning.