Prototype, Nearest Neighbor and Hybrid Algorithms for Time Series Classification

(Extended Abstract)

Christel Wisotzki, Fritz Wysotzki
Fraunhofer-Institute for Information and Data Processing, Branch Lab for Process Optimisation
Kurstraße 33, D-10117 Berlin, email: wisotzki@epo.iitb.fhg.de

1 Introduction

In the paper new methods of classification of time series and other curves are introduced. The classification learning of time series is very important, for example in medical and technical diagnosis, and for forecasting in the case of complicated processes which cannot (or only partly) be modelled mathematically. The presented examples of different real-world applications will give an impression of the varying availability of curve classification methods and of their performance.

The presented classification methods are "holistic" in the sense that distance or similarity measures are used but not description of the curves by feature vectors of fixed length. Three different methods will be adopted for the task of classification learning: construction of prototypes for each class, the use of a kNN classifier ([5], [6]), and a new principle which is called the generalized prototype classifier. The latter can be viewed as a hybrid of the first two methods, improving them in a substantial way. It is well known that, in general, the kNN classification has good results but the algorithm is very expensive. On the other hand, the prototype algorithm is very simple, but classes represented by non-convex regions (e. g. consisting of disjunctive subclasses without a common description) cannot be handled successfully by this method. The generalized prototype classifier keeps the advantages and overcomes the disadvantages of both algorithms.

In the paper, two general cases concerning the available measurements of the curves are considered leading to different preprocessing strategies:

- the measurement points are in the same interval for all curves, or
- the curves are measured on different time intervals.

In both cases, the curves are firstly approximated by spline functions (piecewise polynomial functions). For this a new method of finding the optimal intervals for the spline approximation based on a clustering technique has been developed ([5], [6]). Thus, the intervals where the curve is smoothly replaced, are adaptively defined by the approximation algorithm, depending on noise to be eliminated and the granularity of the approximation which can be controlled by the user.

Therefore, in the first case, distance measures in the functional space of the approximating functions will be used for classification.

In the second case where the measurements are taken from different time intervals, the curves are mapped onto symbol strings. The symbols can be generated by a discretization of spline parameters or directly as names of some specific curve segments (e. g. certain peaks, pieces with constant slope etc.) defined by experts. The description by symbol strings leads to further noise elimination, allows the definition of complex features like repetitions, groupings, and gives the best possibility of translation-invariant recognition (classification). Since syntactical methods for string matching

are very sensitive to noise (e. g. missing symbols), the strings are transformed into graphs by introducing the distance between symbols in the string as relations. In the case of time series, this may correspond to the time intervals between the segments which are named by the symbols. Having done this, measures of graph similarity based on the well-known Zelinka metric for graphs will be used to evaluate the similarity of curves. Besides the advantage of reducing noise sensitivity, this method uses explicitly the distance of symbols in the string as an additional information for classification.

2 Classification Methods for the Approximating Splines, Examples

The used classification methods base on similarity measures between the objects to be classified and certain reference objects, which generalize the classes. If all curves are given in the same time interval, then all approximating functions belong to the same functional space. In this case a similarity measure based on a difference in the corresponding space of the approximating functions can be used. Such similarity measures are independent of the number of subintervals of approximation.

Three classification methods are introduced. They differ from each other by the construction of the reference objects from the training examples.

The Prototype Method: In the training phase, prototypes are generated for all classes as generalizations of the classes. A prototype of a class is the average of the approximating functions of all training objects belonging to this class ([5], [6]). Therefore, all prototypes are in the mentioned functional space of approximating functions. In the test phase, an unseen object is assigned to the class to which prototype its approximating function has the smallest distance. The computational amount is proportional to the number of classes in the test phase.

The kNN Method: The approximating functions of all training objects themselves are taken as reference objects. In the test phase, the k nearest reference objects (k nearest neighbours) are determined for a new object. The latter object is assigned to that class which is the most frequent in the set of the k nearest neighbours. The computational amount in the test phase which is proportional to the number of training examples, exceeds that of the prototype method but, as will be seen below, its performance is essentially better in many cases.

The Generalized Prototype Method: Advantages of the prototype method over the kNN method include compact representation of the training data, fast classification of unseen curves and the ability to interpret the prototypes. For these reasons a generalization of the prototype method was developed. It should overcome the disadvantages and keep the advantages of the two considered methods. The prototype classifier separates two classes by a hyperplane in the functional space of the approximating functions. Therefore a bad classification result points out that the class regions in this space are not convex (e. g. a class region consists of two subclass regions). The generalized prototype classification method is based on class separation by piecewise linear surfaces which is obtained by generating more than one prototype per class. The training phase begins with prototypes for each class, consisting of one object. Now, for a given number p, the prototypes are tought incrementally by the remaining training objects in the following way: For a new object X the set of p nearest prototypes Π is determined. If there is a prototype in Π generalizing the class of X,

then this prototype is extended by X. In the other case a new prototype is opened. This procedure is repeated until the whole training set is used.
In the boundary case in which each object forms a prototype, the generalized prototype method results in the nearest-neighbor method. On the other hand, if p increases then the generalized method becomes more and more similar to the simple prototype method, i. e. there is a number p_0 for which both methods coincide.
The performance of the three classification methods is compared using some examples from different real-world applications - chemistry, physics and economics.
Example 1 (gas chromatograms [6]): In this example 44 gas chromatograms (330 points per curve), divided up into three classes (high, medium and small concentration) were given. The three-fold cross validation has been calculated.
Example 2 (TRMC signals [2]): 300 TRMC signals (500 points per curve) and two classifications (3 classes) according to the quality (Solar1) and according to the state of the measurement system (Solar2) were given. The ten-fold cross validation has been calculated.
Example 3 (Dollar Exchange Rates): 63 curves of the measured daily changes of the exchange rates of the US-Dollar over a year were given. The task was to decide wether the average of the following quarter will increase (class +) or decrease (class -). The training set consists of the first 40 and the test set of the last 23 curves.
Discussion of Results: For the described data cross validation (with the exception of Dollar) has been calculated by the NN method, the generalized prototype method (GPMp) for p = 1, 2, 3 and by the simple prototype method (PM). The mean success rates (performance) and the mean numbers of prototypes (computational amount) are represented in the following table.

Method	NN		GPM1		GPM2		GPM3		PM		default
	num	rate	num	rate	num	rate	num	rate	num	rate	rate
Chrom	29.3	1.00	5.3	0.98	4	0.93	3	0.86	3	0.86	0.68
Solar1	270	0.99	27.3	0.97	15.2	0.89	3	0.62	3	0.62	0.37
Solar2	270	0.98	27.4	0.96	15.5	0.94	3	0.69	3	0.69	0.40
Dollar	40	0.52	21	0.61	2	0.78			2	0.78	0.61

The first two examples clearly demonstrate, that the objective, formulated in section 2, has been achieved for this data. The numbers of prototypes for GPM1 are significantly smaller than the number of training objects. On the other hand, the accuracy of GPM1 differs only negligibly from that of NN. For $p_0 = 3$, the generalized and the simple prototype method coincide. The third example demonstrates, that the generalization of the prototype method has no effect if the simple prototype method gives better results than the NN classifier. But it shows the good performance of the prototype method used for the prediction of exchange rates.

3 Stringclassification

Now the assumption that all curves are given on the same time interval is dropped, i. e. the approximating functions do not belong to the same functional space. The similarity measures and the prototype building from section 2 cannot be used. To define a proper similarity measure for the kNN classification, the approximating splines are transformed into strings. There are different ways for doing it. If typical, relevant

pieces of the considered curves can be described by a set of linguistic rules, the curve can be decomposed with the help of these rules into parts. These curve parts are identified with characters of an alphabet. After that the sequence of symbols corresponds to a string.

The goal is to find a measure for the evaluation of the similarity of two arbitrary strings. Let S and T be two arbitrary strings, where the components belong to the same alphabet. S and T are converted into directed graphs the nodes of which are the characters (symbols) and the arcs between two nodes are labelled by the distances of the symbols in the string (relations). The so-called compatibility graph can be used for the construction of a similarity measure for graphs ([7]). Pairs of identical characters (one from each string) form the nodes of the compatibility graph. Two of its nodes are compatible if the corresponding characters have the same relation (distance) within their strings.

Obviously, two strings have a large similarity if the maximal induced connected subgraph (maximal clique) of their compatibility graph has a quite large number of nodes with respect to the string lengths, i. e. with the help of the cardinal number of the maximal clique a similarity measure can be defined.

The graph metric $d(S,T) = \max\{n(S),n(T)\} - n(S,T)$, $(S,T \in \mathcal{G})$, is based on the cardinal number of the maximal clique of the compatibility graph $n(S,T)$ and was introduced by Zelinka [8] and generalized by Kaden [3] and Sobik [4] to the set $\mathcal{G}$ of all finite directed labelled graphs ($n(S)$ - cardinal number of S). By this way a metric for strings is obtained, too. The methods for computing a maximal clique are NP-complete ([1]). In the present case, the task is simpler due to the fact that the compatibility graph consists of disjoint cliques.

For the curves of example 1 from section 2 linguistic rules were given. After approximating the curves, tranformating into strings, and treating by kNN, mean success rates 0.98 (for $k = 1$) and 1.0 (for $k = 3$) were obtained.

References

[1] Bron, C./Kerbosch, J.: Finding All Cliques of an Undirected Graph. Comm. ACM 1973, Vol. 17, No. 9, 575 - 577

[2] Haffner, C./ Kunst M./ Swiatkowski, C./Seidelman, G.: In-situ quality monitoring during the deposition of a-Si:H films. Applied Surface Science 63(1993) 222 - 226

[3] Kaden, F.: Graphmetriken und Distanzgraphen, in: Beiträge zur angewandten Graphentheorie, Teil 1, ZKI-Informationen 1982, Akademie der Wissenschaften der DDR, 2/82, 1 - 62

[4] Sobik, F.: Graphmetriken und Klassifikation strukturierter Objekte. in: Beiträge zur angewandten Graphentheorie, Teil 1, ZKI-Informationen 1982, Akademie der Wissenschaften der DDR, 2/82, 63 - 122

[5] Wisotzki, C./Wysotzki, F.: Feature Generation and Classification of Time Series. in: Bock, H. H., Lenski, W., Richter M. M. (eds) : Information systems and data analysis, Studies in Classification, Data Analysis, and Knowledge Organization, Springer-Verlag Heidelberg, 1994.

[6] Wisotzki, C./Wysotzki, F.: Lernfähige Klassifikation von Zeitreihen. in: 39. IWK, Band 3, Technische Universität Ilmenau, 1994

[7] Wysotzki, F.: Artificial Intelligence and Artificial Neural Nets. Proc. 1st Workshop on AI, Shanghai Jiaotong-University, Sept.1990, 116 - 122

[8] Zelinka, F.: On a Certain Distance between Isomorphism Classes of Graphs. Casopis pest. mat. 100 (1975), 371 - 373