Time Series Analysis Using Unsupervised Construction of Hierarchical Classifiers

S.A.Dolenko[1], Yu.V.Orlov[1], I.G.Persiantsev[1], A.G.Pipe[2], Ju.S.Shugai[1]

[1]Nuclear Physics Institute, Moscow State University, Vorobjovy gory, Moscow, 119899, Russia
[2]University of the West of England, Bristol, United Kingdom

dolenko@radio-msu.net

Abstract

Recently we have proposed an algorithm of constructing hierarchical neural network classifiers (HNNC), that is based on a modification of error back-propagation. This algorithm combines supervised learning with self-organisation. Recursive use of the algorithm results in creation of compact and computationally effective self-organised structures of neural classifiers. The above algorithm was expanded for unsupervised analysis of dynamic objects, described by time series. It performs segmentation of the analysed time series into parts characterised by different types of dynamics. This paper presents the latest successful results of testing the algorithm of time series analysis on pseudo-chaotic maps.

1. Algorithm for Adaptive Construction of Hierarchical Classifiers

The adaptive method of construction of HNNC for classification of static objects has been described elsewhere (Dolenko et al. 1997, Dolenko et al. 1998) and will not be described here again due to lack of space.

The distinctive features of the proposed approach are the following:

- Each node of the tree is a conventional MLP, the traditionally supervised trained architecture. Our modification of backpropagation, in fact, turns MLP training into an unsupervised mode.
- Training of the classifier is performed simultaneously with clustering, providing optimal selection of features.
- The suggested algorithm of clustering inherits all the well-known advantages of MLP, in particular, the opportunity of training by examples. In this way, *a priori* information on similarity of grouped patterns can be naturally taken into account.
- Tree structure is not set in advance. Recursive application of the algorithm provides adaptive growth of hierarchical structures.
- The parameters of the algorithm make it possible to control the topology of constructed HNNC.
- Application of this approach results in structures with high recognition rate, resistant enough to noise in test data.

The algorithm performance has been tested on model data and on real-world problems of classification of static patterns (printed letters, textures, spectrograms of isolated words and vowels), and it has been demonstrated to be effective for construction of hierarchical structures with high recognition rate (Dolenko et al. 1998).

2. Time series Analysis: Assumptions and the Main Idea

The above algorithm was expanded for unsupervised analysis of dynamic objects, described by time series (Dolenko et al. 1998).

Suppose that the hypothetical dynamic object possesses the following features:

- It has several unknown types of dynamics, while there is no *a priori* information about the types themselves and about the number of such types.
- At each moment the object is described by only one type of dynamics.
- Switching between the types of dynamics can occur at arbitrary time moments.
- The duration of switching is negligible in comparison with the intervals between switching.

The underlying idea of using the algorithm for time series analysis is the following. Time series is divided into segments of equal length, and dynamics describing each such segment is considered to be fixed. Under the assumption that switching between different types of dynamics is instant and rather rare, each segment is at first considered belonging to a separate class with its own dynamics.

Next, the algorithm of construction of HNNC described in Section 1 is applied to the analysed time series. As the result of algorithm working, each segment of time series may be reassigned to another class. Due to the ability of the algorithm to join similar classes, segments with similar or the same types of dynamics will be attributed to the same class. Note that such assignment will be done with no *a priori* information.

In spite of the above assumptions, this model may match numerous practical problems, say, EEG analysis in medicine, continuous speech segmentation, stock market analysis, etc. Some papers (Pawelzik, Kohlmorgen, and Muller 1996; Kehagias and Petridis 1997) describe neural network approaches to the analysis of time series with switching dynamics. However, in these approaches the number of neural networks is set in advance, so the whole structure of networks is non-adaptive. Also, the networks work in parallel instead of hierarchical organisation.

3. The Results of Testing on Model Data

3.1. Description of Data and the Results For the Simplest Case

The algorithm was tested on a model task. The task is classification of well-known pseudo-chaotic sequences:

logistic map, $f(x)=4x(1-x),\ x \in [0,1]$

tent map, $f(x)=2x,\ x \in [0, 0.5]$

$f(x)=2(1-x),\ x \in [0.5, 1]$

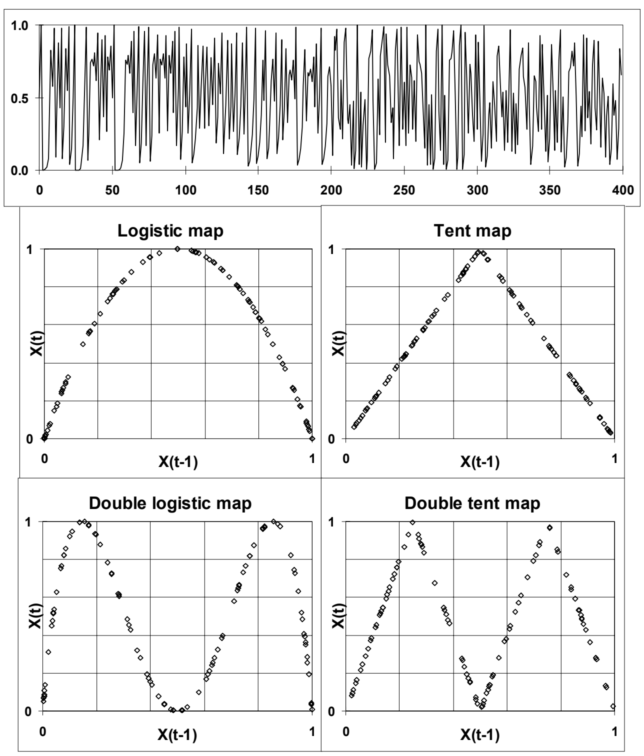


Fig.1. Top, sample pseudo-chaotic time series used in this study. Bottom, parametric maps describing the four pseudo-chaotic time sequences used to obtain the time series.

and double logistic map, and double tent map. Double logistic map is produced by recursion of logistic map, double tent map - by recursion of tent map. Fig.1 presents all the four maps.

These sequences alternated, producing from 25 to 100 points each, while the total size of a training set made from 1000 to 2000 points, depending on the statement of the experiment. Sample time series obtained as a result is also presented in Fig.1.

Each single pattern for NN training is formed of 5 sequential points from the resulted time series. The rightmost point of the window determines what class the pattern belongs to.

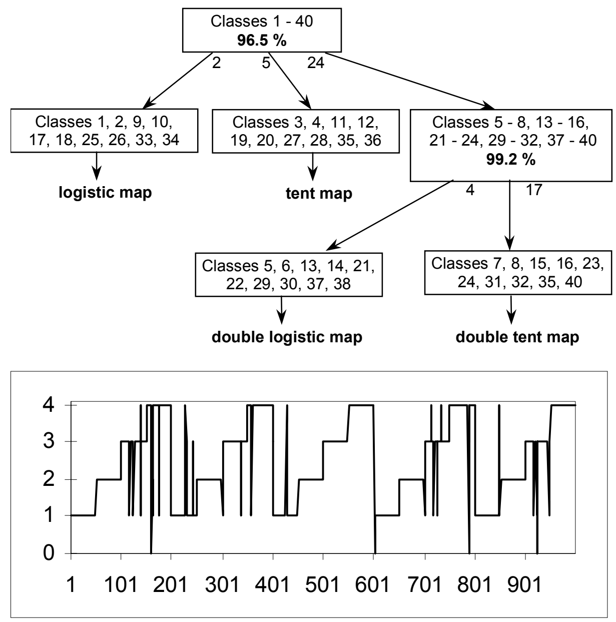


Fig.2. 1000 points (50 points/map 5 times), 40 segments 25 points each. Overall recognition rate 96.1%. Top, HNNC; bottom, class assigned vs point number.

Figure 2 presents the results obtained in one of our experiments. Each of pseudo-chaotic maps in turn produced 50 points, and this process was repeated 5 times. Thus, the total size of time series was 1000 points. This set was divided into 40 segments 25 points in each, so initially the task had 40 classes.

The HNNC constructed in this experiment is shown at the top of Fig.2. All segments of logistic map were grouped into one class at the first level of hierarchy. The same has occurred for segments of tent map. Segments of double logistic and double tent maps at the first level were all grouped together and formed the third class. Recognition rate of the classifier at this level was 96.5%.

At the next level of hierarchy, only segments of double logistic and double tent maps formed the training set, so the total number of classes was reduced to 20. At this level, segments of different chaotic maps were grouped into two different classes, and recognition rate of this node was 99.2%.

Overall recognition rate of this HNNC was 96.1%.

The chart at the bottom of Fig.2 demonstrates segmentation of the analysed time series. This chart shows the number of chaotic map each point was assigned to vs the number of point within the time series. Zero corresponds to unrecognised patterns.

In the next experiment, the size of this time series was 2000 points, and it was divided into 40 segments 50 points in each. The larger amount of points gave better results: recognition rate of the HNNC was 97.6%.

In the next experiment, the size of this time series was 2000 points, and it was divided into 40 segments 50 points in each. The larger amount of points gave better results: recognition rate of the HNNC was 97.6%.

For comparison, we have trained a conventional MLP to classify the same time series into 4 classes corresponding to the 4 chaotic maps (known *a priori*). The best recognition rate obtained in this task statement, was about 95% only.

3.2. Unbalanced Classes

We have also made a set of experiments with unbalanced classes, when some chaotic maps produced much more points than other maps.

In the next experiment (Fig.3), we took 50 points of logistic map and 350 points of each of the other maps, so the proportion was 1:7:7:7. The size of training set was 1100 patterns. This set was divided into 44 segments 25 points in each.

The tree constructed in this experiment is presented at the top of Fig.3. At the first level, all tent map segments were grouped into one class, and all other segments - into a second class. Recognition rate for this node was 99.55%.

At the second level, the number of classes was reduced to 30. All segments of logistic and double logistic maps were grouped together, and segments of double tent map were grouped into another class. Recognition rate for this node was 99.73%.

At the third level, the number of classes was reduced to 16. All segments of logistic map and of double logistic map were grouped into two different classes, and recognition rate for this node was 99.5%.

Overall recognition rate of this HNNC was 99.18%. Recognition rates for each map were the following: logistic - 98%, tent - 100%, double logistic - 98%, double tent - 99.71%.

The chart at the bottom of Fig.3 demonstrates segmentation of the analysed time series.

In the next experiment (Fig.4), we used inverse proportion: we took 900 points of logistic map and 100 points of each other maps, so the proportion was 9:1:1:1. The size of training set was 1200 patterns. This set was divided into 24 segments 50 points in each.

Overall recognition rate of this HNNC was 99.42%.

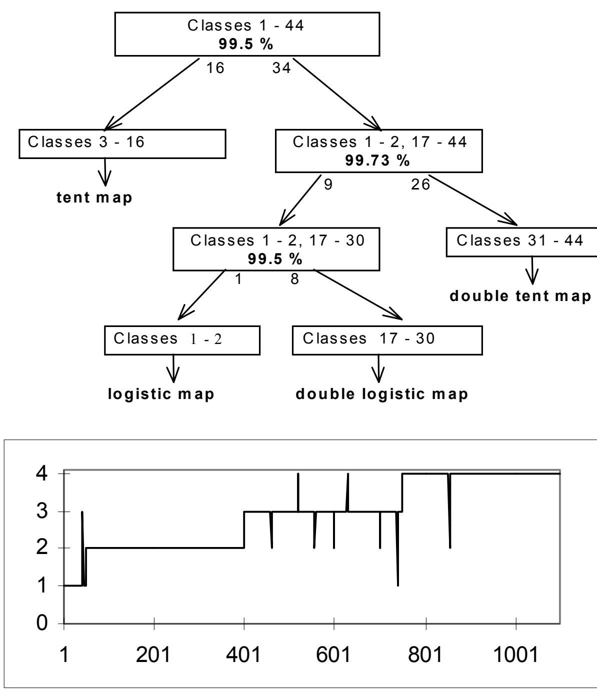


Fig.3. 1100 points. Proportion 1:7:7:7 (50 logistic map, 350 each tent map, double logistic map, and double tent map), 44 segments 25 points each. Overall recognition rate 99.18%. Top, HNNC; bottom, class assigned vs point number.

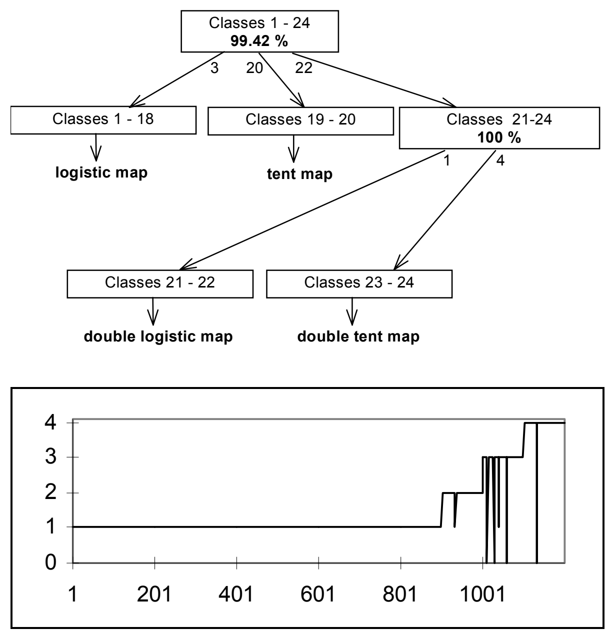
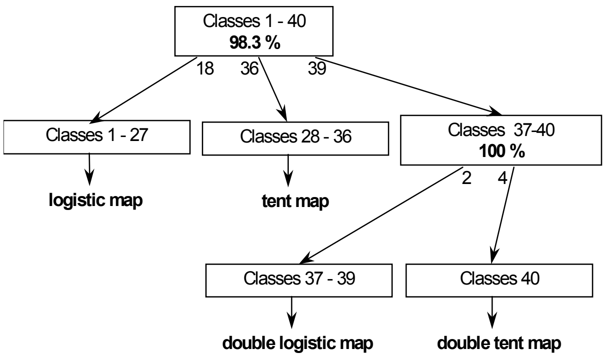


Fig.4. 1200 points. Proportion 9:1:1:1 (900 logistic map, 100 each tent map, double logistic map, double tent map), 24 segments 50 points each. Overall recognition rate 99.42%. Top, HNNC; bottom, class assigned vs point number.

The chart at the bottom of Fig.4 demonstrates segmentation of the analysed time series.

In the last experiment of such kind (Fig.5), unbalance between chaotic maps was the strongest. We took 1350 points of logistic map, 450 points of tent map, 150 points of double logistic map, and only 50 points of double tent map. Thus the proportion was 27:9:3:1. The size of training set was 2000 patterns. This set was divided into 40 segments 50 points in each.



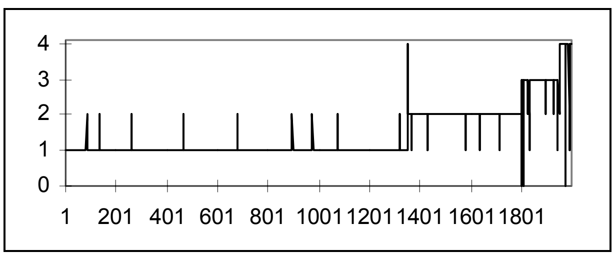


Fig.5. 2000 points. Proportion 27:9:3:1 (1350 logistic map, 450 tent map, 150 double logistic map, 50 double tent map), 40 segments 50 points each. Overall recognition rate 98.3%. Top, HNNC; bottom, class assigned vs point number.

The tree constructed in this experiment is presented at the top of Fig.5. At the first level, all logistic map segments were grouped into one class, all tent map segments - into another class, and all segments of double logistic and double tent maps were grouped into a third class. Recognition rate of the classifier at this level was 98.3%.

At the second level, the number of classes was reduced to 4. All segments of logistic and of double logistic maps were grouped into two different classes, and recognition rate for this node was 100%.

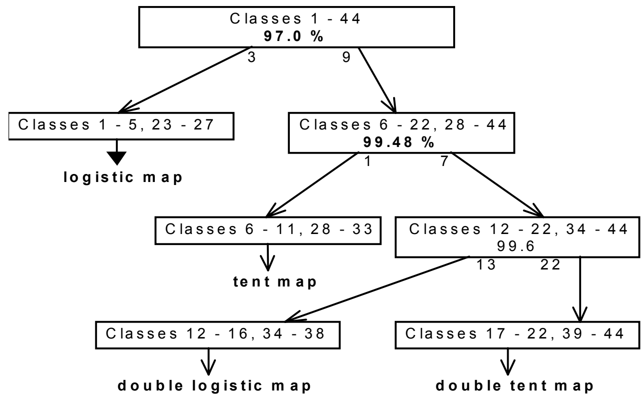
Overall recognition rate of this HNNC was 98.3%. Recognition rates for each map were the following: logistic - 99.33%, tent - 98.2%, double logistic - 91.4%, double tent - 92%.

The chart at the bottom of Fig.5 demonstrates segmentation of the analysed time series.

3.3. Uneven Class Borders

In all the above experiments, the borders of initial classes were chosen coinciding with the points where chaotic maps alternated. This is obviously a simplifying assumption. More difficult situation is when the points of chaotic maps alternation fall within some classes.

To check the work of the algorithm in this situation, we took a set of data where each of pseudo-chaotic maps in turn produced 125 points, and this process was repeated. Thus, the total size of the time series was 1000 points. This set was divided into 44 segments 23 points in each (except the last segment that included only 11 points). Therefore, initially the task had 44 classes, and 7 of them ("mixed" segments - classes 6,11,17,22,28,33, and 39) contained points produced by different maps.



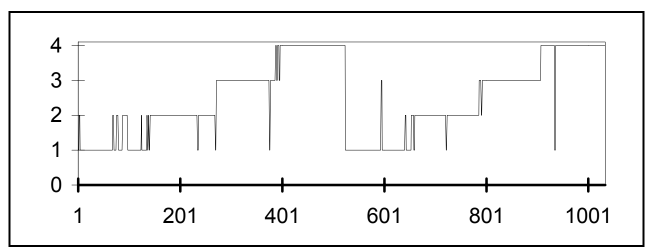


Fig.6. 1000 points (125 points/map 2 times), 44 segments 23 points each, uneven class borders. Overall recognition rate 97.0%. Top, HNNC; bottom, class assigned vs point number.

The tree constructed in this experiment is presented at the top of Fig.6. At the first level, all logistic map segments and all mixed segments in which the majority of the points belonged to logistic map, were grouped into one class, and all the other segments - into a second class. The recognition rate for this node was 97.00%.

At the second level, the number of classes was reduced to 33. All segments of tent map and all mixed segments in which the majority of the points belonged to tent map, were grouped into one class, and all the remaining segments - into a second class. The recognition rate for this node was 99.48%.

At the third level, the number of classes was reduced to 22. All segments of double logistic map were grouped into one class, and all the other segments (these of double tent map and the two remaining mixed segments; in these mixed segments the majority of points belonged to double tent map) were grouped into the other class. The recognition rate for this node was 99.6%.

Overall recognition rate of this HNNC was 96.6%. The chart at the bottom of Fig.6 demonstrates segmentation of the analysed time series.

Note that it is not surprising that the algorithm attributed each of the mixed segments to the map that was used to generate the majority of the points of this segment, as each segment is attributed using the method of simple voting of points (patterns).

So, as the result of work of the algorithm, the initial time series is divided into sections governed by different types of dynamics, and the borders between these sections are determined with the precision equal to plus or minus half of segment length. The actual border position may be anywhere within an area one segment long.

This precision can be increased to half segment (i.e., plus or minus quarter of segment) in the following way. Time series is divided into segments again, but the position of the segments is shifted half a segment aside in respect to the initial division. Then the algorithm is applied once more with the new position of segments.

We have performed such an experiment for several segments of the preceding data set, in the vicinity of the first border between logistic map and tent map (please refer to Fig.7). All segments were shifted right. The algorithm attributed the shifted segment 5 that then covered the area of the supposed border, to logistic map (the type of dynamics to the left of the border). It means that the actual border between logistic map and tent map falls within the area of segment 6 and shifted segment 5 overlap. This conclusion of the algorithm is correct, as really the segment 6 consisted of 10 points of logistic map and 13 points of tent map.

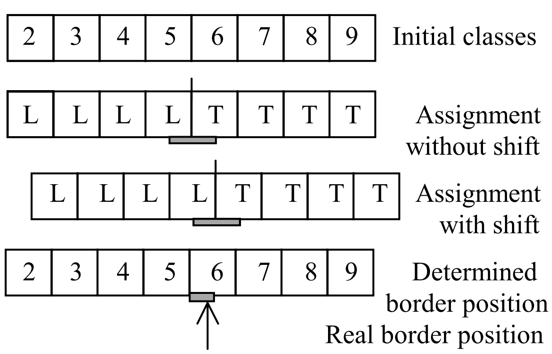


Fig.7. Increasing the precision of determination of the border between two maps. Top, some of segments for the task described by Fig.6. Second line, assignment of segments to Logistic (L) or Tent (T) maps by the structure of Fig.6. Grey thick line marks the possible placement of the border. Third line, assignment of the segments made by the algorithm after shifting all the segments half a segment aside. Last line, the grey line (half segment long) marks the area of possible placement of the border, combining the two above experiments. The arrow marks the real position of the border.

Conclusion

The recently proposed algorithm of constructing hierarchical neural network classifiers (HNNC), based on a modification of error back-propagation, has been expanded for unsupervised analysis of dynamic objects, described by time series. The modified algorithm performs segmentation of the analysed time series into parts characterised by different types of dynamics.

Testing the algorithm on pseudo-chaotic maps brought confirmation of its perspective for time series segmentation and analysis. It works successfully even in the cases of strongly unbalanced classes, and it can reveal the borders between different types of dynamics with the precision of plus or minus quarter of segment.

Future research should include application of the algorithm to real-world problems, e.g., in the study of dynamics of processes in space physics.

Acknowledgements

This work was supported in part by RFBR (Russian Foundation for Basic Research) (grant number 98-01-00537), by Ministry of Science of Russia (GNTP-PIT project number 0201.05.311), and by Program "Universities of Russia" (grant number 02-5360).

References

Dolenko, S.A., Orlov, Yu.V., Persiantsev, I.G., Shugai, Ju.S., and Eremin, E.K. 1997. The Perceptron-Based Hierarchical Structure of Classifiers Constructed by the Adaptive Method. *Pattern Recognition and Image Analysis* 7:18-22.

Dolenko, S.A., Orlov, Yu.V., Persiantsev, I.G., and Shugai, Ju.S. 1998. Construction of Hierarchical Neural Classifiers by Self-Organizing Error Back-Propagation. In: Proceedings of the International ICSC/IFAC Symposium on Neural Computation, Sept 23-25 1998, Vienna, Austria, 113-116. Ed. M.Heiss. Canada/Switzerland: ICSC Academic Press International Computer Science Conventions.

Pawelzik, K., Kohlmorgen, J., and Muller, K.-R. 1996. Annealed Competition of Experts for a Segmentation and Classification of Switching Dynamics. *Neural Computation*, 8:340-356.

Kehagias, A., and Petridis, V. 1997. Time-Series Segmentation Using Predictive Modular Neural Networks. *Neural Computation*, 9:1691-1709.