S. Singh. Neural Network Separation of Temporal Data, *Proc. IEEE International Joint Conference on Neural Networks (IJCNN'99),* Washington D.C., (10-16 July, 1999).

Neural Network Separation of Temporal Data

Sameer Singh
University of Exeter
Department of Computer Science
Exeter EX4 4PT, UK
Email: s.singh@exeter.ac.uk

Abstract

The main motivation for this research is three fold: a) temporal data classification is a challenging problem for pattern recognition tools and sophisticated tools are consistently sought to improve classifier performance on such type of data; b) traditional methods of treating temporal data often fail to adequately use temporal relationships between data points in their separation; and c) the development of neural tools based on new ideas capable of handling highly non-linear and noisy data will be of significant use in time-series, signal processing and speech applications. In this study we experiment with two types of data benchmarks: speech classification benchmark from NIST and 3D extension of the classical spiral benchmark from the Carnegie repository. It has been demonstrated in the past that ordinary neural network techniques for classification working on raw inputs of these benchmarks are inadequate. In this paper we propose a polygon method of temporal feature selection from time-dependent data and investigate neural network performance using a standard MLP architecture on such input.

1. Introduction

The separation of time-series type data is of significant importance in source separation and speech recognition. The task is to either recognise a source or a speaker at a given time on the basis of past observations of the same series. Accurate separation of sources is important for producing commercial products that can be used robustly in noisy environments. The source function generating the data is supposed to be non-parametric in nature. In order to separate two or more sources, one of the many available approaches may be adopted. The recognition system could either use the raw time-series observations for classification or perform some pre-processing for generating useful inputs that are better for classification. The raw time-series values are generally used as input for nearest neighbour methods where a sample class is allocated on the basis of the nearest neighbour in the training data. In the latter method of pre-processing, some statistical characteristics of different series could be used as inputs for a decision tree or neural network tool. These approaches suffer with the following limitations: a) the input representation does not encode temporal relationships between data points for the classification process; b) the nearest neighbour and statistical approaches are particularly weak when different sources are quantitatively similar; and c) small amounts of data is available to form meaningful statistical conclusions on the behavioural nature of given data.

2. Method

The polygon method of temporal feature selection in time-dependent data extracts temporal relationships between successive observations of a given source. Consider a source y = *f*(t) which produces observations y_1, y_2, … y_t. If we have more than one such source y_i= f(t) producing in general observations y^i_1, y^i_2, … y^i_t, then their classification on the basis of their raw observations alone is difficult. The N polygon method constructs a polygon connecting N successive points in time-series data. For each polygon, a number of features are extracted including the sine of the base angle, change in area between polygons, area at the given time and preceding observation. These features are used as input to the neural network for classification of different temporal sources. Some of the advantages of using the polygon method are: the polygon computation is based on current as well as historical information; polygon changes are more resilient to noise and individual changes in the classification process rather than changes in single observations; polygons can be optimised for size to obtain the best results; polygons quantify local as well as global characteristics of different sources. In this paper, these benefits of the polygon method of feature extraction in temporal problems will be shown on classifying data using neural network more effectively.

In the following sections we describe the two benchmarks and feature extraction methods on these. The result section demonstrates that neural networks perform better with these extracted features rather than raw data for classifying data sources.

3. NIST Benchmark

The benchmark consists of 100 observations for six deterministic non-linear sources. Each source is recorded as an amplitude observation for that source at a given time. We use the SINSIN cross-spectral analysis test data set from the NIST data library (ftp://ftp.nist.gov/pub/dataplot/other/reference/SINSIN.DAT). The data is produced by multiple sinusoidal models as follows:

$$y_1 = 10 + 5\sin(2\pi(.1)t + .8) \qquad \ldots(1)$$
$$y_2 = 20 + 3\sin(2\pi(.1)t + (.8 + 0.25\pi)) \qquad \ldots(2)$$
$$y_3 = 20 + 3\sin(2\pi(.1)t + (.8 + 0.50\pi)) \qquad \ldots(3)$$
$$y_4 = 20 + 3\sin(2\pi(.1)t + (.8 + 0.75\pi)) \qquad \ldots(4)$$
$$y_5 = 20 + 3\sin(2\pi(.1)t + (.8 + 1.0\pi)) \qquad \ldots(5)$$
$$y_6 = 20 + 3\sin(2\pi(.3)t + .8) \qquad \ldots(6)$$

where $\pi = 3.14$.

The statistical characteristics of the different sources ($y_1 \ldots y_6$) are the same except the first source (all sources have mean $\mu = 20$, variance $\sigma^2 = 2.1$. A plot of the last five sources is shown in Figure 1 (the first source is at a distinctly different amplitude level than the others and is therefore not displayed).

4. Input selection with NIST data

The temporal structure of each source is determined using a dynamic triangle method. For each successive observation, a triangle (polygon of size 3) is computed with the past two observations. The changes in the structural characteristics of the triangle describes the variation in the temporal properties of different sources. The dynamic triangle method represents the simplest of a set of methods that can be developed based on the use of polygons that can be optimised for their size that gives the best classification results. In the remaining paper, we will refer to this method as the "polygon" method. N-sided polygons can be computed for a given point by using the current plus the last N-1 observations.

Each polygon provides significant advantages over using raw observations:

- The polygon computation is based on current as well as historical information
- Polygon changes are more resilient to noise and individual changes in the classification process rather than single observations
- Polygons can be optimised for size to obtain the best results
- Polygons quantify local as well as global characteristics of different sources

We propose four features for source classification based on the polygon method:

1. The area of the polygon at a given time, ψ
2. The change in the area for successive polygons at a given time, $\Delta\psi$
3. The angle of movement φ as quantified by the value $1/(y_t - y_{t-1})$
4. The actual observation at time t, i.e. y_t

We propose that these are discriminatory features for separating time-series sources. These features will be used for NIST data. In later sections we describe a neural network based on the above inputs that solves the benchmark. We next present the SPIRAL benchmark characteristics and its feature selection process for classification. The spiral benchmark represents one step up from the simple time-series source classification. Here, at a given time, the position of the source is described by more than just one value-for a spiral lying in N dimensions, a vector of N coordinates describes spiral movement. This makes this problem both challenging and interesting to solve as detailed in the following sections.

5. Spiral Benchmark

Recognising the two spiral benchmark is a difficult task for several pattern recognition approaches since spiral data is highly non-linear. The main aim is to learn a function that is able to discriminate between two spirals. The problem has proved difficult to solve using neural approaches (see Touretzky and Pomerleau[1] for a discussion). It has been observed that backpropagation and its relatives encounter significant problems when training the neural network. In particular, deriving the optimal architecture is difficult, and furthermore, the training times are large. In addition, the spiral is under-constrained, i.e. data not lying on the spiral is often misclassified. For this reason, the two spiral problem has been particularly popular for testing novel neural and statistical pattern recognition classifiers. Considerable work has been done in the area since mid 1980s and in 1990s and a number of intelligent approaches have been applied to solving the spiral problem; neural networks:

Fahlman[2], Fahlman and Lebiere[3], Lang and Witbrock[4], Tay and Evans[5]; neurofuzzy methods: Sun and Jang[6]; and data encoding methods: Chua et al.[7], Jia and Chua[8]. In addition, several other studies have tested their proposed pattern recognition methods on this benchmark problem since this process served as an indicator of their success with real-world problems, e.g. Ulgen et al.'s[9] hypercube separation algorithm's initial success with this benchmark confirmed superior results with hand-written character recognition data. Singh[10] used a single nearest neighbour method to recognise the two spiral data. Singh[11] used a fuzzy classifier to recognise spiral data and Singh[12] studied the effect of noise contamination of various types on the recognition performance.

The two spirals can be represented in three or higher dimension. For the 3D case, each point on either spiral is characterised by {x, y, z} coordinates. The two spiral problem in 3D is shown in Figure 2. The aim of a classifier is to recognise the two spirals as distinct.

6. Input selection with Spiral data

The angle θ is calculated as the change from (x_i, y_i) to (x_{i+1}, y_{i+1}) on the same spiral, rather than the angle of a given point from the origin with respect to the x axis. Following the helixes, a close observation of Figure 2 will reveal the fact that this change in θ for the two spirals at any given time is different from each other. This is a very important observation. If we are to calculate the sine of θ, then we find that this additional information gives us the power to discriminate between the two spirals.

For a spiral in 3D, the procedure is to divide the training data into two parts representing the two classes; one for the first spiral and two for the second spiral. For each spiral, we compute the change in angle starting from the first point; if we represent spiral points as (x_i, y_i, z_i), then:

In 3D, the input features for recognition form an eight feature vector {**x**, **y**, **z**, **d**, **sin(θ_x), sin(θ_y), sin(θ_z), Ω**}, where the sine of the angle is taken with respect to all three axes between consecutive points. Here (x, y, z) are the coordinates of a given data point on the spiral, d is the distance between the given data point and previous point, Ω is the distance between the given data point and origin of the cartesian coordinates, and sin(θ) is the sine of the angle of change in helix direction.

7. Results

SINSIN DATA

The SINSIN data described in section 4 is processed for extracting a four featured input pattern (**ψ, Δψ, φ, y_t**). The output of the classifier is the class of the source; since there are six sources, the output is six valued (1/0, 1/0, 1/0, 1/0, 1/0, 1/0) where 1/0 represents an output of 1 or 0 depending on whether the input belongs to a given source or not. Hence, (1, 0, 0, 0, 0, 0) represents the first source.

A neural network is used for source classification based on the backpropagation with momentum method of learning. Neural network development primarily involves the selection of the optimal architecture, which in our case refers to optimising the number of hidden nodes. The main aim is to model a system that neither under- or over-generalises. Weiss and Kulikowski [13] recommend the following procedure:

- Start with a neural network with minimal number of hidden nodes and measure its generalisation performance with the training data
- Increase the number of hidden nodes in a step-wise manner and measure the generalisation error at each increase
- The generalisation error will decrease first and then reach a minimum at the optimal network configuration before starting to increase again.
- Choose a model with minimal complexity that gives the least generalisation error. Such a model best fits the data without under- or over-generalising.

In light of the above evidence, we choose the 4x80x6 model for SINSIN classification. We believe that provided a larger data set, the network could learn such data with fewer hidden nodes. We next show results on the classification success using cross-validation.

Fu [14] describes the cross-validation process as " *K*-fold cross-validation (Stone[15]) repeats *k* times for a sample set randomly divided into *K* disjoint subsets, each time leaving one out for testing and the others for training". The value of K = 10 is usually recommended [13]. Cross-validation requires that the original data set is split in k disjoint sets. At any one time, 90% of the data is used for training and the performance is tested on the remaining 10%. Each training process is called a 'fold'. At the end of 10 folds, all data has been tested. In every fold therefore the training and test patterns are different. The overall performance of the system may be measured using two different parameters: the average recognition rate of training data in percentage (av. R_α), and the average recognition rate of the test data in percentage (av. R_β). As expected, the latter is smaller in practice but is more important since it represents the true performance of the neural network. In our results only test performance on unseen data is shown.

Table 1 shows the results of the 10-fold cross-validation on a neural network trained with a learning rate of .01 and a momentum of .9. The overall result of 84.1% correct classification is shown as an average of the ten folds. Table 1 shows that very good classification rates are obtained by choosing appropriate inputs for the classifier. As mentioned before, these performances can be further improved with larger training sets and optimising for polygon size. In the above experiment, triangular polygons are used.

SPIRAL DATA

The SPIRAL data described in the section 5 is processed for extracting an eight featured input pattern (**x**, **y**, **z**, **d**, **sin(**θ_x**), sin(**θ_y**), sin(**θ_z**), Ω**). The target output of the classifier is the class of the spiral; since there are two spirals, the target output is 0 or 1 depending on whether the input belong to spiral of type 1 or 2 respectively.

A neural network is used for spiral classification based on the backpropagation with momentum method of learning. The optimal values of the learning rate is .1 with a momentum of .9. An optimal architecture of 7x28x2 is chosen where we have 28 hidden nodes.

The neural network is trained for convergence on a total of 194 patterns available in the Carnegie repository. The results obtained on testing unseen data is shown in Table 2 for the eight test sets (3D-1 to 3D-8). The eight test sets represent different boolean combinations of offset added to different features, e.g. 3D-1 represents ($x-\delta$, $y-\delta$, $z-\delta$) and 3D-8 represents ($x+\delta$, $y+\delta$, $z+\delta$). The offset is varied to a maximum value of 1.0 since this represents the midway distance between two spirals: it should be noted that the two spirals make approximately three revolutions in a maximum radius envelope of 6.5. A total of 80 test trials are conducted (eight test sets for each of the ten folds) and the successful recognition rate is calculated for each trial. This is simply calculated as a ratio of the patterns correctly classified to the total number of patterns tested.

The results in Table 2 are very encouraging. The main points of observation are:

i) Neural networks perform extremely well in classifying two spirals in 3D

ii) The temporal inputs to the network are highly efficient features for spiral discrimination

iii) There is a graceful degradation in performance as the noise offset is increased

We propose that the above methodology can be easily extended to recognising two or more spirals in two or higher dimensions by including the angular measurements with respect to additional dimensions introduced.

The results show that as high as 96% correct classification of the 3D spiral can be achieved with a *7x28x2* network (the distance between strands of successive spirals is about 1.1). As the offset δ is increased the recognition rate drops. The increase in δ however leads to a graceful degradation in performance and compared to the result of around 33% correct recognition achieved in training with raw input to neural networks, the results of this study are considerably encouraging.

8. Conclusion

The experimental set-up and results of this study prove that temporal feature extraction is better for classifiers than raw input. These input features are extracted from the raw data using a polygon method. The main feature of such a method is to use more than one data points for generating a polygon whose temporal changes in geometric features uniquely specifies the fingerprint of the source. Our experimentation shows that such refined data input leads to training with fewer local minima and iterations with neural networks.

References

[1] D.S. Touretzky and D. A. Pomerleau, What's hidden in the hidden layers? *Byte*; August issue:227-233, (1989).

[2] S. E. Fahlman, Faster-learning variations on back-propagation: An empirical study. In Proceedings of the 1988 Connectionist Models Summer School, Morgan Kaufmann, (1988).

[3] S. E. Fahlman, and C. Lebiere, The cascade-correlation learning architecture, In Advances in neural information processing systems 2, Touretzky, DS (ed.), Morgan Kaufmann, (1990).

[4] K. J. Lang and M. J. Witbrock, Learning to tell two spirals apart, In Proceedings of the 1988 Connectionist Models Summer School, Morgan Kaufmann, (1988).

[5] L. P. Tay and D. J. Evans, Fast learning artificial neural network (FLANN II) using the nearest neighbour recall.

Neural, Parallel and Scientific Computations vol. 2, issue 1, pp. 17-27 (1994).

[6] C. T. Sun and J. S. Jang. A neuro-fuzzy classifier and its applications, In Proceedings of the IEEE International Conference on Fuzzy Systems, vol. 1, pp. 94-98 (1993).

[7] H. Chua, J. Jia, L. Chen and Y. Gong, Solving the two-spiral problem through input data encoding, *Electronics letters*, vol. 31, issue 10, pp. 813-14 (1995).

[8] Jia, J and Chua, H. Solving two-spiral problem through input data representation, In Proceedings of the IEEE International Conference on Neural Networks, vol. 1, pp. 132-135 (1995).

[9] F. Ulgen, N. Akamatsu and T. Iwasa, The hypercube separation algorithm: a fast and efficient algorithm for on-line handwritten character recognition, *Applied Intelligence,* vol. 6, issue 2, pp.101-116 (1996).

[10] S. Singh, A single nearest neighbour fuzzy approach for pattern recognition, (in press, *International Journal of Pattern Recognition and Artificial Intelligence*, 1998).

[11] S. Singh, 2D spiral recognition using possibilistic measures, *Pattern Recognition Letters,* vol. 19, issue 2, pp. 141-147 (1998).

[12] S. Singh, Effect of noise on generalisation in Massively Parallel Fuzzy Systems, *Pattern Recognition*, vol. 31, issue 11,pp. 25-33, 1998.

[13] S. M. Weiss and C. A. Kulikowski, *Computer systems that learn*. Morgan Kaufmann, San Mateo, CA (1991).

[14] L. Fu, *Neural networks in computer intelligence*, McGraw-Hill, Singapore, pp. 331-348, (1994).

[15] M. Stone, Cross-validatory choice and assessment of statistical predictions, *Journal of the Royal Statistical Society,* vol. 36, issue 1, pp. 111-147 (1974).

Table 1. Ten fold cross-validation results for SINSIN data classification.

Fold	Success rate %
1	85
2	89
3	80
4	84
5	90
6	87
7	82
8	84
9	80
10	80
Average	84.1%

Table 2. 3D Spiral Recognition Rate % for test data as a function of the Offset δ. The recognition Rate *R* is in percentage.

Offset δ	*3D-1*	*3D-2*	*3D-3*	*3D-4*	*3D-5*	*3D-6*	*3D-7*	*3D-8*
.1	96	97	97	96	98	97	97	94
.2	92	96	91	87	93	88	97	84
.3	78	95	78	73	80	73	94	73
.4	69	90	64	56	64	58	89	62
.5	58	85	51	48	55	49	83	52
.6	45	75	47	45	50	43	74	43
.7	40	70	43	45	46	40	67	45
.8	39	64	42	48	45	41	61	46
.9	46	57	44	51	49	45	60	47
1.0	49	56	51	56	52	51	60	49

Figure 1. Data sources in the SINSIN data: only the first 25 points are plotted

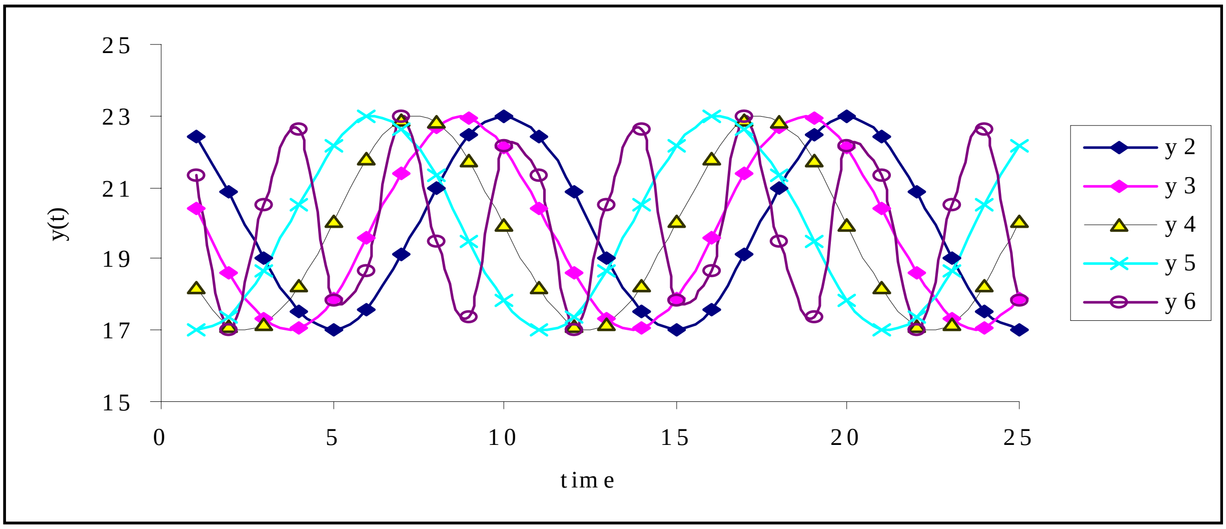
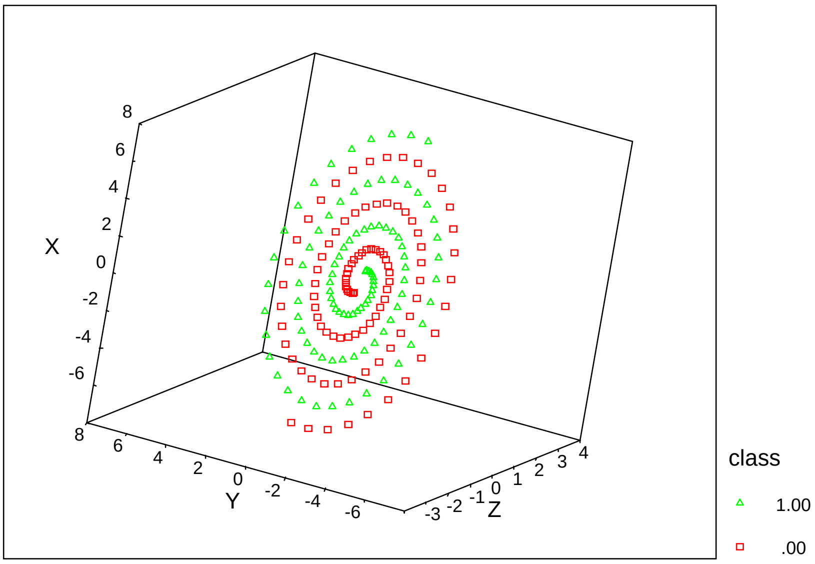


Figure 2. Two spiral problem in 3D


X
Y
Z
class