Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®
Neurocomputing 55 (2003) 779–786

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Letters

A novel pattern classification scheme using the Baker's map

Alan Rogers[a,*], John Keating[b], Robert Shorten[c]

[a]*Department of Electronic Engineering, NUI Maynooth, Maynooth, Co. Kildare, Ireland*
[b]*Department of Computer Science, NUI Maynooth, Maynooth, Co. Kildare, Ireland*
[c]*Hamilton Institute, NUI Maynooth, Maynooth, Co. Kildare, Ireland*

Abstract

We demonstrate a novel application of nonlinear systems in the design of pattern classification systems. We show that pattern classification systems can be designed based upon training algorithms designed to control the qualitative behaviour of a nonlinear system. Our paradigm is illustrated by means of a simple chaotic system—the Baker's map. Algorithms for training the system are presented and examples are given to illustrate the operation and learning of the system for pattern classification tasks.

Keywords: Pattern classification; Chaos Baker's map; Lyapunov dimension

1. Introduction

Conventional pattern classification systems are usually constructed by manipulating the parameters of some nonlinear function; the parameters are chosen such that the output of the function attains prescribed values for classes of input signal [3]. Although static functions are usually chosen for the design of pattern classification systems, nonlinear dynamic systems can also be used for the design of such systems (e.g. Hopfield neural networks). In this letter, we argue that effective pattern classification systems can be designed by manipulating the parameters of nonlinear dynamic systems such that the qualitative behaviour of the function, rather than the steady-state output, acts as a signature of stored patterns. Potential advantages of this approach include: compact

* Corresponding author.
E-mail address: alan.rogers@may.ie (A. Rogers).

0925-2312/$ - see front matter
doi:10.1016/S0925-2312(03)00441-7

representation of stored patterns; increased storage capacity; and the possibility of exploiting the qualitative behaviour of a function for specific applications. The objective of this letter is not to investigate these potential advantages, but rather to illustrate that controlling the qualitative behaviour of nonlinear systems provides a feasible basis for the design of pattern classification systems. Specifically: (i) we show that a typical chaotic nonlinear system, the Baker's map [2], can be used as the basis for the design of pattern classification systems; (ii) we present algorithms for training our Baker's map pattern classification system; (iii) we present examples to illustrate the efficacy of our paradigm.

2. The Baker's map and Lyapunov dimensions

In their classic study of fractal dimensions, Farmer et al. [4] re-introduced the Baker's map [2]. It is a transformation of the unit square $[0,1]\times[0,1]$, and has three parameters, R_1, R_2 and S:

$$x_{n+1} = \begin{cases} R_1 x_n & \text{if } y_n < S, \\ 1/2 + R_2 x_n & \text{if } y_n \geqslant S, \end{cases}$$

$$y_{n+1} = \begin{cases} y_n/S & \text{if } y_n < S, \\ \dfrac{y_n - S}{1-S} & \text{if } y_n \geqslant S. \end{cases} \tag{1}$$

Chaotic maps, such as the Baker's map, are deterministic systems whose long-term behaviour is unpredictable. Despite this uncertainty in behaviour, there are quantitative measures of chaos which can be computed. One such measure is the Lyapunov dimension (D_L) of the chaotic attractor. This is a particular type of fractal dimension, and is related to the average rates of expansion and contraction of the map. From Fig. 1, it can be seen that repeated iteration of the map leads to an attractor, which is the union of a line and a Cantor set, and thus the fractal dimension (Lyapunov

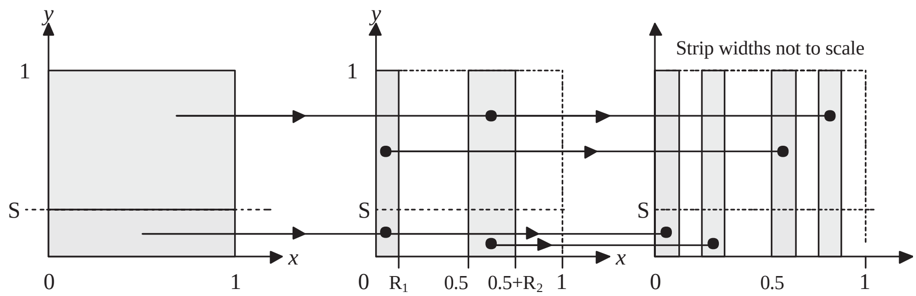

Fig. 1. Action of Baker's map on unit square: transforms square into two strips, then four strips, eight strips, and so on.

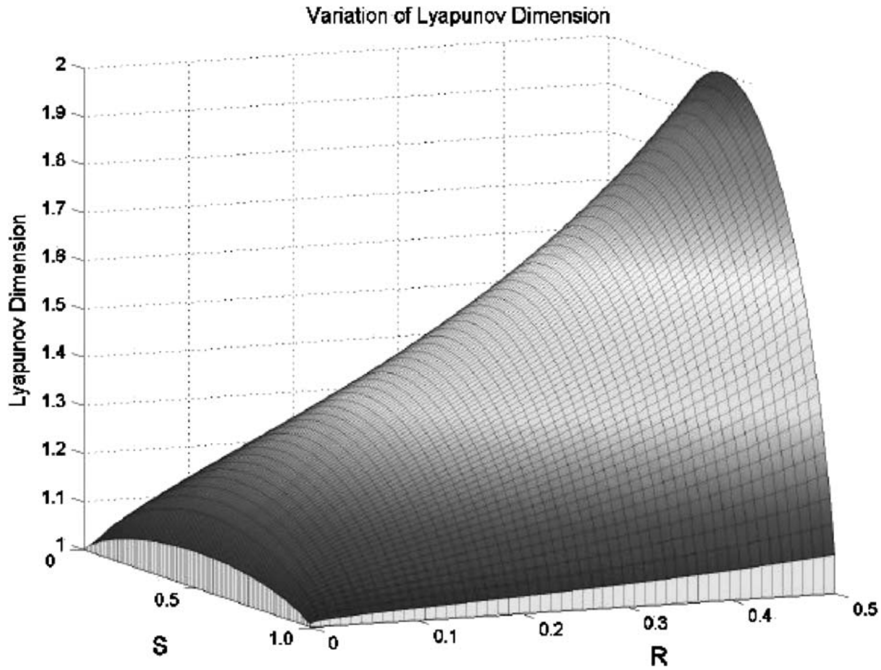

Fig. 2. Variation of Lyapunov dimension with parameters R and S.

dimension) must lie in the range $1 \leqslant D_L \leqslant 2$, depending on the choice of parameter values. The so-called Lyapunov numbers λ_x and λ_y characterise the stability of the map, and are defined as follows:

$$\log \lambda_x = S \log R_1 + (1 - S) \log R_2,$$

$$\log \lambda_y = S \log \frac{1}{S} + (1 - S) \log \frac{1}{1 - S}. \qquad (2a,2b)$$

The Lyapunov dimension D_L was introduced by Kaplan and Yorke [6] and for the Baker's map,

$$D_L = 1 + \frac{\log \lambda_x}{\log 1/\lambda_y}. \qquad (3)$$

For the purposes of 2-bit pattern classification, we require only two parameters, and so let $R_1 = R_2 = R$. The variation of Lyapunov dimension with R and S is shown in Fig. 2. It has the useful property that it is monotonically increasing for $R, S \in (0, 5)$. From Fig. 3, it can be seen that many different parameter pairs, or patterns, will lead to the same Lyapunov dimension, for example, in the figure, $(R, S) = \{(0.1, 0.5), (0.36, 0.1)\}$ correspond to the same Lyapunov dimension D_A (note the Lyapunov dimension D_L varies between 1 and 2 depending on the values of R and S, as can be seen in Fig. 2.

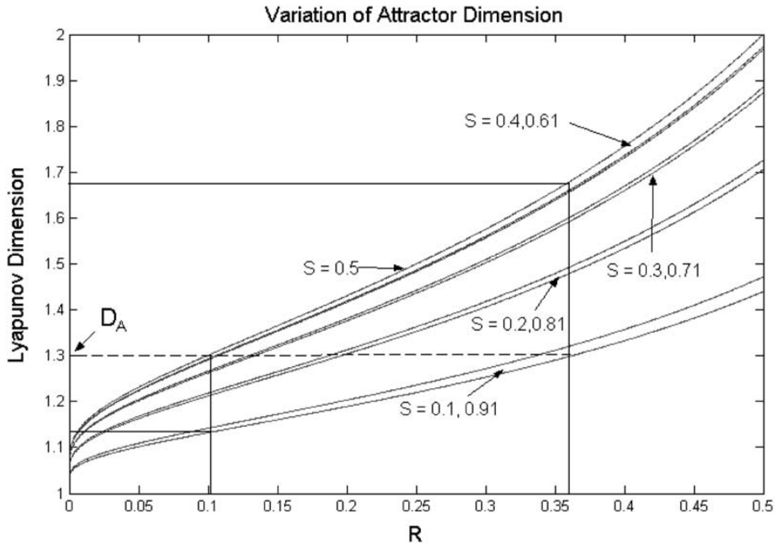

Fig. 3. Side-view of Fig. 2 showing two different sets of parameters having the same Lyapunov dimension D_A.

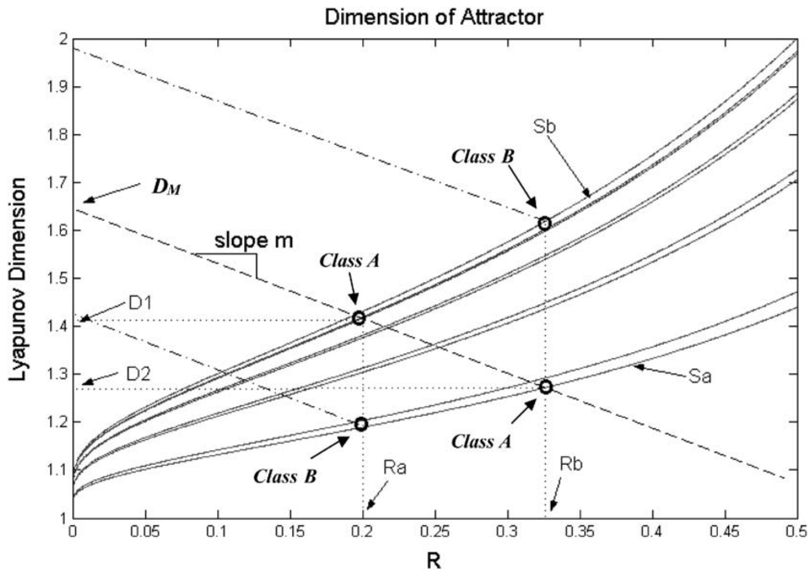

Fig. 4. A more general way of solving the XOR problem: draw a straight line through the two points belonging to class A (say), and find where the line intersects the y-axis.

For a general-purpose XOR-type pattern classifier, described later, we use a linear transformation to map one of the classes onto the same fractal dimension, which we call D_M, the modified Lyapunov dimension as illustrated in Fig. 4. For different types of class arrangement, it is necessary to find some transformation which will separate the classes, so that for instance, patterns in class A lies below some value of D_{M}, and patterns in class B lie above. We illustrate a training algorithm in the next section.

3. Training

Training algorithms for the Baker's map system involves estimating the parameters of the linear mapping onto D_{M} for a given set of inputs and associated class labels. In principle, a number of standard paradigms from the statistical pattern recognition and neural network literature could be used as the basis for a training algorithm for our system [3,5]. Here, we present a modified simulated annealing training algorithm consisting of the following steps [8,7]:

```
Initialise system parameters to state X_0
Initialise annealing parameter T
    Repeat until
     { classification error is below threshold OR
       annealing parameter T is below annealing threshold
        {
         Repeat for large number of iterations
           {
              Perturb system to new state X_i;
              Determine change in cost(s) ΔE_j = E_j(X_i) − E_j(X_{i−1});
              If ΔE_j < 0
                   Accept X_i;
              else if ΔE_j > 0
                   Accept X_i with probability e^{−ΔE_j T};
              end;
           }
        }
        Reduce annealing parameter T;
     }
```

To overcome difficulties associated with adopting a single cost function for general classification tasks, we utilize a technique from the multiple-models and switching literature [9]: we specify a number of cost functions, one of which is guaranteed to converge for the unknown classification task. Each cost function is evaluated online, using training and test data sets, and the algorithm is terminated when one of the cost functions converges or falls below some pre-specified threshold. In the next section we present an example to illustrate our algorithm.

4. XOR pattern classification task

We demonstrate the pattern classification system with a simple XOR-type classification task. It is necessary to assign values to R and S, representing high and low. We arbitrarily choose the following values (though limiting both R and S to $(0, 0.5)$). The training procedure is carried out on a line in the R-D_L. plane, so we find the corresponding values of Lyapunov dimension D_L for each (R, S) pair. This is possible as there exists a closed-form expression for D_L in terms of R and S [4],

$$D_L = 1 - \frac{S\ln[1/S] + (1-S)\ln[1/(1-S)]}{\ln R}. \qquad (4)$$

Binary pattern	R-value	S-value	D_L
0 0	0.2	0.2	1.3109
0 1	0.2	0.4	1.4182
1 0	0.4	0.2	1.5461
1 1	0.4	0.4	1.7345

We now apply the simulated annealing algorithm to find a line, which will classify the patterns correctly, as shown in Fig. 5. Finally, we apply test patterns to the Baker's map, to verify that the correct patterns are detected. The two cost functions used are as follows, where d_i represent the perpendicular distances from the patterns to the line (a) $E = \sum_{\text{classAB}} \tanh(d_i)$ (b) $E = \sum_{\text{classAB}} \tanh(d_i)^2$.

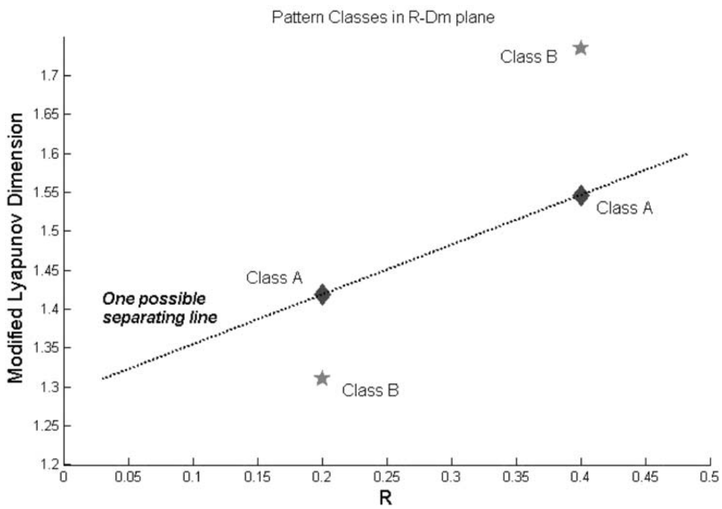

Fig. 5. Patterns/classes in R–D_L plane, and a possible separating line. In practice, each diamond or star can represent a cluster of patterns.

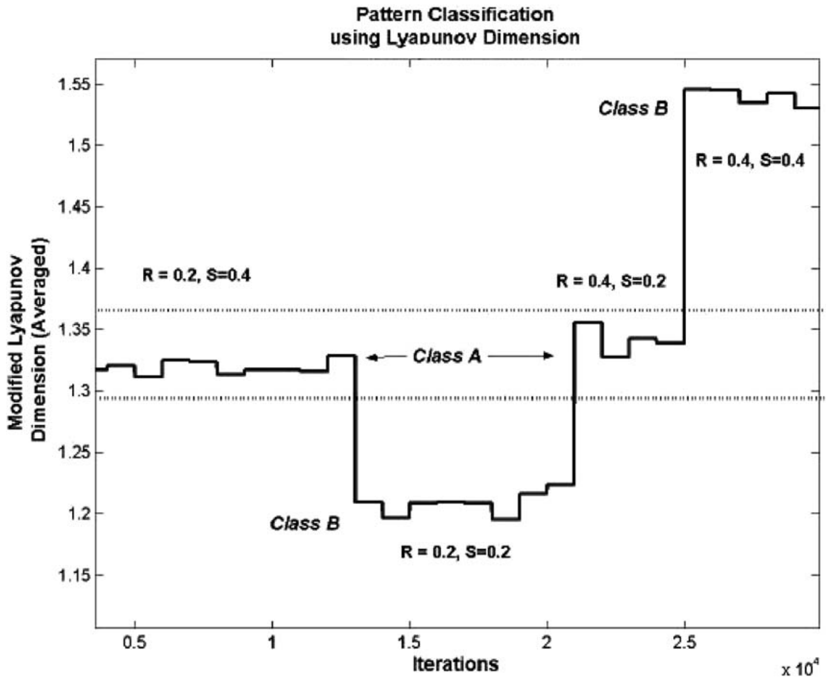

Fig. 6. Patterns between the dotted lines belong to class *A*. We can classify the patterns simply by looking at their modified Lyapunov dimension.

When we apply the simulated annealing algorithm, we find that the slope $m = 0.487$ and y-intercept $= 1.34$. This information is used to separate the classes, as shown in Fig. 6. There is some expected variation in Lyapunov numbers, (and hence the Lyapunov dimension) due to the nature of the map (see Eq. (1)). We find average values of the Lyapunov dimension, using a 500-point averaging window. In Fig. 6, we apply the four possible patterns to the system, as inputs. After modifying the Lyapunov dimension, the points lying below the separating line correctly correspond to the pattern Class B.

5. Conclusions

We have shown that certain properties of chaotic mappings can be utilized to create a simple pattern classification system. Our system has been based on a purely chaotic dynamical system, whereas some other researchers have found uses for nonlinear maps not necessarily operating in a chaotic regime [1]. As the study of nonlinear dynamics reaches a state of comparative maturity, it is hoped that other innovative applications of chaos will be found.

References

[1] Y.V. Andreyev, A. Dmitriev, S.O. Starkov, Information processing in 1-D systems with chaos, IEEE CAS-I Fund. Theory Appl. 44 (1997) 21–28.

[2] J. Balatoni, A. Renji, Publ. Math. Inst. Hungarian Acad. Sci. 1 (9) (1956) 9– 40.

[3] R. Duda, P. Hart, D. Stork, Pattern Classification, Wiley, New York, 2001.

[4] J.D. Farmer, E. Ott, J.A. Yorke, The dimension of chaotic attractors, Physica D 7 (1983) 153–180.

[5] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, New York, 2001.

[6] J. Kaplan, J. Yorke, Chaotic behaviour of multidimensional difference equations, in: H.O. Peitgen, H.O. Walther (Eds.), Functional Differential Equations and the Approximations of Fixed Points, Lecture Notes in Mathematics, Vol. 730, Springer, Berlin, 1979, pp. 204–207.

[7] J.G. Keating, D. Noonan, The structure and performance of trained Boolean networks, in: G. Orchard (Ed.), Neural Computing: Research and Applications, Vol. II (Irish Neural Networks Association, Belfast, 1994, pp. 79–86).

[8] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.

[9] K.S. Narendra, Neural networks for control theory and practice, IEEE Proc. 84 (1996) 1385–1406.