



Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/84834/>

Monograph:

Zhao, Y. and Billings, S.A. (2005) Neighbourhood Detection Using Mutual Information for the Identification of Cellular Automata. Research Report. ACSE Research Report 881 . Department of Automatic Control and Systems Engineering

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



Neighbourhood Detection Using Mutual Information for the Identification of Cellular Automata

Y. Zhao and S.A. Billings



Research Report No. 881

Department of Automatic Control and Systems Engineering
The University of Sheffield
Mappin Street, Sheffield,
S1 3JD, UK

January, 2005

Neighbourhood Detection Using Mutual Information for the Identification of Cellular Automata

Y.Zhao S.A.Billings

Department of Automatic Control and System Engineering, University of Sheffield, UK, S1 3JD

Abstract

Extracting the rules from spatio-temporal patterns generated by the evolution of Cellular Automata (CA) usually requires *a priori* information about the observed system, but in many applications little information will be known about the pattern. This paper introduces a new neighbourhood detection algorithm which can determine the range of the neighbourhood without any knowledge of the system by introducing a criterion based on Mutual Information (MI) and an indication of over-estimation. A coarse-to-fine identification routine is then proposed to determine the CA rule from the observed pattern. Examples, including data from a real experiment, are employed to evaluate the new algorithm.

Key words: Cellular Automata, Identification, Mutual Information

1 Introduction

Cellular Automata (CA) are a class of spatially and temporally discrete mathematical systems characterized by local interactions. Because of the simple mathematical constructs and distinguishing features, CA have been widely used to model aspects of advanced computation, evolutionary computation, and for simulating a wide variety of complex systems in the real world [1], [17], [18] and [21].

In many applications the resulting CA pattern can be observed but the underlying CA rule is unknown. This could be true for example when dealing with

Email addresses: y.zhao@shef.ac.uk (Y.Zhao), s.billings@shef.ac.uk (S.A.Billings).

some natural systems. The key problem the observer faces is to understand how the system works, this involves identifying the rule and then using the identified model of the system to predict the output. The theory of how CA rules can be extracted from observed patterns of spatio-temporal behavior is therefore fundamental to the study of CA. Essentially, this is an inverse problem, which means that the order of cause and effect is reversed: the observer knows the effects instead of the causes and tries to deduce the CA rule. Often solving inverse problems is difficult because the problem itself can be ill posed. Because of these difficulties determining how the transition rules can be extracted from observed patterns of spatio-temporal behavior has attracted few investigations, but if this problem can be solved, many applications may benefit from it.

Adamatzky presented a sequential and parallel algorithm to determine the local CA transition table [2], and introduced a genetic programming solution with automatically defined functions, to evolve a rule for the majority classification task for one-dimensional CA's [3]. However, because most CA use either the von Neumann, the Moore or larger neighbourhoods in order to model systems with long-range interactions, the number of potential rules can be very large and the application of the above algorithms can become very complicated and time consuming. For example, a three-site one dimensional CA will have $2^{2^3} = 256$ possible rules while the number of possible rules will explode to $2^{2^9} = 1.3e + 154$ for a nine-site one dimensional CA. It is therefore often very difficult to scan all the possible rules even using modern computers. To simplify the problem, Yang and Billings showed that CA binary rules can be represented by a simple polynomial model [5]. Using this important observation, the number of parameters to be identified can be substantially reduced, for example, from $1.3e + 154$ to 512. Based on the polynomial model form, the identification of CA can be divided into two parts: neighbourhood detection and parameter estimation. It is often straightforward to estimate the unknown parameters once the correct neighbourhood has been determined, and therefore in this paper we will concentrate on the neighbourhood detection problem.

In this paper a new neighbourhood detection algorithm is introduced based on mutual information to provide an initial indication of the temporal and spatial range in the identification of CA. This initial neighbourhood is then used to prime a CA-OLS (Cellular Automata - Orthogonal Least Squares) algorithm to find the correct model terms and unknown parameters in a CA model. This provides a coarse-to-fine identification approach for CA where the mutual information is used to significantly reduce the potential neighbourhood choices which are then optimised using the CA-OLS identification algorithm.

The paper is organized as follows. The structure of CA and an overview of previous investigations are introduced in section 2. The new neighbourhood detection algorithm based on Mutual Information is proposed in section 3 together with a coarse-to-fine identification routine for CA. Three examples, including two simulation studies and a real system, are demonstrated in sec-

tion 4, and conclusions are given in section 5.

2 Cellular Automata

A Cellular Automata is composed of three parts: a neighbourhood, a local transition rule and a discrete lattice structure. The local transition rule updates all cells synchronously by assigning to each cell, at a given step, a value that depends only on the neighbourhood. Relatively simple binary CA rules can produce highly complex patterns of behaviour, although only two values can be taken in each cell. Because of the advantages, binary CA have attracted many authors to investigate the properties of this class of systems. In the present paper, only binary CA will be considered.

Based on the properties of the transition rule, Cellular Automata can be classified into two types: deterministic Cellular Automata (DCA) and probabilistic Cellular Automata (PCA). The transition rules for DCA are deterministic while those for PCA are statistical because some state flipping of cell values occurs during the evolution of the PCA. Due to the presence of noise, which causes state switching of cell entries during the CA evolution, it is more difficult to identify the rule of PCA than that of DCA. Therefore, the identification of PCA is often investigated independently from the identification of DCA.

2.1 CA Neighbourhoods

The neighbourhood of a cell is the set of cells in both spatial and temporal dimensions that are directly involved in the evolution of the cell. Sometimes this includes the cell itself. The neighbourhood structure varies depending on the construction of the CA. Consider a one-dimensional 3-site CA for example where the cell at position j and at time step t is denoted as $c(j; t)$. The neighbourhood of cell $c(j; t)$ could then be a von Neumann neighbourhood or other exotic neighbourhoods, some of which are shown in Fig. 1 where the black cells denote the cells $c(j; t)$ and the gray cells denote the neighbourhoods. The *Exotic 1* neighbourhood shown in Fig. 1 encompasses cells from the same temporal scale but different spatial scale than the cells in the von Neumann neighbourhood, while the neighbourhood of *Exotic 2* involves cells from two temporal scales. There are many possible neighbourhood structures for two-dimensional CA. The most common structures are the von Neumann neighbourhood, the Moore neighbourhood and the Hexagonal neighbourhood shown in Fig. 2 respectively.

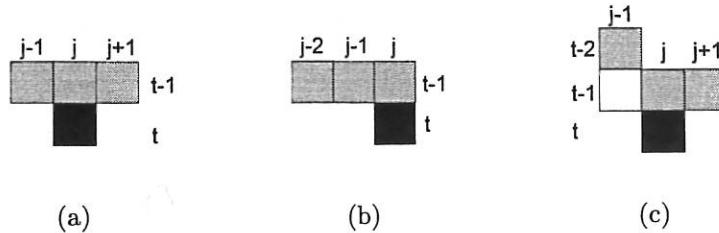


Fig. 1. Three examples of a 3-site neighbourhood for a one-dimensional CA: (a) von Neumann: $\{c(j-1; t-1), c(j; t-1), c(j+1; t-1)\}$; (b) Exotic 1: $\{c(j-2; t-1), c(j-1; t-1), c(j; t-1)\}$; (c) Exotic 2: $\{c(j-1; t-2), c(j; t-1), c(j+1; t-1)\}$

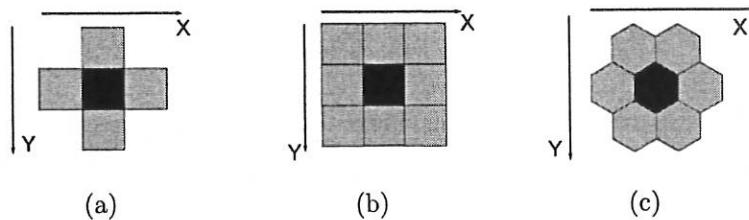


Fig. 2. Three examples of the neighbourhood type for a two-dimensional CA: (a) von Neumann; (b) Moore ; (c) Hexagonal

2.2 Polynomial Form of CA Rules

In the present study, the problem of searching for the neighbourhood and the parameter values associated with a nonlinear logical model will initially be mapped into an equivalent polynomial representation, which was proposed by Yang and Billings [5]. By using the polynomial CA model and using a Cellular Automata Orthogonal Least Square (CA-OLS) routine both the CA neighbourhood and the unknown polynomial model parameters can be determined from observed patterns.

The local rule for a deterministic binary CA can be considered as a Boolean function of the cells within the neighbourhood. Consider a one-dimensional n -site CA for example. Denoting the cell at position j and at time step t as $c(j; t)$, then the neighbourhood of $c(j; t)$ can be expressed as $\{c(j - a_1; t - b_1), \dots, c(j - a_n; t - b_n)\}$, where $a_i (i = 1, \dots, n)$ and $b_i (i = 1, \dots, n)$ denote the spatial and temporal scales of the i th neighbourhood respectively. The Boolean function of $c(j; t)$ can then be expressed in the form

$$c(j; t) = \vartheta_0 \oplus \vartheta_1 c(j - a_1; t - b_1) \oplus \dots \oplus \vartheta_N (c(j - a_1; t - b_1) \times \dots \times c(j - a_n; t - b_n)) \quad (1)$$

where $\vartheta_i (i = 0, \dots, N, N = 2^n - 1)$ are binary numbers and $\vartheta_i = 1$ indicates that the corresponding term is included in the Boolean function while $\vartheta_i = 0$ indicates that the corresponding term is not included. The symbol \oplus denotes the *XOR* operator.

Equation (1) can be transformed into a polynomial expression [5] by expanding the logical operators and applying the Principle of Duality and Absorption to all the terms [5], [7]. Equation (1) then can be expressed as

$$c(j; t) = \theta_0 + \theta_1 c(j - a_1; t - b_1) + \dots + \theta_N (c(j - a_1; t - b_1) \times \dots \times c(j - a_n; t - b_n)) \quad (2)$$

where $\theta_i (i = 0, \dots, N, N = 2^n - 1)$ represent the unknown integer parameters. Now the problem has been mapped into a linear-in-the-parameters model, and the identification of the CA rule is translated into determining the structure of this model and estimating the unknown parameters.

2.3 Previous Studies on Neighbourhood Detection

In most former studies, the neighbourhood was manually predefined as the cells that were immediately close to the cell to be updated. Richards directly selected the Moore structure as the neighbourhood of the pattern generated by dendritic solidification [8]. Adamazky set a minimal neighbourhood before the identification of a one-dimensional CA [2]. But for most systems, especially higher order CA, it will often be very difficult to manually choose a candidate neighbourhood that just covers the exact neighbourhood and rejects the many possible redundant cells. The potential combinatorial possibilities for the neighbourhood are large and neighbourhood detection is a key step in CA identification.

Based on the polynomial equation (2), Yang and Billings proposed a modified orthogonal least squares algorithm, abbreviated as CA-OLS, to detect the neighbourhood and estimate the parameters in CA models [4]-[5]. The preliminary step in this algorithm involves choosing a candidate neighbourhood, which can be coarse but must include all correct neighbourhoods. Consider a one-dimensional CA for example and assume the neighbourhood of the cell $c(j; t)$ is chosen as $\{c(j - 1; t - 1), c(j; t - 1), c(j + 1; t - 1)\}$, a polynomial model, expressed as equation (3), can be generated according to equation (2).

$$\begin{aligned} c(j; t) = & \theta_0 + \theta_1 c(j - 1; t - 1) + \theta_2 c(j; t - 1) + \theta_3 c(j + 1; t - 1) \\ & + \theta_4 c(j - 1; t - 1)c(j; t - 1) + \theta_5 c(j - 1; t - 1)c(j + 1; t - 1) \\ & + \theta_6 c(j; t - 1)c(j + 1; t - 1) + \theta_7 c(j - 1; t - 1)c(j; t - 1)c(j + 1; t - 1) \end{aligned} \quad (3)$$

Determining which terms are significant and which terms are redundant can be derived directly from the Error Reduction Ratio (ERR), which measures the contribution of each candidate term to the updated cell and which is part of the CA-OLS routine. Using ERR the candidate terms can be ranked in order of importance and the insignificant terms can then be discarded.

Mei and Billings [9] recently proposed a new neighbourhood detection routine, which can refine the candidate neighbourhood according to a statistic associated with each combination of candidate neighbourhood cells and the updated cell, from which an exact neighbourhood can be obtained.

3 Neighbourhood Detection Based on Mutual Information

Most previous algorithms for the identification of CA, involve the preliminary step of either guessing the candidate neighbourhood or performing an initial determination of the candidate neighbourhood. The candidate set must be large enough to cover all potential neighbourhoods, but too large a range will often involve massive computational time in the subsequent identification. There is a clear tradeoff between computational time and productivity of the model. Hence, the determination of the candidate neighbourhood becomes crucial and a failure to select an appropriate neighbourhood will often produce an incorrect CA model.

In this paper a new neighbourhood detection approach based on Mutual Information is introduced to detect a coarse range for the CA neighbourhood without any *a priori* neighbourhood information regarding the observed system.

3.1 Mutual Information

Mutual Information (MI), which was initially proposed by Shannon in 1948 [10], is a measurement of the dependence between two variables. If the two variables are independent, the MI between them is zero. If the two variables are strongly dependent, then the MI between them will be close to 1. Mutual Information can measure the state predictability or the memory of a system, represented by a sequence of certain symbols.

Let x be the random variable uniformly chosen from $\{a_1, \dots, a_m\}$ and let y be the random variable chosen from $\{b_1, \dots, b_n\}$. Then, the Mutual Information of x and y , written $I(x; y)$ can be defined by

$$I(x; y) = \sum_{i=1}^m \sum_{j=1}^n p_{a_i, b_j} \log_2 \frac{p_{a_i, b_j}}{p_{a_i} p_{b_j}}$$

where p_{a_i} is the probability when $x = a_i$, p_{b_i} is the probability when $y = b_i$ and p_{a_i, b_j} is the joint probability when $x = a_i$ and $y = b_j$.

3.2 A New Neighbourhood Detection Approach using Mutual Information

Consider the one-dimensional CA case to illustrate the approach and assume the neighbourhood of the cell $c(j; t)$ is $\{c(j - a_1; t - b_1), \dots, c(j - a_n; t - b_n)\}$. The aim is to determine the maximal spatial lag a_n and the maximal temporal lag b_n .

Definition 1 A *case* is defined as a pair of $\{f(R\{j; t\}), c(j; t)\}$, where $R\{j; t\}$ is the neighbourhood of a cell $c(j)$ at time step t and the $c(j; t)$ is the state value of the cell at time step t , and $f(R\{j; t\}) = c_1 + 2c_2 + \dots + 2^{m-1}c_m$ assuming $R\{j; t\} = \{c_1, c_2, \dots, c_m\}$. For example, if the state value of the updated cell $c(j; t)$ is 1 and the state of its neighbourhood $R\{j; t\}$ is $\{0, 1, 1\}$, the *case* can be described as $\{5, 1\}$.

Essentially, $R\{j; t\}$ represents the input and the $c(j; t)$ represents the output of a nonlinear system. If the candidate neighbourhood $R\{j; t\}$ is large enough to cover all the correct neighbourhoods, the Mutual Information between $f(R\{j; t\})$ with $c(j; t)$ should be close to 1. If $R\{j; t\}$ can not contain all the correct neighbourhoods, the Mutual Information between $f(R\{j; t\})$ with $c(j; t)$ will be close to 0. According to this rule, this paper presents a new criteria, which introduces MI as a fitness function to establish a measurement for ranking each candidate neighbourhood.

Assume the number of component cells of the candidate neighbourhood $R\{j; t\}$ is m and the number of the sampled *cases* is N , the new criteria can be expressed as

$$F = I - OE = \sum_{i,j} p_{k_i, l_j} \log_2 \frac{p_{k_i, l_j}}{p_{k_i} p_{l_j}} - \frac{2^m}{N} \quad (4)$$

where $k_i \in \{0, \dots, 2^m - 1\}$ and $l_i \in \{0, 1\}$. In expression (4) p_{k_i} denotes the possibility when $f(R\{j; t\}) = k_i$, p_{l_i} denotes the possibility when $c(j; t) = l_i$ and $f(R\{j; t\})$ is given in from *Definition 1*.

The MI is *over-estimated* when the number of measurements N is small compared to the number of possible discrete states 2^m . The overestimation OE of the MI can be represented by $\frac{2^m}{N}$ [11].

Consider two candidate neighbourhoods $R_1\{j; t\}$ and $R_2\{j; t\}$ of the cell $c(j; t)$ for a deterministic CA system. If both of these cover all potential neighbour-

hoods and $R_1\{j; t\} \in R_2\{j; t\}$, the following relationship is obtained.

$$\begin{aligned} I(f(R_1\{j; t\}); c(j; t)) &= I(f(R_2\{j; t\}); c(j; t)) \\ F(f(R_1\{j; t\}); c(j; t)) &> F(f(R_2\{j; t\}); c(j; t)) \end{aligned} \quad (5)$$

We use $\frac{2^m}{N}$ as an upper bound for the overestimation of the mutual information between any two finite sets of data. According to expression (5), the final candidate neighbourhood can be selected from the peak value of F . For the example above, $R_2\{j; t\}$ should be discarded because of the smaller value of F compared to that of $R_1\{j; t\}$. The method should therefore avoid selecting a neighbourhood with too large a range which can result in a massive waste of computational time at the later stages of CA identification.

The neighbourhood detection procedure can now be summarized as:

Initialization Assume that the maximal temporal search depth is t_{max} and set the initial temporary neighbourhood as $R_0\{c(j; t)\} = \{c(j; t - 1)\}$.

- (1) Collect the *case* $\{f(R_0\{j; t\}), c(j; t)\}$, see *Definition 1*, and calculate the value of F using expression (4).
- (2) Increase the temporal range and set the temporary neighbourhood as $R_1\{c(j; t)\} = \{c(j; t - 1), c(j; t - 2)\}$, and then collect *case* $\{f(R_1\{j; t\}), c(j; t)\}$ and calculate the value of F .
- (3) Repeat *step 2* until the temporal range reaches t_{max} .
- (4) Increase the spatial range and reset the temporal range to $t - 1$, which means setting the temporary neighbourhood as $\{c(j; t - 1), c(j - 1; t - 1), c(j + 1; t - 1)\}$, and then collect the *case* and calculate the value of F .
- (5) Repeat *step 1* to *step 3* until a peak value of F appears in the spatial direction. The temporary neighbourhood with maximal value of F can be selected as the final result.

This neighbourhood detection approach produces a range for the correct neighbourhood, which considerably reduces the model search in later steps. However, the algorithm can not guarantee to provide an exactly correct neighbourhood. There are cases when the exact neighbourhood will be obtained but this may not always be the case. For example, if the neighbourhood of a deterministic pattern is symmetrical in the spatial direction, such as $\{c(j - 1; t - 1), c(j; t - 1), c(j + 1; t - 1)\}$, the proposed approach will be expected to produce the exact correct neighbourhood. However, if the neighbourhood is asymmetrical in the spatial direction, such as $\{c(j; t - 1), c(j + 1; t - 1)\}$, a larger range will be detected. This is perfectly acceptable because essentially all we are trying to do is to use the new MI approach to provide an initial estimate of the temporal and the spatial range. This initial neighbourhood will then be used to prime the CA-OLS algorithm and to find the correct model terms and unknown parameters. The aim therefore is a coarse-to-fine approach where MI is used to significantly reduce the potential neighbourhood choices,

which can then be optimized using the CA-OLS identification algorithm.

3.3 Coarse-To-Fine Method of CA Identification

An important step in the identification of CA, is to use the MI neighbourhood detection approach to restrict the neighbourhood search range by providing an initial candidate neighbourhood. The remaining steps are the selection of significant model terms and estimation of the unknown parameters. In this paper the CA-OLS routine [4] will be used for this final step.

Orthogonal Least Squares (OLS), which was first proposed by Billings in 1988 [6], is widely used in the identification of nonlinear systems. Billings and Yang adapted OLS to the identification of cellular automata and showed that this algorithm can not only estimate the parameters of an equivalent polynomial CA model but can also select the significant model terms. This algorithm is called the *forward regression CA-OLS* routine. To reduce the computation time for large observed data sets or large neighbourhoods Mei proposed a Fast CA-OLS [12], which discarded some redundant operations and improved the efficiency considerably. The final estimates however are the same as ordinary CA-OLS. Other authors have also introduced modifications to the original OLS routine applied to nonlinear system identification to provide improved performance. One algorithm is the Locally Regularized Orthogonal Least Squares (LROLS), proposed by Chen [13]. The local regularization enforces model sparsity and avoids over-fitting in the model parameters, so the choice of cutoff value is less critical than for ordinary OLS. Another algorithm, named the Combined LROLS and the D-Optimality algorithm (LROLS + D-Optimality), was also proposed by Chen [14]. In this algorithm it was no longer necessary to specify the cutoff value and the algorithm can automatically terminate by the introduction of a parameter β . However, several iterations have to be performed to search for the optimal value in both the above modifications, so the computational time of LROLS and LROLS + D-Optimality is considerably greater than that of Fast CA-OLS, and much larger than ordinary OLS.

To illustrate the difference between these algorithms when applied to CA identification massive simulation examples were tested using the four algorithms and the results showed that Fast CA-OLS is typically 10 times faster than the other algorithms and obtains almost the same model with few spurious terms. Comparing the advantages and disadvantages of each method, Fast CA-OLS was selected as the core method to estimate the parameters in the new identification routine for binary CA.

The coarse-to-fine identification procedure for CA can be outlined as:

- (1) Detect the coarse spatial and temporal range of the neighbourhood using the mutual information algorithm.

- (2) Using the result from *step 1* to prime Fast CA-OLS, determine the significant terms using the Error Reduction Ratio (ERR) of each term. The correct neighbourhood should be produced in this step.
- (3) Using the neighbourhood from *step 2*, collect new *cases* then re-prime the Fast CA-OLS to estimate the unknown parameters in equation (2).

For a deterministic CA, when the neighbourhood of the observed pattern is symmetric in the spatial direction, such as $c(j-1; t-1), c(j; t-1), c(j+1; t-1)$, the new MI neighbourhood detection approach can often obtain the exact correct neighbourhood in *step 1*. If the neighbourhood is asymmetric, the detected candidate neighbourhood using MI will often be larger than the exact correct neighbourhood. In such a case, *step 3* can be omitted because both ERR and the correct coefficients can be obtained synchronously in *step 2*. While processing a probabilistic CA, for both the asymmetric and symmetric neighbourhoods, a larger candidate neighbourhood will often be detected using MI and all three steps of the coarse-to-fine procedure must be applied to determine the CA model.

4 Example Studies

Three simulation examples are employed in this section to demonstrate the application of the new coarse-to-fine algorithm. To show all the steps of this algorithm, initially a probabilistic 1-D example with the symmetric neighbourhood and a deterministic 1-D example an the asymmetric neighbourhood are discussed. A 2-D real experimental example will then be discussed to show the application of this algorithm to Belousov-Zhabotinsky reaction data.

4.1 Identification of a probabilistic 1-D 3 site CA pattern

Consider the Rule R18 [1] on a 200×200 lattice with three neighbourhoods $\{c(j-1; t-1), c(j; t-1), c(j+1; t-1)\}$. The rule is shown in Table 1. Noise was introduced to all three components by flipping the states with 40% probability during the evolution of the CA. This adds severe degrees of noise which is much more disruptive than simply using additive noise on the final pattern. The generated noisy pattern with random initial conditions and a periodic boundary using the above rule is shown in Fig. 3.(a). Fig. 3.(b) shows the pattern generated by the same rule using the same initial conditions but with no noise. A comparison of Fig. 3.(a) and Fig. 3.(b) clearly shows the severity of the noise on this example.

Table 1
Rule R18 used in Example 1

$t - 1$	000	001	010	011	100	101	110	111
t	0	0	0	1	0	0	1	0

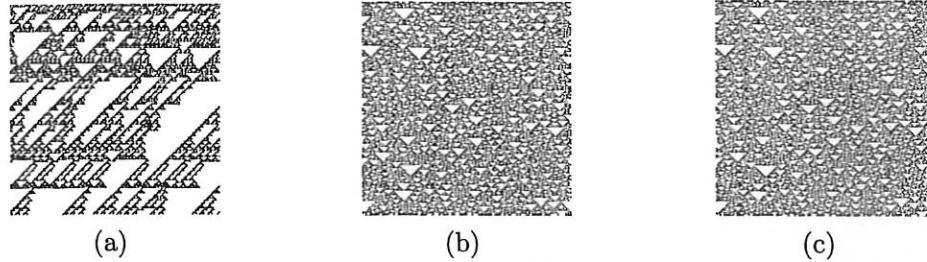


Fig. 3. Patterns from Rule R18 for Example 1. (a) generated pattern with noise; (b) generated pattern with no noise; (c) reconstructed pattern produced by the identified polynomial model

Table 2
The fitness function values of candidate neighbourhood of Example 1

Spatial	Temporal	F
$\{j\}$	$\{t - 1\}$	0.00549
$\{j\}$	$\{t - 1; t - 2\}$	0.01920
$\{j - 1; j; j + 1\}$	$\{t - 1\}$	0.52886
$\{j - 1; j; j + 1\}$	$\{t - 1; t - 2\}$	0.53115
$\{j - 2; j - 1; j; j + 1; j + 2\}$	$\{t - 1\}$	0.52869
$\{j - 2; j - 1; j; j + 1; j + 2\}$	$\{t - 1; t - 2\}$	0.50158

The identification procedure is summarized as follows:

Step 1: As the preliminary step of the coarse-to-fine routine, the initial maximal temporal search depth t_{max} was set to 2, the new neighbourhood detection approach was applied and the values of F for each candidate neighbourhood are shown in Table 2.

For convenience define

$$\begin{aligned}
 s_1 &= c(j - 1; t - 1) & s_2 &= c(j - 0; t - 1) \\
 s_3 &= c(j + 1; t - 1) & s_4 &= c(j - 1; t - 2) \\
 s_5 &= c(j - 0; t - 2) & s_6 &= c(j + 1; t - 2)
 \end{aligned}$$

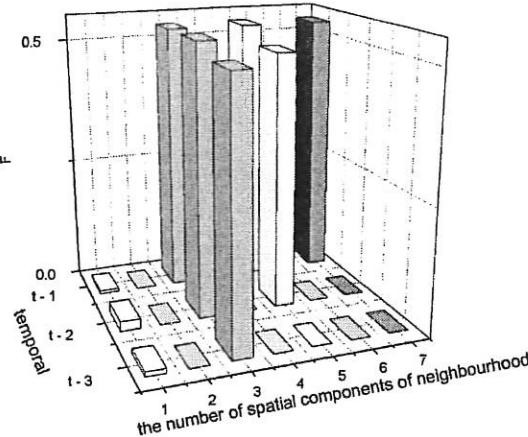


Fig. 4. The F values of each candidate neighbourhood when t_{max} was set to 3 for Example 1

so that from Table 2, F reaches a peak when the candidate neighbourhood is

$$\{s_1, s_2, s_3, s_4, s_5, s_6\} \quad (6)$$

The results using the new algorithm for a larger temporal search depth are illustrated by Fig. 4. Inspection of Fig. 4 shows that the final selected candidate neighbourhood is same as in equation (6). The set from equation (6) is therefore selected as the candidate neighbourhood. A comparison with Rule R18 shows that this includes all the original neighbourhood cells and provides a minimal symmetric spatial range to use to prime the next stage.

Step 2: To determine the exact neighbourhood and the CA model, *cases* were collected from the observed pattern and the Fast CA-OLS was applied to evaluate the contribution of each candidate cell to the updated cell. To save space, only the first 20 terms from the results shown in Table 3 ranked in order of significance are shown.

Inspection of Table 3 shows that the first 8 ERR terms are significantly larger than the remainder. Extracting the most significant result gives the final neighbourhood as

$$\{s_1, s_2, s_3\} \quad (7)$$

Step 3: By collecting *cases* using the neighbourhood in expression (7) and applying Fast CA-OLS again, the coefficients in equation (2) were estimated and these are shown in Table 4, giving the final polynomial model

$$c(j; t) = s_1 + s_3 - s_1 s_2 - 2s_1 s_3 - s_2 s_3 + 2s_1 s_2 s_3 \quad (8)$$

Table 3

The first 20 terms produced by Fast CA-OLS for Example 1

Order	Term	ERR
1	constant	0.23805
2	s_3	0.05097
3	$s_1 \times s_3$	0.10743
4	s_1	0.10089
5	$s_2 \times s_3$	0.08934
6	$s_1 \times s_2 \times s_3$	0.05286
7	s_2	0.04807
8	$s_1 \times s_2$	0.03553
9	$s_2 \times s_4 \times s_5$	0.00176
10	$s_2 \times s_3 \times s_4 \times s_5$	0.00053
11	$s_1 \times s_4$	0.00027
12	$s_1 \times s_5$	0.00025
13	$s_1 \times s_2 \times s_6$	0.00048
14	$s_2 \times s_6$	0.00021
15	$s_1 \times s_3 \times s_4$	0.00017
16	$s_1 \times s_2 \times s_3 \times s_6$	0.00009
17	$s_1 \times s_3 \times s_5$	0.00026
18	$s_2 \times s_3 \times s_6$	0.00010
19	$s_1 \times s_5 \times s_6$	0.00010
20	$s_1 \times s_2 \times s_4$	0.00015

The reconstructed pattern, shown in Fig. 3.(c), was generated using the polynomial model (8) with the same initial conditions as the observed pattern. A comparison of Fig. 3.(b) with Fig. 3.(c) clearly shows that the polynomial model (8) is an excellent representation. The CA rule in Fig. 3.(b) has therefore been successfully identified from the noisy data in Fig. 3.(a).

Table 4
The final terms and coefficients of Example 1

Order	Term	Coefficient
1	s_1	1.00000
2	s_3	1.00000
3	$s_1 \times s_2$	-1.00000
4	$s_1 \times s_3$	-2.00000
5	$s_2 \times s_3$	-1.00000
6	$s_1 \times s_2 \times s_3$	2.00000



Fig. 5. Patterns of Rule R50 for Example 2. (a) generated pattern; (b) reconstructed pattern produced by polynomial model

4.2 Identification of a deterministic 1-D CA pattern with the asymmetric neighbourhood

Consider the *Totalistic Rule R50* ([1]) on a 200×200 lattice with three neighbourhood terms $\{c(j; t-1), c(j+1; t-2), c(j-2; t-1)\}$. The generated pattern with random initial conditions and a periodic boundary condition using Rule R50 is shown in Fig. 5.(a). The identification procedure is summarized as follows:

Step 1: The initial maximal temporal search depth t_{max} was set to 2, the new neighbourhood detection approach was applied and the values of F for each candidate neighbourhood are shown in Table 5 and illustrated in Fig. 6.

For convenience define

$$\begin{aligned}
 s_1 &= c(j-2; t-1) & s_2 &= c(j-1; t-1) \\
 s_3 &= c(j-0; t-1) & s_4 &= c(j+1; t-1) \\
 s_5 &= c(j+2; t-1) & s_6 &= c(j-2; t-2) \\
 s_7 &= c(j-1; t-2) & s_8 &= c(j-0; t-2) \\
 s_9 &= c(j+1; t-2) & s_{10} &= c(j+2; t-2)
 \end{aligned}$$

Table 5

The F values of each candidate neighbourhood for Example 2

Spatial	Temporal	F
$\{j\}$	$\{t - 1\}$	0.00765
$\{j\}$	$\{t - 1; t - 2\}$	0.00991
$\{j - 1; j; j + 1\}$	$\{t - 1\}$	0.00808
$\{j - 1; j; j + 1\}$	$\{t - 1; t - 2\}$	0.15758
$\{j - 2; j - 1; j; j + 1; j + 2\}$	$\{t - 1\}$	0.16394
$\{j - 2; j - 1; j; j + 1; j + 2\}$	$\{t - 1; t - 2\}$	0.91771
$\{j - 3; j - 2; j - 1; j; j + 1; j + 2; j + 3\}$	$\{t - 1\}$	0.17101
$\{j - 3; j - 2; j - 1; j; j + 1; j + 2; j + 3\}$	$\{t - 1; t - 2\}$	0.14195

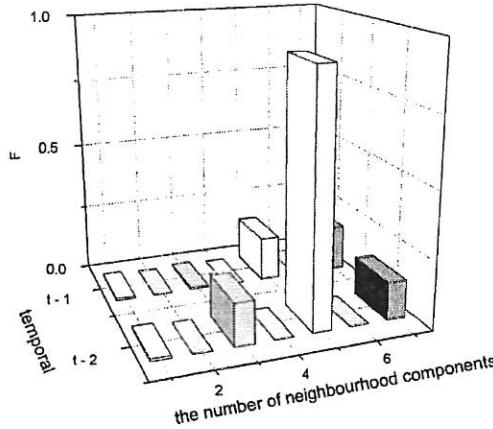


Fig. 6. The search results for Example 2

so that from Table 5, F reaches a peak when the candidate neighbourhood is

$$\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\} \quad (9)$$

The set from equation (9) is therefore selected as the final candidate neighbourhood. A comparison with Rule R50 shows that this includes all the original neighbourhood cells and provides a minimal symmetric spatial range to use to prime the next stage.

Step 2: By collecting *cases* using the neighbourhood in expression (9) and applying the Fast CA-OLS algorithm, the ERR values and the unknown parameters of each term were estimated and are shown in Table 6. The results show that the ERR values of each term are all similar and the component cells

Table 6
The selected terms and coefficients for Example 2

Order	Term	Coefficient	ERR
1	s_1	1.00000	0.0357
2	s_3	1.00000	0.0698
3	s_9	1.00000	0.1690
4	$s_3 \times s_9$	-2.00000	0.0692
5	$s_1 \times s_3$	-2.00000	0.1144
6	$s_1 \times s_9$	-2.00000	0.0618
7	$s_1 \times s_3 \times s_9$	2.00000	0.0744

are exactly the correct neighbourhood. Therefore, *Step 3* can be ignored and the final model can be represented by the expression

$$c(j; t) = s_1 + s_3 + s_9 - 2s_3s_9 - 2s_1s_3 - 2s_1s_9 + 3s_1s_3s_9 \quad (10)$$

The reconstructed pattern, shown in Fig. 5.(b), was generated using the identified polynomial model (10) with the same initial conditions as the observed pattern. A comparison of Fig. 5.(a) with Fig. 5.(b) clearly shows that the polynomial model (10) is an excellent model for this example.

4.3 Identification of the rule of a Belousov-Zhabotinsky reaction

The Belousov-Zhabotinsky (BZ) reaction, which was first discovered by Boris.P.Belousov and Anatol.M.Zhabotinsky in 1951, is a spatio-temporal chemical oscillator. Many researchers have noticed that the patterns generated by the BZ reaction are highly similar to the patterns created by some CA models. In this example, we repeated the famous experiment and an attempt was made to extract the potential rule of the reaction from the recorded data set.

The recipe for the experiment was given by A.T.Winfree [15]. Once the reaction starts blue rings propagate from localized regions on a red background. If the dish is given a gentle shake to break up the rings, spectacular geometries appear.

To capture the image of the experiment, a digital video camera using a USB connection to a PC was employed. Operating at full speed, this camera can record roughly 30 frames per second and the maximal resolution can reach 1024 by 768 pixels. In this experiment, the resolution was selected as 640 by 480 pixels. A schematic diagram of the experimental apparatus is shown in Fig. 7. To enhance the brightness and to prevent an inverted image of the

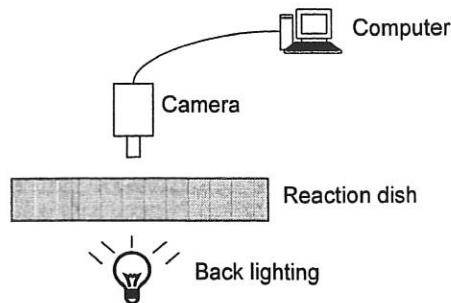


Fig. 7. A schematic diagram of the experimental apparatus for the BZ reaction

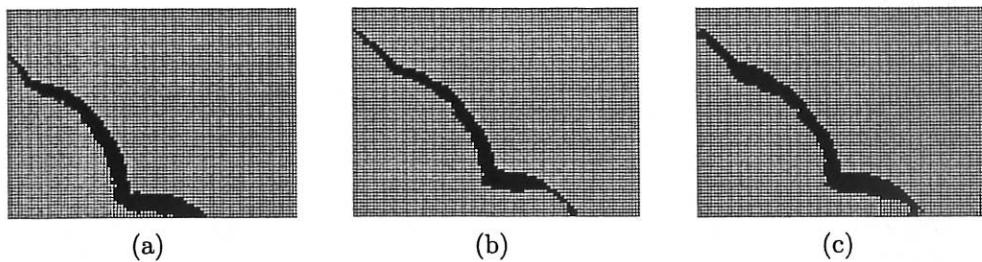


Fig. 8. Patterns from Example 3 (a) the first frame of the sample data; (b) the tenth frame of the system; (c) the predicted pattern of the tenth frame using the identified CA model

camera in the dish, back lighting was added to the bottom of the reaction dish.

Several preliminary steps were carried out to capture the data set. Because only two values can be taken in binary CA, each pixel in the images must be converted to black or white. The image from the BZ reaction is mainly composed of two colors: blue and red. We converted the blue or close to blue pixels to black, and the red or close to red pixels to white. Another important aspect when collecting real images is to set up the lattice of cells over the image. For example, one point in a real image can be described by a rectangle with 4 pixels in a digitized image, or could also be described by a rectangle with 9 pixels if a larger magnification is used. In this experiment, we adjusted the lattice size such that the tip velocity of the fastest change in the covered image was roughly one cell per time unit. We averaged over 4×4 pixel neighbourhoods as a representation of the center cell, to produce a 160×120 pixels image. A snapshot of one frame processed by the above procedure is shown in Fig. 8.(a)

The sampled data was composed of 7 sequential frames each of which comprised 160×120 pixels. Following the coarse-to-fine procedure, the new neighbourhood detection algorithm was applied and the results are shown in Table 7. The results indicate that the candidate neighbourhood should be selected

Table 7

The fitness function values of the candidate neighbourhood for Example 3 (von Neumann and Moor structures is described by Fig. 2)

Spatial	Temporal	<i>F</i>
$\{c(x; y)\}$	$\{t - 1\}$	0.29896
$\{c(x; y)\}$	$\{t - 1; t - 2\}$	0.29698
von Neumann structure	$\{t - 1\}$	0.42895
von Neumann structure	$\{t - 1; t - 2\}$	0.38663
Moore structure	$\{t - 1\}$	0.42410
Moore structure	$\{t - 1; t - 2\}$	0.00000

as the von Neumann structure (see Fig. 2) with temporal lag $t - 1$. Because the von Neumann structure is a minimal neighbourhood for 2-D CA, *Step 2*, to select the significant terms, was omitted and *Step 3* was applied directly. For this example define

$$\begin{aligned}
 s_1 &= c(x; y - 1; t - 1) \\
 s_2 &= c(x - 1; y; t - 1) \\
 s_3 &= c(x; y; t - 1) \\
 s_4 &= c(x + 1; y; t - 1) \\
 s_5 &= c(x; y + 1; t - 1)
 \end{aligned}$$

The results from the Fast CA-OLS algorithm including the ERR values and the coefficients of each terms are shown in Table 8.

The reconstructed pattern, generated by the identified polynomial model, is illustrated in Fig. 8.(c). A comparison of the regenerated pattern with the original pattern of the tenth frame from the sampled data set, shown in Fig. 8.(b), shows that the results from the identification of the BZ reaction are encouraging. Identification of a real dynamical system is often very difficult because information which can be extracted from sampled data is often limited. Moreover, real data are always corrupted by some unpredictable noise and in this case the image has been thresholded to produce a binary image. The aim here was to use a simple rule to represent the main properties of the evolution of this complex pattern. The identified polynomial model, given in Table 8, describes the features of the thresholded pattern acquired from the BZ reaction. These however are only preliminary results and many more experiments need to be conducted and all aspects of the data collection and modelling of this complex system require further study.

Table 8
The final terms and coefficients of Example 3

Order	Term	Coefficient	ERR
1	s_1	1.00000	0.01250
2	s_2	1.00000	0.09375
3	$s_2 \times s_4$	-1.00000	0.03044
4	$s_2 \times s_5$	-1.00000	0.04367
5	$s_2 \times s_3$	-1.00000	0.03668
6	$s_1 \times s_5$	-1.00000	0.01103
7	$s_1 \times s_4$	-1.00000	0.01241
8	$s_1 \times s_3$	-1.00000	0.00833
9	$s_1 \times s_2$	-1.00000	0.01042
10	$s_1 \times s_2 \times s_3$	2.00000	0.50000
11	$s_1 \times s_4 \times s_5$	1.00000	0.01658
12	$s_2 \times s_3 \times s_4$	1.00000	0.01518
13	$s_2 \times s_3 \times s_5$	1.00000	0.03265
14	$s_2 \times s_4 \times s_4$	1.00000	0.03784
15	$s_1 \times s_3 \times s_5$	1.00000	0.00972
16	$s_1 \times s_3 \times s_4$	1.00000	0.01122
17	$s_1 \times s_2 \times s_5$	1.00000	0.00625
18	$s_1 \times s_2 \times s_4$	1.00000	0.00937
19	$s_1 \times s_3 \times s_4 \times s_5$	-1.00000	0.00962
20	$s_1 \times s_2 \times s_3 \times s_4$	-1.00000	0.00174
21	$s_1 \times s_2 \times s_4 \times s_5$	-1.00000	0.00374
22	$s_1 \times s_2 \times s_2 \times s_5$	-1.00000	0.01042

5 Conclusions

A new neighbourhood detection approach has been introduced which is based on finding a coarse range of the neighbourhood initially using mutual information. This method can detect an exact or larger range of neighbourhood than the original even in the presence of noise as shown in the examples. The advantage of the new approach is that it can yield significant improvements in efficiency by not only restricting the search range prior to identification to

save computational time, but also involving all potential neighbourhoods to guarantee the validity of the generated model.

This paper also proposes a coarse-to-fine algorithm for the identification of CA, which decompresses the procedure into three steps. The simulated examples and the real experimental data clearly demonstrate the performance of the new algorithm.

Acknowledgment

The authors gratefully acknowledge that part of this work was financed by EPSRC(UK). We are also very grateful to Alex Routh, Department of Chemical Engineering, University of Sheffield who set up and conducted the real experiment.

References

- [1] Andrew Adamatzky, Cellular Automata: a discrete university, World Scientific, London, 2001.
- [2] Adamatzky A., (1997), Automatic programming of cellular automata: identification approach, *Kybernetes*, 26 (2-3): 126-133.
- [3] Andrew Adamatzky, Identification of Cellular Automata, Taylor Francis, 1994.
- [4] S.A.Billings and Y.X.Yang, "Identification of probabilistic cellular automata," *IEEE Transactions on systems Man and Cybernetics Part B*, vol.33, no.2, pp.225-236, 2003.
- [5] S.A.Billings and Y.X.Yang, "Identification of the neighbourhood and CA rules from spatio-temporal CA patterns," *IEEE Transactions on systems Man and Cybernetics Part B*, vol.30, no.2, pp.332-339, 2003.
- [6] M.Korenberg, S.A.Billings, "Orthogonal parameter estimation algorithm for nonlinear stochastic systems," *International journal of control*, vol.48, no.1, pp.193-210, 1988.
- [7] R.Korfhage, *Logical and Algorithms*. New York: Wiley, 1996.
- [8] F.C.Richards, "Extracting cellular automata rules directly from experimental data," *Phys.D*, vol.45, pp.189-202, 1990.
- [9] S.S.Mei, A.A.Billings and L.Z.Guo, "A Neighbourhood selection method for cellular automata models," *International journal of chaos and bifurcation*, to appear.

- [10] C.E.Shannon, "The mathematical theory of communication," Bell Syst. Techn. Journal 27, 379-423, 1948.
- [11] W.T.Li, "Mutual Information Verus Correlation Functions," Journal of statistical physical, 60(5-6):823-837, 1990.
- [12] S.A.Billings, S.Me, "A new fast cellular automata orthogonal least aquare identification method," Report of AC&SE, University of Sheffield, 2003.
- [13] S.Chen, "Locally regularization assisted orthogonal least squares regression", International journal of control, 2003.
- [14] S.Chen, X.Hong, and C.J.Harris, "Sparse Kernel Regression Modeling Using Combined Locally Regularized Orthogonal Least Squares and D-Optimality Experimental Design," IEEE Transactions on automatic control, vol.48, no.6, pp.1029-1036, 2003.
- [15] A.T.Winfree, "Spiral Waves of Chemical Activity," Science, 175: 634-636, 1972.
- [16] Wuensche,A., (2002) Discrete Dynamics Lab: Tools for inverstigating cellulat automata and discrete dynamical networks, Kybernets, vol 32 no.1/2 2003.
- [17] Andersson C, Rasmussen S, White R, (2002), Urban settlement transitions, Enviroment and Planning B-Planning and Design, 29(6),2002,841-865.
- [18] Li XB, Jiang R, Wu QS, (2003), Cellular automata model simulating complex spatiotemporal structure of wide jams, Physical Review E, 68(1) Part2,2003.
- [19] Voorhees B, (1998), Multifractals defined by nonlinear cellular automata, International Journal of Bifurcation and Chaos, 8(2), 1998,259-280.
- [20] Voorhees B, (1998), Predecessor states for certain cellular automata evolutions, Communications In Mathematical Physical, 117(3), 1988, 431-439.
- [21] Chaudhuri P.P., Chowdhury D. R., Nandi S, and Chattopadhyay S (1997), Additive Cellular Automata: Theory and Applications, IEEE Computer Society Press, 1997.

