Expert Systems With Applications 49 (2016) 99–111

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A cellular automata-based learning method for classification

Sartra Wongthanavasu[a,*], Jetsada Ponkaew[b]

[a] Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand
[b] Department of Computer Science, Faculty of Art and Science, Sisaket Rajabhat University, Sisaket 33000, Thailand

ARTICLE INFO

Keywords:
Cellular Automata
Classification
Decision Support Elementary
Support Vector Machines

ABSTRACT

Over the last few decades, classification applied to numerous applications in science, engineering, business and industries have rapidly been increased, especially for big data. However, classifiers dealing with complicated high dimension problems with non-conforming patterns with high accuracy are rare, especially for bit-level features. It is a challenging research problem. This paper proposed a novel efficient classifier based on cellular automata model, called Cellular Automata-based Classifier (CAC). CAC possesses the promising capability to deal with non-conforming patterns in the bit-level features. It was developed on a new kind of the proposed elementary cellular automata, called Decision Support Elementary Cellular Automata (DS-ECA). The classification capability of DS-ECA is promising since it can describe very complicated decision rule in high dimension problems with less complexity. CAC comprises double rule vectors and a decision function, the structure of which has two layers; the first layer is employed to evolve an input pattern into feature space and the other interprets the patterns in feature space as binary answer through the decision function. It has a time complexity of learning at $O(n^2)$, while the classification for one instance is O(1), where n is a number of bit patterns. For classification performance, 12 datasets consisting of binary and non-binary features are empirically implemented in comparison with Support Vector Machines (SVM) using *k*-fold cross validation. In this respect, CAC outperforms SVM with the best kernel for binary features, and provides the promising results equivalent to SVM on average for non-binary features.

1. Introduction

Researches in machine learning dealing with conforming and non-conforming patterns to discover the state-of-the-art methods have been conducted for several decades. In this regard, most of them lack of capability to handle non-conforming patterns, especially for binary represented features (Biehl, 2012; Scholkopf & Smola, 2001; Skowron & Synak, 2003). In addition, hardware implementations were not supported by the algorithms. Consequently, pattern classifiers based on Cellular Automata (CA) have been explored as a result of the promising structured and spatiotemporal characteristics. However, classifiers based on CA reported to date are few and limited due to complexity and accuracy performance.

For CA-based classifiers, a transition function, also called a rule vector, plays a vital role in classification performance. The rule vector can be ordered using heuristic search; genetic algorithm, simulated annealing (Jinshan & Lingzhi, 2012), etc., for example. The acquisition of the promising rule vector is the most important process. There are many attempts to investigate the behaviors of Elementary Cellular Automata (ECA) in all four classes (Sheng-Uei & Shu, 2003; Wolfram, 2002), i.e., fixed point, periodic, chaotic and complex. In this respect, evolving pattern of ECA in class I and II exhibited characteristics of pattern classification, called attractor basin (Ganguly, Maji, Sikdar, & Chaudhuri, 2002; Maji & Chaudhuri, 2008). There are some promising classifiers based on CA reported to date. CA has been used in remote sensing to implement processes related to simulations over the last few decades. Espínola, Piedra-Fernández, Ayala, Iribarne, and Wang (2015) proposed a new classification algorithm based on CA which not only improves the classification accuracy rate in satellite images by using contextual techniques but also offers a hierarchical classification of pixels divided into levels of membership degree to each class and includes a spatial edge detection method of classes in the satellite image. In addition, the most promising classifiers called Multiple Attractor Cellular Automata (MACA) and Generalized Multiple Attractor Cellular Automata (GMACA) were presented. MACA (Maji, Ganguly, & Chaudhuri, 2003a; Min & Song, 2012; Sikdar, Ganguly, & Chaudhuri, 2005) is a linear CA, known as low cost hardware implementation, working with the only exclusive OR (XOR) operation

* Corresponding author. Tel.: +66 43 362 188; fax: +66 43342910.
E-mail addresses: wongsar@kku.ac.th, sartra.wong@gmail.com (S. Wongthanavasu), jp.sskru@gmail.com (J. Ponkaew).

http://dx.doi.org/10.1016/j.eswa.2015.12.003
0957-4174/

of the neighbors. This leads to limited and incomplete functions. Subsequently, GMACA (Ganguly et al., 2002; Ganguly, Maji, Sikdar, & Chaudhuri, 2004; Maji, Shaw, Ganguly, Sikdar, & Chaudhuri, 2003b; Petrou, 2001; Ponkaew, Wongthanavasu, & Lursinsap, 2011) was developed to eliminate such a drawback faced in the MACA. It utilized a reverse engineering technique for ordering the rule. In this regard, GMACA reported the highest accuracy performance against other methods for error correcting code (ECC) problem in data transmission network for one bit noise. In general, MACA and GMACA are a special class of one-dimensional cellular automata being designed on the basis of the evolving structure of ECA. They evolved using a fix rule vector to change the given input. Due to the exponential growth of the state space and sharp dropped accuracy when a number of bit noises increases, it became the limitation in implement the GMACA for ECC.

The classification based on ECA faced the drawbacks of complexity and classification performance. These lead to the limitation of applications and fertile related researches. This paper proposed a novel efficient generic classifier based on CA, called Cellular Automata-based Classifier (CAC). For efficient reason, it was developed based on a new kind of one dimensional proposed ECA, called Decision Support Elementary Cellular Automata (DS-ECA). The DS-ECA comprises double rule vectors and a decision function. A pattern was changed by a rule vector that is indicated by the result of the decision function. This reduces the classification complexity for a pattern to O(1) and the search space complexity for ordering the rule to O(n^2), where n is a number of bit pattern. The classification performance reported by CAC outperforms SVM using the best kernel.

The paper was organized as follows. Following this section, ECA, MACA and GMACA are presented and given in Section 2. Suggestion 3 provides insightful information of SVM. Then, fundamentals and theoretical analysis in supporting the development of CAC are given in Section 4. Section 5 gives theoretical analysis of CAC and information of compared methods. Extensive test datasets with detailed information consisting of all kinds of features were given in Section 6. Section 7 provides extensive experimental results of the proposed method and SVM. Ultimately, conclusions and discussions were given in Section 8.

2. Pattern classifier based on an evolution of Elementary Cellular Automata

Elementary Cellular Automata (ECA) is the simplest class of CA. It consists of a group of cells arranged in one-dimension with two possible states (0 or 1). A next state S_i^{t+1} for the ith cell is considered from local transition function of its nearest neighbors $f(S_{i-1}^t,\ S_i^t,\ S_{i+1}^t)$ of the present state S_i^t for the cell. For simplicity, a next state of n cells ECA is represented by a matrix M, where row denotes a pattern and column denotes possible binary configuration decoded to decimal as follows:

$$M = [b_{ij}]_{\substack{i=0,1,2\ldots n-1 \\ j=0,1,2\ldots 7}} \tag{1}$$

M be a matrix with size $|n \times 8|$ representing the next state for n cells ECA, called a rule matrix. An element of the matrix $b_{ij} \in \{0,1\}$ is the next state for the ith cell where its nearest neighbors $(S_{i-1}^t,\ S_i^t,\ S_{i+1}^t)$ of the present state S_i^t decoded in decimal equal to j, $j = 0$ to 7. Consequently, a general form of evolving ECA is written using M as follows:

$$S^{t+1} = (M,\ S^t) \tag{2}$$

That is, a next state of n cells ECA, S^{t+1}, is determined by the rule matrix M and its present state S^t. A null boundary condition is considered—that is, the left-neighbor state of the left most cells and the right-neighbor state of the right most cells are 0-state.

Example 1. Suppose a 4-cell ECA of '0011' with null boundary condition and a rule matrix (M) were given following:

$$M = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

To find the next states of '0011' by M, an initial state S^0 is set to '0011'. Then, each cell S_i^0 (where $i = 0, 1, 2, 3$) is implemented by M. The neighboring states of a cell is in the form of $(S_{i-1}^t,\ S_i^t,\ S_{i+1}^t)$ and decoded in decimal j, where $j = 0, 1, 2, 3, 4, 5, 6, 7$, representing the column of the matrix M from left to right. Finally, the next state for the ith cell is an element of the matrix M in the ith row and the jth column. For the '0011' pattern, row $i = 0, 1, 2, 3$ corresponds to the bit pattern, and the column j corresponds to the neighboring states of '000', '001','011' and '110', which is decoded to decimal as '0', '1', '3' and '6', respectively. To consider the left most bit of '0011' which is '0' and its neighboring states of '000' which is '0' in decimal, the first row and the first column element of M is determined, which is '0' in this case. In similar manner, the next state of the second bit of '0011' is determined by looking up the second row and the '001' column, which is 1 in decimal or the first column, of M which is '1'. The third and fourth bits are the third and fourth rows and the fourth and eighth columns of M, respectively. Thus, a next state of '0011' is '0110'. More specifically, if each row of the M matrix is decoded from right to left in decimal, it is called a rule for the ith cell (e.g., $R_1 = 10010110_2 = 150$ (rule 150), for the *first* cell).

There are many attempts to investigate the behaviors of ECA in all four classes (Sheng-Uei & Shu, 2003; Wolfram, 2002), i.e., fixed point, periodic, chaotic and complex, in theory and applications. In this respect, evolving patterns of ECA in class I and II exhibited characteristics of pattern classification, called attractor basin (Ganguly et al., 2002; Maji & Chaudhuri, 2008). Attractor basin consists of a cyclic and non-cyclic phase, representing all traversal paths of states. A solution will be contained as a state in the cyclic phase while distorted patterns of the solution are contained in the other phase.

Definition 1. Suppose a system designed with a rule matrix (M) comprises a set of solutions $Y = \{y_i | y_i \in \{0,1\}^n\}$ and an input x; $x \in \{0,1\}^n$, where n is bits pattern and $i \in I^+$. Thus, pattern classifiers based on the evolving structure of ECA is defined as follows:

$$S^{t+1} = \begin{cases} \left(M,\ S^t\right), & \text{if } S^t \notin Y \\ S^t \text{ and stop}, & \text{otherwise} \end{cases} \tag{3}$$

For an input x, the present state (S^t) is set to x. Then, the next state (S^{t+1}) will be generated using M. The process will be continued until reaching some solution $S^t \in Y$.

Multiple Attractor Cellular Automata (MACA) presented by Maji et al. (2003a, 2003b) is one of the most pattern classifiers based on an evolution of ECA. It is a linear classifier based on ECA, working with the exclusive OR (XOR) logic function ($\oplus$) of the neighboring states $(S_{i-1}^t,\ S_i^t,\ S_{i+1}^t)$. In this method, all possible neighboring states are $S_{i-1}^t \oplus S_i^t$, $S_{i-1}^t \oplus S_{i+1}^t$, $S_i^t \oplus S_{i+1}^t$, S_{i+1}^t, S_i^t, S_{i-1}^t and $S_{i-1}^t \oplus S_i^t \oplus S_{i+1}^t$. Consequently, the possible rules consist of 60, 90, 102, 170, 204, 240 and 150. This can reduce the search space for ordering the rules by heuristic searches to 7^n, where n is a number of bit pattern. Fig. 1 portrayed two attractor basins of 4-bit MACA, {0000, 1100}, using null boundary condition and a rule vector ⟨ 90, 204, 102, 150⟩. In this regard, {0000, 1100} be a set of solutions and the rule vector is ordered by genetic algorithm (GA) in MACA. For example, given an input pattern '0101', MACA sets it to the initial state S^0. Then, the next state (S^1) is generated using the XOR logical function, e.g. $S_0^{t+1} = 0 \oplus S_1^t$,