Evolving Cellular Automata as Pattern Classifier

Niloy Ganguly[1], Pradipta Maji[2], Sandip Dhar[2], Biplab K. Sikdar[2], and P. Pal Chaudhuri[2]

[1] Computer centre, IISWBM, Calcutta, West Bengal, India 700073, n_ganguly@hotmail.com

[2] Department of Computer Science & Technology, Bengal Engineering College (D U), Howrah, West Bengal, India 711103 {pradipta,sdhar,biplab,ppc}@cs.becs.ac.in

Abstract. This paper reports a high speed, low cost pattern classifier based on the sparse network of Cellular Automata. High quality of classification of patterns with or without noise has been demonstrated through theoretical analysis supported with extensive experimental results.

1 Introduction

The internetworked society has been experiencing a explosion of data that is acting as an impediment in acquiring knowledge. The meaningful interpretation of these data is increasingly becoming difficult. Consequently, researchers, practitioners, entrepreneurs from diverse fields are assembling together to develop sophisticated techniques for knowledge extraction. Study of data classification models form the basis of such research. A classification model comprises of two basic operations - classification and prediction. The evolving CA based classifier proposed in this paper derives its strength from the following features:

- The special class of CA referred to as Multiple Attractor Cellular Automata ($MACA$) is evolved with the help of genetic algorithm to arrive at the desired model of CA based classifier.
- In the prediction phase the classifier is capable of accommodating noise based on distance metric.
- The classifier employs the simple computing model of three neighborhood Additive CA having very high throughput. Further, the simple, regular, modular and local neighborhood sparse network of Cellular Automata suits ideally for low cost $VLSI$ implementation.

The Cellular Automata (CA) preliminaries follows in the next section.

2 Cellular Automata Preliminaries

The fundamentals of Cellular Automata we deal with is reported in the book [1]. The classifier reported in this work has been developed around a specific class of CA referred to as *Multiple Attractor CA* ($MACA$).

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 56–68, 2002.

2.1 Multiple Attractor Cellular Automata

The state transition graph of an $MACA$ consists of a number of *cyclic* and *non-cyclic* states. The set of non-cyclic states of an $MACA$ forms inverted trees rooted at the cyclic states. The *cycles* are referred to as *attractors*. *Fig.1* depicts the state transition diagram of a 5-cell $MACA$ with four attractors {00000,00011,00110,00101} having self loop. In rest of the paper, by an attractor we will refer to a cycle of length 1. The states of a tree rooted at the cyclic state α forms the *α-basin.*

With reference to the state transition diagram of a CA, the *depth d* of the CA is the number of edges between a non-reachable state and the attractor. The depth d of the 5-cell $MACA$ of *Fig.1* is 3.

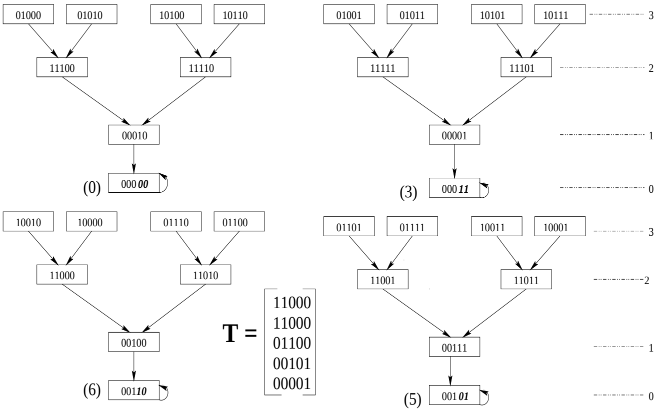

Fig. 1. State transition diagram of a 5-cell $MACA$

The detailed characterization of $MACA$ is available in [1]. A few fundamental results for an n-cell $MACA$ having k number of attractors is next outlined.

Result I: The characteristic polynomial of the $MACA$ is $x^{n-m}(1+x)^m$, where $m = log_2(k)$.

Result II: The characteristic polynomial noted above can be also written in elementary divisor form as
$(1+x)(1+x)\cdot\cdot m \ times \quad x^{d_1}x^{d_2}\cdot\cdot x^{d_p}$ where $d_1 \geq d_2 \cdot\cdot \geq d_p$ and $d_1 + d_2 \cdot\cdot\cdot + d_p = n - m$.

Result III: The minimal polynomial of an $MACA$ is $x^{d_1}(1+x)$, where depth $= d_1$.

Definition 1 *An m-bit field of an n-bit pattern set is said to be pseudo-exhaustive if all possible 2^m patterns appear in the set.*

Theorem 1 *[1] In an n cell $MACA$ with $k = 2^m$ attractors, there exists* m-bit *positions at which the attractors give pseudo-exhaustive 2^m patterns.*

Theorem 2 *[1] The* modulo-2 *sum of two states is the non-zero predecessor of* 0-state *(pattern with all 0's) if and only if the two states lie in the same $MACA$ basin.*

Example 1 *The example $MACA$ of* Fig.1 *is used to illustrate the above results.*

- *It is a 5-cell $MACA$ having 4 number of attractors and the depth of the $MACA$ is 3.*
- Result I: *The characteristic polynomial is $x^3 \cdot (1+x)^2$. Therefore, m=2. This is consistent with the result in the* Fig.1 *where attractor(k) is 4.*
- Result II: *The characteristic polynomial in elementary divisor form is $x^3 \cdot (1+x) \cdot (1+x)$.*
- Result III: *The minimal polynomial is $x^3 \cdot (1+x)$.*
- Result of Theorem 1: *In* Fig.1*, two least significant bit positions constitute the PEF.*
- Result of Theorem 2: *We take an attractor 00011 and any two states 11111, 11101 of the attractor basin. The* modulo-2 *sum of these two patterns is 00010 which is a state in $0-$basin. By contrast, if we take two states 00001 and 11000 belonging to two different attractor basins 00001 and 11000 respectively, their* modulo-2 *sum is 11011 which is a state in a non-zero attractor (00101) basin.*

2.2 *MACA* – As a Classifier

An n-bit $MACA$ with k-attractors can be viewed as a natural classifier (*Fig.2*). It classifies a given set of patterns into k-distinct classes, each class containing the set of states in the attractor basin.

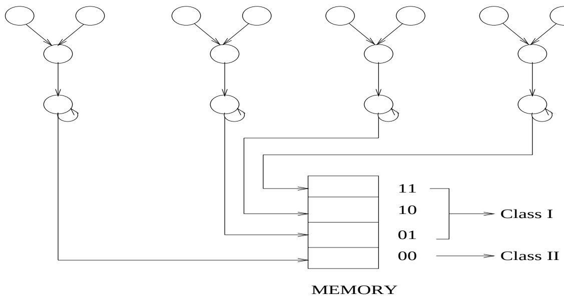

Fig. 2. $MACA$ based Classification Strategy

For an ideal classifier, to distinguish between two classes, we would need one bit. The classifier of *Fig 2* requires two bits to distinguish between two classes which is a memory overhead of the design. A k-attractor two class classifier needs $log_2(k)$ bits.

To classify pattern set into two classes, we should ideally find an $MACA$ with two attractor basins - each basin having the members of the specific class. Even if this ideal situation is not attained, the algorithm should design a $MACA$ based classifier having minimum number of attractor basins - while one subset of basins accommodates the elements of one class, the remaining subset houses the elements of the other class.

MACA based two class classifier. The design of the $MACA$ based Classifier for two pattern sets P_1 and P_2 should ensure that elements of one class (say P_1) are covered by a set of attractor basins that do not include any member from the class P_2. Any two patterns $a \in P_1$ and $b \in P_2$ should fall in different attractor basins. According to *Theorem 2*, the pattern derived out of *modulo-2* sum of a and b $(a \oplus b)$ should lie in a non-zero attractor basin. Let X be a set formed from *modulo-2* sum of each member of P_1 with each member of P_2 that is, $X = \{x_l \mid x_l = (a_i \in P_1) \oplus (b_j \in P_2) \, \forall_{i,j}\}$. Therefore, all members of X should fall in non-zero basin. This implies that the following set of equations should be satisfied.

$$T^d \cdot X \neq 0 \qquad (1)$$

where T is a valid $MACA$ to be employed for designing two class classifier.

Design of Multi-class Classifier. A two class classifier can be viewed as a single stage classifier. For designing a multi-class classifier, this scheme of single stage classification will be repeatedly employed leading to a multi-stage classifier consisting of multiple CA, each CA corresponds to a single stage (*Fig.3*).

Hence, in the rest of the paper, we concentrate on designing an efficient CA based two class classifier.

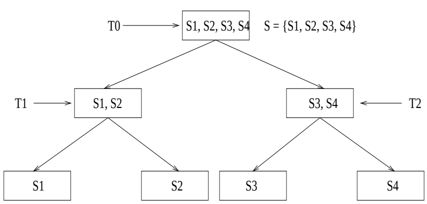

Fig. 3. $MACA$ based multi-class classifier. Note : A leaf node represents a class in the input set $S = \{S_1, S_2, S_3, S_4\}$

3 Genetic Algorithm for Evolution of $MACA$

The aim of this evolution scheme is to arrive at the desired $MACA$ (T matrix) that can perform the task of classification with minimum number of attractors.

This section describes the GA based solution to evolve the $MACA$ with the desired functionality. The three major functions of GA - Random Generation of initial population, Crossover and Mutation, as developed in the current GA formulation, are next discussed.

3.1 Random Generation of Initial Population with Desired Format

To form the population, it must be ensured that each solution randomly generated is an n bit $MACA$ with 2^m number of attractors where m can assume any value from 1 to n. From *Result II* of *Section 2.1*, the elementary divisor form of $MACA$ is $(1+x)(1+x)\cdot\cdot m\ times\ \ x^{d_1}x^{d_2}\cdot\cdot x^{d_p}$ & $d_1 + d_2 \cdots + d_p = (n\ -\ m)$, where the number of $(1+x)$ determines the number of attractors.

As per [3], the elementary divisors, if arranged in different order, produces $MACA$ with different attractor basins. For synthesis, the elementary divisors are first randomly distributed among itself forming a sequence. *Fig 4* shows one such sequence for the characteristic polynomial $x^7(1+x)^3 : x^2 \cdot x^2 \cdot (1+x) \cdot x \cdot (1+x) \cdot (1+x) \cdot x^2$. This is referred to as *pseudo-chromosome format* has been detailed in *Section 3.2*.

Each elementary divisor $(\phi_i(x))$ can be converted to a CA [1]. A tri-diagonal T matrix with characteristic polynomial $\phi_i(x)$ is accordingly synthesized. Let T_i matrices correspond to elementary divisors x^{d_i} (it will be a $d_i \times d_i$ matrix) and T_j matrices correspond to each elementary divisors $(x+1)$. If T_is and T_js are randomly arranged in block diagonal form, the characteristic polynomial of the resultant T is $x^{n-m} \cdot (1+x)^m$ and the minimal polynomial is $x^{d_p} \cdot (1+x)$ and it generates the $MACA$[1].

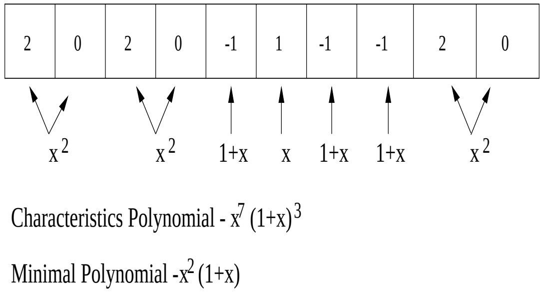

Fig. 4. MACA in Diagonal Representation Form

3.2 Pseudo-Chromosome Format

It is a method of representing an $MACA$ with respect to the sequence in which its $x^{d_i}{'s}$ and $(1+x)'s$ are arranged. It gives a semblance of the chromosome and hence termed as *pseudo-chromosome format*. It a string of n bits where (a)

d_i positions occupied by a x^{d_i} is represented by d_i followed by $(d_i - 1)$ zeros (for example, $x^3 = [300]$), and (b) $(1 + x)$ is represented by -1. The *pseudo-chromosome format* of the $MACA$ is illustrated in *Fig. 4*.

3.3 Crossover Algorithm

The crossover algorithm implemented is similar in nature to the conventional one normally used for GA framework with minor modifications as illustrated in *Fig.5*. The algorithm takes two $MACA$ from the present population (PP) and forms the resultant $MACA$. The *pseudo-chromosome format* has x^{d_i} represented by d_i followed by $(d_i - 1)$ zeros. But in the case of *Fig 5c*, we have 3 followed by a single zero. This is a violation since the property of $MACA$ is not maintained. So we take out those two symbols and form a CA of elementary divisor x^2 and adjust it. The resultant $MACA$ after adjustment is shown in *Fig 5d.*

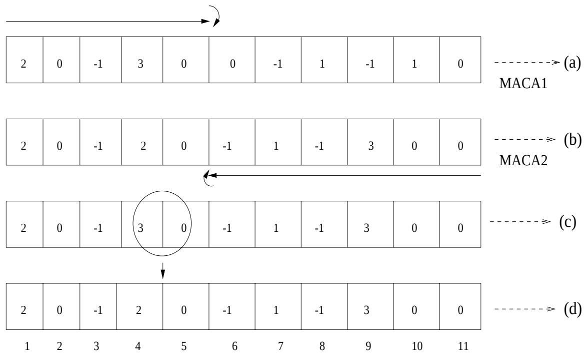

Fig. 5. An Example of Cross-over Technique

3.4 Mutation Algorithm

The mutation algorithm emulates the normal mutation scheme (*Fig.6*. It makes some minimal change in the existing $MACA$ of PP (Present Population) to a new $MACA$ for NP (Next Population). Similar to the single point mutation scheme, the $MACA$ is mutated at a single point.

In mutation algorithm, an $(x + 1)$'s position is altered. Some anomaly crops up due to its alteration. The anomaly is resolved to ensure that after mutation the new CA is also an $MACA$. The inconsistent format, as shown in the *Fig 6b* is the mutated version of *Fig 6a*. The inconsistency of the *pseudo-chromosome format* of *Fig 6b* can be resolved to generate the format of *Fig 6c*.

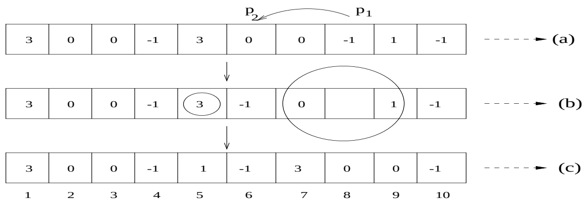

Fig. 6. An Example of Mutation Technique

3.5 Fitness Function

The fitness $\mathcal{F}(s)$ of a particular $MACA$ in a population is determined by the weighted mean of two factors - F_1 and F_2. The fitness criteria F_1 of the $MACA$ is determined by the percentage of patterns satisfying the *relation 1*. F_2 has been defined as -

$$F_2 = 1 - [(m-1)/n]^l \tag{2}$$

where 2^m denotes the number of attractor basins for the n cell CA, and l is equal to 1·8. The value of l is set empirically.

Subsequent to extensive experimentation, we have fixed up the relative weighage of F_1 and F_2 to arrive at the following empirical relation for the fitness function

$$\mathcal{F}(s) = 0.8 \cdot F_1 + 0.2 \cdot F_2 \tag{3}$$

4 Characterization of Attractor Basin – Its Capacity to Accommodate Noise

A classification machine is supposed to identify the zone, the trained patterns occupy. That is, let P_i be a representative pattern of n-bit learnt by the classifier as (say) Class A. Then a new n-bit pattern $\tilde{P}_i$ also belongs to the same class, if the hamming distance (r) between P_i and $\tilde{P}_i$ is small ($r << n$). **The hamming distance r is termed as noise while the pattern $\tilde{P}_i$ is termed as noisy pattern**.

This section provides a comprehensive study on the capacity of the classifier to accommodate noise. The study is developed in four phases

- *Phase I:* Establishes the fact that probing into the nature of pattern distribution of zero basin is equivalent to studying the noise accommodating capacity of the classifier.
- *Phase II:* Establishes the fact that characterization of the zero basin of $MACA$ with two attractor basins can be easily extended to $MACA$ with more than two attractors.
- *Phase III:* Characterizes the zero basin of $MACA$ with two attractor basins.
- *Phase IV:* Generalizes the study with 2^m number of basins, m varying from 1 to n.

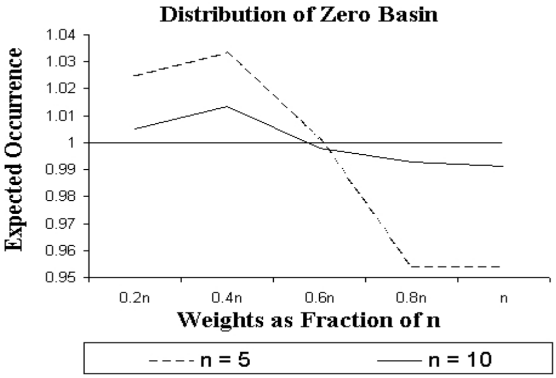

Fig. 7. Expected Distribution of MACA with two attractors ($m = 1$)

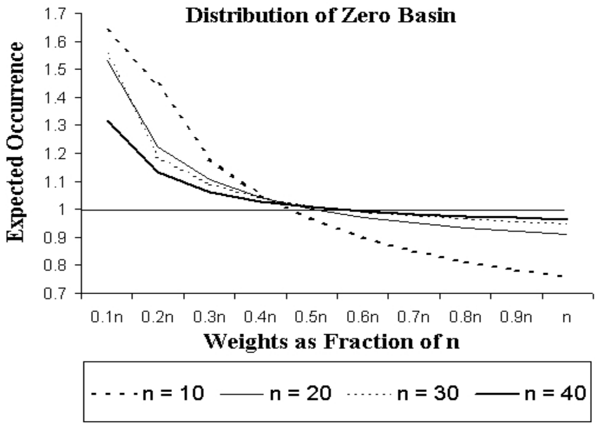

Fig. 8. Expected Distribution of MACA with 4 attractors ($m = 2$)

For shortage of space we only report the end results. The details are available in [3]. Both *Phase III & IV* show that the zero basin has a definite bias for patterns with lesser weight. *Phase III* establishes the relation on $MACA$ with two attractors, while *Phase IV* generalizes the result for multiple attractor $MACA$.

The bias gets reflected in the graphs reported in the *Fig 7-9* for $MACA$ with 2^m attractor basins. The graphs are based on the relation denoting the Expected Occurrence (EO) of a pattern with particular weight (r) in the zero basin of an $MACA$ over the normal unbiased expectation. The general relationship can be noted as

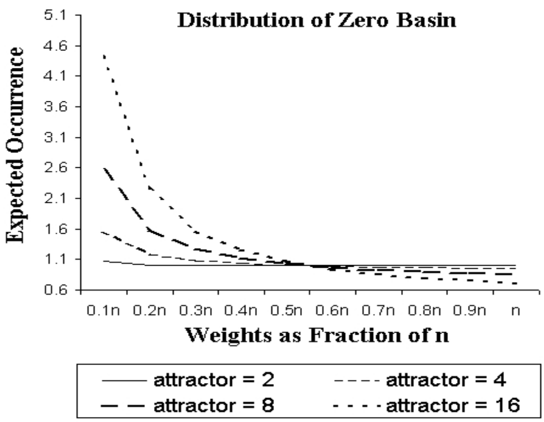

Fig. 9. Expected Distribution of MACA (n =30) with multiple attractors (m = 1,2,3,4)

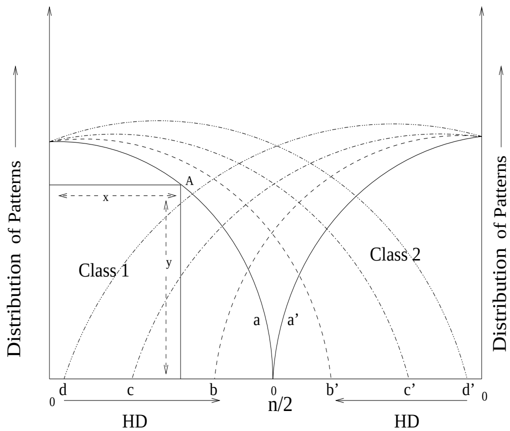

Fig. 10. Distribution of patterns in class 1 and class 2

$$EO(r, m) = N_{MACA}(r, m)/N_{UB}(r, m) \tag{4}$$

where $N_{UB}(r, m)$ shows the expected number of patterns with weight r in an $(n - m)$ dimensional vector subspace. $N_{MACA}(r, m)$ = the expected number of patterns with weight r in the zero basin of an $MACA$ with 2^m attractors. The entire equation is specific for a particular n. It has been shown [3] that the bias of the zero basin for low weight (r) states ($r << n$) creeps up due to the three neighborhood constraints of the CA [3].

The graph in *Fig 7-9* plots the expected occurrence - EO(r,m) - denoted by *relation 4* in the y-axis, while the weight of patterns is plotted on x-axis as a fraction of n - the number of bits in a pattern.

In *Fig 7*, where $m = 1$, it is seen that the expected occurrence doesn't follow a monotonically decreasing function but reaches the peak at slightly higher weight value. However, as m value is increased, (for *graph -8,* $m = 2$), the function becomes monotonically decreasing.

The gradient becomes steeper as the value of m is increased further (*Fig 9*). In graph of *Fig 9* which is plotted for different values of m keeping n ($= 30$) constant. It can be seen that the expectation of lower weight patterns occurring in zero basin increases manifold.

5 Performance Analysis of *MACA* Based Classifier

For the sake of convenience of performance analysis, distributions of patterns in two classes are assumed as shown in *Fig.10.* Each pair of sets on whom classifiers are run are characterized by the curves ($a-a'$, $b-b'$, $c-c'$, $d-d'$). The ordinate of the curves represents number of pairs of patterns having the specified hamming distance. For example, at point A (on the curve for a) has y number of pairs of patterns which are at hamming distance x. The abscissa has been plotted in both direction, from left to right for *Class I* while from right to left for *Class II.* The curves of *Class I* & *II* overlap if $D_{min} < d_{max}$. An ideal distribution $a - a'$ is represented by the continuous line without any overlap of two classes.

In each distribution various values of n are taken. For each value of n, 2000 patterns are taken for each class. Out of this, 1000 patterns are taken from each class to build up the classification model. The rest 1000 patterns are used to test the prediction accuracy of the model. For each value of n, 10 different pairs of pattern sets are built.

The *Table 1* represents the classification efficiency of data set $a - a'$, $b - b'$, $c-c'$. *Column II* represents the different values of m (number of attractor basins) for which *GA* finds the best possible solution. *Column III* to *VI* represent the classification efficiency of training and test data set respectively. Classification efficiency of training set is the percentage of patterns which can be classified in different attractors while that of test data implies the percentage of data which can be correctly predicted. The best result of classification efficiency corresponding to each m in the final generation is taken. This is averaged over for the 10 different pairs of pattern set taken for each value of n.

The following experiments validate the theoretical foundations of the classifier performance reported in earlier sections.

5.1 Expt 1: Study of GA Evolution

The *GA* starts with various values of m. But it soon begins to get concentrated in certain zone of values. The genetic algorithm is allowed to evolve for 50 generations. In each case 80% of the population in the final solution assumes the two or three values of m noted in *Column II* of *Table 1*. The classification performance improves at a very slow gradient on further increase of m.

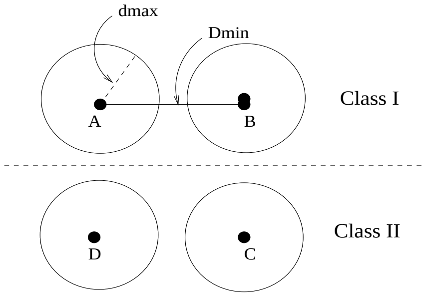

Fig. 11. Clusters Detection by two-class classifier

Table 1. Experiment to find out the value of m

Size (n)	Value of m	Curve a-a'		Curve b-b'		Curve c-c'	
		Training	Testing	Training	Testing	Training	Testing
20	2	85.40	85.60	83.20	82.00	81.20	72.40
	3	96.10	94.35	92.20	93.35	92.20	83.35
40	3	98.20	97.75	97.60	96.80	77.60	66.80
	4	98.35	98.85	97.20	97.45	94.28	87.45
60	3	98.55	97.75	96.90	96.05	86.98	77.55
	4	98.50	98.00	96.90	96.05	91.90	86.60
80	3	98.81	98.65	98.70	97.70	81.70	76.35
	4	99.15	99.20	98.70	97.70	88.79	87.30
	5	99.75	99.70	98.30	97.30	91.65	87.10
100	3	99.65	99.25	98.30	97.45	86.40	77.95
	4	99.67	99.35	98.40	97.30	83.10	80.35

5.2 Expt 2: Experiment Done on $a - a'$ and $b - b'$

It shows that classification accuracy is above 98% in most of the cases. Moreover, the prediction accuracy is also almost equal to the classification accuracy. This validates the theoretical foundation that $MACA$ basins are natural clusters.

5.3 Expt 3: Experiment Done on Data Set $c - c'$

With increasing overlap of data set (curve $c - c'$), a significant gap is created between classification efficiency of test and trained data. The *Column V* & *VI* of *Table 1* shows the result. But even in this case the classification efficiency and the number of attractors required to represent the data set doesn't deteriorate much.

Table 2. Clusters Detection by MACA based Classifier, $n = 100$, $D_{min} = 20$, $d_{max} = 5$

Combi^n of Clusters	Value of m	Performance (%)	
		Training	Testing
A & B, C & D	2	95.90	92.30
	4	99.82	97.10
A & C, B & D	2	94.50	92.30
	4	98.70	96.62
A & D, B & C	2	94.60	90.40
	4	99.20	96.82

5.4 Expt 4a: Experiment with Data Sets Having Implicit Clusters

Even if the data set between two classes have high overlap $(d - d')$, the classifier functions well if the classes have implicit clusters among them. The following experiment is performed to validate this observation.

To perform this experiment, we first randomly generate four pivot patterns (*Fig.11*) which are separated by a fixed hamming distance (say D_{min}). Then around each pivot, we randomly generate a cluster of p number of patterns within d_{max} distance (maximum hamming distance between a pivot and an element in the associated cluster); where $d_{max} <= D_{min}/2$. It is seen that $MACA$ based classifier perform classification task very well even though the distribution of patterns of each class is not ideal. The *Table 2* gives the detail result of cluster detection.

The *Column III* and *IV* depict the classification efficiency of the training and testing data set respectively at different values of m. In the two attractor basins (say a & b) which *Cluster A* & *B* occupy, it is observed that 99% of patterns of *Cluster A* occupies attractor a, while 99% of patterns of *Cluster B* occupies the attractor b. It is similar in case of the class formed from C and D. This is in line with the theoretical formulation that each attractor basin generates a cluster of patterns having lesser hamming distance.

6 Conclusion

The paper presents the detailed design and analysis of Cellular Automata based Pattern Classification technique. Theoretical formulation and experimental results prove beyond doubt the scope of the machine becomes really popular in industry in the near future.

References

1. P.Pal Chaudhuri, D.Roy Choudhury, S. Nandi and S. Chattopadhyay. *Additive Cellular Automata Theory and Applications.* IEEE Computer Society Press, California, USA, 1997.
2. S. Wolfram, *Theory and applications of cellular automata.* Singapore: World Scientific, 1986. ISBN 9971-50-124-4 pbk.
3. Niloy Ganguly, *"Cellular Automata Evolution : Theory and Applications in Pattern Recognition and Classification",* Ph.D thesis (proposal) *CST* Dept. BECDU India.