Pattern Recognition Letters 116 (2018) 114–120

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

A solution to the classification problem with cellular automata

Arif Orhun Uzun[a], Tuğba Usta[a], Enes Burak Dündar[b], Emin Erkan Korkmaz[a,*]

[a] Yeditepe University, İstanbul, Turkey
[b] Boğaziçi University, İstanbul, Turkey

ARTICLE INFO

Article history:
Received 14 April 2018
Available online 9 October 2018

Keywords:
Classification
Cellular automata
Heat Transfer
Big Data

ABSTRACT

Classification is the task of labeling data instances by using a trained system. The data instances consist of various attributes and in order to train the system, a set of already labeled data is utilized. After the training process, success rate of the system is determined with separate test sets. Various machine learning algorithms are proposed for the solution of the problem. On the other side, Cellular Automata (CA) provide a computational model consisting of cells interacting with each other based on some predetermined rules. In this study, a new approach is proposed for the classification problem based on CA. The method maps the data instances in the training data set to cells of an automaton based on the attribute values. When a CA cell receives a data instance, this cell and its neighbors are heated based on a heat transfer function. A separate automaton is heated for each class in the data set and hence a characteristic heat map is obtained for each class at the end of the procedure. Then new instances are classified by using these heat maps. The success rate of the algorithm is compared with the results of other known classification algorithms in the experiments carried out.

1. Introduction

Classification is a supervised learning process in which a set of labelled data is utilized to partition the instance space. If the classification model performs well on the training data utilized, then it becomes possible to predict appropriate labels for unseen data instances. For classification tasks, data instances are represented in the following format, $\{(x_1, l_1),(x_2, l_2),\ldots ,(x_k, l_k)\}$, where x_i is a data instance with n attributes and l_i is the class label of the data instance. The classification task can be defined as deducing the function f: $X \rightarrow L$ where X is the instance space and L is the output space. Accuracy of the function f is calculated with a separate test set. Many different techniques are used for acquiring the function f such as Naive Bayes, support vector machines and decision trees [7,13,18].

Cellular Automata (CA) provide a means for computation based on the interaction of interconnected cells in a discrete system. The interaction is carried by using some predetermined rules. Most popular CA application is Conway's Game of Life [8]. Other than that, CA is mostly used as a simulation tool for various disciplines [2,23].

In this study, a model based on CA is proposed as a solution to the classification problem. For any data set with n attributes and c classes, c two dimensional (2D) automata are utilized in the computation where a 2D-automaton is reserved for each class. Each attribute a_i is associated with a row of this 2D-automaton. Hence, each automaton would have n rows and m columns, where m is parameter that can be set to different values in the algorithm. Then the automaton cells are heated based on the attribute values of the instances belonging to the class associated with this automaton. The warming procedure is repeated for each automaton in the model by using the instances of the classes corresponding to these automata. A characteristic heat map is obtained for each class in the dataset at the end of the procedure. Then the new instances can be classified by using these heat maps belonging to different classes.

Performing the classification task by using these characteristic heat maps is the novel approach proposed in this study. The method is quite simple and is advantageous in terms of runtime performance compared to other classification methods. The details of the warming and the classification procedures utilized in the study can be found in Section 3.

2. Related work

In the literature, various approaches exist where CA is utilized for the classification task. For instance, in [19], a novel method named as Classificational Cellular Automata(CCA) has been pro-

* Corresponding author.
E-mail addresses: burak.dundar@boun.edu.tr (E.B. Dündar), ekorkmaz@cse.yeditepe.edu.tr (E.E. Korkmaz).

https://doi.org/10.1016/j.patrec.2018.10.007
0167-8655/

Fig. 1. Neighborhood Types.

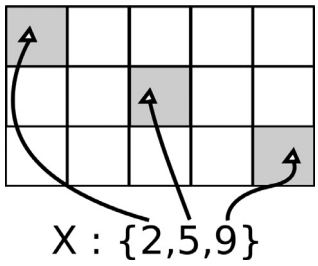


Fig. 2. Mapping procedure.

posed. CCA combines various classification techniques as a Multi Classifier System. In [5], data classification task has been solved where the relationship between the data attributes are extracted by utilizing a cellular learning automaton. In [15], the classification process is again carried out by a CA. An ensemble of classifiers is built by a CA in the study. Each cell of the CA keeps information about the success of the corresponding classifier. Then, a set of transaction rules determine the energy in each cell by using the information kept in it. Learning process is conducted by the interaction of neighbourhood cells based on these transaction rules. For each dataset, the performance may depend on parameters like the size of CA and transaction rule parameters. Fawcett [6] is another example where the classification problem is solved by using CA. The study focuses on the question of how data points with real-valued vectors should be mapped into a discrete space. Moreover, the method proposed uses a voting rule for determining the class of a selected cell.

Also, in [20], a CA model utilizing a Gaussian kernel has been proposed in order to solve the classification problem. Yet, the method is analysed only on 2-dimensional data, not tested on well-known datasets, and not compared with well-known classification methods. Cellular Automata have been widely utilized for simulation, too. In [1], a CA approach has been proposed for the problem of heat conduction. In [14], the simulation of the temperature changes on the earth surface is carried out by cellular automata. On the other side, in [3], simulations of heat and mechanical energy have been demonstrated on cellular automata. These approaches indicate that various real world processes can be simulated via cellular automata including heat conduction.

On the other hand, in [4], clustering problem has been solved by utilizing a heat propagation procedure in an n-dimensional CA framework where n is the number of attributes in the data. The main advantage achieved against the well-known clustering algorithms is that no distance metric is needed in the proposed algorithm. Therefore, clustering process becomes easier for big datasets. However, space complexity of the method proposed in [4] is $O(m^n)$ where m is the number of cells in each dimension and n is the number of attributes. Therefore the method is applicable to low dimensional datasets. In this study, a similar procedure with [4] is utilized, but operation is carried out on two dimensional CA. Hence, the method is applicable to high dimensional datasets, too.

3. Methodology

Cellular Automata provide a framework in which each cell interacts with neighbour cells based on a set of predefined rules. There are two common neighbourhood relationships utilized: the Moore and Von Neumann which are shown in Fig. 1. In n-dimensional space, Von Neumann neighbourhood has $2n$ neighbours, the Moore has $3^n - 1$. In this study, linear neighborhood is utilized in 2D-CA where each cell interacts with only two neighboor cells in the same row.

As noted in Section 1, a separate 2D-CA is utilized for each class in this study. Thereby, characteristics of each class is obtained af-

Algorithm 1: Overall algorithm of the training process.

Function *Train(D_{train},n,c,CA)*:
  D_{train}: The training dataset
  n : Number of attributes in the dataset
  c : Number of classes in the dataset
  CA : An array of c Cellular Automata
  foreach *data_instance* $\in D_{train}$ **do**
    label=$data\_instance[n+1]$
    Map-Data($data\_instance, n, CA[label]$);
  end
  for $i \leftarrow 1$ *to* c **do**
    Set-Heat($CA[i]$);
    Distribute-Heat($CA[i]$);
  end
Function *Map-Data (d,n,A)*:
  for $i \leftarrow 1$ **to** n **do**
    $index = \texttt{FindIndex}(i, D_i)$;
    `// Increment the counter of the corresponding cell in A`
    $A[i][index]_{counter} + +$;
  end
Function *Set-Heat(A)*:
  foreach *cell* $\in A$ **do**
    $cell_{temp} = \ln(cell_{Counter} + 1)$
  end
Function *Distribute-Heat(A)*:
  Generate-Heat-Waves(A);
  while *A has heat to distribute* **do**
    foreach *cell* $\in A$ **do**
      if *cell has incoming_heat* **then**
        IncreaseHeat($cell, incoming\_heat$);
        TransferHeat($cell, incoming\_heat$);
      end
    end
  end

Table 1
Datasets utilized in the experiments.

Data set	# of Data instances	Attributes	Classes
Australian	690	14	2
Banknote	1372	4	2
Breast-Wisconsin	699	9	2
Glass	214	9	6
Haberman	306	3	2
Heart	270	13	2
Iris	150	4	3
Pima	768	8	2

Table 2
Results obtained by the CA-classification on the selected datasets.

Datasets	Average (%)	Best (%)	Runtime
Australian	78.86	86.95	0.29s
Banknote	86.75	89.78	0.23s
Breast-Wisconsin	95.74	98.56	0.22s
Glass	62.68	76.56	0.28s
Haberman	73.87	82.41	0.09s
Heart	76.77	90.12	0.19s
Iris	91.35	97.77	0.13s
Pima	66.32	73.91	0.21s

Table 3
Effect of categorical/continuous attributes in the warming procedure.

Dataset	Attribute no	Type	Used cells
Australian	1	Categorical	8%
	2	Continuous	88%
	4	Categorical	11%

Table 4
Datasets utilized to test runtime performance.

Dataset	Attributes	Classes	# of Data instances
Dataset 1	3	5	166,673
Dataset 2	5	8	198,678
Dataset 3	8	10	364,916
Dataset 4	12	12	110,547

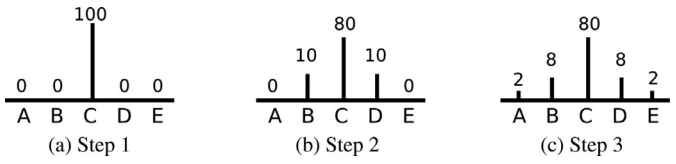


Fig. 3. Heat transfer procedure.

ter the CA is heated by using the instances of the corresponding class. Each row of the 2D-CA is associated with an attribute in the dataset. Then each instance in the class associated with this automaton is mapped to the cells of the automaton based on the attribute values of the instance. Each cell in the automaton is heated according to the number of data instances it is assigned to. Hence, each cell keeps a temperature value and a counter. The counter of a cell denotes the number of data instances assigned to it. Certainly, temperature is the corresponding heat of the cell created by the warming procedure.

The general pipeline of the proposed approach is as follows; at first, the dataset is preprocessed and it is divided into training and test sets. Then, a characteristic heat map is obtained for each class in the dataset by the warming procedure utilized. Finally, the test process is handled.

3.1. Preprocessing

Initially, the dataset is shuffled and divided into training and test sets where the training set consits of 70% of the initial set and the test set contains 30%. Then c distinct 2D-CA are created with the size of n rows and m columns where c is the number of classes and n is the number of attributes in the dataset. m is a user parameter as noted in Section 1. Training the automaton for each class consists of 3 stages. First, the data instances in the training set are mapped to the corresponding cells in the automaton associated with their classes. Then the temperature values are set for each cell in the CA and finally the heat is distributed among the cells based on the rule that is explained in Section 3.4.

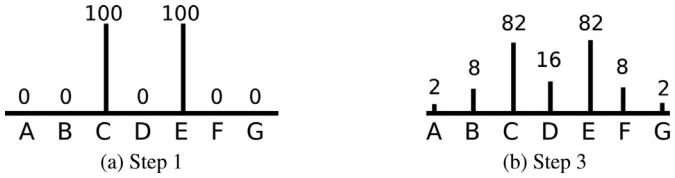


Fig. 4. Heat transfer procedure.

3.2. Mapping data to CA cells

Note that a seperate CA is heated for each class and each row in the automaton is associated with an attribute of the dataset. A data instance is mapped to a distinct cell in each row based on its corresponding attribute value. A counter is kept in each cell since more than one instance can be assigned to the same CA cell. A data instance is a vector $(x_1, x_2, \ldots, x_n, l)$ where x_i is the i^{th} attribute of the instance and l is the class label. Determining the corresponding cell for x_i is carried out by using the Eq. (1). MIN_i and MAX_i are the minimum and maximum values of the ith attribute in the training set. m is the number of cells in the corresponding CA row. m is a critical parameter in the algorithm that can affect the results drastically. If m is too large, then all data instances might be mapped to separate cells. In such a case, the relationship between similar items would be lost. On the other side, if the number of cells in a row is low, then irrelevant groupings of data instances might occur in the automata.

In Fig. 2, an example data instance with three attributes is given. Therefore the autamaton has 3 rows. m is chosen as 5 in this example and hence 5 cells exist in each row of the automaton. It is assumed that the minimum and maximum values for all attributes are 1 and 10. With the mapping procedure, the data instance is mapped to different cells in each row of the automaton based on the attribute values as seen in the figure.

$$x_i^{index} = \left\lfloor \frac{x_i - MIN_i}{(MAX_i - MIN_i)/(m-1)} \right\rfloor + 1 \tag{1}$$

3.3. Setting temperature values in the CA

The mapping process described in the previous section determines the counter values of the cells in the CA for each class. Then the temperature values of the cells are set based on these counter values. This is done by using a function that converts the counter value into a temperature. The logarithmic function presented in Eq. (2) is utilized for this purpose. In the Equation, C denotes a cell in the CA, and $C_{Counter}$ is the counter value set for this cell. Certainly, the cells would have higher temperatures when they are assigned more instances. On the other side the logarithmic scale utilized would prevent the cells to have extreme temperatures due to large number of data instance assignments.

$$C_{temp} = ln(C_{Counter} + 1) \tag{2}$$

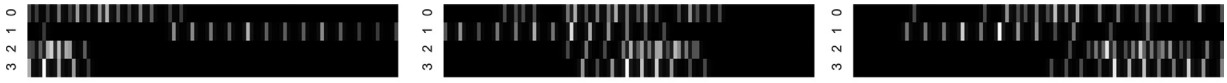
There are two other user parameters utilized for the process. These are named as *range* and *portion*. The first parameter *range* denotes the percentage of cells that will be utilized in the heat transfer process. For instance, if the number of cells in a row is 100 and if the range is 0.1, then a cell can transfer heat energy to the ten closest cells on the left and on the right. On the other side *portion* is the percentage of energy that will be transferred to the neighbor cells. The use of these two parameters are described in detail in Section 3.4.

3.4. Distributing the heat in the CA

The counter values in the CA cells in fact denote the frequency of each attribute value in the classes. Therefore an initial repre-

Table 5
Performance of other classification algorithms.

Method	Iris	Haberman	Banknote	Glass	Brest-W.	Pima	Heart	Australian
Naive Bayes	96	76.47	88.04	47.66	97.42	-	75.36	85.20
Decision Table	92.67	-	-	68.22	-	-	85.07	82.20
J48	96	-	-	69.68	-	-	85.79	-
Random Forest	96	-	-	80.37	-	-	86.08	84.40
ZeroR	33.33	-	-	35.51	-	-	55.50	-
MLP	96	-	95.81	65.88	-	-	85.79	-
GP	96	-	-	66.82	-	-	87.10	-
Boosting	-	-	-	-	-	-	-	82.20
Bagging	-	-	-	-	-	-	-	84.30
SVM	-	77.12	-	-	-	-	-	-
ID3	-	-	-	-	92.99	-	-	-
C4.5	-	-	-	-	95.57	-	-	-
CART	-	-	-	-	92.42	-	-	-
PLS-LDA	-	-	-	-		74.4	-	-
FMM	-	-	-	-		69.28	-	-
LDA-SVM	-	-	-	-	-	75.65	-	-



(a) setosa class (b) versicolor class (c) virginica class

Fig. 5. Heat distributions of each class at the end of training process.

Table 6
Runtime statistics.

Algorithm	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
	Runtime(s)	Success(%)	Runtime(s)	Success(%)	Runtime(s)	Success(%)	Runtime(s)	Success(%)
CA-classification	0.27	99.95	0.52	99.89	1.28	99.79	0.52	99.78
Naive Bayes	0.39	99.99	1	99.98	3.12	99.99	1.54	99.99
ZeroR	0.04	31.65	0.07	26.55	0.15	14.72	0.04	12.52
Decision Table	2.89	99.22	36.41	95.22	1005.96	85.64	124.25	73.46
Random Forest	26.84	99.99	61.39	99.97	230.32	99.98	53.97	99.94
MLP	80.14	99.99	174.52	99.96	535.2	99.97	256.84	99.87

Table 7
Analysis of the number of cells utilized in the automata.

Datasets	Results	Number of cells		
		100	500	1000
Australian	Average Success (%)	79.42	78.68	78.4
	Runtime(s)	0.29	0.86	2.54
Banknote	Average Success (%)	87.07	79	73.67
	Runtime(s)	0.28	0.47	1.5
Breast-W.	Average Success (%)	95.78	95.61	95.73
	Runtime(s)	0.25	0.52	1.4
Glass	Average Success (%)	61.7	55.2	52
	Runtime(s)	0.32	1.11	4.07
Haberman	Average Success (%)	74.02	73.48	73.58
	Runtime(s)	0.15	1.11	4.07
Heart	Average Success(%)	77.4	77.71	77.61
	Runtime(s)	0.26	0.65	2.06
Iris	Average Success(%)	91.88	91.37	91.75
	Runtime(s)	0.14	0.35	0.95
Pima	Average Success(%)	65.93	66.33	66.29
	Runtime(s)	0.27	0.49	1.73

sentation of the data distribution is obtained in the CA after this mapping process. As explained in the previous subsection, these frequency values are converted to temperature values in the next step. Then the heat energy represented by these temperature values are distributed to other cells in the same row. Hence, if there is a group of nearby cells that have high temperature values, they would increase temperature of other cells in the same region even if these cells are are not assigned a data instance at all.

The heat transfer procedure is a critical part of the methodology. As noted above the counter values only denote the frequency of attribute values in the data. These frequencies can be helpful to classify categorical data, but they will not be sufficient for continuous data. The heat transfer procedure explained above enables the system to form patterns in the automaton that will represent the characteristics of the classes in the original data.

Heat is transferred by using the parameter *portion* given by the user. The *portion* determines how much heat a cell will transmit to the neighbours. The warming procedure is illustrated in Figs. 3 and 4. For this procedure the portion is used as 20%. In the first part of Fig. 3, a single row of an automaton with five cells is presented and only the middle cell (C) has a temperature value. Each cell that has a temperature value emits a heat wave in two directions (left and right) according to the algorithm utilized. These heat waves spread to the neighbor cells until they reach to the borders of the area determined by the *range* parameter. This procedure is simulated in the automaton with the following rule. The cells that have a temperature value emit 20% of their energy to the left and right neighbors. That is why in Fig. 3(b), temperature of cell C has dropped down to 80° whereas the temperatures of cells B and D are increased to 10°. However, cells B and D also emit 20% of the energy they have received *(into the direction of the original wave only)* and the final configuration in the third part of the figure is obtained. The heat transfer procedure would stop at this point assuming that A and E are the border cells determined by the *range* parameter.

The heat waves from different sources can overlap in the CA and this situation is illustrated in Fig. 3. In the fist part of the Figure the initial configuration of an automaton with 7 cells is pre-

Table 8
Analysis of the *range* parameter on selected datasets.

Datasets	Results	Range									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Australian	Average Success (%)	79.42	79.8	79.58	79.83	80.05	79.03	79.24	79.79	78.37	79.74
	Runtime(s)	0.29	0.3	0.36	0.48	0.39	0.44	0.43	0.45	0.41	0.42
Banknote	Average Success (%)	87.07	86.95	87.24	86.89	87.27	87.14	86.9	86.84	86.8	86.76
	Runtime(s)	0.28	0.3	0.29	0.28	0.28	0.37	0.35	0.32	0.29	0.29
Breast-Wisconsin	Average Success (%)	95.78	95.66	95.89	95.77	95.79	95.54	95.85	95.62	95.61	95.63
	Runtime(s)	0.25	0.22	0.23	0.26	0.27	0.28	0.28	0.29	0.28	0.29
Heart	Average Success (%)	77.4	76.49	76.88	77.8	76.56	77.41	75.86	75.91	77.58	76.48
	Runtime(s)	0.26	0.26	0.29	0.33	0.3	0.35	0.36	0.35	0.33	0.33
Iris	Average Success (%)	91.88	91.73	91.57	91.19	92.37	91.4	91.33	92	91.75	91.64
	Runtime(s)	0.14	0.15	0.15	0.15	0.16	0.18	0.18	0.16	0.16	0.17

sented. As seen in the figure the cells *C* and *E* are the heat sources. The final configuration is given in Fig. 4(b). Note that the temperature of cell *D* is higher compared to the temperatures of cells *A, B, F* and *G* since it exists between the two heat sources. What is more the cells *C* and *E* end up with temperature 82° instead of 80° in the previous example since they are close enough to exchange energy between each other.

The training process is presented in Algorithm 1. *Train* is the main function in the algorithm. *CA* is a data structure that contains *c* cellular automata where *c* is the number of classes. As explained in the previous sections, for each data instance the class label is determined and then the data instance is mapped to the cells of an automaton based on its class label. After the mapping process, the temperature values of the cells in the CA are determined. Lastly, heat distribution process is carried out on each automaton one by one. In *Distribute – Heat* function, all cells that have a temperature value larger than 0° generate the heat waves. Then each cell checks if there is an incoming heat wave and if it is so, 80% of this energy is preserved and the rest is transmitted to the neighbor cell in the same direction. This distribution process stops when all waves reach to the borders of the region in the automaton determined by the *range* parameter. In Fig. 5, the heat distributions formed in CA for each class at the end of training process is presented for the iris dataset. Cells with lighter tones have higher temperatures. The rows in the CA represent the four attributes in the dataset.

```
1  Function Classify(CA,v,c,n):
       CA : An array of c Cellular Automata
       v  : Data instance
       c  : Number of classes in the dataset
       n  : Number of attributes in the dataset
2      max_heat = −∞;
3      class_label;
4      foreach A ∈ CA do
5          heat = Total-Heat-Value(A,v,c,n);
6          if heat > max_heat then
7              max_heat = heat;
8              class_label = A_label;
9          end
10     end
11     return class_label;
12 Function Total-Heat-Value(A,v,c,n):
13     total_heat = 0;
14     for i ← 1 to n do
15         index = FindIndex(i, v_i);
16         total_heat +=A[i][index]_temperature;
17     end
18     return total_heat;
```

3.5. Testing procedure

After training process is completed, the success rate of the system is determined by using the test set. In order to classify the instances in the test set, each data instance is mapped to all CA one by one again by using the Eq. (1). This time, the temperature values in the corresponding cells of the automaton are summed up in each CA. Hence, a total temperature value is obtained for each CA (hence for each class) for the test data instance. Certainly, the instance is labelled with the class corresponding to the CA that provides the maximum total temperature for the instance.

Algorithm 2 presents this procedure. Again *CA* is the data structure that holds *c* different automata where *c* is the number of classes in the dataset. *Classify* is the main function which calculates the total temperature for a data instance for all automata one by one and then the data instance is classified based on this total temperature value.

4. Experimental results

Eight data sets from UCI data repository are used for evaluating the method [16]. The properties of the datasets are presented in Table 1. As noted in Section 3, 30% of the initial dataset is selected to form the test set. For each dataset, the algorithm is run 100 times with randomly selected training and test sets. Table 2 presents the results obtained on each dataset. The runtime statistics presented in the table are the average performance of the algorithm in these 100 runs.

The results in Table 2 are obtained by the following parameters of the algorithm. The number of columns denoted as *m* in the previous section is set to 100 in all experiments. The temperatures of the cells are determined by the logarithmic function presented in Eq. (2). The cells keep 80% of their energy and propagate 20% of it to neighbor cells. And the range is used as 10%.

Table 9
Performance measures on selected datasets.

Datasets	Results(%)	Algorithms					
		CAC	Naive Bayes	MLP	Decision Table	ZeroR	Random Forest
Australian	Accuracy	79.42	86.8108	84.3954	86.2792	55.5073	87.5843
	Precision	79.90	86.2756	85.2382	88.257	55.5073	88.95
	Recall	79.42	90.6918	87.0328	87.122	100	88.6903
	F1 Score	79.66	88.392	86.0524	87.556	71.3886	88.7693
Banknote	Accuracy	87.29	91.5917	99.7568	93.5359	55.5285	99.052
	Precision	87.27	92.17	99.7817	94.1594	55.5285	99.4286
	Recall	87.29	92.8221	99.7811	94.3109	100	98.8619
	F1 Score	87.28	92.4545	99.7811	94.1887	71.4062	99.1427
Breast-Wisconsin	Accuracy	95.59	97.4344	95.9138	95.2006	65.4945	96.5313
	Precision	95.55	99.1862	97.0256	95.7124	65.4945	97.8151
	Recall	95.59	96.8809	96.7349	97.0263	100	96.8809
	F1 Score	95.57	98.0163	96.8736	96.3593	79.1499	97.3373
Heart	Accuracy	77.77	81.9753	77.1605	79.3827	55.5556	82.4691
	Precision	77.90	81.7335	79.5877	79.1566	55.5556	81.3244
	Recall	77.77	87.5556	80	86.4444	100	89.5556
	F1 Score	77.83	84.4023	79.4688	82.3276	71.4286	85.0569
Iris	Accuracy	92.00	94.6667	95.1111	94.4444	33.3333	95.3333
	Precision	91.55	100	100	100	33.3333	100
	Recall	92.00	100	100	100	100	100
	F1 Score	91.77	100	100	100	50	100

The results of other classification algorithms on the same datasets are presented in Table 5. These results are collected from different studies in the literature, Gupta [10], Ghazvini et al. [9], Verma and Mehta [24], Shruti [22], Shajahaan et al. [21], and Parashar [17]. It can be seen that the CA-classification method proposed in this study produces results compatible with the well-known algorithms in the area. There are even cases where CA-classification is better than some other methods. For example, with the Iris data set, the result obtained with CA-classification is close to the output of Decision Table and and dummyTXdummy- it is better than what ZeroR achieves. Also, CA-classification is better than ID3 and CART methods on Breast-Wisconsin dataset. Lastly, Naive Bayes classification is worse than CA-classification on Glass dataset.

In Table 9, the proposed approach has also been compared with some well known algorithms in terms of accuracy, precision, recall, and F1-score. The performance measures of the other algorithms have been obtained by executing WEKA [11] 10 times.

It should be stated that CA-classification gives better results on datasets that has continuous attributes. Certainly, the variation of numerical or categorical attributes is smaller compared to continuous attributes. In this case, only a specific subset of the cells are heated in the automaton and it becomes more difficult to obtain an adequate characteristic heat map for the classes when such attributes are dominant in the dataset. This aspect is analyzed in Table 3. As it can be seen in the table, in the Australian dataset, the second attribute is continuous and and the warming process assigns a temperature value to 88% of the cells in the corresponding row of the automaton. However, the first attribute in the dataset has only 2 distinct values (0, 1) and fourth one has 3 distinct values (0, 1, 2). For these two attributes a smaller area of the corresponding rows are heated compared to the continuous attribute. Hence, the heat distribution obtained may not be sufficient for a successful classification in such cases.

The datasets utilized in the experiments are widely used for testing purposes in the literature. However, the sizes of these datasets are quite limited. Therefore, some large datasets are generated in order to test the runtime performance of the algorithm. The new datasets are generated by using the Gaussian generator and the properties of the datasets can be seen in Table 4 [12].

The statistics for the other algorithms are obtained by using the WEKA software again [11]. Tests are run 10 times on each set for all methods. The results obtained can be seen in Table 6. ZeroR has the best runtime performance as seen in the table. However, the success rate of ZeroR is not sufficient on these datasets. If ZeroR is excluded, the CA-classification seems to have the best performance. The difference between the runtime performance of CA-classification and Naive Bayes is low, but still it can be claimed the CA-classification is slightly faster. There is a significant difference between CA-classification and other algorithms in terms of runtime performance. Also, it should be noted that the algorithms exhibit a similar performance in terms of accuracy except ZeroR on these datasets.

4.1. The effects of system parameters on the results

Different parameters are utilized in the algorithm. Experiments are also conducted to see the effects of these parameters and to determine if different parameter values would result drastic changes on the output of the system.

Table 8 presents the results obtained with different *range* parameter values. It can be claimed that changing the *range* value merely affects the success rate and the runtime of the method. Table 7 shows the variation in the output of the method when the amount of cells that exist in the automaton is altered. As seen in the table, increasing the number of cells has a negative effect on the results. Both runtime and accuracy decline with larger automata. The runtime performance worsens since the algorithm needs more iterations to dissipate heat among the cells of the automaton in this case. Also, a more sparse heat map is obtained with larger number of cells and this results a drop in the accuracy. Lastly, it should be noted that changing the *portion* parameter value does not have an important effect on the results.

5. Conclusion

In this study, a new approach has been developed for the classification task based on 2D-CA. A novel method is utilized in the algorithm to warm the cells of the automata and obtain a characteristic heat map for the classes in the data. The process of forming characteristic heat maps for the classes in the data is a novel approach that has not been proposed previously in the literature. As discussed in Section 4, the runtime performance of the proposed approach has an advantage against the well-known algorithms, especially when the big-data issue is handled. The implementation

of the proposed approach does not require too much effort since it simply requires two processes: mapping data instances into CA and heat transfer.

As future work, more effective heat distribution functions can be designed. Note that the same number of columns is used in each row of the automaton. The number of columns might be dynamic based on the range or type of the attributes.

The method proposed is open to parallelization. The operations carried out in each row of the automaton are independent of the operations that take place in other rows. The warming process of different automata and different rows can be carried out concurrently. Hence, using a multi-core environment would improve the runtime performance of the algorithm considerably.

Furthermore, deep learning approaches that can automatically extract relevant features from the data are very popular in the literature. We will consider adopting our method so that it can be utilized for feature selection or feature extraction procedures. Yet, the current model can be used as a fast classifier in a deep learning framework.

References

[1] M. Afshar, J. Amani, A. Mahmoodpour, Cellular automata approach in solution of heat conduction problem (2009).

[2] J.I. Barredo, M. Kasanko, N. McCormick, C. Lavalle, Modelling dynamic spatial processes: simulation of urban future scenarios through cellular automata, Landsc. Urban Plan 64 (3) (2003) 145–160.

[3] S. Bobkov, Cellular automata systems application for simulation of some processes in solids (2012).

[4] E.B. Dündar, E.E. Korkmaz, Data clustering with stochastic cellular automata, Intell. Data Anal. 22 (3) (2018).

[5] M. Esmaeilpour, V. Naderifar, Z. Shukur, Cellular learning automata approach for data classification, Int. J. Innov. Comput.Inf. Control 8 (12) (2012) 8063–8076.

[6] T. Fawcett, Data mining with cellular automata, ACM SIGKDD Explor. Newsl. 10 (1) (2008) 32–39.

[7] M.A. Friedl, C.E. Brodley, Decision tree classification of land cover from remotely sensed data, Remote Sens. Environ. 61 (3) (1997) 399–409.

[8] M. Gardner, Mathematical games: the fantastic combinations of john conways new solitaire game life, Sci. Am. 223 (4) (1970) 120–123.

[9] A. Ghazvini, J. Awwalu, A.A. Bakar, Comparative analysis of algorithms in supervised classification: a case study of bank notes dataset, Computer Trends and Technology 17 (1) (2014) 39–43.

[10] A. Gupta, Classification of complex uci datasets using machine learning and evolutionary algorithms, Int. J. Sci. Technol.Res. (IJSTR) 4 (5) (2015) 85–94.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explor. 11 (1) (2009) 10–18.

[12] J. Handl, Cluster generators, 2018, http://personalpages.manchester.ac.uk/mbs/julia.handl/generators.html. [Online; accessed 12-March-2018].

[13] C.-W. Hsu, C.-C. Chang, C.-J. Lin, et al., A practical guide to support vector classification (2003).

[14] C. Juraj, P. Michal, Cellular automata for earth surface flow simulation (2011).

[15] P. Kokol, P. Povalej, M. Lenic, G. Stiglic, Building classifier cellular automata., in: P.M.A. Sloot, B. Chopard, A.G. Hoekstra (Eds.), ACRI, Springer, 2004, pp. 823–830.

[16] M. Lichman, UCI machine learning repository, http://archive.ics.uci.edu/ml 2013.

[17] R. Parashar Burse, A comparative approach for pima indians diabetes diagnosis using lda-support vector machine and feed forward neural network, Int. J. Eng. Res.Technol. (IJERT) 3 (10) (2014) 1192–1194.

[18] T.R. Patil, S. Sherekar, Performance analysis of naive bayes and j48 classification algorithm for data classification, Int. J. Comput. Sci.Appl. 6 (2) (2013) 256–261.

[19] P. Povalej, P. Kokol, T.W. Družovec, B. Stiglic, Machine-learning with cellular automata, in: International Symposium on Intelligent Data Analysis, Springer, 2005, pp. 305–315.

[20] M. Qudsia, M. Saeed, Gaussian cellular automata model for the classification of points inside 2d grid patterns, in: Frontiers of Information Technology (FIT), 2017 International Conference on, IEEE, 2017, pp. 350–355.

[21] S.S. Shajahaan, S. Shanthi, V. ManoChitra, Application of data mining techniques to model breast cancer data, Int. J. Emerg. Technol.Adv. Eng. 3 (11) (2013) 362–369.

[22] K. Shruti, Comparative study of advanced classification methods, Int. J. Recent Innov.Trends Comput. Commun. 3 (3) (2015) 1216–1220.

[23] B.S. Soares-Filho, G.C. Cerqueira, C.L. Pennachin, Dinamicaa stochastic cellular automata model designed to simulate the landscape dynamics in an amazonian colonization frontier, Ecol. Modell. 154 (3) (2002) 217–235.

[24] M. Verma, D.D. Mehta, A comparative study of techniques in data mining, Int. J. Emerg. Technol.Adv. Eng. 4 (4) (2014) 314–321.