To Better Handle Concept Change and Noise: A Cellular Automata Approach to Data Stream Classification

Sattar Hashemi[1], Ying Yang[1], Majid Pourkashani[2], and Mohammadreza Kangavari[2]

[1] Clayton School of Information Technology
Monash University, Australia
{Sattar.Hashemi,Ying.Yang}@infotech.monash.edu.au
[2] Computer Engineering Department, Iran University
Of Science and Technology, Tehran, Iran
{mpkashani,kangavari}@iust.ac.ir

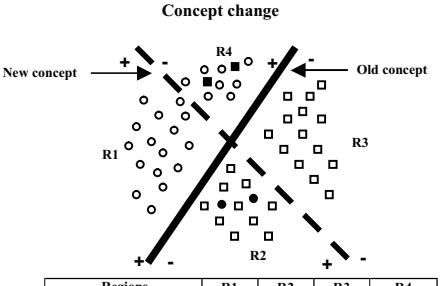
Abstract. A key challenge in data stream classification is to detect changes of the concept underlying the data, and accurately and efficiently adapt classifiers to each concept change. Most existing methods for handling concept changes take a windowing approach, where only recent instances are used to update classifiers while old instances are discarded indiscriminately. However this approach can often be undesirably aggressive because many old instances may not be affected by the concept change and hence can contribute to training the classifier, for instance, reducing the classification variance error caused by insufficient training data. Accordingly this paper proposes a cellular automata (CA) approach that feeds classifiers with most *relevant* instead of most *recent* instances. The strength of CA is that it breaks a complicated process down into smaller adaptation tasks, for each a single automaton is responsible. Using neighborhood rules embedded in each automaton and emerging time of instances, this approach assigns a relevance weight to each instance. Instances with high enough weights are selected to update classifiers. Theoretical analyses and experimental results suggest that a good choice of local rules for CA can help considerably speed up updating classifiers corresponding to concept changes, increase classifiers' robustness to noise, and thus offer faster and better classifications for data streams.

1 Introduction

In data streams, the concept underlying the data may change over time, which can cause the accuracy of current classifiers to decrease. Meanwhile, real-world data are seldom perfect and often suffer from significant amount of noise, which may affect the accuracy of induced classifiers. Dealing with concept changes and differentiating them from noise has become an interesting and challenging task in the machine learning and data mining community [5,7,10]. The traditional approaches proposed for mining data streams, incremental and ensemble classifiers, are mainly based on windowing [5,8,9]. In both cases the implicit idea underlying the method is that *the more recent the data, the more relevant they are to test/train the current learner.*

M.A. Orgun and J. Thornton (Eds.): AI 2007, LNAI 4830, pp. 669–674, 2007.

This paper suggests that the arrival time of an instance is not always the best relevance criterion in changing environments because it decreases the number of instances on which the classifier is induced and hence, affects the overall performance of the classifier. To be more illustrative, let's consider a generic concept change scenario as depicted in Figure 1. The feature space is a 2-d plane with two linearly separable classes, positive and negative, whose instances are shown by circles and squares respectively. The solid line represents the "old" concept before the concept changes, where regions R1 and R4 are positive, and regions R2 and R3 are negative. Instances of the old concept are shown by "empty" shapes. The dashed line represents the "new" concept after the concept changes, where regions R1 and R2 are positive, and R3 and R4 are negative. Instances of the new concept are shown by "filled" shapes, in contrast to the older "empty" ones. Now assume that the old concept is in effect (the solid line) and the algorithm detects concept change when new instances (filled shapes) fall into regions R2 or R4 with unexpected class labels. Once the concept change is detected, the windowing approach will indiscriminately remove instances from every region as long as they are not most recent regardless of whether they are still valid for the new concept. As a result, the windowing approach will unwisely reduce the amount of training data, undermine the learner's ability to approximate the new concept and hence increase the classification error [5, 6].



Regions	R1	R2	R3	R4
Labels in old concept	+	-	-	+
Labels in new concept	+	+	-	-

Fig. 1. Regions R1 and R3 are not affected by the concept change. Their instances are still valid for the new concept and hence should be utilized to update classifiers.

To handle the above problems, the proposed method in this paper uses a cellular automata (CA) approach. Using simple neighborhood rules, The CA approach will identify those negative instances (empty squares) in R2 and those positive instances (empty circles) in R4 are no longer relevant to the learning task. Those instances will be removed while instances in regions R1 and R3 will be retained for updating classifiers. This property provides the leaner with more relevant training instance, which can reduce its classification variance. Meanwhile, noisy instances can be discarded even when they are most recent data. The strength of CA is that it breaks a complicated process down into smaller adaptation tasks, for each a single automaton is responsible. Each individual automaton is only involved with the simple rules which are applied locally to update the overall automata's states. The interested reader can refer to our last research for further information [5].

The rest of the paper is organized as follows. Section 2 offers a formal representation of the CA approach to suppress noise and handle concept change in data streams. Section 3 presents experimental results and analyses. Section 4 gives concluding remarks and suggestions for future work.

2 Methodology

A cellular automata is a discrete dynamical system and includes a regular spatial lattice (gird) of points [1,2,3]. Each point, called a cell or an automaton, is a finite state machine. The inputs to a cell are the states of all cells in its neighborhood. The states of the cells in the lattice are updated according to a local rule. That is, the state of a cell at a given time depends only on its own state and its nearby neighbors' states one time step before. Suppose the instance s_i of the data stream S can be identified by the *d*-tuple of attribute values $(\alpha_1,\alpha_2,...,\alpha_d)$. This *d*-tuple represents a point in a *d*-dimensional problem space. Let $G=\{c_k=(D,t,w,c)\mid D=(\alpha_1,\alpha_2,...,\alpha_d)\}$ be a d-dimensional grid (lattice) of cellular automata, which has a one-to-one relation with the instances in the problem space, and c_k represent the k^{th} cell or automaton in sequential indexing of the cellular automata of concern. Each automaton is assigned three parameters that describe the state of its corresponding data point: *t* as *timetag*, *w* as *weight*, and *c* as *class*. *Timetag* is a monotonically decreasing value that shows the recency of the data. *Weight* is a parameter that accounts for the relevance of the automaton to the current concept. *Class* is the true label of the instance that is used afterwards to validate the learnt concept. We use the super-index notation to refer to parameters of a cell. Thus, c_k^t means the timetag of the cell c_k, c_k^D stands for its corresponding instance, and c_k^w for its weight. Whenever a new instance streams in, the corresponding cell is activated and its timetag and weight are initialized to a pre-defined constant *T* and *W* respectively. These parameters for other active cells are decreased by a pre-defined forgetting factor λ:

$$(c_k^t \leftarrow c_k^t - \lambda \ and \ \ c_k^w \leftarrow c_k^w - \lambda) \ \forall k \ s.t. \ c_k \ is \ active \ \ cell \ . \tag{1}$$

As soon as the timetag and weight of an active cell reach zero, the cell is deactivated. It is worth mentioning that the weight of each cell can also be altered by the local rules of the cells in its neighborhood. Local rules are simple heuristics that aim at feeding classifiers with more relevant instances by taking into account the local situations of the instances in the current concept with respect to each other. By adding local rules, our CA based approach is more appropriate than the naïve recency-based windowing approach. Different rules can be defined depending on the purpose. The only restriction on these rules is the one that has been set by the CA theory: locality. We have adopted a generalized Moore neighborhood definition [4], which includes cells in a hyper-sphere that is centered at the base cell and with a radius of *n* cells.

To deal with noise, our approach simply checks for local class disagreements. For each instance, its true class is checked against the class of its neighbors. If most of the active neighbors have a different class, this instance is considered as noisy. Its weight

is suppressed to an amount below the selection threshold. As a consequence, this instance is moved out of the selection basket of learning. This strategy is based on the heuristic "A positive sample will rarely emerge in-between many negative samples, and vice versa".

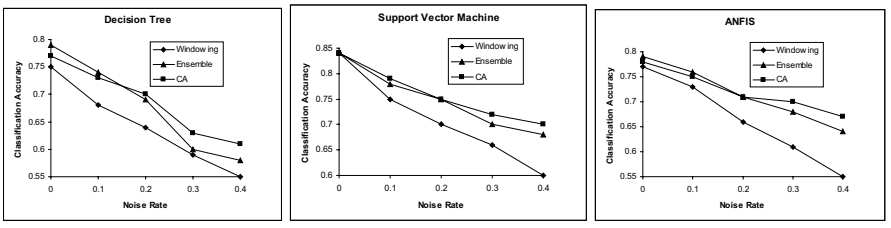
To deal with concept change, an important problem to solve is to differentiate concept change from noise because both affect classifiers' performance but in different ways. Effect of the concept change in a particular neighborhood is consistent while it is not the case for noise. Our proposed method adopts this local heuristic to detect concept change: "A newly misclassified instance, whose class label is different from most neighboring instances but is supported by coming instances from the stream, indicates a concept change". To distinguish between concept change and noise, a two-tailed binomial sign test is applied on misclassification sequence of the classifier. If the result is less than the critical level of 0.05, the drop in accuracy is the effect of concept change, otherwise it is due to noise. Whenever a concept change is detected, the misclassified new instances are considered as representatives of the new concept, while their nearby instances with a different class are considered as representatives of the outdated concept and thereafter suppressed.

3 Experiments

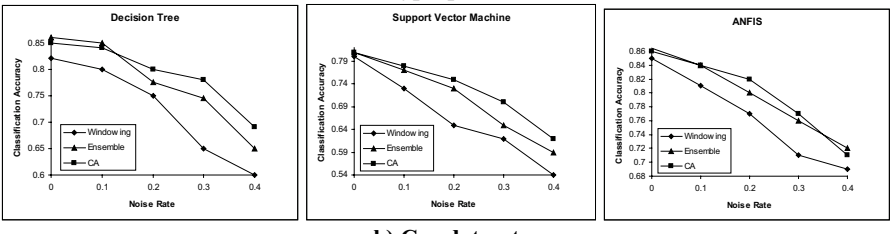
Experiments are conducted to compare the cellular automata approach with the windowing and the ensemble methods for dealing with noise and handling concept change in both synthetic and real-world data streams. Evaluations involve decision trees (DT), support vector machines (SVM) and adaptive neuro-fuzzy system (ANFIS), and also, two public benchmark datasets Hyperplane and Car dataset are used for evaluation [5]. They are relatively large, have concept changes, and are commonly used in published research of data stream classification.

To observe the role that local rules play in dealing with noise, we add different levels of noise (10%, 20%, 30% and 40%) into the data [5]. The presented results demonstrate alternative methods' behavior in face of both concept changes and different levels of noise. The CA removes instances that it has identified as noise and feed other instances to classifiers. The predictive performance of each classification algorithm (DT, SVM and ANFIS) under the windowing, the ensemble and the CA strategy respectively are compared in Figure 2. As the graphs show, the proposed CA approach yields better classification accuracy, which suggests that the local rule heuristic is more robust to noise. Although the ensemble approach performs better at first, but, it is outperformed by the CA approach in noisy environment what approves our early discussion that relying on the most recent instances is not always the best strategy in noisy environments.

To compare the performance of different approaches on handling concept change in data streams, we use the Hyperplane dataset [8] because we can manipulate different amount of concept changes. In contrast, the amount of concept changes in real-word data such as the Car dataset can not be changed. In particular, the amount of concept changes in Hyperplane relates to the number of dimensions k whose weights are changing over time. Using four different values of k, we produce several



a) Hyperplane dataset



b) Car dataset

Fig. 2. Classification accuracy of different classification algorithms under the windowing, the ensemble and the CA approaches as a function of noise level

Table 1. Experiment results in learning drifting concepts

Learner	k	Windowing Approach			Ensemble Approach			CA Approach		
		acc	*ct*	$N_{\hbar}$	*acc*	*ct*	$N_{\hbar}$	*acc*	*ct*	$N_{\hbar}$
Decision Tree	1	75.0%	43	70	79%	29	85	77.0%	31	58
	2	71.2%	47	79	73%	34	92	73.3%	37	67
	3	60.8%	72	95	65%	61	111	64.8%	57	74
	4	56.0%	75	104	60%	67	123	61.3%	60	86
SVM	1	84.0%	47	66	84%	34	80	84.0%	35	54
	2	73.0%	70	80	78%	47	89	77.7%	45	62
	3	69.7%	77	85	74%	50	100	76.0%	48	66
	4	58.0%	100	96	67%	67	119	68.3%	63	73
ANFIS	1	77.0%	33	69	79%	15	72	78.1%	20	56
	2	71.3%	40	77	74.5%	23	80	75.0%	23	73
	3	70.9%	41	86	72.9%	26	98	73.4%	25	78
	4	63.8%	57	102	65%	44	113	66.0%	40	83

Hyperplane datasets to evaluate the performance of our approach from different points of view. Three measures of the performance are calculated for each experiment: 1) convergence time, *ct,* which is the average time the system remains unstable during updating to a concept change, measured by the number of instances; 2) classification accuracy, *acc*, which is the percentage of the classifier's correct predictions, and 3) re-learning requirement, $N_{\hbar}$, as the total number of times the classifier is re-learned from scratch. Results are shown in Table 1. It is observed that almost in every case the CA approach outperforms the windowing and ensemble approaches: it achieves higher classification accuracy; spends less time on updating classifiers upon concept changes; and require less frequently building classifiers from

scratch. Lower ct and $N_{\hbar}$ are desirable because data streams demand fast online classification. In other words, whenever the other approaches present almost equal classification accuracy, it is at the cost of using more time resources.

4 Conclusion and Future Work

A cellular automata (CA) approach is proposed for data stream classifications. Its advantage is to use most relevant instances instead of most recent instances to update classifiers in face of concept changes. By using neighboring instances and simple local rules, the CA approach can be more robust to noise, achieve higher classification accuracy, adapt faster to changed concepts, and less frequently require building classifiers from scratch.

Using cellular automata for data stream classification is a new topic and there are various interesting issues to further investigate. For example, as cellular automata are distributed along a grid with just interactions between neighboring cells, they can be consider as a means for parallel stream mining. For another example, cellular automata have a discrete nature. Hence, an effective online method for discretizing continuous attributes in data streams may be considered for future work.

References

1. Farmer, J.D., Toffoli, T., Wolfram, S. (eds.) Cellular Automata.Proceedings of an Interdisciplinary Workshop, Los Alamos, New Mexico (March 7-11, 1983)
2. Wolfram, S. (ed.): Theory and Applications of Cellular Automata. World Scientific, Singapore (1986) Collection of papers on CA's
3. Folioni, G., Pizzuti, C., Spezzano, G.: A Cellular Genetic Programming Approach to Classification. In: Proceedings of the Genetic and Evolutionary Computation Conference, vol. 2, pp. 1015–1020 (1999)
4. Maji, P., Shaw, C., Ganguly, N., Sikdar, B.K., Chaudhuri, P.P.: Theory and Application of Cellular Automata for data mining. Fundamenta Informaticae (58), 321–354 (2003)
5. Hashemi, S., Yang, Y., Pourkashani, M., Kangavari, M.: To Better Handle Concept Change and Noise: A Cellular Automata Approach to Data Stream Classification. Technical Report, Monash University (2007)
6. Widmer, G., Kubat, M.: Learning is presence of concept drift and hidden contexts. Machine Learning 23(1), 69–101 (1996)
7. Klinkenberg, R.: Learning drifting concepts: example selection vs. example weighting, Intelligent Data Analysis. Special Issue on Incremental Learning Systems Capable of Dealing with Concept Drift 8(3) (2004)
8. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: SIGKDD. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 97–106 (2001)
9. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept drifting data streams using ensemble classifiers. In: SIGKDD. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 226–235 (2003)
10. Zhu, X., Wu, X., Yang, Y.: Effective Classification of Noisy Data Streams with Attribute-Oriented Dynamic Classifier Selection. Knowledge and Information Systems An International Journal (KAIS) 9(3), 339–363 (2006)