Active learning approach to concept drift problem

BARTOSZ KURLEJ and MICHAL WOZNIAK, *Department of Systems and Computer Networks, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland.*
E-mail: bartosz.kurlej@pwr.wroc.pl; michal.wozniak@pwr.wroc.pl

Abstract

The traditional pattern recognition method assumes that the model that is used does not depend on data timing. This assumption is correct for several practical issues but it is not valid for the case where new data frequently becomes available (the so-called *data streams*). We could meet the above-mentioned situation in many practical issues, as spam filtering, intrusion detection/prevention (IDS/IPS) or recognition of client behaviour to enumerate only a few. In these cases, the dependencies between the observation and classes are continually changing. Unfortunately, most pattern recognition methods do not take the so-called *concept drift* into consideration and they cannot adapt to a new model. Therefore, design classifiers for data streams are currently the focus of intense research. Another important issue related to the recognition of data streams is the problem of data labelling. Traditional machine learning methods use supervised learning algorithms, which could produce a classifier on the basis of a set of labelled examples. In this approach, we should take into consideration the cost of data labelling, which is usually passed over. Let us notice that labels are usually assigned by human experts and therefore they cannot label all new examples if they come too fast. Therefore, methods of classifier design which could produce the recognition system on the basis of a partially labelled set of examples (called *active learning*) would be an attractive proposition. This article focuses on the problem of the concept drift using active learning approach for the minimal distance classifiers. The potential for adaptation of the proposed method and its quality are evaluated through computer experiments, carried out on several benchmark data sets.

Keywords: Machine learning, pattern recognition, concept drift, active learning, minimal distance classifier, *k*-nearest neighbours.

1 Introduction and related works

Nowadays, the manual analysis of data collected by an average institution is virtually impossible, and for the efficient management, simple data analysis methods do not suffice because to make smart decisions, hidden knowledge must be extracted from the databases. One of the most popular data mining tasks is the classifier design [9]. The biggest disadvantage of most of these methods is that they 'assume' that the statistical dependencies between the observations of given objects and their classifications remain unchanged. In real situations, the so-called *concept drift* occurs frequently [12, 20]. The potential for considering new training data [23] is an important feature of machine learning methods used in security applications (like spam filters or IDS/IPS) [18] or decision support systems for marketing departments, which need to follow the changing client behaviour [10].

It is obvious that the smaller the data structures used by such systems in making decisions, the more likely it is that the systems can adapt. As an example, the minimal distance methods, known as lazy classifiers [2], can take into consideration each new training element because no structure is used in making the decision. On one hand, such methods are very adaptable, but on the other, the cost of their decision making is high. For such a method, we should also take into consideration that 'old' examples could have an unfavourable effect on quality of a classifier. Other kinds of

doi:10.1093/jigpal/jzr011

machine learning methods invest in building data structures that allow them to make inexpensive and rapid decisions. Unfortunately, these methods are not at all adaptable but they are able to make decisions quickly, the training thereof is costly in terms of time. Therefore, the design of data mining methods, especially the classification ones for data streams, is currently the focus of intense research [1, 4, 16, 21].

For the first time, the term *concept drift* appeared in the paper by Schlimmer and Granger [19] who described the problem of classification of objects with a changing definition of positive object though the time. Generally speaking, the concept drift could be caused by changes in the probabilities of classes or conditional probability distributions of classes [14] and we could distinguish the following sources of this phenomenon:

- Change of prior probabilities for classes.
- Change of class-conditional probability distribution.
- Change of posterior probabilities.

The most popular taxonomy of the *concept drift* was formulated in [17], where the author proposed the following kinds of it:

- Gradual drift (gradual changes, evolutionary changes, concept drift)—if the changes are smooth in nature.
- Sudden drift (substitution, abrupt changes, concept substitution, concept shift)—if the changes are abrupt.
- Recurring context (recurring trends)—if the context changes either periodically or changes in an unordered fashion.

The first methods dealing with the problem of the *concept drift* were STAGGER [19], FLORA [22] and IB3 [3] algorithms. They handle the concept drift problem by introducing two mechanisms: learning and forgetting.

The following alternative approaches can be considered:

- rebuilding a classification model if new data becomes available, which is very expensive and impossible from a practical point of view, especially if the concept drift occurs rapidly;
- detecting concept changes in new data [8] and if these changes are sufficiently 'significant', then rebuilding the classifier; and
- adopting an incremental learning algorithm for the classification model [7].

Traditional machine learning methods use supervised learning algorithms, which could produce a classifier on the basis of a set of labelled examples. In this approach, we should take into consideration the cost of data labelling, which is usually passed over. Let us notice that labels are usually assigned by human experts and therefore they cannot label all new examples if they come too fast. Therefore, methods of classifier design which could produce the recognition system on the basis of a partially labelled set of examples (the so-called active learning) would be an attractive proposition [12].

Changes are discovered by monitoring the unlabelled input data and discover novelties related to outlier detection, or by monitoring classification accuracy [15]. Constant update of a classifier is accomplished by using incremental learning methods that allow adding new training data during the exploitation of a classifier or by data set windowing.

This article focuses on the problem of the concept drift using the active learning approach for minimal distance classifiers. The potential for adaptation of the proposed method and its quality are evaluated through computer experiments, carried out on several benchmark data sets.

The article is organized as follows. The formal description of the problem is described in Section 2. Section 3 presents proposed algorithm of learning. In Section 4, the description and results of experiments are described. Section 5 concludes the article and proposes the main directions of the future work.

2 Problem statement

Let us formulate the classification problem, and introduce the concept drift modelling. The aim of the classification is to assign the object to one of the predefined classes $i \in M = \{1, \ldots, M\}$, on the basis of values of its features $x \in X$, where x is the features vector or the instance and X is called feature space. We define the classifier as any function that maps feature space to the set of class labels $\boldsymbol{M}$

$$\psi : X \rightarrow \boldsymbol{M} \tag{1}$$

We define also loss function $L(i, j)$ that describes the loss incurred for classifying object of class j as class i. The loss function is usually described by the loss matrix and in the simplest form mean loss is equal to classification error—the so-called zero-one loss.

$$L(i, j) = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases} \quad i, j \in \boldsymbol{M} \tag{2}$$

In the classification problem, our task is to find the best classifier that minimizes the expected value of loss function.

$$\psi^{*} = \underset{\psi(x)}{\operatorname{argmin}} \sum_{x \in x} [L(\psi(x), j) p(x)] \tag{3}$$

where j is the true class of $\boldsymbol{x}$ object. Overall performance of Ψ is equal to the expected value of loss function.

In changing environments, we have to deal with different problem descriptions. Our description of classes changes in time, so classifiers should also be dependent on time. The training set is described as the stream of historical data so the classification of $\boldsymbol{x}(t+1) \in X(t+1)$ is made by ψ_t classifier

$$\psi_t : X(t+1) \rightarrow \boldsymbol{M}. \tag{4}$$

And the expected value of loss function is equals:

$$\sum_{x \in \boldsymbol{x}} [L(\psi_t(\boldsymbol{x}), j) p_{t+1}(\boldsymbol{x})] \tag{5}$$

3 Proposed algorithm

One of the proposed solutions to handle the concept drift is the usage of sliding window (time window). In this approach, we use only the newest objects to train the classifier. One of the disadvantages of this solution is the necessity of the correct labelling of classified objects that are usually obtained from an external expert, or usage of a classifier output that leads to propagation of error (poisoning training set).

```
for each element(x,•)
set (x*,y*) = the nearest element from referenceSet
$d_d = \min(|x-x_i|)$ for $x_i$ in referenceSet with label $y_i \neq y^*$
if size(referenceSet) < windowSize or $|x-x^*| > d_e^*$ or $d_d < d_d^*$ then
    set y = getLabel((x, •))
    add (x,y) to referenceSet
if size(referenceSet) > windowSize then
    remove the oldest element from referenceSet
```

FIG. 1. Pseudocode of the proposed active learning algorithm.

Our algorithm uses the active learning paradigm, and lets the algorithm decide if the new object should be labelled by an external expert and used as training data or not. To make this decision, we take two factors into consideration: the movement of boundaries and exploration of new territories of feature space.

The usage of the minimal distance classifier inclined us to define those factors in terms of distance to the nearest objects from classes. We introduce the exploration factor as distance to the nearest object in current object set. To check if current boundaries are still valid, the algorithm will also ask for labelling of every object that is near the current boundary, so its distance to the nearest neighbours from different classes is more or less equal. Pseudocode of the proposed algorithm is presented in Figure 1.

4 Experiments

We seek the answer to question if the active learning methods can be implemented in the concept drift problem.

4.1 Scenario

The experiments are conducted in 1000 steps. In each step, we estimate the performance of the current classifier and next we present a new unlabelled instance to the algorithm, if the algorithm decides to use this instance to improve itself then the correct label is applied to the instance, new window is created and we train the new classifier.

To estimate the performance in the changing environment, we used the test and train method. The first classifier was tested against the whole rotated data set and the error on this set was assigned as performance measure for the current step. The next one of the rotated elements is randomly chosen to act as an instance to learn.

4.2 Data sets

In our experiments, we used three UCI [6] benchmark data sets shown in Table 1. Each data set was normalized before starting the experiments. Normalization was made by scaling each feature value so it has mean 0 and standard deviation 1. At each run of the experiment, we made a random permutation of the whole data set to make the initial training set (only the last *windowSize* of them have impact on initial results).

To simulate the gradual concept drift, we rotated our normalized data sets in every step. For every run of the experiment, we randomly paired the features and whole data set was rotated in the plane defined by each feature pair with an angle of $2\pi/1000$. This means that during the experiment the

TABLE 1. Data sets used in experiments

Data set	Number of features	Number of classes	Number of instances in classes
Iris	4	3	50, 50, 50
Wine	13	3	59, 71, 48
Breast	9	2	458, 241

TABLE 2. Parameters used in experiments

Data set	K	*windowSize*	d_d	d_e
Iris	1,3	40, 50, 75	0, 0.2, 0.5, 1.0	0.2, 0.5, 1.0
Wine	1,3	40, 50, 75	0, 0.4, 1.0, 2.0	0.4, 1.0, 2.0
Breast	1,3	40, 50, 75	0, 0.4, 1.0, 2.0	0.4, 1.0, 2.0

data set was fully rotated around the origin, and in the last iteration the data set is equal to the normalized original one. If the number of feature is odd, then the unpaired one is constant during the run experiment.

4.3 Used algorithm

As the reference algorithm, we use k-nearest neighbour (k-NN) algorithm with fixed sliding window. It is standard k-NN algorithm that uses only the last *windowSize* seen samples as the reference set. Each new object is labelled and added to the reference set while the oldest one is discarded from it.

Our active approach used an additional heuristic algorithm to determine if any newly seen object is useful for our reference set. To distinguish a given object, we used two values: the distance to the nearest point in the current reference set (d_e) and the difference in the distance to two nearest points from the reference set that belong to different classes (d_d). We set two thresholds parameters d_e^* and d_d^* if the new object fulfils condition

$$d_e > d_e^* \text{ or } d_d < d_d^* \tag{6}$$

then it is labelled and added to the reference set while the oldest one is discarded form it.

For both algorithms, initial classifiers were learned on *windowSize* randomly chosen samples from normalized data sets.

Because k-NN algorithm is prone to dimensionality curse, we have to use different parameters for different data sets. Parameter ranges used with every data set are shown in Table 2.

4.4 Performance measures

Each classifier was tested for two factors: mean error in exploitation time and number of instances that were used to build data sets—those instances need to be labelled what usually bounds with costs.

4.5 Results

The experiments were conducted with 5 parameters: k (number of NN), *windowSize*, $d_d, d_{e,}t$ (step number). Each set of parameters was used in 10 independent iterations.

To answer the question if our algorithms average error is significantly lower than error of reference algorithm, we used paired t-test at the significance level equal to 0.05 [5].

Table 3. Mean error for iris data set (*windowSize* = 50)

$d_d \backslash d_e$	0.2	0.5	1.0
0.0	0.0705	0.1059	0.1630
0.2	0.0638	0.0785	0.0893
0.5	0.0602	0.0642	0.0676
1.0	**0.0574**	**0.0583**	0.0625
Reference classifier: 0.0611			

Table 4. Mean number of labelled objects (*windowSize* = 50)

$d_d \backslash d_e$	0.2	0.5	1.0
0.0	477.6	234.6	121.1
0.2	516.8	302.6	216.1
0.5	604.7	451.3	383.5
1.0	722.9	612.6	560.3
Reference classifier: 1000			

Table 5. Mean error for iris data set (*windowSize* = 75)

$d_d \backslash d_e$	0.2	0.5	1.0
0.0	0.0799	0.1121	0.3899
0.2	0.0674	0.0801	0.0988
0.5	0.0616	0.0664	0.0708
1.0	0.0592	0.0618	0.0644
Reference classifier: 0.0569			

Table 6. Mean number of labelled objects (*windowSize* = 75)

$d_d \backslash d_e$	0.2	0.5	1.0
0.0	459.4	228.4	94.0
0.2	511.2	311.4	226.3
0.5	613.2	469.2	402.1
1.0	733.4	625.2	574.4
Reference classifier: 1000			

The influence of d_d and d_e parameters on mean error and number of labelled objects are presented in Tables 3–10. The results are means of all steps in all iterations of the experiment. Bolded results are statistically better than the reference classifier at the 0.05 significance level, and the underlined are not statistically significant on different than the reference level.

We can observe that the active selection of samples significantly reduces the number of objects that need to be labelled without decreasing the performance of a classifier. Some experiments showed that in particular cases it can even improve the overall performance.

We notice that by increasing d_e and decreasing d_d we can decrease mean error but the number of objects that need to be labelled increases. On the following Figures 2–4, we present the observed values of mean error and mean number of the labelled objects for different values of k and *windowSize* parameters.

TABLE 7. Mean error for breast data set (*windowSize* = 50)

$d_d \backslash d_e$	0.4	1.0	2.0
0.0	**0.0432**	**0.0425**	0.0458
0.4	**0.0432**	**0.0424**	**0.0442**
1.0	**0.0432**	**0.0422**	**0.0432**
2.0	**0.0434**	**0.0424**	**0.0424**
Reference classifier: 0.0471			

TABLE 8. Mean number of labelled objects (*windowSize* = 50)

$d_d \backslash d_e$	0.4	1.0	2.0
0.0	596.8	442.5	361.0
0.4	596.8	442.8	362.9
1.0	597.3	444.5	367.8
2.0	609.6	468.9	391.2
Reference classifier: 1000			

TABLE 9. Mean error for breast data set (*windowSize* = 75)

$d_d \backslash d_e$	0.4	1.0	2.0
0.0	**0.0415**	**0.0420**	0.0481
0.4	**0.0415**	**0.0419**	0.0439
1.0	**0.0416**	**0.0417**	0.0429
2.0	**0.0416**	**0.0413**	**0.0420**
Reference classifier: 0.0439			

TABLE 10. Mean number of labelled objects (*windowSize* = 75)

$d_d \backslash d_e$	0.4	1.0	2.0
0.0	592.9	438.8	355.2
0.4	592.9	439.3	358.0
1.0	594.0	442.9	365.9
2.0	612.1	476.0	407.3
Reference classifier: 1000			

4.6 Evaluation of results

On the basis of experimental results, we can notice that the performance at the level of the reference classifier may be obtained with much smaller amount of labelled examples. It has been shown that by active choosing of learning samples we can improve the classifier performance. If *windowSize* parameter is too high for the problem, then active learning performs worse than traditional methods, we think that it is caused by too slow forgetting factor, but that statement should be investigated in the future.

Parameters of algorithm, i.e. d_d and d_e should be tuned to specific data set, and maybe future work could discover a functional relation between the feature space size and the number of instances and the optimal values of those parameters.

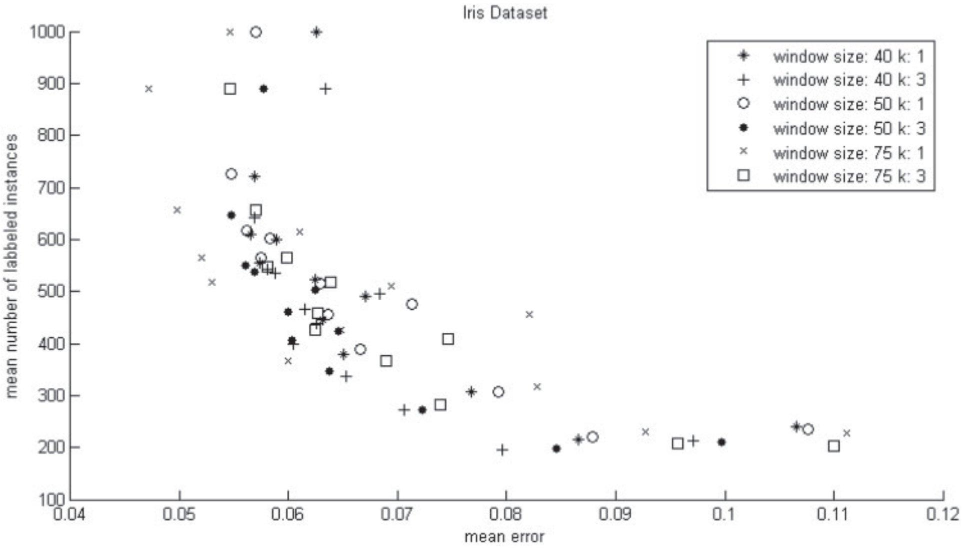

FIG. 2. Results for Iris data set.

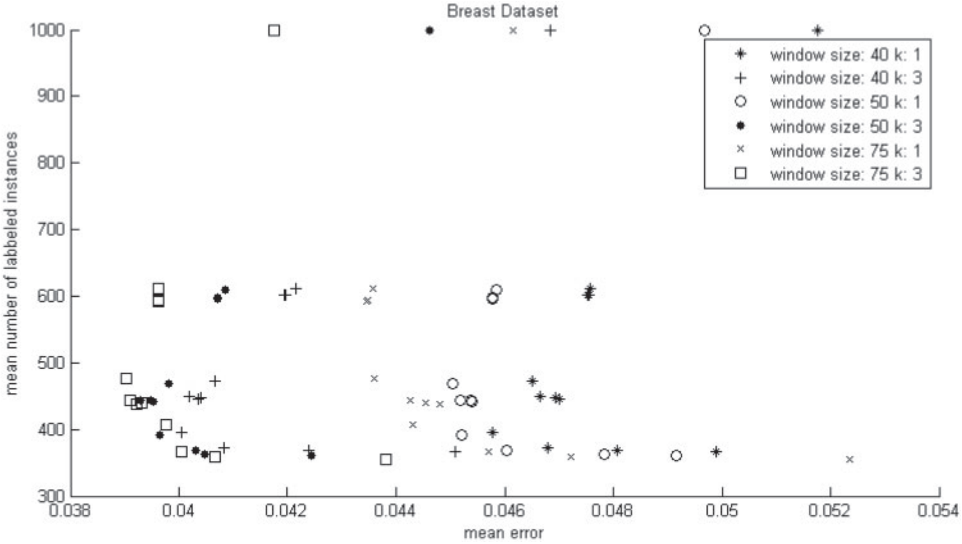

FIG. 3. Results for Breast data set.

5 Final remarks

The article presented a new active learning algorithm dedicated to solve the concept drift problem and compared its performance with the classical approach. The obtained results justified the usage of active learning methods in the concept drift problems. The proposed algorithms with the correct parameterization performed better than the classical approach, an additional gain was the labelling cost of samples that was not used to build our classifier.

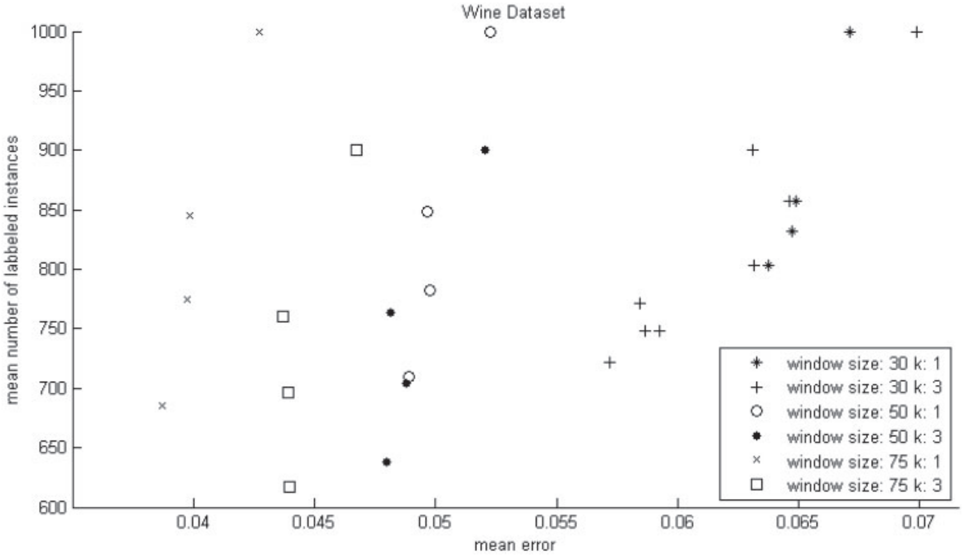

FIG. 4. Results for Wine data set.

Our future research will be focused on:

- developing methods of forgetting samples adapted to the presented algorithm;
- searching for methods to adjust parameters' values especially based on the hybrid approach [11]; and
- applying the presented algorithm to the real problems of the concept drift.

Funding

This work is supported by The Polish Ministry of Science and Higher Education under the grant which is being realized in years 2010–2013.

References

[1] Ch. C. Aggarwal. On classification and segmentation of massive audio data streams. *Knowledge and Information Systems*, **20**, 137–156, 2009.

[2] D. W. Aha (ed.). *Lazy Learning*. Kluwer Academic Publishers, 1997.

[3] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, **6**, 37–66, 1991.

[4] M. Aksela and J. Laaksonen. Adaptive combination of adaptive classifiers for handwritten character recognition. *Pattern Recognition Letters*, **28**, 136–143, 2007.

[5] E. Alpaydin. *Introduction to Machine Learning*, 2nd edn., The MIT Press, 2010.

[6] A. Asuncion and D. J. Newman. UCI ML Repository. University of California, School of Information and Computer Science, 2007. Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html

[7] Y. Ben-Haim and E. Yom-Tov. A streaming parallel decision tree algorithm. *Journal of Machine Learning Research*, **11**, 849–872, 2011.

[8] A. Bifet and R. Gavalda. Learning from time-changing data with adaptive windowing. *Proceedings of SIAM International Conference on Data Mining (SDM'07)*, 2007.

[9] Ch. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[10] M. Black and R. Hickey. Classification of customer call data in the presence of concept drift and noise. *Proceedings of the 1st International Conference on Computing in an Imperfect World Soft-Ware 2002*, Springer, pp. 74–87, 2002.

[11] E. Corchado, A. Abraham, and A. de Carvalho. Hybrid intelligent algorithms and applications. *Information Science*, **180**, 2633–2634, 2010.

[12] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy. Mining data streams: a review. *ACM SIGMOD Record*, **34**, 18–26, 2005.

[13] R. Greiner, A. J. Grove, and D. Roth. Learning cost-sensitive active classifiers. *Artificial Intelligence*, **139**, 137–174, 2002.

[14] M. Kelly, D. Hand, and N. Adams. The impact of changing populations on classifier performance. *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 367–371, 1999.

[15] L. Kuncheva. Classifier ensembles for changing environments. *Lecture Notes in Computer Science*, **3077**, 1–15, 2004.

[16] H. Liu, Y. Lin, and J. Han. Methods for mining frequent items in data streams: an overview. *Knowledge and Information Systems*, **26**, 1–30, 2009.

[17] A. Narasimhamurthy, L. Kuncheva. A framework for generating data to simulate changing environments. *AIAP'07: Proceedings of the 25th IASTED International Multi-Conference,* ACTA Press, pp. 384–389, 2007.

[18] A. Patcha and J. Park. An overview of anomaly detection techniques: existing solutions and latest technological trends. *Computer Networks*, **51**, 3448–3470, 2007.

[19] J. Schlimmer and R. Granger. Incremental learning from noisy data. *Machine Learning*, **1**, 317–354, 1986.

[20] A. Tsymbal. The problem of concept drift: definitions and related work. *Technical Report.* Department of Computer Science, Trinity College, 2004.

[21] A. Ulaş, M. Semerci, O. T. Yıldız, and E. Alpaydın. Incremental construction of classifier and discriminant ensembles. *Information Sciences*, **179**, 1298–1318, 2009.

[22] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, **23**, 69–101, 1996.

[23] X. Zhu, X. Wu, and Y. Yang. Effective classification of noisy data streams with attribute-oriented dynamic classifier selection. *Knowledge and Information Systems*, **9**, 339–363, 2006.

Received 16 December 2010