Investigation of Expert Addition Criteria for Dynamically Changing Online Ensemble Classifiers with Multiple Adaptive Mechanisms

Rashid Bakirov, Bogdan Gabrys

Investigation of expert addition criteria for dynamically changing online ensemble classifiers with multiple adaptive mechanisms

Rashid Bakirov and Bogdan Gabrys

Smart Technology Research Centre, Bournemouth University, United Kingdom
{rbakirov,bgabrys}@bournemouth.ac.uk

Abstract. We consider online classification problem, where concepts may change over time. A prominent model for creation of dynamically changing online ensemble is used in Dynamic Weighted Majority (DWM) method. We analyse this model, and address its high sensitivity to misclassifications resulting in creation of unnecessary large ensembles, particularly while running on noisy data. We propose and evaluate various criteria for adding new experts to an ensemble. We test our algorithms on a comprehensive selection of synthetic data and establish that they lead to the significant reduction in the number of created experts and show slightly better accuracy rates than original models and non-ensemble adaptive models used for benchmarking.

1 Introduction

Ensemble learning in stationary settings has been extensively researched, and it was shown to often be able to outperform single learners [17], [3], [15]. There has been a number of attempts to apply this ensemble learning paradigm to online learning for non-stationary environments. Many of them try to map experts to the continuous data batches or concepts [10] [9] [5] [13].

We focus on the problem of online classification in incremental fashion, where learner is presented with a single data instance, and after its classification, the true label for this instance is revealed. Our aim in this setting is the creation and maintenance of an experts' ensemble which can adapt to the changes in data.

In "batch learning", the data is presented in chunks providing natural data bases for creation of experts whereas incremental learning makes questions such as when, and on what data basis add an expert, very important. These two learning types share other problems of optimal adjustment of weights and use of suitable criterion for expert removal.

In this work we concentrate on the model of ensemble management introduced in [10]. We analyse the algorithm and focus on its shortcomings, such as undesirable behaviour in noisy environments with regard to addition of new unnecessary experts to ensemble. Larger ensembles require more computational resources and are therefore less desirable. The main purpose of this paper is empirical analysis of how the changes in the following main areas of the model affect the ensemble size and accuracy:

- *Expert creation criterion* - original model creates a new expert after every misclassification. We explore alternative criteria based on the average accuracy on the defined window of last data instances.
- *Expert quality* - we investigate the possibility of using data windows for the purposes of expert creation.
- *Expert assessment* - even if the expert has a satisfactory accuracy, it might not always be needed to include it in the ensemble. We look into on-the-fly evaluation of the expert, to decide whether its use is beneficial.

Specifically, we use windows of data instances to determine when to add a new expert, or which data basis to use for its training. Our results show that creating expert from larger data bases leads to the highest accuracy rates, while evaluating their performance helps keeping their number lower than the original model and still reach comparable accuracy rates.

2 Related work

The notion of using more than one expert (ensemble) for making decisions has a long history. Famously, in 1785 Condorcet has established that if the probability p of making a correct decision for each voter is larger than 0.5, then, under certain assumptions, the larger the number of the voters, the higher is the likelihood of reaching the correct decision when choosing among two alternatives [17].

Work of Littlestone and Warmuth [12] is one of the seminal papers on the topic of using expert ensembles for online learning. They consider binary prediction task using multiple experts, with given initial weights, each of whom makes an individual prediction. In the case of wrong prediction, the weights of predictors are multiplied by β such as $0 < \beta < 1$. This work can be considered a special case of [19], which considers continuous prediction and decreasing the weights of experts according to the loss function inversely proportional to their error.

In the last 10 years, there has been an increased interest of data mining community to use ensemble methods while dealing with on-line learning in non-stationary environments. Particularly, the problem of "concept drift" has been often addressed. In this setting, another intuitive reasoning for using ensembles is the intention that each expert should represent a certain concept or a part of it. Among the many classifier algorithms for concept drift scenario we review the most relevant ones for our purposes. A well-known algorithm creating expert ensemble in online mode is DWM [10] , which adapts to drift by creating a new expert each time a datapoint is misclassified by the existing ensemble. New expert gets the weight of one. All experts train online and whenever an expert misclassifies a data instance, its weight is multiplied by $0 < \beta < 1$. After each classification, to reduce the dominance of newly added experts, the weights of existing experts are normalized, so that the highest weighting expert gets a new weight of 1. To reduce the number of experts, they are deleted if their weight is lower than a defined threshold θ. In [11] the same authors present AddExp.D, a variation of this method where the weight assigned to the new experts is

the current weight of the ensemble multiplied by a constant γ. Here the authors bound the error of this type of ensemble on the error of the latest created expert, provided that $\beta + 2\gamma < 1$.

CDC algorithm [18] is a similar approach to adaptive ensemble management. CDC has fixed number of experts; it starts with an empty set and adds a new expert every data instance until the set is full. New "immature" experts have the weight of zero and become "mature" after learning on a defined number of instances (usually equal to the size of the ensemble). The experts are weighted based on their performance on a test set, which must be available with every data instance, and removed if the all of the following conditions are satisfied: a) their weight is below the threshold; b) their weight is the smallest among all members of the ensemble; c) they are mature.

An interesting algorithm which aims to minimize the number of experts is discussed in [1]. Here, only two experts are active at a given time; an active predictor, which is trained on the complete set of the instances, and the test predictor which is trained only on the last n instances. If the test predictor starts predicting better than the active one, the active one is deleted, test predictor becomes active, and a new test predictor is started to be trained.

More recently, similar algorithms have been proposed for time series prediction. [7] proposes creating a new expert every instance while deleting some of the older ones and [8] suggests splitting the data stream into epochs and creating an expert which learns on the most recent epoch every τ instances.

3 Elements of online expert ensemble creation

One of the most researched and well defined reasons for the adaptive models is the problem of concept drift, as introduced in [16]. Concept drift occurs when the statistical distribution of target or input variables (virtual drift), or their relations change over time (real drift) [5]. We are more interested in the real concept drift, more formal definition of which drift can be given as follows. Assume an input vector $\bar{x} = (x_1, x_2, ..., x_n)$ and a function $y = f_t(\bar{x})$ which produces an output y from $\bar{x}$ at time t. If there exists an input vector $\bar{x'}$ and time points t_1, t_2, such that $f_{t_1}(\bar{x'}) \neq f_{t_2}(\bar{x'})$, then this is called concept drift.

In this work we concentrate on the model of ensemble management introduced in [10] and [11] (except the pruning part of the latter) which has been empirically shown to be effective and perform well in many cases. Certain bounds on overall number of mistakes are given in [11] as well. However this approach is not entirely problem free, as it becomes clear from following sections.

Reviewed model involves several layers of adaptation - online training, change of exerts' weights, addition and removal of experts. Clearly, the most drastic adaptation method used is adding new experts, which is why we concentrate on this topic. In the following we will analyse the performance of the discussed model and some of its modifications. For this purpose we use the term *reaction time* - the minimum number of observations, after which algorithm will react to observed change by creating an expert and *convergence time* - number of

observations, after which algorithm will converge (total weight of the experts which are trained on the new concept is larger than the total weight of the experts) to new concept.

3.1 Condition for adding of an expert

Condition for adding an expert largely determines the *reaction time* of an algorithm, and thus plays a significant role in its *convergence time* as well. Reviewed model reacts to misclassifications by creating a new expert. Initially, it is suggested to add an expert every time when the prediction of current ensemble is false. This provides fast reaction time but may, in noisy conditions, result in adding many unnecessary and inaccurate experts. To deal with this problem, in [10] authors suggest that, in noisy domains or for large experiments, only every T-th example could be taken into consideration, which reduces the number of created experts in proportion to T. The drawback of having $T > 1$ is a possibility of slower reaction to the change. This is best manifested during a sudden drift, where, in the worst case, the reaction time is T.

To reduce the effects of noise in a more deterministic way, we propose the averaging window condition for expert creation as an alternative to having $T > 1$ (In [1] a similar condition is used to substitute learners). Here, the strategy is based on the decision of creating an expert from x_n (n-th datapoint) not only as a result of x_n's classification, but on the basis of accuracy in the window of the last l elements, with x_n being the last element of l. We add an expert trained from x_n if the average accuracy of the ensemble in the window is less than fixed threshold value u. If we assume that the change causes algorithm to always misclassify incoming data, then the reaction time to the change in this case can be calculated to be at most $l(1-u)$ rounded up.

The choice of the threshold may be difficult for unknown data. Also, for the datasets where average accuracy may vary with the time, for example due to changing noise levels, using the above static threshold might result in creation of many unnecessary experts or not creation of experts when needed. We introduce a similar algorithm with dynamic threshold value, which we call"maximum accuracy threshold window" (MTW). The dynamic threshold here is similar to the one used in DDM change detector [6]. While classifying incoming data we record the maximum value of $\mu_{acc} + \sigma_{acc}$ where μ_{acc} is mean accuracy and $\sigma_{acc} = \sqrt{\frac{\mu_{acc}(1-\mu_{acc})}{l}}$ is the standard deviation of the Bernoulli process. We create a new expert when the condition $\mu_{cur} - \sigma_{cur} < \mu_{max} - m * \sigma_{max}$ is met. Here μ_{cur} and σ_{cur} are mean accuracy and standard deviation of current window and μ_{max} and σ_{max} are mean accuracy and standard deviation of the window where the maximum value of $\mu_{acc} + \sigma_{acc}$ was recorded. Parameter m is usually set to 3. After creation of new expert, the maximum values are reset. A possible issue in some cases could be that when the accuracy reaches 1, the new expert will be added when there is a single misclassification. To prevent this it is possible to enforce a certain minimum σ_{max} such as 0.1 or 0.15. Using window based conditioning is illustrated in the Figure 1b.

3.2 Data basis for new experts

Assuming uniform class label distribution, training an expert from a single data instance means that this expert will assign the label it has been trained on to all other samples which makes its initial accuracy 1/n in the case of n-label classification problem. Low accuracy of experts trained on insufficient amount of data, also discussed in [20], combined with the high weight of the new expert, may result in noticeable negative effect on the accuracy. To counter this we can use a delay in reaction time to train the new expert on more examples before using it for predictions (note: approaches proposed in this section are analysed standalone and not combined with the ones from the section 3.1 at this moment). The simplest option is to train an expert on l datapoints after its creation and only then add it to the ensemble, as in [18]. We call this "mature" experts (MATEX) approach. To prevent multiple reactions to one change, during the time that expert is being "matured", no new experts are introduced. When the expert is added to the ensemble, it is better trained and thus more accurate than the expert which is created from only one datapoint. The reaction time for a change in this case is l. Another advantage of this approach is reducing the effect of noise on the created expert.

One possibility to reduce the number of unnecessary experts created in this way is assessing their performance. A sufficient condition for expert to benefit the ensemble independent of this expert's weight is predicting better than ensemble (another option could be dynamic weighting based on the accuracy assessment, which is not discussed here). So before adding it to the ensemble we can compare it with the performance of the ensemble in the window of size l. Comparison strategy is similarly used in [1]. The comparison can be done in various ways; comparing the prequential accuracies of the expert and ensemble, or constructing certain test and training sets from the datapoints in the window and using cross-validation. If the validation is successful, then the new expert which has been trained on the whole window is added to the ensemble. Here the reaction time is l. To prevent multiple reactions to one change, during the time that expert is being "validated", no new experts are introduced. Here, the effect of noise is further reduced - when the data suddenly becomes noisy, newly created experts will probably not predict better than existing ensemble and thus will be discarded. Validation approach can be combined with MATEX allowing the expert to train on l_{mature} datapoints, before starting the comparison on l_{val} datapoints. This might help prevent the premature removal of experts but will accordingly increase reaction time to $l_{mature} + l_{val}$. It must be noted that this approach requires additional computational effort for the validation. Using window for the data basis of new expert is illustrated in the Figure 1c.

4 Experimental results

4.1 Methods description

We have experimented with different variations of the methods described in the Section 3. The implemented window based condition schemes from the section

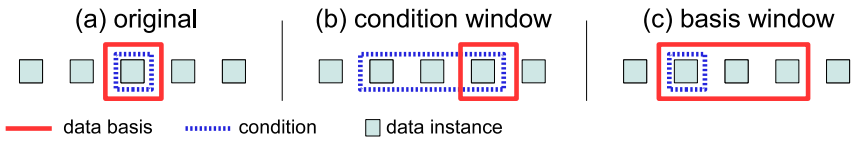

Fig. 1: Using windows for expert adding condition and data base of a new expert

Table 1: Experiments with window based conditions to add an expert

Type	Threshold u(static) / n(dynamic)	Window length	$min(\sigma_{max})$	Codename
Static	0.5	5	N/A	WIN_5_0.5
Static	0.5	10	N/A	WIN_10_0.5
Static	0.7	10	N/A	WIN_10_0.7
Dynamic	3	5	0	MTW_5
Dynamic	3	10	0	MTW_10
Dynamic	3	10	0.1	MTW_10_0.1
Dynamic	3	10	0.15	MTW_10_0.15

3.1 are presented in Table 1. Implemented methods experimenting with experts' data basis (section 3.2) were MATEX with window sizes of 5 and 10, prequential validation (PVAL) with window sizes of 5 and 10, combination of MATEX and prequential validation each having window of 5 and several variations of cross-validation methods (XVAL) using window size of 10 with different sizes of training and testing sets. We also have experimented with the periodical expert additions with periods T of 5, 7, 10 and 11. In our implementation of original algorithms, WIN and MTW we create a new expert from the single datapoint.

We have used different weighting schemes for all of the experiments, specifically static weighting [10] with new expert weight of 1 with β equal to 0.3, 0.5, 0.7 and dynamic weighting [11] with the same values of β and respective values of γ equal to 0.3, 0.2, 0.1. Unlike dynamic weighting, static weighting makes convergence time n_{conv} dependant on the total weight of ensemble at the moment of expert W creation. This allows implicit control of n_{conv} while limiting its explicit control to some extent. The following results are based on $\beta = 0.5$.

4.2 Results on synthetic data

We have synthesised 26 two-dimensional data sets with various properties to examine the behaviour of the algorithms in different situations. We consider rotating hyperplane data and Gaussians with different type of changes - switching between two data sources [14], one Gaussian passing through the other one and returning, Gaussians moving together in one direction and returning (see Figure 2). We have experimented with various magnitudes of changes and levels of artificial noise and decision boundaries overlap (see Table 2).

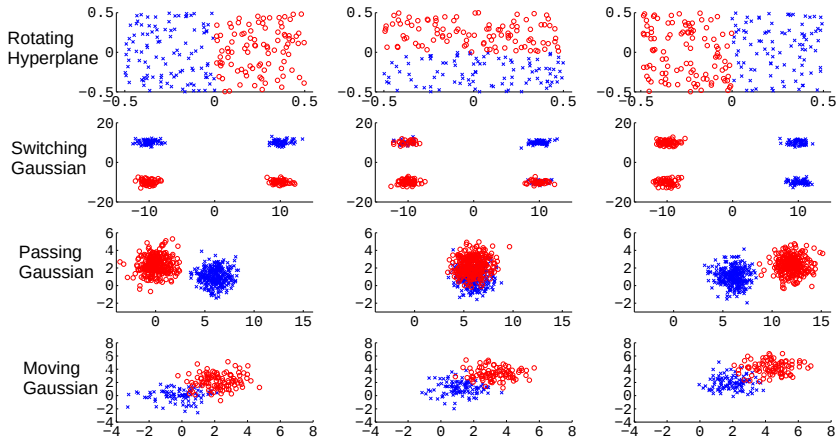

Fig. 2: Changes in experimental datasets. From left to right: data in the start, in the middle, in the end (before possible return to starting position).

We have tested the described methods with Naive Bayes base learner using PRTools 4.2.4 toolbox for MATLAB [4]. MATEX_10 with dynamic weighting showed the best average test and prequential accuracy among all of the methods on 26 datasets, and the best average test accuracy on 6 datasets. Here, the test accuracy is calculated using average predictive accuracy on additional 100 test instances from the same distribution as current training data instance. From the methods with window based expert creation condition (table 1), MTW_10 with dynamic weighting shows the best performance, 0.03% less than the leader. The original DWM showed comparable average test accuracy results (88.4%, 1.3% less than leader) but had 3 times as larger average ensemble size (~15 against ~5). Even in the datasets without noise, average ensemble size of the original DWM is noticeably higher than that of other methods. The same model with dynamic weighting has twice smaller average ensemble size.

In the Table 3 we compare results of different described variations of the original method with the results of original method. Here we use window of 10 for all of the methods. XVAL is leave-one-out cross-validation. Threshold in the WIN is 0.5. Again, MATEX methods with dynamic weighting emerge as a narrow leaders in terms of average test accuracy (see Table 3 for full results). The accuracy rates are quite similar, but we see a noticeable decrease in the number of total created experts and the average ensemble size. Validation methods PVAL and XVAL further reduce the number of total created experts and average ensemble size, while having slightly lower accuracy rates than the leaders and requiring additional computation for validation purposes. To benchmark our results against non-ensemble methods we have run tests with a simple online Naive Bayes classifier without any forgetting, state of the art change detectors DDM [6]

Table 2: Synthetic datasets used in experiments. Column "Drift" specifies number of drifts, the percentage of change in the decision boundary and its type.

Num.	Data type	Instances	Classes	Drift	Noise/overlap
1	Hyperplane	600	2	2x50% rotation	None
2	Hyperplane	600	2	2x50% rotation	10% uniform noise
3	Hyperplane	600	2	9x11.11% rotation	None
4	Hyperplane	600	2	9x11.11% rotation	10% uniform noise
5	Hyperplane	640	2	15x6.67% rotation	None
6	Hyperplane	640	2	15x6.67% rotation	10% uniform noise
7	Hyperplane	1500	4	2x50% rotation	None
8	Hyperplane	1500	4	2x50% rotation	10% uniform noise
9	Gaussian	1155	2	4x50% switching	0-50% overlap
10	Gaussian	1155	2	10x20% switching	0-50% overlap
11	Gaussian	1155	2	20x10% switching	0-50% overlap
12	Gaussian	2805	2	4x49.87% passing	0.21-49.97% overlap
13	Gaussian	2805	2	6x27.34% passing	0.21-49.97% overlap
14	Gaussian	2805	2	32x9.87% passing	0.21-49.97% overlap
15	Gaussian	945	2	4x52.05% move	0.04% overlap
16	Gaussian	945	2	4x52.05% move	10.39% overlap
17	Gaussian	945	2	8x27.63% move	0.04% overlap
18	Gaussian	945	2	8x27.63% move	10.39% overlap
19	Gaussian	945	2	20x11.25% move	0.04% overlap
20	Gaussian	945	2	20x11.25% move	10.39% overlap
21	Gaussian	1890	4	4x52.05% move	0.013% overlap
22	Gaussian	1890	4	4x52.05% move	10.24% overlap
23	Gaussian	1890	4	8x27.63% move	0.013% overlap
24	Gaussian	1890	4	8x27.63% move	10.24% overlap
25	Gaussian	1890	4	20x11.25% move	0.013% overlap
26	Gaussian	1890	4	20x11.25% move	110.24% overlap

and EDDM [2] and Paired Learners method with window size 10 and threshold 0.1 [1]. As expected, online Naive Bayes performs noticeably worse than adaptive methods. Change detectors and paired learners show slightly lower but comparable test accuracy to MATEX methods.

The top performers on some datasets can be different than the average leaders. For instance, validation methods perform better on the dataset with passing Gaussian with 4 drifts. Here, XVAL with dynamic static weighting shows the best accuracy among the methods compared above - 87.2% which is 0.8 % higher than the accuracy of the leader. Intuitively, this can be explained with a large proportion of class intersection area, where the expert creation is not beneficial, and two intersection-free areas where high accuracy experts can be created. In general, expert checking is beneficial for the datasets with variable noise or decision boundary intersection. Figure 3 gives an insight on the performance for selected methods with static weighting from the Table 3 on individual datasets.

Table 3: Results on 26 synthetic datasets, averaged

Method	Average test accuracy	Std. deviation of avg. accuracy	Average total created experts	Average ensemble size
MATEX dynamic weighting	0.898	0.064	54.58	4.96
MATEX static weighting	0.893	0.067	54.62	6.57
MTW dynamic weighting	0.894	0.067	40.38	3.31
MTW static weighting	0.889	0.071	40.54	4.63
DWM periodical dynamic weighting	0.894	0.064	15.58	5.42
DWM periodical static weighting	0.891	0.066	15.81	6.76
XVAL dynamic weighting	0.890	0.068	22.35	2.31
XVAL static weighting	0.893	0.068	21.12	3.31
PVAL dynamic weighting	0.888	0.066	5.23	1.53
PVAL static weighting	0.889	0.066	4.65	1.48
WIN dynamic weighting	0.880	0.073	51.04	3.39
WIN static weighting	0.881	0.072	24.85	4.71
Original dynamic weighting	0.867	0.091	181.15	7.94
Original static weighting	0.884	0.075	156.12	14.97
PAIRED LEARNER	0.891	0.069	4.5	2
DDM	0.88	0.077	2.27	1
EDDM	0.89	0.067	1.92	1
NAIVE_BAYES	0.807	0.137	1	1

5 Conclusions

In this work we discuss shortcomings of the investigated dynamic ensemble classification method introduced in [11], and perform analysis of how using data windows for expert creation condition and data basis affect the number of the created experts and predictive accuracy. Our extensive tests with synthetic data show the viability of suggested approaches for different concept drift scenarios. It is not the aim of the paper to present a better novel classification algorithm, however the most of our proposed variations result in slightly better performance and significantly smaller number of experts than the original model. Expert evaluation techniques reduce the average ensemble size to the minimum while retaining comparable performance. We notice that ensemble methods can perform better than non-ensemble methods based on drift detection and there are promising signs that this performance can be improved even further. We conclude that:

- *Window based expert creation criteria* lead to comparable or slightly higher accuracy rates and a reduction of average ensemble size.
- *Window based expert data bases* result in slightly higher accuracy rates and a significant reduction of average ensemble size.
- *Expert validation* leads to comparable accuracy rates and a drastic reduction of ensemble size, but requires more computational effort.

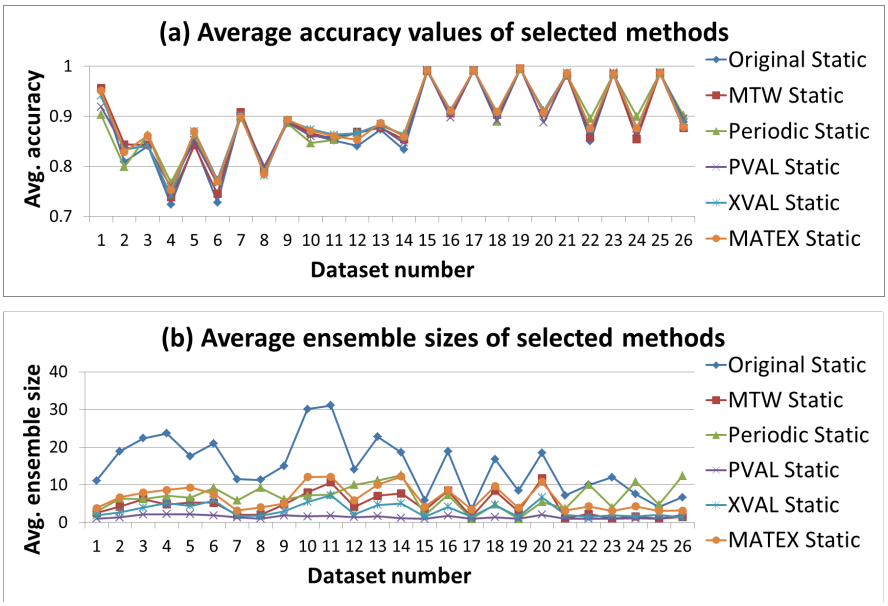

Fig. 3: Average accuracy values and ensemble sizes for selected methods

In the future we aim to present the detailed results, discussing why certain methods perform better on certain datasets. Further improvements of the proposed methods, e.g. dynamic starting weights for experts based on their performance during validation phase, dynamic parameter β and explicit handling of recurring concepts can be looked into. Probabilistic analysis of models is currently under way and the analysis of complexity is planned. Another direction of research is intelligent combination of conditioning, data base selection and validation. We intend evaluating the methods on a selection of real datasets.

Acknowledgements. Research leading to these results has received funding from the EC within the Marie Curie Industry and Academia Partnerships and Pathways (IAPP) programme under grant agreement no. 251617. We express our thanks to Indrė Žliobaitė, Damien Fay and anonymous reviewers.

References

1. Bach, S.H., Maloof, M.a.: Paired Learners for Concept Drift. 2008 Eighth IEEE International Conference on Data Mining pp. 23–32 (Dec 2008)
2. Baena-García, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavaldà, R., Morales-Bueno, R.: Early drift detection method. In: ECML PKDD 2006 Fourth International Workshop on Knowledge Discovery from Data Streams. Berlin, Germany (2006)

3. Dietterich, T.G.: Ensemble Methods in Machine Learning. MULTIPLE CLASSIFIER SYSTEMS Lecture Notes in Computer Science, 2000, Volume 1857/2000, 1-15, DOI: 10.1007/3-540-45014-9_1 (2000)
4. Duin, R.P.W., Juszczak, P., Paclik, P., Pekalska, E., de Ridder, D., Tax, D.M.J., Verzakov, S.: PRTools4.1, A Matlab Toolbox for Pattern Recognition (2007), http://prtools.org
5. Elwell, R., Polikar, R.: Incremental learning of concept drift in nonstationary environments. IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council 22(10), 1517–31 (Oct 2011)
6. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A., Labidi, S. (eds.) Advances in Artificial Intelligence SBIA, pp. 286–295. Springer, Berlin Heidelberg (2004)
7. Hazan, E., Seshadhri, C.: Efficient learning algorithms for changing environments. In: ICML '09 Proceedings of the 26th Annual International Conference on Machine Learning. pp. 393–400 (2009)
8. Jacobs, A., Shalizi, C.R., Clauset, A.: Adapting to Non-stationarity with Growing Expert Ensembles. Tech. rep., Carnegie Mellon University (2010), http://www.santafe.edu/media/cms_page_media/285/AJacobsREU paper.pdf
9. Kadlec, P., Gabrys, B.: Local learning-based adaptive soft sensor for catalyst activation prediction. AIChE Journal 57(5), 1288–1301 (May 2011)
10. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: An ensemble method for drifting concepts. The Journal of Machine Learning Research Volume 8+, 2755–2790 (2007)
11. Kolter, J., Maloof, M.: Using additive expert ensembles to cope with concept drift. In: ICML '05 Proceedings of the 22nd international conference on Machine Learning. pp. 449 – 456. No. 1990 (2005)
12. Littlestone, N., Warmuth, M.: The Weighted Majority Algorithm. Information and Computation 108(2), 212–261 (Feb 1994)
13. Minku, L.L., Yao, X.: DDD: A New Ensemble Approach for Dealing with Concept Drift. IEEE Transactions on Knowledge and Data Engineering 24(4), 619–633 (Apr 2012)
14. Narasimhamurthy, A., Kuncheva, L.: A framework for generating data to simulate changing environments. In: Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications. pp. 384–389. ACTA Press, Anaheim, CA, USA (2007)
15. Ruta, D., Gabrys, B.: Classifier selection for majority voting. Information Fusion 6(1), 63–81 (Mar 2005)
16. Schlimmer, J.C., Granger, R.H.: Incremental Learning from Noisy Data. Machine Learning 1(3), 317–354 (1986)
17. Shapley, L., Grofman, B.: Optimizing group judgmental accuracy in the presence of interdependencies. Public Choice 43(3), 329–343 (1984)
18. Stanley, K.O.: Learning concept drift with a committee of decision trees. Technical report, UT-AI-TR-03-302, Department of Computer Science, University of Texas in Austin (2003)
19. Vovk, V.G.: Aggregating strategies. In: COLT '90 Proceedings of the third annual workshop on Computational learning theory. pp. 371–386. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (Jul 1990)
20. Žliobaitė, I., Kuncheva, L.I.: Theoretical Window Size for Classification in the Presence of Sudden Concept Drift. Tech. rep., CS-TR-001-2010, Bangor University, UK (2010)