KNN classifier with self adjusting memory for heterogenous concept drift

Viktor Losing*†, Barbara Hammer* and Heiko Wersing†
*Bielefeld University, Universitätsstr. 25, 33615 Bielefeld
†HONDA Research Institute Europe, Carl-Legien-Str. 30, 63073 Offenbach am Main

***Abstract*—Learning from non-stationary data streams is gaining more attention recently, especially in the context of Internet of Things and Big Data. It is a highly challenging task, since the fundamentally different types of possibly occurring drift undermine classical assumptions such as i.i.d data. Incremental drift characterizes a continuous change in the distribution such as the signals of a slowly degrading sensor. A suddenly malfunctioning sensor on the other hand causes a severe shift and is defined as abrupt drift. Available algorithms are able to handle different types of drift, however they target either abrupt or incremental drift and often incorporate hyperparameter requiring a priori knowledge about the task at hand.**

We propose a biological inspired, architecture which partitions the data into a short-term and long-term memory. The former is a window of recently seen data-points whose size is adjusted such that the estimated generalization error is minimized. The latter preserves only those information from previous concepts which are non-conflicting to the current one. These memories are combined according to the demands of the present concept to classify unseen data points. We couple our parameter free approach with the K-Nearest Neighbor classifier, however, any other incremental learning algorithm could be used as well.

New artificial and real datasets are proposed to evaluate performance on specific types of drift. Experiments on these as well as on generally known benchmark datasets compare our approach with state of the art methods. The highly competitive results throughout all experiments underline the robustness of our approach.

I. Introduction

II. Concept Drift

Definition and examples

A. Types of Drift

-abrupt, incremental, gradual, reoccuring, virtual

III. Related Work

-Survey by gamma eventuell Ditzler -Active drift detection work Gamma, Bifet ADWIN for abrupt to minimize delay -Passive drift handling with ensembles DACC, LPPNSE - Windowing but usually throws away information which still can be valid, no LTM -SVM Leave one out -KNNPAW -KNN for streaming like the ICDM paper

IV. Architecture

Our architecture is depicted in figure **??**. It consists of 4 different steps. The goal is to keep as much as possible of the current concept in the Short Term Memory(STM) and at the same time preserve former knowledge in the Long Term Memory (LTM). We assume that the recent information is correct and reflects the current concept. Therefore, it is crucial to make sure that the memory in the LTM is not conflicting.

A. Short Term Memory

-STM, partitioning, O(log n) tests, maximizing generalization accuracy, write down in fancy formal way -LTM, keep not compromising -Transfer -choosing of proper prediction model(STM, or LTM, or Both) -Clustering -Size management

B. Efficiency

-distances have to be calculated anyways once, can be stored and afterwards need only to be sorted -fits with other algorithms as long they can be trained incremental and decremental e.g. NB, incremental SVM, but how to solve validation procedure -approximative KNN -adapt stm not every example but every n examples

V. Experiments

We compare our method with well-known state of the art methods for handling drifting streaming data.

Learn++.NSE with Classification and Regression Trees
Proposed in [1], this algorithm processes incoming samples in chunks with a predefined size. A base classifier is trained for each chunk and added to the ensemble. The loss on recent chunks is averaged with a sigmoidal function to compute the final weight of each base classifier. Similar to AdaBoost, instances are weighted such that misclassified inputs have a higher impact on the calculated loss. Chunk-wise trained models have by design an adaption delay depending on the chunk size. The base classifier are in our case Classification and Regression Trees (CART) XXVL reference.

Leveraging bagging with Hoeffding Trees
XXVL either describe Hoeffding Trees or make a reference! Bifet et al. propose in [2]to increase the randomization of online bagging [3] and thereby the diversity of the base classifier. This is done by a higher λ value for the poisson distribution and the usage of output detection codes. Additionaly, they use ADWIN as change detector for every classifier within the ensemble such that whenever a change is detected the worst classifier is replaced by a new one. As base classifier Hoeffding Trees with Gaussian Naive Bayes within the leaves are used.

Dynamic Adaption of Concept Changes with Hoeffding Trees
Within this ensemble algorithm [4] a classifier of the worst half of the pool is removed randomly after a predefined number of examples and replaced by a new one. New examples are only classified by the best classifier of the pool.

TABLE I. THE COMPARED ALGORITHMS.

Abbr.	Classifier	Parameter
HT_A	Hoeffding Tree with ADWIN	
LPPNSE	Learn++.NSE with CART	chunk-Size = various
DACC	Dynamic with HT	n=10
LVGB	Leveraging Bagging with HT	n=10
KNN_S	KNN with sliding window	w=5000, k=5
$\text{KNN}_{\text{W}_\text{A}}$	NN with PAW+ADWIN	w=5000, k=5
KNN_M	KNN with STM+LTM memory	w=5000, k=5

TABLE II. EVALUATED ARTIFICIAL DATASETS.

Dataset	#Samples	#Feat.	#Class	Drift type
SEA Concepts	50K	3	2	abrupt
Rotating Hyperplane	200K	10	2	incremental
Moving RBF	200K	10	5	incremental
Interchanging RBF	200K	2	10	abrupt
Moving Squares	200K	2	4	incremental
Transient Chessboard	200K	2	8	virtual
Mixed Drift	600K	2	8	abr/incr/virt

kNN with Sliding Window
This classifier is a kNN with a fixed window size containing the most recent samples. The sliding window is a standard approach for drift handling, since the window contains usually the most relevant examples for the future predictions. However, the window can contain outdated examples interfering with the current concept leading to a deterioration of the prediction accuracy.
kNN with Probabilistic Adaptive Window and ADWIN
In contrast to the approach with the sliding window, examples are here removed randomly leading to a window which not only contains recent samples but also older ones. The window size is not bounded strictly and varies around a target size (see [5]). This approach also uses ADWIN for change detection and clears the window accordingly.
Table I gives an overview of the algorithms as well as the chosen hyperparameter. A maximum of 5000 samples was allowed as size for the window based approaches. However, we limited it to at most 10% of the whole dataset for those with rather few examples. LPPNSE demands a chunk size which is critical for its performance. To avoid any disadvantage we evaluated several chunksizes and report the best achieved result. No further dataset specific hyperparameter tuning was done, as we wanted to use as little prior knowledge as possible.

A. Datasets

We used own and well known artificial as well as real world datasets to compare our method with state of the art algorithms. Links to all datasets as well as our own ones are available at https://github.com/vlosing/Online-learning. In the following we describe the data more detailed.

For the artificial data either published benchmarks were taken or generated with MOA using common parametrization in the literature. We also added four new datasets allowing the evaluation of particular algorithm properties which are in our opinion not yet considered enough in the community. Table II shows their main characteristics.
SEA Concepts (SEA)
This two class dataset was proposed in [6] and consists of 50000 samples with three attributes of which only two are relevant. Abrupt drift is simulated with four different concepts, changing every 12500 samples, by using different thresholds θ_i such that $f_1 + f_2 > \theta_i$. This dataset includes 10% of noise.

TABLE III. CONSIDERED REAL WORLD DATASETS. THE GIVEN DRIFT TYPE WAS DETERMINED AS DESCRIBED IN ??.

Dataset	#Samples	#Feat.	#Class	Drift type
Weather	18159	8	2	virtual
Elec	27549	6	2	real
CovType	581012	54	784	real
Outdoor	4000	21	40	real
Railto	40000	27	10	real

Rotating Hyperplane (HYP)
A hyperplane in d-dimensional space is defined by the set of points x that satisfy $\sum_{i=1}^{d} w_i x_i = w_0$. The position and orientation of the hyperplane are changed incrementally by continuously adding a term δ to the weights $w_i = w_i + \delta$. We used the generator in MOA with the same parameters as in [5] (10 dimensions, 2 classes, δ=0.001).
Moving RBF (MRBF)
Gaussian distributions with random initial positions, weight and standard deviations are moved with constant speed v in d-dimensional speed. The weight controls the partitioning of all samples among the Gaussians. We used the generator in MOA with the same parameters as in [5] (10 dimensions, 50 Gaussians, 5 classes, v=0.001).
Interchanging RBF (IRBF)
Ten Gaussians with random covariance matrix are exchanging positions every 2000 samples and generate thereby a total of 100 abrupt drifts.
Moving Squares (SQR)
Four equidistantly seperated squares, representing different classes, are moving in horizontal direction with constant speed. The direction is inverted whenever the leading squares reaches a predefined boundary. The nice property of this dataset is that the upper bound of stored recent examples such that old samples do not overlap current ones can be easily calculated (120) and facilitates the analysation of algorithms, especially those using a sliding window approach, for incremental drift.
Transient Chessboard (CHESS) This dataset simulates virtual drift by succesively revealing random squares of a chessboard. Each time after four squares have been revealed, samples covering all squares of the board are shown. facilitating the classification for algorithms which preserve as much information as possible.
Mixed Drift (MIX)
The datasets interchanging RBF, moving squares and transient chessboard datasets are simply positioned next to each other and samples belonging to one of these are introduced alternately. Therefore, incremental, abrupt and virtual drift are taking place at the same time, requiring local adaptation to different drift types.

Unfortunately, only a few real world drift benchmarks are available, of which we used the largest ones. Additionaly, we contributed two challenging new datasets obtained from visual data. The main attributes of all considered real world datasets are given in Table III.
Weather
Elwell et al. introduced this dataset in [1]. Using eight different features such as temperature, pressure wind speed etc. the target is to predict whether it is going to rain on a certain

day or not at the Offutt Air Force Base in Bellevue, Nebraska. A period of 50 years is covered (1949-1999) summing up to 18159 samples with an imbalance towards no rain (69%).

Electricity market dataset

This problem is often used as a benchmark for concept drift classification. Initially described in [7], it was used thereafter for several performance comparisons [8], [9], [5], [10]. A critical note to its suitability as a benchmark can be found in [11]. The dataset holds information of the Australian New South Wales Electricity Market, whose prices are affected by supply and demand. Each sample characterized by attributes such as day of week, timestamp, market demand etc. refers to a period of 30 minutes and the class label identifies the relative change (higher or lower) compared to the last 24 hours. The dataset is often termed ELEC2 and contains 45312 samples. However, we removed those with missing values leading to a total of 27449 points.

Forest Cover Type

Assigns cartographic variables such as elevation, slope, soil type asf of 30×30 meter cells to different forest cover types. Only forests with minimal human-caused disturbances were used, so that resulting forest cover types are more a result of ecological processes. It is often used as a benchmark for drift algorithms [5], [12], [13].

Outdoor Objects

This visual dataset was obtained from images recorded by a mobile robot approaching 40 different objects in a garden environment [14]. The lighting conditions between the approaches are varying significantly caused by different weather conditions and/or cast shadows making the classification quite challenging. Each approach consists of 10 images and is represented in temporal order within the dataset. Even though the representation is quite stable during a single approach, there is a change of varying degree between different approaches of the same object XXVL pictures. The objects are encoded in a normalized 21-dimensional rg-chromaticity histogram.

Railto Bridge Timelapse

Ten of the colorful buildings next to the famous rialto bridge in Venice are encoded in a normalized 27-dimensional rgb histogram. The images were obtained from timelapse videos of 10 consecutive days recorded in may 2016 by a webcam of XXVL. Continuously changing weather and lighting conditions affect the representation XXVL see figures. We excluded overnight recordings since they were too dark for being useful.

1) Assessing the drift type in real world data: While the drift is explicitly generated in artificial datasets, it is rather difficult to identify the drift type or whether drift is present at all in real world datas. Therefore, it is usually assumed in the comunity that there is some drift contained in the commonly evaluated datasets. We propose in this section a method which is able to determine the present drift type in a given dataset. To the best of our knowledge this has not been done so far in literature. Our two staged approach can distinguish between real, virtual or no drift at all. In the first step we are able to conclude whether real drift is present. If that is not the case, we continue with the second step which detects virtual drift. Whenever both tests are negative we infer that no drift is present at all. We heavily use sliding windows of various sizes for this analysis. The type of classifier is interchangeable, however, we used once again KNN.

The idea for the first step is inspired by the observation that whenever real drift is present, sliding windows of smaller sizes tend to deliver higher accuracies than larger ones. This contradicts the classical assumption for i.i.d datasets, backed up by the PAC theory(XXVL reference!), that the more data is stored in a model the less mistakes are done. However, streaming data is not i.i.d and more mistakes are done whenever outdated data from former concepts is in conflict with samples of the current one. A window which contains less of the outdated samples, delivers therefore a better result.

We propose to test whether the accuracies achieved with sliding windows of different sizes are significantly higher than the one obtained without any size restriction. To obtain multiple accuracies for a given window side we generate bootstrapped samples of the dataset maintaining the initial order. Afterwards, we perform a one sided hypothesis test with the traditional p-value of 5%. The method is described exemplarly for one window size s1 in the following.

Given a dataset X with n examples, we generate b bootstrap samples $\Psi = \hat{X}_1, \hat{X}_2 ... X_b$of the dataset and sort them such that the initial order in the dataset is preserved. We train a classifier with window size s_1 and another one with s_n using Ψ and asses their accuracies $a_1, a_2, ...a_b$ and $\tilde{a}_1, \tilde{a}_2, ...\tilde{a}_b$ respectively. The pth percentile of all accuracy differences $d_i = a_i - \tilde{a}_i$ is calculated. The null hypothesis, accuracy $a_i <= \tilde{a}_i$, is rejected if its value is smaller 0.

In case of no real drift, we proceed with the second step and test for virtual drift. Here we permutate the dataset each time before generating the bootstrapped samples and compare the resulting accuracies with those obtained in the first stage using the same window size. The permutation mixes data of all concepts. If P(X) is changed over time an algorithm should achieve higher accuracies, since it is easier to classify only one concept at a time instead of all of them together.

Next to the real world data we also analyzed the drift type of artificial data as a proof of concept. We used 200 bootstrap samples per window size. Table Va shows the mean accuracies of the bootstrapped samples with corresponding standard deviations and the inferred drift type. Our approach classifies correctly the drift type of all artificial datasets. In the case of the artificial datasets the increase/decrease of performance with shrinking window sizes is quite significant for datasets with / without real drift. This is generally less pronounced for real real world data. Nonthelss, the datasets Elec, Outdoor and Rialto clearly benefit from smaller window sizes. Covtype contains little real drift, while the Weather task incorporates only virtual drift.

B. Results

We evaluated the algorithms in the standard online learning setting in which each incoming example is first classified and afterwards used for training. The error rates of all experiments are shown in Table V.

(XXVL name of algorithm) outperforms the other algorithms quite significantly by having nearly half the error rate in average compared to the second best method LVGB. Even more important is in our eyes the fact that while all other methods are struggling at some datasets our aproach delivers very robust results without any hiccup. All drift types are handled better or at least competitive compared to the other algorithms. This

TABLE IV. MEAN ACCURACIES AND CORRESPONDING STANDARD DEVIATIONS OBTAINED FROM BOOTSTRAPPED SAMPLES OF THE DATASETS WITH VARYING SLIDING WINDOW SIZES.

Dataset	KNN$_{\text{S-100}}$	KNN$_{\text{S-500}}$	KNN$_{\text{S-1000}}$	KNN$_{\text{S-5000}}$	KNN$_{\text{S-10000}}$	KNN	Concluded drift type
Weather	82.03(0.12)	84.05(0.24)	84.36(0.25)	84.95(0.18)	84.92(0.20)	84.86(0.16)	not real
Elec	**89.08(0.18)**	**87.46(0.17)**	**86.61(0.16)**	**85.07(0.11)**	**84.22(0.14)**	83.43(0.13)	real
CovType	94.65(0.01)	96.15(0.01)	**97.11(0.01)**	**96.95(0.01)**	**96.92(0.01)**	5	real
Outdoor	89.08(0.18)	87.46(0.17)	86.61(0.16)	85.07(0.11)	84.22(0.14)	83.43(0.13)	real
Railto	89.08(0.18)	87.46(0.17)	86.61(0.16)	85.07(0.11)	84.22(0.14)	83.43(0.13)	real
Chessboard i.i.d.	63.23(0.07)	83.63(0.07)	88.47(0.06)	94.67(0.03)	96.15(0.04)	98.37(0.02)	not real
Transient Chessboard	81.00(0.05)	88.56(0.04)	88.85(0.04)	94.81(0.02)	95.89(0.03)	98.36(0.03)	not real
Moving Squares	**97.73(0.02)**	**67.22(0.05)**	**63.30(0.05)**	56.87(0.11)	57.22(0.04)	57.12(0.10)	real
Interchanging RBF	97.73(0.02)	67.22(0.05)	63.30(0.05)	56.87(0.11)	57.22(0.04)	57.12(0.10)	real

(a) Bootstrapped samples for detecting real drift. Accuracies of size restricted windows which are significantly higer compared to the accuracy achieved without restriction are marked bold. If this is the case at least for one window size it is inferred that the dataset contains real drift.

Dataset	KNN$_{\text{S-100}}$	KNN$_{\text{S-500}}$	KNN$_{\text{S-1000}}$	KNN$_{\text{S-5000}}$	KNN$_{\text{S-10000}}$	KNN	Concluded drift type
Weather	**81.14(0.30)**	**83.20(0.23)**	83.89(0.28)	84.42(0.17)	84.77(0.19)	84.63(0.18)	virtual
Chessboard i.i.d.	63.25(0.05)	83.63(0.06)	88.44(0.06)	94.71(0.04)	96.18(0.02)	98.36(0.01)	None
Transient Chessboard	**63.30(0.09)**	**83.77(0.06)**	**88.48(0.05)**	94.77(0.03)	96.20(0.05)	98.38(0.02)	virtual

(b) Permutated bootstrapped samples for detecting real drift. Accuracies which are significantly higher compared to those achieved in a) with the same window size are marked bold and it is assumed that the corresponding dataset incorporates virtual drift

TABLE V. ERROR RATES OF ALL EXPERIMENTS EVALUATED WITH THE INTERLEAVED TEST-THEN-TRAIN METHOD.

Dataset	HT$_A$	LPPNSE	DACC	LVGB	kNN$_S$	kNN$_{W_A}$	kNN$_M$
SEA	13.8	10.00	10.00	10.00	10.00	10.00	10.00
HYP	10.00	10.00	10.00	10.00	10.00	10.00	10.00
MRBF	10.00	10.00	10.00	10.00	10.00	10.00	10.00
IRBF	10.00	10.00	10.00	10.00	10.00	10.00	10.00
SQR	10.00	10.00	10.00	10.00	10.00	10.00	10.00
CHESS	10.00	10.00	10.00	10.00	10.00	10.00	10.00
MIX	10.00	10.00	10.00	10.00	10.00	10.00	10.00
Art. avg	10.00	10.00	10.00	10.00	10.00	10.00	10.00
Art. rank	10.00	10.00	10.00	10.00	10.00	10.00	10.00
Weather	10.00	10.00	10.00	10.00	10.00	10.00	10.00
Elec	10.00	10.00	10.00	10.00	10.00	10.00	10.00
CovType	10.00	10.00	10.00	10.00	10.00	10.00	10.00
Outdoor	10.00	10.00	10.00	10.00	10.00	10.00	10.00
Railto	10.00	10.00	10.00	10.00	10.00	10.00	10.00
Real avg	10.00	10.00	10.00	10.00	10.00	10.00	10.00
Real rank	10.00	10.00	10.00	10.00	10.00	10.00	10.00
total avg	10.00	10.00	10.00	10.00	10.00	10.00	10.00
total rank	10.00	10.00	10.00	10.00	10.00	10.00	10.00

is particularly clarified in the large accuracy gap achieved with the MIX dataset, which contains incremental, abrupt and virtual drift at the same time.
The methodically most similar method to ours is kNN$_{W_A}$, since it uses also kNN as classifier and actively manages its window. However, it performs in all experiments worse, even in those containing only abrupt drift.
Our results confirm the fact that kNN is in general a very competitive algorithm in the streaming setting. It is quite surprising that the simple sliding window approach of fixed window size kNN$_S$ performs comparably well or even better than more sophisticated methods such as HTAdaptive or L++.NSE.

-Table with results -Evaluation

C. Memory behaviour

-STM-Size behaviour for SquaresIncr, rbfAbrupt -Check prior behaviour for experiments -Pictures of LTM for chess, rbfAbrupt

VI. CONCLUSION

REFERENCES

[1] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1517–1531, Oct 2011.

[2] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in *Machine learning and knowledge discovery in databases*. Springer, 2010, pp. 135–150.

[3] N. C. Oza, "Online bagging and boosting," in *Systems, man and cybernetics, 2005 IEEE international conference on*, vol. 3. IEEE, 2005, pp. 2340–2345.

[4] G. Jaber, A. Cornuéjols, and P. Tarroux, "Online learning: Searching for the best forgetting strategy under concept drift," in *Neural Information Processing*. Springer, 2013, pp. 400–408.

[5] A. Bifet, B. Pfahringer, J. Read, and G. Holmes, "Efficient data stream classification via probabilistic adaptive windows," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ser. SAC '13. New York, NY, USA: ACM, 2013, pp. 801–806. [Online]. Available: http://doi.acm.org/10.1145/2480362.2480516

[6] W. N. Street and Y. Kim, "A streaming ensemble algorithm (sea) for large-scale classification," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 377–382. [Online]. Available: http://doi.acm.org/10.1145/502512.502568

[7] M. Harries and N. S. Wales, "Splice-2 comparative evaluation: Electricity pricing," 1999.

[8] M. Baena-Garcıa, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, and R. Morales-Bueno, "Early drift detection method," in *Fourth*

international workshop on knowledge discovery from data streams, vol. 6, 2006, pp. 77–86.

[9] L. I. Kuncheva and C. O. Plumpton, “Adaptive learning rate for online linear discriminant classifiers,” in *Structural, Syntactic, and Statistical Pattern Recognition*. Springer, 2008, pp. 510–519.

[10] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, “Learning with drift detection,” in *Advances in artificial intelligence–SBIA 2004*. Springer, 2004, pp. 286–295.

[11] I. Zliobaite, “How good is the electricity benchmark for evaluating concept drift adaptation,” *arXiv preprint arXiv:1301.3524*, 2013.

[12] J. Gama, R. Rocha, and P. Medas, “Accurate decision trees for mining high-speed data streams,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 523–528.

[13] N. C. Oza and S. Russell, “Experimental comparisons of online and batch versions of bagging and boosting,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 359–364.

[14] V. Losing, B. Hammer, and H. Wersing, “Interactive online learning for obstacle classification on a mobile robot,” in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–8.