ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Concept Drift Segmentation via Kolmogorov-Trees

Fabian Hinder[1] and Barbara Hammer[1] *

1- Bielefeld University - Cognitive Interaction Technology (CITEC)
Inspiration 1, 33619 Bielefeld - Germany

Abstract. The notion of concept drift refers to the phenomenon that the data distribution changes over time. If drift occurs, machine learning models need adjustment. Since drift can be inhomogeneous, suitable actions depending on the location in data space. In this paper we address the challenge to partition the data space into segments with homogeneous drift characteristics. We formalize this objective as an independence criterion, and derive a robust and efficient training algorithm based thereon. We evaluate the efficiency of the method in comparison to existing technologies: the identification of drifting clusters, and the estimation of a conditional density distribution.

1 Introduction

Data from the real world such as social media entries or measurements of IoT devices are subject to continuous change [1, 2]. Here concept drift can be caused by seasonal changes, changed demands, ageing of sensors, etc. Since the characteristics of drift might induce severe problems of the accuracy of machine learning models, it is important to understand the nature of drift. In recent years, quite a few approaches were proposed to deal with concept drift [3, 4]. These range from non-parametric methods over gradient techniques up to ensemble technologies for dealing with streaming data [5]. In addition to model adaptation schemes, a large number of methods aims for a detection of drift, an identification of change points in given data sets, or a characterization of overarching types of drift [6, 7]. Currently, only few methods aim for a more detailed analysis of the spatial characteristics of drift [4, 8]. These are restricted to an analysis of the drift behavior for two consecutive time points corresponding to abrupt drift. As concept drift can happen in different ways [9], such as abrupt vs. gradual, periodic vs. nonrecurring, slow vs. fast drift, a more general analysis of the spatial peculiarities of drift would be desirable.

The aim of the present work is to uncover the spatial structure of drift by segmenting the observed data into spatial regions which possess an identical or comparable drift characteristics. We will provide a formal definition of such drift segmentation by referring to a suitable formalization in terms of conditional independence. We provide a non-parametric, linear time segmentation algorithm based on this formalization, and we show its practical relevance in a number of benchmarks for downstream tasks, comparing the result to state-of-the art methods. This paper is organized as follows: In the first section we recall the definition of concept drift, provide the definition of drift segmentation, and

*Funding in the frame of the BMBF project ITS.ML, 01IS18041A is gratefully acknowledged.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

compare it to other notions. Then we present two approaches to the segmentation problem in section 3 – we relate the first one to existing methods from decision tree learning and we provide a proof of correctness for the other. In section 4 we compare our approaches to other methods form the respective fields as regards their capability of detecting regions of change, and their suitability as preprocessing methods for conditional density estimation.

2 Problem definition

In classical machine learning one considers a generative process p on the sample space $\mathcal{X}$. A data point is an instance of a random variable $X \sim p$. Many processes in real-world applications are time dependent. One prominent way to to take time into account, is to consider a family of probability measures p_t on $\mathcal{X}$, indexed over a set $\mathcal{T}$, representing time. Probabilities p_t can change over time. Concept drift takes place if $p_t \neq p_s$ for at least one pair $t \neq s$ [9]. One powerful mathematical modeling of such drift processes has been suggested in [10]: one considers random variables $(X, T) \sim \mathbb{P}_{(X,T)}$ representing data and time respectively, with an underlying joint distribution $\mathbb{P}_{(X,T)}$, whereby the $\mathcal{T}$-valued random variable T yields the conditional distribution $\mathbb{P}_{X|T=t} = p_t$. It has been shown in [10] that drift is uniquely characterized by the statistical dependence of X and T.

We aim for drift segmentation as a decomposition of the space of observations $\mathcal{X}$ into regions with the same drifting behavior. As an example, consider sports news, and assume a soccer club A newly joins premier league. Then, news which are not concerning soccer or the home town of A are likely not affected by the drift, while news concerning premier league or news about the home town of A are likely affected. Thus, we would aim for a decomposition of news into three subparts in this case: premier league, home town of A, and the rest. The setting might be more complex, if we consider a longer time period. In mathematical terms, we can formalize this intuition in terms of a decomposition of the space $\mathcal{X}$ which leaves the observed probabilities invariant as follows:

Definition 1. Assume we are given a drift process $(X, T) \sim \mathbb{P}_{(X,T)}$. A *drift segmentation* of (X, T) is a measurable map $L : \mathcal{X} \to \mathbb{N}$, which assigns each element of $x \in \mathcal{X}$ an index $L(x)$ corresponding to the segment it belongs to, such that $\mathbb{P}_{T|L(X)} = \mathbb{P}_{T|X}$. A segmentation is optimal if $\mathcal{L} := L(\mathcal{X}) = \{l_1, ..., l_n\}$ is finite and the number of segments n is minimal. We obtain a partition of $\mathcal{X}$ into segments by considering the preimages $\{x \in \mathcal{X} \mid L(x) = l_i\}$ of L.

This definition can be interpreted as grouping all points with the same drifting behavior in the same segment and be rephrased as follows:

Lemma 1. *Let $L : \mathcal{X} \to \mathbb{N}$. Then L is a drift segmentation if and only if T and X are independent given $L(X)$, i.e. $T \amalg X | L(X)$.*

Proof (sketch). Since L is deterministic, it holds $\mathbb{P}_{T|X} = \mathbb{P}_{T|X,L(X)}$. Furthermore, we have $\mathbb{P}_{T|X,L(X)} = \mathbb{P}_{T|L(X)}$ if and only if $T \amalg X | L(X)$. □

Hence, a finite minimal drift segmentation does not necessarily exist. However, every distribution can be arbitrarily good approximated by one that admits

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

a finite drift segmentation, and finite observational data can always be modeled. Up to our knowledge, drift segmentation has not yet been considered in this form in the literature. However, the notion is closely related to the following two objectives, for which benchmarks exist:

Drift localization: Drift localization as addressed in [4] aims for an identification regions where drift occurs, i.e. elements $x \in \mathcal{X}$ such that $p_t(x) \neq p_s(x)$ for some time points $t \neq s$. Drift segmentation constitutes a generalization of drift localization as it does not only identify the regions where drift occurs, but it also groups them according to their drift characteristic over possibly longer time period. Existing approaches for the localization of drift are typically limited to a single change point [11, 12] or supervised setups [13].

Conditional density estimation: Conditional density estimation $\mathbb{P}_{Y|X}$ for a real-valued variable Y constitutes a prominent problem, which can be addressed with conditional kernel density estimates or least squares approaches, for example [14, 15, 16]. By identifying Y and T, drift segmentation can serve as a preprocessing step for conditional density estimation: a drift segmentation yields a decomposition of $\mathcal{X}$ into regions without local dependence of X and T, hence we obtain approximately X-invariant local distributions of T within these segments. Thus, the inference of $\mathbb{P}_{T|X}$ boils down to an unconditional density estimator on the data contained in the respective region $\mathbb{P}_{T|L(X)}$.

3 Efficient methods for drift segmentation

In the following we will assume $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{T} = [0, 1]$. Albeit the criterion of conditional independence allows us to check whether a given partition is suitable as drift segmentation, the criterion does not provide a good strategy like the simple greedy approach of successively splitting $\mathbb{R}^d$ at the most suitable position, as is illustrated by the following example: Assume $X, T \sim \frac{1}{3}(\delta_{(0,0)} + \delta_{(1,\frac{1}{2})} + \delta_{(2,1)})$, where δ_x denotes the Dirac-measure concentrated at x. Here, any single split keeps X and T dependent. Therefore, we aim for a different approach: we search for a split into l_1 and l_2 such that distributions $\mathbb{P}_{T|X\in l_1}$ and $\mathbb{P}_{T|X\in l_2}$ are as different as possible.

3.1 A mean value approach

Comparing distributions can be done by a reference characteristic such as the expectation of T. That means, we aim for a split l_1, l_2 such that $\mathbb{E}[T|X \in l_1]$ and $\mathbb{E}[T|X \in l_2]$ differ significantly. This task can be performed using Welch's t-test. Note that, there is a close connection between this test approach and the usual variance reduction gain used in regression trees:

$$\sqrt{I_V(l_1, l_2)} = \sqrt{\frac{nm}{n+m}} |\hat{\mathbb{E}}[T|X \in l_1] - \hat{\mathbb{E}}[T|X \in l_2]|,$$

where n, m are the number of samples in the respective subset. Normalizing the right hand side by $(\widehat{\mathrm{var}}(T|X \in l_1)/n + \widehat{\mathrm{var}}(T|X \in l_2)/m)^{\frac{1}{2}}$ we obtain the statistic of Welch's t-test (up to a sign). Hence, the main difference between both strategies is that the latter also takes certainty regarding the statistical stability of the estimated quantity into account.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

3.2 Kolmogorov splitting

The restriction to expectations has major limitations. As an example, consider $(X,T) \sim \frac{1}{2}(\delta_0 \times \mathcal{N}(0,1) + \delta_1 \times \mathcal{N}(0,2))$ which yields $\mathbb{E}[T|X=x] = 0$ for every x, but there is a reasonable splitting between $X=0$ and $X=1$. To overcome this problem we make use of the classical Kolmogorov-Smirnov test, a non-parametric two-sample test of the equality of continuous one-dimensional real valued distributions. Its statistic is based on the difference between empirical distribution functions. For realizations $(x_1,t_1),...,(x_N,t_N)$ of (X,T), ordered such that $t_i \leq t_{i+1}$, one can compute the statistic of the Kolmogorov-Smirnov test for a split l_1, l_2 as

$$\left\| \hat{F}_{T|X\in l_1} - \hat{F}_{T|X\in l_2} \right\|_\infty = \max_{1\leq k\leq N} \left| \frac{k}{n} - \frac{N}{n\cdot(N-n)} \sum_{i=1}^{k} \mathbb{I}_{l_2}(x_i) \right|,$$

where n is the number of samples in l_1, $\mathbb{I}_A$ is the indicator function, and $\hat{F}$ the empirical distribution function. Note that, this is the only estimate needed to compute the p-value. Since it is sufficient to sort the data once at the beginning, a quasilinear $\mathcal{O}(N \log N)$ run time results for the construction of a complete decision tree for concept drift segmentation. In the following, we will use a classical decision tree algorithm to grow a decision tree similar to e.g. CART but using the p-value of the Kolmogorov-Smirnov test as split criterion. We refer to this model as *Kolmogorov trees*. Such trees allow a drift segmentation:

Theorem 1. *If the algorithm of CART, Random Forest or Extra-Tree is used with Kolmogorov splitting as split criterion, then the obtained trees admit arbitrarily good segmentation. This means that for all $\varepsilon > 0$ there exists a tree with leaves L, such that $\mathbb{E}[\|F_{T|L(X)} - F_{T|X}\|_\infty] < \varepsilon$ assuming a sufficient amount of data is present, and such trees are the result of the proposed algorithm.*

Proof (sketch). Let $n \in \mathbb{N}$ be such that $4/n < \varepsilon$. Find a tree that approximates the quantiles $1/n, 2/n, ..., (n-1)/n$ sufficiently well, using the algorithm. The result follows from the triangle inequality. □

Early stopping: Flat decision trees reduce the computational complexity during the prediction phase and increase the generalization ability, but result in less flexible decision rules. As Kolmogorov information gains are p-values, the growths of a tree can be stopped, if all splittings are above a critical threshold: this indicates homogeneous regions such that no further splitting is needed.

4 Experiments

To evaluate our approach we consider two experimental settings, drift localization and conditional density estimation. We compare to state-of-the-art algorithms from these fields. We use forests of 20 Extra-trees grown with Kolmogorov split and early stopping at $p = 0.01$.

Drift Localization / Segmentation: We use our method for the task of drift localization given the presence of drift has been detected at a specific time point [4]. For Kolmogorov trees, regions of drift can be identified as those leaves for which the ratio of samples before and after the detected time of drift differs

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Table 1: Experimental results over 200 runs. Mean accuracy and standard deviation are shown. Significantly ($p = 0.01$) better results are printed boldface. n is the number of noise dimensions, cpt is the number of clusters per time.

cpt	n	Kolmogorov	k-NN	LDD-DSI	kdq-Tree
9	0	**0.87**(±**0.09**)	**0.86**(±**0.07**)	0.60(±0.03)	0.78(±0.11)
9	1	**0.86**(±**0.11**)	0.75(±0.07)	0.49(±0.06)	0.70(±0.09)
18	0	0.73(±0.09)	**0.78**(±**0.05**)	0.60(±0.03)	0.72(±0.08)
18	1	**0.74**(±**0.09**)	0.69(±0.04)	0.48(±0.06)	0.66(±0.06)
18	5	**0.71**(±**0.10**)	0.58(±0.01)	0.37(±0.02)	0.48(±0.05)

from 1. We compare our method to LDD-DSI [11], which refers to the local density between two time windows using a nearest neighbour approach, and kdq-trees [12], which splits the data beforehand and then checks the respective leaf-regions for drift. We also consider k-NN with optimal (post hoc) ratio. We use standard parameters for all methods.

For an evaluation we use artificial data with ground truth, induced by a given cluster structure. While drifting clusters either appear ($T = 1$) or vanish ($T = 0$) at the given time point, non-drifting cluster have a uniform propbability before and after the drift, i.e. $\mathbb{P}(T = t) = \frac{1}{2},\ t \in \{0, 1\}$. Different cluster numbers and dimensionality were used. The latter add noise to the two-dimensional cluster structure. The task is now to decide, whether a sample belongs to a cluster that is drifting, or not. Note that, in this setup, drift localization and segmentation coincide. The results are displayed in Table 1. Kolmogorov trees yield superior results. k-NN is strongly affected by noise. kdq-trees have problems in particular for a larger number of clusters. LDD-DSI is overall weak.

Conditional density estimation: We use Kolmogorov trees for conditional density estimation in supervised regression problems, using three benchmarks from UCI and one from [17] (see Table 2). Kolmogorv trees are used by enhancing leaf nodes with standard (unconditional) kernel density estimates. As reference methods, we use conditional kernel density estimates [14, 15], and least squares methods [16, 15] which model the conditional density as a mixture model and which are also used in (supervised) drift detection [18]. In addition, we use standard regression trees. Hyperparameters are optimized using cross validation. The results are displayed in Table 2. As can be seen the tree based approaches perform best. MSE performs particularly well if a high correlation between density and mean can be observed; Kolmogorov trees perform particularly well for heterogeneous data sets.

Table 2: Experimental results over 200 runs. Table shows mean negative log-likelihood and standard deviation. Significantly ($p = 0.01$) better results are printed in bold face. Number in brackets denotes the number of "pearls".

	Kolmogorov	LS-CDE	MSE	ϵ-KDE
boston	**0.45**(±**0.04**)	0.65(±0.10)	**0.44**(±**0.06**)	1.17(±0.05)
california housing	0.83(±0.03)	0.89(±0.04)	**0.74**(±**0.04**)	1.05(±0.03)
diabetes	1.11(±0.03)	1.18(±0.05)	**1.08**(±**0.04**)	1.73(±0.05)
Gauss necklace (3)	**1.25**(±**0.03**)	1.29(±0.04)	1.31(±0.04)	1.46(±0.05)
Gauss necklace (6)	**1.22**(±**0.02**)	1.25(±0.03)	1.31(±0.04)	1.43(±0.04)

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

5 Discussion

In this work we introduced a new way to tackle concept drift by formalizing drift segmentation as the challenge to decompose the observational space into homogeneous regions as regards underlying drift characteristics. We derived a new information criterion to judge according splits, based on which an efficient tree segmentation algorithm has been derived. We showed the usefulness of this criterion for two different application areas and explored its properties in empirical experiments.

References

[1] A. Bifet and J. Gama. Iot data stream analytics. *Ann. des Télécomm.*, 75(9-10), 2020.

[2] S. Tabassum, F. S. F. Pereira, S. Fernandes, and J. Gama. Social network analysis: An overview. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 8(5), 2018.

[3] Gregory Ditzler, Manuel Roveri, Cesare Alippi, and Robi Polikar. Learning in nonstationary environments: A survey. *IEEE Comp. Int. Mag.*, 10(4):12–25, 2015.

[4] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *IEEE TKDE*, 2018.

[5] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Wozniak. Ensemble learning for data stream analysis: A survey. *Inf. Fusion*, 37, 2017.

[6] S. Aminikhanghahi and D. J. Cook. A survey of methods for time series change point detection. *Knowl. Inf. Syst.*, 51(2):339–367, May 2017.

[7] I. Goldenberg and G. I. Webb. Survey of distance measures for quantifying concept drift and shift in numeric data. *Knowl. Inf. Syst.*, 60(2):591–615, 2019.

[8] G. I. Webb, L. K. Lee, F. Petitjean, and B. Goethals. Understanding concept drift. *CoRR*, abs/1704.00362, 2017.

[9] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):44:1–44:37, March 2014.

[10] F. Hinder, A. Artelt, and B. Hammer. Towards non-parametric drift detection via dynamic adapting window independence drift detection (dawidd). In *ICML*, 2020.

[11] A. Liu, Y. Song, G. Zhang, and J. Lu. Regional concept drift detection and density synchronized drift adaptation. In *IJCAI*, 2017.

[12] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi. An information-theoretic approach to detecting changes in multidimensional data streams. *Interfaces*, 01 2006.

[13] J. Gama and G. Castillo. Learning with local drift detection. In Xue Li, Osmar R. Zaïane, and Zhanhuai Li, editors, *Advanced Data Mining and Applications*, pages 42–55, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[14] P. Hall, R. C. L. Wolff, and Q. Yao. Methods for estimating a conditional distribution function. *Journal of the American Statistical Association*, 94(445):154–163, 1999.

[15] J. Rothfuss, F. Ferreira, S. Walther, and M. Ulrich. Conditional density estimation with neural networks: Best practices and benchmarks. *arXiv:1903.00954*, 2019.

[16] M. Sugiyama, I. Takeuchi, T. Suzuki, T. Kanamori, H. Hachiya, and D. Okanohara. Conditional density estimation via least-squares density ratio estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 781–788. JMLR Workshop and Conference Proceedings, 2010.

[17] L. Fischer, B. Hammer, and H. Wersing. Optimal local rejection for classifiers. *Neurocomputing*, 214:445–457, 2016.

[18] L. Bu, C. Alippi, and D. Zhao. A pdf-free change detection test based on density difference estimation. *IEEE Transactions on Neural Networks and Learning Systems*, 29(2):324–334, 2018.