ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

Beta Distribution Drift Detection for Adaptive Classifiers

Lukas Fleckenstein, Sebastian Kauschke, and Johannes Fürnkranz

TU Darmstadt – Knowledge Engineering Group
Hochschulstrasse 10, 64289 Darmstadt, Germany

Abstract. With today's abundant streams of data, the only constant we can rely on is change. For stream classification algorithms, it is necessary to adapt to concept drift. This can be achieved by monitoring the model error, and triggering counter measures as changes occur. In this paper, we propose a drift detection mechanism for batch-data that fits a beta distribution to the model error, and treats abnormal behavior as drift. It works with any given model, leverages prior knowledge about this model, and allows to set application-specific confidence thresholds. Experiments confirm that it performs well, in particular when drift occurs abruptly.

1 Introduction

In the domain of learning from data streams, it is highly probable that the target concept will change over time, i.e, that concept drift will affect the stream. This usually leads to the necessity of updating the model that deals with the data. Hence, many learning algorithms specifically geared towards this situation have been investigated. Some of them deal with drift implicitly, while others require explicit drift detection. A popular explicit approach towards the supervised learning scenario is to monitor the performance of the learner via its accuracy or error rate. When a significant change in the model's performance is detected, a mechanism for coping with this problem is invoked, for example by updating the model with current information [1, 2, 3].

In this paper, we present *Beta Distribution Drift Detection* (BD^3), a method that leverages previously known information about an existing classifier in the form of a beta distribution, and detects drift by assessing if new batches of data operate within the confidence bounds of that distribution. If previous knowledge does not exist, we gather it in the beginning of the drift detection process. The method works on batches of data, as opposed to other methods that work on single instances at a time. The batchwise approach is generally more stable w.r.t. the trade-off between false alerts and false negatives, and, as we think, applies to more real-world scenarios.

In Section 2 we define the problem, followed by the explanation of our algorithm in Section 3. We describe the experimental setup in Section 4, discuss our results in Section 5, and draw some conclusions in Section 6.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

2 Problem Definition

We assume a model M which classifies a stream of data instances D. The stream consists of batches D_i, $i \in 0...I$, where the number of batches I is large or potentially infinite. For each batch D_i, the model M classifies the n_i instances, producing a corresponding error batch E_i. Each error batch E_i consists of binary classifier errors $e_{i,j}$, where $e_{i,j} = 0$, if M predicts instance $j \in 0...n_i$ correctly, and $e_{i,j} = 1$ otherwise. The goal is to detect whether a batch E_i shows a significant increase in error rate compared to previous batches. We will approach this by fitting beta distributions to the classifier error, as described in Section 3.

3 Beta Distribution Drift Detection Method

The method presented in this paper is based on evaluating the beta distribution of the classification error of a model. For each new batch of data, the current classifier error is compared against this distribution. If it is outside a certain confidence interval, concept drift is assumed.

Model Initialization. The beta distribution has two shape parameters $\alpha, \beta > 0$, which we initially derive from previous knowledge about the error-rate π_0 of the model. As a rule of thumb we use $\alpha_0 = \pi_0 \cdot n_0$ and $\beta_0 = (1 - \pi_0) \cdot n_0$, where n_0 is the number of instances in the first batch that we receive. If π_0 is unknown, we suggest to set it to 0.5 as a starting value.

Batchwise Model Update. Given the most recent batch D_i, the detector receives the corresponding binary error batch E_i. For a given error rate π_i of the model on this batch, the likelihood for observing k_i misclassifications on the n_i samples is given by the binomial distribution $Bin(k_i|n_i, \pi_i)$. By putting a prior on that error rate, we are able to compute the posterior probability of the error rate given some data. Since it is a conjugate prior to the binomial, we choose a beta distribution $Beta(\pi_i|\alpha_i, \beta_i)$, where α_i and β_i represent the number of previously misclassified and correctly classified samples, respectively. It can be shown that:

$$P(\pi_i|E_i) = \frac{P(E_i|\pi_i) \cdot P(\pi_i)}{P(E_i)} = \frac{Bin(k_i|n_i, \pi_i) \cdot Beta(\pi_i|\alpha_i, \beta_i)}{P(E_i)} = Beta(\alpha_i^*, \beta_i^*) \tag{1}$$

with $\alpha_i^* = \alpha_i + k_i$ and $\beta_i^* = \beta_i + (n_i - k_i)$.

Drift Detection. We can now test if the error of a new batch of data is likely to correspond to the classifier's concept, or if a concept change occurred:

1. Compute *warning* and *drift* boundaries that contain 95.0% and 99.7% of the distribution $Beta(\alpha_{i-1}^*, \beta_{i-1}^*)$, and the current error rate $\pi_i = k_i/n_i$.
   If $\pi_i > upper\_bound_{warning}$ signal warning.
   If $\pi_i > upper\_bound_{drift}$ signal drift and reset shape parameters to

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

$\alpha^*_{i-1} = \alpha_0$, $\beta^*_{i-1} = \beta_0$, since the observed error does not correspond to the previous distribution. Reset test counter $t = 0$.

2. Set the parameters of the previous posterior distribution $Beta(\alpha^*_{i-1}, \beta^*_{i-1})$ as prior for the most recent one, $\alpha_i = \alpha^*_{i-1},\ \beta_i = \beta^*_{i-1}$.

3. Compute the most recent shape parameters $\alpha^*_i = \frac{\alpha_i}{\text{decay}} + k_i$, $\beta^*_i = \frac{\beta_i}{\text{decay}} + (n_i - k_i)$. Increment test counter $t = t + 1$.

The parameter *decay* is used to control the influence of prior knowledge given by the previous batches. As the reliability of that prior increases with the number of tests t, we decrease *decay* according to $decay = 1/\exp(a \cdot (t + b)) + 1.1$. We choose $a = 0.15$, $b = -7$ based on preliminary experiments as a trade-off between abrupt and gradual drift detection. It can be shown that $\lim_{i\to\infty} \alpha^*_i + \beta^*_i = n + \frac{n}{decay-1}\ \ \forall\ decay > 1$. Thus, the decay parameter limits the variance of the beta distribution, preventing it from becoming too narrow, which would increase false alerts. Figure 1 shows how different shape parameters α and β influence the beta distribution, even if their mean $\pi = \alpha/(\alpha + \beta)$ remains the same.

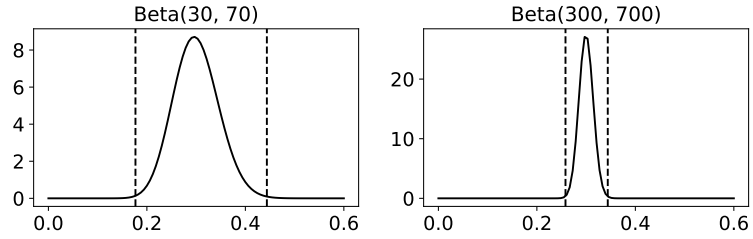

Fig. 1: Two example beta probability density functions with their boundaries that contain 99.7% of the distribution.

4 Experiment Setup

In this section we describe the datasets we chose for evaluation, the evaluation procedure, the used metrics, and the algorithms we compare against.

4.1 Benchmark Algorithms

We compare our novel approach against two tried-and-tested approaches which follow the idea of detecting drift from classifier error:[1]

DDM The Drift Detection Method by Gama et al. [2] relies on detecting statistically significant changes in the performance of a classifier via monitoring the mean error and its standard deviation. A drift is detected if a certain threshold σ is exceeded.

[1]Implemented in `https://scikit-multiflow.github.io`

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

EDDM The Early Drift Detection Method [3] is similar to *DDM*. Instead of monitoring the error rate, it monitors the distances between subsequent errors. This allows to better detect gradual and slow changes, thus eliminating a weakness of *DDM*.

4.2 Evaluation Datasets

We evaluate our findings on four datasets. Each scenario represents a different type of concept drift with varying severity and/or gradation:
Bit-Stream. The first dataset is a stream of bits from a Bernoulli distribution with parameter μ as proposed in [4]. Each stream contains 30 change points, separated by 600 or 2000 bits for which the concept is stable. In order to simulate different drift magnitudes, the maximum absolute difference between the means of two subsequent concepts is restricted in an interval $[a, b]$.
SEA Concepts. SEA is a drifting stream generation scheme [5] and used as a standard test for abrupt concept change. The dataset has three features, two determine the class and the third is noise. We generate streams of 40k instances with three change points at 10k, 15k, and 30k samples, and also vary between 10% and 20% class noise.
Rotating Hyperplane. This dataset has features equal to SEA, but contains a gradual drift behavior. Class labels depend on the placement of the two-dimensional points compared to a hyperplane that rotates during the course of the stream. It starts rotation with a certain angle every 1,000 instances, starting after the first 10k samples, the angles being 20°, 30°, and 40°.
Elec2. This dataset [6] is a real-world dataset of electricity prices. It contains 45,312 instances with eight features and binary class labels that indicate price change (UP or DOWN), and has an unknown number of drifts.

4.3 Evaluation Measures

For the comparison of the algorithms we use the following metrics:
FPR: The false positive rate, where the drift detection method detects a drift when there is actually none.
FNR: The false negative rate, where the model fails to detect a drift, when there is one.
Delay: The average number of batches between the true drift point and the first true positive detection on the same concept.

We run each algorithm 50 times with a batch size of 200 on every dataset and calculate the average values and standard deviations of FPR, FNR, and Delay for those runs. For the synthetic and real-world datasets we use a Naïve Bayes classifier in the *Interleaved Test-Then-Train* or *Prequential* framework, in which a new batch is first used for evaluating the accuracy and afterwards for updating the classifier (cf. [7] for more details). If a warning gets signaled, a second instance of the classifier starts training in parallel and replaces the primary one once a drift follows the warning, as proposed in [2].

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Table 1: Results on the Bit-Stream dataset.

Algorithm		600 bits between changes			1,000 bits between changes		
		[0.1, 0.3]	[0.3, 0.5]	[0.5, 0.7]	[0.1, 0.3]	[0.3, 0.5]	[0.5, 0.7]
DDM	FPR	**0.0310** (± 0.0232)	**0.0215** (± 0.0133)	**0.0197** (± 0.0127)	**0.0106** (± 0.0071)	**0.0062** (± 0.0050)	**0.0085** (± 0.0055)
	FNR	0.5579 (± 0.1996)	0.2254 (± 0.1975)	0.0115 (± 0.0523)	0.5011 (± 0.2180)	0.2180 (± 0.2783)	0.0114 (± 0.0800)
	Delay	0.6122 (± 0.3011)	0.3325 (± 0.1367)	0.0273 (± 0.0468)	2.3346 (± 0.9044)	1.4450 (± 0.3424)	0.4567 (± 0.1183)
EDDM	FPR	0.1719 (± 0.0832)	0.1831 (± 0.0588)	0.2197 (± 0.0461)	0.0861 (± 0.0458)	0.1018 (± 0.0359)	0.1099 (± 0.0299)
	FNR	0.4072 (± 0.1942)	0.0923 (± 0.1138)	**0.0000** (± 0.0000)	0.4656 (± 0.1731)	0.0950 (± 0.1438)	**0.0000** (± 0.0000)
	Delay	0.3508 (± 0.2182)	0.4337 (± 0.1317)	0.0530 (± 0.0603)	2.0603 (± 0.8537)	2.5416 (± 0.4280)	1.3600 (± 0.2792)
BD^3	FPR	0.0521 (± 0.0377)	0.0944 (± 0.0402)	0.0618 (± 0.0406)	0.0552 (± 0.0335)	0.0802 (± 0.0422)	0.0383 (± 0.0286)
	FNR	**0.0312** (± 0.0414)	**0.0000** (± 0.0000)	**0.0000** (± 0.0000)	**0.0270** (± 0.0421)	**0.0000** (± 0.0000)	**0.0000** (± 0.0000)
	Delay	**0.0189** (± 0.0383)	**0.0000** (± 0.0000)	**0.0000** (± 0.0000)	**0.0637** (± 0.1246)	**0.0000** (± 0.0000)	**0.0000** (± 0.0000)

5 Results

In Table 1, we show the results on the abruptly drifting bit-stream. Compared to *DDM* and *EDDM*, BD^3 generally shows low FNR and Delay values. Especially for the smallest change interval of $[0.1, 0.3]$, where *DDM* and *EDDM* show high levels of FNR, BD^3 performs better. On the synthetic datasets (Table 2), BD^3 shows comparable results to *DDM* and *EDDM*. Here, the performance depends heavily on the classifier algorithm, and the chosen drift detector has limited effect. However, the accuracy is marginally above *DDM* and *EDDM*, which we attribute to the lower delay achieved by BD^3. Figure 2 shows the accuracy as a stream on the gradually drifting rotating hyperplane dataset, where BD^3s advantage in delay shows by faster recovery on dips in the curve. On the Elec2 dataset (Table 3), BD^3 is the only drift detector that does not lower the accuracy below the levels of no detector.

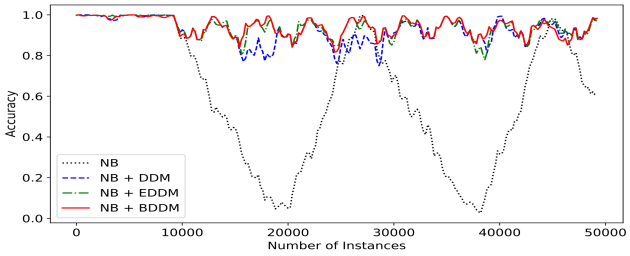

Fig. 2: Accuracy comparison on the Rotating Hyperplane dataset (20°).

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Table 2: Results on synthetic datasets (Accuracy).

Algorithm	SEA Concepts (Abrupt)			Rotating Hyperplane (Gradual)		
	0.0	0.1	0.2	20	30	40
No Detector	0.9137 (± 0.0014)	0.8487 (± 0.0015)	0.7631 (± 0.0015)	0.6046 (± 0.0079)	0.5933 (± 0.0106)	0.5834 (± 0.0080)
DDM	0.9417 (± 0.0111)	0.8643 (± 0.0102)	0.7671 (± 0.0100)	0.9221 (± 0.0073)	0.9027 (± 0.0131)	0.8925 (± 0.0110)
EDDM	0.9471 (± 0.0023)	0.8621 (± 0.0036)	0.7617 (± 0.0038)	0.9283 (± 0.0064)	0.9109 (± 0.0095)	0.9039 (± 0.0068)
BD^3	**0.9496** (± 0.0024)	**0.8717** (± 0.0039)	**0.7750** (± 0.0057)	**0.9315** (± 0.0061)	**0.9133** (± 0.0085)	**0.9053** (± 0.0058)

Table 3: Results on the Elec2 dataset.

	No Detector	DDM	EDDM	BD^3
Accuracy	0.7270	0.7232	0.7243	**0.7307**

6 Conclusion

In this paper, we have shown a novel drift detection method that monitors the classifier error via a beta distribution. Change in the classifier's performance is detected as drift, the sensitivity of the detector can be adjusted via a confidence threshold and decay parameters. Existing knowledge about the classifier's performance can be used to set the initial parameters of the distribution, which allows immediate drift detection. Experimental results show that the method is robust against false positives, while also being fast in detecting concept drift.

References

[1] Albert Bifet and Ricard Gavaldà. Learning from Time-Changing Data with Adaptive Windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining – SDM'07*, pages 443–448, 2007.

[2] João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with Drift Detection. In *Proceedings of the 2004 Brazilian Symposium on Artificial Intelligence – SBIA'04*, pages 286–295, 2004.

[3] Manuel Baena-Garcia, Jose Del Campo-Avila, Raul Fidalgo, Albert Bifet, Ricard Gavalda, and Rafael Morales-Bueno. Early Drift Detection Method. In *Proceedings of the 4th ECML PKDD International Workshop on Knowledge Discovery from Data Streams*, pages 77–86, 2006.

[4] Isvani Frías-Blanco, José Del Campo-Ávila, Gonzalo Ramos-Jiménez, Rafael Morales-Bueno, Agustín Ortiz-Díaz, and Yailé Caballero-Mota. Online and Non-parametric Drift Detection Methods Based on Hoeffding's Bounds. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):810–823, 2015.

[5] W. Nick Street and YongSeog 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 – KDD'01*, pages 377–382, 2001.

[6] Michael Harries. SPLICE-2 Comparative Evaluation: Electricity Pricing. Technical report, The University of South Wales, 1999.

[7] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A Survey on Concept Drift Adaptation. *ACM Computing Surveys*, 46(4):44, 2014.