ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Unsupervised Drift Detection Using Quadtree Spatial Mapping

Bernardo A Ramos[1] and Cristiano L Castro[1,2]
and Tiago A Coelho[3] and Plamen P Angelov[2] *

1- Graduate Program in Electrical Engineering
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

2- School of Computing and Communications
Lancaster University, Lancaster, United Kingdom

3- Universidade Estadual de Feira de Santana, Brazil

Abstract. This paper presents an unsupervised and model-independent concept drift detector based on quadtree spatial analysis (QTS). We used a d-dimensional quadtree to map the feature space and tracked a univariate curve that mimics the spatial behavior of the data stream. This curve serves as a helpful visual tool for analyzing concept drifts. Drifts are identified when there is a significant change in the current spatial mapping. Experimental results show that the proposed outperformed well-known drift detectors in terms of average *precision* and *F1-score*.

1 Introduction

IoT-connected devices and real-time sensors have significantly increased data volume, posing challenges for machine learning, including constraints on storage capacity and the problems inherent to non-stationary data streams.

In data stream literature, changes in data distribution over time due to evolving generator functions are called *concept drifts*, As new samples from different distributions are introduced, machine learning models trained on historical data tend to degrade. Therefore, a concept drift detection mechanism must update the model based on the most recent data.

Regarding concept drift, approaches can be divided into supervised, semi-supervised, and unsupervised detectors. Supervised detectors assume the availability of ground truth labels at the same time as arriving instances. However, this may be unrealistic in real-world applications due to data acquisition failures and latency. As a result, unsupervised alternatives for drift detection tasks have been explored due to these limitations [1, 2].

Unsupervised detector methods must learn from the current raw data and provide the basis to detect distribution changes in the input features. Some detectors rely on data stream statistical properties of first and second order, compute density clusters, and analyze their differences through time [3, 4]. While others build models and track their behavior on newer and older data [5, 6, 7]. In any case, sliding windows and statistical tests are common.

This paper presents a new unsupervised drift detection method named QTS. Even though the concept of monitoring data distribution in space is not new,

*This study was financed in part by CAPES-Brazil - Finance Code 001

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

QTS uses a quadtree to monitor data stream behavior in the mapping space. This allows multivariate data to be analyzed as a univariate curve, providing a visual tool for inspecting data stream changes. This is particularly useful for applications where we don't know the drift characteristics in advance (recurrence, behavior, and velocity). Concept drifts are detected when significant changes occur in the amount of mapped data. Experimental evaluation results show that the proposed method has higher average *precision* and *F1-score* than well-known drift detectors on synthetic and real-world datasets.

2 Unsupervised Quadtree-based Drift Detector (QTS)

QTS analyzes multivariate data distributions from a univariate curve that reflects the spatial occupation of data over time. This is achieved by mapping the data to a height-limited quadtree. Such mapping reduces the cost of storing multivariate data since the streaming is summarized. It also creates a visual tool for inspecting data stream changes. The algorithm for mapping streaming data into a multidimensional quadtree is described in detail in our previous study [4].

The storage capacity of a quadtree depends on the height parameter h and the dimension d. Since h is related to the number of recursive divisions, a limit on this parameter can lead to multiple data points in a child hypercube, breaking the rule of having only one data point per leaf node. To address this, we summarize the data points by taking the mean feature vector of the data points within each child hypercube [4]. The summarization process can be seen in Fig. 1.

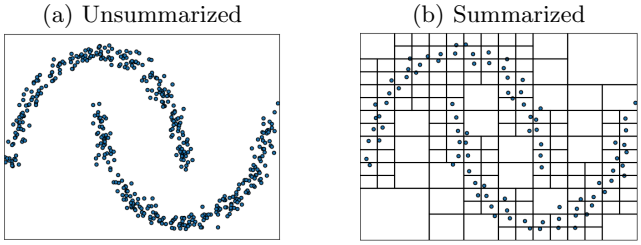

Fig. 1: Half Moons with 500 samples (a) and Quadtree ($h = 4$) with 90 summarized samples (b)

QTS uses a sliding window W to store the most recent data stream samples. Once these samples are inserted into a quadtree, they are summarized, and the amount of data resulting from the summarization is placed in a data structure S_w. The occupancy of S_w is defined as the amount of data occupying the summarized quadtree leaf nodes. As a new sample arrives, the quadtree and the occupancy value for the next S_w are updated. This enables the creation of an occupancy curve over time that reflects the data stream behavior in the mapping space. Fig. 2 shows the corresponding occupancy curve for the well-known benchmark *Electricity* dataset, which has 8 attributes and 45312 samples. It

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

is possible to observe the behavior of occupancy over time showing fluctuations within the same concept and a significant variation between different concepts.

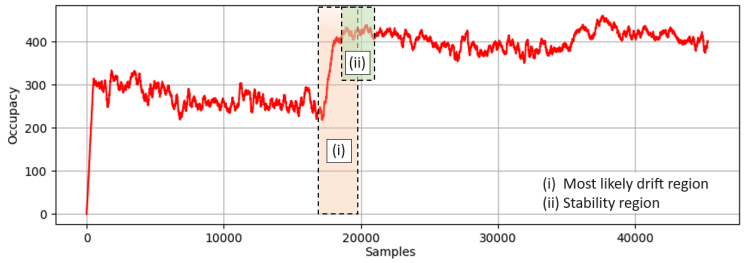

Fig. 2: Occupancy curve over time for Electricity dataset.

Upon the introduction of a new concept, the tree occupancy values change until they stabilize around an average value (see Fig. 2). These values are stored in a static reference window R_w and serve as a reference for the concept. Detecting a concept drift involves identifying significant differences in the quadtree structure. This is done by monitoring the occupancy curve using two parallel strategies, which can be triggered individually depending on variations in the curve:

- this strategy checks whether the most recent occupancy values, stored in S_w, have changed regarding the R_w reference values for a given concept. The reference mean (μ_{Rw}) and standard deviation (σ_{Rw}) of occupancy can be computed from the static window R_w. Likewise, a value representing the actual occupancy can be achieved from the mean (μ_{Sw}) of the S_w values. Thus, inspired by the Chebyshev inequality, a concept drift is alarmed when $\|\mu_{Sw} - \mu_{Rw}\| > 3\sigma_{Rw}$.

- this strategy uses the first derivative of the occupancy curve in S_w to identify a concept drift. A sliding window is set to keep track of the first derivative of the occupancy ($f'(S_w)$). A drift occurs when the distance between mean derivatives on more recent (S_{wRD}) and older data (S_{wOD}) exceeds a threshold: $\left\|\mu_{f'(S_w)_{RD}} - \mu_{f'(S_w)_{OD}}\right\| > \sigma_{f'(S_w)_{OD}}$. As it can be observed, the recommended threshold corresponds to one standard deviation with respect to the mean derivative for the oldest data.

The sizes of R_w, S_w, and W control the detection sensitivity of the QTS method. Using a smaller window size increases detection sensitivity but also raises the risk of false positives. Based on initial experiments, we recommend a window size of 1000 samples for most cases. For the $f'(S_w)$ we recommend using the same size as R_w. The recent data used to estimate the average for comparison with the derivative of old data must be at least 10% of the size of the sliding window $f'(S_w)$.

Fig. 3 illustrates the QTS method in two main stages: (i) a sliding window W stores the most recent data in the first step. A quadtree summarizes the data from W, and the amount of data resulting from the summarization (occupancy) is stored in S_w (Fig. 3a); (ii) the S_w values are processed for drift detection.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

A derivative sliding window $f'(S_w)$ is calculated on S_w data (Fig. 3b), and a static reference window R_w is kept at the beginning of each new concept (Fig. 3c). The pseudocode for QTS algorithm can be found in the following link[1].

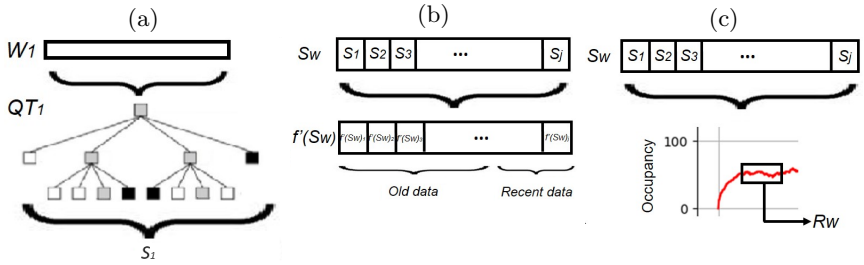

Fig. 3: Overview of the QTS drift detection method.

3 Experiments and Results

To evaluate the effectiveness of the proposed method, QTS was compared with three known unsupervised drift detection methods: D3 [2], OCDD [6], and STUDD [7]. The experiments were performed over 9 synthetic datasets, 7 drawn from MOA [2], and 4 real-world datasets, that have been commonly used in earlier studies on concept drift [3].

The evaluation methodology uses the timestamps in which the drifts occurred to measure the detectors' ability to distinguish between true detection and false alarms. As recommended in [8], we define a True Positive (TP) for every timestamp following a detection within a fixed delay range of 1000 timestamps after a concept drift occurred and a False Negative (FN) for missing a detection for every timestamp within the same delay range; a False Positive (FP) as a detection outside this range or an extra detection in the range. Detection quality is then evaluated by $\textit{recall} = \#TP/(\#TP + \#FN)$, $\textit{precision} = \#TP/(\#TP + \#FP)$, and $\textit{F1-score} = 2 * (\textit{precision} * \textit{recall})/(\textit{precision} + \textit{recall})$ metrics.

Evaluation for real-world datasets involves a pipeline that integrates the drift detector with a classification model to measure the accuracy metric over time. The model used in all cases was a Random Forest Classifier with default parameters, trained with an initial set of samples and updated whenever a concept drift was detected. The process involves discarding the current classifier and training a new model with a new set of samples. The methods are then compared using two baseline models, with no integrated drift detectors: BL1 and BL2. In BL1, the classifier is trained only once and is not updated further. In BL2, the classifier is retrained after every K sample, where the value of K is modified for each dataset so that the number of retraining matches the number of QTS detections.

[1] https://github.com/beraram/QTS-method
[2] available at: `https://moa.cms.waikato.ac.nz/`

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Table 1 shows the results achieved on the synthetic datasets: *Recall*, *Precision*, *F1-score*, *#Detections* and the amount of truly detected *Drifts*. The best scores for each dataset are highlighted in bold. The QTS outperforms other methods in most datasets, having the best averages for every metric. It detected all the concept drifts with a high *precision* and a good *F1-score*.

Dataset	Method	Recall	Precision	F1-Score	#Detections	Drifts
TOY	QTS	**0.8443**	**0.9978**	**0.9147**	4	3/3
	STUDD	0	0	0	1	0/3
	D3	0.396	0.9905	0.5658	4	2/3
	OCDD	0	0	0	3	0/3
RBF	QTS	0.331	1	0.4974	1	1/1
	STUDD	0.28	1	0.4375	1	1/1
	D3	**0.899**	**1**	**0.9468**	1	1/1
	OCDD	0.305	0.7274	0.4298	7	1/1
SEA	QTS	0.228	**1**	0.3713	1	1/1
	STUDD	0	0	0	0	0/1
	D3	0	0	0	0	0/1
	OCDD	**0.237**	0.8976	**0.375**	4	1/1
DS_RS	QTS	**0.994**	**1**	**0.997**	1	1/1
	STUDD	0	0	0	0	0/1
	D3	0.899	0.8727	0.9468	1	1/1
	OCDD	0.869	0.8727	0.8709	15	1/1
DS_RS_G	QTS	**0.861**	**1**	**0.9253**	1	1/1
	STUDD	0	0	0	0	0/1
	D3	0	0	0	0	0/1
	OCDD	0.749	0.8734	0.8064	13	1/1
DS_HR	QTS	**0.989**	**1**	**0.9945**	1	1/1
	STUDD	0	0	0	1	0/1
	D3	0	0	0	0	0/1
	OCDD	0.6867	0.8734	0.7689	12	1/1
DS_HR_G	QTS	**0.895**	**1**	**0.9446**	1	1/1
	STUDD	0.2851	1	0.4437	1	1/1
	D3	0	0	0	0	0/1
	OCDD	0.537	0.8438	0.6563	12	1/1
DS_HS	QTS	**0.999**	**1**	**0.9995**	1	1/1
	STUDD	0	0	0	0	0/1
	D3	0.899	0.8727	0.9468	1	1/1
	OCDD	0.799	0.8717	0.8338	14	1/1
DS_SR_G	QTS	**0.8372**	0.9993	**0.9111**	3	1/1
	STUDD	0.1747	**1**	0.2974	1	1/1
	D3	0	0	0	0	0/1
	OCDD	0.7091	0.9938	0.8277	17	1/1
Average	QTS	**0.7754**	**0.9997**	**0.8395**	1.5556	11/11
	STUDD	0.0672	0.2727	0.1071	0.3636	3/11
	D3	0.3437	0.4151	0.3785	0.7778	5/11
	OCDD	0.5435	0.7726	0.6188	10.7778	8/11

Table 1: Results of unsupervised drift detectors over the synthetic datasets.

Table 2 shows the accuracy results on the real-world datasets. As observed, QTS performed well and was competitive with other unsupervised methods. In all cases, QTS was better or equal to BL1 and outperformed BL2 in three of four datasets. Additionally, QTS was better or equal to D3 in all datasets, outperformed OCDD in three out of four datasets, and outperformed STUDD in two datasets, with similar results in cases where it was worse.

4 Conclusions

This paper presented a novel unsupervised drift detector (QTS) that maps the streaming data to a height-limited d-dimensional quadtree. Such mapping reduces the cost of storing multivariate data since the streaming is summarized;

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Dataset	QTS	STUDD	D3	OCDD	BL1	BL2
ELECTRICITY	0.7318	**0.7394**	0.6782	0.7100	0.6612	0.7034
WEATHER	0.7409	**0.7413**	0.7409	0.7235	0.7409	0.7358
AIRLINES	**0.5958**	0.5483	0.5565	0.5733	0.5490	0.5877
POSTURE	0.5291	0.5258	0.4631	0.5329	0.4620	**0.5572**
AVERAGE RANK	2.25	3	4.25	3.5	5	3

Table 2: Accuracy results over the real-world datasets.

it also creates a visual tool for inspecting data stream changes, which is particularly suitable for applications in which we do not know in advance the drift behavior. Two strategies were proposed for identifying changes through the Quadtree occupancy curve: one inspired by the Chebyshev inequality for detecting slower changes and another using the occupancy's mean derivative for more subtle changes. Experimental results on synthetic and real data streams indicated that QTS is more precise in detecting drifts than the other tested methods.

References

[1] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *IEEE TKDE*, 31(12):2346–2363, 2018.

[2] Ö. Gözüaçık, A. Büyükçakır, H. Bonab, and F. Can. Unsupervised concept drift detection with a discriminative classifier. In *Proceedings of the 28th ACM-CIKM*, pages 2365–2368, 2019.

[3] A. Iwashita and J. Papa. An overview on concept drift learning. *IEEE Access*, 7:1532–1547, 2018.

[4] R. Coelho, L. Torres, and C. Castro. Concept drift detection with quadtree-based spatial mapping of streaming data. *Information Sciences*, 625:578–592, 2023.

[5] R. Gemaque, A. Costa, R. Giusti, and E. Dos Santos. An overview of unsupervised drift detection methods. *Data Mining and Knowledge Discovery*, 10(6), 2020.

[6] Ö. Gözüaçık and F. Can. Concept learning using one-class classifiers for implicit drift detection in evolving data streams. *Artificial Intelligence Review*, 54:3725–3747, 2021.

[7] V. Cerqueira, H. Gomes, A. Bifet, and L. Torgo. Studd: A student–teacher method for unsupervised concept drift detection. *Machine Learning*, 2022.

[8] S. Yu, Z. Abraham, H. Wang, M. Shah, Y. Wei, and J. Príncipe. Concept drift detection and adaptation with hierarchical hypothesis testing. *Journal of the Franklin Institute*, 356(5):3187–3215, 2019.