ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1549

Towards Online Concept Drift Detection with Feature Selection for Data Stream Classification

Mahmood Hammoodi [1] and **Frederic Stahl** [2] and **Mark Tennant** [3]

Abstract. Data Streams are unbounded, sequential data instances that are generated very rapidly. The storage, querying and mining of such rapid flows of data is computationally very challenging. Data Stream Mining (DSM) is concerned with the mining of such data streams in real-time using techniques that require only one pass through the data. DSM techniques need to be adaptive to reflect changes of the pattern encoded in the stream (concept drift). The relevance of features for a DSM classification task may change due to concept drifts and this paper describes the first step towards a concept drift detection method with online feature tracking capabilities.

1 INTRODUCTION

Velocity in Big Data Analytics [6] refers to data that is generated at a high speed in real-time and challenges our computational capabilities in terms of storing and processing the data [2]. DSM requires techniques that are incremental, computationally efficient and can adapt to *concept drift* for applications such as real-time analytics of chemical plant data in the chemical process industry [10], intrusion detection in telecommunications [9], etc. A concept drift occurs if the pattern encoded in the data stream changes. DSM has developed various real-time versions of established predictive data mining algorithms that adapt to concept drift and keep the model accurate over time, such as CVFDT [8] and G-eRules [11]. The benefit of classifier independent concept drift detection methods is that it allows providing information about the dynamics of the data generation. Common drift detection methods are for example ADaptive sliding WINdow (ADWIN) [4], Drift Detection Method (DDM) by [7] and the Early Drift Detection Method (EDDM) by [3]. However, to the best of our knowledge, no drift detection method provides insights into which features are involved in the concept drift, which is potentially valuable information. For example, if a feature is contributing to a concept drift it can be assumed that the feature may have become either more or less relevant for the concept encoded in the stream after the drift. This knowledge about a feature's contribution to concept drift could be used to develop an efficient real-time feature selection method that does not require examining the entire feature space for online feature selection. This paper proposes a concept drift detection method for data stream classification that also feeds forward information about the involvement of individual features in the drift for feature selection purposes. The proposed method could be used with any learning algorithms either as a real-time wrapper for a batch classifier or realised inside a real-time adaptive classifier. This paper is organised as follows: Section 2 introduces the proposed concept drift and feature selection method, Section 3 evaluates the methodology briefly as a proof of concept and Section 4 provides concluding remarks.

2 REAL-TIME FEATURE SELECTION USING USING ADAPTIVE MICRO-CLUSTERS

The work presented in this paper is based on the Micro-Cluster structure of the MC-NN classifier [12] developed by one of the authors of this paper. MC-NN Micro-Clusters are an extension of the Micro-Clusters used in the CluStream data stream clustering algorithm [1]. The notation used for a Micro-Cluster has been taken from [1]. Essentially Micro-Clusters in MC-NN aim to keep a recent accurate summary of the data stream. The structure of MC-NN Micro-Clusters is: $< CF2^x, CF1^x, CF1^t, n, CL, \epsilon, \Theta, \alpha, \Omega >$ $CF2^x$ is a vector with the sum of squares of the features; $CF1^x$ a vector with the sum of feature values; $CF1^t$ a vector with the sum of time stamps; n is the number of data instances in the cluster; CL is the cluster's majority class label; ϵ the error count; Θ the error threshold (default 5000) for splitting the Micro-Cluster; α is the initial time stamp and Ω a threshold for the Micro-Cluster's performance (default 50). The centroid of a Micro-Cluster is calculated by $\frac{CF1^x}{n}$.

Loosely speaking MC-NN updates Micro-Clusters by adding a new instance to its nearest Micro-Cluster if it matches the CL it decrements the error ϵ by 1. Otherwise it adds the data instances to its nearest Micro-Cluster that matches the CL but increases the error count ϵ of both involved Micro-Clusters by 1. If a Micro-Cluster's error count reaches Θ it splits into to new Micro-Clusters placed about the original Micro-Cluster's feature of greatest variance and the original Micro-Cluster is removed in order to fit the data stream better. The variance for a feature x can be calculated by $Variance[x] = \sqrt{\left(\frac{CF2^x}{n}\right) - \left(\frac{CF1^x}{n}\right)^2}$, the assumption is that the larger the variance, the greater the range of values that have been seen for this feature and thus is may contribute to misclassification. New Micro-Clusters are generated with the old Micro-Clusters' centroid values. The centroids values of the 2 new Micro-Clusters, for the attribute that has the largest variance is 'altered' by either adding or subtracting the variance amount (adding in one Micro-Cluster, subtracting in the other). The participation of the cluster on absorbing instances is monitored over time and if a cluster has not participated recently in classifications it is removed. The clusters' participation is measured with the *Triangle Number* $\Delta(T) = ((T^2 + T)/2)$ which can be calculated from $CF1^t$. The lower $CF1^t$ the lower the participation of the Micro-Cluster, but the triangle number gives more importance to recent instances than older ones. In order to detect a concept drift, we track the total number of Micro-Cluster splits and

[1] University of Reading, United Kingdom, email: m.s.h.hammoodi@pgr.reading.ac.uk

[2] University of Reading, United Kingdom, email: F.T.Stahl@reading.ac.uk

[3] University of Reading, United Kingdom, email: m.tennant@pgr.reading.ac.uk

removals. The assumption is that the larger either or both numbers are, the more likely it is that a concept drift happened.

Using a windowing approach upon the data stream, a running average of the split and death rates are calculated concurrently. If the percentage of the split and death rates differs from the mean (of the statistical windows) by 50% (default value), this is consider as a concept drift. Then a closer look can be taken into the individual features through examining their change in velocity. This is tracked through an extension of MC-NN's Micro-Cluster structure by: $< CF1^{hx}, n_h >$. Where the components of the structure above are equivalent to $CF1^x$ and n. However, the h denotes that these components are *historical* summaries (taken from the statistical windows); the value of h is a fixed user defined parameter and denotes how many time stamps the historical summaries are behind the recent ones (default 10,000). The *velocity* of a feature x can then be calculated by $V_x = \frac{CF1^x}{n} - \frac{CF1^{hx}}{n_h}$. A high velocity during a concept drift indicates that the feature changed. The assumption here is that this particular feature may have changed its contribution towards the classification technique, whereas the remaining ones have not. Thus feature selection can be limited to examining only features that have changed their velocity where there is a concept drift detected. Section 3 evaluates this approach as a principal proof of concept.

3 EXPERIMENTAL EVALUATION

This section aims to show that the proposed methodology can identify concept drift in a data stream and at the same time detect which features were involved. Random Tree data stream generator, which was introduced in [5] and generates a stream based on a randomly generated tree, was used for a proof of concept. New examples are generated by assigning uniformly distributed random values to features, which then determine the class label using the randomly generated tree.

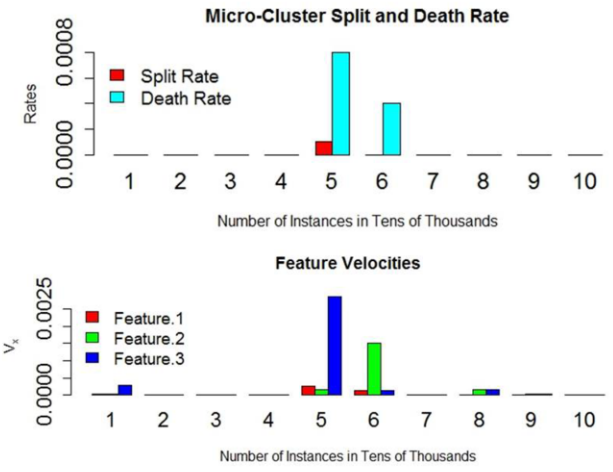

Figure 1. The top of the figure shows the Micro-Cluster split and death rate and the bottom of the figure shows the feature velocities.

We generated a stream of 100,000 instances, 3 features and 2 classification labels. Features 1 and 2 are relevant to determine the class label and feature 3 is random. After 50,000 instances features 2 and 3 were swapped making feature 2 irrelevant and feature 3 relevant for classification tasks. We noticed that the method accurately detected a drift at 50,000 instances as Micro-Clusters were reset. For the experiment the default parameters stated in Section 2 of the method were used unless stated otherwise. In the top half of Figure 1 it can be seen that the split and death rates at the time of concept drift increase, indicating that the current set of Micro-Clusters does not fit the concept encoded in the data anymore. The bottom of Figure 1 shows the velocities of the features in the Micro-Clusters. In this case we know that the concept appeared due to swapping features 2 and 3, hence we would expect a higher velocity of these two features. Figure 1 shows this change in velocity. Thus the method is capable to detect a concept drift but also delivers an indication which features are involved, which can be used to perform online real-time feature selection.

4 CONCLUSIONS

This paper introduced a novel Micro-Cluster based methodology for drift detection in data streams. Different compared with existing drift detection techniques, the proposed method is also capable to detect which features have been involved in the drift through the velocity of Micro-Clusters in different dimensions; and thus can be used to implement real-time feature selection techniques. The experimental proof of concept shows that the methods can successfully detected concept drifts and identify drifting features. Ongoing and future work comprises an in depth evaluation of the method and the development of a real-time feature selection technique.

REFERENCES

[1] C. Aggarwal, J. Han, J. Wang, and P.Yu, 'A framework for clustering evolving data streams', in *Proceedings of the 29th VLDB Conference*, Berlin Germany, (2003).

[2] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom, 'Models and issues in data stream systems', in *In PODS*, pp. 1–16, (2002).

[3] Manuel Baena-Garcıa, José del Campo-Ávila, Raúl Fidalgo, Albert Bifet, R Gavalda, and R Morales-Bueno, 'Early drift detection method', in *Fourth international workshop on knowledge discovery from data streams*, volume 6, pp. 77–86, (2006).

[4] Albert Bifet and Ricard Gavald, 'Learning from time-changing data with adaptive windowing', in *In SIAM International Conference on Data Mining*, pp. 443–448.

[5] Pedro Domingos and Geoff Hulten, 'Mining high-speed data streams', *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '00*, 71–80, (2000).

[6] M Ebbers, A Abdel-Gayed, V Budhi, and F Dolot, *Addressing Data Volume, Velocity, and Variety with IBM InfoSphere Streams V3.0*, IBM Redbooks, 2013.

[7] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues, 'Learning with drift detection', in *Advances in artificial intelligence–SBIA 2004*, 286–295, Springer, (2004).

[8] Geoff Hulten, Laurie Spencer, and Pedro Domingos, 'Mining Time-Changing Data Streams', in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, volume 18, pp. 97–106. ACM, (2001).

[9] A. Jadhav, A. Jadhav, P. Jadhav, and P. Kulkarni, 'A novel approach for the design of network intrusion detection system(NIDS)', in *Sensor Network Security Technology and Privacy Communication System (SNS PCS), 2013 International Conference on*, pp. 22–27, (May 2013).

[10] Petr Kadlec, Bogdan Gabrys, and Sibylle Strandt, 'Data-driven Soft Sensors in the process industry', *Computers and Chemical Engineering*, **33**(4), 795–814, (2009).

[11] Thien Le, Frederic Stahl, João Bártolo Gomes, Mohamed Medhat Gaber, and Giuseppe Di Fatta, 'Computationally Efficient Rule-Based Classification for Continuous Streaming Data', in *Research and Development in Intelligent Systems XXIV*, p. 2014, (2008).

[12] Mark Tennant, Frederic Stahl, and João Bártolo Gomes, 'Fast adaptive real-time classification for data streams with concept drift', in *Internet and Distributed Computing Systems*, 265–272, Springer, (2015).