A cluster-based prototype reduction for online classification *

Kemilly Dearo Garcia[1,2], André C.P.L.F. de Carvalho[2][0000−0002−4765−6459], and João Mendes-Moreira[3,4][0000−0002−2471−2833]

[1] University of Twente, Netherlands, kemilly.dearo@usp.br
[2] ICMC, University of São Paulo, Brazil, andre@icmc.usp.br
[3] Faculty of Engineering, University of Porto, Portugal, jmoreira@fe.up.pt
[4] LIAAD-INESC TEC, Porto, Portugal

Abstract. Data stream is a challenging research topic in which data can continuously arrive with a probability distribution that may change over time. Depending on the changes in the data distribution, different phenomena can occur, for example, a concept drift. A concept drift occurs when the concepts associated with a dataset change when new data arrive. This paper proposes a new method based on k-Nearest Neighbors that implements a sliding window requiring less instances stored for training than existing methods. For such, a clustering approach is used to summarize data by placing labeled instances considered similar in the same cluster. Besides, instances close to the uncertainty border of existing classes are also stored, in a sliding window, to adapt the model to concept drift. The proposed method is experimentally compared with state-of-the-art classifiers from the data stream literature, regarding accuracy and processing time. According to the experimental results, the proposed method has better accuracy and less time consumption when fewer information about the concepts are stored in a single sliding window.

Keywords: kNN Prototyping · Data Stream · Online Clustering.

1 Introduction

In real world data analysis, data can continuously arrive in streams, with a probability distribution that can change over time. These data are known as data streams. Depending on the changes in the data distribution, different phenomena can occur, like concept drift [8]. In these situations, it is important to adapt the classification model to the current stream, otherwise its predictive performance can decrease over time.

* This work was partially funded by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project POCI-01-0145-FEDER-016883.

Several algorithms proposed for data stream mining are based on online learning [7, 4, 8, 9]. Some of them are based on the kNN (k-Nearest Neighbor) algorithm. In the data stream mining, the kNN algorithm maintains a sliding window with a certain amount of labeled data, which is used as its training data.

Other algorithms from the literature deal with concept drift by explicitly detecting changes in parts of the stream, comparing the current concept with previous concepts from time to time [9]. Some of them continuously calculate the model classification error. For such, they assume that the label of the data arriving in the stream is available.

However, there is a cost associated with the data labeling process that can become prohibitive or unfeasible when data arrive in high speed or volume. In online classification, the labeling of incoming instances can have a high cost [11]. The lack of the label make the measure of classification error a difficult task.

Despite its simplicity, kNN has been largely used in literature, because it is nonparametric, favoring its use in scenarios with few available information and with known concepts changing over time [4]. However, the use of a sliding window may ignore instances with relevant information about persistent concepts. Furthermore, the size of a sliding window affects its efficient use.

This article proposes SWC (Sliding Window Clusters), a method based on kNN that implements a sliding window whose number of instances stored can be reduced. SWC summarizes data streams by creating a set of clusters, each one representing similar labeled instances. Instances close to decision border of each cluster are also stored, so they can be used to adapt the model to concept drift.

An experimental evaluation shows that SWC can increase the predictive performance and reduce both computational and time consumption than related methods based on kNN and sliding window.

This paper is structured as follows. Section 2, presents previous related works using kNN and sliding window. Data stream and concept drift are introduced in 3. The proposed method is described in Section 4. Sections 5 presents the experimental setup and analyses the results obtained. Finally, Section 6 has main conclusions and points out future work directions.

2 Related Work

This section briefly presents previous works using kNN for data stream classification with concept drift. These works use variations of sliding window to store the training instances.

One alternative of online learning is to randomly select instances to maintain or discard in the sliding window. This is the case of the method PAW (Probabilistic Approximate Window) method [4], a probabilistic measure used to decide which instance will be discarded from the sliding window when a new instance arrive. Thus, the size of the window is variant and represents a mix of outdated and recent relevant instances. The $k\text{NN}_W$ method combines the PAW method couplet with the kNN classifier.

Another related method, ADWIN (ADaptive sliding WINdowing) [2], is a concept drift tracker able to monitor changes in data streams. The algorithm automatically grows the sliding window when no change is detected in the stream. When a change is detected, the algorithm shrinks the sliding window and forgets the sub-window that is outdated. In the combination of kNN with PAW and ADWIN [4], $k\text{NN}_{WA}$, ADWIN is used to keep only the data related to the most recent concept from the stream, the rest of instances are discarded.

A deficiency of updating instances using a sliding window is the possibility to forget old but relevant information. To avoid losing relevant information, another method, named SAM (Self Adjusting Memory) [9], to adapt a model to concept drift by explicitly separating current and past information. SAM uses two memory structures to store information, one based on short-term-memory and the other on long-term-memory. The short-term-memory contains data associated with the current concept and the long-term-memory maintains knowledge (old models) from past concepts. SAM is couplet with a kNN classifier.

The implementations and variations of kNN for data stream mining are available in the MOA framework. Due to memory and computational limitations, the implementations use a fixed size window of 1000 labeled instances.

3 Problem Formalization

A possible unbounded amount of data can sequentially arrive in a data stream. These data can often undergo chances in their distribution over time, which may require the adaptation of the current model context [8].

Formally, a data stream is a sequence of instances, potentially infinity that can be represented by [7]:

$$D_{tr} = \{(X_1, y_1), (X_2, y_2), ..., (X_{tr}, y_{tr})\}$$

where X_{tr} is an instance arriving in time tr and y_{tr} is the target class. Each instance needs to be processed only once due to finite resources.

Concept drift is a change in the distribution probability of target classes [8]. Formally, a distribution P in a given time tr conditioned by the instance X and label y can suffer changes affecting the conditional probability $P_{tr+1}(X, y)$. As a result, a model built during time tr could be outdated in time $tr + 1$.

$$X : P_{tr}(X, y) \neq P_{tr+1}(X, y)$$

4 Methodology

In data stream mining, an ideal classifier should be able to learn the current concept in feasible time without forgetting relevant past information [4].

The proposed method is described in Algorithm 1. Instead of storing all instances that fit in a sliding window (for representing both old and current concepts), SWC stores compressed information about concepts and instances

Algorithm 1 SWC: Online Window Update

input: X_{tr}, W, T, ρ
output: W
rand $\leftarrow$ *random(0, 1)*
if *rand* $\leq \rho$ **then**
  for all w in W **do**
    Let X_{tr} be the nearest to w ($w \in W$)
    $dist \leftarrow EuclidianDistance(X_{tr}, w)$
    if $dist < w_{radius}$ **then**
      $W \leftarrow UpdateCluster(X_{tr}, w)$
    else
      if $dist \leq T$ **then**
        $W \leftarrow W \cup X_{tr}$
return W

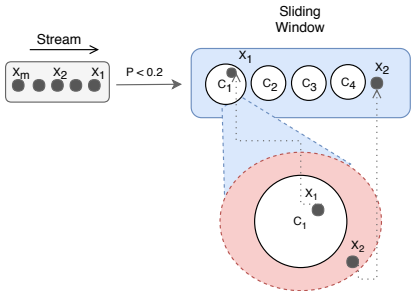

Fig. 1: Instances X_1 and X_2 are stored within the sliding window. The first instance, X_1, is closer to cluster C_1 and inside its radius. The second instance, X_2, is outside cluster area, but close to the uncertainty border.

close to uncertainty border of each class. As the previous methods, SWC is combined with the kNN classifier in the MOA framework [3].

A more detailed description of how SWC works is presented next. Initially, all instances arriving from the stream are stored in the form of clusters. The clusters are created using the CluStream algorithm [1]. A constrain implying that each cluster must contain only instances from the same class was included.

As data arrive in the stream, a parameter based on probability, ρ, is used to decide if a new instance, X_{tr}, will be incorporated to the model W. If X_{tr} is inside a radius from a existing cluster, the instance is incorporated to this cluster. However, if X_{tr} is outside, but is close to a uncertainty border, X_{tr} is incorporated to the model alone, outside the existing clusters. For such, the uncertainty border is defined as the area outside the radius of a cluster, but inside a given threshold.

As is illustrated in Figure 1, if the instance, X_1, is inside the radius of the closest cluster, then it will be incorporated to the existing cluster, however if the instance, X_2, is closer to a uncertainty border, it is stored alone.

It must be observed that not all instances in the stream are included into the sliding window. For each instance arriving in the stream, SWC randomly decides if the instance will be learned or not. A similar procedure is used in [4], which uses a probability $\rho = 0.5$. SWC uses a lower probability, consider that the learning process can be done with a lower probability of $\rho = 0.2$, without significant predictive performance loss, but with a lower processing cost.

5 Experimental Evaluation

This section experimentally compares SWC with other methods implemented in the MOA framework that use kNN with sliding window, namely kNN, $k\text{NN}_W$, $k\text{NN}_{WA}$ and SAM. The experimental evaluated used was Interleaved Test-Train to incremental learning [4].

5.1 Datasets

Table 1 describes the datasets used in the experiments. Before the streaming, in a offline phase, all methods started with a batch of labeled data representing 10% of the each dataset. The remaining data arrived in the stream. Real and artificial datasets were used.

Table 1: Characteristics of datasets evaluated.

Datasets	Samples	Features	Class
SEA	5.000 / 50.000 (total)	3	2
Mixed Drift	60.000 / 600.000 (total)	2	15
Rotating Hyperplane	20.000 / 200.000 (total)	10	2
Forest Cover Type	58.101 / 581.012 (total)	54	7
Airlines	53.938 / 539.383 (total)	4	2
Moving RBF	20.000 / 200.000 (total)	10	5

Artificial Datasets

The SEA Concepts Dataset [10] has four concepts. A concept drift occurs at each 15.000 instances, with different thresholds for the concept.

The Rotating Hyperplane dataset is based on a hyperplane of d-dimensional space which is continuously changing in position and orientation. It is available in the MOA framework and was used in [4, 9].

Moving RBF is a dataset, generated by MOA framework, based on Gaussian distributions with random initial positions, weights and standard deviations. Over time, this Gaussian distributions suffer changes. This dataset is used by [4, 9].

Mixed Drift [4, 9] is a mix of tree datasets: Interchanging RBF, Moving Squares and Transient Chessboard. Data from each dataset are alternatively presented in the stream.

Real World Datasets

The Forest Cover Type [6] data set is is a well known benchmark for the evaluation of algorithms for data stream mining, being constantly used to validate proposed methods [9, 4, 11].

The Airlines dataset has data from US flight control [11]. It has two classes, one indicating that a flight will be delayed, and the other that the flight will arrive on time.

5.2 Results and discussion

The proposed method, SWC, is compared with the methods kNN, $k\text{NN}_W$, $k\text{NN}_{WA}$ and SAM. For all methods, one nearest neighbor ($k = 1$) is adopted. The remaining parameters use default values, including a fixed window size ($w = 1000$).

The ρ parameter, chance of updating the model, in the SWC method is defined for an acceptable trade-off between accuracy and time cost. A parameter of threshold $T = 1.1$, uncertainty border, is also defined for each cluster. The threshold is multiplied by the radius of each cluster and indicates how much the cluster can expand. Both parameters were explained in Section 4.

Experiments were performed to decide the value of ρ and for SWC. Figure 2 shows that there is an increase of accuracy with $\rho = 0.5$, meaning that a instance has 50% of chance to be learned by the model. However, the selected value was $\rho = 0.2$, which results in a better balance between accuracy and time cost.

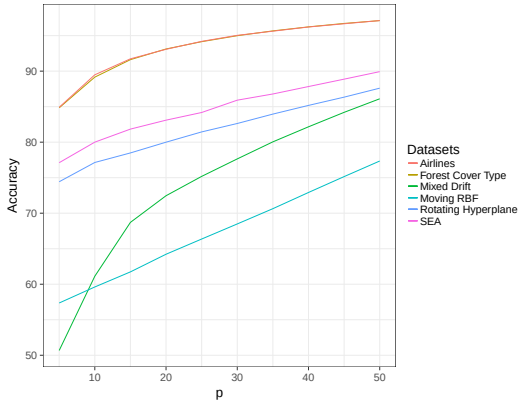

Fig. 2: SWC accuracy performance of all datasets varying ρ value in 5% to 50%.

Table 2 shows the average accuracy and total time cost. It must be observed that accuracy is the measure of instances correctly classified over test/train interleaved evaluation [4].

The results shows that SWC is competitive with state-of-the-art SAM and is considerably faster. The method baseline kNN presented the worst performance, which was expected, once it does not learn over time. However, it is a good baseline to measure how much time each other method take to learn new instances.

Both methods $k\text{NN}_W$ and $k\text{NN}_{WA}$ present similar accuracy rates. However, $k\text{NN}_{WA}$ has a higher cost, due the use of ADWIN.

Finally, although SAM and SWC obtained predictive accuracies similar to SWC, for some cases, SWC was better. Besides, SWC is faster, due to the use of only one sliding window with compressed concepts and relevant instances.

Table 2: Accuracy and time cost (in seconds) for each method.

Dataset	kNN	$k\text{NN}_W$	$k\text{NN}_{WA}$	SAM	SWC
SEA	77.24 (6)	77.25 (9)	77.25 (10)	80.53 (18)	83.49 (8)
Mixed Drift	16.82 (68)	53.62 (94)	53.62 (102)	80.53 (2724)	72.46 (73)
Rotating Hyperplane	50.00 (63)	66.42 (88)	68.42 (91)	70.27 (318)	80.17 (70)
Forest Cover Type	23.46 (614)	54.56 (898)	55.56 (1024)	89.84 (3422)	93.12 (394)
Airlines	54.37 (91)	52.53 (163)	52.53 (146)	88.37 (1530)	93.07 (120)
Moving RBF	26.07 (62)	59.98 (89)	59.97 (100)	69.92 (1788)	64.22 (71)

To assess their statistical significance, a Friedman rank sum test combined with Nemenyi post-hoc test [5], both with a significance level of 5%, was applied to the experimental results. A $p - value = 0.000441$ was obtained in the Friedman test, showing a significant difference between the five methods. Additionally, the Nemenyi post-hoc test, Table 3, showed meaningful statistical differences between the following pair of methods: SWC$\succ k$NN. There is no significant difference between all remaining pairs. However we emphasize that SWC$\succ k\text{NN}_W$ and SWC$\succ k\text{NN}_{WA}$ have relatively low p-values (less than 10%).

Table 3: P-values obtained for the multiple comparison post-hoc Nemenyi test.

	kNN	$k\text{NN}_W$	$k\text{NN}_{WA}$	SAM
$k\text{NN}_W$	0.8536	-	-	-
$k\text{NN}_{WA}$	0.7591	0.9998	-	-
SAM	0.0090	0.1506	0.2201	-
SWC	0.0024	0.0621	0.0987	0.9962

6 Conclusion and Future Work

This paper presented a new method, SWC, based on k-Nearest Neighbors that implements a sliding window that stores less training instances than related

methods. SWC stores in a sliding window clusters and instances close to uncertainty border of each class. The clusters are compressed stable concepts and the instances are possible drifts of these concepts.

Considering accuracy performance, time and storage cost, SWC was experimentally compared with state-of-the-art related methods. According to the experimental results SWC presented higher predictive performance, with lower processing and memory cost than the compared methods.

As future work, the authors want to distinguish concept drift from novelty detection and study an efficient alternative to discard outdated information. Besides, they intend to include an unsupervised concept drift tracker.

References

1. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: VLDB 2003, Proceedings of 29th International Conference on Very Large Data Bases, September 9-12, 2003, Berlin, Germany. pp. 81–92 (2003)
2. Bifet, A., Gavaldà, R.: Learning from time-changing data with adaptive windowing. In: Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA. pp. 443–448 (2007)
3. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: massive online analysis. Journal of Machine Learning Research **11**, 1601–1604 (2010)
4. Bifet, A., Pfahringer, B., Read, J., Holmes, G.: Efficient data stream classification via probabilistic adaptive windows. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, Coimbra, Portugal, March 18-22, 2013. pp. 801–806 (2013)
5. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7**, 1–30 (2006)
6. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
7. Faria, E.R., Gama, J., Carvalho, A.C.P.L.F.: Novelty detection algorithm for data streams multi-class problems. In: Proceedings of the 28^{th} Annual ACM Symposium on Applied Computing, SAC '13, Coimbra, Portugal, March 18-22, 2013. pp. 795–800 (2013)
8. Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Comput. Surv. **46**(4), 44:1–44:37 (2014). https://doi.org/10.1145/2523813, http://doi.acm.org/10.1145/2523813
9. Losing, V., Hammer, B., Wersing, H.: KNN classifier with self adjusting memory for heterogeneous concept drift. In: IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain. pp. 291–300 (2016), https://doi.org/10.1109/ICDM.2016.0040
10. Street, W.N., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001. pp. 377–382 (2001)
11. Zliobaite, I., Bifet, A., Pfahringer, B., Holmes, G.: Active learning with drifting streaming data. IEEE Trans. Neural Netw. Learning Syst. **25**(1), 27–39 (2014)