Online Learning with Reoccurring Drifts: The Perspective of Case-Based Reasoning

Marie Al-Ghossein[1⋆], Pierre-Alexandre Murena[1,2⋆], Antoine Cornuéjols[2], and Talel Abdessalem[1,3]

[1] LTCI - Télécom ParisTech, Paris, France
[2] UMR MIA518 - AgroParisTech INRA, Paris, France
[3] UMI CNRS IPAL NUS

Abstract. While machine learning usually focuses on learning one single concept from a batch of instances, the development of new media for data acquisition has led to the emergence of data streams. In such streams, the data distribution can change over time, and in particular previous states can reoccur. Handling such re-occurrences requires to manage a memory of past states. In this paper, we show that a parallel can be drawn between this task and the framework of case-based reasoning. Based on this parallel, we propose a general methodology and apply it to the problem of online topic modeling.

Keywords: Incremental learning · Concept Drift · Topic Modeling.

1 Introduction

The recent emergence of new sources of data related to the Internet of Things, online platforms or social media, among others, has led to a change in the way data is acquired in machine learning. Traditionally, all learning data is available in a batch and points are supposed to be drawn independently from a single distribution. In these new environments, data arrive in the form of streams that have to be handled online at high rates. One of the challenges raised by data stream mining, among others, concerns *concept drift*, i.e., change in the distributions [28].

Depending on the problem, several types of drifts can be observed [13]. *Abrupt* drifts correspond to a sudden change of distributions, switching from one distribution to another. On the contrary, *incremental* drifts correspond to slight evolutions of the distribution at each time step. *Reoccurring* drifts are particularly frequent and model the reoccurrence of past states (either cyclic or episodic). In general, algorithmic methods for data stream mining are biased toward one class of drifts and behave poorly on other categories. Active methods, such as ADWIN [2], focus on detecting drifts and adapt their models only when a drift is detected: thus, they are more adapted to abrupt drifts. On the contrary, passive methods adapt their model at each step, no matter if a drift actually happened

⋆ Both authors contributed equally.

or not: these methods are particularly efficient for incremental drifts. Dealing with reoccurring drifts requires a bit more adaptation and especially the use of a memory to evaluate the relatedness of the current observation with the past.

Such a use of memory is highly similar to the problems encountered in the domain of Case-based Reasoning (CBR). In particular, the four steps of CBR are observed in memory management for stream mining. Retrieval is implied in the process of detecting similar states in the past (*did the drift lead to a previously encountered distribution?*); Reuse brings a solution to the current case based on the retrieved cases; Revision exploits information of the new case to adapt current cases; Retention evaluates if the new case has to be kept in memory [6]. Exploiting this analogy, we propose to discuss in this paper the correlations between memory-based systems for data stream mining and CBR. We illustrate our reflection with the application of online topic modeling, which consists in evaluating topics of texts in streams [23].

The remainder of this paper is organized as follows. In Section 2, we present a general introduction to online learning as well as a couple of works that already exploit the similarity of online learning and CBR. In Section 3, we present our idea and discuss how each step of CBR can be adapted as a step in online learning. In Section 4, we propose the application of online topic modeling and illustrate our idea with some examples. Finally, we conclude our paper in Section 5 with a discussion of the potential implications and perspectives.

2 Related Work

Online learning and memory. Unlike batch learning, data stream mining requires to use a dynamic memory to store and forget data or concepts. [13] makes the distinction between short term memory, which is filled with current data, and long term memory which stores generalization of data, hence models.

Short term memory captures the current state of the stream at the time of the observation. A first family of methods stores the most recent data in a window, the length of which can be either fixed [28] or variable [12], [21]. Such methods are by nature inefficient for reoccurring concept drifts, since they are motivated by the assumption that most recent data are the best representations of current state. Other solutions are employed to overcome the main drawbacks of windowing strategies. Instance weighting strategies, for instance, keep all data in memory but the examples are weighted depending on their age in the stream [20].

Management of long term memory is the main concern of ensemble approaches. Online ensemble methods are based on the use of a pool of models that can be used and combined in order to describe current observations. Three strategies can be employed: training the models in advance and combining them dynamically [27], [28], continuously updating the models [10], or adding/removing (activating/desactivating) models [26].

Recurring concept drifts. The question of reoccurring drifts is essential in applications where seasonal effects can be observed or where the environment

can oscillate between several states. Various algorithms have been designed to tackle this issue.

The first method that was explicitly designed for reoccurring drifts (working with categorical attributes) is FLORA3 [28], an evolution of the original window-based FLORA method. When a drift is detected, FLORA3 inspects a pool of saved models instead of relearning a brand new model from scratch. The reuse procedure can be decomposed into three steps: Finding the optimal model (i.e., the model which makes the best predictions on the current data), update the chosen model (in order to make it consistent with current state) ,and comparing the updated version of the model to its memorized version. As an alternative to FLORA3, SPLICE-2 [15] offers another adaptation to recurring concept drifts on categorical features. The algorithm considers batches of data on which the concept is supposed to be stable. These batches are then clustered together, based on a notion of context similarity. In [29], the past history is modeled by a Markov chain and the future state is predicted according to the computed transition matrix.

Ensemble approaches are ideal for recurring drifts. For instance, Ensemble Building (EB) [24] aims to combine multiple classifiers with weights depending on their scores. If none of the known classifiers have good prediction rate on the currently observed chunk, a new classifier is trained and added to the pool. In a slightly different way, [11] chooses current models from a pool of previously learned model. The models are stored in memory, as well as their associated referee. In [17], the traces of past relevant concepts are stored in the pool of base-learners. These base-learners are learned each time that no existing classifier is a good predictor on the current window of examples. A diversity criterion on the pool of base-learners guarantees that the pool is both diverse and not cluttered.

The approach of [18] is very similar but exploits an idea that is close to CBR: Batch examples are selected by the algorithm and transformed to *conceptual vectors*. These vectors are then clustered together and a new classifier is learned for each cluster. Finally, the more generic algorithm Learn^{++}.NSE [9] is also perfectly tuned for recurring drifts: The algorithm is based on a passive incremental approach and proposes a weighted majority vote on a pool of classifiers.

CBR and online learning. Interestingly enough, the similarities between the main questions of CBR and online learning have not been exploited much. Apart from the ensemble techniques mentioned above which are implicitly related to CBR (in particular [18]), some methods use CBR in an explicit way. In [25], all new observations are directly stored in memory but, depending on their relevance to the context, they can be deactivated or reactivated. It is shown that this strategy improves the robustness of lazy learning algorithms to concept drift. CBR is used in the context of spam classification with concept drift [8]: The case base is filled with a vector representation of emails and managed using a Case Base Editing strategy [7] which removes both noisy and redundant cases. This case base editing strategy is also used by [22]. The problem of instance-based learning has also been expressed in the context of data streams [1]: The proposed

method updates the case base at each detection of a drift, implying the removal of a large number of cases.

3 Drift Adaptation seen as a CBR Problem

In this section, we present an interpretation of online learning in terms of case-based reasoning. The presented notions are given at an abstract level: an applied example is proposed in the next section.

3.1 General Process

In a context of stream mining, it is not possible to have a full CBR process at each step. The methodology we propose allies the performance qualities of active methods for stream mining and the use of memory, which is typical of CBR.

The data stream is analyzed by a drift detection algorithm (for instance ADWIN [2]) on the base of a *score*. The purpose of this algorithm is to detect when the data distribution changed and when an adaptation is needed. Since a drift is necessarily detected with some delay, a drift detection comes with a batch of instances $\mathcal{D}$ generated by the new distribution. The score is computed based on a representation model of the data. It can correspond to the error rate of the model or to its likelihood for instance. In the following, we will denote by $score(\mathcal{D}, \mathcal{M})$ the score of data $\mathcal{D}$ relative to the model $\mathcal{M}$.

Instead of relearning the model from the batch selected by ADWIN, we propose to select the model from a case base and to adapt it in order to fit the new data. This use of case base is ideal for dealing with recurring concept drifts, as suggested by the state of the art.

3.2 Case Representation

One of the central questions of CBR concerns the management of the case base and the representation of cases. In the context of online learning, we propose the following storage process. A case corresponds to a data point, after or before any transformation process. As suggested by [18], the points are then grouped into clusters corresponding to concepts. Each of the clusters is associated to a unique decision model which can be either discriminative (e.g., a classifier in supervised setting) or generative (e.g., a probability distribution in unsupervised setting).

In a perspective of reusing previously solved cases to address new questions, this representation consists of a factorized representation of problems: the solution (here the decision model) is shared by several cases.

3.3 Case Retrieval

When a drift is detected, the first question is how to associate the batch of points to a corresponding group of cases. Using the representation we proposed,

the relatedness of a batch to any case inside a cluster can be measured by its relatedness to its associated model. As a good candidate for this measure, we propose to use the score function.

The optimal cluster of cases is chosen to be the cluster such that the associated model maximizes $score(\mathcal{D}, \mathcal{M})$. Note that, especially for the first drifts, none of the learned models might describe well the observed data. In order to discard incorrect models, a threshold can be given for the score, under which no cases are selected. In the scope of this paper, we will ignore this problem.

3.4 Case Reuse

The retrieved cases do not necessarily correspond exactly to the current distribution of data. In order to cope with this problem, the decision model in use is retrained on a specific batch of data. This batch contains the points in the case cluster and the points in batch $\mathcal{D}$. This reused model thus incorporates both knowledge from the past and from current data. The model is taken as the reference model for the next observations, until a new drift is detected.

3.5 Case Revision

In the time interval between two drifts, we propose a case revision based on two aspects. On the one hand, the description model is updated online for each new observation, using a stochastic optimization scheme [5]. On the other hand, the most relevant data instances are kept in a short-term memory, in order to feed the case in the retainment phase. The relevance of an instance is evaluated with the score function, for the current model. These two actions are complementary: the model update is important in order to keep the decision model up-to-date, while the data selection contributes to an optimal case design.

3.6 Case Retainment

When a drift is detected, the model has to be saved in the case base. Two possibilities appear: either to re-write the selected case or to create a new case. This decision is motivated by the impact of creating a new model onto the global case base. If $(\mathcal{M}^{old}, \mathcal{D}^{old})$ designates the previous model and the cases associated to it, and $(\mathcal{M}^{new}, \mathcal{D}^{new})$ designates the current model and the data stored in short-term memory, one possibility to discriminate the two options is to compare $score(\mathcal{D}^{old}, \mathcal{M}^{new})$ and $score(\mathcal{D}^{old}, \mathcal{M}^{old})$. If the first score is higher, the new model is better at describing data from previous case model and thus the model has to be overridden. Otherwise, the previous model was satisfactory and the new model is relevant only for the new cases. Thus a new model has to be created and is associated to the instances in short-term memory.

In the case where the previous model is overridden, the cases stored in short-term memory are added to the case cluster of the model. In simple applications, where the number of cases per cluster is limited, only the cases with higher score are kept.

4 Application: Online Topic Modeling with AWILDA

In this section, we propose an application of the described methodology in the case of online topic modeling.

4.1 Presentation of the Problem

Topic modeling is a machine learning technique that processes documents as bag-of-words and represents them as vectors of topics. One of the most prominent methods used for topic modeling is Latent Dirichlet Allocation (LDA) [4], where documents are modeled as mixtures over latent topics, and each topic is characterized by a distribution over words.

Taking into account the order in which documents are generated is important since it allows one to track the evolution of topics over time. Variants of LDA have been proposed to incorporate temporal dynamics into the topic model [3]. However, these variants are not able to handle streams of documents arriving in real-time: They require the whole documents to be accessible in order to infer the corresponding model.

AWILDA [23] is designed to process documents arriving in a stream, one by one, and combines online LDA [16] and ADaptive Windowing (ADWIN) [2], a technique for drift detection. The main idea is that a change in the distribution generating the documents will result in a drift in the likelihood of the LDA model currently used. This change is detected in AWILDA using an ADWIN component that processes the series of likelihoods and detects when the LDA model is no longer adapted to the recently received documents. When this is the case, ADWIN returns the sub-window of documents corresponding to the new distribution. These documents are used to retrain the new topic model. Further details about AWILDA can be found in [23].

To the best of our knowledge, there is no previous work handling reoccurring drifts that are present in topic modeling. Such scenarios could easily occur in the news domain for example, where we observe unexpected events related to specific topics that recurrently appear over time and affect the distribution of words and topics in documents. In the following, we present the variant of AWILDA that is able to adapt to reoccurring drifts and that we explore in this work, followed by experiments demonstrating our approach.

4.2 AWILDA with Reoccurring Drifts

Textual content written by individuals and shared online on several platforms (e.g., tweets, news, reviews) is usually affected by their specific context that is in turn influenced by real-life events. It is essential to account for changes happening in the distribution of topics and words in order to improve document modeling. While AWILDA retrains a model at each detected drift, it cannot leverage previous learned information about a concept when it reappears due to its possible recurrence. We propose to store learned models that are no longer adapted to the current context and reuse them later when they are valid again.

In terms of the methodology described in Section 3, this problem can be described as follows. Each point corresponds to a document (described as a bag-of-words) and documents arrive sequentially as a stream. The task we address here is a modeling task: The purpose is to identify a good model that fits the data in real time. As a consequence, the model used to select the cases to cluster corresponds to the LDA model itself. The score function that we use is the log-likelihood, which measures the probability of observed documents to be generated by the model.

4.3 Experimental Results

In this subsection, we present the experiments we conducted on two datasets from different domains in order to demonstrate our approach.

Datasets. The first dataset gathers hotel reviews posted on TripAdvisor [14] and is denoted by *trip*. The dataset comprises approximatively 200k reviews published from October 2001 to November 2009 and related to hotels located in ten different cities. We expect to observe a recurrence of concepts in this type of dataset due to the seasonality effect that influences the behavior of tourists and the hotel aspects they attach importance to. The second dataset contains a collection of 9k news articles published in German on several news portals, during the month of February 2016 (*plista dataset* [19]) and is denoted by *news*. Documents from both datasets have been preprocessed by mainly removing stop words, removing words occurring once, and stemming remaining words.
Evaluation. The main goal in topic modeling is to maximize the likelihood on unseen documents. In order to evaluate the topic model, we measure, for each received document, the perplexity which is defined in [4]. Perplexity measures the capacity of the model to generalize to new data. A lower value of perplexity indicates a better generalization capacity.
Methods. We compare our approach, denoted by CB-AWILDA, to AWILDA [23] that is able to analyze documents arriving in a stream. AWILDA is better suited to handle abrupt drifts: The model is retrained for each detected drift using the documents corresponding to the new distribution. AWILDA and CB-AWILDA are considered to be receiving a stream of document in real-time and to process documents sequentially. We use the first 20% of the document stream to initialize the models and we measure perplexity for all the documents received afterwards. We report the results obtained by fixing the number of topics to 5, and the minimum number of cases to 2.
Results. Figure 1 shows the perplexity measured on the document streams of *trip* and *news* for AWILDA and CB-AWILDA. The performance of both methods at the beginning of the process is relatively similar. This is expected since the learning process is the same before any drift is detected. As more documents are received, CB-AWILDA outperforms AWILDA for the task of document modeling. For each detected drift, AWILDA is retrained using the documents related to the new distribution. This is thus pushing the model to forget previously learned information that may be valid in the future. On the

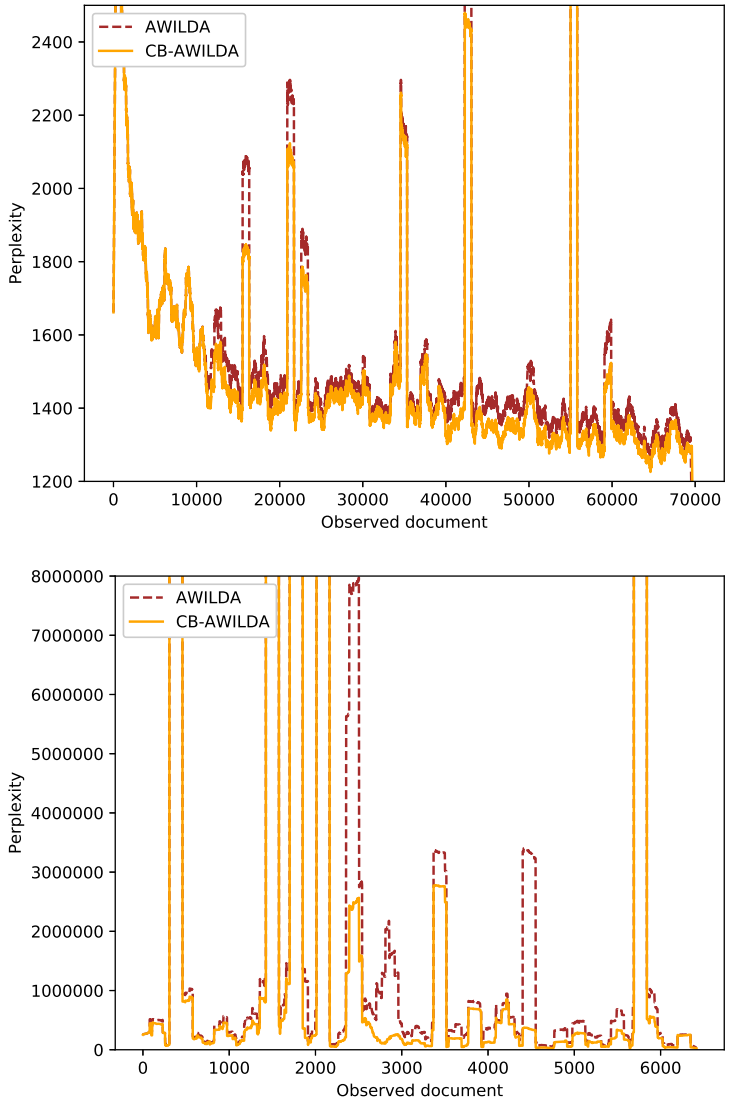

Fig. 1. Evaluation of AWILDA and CB-AWILDA for the task of document stream modeling on the *trip* (first figure) and the *news* (second figure) datasets.

other hand, CB-AWILDA leverages previously seen documents that correspond to the current distribution and uses them in the learning process. CB-AWILDA is therefore more adapted to the documents that are currently being received, which results in a better performance in terms of perplexity.

5 Conclusion

In this paper, we address the problem of online learning with reoccurring drifts and we formulate its solution in the context of the case-based reasoning framework. Observations are represented as cases and are grouped into clusters corresponding to concepts and associated to an adapted model. When a drift is detected, we retrieve the case related to the concept that is currently being observed. Retrieved cases are used to update the decision model and can be updated or overridden if necessary. We propose an application of our approach for online topic modeling. We show that taking into account reoccurring drifts improve the task of document modeling.

Future work includes the application of the proposed solution to other tasks. In particular, online recommender systems could benefit from such an approach in order to adapt to reoccurring drifts appearing on the user and item level due to seasonal and unexpected events.

References

1. Beringer, J., Hüllermeier, E.: Efficient instance-based learning on data streams. Intelligent Data Analysis **11**(6), 627–650 (2007)
2. Bifet, A., Gavalda, R.: Learning from time-changing data with adaptive windowing. In: Proceedings of the 2007 SIAM international conference on data mining. pp. 443–448. SIAM (2007)
3. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: Proceedings of the 23rd international conference on Machine learning. pp. 113–120. ACM (2006)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)
5. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, pp. 177–186. Springer (2010)
6. De Mantaras, R.L., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., T COX, M., Forbus, K., et al.: Retrieval, reuse, revision and retention in case-based reasoning. The Knowledge Engineering Review **20**(3), 215–240 (2005)
7. Delany, S.J., Cunningham, P.: An analysis of case-base editing in a spam filtering system. In: European Conference on Case-Based Reasoning. pp. 128–141. Springer (2004)
8. Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L.: A case-based technique for tracking concept drift in spam filtering. Knowledge-based systems **18**(4-5), 187–195 (2005)
9. Elwell, R., Polikar, R.: Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks **22**(10), 1517–1531 (2011)

10. Fern, A., Givan, R.: Online ensemble learning: An empirical study. Machine Learning **53**(1-2), 71–109 (2003)
11. Gama, J., Kosina, P.: Tracking recurring concepts with meta-learners. In: Portuguese Conference on Artificial Intelligence. pp. 423–434. Springer (2009)
12. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Brazilian symposium on artificial intelligence. pp. 286–295. Springer (2004)
13. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) **46**(4), 44 (2014)
14. Ganesan, K., Zhai, C.: Opinion-based entity ranking. Information retrieval **15**(2), 116–150 (2012)
15. Harries, M.B., Sammut, C., Horn, K.: Extracting hidden context. Machine learning **32**(2), 101–126 (1998)
16. Hoffman, M., Bach, F.R., Blei, D.M.: Online learning for latent dirichlet allocation. In: advances in neural information processing systems. pp. 856–864 (2010)
17. Jaber, G., Cornuéjols, A., Tarroux, P.: A new on-line learning method for coping with recurring concepts: the adacc system. In: International Conference on Neural Information Processing. pp. 595–604. Springer (2013)
18. Katakis, I., Tsoumakas, G., Vlahavas, I.: Tracking recurring contexts using ensemble classifiers: an application to email filtering. Knowledge and Information Systems **22**(3), 371–391 (2010)
19. Kille, B., Hopfgartner, F., Brodt, T., Heintz, T.: The plista dataset. In: Proceedings of the 2013 International News Recommender Systems Workshop and Challenge. pp. 16–23. ACM (2013)
20. Klinkenberg, R.: Learning drifting concepts: Example selection vs. example weighting. Intelligent data analysis **8**(3), 281–300 (2004)
21. Kuncheva, L.I., Žliobaitė, I.: On the window size for classification in changing environments. Intelligent Data Analysis **13**(6), 861–872 (2009)
22. Lu, N., Lu, J., Zhang, G., De Mantaras, R.L.: A concept drift-tolerant case-base editing technique. Artificial Intelligence **230**, 108–133 (2016)
23. Murena, P.A., Al Ghossein, M., Abdessalem, T., Cornuéjols, A.: Adaptive window strategy for topic modeling in document streams, accepted in International Joint Conference on Neural Networks 2018
24. Ramamurthy, S., Bhatnagar, R.: Tracking recurrent concept drift in streaming data using ensemble classifiers. In: Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on. pp. 404–409. IEEE (2007)
25. Salganicoff, M.: Tolerating concept and sampling shift in lazy learning using prediction error context switching. In: Lazy learning, pp. 133–155. Springer (1997)
26. 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. pp. 377–382. ACM (2001)
27. Tsymbal, A., Pechenizkiy, M., Cunningham, P., Puuronen, S.: Handling local concept drift with dynamic integration of classifiers: Domain of antibiotic resistance in nosocomial infections. In: Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on. pp. 679–684. IEEE (2006)
28. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine learning **23**(1), 69–101 (1996)
29. Yang, Y., Wu, X., Zhu, X.: Mining in anticipation for concept change: Proactive-reactive prediction in data streams. Data mining and knowledge discovery **13**(3), 261–289 (2006)