

Automating concept-drift detection by self-evaluating predictive model degradation

*Original*

Automating concept-drift detection by self-evaluating predictive model degradation / Cerquitelli, Tania; Proto, Stefano; Ventura, Francesco; Apiletti, Daniele; Baralis, ELENA MARIA. - ELETTRONICO. - arXiv:1907.08120:(2019), pp. 1-5.

*Availability:*

This version is available at: 11583/2749928 since: 2019-09-05T12:19:52Z

*Publisher:*

*Published*

DOI:

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# Automating concept-drift detection by self-evaluating predictive model degradation

Tania Cerquitelli, Stefano Proto, Francesco Ventura, Daniele Apiletti, Elena Baralis

Department of Control and Computer Engineering

Politecnico di Torino

Turin, Italy

name.surname@polito.it

## ABSTRACT

A key aspect of automating predictive machine learning entails the capability of properly triggering the update of the trained model. To this aim, suitable automatic solutions to self-assess the prediction quality and the data distribution drift between the original training set and the new data have to be devised.

In this paper, we propose a novel methodology to automatically detect prediction-quality degradation of machine learning models due to class-based concept drift, i.e., when new data contains samples that do not fit the set of class labels known by the currently-trained predictive model.

Experiments on synthetic and real-world public datasets show the effectiveness of the proposed methodology in automatically detecting and describing concept drift caused by changes in the class-label data distributions.

## KEYWORDS

AutoML, predictive analytics, concept drift, model degradation, unsupervised self-evaluation

## 1 INTRODUCTION

In many application domains, from industrial production environments to smart cities, from network traffic classification to text mining, it is quite common that the nature of the collected data changes over time due to the evolution of the phenomena under analysis, e.g., equipment maintenance, road topology changes, updated configurations, and environmental factors. However, collecting historical training sets including all possible class labels may be very hard, too expensive, or even unfeasible. Hence, when performing predictions, new data belonging to unseen class labels, i.e., labels that the predictive model did not know at training time, are likely to be presented at some point in the future, leading to wrong predictions.

Frequently updating the prediction models by extending the training set to the new data can be computationally intensive and, worse, may require the intervention of domain experts along with data scientists to translate the changes of the phenomenon into appropriate choices for the predictive task. For these reasons, it is often unfeasible or at least highly sub-optimal to simply retrain the model very frequently.

To this aim, we introduce a self-evaluation step to automatically identify and quantify the degradation of the prediction quality over time. An additional challenge in this approach is provided by the absence of the ground-truth predictions for the newly classified samples. Hence, the solution is based on *unsupervised* evaluation

metrics. These metrics can provide a twofold contribution: (i) automatically trigger a predictive model retraining, by determining when it is required to better fit the new data, and (ii) describe the changes in data distributions motivating the model update.

Many works addressed the problem of detecting data distribution changes in machine learning. In [17], three different strategies are presented: (i) instance selection, (ii) instance weighting and (iii) ensemble learning. In the first case, the goal is to consider samples relevant to the current concept: these approaches are mainly based on trailing windows moving over the latest instances. The instance-weighting strategy is based on learning algorithms able to consider weights separately for each instance, whereas the latter exploits ensembles to better capture different nuances in the data. In our solution, we exploit a trailing window on new data samples similarly to strategy (i), but we propose a new evaluation metrics to self-evaluate degradation between the training set and the current dataset, since the training set describes what the current predictive model has learnt.

A survey on concept drift adaptation is provided in [5]. In this context, we focus on class-based concept drift for predictive analytics, for which a unifying view is provided and its nuances are explored in [10]. Different techniques for monitoring two distributions at different time windows have been proposed. An entropy-based measure for data streams has been proposed in [19] to detect abrupt concept drifts due to context changes produced by sensors (e.g., wearable devices). In [3] the authors propose an adaptive sliding window technique along with a Naïve Bayes predictor to monitor the performance of a predictive model over time. Another approach addressing concept drift detection in textual data, based on support vector machines, is presented in [7]. An analysis of discrete-time Markov chains affected by concept drift has been recently proposed in [14] providing a collection of change-detection mechanisms and an adaptive learning algorithm suited to this specific context. Recent approaches aiming at detecting concept drift in the context of online learning with imbalanced classes are presented in [20], whereas the challenge of detecting model degradation when incremental learning is applied was addressed in [15]. Many related works validate their approaches on controlled synthetic datasets with known concept drifts, besides considering real-world datasets.

The solution presented in this paper improves the state-of-the-art by defining a methodology for detecting concept drifts on new incoming data over time (i) automatically (self-evaluation), (ii) in a general purpose manner (not tailored to a specific use case or application domain, nor to a specific data type), and (iii) with scalability in mind. The algorithm has been designed to be horizontally

scalable on Big Data contexts by means of the Map Reduce programming paradigm and is implemented on top of Apache Spark.

The paper is organized as follows. Section 2 introduces the overall methodology to automatically provide up-to-date predictive models when required. Section 3 presents the new self-evaluation strategy proposed to detect predictive degradation and concept drift, while Section 4 discusses some preliminary experiments to evaluate the proposed approach. Section 5 draws conclusions and discusses future extensions of the research work.

## 2 AUTOMATED CONCEPT DRIFT MANAGEMENT

We present a novel unsupervised methodology able to automatically detect class-based concept drifts by evaluating the degradation of the predictions. Specifically, we aim at identifying when additional or different class labels are required because the current ones misrepresent the new data samples.

The proposed methodology consists of three steps, as depicted in Figure 1. (i) *Model degradation self-evaluation* step, performed through a novel unsupervised approach assessing the degradation of a prediction model over time. (ii) *Semi-supervised data labeling* step, to assign labels to the new automatically-discovered classes of data. A small subset of representative samples of the new classes will be manually inspected by domain experts. Their label assignments will be used for the remaining samples in each corresponding class. (iii) *Automated KDD (Knowledge Discovery Process) to build a new predictive model*, able to correctly fit the new incoming data distributions and classification labels. This step can be automatically triggered based on the results of the previous ones, e.g., when the model degradation is higher than a given threshold. The latter step has already seen applications of state-of-the-art approaches tailored to specific data types, such as in [2] for predictive maintenance, in [4, 12] for addressing topic detection among textual document collections, and for network traffic characterization in [1].

The contribution of this paper focuses on step (i), which is a core step required to provide a solution for self-evaluating the model degradation over time, aimed at detecting concept drifts. The contribution is detailed in Section 3.

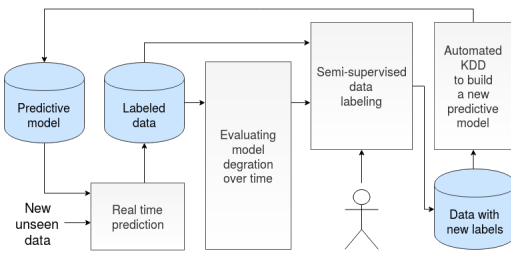


Figure 1: Building blocks of the proposed framework.

## 3 SELF-EVALUATING MODEL DEGRADATION

The knowledge of a prediction model is based on the information learnt from the training samples (historical data with labels). Hence, predictions for new data describing evolving phenomena may become misleading or erroneous, due to the different data

distributions of the new samples. To efficiently capture this model degradation over time, the proposed approach performs (i) *Baseline computation*, by computing unsupervised quality metrics on the training set, and (ii) *Self-evaluation*, by periodically recomputing the same metrics on the new data and comparing it to the metrics in (i). While new data are continuously classified by the model over time, the quality assessment has to be performed periodically. Since the quality metrics depend on the cardinality of the new data and their labels, in the current version we trigger self-evaluation when any class label is affected by a percentage increase in the number of its new samples with respect to the previous run above a given threshold (e.g., 20%).

Finally, to fully automate the process, a degradation threshold should be defined to trigger predictive model rebuilding. This threshold depends on the expectations of domain experts and end-users, the risks and costs related to the specific application, and also the number of records and classes of the dataset. Its evaluation is out of the scope of this paper.

**Self-evaluating predictive model quality over time.** Traditional evaluation techniques for predictive analytics (e.g., f-measure, precision, recall) are not applicable to our context since they require ground-truth labeled data, which is missing for newly-classified samples. Hence, we exploit an unsupervised index that, given a dataset of samples divided into classes (labels), is able to quantify both the intra-class cohesion and the inter-class separation. The degradation is then defined as a negative change in the index value of the newly-classified data with respect to the value obtained on the training set, considered as a baseline.

The proposed approach is independent of a specific quality index selection. In this work we exploit the Silhouette [13], a succinct measure of the fit of each sample within its predicted class. It measures how similar a sample is to its own class (cohesion) compared to other classes (separation). The Silhouette value ranges from -1 to +1, with higher values indicating a better match to the assigned class and a poor match to other classes. In order to compute the Silhouette, for each sample, the pairwise distance between itself and each other sample of the dataset have to be calculated. For this reason, the traditional computation of the Silhouette coefficient is not suitable for Big Data scenarios. However, scalable approaches have been proposed in literature. We adopt the approach proposed in [18]. The Silhouette measure can be used for any data type by using the appropriate distance-similarity measure (e.g., Euclidean distance for structured and numerical data, cosine similarity for textual data, Jaccard for Boolean data [16]).

**Estimating model degradation.** To this aim, we compare two quality index values: (i) the baseline, computed on the training set only, and (ii) the current one, computed on newly-labeled data. The change in quality has to be quantified separately for each class. Hence, at each index computation step, for each class, a curve is plotted, representing the Silhouette values of each point, sorted in increasing order. To enable pairwise Silhouette curve comparison, the two curves should be characterized by the same number of points. Hence, a down-sampling of the curve characterized by the largest cardinality may be required.

An upward shift of the Silhouette curve represents an improvement in terms of intra-class cohesion and inter-class separation, while, on other hand, a downward shift denotes a degradation. The

degradation of the Silhouette curve is able to detect the presence of new samples not fitting the distribution of the data seen at model-training time. Hence, it is likely that the current prediction model is not able to correctly assign the class labels to these new samples. Furthermore, the correct class labels might be new (additional) ones, unknown for the current model.

To quantify the shift of the Silhouette curves, the MAAPE (*Mean Arc-tangent Absolute Percentage Error*) [6] has been used, although other error metrics could be used alternatively. Given a prediction model trained on a set of classes  $C$  at time  $t_0$ , whose training set has a Silhouette value  $Sil_{t_0}$ , the degradation of a class  $c \in C$  at time  $t$  is described by the following relation:

$$DEG(c, t) = \alpha * \text{MAAPE}(Sil_{t_0}, Sil_t) * \frac{N_c}{N} \quad (1)$$

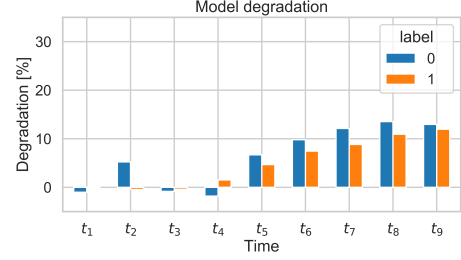
$$\alpha = \begin{cases} 1 & \text{if } : \overline{Sil_{t_0}} \geq \overline{Sil_t} \\ -1 & \text{if } : \overline{Sil_{t_0}} < \overline{Sil_t} \end{cases} \quad (2)$$

The coefficient  $\alpha$  defines if the degradation is positive, meaning a possible reduction in performance of the prediction model, or negative, when the new data fit the training distribution, hence increasing the cohesion of class  $c$ . From (1), the degradation is modelled as the MAAPE error between the baseline Silhouette  $Sil_{t_0}$  on the training set and the possibly-degraded Silhouette  $Sil_t$  computed at time  $t$  on newly-labeled data. The degradation estimation is weighted by the ratio  $N_c/N$ , where  $N_c$  is the number of new records assigned to class  $c$  and  $N$  is the total number of new samples, with  $N$  capped at the number of samples in the training set  $N_{train}$ . The cap is introduced to allow a fair comparison when  $N \gg N_{train}$ . To this aim,  $Sil_t$  is computed on a trailing window containing the latest  $N$  samples up to the number of samples of the training set  $N_{train}$ . Finally, the degradation of the whole model at time  $t$  is computed as the sum of the degradation values for each class:  $\sum_{c \in C} DEG(c, t)$ .

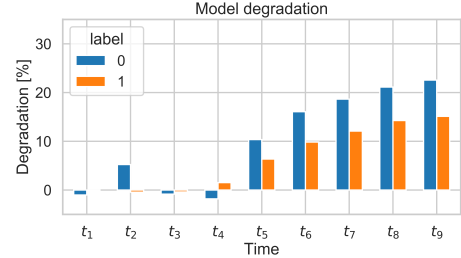
As a heuristics to automate the degradation assessment, we are currently triggering the self-evaluation when at least one class has seen an increase of a given percentage in  $N_c$  with respect to the latest computation (e.g., 20%). This heuristics allows us to avoid delaying the detection of concept drift by providing excessive inertia to the approach. Finally, we are considering to trigger a full model rebuild when the overall degradation or at least a single-class degradation are above given thresholds (e.g., 10-15% overall and 5-10% single class). The assessment of the thresholds and the evaluation of different heuristics will be addressed as future work.

#### 4 PRELIMINARY EXPERIMENTAL RESULTS

We present experimental results on two datasets containing concept shifts represented by the presence of data belonging to previously unseen classes, unknown to the predictive model. The first dataset, D1, is a synthetic dataset created with the *scikit-learn* Python library [11]. It has been generated with 4 normally distributed classes and 800,000 records, 200,000 for each class, and 10 features. Dataset D2, on the contrary, is a real-world dataset containing Wikipedia articles, extracted from 3 classes: *mathematics*, *literature*, and *food-drink*. For each class, a selection of 1,000 articles appearing in the Wikipedia index for that class have been downloaded. Each article has been pre-processed to obtain an embedded representation of 100



(a) Degradation with class 2.



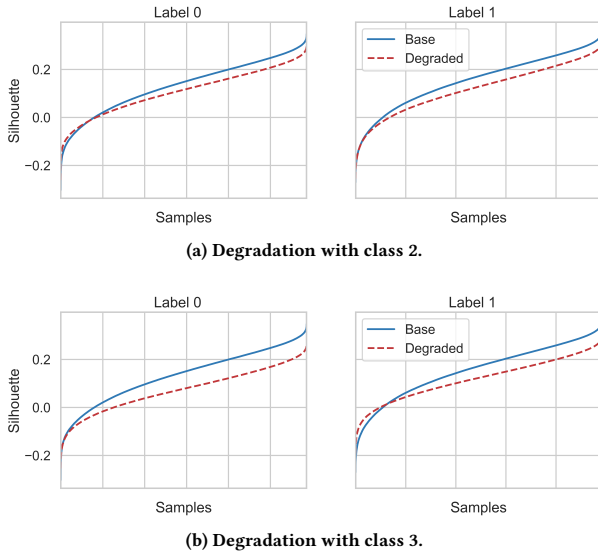
(b) Degradation with class 3.

Figure 2: Dataset D1. Model degradation over time, with training on classes 0 and 1.

features for it. The document-embedding process takes advantage of a Doc2Vec model [9], pre-trained on the English Wikipedia corpus [8]. For all datasets, a Random Forest classifier has been used as predictive model. Using a 3-fold cross-validation, the average f-measure of the predictive model is 0.964 for dataset D1, and 0.934 for dataset D2.

The experimental goal is to show the capability of the proposed methodology to correctly assess the model degradation over time when concept drifts are introduced. For both experiments, the training set is composed by a stratified sample over classes 0 and 1 of about 60% of records in each class. The remaining part of the dataset (40% class 0, 40% class 1, the unknown class 2 and/or 3 according to the dataset) is used as test set to assess model degradation. The first four test sets, from  $t_1$  to  $t_4$ , contain only data known to belong to the classes included in the training set, whereas the last test sets, from  $t_5$  to  $t_n$ , contain both known-class samples and unknown-class samples, in different proportions. The test sets have been designed to simulate the flow of time, so subsequent tests extend the data of previous ones, e.g.,  $t_1$  contains half of the test set for class 0,  $t_4$  contains the whole test set for classes 0 and 1,  $t_5$  contains all the test samples for classes 0 and 1 and 20% of the unknown class (class 2 or 3), then at each step another 20% of the unknown class is added for the evaluation of degradation until  $t_n$ .

Figure 2 shows the MAAPE degradation percentage for the training classes 0 and 1 in D1 at different time periods, when data belonging to the previously unseen classes 2 (Figure 2a) and 3 (Figure 2b) arrive starting from time  $t_5$ . While from  $t_1$  to  $t_4$  the overall MAAPE degradation is always below 5%, from  $t_5$  the overall MAAPE degradation, i.e. the sum of both classes (0 and 1), is constantly higher

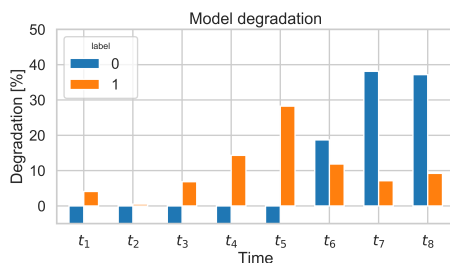


**Figure 3: Dataset D1. Baseline and degraded Silhouette curves for each class label at time  $t_9$ .**

than 10%, and the trend is coherent with the introduced drift proportion. Figures 3a and 3b report the detailed Silhouette degradation of the known classes 0 and 1 at time  $t_9$ , when there are new samples of the unknown classes 2 and 3 in the test set.

Figure 4 shows the average percentage degradation of the Wikipedia document classification model. From the histogram, it is possible to see that, up to time  $t_4$ , the new unseen data classified by the model fits well the learnt distribution, with an overall MAAPE below 15%. At time  $t_5$ , samples of an unknown class start arriving and the overall MAAPE raises above 28%.

In both datasets, the sum of the degradation for all known classes correctly detected when the shift was introduced, even if the proportion of new shifted data was low (i.e., 20% of the unknown class). Specifically, we note that the overall degradation at least duplicates from the pre-drift ( $t_1$  to  $t_4$ ) to the post-drift ( $t_5$  to  $t_n$ ), hence proving to be a promising detector of these events.



**Figure 4: Dataset D2. Model degradation over time, with training on classes 0 (food-drink) and 1 (literature) and degradation introduced by the new class 2 (mathematics).**

## 5 CONCLUSIONS AND FUTURE WORKS

This paper presented a novel strategy to detect concept drift, due to the arrival of new unseen samples not fitting the data distribution available at model-training time. The proposed approach is based on the evaluation of the predictive model degradation through an unsupervised metric (i.e. the Silhouette index) in a self-evaluating fashion, and shows promising experimental results on two datasets.

Future directions of this research include: (i) a comparison with state-of-the-art techniques focusing on drift recognition efficiency, (ii) the introduction of alternative unsupervised metrics besides the Silhouette index, (iii) the improvement of the self-evaluation triggering mechanism (currently set as a percentage of new data), and (iv) further experiments to assess the generality and the performance of the approach over different real-world datasets presenting known concept drifts.

## REFERENCES

- [1] Daniele Apiletti, Elena Baralis, Tania Cerquitelli, Paolo Garza, Danilo Giordano, Marco Mellia, and Luca Venturini. 2016. Selina: a self-learning insightful network analyzer. *IEEE Transactions on Network and Service Management* 13, 3 (2016), 696–710.
- [2] Daniele Apiletti, Claudia Barberis, Tania Cerquitelli, Alberto Macii, Enrico Macii, Massimo Poncino, and Francesco Ventura. 2018. iSTEP, an Integrated Self-Tuning Engine for Predictive Maintenance in Industry 4.0. In *IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications, ISPA/IUCC/BDCloud/SocialCom/SustainCom 2018, Melbourne, Australia, December 11–13, 2018*, 924–931. <https://doi.org/10.1109/BDCloud.2018.00136>
- [3] Albert Bifet and Ricard Gavalda. 2007. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 443–448.
- [4] Evelina Di Corso, Tania Cerquitelli, and Francesco Ventura. 2017. Self-tuning techniques for large scale cluster analysis on textual data collections. In *Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3–7, 2017*, 771–776. <https://doi.org/10.1145/3019612.3019661>
- [5] João Gama, Indrè Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 46, 4 (2014), 44.
- [6] Sungil Kim and Heeyoung Kim. 2016. A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting* 32, 3 (2016), 669–679.
- [7] Ralf Klinkenberg and Thorsten Joachims. 2000. Detecting Concept Drift with Support Vector Machines. In *ICML*, 487–494.
- [8] Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368* (2016).
- [9] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. 1188–1196.
- [10] Jose G Moreno-Torres, Troy Raeder, RociO Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern Recognition* 45, 1 (2012), 521–530.
- [11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
- [12] Stefano Proto, Evelina Di Corso, Francesco Ventura, and Tania Cerquitelli. 2018. Useful ToPIC: Self-Tuning Strategies to Enhance Latent Dirichlet Allocation. In *2018 IEEE International Congress on Big Data, BigData Congress 2018, San Francisco, CA, USA, July 2–7, 2018*, 33–40. <https://doi.org/10.1109/BigDataCongress.2018.00012>
- [13] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53 – 65.
- [14] M. Roveri. 2019. Learning Discrete-Time Markov Chains Under Concept Drift. *IEEE Transactions on Neural Networks and Learning Systems* (2019), 1–13. <https://doi.org/10.1109/TNNLS.2018.2886956>
- [15] Y. Sun, K. Tang, Z. Zhu, and X. Yao. 2018. Concept Drift Adaptation by Exploiting Historical Knowledge. *IEEE Transactions on Neural Networks and Learning Systems* 29, 10 (Oct 2018), 4822–4832. <https://doi.org/10.1109/TNNLS.2017.2775225>
- [16] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2005. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston,

MA, USA.

- [17] Alexey Tsymbal. 2004. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin* 106, 2 (2004), 58.
- [18] Francesco Ventura, Stefano Proto, Daniele Apiletti, Tania Cerquitelli, Simone Panicucci, Elena Baralis, Enrico Macii, and Alberto Macii. 2019. A new unsupervised predictive-model self-assessment approach that SCALES. In *2019 IEEE International Congress on Big Data (BigData Congress)*. IEEE, 144–148. <https://doi.org/10.1109/BigDataCongress.2019.00033>
- [19] P. Vorburger and A. Bernstein. 2006. Entropy-based Concept Shift Detection. In *Sixth International Conference on Data Mining (ICDM'06)*. 1113–1118. <https://doi.org/10.1109/ICDM.2006.66>
- [20] S. Wang, L. L. Minku, and X. Yao. 2018. A Systematic Study of Online Class Imbalance Learning With Concept Drift. *IEEE Transactions on Neural Networks and Learning Systems* 29, 10 (Oct 2018), 4802–4821. <https://doi.org/10.1109/TNNLS.2017.2771290>