Streaming Decision Trees for Lifelong Learning

Łukasz Korycki(✉) and Bartosz Krawczyk

Department of Computer Science, Virginia Commonwealth University, Richmond, VA, USA
`koryckil@vcu.edu`, `bkrawczyk@vcu.edu`

Abstract. Lifelong learning models should be able to efficiently aggregate knowledge over a long-term time horizon. Comprehensive studies focused on incremental neural networks have shown that these models tend to struggle with remembering previously learned patterns. This issue known as catastrophic forgetting has been widely studied and addressed by several different approaches. At the same time, almost no research has been conducted on online decision trees in the same setting. In this work, we identify the problem by showing that streaming decision trees (i.e., Hoeffding Trees) fail at providing reliable long-term learning in class-incremental scenarios, which can be further generalized to learning under temporal imbalance. By proposing a streaming class-conditional attribute estimation, we attempt to solve this vital problem at its root, which, ironically, lies in leaves. Through a detailed experimental study we show that, in the given scenario, even a rough estimate based on previous conditional statistics and current class priors can significantly improve the performance of streaming decision trees, preventing them from catastrophically forgetting earlier concepts, which do not appear for a long time or even ever again.

Keywords: Lifelong learning · Continual learning · Catastrophic forgetting · Data streams · Decision trees.

1 Introduction

Modern machine learning calls for algorithms that are able not only to generalize patterns from a provided data set but also to continually improve their performance while accumulating knowledge from constantly arriving data [12]. Lifelong learning aims at developing models that will be capable of working on constantly expanding problems over a long-time horizon [18]. Such learning models should keep utilizing new instances (i.e., like online learning), new classes (i.e., like class-incremental learning), or even new tasks (i.e., like multi-task learning). Whenever new information becomes available it must be incorporated into the lifelong learning model to expand its knowledge base and make it suitable for predictive analytics over a new, more complex view on the analyzed problem [15]. This requires a flexible model structure capable of continual storage of incrementally arriving data. At the same time, adding a new class or task to the model may cause an inherent bias towards this newly arrived distribution, leading to a decline of performance over previously seen classes/tasks [22]. This phenomenon is known as catastrophic forgetting and must be avoided at all costs, as robust lifelong learning models should be capable of both accumulating new knowledge and retaining the previous one [10]. Most of the research done in this domain focuses on deep neural network architectures. However, lifelong learning has many parallels with data stream mining domain, where other models (especially decision trees and their ensemble versions) are highly effective and popular [12]. Therefore, adapting streaming decision trees is an attractive potential solution to the considered issues, due to their advantages, such as lightweight structure and interpretability.

Research goal. To propose a lifelong learning version of streaming decision trees that will be enhanced with a modified splitting mechanism offering robustness to catastrophic forgetting, while maintaining all the advantages of this popular streaming classifier.

Motivation. Streaming decision trees are highly popular and effective algorithms for learning from continuously arriving data. They offer a combination of a lightweight model, adaptiveness, and interpretability while being able to handle ever-growing streams of instances. Streaming decision trees have not been investigated from the perspective of lifelong learning problems that impose the need for not only integrating new knowledge into the model, but also retaining the previously learned one. This calls for modifications of the streaming decision tree induction algorithms that will make them robust to catastrophic forgetting when creating new splits over newly appearing classes or tasks.

Overview. We offer a detailed analysis of Hoeffding Trees in the lifelong learning set-up. We show that neither these trees, nor any streaming ensemble technique using them, can retain useful knowledge over time. Their success in data stream mining can be attributed to their ability to adapt to the newest information, but no research so far has addressed the fact that they cannot memorize learned concepts well over a long-term time horizon. We identify this a fundamental problem can be found at leaves of the streaming decision trees, as they are not able to maintain information about distributions of previously seen classes, and propose a potential solution to the problem.

Main contributions. This paper offers the following contributions to the lifelong learning domain.

- **Streaming decision trees for lifelong learning.** We propose the first approach for using streaming decision trees for lifelong learning tasks, introducing a modification of Hoeffding Tree that is capable of both incremental addition of new knowledge, as well as retaining the previously learned concepts over a long-term time horizon.
- **New splitting mechanism robust to catastrophic forgetting.** We show that splitting procedure for creating new leaves in Hoeffding Tree directly contribute to the occurrence of catastrophic forgetting. To alleviate this problem, we enhance the streaming tree induction with the propagation of class-conditional attribute estimators and utilization of the class priors during entropy calculation and Bayesian classification.

- **Decision tree ensembles for lifelong learning.** We show that the proposed modification of Hoeffding Tree can be used to create highly effective ensembles robust to catastrophic forgetting, allowing us to introduce Incremental Random Forest for lifelong learning.
- **Detailed experimental study.** We evaluate the robustness of the proposed streaming decision trees through a detailed experimental study in the lifelong learning setting. We evaluate not only the global and per-class accuracy over time, but additionally the propagation of errors and model retention after being exposed to multiple new classes.

2 Related Works

Data streams. Learning from data stream focuses on developing algorithms capable of batch-incremental or online processing of incoming instances [4]. Due to the high velocity of data, time and memory constraints are important, as algorithms should be lightweight and capable of fast decision-making [12]. The focus is put on adaptation to the current state of the stream, as concept drift may dynamically impact the properties of data [13]. Thus, streaming algorithms offer high-speed and adaptive learners that provide powerful capabilities for learning from new information [3]. At the same time, knowledge aggregation and retaining mechanisms are not commonly investigated, making streaming algorithms unsuitable for lifelong learning.

Catastrophic forgetting. Lifelong learning focuses on preserving knowledge learned over a long-term time horizon, mainly with the usage of deep neural networks [18]. It has been observed that these models are biased toward the newest class, while gradually dropping their performance on older classes, which is known as catastrophic forgetting [21]. Several interesting solutions have been proposed to make neural networks robust to this phenomenon, such as experience replay [6], masking [14] or hypernetworks [16]. Despite the fact that not only neural networks suffer from catastrophic forgetting, the research on avoiding its occurrence in other learning models is still very limited.

3 Decision Trees and Lifelong Learning

Typical scenarios of lifelong learning and catastrophic forgetting involve cases in which classes arrive subsequently one after another. This means that once a given class was presented it may never appear again. Extensive works on using neural networks in such scenarios showed that such settings lead to severe learning problems for them, as mentioned in Sec. 2. While very little attention has been given to decision trees in similar scenarios, our preliminary studies of hybridizing convolutional networks with tree-based classifiers for lifelong learning indicated that streaming decision trees may struggle with exactly the same problems as neural networks. In this section, we want to emphasize this issue and propose a possible solution.

3.1 Forgetting in Streaming Decision Trees

Online decision trees have been proven to be excellent algorithms for learning from stationary and non-stationary data streams [2]. However, a more in-depth analysis of the conducted experimental research may reveal that algorithms like Hoeffding Tree [5] and Adaptive Random Forest [7] have been evaluated mainly in scenarios where incoming data per class is generally uniformly distributed over time, which means that instances of different classes are reasonably mixed with each other, without long delays between them [12]. Although researchers usually take into consideration the dynamic imbalance of analyzed streams [11], they still assume that instances of all classes appear rather frequently, even if ratios between them are skewed. The class-incremental scenarios are edge cases of extreme temporal imbalance, where the older classes do not appear ever again and the newer ones completely dominate the learning process. Let us introduce the main components of the state-of-the-art streaming decision trees and analyze what consequences the given scenario has for them.

Entropy and splits. The Hoeffding Tree model is built upon two fundamental components used at leaves: (i) Hoeffding bound that determines when we should split a node, and (ii) node statistics that are used for finding the best splits. The former is defined as:

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}, \tag{1}$$

where R is a value range, equal to $R = \log C$ for information gain calculations (C is the total number of classes), n is a number of examples seen at a node and δ is a confidence parameter. If a difference between the best potential split and the current state of the node is greater than ϵ, then there is a $1 - \delta$ confidence that the attribute introduces superior information gain and it should be used to create a split. We can express it using the following condition:

$$\Delta G(x_i, s_j) = E(x_i, s_j) - E_0 > \epsilon, \tag{2}$$

where the best potential information gain $\Delta G(x_i, s_j)$ is equal to the difference between the entropy after the best possible split $E(x_i, s_j)$ on an attribute x_i using a split value s_j, and before the split E_0. Although the condition alone is not directly related to the forgetting problem, the entropy values are, as we will show in the next steps.

The entropy for a given binary split s_j on an attribute x_i can be calculated as:

$$E(x_i, s_j) = \sum_{k=1}^{C} -p(c_k|x_i \leq s_j) \log(p(c_k|x_i \leq s_j)) - p(c_k|x_i > s_j) \log(c_k|x_i > s_j) \tag{3}$$

which simply boils down to the entropy on the left ($x_i \leq s_j$) from the split s_j and on the right ($x_i > s_j$). For the current entropy E_0 at the node we simply have:

$$E_0 = \sum_{k=1}^{C} -p(c_k) \log(p(c_k)). \tag{4}$$

Based on the given formulas, in order to find the best potential splits over all attributes and classes, we need to maintain two groups of estimators at leaves: (i) class priors $p(c_k)$, and (ii) conditional class probabilities $p(c_k|x_i)$. The former estimations can be easily obtained by counting occurrences of each class:

$$p(c_k) = \frac{n_k}{n}, \tag{5}$$

where n_k is the number of instances of class k counted for a node and n is the total number of examples received. For the latter values we use the fact that we have discrete classes and apply the conditional probability formula:

$$p(c_k|x_i) = \frac{p(x_i|c_k)p(c_k)}{p(x)}, \tag{6}$$

where $p(x)$ is the normalizing constant for all classes. The prior probability $p(c_k)$ can be omitted here, as a part of the prior scaling, to alleviate the class imbalance problems [11]. The required class-conditional attribute probabilities $p(x_i|c_k)$ are modeled using Gaussian estimators, which provide a quick and memory efficient way of obtaining the required values [19]. We use triplets consisting of a count $n_{k,i}$, mean $\mu_{k,i}$ and variance $\sigma_{k,i}$ for all pairs of classes c_k and attributes x_i. By having those models we can easily apply Eq. 6 to obtain $p(c_k|x_i \leq s_j)$ and $p(c_k|x_i > s_j) = 1.0 - p(c_k|x_i \leq s_j)$. We end up with $p(x_i \leq s_j|c_k)$, which can be calculated using the cumulative distribution function for the standard normal distribution $\Phi_k(s_j)$. It can be expressed using the error function:

$$p(x_i \leq s_j|c_k) = \Phi_k(s_j) = 0.5(1 + erf_k(s_j/\sqrt{2}), \tag{7}$$

where the value of the error function can be calculated using the stored triplets.

Finally, after finding the best possible split s_j for an attribute x_i that minimizes the entropy after a split (Eq. 3) and passing the Hoeffding bound test (Eq. 2) we can split the node and estimate the total number of instances that will go to the left and right child:

$$p_l(c_k) = p(c_k)p(c_k|x_i \leq s_j) = 1 - p_r(c_k), \tag{8}$$

where $p_l(c_k)$ and $p_r(c_k)$ are priors for the left and right child for the given class c_k, and x_i is the selected split attribute.

By default, we omit the estimation of all $p(c_k|x_i)$ after the split as it is a non-trivial task, which most likely cannot be quickly solved in the current form of the algorithm. This fact has a crucial impact on the streaming decision trees in the class-incremental scenario as we will show in the subsequent paragraphs.

Classification at leaves. After forwarding an incoming instance to a leaf in the decision tree, it is classified using majority voting based on the class priors. To improve the classification process the simple procedure is often combined with a naive Bayes classifier [1], which can be easily applied using the already stored estimators:

$$p(c_k|\mathbf{x}) = \frac{p(\mathbf{x}|c_k)p(c_k)}{p(\mathbf{x})}, \tag{9}$$

where $\mathbf{x}$ is the vector of input attributes and $p(\mathbf{x}|c_k)$ is equal to:

$$p(\mathbf{x}|c_k) = \prod_{i=1}^{m} p(x_i|c_k), \tag{10}$$

where m is the number of features. Each $p(x_i|c_k)$ can be obtained using the Gaussian density function.

Forgetting scenario. After the introduction of the leaf components and required calculations, let us now consider what will happen in the class-incremental scenario after subsequent splits. In Fig. 1 we can see an example of a sequence of 3 class batches. At the beginning, there are only instances of the first class (C0) for which the algorithm accumulates values for the prior count (Eq. 5) and conditional estimators (Eq. 6) only at the root, since there is no need for a split.

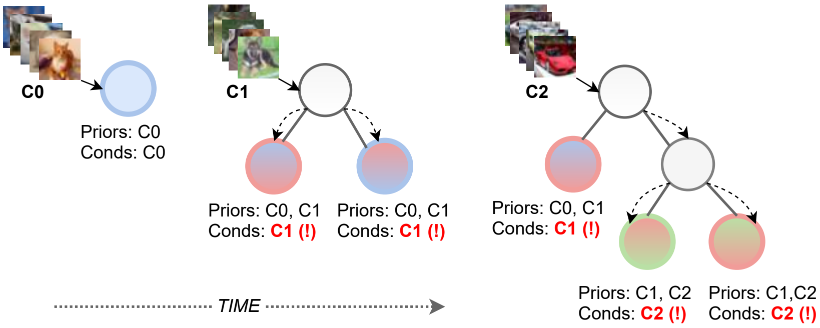

Fig. 1: Catastrophic forgetting in streaming decision trees learning from a class-incremental sequence.

Next, the second class (C1) starts arriving and at some point the Hoeffding Tree algorithm finds a good split, which creates two additional nodes and distributes priors accordingly to Eq. 8. After this step, the child nodes have some smaller priors for C0 and C1, however, the conditional estimators have been reset by default. Although we can assume that after the split some instances of class C1 can still appear and rebuild the conditional estimators, there is no chance that the same will happen for C0, which means that while its priors will be good for now, its conditional estimators (Eq. 6) will remain equal to zero, resulting in an inability of the naive Bayes classifier (Eq. 9) to recognize this class.

When the next class starts coming (C2) we can already observe a problem – since there are no instances of C0, its $p(c_0|x_i)$ is still equal to zero, which leads to the situation in which the older class is completely ignored during the entropy calculations when looking for a split (Eq. 3). Finally, once a new split is created, there will be no prior for the class at the newest leaves, since based on Eq. 8 it has to be zeroed. This concludes the learning process for class C0 which has been completely erased at the third level of the tree, and which may very likely disappear from the model completely. Even worse is the fact that the same will most likely happen to C1 and C2 as soon as new classes arrive.

Based on the analysis, we can conclude that in the class-incremental scenario, catastrophic forgetting in streaming decision trees manifests itself in three ways: **(i)** by excluding older classes from a meaningful contribution to the best split criterion, **(ii)** by disabling the conditional classification, and finally **(iii)** by erasing priors which leads to complete class forgetting at a given node.

3.2 Overcoming Catastrophic Forgetting

The observations from the previous section clearly indicate that the source of the problem with forgetting can be found at leaves and their conditional estimators. It is worth emphasizing that this issue practically does not exist in most of the commonly used data stream benchmarks, which provide instances of different classes for most of the time during the learning process. In such a case, the estimators can always rebuild themselves after new instances arrive, preventing them from forgetting most of the classes. The longer are gaps between subsequent instances of one class, the higher is the chance that the class will be temporarily or forever forgotten.

To make a step towards solving the introduced problem in Hoeffding Trees, we propose using a rough class-conditional attribute estimation after the split to prevent the model from forgetting older classes. The approach consists of two modifications: **(i)** propagating class-conditional attribute estimators (needed for Eq. 7) to children of a node being split, and **(ii)** keeping the class priors in the entropy and naive Bayes calculations to calibrate the rough estimation.

Estimator propagation. We can simply achieve the first step by copying the Gaussian parameters of each class-conditional distribution $p_{t-1}(x_i|c_k)$ before split at time step t to the left node with $p_{t,l}(x_i|c_k)$ and to the right one with $p_{t,r}(x_i|c_k)$, which results in:

$$p_{t,l}(x_i|c_k) = p_{t,r}(x_i|c_k) = p_{t-1}(x_i|c_k), \tag{11}$$

for each class c_k and attribute x_i. This is obviously a very rough estimate, however, since we assume simple Gaussian distributions, the error does not have to be critical and may provide more benefits than obstructions. Most likely, providing any platform for an older class is more important than the risk of making the estimation error. In addition, the estimate may still be fine-tuned by instances that come to this node before the class batch ends.

Prior scaling. By sticking to the prior probabilities $p(c_k)$ in the entropy calculations (Eq. 3) and Bayesian classification (Eq. 9), we attempt to somehow adjust the rough estimate from the previous step. Since the split class priors are relatively well-estimated, we can utilize them to softly scale the class-conditional distributions to become more adequate to the state after the split. Although this step does not change the shape of the distribution horizontally, it may increase or decrease the influence of the distribution by scaling it vertically based on the formula:

$$p_t(x_i|c_k) = p_{t-1}(x_i|c_k)p_t(c_k). \tag{12}$$

Ensembles. Finally, the modified Hoeffding Tree can be simply used as a base learner of the Incremental Random Forest, which is an Adaptive Random Forest without change detectors and node replacement mechanisms. The only difference between the standard forest and the ensemble using our modified tree is that we have to keep statistics for all attributes at leaves, not only for those within a random subspace, since we do not know which attributes will be needed at a lower level. By combining the robustness of ensemble techniques with improvements of the base learner we may potentially alleviate the catastrophic forgetting problem even more.

4 Experimental Study

In the following experiments, we aim at proving that our proposed modifications of the Hoeffding Tree algorithm are capable of alleviating the catastrophic forgetting in decision trees learning from class-incremental streams, allowing for the application of these models in such scenarios. Our goal was to answer the following research questions.

- **RQ1**: Does the proposed algorithm effectively address the problem of catastrophic forgetting in streaming decision trees?
- **RQ2**: Can the presented decision tree be utilized as a base learner of a random forest? Does it further improve the classification performance?
- **RQ3**: Is it possible to solve the presented problem by using a different already available ensemble technique?

In order to to improve reproducibility of this work, all of the presented algorithms and details of the evaluation have been made available in a public repository: github.com/lkorycki/lldt.

4.1 Data

To evaluate the baseline and proposed models in the scenario of lifelong learning and catastrophic forgetting, we used popular visual data sets commonly used for the given task. The first three were used as simpler sequences consisting of 10 classes: **MNIST**, **FASHION**, **SVHN**. Next, we utilized 20 superclasses of the CIFAR100 data set (**CIFAR20**), as well as we extracted two 20-class subsets of the IMAGENET: **IMAGENET20A** and **IMAGENET20B**. All of the sets were transformed into class-incremental sequences in which each batch contained only one class and each class was presented to a classifier only once. All of the evaluated models were processing the incoming batches in a streaming manner, one instance after another.

The MNIST and FASHION data sets were transformed into a series of flattened arrays (from raw images), which provided us with feature vectors of size

784. The rest of the used benchmarks were pre-processed using pre-trained feature extractors. For SVHN and CIFAR20 we used ResNeXt-29 with its cardinality equal to 8 and using widen factor equal to 4. We extracted the output of the last 2D average pooling and processed it with an additional 1D average pooling, which resulted in a feature vector consisting of 512 values. For the IMAGENET-based sets we directly utilized the output of the last average pooling layer of the ResNet18 model, which once again gave us 512-element vectors.

4.2 Algorithms

In our experiments, we compared the proposed single tree (**HT+AE**) with the original streaming algorithm (**HT**) [5], as well as the incremental random forest using our base learner (**IRF+AE**) with its baseline (**IRF**) to answer the first two research questions. Next, we evaluated other ensemble techniques to check whether it is possible that a solution to the introduced problem lies solely in a different committee design (the last research question). We investigated drift-sensitive Adaptive Random Forest (**ARF**) [7], online bagging without random subspaces per node (**BAG**) [17], online random subspaces per tree (**RSP**) [8] and the ensemble of 1-vs-all classifiers (**OVA**) [9].

All of the algorithms used Hoeffding Trees as base learners with confidence set to $\delta = 0.01$, bagging lambda equal to $\lambda = 5$, split step $s = 0.1$ (10% of a difference between the maximum and minimum attribute value) and split wait equal to $w = 100$ for all sets except for the slightly smaller IMAGENET-based ones for which we set $w = 10$. All of the ensembles used $n = 40$ base learners.

4.3 Evaluation

Firstly, for all of the considered sequences, we measured **hold-out accuracy** [20] per each class after each class batch and used it to calculate the average accuracy per batch and the overall average for a whole sequence. Secondly, we collected data for **confusion matrices** after each batch to generate the average matrices which could help us illustrate the bias related to catastrophic forgetting. Finally, we measured the **retention** of the baseline and improved algorithms to show how well the given models remember previously seen concepts.

4.4 Results

Analysis of the average predictive accuracy. Tab. 1 presents the average accuracy over all classes for all six used class–incremental benchmarks. This is the birds eye view on the problem and the performance of the analyzed methods, allowing us to assess the general differences among the algorithms. We can see that the standard HT and IRF were significantly outperformed by the proposed HT+AE and IRF+AE approaches. For HT the proposed propagation of class-conditional attribute estimators and storing the class priors led to very significant improvements on all data sets, which is especially visible on CIFAR20 (almost

0.3) and IMAGENET20A (0.2). Similar improvements can be observed for IRF, especially for CIFAR20 where the modifications led to 0.28 improvement. The SVHN benchmark shows the smallest improvements out of all six data sets, which can be explained by the extractor potentially being very strongly fine-tuned for this problem. Thus extracting well-separated class embeddings may slightly alleviate the catastrophic forgetting on its own (although the proposed modifications still help).

The impact of different ensemble architectures. To truly understand the impact of catastrophic forgetting on HT and IRF, we decided to see if other ensemble architectures may behave better in class-incremental lifelong learning scenarios. Tab. 1 presents results for four other popular streaming ensemble architectures. We can see that all of them performed poorly on every data set, offering inferior predictive accuracy to the baseline IRF. This shows that the choice of an ensemble architecture on its own does not offer improved robustness to catastrophic forgetting. As a result, we have a good indication that our modifications of the HT splitting procedure are the sole source of the achieved impressive gains in accuracy. However, a more in-depth analysis of these models will allow us to gain better insights into the nature of catastrophic forgetting in streaming decision trees.

Table 1: The average accuracy on all class-incremental sequences.

Model	MNIST	FASHION	SVHN	CIFAR20	IMGN20A	IMGN20B
HT	0.6283	0.5720	0.8845	0.3511	0.4589	0.5301
HT+AE	**0.8398**	**0.7037**	**0.9510**	**0.6497**	**0.6530**	**0.6730**
IRF	0.8662	0.7355	0.9334	0.4467	0.6890	0.7500
IRF+AE	**0.9645**	**0.8698**	**0.9733**	**0.7298**	**0.7777**	**0.8121**
ARF	0.2929	0.2929	0.2929	0.1799	0.2411	0.2849
OVA	0.3416	0.2929	0.5033	0.1805	0.3842	0.3847
BAG	0.7096	0.6446	0.9029	0.3737	0.5709	0.6635
RSP	0.6202	0.5898	0.9087	0.3734	0.6337	0.6995

Analysis of the class-batch performance. Fig. 2 depicts the average accuracy after each class appearing incrementally. This allows us to visually analyze the stability of the examined methods and their response to the increasing model size (when more and more classes need to be stored and remembered). We can see that both proposed HT+AE and IRF+AE offered significantly improved stability over the baseline approaches, maintaining their superior predictive accuracy regardless of the number of classes. Additionally, we can see that the baseline models tended to deteriorate faster when the number of classes became higher (e.g., HT and IRF on MNIST and FASHION). At the same time, the proposed modifications could accommodate all the classes from the used benchmarks with-

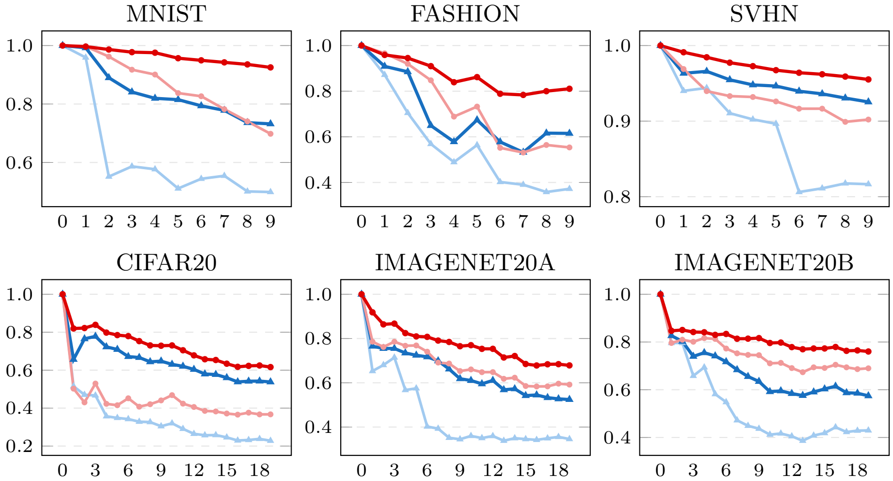

Fig. 2: The average class accuracy for the baseline tree-based models (▲ HT, ● IRF) and the proposed ones (▲ HT+AE, ● IRF+AE) after each class batch.

out destabilization of their performance. It is worth noting that HT+AE was often capable of outperforming IRF. This is a very surprising observation, as the modification of class-conditional estimators allows a single decision tree to outperform a powerful ensemble classifier. This shows that the proposed introduction of robustness to catastrophic forgetting into streaming decision trees is a crucial improvement of their induction mechanisms.

Analysis of the class-based performance. Fig. 3 depicts the accuracy per batch on selected classes. This allows us to understand how the appearance of new classes affects the performance on previously seen ones. We can clearly see that both HT and IRF were subject to catastrophic forgetting, very quickly forgetting the old classes. While they were very good at learning the newest concept, their performance degraded with every newly arriving class, showing their capabilities of aggressively adapting to new knowledge, but not retaining it over time. This was especially vivid for the first class for each data set (C0), where it was completely forgotten (i.e., accuracy on it drops to zero) as soon as 1-2 new classes appeared. The proposed propagation of class-conditional attribute estimators and storing the class priors in HT+AE and IRF+AE led to a much better retaining of knowledge extracted from old classes. In some cases (e.g., CIFAR20 or IMAGENET20) we can see that the accuracy for old classes remained almost identical through the entire duration of the lifelong learning process. This is a highly sought-after property and attests to the effectiveness of our proposed modifications.

Analysis of the confusion matrices. Fig. 4 depicts the confusion matrices averaged over all examined data sets (10 classes from each for the visualization sake). Based on that we can directly compare how errors are distributed among

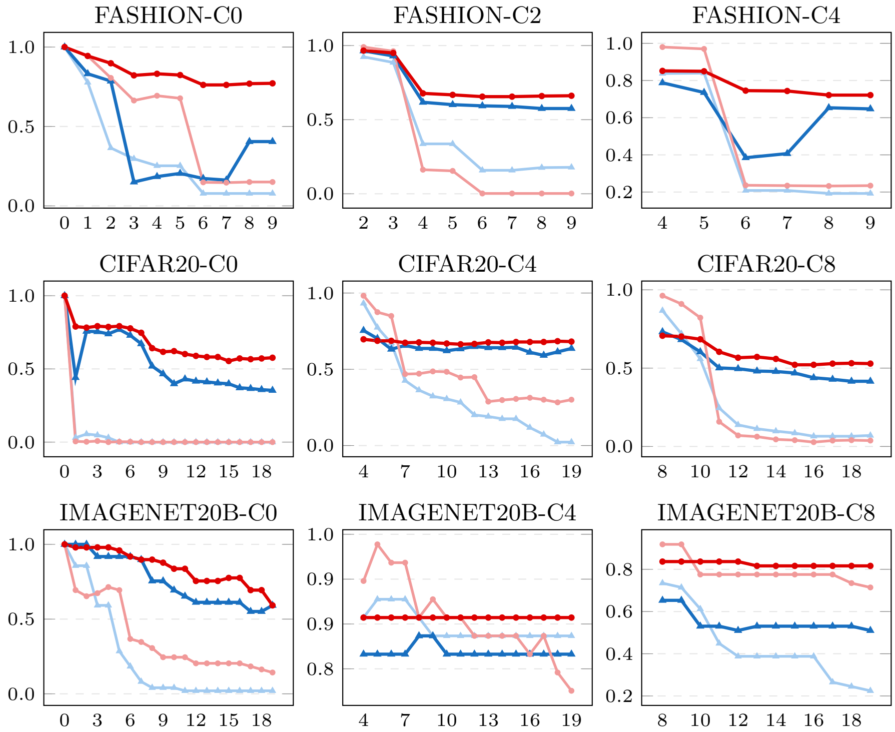

Fig. 3: The average accuracy for selected classes of FASHION, CIFAR20 and IMAGENET20B for the baseline tree-based models (▲ HT, ● IRF) and the proposed ones (▲ HT+AE, ● IRF+AE) after subsequent class batches.

classes for HT vs. HT+AE and IRF vs. IRF+AE. We can see that the proposed modifications in HT+AE and its ensemble version led to a much more balanced lifelong learning procedure that both avoided the bias towards the newest class (i.e., is robust to catastrophic forgetting) and the bias towards older classes (i.e., offers capabilities for incorporating new information into the model in an effective manner). These confusion matrices further confirm our observations made in previous points of this discussion that our proposed modifications lead to robust streaming decision tree induction for lifelong learning.

Analysis of the retention of information. Fig. 5 depicts the average retention of information about a class after +k new classes appeared. This helps us analyze how each of examined models manages its knowledge base and how flexible it is to add new information to it. An ideal model would perfectly retain the performance on every previously seen class, regardless of how many new classes it has seen since then. We can see that the baseline HT and IRF offered very good performance on the newest class, but drastically dropped it after seeing as little as 2 new classes. This further enforces our hypothesis that standard

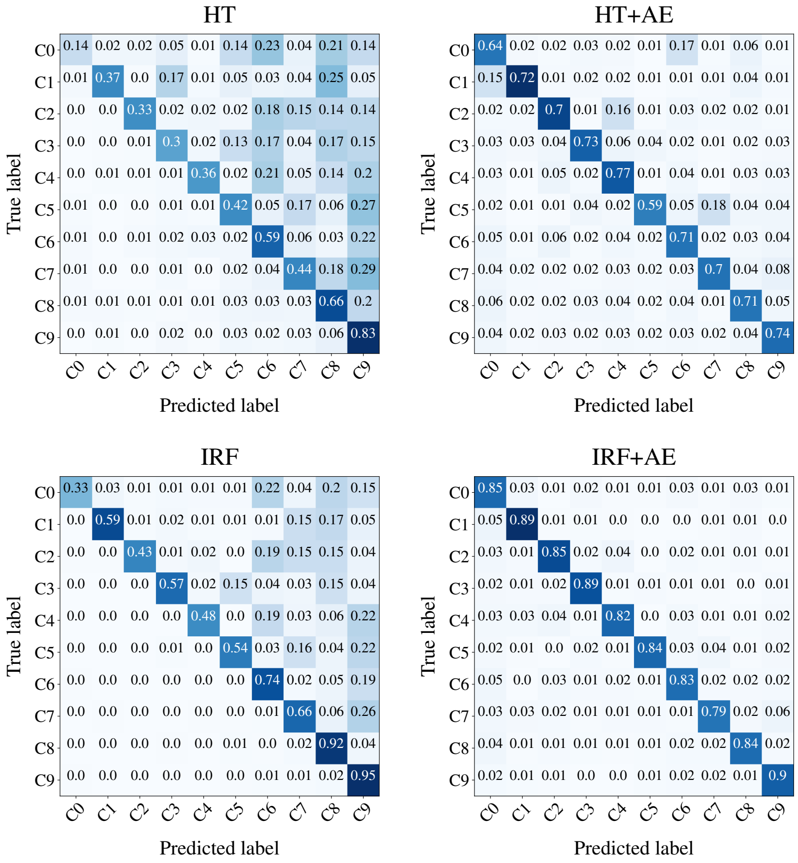

Fig. 4: The average confusion matrices.

decision trees and their ensembles cannot avoid catastrophic forgetting and thus cannot be directly used for lifelong learning. However, when we enhance HT with the proposed propagation of class-conditional attribute estimators and storing the class priors, we obtain a streaming decision tree that can learn new information almost as effectively as its standard counterpart, while offering excellent robustness to catastrophic forgetting (**RQ1 answered**). Furthermore, we can see that HT+AE can be utilized as a base learner for ensemble approaches, leading to even further improvements in its accuracy and information retention (**RQ2 answered**).

Batch-based performance of the ensemble architectures. Fig. 6 depicts the average accuracy after each class appearing incrementally for the reference ensemble approaches. This confirms our observations from the earlier point that the ensemble architecture itself does not have any impact on the catastrophic forgetting occurrence. Reference methods use different ways of data partitioning

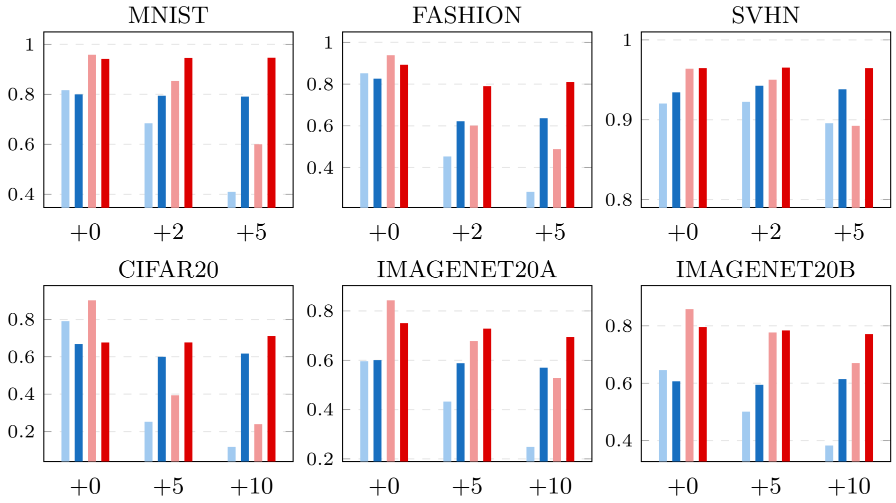

Fig. 5: The average retention after +k class batches since the moment a class appeared for: ■ HT, ■ HT+AE, ■ IRF, ■ IRF+AE

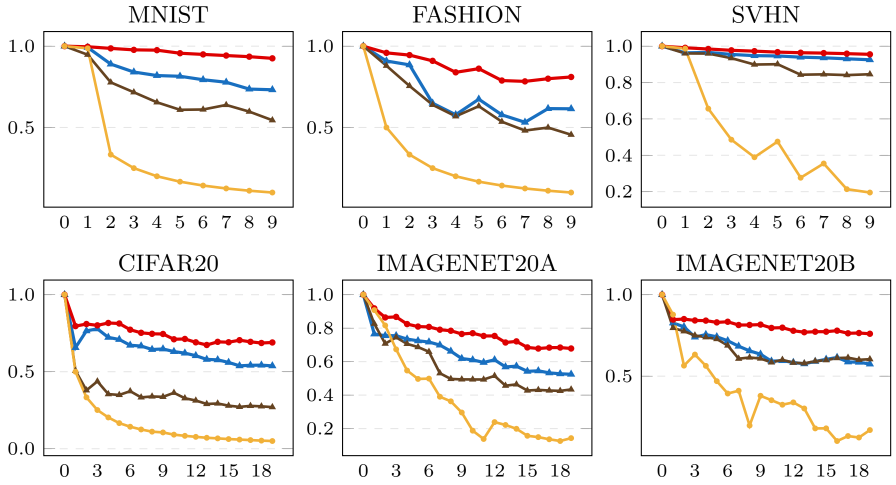

Fig. 6: The average class accuracy for other baseline models (▲ OVA, ▲ BAG) and the proposed ones (● HT+AE, ● IRF+AE) after each class batch.

(subsets of instances, features, or classes), but none of them allowed for better retention of old information. What is highly interesting is that HT+AE (a single decision tree) could outperform any ensemble of trees that do not use our proposed modifications. This shows the importance and significant impact of propagation of class-conditional attribute estimators and storing the class priors on the usefulness of streaming decision trees for lifelong learning. Therefore, catastrophic forgetting can be avoided by using a robust base learner, not changing the ensemble structure (**RQ3 answered**).

5 Summary

In this work, we identified and emphasized the issue of catastrophic forgetting that occurs when traditional streaming decision trees attempt to learn in class-incremental lifelong learning scenarios. Through an in-depth analysis of the Hoeffding Tree algorithm, we found out that the source of the algorithm's weakness comes from the lack of additional support for class-conditional attribute estimators, which tend to forget older classes after splits. The issue critically affects different aspects of tree-based learning, ranging from the procedure for finding new splits to classification on leaves.

To solve the introduced problem, we proposed a rough estimation of the conditional distributions after a split, based on distributions and priors aggregated at a node before it is divided. Our extensive experimental study has shown that this simple yet effective approach is capable of providing excellent improvements for both single trees and incremental forests. As a result, we proved that the proposed method turns the standard streaming trees into learners suitable for lifelong learning scenarios.

In future works, we plan to find more precise estimators, which may need to be supported by some local experience replay utilizing small buffers of either input instances or prototypes.

References

1. Bifet, A., Gavaldà, R.: Adaptive Learning from Evolving Data Streams. In: Adams, N.M., Robardet, C., Siebes, A., Boulicaut, J. (eds.) Advances in Intelligent Data Analysis VIII, 8th International Symposium on Intelligent Data Analysis, IDA 2009, Lyon, France, August 31 - September 2, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5772, pp. 249–260. Springer (2009)
2. Bifet, A., Zhang, J., Fan, W., He, C., Zhang, J., Qian, J., Holmes, G., Pfahringer, B.: Extremely Fast Decision Tree Mining for Evolving Data Streams. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017. pp. 1733–1742. ACM (2017)
3. Cano, A., Krawczyk, B.: Kappa Updated Ensemble for drifting data stream mining. Mach. Learn. **109**(1), 175–218 (2020)
4. Ditzler, G., Roveri, M., Alippi, C., Polikar, R.: Learning in Nonstationary Environments: A Survey. IEEE Comput. Intell. Mag. **10**(4), 12–25 (2015)
5. Domingos, P.M., Hulten, G.: Mining high-speed data streams. In: Ramakrishnan, R., Stolfo, S.J., Bayardo, R.J., Parsa, I. (eds.) Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000. pp. 71–80. ACM (2000)
6. Fujimoto, S., Meger, D., Precup, D.: An Equivalence between Loss Functions and Non-Uniform Sampling in Experience Replay. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)

7. Gomes, H.M., Bifet, A., Read, J., Barddal, J.P., Enembreck, F., Pfahringer, B., Holmes, G., Abdessalem, T.: Adaptive random forests for evolving data stream classification. Mach. Learn. **106**(9-10), 1469–1495 (2017)
8. Gomes, H.M., Read, J., Bifet, A.: Streaming Random Patches for Evolving Data Stream Classification. In: 2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019. pp. 240–249. IEEE (2019)
9. Hashemi, S., Yang, Y., Mirzamomen, Z., Kangavari, M.R.: Adapted One-versus-All Decision Trees for Data Stream Classification. IEEE Trans. Knowl. Data Eng. **21**(5), 624–637 (2009)
10. Korycki, Ł., Krawczyk, B.: Class-Incremental Experience Replay for Continual Learning under Concept Drift. CoRR **abs/2104.11861** (2021), http://arxiv.org/abs/2104.11861
11. Korycki, Ł., Krawczyk, B.: Online Oversampling for Sparsely Labeled Imbalanced and Non-Stationary Data Streams. In: 2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020. pp. 1–8. IEEE (2020)
12. Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Wozniak, M.: Ensemble learning for data stream analysis: A survey. Inf. Fusion **37**, 132–156 (2017)
13. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., Zhang, G.: Learning under Concept Drift: A Review. IEEE Trans. Knowl. Data Eng. **31**(12), 2346–2363 (2019)
14. Mallya, A., Davis, D., Lazebnik, S.: Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 11208, pp. 72–88. Springer (2018)
15. Mishra, M., Huan, J.: Learning Task Grouping using Supervised Task Space Partitioning in Lifelong Multitask Learning. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015. pp. 1091–1100. ACM (2015)
16. von Oswald, J., Henning, C., Sacramento, J., Grewe, B.F.: Continual learning with hypernetworks. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020)
17. Oza, N.C.: Online bagging and boosting. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, Hawaii, USA, October 10-12, 2005. pp. 2340–2345. IEEE (2005)
18. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks **113**, 54–71 (2019)
19. Pfahringer, B., Holmes, G., Kirkby, R.: Handling Numeric Attributes in Hoeffding Trees. In: Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference, PAKDD 2008, Osaka, Japan, May 20-23, 2008 Proceedings. Lecture Notes in Computer Science, vol. 5012, pp. 296–307. Springer (2008)
20. Raschka, S.: Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. CoRR **abs/1811.12808** (2018)
21. Yao, X., Huang, T., Wu, C., Zhang, R., Sun, L.: Adversarial Feature Alignment: Avoid Catastrophic Forgetting in Incremental Task Lifelong Learning. Neural Comput. **31**(11), 2266–2291 (2019)
22. Zaidi, N.A., Webb, G.I., Petitjean, F., Forestier, G.: On the Inter-relationships among Drift rate, Forgetting rate, Bias/variance profile and Error. CoRR **abs/1801.09354** (2018), http://arxiv.org/abs/1801.09354