Lightweight Conditional Model Extrapolation for Streaming Data under Class-Prior Shift

Paulina Tomaszewska
Faculty of Mathematics and Information Science
Warsaw University of Technology
Warsaw, Poland

Christoph H. Lampert
Machine Learning and Computer Vision Group
Institute of Science and Technology Austria (ISTA)
Klosterneuburg, Austria

ABSTRACT

We introduce *LIMES*, a new method for learning with non-stationary streaming data, inspired by the recent success of meta-learning. The main idea is not to attempt to learn a single classifier that would have to work well across all occurring data distributions, nor many separate classifiers, but to exploit a hybrid strategy: we learn a single set of model parameters from which a specific classifier for any specific data distribution is derived via classifier adaptation. Assuming a multi-class classification setting with class-prior shift, the adaptation step can be performed analytically with only the classifier's bias terms being affected. Another contribution of our work is an extrapolation step that predicts suitable adaptation parameters for future time steps based on the previous data. In combination, we obtain a lightweight procedure for learning from streaming data with varying class distribution that adds no trainable parameters and almost no memory or computational overhead compared to training a single model. Experiments on a set of exemplary tasks using Twitter data show that *LIMES* achieves higher accuracy than alternative approaches, especially with respect to the relevant real-world metric of lowest within-day accuracy.

1 Introduction

In our current era of connected devices, data is often generated in a continuous manner, e.g. by mobile devices, intelligent sensors or surveillance equipment. This creates a challenge for machine learning systems, especially when the available computational and memory resources are limited, such as when learning *at the edge*. On the one hand, systems cannot store all prior data as the necessary amount of storage would grow linearly over the runtime of the system. On the other hand, the systems have to make predictions continuously, interleaved with learning, rather than following the traditional *offline* setting with separate training and prediction phases. Furthermore, continuously generated data is typically non-stationary: its data distribution varies over time, e.g. due to changing environmental conditions or changing user behavior.

For a machine learning system, a time-varying data distribution is problematic, because it means that previously observed data might not be representative of future data. Consequently, simply learning a single fixed model, as is usually done in the standard setting of independent and identically distributed (i.i.d.) data, leads to suboptimal results. Alternatively, it is possible to learn multiple models, essentially starting a new one whenever the data distribution changes. This, however, can also lead to suboptimal results when each model achieves acceptable prediction quality only after having seen enough data. Furthermore, learning multiple models leads to increased memory requirements.

In this work, we introduce *LIMES* (short for Lightweight Model Extrapolation for Streaming data) that offers a new way for learning classifiers from non-stationary streaming data in resource-constrained settings. Inspired by recent progress in *meta-learning*, *LIMES*'s core idea is to combine the resource efficiency of having only a single set of parameters with the flexibility of learning different models for different data distributions. We construct a base model that can be adapted efficiently to different target distributions together with a procedure for forecasting the parameters necessary for adapting the base model to be used on future data, even before any examples of that data have been observed.

Specifically, we adopt the scenario of multi-class classification under *class-prior shift*, in which the class frequencies change over time, while the class-conditional data densities remain approximately constant. One of our contributions is to show how in this setting it is possible to learn a model from multiple data distributions, and how make use of it for yet another data distribution, with hardly any computational or memory overhead. Furthermore, we introduce an *extrapolation* step that forecasts the class distribution of future data based on the observed previous data.

We show experimentally on exemplary tasks using Twitter data that *LIMES* achieves higher accuracy than either learning a single model or an ensemble of multiple models. The code is available at https://github.com/ptomaszewska/LIMES.

2 Method

In this section, we formalize the learning scenario, we study and introduce the proposed *LIMES* method.

2.1 Learning Scenario

We assume a time-varying data distribution $p_t(x, y)$ for $t = 1, 2, \ldots$, where $x \in \mathcal{X}$ are the designated inputs to the learning system and $y \in \mathcal{Y} = \{1, \ldots, L\}$ are the output labels (Bartlett [1992]). The goal of a learning system is to create a predictive model, $f_t : \mathcal{X} \rightarrow \mathcal{Y}$, for any time step t, with as small as possible *risk*, $R_t(f) = \mathbb{E}_{(x,y) \sim p_t}[\ell(y, f_t(x))]$, where ℓ is the 0/1-loss function, $\ell(y, y') = [\![y \neq y']\!]$[1] For this task, the learning system can make use of observed data from prior time steps, $S^1, \ldots, S^{t-1}$, where each $S^\tau = \{(x_1^\tau, y_1^\tau), \ldots, (x_{n_\tau}^\tau, y_{n_\tau}^\tau)\} \sim p_\tau(x, y)$, for $\tau = 1, \ldots, t-1$. In line with practical resource-constrained applications, we assume a continuously running learning system that is kept up-to-date by *periodic retraining* (Graepel et al. [2010], McMahan et al. [2013], Li et al. [2015]). That means, only at the end of each time step, it is possible to update the model parameters. The obtained model is then used to make predictions for all data during the next time step.

A typical example of this setting is the real-time classification of social media streams. For example, for our experimental evaluation in Section 3, the inputs are tweets and the outputs are the countries from which the tweets originated. The time index corresponds to the time intervals over which the data distribution can be assumed to be relatively stable, in our case *hours*. Within a few time steps, the data distribution can change quite rapidly. Intuitively, users from different countries tweet predominantly at different times due to their respective time zones. This causes a time-varying data distribution in which class proportions fluctuate substantially over time.

2.2 Learning Methods

A number of possible approaches can be used to tackle the learning problem introduced above. The most classical is standard *empirical risk minimization* (Vapnik [2013]), in which a model, f, from a class of possible ones, $\mathcal{F}$, is sought that minimizes the training error across all the data observed so far,

$$f_{t+1} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \sum_{\tau=1}^{t} \hat{R}_\tau(f), \qquad (1)$$

where

$$\hat{R}_\tau(f) = \sum_{(x,y) \in S^\tau} \ell(y, f(x)) \qquad (2)$$

is the training error (empirical risk) of a model f on the data from the time step τ. Optionally, a regularizer or other terms can be added to improve generalization or to make the optimization problem better behaved.

Empirical risk minimization is widely used when data is sampled i.i.d. from a stationary distribution. However, for learning from non-stationary data, it has a number of shortcomings. First, Equation (1) will not actually learn a model adapted for the coming time step $t + 1$, but rather one that works well on average across all data distributions in the past. Second, minimizing Equation (1) is intractable in a practical streaming setting. It would require having stored all training sets, $S^1, \ldots, S^t$. which is impractical for systems that are meant to run continuously over a long time with limited resources, such as *at the edge* (Murshed et al. [2021]).

Alternatively, one can find a model by training only on the most recently available data,

$$f_{t+1} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \hat{R}_t(f). \qquad (3)$$

---

[1]The *Iverson* brackets $[\![P]\!]$ for a predicate P take the value 1 if P is true and 0 otherwise.

This *restart* approach overcomes the second problem and mitigates the first one. However, the procedure completely ignores non-recent data. Therefore, the models it produces are usually limited in the accuracy they can achieve.

Incremental training (Giraud-Carrier [2000]) avoids the problem of having to store all data, but still allows models to benefit from all available data. Each f_{t+1} is obtained by running a minimization routine on the most recent empirical risk, $\hat{R}_t$, as in Equation (3). However, instead of running the optimization routine until convergence, one initializes the process at the parameters of the previous model, f_t, and performs early stopping, e.g. after one pass through the data. Models learned by incremental training depend on all available data. At the same time, they tend to be biased towards recently seen data. The latter can be a benefit if the data distribution changes slowly, but it can also be a disadvantage, if data from more distant time steps is valuable, such as for periodic data distributions.

A way to preserve information from more distant time step is to combine incremental training with *model ensembling* (Krawczyk et al. [2017], Street and Kim [2001], Minku and Yao [2011], Muhlbaier and Polikar [2007]). Instead of training a single classifier, one trains multiple prototypical models, $g_1, \ldots, g_K$, on different subsets of the available data. The desired f_{t+1} is then constructed out of this ensemble based on contextual information or, if available, data from the target distribution (Cruz et al. [2018], Almeida et al. [2018]). The cost of the gained flexibility is a K-fold increase in memory requirements in order to store all models. At the same time, the amount of training data necessary to achieve satisfactory accuracy tends to be larger as well, when each model is trained only on a subset of all observed data.

Recently, the problems of *class-incremental learning* (Rebuffi et al. [2017]) and *task-incremental learning* (Li and Hoiem [2017]) have attracted a lot of attention in the machine learning community, see e.g. (Delange et al. [2021]) for a survey. These problems are special cases of the problem of learning with non-stationary data, based on additional assumptions about which data appears when and to which tasks the resulting functions will be applied. Specialized algorithms have been derived that aim at giving deep neural networks the ability to learn continually without suffering from catastrophic forgetting (McCloskey and Cohen [1989], French [1999]). However, those recent methods are not applicable in the situation in which we are interested. On the one hand, that is because they solve different problems, and on the other hand, because they require additional resources either for storing part of the observed data (Rebuffi et al. [2017], He et al. [2020]) or growing the model architecture over time (Rusu et al. [2016], Yoon et al. [2018]).

2.3   Classifier Adaptation

The *LIMES* method we propose, combines the best aspects of the different approaches discussed before. Like in incremental training, only a single model is trained, for which all available data is used. Like in ensemble learning, the predictions are made with a model specifically adapted to the target distribution, thereby allowing for higher prediction accuracy. The core concept to achieve both properties simultaneously is a form of *classifier adaptation* (Saerens et al. [2002], Royer and Lampert [2015]), that we discuss in the following.

We adopt a setting of probabilistic multi-class classification, such as implemented by logistic regression or any neural networks with softmax output layer. Such a model represents conditional distributions over the label set, $\mathcal{Y} = \{1, \ldots, L\}$, in log-linear form

$$\hat{p}(y|x; W, b) \propto \exp(w_y^\top \phi(x) + b_y). \tag{4}$$

The model parameters are the weight vectors $W = (w_1, \ldots, w_L) \in \mathbb{R}^{d \times L}$, and per-class bias terms $b = (b_1, \ldots, b_L) \in \mathbb{R}^L$. The feature map $\phi : \mathcal{X} \to \mathbb{R}^d$ can be fixed (in the case of logistic regression) or learned together with the classifier parameters (in the case of a neural network).

The proportionality constant for the right hand side of Equation (4) can be computed efficiently for any $x \in \mathcal{X}$ as $\frac{1}{Z(x)}$ with $Z(x) = \sum_{z \in \mathcal{Y}} \exp(w_z^\top \phi(x) + b_z)$. We generally suppress this step from our notation for the sake of conciseness.

The probabilistic model allows one to make predictions for new inputs $x \in \mathcal{X}$ by

$$f(x) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \, \hat{p}(y|x; W, b) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \, [w_y^\top \phi(x) + b_y]. \tag{5}$$

The parameters W, b (and potentially ϕ) are typically trained to make the conditional model distribution, $\hat{p}(y|x; W, b)$, as close as possible to the conditional distribution of the training data, $p(y|x)$. In the case that they are identical, Equation (5) yields the *Bayes-optimal* prediction rule, i.e. the classifier has minimal expected error on future data from the same distribution. However, if the data distribution at the prediction time, $q(x, y)$, differs from $p(x, y)$, then the classifier f can become suboptimal, i.e. have higher error rate than necessary. Unfortunately, the latter situation is the rule rather than the exception in the setting we study. For a time-varying data distribution, the data available at training time practically never is distributed identically to the data at prediction time.

There is no general solution for this problem of *distribution mismatch*, unless we assume that $p(x, y)$ and $q(x, y)$ are in some way related to each other. One such relatedness assumption, which is often justified in practice, is that the two distributions differ only by a *class-prior shift*, that is, $q(y) \neq p(y)$ but $q(x|y) = p(x|y)$ (Quiñonero-Candela et al. [2008]). In that case, it is possible to *adapt f* to a classifier that yields optimal predictions for the new situation. To see this, we first write

$$q(y|x) = \frac{q(x|y)q(y)}{q(x)} = \frac{p(x|y)q(y)}{q(x)} = p(y|x)\frac{q(y)}{p(y)}\frac{p(x)}{q(x)}, \tag{6}$$

where the first and third equality are Bayes' rule and the second equality follows from the class-prior shift assumption. Consequently, the optimal decisions for the distribution $q(x, y)$ is

$$\operatorname*{argmax}_{y \in \mathcal{Y}} q(y|x) = \operatorname*{argmax}_{y \in \mathcal{Y}} p(y|x)\frac{q(y)}{p(y)}, \tag{7}$$

where we have used the fact that the argmax is not affected by multiplication with factors that do not depend on y.

For a log-linear model parameterized as in Equation (4)

that approximates $p(y|x)$, we obtain the *adapted model*

$$\hat{q}(y|x; W', b') \propto \exp(w_y'^\top \phi(x) + b_y') \tag{8}$$

that approximates $q(y|x)$ by setting, for all $y \in \mathcal{Y}$,

$$w_y' = w_y, \quad b_y' = b_y + \log \frac{q(y)}{p(y)}, \tag{9}$$

as long as $q(y)$ and $p(y)$ are known, or can at least be estimated sufficiently well. We denote the classifier associated with the adapted model (8) as $f_{p \to q}$.

2.4 Lightweight Model Extrapolation for Streaming Data

The ability to adapt a learned model from one data distribution to another has been used previously, for example for *domain adaptation* (Quiñonero-Candela et al. [2008]) or to exploit changing class proportions at prediction time (Royer and Lampert [2015]).

In contrast to prior work, we use it at training time to learn a single set of parameters that will work well –after respective adaptation– under many different conditions. Conceptually, the step resembles meta-learning (Finn et al. [2017, 2019]), where also a base model is learned in a way such that it can be adapted easily to new situations. Specifically, we form a single log-linear model f_u with modeled conditional distribution as in Equation (4).

Instead of training this model to minimize the average loss across all time steps, we aim for the adapted models to result in minimal loss,

$$\min_{W,b} \quad \sum_{\tau=1}^{t} \hat{R}_\tau(f_{u \to p_\tau}), \tag{10}$$

where u denotes the distribution in which all classes have equal probability. Despite the similar formulation, learning a model using Equation (10) differs fundamentally from learning a model with Equation (1). For the latter, the terms, $\hat{R}_1(f), \ldots, \hat{R}_t(f)$, in the summation compete with each other: the parameters that optimally minimize one of them are not optimal for any of the others, when their respective training data is distributed differently. As a consequence, the optimization has to settle for finding parameters that perform well on average, but are never truly optimal. This effect is purely due to the time-varying nature of the data, so neither observing larger datasets nor more of them will overcome it.

In the newly proposed formulation (10) the terms, $\hat{R}_1(f_{u \to p_1}), \ldots, \hat{R}_t(f_{u \to p_t})$, support each other: up to finite sample effects, the parameters that minimize any of the terms $\hat{R}_\tau(f_{u \to p_\tau})$ for $\tau = 1, \ldots, t$, are identical, and they are also the same ones that would minimize the risk of the modeled f_u over a dataset sampled from u, i.e. with uniform class distribution. The reason is that the differences in class distributions are compensated by the corresponding adaptation step, so the model parameters do not have to change. As a consequence, the more datasets we observe or the larger they are, the higher the precision with which we can recover the optimal parameters for all time steps.

As in the case of empirical risk minimization, solving Equation (10) exactly is intractable because it would require us to store all past data. Instead, we again use incremental training: we update the parameters by always minimizing the most

Figure 1: Class distribution in `geo-tweets` dataset across 24 hours (March 1, 2020) for five most frequent countries (`US`: USA, `BR`: Brazil, `JP`: Japan, `GB`: Great Britain, `PH`: Philippines).

recent empirical risk

$$\min_{W,b} \quad \hat{R}_t(f_{u \to p_t}), \tag{11}$$

starting at the previous parameter values and using early stopping after a single pass through the data.

Forming $f_{u \to p_\tau}$ for any $\tau = 1, \dots, t$, requires knowledge of the class probabilities $p_\tau(y)$. These are not known exactly, but after having observed the training set S^τ, we can estimate them easily by the observed class frequencies, $\hat{p}_\tau(y) = \frac{1}{n_\tau} \sum_{i=1}^{n_\tau} [\![y_i^\tau = y]\!]$. Those are just L-dimensional vectors, so they can be stored, even continuously, with only a very small memory footprint.

The model f_u resulting from the above procedure can be expected to work well for data with a uniform class distribution, and also for any other class distribution if we adapt it suitably.

This property is particularly useful for constructing a model for the next time step, $t + 1$: instead of having to train on a data set $S^{t+1} \sim p_{t+1}$, which is not available yet, we can simply perform the very efficient adaptation step (8) to form $f_{u \to p_{t+1}}$. This, however, requires an estimate of the class probabilities at the following time step, $p_{t+1}(y)$, which are a priori unknown. To overcome this problem, we propose a history-based *forecasting step* that extrapolates the class distribution. First, we find the time step, t^*, among all earlier ones whose class distribution is most similar to the one at the current time step, t.

$$t^* = \operatorname*{argmin}_{\tau=1,\dots,t-1} \operatorname{dist}(\hat{p}_\tau(y), \hat{p}_t(y)), \tag{12}$$

where dist denotes the L^1-distance. We then use $\hat{p}_{t^*+1}$ as a proxy for $\hat{p}_{t+1}$. In other words, we forecast that the situation one time step after the current, $t + 1$, will be similar to the situation that had emerged one time step after a similar situation had occurred in the past, $t^* + 1$.

The combination of both steps, adaptation at training time and forecasting of the next class distribution, we call *LIMES* (Lightweight Model Extrapolation for Streaming Data). It has a number of desirable properties. First, it is easy to implement, as it requires neither a new model architecture nor a special form of optimization. Instead, one simply trains a model continuously on the incoming data, and only adjusts the model's bias terms at each new time step. *LIMES*'s computational and memory overheads are negligible, as only the vectors of prior class distributions have to be stored and searched. For very long-running systems, the forecasting step could be made even more efficient (time and memory complexity $O(1)$ instead of $O(t)$) by either using a rolling history buffer, or by selecting a finite number of prototypical histories in a data-dependent way. Such steps are orthogonal to our main contribution in this work, so we leave them to future work.

3   Experimental Evaluation

We evaluate *LIMES* on the `geo-tweets` dataset[2]. It consists of a large number of tweets with geographic information that were collected using Twitter's free Streaming API[3] from April to July 2020. The associated classification task is to predict from which of 250 countries a tweet was sent based on its textual content and/or the user metadata. Because

---

[2]`https://cvml.ist.ac.at/geo-tweets`
[3]`https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data`

users in different time zones are active at different times of the day, the country distribution fluctuates strongly across time. Figure 1 illustrates this effect: the x-axis shows the hours of a day in GMT time zone. Between hours 14 and 23 as well as 0 and 6, the most active countries on Twitter are the USA and Brazil. At their peak (hours 22-4), they together are responsible for more than half of all tweets in the dataset. In contrast, between hours 7-14, tweets from the USA and Brazil are less frequent, whereas Japan, Great Britain and the Philippines are responsible for larger shares.

To reflect the time-varying nature of the data, we split the data into hourly chunks, which on average contain approximately 127000 tweets (minimum: 61000, maximum: 149000). We use random subsets of 80% of each chunk for training and 20% for evaluation. In order to assess statistical variations, we subdivide the data further. First, we create two subsets, one containing tweets of the 1st to 5th day of each month (*"early"*) and one for the 11th to 15th (*"late"*). From each subset, we create ten realizations of the time series by subsampling the data with a stride of 10 from different starting indices.

From each tweet we extract two 512-dimensional feature vectors using the pretrained `distiluse-base-multilingual-cased-v1` multi-lingual sentence embedding network (Reimers and Gurevych [2019]). One is of the tweet text itself and one is of the text in the *location* meta-data, which is a user-provided free text that might or might not be related to the actual location of the user. As target labels, we use the tweet's country of origin, which for geo-tagged tweets can be extracted from the tweet's meta-data. Specifically, we infer the country either from the lat-long coordinates, or, if those are not provided, we use the user-specified *country_code* entry of the meta-data.

The input features can be extracted also for tweets without geo-tags. Therefore, the classifier resulting from the described training method can be used to predict the country of origin from any tweet, which is an often-required step in social media analysis (Han et al. [2014]). Note, however, that in this work, we are not trying to compete with existing methods for this specific task, as the Twitter data serves just as a real-world testbed for our proposed general-purpose model adaptation technique.

As an instantiation of *LIMES*, we implement a logistic regression model in the *keras* toolkit with *tensorflow* backend. The training is performed incrementally by Equation (10) using the Adam optimizer with mini-batches of size 100. No regularization is used and all hyperparameters are left at their defaults (learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$).

As baseline, we use alternative methods discussed in Section 2.2, relying on the same model architecture and optimization routines as for *LIMES*. *Incremental* is a classifier that is trained incrementally to minimize (1), i.e. identically to *LIMES* but without adaptation and forecasting. *Random* uses the same adaptation as *LIMES*, but using a randomly selected time step from history instead of the extrapolation. *Ensemble* is an ensemble of 24 classifiers, one per hour. Each classifier is trained incrementally on the data of the respective hour. For prediction at time step $t + 1$, the model with index $(t + 1)$ mod 24 is used. *Restart* trains a new model at every time step t from the data S^t and uses it for predictions at step $t + 1$.

All four methods are tested in 60 experimental settings: 2 subsets (*early*, *late*), 3 feature types (*tweet*, *location* or their concatenation), and 10 different data realization. In each case, we measure the accuracy of the resulting classifier. We then report average and standard deviation across the ten realizations, once for the average accuracy across all hours (*avg-of-avg*) and once for the average of the minimum accuracy within each day (*avg-of-min*). The latter value is often a quantity of interest for real-world systems, because customer satisfaction tends to depend not on the average of their experience but rather on the worst case.

3.1 Results

Table 1 summarizes the results. Overall, one can see that the task of predicting the country from the tweet text is more difficult than from the user-provided meta-data. Combining both feature types yields the highest accuracy. In all cases, the minimal accuracy across the day is substantially lower than the average. This indicates that the difficulty of the classification task varies over time, presumably due to changes in the label distribution.

Comparing the different methods, we observe a ranking that is stable across all experimental settings: *LIMES* achieves the best results, followed by *incremental* and *random*. *Ensemble* performs substantially worse, and *restart* even worse than that. Given that *incremental* differs from *LIMES* only in the lack of an adaptation and forecasting step, we can conclude that the way we propose the adaptation to the class distribution indeed has a positive effect. For the *avg-of-avg accuracy* measure, the gain due to adaptation and forecasting is the biggest for the most difficult task: 0.53–0.65% when predicting the country based on the tweet text itself. It gets smaller, the easier the task becomes: 0.13–0.19% for *location* features, and 0.07–0.12% for the concatenation of both. More apparent, however, is the positive effect on the *avg-of-min accuracy*: for *tweet* features, it is 2.64–3.00%, for *location* 0.71–0.83% and for their concatenation 0.53–0.60%.

Figure 2 provides an illustrative explanation of these results. Subfigure (a) shows the four methods' accuracy curves over the first 120 hours of one exemplary learning run (*tweet* features, subset *early*, realization 0). First, one can clearly

6

Table 1: Experimental Results: classifier accuracy of the proposed *LIMES* method and baselines as mean and standard deviation across 10 data splits for six experimental settings (see main text for details). Reported are the average per-day accuracy (*avg-of-avg*) and the minimum-across-the-day accuracy (*avg-of-min*). The difference of LIMES to the other methods is significant according to a Wilcoxon signed rank test at a below 0.5% level in all cases.

(a) features: *tweet*, subset *0–5*			(b) features: *location*, subset *0–5*			(c) features: *tweet–location*, subset *0–5*		
method	**avg**	**min**	**method**	**avg**	**min**	**method**	**avg**	**min**
LIMES	$52.35_{\pm 0.04}$	$43.15_{\pm 0.21}$	LIMES	$81.23_{\pm 0.03}$	$75.20_{\pm 0.07}$	LIMES	$85.54_{\pm 0.03}$	$79.92_{\pm 0.11}$
incremental	$51.81_{\pm 0.05}$	$40.51_{\pm 0.25}$	incremental	$81.09_{\pm 0.03}$	$74.49_{\pm 0.14}$	incremental	$85.48_{\pm 0.03}$	$79.40_{\pm 0.14}$
random	$48.85_{\pm 0.13}$	$35.13_{\pm 0.71}$	random	$79.72_{\pm 0.06}$	$71.84_{\pm 0.50}$	random	$84.13_{\pm 0.04}$	$76.80_{\pm 0.19}$
ensemble	$38.94_{\pm 0.03}$	$29.48_{\pm 0.18}$	ensemble	$68.48_{\pm 0.05}$	$62.09_{\pm 0.13}$	ensemble	$72.37_{\pm 0.04}$	$66.16_{\pm 0.15}$
restart	$34.28_{\pm 0.05}$	$24.62_{\pm 0.27}$	restart	$54.33_{\pm 0.05}$	$44.28_{\pm 0.17}$	restart	$58.36_{\pm 0.04}$	$48.08_{\pm 0.18}$

(d) features: *tweet*, subset *10–15*			(e) features: *location*, subset *10–15*			(f) features: *tweet–location*, subset *10–15*		
method	**avg**	**min**	**method**	**avg**	**min**	**method**	**avg**	**min**
LIMES	$52.15_{\pm 0.04}$	$42.90_{\pm 0.22}$	LIMES	$81.17_{\pm 0.04}$	$75.07_{\pm 0.16}$	LIMES	$85.48_{\pm 0.03}$	$79.74_{\pm 0.11}$
incremental	$51.51_{\pm 0.03}$	$39.90_{\pm 0.22}$	incremental	$80.98_{\pm 0.04}$	$74.24_{\pm 0.17}$	incremental	$85.36_{\pm 0.02}$	$79.15_{\pm 0.10}$
random	$48.63_{\pm 0.13}$	$34.79_{\pm 0.62}$	random	$79.62_{\pm 0.05}$	$71.79_{\pm 0.30}$	random	$84.09_{\pm 0.07}$	$76.64_{\pm 0.19}$
ensemble	$38.74_{\pm 0.05}$	$29.39_{\pm 0.16}$	ensemble	$68.18_{\pm 0.04}$	$61.30_{\pm 0.22}$	ensemble	$72.09_{\pm 0.05}$	$65.60_{\pm 0.25}$
restart	$33.81_{\pm 0.05}$	$24.21_{\pm 0.19}$	restart	$53.94_{\pm 0.09}$	$43.34_{\pm 0.39}$	restart	$57.73_{\pm 0.06}$	$46.53_{\pm 0.26}$

(a) Exemplary curve (realization 0) of per-hour accuracy for first 120 hours (*March 1st–5th, 2020*).

(b) Curve of average per-day accuracy over 20 days

(c) Curve of minimum-across-day accuracy over 20 days

Figure 2: Classifier accuracy over time for the proposed *LIMES* and baseline methods in one of the tested settings (*tweet* features, subset *early*).

see the 24-hour cycle by which the accuracies of all classifiers fluctuate. The accuracies are highest at around 2:00 GMT. As we know from Figure 1, this corresponds to the time the USA and Brazil are by far the most common countries from which tweets originate. The accuracy is the lowest around 9:00 GMT, which is the time when tweets come from several different countries in comparable amounts.

Second, one can also see that *ensemble*, which does not share information between different hours, starts working only after 24 hours and afterwards increases its prediction quality quite slowly. In contrast, the other models make reasonable predictions already after the first hour, because they transfer information between different hours.

Finally, one observes that *LIMES* is superior over *incremental* mostly in the *valleys* around 9:00 GMT, i.e. during the hours where the classification problem is the hardest. Presumably, this is because *incremental* learns parameters that are adapted to the average data distribution, and that is closer to the simple situation with two dominant countries than to the difficult situations where many countries are similarly likely.

Subfigures (b) and (c) make the difference between the methods more visible by suppressing the daily fluctuations. They show the accuracy values for all 20 days aggregated as the per-day-average and per-day-minimum, respectively. The line plots show mean of results over all ten realizations, whereas the error bars reflect the standard deviation. As expected, standard deviation is the biggest in the case of *random* method as for each time step, it draws a class distribution vector from historical values. This effect of randomness is alleviated when the average per-day is computed.

At subfigures (b) and (c), one can see the difference between *LIMES* and *incremental* well: it is small but consistent for the per-day-average accuracy, and bigger for the per-day-minimum accuracy.

Incremental training is clearly effective, as can be seen from the fact that the accuracy curves for the three incrementally trained methods, *LIMES*, *incremental* and *ensemble*, increase over time. In contrast, *restart*'s accuracy is approximately constant (up to the hourly fluctuations) over the runtime of the system.

4 Summary and Conclusions

In this work, we introduced *LIMES*, a lightweight method for continuous classifier training from streaming data with class-prior shift. Inspired by the success of recent meta-learning methods, we introduced a system that trains multiple models, each adapted to a specific class distribution, but requires only a single set of parameters, as each model is derived from a base model by analytic classifier adaptation. The adaptation parameters are extrapolated for the next data distribution based on the data observed so far. Our experiments on the large-scale `geo-tweets` dataset show that this process results in improved prediction quality compared to common baselines, especially when judged by the worst-case measure of the lowest accuracy across a day.

Despite the promising results, the problem of continual learning from data streams with time-varying distribution is far from solved. In particular, our results suggest that new methods are required to tackle the problem that classifiers for certain data distributions are harder to learn than others. This suggests that one might have to adapt not only the model parameters, but also the model architecture. We plan to address this aspect in future work.

References

Peter L Bartlett. Learning with a slowly changing distribution. In *Workshop on Computational Learning Theory (COLT)*, 1992.

Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. Web-scale Bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *International Conference on Machine Learing (ICML)*, 2010.

H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Conference on Knowledge Discovery and Data Mining (KDD)*, 2013.

Cheng Li, Yue Lu, Qiaozhu Mei, Dong Wang, and Sandeep Pandey. Click-through prediction for advertising in Twitter timeline. In *Conference on Knowledge Discovery and Data Mining (KDD)*, 2015.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

MG Sarwar Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. Machine learning at the network edge: A survey. *ACM Computing Surveys (CSUR)*, 54(8):1–37, 2021.

Christophe Giraud-Carrier. A note on the utility of incremental learning. *AI Communications*, 13(4):215–223, 2000.

Bartosz Krawczyk, Leandro L Minku, Joao Gama, Jerzy Stefanowski, and Michał Woźniak. Ensemble learning for data stream analysis: A survey. *Information Fusion*, 37:132–156, 2017.

W Nick Street and YongSeog Kim. A streaming ensemble algorithm (SEA) for large-scale classification. In *Conference on Knowledge Discovery and Data Mining (KDD)*, pages 377–382, 2001.

Leandro L Minku and Xin Yao. DDD: a new ensemble approach for dealing with concept drift. *IEEE Transactions on Knowledge and Data Engineering*, 24(4):619–633, 2011.

Michael D Muhlbaier and Robi Polikar. Multiple classifiers based incremental learning algorithm for learning in nonstationary environments. In *International Conference on Machine Learning and Cybernetics*, volume 6, 2007.

Rafael M.O. Cruz, Robert Sabourin, and George D.C. Cavalcanti. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41:195–216, 2018.

Paulo R.L. Almeida, Luiz S. Oliveira, Alceu S. Britto, and Robert Sabourin. Adapting dynamic classifier selection for concept drift. *Expert Systems with Applications*, 104:67–85, 2018.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCarl: Incremental classifier and representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(12):2935–2947, 2017.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. Incremental learning in online scenario. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural Computation*, 14(1):21–41, 2002.

Amelie Royer and Christoph H Lampert. Classifier adaptation at prediction time. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. MIT Press, 2008.

Chelsea Finn, P. Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learing (ICML)*, 2017.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International Conference on Machine Learing (ICML)*, 2019.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2019.

Bo Han, Paul Cook, and Timothy Baldwin. Text-based Twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 49:451–500, 2014.