Machine Learning API Shift Assessments: Change is Coming!

Anonymous Authors[1]

Abstract

A growing number of applications rely on machine learning (ML) prediction APIs. Model updates or retraining can change an ML API silently. This leads to a key challenge to API users, who are unaware of if and how the ML model has been changed. We take the first step towards the study of ML API shifts. We first evaluate the performance shifts from 2020 to 2021 of popular ML APIs from Amazon, Baidu, and Google on a variety of datasets. Interestingly, some API's predictions became notably worse for a certain class and better for another. Thus, we formulate the API shift assessment problem as estimating how the API model's confusion matrix changes over time when the data distribution is constant. Next, we propose MASA, a principled adaptive sampling algorithm to efficiently estimate confusion matrix shifts. Empirically, MASA can accurately estimate the confusion matrix shifts in commercial ML APIs with up to 77sampling. This paves the way for understanding and monitoring ML API shifts efficiently.

1. Introduction

More and more machine learning (ML) applications are deployed via ML prediction APIs. For example, one can use Amazon text Comprehend (Ama) to determine the polarity of a text review written by a customer. These APIs do not require collecting data or training one's own models, and thus have been gaining popularity (Chen et al., 2020).

The performance of those third-party ML APIs over time, though, is not permanent. ML API providers continuously collect new data or change their model architectures (Qi et al., 2020) to update their services, which may either help or hurt downstream applications' performance silently. For example, we observe a 1.1% overall accuracy drop of Amazon Comprehend API on the IMDB dataset in April 2021 compared to its evaluation in April 2020, as shown in Figure 1 (a) and (b). Moreover, change of the confusion matrix of an API is often more informative than overall accuracy alone. For example, as shown in Figure 1 (c), 5% accuracy drops for positive text messages, but 4% accuracy rises for negative texts. Those performance shifts are of serious concern due to potential disruptions to downstream tasks as well as consistency required for audits. Thus, a natural and important question is *how to precisely assess ML API performance shifts over time*.

Contributions. This work formalizes the problem of assessing API shifts as estimating the confusion matrix differences on the same data set. A straightforward approach is to compare the API's prediction on randomly sampled data. However, this may require a large number of API calls, prohibitively expensive as each API call costs a fee. To address this challenge, we propose MASA, a principled algorithm for ML API shift assessments. To estimate the shifts, MASA clusters the dataset first and then adaptively draws data from different clusters to query the API. MASA automates its sampling rate from different data clusters based on the uncertainty in the confusion matrix estimation. For example, it may query the ML API on more samples with the true label "negative" than "positive", if it is less sure about the estimated performance change on the former. MASA leads to a low computation and space cost as well as a fast estimation error rate, by employing an upper-confidence-bound approach to estimate the uncertainties.

Empirically, we observe that MASA substantially improves the quality of estimation for API shifts. In preliminary experiments on real world ML APIs, MASA's assessment error is often an order of magnitude smaller than that of standard uniform sampling with same sample size (see, e.g., Figure 1 (d)). To achieve the same tolerable estimation error, MASA can reduce the required sample size by up to 77%.

Related Work. **Distribution shifts in ML deployments:** Performance shifts in ML systems have been observed in many applications. Most of them are attributed to distribution shifts, such as covariate shifts (Shimodaira, 2000) or label shifts (Lipton et al., 2018). API shifts are orthogonal to distribution shifts: instead of attributing the performance shifts to data distribution changes, API shifts concern with ML model changes on the same dataset. To our knowledge, this is the first work to investigate ML API shifts.

Deploying and monitoring ML APIs: Several issues in deployed ML APIs have been studied, such as biases (Buolamwini & Gebru, 2018) and bugs (Ribeiro et al., 2020).

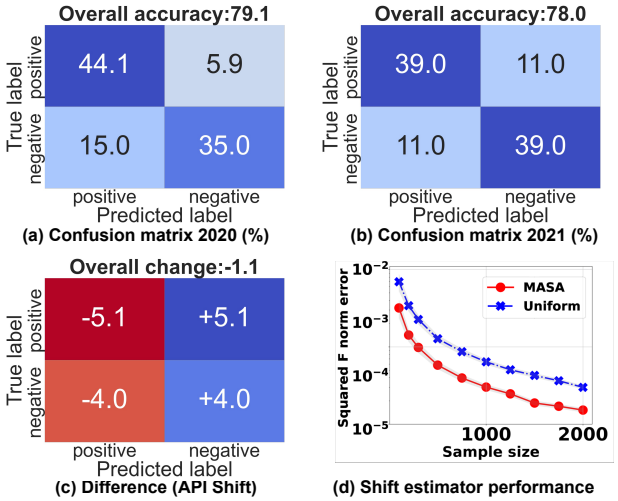

Figure 1. ML API shift for Amazon Comprehend API on IMDB, a sentiment analysis dataset. (a) and (b) give its (normalized) confusion matrix in April 2020 and 2021, respectively. Overall accuracy drops by 7%. This is because the 2021 model incorrectly predicts more "positive" texts as "negative", while less "negative" as "positive", as shown in (c). (d) Given a sample budget, the proposed MASA can assess the API shift with much smaller error in Frobenius norm compared to standard uniform sampling.

(Chen et al., 2020) considers the trade-offs between accuracy performance and cost via exploiting multiple APIs. On the other hand, the proposed MASA focuses on estimating (silent) API performance changes cheaply and accurately, which has not been studied before.

2. The API Shift Problem

Empirical assessment of ML API shifts. Let us start by an interesting observation: *Commercial ML APIs' performance can change substantial over time on the same datasets*. We evaluated three commercial ML APIs, Amazon Comprehend (Ama), Baidu AI (Bai) and Google NLP (GoN), on four standard sentiment analysis datasets, YELP, IMDB, WAIMAI, and SHOP. Figure 2 summarizes the accuracy changes.

A couple of interesting empirical findings exist. First, API performance changes are quite common. In fact, as shown in Figure 2, API performance changes exceeding 1% occurred in about 33% of all (twelve) considered ML API-dataset combinations. Since the data distribution remains fixed, such a change is due to ML APIs' updates. Second, API updates can lead to either overall accuracy increase or decrease, depending on the datasets. For example, as shown in first row of Figure 2, the Amazon Comprehend API's accuracy increases on YELP, WAIMAI, and SHOP, but decreases on

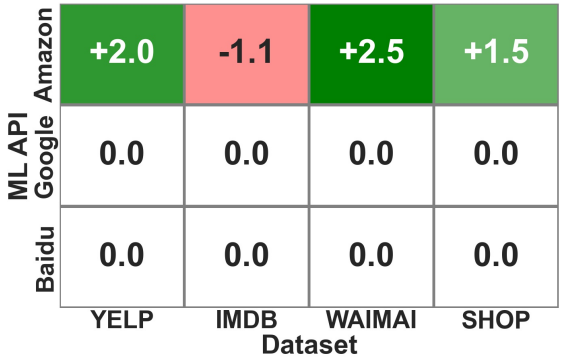

Figure 2. Observed overall accuracy changes (%). Each row corresponds to an ML API, and each column represents a dataset. The entry is the overall accuracy difference between evaluation in spring 2020 and spring 2021.

IMDB. It is also worth mentioning that the magnitude of the performance change can be quite different. In fact, the overall accuracy differences range from less than 1% to 3% across the cases we evaluated,.

Fine-grained assessment of API shift. Overall accuracy is often not enough. In fact, our discussion with practitioners revealed that attribution to per class change is often much more informative (Tsipras et al., 2020). One natural way to quantify an ML API's performance by its confusion matrix. Thus, we assess the change of the confusion matrix over time as a measure of API shift.

Formally, consider an ML service for a classification task with L labels. For a data point x from some domain $\mathcal{X}$, let $\hat{y}(x) \in [L]$ denote its predicted label on x, and $y(x)$ be the true label. The confusion matrix is denoted by $\boldsymbol{C} \in \mathbb{R}^{L\times L}$ where $\boldsymbol{C}_{i,j} \triangleq \Pr[y(x) = i, \hat{y}(x) = j]$. Given a confusion matrix of the ML API measured previously, $\boldsymbol{C}^o$, the ML API shift is defined as $\Delta\boldsymbol{C} \triangleq \boldsymbol{C} - \boldsymbol{C}^O$. Confusion matrix differences are strictly more informative than overall accuracy. For example, the trace of $\Delta\boldsymbol{C}$ is simply the overall accuracy. Moreover, it also explains which label gets harder or easier for the updated API.

3. MASA: ML API Shift Assessment

This section presents MASA, an algorithmic framework for efficient ML API shifts assessments. Suppose the old confusion matrix $\boldsymbol{C}^o$ and a large labeled dataset D are available. Given a query budget N, our goal is to generate $\Delta\hat{\boldsymbol{C}}$, an estimation of the API shifts as accurately as possible by querying the ML API $\hat{y}(\cdot)$ on N samples drawn from D.

MASA achieves its goal via an adaptive sampling approach (Figure 3). The dataset D is first divided into several partitions (clusters). Then MASA determines on which sample

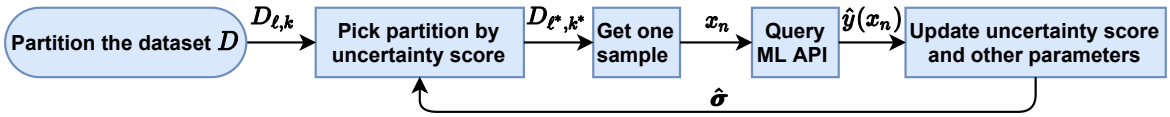

Figure 3. How MASA works. MASA first partitions the dataset. Then it picks which partition to sample based on an uncertainty measurement, queries the ML API on the drawn sample, and uses its prediction to update uncertainty and estimated shifts. This is repeated until the ML API has been queried N times. Finally, the estimated shifts on all partitions are fused to obtain the desired API shifts.

to query the ML API adaptively in an iterative manner: at each iteration, it selects one data partition based on some uncertainty measure, and queries the ML API on one sample randomly drawn from this partition. The API's prediction is obtained to update the uncertainty measure as well as the estimated shift $\Delta\hat{\boldsymbol{C}}$. This process is repeated until the ML API has been queried N times or if a stopping rule is reached. We explain each step in detail as follows.

Data Partitioning. Note that not all samples are equally informative for estimating API shifts. Consider, for example, a vision API makes perfect predictions on "dog" images, and guesses randomly on "cat" pictures. The "dog" images are less informative, as even a small sample of "dog" queries would tell that there is essentially no confusion for this class.

Thus, it is natural to partition all data points based on factors that may correlate with their informativeness, and sample from those partitions separately. In MASA, we use partitions $D_{i,k}$ that each contain the points with true label i and difficulty level k. The difficulty level is an integer indicating how hard it is to predict the data point's label. It needs not be perfect, and can be simply the discretized prediction confidence generated by some simple ML models. If the uncertainty or variability of the ML API's prediction on each partition is different, then drawing a different number of samples from each partition may improve the shift assessment performance compared to standard uniform sampling.

Budget Allocation Problem. Given the data partition, two questions arise: (i) how many samples should be drawn from each partition, and (ii) how to estimate the ML API shifts given available samples. The second question is relatively straightforward. We use standard maximum likelihood estimation for conditional accuracy $\Pr[\hat{y}(x) = j | x \in D_{i,k}]$ using all samples drawn from partition $D_{i,k}$. Then we can use their sum weighted by each partition's size to form an estimator of $\Pr[y(x) = i, \hat{y}(x) = j]$.

Now we consider the first question (budget allocation). As discussed in the last paragraph, more samples should be allocated to partitions with larger uncertainty. In fact, given a sample budget N, allocating $\boldsymbol{N}^*_{i,k} = \frac{\boldsymbol{p}_{i,k}\boldsymbol{\sigma}_{i,k}}{\sum_{i,k}\boldsymbol{p}_{i,k}\boldsymbol{\sigma}_{i,k}} N$ samples to partition $D_{i,k}$ minimizes $\mathbb{E}\left[\|\Delta\boldsymbol{C} - \Delta\hat{\boldsymbol{C}}\|_F^2\right]$, the squared Frobenius norm error of the estimated API shift. Here, $\boldsymbol{p}_{i,k} \triangleq \Pr[x_i \in D_{i,j}]$ denotes the (normalized) partition size. $\boldsymbol{\sigma}^2_{i,k} \triangleq (1 - \sum_{j=1}^{L} \Pr^2[\hat{y}(x) = j | x \in D_{i,k}])$ is the uncertainty score of $D_{i,k}$.

Uncertainty-Aware Adaptive Sampling. The optimal allocation replies on the uncertainty score of each partition. However, the uncertainty score is unknown and also requires estimation. Thus, MASA adopts an iterative process to allocate sample budget and update the uncertainty estimation adaptively. At each iteration, it first selects partition D_{i^*,k^*} to draw a sample via an upper-confidence-bound manner

$$(i^*, k^*) \leftarrow \arg\max_{i,k} \frac{\boldsymbol{p}_{i,k}}{\boldsymbol{N}_{i,k}} \left(\hat{\boldsymbol{\sigma}}_{i,k} + \sqrt[4]{\frac{a}{\boldsymbol{N}_{i,k}}}\right)$$

Here, for any partition $D_{i,k}$, $\boldsymbol{N}_{i,k}$ is the number of samples already drawn before this iteration, $\hat{\boldsymbol{\sigma}}_{i,k}$ is the current estimation of its uncertainty score, and $a > 0$ is a parameter to balance between exploiting and exploration. To obtain an initial value of $\hat{\boldsymbol{\sigma}}_{i,k}$, one can draw two initial samples from each partitions before using the above equation.

Given a drawn sample and the ML API's prediction, MASA then updates the estimation of the uncertainty score $\hat{\boldsymbol{\sigma}}$ as well as the estimated shifts. Naively applying their definition can be computationally expensive and also numerically unstable. Fortunately, we found an incremental update approach requiring constant space and computation cost, similar to online mean estimation (Cotton, 1975).

4. Preliminary Experiments

Now we give empirical evaluation of MASA on shifts estimation of several real world ML services for various tasks. Our goal is to (i) understand if and when MASA assess the API shifts efficiently, and (ii) examine how much sample cost MASA can reduce compared to standard sampling.

Tasks, ML APIs, and datasets. As a proof-of-concept, we focus on the sentiment analysis task. As shown in Section 2, we have observed four cases where >1% overall accuracy changes exist. Thus, we focus on evaluating MASA's

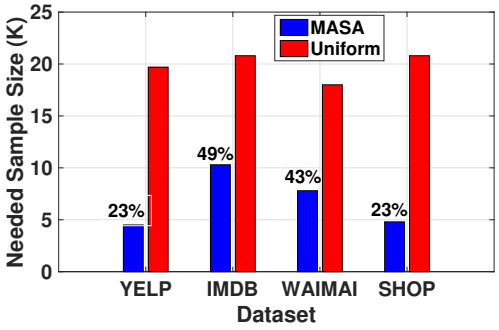

Figure 4. Required sample size to ensure (with high probability) less than 1% Frobenius estimation error of Amazon API's shift.

performance on those cases. All experiments were averaged over 1500 runs. We created partitions using difficulty levels induced by a cheap GitHub model with $K = 3$.

Budget savings achieved by MASA. In many applications, it is often enough to obtain an estimated API shift close to the true shift, e.g., within a 1% Frobenius norm error. Thus, a natural question arises: *to reach the same tolerable estimation error, how much sampling cost can* MASA *reduce compared to standard sampling approaches?*

To answer this question, we compare MASA with uniform sampling. For each approach, we measure the number of samples needed to reach 1% Frobenius norm error with probability 95%, via an upper bound on the estimated Frobenius error. As shown in Figure 4, MASA usually requires more than 70% fewer samples to reach such tolerable Frobenius norm error than the uniform and stratified sampling, primarily due to its shift estimation is more accurate.

Estimation error and query budget trade-offs. Then we study the trade-offs between API shift estimation error and sample size achieved by MASA, shown in Figure 5. Note that MASA consistently outperforms standard uniform sampling for any fixed sample size. In fact, the achieved estimation error of MASA is usually an order of magnitude smaller than that of uniform sampling. This verifies that MASA can provide more accurate assessments of API shifts in diverse applications.

5. Conclusion

This paper studies the problem of characterizing ML API shifts. Our preliminary empirical study shows that API model updates are frequent, and that some updates can reduce performance substantially. To assess API shifts, we propose an algorithmic framework, MASA, which provides significant estimation error and sample size reduction. We are working on in-depth theoretical analysis and comprehensive empirical evaluation. Extending our approach to more complicated ML tasks is another interesting open question.

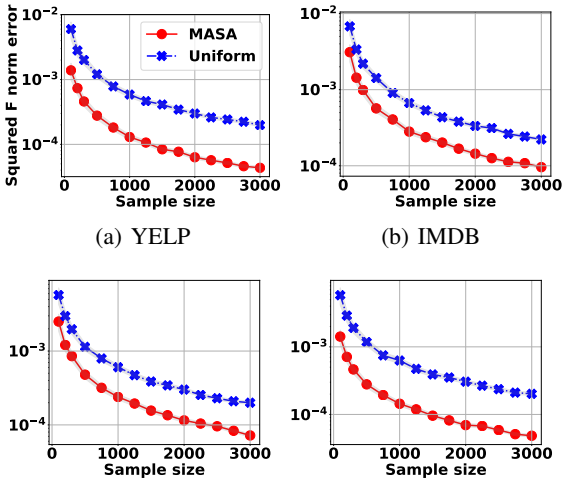

Figure 5. Amazon API shift estimation performance and sample size trade-offs. We compare the expected squared Frobenius norm error of MASA versus standard uniform sampling. For any sample size, MASA consistently leads to an estimation error much smaller than uniform sampling across different dataset.

References

Amazon Comprehend API. https://aws.amazon.com/comprehend. [Accessed March-2020 and March-2021].

Baidu API. https://ai.baidu.com/. [Accessed March-2020 and March-2021].

Google NLP API. https://cloud.google.com/natural-language. [Accessed March-2020 and March-2021].

Buolamwini, J. and Gebru, T. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *FAT*, 2018.

Chen, L. et al. FrugalML: How to use ML Prediction APIs more accurately and cheaply. In *NeurIPS*, 2020.

Cotton, I. W. Remark on stably updating mean and standard deviation of data. *Commun. ACM*, 18(8):458, 1975.

Lipton, Z. C. et al. Detecting and correcting for label shift with black box predictors. In *ICML*, 2018.

Qi, H. et al. Benchmarking intent detection for task-oriented dialog systems. *CoRR*, abs/2012.03929, 2020.

Ribeiro, M. T. et al. Beyond accuracy: Behavioral testing of NLP models with checklist. In *ACL*, 2020.

Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *JSPI*, 2000.

Tsipras, D. et al. From ImageNet to Image Classification: Contextualizing Progress on Benchmarks. In *ICML*, 2020.