Online Adaptive Metrics for Model Evaluation on Non-representative Offline Test Data

Enrico Piovano
Amazon.com, Inc
piovano@amazon.com

Dieu-Thu Le
Amazon.com, Inc
deule@amazon.com

Bei Chen
Amazon.com, Inc
chenbe@amazon.com

Melanie Bradford
Amazon.com, Inc
neunerm@amazon.com

***Abstract*—A major challenge encountered in the offline evaluation of machine learning models before being released to production is the discrepancy between the distributions of the offline test data and of the online data, due to, e.g., biased sampling scheme, data aging issues and occurrence(s) of regime shift. Consequently, the offline evaluation metrics often do not reflect the actual performance of the model online. In this paper, we propose online adaptive metrics, a computationally efficient method which re-weights the offline metrics based on calculating the joint distributions of the model hypothesis over the offline test data VS. the online data. It provides offline metrics which estimate the production performance of the model by taking into account the test data biases. The proposed method is demonstrated by real life examples on text classification and a commercial natural language understanding system. We show that the online adaptive metrics can provide accurate predictions of online recall and precision even with a small test dataset.**

I. Introduction

Machine Learning (ML) systems in production need to be frequently updated in order to add new functionalities and improve accuracy. These updates are often achieved by regularly releasing new models with additional functionalities and improved accuracy. Before a new model is deployed into production, the evaluation of its performance using offline test data is a key step to assess its potential improvement and to uncover possible degradation. Hence, it is critical that the offline test data are representative of the online live data in terms of distribution to reflect the real performance of the new model in production. However, it is often not the case in a real-world scenario.

A major challenge encountered in the offline evaluation of ML models before being released to production is the discrepancy between the offline and the online data. There are few main causes contribute to this issue: 1) Instead of random sampling, the offline data are collected using a biased sampling scheme. For example, Figure 1 shows a natural language understanding (NLU) application on the distribution of utterance intent counts: the offline data are sampled according to the Gaussian distribution while the online data are skewed and heavy-tailed. 2) Limited resource available to annotate a large volume of data and, consequently, to achieve sufficient test data size the annotation process take place over a long time frame, during which multiple seasonalities and regime shifts can occur.

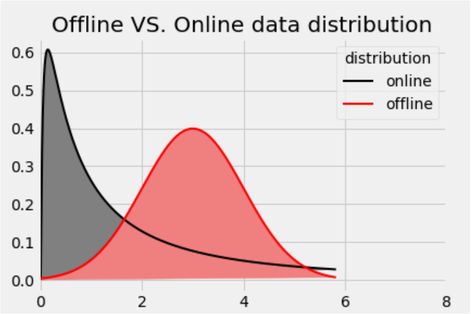

Fig. 1: Comparison of offline and online distribution of the utterance intent counts in a NLU application.

A. *Offline evaluation vs. live performance*

As a consequence of non-representative offline test sets, the offline evaluation does not always reflect the performance in production (aka. online performance). In this paper, we address the problem of predicting the production performance with a non-representative offline test data. Specifically, we compare a certain candidate model, which can be, for instance, the next one to be released in production, vs. a baseline model, which is the one currently in production. The proposed method is under the assumption that during the testing phase, we can pull a set of random data from the online live traffic with no ground truth. The joint distributions of the hypothesis of the candidate vs. baseline model for both the offline test set and the set of non-annotated utterances are then computed in form of two-dimensional matrices. The entry-wise difference between these two matrices is used as an indicator of the mismatch between the test data distribution and the live data distribution. This gap is then used to re-weight the offline metrics and align them to the production performance. Our approach is computationally very fast. It can be added to any offline test scheme as a post-calibration step for accuracy metrics without requiring any model architecture change. To demonstrate the proposed method, we consider two open-source text classification datasets and a commercial NLU system and we show that the proposed methodology allows to obtain online adaptive metrics values closer to the production performance of the models.

B. Related Work

There are multiple studies in the literature contributed to the task of tracking model online performance. They typically focus on detecting change point or drift in mean and variance [5, 9], for example, using reference windows together with statistical tests to monitor if there is significant drop in model live performance [16], scanning key evaluation metrics such as accuracy, recall and prevision over time [10], monitoring training distribution and controlling the trace of the online error of the underlying algorithm [7]. This line of studies aim to detect model drifts, but they do not deal with adjusting the offline metrics to better align with the model online performance.

Other relevant studies proposed algorithms to mitigate the issue of the mismatch between training and testing data [8, 15, 1]. For example, [17] presented three data splitting techniques, cross-validation, bootstrap and systematic sampling, to compare their performance on generalization in testing data. Another direction of research targets at removing biased in the training data. [2] proposed a theoretical framework for sample selection bias correction, in particular for cluster-based estimation and kernel mean matching. [11] described acquisition strategies for active testing [13] to form an ideal test set by removing bias while reducing the variance of the estimator [6]. The bias correction based methods require deriving weights of the sampled data and re-calculation of the model, which is computationally very expensive in the online operation of large scale models.

In this paper, we propose a novel approach to compute online adaptive metrics to predict production performance from poor and non-representative offline test sets. The main advantages of our approach lie in its generality, easy applicability and minimal computational cost.

II. Preliminaries

Let us consider the generic multi-class classification problem, where the K classification classes are given by the set $\mathbf{C} = \{C_1, C_2, ..., C_K\}$, for $C_k \in \mathbb{N}$ where $k \in \{1, ...K\}$. We denote the offline test data of length N, indicated with the superscript (o), as

$$\mathbf{D^{(o)}} = \{\mathbf{d_1^{(o)}}, \mathbf{d_2^{(o)}}, ..., \mathbf{d_N^{(o)}}\},$$

where $d_i^{(o)}$, $i \in \{1, ...N\}$ represents each data point in $\mathbf{D^{(o)}}$. Similarly, let us denote the sample of online data during the testing phase of length M, indicated with the superscript (l), as

$$\mathbf{D^{(l)}} = \{\mathbf{d_1^{(l)}}, \mathbf{d_2^{(l)}}, ..., \mathbf{d_M^{(l)}}\},$$

where $d_j^{(l)}$, $j \in \{1, ...M\}$ represents each data point in $\mathbf{D^{(l)}}$.

Suppose the current model in production is the baseline model B and we have a candidate model C with some improved features potentially to replace B. Now the goal is to compare C with the baseline model B in terms of accuracy defined by the evaluation metrics.

Let us denote the hypothesis of B over any data d as $H^{(B)}(d)$, the hypothesis of C as $H^{(C)}(d)$, the ground truth (aka. reference) as $G(d)$, where $H^{(B)}(d)$, $H^{(C)}(d)$ and $G(d) \in \mathbf{C}$. We assume that the ground truth is known for $\mathbf{D^{(o)}}$ but not for $\mathbf{D^{(l)}}$, which is a common case in practice.

A. Joint Distribution Matrices

We define here the joint distribution matrices which will be used in the rest of paper. First, we define the joint distribution of the hypotheses of the baseline model B and of the candidate model C over the offline test data $\mathbf{D^{(o)}}$ as the $K \times K$ matrix, $\mathbf{M}_{BC}^{(o)}$, whose $(i,j)^{th}$ entry is defined as

$$M_{BC}^{(o)}(i,j) = \frac{1}{N} |\forall d \in \mathbb{D}^{(o)} : H_B(d) = C_i, H_C(d) = C_j|$$

subject to $\sum_{i,j} \mathbf{M}_{BC}^{(o)}(i,j) = 1$, for $i,j \in \{1, ...K\}$.

Next, we define the joint distribution of the hypotheses of the baseline model B, of the candidate model C and of the ground truth G over the offline test data as the $K \times K \times K$ matrix, $\mathbf{M}_{BCG}^{(o)}$, whose $(i,j,k)^{th}$ entry is defined as

$$\begin{aligned}&\mathbf{M}_{BCG}^{(o)}(i,j,k)\\ &\quad = \tfrac{1}{N} |\forall d \in \mathbb{D}^{(o)} : H_B(d) = C_i, H_C(d) = C_j, G(d) = C_k|\end{aligned}$$

subject to $\sum_{i,j,k} \mathbf{M}_{BCG}^{(o)}(i,j,k) = 1$, for $i,j,k \in \{1, ...K\}$. Note that the following relationship between $\mathbf{M}_{BCG}^{(o)}$ and $\mathbf{M}_{BC}^{(o)}$ holds: f

$$\mathbf{M}_{BC}^{(o)}(i,j) = \sum_{k=1}^{K} \mathbf{M}_{BCG}^{(o)}(i,j,k),$$

for all $i,j \in \{1, ...K\}$.

Finally, we define the joint distribution of the hypotheses of the baseline model B and of the candidate model C over the live data $\mathbf{D^{(l)}}$ as the $K \times K$ matrix, $\mathbf{M}_{BC}^{(l)}$, whose $(i,j)^{th}$ entry is defined as

$$\mathbf{M}_{BC}^{(l)}(i,j) = \frac{1}{M} |\forall d \in \mathbb{D}^{(l)} : H_B(d) = C_i, H_C(d) = C_j|$$

subject to $\sum_{i,j} \mathbf{M}_{BC}^{(l)}(i,j) = 1$, for $i,j \in \{1, ...K\}$. Note that, as we assume here that M is large enough, $\mathbf{M}_{BC}^{(l)}$ well approximates the true joint distribution of the hypotheses of the B and C over the live data from the Glivenko–Cantelli Theorem [14].

Toy Example: To illustrate the above notations, we present a simple binary classification example where

$$\mathbf{C} = \{C_1, C_2\}.$$

Suppose that the joint distribution matrix over $\mathbf{D^{(o)}}$ is

$$\mathbf{M}_{BC}^{(o)} = \begin{bmatrix} 0.6 & 0.05 \\ 0.15 & 0.2 \end{bmatrix}.$$

That is, in the offline data, 60% is hypothesized as C_1 by both models B and C; 5% is hypothesized as C_1 by B and C_2 by C; 15% is hypothesized as C_2 by B and C_1 by C; finally 20% is hypothesized as C_2 by both B and C. Suppose the corresponding $\mathbf{M}_{BCG}^{(o)}$ matrix is given by $\mathbf{M}_{BCG}^{(o)}(:,:,C_1) = \begin{bmatrix} 0.2 & 0.02 \\ 0.03 & 0.1 \end{bmatrix}$ and $\mathbf{M}_{BCG}^{(o)}(:,:,C_2) = \begin{bmatrix} 0.4 & 0.03 \\ 0.12 & 0.1 \end{bmatrix}$. That is,

20% of $\mathbf{D}^{(\mathbf{o})}$ is hypothesized as C_1 by B and C and it has ground-truth C_1; 40% is hypothesized as C_1 by B and C and it has ground-truth C_2, and so on. Note that similar interpretation applies to $\mathbf{M}_{BC}^{(l)}$ over $\mathbf{D}^{(\mathbf{l})}$.

B. Evaluation Metrics

For each class $C_k \in \mathbb{N}$, $k \in \{1, ...K\}$, we define the **recall**s of the baseline model B and the candidate model C respectively as $R_B^{(\cdot)}(C_k)$ and $R_C^{(\cdot)}(C_k)$, the **precision**s of B and C respectively as $P_B^{(\cdot)}(C_k)$ and $P_C^{(\cdot)}(C_k)$, where the superscript $(\cdot)$ can be either (o) or (l), indicating that the metric is either for offline or online data. Furthermore, let us denote $A_B^{(\cdot)}$ as the accuracy of B and $A_C^{(\cdot)}$ as the accuracy of C.

The recall and precision can be written in the following way, using the joint distribution matrices described in Section II-A:

$$R_B^{(\cdot)}(C_k) = \frac{\sum_j \mathbf{M}_{BCG}^{(\cdot)}(k,j,k)}{\sum_{i,j} \mathbf{M}_{BCG}^{(\cdot)}(i,j,k)} \tag{1}$$

$$R_C^{(\cdot)}(C_k) = \frac{\sum_i \mathbf{M}_{BCG}^{(\cdot)}(i,k,k)}{\sum_{i,j} \mathbf{M}_{BCG}^{(\cdot)}(i,j,k)} \tag{2}$$

$$P_B^{(\cdot)}(C_k) = \frac{\sum_j \mathbf{M}_{BCG}^{(\cdot)}(k,j,k)}{\sum_j \mathbf{M}_{BC}^{(\cdot)}(k,j)} \tag{3}$$

$$P_C^{(\cdot)}(C_k) = \frac{\sum_i \mathbf{M}_{BCG}^{(\cdot)}(i,k,k)}{\sum_i \mathbf{M}_{BC}^{(\cdot)}(i,k)} \tag{4}$$

$$A_B^{(\cdot)} = \sum_{k,j} \mathbf{M}_{BCG}^{(\cdot)}(k,j,k) \tag{5}$$

$$A_C^{(\cdot)} = \sum_{k,i} \mathbf{M}_{BCG}^{(\cdot)}(i,k,k) \tag{6}$$

where the superscript $(\cdot) \in \{(o),(l)\}$, $C_k \in \mathbb{N}$, $k \in \{1, ...K\}$.

C. Offline Test Set Generation

As explained in Section I, an ideal test set should be generated by random sampling and annotating data from the current traffic in production. If the test set size would be sufficiently large, the test data distribution would match the live distribution, where the latter is approximated by the pulled non-annotated data distribution, and we would also have $\mathbf{M}_{BC}^{(o)} \approx \mathbf{M}_{BC}^{(l)}$. However, due to biased data sampling, seasonality and drift effects, the annotated test set has often a completely different data distribution compared to the live traffic.

Toy Example (continued): Assume that the live traffic data have the following joint distribution matrix: $\mathbf{M}_{BC}^{(l)} = \begin{bmatrix} 0.6 & 0.05 \\ 0.15 & 0.2 \end{bmatrix}$. Few factors could drive $\mathbf{M}_{BC}^{(l)}$ to deviate from $\mathbf{M}_{BC}^{(o)}$, e.g., biased sampling, seasonality and model drift. For example, C_1 is more critical for the use case where high precision is required. It is important that the confused utterances hypothesized as C_1 by the candidate model and as C_2 by the baseline model are in fact belonging to C_1. It is a common practice to draw a large fraction of those confused utterances for re-annotation to be added to the offline test set, which introduce sampling bias.

III. Proposed Methodology

The proposed method to generate online adaptive metrics is based on two main assumptions:

1) The joint distribution of B and C over the online data can be approximated by $\mathbf{M}_{BC}^{(l)}(i,j)$, $i,j \in \{1,...,K\}$. Note that we can pull an adequate number of randomly sampled data (of size M) from the online data in our problem setup, as mentioned in Section I and II (Glivenko–Cantelli Theorem [14]).
2) The conditional probability of the ground-truth G with respect the hypothesis H_B and H_C over the online data, i.e.,

$$p[G(d) = C_k | H_B(d) = C_i, H_C(d) = C_j],$$

can be well approximated by the same conditional probability over the offline data, which is given by

$$\mathbf{M}_{BCG}^{(o)}(i,j,k) / \mathbf{M}_{BC}^{(o)}(i,j),$$

for $C_k \in \mathbb{N}$ where $k \in \{1, ...K\}$.

Based on the assumptions above, we propose a re-weighting method to modify the **offline** recall, precision and accuracy as defined in equations (1)-(6) when $(\cdot)$ being (o), in order to minimize the bias in those metrics caused by the discrepancy in distribution between offline and online data. Due to assumption (1), we replace $\mathbf{M}_{BC}^{(o)}(i,j)$ with $\mathbf{M}_{BC}^{(l)}(i,j)$ in $P_B^{(o)}$ and $P_C^{(o)}$. Also, due to assumption (2), we have

$$\mathbf{M}_{BCG}^{(l)}(i,j,k) \approx \mathbf{M}_{BCG}^{(o)}(i,j,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(i,j)}{\mathbf{M}_{BC}^{(o)}(i,j)}, \tag{7}$$

so we replace $\mathbf{M}_{BCG}^{(o)}(i,j,k)$ with $\mathbf{M}_{BCG}^{(o)}(i,j,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(i,j)}{\mathbf{M}_{BC}^{(o)}(i,j)}$ in $R_B^{(o)}$, $R_C^{(o)}$, $P_B^{(o)}$, $P_C^{(o)}$, $A_B^{(o)}$ and $A_C^{(o)}$.

Specifically, the proposed Online Adaptive Metrics (OAM) for recall, precision and accuracy are then given by:

$$\hat{R}_B^{(l)}(C_k) = \frac{\sum_j \mathbf{M}_{BCG}^{(o)}(k,j,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(k,j)}{\mathbf{M}_{BC}^{(o)}(k,j)}}{\sum_{i,j} \mathbf{M}_{BCG}^{(o)}(i,j,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(i,j)}{\mathbf{M}_{BC}^{(o)}(i,j)}} \tag{8}$$

$$\hat{R}_C^{(l)}(C_k) = \frac{\sum_i \mathbf{M}_{BCG}^{(o)}(i,k,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(i,k)}{\mathbf{M}_{BC}^{(o)}(i,k)}}{\sum_{i,j} \mathbf{M}_{BCG}^{(o)}(i,j,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(i,j)}{\mathbf{M}_{BC}^{(o)}(i,j)}} \tag{9}$$

$$\hat{P}_B^{(l)}(C_k) = \frac{\sum_j \mathbf{M}_{BCG}^{(o)}(k,j,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(k,j)}{\mathbf{M}_{BC}^{(o)}(k,j)}}{\sum_j \mathbf{M}_{BC}^{(l)}(k,j)} \tag{10}$$

$$\hat{P}_C^{(l)}(C_k) = \frac{\sum_i \mathbf{M}_{BCG}^{(o)}(i,k,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(i,k)}{\mathbf{M}_{BC}^{(o)}(i,k)}}{\sum_i \mathbf{M}_{BC}^{(l)}(i,k)} \tag{11}$$

$$\hat{A}_B^{(l)} = \sum_{k,j} \mathbf{M}_{BCG}^{(o)}(k,j,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(k,j)}{\mathbf{M}_{BC}^{(o)}(k,j)} \tag{12}$$

$$\hat{A}_C^{(l)} = \sum_{i,k} \mathbf{M}_{BCG}^{(o)}(i,k,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(i,k)}{\mathbf{M}_{BC}^{(o)}(i,k)} \tag{13}$$

Given the offline data $\mathbf{D}^{(o)}$ and online data $\mathbf{D}^{(l)}$. The proposed algorithm is outlined in the following pseudo code.

Algorithm 1: Online Adaptive Metrics algorithm

```
Compute the following:
$\mathbf{M}_{BC}^{(o)} \leftarrow \frac{1}{N}|\forall d \in \mathbb{D}^{(o)} : H_B(d) = C_i, H_C(d) = C_j|$
$\mathbf{M}_{BCG}^{(o)} \leftarrow \frac{1}{N}|\forall d \in \mathbb{D}^{(o)} : H_B(d) = C_i, H_C(d) = C_j, G(d) = C_k|$
$\mathbf{M}_{BC}^{(l)} \leftarrow \frac{1}{M}|\forall d \in \mathbb{D}^{(l)} : H_B(d) = C_i, H_C(d) = C_j|$
In all equations for recall/precision/accuracy substitute (indicated as sobst.) the following matrices:
for $i \in \{1, ...K\}$ do
    for $j \in \{1, ...K\}$ do
        subst. $\mathbf{M}_{BC}^{(o)}(i,j)$ with $\mathbf{M}_{BC}^{(l)}(i,j)$
        subst. $\mathbf{M}_{BCG}^{(o)}(i,j,k)$ with $\mathbf{M}_{BCG}^{(o)}(i,j,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(i,j)}{\mathbf{M}_{BC}^{(o)}(i,j)}$
    end for
end for
```

The following lemma guarantees that the probabilities defined in the RHS of (7) sums to 1, hence the RHS of (7) is still a joint distribution and it can be used to compute the online adaptive metrics.

Lemma 1. *For $i,\ j,\ k \in 1, ...K$,*

$$\sum_{i,j,k} \mathbf{M}_{BCG}^{(o)}(i,j,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(i,j)}{\mathbf{M}_{BC}^{(o)}(i,j)} = 1.$$

Proof.

$$\begin{aligned} &\sum_{i,j,k} \mathbf{M}_{BCG}^{(o)}(i,j,k) \cdot \frac{\mathbf{M}_{BC}^{(l)}(i,j)}{\mathbf{M}_{BC}^{(o)}(i,j)} \\ &= \sum_{i,j} \frac{\mathbf{M}_{BC}^{(l)}(i,j)}{\mathbf{M}_{BC}^{(o)}(i,j)} \cdot \sum_k \mathbf{M}_{BCG}^{(o)}(i,j,k) \\ &= \sum_{i,j} \frac{\mathbf{M}_{BC}^{(l)}(i,j)}{\mathbf{M}_{BC}^{(o)}(i,j)} \cdot \mathbf{M}_{BC}^{(o)}(i,j) \\ &= \sum_{i,j} \mathbf{M}_{BC}^{(l)}(i,j) = 1 \end{aligned}$$

□

Next, we present the theorem which indicates that the adapted offline metrics $\hat{R}_B^{(l)}$, $\hat{R}_C^{(l)}$, $\hat{P}_B^{(l)}$, $\hat{P}_C^{(l)}$, $\hat{A}_B^{(l)}$ and $\hat{A}_C^{(l)}$ respectively converge to the online ones $R_B^{(l)}$, $R_C^{(l)}$, $P_B^{(l)}$, $P_C^{(l)}$, $A_B^{(l)}$ and $A_C^{(l)}$ in distribution. Suppose N is the offline sample size.

Theorem 2. *As $N \to \infty$, for $i,\ j,\ k \in 1, ...K$, we have that*

$$M_{BC}^{(l)}(i,j) \cdot \frac{M_{BCG}^{(o)}(i,j,k)}{M_{BC}^{(o)}(i,j)} \to M_{BCG}^{(l)}(i,j,k).$$

Proof. Here we present a sketch of the proof. Given that

$$\frac{M_{BCG}^{(l)}(i,j,k)}{M_{BC}^{(l)}(i,j)} = p(G = k | H_B = i, H_C = j)),$$

where $p(G = k|H_B = i, H_C = j)$ is the probability of the reference G to belong to the class k given the hypothesis of the models B and C being i and j. This probability can be estimated from $\mathbf{D}^o$ as

$$\hat{p}(G = k|H_B = i, H_C = j) = \frac{M_{BCG}^{(o)}(i,j,k)}{M_{BC}^{(o)}(i,j)}.$$

As $N \to \infty$, under randomly sampling, we have

$$\hat{p}(G = k|H_B = i, H_C = j) \to p(G = k|H_B = i, H_C = j).$$

Hence

$$\frac{M_{BCG}^{(o)}(i,j,k)}{M_{BC}^{(o)}(i,j)} \to \frac{M_{BCG}^{(l)}(i,j,k)}{M_{BC}^{(l)}(i,j)},$$

and the result follows. □

From Theorem 2 we obtain that, if the number of test data N is sufficiently large, the adapted offline metric will coincide with the online ones.

A. Why is the joint distribution needed ?

The online adaptive metrics are based on the entry-level difference between the offline and online joint distributions of the candidate vs. baseline model. An alternative way would be to adjust the metrics of each model by considering the difference between the offline and online distributions of its hypothesis independently. We will show in the experimental results that the proposed online adaptive method largely outperforms considering each model with independent distribution. In fact, by considering independent distributions, differences and biased sampling across the entries of the joint distributions are not captured. More specifically, the method is robust to a bias sampling which not only privileges some classes over others, but which privileges some of the entries of the joint distributions with respect to others. As explained over the paper, these are biased sampling methods widely utilized in real-world applications, for instance to better understand the impact of certain data which were hyphothesized in a class by the baseline and in another by the candidate, and the proposed method allows to cover them.

IV. Experimental Setup

In this section, we describe the experimental setup on three different datasets: two public text classification datasets and a commercial natural language understanding (NLU) system. The aim of our experiments is to quantify the impact of the online adaptive metrics on the offline non-representative test set. Our expectation is that the online adaptive metrics will be closer to the metrics evaluated on online data than the original offline metrics, without having access to the ground truth of the online data. To this end, we report the results on the online, offline metrics and the online adaptive metrics. Also, we provide the results considering independent model distributions explained in Section III-A. Specifically, we report the following metrics for both baseline and candidate models.

- **Online metrics**: Metrics evaluated on ground truth of online data.
- **Offline metrics**: Metrics evaluated on ground truth of offline data.
- **Online adaptive metrics with independent distribution (OAM IndDis)**: Reweighted offline metrics on online data using independent distribution.
- **Online adaptive metrics with joint distribution (OAM JointDis)**: Reweighted offline metrics on online data using joint distribution.

A. Text classification public datasets

We first apply the proposed method to two public datasets: (1) conference paper classification: which contains research article titles and the categories of these papers (the conferences' names) [1]. The task is to classify each research article to the correct conference based on the content of their titles. (2) news headlines classification: which contains news titles and the categories of these news articles [2]. The task is to classify each news headline to the correct type based on the content of the title.

We run two different text classification models on these datasets to act as the baseline model, which can correspond to the old model currently running in production, and the candidate model, which can correspond the newer model to be released into production. For a baseline model of the first dataset, we used a Multinomial Naive Bayes supervised classification [12] with TFIDF as features. For the second dataset, we apply GridSearch on SVM classification model [3]. As candidate model, we use a BERT transformer model [4] with a sequence classification head on top of the pooled outputs for both datasets.

We split both datasets into two sets: one used for training the models and the other used to simulate the overall online live data (so, the production traffic). After the splitting, the first dataset contains 627 samples for the overall online live traffic and the second dataset has 50214 samples. From the simulated live data of both datasets, we build the joint distributions of the hypothesis $\mathbf{M}_{BC}^{(l)}$. From the $\mathbf{M}_{BC}^{(l)}$ of each dataset, we random generate another joint distribution matrix which will correspond to the one of the offline test set $\mathbf{M}_{BC}^{(o)}$. From the first dataset we then random sample $N = 100$ data out of the 627 live data, and for the second dataset $N = 2500$ out the 50214 samples, according to the respective $\mathbf{M}_{BC}^{(o)}$, to generate the corresponding offline test set. As the first dataset is made of only 5 classes, we can plot here the online and offline joint distributions used in our experiment. To simplify the visualization, we plot them in percentage.

$$\mathbf{M}_{BC}^{(l)} = \begin{bmatrix} 13.18\% & 3.50\% & 0.16\% & 0.32\% & 0.48\% \\ 0.00\% & 32.36\% & 0.00\% & 0.48\% & 0.00\% \\ 0.32\% & 7.02\% & 3.29\% & 1.12\% & 0.32\% \\ 0.80\% & 5.42\% & 0.16\% & 14.45\% & 0.32\% \\ 1.59\% & 2.07\% & 0.00\% & 3.29\% & 9.35\% \end{bmatrix}$$

$$\mathbf{M}_{BC}^{(o)} = \begin{bmatrix} 11.00\% & 5.00\% & 0.00\% & 1.00\% & 2.00\% \\ 0.00\% & 8.00\% & 0.00\% & 2.00\% & 0.00\% \\ 1.00\% & 12.00\% & 6.00\% & 6.00\% & 1.00\% \\ 4.00\% & 10.00\% & 0.00\% & 8.00\% & 1.00\% \\ 2.00\% & 9.00\% & 0.00\% & 6.00\% & 5.00\% \end{bmatrix}$$

To better visualize the discrepancy between the two matrices, we show here the relative difference in percentage of the offline distribution compared to the online one. The result is:

$$\begin{bmatrix} -16.54\% & 42.86\% & -100.00\% & 212.50\% & 316.67\% \\ 0.00\% & -75.28\% & 0.00\% & 316.67\% & 0.00\% \\ 212.50\% & 70.94\% & 82.37\% & 435.71\% & 212.50\% \\ 400.00\% & 84.50\% & -100.00\% & -44.64\% & 212.50\% \\ 25.79\% & 334.78\% & 0.00\% & 82.37\% & -46.52\% \end{bmatrix}$$

The minimum number of online samples to accurately estimate the online joint distribution depends from the number of classes K and larger K requires more data. For instance, considering at least 250 and 15000 random samples from the live data of the first and second dataset, respectively.

The results for the text classification datasets are summarized in both Table I and Table II. For each model, we reported the online vs. offline evaluation metrics, our proposed OAM JointDis and OAM IndDis described in Section III-A. We used both accuracy (Acc) as well as precision and recall as metrics. Proving the precision and recall for each class would have not been feasible in terms of space given the large number of classes, hence we report the standard deviation across all classes between the real values of the recall/precision and the estimated ones by the utilized evaluation method.

As we can see in Table I and Table II, by applying the proposed approach, we obtain metrics which are much closer to the real value compared to before re-weighting or just applying re-weighting independently to each model for both datasets. We can also notice that applying a re-weighting approach considering the models independently and not the join distribution does not lead to closer metrics compared to the real value in general, as the biased sampling is made across the entries of the joint distribution and re-weighting on individual distributions cannot help in general.

[1] https://github.com/susanli2016/Machine-Learning-with-Python/blob/master/research_paper.csv

[2] https://www.kaggle.com/rmisra/news-category-dataset

Conference's Paper Classification				
Model	Setting	Acc.	std. Rec.	std. Prec.
Baseline (Multi. Naive Bayes)	Online metrics	0.727	0	0
	Offline metrics	0.510	0.192	0.219
	OAM IndDis	0.488	0.204	0.219
	OAM JointDis	**0.695**	**0.082**	**0.134**
Candidate (BERT)	Online metrics	0.761	0	0
	Offline metrics	0.620	0.234	0.111
	OAM IndDis	0.669	0.123	0.111
	OAM JointDis	**0.774**	**0.079**	**0.072**

TABLE I: Classification results for both baseline and candidate models on conference paper dataset.

News Classification				
Model	Setting	Acc.	std. Rec.	std. Prec.
Baseline (SVM Grid Search)	Online metrics	0.560	0	0
	Offline metrics	0.160	0.319	0.363
	OAM IndDis	0.111	0.330	0.363
	OAM JointDis	**0.526**	**0.096**	**0.148**
Candidate (BERT)	Online metrics	0.752	0	0
	Offline metrics	0.556	0.183	0.158
	OAM IndDis	0.569	0.197	0.158
	OAM JointDis	**0.700**	**0.160**	**0.043**

TABLE II: Classification results for both baseline and candidate models on news headline dataset.

NLUER	
Domain	Rel. Impro. Corr. Offl. Metr. and Onl. Meas. (%)
Music	**15.65%**
Video	**84.85%**
Friction	
Domain	Rel. Impro. Corr. Offl. Metr. and Onl. Meas. (%)
Music	**5.63%**
Video	**18.99%**

TABLE III: Relative improvement on the correlation between offline metrics and online measurements for a commercial dataset using the proposed re-weighting method

B. Evaluation on a commercial dataset

We then evaluate the online adaptive metrics on a real commercial dataset of a NLU system of a voice assistant, as Alexa, Google Assistant, Siri, etc. A typical NLU system consists of a domain classifier (e.g., music, video), a named entity recognized (e.g., song name, device) and an intent classifier (e.g., play music intent). Before a new NLU model is delivered in production, it is tested over anonymized test data, collected over time and often affected by biased sampling and drift effects. When the model is delivered in production, manual or automatic metrics are used for online evaluation. A manual metric, called NLUER (natural language understanding error rate), consists of sampling some random utterances, annotate them and, for each reference domain, calculate how many utterances within that domain have not been hypothesized correctly by the model. It is readily seen that (1-NLUER) is a recall-based metric. Another metric, instead automatic, is called friction and it computes, for each domain, how many utterances hyphothesized within that domain by the model have not worked end-to-end on the voice assistant. It is readily seen that (1-friction) is a precision-based metric. As the offline test set is often not representative of the customer traffic, it is expected a mismatch between offline and online metrics.

In this experiment, we show how the online adaptive evaluation method can overcome this problem by providing better matching metrics that show more correlations to online metrics. For the following evaluations, we collected offline performance evaluations of multiple model releases and map them to the online performance measurement for these releases. We calculate the correlations of the offline metrics with the online measurements, both before and after re-weighting, using the Pearson correlation coefficient. We report the correlations on the main domains: Music and Video. We show in Table III the relative improvement in percentage in correlation between offline and online metrics by utilizing the online adaptive evaluation method compared to considering the offline test set. Overall, for both online metrics, we see that the proposed method gives a much better indication of the online performance compared to the original offline metrics.

V. Conclusions

In this paper we target at solving the problem that the model performance evaluated on offline test data does not well reflect its online performance due to the discrepancy between the offline and online data in distribution. Instead of re-collecting the offline test data and re-training the model, we address the problem with a computationally light online adaptive re-weighting approach. In particular, we propose to post calibrate the offline evaluation metrics using the weights computed from the ratio of offline and online joint distribution matrices. Experimental results on both public datasets and a commercial dataset have shown that our proposed online adaptive metrics could align better to online metrics than the original accuracy, precision and recall offline metrics. We plan to establish a theoretical framework on bias correction of offline model evaluation metrics in our future work.

References

[1] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 81–88, New York, NY, USA, 2007. Association for Computing Machinery.

[2] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. Sample selection bias correction theory. *CoRR*, abs/0805.2775, 2008.

[3] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[5] A. Dries and U. Rückert. Adaptive concept drift detection. *Stat. Anal. Data Min.*, 2:311–327, 2009.

[6] Sebastian Farquhar, Yarin Gal, and Tom Rainforth. On statistical bias in active learning: How and when to fix it, 2021.

[7] João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In Ana L. C. Bazzan and Sofiane Labidi, editors, *Advances in Artificial Intelligence – SBIA 2004*, pages 286–295, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[8] C. Gonz. Optimal data distributions in machine learning. 2015.

[9] R. Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intell. Data Anal.*, 8:281–300, 2004.

[10] R. Klinkenberg and T. Joachims. Detecting concept drift with support vector machines. In *ICML*, 2000.

[11] Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation, 2021.

[12] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, USA, 2008.

[13] Phuc Nguyen, Deva Ramanan, and Charless Fowlkes. Active testing: An efficient and robust framework for estimating accuracy, 2018.

[14] Edwin JG Pitman. *Some basic theory for statistical inference: Monographs on applied probability and statistics*. Chapman and Hall/CRC, 2018.

[15] S. Sivasankaran, E. Vincent, and I. Illina. Discriminative importance weighting of augmented training data for acoustic model training. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4885–4889, 2017.

[16] Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of Knowledge Discovery and Data Mining (KDD)*, pages 226–235, 2003.

[17] Yun Xu. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing*, 2018.