Online Active Model Selection for Pre-trained Classifiers

Mohammad Reza Karimi* **Nezihe Merve Gürel*** **Bojan Karlaš**
Johannes Rausch **Ce Zhang** **Andreas Krause**
ETH Zürich

Abstract

Given k pre-trained classifiers and a stream of unlabeled data examples, how can we actively decide when to query a label so that we can distinguish the best model from the rest while making a small number of queries? Answering this question has a profound impact on a range of practical scenarios. In this work, we design an online selective sampling approach that actively selects informative examples to label and outputs the best model with high probability at any round. Our algorithm can be used for online prediction tasks for both adversarial and stochastic streams. We establish several theoretical guarantees for our algorithm and extensively demonstrate its effectiveness in our experimental studies.

1 INTRODUCTION

Model selection from a set of pre-trained models is an emerging problem in machine learning and has implications in several practical scenarios. Industrial examples include cases in which a telecommunication company or a flight booking company has multiple ML models trained over different sliding windows of data and hopes to pick the one that performs the best on a given day. For many real-world problems, unlabeled data is abundant and can be inexpensively collected, while labels are expensive to acquire and require human expertise. Consequently, there is a need to robustly identify the best model under limited labeling resources. Similarly, one often needs reasonable predictions for the unlabeled data while keeping the labeling budget low.

Depending on the data availability, one can consider two settings: (i) the *pool-based* setting assumes that the learner has access to a pool of unlabeled data, and she can select informative data samples from the pool to achieve her task, and (ii) the *online* setting assumes the data is arriving one example at a time (i.e., in a stream), and the learner decides to ask for the example's label on the go or to just throw it away. While offering fewer options on which data to label, this setting alleviates the scalability challenge of storing and processing a large pool of examples in the pool-based setting.

Another important aspect is the nature of the data: the instance-label pairs might be sampled *i.i.d.* from a fixed distribution, or chosen *adversarially* by an adversary. While sometimes the i.i.d. assumption is reasonable, there are practical scenarios where this assumption fails to hold. These include cases where there are temporal or spatial dependencies or non-stationarities in the dataset/stream. In these situations, it may be safer not to make assumptions on the data and rather consider worst-case data streams.

Contributions We develop a novel, principled and efficient model selection approach –Model Picker– for the online setting. Our query strategy is randomized and leverages hypothetical query answers to decide which data examples are likely to be informative for identifying the best model. We prove that our algorithm has no regret for adversarial streams, i.e., its performance for sequential label prediction is close to the best model for that stream in hindsight. Our bounds match (up to a constant) those of existing online algorithms that have access to all labels. We also establish bounds on the number of label queries and the quality of the output model of Model Picker. We furthermore conduct extensive experiments, comparing our algorithm with a range of other methods. To reach the same accuracy, competing methods can often require up to 2.5× more labels. Apart from the relative performance, on the ImageNet dataset, Model Picker requires a mere 13% labeled instances to select the best among 102 pre-trained models with 90% confidence, while having up to 1.3× lower regret. These results establish Model Picker as the state-of-the-art for this problem. We also make everything open and

*Equal contribution. Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

reproducible.[1]

2 RELATED WORKS

Our approach relates to several bodies of literature. For each related area, we reference similar works that match the objective of our paper.

Active Model Selection Madani et al. (2004) develop their method for the online setting. They seek to identify the best model via probing models, one at a time, with i.i.d. samples, while having a fixed budget for the number of probes. In contrast, our approach applies even to adversarial streams and allows one to make predictions online, while minimizing the number of queries made. Most of other previous works (Sawade et al., 2012; Gardner et al., 2015; Ali et al., 2014; Sawade et al., 2010; Katariya et al., 2012; Kumar and Raj, 2018; Leite and Brazdil, 2010) focus on pool-based sampling of informative instances, where the learner ranks the entire pool of unlabeled data and greedily selects the most informative examples. This setting substantially differs from the streaming setting, and we focus on the latter for reasons of scalability and applicability to many real-world situations.

Active Learning Active learning aims to query the label of those instances that help improving the *training* of classifiers, rather than selecting among pre-trained models. Here we review those methods that can potentially be adapted for model selection. The celebrated query-by-committee (QBC) paradigm (Seung et al., 1992) forms a committee of classifiers to vote on the labeling of incoming examples. The query decision is made based on the degree of disagreement among the committee members. The general strategy is to query those instances that help the learner prune the committee and only keep those classifiers with higher accuracies. There are other QBC approaches in active learning, such as Cohn et al. (1994); McCallum and Nigam (1998); Abe and Mamitsuka (1998); Melville and Mooney (2004); Settles and Craven (2008); Zhu et al. (2007). One limitation of these algorithms is that they often focus on pool-based sampling, which limits their scalability. Several other approaches consider active learning in the streaming setting. The seminal works of Dasgupta et al. (2008) and Balcan et al. (2009), followed by Beygelzimer et al. (2010); Zhang and Chaudhuri (2014), use disagreement-based strategies. The idea of using importance weights in active learning is studied by a series of works including Beygelzimer et al. (2008); Sugiyama (2006); Beygelzimer et al. (2011); Bach (2007), where importance weights are introduced to correct sampling bias and provide statistically consistent convergence to the optimal classifier in the PAC learning setting. All of the above approaches on stream-based active learning focus on i.i.d. streams and try to improve the supervised training of classifiers, whereas our approach applies to the more general adversarial streams and performs no training.

Online Learning and Bandits Sequential label prediction is an important problem in online learning. The setting closest to ours is *label-efficient prediction* (LEP) (Cesa-Bianchi et al., 2005), where they query the label with a *fixed* probability at each round, and that probability also appears in the regret bound. However, we use the side information of the models predictions to *adapt the probability* of querying to the information content of the instance at hand, thereby significantly reducing the required labels in practice and lowering the regret, as demonstrated theoretically and in our experiments. Moreover, there is no study of the quality of the model outputted at the end of the stream, for neither adversarial nor stochastic streams. Another problem similar to ours is *consistent online learning* (Karimi et al., 2019; Altschuler and Talwar, 2018), where the learner seeks to minimize the number of switches of her actions, while observing the loss every round, even if she does not update her strategy. In our setting, however, we do *not* know the loss in the rounds we do not query. Similar challenges arise in the multi-armed bandit literature. In a way, our setup lies between the usual prediction with experts advice and multi-armed bandit problems. Our algorithm is related in spirit to the EXP3 algorithm (Auer et al., 2002) for adversarial bandits. The key difference is that EXP3 uses the probability of selecting an arm to construct an unbiased loss estimator, whereas we consider the probability of observing the *whole* loss. While similar in spirit, the standard EXP3 analysis fails to yield a regret bound, as discussed in the footnote of page 4.

3 PROBLEM STATEMENT AND BACKGROUND

Assume that we have k pre-trained classifiers (*experts*). Let $\mathcal{X}$ and $\mathcal{C}$ be the set of all possible inputs and classes, respectively. Our sequential prediction problem is a game played in rounds. Consider a stream of data $\{(x_t, c_t) \in \mathcal{X} \times \mathcal{C}\}_{t \geq 1}$ generated by an unknown mechanism. At round t, x_t together with all classifiers predictions $\boldsymbol{p}_t \in \mathcal{C}^k$ is revealed to the learner. She then selects one of the experts $I_t \in [k]$[2] and incurs a loss of 1 if that expert misclassifies x_t. Finally, the learner decides whether to query the label c_t. If no query is made,

[1] The code is available at https://github.com/DS3Lab/online-active-model-selection

[2] In here and what follows, $[k] = \{1, \ldots, k\}$.

then c_t remains hidden, otherwise, the learner observes c_t and the loss $\boldsymbol{\ell}_t \in \{0,1\}^k$ defined as $\boldsymbol{\ell}_{t,i} = \mathbb{I}_{\{\boldsymbol{p}_{t,i} \neq c_t\}}$, with $\mathbb{I}_{\{\cdot\}}$ being the indicator function. Note that I_t can only depend on the past inputs and the observed labels. The goal of the learner is to select I_t in such a way that up to any round T, the total misclassifications she makes is close to the total mistakes of the best expert up to time T in hindsight. This performance measure is formalized as the *regret* of the learner:

$$\mathcal{R}_T = \sum_{t=1}^T \boldsymbol{\ell}_{t,I_t} - \min_{i\in[k]} \sum_{t=1}^T \boldsymbol{\ell}_{t,i}.$$

A prediction strategy satisfying $\limsup_{t\to\infty} \mathcal{R}_t/t \leq 0$, is called a *no-regret* algorithm.

If the stream is generated by sampling (x_t, c_t) i.i.d. from a fixed distribution, it is called a *stochastic stream*, otherwise we call it *adversarial*, as if an oblivious adversary has chosen the stream for the learner. It is known (Hazan, 2019) that if the learner follows a deterministic strategy, she can be forced by the adversary to have linear regret. Hence, the learner should randomize and select $I_t \sim \boldsymbol{w}_t$, where $\boldsymbol{w}_t$ is some distribution over the experts, reflecting how good the learner thinks the experts are at round t. In this case, we are interested in the *expected regret* $\mathbb{E}[\mathcal{R}_T]$, where the expectation is w.r.t. the (possible) randomness in the stream, as well as the randomness of the learner. By the tower property of expectation, $\mathbb{E}[\boldsymbol{\ell}_{t,I_t}] = \mathbb{E}[\mathbb{E}[\boldsymbol{\ell}_{t,I_t} \mid \boldsymbol{w}_t, \boldsymbol{\ell}_t]] = \mathbb{E}\langle \boldsymbol{w}_t, \boldsymbol{\ell}_t\rangle$, and hence, we could write

$$\mathbb{E}[\mathcal{R}_T] = \mathbb{E}\left[\sum_{t=1}^T \langle \boldsymbol{w}_t, \boldsymbol{\ell}_t\rangle - \min_{i\in[k]} \sum_{t=1}^T \boldsymbol{\ell}_{t,i}\right].$$

On top of the preceding task, it is often desirable that at each round t, the learner *recommends* (or outputs) an expert π_t as the best expert so far. This recommendation is suited for model selection tasks, where one needs not only the predictions per round, but also a recommendation about which classifier is the best one. We measure the quality of π_t in two ways: the probability of returning the true best model of the stream so far (*identification probability*), and the gap between the accuracy of the recommended model and the best one (*accuracy gap*). The choice of measure depends on the application: if one is interested only in identifying the best model, then the first measure, and if one just cares about getting a model that has an accuracy close to the best classifier, then the second measure is more relevant.

4 ALGORITHM AND ANALYSIS

In this section we set up the notation and present the Model Picker algorithm, along with several theoretical results regarding its performance.

Algorithm 1: Model Picker

Set $\widehat{\boldsymbol{L}}_{0,i} = 0$ for all $i \in [k]$
for $t = 1, 2, \ldots$ **do**
  $\eta_t \coloneqq \sqrt{(\log k)/(2t)}$
  Compute the distribution $\boldsymbol{w}_t$ over models, with $\boldsymbol{w}_{t,i} \propto \exp\{-\eta_t \widehat{\boldsymbol{L}}_{t-1,i}\}$
  Get predictions $\boldsymbol{p}_t$ of models for the observed data instance x_t
  Recommend $\pi_t := \arg\max_{i\in[k]} \boldsymbol{w}_{t,i}$ as the best model up to round t
  Sample $I_t \sim \boldsymbol{w}_t$ and output $\boldsymbol{p}_{t,I_t}$ as the predicted label for this instance
  Compute q_t as in (1) and sample $Q_t \sim \mathrm{Ber}(q_t)$
  if $Q_t = 1$ **then**
    Query the label c_t
    $\widehat{\boldsymbol{L}}_{t,i} = \widehat{\boldsymbol{L}}_{t-1,i} + \frac{1}{q_t}\mathbb{I}_{\{\boldsymbol{p}_{t,i}\neq c_t\}}$, $\forall i \in [k]$
  else
    $\widehat{\boldsymbol{L}}_{t,i} = \widehat{\boldsymbol{L}}_{t-1,i}, \quad \forall i \in [k]$
  end
end

4.1 The Algorithm

At any round t, our algorithm, based on the predictions $\boldsymbol{p}_t$ and current distribution $\boldsymbol{w}_t$ decides to query the label with probability q_t (to be determined later). Let $Q_t \sim \mathrm{Ber}(q_t)$ be the indicator of querying. Our algorithm then constructs a *loss estimate* $\hat{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t/q_t \cdot Q_t$. With this trick, we can think that the learner observes the loss sequence $\{\hat{\boldsymbol{\ell}}_t\}_{t\geq 1}$. We then construct $\boldsymbol{w}_t$ similar to the Exponential Weights (EW) algorithm (Littlestone and Warmuth, 1994) for this loss estimate sequence and with decaying learning rates $\{\eta_t\}$. The detailed algorithm is depicted in Algorithm 1. In what follows, we also set $\widehat{\boldsymbol{L}}_t = \sum_{s\leq t} \hat{\boldsymbol{\ell}}_s$ and $\boldsymbol{L}_t = \sum_{s\leq t} \boldsymbol{\ell}_s$.

Query Probability Instead of observing c_t with a constant probability (as done by Cesa-Bianchi et al. (2005)), we *adaptively* set this probability according to the predictions $\boldsymbol{p}_t$ and our current distribution over the experts $\boldsymbol{w}_t$. Notice that, based on the predictions, we know that the true loss vector $\boldsymbol{\ell}_t$ is among $\{\boldsymbol{\ell}_t^c : c \in \mathcal{C}\}$, where $\boldsymbol{\ell}_t^c$ is the hypothetical loss vector if the true label was c, i.e., $\boldsymbol{\ell}_{t,i}^c = \mathbb{I}_{\{\boldsymbol{p}_{t,i}\neq c\}}$. We define

$$\begin{aligned} v(\boldsymbol{p}_t, \boldsymbol{w}_t) &= \max_{c\in\mathcal{C}} \operatorname*{Var}_{J\sim\boldsymbol{w}_t} \boldsymbol{\ell}_{t,J}^c \\ &= \max_{c\in\mathcal{C}} \langle \boldsymbol{w}_t, \boldsymbol{\ell}_t^c\rangle(1 - \langle \boldsymbol{w}_t, \boldsymbol{\ell}_t^c\rangle) \end{aligned}$$

to be the maximum possible variance among different possible losses w.r.t. the distribution $\boldsymbol{w}_t$, and we set

$$q_t = \begin{cases} \max\{v(\boldsymbol{p}_t, \boldsymbol{w}_t), \eta_t\} & \text{if } v(\boldsymbol{p}_t, \boldsymbol{w}_t) \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

When $v(\boldsymbol{p}_t, \boldsymbol{w}_t)$ is nonzero, as seen above, we utilize a lower bound on q_t to prevent unboundedness issues. This lower bound, however, decreases over time.

The intuition behind the definition of $v(\boldsymbol{p}_t, \boldsymbol{w}_t)$ is as follows. Hypothetically, if the true label is c and we observe it, the distribution $\boldsymbol{w}_t$ over the models would be updated to $\boldsymbol{w}_{t+1}$ according to the loss $\boldsymbol{\ell}_t^c$. If we miss this update, as shown in Appendix A.1, the amount of regret we accumulate (due to not updating $\boldsymbol{w}_t$ to $\boldsymbol{w}_{t+1}$) is proportional to the variance of $\boldsymbol{\ell}_t^c$. Hence, the maximum variance among all hypothetical losses is a measure of the importance of the instance at hand and we use this value in our query probability. Note that if all models make the same prediction, observing the true label has no effect on the regret, and this behaviour is also reflected in (1), as q_t would be equal to zero in this case.

In what follows, we first tackle the general case of adversarial streams and prove bounds on regret, number of queries, and accuracy gap of Model Picker. We then strengthen our results for the stochastic setting and give improved bounds as well as a bound for the identification probability. All omitted proofs can be found in Appendix A.

Notation In what follows we define the conditional expectation $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot \mid \mathcal{F}_{t-1}]$, where $\mathcal{F}_{t-1}$ is the σ-algebra generated by all the random variables up to and including time $t-1$. Moreover, denote by $a \vee b := \max\{a, b\}$. We use $\langle \cdot, \cdot \rangle$ to denote the inner product of vectors. For a label $c \in \mathcal{C}$, we set $\boldsymbol{w}_{t,c} := \langle \boldsymbol{w}_t, \boldsymbol{\ell}_t^c \rangle$.

4.2 Guarantees for Adversarial Streams

We first prove that our algorithm has no regret. It is known (Cesa-Bianchi et al., 1997) that the regret of any online algorithm that observes all of the labels is at least $\Omega(\sqrt{T \log k})$. Our regret bound matches this lower bound, even though we do not see all the labels. Compared to LEP, our regret bound is smaller: they prove that for a fixed query probability ε, the regret is bounded by $\sqrt{2T \log k/\varepsilon}$, and for getting a regret of $O(\sqrt{T \log k})$ one has to set ε to be a constant. This forces the number of queries to be linear in T. However, there are no additional terms in our regret bound, as the probability of querying is adapted to the stream.

Theorem 1 (Regret)**.** *For adversarial streams, the expected regret of Algorithm 1 is bounded above by*

$$\mathbb{E}[\mathcal{R}_T] \leq 2\sqrt{2T \log k}.$$

Proof. We bring a few important observations that help us in the proof. Observe that we can remove those rounds where $\boldsymbol{\ell}_{t,i} = 1, \forall i \in [k]$, since expert I_t and the best expert in hindsight make the same mistake at round t. In the remaining rounds, $\hat{\boldsymbol{\ell}}_t$ has the same conditional expectation as $\boldsymbol{\ell}_t$:

$$\mathbb{E}_t[\hat{\boldsymbol{\ell}}_t] \overset{\text{(a)}}{=} \mathbb{E}_t[\mathbb{E}_t[\hat{\boldsymbol{\ell}}_t \mid \boldsymbol{p}_t, \boldsymbol{\ell}_t]] \overset{\text{(b)}}{=} \mathbb{E}_t[\tfrac{\boldsymbol{\ell}_t}{q_t} \mathbb{E}_t[Q_t \mid \boldsymbol{p}_t, \boldsymbol{\ell}_t]] = \boldsymbol{\ell}_t,$$

where (a) is by the tower property of expectation and (b) is by the definition of $\hat{\boldsymbol{\ell}}_t$ and the fact that q_t is $\sigma(\boldsymbol{p}_t, \mathcal{F}_{t-1})$-measurable. This, together with $\mathbb{E}[\min_{i \in [k]} \widehat{\boldsymbol{L}}_{T,i}] \leq \min_i \mathbb{E}[\widehat{\boldsymbol{L}}_{T,i}] \leq \min_i \boldsymbol{L}_{T,i}$, immediately implies that the expected regret of the algorithm for the loss sequence $\{\boldsymbol{\ell}_t\}$ is upper bounded by the expected regret for $\{\hat{\boldsymbol{\ell}}_t\}$. Hence, in what follows, we bound the expected regret for the latter.

The expected regret can be decomposed as

$$\mathbb{E}[\mathcal{R}_T] = \mathbb{E}(\textstyle\sum_t m_t - \widehat{\boldsymbol{L}}_{T,*}) + \mathbb{E} \sum_t \mathbb{E}_t[\langle \boldsymbol{w}_t, \hat{\boldsymbol{\ell}}_t \rangle - m_t],$$

where $m_t := -\eta_t^{-1} \log \langle \boldsymbol{w}_t, e^{-\eta_t \hat{\boldsymbol{\ell}}_t} \rangle$ is the mix loss and $\widehat{\boldsymbol{L}}_{T,*} := \min_i \widehat{\boldsymbol{L}}_{T,i}$. Bounding the first part is standard and by Lemma 2, it is at most $\log k/\eta_T$. For the second term in the regret decomposition, we show in Lemma 4 that the t^{th} term in the sum is bounded by η_t. Our proof of this lemma heavily relies on how we defined $v(\boldsymbol{p}_t, \boldsymbol{w}_t)$ and the form of our estimated losses. Plugging in $\eta_t = \sqrt{\log k/2t}$ finishes the proof.[3] □

Our next result concerns the number of queries. We show that in the adversarial setting, this number depends linearly on the total mistakes of the best model (not taking into account the rounds where all models misclassify the instance). For example, if the best model is perfect, the query count is $O(\sqrt{T})$.

Theorem 2 (Queries)**.** *Assume that in every round there are at least two models that disagree. Also assume that the total number of mistakes of the best model satisfies $\boldsymbol{L}_{T,*} \leq (\frac{|\mathcal{C}|-1}{|\mathcal{C}|} - \varepsilon)T$ for some $\varepsilon > 0$. Then, for $T \geq 4 \log k/\varepsilon^2$, the expected number of queries up to round T is at most $5\sqrt{T \log k} + 2\boldsymbol{L}_{T,*}$.*

Proof. The main idea is to relate the number of updates to the regret. First, we bound q_t from above by $\eta_t + \sum_{c \in \mathcal{C}} \boldsymbol{w}_{t,c}(1 - \boldsymbol{w}_{t,c})$, as maximum is smaller than the sum. Then, using the concavity of $a(1-a)$ and Jensen's inequality we further bound the sum over classes by $r_t(2 - \frac{|\mathcal{C}|}{|\mathcal{C}|-1} r_t)$, where $r_t = \langle \boldsymbol{w}_t, \ell_t \rangle$. The proof finishes

[3] The attentive reader familiar with OMD/FTRL might have realized that the proof deviates from the usual proof methods. In a nutshell, if we consider a general regularizer, following the usual proofs, one has to bound the stability of the algorithm, which boils down to bounding $\|\hat{\boldsymbol{\ell}}_t\|_{t,*}^2$ by a constant, where $\|\cdot\|_t$ is the local norm at round t induced by the inverse Hessian of the regularizer. As $\hat{\boldsymbol{\ell}}_t \in [0, 1/\eta_t]$, it can scale up to $O(\sqrt{T})$ and there is no trivial way to bound the norm, as the norms are equivalent in $\mathbb{R}^k$.

by summing over t and carefully invoking Jensen's inequality again. □

Remark. If all models are bad (i.e., if $L_{T,*} \approx T$), then our algorithm *can* query a lot, and the bound above is not loose. A simple adversarial example is illustrated in Appendix B. Better bounds on the number of queries are possible with more assumptions on the stream, e.g., when the stream is stochastic.

We now consider the quality of MODEL PICKER's recommendations for model selection. In the full generality of the adversarial setting, one cannot say much about the identification probability. However, if we restrict the adversary and assume that after some round t_0, the cumulative loss of the models start to deviate and keep a minimal gap, we can give a sharp lower bound on the identification probability, as well as a stronger bound on accuracy gap. We call an adversary (t_0, Δ)-*restricted* if there exists some expert $i^* \in [k]$ so that for all $t \geq t_0$, $\boldsymbol{L}_{t,j} \geq \boldsymbol{L}_{t,i^*} + \Delta t$ for all $j \neq i^*$.

If the algorithm recommends π_t at round t, its accuracy gap is defined as $\frac{1}{t}(\boldsymbol{L}_{t,\pi_t} - \boldsymbol{L}_{t,i^*})$ and its identification probability is $\Pr\{\pi_t = i^*\}$, where i^* is the best model up to round t, i.e., $i^* = \arg\min_{i\in[k]} \boldsymbol{L}_{t,i}$ (notice the use of $\boldsymbol{L}$ instead of $\widehat{\boldsymbol{L}}$ in both definitions).

Theorem 3 (Accuracy Gap)**.** *Under no assumptions on the adversary, modify the algorithm to recommend $\pi_t = I_\tau$, where $\tau \in [t]$ is selected uniformly at random. Then, to reach an expected accuracy gap of at most ε, it is enough to have $t \geq 8 \cdot \log k/\varepsilon^2$.*

Moreover, if the adversary is (t_0, Δ)-restricted, by recommending $\pi_t = \arg\max_{i\in[k]} \boldsymbol{w}_{t,i}$ and

$$t \geq \min\left\{31 \cdot \frac{\log k}{\Delta^4} \log^2(\tfrac{1}{\varepsilon}), t_0\right\},$$

one gets an expected accuracy gap of at most ε.

The proof of the first part is based on our regret bound and is standard. The second part is a simple corollary of Theorem 4 below. The difference between the two guarantees is twofold: while the first guarantee is instance independent, its dependence on $1/\varepsilon$ is quadratic. However, the second guarantee comes with poly-logarithmic dependence on $1/\varepsilon$, but with an instance-dependent constant $1/\Delta^4$.

Theorem 4 (Identification Probability)**.** *If the adversary is (t_0, Δ)-restricted, the probability that we misidentify the best model at round $T \geq t_0$ is at most*

$$\Pr\{\pi_T \neq i^*\} \leq k \cdot e^{-0.18\Delta^2\sqrt{T \log k}}$$

This theorem, together with Theorem 6 below, clearly shows why MODEL PICKER is successful in model selection tasks, as the probability of misidentifying the best model decreases (close to) exponentially fast, even if the stream is (restricted) adversarial. The proof is similar to Theorem 6 and is based on martingale arguments.

4.3 Guarantees for Stochastic Streams

In this section, we assume that the stream is i.i.d. and provide stronger results. Let $i^* \in [k]$ be the model with the highest expected accuracy, and define $\Delta_j = \mathbb{E}[\boldsymbol{\ell}_{\cdot,j} - \boldsymbol{\ell}_{\cdot,i^*}]$ for all $j \in [k]$ to be the gap between the accuracies of model j and the best model. Also define $\theta_j = \Pr\{\boldsymbol{\ell}_{\cdot,j} \neq \boldsymbol{\ell}_{\cdot,i^*}\}$ to be the probability that exactly one of j and i^* correctly classify a sample. Define

$$\lambda = \min_{j\in[k]\setminus\{i^*\}} \Delta_j^2/\theta_j.$$

Intuitively, λ measures the hardness of the instance for our algorithm. Set $\Delta = \min_{i\neq i^*} \Delta_i$ and assume that $\Delta > 0$ (i.e., there is a unique best model). To simplify the exposition, we always assume, w.l.o.g., that in all rounds at least two models disagree, as the rounds in which all models agree do not contribute to the regret or to the number of queries. The *pseudo-regret* is defined as $R_T = \mathbb{E}\sum_t \langle \boldsymbol{w}_t, \boldsymbol{\ell}_t\rangle - T\Delta$.

We first improve Theorem 2 and show on average MODEL PICKER asks $O(\sqrt{T \log k} \cdot |\mathcal{C}|/\Delta)$ labels. The dependence on $1/\Delta$ has the following intuition: it takes on average $1/\Delta$ rounds to observe an instance where the best model performs better than the rest. The bound shows that MODEL PICKER needs no more than $O(\sqrt{T \log k})$ of these instances to build up sufficient confidence in the best model.

Theorem 5 (Queries)**.** *The expected number of queries up to round T is bounded by*

$$\mathbb{E}\left[\sum_{t=1}^T Q_t\right] \leq \sqrt{2T \log k}(1 + 4\tfrac{|\mathcal{C}|}{\Delta}).$$

Proof. Notice that the expected regret is lower bounded by the pseudo-regret and upper bounded by our adversarial regret bound (Theorem 1). These bounds imply

$$\mathbb{E}\sum_{t=1}^T (1 - \boldsymbol{w}_{t,i^*})\Delta \leq R_T \leq \mathbb{E}[\mathcal{R}_T] \leq 2\sqrt{2T \log k}.$$

Hence, $\mathbb{E}\sum_{t=1}^T (1 - \boldsymbol{w}_{t,i^*}) \leq \frac{2\sqrt{2T \log k}}{\Delta}$. This means that $\boldsymbol{w}_{t,i^*} > \frac{1}{2}$ most of the times: if N is the number of rounds such that $\boldsymbol{w}_{t,i^*} \leq \frac{1}{2}$, we have

$$\frac{1}{2}\,\mathbb{E}\,N \leq \mathbb{E}\sum_{t=1}^T (1 - \boldsymbol{w}_{t,i^*}) \leq \frac{2\sqrt{2T \log k}}{\Delta}.$$

Now, by the definition of q_t we have $q_t \leq \eta_t + \sum_{c\in\mathcal{C}} \boldsymbol{w}_{t,c}(1 - \boldsymbol{w}_{t,c})$. For a class $c \in \mathcal{C}$ that is present among the models predictions at round t, we can write $\boldsymbol{w}_{t,c}(1-\boldsymbol{w}_{t,c}) = (\boldsymbol{w}_{t,i^*}+a)\cdot b$, for some $a, b \geq 0$ with $b \leq$

$1 - \boldsymbol{w}_{t,i^*}$. When $\boldsymbol{w}_{t,i^*} \geq \frac{1}{2}$, we have $q_t \leq b \leq 1 - \boldsymbol{w}_{t,i^*}$. If $\boldsymbol{w}_{t,i^*} \leq \frac{1}{2}$ we bound q_t by $\frac{1}{4}$. Summing over t and using the bound on N, we finishes the proof. □

The next three results are parallel to the ones in the previous section. By adopting careful martingale arguments, we first show that the probability of misidentifying the best model decreases (close to) exponentially with a rate depending on λ.

Theorem 6 (Identification Probability). *For $T > 2 \log k$, the probability that we misidentify the best model at round T is at most*

$$\Pr\{\pi_T \neq i^*\} \leq k \cdot e^{-0.18\lambda\sqrt{T \log k}}.$$

Proof Sketch. Notice that $\xi_t = \Delta_j - \hat{\boldsymbol{\ell}}_{t,j} + \hat{\boldsymbol{\ell}}_{t,i^*}$ is a martingale difference sequence. Using a variation of Freedman's inequality for martingales and a careful analysis, one arrives at the theorem. □

Bounds on accuracy gap follow easily. The idea is that by Theorem 6, the best arm is always recommended, except for a constant number of rounds.

Theorem 7 (Accuracy Gap). *For*

$$T \geq 31 \cdot \frac{\log^2(k \max_i \Delta_i)}{\lambda^2 \log k} \log^2(\tfrac{1}{\varepsilon}),$$

recommending π_T results in an expected accuracy gap of at most ε.

To bound the regret, Theorem 1 is still applicable. Additionally, if one predicts according to $I_t = \pi_t$ (a.k.a. Follow The Leader strategy), the following theorem shows that the pseudo-regret is bounded by a *constant*.[4]

Theorem 8 (Regret). *If in Algorithm 1 one sets $I_t = \pi_t$ for all t, then the pseudo-regret bounded by a constant:*

$$R_T \leq 62 \max_i \Delta_i \tfrac{k}{\lambda^2 \log k}.$$

5 EXPERIMENTS

We conduct an extensive set of experiments to demonstrate the practical performance of Model Picker for online model selection and sequential label prediction. We first run experiments on common data sets where the instances come i.i.d. from a fixed data distribution. This setting allows us to empirically assess the performance in the stochastic setting. We then consider a more challenging scenario where examples come from a drifting data distribution, which we treat as an adversarial stream.

Datasets and Model Collection We conduct our experiments using various models trained on common datasets such as the SemEval 2019 dataset (EmoContext) for emotion detection (SemEval) and the long-term gas sensor drift dataset (Drift) from the UCI Machine Learning Repository (Vergara, 2012; Vergara et al., 2012) as well as on more complex datasets of natural images such as CIFAR-10 and ImageNet. These datasets cover a wide range of scale: CIFAR-10, EmoContext and Drift are of smaller scale while ImageNet is a large scale dataset. Each dataset consists of a large test set (which we later use to construct streams of examples) and (possibly multiple) training sets. For each dataset, we collect a collection of pre-trained models by training various models on the training sets. We provide a detailed explanation on the characteristics of our model collection in Appendix C.1.

For CIFAR-10, we trained 80 classifiers varying in model, architecture, and parameter settings available on Pytorch Hub[5]. The ensemble contains models having accuracies between 55-92% on a test set consisting of 10 000 CIFAR-10 images. The ImageNet dataset poses a 1 000-class classification problem. We collected 102 image classifiers that are available on TensorFlow Hub[6]. The accuracy of these models is in the range 50-82%. For the test set, we use the whole official test set with 50 000 images. For the EmoContext dataset, we collected 8 pre-trained models that are the development history of a participant in SemEval 2019. The accuracy of the models varies in 88-92% on a test set of size 5 509. Lastly, for the Drift dataset, we trained an SVM classifier on each of 9 batches of gas sensor data that were measured in different months. We use the last batch as a test set, which is of size 3 000. Due to the drift behaviour of sensor data among different time intervals, the accuracy of the models on the test set is relatively low, and lies in 25-60%.

Baselines To compare with existing selective sampling strategies, we implement variations of QBC, namely, vote entropy (Entropy) and structural QBC (S-QBC) as well as label efficient prediction (Efficient) and importance weighted active learning (Importance), as described below. Typically, these methods follow a coin flipping strategy: upon seeing an instance x_t, a coin is flipped with a bias q_t, and the label t is requested if and only if the coin comes up heads.

Label Efficient Prediction/Passive Learning. We implement (Cesa-Bianchi et al., 2005) by querying the label

[4] In full information, when one observes all the labels, the FTL strategy fails to have the no-regret property in the adversarial setting. However, it has been shown that it favors a constant regret bound in stochastic settings. We show that our algorithm has the same behaviour.

[5] https://pytorch.org/hub/

[6] https://tfhub.dev/

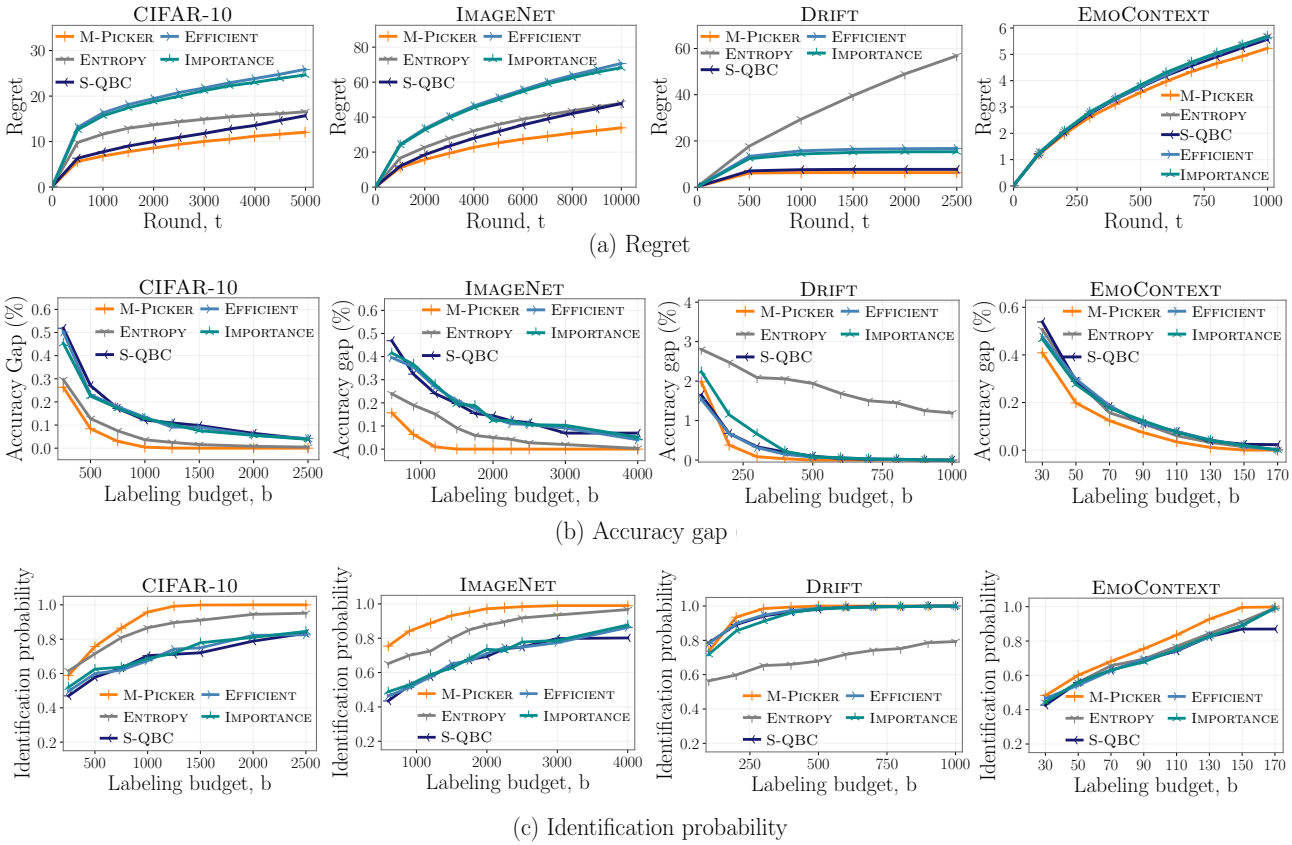

Figure 1: Performance of Model Picker (M-Picker) and other adapted baselines on four datasets {CIFAR-10, ImageNet, Drift, EmoContext}. Model Picker is able to output the true best model with high probability, while querying up to 2.6× fewer labels than the best competing method.

of each round randomly with a fixed probability $q_t = \varepsilon$. For a fair comparison, we restrict our interest merely to the data instances in which at least two models disagree, as others are non-informative in the ranking of models. In our evaluation, we set the query probability to $\varepsilon = b/T$ for having an expected number of b queries in a stream of size T. Note that our way of setting ε depends on the whole stream for having comparable results in terms of the number of queries, as we shall drop the non-informative samples first.

QBC/Vote Entropy. We use the method of Dagan and Engelson (1995) and adapt it to the streaming setting as a disagreement-based selective sampling baseline. Upon seeing each instance, we measure the disagreement between the model predictions to compute the query probability. In our implementation, we consider every pre-trained model as a committee member and use vote entropy as the disagreement measure.

Structural QBC. The (interactive) structural QBC algorithm (Tosh and Dasgupta, 2018) is built upon the QBC principle, and its query probability is specified via the disagreement between competing models that are drawn from a posterior distribution $\boldsymbol{\rho}_t$. After each new query, the posterior is updated as $\boldsymbol{\rho}_{t+1} \propto \boldsymbol{\rho}_t \exp(-\beta \boldsymbol{\ell}_t)$, where β is a fixed parameter. In our implementation, at each round t, we draw two models i and j from $\boldsymbol{\rho}_t$ with replacement and set the query probability to be the fraction of disagreement between i and j up to round t, that is, $q_t = \frac{1}{t}\sum_{s\le t} \mathbb{I}_{\{\boldsymbol{p}_{s,i} \neq \boldsymbol{p}_{s,j}\}}$.

Importance Weighted Active Learning. We implemented the importance weighted active learning algorithm introduced by Beygelzimer et al. (2008), as well as its variant for efficient active learning (Beygelzimer et al., 2010, 2011). Among these two adaptations of importance weights, we only focus on the superior (Beygelzimer et al., 2008) in our empirical evaluation and leave the others to Appendix C.2.

It is crucial to note that none of the methods above are tailored for the task of *ranking pre-trained models* and (except for Cesa-Bianchi et al. (2005)) for *sequential label prediction*. Yet we consider them as selective sampling baselines; see Appendix C.2 for further discussions about our baselines.

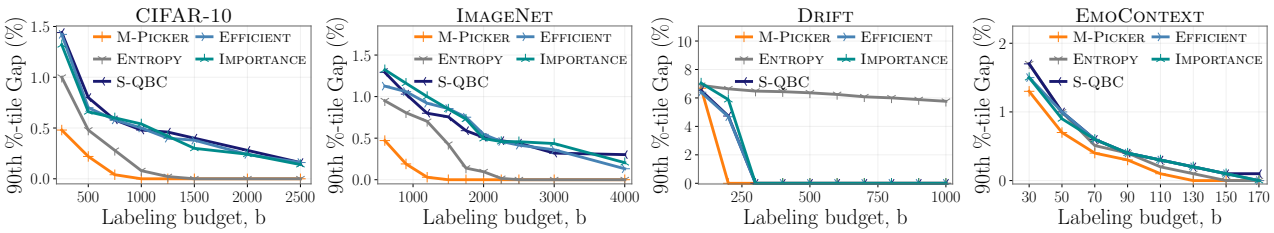

Figure 2: Worst-case analysis on the outputted models: 90th percentile accuracy gap

5.1 Experimental Setup

Evaluation Protocol and Tuning For a fair comparison, we focus on the following protocol. We sequentially draw T i.i.d. instances uniformly at random from the entire pool of test instances, then input it into each algorithm as a stream, and call it a realization. In each realization, the pre-trained model with the highest accuracy on that stream (considering all labels) is denoted as the *true* best model of the realization.

For each realization and up to any round t, Model Picker outputs $\pi_t = \arg\max_i \boldsymbol{w}_{t,i}$ as the best model, and other methods output the model having the highest accuracy on the queried labels. Upon exhausting the stream, we evaluate the performance of each method based on the model that is outputted. We realize this process many times to have an estimate of the expected performance.

For comparing the methods under the same budget constraint, we tune the (hyper-)parameter of each method to query the same number of instances, and compare their average performance under various labeling budgets. For Structural QBC, we treat β (in the posterior) as the hyperparameter. For QBC with vote entropy, importance weighted active learning and Model Picker, we *introduce* a hyperparameter β to scale the query probability according to the given labeling budget.[7] Note that by default, Model Picker needs no hyperparameters, and we introduce β for the sole reason of fair comparison with other methods. We perform hyperparameter selection via a grid search. The hyperparameters used for each budget, together with a large range of hyperparameters and their respective budgets can be found in Appendix C.4.

Performance Metrics For a given labeling budget, we consider the following key quantities as performance measures: *Regret* for a fixed labeling budget, *Accuracy gap* between the outputted model and the true best model, and *Identification Probability*, which is the fraction of realizations that methods return the true best model of that realization.

Scaling and Computation Cost We conduct our experiments on different stream sizes. We choose sizes of 5 000, 10 000, 1 000 and 2 500 for CIFAR-10, ImageNet, EmoContext and Drift test sets, respectively. We implement Model Picker, along with all other baseline methods in Python. All the baseline methods combined, each realization takes between 1 second (for EmoContext) and 4 minutes (for ImageNet) when executed on a single CPU core. Model Picker alone takes between 75 miliseconds (for EmoContext) and 47 seconds (for ImageNet). For all datasets we run 500 independent realizations for each budget constraint. To improve the overall runtime, we run the realizations in parallel over a cluster with 400 cores.

5.2 Experimental Results

We review our numerical results for each of the metrics introduced earlier. We refer to Appendix C.3 for an extensive discussion of our findings. For each of our metrics, we observe the following:

Regret We measure the regret across all rounds and for those budgets where Model Picker returns the best model with high confidence. Namely, we set the budget to 1 250, 1 200, 130 and 1 000 for the CIFAR-10, ImageNet, EmoContext and Drift datasets, respectively. The regret behaviour is shown in Figure 1(a). In all cases, the regret grows sub-linearly for all algorithms. The regret of our algorithm in all cases is smaller up to a factor of 1.3×, which shows that Model Picker can be used for sequential label prediction tasks as well as model selection.

Accuracy Gap Next, we consider the average accuracy gap over the realizations. Figure 1(b) shows that the accuracy gaps for Model Picker are much smaller than that of other adapted methods under the same budget constraints. Quantitatively, in both CIFAR-10 and ImageNet datasets, Model Picker achieves the same expected accuracy gap as Entropy by querying nearly 2.5× less labels. For the Drift dataset,

[7] It is straightforward to see that by scaling the value of $v(\boldsymbol{p}_t, \boldsymbol{w}_t)$ by some constant, one still gets similar theoretical results. The regret bounds, as well as the bounds on the confidence and accuracy gap will be scaled accordingly.

for instance, MODEL PICKER returns a model that is within a 0.1%-neighborhood of the accuracy of best model after querying merely 11% of the entire stream of examples (when the budget is 270 for a stream of size 2 500). Note that active learning over drifting data distribution is a very challenging task, and EFFICIENT (Label Efficient Prediction/Passive Learning) is considered the strongest baseline (Settles, 2009). Our experiments thus suggest that, even for small labeling budgets, MODEL PICKER returns a model whose accuracy is close to that of the best model, if not the best model itself.

Identification Probability As illustrated in Figure 1(c), MODEL PICKER achieves significant improvements of up to 2.6× in labeling cost while returning the true best model and requesting far fewer labels than other adapted methods. For CIFAR-10, IMAGENET, EMOCONTEXT and DRIFT datasets, MODEL PICKER queries 2.5×, 2.5×, 1.2× and 1.7× fewer labels respectively than that of the best competing method (mainly ENTROPY) to reach confidence levels 95%, 97%, 92% and 97%, respectively. This shows that MODEL PICKER is able to achieve the same identification power as the adapted baselines at a much lower labeling cost.

5.2.1 On the Robustness of Model Picker

Practitioners are often interested in the relative quality of the output model compared to the true best model in a single trial. In this regard, and in the spirit of Theorems 4 and 6, we conduct further numerical analysis on the accuracy of the outputted models over a large number of realizations to investigate if MODEL PICKER performs well *with high probability*. We compute the 90th percentile of accuracy gap as a proxy for the behaviour of the algorithms in the high probability regime (see Figure 2). In the DRIFT dataset, for instance, MODEL PICKER returns the true best model after querying merely 8% of the labels (when the budget is 200 with a stream size of 2 500). For the CIFAR-10 and IMAGENET datasets, MODEL PICKER returns a model that is within a 0.1%-neighborhood of the accuracy of best model after querying nearly 12% of the entire stream of examples whereas the best competing method achieves this after querying 24% of the same stream of examples. Moreover, MODEL PICKER outputs the true best model after querying 15% and 20% of the entire stream of examples, respectively. These results clearly demonstrate the robustness of MODEL PICKER.

6 CONCLUSIONS

We introduced an online active model selection approach –MODEL PICKER– to selectively query the labels of instances that are informative for ranking pre-trained models and to sequentially predict unseen labels. Our framework is generic, easy to implement, and applies across various classification tasks. We derived theoretical guarantees and illustrate the effectiveness of our method on several real-world datasets.

Acknowledgements

We thank the reviewers for their constructive feedback. This research was supported by the SNSF grant 407540_167212 through the NRP 75 Big Data program and by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme grant agreement No 815943. CZ and the DS3Lab gratefully acknowledge the support from the Swiss National Science Foundation (Project Number 200021_184628), Innosuisse/SNF BRIDGE Discovery (Project Number 40B2-0_187132), European Union Horizon 2020 Research and Innovation Programme (DAPHNE, 957407), Botnar Research Centre for Child Health, Swiss Data Science Center, Alibaba, Cisco, eBay, Google Focused Research Awards, Oracle Labs, Swisscom, Zurich Insurance, Chinese Scholarship Council, and the Department of Computer Science at ETH Zurich.

References

Naoki Abe and Hiroshi Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML 1998, page 1–9, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1558605568. 2, 19

Alnur Ali, Rich Caruana, and Ashish Kapoor. Active learning with model selection. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1673–1679. AAAI Press, 2014. 2

Jason Altschuler and Kunal Talwar. Online learning over a finite action set with limited switching. In *Conference On Learning Theory*, pages 1569–1573, 2018. 2

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002. 2

Francis R. Bach. Active learning for misspecified generalized linear models. In *Advances in Neural Information Processing Systems 19*, pages 65–72. MIT Press, 2007. 2

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009. 2

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. *ACM International Conference Proceeding Series*, 382, 12 2008. doi: 10.1145/1553374.1553381. 2, 7, 20

Alina Beygelzimer, Daniel J Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems 23*, pages 199–207. Curran Associates, Inc., 2010. 2, 7, 20

Alina Beygelzimer, Daniel Hsu, Nikos Karampatziakis, John Langford, and Tong Zhang. Efficient active learning. In *Proceedings of the 28th International Conference on Machine Learning*, 2011. 2, 7, 20

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. 19

Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997. 4

Nicolo Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 51(6): 2152–2162, 2005. 2, 3, 6, 7

David A. Cohn, Les E. Atlas, and Richard E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994. 2

Ido Dagan and Sean P. Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier, 1995. 7, 19

Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems 20*, pages 353–360. Curran Associates, Inc., 2008. 2

Steven de Rooij, Tim van Erven, Peter D. Grünwald, and Wouter M. Koolen. Follow the Leader If You Can, Hedge If You Must. *arXiv:1301.0534 [cs, stat]*, 2013. 13

Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995. 19, 20

Jacob Gardner, Gustavo Malkomes, Roman Garnett, Kilian Q Weinberger, Dennis Barbour, and John P Cunningham. Bayesian active model selection with an application to automated audiometry. In *Advances in Neural Information Processing Systems 28*, pages 2386–2394. Curran Associates, Inc., 2015. 2

Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019. 3

Mohammad Reza Karimi, Andreas Krause, Silvio Lattanzi, and Sergei Vassilvtiskii. Consistent online optimization: Convex and submodular. In *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 2241–2250. PMLR, 16–18 Apr 2019. 2

Namit Katariya, Arun Iyer, and Sunita Sarawagi. Active evaluation of classifiers on large datasets. In *2012 IEEE 12th International Conference on Data Mining*, pages 329–338. IEEE, 2012. 2

Anurag Kumar and Bhiksha Raj. Classifier risk estimation under limited labeling resources. In *Advances in Knowledge Discovery and Data Mining*, pages 3–15, Cham, 2018. Springer International Publishing. ISBN 978-3-319-93034-3. 2

Rui Leite and Pavel Brazdil. Active testing strategy to predict the best classification algorithm via sampling and metalearning. In *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, page 309–314, NLD, 2010. IOS Press. ISBN 9781607506058. 2

Nick Littlestone and Manfred K Warmuth. *The Weighted Majority Algorithm*. Elsevier, 1994. 3

Omid Madani, Daniel J. Lizotte, and Russell Greiner. Active model selection. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, page 357–365, Arlington, Virginia, USA, 2004. AUAI Press. ISBN 0974903906. 2

Andrew McCallum and Kamal Nigam. Employing em and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 350–358, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1558605568. 2

Prem Melville and Raymond J. Mooney. Diverse ensembles for active learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 74, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. 2, 19

Hariharan Narayanan and Alexander Rakhlin. Random walk approach to regret minimization. In *Advances in Neural Information Processing Systems*, pages 1777–1785, 2010. 13

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics. 19

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke

Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. 19

Christoph Sawade, Niels Landwehr, Steffen Bickel, and Tobias Scheffer. Active risk estimation. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 951–958, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077. 2

Christoph Sawade, Niels Landwehr, and Tobias Scheffer. Active comparison of prediction models. In *Advances in Neural Information Processing Systems 25*, pages 1754–1762. Curran Associates, Inc., 2012. 2

Yevgeny Seldin and Gábor Lugosi. An Improved Parametrization and Analysis of the EXP3++ Algorithm for Stochastic and Adversarial Bandits. *arXiv:1702.06103 [cs, stat]*, 2017. 16

SemEval, 2019. URL https://www.humanizing-ai.com/emocontext.html. 6

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009. 9

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, page 1070–1079, USA, 2008. Association for Computational Linguistics. 2, 20

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294, 1992. 2

Aleksandrs Slivkins. Introduction to Multi-Armed Bandits. *arXiv:1904.07272 [cs, stat]*, September 2019. 18

Masashi Sugiyama. Active learning for misspecified models. In *Advances in Neural Information Processing Systems 18*, pages 1305–1312. MIT Press, 2006. 2

Christopher Tosh and Sanjoy Dasgupta. Interactive structure learning with structural query-by-committee. In *Advances in Neural Information Processing Systems 31*, pages 1121–1131. Curran Associates, Inc., 2018. 7

Alexander Vergara. UCI machine learning repository, 2012. URL http://archive.ics.uci.edu/ml/datasets/Gas+Sensor+Array+Drift+Dataset. 6

Alexander Vergara, Shankar Vembu, Tuba Ayhan, Margaret A Ryan, Margie L Homer, and Ramón Huerta. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166:320–329, 2012. 6

Chicheng Zhang and Kamalika Chaudhuri. Beyond disagreement-based agnostic active learning. In *Advances in Neural Information Processing Systems 27*, pages 442–450. Curran Associates, Inc., 2014. 2

Xingquan Zhu, Peng Zhang, Xiaodong Lin, and Yong Shi. Active learning from data streams. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 757–762. IEEE, 2007. 2