THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Dynamic Ensemble Active Learning: A Non-Stationary Bandit with Expert Advice

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Dynamic Ensemble Active Learning:
A Non-Stationary Bandit with Expert Advice

Kunkun Pang*, Mingzhi Dong†, Yang Wu‡, Timothy M. Hospedales*
*University of Edinburgh,   Email: k.pang@ed.ac.uk, t.hospedales@ed.ac.uk
†University College London,   Email: mingzhi.dong.13@ucl.ac.uk
‡Nara Institute of Science and Technology,   Email: yangwu@rsc.naist.jp

Abstract—**Active learning aims to reduce annotation cost by predicting which samples are useful for a human teacher to label. However it has become clear there is no best active learning algorithm. Inspired by various philosophies about what constitutes a good criteria, different algorithms perform well on different datasets. This has motivated research into ensembles of active learners that learn what constitutes a good criteria in a given scenario, typically via multi-armed bandit algorithms. Though algorithm ensembles can lead to better results, they overlook the fact that not only does algorithm efficacy vary across datasets, but also during a single active learning session. That is, the best criteria is non-stationary. This breaks existing algorithms' guarantees and hampers their performance in practice. In this paper, we propose dynamic ensemble active learning as a more general and promising research direction. We develop a dynamic ensemble active learner based on a non-stationary multi-armed bandit with expert advice algorithm. Our dynamic ensemble selects the right criteria at each step of active learning. It has theoretical guarantees, and shows encouraging results on 13 popular datasets.**

I. INTRODUCTION

The key barrier to scaling or applying supervised learning in practice is often the cost of obtaining sufficient annotation. Active Learning (AL) aims to address this by designing query algorithms that effectively predict which points will be useful to annotate, thus enabling efficient allocation of human annotation effort. There are many different AL algorithms, each with appealing – yet completely different – motivations for what constitutes a good question to ask underpinning their design. For example, uncertainty or margin-based sampling [1], [2] suggests querying the most uncertain or ambiguous point that is the closest point to the decision boundary. Expected error reduction [3], [4] queries points that the current model predicts will reduce its future error. Another typical approach is to label the most representative samples [5], [6], [7] to ensure the major clusters within the dataset are correctly estimated. Besides these approaches, query-by-committee active learning queries points based on the disagreement between a committee of classifiers [8], [9], [10]. More recent studies investigated hybrid criteria that balance multiple motivations [11], [12], [13].

These are all good ideas, yet there are situations where each is ineffective. For example, if the classes are heavily overlapped in an area of feature-space, uncertainty sampling will be tied up querying points in an impossible to solve region. If the current model is very poor, expected error reduction cannot accurately estimate its own future error. If the main data clusters are already well classified, representative sampling focused approaches may not fine-tune between them.

These thought experiments are reflected empirically. The best algorithm for pool-based AL in practice varies both across datasets and also with the progress of learning within a given dataset [14], [15]. This observation has motivated research into both learning dataset and time-specific weightings for an AL algorithm ensemble. [16], [17] developed heuristics for switching AL algorithms that are typically good at early versus late stage learning. In contrast, [14], [15] developed methods for rapid online learning of a dataset-specific weighting for algorithms within an AL-ensemble.

The key insight of the Combination of Active Learning Online (COMB) [14] and Active Learning by Learning (ALBL) [15] algorithms is to formalise the query criteria selection task as a multi-armed bandit (MAB) problem. MAB problems have been well studied and many powerful algorithms with optimality guarantees exist. For example, if each query criterion in the ensemble is considered to be a bandit arm, and the learning improvement after executing a criterion is considered to be the arm's reward, then MAB algorithms such as EXP3 (Exponential-weight algorithm for Exploration and Exploitation) [18] can be applied to quickly learn the efficacy of the arms (AL criteria) and is guaranteed to achieve a near optimal overall reward (learning improvement). A variant of this is to consider data-points to be arms, and AL criteria to be experts providing advice about promising arms. Then MAB with *expert advice* algorithms such as EXP4.P (Exponential-weight algorithm for Exploration and Exploitation using Expert advice with high probability regret bound) [19] optimise exploration and exploitation of experts, and achieve provably near optimal reward.

The fundamental limitation of existing MAB-based approaches to AL is that their underlying MAB algorithms do not take into account the temporal dynamics of active learning: different criteria are effective at different learning stages [16], [17]. This issue is illustrated by Fig. 1(a,c,e), where the most effective criterion varies across the entire time horizon. On *fourclass*, Density (DE) sampling is slightly better at first and uncertainty (US) is consistently good later

on. Similarly in *ILPD* or *german*, representative (RS) and density (DE) sampling are better at the crucial early stage before uncertainty becomes better. A second issue is that the scale of an accuracy-based reward falls dramatically over time (Fig. 1(b,d,f)). Because of this stationary bandit learners will be unduly biased by the high reward from an initial observation and fail to adapt subsequently. For example in ILDP a stationary learner may fail to make the switch from DE to US as later rewards in favour of US are small in scale compared to the initial reward in favour of DE.

Therefore there are non-stationary aspects both in reward scale, and in reward distribution per-arm (MAB perspective) or per-expert (MAB with expert advice perspective). Thus the MAB problem is formally non-stationary, violating a fundamental assumption required to guarantee existing MAB algorithms' optimality bounds.

Here we develop a performance guaranteed stochastic MAB with expert advice[1] algorithm in a *non-stationary* environment. Applying this to AL means that, like [15], if there is a single best (but a priori unknown) AL algorithm for a dataset, we are able to quickly discover it and thus approach the performance of an oracle that knows the best algorithm for each dataset. But importantly when different algorithms' efficacies vary over time within one dataset, we can adapt to this and approach the performance of an oracle that knows the best AL algorithm *at each iteration*.

II. BACKGROUND AND RELATED WORK

A. Active Learning

We denote the pool of data with M samples as $\mathcal{D} = \{x_i, l_i, \ldots, x_M, l_M\}$ where the instances are $x_i \in \mathbb{R}^d$ and the labels are $l_i \in \{1, \ldots, C\}$. In an active learning scenario, the data $\mathcal{D}$ are initially a labelled set $\mathcal{L}$ and unlabelled set $\mathcal{U} = \mathcal{D} \setminus \mathcal{L}$ where $|\mathcal{L}| \ll |\mathcal{U}|$. Training an initial classifier f_0 on the samples in the initial set $\mathcal{L}$, the algorithm starts to query instances x_q from $\mathcal{U}$ during iterations $t = 1, \ldots, T$. After the supervision l_q of instance x_q is obtained, x_q is removed from the unlabelled set $\mathcal{U}$ and added to the labelled set $\mathcal{L}$, from which classifier f_t is retrained.

B. Bandit Algorithms

Multi-armed Bandit In multi-armed bandit (MAB) problems, a player pulls a lever from a set $\mathcal{K} = \{1, \ldots, K\}$ of slot machines in a sequence of time steps $\mathcal{T} = \{1, \ldots, T\}$ to maximise her payoff. During the game, she only observes the reward $r^k(t) \in [0, 1]$ of the specific arm pulled k at time step t. The aim of the player is to maximise their return, which is the sum of the rewards over the sequence of pulls. This requires a trade-off between exploration (collect information to estimate the arm with the highest return) and exploitation (focus on the arm with the highest estimated return). Training a bandit learner to solve a MAB problem is then formalized

[1]We use terminology from [18]. It also has other names, including 'contextual bandit' [19], [20], 'partial-label problem' [21], and 'associative bandit problem' [22].

(a) Proportion of Wins: "fourclass"

(b) Relative Accuracy Increments: "fourclass"

(c) Proportion of Wins: "ILPD"

(d) Relative Accuracy Increments: "ILPD"

(e) Proportion of Wins: "german"

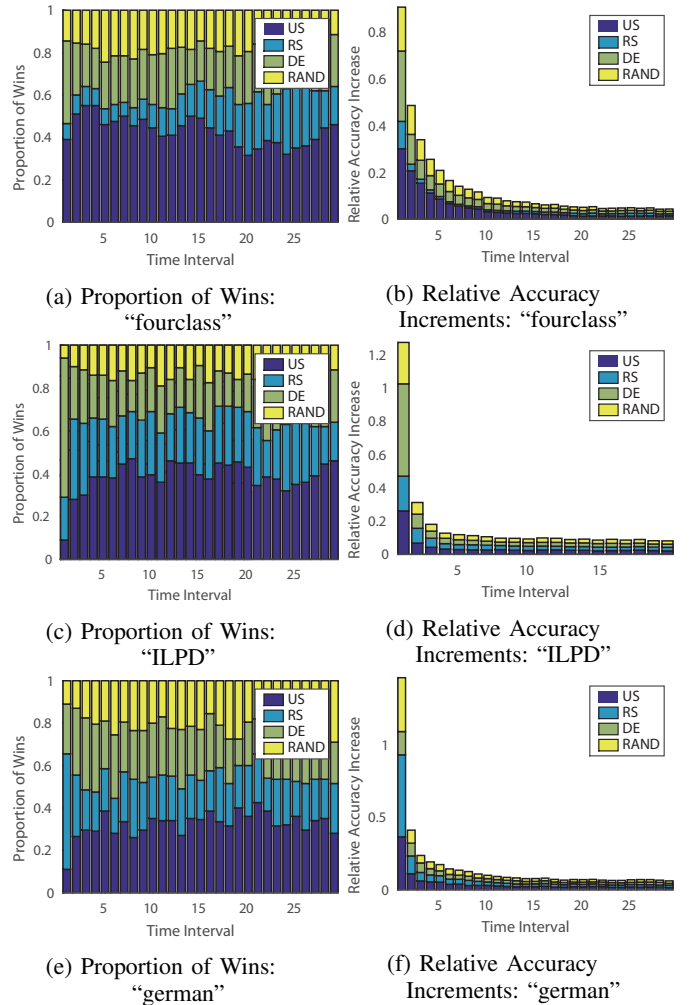(f) Relative Accuracy Increments: "german"

Fig. 1: Examples of non-stationary AL in UCI datasets "fourclass" and "ILPD" using four algorithms/criteria: US, RS, DE, and RAND. Above: Proportion of times each criterion generates the largest increase in accuracy. Below: Relative increase in accuracy. In the relative part all increments are re-scaled by subtracting the minimum increment of accuracy over all criteria in each bin.

as minimising the *regret* between the actions chosen by the player's strategy $a_k \sim \pi$, and the best arm.

For example, the EXP3 algorithm [18] minimises, for any finite T, the "*static* regret" between the player's reward and the best arm in retrospect: $\max_k \sum_{t=1}^T r^k(t) - \mathbb{E}(\sum_{t=1}^T r^\pi(t))$.

Contextual Multi-armed Bandit The goal of contextual bandits is to build a relationship between available context information $h_t \in \mathbb{R}^d$ and the reward distribution over all arms. For example, LinUCB [23] makes the linear realizability assumption that there exists an unknown weight vector $\theta^* \in \mathbb{R}^d$ with $||\theta^*|| \leq 1$ so that regret $\sum_{t=1}^T r_{a_t^*}(t) - \sum_{t=1}^T r_{a_t}(t)$ is minimized, where $r_{a_t^*}(t) = \theta^{*\top} h_t$ and $r_{a_t}(t) = \theta^\top h_t$. However, learning to predict the reward for each data point accurately appears to be an even harder problem given the

limited information from only expert suggestions (Fig 1). More importantly, given the changing reward distribution over time, there is no constant relation between context and reward.

Multi-armed Bandit with Expert Advice Expert information about the likely efficacy of each arm is often available. [18] thus introduced an adversarial MAB with expert advice algorithm EXP4 that exploits N experts giving advice vectors (probabilities $\boldsymbol{\xi}^n(t) \in [0,1]^K$ over levers) to the learner at each time. In contrast to MAB without expert advice, the goal is now to identify the best expert rather than the best arm. In this setting the regret to minimise is the difference between the return of the best expert in retrospect and the player:

$$\max_n \sum_{t=1}^{T} y_t^n - \mathbb{E}(\sum_{t=1}^{T} y_t^\pi) \qquad (1)$$

where $y_t^n = \sum_{k=1}^{K} \xi_k^n(t) \times r^k(t)$ is the expected reward of an expert and y_t^π is the expected reward of our policy.

C. Bandits for Active Learning

For active learning using a MAB with expert advice algorithm, the N experts correspond to our ensemble of active learning criteria and the K arms are available points in the pool. Each expert (criterion) n provides a probability vector encoding preference $\boldsymbol{\xi}^n(t)$ over arms (instances). Active learners based on MAB with expert advice aim to learn the best criterion for a specific dataset. In COMB [14], the authors propose to use MAB with expert advice in active learning and heuristically designed the classification entropy maximization (CEM) score as the reward of the EXP4 bandit algorithm [18]. A more recent paper [15] (ALBL) proposed to replace the CEM reward with an unbiased estimation of test accuracy Important Weighted Accuracy (IWA) and used an upgraded bandit algorithm EXP4.P [19], which improves the earlier EXP4 method. Similarly, another recent paper [24] applied linear upper confidence bound contextual bandit algorithm (LinUCB) to train an ensemble and transferred the knowledge to other datasets. All of these algorithms enable the selection of a suitable active learning criteria for a given dataset. Our contribution is also to perform AL in a dataset-specific way by optimally tuning the exploration and exploitation of an ensemble of AL algorithms; but more importantly to do so dynamically, thus allowing the optimal tuning to vary as learning progresses. Unlike [14], [15], [24] we are able to deal with the non-stationary nature of this process. And unlike the heuristics in [16], [17], we have a theoretical guarantees, and can work with more than two criteria.

D. Non-stationary Property of Active Learning

Demonstration of Non-stationarity We describe a preliminary experiment to demonstrate empirically the existence of non-stationary reward distributions for a MAB formalisation of AL. Following the learning trajectory of our method, we use an oracle to score all the available query points at each iteration (i.e., hypothetically label each point, update the classifier, and check the test accuracy). Using the actual test accuracy as the reward, we can obtain the true expected reward of the nth expert $y_t^n = \boldsymbol{\xi}^n(t)\boldsymbol{r}(t)$ at each time step t. Fig. 1 summarises the resulting average reward obtained in every 10 iterations of AL. Based on this, we can further compute the proportion of times that each criterion would obtain the highest reward. It can be seen that the MAB problem is non-stationary as the rewards vary systematically, and there is not a single criterion (expert) which obtains the highest proportion of wins throughout learning. Additionally, the ideal combination of criteria varies across datasets. For example, as illustrated in Fig 1, density and uncertainty sampling show better complementary in ILPD, while representative and uncertainty sampling are more complementary in german dataset.

Existing MAB ensembles are not robust to non-stationarity The non-stationary property in the MAB formalisation of AL also highlights the key weakness of COMB and ALBL: they use EXP4/EXP4.P [18], [19] expert advice bandit algorithms which provide guarantees against an inappropriate (static) regret that is only relevant in a stationary problem. In a non-stationary problem, it is clear that even an algorithm that perfectly estimates the best single expert (optimal w.r.t static oracle Eq. 1) can be arbitrarily worse than one which can choose the best expert at each step (optimal w.r.t dynamic oracle). In this paper, we develop an non-stationary stochastic MAB algorithm REXP4 (Restarting Exponential-weight algorithm for Exploration and Exploitation using Expert advice) with bounds against a stricter dynamic oracle notion of optimality more suited for (non-stationary) AL.

Prior attempts at non-stationary active learners A few previous active learning studies also observed that different algorithms are effective at different stages of learning and proposed heuristics for switching two base query criteria (e.g., density sampling at an early stage, and uncertainty sampling later on) [16], [17]. But these only adapt 2 criteria (density and uncertainty) unlike MAB ensembles which learn to combine many criteria, and their heuristics do not provide a principled and optimal way to learn when to switch.

Prior attempts at non-stationary MABs Some previous studies have extended MAB without expert advice learning to the non-stationary setting [25], [26] and provided regret bounds to guarantee the algorithms' performance. However bandits with expert advice are preferable because they can achieve tighter learning bounds [18], [15] and they do not treat each criterion as a black box, so that one observation can be informative about many arms. Consider an AL situation where two criteria prefer the same instance. In the MAB interpretation (criteria=arms), after observing a reward, you only learn about the criterion/arm chosen at that iteration. In the MAB with expert advice interpretation (criteria=experts), the observed reward generates updates about the efficacy of all criteria that expressed opinions about the point.

Those few MABs extended to the non-stationary setting have other stronger assumptions. For example, the discounted/sliding-window UCB algorithm [25] assumes the

nature of the non-stationarity is that the reward distribution is piece-wise and the number of changes is known. Similarly [27] makes the easier piecewise assumption, and also that the retrospective rewards for un-pulled arms are available – but they are not in active learning. In [28], the authors proposed to measure the total statistical variance of the consecutive distributions at each time interval. Their result provides a big picture of the regret landscape for full information and bandit settings. Their proposed method addresses non-stationary environments but only for the regular MAB problem. Despite the use of the term expert in the title, it does not address the Expert-advice variant of the MAB problem relevant to us. It addresses arms rather than experts over arms.

We propose a non-stationary MAB with expert advice algorithm that has performance guarantees, and validate its practical application to active learning.

III. NON-STATIONARY MULTI-ARMED BANDIT WITH EXPERT ADVICE FOR ACTIVE LEARNING

A. Non-stationary Multi-Armed Bandit with Expert Advice: REXP4

To formalise the problem, we assume the expected reward y_t^n of each expert n can change at any time step t. The total variation of the expected reward over all T steps is

$$\sum_{t=1}^{T-1} \sup_n |y_t^n - y_{t+1}^n| \tag{2}$$

Following [29], [26], we assume this total variation in expected reward is bounded by a variation budget V_T. The variation budget captures our assumed constraints on the non-stationary environment. It allows a wide variety of reward changes – from continuous drift to discrete jumps – yet provides sufficient constraint to permit a bandit algorithm to learn in a non-stationary environment. Temporal uncertainty set $\mathcal{V}$ is defined as the set of reward vector sequences that are subject to the variation budget V_T over all T steps.

$$\mathcal{V} = \left\{ y \in [0,1]^{N \times K} : \sum_{t=1}^{T-1} \sup_n |y_t^n - y_{t+1}^n| \leq V_T \right\}$$

To bound the performance of a bandit learner in a non-stationary environment, we work with the regret between the learner and a *dynamic* oracle. The regret is defined as the worst-case difference between the expected policy return and the return of using the best expert at each time t.

Definition 1. *Dynamic Regret for Multi-Armed Bandit with Expert Advice*

$$R^\pi(\mathcal{V}, T) = \sup_{y \in \mathcal{V}} \left\{ \sum_{t=1}^T y_t^* - \mathbb{E}^\pi [\sum_{t=1}^T y_t^\pi] \right\} \tag{3}$$

where $y_t^* = \max_n y_t^n$ is the best possible expected reward among all experts at time t. Our regret is against this dynamic oracle, in contrast to prior MABs' static oracle (Eq 1).

Our non-stationary MAB with expert advice algorithm REXP4 minimises the dynamic regret in Eq 3. As shown in

---

Algorithm 1 Pseudocode of algorithm REXP4

Inputs: $\gamma \in (0,1]$ and an epoch size Δ_T
1) Set Epoch index $j = 1$
2) Repeat while $j \leq \lceil T/\Delta_T \rceil$
   - Set $\tau = (j-1)\Delta_T$
   - Initialisation: for any expert n set weight $w_n(t) = 1$
   - Repeat for $t = \tau + 1, \ldots, \min\{T, \tau + \Delta_T\}$, Call EXP4 Algorithm[18]
   - Set $j = j + 1$ and return to the beginning of step 2

Algorithm 1, it trades off between the need to remember and forget by breaking the task into batches and applying EXP4 [18] on each batch. As the reward distribution changes, it adapts to the change as by re-estimating each expert's reward distribution at each batch. We show the worst case bound on the regret between this REXP4 procedure and the dynamic oracle.

B. Regret Bound for REXP4

The regret bound for REXP4 is illustrated in the following theorem. The theorem is proved by following the proof structure of [26] and replacing the term μ in [26] with the expected reward term y in our paper.

Theorem 1. *Let π be the REXP4 policy with a batch size $\Delta_T = \lceil (A \log N)^{1/3} (T/V_T)^{2/3} \rceil$ and $\gamma = \min\{1, \sqrt{\frac{A \log N}{(e-1)\Delta_T}}\}$. Then, there is some constant C such that for every $T \geq 1, K \geq 2, N \geq 2$, and $V_T \in [A^{-1}, A^{-1}T]$*

$$R^\pi(\mathcal{V}, T) \leq C(A \log N \cdot V_T)^{1/3} T^{2/3} \tag{4}$$

where $A = \min\{N, K\}$ indicates the smaller number of experts or arms.

The result is an upper bound on the regret between our REXP4 policy and the dynamic oracle. As $A = \min\{N, K\}$, it is favourable if either the number of experts N or arms K is small. This also means it is relatively robust to many arms (as in AL, where arms=data points). If V_T is sub-linear in T (total variation in reward grows slower than timesteps), then performance converges to that of the oracle.

C. Dynamic Ensemble Active Learning

Based on our REXP4 algorithm for MAB with expert advice, we present DEAL-REXP4 (Dynamic Ensemble Active Learning) for active learning based on REXP4. Our dynamic ensemble learner will update both base learner f_t and active criteria weights $\boldsymbol{w}(t)$ iteratively. More specifically, each ensemble criterion will predict scores $\boldsymbol{s}_t^n$ for all unlabelled instances. We use exponential ranking normalisation $-\exp(-\alpha \, rank)$ to avoid the issue of different criterion scales, and apply the Gibbs measure $\frac{\exp(-\beta \boldsymbol{s}_t^n)}{\sum_k \exp(-\beta s_{t,k}^n)}$ where the parameters α, β control the sharpness of the distribution. The $rank$ denotes the ranking position of the instance's score where the ranking order is determined by the criterion strategy's ordering. For example, the entropy criterion prefers

Algorithm 2 DEAL: Dynamic Ensemble Active Learning

Inputs: $\gamma \in (0,1]$, initial weight $\boldsymbol{w}(1) = 1$, $\Delta_T = 10$, $\tau = 1$, labelled set $\mathcal{L}_0$, unlabelled set $\mathcal{U}_0$, initial classifier f_0
for $t = 1 \rightarrow T$ **do**

1) Get scores of instance $\boldsymbol{s}_t^n$ from criteria
2) Normalised the score vector $\boldsymbol{s}_t^n = -\exp(-\alpha \, \boldsymbol{rank})$
3) Obtain the advise vector with $\boldsymbol{\xi}^n(t) = \frac{\exp(-\beta \boldsymbol{s}_t^n)}{\sum_k \exp(-\beta s_{t,k}^n)}$
4) Set $W_t = \sum_{n=1}^N w_n(t)$ and for $k = 1, \ldots, K$ set

$$p_k(t) = (1-\gamma) \sum_{i=n}^N \frac{w_n(t)\xi_k^n(t)}{W_t} + \frac{\gamma}{K}$$

5) Query the label of instance $\boldsymbol{x}_{k_t}$ randomly from $\mathcal{U}_t$ according to probability $p_1(t), \ldots, p_K(t)$
6) Move the instance $\boldsymbol{x}_{k_t}$ from $\mathcal{U}_t$ to $\mathcal{L}_t$
7) Retrain the classifier f_t and receive reward $r_t^k \in [0,1]$
8) For $k = 1, \ldots, K$ set

$$\hat{r}_k(t) = \begin{cases} r_k(t)/p_k(t) & \text{if } k = k_t \\ 0 & \text{otherwise} \end{cases}$$

9) For $n = 1, \ldots, N$ set

$$\hat{y}_t^n = \boldsymbol{\xi}^n(t)^T \hat{\boldsymbol{r}}(t)$$
$$w_n(t+1) = w_n(t)\exp(\gamma \hat{y}_t^n / K)$$

10) $\tau = \tau + 1$
  if $\tau > \Delta_T$ **then** Reset $\tau = 1$ and $\boldsymbol{w}(t+1) = \boldsymbol{1}$
  end if
end for

Fig. 2: Illustration of DEAL System. Light blue: Taking the unlabelled set $\boldsymbol{X}_{\mathcal{U}_t}$ as the input, each expert will output a score that is normalised before input to the DEAL active learner. ξ_K^N is the Nth criterion score of Kth instance. Orange: the active learner to make a decision. Green: updating the labelled set, unlabelled set, and the classifier. Light yellow: The restart detection scheme. Ensemble weights are then updated differently between (light red) or at (dark red) restarts.

points with maximum entropy, so the maximum entropy point has rank 1. Similarly, the minimum margin criterion prefers points with low distance to margin, so the minimum distance point has rank 1. Based on the current suggestions from the criteria members, the active learning ensemble $\boldsymbol{w}(t)$ will select an instance for label querying. Then, the base learner f_{t+1} will be updated with the new labelled data and the active learner $\boldsymbol{w}(t+1)$ will be updated successively based on the performance improvement of the updated base learner. To learn the non-stationary reward distribution, we use our proposed REXP4 algorithm to learn the weights of active learning criteria in an online adaptive way by introducing the restart scheme. Giving the current within-batch index $\tau \in \{1, \cdots, \Delta_T\}$, the restart scheme will be activated when $\tau > \Delta_T$, otherwise updates follow the EXP4 rule. The details are described in Algorithm 2 with an illustration in Fig. 2.

In DEAL-REXP4 we set the reward as the resulting accuracy after a classifier update. Thus in the context of active learning, the bound given in Eq. 4 means that we know that the total area under the reward curve obtained by DEAL-REXP4 is within a bound of the best case scenario that would occur only *if we had known the best criterion to use at each iteration.* Moreover, if the variation budget V_T grows sublinearly with T, DEAL-REXP4 converges towards this best-expert-per-iteration upper bound scenario.
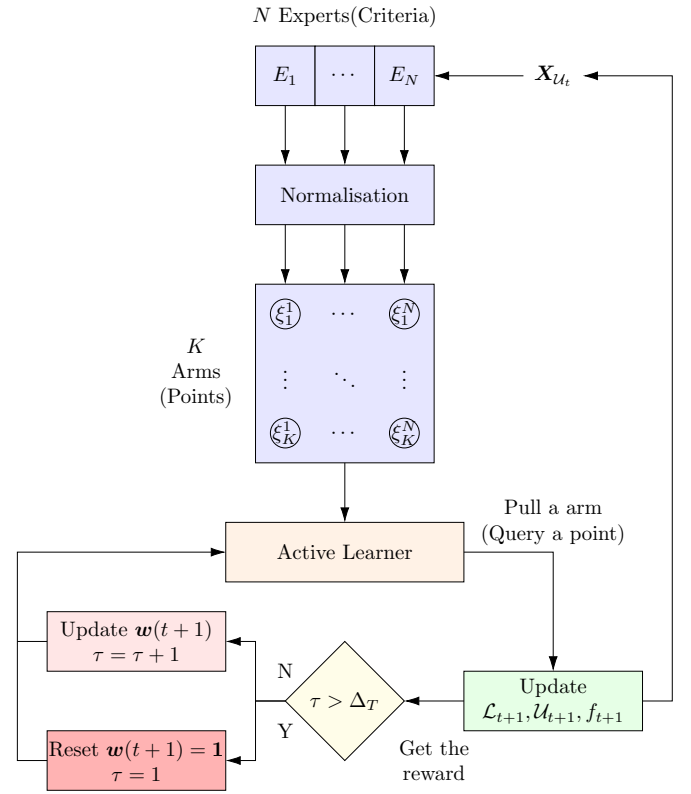
D. Discussion of Static and Dynamic Active Learning

We divide active learning algorithms into static/dynamic based on the stationary/non-stationary assumption on the importance of each criteria over different time periods.

Static Active Learning Single criterion algorithms are all static, since they solve active learning with only one criterion. Regarding active learning algorithms with multiple motivations: if they are formalised as a single fixed mixture of criteria, they are also static. Since the coefficients of different motivations are fixed over all time steps, they assume that a single weighted combination is suitable at any learning stage. For example, Query Informative and Representative Examples (QUIRE) [11], Learning Active Learning (LAL) [33], and Discriminative and Representative Queries for Batch Mode Active Learning (BMDR) [12] are static active algorithms with multiple motivations.

Previously proposed ensemble algorithms ALBL [15], COMB [14], and Linear Strategy Aggregation (LSA) [24] are also static in the sense that, although the weight proportion

TABLE I: Summary of Active Learning Algorithms

Single Criterion

Algorithm	Motivation	Stationarity	Importance of Criterion	Ensemble Members	Property
US [1], [30], [31]	Querying the least confidence	Stationary	Fixed	US	Static
RS [32]	Query a cluster within Margin	Stationary	Fixed	RS	Static
DE [16]	Query the major cluster	Stationary	Fixed	DE	Static

Multiple Criteria

Algorithms	Motivations	Stationarity	Importance of Criterion	Ensemble Members	Property
QUIRE [11]	Combining informativeness and representativeness	Stationary	Equal effect	QUIRE	Static
BMDR [12]	Combining discriminative and representativeness	Stationary	Equal effect	BMDR	Static
LAL [33]	Combining Multiple motivations	Stationary	Equal effect	Any Criteria	Static
DUAL[16]	Switching from DE to US once	Non-stationary	Varying	US, DE	Dynamic
ALGD [17]	Switching between DE to US	Non-stationary	Varying	US, DE	Dynamic

Bandit Ensemble Algorithms

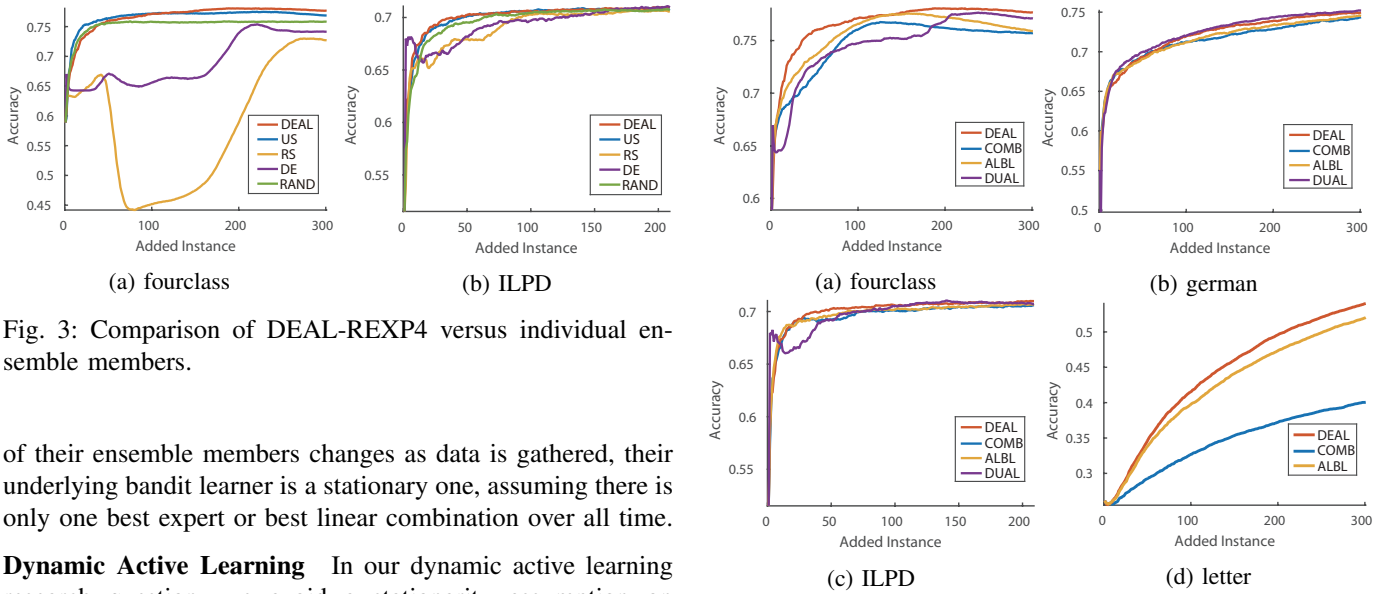| Algorithm	Bandit	Regret	Stationarity	Importance of Criterion	Ensemble Members	Property
COMB [14]	EXP4 [18]	$\max_n \sum_{t=1}^T y_t^n - \mathbb{E}(\sum_{t=1}^T y_t^\pi)$	Stationary	Single best	Any Criteria	Static
ALBL [15]	EXP4.P [19]	$\max_n \sum_{t=1}^T y_t^n - \mathbb{E}(\sum_{t=1}^T y_t^\pi)$	Stationary	Single best	Any Criteria	Static
LSA [24]	LinUCB [23]	$\sum_{t=1}^T r_{a_t^*}(t) - \sum_{t=1}^T r_{a_t}(t)$	Stationary	Single best combination	Any Criteria	Static
DEAL	REXP4	$\sum_{t=1}^T \max_n y_t^n - \mathbb{E}(\sum_{t=1}^T y_t^\pi)$	Non-Stationary	Dynamic best	Any Criteria	Dynamic

Fig. 3: Comparison of DEAL-REXP4 versus individual ensemble members.

(a) fourclass (b) ILPD

Fig. 4: Comparison of active learning with our DEAL-REXP versus alternative state of the art bandit algorithms.

(a) fourclass (b) german (c) ILPD (d) letter

of their ensemble members changes as data is gathered, their underlying bandit learner is a stationary one, assuming there is only one best expert or best linear combination over all time.

Dynamic Active Learning In our dynamic active learning research question, we avoid a stationarity assumption on criteria importance over time. A non-stationary algorithm should adapt its weighting proportions over time in response to learning progress. Prior attempts propose heuristics for classifier switching or reweighting [16], [17] between density and uncertainty sampling. Our DEAL-REXP4 improves on these in that it can use an arbitrary number of criteria of any type beyond 2 specified criteria; and in contrast to prior heuristics, it contains a principled underlying learner with theoretical guarantees. We provide a summary of related prior active learning algorithms in Table I, where the generality and strong notion of regret in DEAL-REXP4 is clear.

IV. EXPERIMENTS AND RESULTS

To evaluate our algorithm, we use 13 datasets from UCI[2] and LibSVM[3] repositories. These datasets are selected following previous relevant papers [24], [15], [11], [6]. We use linear SVM [34] as the base learner. If the datasets do not include

a pre-defined training/testing split, we randomly split 60% for training and the rest for testing. In each trial, we start with 1 randomly labelled point per class. Each experiment is repeated 200 times and the average testing accuracy is reported.

Criteria Ensemble: The ensemble of base learners includes: *US*: picking the instances with max-entropy (min margin) instance in binary class datasets [1], [30] or minimum Best-versus-Second-Best (BvSB) [31] in multiclass datasets. *RS*: clustering the points near the margin [32] then scoring unlabelled points by their distances to the largest centroid. *Distance-Furthest-First (DFF)*: Focuses on exploration by selecting the furthest unlabeled instance to the nearest labeled instance [35]. We use *DFF* which selects the furthest unlabelled instance to the nearest labelled instance [35] to replace the RS in multiclass datasets as originally RS is designed

TABLE II: Win/Tie/Loss counts of DEAL-REXP4 versus ensemble members in terms of AUC at specified learning stage.

Rank	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	Total
1st	0/4/9	0/3/10	0/3/10	0/2/11	0/2/11	0/4/9	0/4/10	0/3/10	0/3/10	0/3/10	0/30/100
2nd	2/6/5	4/5/4	4/6/3	5/4/4	6/3/4	6/3/4	6/3/3	6/4/3	6/4/3	5/6/2	50/45/35
3rd	7/5/1	7/4/2	7/5/1	7/5/1	7/4/2	7/4/2	7/6/0	7/6/0	8/5/0	8/5/0	72/49/9
4th	11/2/0	12/1/0	13/0/0	13/0/0	13/0/0	13/0/0	13/0/0	13/0/0	13/0/0	13/0/0	127/3/0
Total	20/17/15	23/13/16	24/14/14	25/11/16	26/9/17	26/11/15	26/13/13	26/13/13	27/12/13	26/14/12	249/127/144

TABLE III: Win/Tie/Loss counts of DEAL-REXP4 and state of the art alternatives at specified learning stages.

Algorithm	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	Total
Non-Stationary Datasets											
ALBL	8/15/3	9/13/4	9/12/5	9/11/6	7/13/6	6/14/6	6/15/5	4/16/6	4/16/6	4/16/6	66/141/53
COMB	4/13/9	2/14/10	2/13/11	2/12/12	1/12/13	2/12/12	2/13/11	1/15/10	0/16/10	0/16/10	16/136/108
DUAL	7/9/8	7/7/10	7/7/10	8/6/10	9/8/7	8/8/8	6/11/7	7/12/5	7/14/3	7/15/2	73/97/70
DEAL	6/15/5	9/14/3	11/12/3	12/11/3	12/11/3	12/12/2	11/13/2	12/11/3	11/12/3	10/13/3	**106/124/30**
Stationary Datasets											
ALBL	4/6/1	6/3/2	7/3/1	6/3/2	7/2/3	6/3/2	6/3/2	5/5/1	5/5/1	4/6/1	**56/39/15**
COMB	2/4/5	2/2/7	1/4/6	1/2/8	1/2/8	1/2/8	1/3/7	1/4/6	1/4/6	1/4/6	12/31/67
DUAL	0/2/7	3/1/5	3/3/3	5/2/2	5/2/2	5/2/2	4/4/1	4/4/1	4/4/1	4/4/1	37/28/25
DEAL	7/4/0	6/2/3	3/4/4	4/3/4	4/2/5	4/3/4	3/4/4	2/5/4	2/5/4	2/6/3	37/38/35

for binary class datasets. Both are motivated by exploring the datasets, but DFF does not depend on binary classifiers. *Density Estimation (DE):* Picking the instance with maximum density in a GMM with 20 diagonal covariance components [16]. *RAND:* Randomly selecting points can be hard to beat on datasets unsuited to a given criterion. Moreover, including a random expert (for exploration) is necessary to guarantee the performance of the EXP4 subroutine [18], [19].

Competitors: We compare our method to ALBL [15], COMB [14] and DUAL [16]. For COMB, we follow their recommended settings with CEM reward and $\beta = 100$. For the ALBL, we use their settings and importance-weighted accuracy reward.

For direct comparison, ALBL, COMB and REXP4 use the same ensemble of criteria described above. DUAL is engineered for a specific pair of criteria, so we apply its original version using Uncertainty Sampling and Density-Weighted Uncertainty Sampling. It is also only defined for binary classification problems unlike the others.

DEAL-REXP4 Settings: For reward, we follow [15], [24] in using the IWA for unbiased estimation of test accuracy. To produce probabilistic preferences for points from all AL criteria, we use exponential ranking normalisation and a Gibbs measure with $\alpha = 0.1, \beta = 100$. We use batch size $\Delta_T = 10$ throughout. The choice $\Delta_T = 10$ is based on observing the typical coarse duration of performance gaps among different criteria. For example, RS wins first 20 iterations in Fig. 3(b). The reason for parameterizing in terms of Δ_T rather than V_T is that it has intuitive meaning in AL context (batch-size), yet implies a corresponding variation budget for any given T (Theorem 1).

Characterising dataset (non)stationarity: We first investigate each dataset to characterise its (non)stationarity. We use our DEAL trajectory, and use an oracle to measure the % wins of each criterion at each batch Δ_T in terms of performance increase. A dataset with stationary reward distribution would tend to have a consistent winner, and vice-versa. Although (non)stationarity is a continuum, we will describe a dataset as stationary if at least two criteria have a fraction of wins above threshold $\theta = 10\%$.

DEAL versus Individual Criteria Examples comparing the performance of DEAL and individual criteria in the ensemble are shown in Fig. 3. There is no single criterion that works best for all datasets, moreover different criteria are effective at different stages of learning. While DEAL is not best across all datasets and all time-steps (this would require the actual dynamic oracle upper bound), it performs well overall. This is summarised quantitatively across all 13 datasets in Tab. II. Each method's performance is evaluated by the area under the learning curve at different proportions of added instances. The results show the number of wins/ties/losses of DEAL versus the alternative ensemble member of specified highest rank according to two-sided t-test. This shows for example that DEAL often ties with the top-ranked ensemble member (30 draws vs 1st rank), is usually at least as good as the second ranked member (50 wins and 45 ties vs only 35 losses) and is never the worst (0 losses vs 4th rank).

Comparison vs State-of-the-Art We compare our DEAL-REXP4 with state-of-the-art alternatives to tuning an AL-ensemble. Sometimes DUAL performs well, but it is highly variable depending on whether the criterion switch heuristic makes a good choice or not, as seen in Fig. 4. Tab. III summarises the results across all datasets in terms of AUC wins/draws/losses of each approach against the alternatives. DUAL has a lower row-total as it is defined for binary problems only, so not evaluated on wine and letter datasets. The main observation is that DEAL outperforms the alternatives particularly on non-stationary datasets. On stationary datasets we are only slightly worse than ALBL. This is expected as REXP4 performs forgetting in order to adapt to changes in expert efficacy, meaning that we cannot exploit the best criterion as aggressively as ALBL's EXP4.P MAB learner. Nevertheless, overall DEAL is fairly robust to stationary

datasets (small margin behind ALBL), while ALBL is not robust to non-stationary datasets (larger margin behind DEAL).

V. CONCLUSION

We proposed a non-stationary multi-armed bandit with expert advice algorithm REXP4, and demonstrated its application to online learning of a criterion ensemble in active learning. The theoretical results provide bounds on REXP4's optimality. The empirical results show that active learning with DEAL-REXP4 tends to perform near the best criterion in the ensemble. It performs comparable to state of the art alternative ensembles on stationary datasets, and outperforms them on non-stationary datasets.

REFERENCES

[1] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '94. New York, NY, USA: Springer-Verlag New York, Inc., 1994, pp. 3–12.

[2] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Proc. 17th International Conf. on Machine Learning"*, vol. 2, pp. 45–66, Mar. 2002.

[3] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 441–448.

[4] T. M. Hospedales, S. Gong, and T. Xiang, "A unifying theory of active discovery and learning," in *European Conference on Computer Vision*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 453–466.

[5] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds. MIT Press, 1995, pp. 705–712.

[6] R. Chattopadhyay, Z. Wang, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Batch mode active sampling based on marginal probability distribution matching," *ACM Trans. Knowl. Discov. Data*, vol. 7, no. 3, pp. 13:1–13:25, Sep. 2013.

[7] K. Yu, J. Bi, and V. Tresp, "Active learning via transductive experimental design," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 1081–1088.

[8] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT '92. New York, NY, USA: ACM, 1992, pp. 287–294. [Online]. Available: http://doi.acm.org/10.1145/130385.130417

[9] N. Abe and H. Mamitsuka, "Query learning strategies using boosting and bagging," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ser. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 1–9.

[10] C. C. Loy, T. M. Hospedales, T. Xiang, and S. Gong, "Stream-based joint exploration-exploitation active learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[11] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples." in *Advances in Neural Information Processing Systems 23*, 2010, pp. 892–900.

[12] Z. Wang and J. Ye, "Querying discriminative and representative samples for batch mode active learning," *ACM Trans. Knowl. Discov. Data*, vol. 9, no. 3, pp. 17:1–17:23, Feb. 2015.

[13] Z. Wang, B. Du, L. Zhang, L. Zhang, and X. Jia, "A novel semisupervised active-learning algorithm for hyperspectral image classification," *IEEE Trans. Geoscience and Remote Sensing*, vol. 55, no. 6, pp. 3071–3083, 2017.

[14] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *Journal of Machine Learning Research*, vol. 5, pp. 255–291, Dec. 2004.

[15] W.-N. Hsu and H.-T. Lin, "Active learning by learning," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15. AAAI Press, 2015, pp. 2659–2665.

[16] P. Donmez, J. G. Carbonell, and P. N. Bennett, "Dual strategy active learning," in *Proceedings of ECML 2007*. Springer Verlag, September 2007.

[17] T. M. Hospedales, S. Gong, and T. Xiang, "Finding rare classes: Active learning with generative and discriminative models." *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 2, pp. 374–386, 2013.

[18] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM*, vol. 32, no. 1, pp. 48–77, 2002.

[19] A. Beygelzimer, J. Langford, L. Li, L. Reyzin, and R. E. Schapire, "Contextual bandit algorithms with supervised learning guarantees." in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 19–26.

[20] J. Langford and T. Zhang, "The epoch-greedy algorithm for multi-armed bandits with side information," in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 817–824.

[21] S. M. Kakade, S. Shalev-shwartz, and A. Tewari, "Efficient bandit algorithms for online multiclass prediction," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 440–447.

[22] A. L. Strehl, C. Mesterharm, M. L. Littman, and H. Hirsh, "Experience-efficient learning in associative bandit problems," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 889–896.

[23] W. Chu, L. Li, L. Reyzin, and R. E. Schapire, "Contextual bandits with linear payoff functions," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 208–214.

[24] H. Chu and H. Lin, "Can active learning experience be transferred?" in *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, 2016, pp. 841–846.

[25] A. Garivier and E. Moulines, "On upper-confidence bound policies for non-stationary bandit problems," in *Algorithmic Learning Theory*, 2008.

[26] O. Besbes, Y. Gur, and A. Zeevi, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 199–207.

[27] J. Y. Yu and S. Mannor, "Piecewise-stationary bandit problems with side observations," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09, 2009, pp. 1177–1184.

[28] C.-Y. Wei, Y.-T. Hong, and C.-J. Lu, "Tracking the best expert in non-stationary stochastic environments," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 3972–3980.

[29] O. Besbes, Y. Gur, and A. J. Zeevi, "Non-stationary stochastic optimization," *Operations Research*, vol. 63, pp. 1227–1244, 2015.

[30] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.

[31] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification." in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[32] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang, "Representative sampling for text classification using support vector machines," in *Advances in Information Retrieval*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 393–407.

[33] K. Konyushkova, R. Sznitman, and P. Fua, "Learning active learning from data," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4225–4235.

[34] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[35] D. S. Hochbaum and D. B. Shmoys, "A best possible heuristic for the k-center problem," *Math. Operations Research*, vol. 10, no. 2, pp. 180–184, 1985.