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

Citation for published version:

Digital Object Identifier (DOI):
10.1109/ICPR.2018.8545422

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR)

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.
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 algorithm (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 \( D = \{ x_1, l_1, \ldots, x_M, l_M \} \) where the instances are \( x_i \in R^d \) and the labels are \( l_i \in \{ 1, \ldots, C \} \). In an active learning scenario, the data \( D \) are initially a labelled set \( L \) and unlabelled set \( U = D \setminus L \) where \( |L| \ll |U| \). Training an initial classifier \( f_0 \) on the samples in the initial set \( L \), the algorithm starts to query instances \( x_i \) from \( U \) during iterations \( t = 1, \ldots, T \). After the supervision \( l_q \) of instance \( x_i \) is obtained, \( x_i \) is removed from the unlabelled set \( U \) and added to the labelled set \( L \), 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 \( K = \{ 1, \ldots, K \} \) of slot machines in a sequence of time steps \( T = \{ 1, \ldots, T \} \) 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 formalised 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: \( \max_{\theta} \sum_{t=1}^{T} r_k(t) - E(\sum_{t=1}^{T} r^\pi(t)) \).

Contextual Multi-armed Bandit The goal of contextual bandits is to build a relationship between available context information \( h_t \in R^d \) and the reward distribution over all arms. For example, LinUCB [23] makes the linear realizability assumption that there exists an unknown weight vector \( \theta^* \in R^d \) with \( | |\theta^*|| \leq 1 \) so that regret \( \sum_{t=1}^{T} r_k(t) - \sum_{t=1}^{T} r_{\theta^*}(t) \) is minimized, where \( r_{\theta^*}(t) = \theta^\top h_t \) and \( r_{\theta^*}(t) = \theta^\top h_t \).

However, learning to predict the reward for each data point accurately appears to be an even harder problem given the

---

\(^1\)We use terminology from [18]. It also has other names, including ‘contextual bandit’ [19], [20], ‘partial-label problem’ [21], and ‘associative bandit problem’ [22].
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 $\xi^n(t) \in [0, 1]^K$ 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:

$$\max_n \sum_{t=1}^T y_n^t - \mathbb{E}(\sum_{t=1}^T y_n^t)$$

(1)

where $y_n^t = \sum_{k=1}^K \xi_k^n(t) \times r^k(t)$ is the expected reward of an expert and $y_n^t$ 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 $\xi^n(t)$ 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 $n$th expert $y_n^t = \xi^n(t)r^t(t)$ 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 $y_n^t$ of each expert $n$ can change at any time step $t$. The total variation of the expected reward over all $T$ steps is

$$\sum_{t=1}^{T-1} \sup_n |y_n^t - y_n^{t+1}|$$

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.

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

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**

$$R^\mathcal{V}(V,T) = \sup_{\mathbf{y} \in \mathcal{V}} \left\{ \sum_{t=1}^{T} y_n^t - \mathbb{E}^T \left( \sum_{t=1}^{T} y_n^t \right) \right\}$$

where $y_n^t = \max_n y_n^t$ 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, 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 $\mu$ in [26] with the expected reward term $y$ in our paper.

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

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

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 $w(t)$ iteratively. More specifically, each ensemble criterion will predict scores $s^n_t$ for all unlabelled instances. We use exponential ranking normalisation $-\exp(-\alpha \text{ rank})$ to avoid the issue of different criterion scales, and apply the Gibbs measure $\sum_{s} \exp(-\beta s)$, where the parameters $\alpha, \beta$ 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
Moreover, if the variation budget grows sub-linearly with $T$, DEAL-REXP4 converges towards this best-expert-per-iteration upper bound scenario.

Algorithm 2 DEAL: Dynamic Ensemble Active Learning

**Inputs:** $\gamma \in (0, 1]$, initial weight $w(1) = 1$, $\Delta_T = 10$, $\tau = 1$, 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 $s^n_{t}$ from criteria
2) Normalise the score vector $s^n_{t} = -\exp(-\alpha \text{rank})$
3) Obtain the advise vector with $\xi^n(t) = \frac{\sum \exp(-\beta s^n_{t,k})}{\sum \exp(-\beta s^n_{t,k})}$
4) Set $W_t = \sum_{n=1}^{N} w_n(t)$ and for $k = 1, \ldots, K$ set
\[
p_k(t) = (1 - \gamma) \frac{w_n(t) \xi^n_k(t)}{W_t} + \frac{\gamma}{K}
\]
5) Query the label of instance $x_{k_n}$ randomly from $\mathcal{U}_t$ according to probability $p_1(t), \ldots, p_K(t)$
6) Move the instance $x_{k_n}$ from $\mathcal{U}_t$ to $\mathcal{L}_t$
7) Retrain the classifier $f_t$ and receive reward $r^k_t \in [0, 1]$
8) For $k = 1, \ldots, K$ set
\[
\tilde{r}_k(t) = \begin{cases} 
    r_k(t)/p_k(t) & \text{if } k = k_n \\
    0 & \text{otherwise}
\end{cases}
\]
9) For $n = 1, \ldots, N$ set
\[
\hat{y}^n_t = \xi^n(t) \tilde{r}_k(t)
\]
\[
w_n(t+1) = w_n(t) \exp(\gamma \hat{y}^n_t/K)
\]
10) $\tau = \tau + 1$
if $\tau > \Delta_T$ then Reset $\tau = 1$ and $w(t + 1) = 1$
end if
end for

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Bandit Ensemble Algorithms</th>
<th>Motivations</th>
<th>Stationarity</th>
<th>Importance of Criterion</th>
<th>Ensemble Members</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>US [1], [30], [31]</td>
<td>EXP4 [18]</td>
<td>max_{a \in A} \sum_{t=1}^{T} y^a_t - E(\sum_{t=1}^{T} y^*_t)</td>
<td>Stationary</td>
<td>Single best</td>
<td>Any Criteria</td>
<td>Static</td>
</tr>
<tr>
<td>RS [32]</td>
<td>EXP4 [19]</td>
<td>max_{a \in A} \sum_{t=1}^{T} y^a_t - E(\sum_{t=1}^{T} y^*_t)</td>
<td>Stationary</td>
<td>Single best</td>
<td>Any Criteria</td>
<td>Static</td>
</tr>
<tr>
<td>DE [16]</td>
<td>LSA [23]</td>
<td>\sum_{t=1}^{T} \max_{a \in A} y^a_t - E(\sum_{t=1}^{T} y^*_t)</td>
<td>Stationary</td>
<td>Single best combination</td>
<td>Any Criteria</td>
<td>Static</td>
</tr>
<tr>
<td>COMB [14]</td>
<td>DEAL [17]</td>
<td>\sum_{t=1}^{T} \max_{a \in A} y^a_t - E(\sum_{t=1}^{T} y^*_t)</td>
<td>Non-Stationary</td>
<td>Dynamic best</td>
<td>Any Criteria</td>
<td>Dynamic</td>
</tr>
</tbody>
</table>

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

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$^2$ 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 unlabelled 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

---

$^2$https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html
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 $\Delta_T$ 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 $\Delta_T$ 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 $\theta = 10\%$.

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>5%</th>
<th>10%</th>
<th>15%</th>
<th>20%</th>
<th>25%</th>
<th>30%</th>
<th>35%</th>
<th>40%</th>
<th>45%</th>
<th>50%</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALBL</td>
<td>20</td>
<td>13</td>
<td>2</td>
<td>4</td>
<td>12</td>
<td>13</td>
<td>13</td>
<td>8</td>
<td>12</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>COMB</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>13</td>
<td>3</td>
<td>9</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>DUAL</td>
<td>7</td>
<td>7</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>11</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DEAL</td>
<td>12</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>5%</th>
<th>10%</th>
<th>15%</th>
<th>20%</th>
<th>25%</th>
<th>30%</th>
<th>35%</th>
<th>40%</th>
<th>45%</th>
<th>50%</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALBL</td>
<td>20</td>
<td>13</td>
<td>2</td>
<td>4</td>
<td>12</td>
<td>13</td>
<td>13</td>
<td>8</td>
<td>12</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>COMB</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>13</td>
<td>3</td>
<td>9</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>DUAL</td>
<td>7</td>
<td>7</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>11</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DEAL</td>
<td>12</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

**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


