Expertise Drift in Referral Networks

Ashiqur R. KhudaBukhsh
Carnegie Mellon University
Pittsburgh, Pennsylvania
akhudabu@cs.cmu.edu

Jaime G. Carbonell
Carnegie Mellon University
Pittsburgh, Pennsylvania
jgc@cs.cmu.edu

ABSTRACT

Learning-to-refer is a challenge in expert referral networks, wherein Active Learning helps experts (agents) estimate the skills of other connected experts for different categories of tasks that the initial expert cannot solve and therefore must seek referral to experts with more appropriate expertise. Prior research has investigated different reinforcement action selection algorithms to assess viability of the learning setting both with uninformative priors and with partially available noisy priors, where experts are allowed to advertise a subset of their skills to their colleagues. Prior to this work, time-varying expertise drift (e.g., experts learning with experience) has not been considered though it is an aspect that may often arise in practice. This paper addresses the challenge of referral learning with time-varying expertise, proposing `Hybrid`, a novel combination of `Optimistic Thompson Sampling`, `Pessimistic Thompson Sampling` and Distributed Interval Estimation Learning (`DIEL`). In our extensive empirical evaluation, considering both biased and unbiased drift, the proposed algorithm outperforms the previous state-of-the-art (`DIEL`) and approaches the drift-aware oracle upper bound.

KEYWORDS

Active Learning; Referral Networks; Expertise Drift

ACM Reference Format:
Ashiqur R. KhudaBukhsh and Jaime G. Carbonell. 2018. Expertise Drift in Referral Networks. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

Learning-to-refer in referral networks is a recently proposed Active Learning challenge where experts (teachers or autonomous agents) can redirect difficult instances (or problems) to colleague experts based on estimates of the colleagues' topic-conditioned skills. Such a network of experts is common in human professional networks such as in clinical contexts and also in consultancy firms. Recent work [20, 21] has compared a wide variety of referral learning algorithms in the stationary expertise setting, i.e., where distributional parameters of expertise do not change over time. In this setting, Distributed Interval Estimation Learning (`DIEL`), a simple yet effective algorithm, was found to outperform `UCB` variants [3, 4], `Q-Learning` [13, 27] and ϵ-`Greedy` algorithms on both real and synthetic data [20]. A different direction along the lines of adversarial Machine Learning research [5, 14], [17–19] has proposed algorithms to work with partially available noisy priors and mechanisms to truthfully elicit such priors. However, none of the past works addressed time-varying expertise that often arise in real world; expertise may change via refinement of existing skills, acquisition of newer skills, decay of unpracticed skills, and could possibly depend on practical factors like fatigue, workload etc. Learning to track drifting expertise of colleagues in a referral network is the primary focus of this paper.

The *partial information* [6] or the *information obstacle* [5] present in *multi-armed bandit* (MAB) settings (a gambler trying to maximize the total amount of reward she receives by pulling one of the k arms at a time, each arm has an unknown reward distribution) is a key challenge in referral networks too. When an expert refers a task to a colleague, there is no way to know how other colleagues would have performed on the same task. Moreover, local visibility of rewards, and the distributed nature of learning, i.e., each expert is independently estimating topical expertise of her colleagues, contributes to the challenges of *learning-to-refer*. For practical viability, early-learning-phase performance gain is crucial and over a large network, as we cannot afford an unbounded number of samples to estimate topical expertise. Understandably, *learning-to-refer* becomes even more challenging with non-stationary expertise since weak experts who were discarded for future consideration on any given topic, could gain expertise over time, becoming real contenders who should not be ignored at a later time in optimizing referral decisions.

Our contributions are the following: First, we introduce time-varying expertise in referral networks, a practical consideration not previously addressed in the literature to the best of our knowledge. Second, in addition to bidirectional drift, the typical drift model in the literature, we also consider drift with positive bias, where agents mostly improve with practice. Third, we widen the pool of referral learning algorithms by including `Thompson Sampling` and its variants, an important set of algorithms with known finite-time regret bounds, used in practical applications, and with a sustained interest in the research community [2, 7, 11, 23]. Finally, we propose `Hybrid`, a novel algorithm combining components from `DIEL`, the state-of-the-art referral learning algorithm, and a conservative approach used in `Thompson Sampling`. There is little established theoretical basis for the dynamic MAB setting (for example, `Dynamic Thompson Sampling` [12] has no known finite-horizon regret bound and `DIEL` (which outperforms `UCB` variants) is based on earlier algorithms with no known finite-horizon regret bound even in the static case). However, this paper is geared towards the design of a learning algorithm robust to expertise drift in referral networks, a challenging problem not previously studied, rather than a theoretical analysis. Our empirical evaluation indicates that our proposed hybrid algorithm is more robust to expertise drift and tracks drift better than `DIEL` or `Thompson Sampling` at the network

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

level, improving overall referral accuracy. Although our primary focus is on referral networks where aggregate performance of the network is the measuring index, Hybrid's strong network-level performance encourages us to believe our work is applicable to the broader and more general context of multi-armed bandit setting with non-stationary reward distributions.

The rest of the paper is organized as follows. Previous work is summarized in Section 2. Section 3 presents our preliminaries on referral networks, key assumptions, and modeling choices for expertise drift. Sections 4, 5 and 6 describe the distributed learning algorithms we used for comparison, our experimental setup, and the results. We end with some general conclusions and an outlook on future work in section 7.

2 RELATED WORK

The referral learning framework was first proposed in [21], and subsequently extended [20] with performance comparisons over of a wide range of competing algorithms, multi-hop referrals, and consideration of practical factors such as capacity constraints and evolving networks. [20, 21] considered non-informative priors. In an augmented setting [17, 18], similar to the line of research in multi-armed bandits with history where algorithms do not start from scratch [25], experts are allowed a one time local network advertisement of a subset of their skills which essentially extends the setting to partially available noisy priors where eliciting truthful advertisements and effectively initializing with the available priors are the primary challenges. In this paper, we work with the uninformative prior setting and consider time-varying expertise which none of the previous works in referral networks has addressed. Our results expose DIEL's, the state-of-the-art referral learning algorithm's susceptibility to expertise drift as we propose new algorithms that demonstrate superior tracking of drifting experts.

Prior work on Interval Estimation Learning (the basic building block of DIEL) to track time-varying accuracy [8] used a particle filtering approach. Whereas this approach is elegant, it is infeasible in our case because it requires a large number of samples even for a single central learner, and the distributed nature of learning by each member of the referral network only exacerbates the problem.

From each expert's point of view, the core problem of learning appropriate referrals for a given topic is viewed as a mutli-armed bandit (MAB) problem where referral choices are the arms. In the MAB literature, time-varying reward distributions were introduced in [28]. Dynamic Thompson Sampling [12], an extension of Thompson Sampling [26], were suggested for these *restless bandits*. Our work is different from previous *restless bandits* literature in the following ways: First, there is an obvious difference in scale as we are dealing with multiple agents learning several threads of referral policies for each topic and none of these algorithms for *restless bandits* has been used in the context of referral learning before nor in similar distributed network learning. Second, for modelling drift in reward distribution, Brownian perturbation [12] was used. However, we notice that human expertise often improve with time and hence require considering positively biased drifts. We also present a less common approach in tackling drifts including concept drift [10] where the most popular approaches are window-based [12].

Previous research on referral networks primarily focused on IEL-based algorithm (Interval Estimation Learning) [15], UCB (Upper Confidence Bound) class of algorithms [1, 3, 4, 22], ϵ-Greedy and its variants [4, 20], and Q-Learning algorithms [13, 27]. In our work, we compare Thompson Sampling, an algorithm with a long history [26] that has received a recent surge of interest with proofs on finite-horizon bounds [2, 16], empirical evidence of strong performance [7] and practical application. While Optimistic Thompson Sampling, a variant of Thompson Sampling, has been previously proposed in the literature, to the best of our knowledge, we are showing for the first time that Pessimistic Thompson Sampling, a counter-intuitive action selection strategy, when combined with the variance term present in DIEL, could be a viable algorithm robust to expertise drift. Our proposed algorithm, a combination of two Thompson Sampling variants and DIEL, uses a performance gradient-based switching criterion between the algorithms similar to [9].

3 REFERRAL NETWORK

3.1 Preliminaries

Essentially, a *referral network* is a graph (V, E) of size k; each vertex v_i corresponds to an expert e_i $(1 \leq i \leq k)$ and each bidirectional edge $\langle v_i, v_j \rangle$ represents a *referral link* indicating e_i and e_j can refer problem instances to each other. A *subnetwork* of expert e_i is the set of her colleagues, i.e., the set of experts linked to an expert e_i by a referral link. A *referral scenario* consists of a set of m instances $(q_1, \ldots, q_m)$ belonging to n topics $(t_1, \ldots, t_n)$ are to be addressed by the k experts $(e_1, \ldots, e_k)$.

For a per-instance query budget of $Q = 2$, the referral mechanism for a task (we use task and instance interchangeably) q_j consists of the following steps.

(1) A user (learner) issues an *initial query* to an expert e_i (*initial expert*) chosen uniformly at random from the network.
(2) Expert e_i examines q_j and solves it if able and communicates the solution to the learner. This depends on the *expertise* (defined as the probability that e_i can solve q_j correctly) of e_i wrt. q_j.
(3) If not, she issues a *referral query* to a *referred expert* within her subnetwork. The *Learning-to-refer* challenge is improving the estimate of who is most likely to solve the problem.
(4) If the referred expert succeeds, she communicates the solution to the initial expert, who in turn, communicates it to the user.

Note that if the per-instance query budget >2, the recipient of a referral can herself re-refer to another expert.

We follow the same set of *assumptions* made in [21] a detailed description of which can be found in [18], but we remove the stationarity assumption on individual expert skills per topic. Some of the important assumptions are: the network connectivity depends on (cosine) similarity between the topical expertise, and the distribution of topical-expertise across experts can be characterized by a mixture of Gaussian distributions. We made the modeling choice regarding network connectivity because of the general observation that people sharing common expertise areas are more likely to know each other. Gaussian distribution is widely used to model real-valued random variables (e.g., height, weight,

expertise) in natural and social sciences. For topical-expertise distribution, we considered a mixture of two Gaussians (with parameters $\lambda = \{w_i^t, \mu_i^t, \sigma_i^t\}\ i = 1, 2.$). One of them ($\mathcal{N}(\mu_2^t, \sigma_2^t)$) has a greater mean ($\mu_2^t > \mu_1^t$), smaller variance ($\sigma_2^t < \sigma_1^t$) and lower mixture weight ($w_2^t << w_1^t$). Intuitively, this represents the expertise of experts with specific training for the given topic, contrasted with the lower-level expertise of the layman population.

3.2 Expertise Drift

In previous work, [20, 21], the expertise of an expert e_i on $topic_p$ was modeled as a truncated Gaussian distribution with small variance:

$$expertise(e_i, q_j \in topic_p) \sim \mathcal{N}(\mu_{topic_p,e_i}, \sigma_{topic_p,e_i}),$$
$$\forall p, i : \sigma_{topic_p,e_i} \leq 0.2, 0 \leq \mu_{topic_p,e_i} \leq 1.$$

We use a truncated Gaussian since *expertise* is a probability, it must remain within $[0, 1]$. Small variance implies an expert's within-topic expertise does not vary by a large amount. In a time-varying expertise setting, expertise of an expert e_i on $topic_p$ is expressed as

$$expertise(e_i, q_j \in topic_p) \sim \mathcal{N}(\mu_{topic_p,e_i,epoch_k}, \sigma_{topic_p,e_i}),$$
$$\mu_{topic_p,e_i,epoch_{k+1}} = \mu_{topic_p,e_i,epoch_k} + \mathcal{N}(\mu_{drift}, \sigma_{drift})$$

For convenience, we assume discrete changes at epoch boundaries, and within a given epoch, we assume the distributional parameters on expertise do not change. The epochs can be small, approximating continuous change. When μ_{drift} is 0, the unbiased drift is similar to the Brownian perturbation previously considered in [12]. The epochs can have arbitrary length and an expert has no knowledge of the epoch-lengths of their colleagues. After every discrete change, we ensure that $\mu_{topic_p,e_i,epoch_{k+1}}$ always remains within $[0, 1]$ by setting it to 0 (or 1) if it is less than 0 (or greater than 1). Once $\mu_{topic_p,e_i,epoch_{k+1}}$ reaches the boundary (0 or 1), we assume that it remains there until drift in the opposite direction moves it away from the boundary.

The expertise of people often improve over time by acquiring a new skill, explicit learning on how to improve a skill, or just practice through solving more problems. We consider this case in our positive-bias drifts (with $\mu_{drift} > 0$), where the overall expertise of the experts in the network improves on certain topics over time.

3.3 Reward Assumptions

From the point of view of a single expert, for a given topic, learning referral policy maps to the classic *multi-armed bandit setting* with each arm corresponds to a referral choice, and similar to the unknown reward distributions of the arms, the expertise of the colleagues is not known in this case. In order to learn an effective referral strategy, whenever an expert refers a task to her colleague, and depending on the outcome of the task, she assigns a reward to the referred colleague. The computational aspect (e.g., what type of information regarding the sequence of rewards is necessary?, how to score an expert depending on her past performance?) of the referral decision is described in our following section, here we outline the main assumptions related to rewards.

All our rewards are

- **bounded**: All our rewards are bounded within the the range $[0, 1]$. In all our experiments, we considered binary rewards, with a failed and successful task receiving a reward of 0 and 1, respectively.
- **i.i.d**: The reward for a given expert on a specific instance belonging to a topic is independent of any reward observed from any other experts and any reward or sequence of rewards belonging to that topic or any other topic by the same expert.
- **locally assigned and locally visible**: Rewards are both locally assigned and locally visible. For example, $reward(e_i, t, e_j)$, a function of initial expert e_i, referred expert e_j and topic t, is assigned by e_i and visible to e_i only.

4 DISTRIBUTED REFERRAL LEARNING

As we already mentioned, considering a single expert and a given topic, *learning-to-refer* is an action selection problem (the problem of selecting an appropriate referral maps to selecting an effective arm in the *multi-armed bandit* setting). In a distributed setting, each expert maintains an action selection thread for each topic in parallel. In order to describe an action selection thread, we first fix topic to T and expert to e.

Let $q_1, \ldots, q_N$ be the first N referred queries belonging to topic T issued by expert e to any of her K colleagues denoted by $e_1, \ldots, e_K$. For each colleague e_i, e maintains a reward vector $\mathbf{r}_{i,n_i}$ where $\mathbf{r}_{i,n_i} = (r_{i,1}, \ldots, r_{i,n_i})$, i.e., the sequence of rewards observed from expert e_i on issued n_i referred queries. Understandably, $N = \sum_{i=1}^{K} n_i$. Let $m(e_i)$ and $s(e_i)$ denote the sample mean and sample standard deviation of these reward vectors. Some of the algorithms we consider require initializing these reward vectors; we will explicitly mention any such initialization. In addition to the reward vectors, for each colleague e_i, e maintains S_{e_i} and F_{e_i} where S_{e_i} denotes the number of observed successes (reward = 1) and F_{e_i} denotes the number of observed failures (reward = 0). Clearly, without any initialization of the reward vectors, $\forall(S_{e_i} + F_{e_i}) > 0$, $m(e_i) = \frac{S_{e_i}}{S_{e_i}+F_{e_i}}$ (i.e., empirical mean is the ratio of total number of observed successes and total number of observations).

Like any other action selection problem, *learning-to-refer* also poses the classic exploration-exploitation trade-off: on one hand, we would like to refer to an expert who has performed well in the past on this topic (exploitation), while ensuring enough exploration to make sure we are not missing out on stronger experts. We next provide a short description of different action selection algorithms that include `DIEL`, the state-of-the-art, `Thompson Sampling`, a well-known high-performance action selection algorithm previously not used in the context of referral learning, and building blocks of our proposed algorithm, `Hybrid`. At a high level, each of the algorithms computes a score for every expert e_i (denoted by $score(e_i)$) and selects the expert with the highest combined score breaking any remaining ties randomly.

4.1 Action Selection Algorithms

DIEL: Distributed Interval Estimation Learning (`DIEL`) is the known state-of-the-art referral learning algorithm [20]. At each step, `DIEL` [18]

selects the expert e_i with highest $m(e_i) + \frac{s(e_i)}{\sqrt{n_i}}$. Every action is initialized with two rewards of 0 and 1, allowing us to initialize the mean and variance.

The intuition behind selecting an expert with a high expected reward ($m(e_i)$) and/or a large amount of uncertainty in the reward ($s(e_i)$) is the following. A large variance implies greater uncertainty, indicating that the expert has not been sampled with sufficient frequency to obtain reliable estimates. Selecting such an expert is an *exploration step* which will increase the confidence of e in her estimate. Also, such steps have the potential of identifying a highly skilled expert. Selecting an expert with a high $m(e_i)$ amounts to exploitation. Initially, the choices made by e tend to be explorative since the intervals are large due to the uncertainty of the reward estimates. With an increased number of samples, the intervals shrink and the referrals become more exploitative.

Algorithm 1: DIEL(e, T)

Initialization: $\forall i, n_i \leftarrow 2, \mathbf{r}_{i,n_i} \leftarrow (0, 1)$
Loop: Select expert e_i who maximizes
$score(e_i) = m(e_i) + \frac{s(e_i)}{\sqrt{n_i}}$
Observe reward r
Update $\mathbf{r}_{i,n_i}$ with r, $n_i \leftarrow n_i + 1$

Thompson Sampling (TS): At each step, for each expert e_i, TS first samples θ_i from $Beta(S_{e_i} + 1, F_{e_i} + 1)$. Next, TS selects the action with highest θ_i. When the number of observations is 0, θ_i is sampled from $Beta(1, 1)$, which is $U(0, 1)$; this makes all colleagues equally likely to receive referral. As the number of observations increases, the distribution for a given expert becomes more and more centered around the empirical mean favoring experts with better historical performance.

Next, we describe the basic building blocks of our primary contribution, including a switching algorithm between two action selection strategies. We first start with two Thompson Sampling variants: Optimistic Thompson Sampling and Pessimistic Thompson Sampling. Optimistic Thompson Sampling is an existing variant [24] while Pessimistic Thompson Sampling is a counter-intuitive sampling strategy without any prior mention in the literature. However, we found this strategy to be useful when combined with DIEL forming Pessimistic TS-DIEL described next.

Optimistic Thompson Sampling (Optimistic TS): Optimistic TS is very similar to TS with an additional restriction: θ_i is never allowed to be less than the mean observed reward $m(e_i)$; θ_i is set to $m(e_i)$ whenever it is less than $m(e_i)$ (in the boundary condition when number of observed samples is zero, $m(e_i)$ is considered to be zero). The reason this sampling technique is called optimistic is because this technique always assumes that the true mean is at least as high as the sampled mean. Note that, each time we refer to e_i where $\theta_i > m(e_i)$, we are essentially performing an *exploration step*.

Pessimistic Thompson Sampling (Pessimistic TS): Pessimistic TS behaves the opposite way to Optimistic TS: θ_i is never allowed to be greater than the mean observed reward $m(e_i)$ and is set to $m(e_i)$ whenever it is greater than $m(e_i)$. Note that, each time we select an expert e_i where $\theta_i < m(e_i)$, we are essentially performing an *exploitation step*. Also, notice that without any initialization of the mean, if any action fails at the first execution, it will never be chosen again. To circumvent this deficiency, the mean of every action is initialized the same way as DIEL, enabling the possibility of future selection.

Algorithm 3: TS(e, T)

Initialization: $\forall i, S_{e_i} \leftarrow 0, F_{e_i} \leftarrow 0$
Loop: Select expert e_i who maximizes
$score(e_i) = \theta_i$
Observe reward r
if $r == 1$ **then**
$S_{e_i} \leftarrow S_{e_i} + 1$
else
$F_{e_i} \leftarrow F_{e_i} + 1$
end

Algorithm 4: Optimistic TS(e, T)

Initialization: $\forall i, S_{e_i} \leftarrow 0, F_{e_i} \leftarrow 0$
Loop: Select expert e_i who maximizes
$score(e_i) = max(\theta_i, m(e_i))$
Observe reward r
if $r == 1$ **then**
$S_{e_i} \leftarrow S_{e_i} + 1$
else
$F_{e_i} \leftarrow F_{e_i} + 1$
end

Algorithm 5: Pessimistic TS(e, T)

Initialization: $\forall i, S_{e_i} \leftarrow 1, F_{e_i} \leftarrow 1, n_i \leftarrow 2$, and $\mathbf{r}_{i,n_i} \leftarrow (0, 1)$
Loop: Select expert e_i who maximizes
$score(e_i) = min(\theta_i, m(e_i))$
Observe reward r
if $r == 1$ **then**
$S_{e_i} \leftarrow S_{e_i} + 1$
else
$F_{e_i} \leftarrow F_{e_i} + 1$
end

Pessimistic TS-DIEL: As described in Algorithm 6,this action selection strategy is a novel combination of DIEL and Pessimistic TS. Essentially, the strategy replaces mean observed reward with adjusted θ_i of Pessimistic TS.

Notice that, in presence of expertise drift, having a conservative approach towards estimating the mean could prove beneficial because the empirical (historical) mean may overestimate the true-mean (post drift). We show an extreme two-expert setting to illustrate this. Say, at time $t = 0$ to 10 tasks, $expertise(e_1) = 1$, $expertise(e_2) = 0.75$. At time $t = 11$ tasks and beyond, $expertise(e_1) = 0.66$, $expertise(e_2) = 0.75$. We ran this simulation for 1000 times till $t = 1000$. `DIEL` converged to e_2, the stronger expert (68.3%), substantially less than `pessimistic TS-DIEL` (83.6%).

Algorithm 6: Pessimistic TS-DIEL(e, T)

Initialization: $\forall i, S_{e_i} \leftarrow 1, F_{e_i} \leftarrow 1, n_i \leftarrow 2$, and $\mathbf{r}_{i,n_i} \leftarrow (0, 1)$
Loop: Select expert e_i who maximizes $score(e_i) = min(\theta_i, m(e_i)) + \frac{s(e_i)}{\sqrt{n_i}}$
Observe reward r
Update $\mathbf{r}_{i,n_i}$ with r, $n_i \leftarrow n_i + 1$
  if $r == 1$ **then**
    $S_{e_i} \leftarrow S_{e_i} + 1$
  else
    $F_{e_i} \leftarrow F_{e_i} + 1$
  end

We now describe `Hybrid`, a combination of `Optimistic TS` and `Pessimistic TS-DIEL`.

Hybrid: Initially, `Hybrid` starts as `Optimistic TS` which favors early exploration. If the performance-improvement gradient is low, it switches to favoring exploitation through `Pessimistic TS-DIEL`. The switching criterion is conditioned on topic and described in Algorithm 7. $perf_{w_i}$ is the mean reward obtained in referral-window w_i (set to 100 referrals). If the performance improvement w.r.t. the best so far performance $perf_{best}$, is below a threshold, either `Optimistic TS` has reached saturation, or the performance suffered because of drift and `Hybrid` switches to `Pessimistic TS-DIEL` for subsequent conservative exploitation. In our experiments, we set the value of *threshold* to 0 while noting that the performance wasn't highly sensitive to the choice of value as we observed indistinguishable performance difference with small values in $[+0.05, -0.05]$. Tuning was performed through a parameter sweep on a small background data set generated with similar distributional parameters.

5 EXPERIMENTAL SETUP

Baselines and upper bounds: `DIEL`, the previously-known state-of-the-art referral learning algorithm on non-drift setting, is our baseline. Additionally, we included three `Thompson Sampling` variants and two topical upper bounds for performance comparison. `Thompson Sampling` variants and the `DIEL` version we used [18, 20] are parameter free. The *threshold* parameter of `Hybrid` is set to 0. We considered two upper bounds: `Drift-Blind` and `Drift-Aware`. The `Drift-Aware` upper bound is the performance of a network where every expert has access to an oracle that knows the true topic-mean (i.e., $mean(Expertise(e_i, q) : q \in topic_p)\ \forall i, p$) of every expert-topic pair. The `Drift-Blind` upper bound is the performance of a network where every expert has access to an oracle that only knows the true topic-mean of every expert-topic pair at the beginning of the simulation but is agnostic of any subsequent drift.

Algorithm 7: Hybrid(e, T)

execute `Optimistic TS`
$perf_{best} \leftarrow perf_{w_1}$
$switchFlag \leftarrow 0$
for $i = 2, 3, \ldots$ **do**
  if $switchFlag == 0$ **then**
    execute `Optimistic TS`
    $perf_\Delta \leftarrow perf_{w_i} - perf_{best}$
    if $\text{perf}_\Delta < threshold$ **then**
      $switchFlag \leftarrow 1$
    end
    if $\text{perf}_\Delta > 0$ **then**
      $perf_{best} \leftarrow perf_{w_i}$
    end
  else
    execute `Pessimistic TS-DIEL`
  end
end

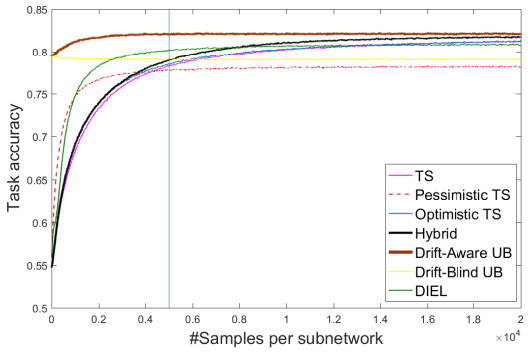
Data set: Our test set for performance evaluation is the same data set used in [17][1], which is a random subset of 200 *referral scenarios* also used in [18, 20, 21]. Each *referral scenario* consists of a network of 100 experts and 10 topics. In our simulation, we start with the same parameter values describing topical expertise of each expert. As the simulation progresses, the expertise drifts according to the drift parameter values are described in Table 1. For modeling expertise drift, we believe a slow, gradual change in expertise is more realistic than abrupt changes. Hence, we considered the distribution for expertise as piece-wise stationary and selected small values for μ_{drift} and σ_{drift}. Recall that in a time-varying expertise setting, expertise of an expert e_i on $topic_p$ is modeled as $\mu_{topic_p, e_i, epoch_{k+1}} = \mu_{topic_p, e_i, epoch_k} + \mathcal{N}(\mu_{drift}, \sigma_{drift})$. We use #samples as a proxy for time as is typical in Machine Learning for evolving or streaming scenarios. For each expert, the epoch boundaries are chosen uniformly at random. The total number of epochs for a given topic is set to 40 (with 10 topics, this essentially means, the total number of time the expertise of an expert changes is 400).

Drift	μ_{drift}	σ_{drift}
weak, unbiased	0	0.03
strong, unbiased	0	0.06
weak, small positive bias	0.005	0.03
strong, small positive bias	0.005	0.06
strong, large positive bias	0.05	0.06

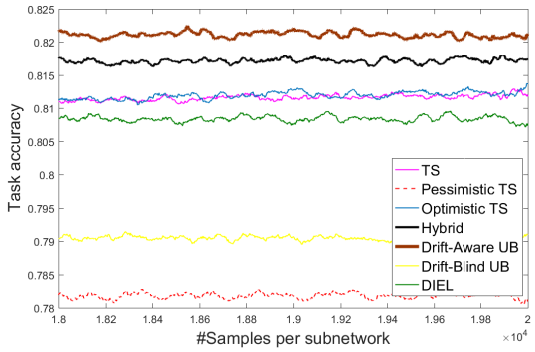
Table 1: Drift parameters

Performance Measure: We use the same performance measure, overall task accuracy of our multi-expert system, as in previous

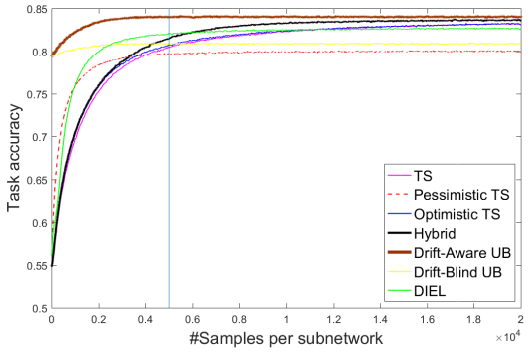
[1]The data set can be downloaded from https://www.cs.cmu.edu/~akhudabu/referral-networks.html



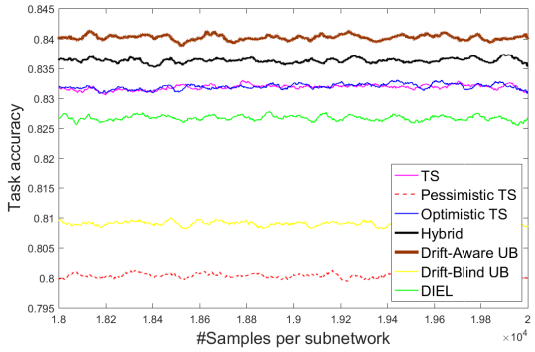
(a) Unbiased, strong drift



(b) Steady-state performance closeup: zooming in to 1(a) upper right portion



(c) Small positive bias, strong drift



(d) Steady-state performance closeup: zooming in to 1(c) upper right portion

Figure 1: Performance comparison of referral learning algorithms.

work in referral networks. So if a network receives n tasks of which m tasks is solved (either by the *initial expert* or the *referred expert*), the overall task accuracy is $\frac{m}{n}$. Q, the per-instance query budget, is set to 2. Each algorithm is run on the data set of 200 referral networks and the average over such 200 simulations is reported in our results section. In order to facilitate comparability, for a given simulation across all algorithms, we chose the same sequence of initial expert and topic pairs; for each expert in a network, the epoch length and expertise shift for each given topic are identical across different referral algorithm runs.

Computational Environment: Experiments were carried out on Matlab R2016 running Windows 10.

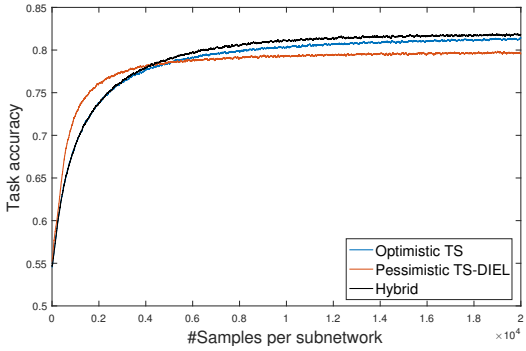


Figure 2: Components of `Hybrid`.

6 RESULTS

Figure 1 compares the performance of referral learning algorithms in the presence of strong drift (weaker drift shows qualitatively similar results). Our results demonstrate the following points:

First, the `Drift-Aware` upper bound outperforms the `Drift-Blind` by a considerable margin, as expected. In fact, apart from `Pessimistic TS`, all algorithms eventually outperformed the `Drift-Blind` upper bound. This underscores the importance of tracking drift in expertise estimation and continual learning, since starting with a perfect information on the topical mean of every expert-topic pair was not enough to overcome expertise-drift tracking, even if starting with imperfect estimates.

Next, we evaluate the relative expertise-tracking performance of algorithms in the literature. The vertical line at 5000 samples per subnetwork marks the horizon considered in previously reported results. Earlier results demonstrated `DIEL` outperformed several algorithms including `UCB` variants, `Q-Learning` variants [20, 21] in the stationary expertise setting. In our new results, we find that even

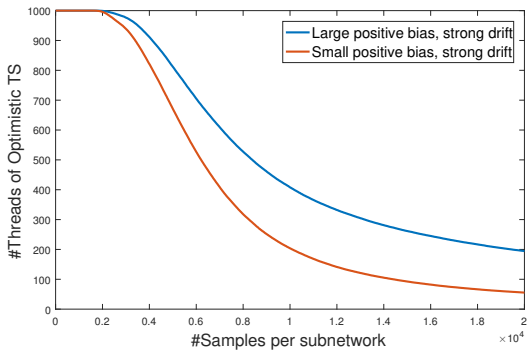


Figure 3: Switching behavior of Hybrid.

in presence of drift, DIEL still outperforms the TS variants when the number of observed samples is small, once again highlighting the early performance gain that made DIEL suitable for multi-hop referral learning and proactive skill posting. However, with a larger number of samples under the expertise-drift condition, we find that both TS algorithms eventually outperform DIEL, thus presenting better long-term steady-state performance, and superior tracking of drifting experts.

Next, we focus on Pessimistic TS, a TS variant never considered in the literature before. As expected, Pessimistic TS performs poorly compared to the other two TS variants indicating that it is not a viable standalone action selection strategy. However, combining DIEL and an effective switching strategy after sufficient exploration proved to be most resilient combination to time-varying expertise. This result shows that combining components from different action selection strategies could result in high-performance algorithms.

Finally, we focus on Hybrid, our primary proposed algorithm. As shown in Figure 2, Hybrid outperformed both its component algorithms by combining the benefit obtained through early exploration of Optimistic TS and subsequent exploitation through Pessimistic TS-DIEL. The effective switching criterion ensured sufficient exploration performed before the switch and less exploration later to continue to track expertise drift. As shown in Figure 1, Hybrid outperforms DIEL, TS and Optimistic TS, the three algorithms from the literature, among which DIEL was the top performer in the stationary expertise setting. The small performance gap between Hybrid and Drift-Aware upper bound indicates that any other referral learning algorithm will have at most little advantage[2].

Note that each expert decides independently when to switch from Optimistic TS to Pessimistic TS-DIEL for each topic. With 10 topics and 100 experts in the network, this effectively means at the beginning, 1000 threads of Optimistic TS are running in parallel. We were curious to see when the strategy switch occurred in aggregate. Figure 3 presents the switching behavior of Hybrid in presence of strong, biased drift. Since the switch only happens if Optimistic TS stops improving significantly, the gradual shift indicates that for different topic-expert pair, that strategy shift arrives at different operating points depending on the composition of the subnetwork around each expert, the expertise of the reachable experts and corresponding drift.

Our results with weak expertise drifts are qualitatively similar. Figure 5 compares the performance of Optimistic TS, DIEL and Hybrid with weak, unbiased drift and shows that the relative orderings found in previous results are preserved (DIEL and Optimistic TS have indistinguishable steady-state performance).

Finally, we present our result with a large positive bias, and strong drift in Figure 4. The relative ordering of previous performance is preserved with both DIEL and Optimistic TS outperforming the Drift-blind upper bound. However, in this case, we found that the drift-tracking of Hybrid is near-perfect as shown in the steady-state close-up in Figure 4(b), where Hybrid is indistinguishable from the drift-aware oracle.

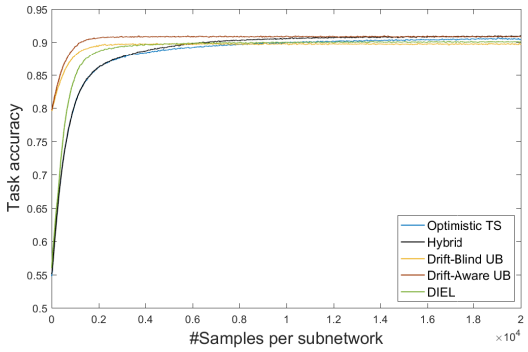
7 CONCLUSIONS AND FUTURE WORK

Learning to refer is a recent Active Learning setting where experts can redirect difficult tasks they cannot solve to other connected experts. In this work, we introduced the notion of time-varying expertise in referral networks, an important practical factor not considered in the literature. Our results indicate that DIEL, the state-of-the-art referral learning algorithm on referral networks without time-varying expertise, is vulnerable to expertise drift. Hence, we proposed a novel combination of Thompson Sampling and DIEL that performs a gradient-based switch between action selection strategies which outperformed DIEL on different types of drift conditions. Moreover, our proposed algorithm achieved a performance close to the theoretical upper bound.

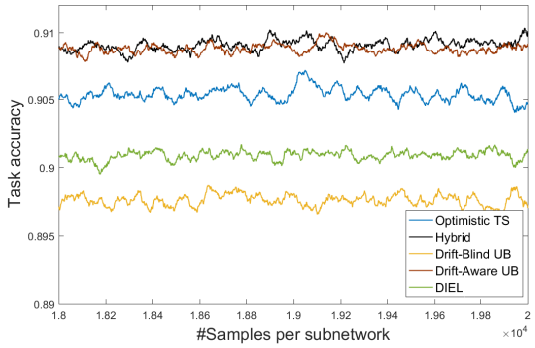
Our work can be extended in the following ways.

- **Evolving networks**: When the estimated $perf_\Delta$ falls below a threshold, the assumption is the exploration component of our hybrid algorithm has largely saturated. However, in evolving networks where new experts can join in and old experts can drop off of the network, this could also mean that a subset of old experts should be replaced by a set of new experts with initially unknown expertise. Distinguishing between the case of network composition change and expertise drift to select the best learning strategy under both conditions presents a new challenge.
- **Topic-dependent drift:** In this work, we assumed the distribution parameters for drift do not vary across topics. However, in real world, some topics may be prone to rapid skills change, whereas others are more stable. It is not yet clear if the proposed methods are robust to a mixture of drift distributions.
- **Expertise-level-dependent drift:** We assumed that the nature of drift is independent of the present expertise. However, in real life, a strong expert is unlikely to lose or improve her skill rapidly, whereas a weak expert may be more likely to substantially improve in a short span of time, i.e. a student rapidly learning to become a true expert. Extending our work to expertise-level-dependent drift could be a possible future direction.

[2]We also tried combinations of Dynamic Thompson Sampling and DIEL with a moving window of observed samples. We did not obtain any perceivable performance benefit. The dynamic versions of any of the TS algorithms also didn't offer any performance benefit.



(a) Large positive bias, large drift



(b) Steady-state performance closeup: zooming in to 4(a) upper right portion

Figure 4: Performance comparison of referral learning algorithms with large positive bias, large drift.

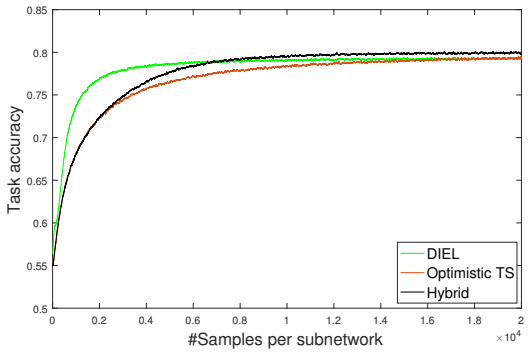


Figure 5: Performance comparison with unbiased weak drift.

- **Finite horizon bound for algorithms with variance term:** In our experimental results both on real data and synthetic data [20], we have found that `DIEL`'s finite horizon performance is substantially better than a wide range of algorithms. In [4], an algorithm `UCB1-tuned` was found to have superior empirical performance than `UCB1`. In this paper, we also found that `Pessimistic TS-DIEL` could be a useful component for dealing with expertise drift. However, none of these algorithms' theoretical finite-horizon regret bounds are known; they all have a variance term in common (which is precisely the reason for the difficulty in proving the finite-horizon regret bound). We would like to attract the attention of the MAB community towards this observation to see whether tight regret bounds might be determined, as many of these algorithms have demonstrated strong performance in practice.

ACKNOWLEDGMENTS

The authors would like to thank Manuel Blum, Jeffrey P. Bigham and anonymous AAMAS reviewers for their constructive feedback to improve the manuscript.

REFERENCES

[1] Rajeev Agrawal. 1995. Sample mean based index policies with O (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability* (1995), 1054–1078.
[2] Shipra Agrawal and Navin Goyal. 2012. Analysis of Thompson Sampling for the Multi-armed Bandit Problem.. In *COLT*. 39–1.
[3] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. 2007. Tuning bandit algorithms in stochastic environments. In *International Conference on Algorithmic Learning Theory*. Springer, 150–165.
[4] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47, 2-3 (2002), 235–256.
[5] Moshe Babaioff, Yogeshwer Sharma, and Aleksandrs Slivkins. 2014. Characterizing truthful multi-armed bandit mechanisms. *SIAM J. Comput.* 43, 1 (2014), 194–230.
[6] Giuseppe Burtini, Jason Loeppky, and Ramon Lawrence. 2015. A survey of online experiment design with the stochastic multi-armed bandit. *arXiv preprint arXiv:1510.00757* (2015).
[7] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*. 2249–2257.
[8] Pinar Donmez, Jaime Carbonell, and Jeff Schneider. 2010. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM, 826–837.
[9] Pinar Donmez, Jaime G Carbonell, and Paul N Bennett. 2007. Dual Strategy Active Learning. *Machine Learning ECML 2007* (2007), 116–127.
[10] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)* 46, 4 (2014), 44.
[11] Ole-Christoffer Granmo. 2010. Solving two-armed bernoulli bandit problems using a bayesian learning automaton. *International Journal of Intelligent Computing and Cybernetics* 3, 2 (2010), 207–234.
[12] Neha Gupta, Ole-Christoffer Granmo, and Ashok Agrawala. 2011. Thompson sampling for dynamic multi-armed bandits. In *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, Vol. 1. IEEE, 484–489.
[13] H. V. Hasselt. 2010. Double Q-Learning. In *Advances in Neural Information Processing Systems*. 2613–2621.
[14] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and JD Tygar. 2011. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*. ACM, 43–58.
[15] Leslie Pack Kaelbling. 1993. *Learning in embedded systems*. MIT press.
[16] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. 2012. Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*. Springer, 199–213.
[17] Ashiqur R. KhudaBukhsh, Jaime G. Carbonell, and Peter J. Jansen. 2016. Proactive-DIEL in Evolving Referral Networks. In *European Conference on Multi-Agent Systems*. Springer, 148–156.
[18] Ashiqur R KhudaBukhsh, Jaime G Carbonell, and Peter J Jansen. 2016. Proactive Skill Posting in Referral Networks. In *Australasian Joint Conference on Artificial Intelligence*. Springer, 585–596.
[19] Ashiqur R. KhudaBukhsh, Jaime G. Carbonell, and Peter J. Jansen. 2017. Incentive Compatible Proactive Skill Posting in Referral Networks. In *European Conference on Multi-Agent Systems*. Springer.

[20] Ashiqur R. KhudaBukhsh, Jaime G. Carbonell, and Peter J. Jansen. 2017. Robust Learning in Expert Networks: A Comparative Analysis. In *International Symposium on Methodologies for Intelligent Systems (ISMIS)*. Springer, 292–301.
[21] Ashiqur R KhudaBukhsh, Peter J Jansen, and Jaime G Carbonell. 2016. Distributed Learning in Expert Referral Networks. In *European Conference on Artificial Intelligence (ECAI)*. 1620–1621.
[22] Tze Leung Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6, 1 (1985), 4–22.
[23] Benedict C May. 2013. *Bayesian sampling in contextual-bandit problems with extensions to unknown normal-form games*. Ph.D. Dissertation. University of Bristol.
[24] Benedict C May, Nathan Korda, Anthony Lee, and David S Leslie. 2012. Optimistic Bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research* 13, Jun (2012), 2069–2106.
[25] Pannagadatta Shivaswamy and Thorsten Joachims. 2012. Multi-armed bandit problems with history. In *Artificial Intelligence and Statistics*. 1046–1054.
[26] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3/4 (1933), 285–294.
[27] Christopher JCH Watkins and Peter Dayan. 1992. Q-Learning. *Machine Learning* 8, 3-4 (1992), 279–292.
[28] Peter Whittle. 1988. Restless bandits: Activity allocation in a changing world. *Journal of applied probability* 25, A (1988), 287–298.