Distributed Learning in Referral Networks

Ashiqur Rahman KhudaBukhsh

CMU-CS-17-131
December 2017

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee
Jaime Carbonell (Chair)
Manuel Blum
Manuela Veloso
Victor Lesser (University of Massachusetts Amherst)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was sponsored by the Boeing under grant number CMUBAGTA1 and the National Science Foundation under grant numbers IIS-0855958, IIS-1065251, IIS-1216282, and IIS-1649225. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Active learning, Reinforcement learning, Referral network, Proactive skill posting

To the constant presence and encouragement of Chinmoy Chowdhury. To the inspiring memories of Satyabrata Das and Anjali Mitter.

Abstract

Human experts as autonomous agents in a referral network must decide whether to accept a task or refer to a more appropriate expert, and if so to whom. In order for the referral network to improve over time, the experts must learn to estimate the topical expertise of other experts. This thesis extends concepts from Multi-agent Reinforcement Learning and Active Learning to referral networks. Among a wide array of algorithms evaluated, Distributed Interval Estimation Learning (`DIEL`), based on Interval Estimation Learning, was found to be promising for learning appropriate referral choices, compared to `Greedy`, `Q-learning`, `Thompson Sampling` and Upper Confidence Bound (`UCB`) methods. `DIEL`'s rapid performance gain in the early phase of learning makes it a practically viable algorithm, including when multiple referral hops are allowed. In addition to a synthetic data set, we compare the performance of several top-performing referral algorithms on a referral network of high-performance Stochastic Local Search (SLS) solvers for the propositional satisfiability problem (SAT). Our experimental results demonstrate that the referral learning algorithms can learn appropriate referral choices in the real task of solving satisfiability problems where expertise does not obey any known parameterized distribution. Apart from evaluating overall network performance, we conduct a robustness analysis across the learning algorithms, with an emphasis on capacity constraints (limits on number of tasks per time period), evolving networks (changes in connectivity or agents joining or leaving the referral network) and expertise drift (skills improving over time or atrophying through disuse) — situations that often arise in real-world scenarios but are largely ignored in the Active Learning literature. Several high-performance referral learning algorithms proved to be robust to capacity constraints and evolving networks, while `Hybrid`, a novel combination of multiple algorithms, proved the most resilient to expertise drift. In an augmented learning setting, where experts may report their top skills to their colleagues, we proposed three algorithms, proactive-`DIEL`, proactive-`Q-Learning`, and proactive-ϵ-`Greedy`. All algorithms exhibited robustness to noisy self-skill estimates, evolving networks and strategic misreporting.

Acknowledgments

PhD is a long road, there were many who deeply influenced my thoughts and the way I look at things and made the journey worth traveling. First and foremost, I would like to thank Jaime Carbonell, my advisor. In my five years with him, I was fortunate to discuss with him a wide range of research problems and received a wealth of knowledge and wisdom on how to conduct meaningful research. As a side-benefit, our occasional discussions on Chess problems propelled my chess-rating from 1600 to 2000. I am thankful to Jaime for encouraging independent research and giving me the courage and support to handle failures, an integral part of research life. I am also grateful that he took me under his wings and walked me through the process of grant-writing, an invaluable skill that I acquired during my PhD stint.

I would like to express my sincere gratitude to my committee members, Manuel Blum, Manuela Veloso, and Victor Lesser, for their guidance and insightful feedback. I consider myself fortunate that I got to interact with a great mind like Manuel Blum, first as a student in his Graduate Algorithm class and later as a teaching assistant. I would forever cherish our discussions which were not confined to only computer science but involved our other common interest, poetry.

I co-authored all the published papers in this thesis with Peter Jansen, my research collaborator who improved both my scholastic skills and chess acumen! I am grateful to Roger Dannenberg for giving me exciting projects on Computer Music to work with. I would like to thank my collaborators from Microsoft Research and Yahoo! Research, Paul Bennett, Ryen White, Narayan Bhamidipati, Ravi Kant and Shaunak Mishra for giving me a great perspective on industry-research and teaching me how to set a tangible research goal with measurable progress. I am thankful to my masters thesis mentors, Kevin-Leyton Brown and Holger Hoos, for teaching me the ABC of paper-writing. The SATenstein solvers, which solved a major evaluation-challenge in my PhD thesis, was a joint-work with them and Lin Xu.

A special thanks to Chun Kai Ling and Andrew Hsi for helping me with the thesis defense talk, and Gloriana St. Clair for proofreading my thesis.

Life of an international student is challenging in many ways. OIE makes it one less challenge. When I joined CMU, a graduating student told us “Deb is an angel”. I cannot emphasize enough how spot on that statement was. She and Catherine Copetas together take wonderful care of the CSD PhD students, which I am going to miss after the PhD.

One of the biggest perks at CMU is the diverse and talented peer group we get time to spend with. Across numerous lunch, dinner (we do not cook that much), Table Tennis, Chess and Tennis sessions (we do waste a lot of time on other things though), Shayak Sen, Aram Ebtekar, Shriphani Palakodety, Sourav Chatterjee, Siddhartha Jain, David Kurokawa, Zack Coker, Ameya

Velingker, Salim Akhter Chowdhury, Aniruddha Basak, Alejandro Carbonara, Ankit Sharma, Anvesh Komuravelli and Mehdi Samadi made life at CMU a fun-filled one. I would like to thank my housemates, Shayak Sen, Kristina Sojakova, Aniruddha Basak, Sourabh Choudhury and Benjamin Smith for listening to my rants and raves and being supportive friends. I would also like to thank Imon, Arko, Sayantan, Sudipto, Angshul and Hirak, my friends from India who were always there when I needed them.

During my PhD, I pursued multiple interests and made many new friends. As a disciplined swimmer (I missed six days of swimming in my last three years), I met Martin Staniland, James Reilly, Justin Terreri, Phong Mai, Jim Seguin, Leon Chudnovsky, Jim Peele, Albert Pacella and James Sahovey and enjoyed many wonderful conversations with them. I am thankful to Martin Staniland, Alberta Sbragia and James Reilly for sharing their wisdom and life-lessons with me.

As a composer and musician, I was an active member of the Peace Band, a musical band comprising Indian graduate students at CMU. The rehearsal sessions with so many accomplished musicians (and fine friends) were great stress-busters, and I got to learn a lot from my fellow band members. I would like to thank Hafiz Karmali and Dipanjan Chatterjee for giving me the opportunity to be a part of New York theatre.

As a poet and columnist, I got tremendous support from my editors and publishers. I am thankful to Ekram Ali who taught me the fine art of capturing rich creative thoughts within the space constraints of a fortnightly column. My publisher, Rohon Kuddus, was always understanding about my paper deadlines; his calm presence helped me complete two collection of poems with him. I would like to thank my friends from the literary circle: Sahajiya Nath, Anuradha Biswas, Luna Rushdi, Payel Nandi and Pragnadipa Halder for inspiring me to write better.

Adjusting oneself to a setting miles away from his or her home can get tough. The occasional bouts of homesickness, a series of negative research results, an unexpectedly long homework - so many things can make life difficult at times. I am thankful that in my home-away-from-home, I have a loving family who was always there for me. Rajesh Bhattacharjee, Ayantika Ghosh, Anindit Mukherjee and Sohinee Bhattacharyya: thank you for so many wonderful road trips, movie nights and “adda”s. I consider myself lucky to have Gloriana St. Clair who always helped me with the right life advice (and great book and film recommendations). I am thankful to Emile Bominaar and Olga Ozhogina (I fondly call them chachajaan and kakimoni) for their parent-like affection.

Finally, I would like to thank my parents, my sisters and close family members for motivating and supporting me with my studies. I would refrain from spending too many words to describe their role in my life, simply because I will run the risk of overshooting the technical content of this thesis!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Whom do you ask when you don't know whom to ask?

Consider a network of experts with differing expertise, where any expert may receive a problem (aka a task or a query) and must decide whether to work on it or to refer the problem, and if so to which other expert. For instance, in a clinical network, a physician may diagnose and treat a patient or refer the patient to another physician whom she believes may have more appropriate knowledge, given the presenting symptoms. The referring physician may charge a referral fee and the receiving physician may charge a larger fee for diagnosing and treating the patient. Referral networks are common across other professions as well, such as members of large consultancy firms. If the experts are software agents, then the need for referral may be greater, given the likely narrower "expertise" typical of intelligent agents (including old-style expert systems). We can also envision a hybrid referral network comprising automated agents and possibly crowd-source human experts.

How does a network or how do individual experts in the network learn to refer effectively? Human referral networks are neither hardwired nor static. Potentially much larger networks of automated experts or hybrid networks with dynamic membership must likewise learn to refer with membership and expertise drift – and that learning should be distributed, without any "boss agent" telling all the others when to try and solve a problem or when to refer and if so to whom.

In this thesis, we explore and extend several well-known reinforcement learning algorithms to meet this *learning-to-refer* challenge. Rather than focusing on agents and problem-solving mechanisms, our primary focus is on this new Distributed Active Learning approach in referral networks. Crowdsourcing (Yuen et al., 2011) remains popular in labeling tasks (Cheng and Bernstein, 2015; Ahn et al., 2006), but also for more complex tasks requiring targeted skills (Bernstein et al., 2010; Yu, 2011; Yu and Nickerson, 2013; Heimerl et al., 2012). We see our work as a con-

fluence of these trends, where agents (e.g., experts, turkers, autonomous systems) have varying expertise, and targeting the right agent to the right job is key.

We see our work as a logical extension to Proactive Learning, that addressed several factors that arise in a real-world Active Learning setting. In many modern-day Machine Learning domains, the volume of labeled data is substantially smaller than the volume of unlabeled data, and in general, obtaining labeled samples is much harder than obtaining unlabeled samples. For instance, amassing a large pool of images by crawling a photo-sharing site like Flickr is much easier than collecting photos labeled with abstract concepts like 'war' or 'loneliness'. Since labeling comes with a cost, and usually all data-points are not equally informative, seeking labels for highly informative data points is often an efficient learning strategy. To this end, Active Learning, a Machine Learning paradigm where the learner takes an active role in the learning process by choosing which data points to label, is often found to be effective.

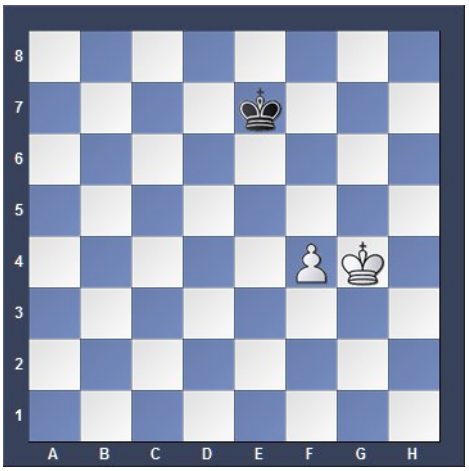
In Active Learning, the learning method is provided an initial seed set of a small number of labeled samples and a larger pool of unlabeled instances. The objective is to select optimal data instances to label such that the learning method when trained with the additional data, will improve its performance. While Active Learning (Lewis and Catlett, 1994; Lewis and Gale, 1994; Settles, 2010) continues to remain a useful tool for decades, several assumptions of Active Learning do not hold in practical scenarios. For instance, Active Learning assumes presence of just one, omniscient, indefatigable, infallible oracle whereas most modern day learning tasks involve multiple annotators with varying accuracy and expertise. Proactive Learning (Donmez and Carbonell, 2008; Donmez et al., 2010a) has relaxed these assumptions along several dimensions and proposed methods to deal with multiple, fallible annotators with time-varying accuracy.

In this thesis, we focus on an aspect not considered in Proactive Learning or any previous Active Learning literature: communication between experts in the form of referrals. With modern day machine learning applications reaching sophistication to a point that the labeling task often needs specialized expertise (is this musical piece Acid Jazz or Bebop? or, is the painting in Figure 1.1(a) Fauvist or impressionist[1]? or, with White to move, is the Chess position shown in Figure 1.1(b) a forced-win for White?), it is impractical to assume oracles would be omniscient, having equally strong skills/knowledge across different tasks/topics. Hence, communication between experts and the ability to redirect difficult instances to one another based on their topical expertise would be crucial for future applications. To address this gap in the Active Learning literature, we introduce referral network, a new Active Learning framework where experts (teachers or agents) can redirect difficult instances to their colleagues.

Whom do you refer when your top-choice abruptly departs the network, or gets unresponsive

[1]Image source: `https://en.wikipedia.org/wiki/Fauvism`

(a) Fauvist or Impressionist?



(b) With White to move, is this end-game a forced win for White?

Figure 1.1: Questions requiring specialized expertise

due to workload, or gradually loses her skills? Several practical factors can hinder or help the *learning-to-refer* challenge. For example, a sudden departure of a strong colleague can necessitate an immediate lookout for an alternative. Similarly, arrival of a strong expert for some topic areas in the network will only be useful when the colleagues are able to quickly identify her as one and hence refer appropriate tasks to her. The colleague can proactively share this information to expedite the search, but her own estimates of her skills could be noisy (she could strategically lie to attract more business as well!). An extremely skilled expert but who is too-busy-to-answer would also require fallback options. Also, experts who got dismissed as weak can improve their skill requiring the referral learning algorithms to have a balanced re-sampling and re-estimation approach. Many such considerations are often swept under the proverbial rug of Active Learning assumptions. In this thesis, we both consider these factors while assessing the performance of algorithms and design new algorithms to meet these challenges better.

While this thesis is primarily focused on the big picture, that is, how to learn referrals in a large distributed setting under various conditions, the techniques presented are fairly general and can be applied to a standalone multi-armed bandit setting (a gambler trying to optimize the total amount of reward she receives by pulling one of the k arms at a time, each arm has an unknown reward distribution). Specifically, our work on expertise drift and proactive skill posting addressing the cold-start problem (KhudaBukhsh et al., 2016a,b, 2017a) could be of interest to the larger bandit community. Also, referral learning could be combined with other forms of learning, a case we have not addressed in this thesis. For instance, the problem-solving skills of an expert can improve over time through learning from the solutions received through referrals.

In addition to learning how to refer better, such other forms of learning would also improve the overall performance of the network. We have considered drift-scenarios where experts largely improve over time (see, Chapter 4.3); however, no other form of learning of experts is explicitly modeled (i.e., with specified learning rate parameters) in this thesis.

In what follows, we first illustrate our referral mechanism with a toy example of a five-expert referral network in which we demonstrate how effective referrals can dramatically improve the network performance. Next, we outline the three major research directions we pursued in this thesis and highlight our contributions. We end this chapter with a road-map to the rest of the thesis.

1.1 Referral Mechanism

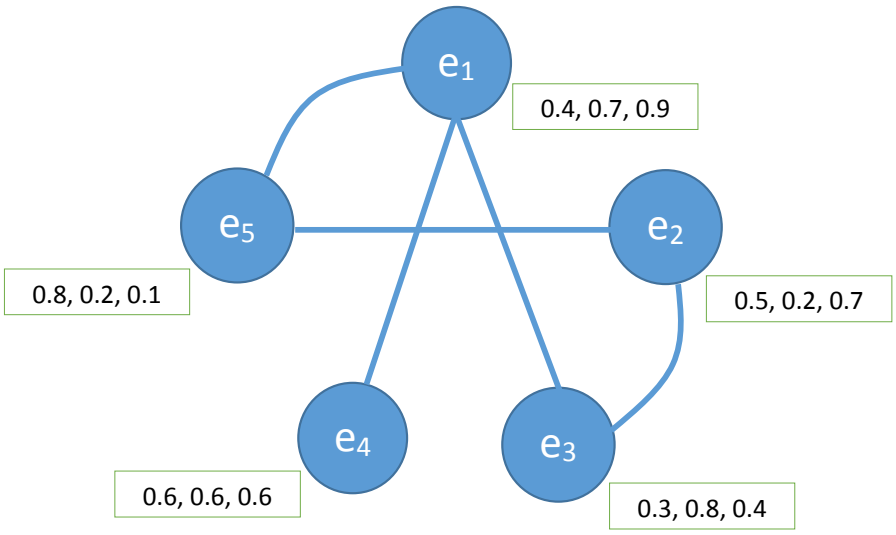


Figure 1.2: A referral network with five experts

Our referral model assumes an initial sparse topology of a referral graph where each expert knows a handful of colleagues so that $E \sim O(V)$ (E and V denote the number of edges and vertexes in the network, respectively). Learning consists of each expert improving its estimates of the ability of colleagues to solve different classes of problems. These colleague experts may be in the initial network, or added to the network over time as the network topology evolves.

We illustrate the referral mechanism and its effectiveness with the simple graph of Figure 1.2 which represents a network of five experts. The nodes of the graph are the experts, and the edges indicate that the experts 'know' each other, that is, they can send or receive referrals and

communicate results. In the domain, three different topics (subdomains) can be distinguished – call them t_1, t_2, and t_3 – and the figures in brackets indicate an expert's expertise in each of these.

In this referral network, with a query belonging to t_2, if there was no referral, the client may consult first e_2 and then possibly e_5, leading to a probability of getting the correct answer of $0.2+(1-0.2)\times 0.2 = 0.36$. With referrals, an expert handles a problem she knows how to answer, and otherwise if she had knowledge of all the other experts' expertise she could ask e_2 who would refer to e_3 for the best skill in t_2, leading to a solution probability of $0.2 + (1 - 0.2) \times 0.8 = 0.84$.

For a query budget Q of 2, the steps in our learning setting are the following.

1. A user issues q_j (*initial query*) to a randomly chosen expert e_i (*initial expert*)
2. Initial expert e_i examines the instance and solves it if possible. This depends on the *expertise* of e_i wrt. q_j.
3. If not, a *referral query* is issued by e_i to a *referred expert*, e_j, within her subnetwork. *Learning-to-refer* involves 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.

The first two steps in our referral network are identical to Active Learning. Step 3 and 4 are the extension to the Active Learning setting proposed in this work. Understandably, with a higher per-instance query budget, step 4 can loop back to step 2 and the referred expert can re-refer instances to other experts as long as budget permits.

In the above example, even with referrals, e_2 could redirect the instance to e_5 when she does not know e_3 is the strongest colleague in that topic, leading to no improvement in the solution probability. Hence, for effective referrals, close-to-accurate estimations of colleagues' topical expertise is crucial. Obtaining such estimations is the *learning-to-refer* challenge we are addressing in this thesis. Essentially, *learning-to-refer* involves querying one colleague at a time balancing exploration (search for stronger experts) and exploitation (referring more problems to the strongest expert found so far) and improving the estimates with newer observations. In the most general setting, the starting point could be completely uninformative prior where an expert has no information about the strengths and weaknesses of her colleagues. We evaluated the performance of a wide array of reinforcement learning algorithms existing in the literature in this setting. However, in real world, colleagues often share information about their strengths. We model this assumption in an augmented learning setting dubbed proactive skill posting. In both cases, the learning is distributed, i.e., each expert learns to improve its referral strategies from its unique vantage point in the referral network, greatly increasing the need for referral algorithms

that learn fast. In the following section, we present a structured outline of our three primary research directions.

1.2 Research Directions

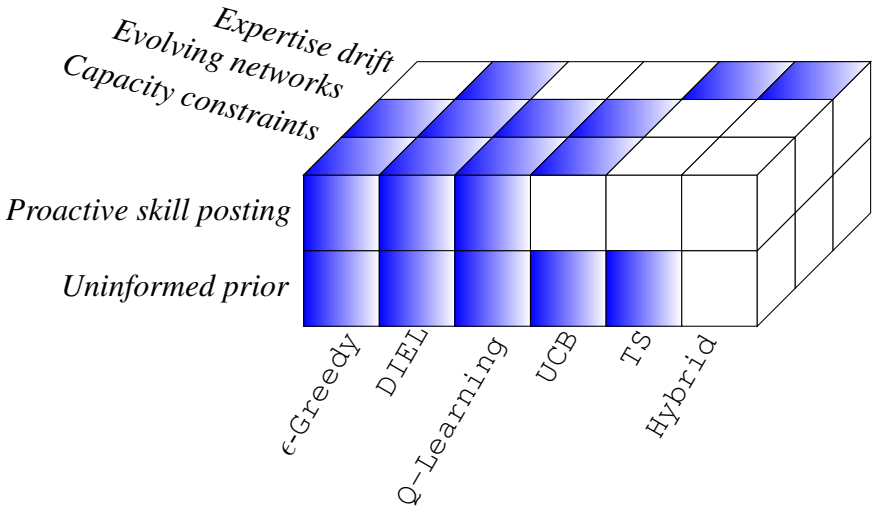


Figure 1.3: Research directions in learning-to-refer: blue denotes successful research results

As shown in Figure 1.3, the primary research thrusts on referral networks span three largely orthogonal dimensions: use of prior knowledge, learning-to-refer algorithms and robustness to a dynamically evolving environment. A blue surface indicates availability of positive experimental results involving the labels on the corresponding axes. Along the algorithm axis, we show a subset of representative algorithms to illustrate our contributions (for a complete list of algorithms, see Table 3.4).

In terms of availability of priors, we address learning-to-refer in two primary conditions: a) uninformative priors, which presents a cold-start problem of forming initial reliable estimates of colleagues' topical expertise, and b) informative priors where agents proactively advertise their stronger skills to their colleagues. Our first line of exploratory study focuses on assessing the viability of our proposed learning setting through comparative evaluation of referral learning algorithms, while our second direction is closer to real-world setting as in practice, experts often do not start from completely uninformative prior. Moreover, in real life, we often see that experts clearly mention which type of tasks they are particularly good at and also often forge links to their colleagues via social networks. In turn, their colleagues may re-estimate their beliefs of expertise levels based on actual performance. In our augmented learning setting, dubbed proactive skill

posting, a one-time local-network advertisement of a subset of skills is allowed. In addition to designing algorithms that take advantage of such advertised priors, we focus on resilience to dynamic addition/drop off of experts in the network, noisy self-skill estimation and strategic lying.

The algorithmic investigations primarily focus on:

1. evaluating algorithms with known finite-time regret bounds (e.g., Upper Confidence Bound (`UCB`) variants, `Thompson Sampling` variants) or strong empirical performance (e.g., Interval Estimation Learning), developing novel algorithms or new variants (e.g., `Pessimistic TS-DIEL`), and hybrid algorithms, and testing them both on synthetic and non-synthetic data.
2. designing proactive-posting versions of the algorithms to take advantage of advertised priors, and evaluating these both on synthetic data and non-synthetic data.
3. empirical analysis of tolerance to noisy-skill estimates and strategic lying and theoretical analysis of incentive compatibility in the limit.

Along the robustness axis, we focus on:

1. **Capacity constraints**: The theoretical assumption that an oracle can handle an unbounded number of requests within a specific time period is impractical. In our work, we relax this assumption and consider experts can only handle a limited number of tasks within a specific time period beyond which it becomes *over-loaded* and refuses to solve any further task until load situation improves. We evaluate our referral-learning algorithms under this assumption varying the parameter that sets the limit to task-threshold.
2. **Evolving networks**: In practice, the composition of network may change with time through joining/leaving of new/old experts and forging/breaking of new/existing links between experts. We considered evolving networks both in our uninformed prior setting and proactive skill posting setting. In uninformed prior setting, the challenge is to quickly find alternative experts when a strong expert leaves and rapid identification of strength areas of newly joined experts. In proactive skill posting, the challenge is to elicit truthful prior advertisements to expedite the integration of the newly joined experts.
3. **Expertise drift**: Experts can improve over time through practice, acquisition of new skill or degrade due to fatigue, age and disuse of skills etc. Tracking expertise drift is crucial for a referral network's overall performance as experts dismissed early for being weak can end up being strong contenders for future referrals. Inability to identify such late bloomers or sticking long with experts who were once highly-skilled but gradually lost their sheen could negatively affect network performance.

Not forming a major research direction in the thesis, but from an evaluation perspective, one important point we had to address was to identify real-world data set on which we can test the performance of referral learning algorithms in the wild. For this, we would need multiple experts with varying topical expertise on several topics. We took a novel approach in our non-synthetic data set design; we constructed several referral networks of high-performance solvers for the propositional satisfiability problem (SAT) where solvers map to experts, topic map to SAT distributions (e.g., quasi-group completion problem, bounded model checking, factoring etc.) and the task is the real task of solving a SAT instance. While these solvers already existed (KhudaBukhsh et al., 2009, 2016d), the use of these solvers in the context of referral learning or even the more general multi-armed bandit setting was never done before.

1.2.1 Contributions

Our three main contributions in this thesis are the following:

1. We propose referral network, a novel Active Learning setting where experts (agents or teachers) can redirect difficult instances to one another. Through a series of experiments on synthetic and non-synthetic data, we establish the viability of referral networks even with uninformative priors as several algorithms (most prominently, `Distributed Interval Estimation Learning`, `DIEL`) proved to meet the distributed *learning-to-refer* challenge including when multi-hop referrals are allowed.
2. We propose proactive skill posting, an augmented learning setting where experts are allowed one-time local-network advertisement of a subset of their skills. We design proactive versions, a class of modified referral algorithms that takes advantage of such advertised priors in a manner resilient to strategic lying and noisy self-skill estimates.
3. We perform a thorough robustness analysis of these algorithms considering factors like capacity constraints (limits on number of tasks per time period), evolving networks (changes in connectivity or agents joining or leaving the referral network) and expertise drift (skills improving over time or atrophying through disuse). For expertise drift, we propose `Hybrid` a novel combination of `Optimistic Thompson Sampling`, `Pessimistic Thompson Sampling` and `DIEL` which were most resilient against drift. For evolving networks, we proposed proactive-`DIEL` which could withstand a large amount of network change at regular interval without much loss of performance.

1.3 Thesis Organization

The rest of the thesis is organized in the following way. We present the literature relevant to our contributions in Chapter 2. Our three contributions are presented in the following three chapters. In Chapter 3, we focus on the uninformative prior setting and present our initial set of assumptions related to expertise, network and reward, and our data sets (synthetic and non-synthetic), describe a wide pool of referral-learning algorithms we considered and analyze their performance both on single-hop and multi-hop referral settings. In Chapter 4, we assess the robustness of our referral-learning algorithms under several robustness criterion, mainly focusing at capacity constraints, evolving networks and expertise drift with expertise drift requiring novel algorithm design. In Chapter 5, we present our augmented learning setting, proactive skill posting where experts are allowed a one-time local network advertisement of a subset of their skills to their colleagues. In this chapter, the key challenges are robustness to noisy self-skill estimates and tolerance to strategic lying. We also revisited the problem of evolving networks and evaluated the performance of algorithms when skill posting is allowed. Finally, in Chapter 6, we summarize our contributions and outline future research directions.

Chapter 2

Related Work

In this chapter, we present a literature review of prior work relevant to our contributions. We break down the chapter in the following way. First, we present prior work relevant to the referral framework, and DIEL, the top-performing algorithm in our uninformative prior setting. Next, we discuss prior research related to our three major stress areas in robustness analysis: capacity constraints, evolving networks and expertise drift, after which, we compare and contrast with literature relevant to our augmented learning setting, proactive skill posting. We finally conclude with pointing to literature related to some additional aspects and our real-world data set.

2.1 Referral Framework

The inspiration for the referral framework dates back to referral chaining, first proposed in (Kautz et al., 1996) and subsequently extended in (Foner, 1997; Yolum and Singh, 2003; Yu, 2002; Yu et al., 2003; Yu and Singh, 2003). In particular, Yu's dissertation (Yu, 2002) made an extensive study of the properties of a simulated referral network under various conditions several of which are consistent with the simulated networks in this thesis. However, Yu's work made a demarcation between expertise and referring capabilities: agent's expertise (her own ability to solve problems) and sociability (her own ability to refer) and consider both co-operative and non-co-operative agents in the networks who are identified using a reputation based system. In contrast, in our work, we make no distinction between sociability and expertise in multi-hop settings and experts who are connected through referral links are assumed to be collaborative (in proactive skill posting, experts may misreport their skills). Moreover, Yu's work did not address learning to refer, capacity constraints, or dynamic adaptation. Beyond synthetic data, we evaluated the performance of several referral learning algorithms on a SAT solver data set, where neither expertise nor noise in estimating self-skill obeys any known parameterized distribution. Also, we

moved beyond the initial referral framework through augmenting the learning setting with a one time local-network advertisement of a subset of skills and proposed modified algorithms that can take advantage of such partially available noisy priors.

Among the referral learning algorithms we considered, Distributed Interval Estimation Learning (DIEL) performed the best in the static expertise scenario and a modified version, proactive-DIEL, proved to be the strongest in the proactive skill posting setting. DIEL built upon earlier research on interval estimation learning (IEL). IEL, a reinforcement learning technique that strikes a balance between exploration and exploitation by combining mean and variance of the observed rewards (as opposed to a strategy that switches between the two, such as in (Osugi and Scott, 2005).), was first proposed in (Kaelbling, 1993; Kaelbling et al., 1996). In the context of Active Learning, IEL has been successfully used in jointly learning the accuracy of labeling sources and obtaining the most informative labels in (Donmez et al., 2009a). IEL is used to compute expected rewards of individual oracles and then a multiplicative threshold on the best oracle's performance is used to eliminate inferior oracles. We used the version from (Donmez et al., 2009a) as the basis of our DIEL method, adjusting for key differences such as the fact that learning in referral networks is *distributed*, that is, each expert learns to improve its labeling strategy from its unique vantage point in the referral network, greatly increasing computational challenges of our simulation. Unlike (Donmez et al., 2009a), in this work we consider heterogeneous tasks and the primary challenge is to estimate topical expertise. While designing proactive-DIEL, an algorithm that takes advantage of proactive skill posting, we found that dropping the student-t distribution parameter improved DIEL's performance in the early learning phase (KhudaBukhsh et al., 2016a). We used the modified version for all of the subsequent experiments.

The problem of learning appropriate referrals can be cast in various ways. One direction we considered was multi-armed bandit selection 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. The three robustness criterion we focused on: capacity constraints, evolving networks and expertise drift, map to the *sleeping bandits* (Kleinberg et al., 2010), *mortal bandits* (Chakrabarti et al., 2009) and *restless bandits* (Whittle, 1988) in the literature, respectively. Accordingly, we enlisted several multi-armed bandit algorithms (Agrawal, 1995; Lai and Robbins, 1985; Audibert et al., 2007) for performance comparison . However, 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 has been used in the context of referral learning before nor in similar distributed network learning. Similarly, we also included Q-Learning variants (van Hasselt, 2010; Watkins and Dayan, 1992) and Thompson Sampling variants (Thompson, 1933; May et al., 2012) for compara-

tive analysis which haven't been explored in this context before. A detailed description of our referral algorithms is presented in section 3.4 along with additional relevant literature.

An analysis of referral networks also exhibits similarities with the study of task allocation (Abdallah and Lesser, 2006; Zhang et al., 2009; Zhang and Lesser, 2007), where minimizing turn-around time corresponds to the maximizing the probability of a correct answer. Studies were made of network changes and load balancing for distributed reinforcement learning (Zhang et al., 2010), and of topic hierarchies (in the context of information retrieval). We focus on using a referral network as a substrate for Active Learning algorithms, which has not been previously studied. Also, the `FAL` algorithm described in (Zhang et al., 2009) uses a variant of ϵ-`greedy Q-learning` similar to one we compared favorably against in the current work.

There exists a wide body of literature relaxing several assumptions of classical Active Learning geared towards practical considerations. Instead of requesting one label at a time, batch mode learning is considered in (Brinker, 2003; Xu et al., 2007; Guo and Schuurmans, 2008; Hoi et al., 2006a,b; Yang and Carbonell, 2013). Similar to our setting with experts with varying skill levels, noisy oracles have been studied in (Snow et al., 2008; Donmez et al., 2009b, 2010b; Sheng and Ling, 2006). In our work, we assume labeling cost as a constant. However, in real-world, it could be variable depending on factors like difficulty level of tasks, scarcity of relevant experts, required time or resource to solve etc. Cost-sensitive Active Learning has received attention in (Kapoor et al., 2007; Vijayanarasimhan and Grauman, 2011; King et al., 2004). Similar to our work where queries belong to multiple topics, multi-task learning is considered in (Reichart et al., 2008; Qi et al., 2008). However, none of these previous works considered communication between multiple noisy oracles in the form of referrals.

In crowdsourcing, communication between experts and non-experts has been studied – in a system dubbed Skierarchy (Nallapati et al., 2012) domain experts break down a complex task into simpler micro-tasks and actively supervise the non-expert crowd. However, in our approach, there is no such simplifying assumption: it is highly unlikely that one expert Pareto dominates other experts in a professional network across all topics. Instead, referral is bi-directional, and the main focus is on learning appropriate referral choices in a distributed manner, rather than by specific boss agents. Although not connected through referral networks, multiple annotators are often used to disambiguate noisy labels. In presence of annotator disagreements, *Learning from crowd*, proposed in (Raykar et al., 2010; Whitehill et al., 2009), presents a probabilistic model to model the labeling process and a subsequent Expectation-Maximum (EM) step is used to obtain maximum likelihood estimates (MLEs) of unobserved variables. The trade-off between expert labelers and noisy labelers, a related problem, has been studied in (Sheng et al., 2008; Snow et al., 2008; Sorokin and Forsyth, 2008). In the MAB literature, communications between the

players to find a close-to-optimal arm has been considered in (Hillel et al., 2013). In this setting, k players communicate among themselves in a load-balanced manner to find an ϵ-optimal arm and the primary challenge is to find a close-to-optimal arm in a large set of arms through distributed search. In our work, the challenge is different: modeling real world, individual experts are connected only with a handful of colleagues; however, learning being distributed, each expert has to learn her own referral policies for individual topics for which early learning advantage is crucial.

2.2 Robustness Analysis

Our work on expertise drift fits in the broader context of multi-agent learning in non-stationary setting (Silva et al., 2006; Noda, 2009; Kaisers and Tuyls, 2010; Abdallah and Kaisers, 2016; Bowling and Veloso, 2001). In the context of Proactive Learning, prior work on Interval Estimation Learning (the basic building block of `DIEL`) to track time-varying accuracy (Donmez et al., 2010b) 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. In the MAB literature, time-varying reward distributions were introduced in (Whittle, 1988) and subsequently had several contributions addressing the challenge (see, e.g., (Weber and Weiss, 1990; Bertsimas and Niño-Mora, 2000; Liu and Zhao, 2010; Yu and Mannor, 2009; Hartland et al., 2006)). `Dynamic Thompson Sampling` (Gupta et al., 2011), an extension of `Thompson Sampling` (Thompson, 1933), were suggested for these *restless bandits*. Our work is different from previous *restless bandits* literature by introducing richer algorithms and operating in scale, with multiple agents learning several threads of referral policies for each topic. Brownian perturbation (Gupta et al., 2011) for modeling random drift is insufficient for capturing human expertise change, as it often improves with time and hence requires considering positively biased drifts. We also present a less common approach in tackling drift including concept drift (Tsymbal, 2004; Gama et al., 2014) where the most popular approaches are window-based (Gupta et al., 2011; Garivier and Moulines, 2008). A related problem is that of fault detection-isolation Lai (2001). However, the goals are different; as opposed to detecting the change and classifying the post-change distribution within a finite set of possibilities, we are primarily concerned with addressing the drift by incurring minimum possible regret. Our setting is also more complex with several possible change points; a similar problem is addressed in Akakpo (2008).

For expertise drift, we proposed a hybrid combination of `Optimistic Thompson Sampling`,

`Pessimistic Thompson Sampling` and `DIEL` which adaptively switches between different algorithms striking a balance between exploration and exploitation. Several lines of work in the past have studied adaptive strategies in algorithm design. In the Active Learning literature, one such example is (Donmez et al., 2007) where performance improvement over static strategies is brought by adaptively updating strategy selection parameters. In a completely different domain, (Wei et al., 2008) presents a hybrid stochastic local search SAT solver that switches between two heuristics to strike a balance between search diversification and intensification.

Capacity constraints have been largely ignored in Active Learning literature other than Proactive Learning (Donmez and Carbonell, 2008) where unavailability of oracles is considered in the form of oracle reluctance which is subsequently used to estimate the reluctance for labeling nearby points. In the MAB literature, *Sleeping bandits* has been considered in (Kleinberg et al., 2010; Blum and Mansour, 2007; Freund et al., 1997). Apart from the contrast in the scale we are operating, our work is also different in a sense that capacity constraints are influenced by expertise; highly-skilled experts are more likely to get busier hence unavailable for future requests. We see some direct applications in load-balancing referral network through dispersion games, specifically anti-coordination games (Grenager et al., 2002). However, the two research questions are different:

Learning-to-refer: given a network of experts, how to learn effective referrals?

Anti-coordination game: given a set of experts, how to create a referral network by connecting experts with minimal skill overlap such that the overall network performance is improved?

Similar to capacity constraints, evolving networks also received little attention from previous work in Active Learning. *Mortal bandits*, MAB equivalent of evolving networks, has been studied in (Chakrabarti et al., 2009). In (Chakrabarti et al., 2009), two different approaches to model mortality were considered of which the *timed death* corresponds to our setting. Our work is different in seeing a connection between the cold-start problem and evolving networks and proposing proactive skill posting, an augmented learning setting, and proactive versions of algorithms that tackle evolving networks with greater agility.

2.3 Proactive Skill Posting

Cold-start problem, the primary challenge proactive skill posting aims to address, is a well-studied problem in recommender systems (the *new user* problem) (Park et al., 2006; Leung et al., 2008; Maneeroj and Takasu, 2009; Chen and He, 2009; Loh et al., 2009; Weng et al., 2008; Kim et al., 2010). Whenever a new user joins the system, it is difficult to come up with meaningful recommendations since their is little prior information (e.g., movies rated by the user) about the

user. Analogous to that, when a new expert joins a referral network, she does not know whom to refer and her colleagues are equally uninformed about her expertise areas. Proactive skill posting is also related to the bandit literature with *side-information* (Langford et al., 2008; Shivaswamy and Joachims, 2012) in the sense that algorithms do not start from scratch. However, a key difference is that, instead of from observed trials (Shivaswamy and Joachims, 2012) or shape of the reward distribution (Bouneffouf and Feraud, 2016), the *side-information* in our case is obtained through advertisement of skills by the experts themselves (who may in fact willfully misreport to attract more business). This ties our work broadly to the vast literature in adversarial machine learning (Huang et al., 2011; Papernot et al., 2016; Newsome et al., 2006) and truthful mechanism design (Babaioff et al., 2009; Biswas et al., 2015; Tran-Thanh et al., 2012b,a). Among a large body of literature in truthful mechanism design (Babaioff et al., 2009; Biswas et al., 2015; Tran-Thanh et al., 2012b,a) we highlight a few key differences with the *budgeted multi-armed bandit mechanism* motivated by crowdsourcing platforms presented in (Biswas et al., 2015). First, our setting is distributed; while *learning-to-refer* can be interpreted as a multi-armed bandit problem where each arm is a referral choice, we are in fact dealing with several such parallel multi-armed bandit problems. Also, in our setting, experts have varying topical expertise which increases the scale of the problem, as each expert needs to estimate the expertise of her colleagues for each of the topics. In contrast, (Biswas et al., 2015) considered homogenous tasks. Reflecting real-world scenarios where experts may not know their skills on all topics and also may not have the time budget to inform their colleagues about their skills on individual topics, proactive-`DIEL` deals with partially available priors, i.e., experts are restricted by an advertisement budget and do not bid for all the topics (a factor (Biswas et al., 2015) did not need to consider because of homogeneous tasks). Finally, much of our focus is on a thorough empirical performance evaluation on both synthetic data and real-world data where certain distributional assumptions on expertise and skill estimates may or may nor hold.

2.4 Other Related Work

In part of this work, we used `SATenstein` (KhudaBukhsh et al., 2009), a highly parameterized Stochastic Local Search (SLS) SAT solver. `SATenstein` has a design space of 2.01×10^{14} candidate solvers which includes most of the high-performance SLS SAT solvers proposed in the literature. By using an automatic algorithm configurator, `SATenstein` can be configured on specific SAT distributions. We used 100 such solvers obtained from the experiments in (KhudaBukhsh et al., 2016d) that allow us to evaluate referral performance on a real task of SAT solving. The data set can be utilized as meaningful benchmarks for evaluating MAB algorithms

in general where there is a serious lack of empirical evaluations beyond synthetic data set barring few (Chapelle and Li, 2011; Kandasamy et al., 2017)

Our work was also influenced by several other areas, such as Agents and Simulation, (e.g., (Axelrod, 2003)), Networks and Emerging Properties,(e.g., (Manavalan and Singh, 2012; Yu, 2002)), Data Mining in Social Networks, (e.g., (Jensen and Neville, 2002)), Expertise and Expertise Finding, (e.g., (McDonald and Ackerman, 2000; Pushpa et al., 2010; Lin et al., 2017)), Computational Trust (e.g., (Sabater and Sierra, 2005; Sherchan et al., 2013)) of which a small selection is presented in the Bibliography.

Chapter 3

Referral Setting with Uninformative Priors

In this chapter, we focus on the uninformative prior setting, the most challenging setting in which experts start with no prior information on the expertise of their colleagues. Our primary research goal here is to identify a set of referral learning algorithms that can address the *learning-to-refer* challenge and evaluate them both on synthetic and non-synthetic data.

We first present the three research questions we are interested in finding answers to and explain why these questions are important and their associated challenges. Next, we present the preliminaries to understand referral mechanism, basic notations, and our assumptions that guided our data set generation process. In subsequent chapters, we relaxed several of our initial assumptions that led to re-evaluation (e.g., algorithms' handling capacity constraints) or re-designing of algorithms (e.g., proactive algorithms to handle evolving networks) and at times, both new algorithms and data set for performance evaluation (e.g., data set and algorithms for expertise drift). Whenever, we present an assumption that is relaxed in a later stage, we provide pointer to the relevant section where it is relaxed. However, instead of designing novel algorithms, in this chapter, we mainly compared several existing re-inforcement learning algorithms to assess their suitability in learning referrals (exceptions include a variant of ϵ-`Greedy` and ϵ-`DIEL`). We provide a short description of these referral-learning algorithms along with relevant literature. Following the algorithm description, we present our experimental setup and analyze the experimental results both on synthetic and non-synthetic data. We conclude with revisiting our research questions and presenting our main takeaways from this chapter.

3.1 Research Questions and Challenges

In this chapter, we focus on the following research questions:

How to learn effective referral choices? A key challenge in learning in a distributed setting

without a central "boss" agent is local visibility of rewards, i.e., each expert has to learn its own referral choices for each topic and her observations are not visible to others so there is no scope of learning from others' mistakes or benefiting from others' findings. For practical viability, it is crucial that the learning algorithm shows rapid improvement in the early phase of learning.

Do close-to-optimal local decisions translate into close-to-optimal global decisions? We are primarily interested in the overall performance of the network. In a distributed setting, each expert learns her own referral policies, and depending on her colleagues' expertise and size of the subnetwork would require varying number of samples to approach optimal policies. How do several experts learning in parallel affect the overall network performance, and if close-to-optimal decisions translate into close-to-optima global decisions are important questions to consider.

How should we evaluate performance beyond synthetic data? Finding data sets suitable for the referral setting is a nontrivial challenge. For this, we would require a large number of experts with differential domain expertise. The flexibility to consider both binary and bounded continuous reward will be a plus.

3.2 Preliminaries

Referral network: Represented by a graph (V, E) of size k in which each vertex v_i corresponds to an expert e_i $(1 \leq k)$ and each bidirectional edge $\langle v_i, v_j \rangle$ indicates a *referral link* which implies e_i and e_j can refer problem instances to each other.

Subnetwork: The *subnetwork* of an expert e_i is the set of experts linked to e_i by a referral link.

Scenario: Set of m instances $(q_1, \ldots, q_m)$ belonging to n topics $(t_1, \ldots, t_n)$ that are to be addressed by the k experts $(e_1, \ldots, e_k)$.

Expertise: Expertise of an expert/question pair $\langle e_i, q_j \rangle$ is the probability with with e_i can solve q_j.

Referral mechanism: For a query budget Q = 2, consists of the following steps.

1. A user issues q_j (*initial query*) to a randomly chosen expert e_i (*initial expert*)
2. Initial expert e_i examines the instance and solves it if possible. This depends on the *expertise* of e_i wrt. q_j.
3. If not, a *referral query* is issued by e_i to a *referred expert*, e_j, within her subnetwork. *Learning-to-refer* involves 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 query budget >2, the recipient of a referral can herself re-refer to another expert.

3.3 Expertise and Network Assumptions

We now present our assumptions on the network, expertise and rewards (observability, range etc.). Our assumptions are primarily geared towards constructing our synthetic data set. Since many of these assumptions may not hold in the wild, in addition to our experiments on synthetic data, we test our referral algorithms' performance on a data set of SAT solvers (see, Chapter 3.7) and also used standard random graph generators (see, Chapter 4.4).

3.3.1 Expertise and Expert Assumptions

Topic-wise distributional assumption: We take the expertise distribution for a given topic *t* to be a mixture of two truncated 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.

Instance-wise distributional assumption: We model the expertise of a given expert on instances under a topic by a truncated Gaussian distribution with small variance. i.e.,

$\textit{Expertise}(e_i, q_j) \sim \mathcal{N}(\mu_{topic_p,e_i}, \sigma_{topic_p,e_i})$,

$\forall q_j \in topic_p, \forall p, i : \sigma_{topic_p,e_i} \leq 0.2$.

The distributional assumptions on expertise are relaxed in Chapter 3.7. In our experiments with well-known high-performance SAT solvers as experts, and finding a satisfiable model for a SAT instance is the task, our expertise assumptions no longer hold.

In our experiments with SAT solvers as experts, verification of a solution is rather straightforward; one can easily verify if a model satisfies a propositional satisfiability instance. However, in several real-world problem domains, verifiability of solutions could be difficult, and in those cases, we consider the *initial expert* takes on faith that if the *referred expert* solved the problem, she probably solved it correctly. For instance, if a general practitioner referred a patient to a cardiologist and the latter diagnosed a heart valve malfunction leading to corrective surgery, there is no way for that general practitioner to know for sure if that was the optimal diagnosis, even if the patient was cured (e.g., a less invasive process resulting from a slightly different diagnosis such as a fluttering valve might have been "optimal"). Judging answer optimality is an interesting research challenge, but that is not the focus of this thesis. Consensus opinion by

multiple experts would apply as a surrogate to ground truth if budget allowed, but in most real settings budget does not allow. Hence, in all our experiments, we make a simplifying assumption that an expert "knows" when she is able/unable to solve a task and communicates the solution or the failure truthfully. We additionally assume that an expert's referral decision to a colleague is independent of the referral behavior of the colleague, i.e., there is no *quid pro quo* or any other form of side-deals influencing the referral decision.

3.3.2 Network Assumptions

The probability that a referral link exists between expert e_i and e_j is a function of how similar the two experts are, which we modeled as
$P(\textit{ReferralLink}(v_i, v_j)) = \tau + c\, Sim(e_i, e_j)$. We made this modeling choice because of the general observation that people sharing common expertise areas are more likely to know each other. For instance, two Machine Learning researchers are more likely to know each other as opposed to a Data Scientist and a Cyber-security expert. Within the Machine Learning community, two researchers working on Active Learning will be even more likely to know each other than an Active Learning researcher and a Reinforcement Learning researcher. As a similarity metric we used *cosine similarity of topic-means*. The parameter τ captures any extraneous reason two experts can be connected, e.g., same geolocation, common acquaintances, etc. Our network assumptions are relaxed in Chapter 4.4 in which we consider networks constructed using well-known random graph generators.

3.3.3 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* where each arm corresponds to a referral choice. 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 (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 involving synthetic data, we considered binary rewards, with a failed and successful task receiving a reward of 0 and 1, respectively. For our experiments with SAT solvers,

we considered both binary and continuous reward. The policy to assign binary reward is straight-forward: a solved instance receives a reward of 1 and a timed-out instance receives a reward of 0. Continuous reward is modeled by time-to-solution.

- **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. In case of multi-hop referrals, suppose an instance is first received by expert A who redirects it to B, B redirects it to C who eventually solves it. C will inform B the solution who in turn will inform the solution to A. So A will learn B solved the instance and will assign a reward 1 to B.

We further assume that an expert can accurately identify the topic of a query (relaxed in Chapter 4.4), the distributional parameters of expertise do not change over time (relaxed in Chapter 3.7), and that experts have no capacity constraints (relaxed in Chapter 4.1), and experts do not have any bias to specific colleagues.

3.4 Referral-Learning Algorithms

As we already mentioned, from the point of view of a single expert, learning appropriate referral choices for a given topic is an action selection problem (*multi-armed bandit* problems belong to this broader class of action selection problems). We first fix topic to T and expert to e and describe each algorithm in that context. 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, $N = \sum_{i=1}^{K} n_i$. $m(e_i)$ and $s(e_i)$ denote the sample mean and sample standard deviation of these reward vectors. For all our experiments involving synthetic data, we consider binary reward. For example, if an expert solved the first two referred queries and fails in the next two, the reward vector will look like (1, 1, 0, 0). Some of the algorithms we would discuss require initializing these reward vectors; we will explicitly mention whenever such initialization is required. For each expert 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, $\forall(S_{e_i} + F_{e_i}) > 0$, $m(e_i) = \frac{S_{e_i}}{S_{e_i}+F_{e_i}}$.

The learning to refer challenge is the following question: given a new referred query, q,

Category	Algorithm	Parameters
IEL	`DIEL` (KhudaBukhsh et al., 2016a)	None
Greedy	`DMT` (KhudaBukhsh et al., 2016c)	None
Greedy	ϵ`-Greedy` (Auer et al., 2002)	c
Greedy	ϵ`-Greedy1` (KhudaBukhsh et al., 2017b)	α
UCB	`UCB1` (Agrawal, 1995)	None
UCB	`UCB2` (Auer et al., 2002)	α
UCB	`UCBNormal` (Lai and Robbins, 1985)	None
UCB	`UCBV` (Audibert et al., 2007)	θ
Q-learning	`Q-learning` (Watkins and Dayan, 1992)	α, γ, ϵ
Q-learning	`DQ-learning` (van Hasselt, 2010)	α, γ, ϵ
Thompson Sampling	`Thompson Sampling` (Thompson, 1933)	None
Thompson Sampling	`Optimistic Thompson Sampling` (May et al., 2012)	None

Table 3.1: Referral algorithms

with past referred queries $q_1, \ldots, q_N$ and reward vectors $\mathbf{r}_{1,n_1}, \ldots, \mathbf{r}_{K,n_K}$, which expert should e refer to? The referral learning algorithms we used for performance comparison are listed in Table 3.1 along with the list of configurable parameters and the references to the versions used in this thesis. At a high level, each of the referral algorithms presented in Table 3.1 computes a score for every expert e_i (denoted by *score*(e_i)) and selects the expert with the highest score breaking any remaining ties randomly. 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 (exploitation), while ensuring enough exploration to make sure we are not missing out on stronger experts. For exploration purpose, some of these algorithms (e.g., ϵ`-Greedy`) contain a diversification component which allows it to randomly select an expert with some small probability. Some algorithms use reward variance (e.g., `DIEL`) or sampling frequency (e.g., `UCB1`) for similar purpose. In what follows we give a short description of the learning algorithms along with the pseudo-code.

3.4.1 `DIEL`

First proposed in (Kaelbling, 1993), Interval Estimation Learning (`IEL`) has been extensively used in stochastic optimization (Donmez et al., 2009a) and action selection problems (Wiering and Schmidhuber, 1998; Berry and Fristedt, 1985). Action selection using Distributed Interval Estimation Learning (`DIEL`) works in the following way (Donmez et al., 2009a; KhudaBukhsh et al., 2016a). First, for each expert e_i, $UI(e_i)$, the upper confidence interval for the mean reward

is estimated by

$$UI(e_i) = m(e_i) + \frac{s(e_i)}{\sqrt{n_i}} \tag{3.1}$$

Next, DIEL selects the expert with the highest upper confidence interval. Every reward vector is initialized with two rewards of 0 and 1, allowing us to initialize the mean and variance, i.e., $\forall i, n_i = 2$ and $\mathbf{r}_{i,n_i} = (0, 1)$.

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 $m(e)$ 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, $n_i \leftarrow n_i + 1$

3.4.2 DMT

Algorithm 2: DMT(e, T)

Initialization: $\forall i, n_i \leftarrow 2$, $\mathbf{r}_{i,n_i} \leftarrow (0, 1)$
Loop: Select expert e_i who maximizes

$$score(e_i) = m(e_i)$$

Observe reward r
Update $\mathbf{r}_{i,n_i}$ with r, $n_i \leftarrow n_i + 1$

Unlike DIEL, DMT only considers the mean observed reward and always greedily picks the expert with the highest observed reward.

It is trivial to prove that DMT can easily get stuck with a sub-optimal choice and thus is not an efficient choice for a referral algorithm. To simplify the proof, we consider continuous reward distributions with fixed range (0,1). Suppose e only has two colleagues e_1 and e_2. e_1 has a constant reward distribution of 0.25 and e_2 has a Bernoulli distribution with $p = 0.75$. With probability 0.25 (if the first observed reward from e_2 is 0), e can remain forever stuck with e_1.

3.4.3 ϵ-Greedy

DMT, being purely greedy, can easily get stuck with a sub-optimal referral choice. ϵ−Greedy performs a diversification step with a probability ϵ. i.e., with probability ϵ, it randomly chooses one of the connected experts for referral. We consider two different variants of ϵ−Greedy1. ϵ−Greedy1 ϵ−Greedy1 differs from ϵ−Greedy only in its way of setting the diversification probability parameter (set to $\frac{\alpha * K}{N}$ where K is the subnetwork size, i.e., the total number of referral choices).

Algorithm 3: ϵ-Greedy(e, T)

Initialization: $\forall i, n_i \leftarrow 2$, $\mathbf{r}_{i,n_i} \leftarrow (0, 1)$
Loop: Define e_{best} = arg max $m(e_i)$
With probability ϵ, refer to randomly chosen expert e_i
With probability $1 - \epsilon$, refer to e_{best}, $i \leftarrow best$
Observe reward r
Update $\mathbf{r}_{i,n_i}$ with r, $n_i \leftarrow n_i + 1$

3.4.4 UCB1

UCB1 belongs to the well-studied *upper confidence bound* family of algorithms first proposed in (Lai and Robbins, 1985). The proposed algorithm was subsequently simplified in (Agrawal, 1995; Auer et al., 2002) and we use the version presented in (Auer et al., 2002). Over the last few decades, UCB class of algorithms has received substantial attention in the bandit community and several well-known variants exist (see, e.g., KL−UCB (Garivier and Cappé, 2011), MOSS (Audibert and Bubeck, 2010), UCBV (Audibert et al., 2007), and Bayes−UCB Kaufmann et al. (2012)).

Similar to DIEL and the Greedy variants, the exploitation component of UCB1 is also mean-based. However, the exploitation function based on sampling frequency is different. UCB1 selects the expert with highest $m(e_i) + \sqrt{\frac{2lnN}{n_{e_i}}}$. This implies among two experts with equal mean reward, UCB1 will favor the least sampled one.

Algorithm 4: UCB1(e, T)

Initialization: Refer to each expert once
For the n-th query select expert e_i who maximizes
score$(e_i) = m(e_i) + \sqrt{\frac{2lnN}{n_i}}$
Observe reward r
Update: update $\mathbf{r}_{i,n_i}$ with r, $n_i \leftarrow n_i + 1$

3.4.5 `UCB2`

`UCB2` executes in an episodic fashion. Once an expert is selected, for an entire episode, all referrals go to her. For each expert e_i, it first initializes $epoch_{e_i}$ to 0 where $epoch_{e_i}$ denotes the number of episodes e_i has been chosen for referral. In the beginning, each expert is referred once. Once the last chosen expert e_j has been referred for $episode_{e_j}$ times, the new expert is selected by maximizing $m(e_i) + \sqrt{\frac{(1+\alpha).ln(eN\tau(epoch_{e_i}))}{2\tau(epoch_{e_i})}}$ where

$$\tau(epoch_{e_i}) = (1+\alpha)^{epoch_{e_i}} \tag{3.2}$$

$$episode_{e_j} = \tau(epoch_{e_j} + 1) - \tau(epoch_{e_j}) \tag{3.3}$$

and α is a configurable parameter.

Algorithm 5: UCB2(e, T)

Parameters: α
Initialization: $\forall i, epoch_{e_i} \leftarrow 0$, refer to each expert once
Loop:

1. Select expert e_i maximizing *score*$(e_i) = m(e_i) + \sqrt{\frac{(1+\alpha).ln(eN\tau(r_{e_i}))}{2\tau(r_{e_i})}}$
2. Refer to expert e_i for $episode_{e_i}$ times
   Observe reward r
   Update $\mathbf{r}_{i,n_i}$ with r, $n_i \leftarrow n_i + 1$
3. Update $epoch_{e_i} \leftarrow epoch_{e_i} + 1$

3.4.6 `UCB-normal`

`UCB-normal` performs any action that has been executed less than $\lceil logN \rceil$. If no such action exists, the action with highest $m(a) + \sqrt{16.\frac{sq(a) - n_a.m(a)^2}{n_a - 1}.\frac{ln(N-1)}{n_a}}$ is chosen ($sq(a)$ is the sum of squared reward obtained from action a).

Algorithm 6: UCBNormal(e, T)

Initialization: Select an expert that has been executed less than $\lceil logN \rceil$
For the n-th query select expert e_i who maximizes
score(e_i) = $m(a) + \sqrt{16.\frac{sq(a)-n_a.m(a)^2}{n_a-1}.\frac{ln(N-1)}{n_a}}$
Observe reward r
Update: update $\mathbf{r}_{i,n_i}$ with r, $n_i \leftarrow n_i + 1$

3.4.7 UCBV

Similar to DIEL, UCBV also uses variance to compute expected reward. However, it uses a different exploration function, $\frac{logN}{n_a}$. UCBV selects the action with highest $m(a) + s(a).\sqrt{\frac{2\theta log(N)}{n_a}} + \frac{3\theta log(N)}{n_a}$. (Audibert et al., 2007) reported a value of 1.2 for the parameter θ to guarantee logarithmic convergence.

Algorithm 7: UCBV(e, T)

Initialization: Refer to each expert once
For the n-th query select expert e_i who maximizes
score(e_i) = $m(e_i) + s(e_i).\sqrt{\frac{2\theta log(N)}{n_i}} + \frac{3\theta log(N)}{n_i}$
Observe reward r
Update: update $\mathbf{r}_{i,n_i}$ with r, $n_i \leftarrow n_i + 1$

3.4.8 Q-Learning

Q-Learning (Watkins and Dayan, 1992) is a well-known model-free reinforcement learning algorithm used in a wide variety of applications. Under some mild conditions (Jaakkola et al., 1994; Tsitsiklis, 1994; Littman and Szepesvári, 1996), it is proven that the policy Q converges to the optimal policy Q^* in the limit. Q-Learning is an important internal component to several gradient-based learning algorithms specifically designed for multi-agent interactions (Abdallah and Lesser, 2008; Bowling, 2005; Zhang and Lesser, 2010). Q-Learning has several well-known variants (see, e.g., Speedy Q-Learning (Azar et al., 2011), Double Q-Learning (van Hasselt, 2010), Frequency Adjusted Q-Learning (Kaisers and Tuyls, 2010), Repeated Update Q-Learning (Abdallah and Kaisers, 2016) etc.) of which, we study the performance of Double Q-Learning for learning referrals.

At each step, Q-learning select an action (in our case, a referral decision) and observes the state-transition (in our case, no state transition happens since we have a single state) and reward. After each step, the Q function is updated using the following equation.

$$Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \alpha(r + \gamma \, max \, Q_t(s',a') - Q_t(s,a)) \tag{3.4}$$

Q-learning has two parameters: γ and α. γ is called discount factor which controls the relative importance of future rewards. α is called the learning rate which controls the speed of learning. The two most common `exploration` strategies for Q-learning is ϵ-`Greedy` exploration and Boltzmann exploration (Sutton and Barto, 1998). As shown in Algorithm 8, the variant of `Q-Learning` we used is ϵ-`Greedy` for which we have an additional parameter, ϵ that balances the exploration-exploitation tradeoff.

Algorithm 8: Q-Learning(e, T)

Initialization: Initialize Q arbitrarily, observe s

Loop:

1. With probability 1 - ϵ, select expert e_i who maximizes the Q function
2. With probability ϵ, randomly choose an expert
3. Update Q function using equation 3.4

3.4.9 `DQ-Learning`

Double Q-Learning, or `DQ-learning`, maintains two Q functions: Q^A and Q^B. After each action and observing the subsequent reward and state (in our case, we have just one constant state), one of the two Q functions is chosen for update. Maintaining two such Q functions has the following benefit. Unlike traditional Q-Learning that often overestimates the maximum expected reward, the double estimator approach sometimes underestimates the maximum expected reward. In practice, `DQ-Learning` tends to converge faster than `Q-Learning` (for further details, see (van Hasselt, 2010)). For the `Q-Learning` component in `DQ-learning`, we considered ϵ-`Greedy-Q-Learning`.

3.4.10 `Thompson Sampling`

`Thompson Sampling` was first proposed in the 1930's (Thompson, 1933) and the finite-time regret bound remained unsolved for decades (Agrawal and Goyal, 2012) until recent results on its competitiveness with algorithms with provable regret bounds renewed interest (Chapelle and Li, 2011; Graepel et al., 2010).

In all our experiments, we consider `Thompson Sampling` (`TS`) with Beta priors (other types of priors include Jeffreys prior in Korda et al. (2013), Gaussian prior in Agrawal and Goyal

Algorithm 9: DQ-Learning(e, T)

Initialization: Initialize Q^A, Q^B arbitrarily, observe s

Loop:

1. Refer to expert based on $Q^A(s, .)$ and $Q^B(s, .)$, observe r, s'
2. Choose (e.g., random) either UPDATE(A) or UPDATE(B)
3.

if *UPDATE(A)* **then**

Define a^* = arg max$_a$ $Q^A(s', a)$

$Q^A(s, a) \leftarrow Q^A(s, a) + \alpha(s, a)(r + \gamma\, Q^B(s', a^*) - Q^A(s, a))$

else

Define b^* = arg max$_a$ $Q^B(s', a)$

$Q^B(s, a) \leftarrow Q^B(s, a) + \alpha(s, a)(r + \gamma\, Q^A(s', a^*) - Q^B(s, a))$

end

4. $s \leftarrow s'$

(2013)). At each step, for each expert e_i, `TS` first samples θ_i from $Beta(S_{e_i} + 1, F_{e_i} + 1)$ where S_{e_i} and F_{e_i} are the number of observed successes (reward = 1) and failures (reward = 0). Next, `TS` selects the expert 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.

Algorithm 10: TS(e, T)

Initialization: $\forall i, S_{e_i} \leftarrow 0, F_{e_i} \leftarrow 0$

For the n-th query select expert e_i who maximizes

score(e_i) = $Beta(1 + S_{e_i}, 1 + F_{e_i})$

Observe reward r

Update:

if $r == 1$ **then**

$S_{e_i} \leftarrow S_{e_i} + 1$

else

$F_{e_i} \leftarrow F_{e_i} + 1$

end

3.4.11 `Optimistic Thompson Sampling`

`Optimistic TS` (May et al., 2012) is very similar to `TS`. The only restriction is 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 we always believe 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*.

For continuous rewards, we modify both `Optimistic TS` and `TS` the same way as presented in Agrawal and Goyal (2012). We consider the observed reward as the parameter for a Bernoulli trial. The outcome of the trial is then considered as the reward.

Algorithm 11: Optimistic TS(e, T)

Initialization: $\forall i, S_{e_i} \leftarrow 0, F_{e_i} \leftarrow 0$
For the n-th query select expert e_i who maximizes
score(e_i) = max($m(e_i), Beta(1 + S_{e_i}, 1 + F_{e_i})$)
Observe reward r
Update:
if $r == 1$ **then**
    $S_{e_i} \leftarrow S_{e_i} + 1$
else
    $F_{e_i} \leftarrow F_{e_i} + 1$
end

3.4.12 `Expertise-Blind`

The baseline for our experiments is an `Expertise-Blind` algorithm where the initial expert randomly chooses a connected expert for referral. Essentially, the baseline allows us to determine how much learning appropriate referral adds to the performance.

3.4.13 `Upper Bound`

Our 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.

Parameter	Description	Distribution
τ	P(*ReferralLink*(v_i, v_j))	Uniform(0.01, 0.1)
c	$= \tau + c\ Sim(e_i, e_j)$.	Uniform(0.1,0.2)
μ_1	Truncated mixture of two	Uniform(0,b)
μ_2	Gaussians for topics	Uniform(b,1)
		$b \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$
σ_1		Uniform(0.2,0.4)
σ_2		Uniform(0.05,0.15)
w_2		$\mathcal{N}(0.03, 0.01)$, $w_2 \geq 0$

Table 3.2: Parameters for synthetic data set

3.5 Experimental Setup

Referral-learning algorithms: We compared the performance of twelve referral-learning algorithms (listed in Table 3.1), a topical upper bound, and a random (expertise-blind) baseline. Each parameterized referral algorithm was tuned on a separate training set constructed using the same parameter distribution described in Table 3.2 (KhudaBukhsh et al., 2016c). The ϵ-Greedy algorithm, as presented in (Auer et al., 2002), requires prior knowledge about the reward distribution in order to set the value of the hyper-parameter d. However, we found that estimating d from the observations created sub-par performance. Setting instead ϵ to $\frac{\alpha * K}{N}$ (where K is the subnetwork size and N is the number of total observations) gave rise to a good performance when appropriately configured. We followed a similar procedure to set ϵ for ϵ`-Greedy Q-learning`,

Algorithm configuration: We first generated a small training data set (10 scenarios) using the same distributional parameters used for our test set. Then, for each algorithm, we ran 100 random instantiations of the algorithm on the training data set and selected the configuration that performed best on this set. For computational tractability, we chose a smaller training data set. It is unlikely that our configured algorithms suffered from overfitting because all our configured algorithms achieved superior performance on the test set than corresponding performance reported in (KhudaBukhsh et al., 2016c).

Synthetic data set: Our test set for performance evaluation (KhudaBukhsh et al., 2016c,a). consisted of 1000 scenarios, each with 100 experts, 10 topics and a referral network. For our per-instance query budget, Q, we chose the values 2, 3, and 4, corresponding to single-hop, two-hop and three-hop referrals, respectively.

Upper bounds and baseline: Our upper bound for single-hop referral (KhudaBukhsh et al., 2016c) 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.

For two-hop referrals, we use an upper bound based on calculating optimal referral choices up to depth 2 – say an expert e_i has two referral choices, e_j and e_k, with an expertise of 0.4 and 0.5 respectively, and the maximum expertise under e_j and e_k's subnetworks are 0.9 and 0.2 respectively; our upper bound will choose e_j over e_k because of a higher overall solution probability. Finally, the baseline is an `Expertise-Blind` algorithm where the initial expert randomly chooses a connected expert for referral.

Non-synthetic data set: The 100 `SATenstein` solvers we used are obtained by configuring `SATenstein2.0` version (described in (KhudaBukhsh et al., 2016d)) on six well-known SAT distributions. Each of these solvers is configured on one of the six SAT distributions listed in Table 3.3. We used the test sets of the SAT distributions as our pool of tasks. For selecting an instance from a distribution, we used random sampling with replacement. Detailed descriptions of the SAT distributions can be found in (KhudaBukhsh et al., 2009).

Performance Measure: Our performance measure is the overall task accuracy of our multi-expert system. So if a network receives n tasks of which m tasks are solved (either by the *initial expert* or the *referred expert*), the overall task accuracy is $\frac{m}{n}$. For our experiments on synthetic data set, each algorithm is run on the data set of 1000 referral networks and the average over such 1000 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.

Computational environment: We carried out our experiments involving SAT solvers on a cluster of dual-core 2.4 GHz machines with 3 MB cache and 32 GB RAM running Linux 2.6. Experiments on synthetic data were carried out on Matlab R2016 running Windows 10.

3.6 Performance Comparison on Synthetic Data

We conduct the experiments on our synthetic data in two stages. In the first stage, for single-hop referral, we consider all algorithms presented in Table 3.1. For multi-hop referral, we consider a subset of high-performance algorithms identified in the single-hop setting experiments.

3.6.1 Single-hop Referral

For single referral, Table 3.4 presents the mean task accuracy across the entire data set at specific points of the horizon (samples per subnetwork). For a given horizon, the best performance (not considering the upper-bound oracle) is highlighted in bold. Our results show that except during the very early stages of the simulation, `DIEL` dominated all the other referral algorithms with

Distribution	Description	Generator Parameters	Size
QCP	SAT-encoded quasi-group completion problems (Gomes and Selman, 1997)	order $O \in [10, 30]$; holes $H = h * O^{1.55}$, $h \in [1.2, 2.2]$	1000 instances
SW-GCP	SAT-encoded small-world graph-colouring problems (Gent et al., 1999)	ring lattice size $S \in [100, 400]$; nearest neighbors connected: 10; rewiring probability: 2^{-7}; chromatic numbers: 6	1000 instances
R3SAT	uniform-random 3-SAT instances (Simon, 2002)	variable: 600; clauses-to-variables ratio: 4.26	250 instances
HGEN	random instances generated by HGEN2 (Hirsch, 2002)	variable $n \in [200, 400]$	1000 instances
FAC	SAT-encoded factoring problems (Uchida and Watanabe, 1999)	prime number $\in [3000, 4000]$	1000 instances
CBMC	SAT-encoded bounded model checking (Clarke et al., 2004), preprocessed by SatELite (Eén and Biere, 2005)	array size $s \in [1, 2000]$; loop unwinding $n \in 4, 5, 6$	302 instances

Table 3.3: Six benchmark SAT distributions mapping to topics

a performance approaching the optimal upper bound. A paired t-test reveals that beyond the crossover point (1000 samples per subnetwork), `DIEL` is better than all other referral algorithms with p-value less than 0.0001. Algorithms with provable performance guarantees may catch up with `DIEL` given a sufficiently large horizon, but from a practical standpoint, `DIEL` is an effective referral algorithm to handle real-world scenarios. We extended a random subset of 200 scenarios up to a horizon of 20,000 samples per subnetwork, at which time none of the top performing referral-algorithms from each category had caught up with `DIEL`.

The large performance difference between the upper bounds and the baseline shows that effective referral mechanisms can make a significant difference. Also, in every category, the best algorithm performed substantially better than the baseline indicating that `learning-to-refer` is a tractable challenge even with an uninformative prior. In a more realistic setting, with informative priors on several expert-topic pairs, fewer samples may be required.

For the remaining results, we retained only the best-performing algorithms in each category, as follows: `DIEL` (IEL category), ϵ`-Greedy1` (Greedy category), `UCB1` (UCB category) and `DQ-learning` (Q-learning category). For comparison, we included additionally, `DMT`, a horizon-free algorithm. For the remainder of the thesis, we denote ϵ`-Greedy1` as ϵ`-Greedy`.

	500	1000	1500	2000	3000	4000	5000
Upper Bound	79.47	79.31	79.27	79.42	79.38	79.41	79.47
DIEL	67.73	**73.35**	**75.35**	**76.33**	**77.33**	**77.76**	**77.96**
DMT	**70.63**	73.06	73.83	74.20	74.54	74.71	74.69
ϵ-Greedy	55.33	56.63	57.80	58.95	60.57	61.95	62.97
ϵ-Greedy1	70.22	72.91	73.97	74.48	74.92	75.19	75.32
UCB1	57.78	59.60	60.80	61.61	63.28	64.49	65.49
UCB2	63.71	64.16	64.19	64.21	64.18	64.19	64.28
UCB-normal	54.38	54.47	54.71	54.97	56.43	58.91	61.39
UCBV	54.99	55.92	56.44	56.87	57.83	58.60	59.15
Q-Learning	65.46	69.19	70.98	72.08	73.46	74.27	74.75
DQ-learning	70.23	72.68	73.74	74.37	75.14	75.60	75.91
TS	62.96	66.93	69.38	71.07	73.26	74.56	75.49
Optimistic TS	63.90	67.67	70.05	71.62	73.63	74.86	75.66
Expertise-Blind	54.48	54.46	54.45	54.41	54.48	54.44	54.60

Table 3.4: Performance comparison of referral algorithms.

3.6.2 Multi-hop Referral

In a multi-hop setting, a referred expert can continue referring an instance to another expert as long as the budget permits (excluding cyclic referrals). Figure 3.2 compares our top-performing referral algorithms with query budget 3 and 4. In Figure 3.2(a), we compare the performance with an upper bound that calculates optimal choice to depth 2 (optimal choice to depth 1 but the same query budget 3 achieved a task accuracy of 93.05%). Understandably, with a higher query budget, the overall task accuracy of every learning algorithm increases. However, DIEL's rapid performance gain in the early phase of learning still enables it to obtain a superior performance. The practical benefit of DIEL against algorithms with theoretical convergence guarantees is particularly evident when compared against UCB1. In fact, DIEL with a lower query budget (Q = 3) achieves a better performance than UCB1 with a higher budget (Q = 4).

However, in three-hop setting, we found that Optimistic TS marginally outperformed DIEL towards the end of the horizon. The exploration component of DIEL dies down quickly, while Optimistic TS continually explores albeit at an increasingly reduced rate. This continual exploration enables Optimistic TS to find experts who were themselves weak but eventually found stronger referral options thus improving their effective strength. To address this issue, we proposed ϵ-DIEL, a variant of DIEL with an additional exploration component. Recall that, DIEL picks an expert with the highest score where the score is defined as $m(e_i) + \frac{s(e_i)}{n_{e_i}}$. We break down the score into two factors: $m(e_i)$, the mean observed reward, as the *exploitation*

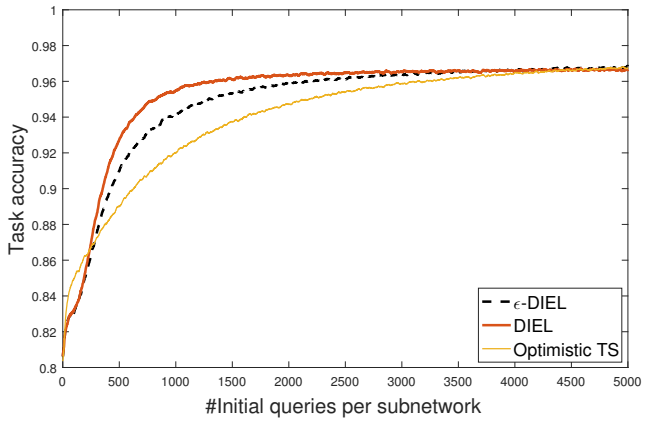
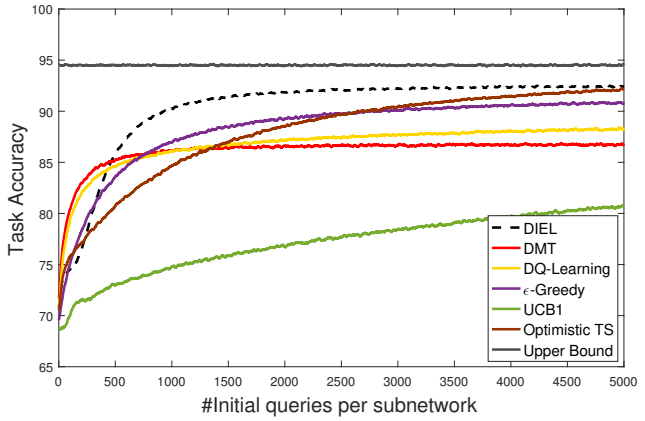
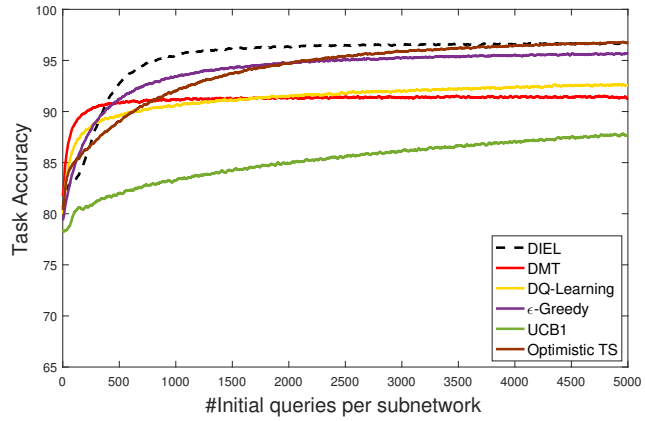


Figure 3.1: Performance comparison of ϵ-`DIEL`, `DIEL` and `Optimistic TS` with query budget $Q = 4$



(a) Two-hop



(b) Three-hop

Figure 3.2: Multi-hop referral

Algorithm 12: ϵ-DIEL(e, T)

Initialization: $\forall i, n_i \leftarrow 2$, $\mathbf{r}_{i,n_i} \leftarrow (0, 1)$
Loop: Define e_{best} = arg max $m(e_i) + \frac{s(e_i)}{\sqrt{n_i}}$
  if exploration$(e_{best}) \leq thresh$ **then**
    with probability ϵ, refer to randomly chosen expert e_i
    with probability $1 - \epsilon$, refer to e_{best}, $i \leftarrow best$
  else
    refer to e^*, $i \leftarrow best$
  end
  Observe reward r
  Update $\mathbf{r}_{i,n_i}$ with r, $n_i \leftarrow n_i + 1$

factor and $\frac{s(e_i)}{n_{e_i}}$, the sample-size adjusted variance term, as the *exploration factor*. When `DIEL` settles down with a referral choice, the *exploration factor* of the best expert slowly goes down as the best expert continues to receive referrals because of a high *exploitation factor*. When the *exploration factor* is below a threshold, we introduce an ϵ-`Greedy`-like exploration step, i.e., randomly picking an expert for referral with a small probability ϵ. As shown in Algorithm 12, ϵ-`DIEL` has one parameter, $thresh$, set to 0.1 in our experiments. Figure 3.1 shows that with this modification to `DIEL`, ϵ-`DIEL` obtains marginally better steady-state performance than `Optimistic TS` while sacrificing some early learning advantage as obtained in `DIEL`.

Multi-hop referrals introduce non-stationarity in expertise in a sense that a weak expert can find a strong colleague in a later part of the simulation which effectively changes her expertise. Our results reveal `DIEL`'s vulnerability to expertise drift; we perform an in-depth analysis of this phenomenon and propose a novel algorithm `Hybrid`, a combination of `Thompson Sampling` variants and `DIEL`, in Chapter 4.3.

3.7 Evaluation on Referral Network of SAT Solvers

In order to determine the true impact of learning-to-refer, we move beyond synthetic domains, starting with a network of expert SAT solvers. For a given propositional formula F, the satisfiability problem (SAT) asks if there exists a complete assignment of truth values to the variables of F under which F evaluates to true (see, e.g., (Biere et al., 2009)). Many of the hard combinatorial problems that arise in practical scenarios belong to the complexity class of so-called $\mathcal{NP}$-complete problems (Cook, 1971; Applegate et al., 2011; Biere et al., 1999; Fraenkel, 1993; Pop et al., 2002), reduce to SAT, one of the most-studied $\mathcal{NP}$-complete problems. For example, SAT solves the core problems arising in applications such as planning (Kautz and Selman, 1996,

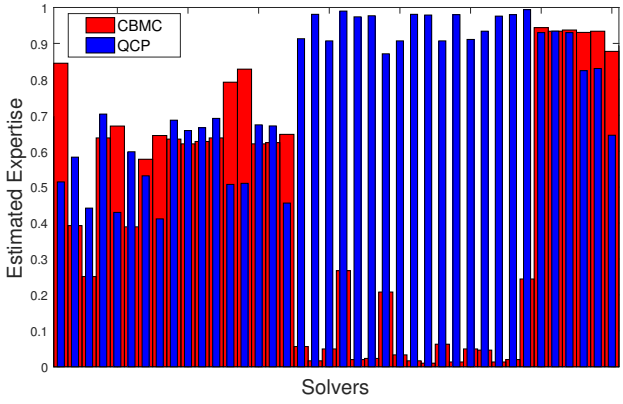


Figure 3.3: Expertise estimates of a subset of solvers on background data of two SAT distributions `CBMC` (bounded model checking) and `QCP` (quasi-group completion problems)

1999), scheduling (Crawford and Baker, 1994), graph-coloring (Gelder, 2008), bounded model checking (Biere et al., 1999), and formal verification (Stephan et al., 1996). Given the efficiency of modern SAT solvers, many of these otherwise intractable problems are solved at significant scale, even given the theoretical $\mathcal{NP}$ limit.

High-performance Stochastic Local Search (SLS) solvers map to experts and Boolean satisfiability (SAT) instances to tasks/queries. This mapping is particularly appealing because SLS solvers work better or worse (= skills) depending on the SAT distributions (= topics). There are also large number of SAT-solver experts with varying topical expertise (see, Figure 3.3) available in the form of `SATenstein` solvers (KhudaBukhsh et al., 2009) (which map to our experts). In our experiments, we use 100 `SATenstein` solvers configured on six well-known SAT distributions (which map to our topics), which include real applications such as factoring and model-checking (KhudaBukhsh et al., 2016d). Moreover, verifying if the task is solved correctly is straightforward: the solver finds a satisfying model of the instance. Finally, we may measure time-to-solution in order to explore continuous rewards, rather than just binary, i.e., whether the SAT instance is solved within the allocated time bound. In real life, many tasks involve task-responses beyond simple binary states (e.g., what fraction of all constraints an optimization algorithm satisfies, by how far the prediction of a stock value is off, in a scale of 1-10, how confident the doctor is in diagnosing her patient with stage-two melanoma). Hence, evaluating referral algorithms on a continuous reward setting is important as an effective referral will maximize not only solution likelihood but also solution quality.

SAT being an $\mathcal{NP}$-complete problem, run times do not have any known closed-formd parameterized distribution which allows us to evaluate performance in the wild. Because of this,

the data set can be utilized as meaningful benchmarks for evaluating MAB algorithms in general where there is a serious lack of empirical evaluations beyond synthetic data set barring few (Chapelle and Li, 2011; Kandasamy et al., 2017).

We set the budget C for solving each instance to 1 CPU second, which is the maximum time in which, on a similar computing architecture, configured high-performance `SATenstein` solvers were found to solve a majority of the instances in their expertise area (KhudaBukhsh et al., 2016d) (This was corroborated in our experiments). The reward is computed as (C - r_t) where r_t is the run time (when a solver fails to solve an instance, r_t = 1). With C set to 1 in our experiments, the reward is bounded by [0, 1) with a failed task fetching a reward of 0 and higher rewards implying faster solutions. So in this setting, through continuous reward, we have incorporated solution quality (in this case, run time) in our experiments. We also considered binary rewards in our experiments; for which, we simply assign a reward of 1 when the referred solver finds a satisfying model and 0 when the referred solver times out.

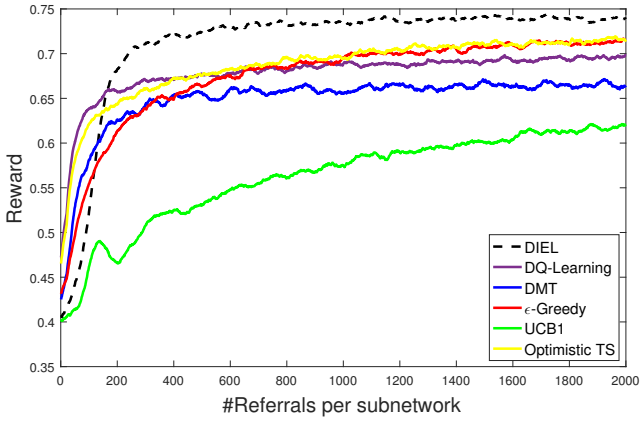
Figure 3.4 presents the performance comparison of referral-learning algorithms where experts are SAT solvers and topics are SAT problem distributions on 10 randomly chosen referral networks. We found that `DIEL` outperformed all other algorithms, with `Optimistic TS`, `DMT`, `DQ-learning`, and ϵ`-Greedy1` achieving a performance close to `DIEL` (even when we extended the runs to 4000 referrals per subnetwork for ϵ`-Greedy` and `Optimistic TS`, they had not yet caught up with `DIEL`). Similar to the results obtained on our synthetic data, we found that `UCB` had the slowest rate of improvement in the initial stage of learning. These results highlight the following. First, with real experts, a well-defined task and very few distributional assumptions on expertise, we found that learning effective referral choices that improve solution quality is possible. Second, `DIEL`'s superiority over other referral-learning algorithms is not just restricted to synthetic data, nor the consequence of binary rewards.

3.8 Revisting the Research Questions

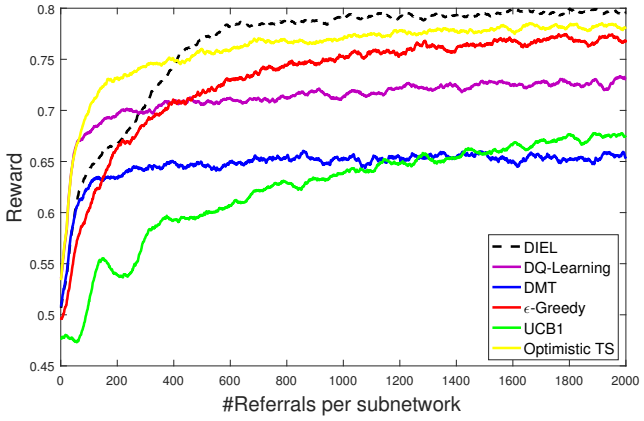
We now revisit the research questions and present our main takeaways:

How to learn effective referral choices? Since the problem of learning appropriate referrals can be cast in various ways, our primary goal was not to design new algorithms, we rather focused on each different paradigm and identify few key algorithms in those settings, thus forming a diverse pool of existing algorithms. We considered the following categories of algorithms: multi-armed bandit algorithms (`UCB` class and `Thompson Sampling` variants), Q-learning variants, Interval Estimation Learning (IEL) algorithms and Greedy variants.

We found that several algorithms showed competitive performance with `DIEL` leading the



(a) Continuous Reward



(b) Binary Reward

Figure 3.4: Performance comparison on SAT solvers as experts and SAT solving as the task

pack. In multi-hop setting, `Optimistic TS`'s performance improves with an increased number of hops. We also presented a novel algorithm ϵ-`DIEL` that performs comparably with `Optimistic TS`. However, this result reveals `DIEL`'s vulnerability to expertise drift, a case we analysis in Chapter 4.3.

Do close-to-optimal local decisions translate into close-to-optimal global decisions? Yes. In our distributed setting, each expert were learning referral policies for each topic on its own. Yet, the performance of `DIEL` and many other algorithms were close to the oracle `Upper Bound`.

How should we evaluate performance beyond synthetic data? We considered a suite of high-performance SAT solvers as experts to overcome this challenge. The SAT solver data set was useful in evaluating performance both with continuous and binary reward allowing us to test algorithms on conditions where expertise do not obey any known parameterized form.

Chapter 4

Robustness to Practical Factors

Several practical factors may impact the performance of a referral network. In this chapter, we primarily focus on three of them:

- capacity constraints (limits on number of tasks per time period)
- evolving networks (changes in connectivity or agents joining or leaving the referral network) and
- expertise drift (skills improving over time or atrophying through disuse).

From a practical standpoint, assuming that experts can handle an unbounded number of requests within a specific time-window, is unrealistic. Capacity constraints put a restriction on the number of tasks an expert can solve within a given time period (we consider the number of tasks the overall network receives as a proxy for time). Beyond the permissible task-threshold, an expert becomes unavailable for a while till her load situation improves. In evolving networks, we consider a different type of unavailability of experts. We assume that the network experiences attrition of existing experts and an influx of new experts, and assess the robustness of our referral-learning algorithms. In expertise drift, the initial stationary assumption on the distributional parameters of expertise no longer holds modeling experts improving with time or losing skills because of factors like fatigue, disuse etc.

For capacity constraints and evolving networks, our experiments are limited to performance assessment of existing high-performance referral-learning algorithms in the static case. For expertise drift, by far the most challenging robustness criterion, we design new algorithms and compare the performance with the top-performing algorithm in the stationary expertise setting. Additionally, we studied networks generated using well-known random graph generators and effects on learning behavior when topic-identification is noisy.

The remainder of the chapter is organized as follows. We first analyze the performance

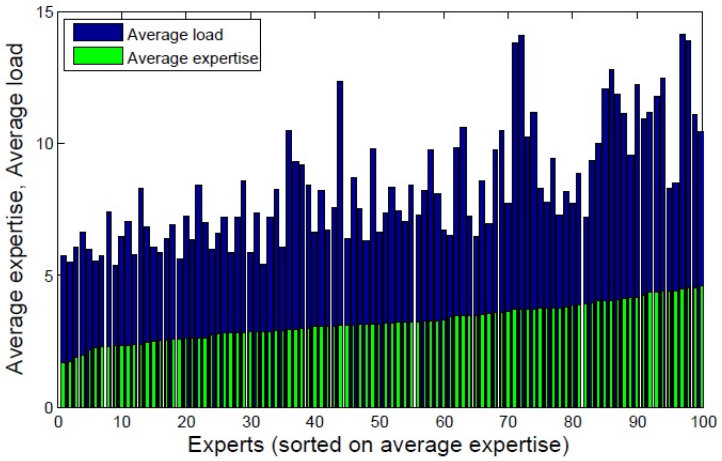


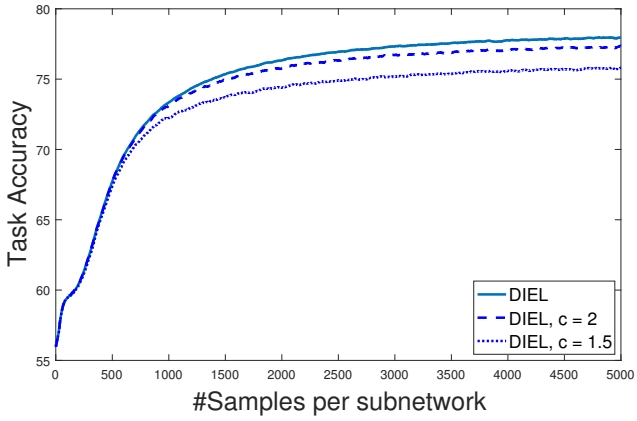
Figure 4.1: Average load on experts, overlaid with expertise. Experts are sorted in the ascending order on their mean expertise. Average load is computed per 1000 queries.

of referral-learning algorithms in presence of capacity constraints following which, we relax the static assumption on the network and consider two types of network changes: distributed change, single-point change. After studying the learning behavior on evolving networks, we present a detailed analysis of expertise drift. We conclude with reporting some additional robustness results.

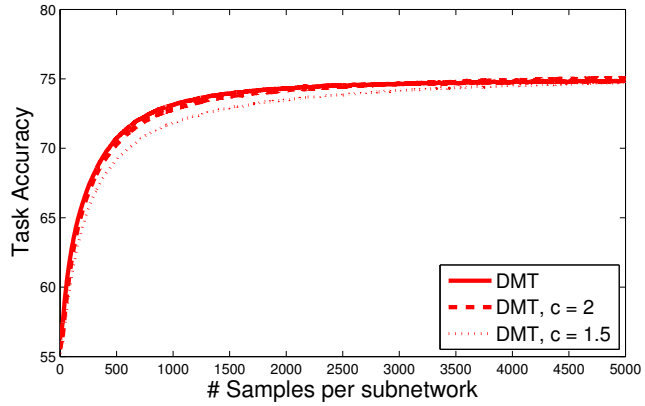
4.1 Capacity Constraints

Capacity constraints on experts are rarely considered in Active Learning (though Proactive Learning (Donmez and Carbonell, 2008) considers similar aspects). In reality, of course, experts can handle only a limited number of tasks at any given time, while the better experts may receive an ever increasing stream of requests. As a first step to analyze this issue, we note that average expertise and average load was moderately correlated (correlation coefficient 0.69) in our simulations (see, e.g., Figure 4.1).

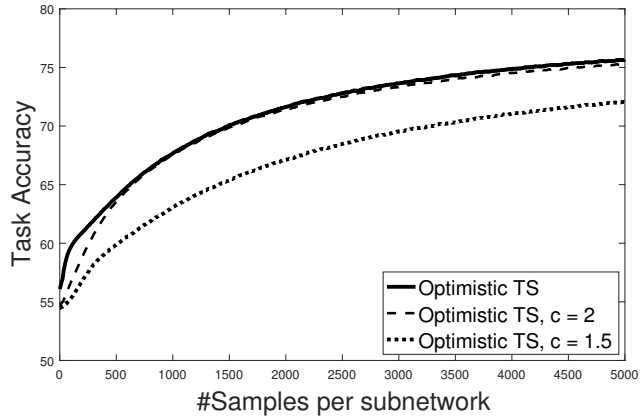
Let $load(e_i, m)$ denote the number of tasks expert e_i received among the last m tasks (*initial* or *referred*) the network received. In a network of k experts, a fair load for every expert is $\frac{m}{k}$. An expert is overloaded if $load(e_i, m) \geq c * \frac{m}{k}$, where c (load-factor) is greater than 1. While we can imagine an overloaded expert charging more money to solve a problem (or delaying her response, or referring to another expert), in our experiments we assumed that she becomes completely unavailable until the load situation improves. Figure 4.3 presents the average number of overloaded experts when `DIEL` and `DMT` are used with m set to 1000 queries and c set to



(a) Load-balanced `DIEL`



(b) Load-balanced `DMT`



(c) Load-balanced `Optimistic TS`

Figure 4.2: Performance of `DIEL`, `DMT` and `Optimistic TS` for different values of the load-factor c

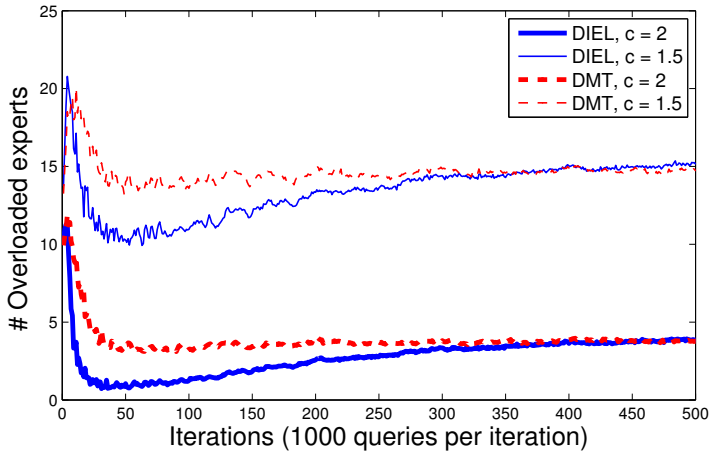


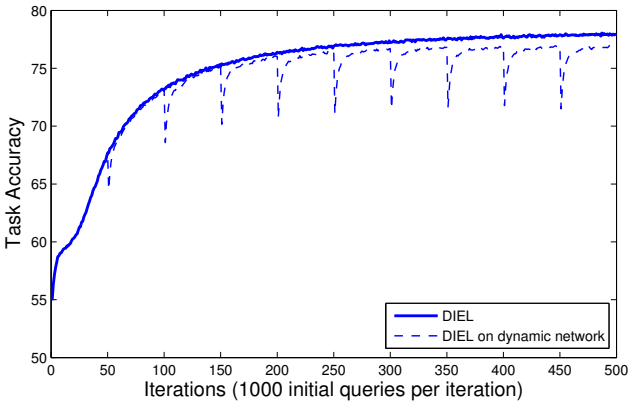
Figure 4.3: Number of overloaded experts per 1000 queries for different values of the load-factor c

1.5 and 2. We found that both DIEL and DMT had some overloaded experts throughout the entire period of time with a high number of overloaded experts in the early phase. Algorithms with greater exploration tend to have fewer overloaded experts in the early learning phase. As expected, the number of overloaded experts also increased considerably with a tighter value of c.

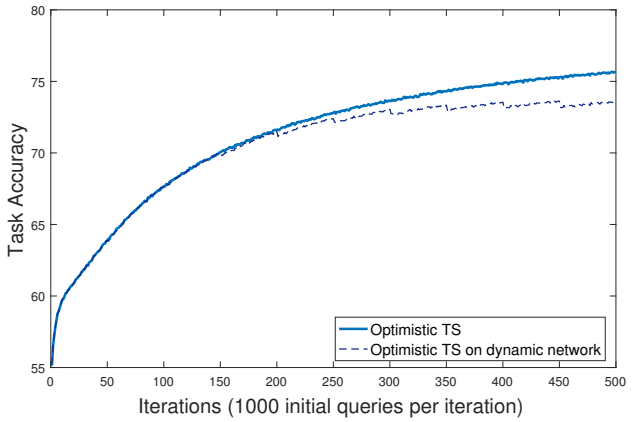
Even with a tighter value of c of 1.5, we find that the performance of the referral algorithms degrades gracefully, and sometimes even paradoxically improves, a phenomenon due to the forced exploration resulting from load balancing. For example, in Figure 4.2, while DIEL exhibits a graceful performance degradation with increased load factor, the load-balanced version of DMT with a load factor of 2 does slightly better than DMT without any capacity constraint. That we observed this property with all the referral algorithms leads us to conjecture that load balancing is facilitated by the distributed nature of the learning setting.

4.2 Evolving Networks

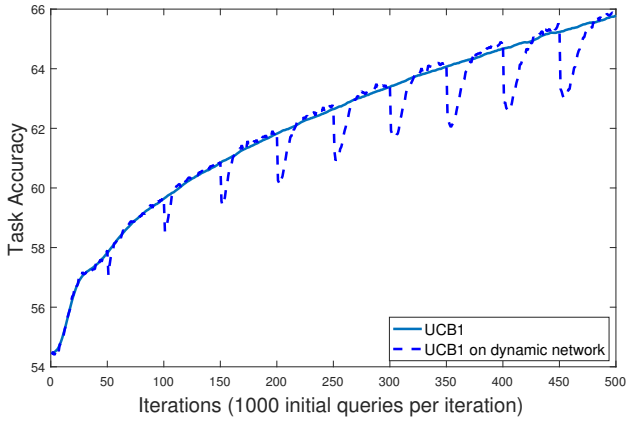
In practice, referral networks are not static; they evolve over time with new links being forged, experts dropping out, and new ones joining. We focused primarily on addition/deletion of new/old experts to the network. We considered this both in the form of a *single point change* (i.e. a catastrophic event, with 20% of the experts in the network replaced at iteration 100, where an iteration consists of 1000 initial queries), and a *distributed change* (modelling more closely a real-world gradual change: 5% of the network changes every 50 iterations). Note that, Our main reason to opt for this choice was to obtain a clearer visualization of the effect on network per-



(a) `DIEL`

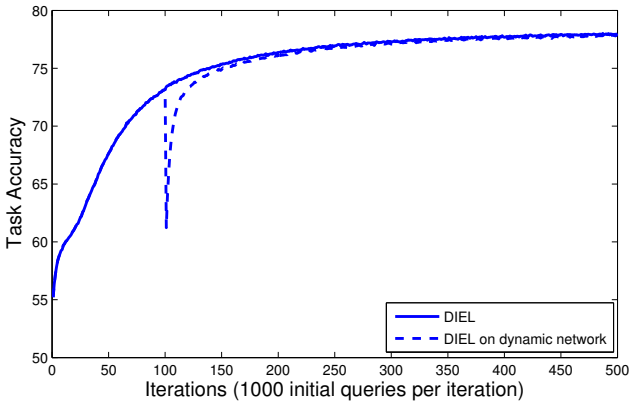


(b) `Optimistic TS`

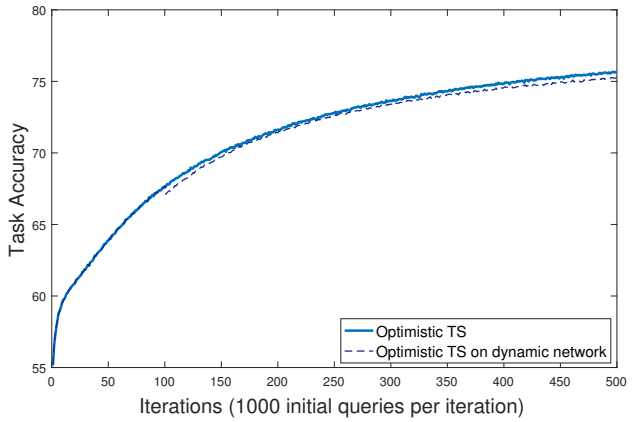


(c) `UCB1`

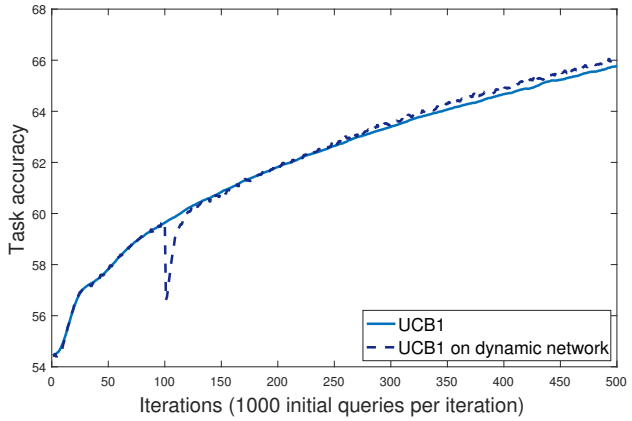
Figure 4.4: Performance of referral-learning algorithms with distributed 5% network change



(a) `DIEL`



(b) `Optimistic TS`



(c) `UCB1`

Figure 4.5: Performance of referral-learning algorithms with single point 20% network change

formance; we also ran experiments where the network changes are distributed across time-steps and found qualitatively similar performance.

Figure 4.4(a) compares the performance of DIEL on a static network with that on a dynamic network with distributed change. Our results show that DIEL coped fairly well with a distributed change, and in spite of multiple changes in the network at a regular interval, the final DIEL performance on a dynamic network (task accuracy 76.91%) is slightly worse than DIEL on a static network, but still better than any other referral learning algorithm presented in Table 3.4. Hence, the performance of DIEL on a dynamic network with distributed change is better than the performance of any other referral-learning we studied even on a static network. In addition, we ran experiments where no experts leave or join, but new referral links get created. Then too, the performance of the referral learning algorithms proved robust, exhibiting qualitatively similar characteristics. We found none of the algorithms had any problem with adjusting to a single point network change. Figure 4.5 shows the result for DIEL, UCB1 and Optmistic Thompson Sampling. Other referral algorithms showed qualitatively similar behavior.

The relative performance of the referral-learning algorithms raise few interesting observations. The performance dip in Optimistic TS after a network change occurs is much lesser than the performance dip in DIEL or UCB1. In fact, for the first few distributed changes, Optimistic TS showed no loss of performance. However, when the amount of change (cumulative or one time) is large, DIEL showed better recovery than Optimistic TS. In comparison with their static-network counterpart, surprisingly, UCB1 showed an improvement. Recall that, the exploration component of UCB1 is sampling-frequency based, favoring least sampled experts more. Hence, when a new set of experts arrive at a later stage of simulation, the newer experts are likely to get sampled more often. This effectively, temporarily restricts the search-space as UCB1 focuses more on the newer experts while favoring only those old experts with strong historical performance. This could be a possible explanation for the modest performance gain observed in UCB1.

At a high-level, capacity constraints and evolving networks are somewhat inter-related; in both cases, mainly one underlying previous assumption changes: availability of experts. When we consider capacity constraints, our assumption that an expert can handle an unbounded number of requests in a given amount of time no longer holds; the expert becomes unavailable whenever the number of requested tasks crosses some certain threshold. Similarly, evolving networks could be viewed as experts being available for a certain time-period then becoming completely unavailable. Intuitively, unavailability of a stronger expert will hurt the overall performance more than that of a weaker expert. However, one subtle difference is that stronger experts are more likely to get busier, hence relative expertise is a factor in capacity constraint but not in evolving

networks. If we look at the load-restricted behavior of `Optimistic TS`, we will find that similar to the dynamic-network behavior, `Optimistic TS` showed a strong resilience to less stricter load-conditions (*load-factor* 2), but performed substantially worse when *load-factor* was 1.5.

Evolving networks is revisited in Chapter 5 where we introduce proactive skill posting. In Chapter 5, we examined `DIEL`'s tolerance to more severe extent of distributed change (20% of the network changes every 50 iterations). We found that proactive skill posting was particularly useful when network evolves at such high rate, and the performance of proactive-`DIEL` was substantially better than that of `DIEL`.

4.3 Expertise Drift

The *partial information* (Burtini et al., 2015) or the *information obstacle* (Babaioff et al., 2014) present in multi-armed bandit (MAB) settings 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 in optimizing referral decisions. Hence, devising algorithms to deal with time-varying expertise would be a meaningful research challenge.

In what follows, we first present the research questions we considered. Next, we describe our modeling choices for expertise drift. In addition to Brownian bidirectional drift, the typical drift model in the literature, we also consider drift with positive bias, where agents mostly improve with practice. Finally, we present our experimental setup and results.

4.3.1 Research Questions

How to detect a drifting expert? All referral algorithms we have presented so far, will be able to detect an initially good expert whose performance deteriorates. However, if an expert initially exhibits low performance and then improves, designing referral algorithms that quickly detect

such expertise shifts could be a challenging task, given that previously dismissed low performers would have a low probability of being sampled by the current algorithms.

How to adapt to the drift, and quickly?

Realizing that our top-choice is no longer performing to her previous potential is important. However, from the network's performance point of view, an even more important aspect is to find another strong expert.

Can hybrid algorithms be useful in presence of expertise drift? Algorithms exhibit a varying range of approaches to strike up a balance between exploration and exploitation. Combining multiple algorithms could be proven beneficial in tackling expertise drift.

4.3.2 Modeling Drift

In previous work, (KhudaBukhsh et al., 2016c,a,b, 2017b), the expertise of an expert e_i on $topic_p$ was modeled as a truncated Gaussian distribution with small variance:

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

Here, we introduce the notion of drift in the following way. In a time-varying expertise setting, expertise of an expert e_i on $topic_p$ is expressed as

$\textit{expertise}(e_i, q_j \in topic_p) \sim \mathcal{N}(\mu_{topic_p,e_i,epoch_k}, \sigma_{topic_p,e_i}),$

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

We assume discrete changes at epoch boundaries, and for a given epoch, we assume the distributional parameters on expertise do not change. When μ_{drift} is 0, the unbiased drift is similar to the Brownian perturbation previously considered in (Gupta et al., 2011). Since *expertise* is a probability, it must remain within $[0, 1]$. We assume that once *expertise* reaches the boundary, it remains there until a drift moves it out of 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 biased drifts (with $\mu_{drift} > 0$), where the overall expertise of the experts in the network improves on certain topics over time.

4.3.3 Referral Algorithms

Similar to our previous algorithm descriptions, we fix the expert to e and topic to T. Recall that, for a given expert e and topic T, S_{e_i} and F_{e_i} denote the total number of observed successes and failures, respectively. Also, $m(e_i)$ and $s(e_i)$ denote the sample mean and sample standard deviation of the corresponding reward vector.

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 13: Pessimistic TS(e, T)

Initialization: $\forall i, S_{e_i} \leftarrow 0, F_{e_i} \leftarrow 0$
For the n-th query select expert e_i who maximizes
score(e_i) = min$(m(e_i), Beta(1 + S_{e_i}, 1 + F_{e_i}))$
Observe reward r
Update:
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

This action selection strategy is a new combination of `DIEL` and `Pessimistic TS`. As described in Algorithm 14,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).

Hybrid

Our `Hybrid` algorithm is a combination of `Optimistic TS` and `Pessimistic TS-DIEL`. 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 15. $perf_{w_i}$ is the

Algorithm 14: Pessimistic TS-DIEL(e, T)

Initialization: $\forall i, S_{e_i} \leftarrow 0, F_{e_i} \leftarrow 0$
For the n-th query select expert e_i who maximizes
score(e_i) = min$(m(e_i), Beta(1 + S_{e_i}, 1 + F_{e_i}))$
Observe reward r
Update:
if $r == 1$ **then**
    $S_{e_i} \leftarrow S_{e_i} + 1$
else
    $F_{e_i} \leftarrow F_{e_i} + 1$
end

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 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 $[+0.05, -0.05]$.

Algorithm 15: 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 perf$_{\Delta} < threshold$ **then**
            $switchFlag \leftarrow 1$
        end
        if perf$_{\Delta} > 0$ **then**
            $perf_{best} \leftarrow perf_{w_i}$
        end
    else
        execute `Pessimistic TS-DIEL`
    end
end

4.3.4 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 (KhudaBukhsh et al., 2016a, 2017b) 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., $\textit{mean}(\textit{Expertise}(e_i, q) : q \in \textit{topic}_p) \; \forall i, p$) 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.

Data set: Our test set for performance evaluation is the same data set used in (KhudaBukhsh et al., 2016b), which is a random subset of 200 scenarios also used in (KhudaBukhsh et al., 2016c,a, 2017b). Each *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 4.1. 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 $\mu_{topic_p,e_i,epoch_{k+1}} = \mu_{topic_p,e_i,epoch_k} + \mathcal{N}(\mu_{drift}, \sigma_{drift})$. 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 4.1: Drift parameters

Performance Measure: We use the same performance measure, overall task accuracy of our multi-expert system, as in previous work in referral networks. 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. Q, the per-instance query budget, is set to 2.

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

4.3.5 Experimental Results

Figure 4.6 and 4.7 compare 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 imperfect.

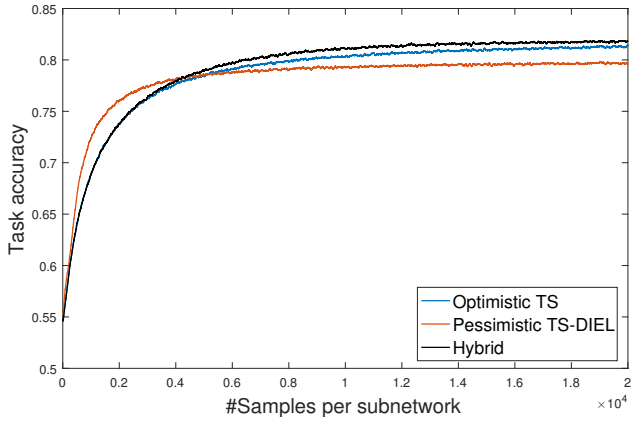
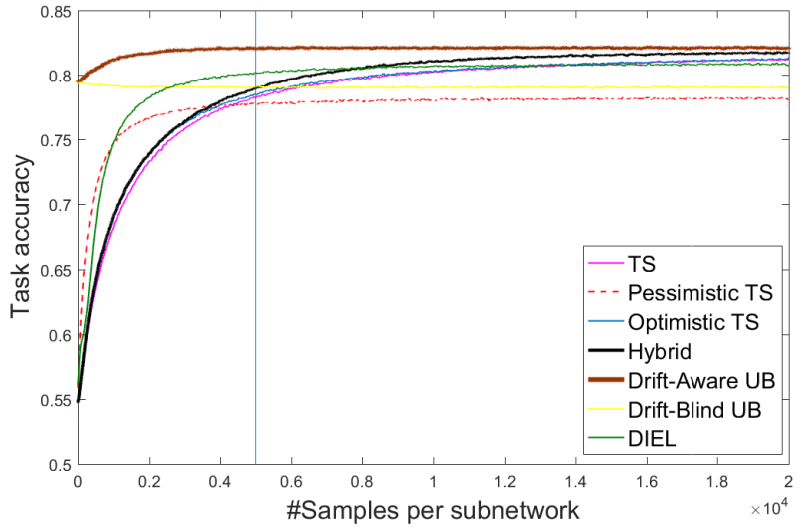
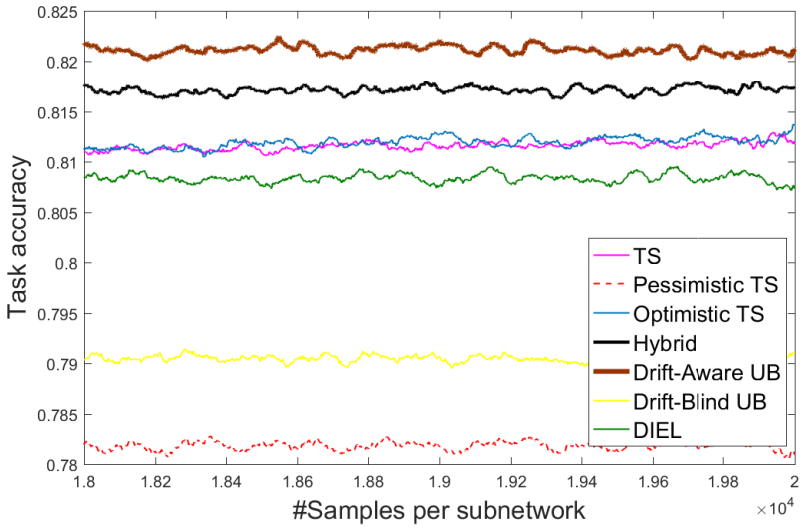


Figure 4.8: Components of `Hybrid`

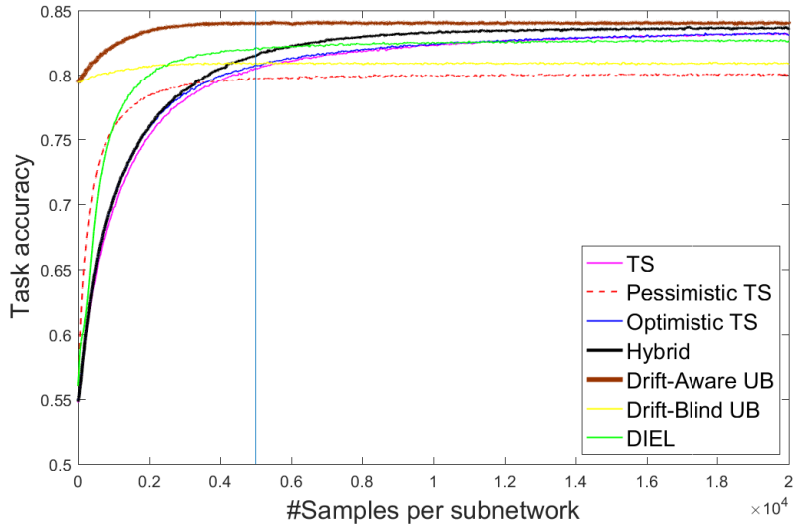


(a) Unbiased, strong drift

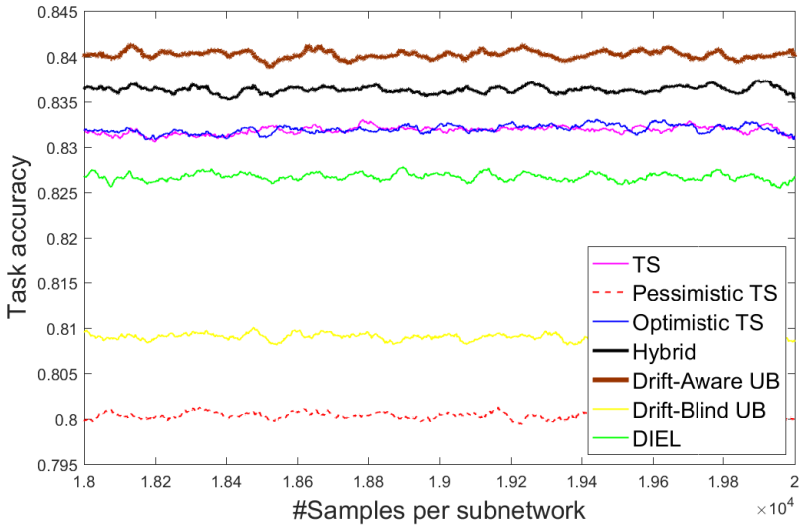


(b) Unbiased, strong drift, steady state

Figure 4.6: Performance comparison of referral learning algorithms



(a) Small positive bias, strong drift



(b) Small positive bias, strong drift, steady state

Figure 4.7: Performance comparison of referral learning algorithms

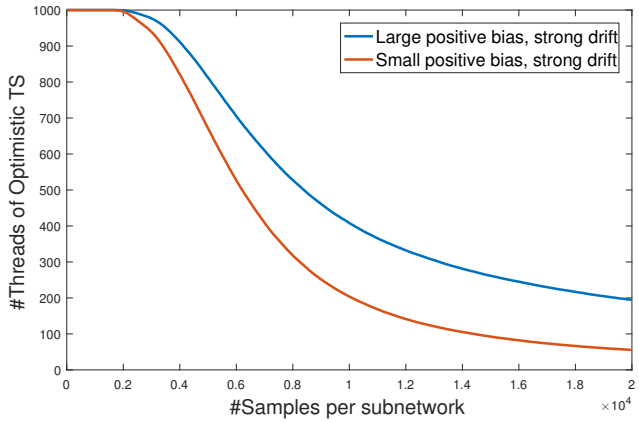


Figure 4.9: Switching behavior of `Hybrid`

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 (KhudaBukhsh et al., 2017b, 2016c) in the stationary expertise setting. In our new results, we find that even 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 than the other two `TS` variants indicating that it is not a viable standalone action selection strategy. However, combining `DIEL` and an effective switching after sufficient exploration proved to be most resilient to time-varying expertise. This result shows that combining components from different action selection strategies could result into high-performance algorithms.

Finally, we focus on `Hybrid`, our primary proposed algorithm. As shown in Figure 4.8, `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 **??**, `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.

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 4.9 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.

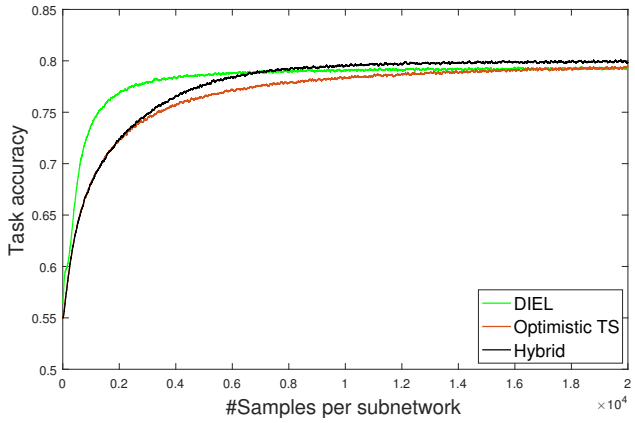


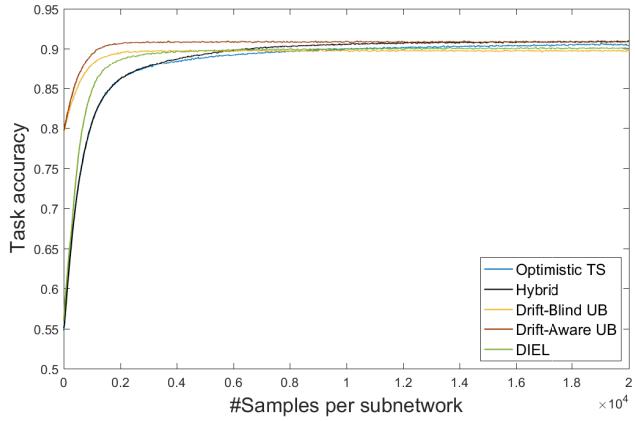
Figure 4.10: Performance comparison with unbiased weak drift

Our results with weak expertise drifts are qualitatively similar. Figure 4.10 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).

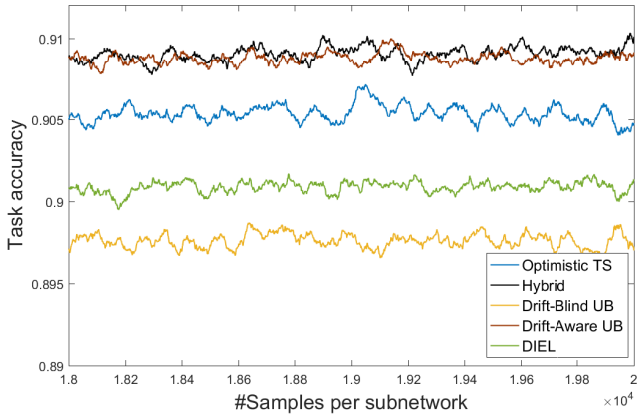
Finally, we present our result with large positive bias, and strong drift in Figure 4.11. The relative ordering of previous performance is preserved with both `DIEL` and `Optimistic TS` outperforming the `Drift-blind` upper bound. However, in this case, we found that the drift-tracking of `Hybrid` is near-perfect as shown in steady-state close-up in Figure 4.11(b), where `Hybrid` is indistinguishable from the drift-aware oracle.

4.3.6 Revisiting the Research Questions

We now take a look at the research questions and present our main takeaways.



(a) Large positive bias, large drift



(b) Steady-state performance closeup: zooming in to 4.11(a) upper right portion

Figure 4.11: Performance comparison of referral learning algorithms with large positive bias, large drift

How to detect a drifting expert? We found that a combination of `Optimistic TS` and `Pessimistic TS-DIEL` performed the best in detecting drifting experts.

How to adapt to the drift, and quickly?

From `Hybrid`'s success, we conclude that a combined strategy of early exploration and subsequent conservative exploitation is a good adaptive strategy. A performance gradient dependent switching point helped `Hybrid` to find a suitable transition point.

Can hybrid algorithms be useful in presence of expertise drift?

Yes. `Hybrid`, a novel combination of `Optimistic TS`, `Pessimistic TS` and `DIEL`, were found to be the most resilient against expertise drift. In fact, in fact we propose a new and structured direction in designing mixed-strategy algorithms in our future work section.

4.4 Additional Robustness Experiments

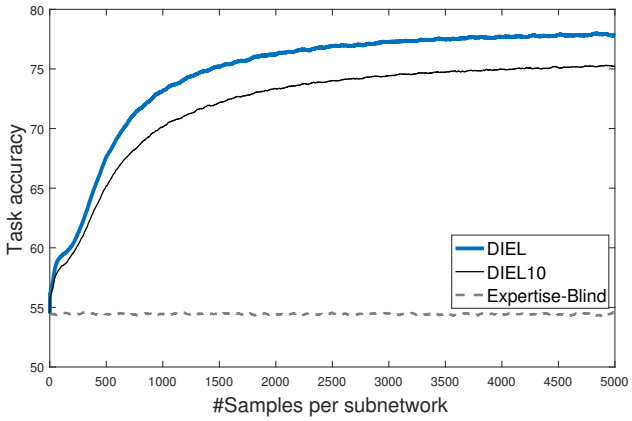
In this section, we describe some additional experiments we conducted considering noisy topic identification, larger networks, and well-known graph generators.

Noisy Topic Identification

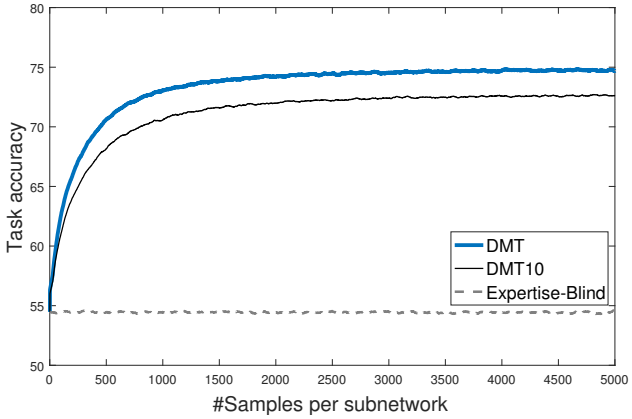
We initially assumed that all experts can always accurately identify the topic of a task. However, for many real tasks identifying the topic could be difficult and prone to error. For instance, in medical diagnosis, similar-looking symptoms may lead to vastly different diseases. In our next set of experiments, we relax the accurate task-topic identification assumption and found that even with a topic misclassification rate of 10%, all algorithms performed substantially better than the expertise-blind baseline. In Figure 4.12, we show results for `DIEL`, `DMT` and `Optimistic TS`; the results for other algorithms are qualitatively similar. Note that, here we assumed that in case of a misclassification, all other topics are equally likely which may not be the case on a practical setting where certain topics can get confused with some specific topics more often than the other.

Network Size and Distribution:

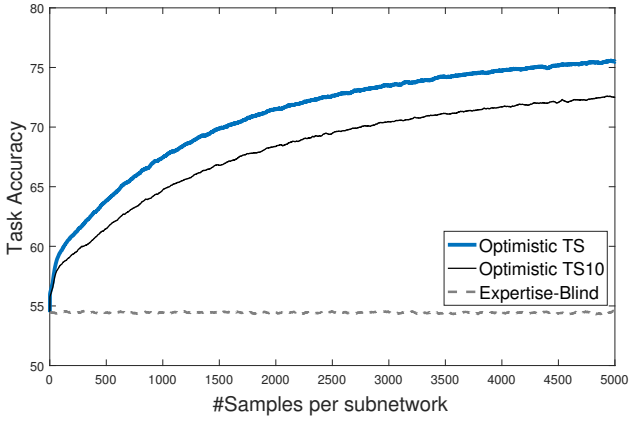
In Chapter 3, we have shown that our referral-learning algorithms perform well even when expertise do not obey any known parameterized distribution. Through an extensive series of experiments, both considering continuous and binary rewards, we have demonstrated a strong performance on a referral network of SAT solvers. In our next set of experiments, we strive to



(a) DIEL



(b) DMT



(c) Optimistic TS

Figure 4.12: Performance of DIEL, DMT and Optimistic TS with topic misclassification

	500	1000	1500	2000	3000	4000	5000
Upper-Bound[SW]	79.48	78.21	78.13	78.81	78.73	78.32	79.75
DIEL[SW]	68.63	**73.10**	**75.10**	**76.57**	**76.42**	**77.11**	**77.23**
DQ[SW]	**70.60**	72.59	73.59	73.99	74.57	75.09	75.69
OTS[SW]	64.34	68.24	70.21	71.50	73.61	74.54	75.09
Expertise-Blind[SW]	54.08	53.93	54.47	54.78	54.64	54.50	54.31
Upper-Bound[POW]	82.48	82.23	82.51	82.55	82.33	81.93	82.35
DIEL[POW]	65.41	**72.80**	**76.77**	**77.86**	**79.55**	**80.28**	**80.14**
DQ[POW]	**70.99**	73.51	74.91	75.61	76.89	76.96	77.68
OTS[POW]	63.21	66.61	69.02	71.33	73.59	75.31	76.06
Expertise-Blind[POW]	54.84	54.31	54.88	53.78	54.23	54.78	54.89
Upper-Bound[PREF]	82.10	82.36	82.21	82.29	82.37	82.52	82.42
DIEL[PREF]	64.20	**73.20**	**75.75**	**77.74**	**79.48**	**80.33**	**80.56**
DQ[PREF]	**71.06**	73.79	75.09	75.66	76.86	77.60	78.19
OTS[PREF]	62.82	66.52	69.09	71.32	73.79	75.02	76.67
Expertise-Blind[PREF]	54.96	54.36	54.08	54.00	54.65	53.85	54.17

Table 4.2: Performance comparison of referral algorithms

test the robustness to network topologies while keeping the expertise assumptions unchanged (e.g., topical expertise belongs to mixture of Gaussians).

Although we had already shown a measure of robustness to different network topologies, we investigated a few additional typical network types, while holding the expertise assumptions constant: constructing our referral network by means of well-known random graph generators (Watts and Strogatz, 1998; Barabási and Albert, 1999; Holme and Kim, 2002), we found that there was no overall qualitative performance change. We denote the data sets as PREF (preferential graphs), POW (generated with power law distribution) and SW (*small world graphs* known to model collaboration networks). We focus on the top three algorithms in our single-hop referral setting (see, Table 3.4): DIEL, DQ and Optimistic TS. We also include the Expertise-Blind and Upper-Bound baselines to compare and contrast the learning performance.

Table 4.2 reports the performance of our referral-learning algorithms on referral networks constructed by well-known random graph generators at different points in the horizon. For a given network distribution and a specific point in the horizon, the best performance (excluding the Upper Bound) is highlighted in bold. In all three network distributions, DIEL outperformed Optimistic TS and DQ highlighting DIEL's robustness to network topologies. Additionally, we found that all three referral-learning algorithms performed substantially better than the Expertise-Blind baseline and the results were consistent with our previous experiments.

We also ran experiments on a data set of referral networks of 1000 experts with similar subnetwork size. Since our learning is distributed, with each expert learning its own referral policies, we found that our algorithms scaled linearly with the number of the experts and there was no significant change in the results.

Chapter 5

Proactive Skill Posting

In all our experiments so far, we assumed an uninformative prior on the expertise of colleagues. We have identified a set of algorithms suitable for the *learning to refer* challenge even in presence of capacity constraints, evolving networks or expertise drift. However, experts starting with no information whatsoever about her colleagues may not correspond to a real-world setting. In real life, we often see that experts clearly mention which type of tasks they are particularly good at and also often forge links to their colleagues via social networks. Such information can particularly help in the early phase of learning referrals. Gradually, with further observations, colleagues may re-estimate their beliefs of expertise levels based on actual performance. In this chapter, we introduce proactive skill posting, an augmented learning setting with a one time local-network advertisement of expertise-by-topic by each expert in the network. In real life, such advertisement will have a budget; it is infeasible for every expert to inform every colleague an exhaustive list of their topical expertise. Also, it is unlikely that an expert will have close-to-accurate estimates of her skills on her weaker topics. Hence, our proactive skill posting assumes advertisement of a subset of strong skills. However, even in one's true expertise area, estimating self-skill could be noisy (see, e.g., MacKay et al. (2014)). In this chapter, we present proactive versions of referral-learning algorithms that take advantage of advertised noisy priors and address the cold-start problem.

There are two aspects to the cold-start problems in learning distributed referral networks. The first is when a new expert joins the referral network, other experts may not know that person's skill set and therefore are not in a good position to refer any problem(s) his or her way. The second is the dual problem of that new expert knowing few, if any, skills of the established expert, and therefore not knowing to whom he or she may refer. We propose to address both the inward and outward cold start problems via proactive skill posting. However, the success in the extended learning setting depends on a *truthful mechanism* to elicit the true skills of the experts

in the network since the experts, as selfish agents, try to maximize the number of tasks they receive to maximize fees. Also, as already mentioned, well-meaning experts may overestimate (or underestimate) their skills. So exploiting the advertised prior would require a mechanism to both elicit true expertise information and at the same time, being tolerant to noisy self-skill estimates.

The rest of the chapter is organized as follows. We first present the research questions and their associated challenges and benefits. Next, we present the preliminaries specific to our augmented learning setting after which, we explore the impact of prior availabilities in two algorithms, `DIEL` and `DMT`, with mean-based exploitation. Following the preliminary study, we describe the two key components: effective initialization to incorporate advertised priors and a reward-penalty mechanism to discourage strategic lying, of the proactive versions of `DIEL`, `Q-Learning`, and ϵ-`Greedy`. We summarize our empirical findings and conclude with revisiting the research questions.

5.1 Research Questions

In this chapter, we are primarily concerned with the following research questions:

Does access to (noisy) priors on their colleagues' expertise improve an expert's referral performance? The key challenge is initializing the algorithms with the noisy priors in a way such that the search for an effective referral choice becomes biased towards stronger experts while ensuring the weaker ones also get enough exploration.

Does access to (noisy) priors on a subset of topics improve an expert's referral performance? It is unlikely that all experts will have accurate or close-to-accurate estimates on their own skills across all topics. Also, it is infeasible for experts to inform about every skill to every connected expert. A more realistic scenario is an expert advertises some of her top skills to her colleagues. The key is to use the available information present in the *explicit bids* to set an upper-bound on the *implicit bids*.

How to design proactive skill posting algorithms that discourage strategic lying to attract more business? The challenge of using partially available prior is two-fold. On one hand we would strive to design algorithms robust to noise in self-skill estimates. On the other hand, we would like to prevent experts getting more business through strategic lying, i.e., claiming they are stronger than they actually are.

How extensible are the proactive skill posting techniques? Since different algorithms put different emphasis on exploration and exploitation, the same initialization and reward-penalty mechanism may not work for all algorithms.

5.2 Preliminaries

Advertising unit: a tuple $\langle e_i, e_j, t_k, \mu_{t_k} \rangle$, where e_i is the *target expert*, e_j is the *advertising expert*, t_k is the topic and μ_{t_k} is e_j's (advertised) topical expertise. Similar to rewards in our uninformative prior setting, an advertising unit is also locally visible, i.e., only the *target expert* gets to see the advertised prior for a given unit.

Advertising budget: In practice, experts have limited time to socialize with different colleagues and get to know each other's experience. We incorporate this notion through the notion of budget and assume each expert is allocated a budget of B advertising units, where B is twice the size of that expert's subnetwork. Effectively means that each expert reports her top two skills to everyone in her subnetwork.

Advertising protocol: a one-time advertisement that happens right at the beginning of the simulation or when an expert joins the network. The advertising expert e_j reports to each target expert e_i in her subnetwork the two tuples $\langle e_i, e_j, t_{best}, \mu_{t_{best}} \rangle$ and $\langle e_i, e_j, t_{secondBest}, \mu_{t_{secondBest}} \rangle$, i.e., the top two topics in terms of the advertising expert's topic means.

Explicit bid: A topic that is advertised in the above advertising protocol.

Implicit bid: A topic that is not advertised, for which an upper bound can be assumed.

5.3 Impact of Informative Prior

To set the stage for proactive skill posting, where experts have a noisy estimate of a subset of their skills which they advertise to their colleagues, we first examine to which extent informative priors on the means can be incorporated into referral algorithms. For this, we considered two algorthims: `DIEL` and `DMT`, in which empirical mean is the exploitation component. In Figure 5.1, we contrast the performance of `DIEL` and `DMT` in an uninformed prior setting versus an informed prior setting where experts have access to a noisy oracle.

Specifically, our experimental design is the following. Every expert has access to a noisy oracle than can estimate the true topic-mean of every other expert-topic pair within an error bound, i.e. $|\mu_{e_i,t_k} - \hat{\mu}_{e_i,t_k}| \leq \delta$ where $\hat{\mu}_{e_i,t_k}$ is the estimated topic-mean of expert e_i on topic t_k. Unlike our previous initialization scheme, instead of setting all reward vectors (0,1), for all experts, *reward*(e_i, t_k, e_j) are initialized with two rewards of $\hat{\mu}_{e_i,t_k}$. The initialization of two rewards is essentially done to set the initial variance of `DIEL` to zero. The performance of `DMT`

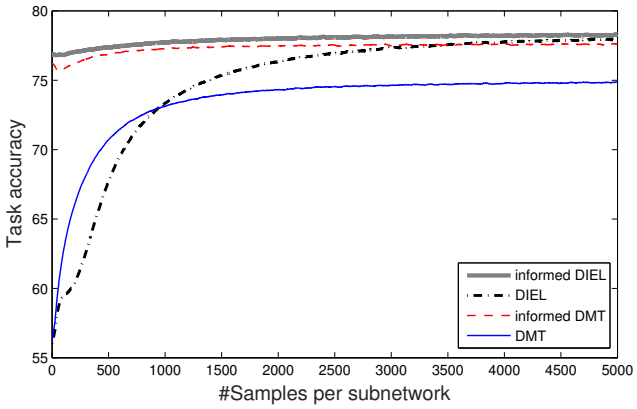


Figure 5.1: DIEL and DMT with informative prior

would remain unchanged even if we initialized with a single reward. We chose the same for DMT primarily for consistency.

In figure 5.1, we see that even when δ is as high as 0.2, the performance of both DIEL and DMT substantially improved; and the cold-start problem was duly addressed as the early-learning phase advantage was considerably large. An interesting side-observation is that, an uninformed DIEL still managed to outperform an informed DMT. This result highlights again that DIEL's variance-based exploration component creates a strong impact in learning.

5.4 Proactive Skill Posting

In this section, we outline the design modifications on three algorithms, DIEL, ϵ-Greedy, and Q-Learning, that led to their corresponding proactive versions. Any proactive algorithm differs from its non-proactive counterpart in two areas: initialization and reward mechanism. The goal of initialization is to make the algorithms' search more biased towards stronger experts. The goal of reward-penalty mechanism is to ensure no expert can get more business by overstating their priors.

5.4.1 Initialization

proactive-DIEL

Rather than DIEL sets *reward*(e_i, t_k, e_j) for each i, j and k with a pair $(0, 1)$ in order to initialize mean and variance, proactive-DIEL initializes *reward*(e_i, t_k, e_j) for each advertisement unit $\langle e_i, e_j, t_k, \mu_{t_k} \rangle$ with two rewards of μ_{t_k} (explicit bid).

To initialize topics for which no advertisement units are available (implicit bid), we initialize the rewards as if the expert's skill was the same as on her second best topic, that is, with two rewards of $\mu_{t_{secondBest}}$, effectively being an upper bound on the actual value.

proactive-ϵ-Greedy

Recall that, similar to DIEL, ϵ-Greedy was also initialized with a reward-pair of $(0, 1)$. Proactive-ϵ-Greedy is initialized the same as proactive-DIEL; with two rewards of μ_{t_k} for the explicit bids and $\mu_{t_{secondBest}}$ acting as an upper-bound for the implicit bids.

proactive-Q-Learning

Proactive-Q-Learning uses the same initialization and a similar technique to bound unknown priors with reported second-best skills as proactive-DIEL and proactive-ϵ-Greedy. Instead of randomly initializing the Q-Function as in Q-Learning, the Q-function for each action is initialized with its advertised mean or corresponding $\mu_{t_{secondBest}}$ in absence of such advertisement unit.

5.4.2 Reward Update Function

Instead of just observing the reward r, and appending it to the corresponding reward vector and update necessary data structures (e.g., as in Thompson Sampling, maintaining the number of observed successes and failures for a given expert-topic pair), in proactive skill posting setting, the reward update function plays a dual role. The first role is same as in the uninformed prior setting, keeping track of the historical performance of colleagues. The second role is non-trivial; penalizing experts for misstating their priors.

We have explored two different approaches to penalize misreporting experts. The first approach is simpler (KhudaBukhsh et al., 2016a), penalizing experts when they fail to solve an instance. The second approach estimates distrust (KhudaBukhsh et al., 2017a), and penalizes regardless of success and failure.

In our results section, we compare and contrast these two mechanisms on four different aspects: steady-state performance gain, tolerance to noisy self-skill estimates, incentive compatibility and extensibility.

5.4.3 Penalty on Failure

We call the first scheme as *Penalty on Failure*. Using this approach, we obtained two proactive algorithms: proactive-DIEL and proactive-ϵ-Greedy.

When a referred expert e_j succeeds in solving a task on topic t_k, *update* in proactive-DIEL's *update* function, like DIEL's, assigns an additional reward of 1 to *reward*(e_i, t_k, e_j). When e_j fails, however, instead of always appending a 0 to *reward*(e_i, t_k, e_j), proactive-DIEL, in the presence of an advertisement unit $\langle e_i, e_j, t_k, \mu_{t_k} \rangle$, appends a (negative) penalty P with probability μ_{t_k}. This way, over-reporting of skill leads to more frequent incurrence of the penalty. In the absence of an advertisement unit (i.e., for implicit bids), a penalty P is still appended, but with a probability equal to the sample mean of e_j observed by e_i on topic t_k. In our experiments, we set P to -0.35.

Algorithm 16: Penalty on Failure

if *referredExpert* succeeds **then**
  penaltyProbability $\leftarrow 0$
else
  if *topic t* is *explicitBid* **then**
    penaltyProbability $\leftarrow \mu_{advertised}$
  else
    penaltyProbability $\leftarrow \hat{\mu}_{observed}$
  end
end

Proactive-ϵ-Greedy was adapted essentially the same way as proactive-DIEL, the only minor difference being that a failed task does not receive a penalty if it was a diversification step.

5.4.4 Penalty on Distrust

We denote this approach as *Penalty on Distrust*. We obtained three proactive algorithms using this approach: proactive-DIEL$_t$, proactive-ϵ-Greedy$_t$ and proactive-Q-Learning$_t$[1].

In this approach, our penalty incorporates a factor we may call *distrust*, as it estimates a likelihood the expert is lying, given our current observations:

$penalty = C_2\ distrust$, where

distrust = *distrustFactor*$_1$ + *distrustFactor*$_2$;

distrustFactor$_1$ = $|\mu_{t_{best}} - \hat{\mu}_{t_{best}}|\ \zeta(n_{t_{best}})$ and,

distrustFactor$_2$ = $|\mu_{t_{seconBest}} - \hat{\mu}_{t_{secondBest}}|\ \zeta(n_{t_{secondBest}})$

[1]The subscript t stands for trust.

where $\zeta(n_t) = \frac{n_t}{n_t+C_1}$, a factor intended to gradually ramp up to 1 towards steady state, where n_t is the number of observations for topic t. Basically, *distrustFactor*$_1$ and *distrustFactor*$_2$ estimate how much the advertised skill posting is off from the estimated mean, for the best skill and second-best skill respectively (i.e., the explicit bids). C_1 and C_2 are the two configurable parameters of this mechanism. Intuitively, larger the value of C_2, greater is the discouragement for strategic lying; larger the value of C_1, the slower grows the distrust. In all our experiments, C_1 was set to 50. C_2 was set to 1, 2 and 3 for proactive-$\texttt{DIEL}_t$, proactive-ϵ-$\texttt{Greedy}_t$, and proactive-$\texttt{Q-Learning}_t$, respectively.

The distrust-based penalty mechanism differs from the previous approach in that all tasks receive a penalty regardless of whether the referred expert solves it or not. Second, the two mechanisms penalize the extent of misreporting in different ways, as the previous method fails to penalize underbidding. We can show a simple two-expert subnetwork to illustrate how underbidding could be used to attract more business in the earlier scheme. Consider two experts, e_1 and e_2, have identical expertise (1 - ϵ, $\epsilon \leq 0.5$) across all tasks. e_1 reports truthfully while e_1 underbids and advertises (1 - 2ϵ). For a penalty of r ($r > 0$), the expected mean reward for e_1 will be (1 - ϵ) - ϵ (1 - ϵ) r. Due to underbidding, e_2 will have an unfair advantage over e_1 as her expected mean reward will be (1 - ϵ) - ϵ (1 - 2ϵ) r, larger than e_1.

We now provide proof sketches demonstrating Bayesian-Nash incentive compatibility in the limit for our distrust-based mechanism, *Penalty on Distrust*.

Theorem 1 *Under the assumption that all actions are visited infinitely often, in the limit, strategic lying is not beneficial in proactive-*$\texttt{Q-Learning}_t$.

We give a proof sketch by showing that a lying expert will have a non-zero penalty in the limit.

$$\lim_{n\to\infty} \hat{\mu}_{t_{best}} = \mu_{t_{best}} \tag{5.1}$$

$$\lim_{n\to\infty} \hat{\mu}_{t_{secondBest}} = \mu_{t_{secondBest}} \tag{5.2}$$

$$\lim_{n\to\infty} \zeta(n) = 1 \tag{5.3}$$

Hence, for a truthful expert both $distrust$ and $penalty$ approach zero in the limit. However, for a lying expert at least one of the estimates ($distrustFactor_1$ or $distrustFactor_2$) is off by a positive constant c. Hence, in the limit, $distrust \geq c$ and $penalty \geq C_2 c$, therefore a truthful expert will always receive more reward than if she lies and since $\texttt{Q-Learning}$ considers a

Reward update function	Algorithm	Parameters
Penalty on Failure	proactive-DIEL (KhudaBukhsh et al., 2016a)	P
Penalty on Failure	proactive-ϵ-Greedy (KhudaBukhsh et al., 2016b)	α, P
Penalty on Distrust	proactive-$\texttt{DIEL}_t$ (KhudaBukhsh et al., 2017a)	C_1, C_2
Penalty on Distrust	proactive-ϵ-$\texttt{Greedy}_t$ (KhudaBukhsh et al., 2017a)	α, C_1, C_2
Penalty on Distrust	proactive-$\texttt{Q-Learning}_t$ (KhudaBukhsh et al., 2017a)	α, γ, ϵ, C_1, C_2

Table 5.1: Proactive referral algorithms

discounted sum of rewards, eventually, a truthful expert will have a larger Q-value than if she lies. Ergo, strategic lying is not beneficial when all other experts are truthful.

Theorem 2 *Under the assumption that all actions are visited infinitely often, in the limit, strategic lying is not beneficial in proactive-ϵ-$\texttt{Greedy}_t$.*

The proof is essentially the same as the previous proof.

Theorem 3 *Under the assumption that all actions are visited infinitely often, in the limit, strategic lying is not beneficial in proactive-$\texttt{DIEL}_t$.*

In our previous proof, we already showed that in the limit, a lying expert will always receive a higher penalty than a truthful expert which will effectively lower the reward mean.

For any reward sequence $r_1, r_2, \ldots, r_n$, and a penalty sequence $p_1, p_2, \ldots, p_n$,
$-max(p_1, p_2, \ldots, p_n) \leq r_i \leq 1 - min(p_1, p_2, \ldots, p_n)$,
$1 \leq i \leq n$.
Now, $distrust \leq 2$. Hence, $0 \leq p_i \leq 2C_2$, $1 \leq i \leq n$.
Hence, $-2C_2 \leq r_i \leq 1$, i.e., all rewards are finite and bounded. This means, in the limit, the variance of the reward sequence is finite and bounded. Hence,

$$\lim_{n\to\infty} UI(a) = \lim_{n\to\infty} (m(a) + \frac{s(a)}{\sqrt{n}}) = m(a) \tag{5.4}$$

This means, in the limit, the reward for DIEL will be dominated by its mean reward. Since a lying expert will always incur higher penalty than a truthful expert, an expert will have a higher reward mean when it behaves truthfully.

Unlike the Q-learning variants and ϵ-Greedy algorithms, there is no guarantee for DIEL that all actions are visited infinitely often, although a variant can guarantee that condition with random visits at ϵ probability, and perform similarly in the finite case for small enough ϵ.

5.5 Experimental Setup

Referral algorithms and baselines: We have proposed five proactive algorithms listed in Table 5.1. Our baselines are the corresponding non-proactive versions.

Data set: As our synthetic data set, we used the same 1000 scenarios used in all our experiments in the uninformed prior setting (except the data set where expertise drift occurs). Each scenario consists of 100 experts connected through a referral network and 10 topics. For our experiments involving SAT solvers, we used the same 100 `SATenstein` (version 2.0) solvers obtained from the experiments presented in KhudaBukhsh et al. (2016d) as experts as in our previous experiments.

Algorithm configuration: The version of `DIEL` we used is parameter free. The remaining parameterized algorithms are configured by selecting 100 random instantiations of each algorithm and running them on a small background data set (generated with the same distributional parameters as our evaluation set). We selected the parameter configuration with the best performance on the background data.

Performance measure: Following our previous experiments on uninformed prior setting, we used overall task accuracy as our performance measure. In order to empirically evaluate Bayesian-Nash incentive compatibility, we followed the same experimental protocol followed in KhudaBukhsh et al. (2016a) (described in Chapter 5.6.2).

Computational environment: Experiments on synthetic data were carried out on Matlab R2016 running Windows 10. Experiments on SAT solver referral networks were carried out on a cluster of dual-core 2.4 GHz machines with 3 MB cache and 32 GB RAM running Linux 2.6.

5.6 Results

Before presenting our results in further detail, here, we first list our **key findings**:

- A comparative analysis of the proactive referral algorithms on our synthetic set shows that all proactive versions of the referral algorithms we proposed beat the original versions under the condition of truthful reporting of skills.
- All proactive referral algorithms are robust to noisy self-skill-estimates.
- All proactive referral algorithms demonstrate strong empirical evidence of being Bayesian-Nash incentive compatible.
- Performance results on the SAT Solver application indicate that the main conclusions derived from synthetic data, hold for more realistic data as well.
- The new algorithms are robust to dynamic changes in evolving networks.

$\mu_{t_{best}}$	$\mu_{t_{secondBest}}$	proactive DIEL	proactive $\texttt{DIEL}_t$	proactive ϵGreedy	proactive $\epsilon\texttt{Greedy}_t$	proactive $\texttt{Q-Learning}_t$
Truthful	Truthful	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Truthful	Overbid	0.99	**1.02**	0.99	**1.03**	0.97
Overbid	Truthful	**1.00**	**1.19**	0.98	**1.24**	**1.35**
Overbid	Overbid	0.97	**1.25**	0.98	**1.36**	**1.39**
Truthful	Underbid	**1.04**	**1.15**	**1.00**	**1.08**	**1.21**
Underbid	Truthful	**1.09**	**1.16**	**1.06**	**1.10**	**1.17**
Underbid	Underbid	**1.22**	**1.32**	**1.12**	**1.24**	**1.56**
Underbid	Overbid	**1.11**	**1.15**	**1.09**	**1.09**	**1.14**
Overbid	Underbid	**1.04**	**1.50**	**1.04**	**1.34**	**1.63**

Table 5.2: Comparative study on empirical evaluation of Bayesian-Nash incentive-compatibility. Strategies where being truthful is no worse than being dishonest are highlighted in bold.

5.6.1 Overall Performance Gain

Figure 5.2 compares the performance of the proactive algorithms with their non-proactive versions under the assumption of truthful reporting and accurate self-skill estimates. The two main aspects of note are performance in the early learning phase, and steady state performance. We first observe that, as expected, all new proactive algorithms did better than their non-proactive counterparts, both in steady state and during the early phase of learning, while noting that the gap between DIEL and its proactive versions was less than the corresponding difference for the other two algorithms. We also obtained a modest performance gain over the *penalty on failure* mechanism and both proactive-$\texttt{DIEL}_t$ and proactive-ϵ-$\texttt{Greedy}_t$ did slightly better than the algorithms obtained using *penalty on failure*.

5.6.2 Discouraging Strategic Lying

So far, we have shown that our proposed proactive referral algorithms address the cold start problem better than their non-proactive counterparts and are immune to a small amount of Gaussian noise in self-skill estimates. Here, we strive to deal with the case of deliberate (strategic) misreporting, e.g. experts trying to get more business by overstating (or counter-intuitively, understating) their skills. Note that, since a noisy bid can be interpreted as deliberate misreporting and vice-versa, robustness to noisy self-skill estimates and robustness to strategic lying are two orthogonal goals.

Since proving incentive compatibility in a multi-expert distributed learning setting is a chal-

lenging task, we treat the number of referrals received as a proxy for payment and empirically analyze Bayesian-Nash incentive compatibility the following way. Since we are interested in knowing if their exists any specific strategy combination (e.g., truthfully report best-skill but overbid second-best skill) that could fetch more referrals, we consider all possible such combinations (listed in Table 5.2). For a given strategy s and scenario $scenario_i$, we first fix one expert, say e_l^i. Let *truthfulReferrals*(e_l^i) denote the number of referrals received by e_l^i beyond a steady-state threshold (i.e., a referral gets counted if the initial expert has referred 1000 or more instances to her subnetwork) when e_l^i and all other experts report truthfully. Similarly, let *strategicReferrals*(e_l^i) denote the number of referrals received by e_l^i beyond a steady-state threshold when e_l^i misreports while everyone else advertises truthfully. We then compute the following Incentive Compatibility factor (*ICFactor*) as :

$$\textit{ICFactor} = \frac{\sum_{i=1}^{1000} \textit{truthfulReferrals}(e_l^i)}{\sum_{i=1}^{1000} \textit{strategicReferrals}(e_l^i)} .$$

A value greater than 1 implies truthful reporting fetched more referrals than strategic lying.

Table 5.2 presents the *ICFactor*s for each algorithm and each strategy combination. We see that, beyond the steady-state threshold, strategic misreporting is hardly beneficial and in fact counterproductive in most cases. Proactive-$\texttt{DIEL}_t$ was (slightly but consistently) better at discouraging each strategy combination than proactive-`DIEL`. The only case truthful advertising fetched slightly fewer referrals for proactive-$\texttt{Q-Learning}_t$ is when an expert truthfully reports her top skill but overbids her second-best skill (in fact a hard case for all the algorithms). This is likely the result of the way the posted second-best skill is used to bound implicit bids. However, on doubling the horizon (i.e., considering 10,000 samples per subnetwork), we found that proactive-$\texttt{Q-Learning}_t$'s *ICFactor* improved to 1.04.

5.6.3 Robustness To Noisy Skill Estimates

So far, we have shown that our proposed proactive referral algorithms address the cold start problem better than their non-proactive counterparts and provide stronger discouragement to strategic lying. However, even when experts post their skills truthfully, their self-estimates may not be precise (see, e.g., MacKay et al. (2014)). Imprecise skill estimation in proactive skill posting was first explored in (KhudaBukhsh et al., 2016a,b). Note that, since a noisy bid can be interpreted as deliberate misreporting and vice-versa, robustness to noisy self-skill estimates and robustness to strategic lying are two major goals and there lies an inherent trade-off between them.

We assume Gaussian noise on the estimates in the form of $\hat{\mu} = \mu + \mathcal{N}(0, \sigma_{noise})$, where $\hat{\mu}$

is an expert's own estimate of her true topic-mean μ, and σ_{noise} is a small constant (0.05 or 0.1 in our experiments). Figure 5.3 compares the performance of the proactive referral algorithms with noisy estimates with the noise-free case and their non-proactive versions. Resilience to the noise depends on the algorithm. In proactive-$\texttt{DIEL}_t$, a small amount of noise (0.05) degrades the steady-state performance, but retains a small advantage over the non-proactive version. While both versions of noisy proactive-$\texttt{DIEL}_t$ do substantially better in the early-learning phase, there is no steady-state performance gain in the presence of larger noise. Proactive-ϵ-$\texttt{Greedy}_t$ was the most resilient (not shown in the figure): even with a larger noise value, it kept a significant lead over the non-proactive version even in the steady state (task accuracy: 77.33% (σ_{noise} = 0.1), 76.76% (σ_{noise} = 0.05), and 75.26% for the non-proactive version). Proactive-$\texttt{Q-Learning}_t$ was the most sensitive: with smaller noise value, the early-learning-phase gain disappears again in the steady state; with higher noise value, proactive skill posting became counter-productive.

5.6.4 Evolving Networks

In our earlier experiments, we found that proactive-DIEL was resilient to small amount of distributed network change (5% network change) and a large amount of one-time network change (20% network change). Here, we revisit the problem of evolving networks analyzed in Chapter 4.2 but unlike our previous experiments, we consider more severe extents of distributed network change. We have already seen that a primary benefit of proactive methods is that they address the cold-start problem. Rapid improvement in the early learning phase is perhaps even more important for evolving networks. Figure 5.4 presents an extreme case of 20% network change at regular interval. We found that the proactive-DIEL handled the network changes much better than the original DIEL even in presence of noise in self-skill estimates.

5.6.5 SAT Solver Referral Network

So far, we have presented results on our synthetic data with well-behaved and predetermined distributions for expertise as well as noise on the self-skill estimates. As in (KhudaBukhsh et al., 2016b), we also ran several experiments on a referral network of high-performance Stochastic Local Search (SLS) solvers, a more realistic situation in which expertise or noise in self-skill estimates do not obey known parameterized distributions. Our experts are 100 SATenstein solvers with varying expertise on six SAT distributions (map to topics). We ran experiments on 10 randomly chosen referral networks from our synthetic data set. In order to save computational cycles, in these experiments, we only focus on the referral behavior. This explains why our choice of horizon is smaller (also, the number of topics is less than the synthetic data set). On a given

SAT instance, the referred SATenstein solver is run with a cutoff time of 1 CPU second. A solved instance (a satisfying model is found) fetches a reward of 1, a failed instance (timeout) fetches a reward of 0.

Figure 5.5 compares the performance of proactive and non-proactive algorithms on this data set. Figure 5.5(a) shows that proactive-$\texttt{DIEL}_t$ retains the early-learning phase advantage over DIEL, but the slight performance gain in steady state is missing. On the other hand, Figure 5.5(b) shows qualitatively similar behavior as the synthetic data set: throughout the learning phase, proactive-ϵ-$\texttt{Greedy}_t$ maintained a modest lead over its non-proactive version.

5.7 Revisting the Research Questions

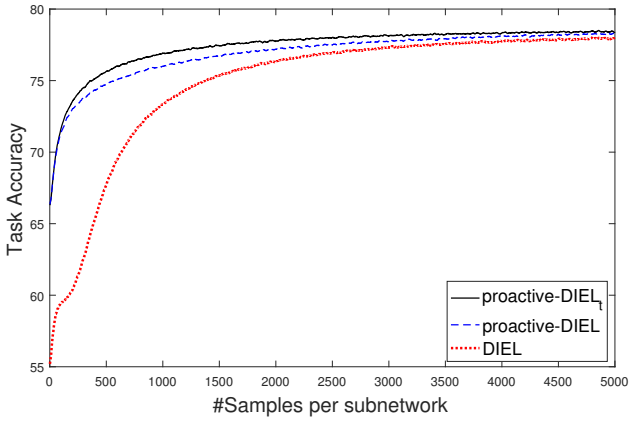
In this chapter, we are primarily concerned with the following research questions:

Does access to (noisy) priors on their colleagues' expertise improve an expert's referral performance? Yes, the performance of DIEL and DMT both improved substantially in presence of noisy priors.

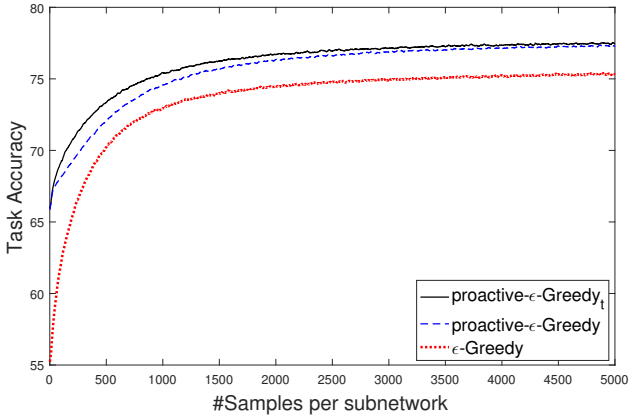
Does access to (noisy) priors on a subset of topics improve an expert's referral performance? Yes, we have presented five proactive algorithms that use initialization techniques to bound the *implicit bids* and thus perform better than their non-proactive counterparts.

How to design proactive skill posting algorithms that discourage strategic lying to attract more business? We have presented two penalty mechanisms: *Penalty on failure* and *Penalty on distrust*, to discourage strategic lying.

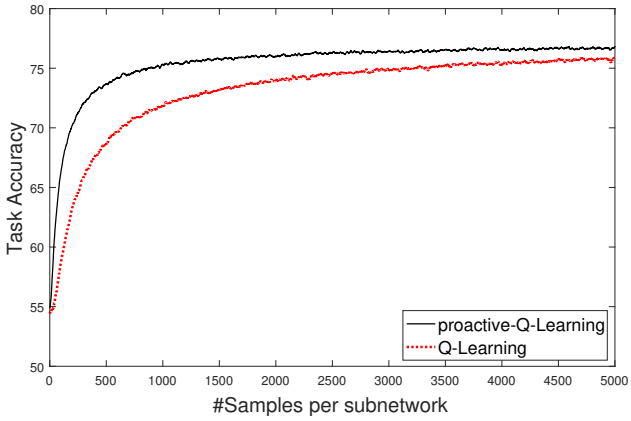
How extensible are the proactive skill posting techniques? Out of the five categories of referral-learning algorithms we studied, we successfully proposed proactive variants belonging to three of them.



(a) Proactive-`DIEL`

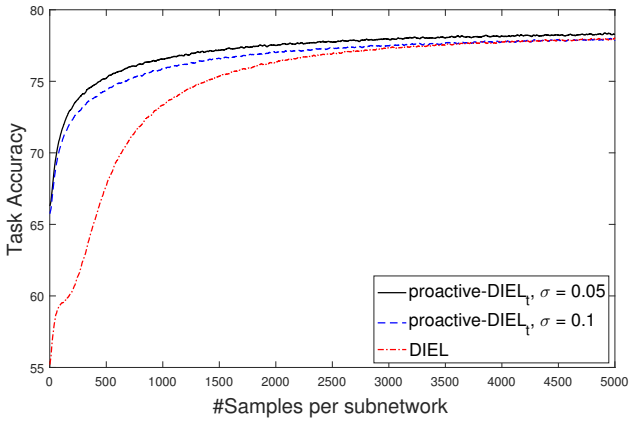


(b) Proactive-ϵ-`Greedy`.

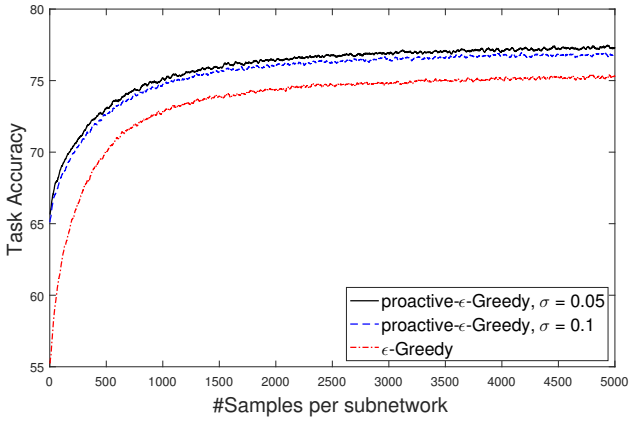


(c) Proactive-`Q-Learning`.

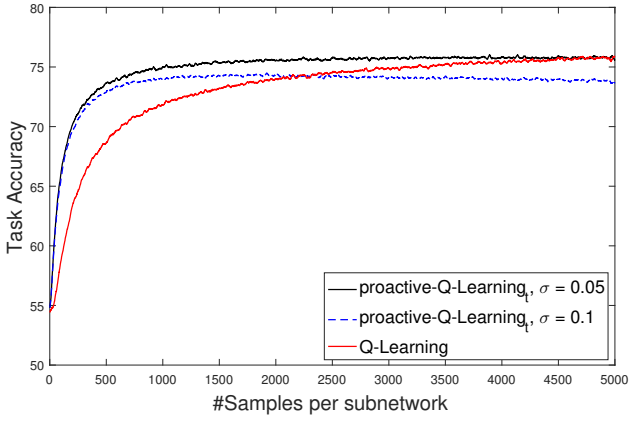
Figure 5.2: Performance comparison of proactive algorithms and corresponding non-proactive versions



(a) Noise tolerance of proactive-$\texttt{DIEL}_t$

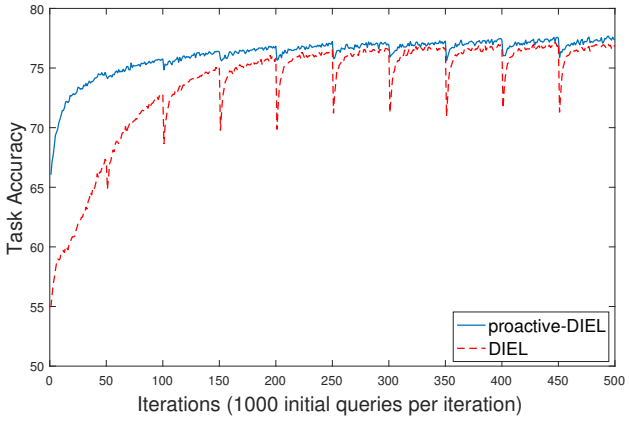


(b) Noise tolerance of proactive-ϵ-$\texttt{Greedy}_t$

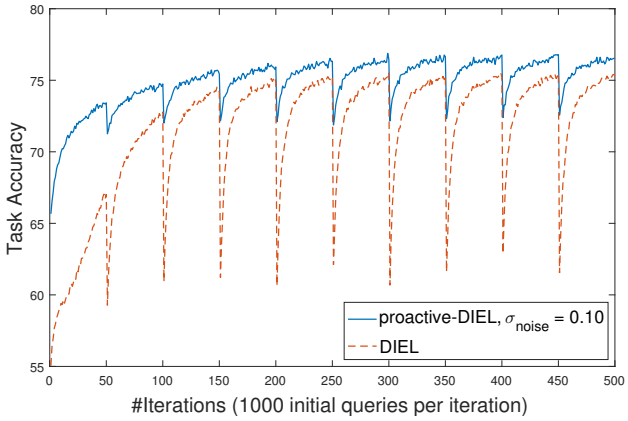


(c) Noise tolerance of proactive-$\texttt{Q-Learning}_t$

Figure 5.3: Robustness to noisy skill estimates

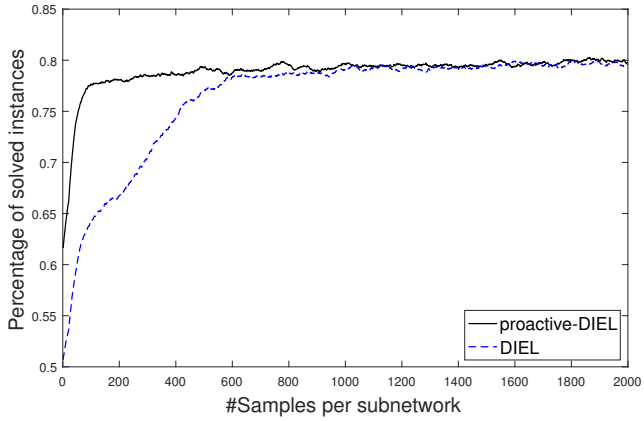


(a) 5% network change

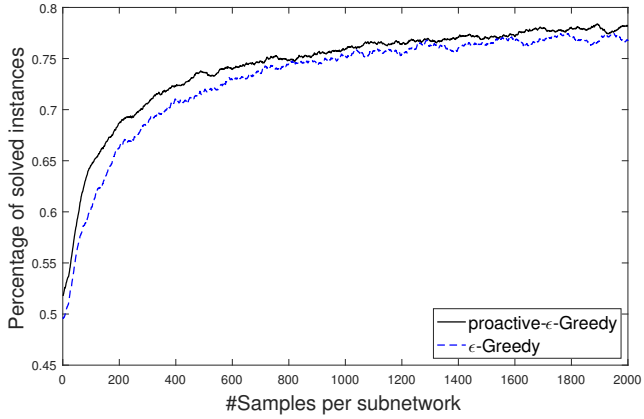


(b) 20% network change and noisy self-skill estimates

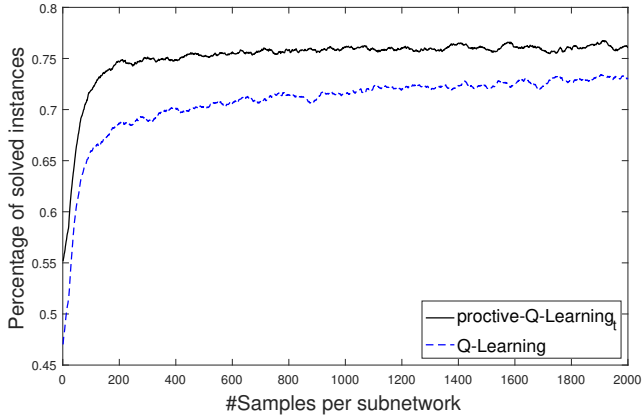
Figure 5.4: Proactive-DIEL on dynamic networks.



(a) Performance comparison between `DIEL` and proactive-$\texttt{DIEL}_t$



(b) Performance comparison between ϵ-`Greedy` and proactive-ϵ-$\texttt{Greedy}_t$



(c) Performance comparison between `Q-Learning` and proactive-$\texttt{Q-Learning}_t$

Figure 5.5: Performance comparison on SAT solver referral networks

Chapter 6

Conclusions and Future Work

In this thesis, we have presented referral networks, an Active Learning framework where experts can redirect difficult instances to their colleagues. We found that effective referrals substantially improve the overall performance of the network. Through a series of extensive empirical evaluations both on synthetic and non-synthetic data, we identified a set of algorithms that can successfully address the *learning-to-refer* challenge even in presence of practical constraints like capacity constraints, evolving network and drift in expertise. In order to tackle the cold-start problem, we have proposed an augmented learning setting, proactive skill posting, in which experts are allowed a one-time local-network-advertisement of a subset of their skills.

6.1 Summary of Contributions

In this thesis, we have proposed a novel learning setting, identified key algorithms suitable for tackling the learning-to-refer challenge, assessed their relative merits and shortcomings considering several robustness criterion, and proposed modified versions of a subset of algorithms to deal with an augmented learning setting, also, first proposed here, where experts are allowed to share private information with their colleagues.

In particular, we proposed solutions to three inter-linked research questions:

1. In an Active Learning setting, how can we use communication between experts (agents or teachers) to improve overall performance?
2. Are our proposed methods to learn referrals robust to practical factors like capacity constraints, evolving networks and expertise drift?
3. How side-information like advertised priors can be incorporated into algorithms in a truthful manner to tackle the cold-start problem?

6.1.1 Referral Networks

The existence of a single, omniscient, indefatigable and infallible oracle assumption in classical Active Learning has some serious limitations when applied to practical Machine Learning applications, as we often have multiple annotators with varying expertise and availibilities. Proactive Learning has extended Active Learning along several dimensions by considering multiple oracles with time-varying accuracy. In this work, we introduce the notion of communication between experts in the form of referrals, an aspect not considered in Proactive Learning or any other previous Active Learning research. We propose a novel learning framework where agents (teachers) are connected through a referral network and are allowed to redirect difficult instances to each other. With modern machine learning applications attempting to solve increasingly non-trivial tasks and requiring sophisticated training data in order to achieve proficiency level comparable to human-experts, we see such network of experts improving the overall effectiveness of teachers.

In this part of our work, we were primarily interested in assessing the viability of this learning setting when starting from uninformative priors. Our main goal was to identify a set of algorithms that exhibit rapid improvement in the early phase of learning. To this end, we have found Distributed Interval Estimation Learning, `DIEL`, to be the most-effective algorithm. Other high-performance algorithms include ϵ-`Greedy`, `Double Q-Learning` and `Optimistic Thompson Sampling`. We have also analyzed multi-hop referrals, a more general setting of referral networks and found `DIEL` as a strong candidate with `Thompson Sampling` catching up when number of hop increases.

6.1.2 Robustness Analysis

The notion of a network adds newer challenges such as reluctance of busier experts creating the need for finding comparable alternatives and attrition and influx of experts requiring the need for rapid integration of newly joined experts. Also, adding to the difficulty of *learning to refer*, expertise could improve (or degrade). We conducted an extensive set of experiments to assess the robustness of our referral learning algorithms. Specifically, we considered practical factors such as capacity constraints, evolving networks, and expertise drift - that are largely ignored in the Active Learning literature.

We found that all learning algorithms exhibit graceful degradation in presence of capacity constraints and evolving networks. For evolving networks, our augmented learning setting, proactive skill posting, proved to be particularly useful. For expertise drift, we found `DIEL`, the best-performing algorithm when the distributional parameters of expertise do not change, was susceptible. We designed `Hybrid`, a new performance-gradient based switching algorithm, that

tackled expertise drift the best.

6.1.3 Proactive Skill Posting

In practice, experts do not start from completely uninformative priors on the skills of their colleagues. Moreover, experts often tell their colleagues about their expertise areas. We modelled such sharing of noisy priors in our augmented learning setting, proactive skill posting. The key challenge in this setting was to elicit truthful information from the experts since experts as selfish agents, may overstate their skills to attract more business.

We extended algorithms on uninformed prior setting with an initialization and penalty mechanism and proposed five new proactive algorithms. Our algorithms proved to be robust to noisy self-skill estimates, strategic lying and evolving networks. All proactive algorithms obtained improved steady-state performance and substantial early-learning phase advantage over their non-proactive counterparts. Of the two penalty mechanisms we studied, *penalty on distrust* demonstrated slightly better empircal performance than *penalty on failure* in countering strategic misreporting.

6.2 Future Directions

Our work can be extended in several ways. In this section, we outline some of the prominent research directions.

6.2.1 Referral Networks

Considering richer referral frameworks, referral bias, correlated topics and hierarchical topic-structure could be challenging research directions in the basic referral framework.

Richer Referral Framework

The referral mechanism can be modified in several ways. We outline some interesting directions we intend to explore in the future.

Visibility of the actual solver in a multi-hop referral chain: In a multi-hop referral chain, only the expert who referred directly to the colleague who solved the problem knows about the true solver. If the expert not only communicates the solution, but also who solved it in the multi-hop refferal chain, the additional information can be utilized to improve the expertise estimates and also take an informed decision depending on the remaining query budget (i.e., if I know A is

weak on a given topic but has a strong colleague in her subnetwork, I will only refer to A when the remaining query budget for the task is 2 or more).

Richer handling of overload situations: In our experiments involving capacity constraints, we assumed that an expert becomes completely unavailable when she is overloaded. However, in a practical scenario, she should still be able to refer or maintain a queue of tasks to solve later. Also, since we found that a highly-skilled expert is more likely to get overloaded, a meaningful research question could be: if an expert knew the fraction of the time colleagues were busy (not accepting new tasks), can we utilize this information for an improved estimate of their skills?

Shared updates: So far, we assumed each expert is learning effective referrals completely on their own. However, expert-pairs or team of experts sharing a common goal is a real-world phenomenon. We intend to explore shared updates where one expert can benefit from observations of other experts.

Experts with Bias

So far we assumed that no expert had any bias toward or against any other expert in the network. Unless it is a diversification step, every time an expert makes a referral decision, she picks the expert with the highest score breaking ties randomly, and an expert's referral decisions are independent of any reciprocal behavior. However, it is easy to think of practical applications for which this may not hold. For example, if resolving a label earns money, then we would not expect any asymmetric referral relation – where expert A refers often to expert B but B doesn't return the favor – to continue for long. Our experiments indicate that such relations may in fact be rather common: from Table 6.1 we learn that 42% of the referral links had an 80-20 referral share or worse. Conversely, two (or more) experts on the same topic may *collude* by systematically referring instances of that topic to each other. Devising learning algorithms immune to such strategic behavior could be a promising line of future work.

Referral Share	Percentage of referral links exhibiting the behavior
$> 60\%$	78.62%
$> 70\%$	59.57%
$> 80\%$	41.81%
$> 90\%$	19.64%

Table 6.1: Asymmetry in referral. The left column indicates referral link-wise referral share.

Hierarchical Topic Structure

In our experiments, we have considered noisy topic identification and found our referral-learning algorithms to be robust enough when the topic misclassification rate is small. Extending referral network to a model with hierarchical topic structure, where an expert may not be able to identify the specific subtopic of a task but only one of its super-topics, may lead to more nuanced referral behavior that would be worth studying.

Correlated Topics

We considered that expertise on one topic is independent of expertise on any other topic. However, in practical scenarios, we often see that highly-skilled experts on a specific area also exhibit proficiency in related areas. If information about such topical correlations is available, a meaningful future research question could be: is it possible to algorithmically propagate an expert's skill on $topic_k$ to correlated topic $topic_{k+1}$, either as a prior if we have no knowledge of the expert's skills on $topic_{k+1}$ or as an updated estimate otherwise? Note that, when knowledge in topical correlations is available to an expert, the bounding technique using the second-best skill in proactive skill posting can be modified with tasks correlated to the best and second-best skill receiving higher initial estimates than tasks uncorrelated to the top-two skills.

Non-stationarity in Topic Distributions

In our experiments, we have considered each topic is equally likely and the *initial expert* is chosen uniformly at random. However, in real-world, certain topics could be more predominant than others and there could be extremely rare task-classes. In such cases, experts may require to prioritize their referral-learning with a greater emphasis on popular topics. Also, trending topics would experience a sudden surge of task-requests requiring experts to find out colleagues who can handle such topics if not already known. Modeling non-stationarity in topic distributions and analyzing its effect on referral networks could be an interesting research direction.

We have only considered uniform distribution while selecting the *initial expert*. However, the choice of initial expert may not be uniform and may also depend on the topic. For instance, a medical professional is far more likely to receive a medical question than a programming language question. In our experiments involving capacity constraints, we have seen that the load-situations of experts were correlated with their expertise. In those experiments, the distribution for selecting the *initial expert* would start out as uniform, but eventually it will exhibit non-stationarity depending on the load-situation and availability of the individual experts. However,

we haven't explicitly modeled non-stationarity in our distributional choice for the selection of the *initial expert* and also restricted ourselves only to uniform distribution.

Finite Horizon Bound for DIEL

In our experimental results both on real data and synthetic data KhudaBukhsh et al. (2017b), we have found that DIEL's finite horizon performance is substantially better than a wide range of algorithms. In Auer et al. (2002), an algorithm UCB1-tuned was found to have superior empirical performance than UCB1. In our experiments involving expertise drift, we also found that Pessimistic TS-DIEL could be a useful component for dealing with evolving expertise. 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.

6.2.2 Proactive Skill Posting

Our work on proactive skill posting can be extended in the following ways:

Extensibility

Whereas we succeeded in designing proactive skill posting versions of DIEL and ϵ-Greedy and obtained preliminary results for Q-Learning with a different penalty-mechanism, the challenge of robust priors for MABs has broader appeal beyond referral networks, including proactive versions of Thompson Sampling and its many applications. However, the three-fold goal of improved cold-start performance, robustness to noisy self-skill estimates and immunity to strategic lying is difficult to achieve and requires careful case-by-case algorithmic design. For example, the initialization mechanism of DIEL (or ϵ-Greedy) would not work in the case of UCB1 since UCB1 uses a different exploration technique from DIEL. A deeper exploration of algorithm-specific and more general proactive-posting to provide incentive-compatible guarantees for a wide range of MAB-relevant algorithms could be an important follow-up work.

Market-aware Skill Posting

With K topics, the current assumption is that experts are only aware of their individual skills, and advertise their top skills, e.g., $\mu_{t_{best}}$ and $\mu_{t_{secondBest}}$, to their colleagues. However, in more realistic

settings, topics can have variable difficulties, and experts often have aggregate knowledge of skill distributions; i.e., they may know whether their skills are unique or common. Specifically, we propose that every expert maintains a noisy estimate of $\overline{\mu_{t_k}}$ (average network skill on each topic t_k) and reports the skills with her largest relative advantage μ_Δ (where for a given expert topic pair $\langle e_i, t_k \rangle$, $\mu_{\Delta_{t_k}} = \mu_{e_i,t_k} - \overline{\mu_{t_k}}$). Our preliminary results indicate an advantage to exploiting such topic-distributional information, we have yet to achieve incentive compatibility in this setting, especially with tolerance to noisy self-skill or topic difficulty estimates.

So far, our results indicate that proactive algorithms are tolerant to small amount of noise in self-skill estimates. In presence of a larger amount of noise, even when $\overline{\mu_{t_k}}$ is not known, the advertised priors can be regularized relative to the subnetwork. For instance, if it is common knowledge that a particular topic is difficult and an expert's first few attempts indicate that she grossly overestimated her skill, her reported skill can be re-adjusted with the average of other colleagues' advertised priors.

Strategyproofness

While misreporting was shown to be of little or no benefit when other experts report truthfully, a stronger degree of incentive compatibility, strategyproofness, would require proving truth is the optimal strategy for each expert no matter what other experts do. A further investigation on what modifications to the proactive algorithms, or which additional conditions or constraints would be required to achieve this stronger guarantee could be a challenging research goal. Unlike our previous experiments with Bayesian-Nash incentive compatibility, empirical evaluation will be much harder given that we need to sample from a vast strategy-space of a larger number of experts.

Continuous Rewards

In real life, many tasks involve task-responses beyond simple binary states (e.g., what fraction of all constraints an optimization algorithm satisfies, by how far the prediction of a stock value is off, in a scale of 1-10, how confident the doctor is in diagnosing her patient with stage-two melanoma). Exploring reward mechanisms to handle continuous rewards could further improve network performance and broaden its impact, as an effective referral will maximize not only solution likelihood but also solution quality.

6.2.3 Robustness Analysis

Many of our robustness criterion were tested in isolation, i.e., when we were interested in assessing the effect of evolving networks, we did not consider capacity constraints. Our first two proposed directions combine expertise drift with skill posting and evolving networks. Different notions of drift and a structured approach to design mixed-strategy multi-armed bandit algorithms conclude our future research directions.

Skill-posting with Drift

In proactive skill posting with skill drift, a truthful expert will get penalized if she improves (or gets worse) over time, as her initial estimates are no longer valid. We propose the following modifications to proactive skill posting for tackling expertise drift.

- **Updating the posted advertisement:** Since the penalty is a function of advertised prior, it is crucial for the drifting experts to keep their colleagues informed about their current skill level. We introduce the notion of *updating advertisement* (constrained by a budget similar to previous setting) that an expert can occasionally use to update their colleagues. For a gradual change, an update advertisement of the nature $\langle e_i, e_j, t_k, \downarrow \rangle$ (in case of degradation) or $\langle e_i, e_j, t_k, \uparrow \rangle$ (in case of improvement) could be sent out to the colleagues. For a sudden large shift, an update advertisement of $\langle e_i, e_j, t_k, \Delta_\mu \rangle$ can be used where Δ_μ is the amount of shift.
- **Shared updates:** In a non-stationary drift setting, shared updates (see, Chapter 6.2.1) could be particularly useful where one expert can share an observed recent improvement (or sudden loss of skill) of a common colleague.

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.

Mixed Strategy Algorithm Design

Thus far we primarily focused on a performance-gradient-based algorithm switch. We propose a richer systematic exploration of the algorithm configuration challenge, avoiding combinatorial search in configuration space. Specifically, say, we intend to create a mixed-strategy referral algorithm from K algorithms. We declare K parameters $w_1, \ldots, w_K$ such that algorithm i is executed with a probability $\frac{w_i}{W}$ where $W = \sum_{i=1}^{K} w_i$. We can obtain a mixed-strategy referral algorithm by configuring the parameterized algorithm for discrete domains of w_is on a referral networks data set with the overall task accuracy as the objective function to maximize. For this, leveraging the advances in the algorithm configuration literature (see, e.g., (Hutter et al., 2009; Ansótegui et al., 2009, 2015; López-Ibánez et al., 2011; Lang et al., 2015; Ansel et al., 2014; Hutter et al., 2011)) could be useful.

Bibliography

S. Abdallah and M. Kaisers. Addressing environment non-stationarity by repeating q-learning updates. *The Journal of Machine Learning Research*, 17(1):1582–1612, 2016. 2.2, 3.4.8

S. Abdallah and V. R. Lesser. Learning the task allocation game. In *Proc of AAMAS '06*, pages 850–857. ACM, 2006. 2.1

S. Abdallah and V. R. Lesser. A multiagent reinforcement learning algorithm with non-linear dynamics. *Journal of Artificial Intelligence Research*, 33:521–549, 2008. 3.4.8

R. Agrawal. Sample mean based index policies with o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, pages 1054–1078, 1995. 2.1, **??**, 3.4.4

S. Agrawal and N. Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *COLT*, pages 39–1, 2012. 3.4.10, 3.4.11

S. Agrawal and N. Goyal. Further optimal regret bounds for thompson sampling. In *Artificial Intelligence and Statistics*, pages 99–107, 2013. 3.4.10

L. Von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64. ACM, 2006. 1

N. Akakpo. Detecting change-points in a discrete distribution via model selection. *arXiv preprint arXiv:0801.0970*, 2008. 2.2

J. Ansel, S. Kamil, K. Veeramachaneni, J. Ragan-Kelley, J. Bosboom, U. O'Reilly, and S. Amarasinghe. Opentuner: An extensible framework for program autotuning. In *Proceedings of the 23rd international conference on Parallel architectures and compilation*, pages 303–316. ACM, 2014. 6.2.3

C. Ansótegui, M. Sellmann, and K. Tierney. A gender-based genetic algorithm for the automatic configuration of algorithms. *Principles and Practice of Constraint Programming-CP 2009*, pages 142–157, 2009. 6.2.3

C. Ansótegui, Y. Malitsky, H. Samulowitz, M. Sellmann, and K. Tierney. Model-based genetic

algorithms for algorithm configuration. In *IJCAI*, pages 733–739, 2015. 6.2.3

D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook. *The traveling salesman problem: a computational study*. Princeton university press, 2011. 3.7

J. Y. Audibert and S. Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11(Oct):2785–2836, 2010. 3.4.4

J. Y. Audibert, R. Munos, and C. Szepesvári. Tuning bandit algorithms in stochastic environments. In *International Conference on Algorithmic Learning Theory*, pages 150–165. Springer, 2007. 2.1, **??**, 3.4.4, 3.4.7

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002. **??**, **??**, 3.4.4, 3.5, 6.2.1

R. Axelrod. Advancing the art of simulation in the social sciences. *Journal of the Japanese and International Economies*, 12(3):16–22, 2003. 2.4

M. G. Azar, R. Munos, M. Ghavamzadeh, and H. J. Kappen. Speedy q-learning. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 2411–2419. Curran Associates Inc., 2011. 3.4.8

M. Babaioff, Y. Sharma, and A. Slivkins. Characterizing truthful multi-armed bandit mechanisms. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 79–88. ACM, 2009. 2.3

M. Babaioff, Y. Sharma, and A. Slivkins. Characterizing truthful multi-armed bandit mechanisms. *SIAM Journal on Computing*, 43(1):194–230, 2014. 4.3

A. L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439): 509–512, 1999. 4.4

M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: a word processor with a crowd inside. In *Proc. of UIST '10*, pages 313–322. ACM, 2010. 1

D. A. Berry and B. Fristedt. *Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability)*, volume 12. Springer, 1985. 3.4.1

D. Bertsimas and J. Niño-Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Operations Research*, 48(1):80–90, 2000. 2.2

A. Biere, A. Cimatti, E. M. Clarke, M. Fujita, and Y. Zhu. Symbolic model checking using sat procedures instead of bdds. In *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*, pages 317–320. ACM, 1999. 3.7

A. Biere, M. Heule, and H. van Maaren. *Handbook of satisfiability*, volume 185. IOS press, 2009. 3.7

A. Biswas, S. Jain, D. Mandal, and Y. Narahari. A truthful budget feasible multi-armed bandit mechanism for crowdsourcing time critical tasks. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1101–1109. International Foundation for Autonomous Agents and Multiagent Systems, 2015. 2.3

A. Blum and Y. Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(Jun):1307–1324, 2007. 2.2

D. Bouneffouf and R. Feraud. Multi-armed bandit problem with known trend. *Neurocomputing*, 205:16–21, 2016. 2.3

M. Bowling. Convergence and no-regret in multiagent learning. In *Advances in neural information processing systems*, pages 209–216, 2005. 3.4.8

M. Bowling and M. Veloso. Rational and convergent learning in stochastic games. In *International joint conference on artificial intelligence*, volume 17, pages 1021–1026. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001. 2.2

K. Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 59–66, 2003. 2.1

G. Burtini, J. Loeppky, and R. Lawrence. A survey of online experiment design with the stochastic multi-armed bandit. *arXiv preprint arXiv:1510.00757*, 2015. 4.3

D. Chakrabarti, R. Kumar, F. Radlinski, and E. Upfal. Mortal multi-armed bandits. In *Advances in neural information processing systems*, pages 273–280, 2009. 2.1, 2.2

O. Chapelle and L. Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011. 2.4, 3.4.10, 3.7

T. Chen and L. He. Collaborative filtering based on demographic attribute vector. In *Future Computer and Communication, 2009. FCC'09. International Conference on*, pages 225–229. IEEE, 2009. 2.3

J. Cheng and M. S. Bernstein. Flock: Hybrid crowd-machine learning classifiers. In *Proc. of CSCW 2015*, pages 600–611. ACM, 2015. 1

E. Clarke, D. Kroening, and F. Lerda. A tool for checking ANSI-C programs. In *Proceedings of the Tenth International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2004)*, pages 168–176, 2004. **??**

S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971. 3.7

J. M. Crawford and A. B. Baker. Experimental results on the application of satisfiability algorithms to scheduling problems. In *AAAI*, volume 2, pages 1092–1097, 1994. 3.7

P. Donmez and J. G. Carbonell. Proactive Learning : Cost-Sensitive Active Learning with Multiple Imperfect Oracles. *Proceedings of CIKM '08*, 08:619–628, 2008. 1, 2.2, 4.1

P. Donmez, J. G. Carbonell, and P. N. Bennett. Dual Strategy Active Learning. *Machine Learning ECML 2007*, pages 116–127, 2007. 2.2

P. Donmez, J. G. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (2009)*, page 259, 2009a. 2.1, 3.4.1

P. Donmez, J. G. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 259–268. ACM, 2009b. 2.1

P. Donmez, J. G. Carbonell, and J. Schneider. A Probabilistic Framework to Learn from Multiple Annotators with Time-Varying Accuracy. *Proceedings of the SIAM International Conference on Data Mining (SDM 2010)*, pages 826–837, 2010a. 1

P. Donmez, J. G. Carbonell, and J. Schneider. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 826–837. SIAM, 2010b. 2.1, 2.2

N. Eén and A. Biere. Effective preprocessing in SAT through variable and clause elimination. In *Proceedings of the Eighth International Conference on Theory and Applications of Satisfiability Testing (SAT'05)*, pages 61–75, 2005. **??**

L. N. Foner. Yenta: a multi-agent, referral-based matchmaking system. In *Proceedings of the first international conference on Autonomous agents*, pages 301–307. ACM, 1997. 2.1

A. S. Fraenkel. Complexity of protein folding. *Bulletin of mathematical biology*, 55(6):1199–1210, 1993. 3.7

Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth. Using and combining predictors that specialize. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 334–343. ACM, 1997. 2.2

J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4):44, 2014. 2.2

A. Garivier and O. Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual Conference On Learning Theory*, pages 359–376, 2011. 3.4.4

A. Garivier and E. Moulines. On upper-confidence bound policies for non-stationary bandit problems. *arXiv preprint arXiv:0805.3415*, 2008. 2.2

A. Van Gelder. Another look at graph coloring via propositional satisfiability. *Discrete Applied Mathematics*, 156(2):230–243, 2008. 3.7

I. P. Gent, H. H. Hoos, P. Prosser, and T. Walsh. Morphing: Combining structure and randomness. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*, pages 654–660, 1999. **??**

C. P. Gomes and B. Selman. Problem structure in the presence of perturbations. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI'97)*, pages 221–226, 1997. **??**

T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 13–20, 2010. 3.4.10

T. Grenager, R. Powers, and Y. Shoham. Dispersion games: general definitions and some specific learning results. In *AAAI/IAAI*, pages 398–403, 2002. 2.2

Y. Guo and D. Schuurmans. Discriminative batch mode active learning. In *Advances in neural information processing systems*, pages 593–600, 2008. 2.1

N. Gupta, O. C. Granmo, and A. Agrawala. Thompson sampling for dynamic multi-armed bandits. In *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, volume 1, pages 484–489. IEEE, 2011. 2.2, 4.3.2

C. Hartland, S. Gelly, N. Baskiotis, O. Teytaud, and M. Sebag. Multi-armed bandit, dynamic environments and meta-bandits. 2006. 2.2

K. Heimerl, B. Gawalt, K. Chen, T. Parikh, and B. Hartmann. Communitysourcing: engaging local crowds to perform expert work via physical kiosks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1539–1548. ACM, 2012. 1

E. Hillel, Z. S. Karnin, T. Koren, R. Lempel, and O. Somekh. Distributed exploration in multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 854–862, 2013. 2.1

E. A. Hirsch. Random generator hgen2 of satisfiable formulas in 3-CNF. `http:`

//logic.pdmi.ras.ru/~hirsch/benchmarks/hgen2-1.01.tar.gz. Last accessed on Sept. 16, 2013., 2002. **??**

S. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th international conference on World Wide Web*, pages 633–642. ACM, 2006a. 2.1

S. Hoi, R. Jin, J. Zhu, and M.. R. Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, pages 417–424. ACM, 2006b. 2.1

P. Holme and B. J. Kim. Growing scale-free networks with tunable clustering. *Physical review E*, 65(2):026107, 2002. 4.4

L. Huang, A. D. Joseph, B. Nelson, B. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 43–58. ACM, 2011. 2.3

F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle. Paramils: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267–306, 2009. 6.2.3

F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. *LION*, 5:507–523, 2011. 6.2.3

T. Jaakkola, M. I. Jordan, and S. P. Singh. Convergence of stochastic iterative dynamic programming algorithms. In *Advances in neural information processing systems*, pages 703–710, 1994. 3.4.8

D. Jensen and J. Neville. Data Mining in Social Networks. In *National Academy of Sciences Symposium on Dynamic Social Network Modeling and Analysis*, 2002. 2.4

L. P. Kaelbling. *Learning in embedded systems*. MIT press, 1993. 2.1, 3.4.1

L. P. Kaelbling, M. L. Littman, and A. P. Moore. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996. 2.1

M. Kaisers and K. Tuyls. Frequency adjusted multi-agent q-learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 309–316. International Foundation for Autonomous Agents and Multiagent Systems, 2010. 2.2, 3.4.8

K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Poczos. Asynchronous parallel bayesian optimisation via thompson sampling. *arXiv preprint arXiv:1705.09236*, 2017. 2.4, 3.7

A. Kapoor, E. Horvitz, and S. Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI*, volume 7, pages 877–882, 2007. 2.1

E. Kaufmann, O. Cappé, and A. Garivier. On bayesian upper confidence bounds for bandit problems. In *Artificial Intelligence and Statistics*, pages 592–600, 2012. 3.4.4

H. Kautz and B. Selman. Pushing the envelope: Planning, propositional logic, and stochastic search. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1194–1201, 1996. 3.7

H. Kautz and B. Selman. Unifying sat-based and graph-based planning. In *IJCAI*, volume 99, pages 318–325, 1999. 3.7

H. Kautz, B. Selman, and A. Milewski. Agent amplified communication. pages 3–9, 1996. 2.1

A. R. KhudaBukhsh, L. Xu, H. H. Hoos, and K. Leyton-Brown. Satenstein: Automatically building local search sat solvers from components. In *IJCAI*, volume 9, pages 517–524, 2009. 1.2, 2.4, 3.5, 3.7

A. R. KhudaBukhsh, J. G. Carbonell, and P. J. Jansen. Proactive Skill Posting in Referral Networks. In *Australasian Joint Conference on Artificial Intelligence*, pages 585–596. Springer, 2016a. 1, 2.1, **??**, 3.4.1, 3.5, 4.3.2, 4.3.4, 5.4.2, **??**, 5.5, 5.6.3

A. R. KhudaBukhsh, J. G. Carbonell, and P. J. Jansen. Proactive-DIEL in Evolving Referral Networks. In *European Conference on Multi-Agent Systems*, pages 148–156. Springer, 2016b. 1, 4.3.2, 4.3.4, **??**, 5.6.3, 5.6.5

A. R. KhudaBukhsh, P. J. Jansen, and J. G. Carbonell. Distributed learning in expert referral networks. In *European Conference on Artificial Intelligence (ECAI), 2016*, pages 1620–1621, 2016c. **??**, 3.5, 3.5, 4.3.2, 4.3.4, 4.3.5

A. R. KhudaBukhsh, L. Xu, H. H. Hoos, and K. Leyton-Brown. Satenstein: Automatically building local search sat solvers from components. *Artificial Intelligence*, 232:20–42, 2016d. 1.2, 2.4, 3.5, 3.7, 5.5

A. R. KhudaBukhsh, J. G. Carbonell, and P. J. Jansen. Incentive compatible proactive skill posting in referral networks. In *European Conference on Multi-Agent Systems*, page [to appear]. Springer, 2017a. 1, 5.4.2, **??**, **??**, **??**

A. R. KhudaBukhsh, J. G. Carbonell, and P. J. Jansen. Robust learning in expert networks: A comparative analysis. In *International Symposium on Methodologies for Intelligent Systems*, pages 292–301. Springer, 2017b. **??**, 4.3.2, 4.3.4, 4.3.5, 6.2.1

H. N. Kim, A. T. Ji, I. Ha, and G. S. Jo. Collaborative filtering based on collaborative tagging for

enhancing the quality of recommendation. *Electronic Commerce Research and Applications*, 9(1):73–83, 2010. 2.3

R. D. King, K. E. Whelan, F. M. Jones, P. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 2004. 2.1

R. Kleinberg, A. Niculescu-Mizil, and Y. Sharma. Regret bounds for sleeping experts and bandits. *Machine learning*, 80(2-3):245–272, 2010. 2.1, 2.2

N. Korda, E. Kaufmann, and R. Munos. Thompson sampling for 1-dimensional exponential family bandits. In *Advances in Neural Information Processing Systems*, pages 1448–1456, 2013. 3.4.10

T. L. Lai. *Sequential analysis*. Wiley Online Library, 2001. 2.2

T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985. 2.1, **??**, 3.4.4

M. Lang, H. Kotthaus, P. Marwedel, C. Weihs, J. Rahnenführer, and B. Bischl. Automatic model selection for high-dimensional survival analysis. *Journal of Statistical Computation and Simulation*, 85(1):62–76, 2015. 6.2.3

J. Langford, A. Strehl, and J. Wortman. Exploration scavenging. In *Proceedings of the 25th international conference on Machine learning*, pages 528–535. ACM, 2008. 2.3

C. Leung, S. Chan, and F. L. Chung. An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge-Based Systems*, 21(7):515–529, 2008. 2.3

D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the eleventh international conference on machine learning*, pages 148–156, 1994. 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*, pages 3–12. Springer-Verlag New York, Inc., 1994. 1

S. Lin, W. Hong, D. Wang, and T. Li. A survey on expert finding techniques. *Journal of Intelligent Information Systems*, pages 1–25, 2017. 2.4

M. L. Littman and C. Szepesvári. A generalized reinforcement-learning model: Convergence and applications. In *ICML*, pages 310–318, 1996. 3.4.8

K. Liu and Q. Zhao. Indexability of restless bandit problems and optimality of whittle index for

dynamic multichannel access. *IEEE Transactions on Information Theory*, 56(11):5547–5567, 2010. 2.2

S. Loh, F. Lorenzi, R. Granada, D. Lichtnow, L. K. Wives, and J. de Oliveira. Identifying similar users by their scientific publications to reduce cold start in recommender systems. In *WEBIST*, volume 9, pages 593–600, 2009. 2.3

M. López-Ibánez, J. Dubois-Lacoste, T. Stützle, and M. Birattari. The irace package, iterated race for automatic algorithm configuration. Technical report, Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011. 6.2.3

T. L. MacKay, N. Bard, M. Bowling, and D. C. Hodgins. Do pokers players know how good they are? accuracy of poker skill estimation in online and offline players. *Computers in Human Behavior*, 31:419–424, 2014. 5, 5.6.3

P. Manavalan and M. P. Singh. Emerging properties of knowledge sharing referral networks: Considerations of effectiveness and fairness. *Lecture Notes in Computer Science*, pages 13–23, 2012. 2.4

S. Maneeroj and A. Takasu. Hybrid recommender system using latent features. In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*, pages 661–666. IEEE, 2009. 2.3

B. C. May, N. Korda, A. Lee, and D. S. Leslie. Optimistic bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research*, 13(Jun):2069–2106, 2012. 2.1, **??**, 3.4.11

D. W. McDonald and M. S. Ackerman. Expertise recommender: a flexible recommendation system and architecture. *CSCW '00 Proceedings of the 2000 ACM conference on Computer Supported Cooperative Work*, pages 231–240, 2000. 2.4

R. Nallapati, S. Peerreddy, and P. Singhal. Skierarchy: Extending the power of crowdsourcing using a hierarchy of domain experts, crowd and machine learning. Technical report, DTIC Document, 2012. 2.1

J. Newsome, B. Karp, and D. Song. Paragraph: Thwarting signature learning by training maliciously. In *International Workshop on Recent Advances in Intrusion Detection*, pages 81–105. Springer, 2006. 2.3

I. Noda. Recursive adaptation of stepsize parameter for non-stationary environments. In *ALA*, pages 74–90. Springer, 2009. 2.2

T. Osugi and S. Scott. Balancing Exploration and Exploitation: A New Algorithm for Active

Machine Learning. *Fifth IEEE International Conference on Data Mining*, pages 330–337, 2005. 2.1

N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, B. Z. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016. 2.3

S. T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste. Naïve filterbots for robust cold-start recommendations. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 699–705. ACM, 2006. 2.3

M. Pop, S. L. Salzberg, and M. Shumway. Genome sequence assembly: Algorithms and issues. *Computer*, 35(7):47–54, 2002. 3.7

S. Pushpa, K. S. Easwarakumar, S. Elias, and Z. Maamar. Referral based expertise search system in a time evolving social network. *Proceedings of the Third Annual ACM Bangalore Conference on - COMPUTE '10*, pages 1–8, 2010. 2.4

G. J. Qi, X. S Hua, Y. Rui, J. Tang, and H. J. Zhang. Two-dimensional active learning for image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2.1

V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010. 2.1

R. Reichart, K. Tomanek, U. Hahn, and A. Rappoport. Multi-task active learning for linguistic annotations. In *ACL*, volume 8, pages 861–869, 2008. 2.1

J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial intelligence review*, 24(1):33–60, 2005. 2.4

B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010. 1

V. S. Sheng and C. X. Ling. Feature value acquisition in testing: a sequential batch test algorithm. In *Proceedings of the 23rd international conference on Machine learning*, pages 809–816. ACM, 2006. 2.1

V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM, 2008. 2.1

W. Sherchan, S. Nepal, and C. Paris. A survey of trust in social networks. *ACM Computing Surveys (CSUR)*, 45(4):47, 2013. 2.4

P. Shivaswamy and T. Joachims. Multi-armed bandit problems with history. In *Artificial Intelligence and Statistics*, pages 1046–1054, 2012. 2.3

B. C. Da Silva, E. W. Basso, A. Bazzan, and P. M. Engel. Dealing with non-stationary environments using context detection. In *Proceedings of the 23rd international conference on Machine learning*, pages 217–224. ACM, 2006. 2.2

L. Simon. SAT competition random 3CNF generator. `www.satcompetition.org/2003/TOOLBOX/genAlea.c`. Last accessed on Sept. 16, 2013., 2002. **??**

R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics, 2008. 2.1

A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008. 2.1

P. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. Combinational test generation using satisfiability. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(9):1167–1176, 1996. 3.7

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998. 3.4.8

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. 2.1, 2.2, **??**, 3.4.10

L. Tran-Thanh, A. Chapman, A. Rogers, and N. R. Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. *arXiv preprint arXiv:1204.1909*, 2012a. 2.3

L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings. Efficient crowdsourcing of unknown experts using multi-armed bandits. In *European Conference on Artificial Intelligence*, pages 768–773, 2012b. 2.3

J. N. Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16 (3):185–202, 1994. 3.4.8

A. Tsymbal. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, 106(2), 2004. 2.2

T. Uchida and O. Watanabe. Hard SAT instance generation based on the factorization problem. `http://www.is.titech.ac.jp/~watanabe/gensat/a2/GenAll.tar.`

gz, 1999. **??**

H. van Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems*, pages 2613–2621, 2010. 2.1, **??**, 3.4.8, 3.4.9

S. Vijayanarasimhan and K. Grauman. Cost-sensitive active visual category learning. *International Journal of Computer Vision*, 91(1):24–44, 2011. 2.1

C. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992. 2.1, **??**, 3.4.8

D. J. Watts and S. H. Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684): 440–442, 1998. 4.4

R. R. Weber and G. Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27(3):637–648, 1990. 2.2

W. Wei, C. M. Li, and H. Zhang. A switching criterion for intensification, and diversification in local search for sat. *Journal on Satisfiability, Boolean Modeling and Computation*, 4:219–237, 2008. 2.2

L. T. Weng, Y. Xu, Y. Li, and R. Nayak. Exploiting item taxonomy for solving cold-start problem in recommendation making. In *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, volume 2, pages 113–120. IEEE, 2008. 2.3

J. Whitehill, T. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009. 2.1

P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, 25(A):287–298, 1988. 2.1, 2.2

M. Wiering and J. Schmidhuber. Efficient model-based exploration. In *Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior (SAB98)*, pages 223–228, 1998. 3.4.1

Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. In *ECiR*, volume 7, pages 246–257. Springer, 2007. 2.1

L. Yang and J. G. Carbonell. Buy-in-bulk active learning. In *Advances in Neural Information Processing Systems*, pages 2229–2237, 2013. 2.1

P. Yolum and M. P. Singh. Dynamic communities in referral networks. *Web Intelligence and Agent Systems*, 1(2):105–116, 2003. 2.1

B. Yu. *Emergence and evolution of agent-based referral networks*. PhD thesis, North Carolina State University, 2002. 2.1, 2.4

B. Yu and M. P. Singh. Searching social networks. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems AAMAS 03*, 2003. 2.1

B. Yu, M. Venkatraman, and M. P. Singh. An adaptive social network for information access: Theoretical and experimental results. *Applied Artificial Intelligence*, 17:21–38, 2003. 2.1

J. Y. Yu and S. Mannor. Piecewise-stationary bandit problems with side observations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1177–1184. ACM, 2009. 2.2

L. Yu. Crowd creativity through combination. In *Proc. of Creativity and Cognition 2015*, pages 471–472. ACM, 2011. 1

L. Yu and J. V. Nickerson. An internet-scale idea generation system. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 3(1):2, 2013. 1

M. C. Yuen, I. King, and K. S. Leung. A survey of crowdsourcing systems. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 766–773. IEEE, 2011. 1

C. Zhang and V. R. Lesser. Multi-agent learning with policy prediction. In *AAAI*, 2010. 3.4.8

C. Zhang, V. R. Lesser, and P. Shenoy. A Multi-Agent Learning Approach to Online Distributed Resource Allocation. In *Proc. of IJCAI-09*, volume 1, pages 361–366, Pasadena, CA, 2009. URL `http://mas.cs.umass.edu/paper/467`. 2.1

C. Zhang, V. R. Lesser, and S. Abdallah. Self-Organization for Coordinating Decentralized Reinforcement Learning. In K. van der Hoek, editor, *Proc. of AAMAS '10*, pages 739–746, Toronto, 2010. URL `http://mas.cs.umass.edu/paper/482`. 2.1

H. Zhang and V. R. Lesser. A reinforcement learning based distributed search algorithm for hierarchical peer-to-peer information retrieval systems. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 47. ACM, 2007. 2.1