# Incentive Compatible Proactive Skill Posting in Referral Networks

Ashiqur R. KhudaBukhsh[1], Jaime G. Carbonell[1], and Peter J. Jansen[1]

Carnegie Mellon University
{akhudabu, jgc, pjj}@cs.cmu.edu

**Abstract** Learning to refer in a network of experts (agents) consists of distributed estimation of other experts' topic-conditioned skills so as to refer problem instances too difficult for the referring agent to solve. This paper focuses on the cold-start case, where experts post a subset of their top skills to connected agents, and as the results show, improve overall network performance and, in particular, early-learning-phase behavior. The method surpasses state-of-the-art, i.e., proactive-DIEL, by proposing a new mechanism to penalize experts who misreport their skills, and extends the technique to other distributed learning algorithms: proactive-$\epsilon$-Greedy, and proactive-Q-Learning. Our proposed new technique exhibits stronger discouragement of strategic lying, both in the limit and finite-horizon empirical analysis. The method is shown robust to noisy self-skill estimates and in evolving networks.

**Keywords:** active learning; referral networks; proactive skill posting

## 1 Introduction

Learning-to-refer in expert referral networks is a recently proposed active learning setting where an expert can refer problem instances to appropriate colleagues if she finds the task at hand difficult to solve [1]. Such a network draws inspiration from the real world examples of expert networks, such as among physicians or within consultancy firms. Initially designed for uninformative priors, an extension of the learning setting is proposed in [2] where experts are allowed a one-time local-network advertisement of a subset of their skills to their colleagues. The success in the extended learning setting depends on a *truthful mechanism* to elicit the true skills of the experts in the network. The experts, as selfish agents, try to maximise the number of tasks they receive to maximize fees. In this paper, we propose a novel penalty mechanism (applied to a diverse set of action selection algorithms) that shows stronger discouragement to strategic lying, including incentive compatibility for some referral algorithms, and also obtains a modest performance improvement.

While we study and contrast the behavior in the limit of our proposed mechanism against past work (see, Section 3.3), and show that theoretically, our mechanism discourages willful misreporting better than previous work, many of our experimental results deal with finite-horizon behavior (see, Section 5.2), acknowledging that in a practical setting, we cannot afford an unbounded number

of samples to identify truthful, skilled workers. Although our primary focus is on referral networks, the challenge that we are addressing is relevant to the multi-armed bandit problem with partially-available noisy priors, a fairly general problem that may arise in several applications. We also see our work as a part of the growing trend of several lines of research on adversarial Machine Learning [3].

A key aspect on which we differ from past works on multi-armed bandits [4–7] is our choice of data sets: in addition to constructing traditional synthetic data that obeys well-known distributions, we evaluate algorithms on a referral network of high-performance SAT (propositional satisfiability problem) solvers where neither expertise nor noise in estimating skill obey known parameterized distributions.

## 2 Related Work

| | proactive-DIEL [2] | proactive-DIEL$_t$ | proactive-$\epsilon$Greedy [8] | proactive-$\epsilon$Greedy$_t$ | proactive-Q-Learning$_t$ |
|---|---|---|---|---|---|
| Incentive Compatibility | ✓ [2] | ✓✓ | ✓ | ✓✓ | ✓✓ |
| Tolerance to noisy skill-estimates | ✓ [2] | ✓ | ✓ | ✓ | ✓✓ |
| Early performance gain | ✓ [2] | ✓✓ | ✓ | ✓✓ | ✓✓ |
| Steady-state performance gain | ✓ [2] | ✓✓ | ✓ | ✓✓ | ✓✓ |
| Robustness to evolving networks | ✓ [8] | ✓✓ | ✓ | ✓✓ | ✓✓ |

**Table 1.** Summary of contributions: Blue columns represent new algorithms first proposed in this paper. Blue cells indicate new experimental results (e.g., cell (3,1), (3,5)), a check mark indicates that a property holds, and two check marks indicate we improve the known state of the art (including the case where there were no known previous baselines to compare against).

Our starting point for this work was the augmented setting of referral learning [1, 9] first proposed in [2] and then extended in [8]. [2] proposed several modifications to Distributed Interval Estimation Learning (DIEL), up to then the best referral learning algorithm on uninformative priors. The modified algorithm, *proactive*-DIEL, demonstrated superior performance, especially during the initial learning phase, even in the presence of noise in skill self-estimates. It also showed empirical evidence of being near-Bayesian-Nash Incentive Compatible, i.e., misreporting skills to receive more referrals provided little or no benefit when all other experts report truthfully. More recently, [8] showed that the mechanism proposed in proactive-DIEL can be adapted with minor modifications to another algorithm ($\epsilon$-Greedy), and that the new algorithm is robust to noisy self-skill estimates. Compared to the experiments reported in [8], we achieve stronger incentive compatibility covering a wider range of referral algorithms while showing comparable or better resilience to noise and dynamic network changes.

In our work, the baseline algorithms are the non-proactive referral algorithms, of which DIEL is the known state of the art in the non-proactive setting. DIEL, a reinforcement learning technique balancing the exploration-exploitation trade-off, traces back to a chain of research on interval estimation learning, first proposed in [10, 11] and has been successfully used in jointly learning the accuracy of labeling sources and obtaining the most informative labels in [12]. Adversarial

Machine Learning focuses on a wide variety of issues, ranging from adversarial attempts to alter or influence the training data [13] to intrusion attacks by crafting negatives that would pass a classifier (false negatives) [14]. A comprehensive survey is available in [3]. In our work, deliberate skill misreporting from an expert would not only make it difficult for connected experts to learn appropriate referral choices, but it may potentially enable a weaker expert receive more business at the expense of a stronger expert and thus reducing the overall network performance.

While we note that there exists a large body of literature on truthful mechanism design [7, 15–17], a few key differences set us apart from *budgeted multi-armed bandit mechanism* motivated by crowdsourcing platforms presented in [16]. Our setting is *distributed*; hence it consists of many parallel multi-armed bandit problems. Also, 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, [16] considered only homogenous tasks. Reflecting real-world scenarios where experts have differential expertise across topics, and communication/advertisement is focused on the top skills, proactive-DIEL deals with partially available priors, i.e., experts are restricted to bidding for business in their top skill areas only, (a factor [16] did not need to consider because of homogeneous tasks). Unlike budget-limited MAB [16, 18, 19], the budget restriction in our case is on the advertisement; although we focus on a finite-horizon performance analysis, there is no restriction on exploration or exploitation as such. Finally, we present proof sketches for incentive compatibility in the limit, as well as empirical performance evaluation on both synthetic data and real-world data without distributional assumptions.

## 3 Referral Network

### 3.1 Preliminaries

We summarize our basic notation, definitions, and assumptions, mostly from [1, 2], where further details regarding expertise, network parameters, proactive skill posting mechanism and simulation details can be found.

**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 co-refer problem instances.

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

**Scenario**: Set of $m$ instances $(q_1, \ldots, q_m)$ belonging to $n$ topics $(t_1, \ldots, t_n)$ 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 which $e_i$ can solve $q_j$.

**Referral mechanism**: For a query budget $Q$ (following [1, 2], we kept fixed to $Q = 2$ across all our current experiments), this consists of the following steps.

1. A user issues an *initial query* $q_j$ to a randomly chosen *initial expert* $e_i$.
2. The 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, with a query budget of $Q-1$. *Learning-to-refer* involves improving the estimate of who is most likely to solve the problem.
4. If the referred expert succeeds, she sends the solution to the initial expert, who sends it to the user.

**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 $\mu_{t_k}$ is $e_j$'s (advertised) topical expertise.

**Advertising budget**: the number of advertising units available to an expert, following [2], set to twice the size of that expert's subnetwork; each expert reports her top two skills to her subnetwork.

**Advertising protocol**: a one-time advertisement that happens 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 advertised in the above protocol.
**Implicit bid**: A topic that is not advertised, for which an upper skill bound $<$ expert's two top advertised skills.

### 3.2 Referral Algorithms

From an individual expert's point of view, the referral decision is an action selection problem. We give a short description of action selection for the non-proactive referral algorithms, and then extend to proactive skill positing.

`DIEL`: `DIEL` uses Interval Estimation Learning to select action $a$ for which the upper-confidence interval $UI(a)$ is largest, where
$UI(a) = m(a) + \frac{s(a)}{\sqrt{n}}$
$m(a)$ is the mean observed reward, $s(a)$ is the standard deviation of the observed rewards and $n$ is the number of observations so far. The intuition behind `DIEL` is to combine exploitation (via high mean) and exploration (via high variance). As in [1, 2], we initialized the mean reward, standard deviation and number of observations for all actions to 0.5, 0.7071 and 2 respectively as a non-informative prior.

$\epsilon$-`Greedy`: Unlike `DIEL`, $\epsilon$-`Greedy` only considers the mean observed reward to determine the most promising action [4]. It explores via an explicit probabilistic diversification step – randomly selecting a connected expert for referral. We set $\epsilon$ as in in [8]: Letting $\epsilon = \frac{\alpha * K}{N}$ (where $K$ is the subnetwork size and $N$ is the total observations) we configured $\alpha$ by a parameter sweep on a training set as in [1].

`Q-Learning`: `Q-Learning` [20] is a model-free reinforcement learning technique used to learn an optimal action selection policy provided that all actions are sampled repeatedly in all states. To ensure this, we combined `Q-Learning` with $\epsilon$-Greedy as an action-selection component. For all of the above algorithms, a successful task receives a reward of 1 and a failed task receives a reward of 0.

### 3.3 Proactive Referral Algorithms

We extend the non-proactive referral algorithms to the augmented setting with proactive skill posting, both in previous work [2, 8] and the current work.

**proactive-DIEL**: In [2], proactive-DIEL was derived from DIEL by enabling each expert to post a self-estimated skill prior initializing the mean expected reward. Given advertisement unit $\langle e_i, e_j, t_k, \mu_{t_k} \rangle$ the $reward_{mean}(e_i, t_k, e_k)$ (mean reward received by expert $e_k$ on topic $t_k$ as observed by expert $e_i$) is initialized to $\mu_{t_k}$ (explicit bid). When not, proactive-DIEL initializes $reward_{mean}(e_i, t_k, e_k)$ to $\mu_{t_{secondBest}}$, which is in effect an upper bound.

Since each expert has an incentive to maximize its income by drawing new business, a probabilistic penalty mechanism was added to discourage misreporting. The probability *penaltyProbability* with which a *penalty* (kept to 0.35 in [2]) is applied, is computed as described in algorithm 1 below.

**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**

**Algorithm 1:** PENALTY MECHANISM

**proactive-$\epsilon$-Greedy**: proactive-$\epsilon$-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. Since one of our primary contributions is a better mechanism to prevent strategic misreporting, we describe this in the context of proactive-Q-Learning$_t$, an algorithm also first proposed here.

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

However, we take a marked deviation in defining the penalty function, which 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,
$distrustFactor_2 = |\mu_{t_{seconBest}} - \hat{\mu}_{t_{secondBest}}| \zeta(n_{t_{secondBest}})$

where $\zeta(n_t) = \frac{n_t}{n_t + C_1}$, a factor ramping up to 1 in the 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 is off from its estimated mean, for the best skill and second-best skill respectively. $C_1$ and $C_2$ are the two configurable parameters of this mechanism; the larger the value, greater is the discouragement for strategic lying. In all our experiments, $C_1$ was set to 50. $C_2$ was set to 1, 2 and 3 for proactive-DIEL$_t$, proactive-$\epsilon$-Greedy$_t$, and proactive-Q-Learning$_t$, respectively.

The newly proposed penalty mechanism differs from the old method 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 - $\epsilon$, $\epsilon \leq 0.5$) across all tasks. $e_1$ reports truthfully while $e_1$ underbids and advertises (1 - $2\epsilon$). For a penalty of $r$ ($r > 0$), the expected mean reward for $e_1$ will be (1 - $\epsilon$) - $\epsilon$ (1 - $\epsilon$) $r$. Due to underbidding, $e_2$ will have an unfair advantage over $e_1$ as her expected mean reward will be (1 - $\epsilon$) - $\epsilon$ (1 - $2\epsilon$) $r$, larger than $e_1$.

**proactive-DIEL$_t$ and proactive-$\epsilon$-Greedy$_t$:** proactive-DIEL$_t$ and proactive-$\epsilon$-Greedy$_t$ denote the corresponding proactive versions with the new penalty mechanism.

We provide proof sketches demonstrating Bayesian-Nash incentive compatibility in the limit for our new mechanism.

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

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

$$\lim_{n\to\infty} \hat{\mu}_{t_{best}} = \mu_{t_{best}} \tag{1}$$

$$\lim_{n\to\infty} \hat{\mu}_{t_{secondBest}} = \mu_{t_{secondBest}} \tag{2}$$

$$\lim_{n\to\infty} \zeta(n) = 1 \tag{3}$$

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 Q-Learning considers a 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-$\epsilon$-Greedy$_t$.*

*Proof.* 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-DIEL$_t$.*

*Proof.* 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,

$$\lim_{n\to\infty} UI(a) = \lim_{n\to\infty} (m(a) + \frac{s(a)}{\sqrt{n}}) = m(a) \tag{4}$$

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.

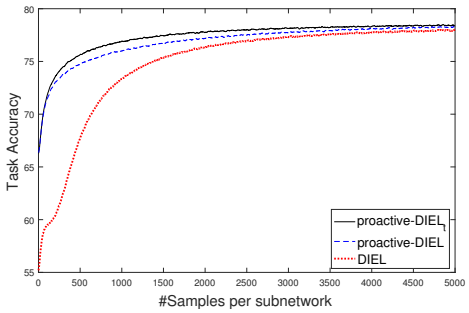
Unlike the Q-learning variants and $\epsilon$-Greedy algorithms, there is no guarantee for DIEL that all actions are visited infinitely often, although a variant can guarantee that condition with random visits at $\epsilon$ probability, and perform similarly in the finite case for small enough $\epsilon$.

## 4 Experimental Setup

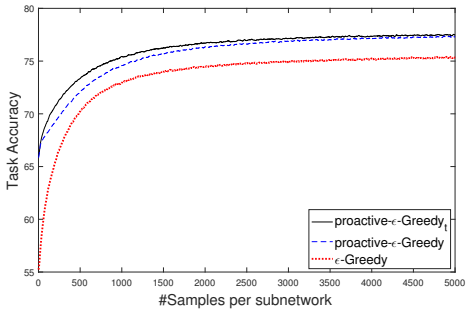
**Data set**: as our synthetic data set, we used the same 1000 scenarios used in [1, 2]. Each scenario consists of 100 experts connected through a referral network and 10 topics. For our experiments involving SAT solvers, we used 100 SATenstein (version 2.0) solvers obtained from the experiments presented in [21] as experts. As topics we use the six SAT distributions on which SATenstein is configured. The details of the SAT distributions can be found in [21].
**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 [1, 2], 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 [2] (described in Section 5.2).
**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 Results

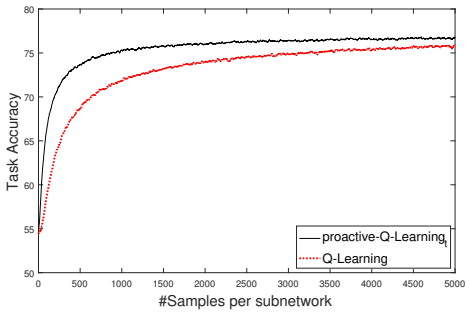
## 5.1 Overall Performance Gain




(a) Proactive-DIEL




(b) Proactive-$\epsilon$-Greedy.




(c) Proactive-Q-Learning.

**Figure 1.** Performance comparison with previous proactive algorithms and corresponding non-proactive versions.

Figure 1 compares the performance of the proactive algorithms with their non-proactive versions under the assumption of truthful reporting and accurate

self-skill estimates. We also compare against the older proactive algorithms of which proactive-DIEL can be considered state of the art. 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 state of the art and both proactive-$\texttt{DIEL}_t$ and proactive-$\epsilon$-$\texttt{Greedy}_t$ did slightly better than the earlier proactive referral algorithms.

### 5.2 Incentive Compatibility

| $\mu_{t_{best}}$ | $\mu_{t_{secondBest}}$ | proactive DIEL | proactive $\texttt{DIEL}_t$ | proactive $\epsilon$Greedy | proactive $\epsilon\texttt{Greedy}_t$ | proactive Q-Learning$_t$ |
|---|---|---|---|---|---|---|
| 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 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.

Next, we focus on the case of deliberate (strategic) misreporting, i.e. experts trying to get more business by overstating (or counter-intuitively, understating) their skills. While our theoretical results (see, Section 3.3) indicate $proactive_t$ algorithms are incentive compatible in the limit, empirical evaluation on a finite horizon addresses practical benefits.

Following [2], we treat the number of referrals received as a proxy for expert benefit, and we empirically analyze Bayesian-Nash incentive compatibility by examining all specific strategy combination (e.g., truthfully report best-skill but overbid second-best skill) that could fetch more referrals (listed in Table 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 :

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

A value greater than 1 implies truthfulness in expectation, i.e., truthful reporting fetched more referrals than strategic lying.

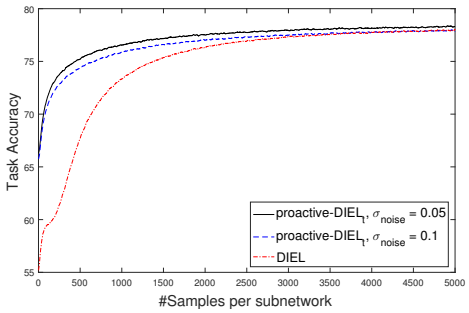
Table 2 presents the *ICFactor*s for each algorithm and each strategy combination. We see that, beyond the steady-state threshold, strategic misreporting is hardly beneficial and in fact counterproductive in most cases. Proactive-DIEL$_t$ was (slightly but consistently) better at discouraging each strategy combination than proactive-DIEL. The only case truthful advertising fetched slightly fewer referrals for proactive-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-Q-Learning$_t$'s *ICFactor* improved to 1.04.

### 5.3 Robustness To Noisy Skill Estimates, Evolving Networks

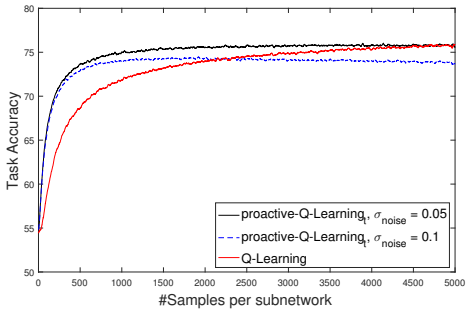
So far, we have shown that our proposed proactive referral algorithms address the cold start problem better than their non-proactive counterparts and provide stronger discouragement to strategic lying. However, even when experts post their skills truthfully, their self-estimates may not be precise. Imprecise skill estimation in proactive skill posting was first explored in [2, 8]. 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. Following [2], 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 $\mu$, and $\sigma_{noise}$ is a small constant (0.05 or 0.1 in our experiments).

Figure 2 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-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-DIEL$_t$ do substantially better in the early-learning phase, there is no steady-state performance gain in the presence of larger noise. Proactive-$\epsilon$-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% ($\sigma_{noise} = 0.1$), 76.76% ($\sigma_{noise} = 0.05$), and 75.26% for the non-proactive version). Proactive-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.

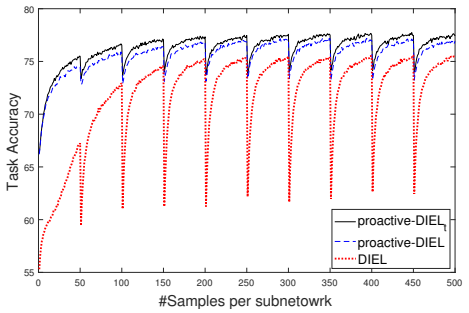
Referal networks may be dynamic, with new experts joining in and old experts leaving. We have already seen that a primary benefit of proactive methods is that they address the cold-start problem. Rapid improvement in the early learning




(a) Noise tolerance of proactive-DIEL$_t$.



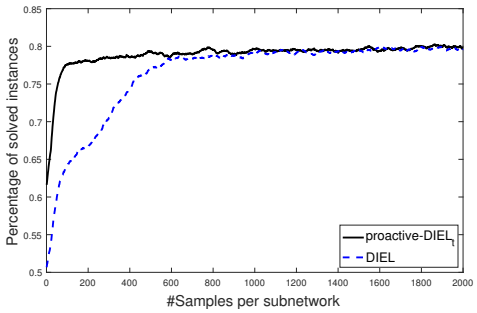

(b) Noise tolerance of proactive-Q-Learning$_t$.



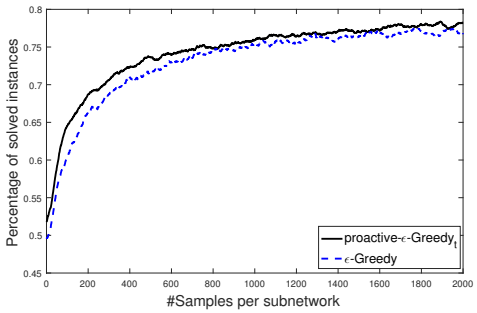

(c) Resilience to evolving networks.

**Figure 2.** Robustness to noisy skill estimates, evolving networks.

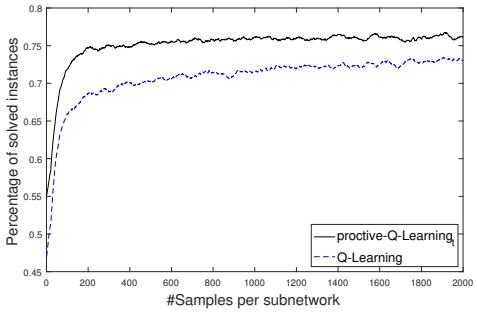
phase is perhaps even more important for evolving networks. Figure 2(c) presents an extreme case of 20% network change at regular interval. We found that the proactive algorithms handled the network changes much better than the original DIEL, with proactive-DIEL$_t$ marginally outperforming proactive-DIEL.




(a) Performance comparison between DIEL and proactive-DIEL$_t$.




(b) Performance comparison between $\epsilon$-Greedy and proactive-$\epsilon$-Greedy$_t$.




(c) Performance comparison between Q-Learning and proactive-Q-Learning$_t$.

**Figure 3.** Performance comparison on SAT solver referral network.

### 5.4 SAT Solver Referral Network

As in [8], 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 3 compares the performance of proactive and non-proactive algorithms on this data set. Figure 3(a) shows that proactive-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 3(b) shows qualitatively similar behavior as the synthetic data set: throughout the learning phase, proactive-$\epsilon$-Greedy$_t$ maintained a modest lead over its non-proactive version.

## 6 Conclusions

We proposed an incentive compatible mechanism improving the state of the art for referral learning, both in overall performance and in discouraging strategic lying. We extended the algorithms (DIEL, $\epsilon$-Greedy) as well as proposed a new one (Q-Learning) to use the new mechanism, and compared their behavior both with and without noise on the self-skill estimates, indicating $\epsilon$-Greedy to be the most and Q-Learning the least robust. Similar experiments on automated agents (SAT solvers) confirmed the results on synthetic data.

# Bibliography


[1] KhudaBukhsh, A.R., Jansen, P.J., Carbonell, J.G.: Distributed Learning in Expert Referral Networks. In: European Conference on Artificial Intelligence (ECAI), 2016. (2016) 1620–1621

[2] KhudaBukhsh, A.R., Carbonell, J.G., Jansen, P.J.: Proactive Skill Posting in Referral Networks. In: Australasian Joint Conference on Artificial Intelligence, Springer (2016) 585–596

[3] Huang, L., Joseph, A.D., Nelson, B., Rubinstein, B.I., Tygar, J.: Adversarial machine learning. In: Proceedings of the 4th ACM workshop on Security and artificial intelligence, ACM (2011) 43–58

[4] Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine Learning **47**(2-3) (2002) 235–256

[5] Chakrabarti, D., Kumar, R., Radlinski, F., Upfal, E.: Mortal multi-armed bandits. In: Advances in neural information processing systems. (2009) 273–280

[6] Xia, Y., Li, H., Qin, T., Yu, N., Liu, T.: Thompson sampling for budgeted multi-armed bandits. CoRR **abs/1505.00146** (2015)
[7] Tran-Thanh, L., Chapman, A.C., Rogers, A., Jennings, N.R.: Knapsack based optimal policies for budget-limited multi-armed bandits. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence. (2012)
[8] KhudaBukhsh, A.R., Carbonell, J.G., Jansen, P.J.: Proactive-DIEL in Evolving Referral Networks. In: European Conference on Multi-Agent Systems, Springer (2016)
[9] KhudaBukhsh, A.R., Carbonell, J.G., Jansen, P.J.: Robust learning in expert networks: A comparative analysis. In: International Symposium on Methodologies for Intelligent Systems, Springer (2017) 292–301
[10] Kaelbling, L.P.: Learning in embedded systems. MIT press (1993)
[11] Kaelbling, L.P., Littman, M.L., Moore, A.P.: Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research **4** (1996) 237–285
[12] Donmez, P., Carbonell, J.G., Schneider, J.: Efficiently learning the accuracy of labeling sources for selective sampling. Proc. of KDD 2009 (2009) 259
[13] Newsome, J., Karp, B., Song, D.: Paragraph: Thwarting signature learning by training maliciously. In: International Workshop on Recent Advances in Intrusion Detection, Springer (2006) 81–105
[14] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: Security and Privacy (EuroS&P), 2016 IEEE European Symposium on, IEEE (2016) 372–387
[15] Babaioff, M., Sharma, Y., Slivkins, A.: Characterizing truthful multi-armed bandit mechanisms. In: Proceedings of the 10th ACM conference on Electronic commerce, ACM (2009) 79–88
[16] Biswas, A., Jain, S., Mandal, D., Narahari, Y.: 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. (2015) 1101–1109
[17] Tran-Thanh, L., Stein, S., Rogers, A., Jennings, N.R.: Efficient crowdsourcing of unknown experts using multi-armed bandits. In: European Conference on Artificial Intelligence. (2012) 768–773
[18] Xia, Y., Qin, T., Ma, W., Yu, N., Liu, T.Y.: Budgeted multi-armed bandits with multiple plays. In: Proceedings of 25th International Joint Conference on Artificial Intelligence. (2016)
[19] Xia, Y., Ding, W., Zhang, X.D., Yu, N., Qin, T.: Budgeted bandit problems with continuous random costs. In: Proceedings of the 7th Asian Conference on Machine Learning. (2015) 317332
[20] Watkins, C.J., Dayan, P.: Q-Learning. Machine Learning **8**(3) (1992) 279–292
[21] KhudaBukhsh, A.R., Xu, L., Hoos, H.H., Leyton-Brown, K.: Satenstein: Automatically building local search sat solvers from components. Artificial Intelligence **232** (2016) 20–42