Learning to Hire Teams

Adish Singla
ETH Zurich
adish.singla@inf.ethz.ch

Eric Horvitz
Microsoft Research
horvitz@microsoft.com

Pushmeet Kohli
Microsoft Research
pkohli@microsoft.com

Andreas Krause
ETH Zurich
krausea@ethz.ch

Abstract

Crowdsourcing and human computation are being employed in sophisticated projects that require the solution of a heterogeneous set of tasks. We explore the challenge of composing or hiring an effective team from an available pool of applicants for performing tasks required for such projects on an ongoing basis. How can one optimally spend budget to learn the expertise of workers as part of recruiting a team? How can one exploit the similarities among tasks as well as underlying social ties or commonalities among the workers for faster learning? We tackle these decision-theoretic challenges by casting them as an instance of online learning for best action selection with side-observations. We present algorithms with PAC bounds on the required budget to hire a near-optimal team with high confidence. We evaluate our methodology on simulated problem instances using crowdsourcing data collected from the Upwork platform.

Introduction

The success of a project or a collaborative venture depends critically on acquiring a team of contributors. Beyond increased performance and productivity, hiring a strong team can be important for effective collaboration, enhanced engagement, and increased retention of workers.

"A small team of A+ players can run circles around a giant team of B and C players." – Steve Jobs

Crowdsourcing via online marketplaces further underscores the promise of developing procedures for identifying potential contributors and composing teams, even when a job requester and workers may be half a world apart. To date, online crowdsourcing markets have largely focused on micro-tasking through enlisting non-expert crowd of workers who work independently on simple tasks such as performing image annotation. With the increasing complexity of tasks that are crowdsourced, as well as enterprises outsourcing their work, the need to hire skilled workers with an eye to considerations of complementarity and coordinative efforts in a collaboration around problem solving is becoming important. Contract-based crowdsourcing is another emerging paradigm where workers are recruited on a contract for performing tasks on an ongoing basis. Online
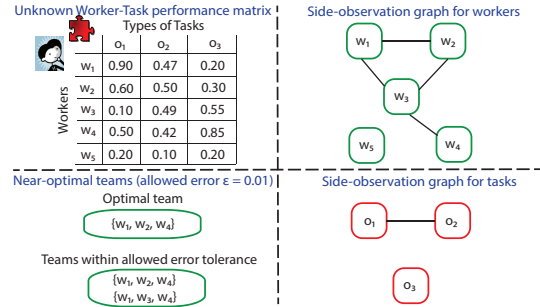
Figure 1: Illustration of the approach on toy example with five workers and three types of tasks.

platforms are offering new capabilities to deal with the rise of expertise-driven crowdsourcing. For example, the Upwork platform provides opportunities for workers to do self-assessments via taking tests on a voluntary basis. The platform provides support for recruiters to conduct interviews and perform online tests for job applicants.

Our Approach

Tasks, workers and the team. We consider the crowdsourcing setting where the job requester has a predefined heterogeneous set of types of tasks that need to be solved on an ongoing basis. For instance, consider an enterprise whose goal is to outsource a project that has three categories of tasks: (*i*) web development, (*ii*) English to Spanish translation, and (*iii*) video editing. The project would have ongoing assignments of tasks that would belong to one of these three categories. There are M types of tasks and N workers (*i.e.*, pool of job applicants) denoted by the sets $\mathcal{O} = \{o_1, \ldots, o_M\}$ and $\mathcal{W} = \{w_1, \ldots, w_N\}$, respectively. The recruiter's goal is to select or hire a team of workers denoted by $\mathcal{S}^*$, of size at most M from the set $\mathcal{W}$, comprising the highest performing worker for each task type $o \in \mathcal{O}$. When a new task needs to be executed, it is assigned to the hired team and can be performed by one of the team's workers.

Learning workers' expertise. In order to make informed hiring decisions, the recruiter needs to learn the workers' expertise for a given type of task. We model the performance of a worker for a given task as a bounded random variable, represented by an unknown performance matrix $\mu : N \times M \to \mathcal{R}_{\geq 0}$. In order to learn μ, the recruiter can perform an online *test* (*e.g.*, evaluate the performance of the worker for a task type via assignment of gold-standard question for which

the ground truth is available). Assigning task type $o_j \in \mathcal{O}$ to worker $w_i \in \mathcal{W}$ at time t yields a performance value (as feedback) denoted by random variable $X_{(i,j)}^t$, sampled *i.i.d.* from an unknown distribution with mean value $\mu_{(i,j)}$. The algorithm can repeatedly assign a task type o_j to worker w_i in order to reduce uncertainty and get a good estimate of the performance $u_{(i,j)}$.

Exploiting commonalities. Typically, the task types and the total number of job applicants could be large and hence may require performing large numbers of test assignments in order to learn workers' expertise. However, in order to speed up learning, we may be able to exploit the similarities among the tasks (*e.g.*, via group testing of two tasks) and underlying social ties or commonalities among the workers (*e.g.*, clustering based on demographics). We consider settings where the workers and tasks are embedded in two known underlying graphs, denoted by $G_w(V_w, E_w)$ and $G_o(V_o, E_o)$, respectively. The edges in these graphs capture the model of side-observations that may be possible to obtain at no additional cost (Mannor and Shamir 2011). In our model, when worker w_i is assigned task type o_j at time t, apart from observing the performance $X_{(i,j)}^t$, the following additional set of observations become available:

- $X_{(i,q)}^t \ \forall \ q : \{o_j, o_q\} \in E_o$, the additional observations associated with types of tasks neighboring o_j in G_o.
- $X_{(p,j)}^t \ \forall \ p : \{w_i, w_p\} \in E_w$, the additional observations associated with the workers neighboring w_i in G_w.

In Figure 1, assigning task type o_1 to worker w_2 at time t would yield set of observations given by $X^t = \{X_{(2,1)}^t, X_{(2,2)}^t, X_{(1,1)}^t, X_{(3,1)}^t\}$.

Budgeted hiring. The goal is to design algorithm that can efficiently learn the performance matrix $\mu[N, M]$ and output a near-optimal team. We measure the efficiency of such an algorithm in terms of the total number of tests required or equivalently the budget spent. In our model, a team $\mathcal{S}$ is ϵ-optimal, when, for each task type $o_j \in \mathcal{O}$, we have:

$$\forall o_j \in \mathcal{O}, \ \max_{w_i \in \mathcal{W}} \mu_{(i,j)} - \max_{w_i \in \mathcal{S}} \mu_{(i,j)} \leq \epsilon \qquad (1)$$

We seek algorithms with PAC bounds, *i.e.*, for given constants (ϵ, δ), the algorithm should output an ϵ-optimal team with probability of at least $(1 - \delta)$.

Algorithms for Budgeted Hiring

We now present several key insights in the design of algorithms UEXPSELECT and AEXPSELECT. Details, performance analyses, and experimental evaluation can be found in an extended version of this paper (Singla et al. 2015).

Algorithms UEXPSELECT and AEXPSELECT. We first consider the simple setting of hiring for one task type (*i.e.*, $M = 1$) in the absence of side-observations. We consider the recruiting of team members from among N workers as the set of actions at hand, and reduce the decision problem to the problem of best action selection in multi-armed bandits (MAB). Even-Dar, Mannor, and Mansour (2006) proposes a simple algorithm NAIVE for this problem based on uniform exploration of all of the actions and is the main building block for our proposed algorithm UEXPSELECT. However, such a uniform exploration ignores potential differences in the difficulties of assessing actions. To
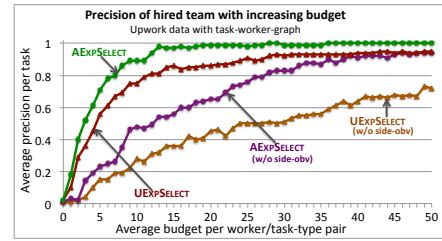

Figure 2: Results on Upwork data.

tackle this problem, Kalyanakrishnan et al. (2012) design an adaptive algorithm LUCB-1 using upper and lower confidence bounds. We use LUCB-1 as the main building block for our proposed algorithm AEXPSELECT. Next, we extend the algorithms to multiple types of tasks ($M > 1$) by jointly learning over all of these task types, and adaptively allocating the budget across them capturing the inherent hardness of best action selection problem per task type (Gabillon et al. 2011). Lastly, we extend our algorithms to account for side-observations. We have separate side-observation graph over tasks and over workers as discussed above. The proposed side-observation model can be jointly represented as the Cartesian product of two graphs given by $G_w \square G_o$, denoted as G_{wo}. Furthermore, the side-observation models have only been applied to regret minimization settings for MAB problems (Mannor and Shamir 2011) and we extend them to the best action selection settings by operating on the dominating set of the graph G_{wo}.

Experimental evaluation. We evaluated our algorithms via simulations using data from the Upwork platform. We consider as a primary metric the quality of the team generated by the algorithms for given budget measured through average precision per task—for a given output S, and any task type o_j, the precision for o_j is defined to be 1 if S contains an ϵ-optimal worker for o_j, else 0. Figure 2 shows the results for varying the average budget spent per worker/task-type pair. AEXPSELECT shows significantly faster convergence towards selecting the optimal team in comparison to UEXPSELECT. Also, the results demonstrate the effectiveness of the side-observations as we see a significant boost in terms of faster learning in both of the algorithms by exploiting the side-observations.

References

Even-Dar, E.; Mannor, S.; and Mansour, Y. 2006. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. In *JLMR*.

Gabillon, V.; Ghavamzadeh, M.; Lazaric, A.; and Bubeck, S. 2011. Multi-bandit best arm identification. In *NIPS*.

Kalyanakrishnan, S.; Tewari, A.; Auer, P.; and Stone, P. 2012. PAC subset selection in stochastic multi-armed bandits. In *ICML*.

Mannor, S., and Shamir, O. 2011. From bandits to experts: On the value of side-observations. In *NIPS*.

Singla, A.; Horvitz, E.; Kohli, P.; and Krause, A. 2015. Learning to hire teams (full version). http://arxiv.org/abs/1508.02823.