Joint Crowdsourcing of Multiple Tasks

Andrey Kolobov[*]
akolobov@microsoft.com
Microsoft Research
Redmond, USA, WA-98052

Mausam **Daniel S. Weld**
{mausam, weld}@cs.washington.edu
University of Washington
Seattle, USA, WA-98195

Introduction

Allocating tasks to workers so as to get the greatest amount of high-quality output for as little resources as possible is an overarching theme in crowdsourcing research. Among the factors that complicate this problem is the lack of information about the available workers' skill, along with unknown difficulty of the tasks to be solved. Moreover, if a crowdsourcing platform customer is limited to a fixed-size worker pool to complete a large batch of jobs such as identifying a particular object in a collection of images or comparing the quality of many pairs of artifacts in crowdsourcing workflows, she inevitably faces the tradeoff between getting a few of these tasks done well or getting many done poorly.

In this paper, we propose a framework called JoCr (**Jo**int **Cr**owdsourcing, pronounced as "Joker") for analyzing joint allocations of many tasks to a pool of workers. JoCr encompasses a broad class of common crowdsourcing scenarios, and we pose the challenge of developing efficient algorithms for it. In the settings modeled by JoCr, a customer needs to get answers to a collection of multiple-choice questions and has a limited worker pool at her disposal. Each question has a certain level of difficulty (possibly unknown) and each worker has a certain level of ability (possibly unknown as well). The chance of a worker answering a question incorrectly is an increasing function of question difficulty and decreasing function of worker skill, as described in (Dai, Mausam, and Weld 2010). Adopting the "consensus task" model of (Kamar, Hacker, and Horvitz 2012), we allow each question to be assigned to several workers, whose responses can be aggregated in order to increase the chance of correct answer. Last but not least, the framework admits constraints on the number of questions that can be allocated per person that prevent any given worker from getting overwhelmed. We describe several possible optimization problems formalized by JoCr and suggest promising methods for solving them approximately.

The JoCr Framework

The JoCr framework we put forth here models the assignment of a set $\mathcal{Q}$ of questions to a pool $\mathcal{W}$ of workers. In scenarios that motivated JoCr, the questions are roughly of the same kind, e.g., "Does the area in satellite image I contain a missile launch pad?", but JoCr does not require this explicitly. However, it assumes that the customer pays any worker the same amount of money c for answering any question and receives some fixed reward λ from an external source for getting a question answered correctly. In addition, JoCr stipulates that each question q should have a reasonably small set of possible answers $\mathcal{A}_q$, which is the case for many crowdsourcing tasks, from identifying objects in images to multiple-choice questions. This assumption makes it feasible to compile several workers' responses to a question in order to gain insights about the question's true answer.

JoCr models workers' responses to assigned questions as being probabilistic in nature. Indeed, workers are not guaranteed to answer a given question correctly. Rather, their accuracy depends on their ability, mood, motivation, and many other intrinsic characteristics, as well as question difficulty. JoCr describes worker w's qualities relevant for perfoming a task with a *skill* parameter $\gamma_w \in [0, 1]$. Similarly, for a question q, its difficulty is captured by a parameter $d_q \in [0, 1]$. JoCr assumes that a worker's probability of answering a specific question correctly is a function $p(d_q, \gamma_w)$ of only these two variables. One such function has been proposed in (Dai, Mausam, and Weld 2010). Crucially, JoCr does not assume that either γ_w's or d_q's are known — typically, they have to be learned. The necessity to learn them is the reason for characterizing worker skill with a single number, since this parametrization is likely to be trainable even with meager data quantities.

Last but not least, JoCr imposes a constraint M_w on the maximum number of questions that can be assigned to a worker, in order to ensure that no single person gets overwhelmed by assignments and can complete them promptly.

To formalize the above intuitions, we define an *instance* of JoCr as a tuple $F = \langle \mathcal{Q} = \{\langle q, d_q, \mathcal{A}_q \rangle\}, \mathcal{W} = \{\langle w, \gamma_w, M_w \rangle\}, c, \lambda \rangle$. Letting $S_{q,w}$ be an indicator variable that has value 1 iff question q has been assigned to worker w, a solution to a JoCr instance F is a setting $\mathcal{S}$ of variables $S_{q,w}$ for all questions q and workers w that obeys $\sum_{q'} S_{q',w} \leq M_w$ for every worker.

JoCr Solution Concepts and Their Optimality

JoCr's probabilistic model of worker accuracy suggests several interesting optimization objectives and ways of constructing question-to-worker assignments to meet them.

[*] The author did part of the work for this paper while being a Ph.D. student at the Dept. of CSE of the University of Washington.

First, recall that a single worker's response to question q is likely to be only an estimate of q's true answer. However, if priors over γ_w, d_q, and $\mathcal{A}_q$ are available and $|\mathcal{A}_q|$ is reasonably small, assigning question q to an additional worker w can give significant (although generally incomplete) extra information about a correct answer to q. Therefore, in JoCr, allocating a given question to *several* workers and then aggregating the results is usually a worthwhile strategy. While assigning an easy question to a few low-skilled workers is likely sufficient, hard questions can take many proficient workers to produce a reliable answer. Moreover, note that the decision whether to assign a question to more workers generally depends on the responses of the already hired ones. Therefore, although allocations can be static, hiring additional workers *dynamically*, based on the available answers, can be more effective. At the same time, the choice between static and dynamic assignment construction may be dictated by the specifics of the crowdsourcing platform, so both types of solutions need to be considered.

So far, we have ignored the issue of what constitutes an *optimal* solution to a JoCr instance. One reasonable objective function for solutions to maximize is the expected reward less the total cost paid to the workers. More specifically, let $R_{q,w}|S_{q,w}$ be a random variable over worker w's answers to question q if q has been assigned to w. As mentioned above, different workers' responses to q can be aggregated into a posterior over q's answers. Let a function $f(r_1, \ldots, r_{|\mathcal{W}|})$ be such an aggregator, and suppose we estimate the correct answer a^*_q to q by returning the MAP value of f, i.e., by computing $\hat{a}^*_q =$ MAP$[f(R_{q,w_1}|S_{q,w_1}, \ldots, R_{q,w_{|\mathcal{W}|}}|S_{q,w_{|\mathcal{W}|}})]$. In this case, the utility of any solution $\mathcal{S}$ (a setting of variables $S_{q,w}$) can be defined as $U(\mathcal{S}) = \lambda\mathbb{E}[\sum_q \mathbb{I}_{a^*_q == \hat{a}^*_q}] - c\sum_{q,w} S_{q,w}$. For an optimal assignment $\mathcal{S}^*$ this utility is the highest.

Approaches to Solving JoCr

In general, JoCr instances can be analyzed under several sets of assumptions, with either a static or dynamic allocation sought in each case:

Known question difficulty, known worker skill. Under these conditions, a promising technique for both a static and a dynamic allocation would be incrementally constructing a solution by assigning questions to workers so that each question's assignment results in a partial allocation of the largest utility without violating the per-worker allocation limits M_w. It is an open research question whether this schema yields a solution with optimality guarantees, as it does for the seemingly related problems such as knapsack.

Question difficulty, worker skill, or both are unknown. All JoCr instances with these characteristics intoduce an exploration-exploitation tradeoff, since the unknown parts of the model need to be learned in order to come up with a good question assignment. Resolving this tradeoff is the core focus of the literature on reinforcement learning (Sutton and Barto 1998) and multi-armed bandits (MAB) (Auer, Cesa-Bianchi, and Fischer 2002). In particular, in the setting with unknown worker skill but known question difficulty, each worker can be viewed as a MAB's "arm" whose reward distribution is parametrized by the worker's skill. Nonetheless, solving such JoCr instances may be significantly more difficult than vanilla MAB problems, since the number of allowed arm "pulls" is limited by the per-worker allocation constraints M_w.

Relation to Existing Crowdsourcing Models

Perhaps the most closely related crowdsourcing model to JoCr is the Generalized Task Markets (GTM) framework (Shahaf and Horvitz 2010). GTM models the process by which decomposable high-level tasks can be solved by a crowd whose labor is organized in a workflow. The principal distinction of GTM from JoCr is GTM's very high-level specification of tasks and workers that prevents designing a general algorithm for constucting task allocations. JoCr, by making explicit a probabilistic dependence of workers' performance on task difficulty, allows its instances to be solved with a small set of problem-independent techniques.

Another setting that, similar to JoCr, requires assignment of multiple tasks to multiple workers and explicitly considers the possibility of worker skill being unknown has been studied in (Tran-Thanh et al. 2012). It uses MAB theory to construct a task allocation while learning model parameters. However, it assumes that each task can only be assigned to one worker, which suggests that the allocation problem considered there is easier than JoCr instances with unknown parameters.

Conclusion

We have proposed a framework called JoCr for analyzing joint allocation of a set of tasks of varying difficulty to a set of workers with diverse skill levels. JoCr allows the model parameters to be unknown and assumes that responses of different workers to the same question can be aggregated to better approximate a correct answer. This provides JoCr instances with enough structure to be potentially solvable with problem-independent algorithms and, at the same time, to describe many common crowdsourcing scenarios, such as identifying an object in a collection of images. In the future, we will design efficient approximation techniques for JoCr and show their merit on practical crowdsourcing problems.

Acknowledgments. This research has been funded by WRF/TJ Cable Professorship, ONR grant N00014-12-1-0211, and NSF grants IIS 1016713 and IIS 1016465.

References

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multi-armed bandit problem. *Machine Learning* 47(2-3):235–256.

Dai, P.; Mausam; and Weld, D. S. 2010. Decision-theoretic control of crowd-sourced workflows. In *AAAI'10*.

Kamar, E.; Hacker, S.; and Horvitz, E. 2012. Combining human and machine intelligence in large-scale crowdsourcing. In *AAMAS'12*.

Shahaf, D., and Horvitz, E. 2010. Generalized task markets for human and machine computation. In *AAAI'10*.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

Tran-Thanh, L.; Stein, S.; Rogers, A.; and Jennings, N. R. 2012. Efficient crowdsourcing of unknown experts using multi-armed bandits. In *ECAI'12*.