Title	Know when to run : recommendations in crowdsourcing contests
Author(s)	Mo, Jiahui; Sarkar, Sumit; Menon, Syam
Citation	Mo, J., Sarkar, S., & Menon, S. (2018). Know When to Run: Recommendations in Crowdsourcing Contests. MIS Quarterly, 42(3), 919-944. doi:10.25300/MISQ/2018/14103
Date	2018
URL	http://hdl.handle.net/10220/46105
Rights	© 2018 Management Information Systems Research Center. This paper was published in MIS Quarterly and is made available as an electronic reprint (preprint) with permission of Management Information Systems Research Center. The published version is available at: [http://dx.doi.org/10.25300/MISQ/2018/14103]. One print or electronic copy may be made for personal use only. Systematic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper is prohibited and is subject to penalties under law.

MIS Quarterly

RESEARCH ARTICLE

KNOW WHEN TO RUN: RECOMMENDATIONS IN CROWDSOURCING CONTESTS[1]

Jiahui Mo
Nanyang Business School, Nanyang Technological University, SINGAPORE {jhmo@ntu.edu.sg}

Sumit Sarkar and Syam Menon
Naveen Jindal School of Management, University of Texas at Dallas, Richardson, TX 75080 U.S.A.
{sumit@utdallas.edu} {syam@utdallas.edu}

Crowdsourcing contests have emerged as an innovative way for firms to solve business problems by acquiring ideas from participants external to the firm. As the number of participants on crowdsourcing contest platforms has increased, so has the number of tasks that are open at any time. This has made it difficult for solvers to identify tasks in which to participate. We present a framework to recommend tasks to solvers who wish to participate in crowdsourcing contests. The existence of competition among solvers is an important and unique aspect of this environment, and our framework considers the competition a solver would face in each open task. As winning a task depends on performance, we identify a theory of performance and reinforce it with theories from learning, motivation, and tournaments. This augmented theory of performance guides us to variables specific to crowdsourcing contests that could impact a solver's winning probability. We use these variables as input into various probability prediction models adapted to our context, and make recommendations based on the probability or the expected payoff of the solver winning an open task. We validate our framework using data from a real crowdsourcing platform. The recommender system is shown to have the potential of improving the success rates of solvers across all abilities. Recommendations have to be made for open tasks and we find that the relative rankings of tasks at similar stages of their time lines remain remarkably consistent when the tasks close. Further, we show that deploying such a system should benefit not only the solvers, but also the seekers and the platform itself.

Keywords: Competition, performance, winner prediction, probability models, rankings

Introduction

Crowdsourcing contests have emerged as innovative ways for firms to solve business problems. In a typical contest, a firm (the *seeker*) posts the requirements for a task to be solved on a crowdsourcing contest platform, along with information on the time frame for the contest and the prize money the winner will receive. Contests are usually open to anyone who wishes to participate (the *solvers*); the firm evaluates solutions and picks a winner once the deadline is over. Typically, a single winner receives the entire prize money.

Successful outcomes for both seekers and solvers have resulted in crowdsourcing platforms becoming quite popular, and the number of tasks posted on such platforms has increased rapidly. For example, the number of contests handled by *99designs* (a popular crowdsourcing platform) went from about 100,000 in 2012 (Lacy 2012) to 423,000 by July 2015 (*99designs.com*). Roth et al. (2016) report that Unilever's *Foundry IDEAS*, a hub to organize Unilever's crowdsourcing briefs, will "increase its use of crowdsourced ideation tenfold by 2020."

[1]Jeffrey Parsons was the accepting senior editor for this paper. Huimin Zhao served as the associate editor.

As crowdsourcing platforms attract more seekers and solvers, the number of open tasks at any point in time can become quite large. For instance, some categories on *99designs* have around 1,000 open tasks at a time, with a typical task staying open for 4 to 7 days. Thus, a solver visiting the platform has to evaluate a large number of tasks to identify one in which to participate. Kucherbaev et al. (2014) note that the majority of solvers spend over 25% of their time searching for tasks. As solvers need to complete tasks in a short amount of time (ranging from a few hours to a few days), spending a substantial amount of time and effort just to identify tasks in which to participate can have a significant impact, both on the number of tasks in which they can participate and on the quality of solutions they can provide.

Platforms usually list open tasks sorted along a predetermined dimension; one common approach is to list tasks based on their posting times, with the most recently posted task appearing on top. Platforms may also provide alternative criteria on which a solver can sort, such as the the prize amount or the time remaining before a task will close. While such options provide solvers some flexibility, it remains very difficult, if not impossible, for them to identify and examine in detail all tasks in which they could be interested. Consequently, solvers end up searching through a limited number of tasks before deciding on the one(s) in which to participate, often examining only those tasks that appear on the first couple of pages of the task listings (Chilton et al. 2010).

This abbreviated search behavior can have serious implications for all concerned. Solvers can miss out on tasks for which they are better suited, just because they are harder to find (Ambati et al. 2011); this hurts the solver's chances of winning and earning money. Further, this can lead to seekers receiving solutions of poorer quality compared to when solvers can find tasks for which they arebest suited. Indeed, seekers may not even receive the minimum number of submissions that makes the exercise worthwhile, and many platforms refund the prize money in such situations (e.g., *Zhubajie*, *DesignCrowd, 99designs*). Dissatisfied seekers may leave the platform, which could lose revenues not just from current tasks, but also from future business opportunities.

Recognizing these concerns, some platforms have started providing recommendations to solvers in order to reduce their search costs. These recommendations are typically based on systems deployed for e-commerce sites. For example, *Freelancer* recommends tasks whose requirements best match the solver's skills (which are usually explicitly provided by the solvers when they register at the site). *Zhubajie* recommends tasks that are most similar (based on some platform-determined metric) to closed tasks in which the solver has participated.

Although solvers may participate in crowdsourcing platforms for a variety of reasons, their primary motivation is to win (Brabham 2010; Lakhani et al. 2007). However, it is not easy for solvers to find winnable tasks. This is because winning a task does not depend on one solver's performance alone; every other solver participating in the task is in it to win, and their performance will impact the success chances of a solver considering participation in the task. Information about competitors is often not available to solvers, and even if it were, they would find it difficult to assess the abilities of all competing solvers for each open task. Therefore, any guidance to increase a solver's chances of winning would be worthwhile for everybody involved: solvers are more likely to win contests, seekers are more likely to get better solutions, and the platform is likely to earn more money and retain more solvers and seekers.

We propose a framework to recommend tasks that a solver is most likely to win, given the current set of participants in open tasks. A key component of the framework is an approach to determine the probability that a solver who has just logged into the system (the *focal* or *target* solver) will win an open task if she participates in it; this probability would depend on her performance. The existence of competition on crowdsourcing contest platforms provides an interesting contrast to domains where traditional recommendation methodologies are employed. Traditional systems rely on the characteristics of users and products to make recommendations. While this is appropriate where multiple users can buy the same product, typically only one of the participants can win in a crowdsourcing contest. This is a fundamental difference, and a recommendation system tailored for a crowdsourcing contest environment should consider not only the performance of a focal solver, but also that of other solvers participating in a task.

The predictive ability of a model depends on the nature of information available to it. Information can originate from various sources, with the characteristics of the solvers and the tasks being two important ones. In our context, another relevant information source is the way platforms enable seekers to rank solvers and select winners. We consider three ranking mechanisms (scenarios) that capture the prominent winner selection approaches. In one case, the seeker selects a winner, providing no additional ranking information. At the other extreme, the platform requires seekers to rank every submitted entry. The third scenario lies in between, with the seekers shortlisting solvers before eventually choosing a winner. We consider several prediction models under each scenario, and identify those that best capture available information in each case. For a target solver, once winning probabilities are estimated for each task, a desired number of tasks can be recommended by comparing these probabilities (or expected payoffs).[2]

We summarize below the research questions we address, along with related contributions.

1. *What is the theoretical basis for identifying inputs to a recommender system for a crowdsourcing contest environment?* We identify Campbell's (1990) model of performance as the theoretical foundation for identifying inputs to a recommender system. We reinforce it with established theories from learning, motivation, and tournaments[3] to better explain the roles of the different predictors of performance.

2. *How can competition, an integral part of crowdsourcing contests, be factored into recommendation methodologies?* We use the reinforced version of Campbell's theory to identify the characteristics of solvers participating in a task that could impact a focal solver's winning probability. We then identify prediction models that can be used, and show how they can be adapted to incorporate the competitive aspects of the different scenarios. These models predict the relative performance of solvers within a task, thereby enabling us to capture competition.

3. *How useful is this framework expected to be in real-world environments?* We validate the framework (and associated methodologies) by collecting task and solver data from a live crowdsourcing platform (*99designs*) and conducting a broad set of experiments.

    (a) *How do the winner prediction models perform?* We compare the performance of the adapted models to two benchmarks. We first establish that all of the models presented in the paper have value as they are significantly better than selecting winners randomly. We then show that these models generally perform better than a benchmark which uses a metric that the platform itself views as the best measure of a solver's abilities.

    (b) *How does the recommender system compare with the solvers' own task choices?* We show our approach has the potential to help solvers improve their success rates substantially. This result holds across the board, irrespective of the solvers' current success rates.

    (c) *Given that tasks would be recommended while they are open, how reliable would these recommendations turn out to be after the tasks close?* We examine whether the existing competition for different open tasks at the same stage in their overall time lines are representative of the eventual competition that a target solver will face at the time the tasks close. We find the likelihood of the solver winning a task *relative to another* while the tasks are open is not very different from when the tasks close.

4. *What is the potential impact to the seekers, and to the platform itself?* We conduct a simulation experiment to show that seekers, on the whole, should benefit from the use of the system. The platform also benefits, both in terms of expected revenues and in terms of the potential improvement in solvers' and seekers' satisfaction with the site. Thus, all stakeholders (solvers, seekers, and the platform) should benefit from such a system.

Literature Review

We review relevant literature on recommender systems, crowdsourcing, and winner prediction in competitive environments.

Recommender Systems

Recommender systems have been studied extensively over the years, and many books (e.g., Aggarwal 2016) and surveys (e.g., Bobadilla et al. 2013) exist on the topic. In a recent review, Li and Karahanna (2015) categorize the literature on recommender systems into three streams: one that focuses on understanding the customer, another that develops approaches to generate recommendations, and a third that evaluates the impact of deployed systems.

Understanding the customer usually involves building user profiles to capture their preferences (Adomavicius and Tuzhilin 2001). The information needed to build profiles can come from a variety of sources, including demographic data, browsing history (e.g., Atahan and Sarkar 2011), consumer reviews (e.g., Chen et al. 2015), and social networks (e.g., Arazy et al. 2010; Li et al. 2017). Domain-specific aspects and contextual information can also be important (Adomavicius and Tuzhilin 2015); for instance, Provost et al. (2015)

[2]Probabilities are appropriate if different tasks had the same prize amounts, as this is equivalent to comparing expected payoffs. When the prize amounts for the tasks are different, expected payoffs could be used directly.

[3]A tournament is a contest where participants compete for a prize, and is designed to encourage an optimal level of effort from the participants (Becker and Huselid 1992).

use a *geosimilarity network* to target advertisements in the context of mobile networks. However, prior work on recommender systems has not examined competitive environments. Given this unique context, the solver profiles we develop are novel and based on extant theories of performance.

The generation of recommendations involves effectively delivering offerings (e.g., items, tasks) to customers. Aggarwal (2016) classifies approaches to identify recommendations into various categories. Content-based methods (e.g., Balabanovic and Shoham 1997) use item attribute descriptions to make recommendations. These approaches have roots in information retrieval (Manning et al. 2008), where similarity-based approaches have been extensively used. Collaborative methods leverage other users with similar preferences (Goldberg et al. 1992) through two main approaches: memory- (or neighborhood-) based, and model-based methods. Memory-based methods include user- and item-based collaborative filtering where ratings of similar people, or on similar items, are used to make recommendations (Breese et al. 1998). Model-based methods impose a model on the data and estimate the parameters of the imposed model, for example, graph based models (Banerjee et al. 2016; Huang et al. 2007), matrix factorization approaches (Koren et al. 2009), and hidden Markov models (Sahoo et al. 2012). Utility-based systems, where probabilities are computed to estimate the extent to which a user will like the recommended item, form an important class of knowledge-based methods. For example, Ghoshal et al. (2015) use such an approach to combine association rules and improve recommendation quality. Our framework incorporates a utility-based approach to make recommendations in the novel competitive environment. We show how machine learning and econometric techniques can be adapted to estimate a solver's *probability of winning*. This metric has not been considered in prior works on recommender systems.

The third category of research examines the effect of recommender systems on the behavior of consumers and on the market. Studies include the impacts on product demand (Lin et al. 2017; Oestreicher-Singer and Sundararajan 2012b), consumer satisfaction (Ho et al. 2011), trust (Komiak and Benbasat 2006), and sales diversity (Oestreicher-Singer and Sundararajan 2012a). We evaluate the potential impact of our framework on solution quality and task success.

The vast majority of the literature on recommender systems has been for traditional e-commerce environments. Some recent work has extended content-based or collaborative approaches to contexts beyond typical e-commerce settings, such as micro-task crowdsourcing environments (Ambati et al. 2011; Yuen et al. 2012).

A critical distinction between micro-task platforms and crowdsourcing contest platforms is that micro-task platforms are noncompetitive environments whereas a crowdsourcing contest involves solvers who compete with each other wherein only one of them will eventually be the winner. Solvers in such environments are usually interested in tasks that maximize their odds of winning (Brabham 2010). A solver's chance of winning a crowdsourcing contest is greatly influenced by the other solvers participating in the task, and traditional recommendation systems (whether for e-commerce or for micro-task platforms) do not capture this competitive aspect of crowdsourcing contests. Our approach is tailored for contest environments.

Crowdsourcing

There are many types of crowdsourcing applications, with micro-task platforms (e.g., Amazon's *Mechanical Turk*), citizen science based platforms (e.g., *iSpot*), crowdsourcing contests (e.g., *TaskCN*, *99designs*, *InnoCentives*), and crowdsourcing ideation (e.g., *Dell IdeaStorm*) being the best known. Our research focuses on crowdsourcing contests.

Research on crowdsourcing contests falls into several streams. One deals with designing mechanisms and builds on the economic literature on tournaments (e.g., Moldovanu and Sela 2001), examining aspects such as prize structures (e.g., one winner or multiple winners), feedback mechanisms (with feedback or without) and submission rules (public or hidden submissions). For example, Terwiesch and Xu (2008) build an analytical model to examine the type of innovation problems most suited to crowdsourcing contests and how to design the reward structure optimally. Archak and Sundararajan (2009) investigate the effects of multiple reward structures on the strategic behavior of both risk-neutral and risk-averse contestants.

Another stream explores solvers' incentives, such as their motivation to participate and invest different levels of effort. Brabham (2010) identifies winning opportunity as the primary reason solvers participate in tasks. Yang et al. (2009) empirically show that seeker feedback can encourage solvers to put in more effort. Boudreau et al. (2011) find that greater rivalry reduces the incentives of competitors to participate in a contest, but increases the likelihood that at least one competitor will provide a particularly good solution. Liu et al. (2014) show, in an experimental setting, that a higher reward induces more submissions of higher quality.

A third stream examines the submission strategies of solvers during the course of a task, and the impact of these strategies on their chances of winning. Yang et al. (2011) find that solvers who submit early or late in a task's duration tend to win more than other solvers; they also find a solver's past experience to be a good predictor of future winning probability. Bockstedt et al. (2015) investigate how solver diversity in terms of national wealth and culture impacts the number of submissions. Bockstedt et al. (2016) estimate a solver's chances of winning by viewing three dimensions of submission behavior: (1) the position of a solver's first submission (measured as the number of submissions to the task prior to the solver's first submission), (2) the number of submissions made by a solver, and (3) the length of active participation by a solver (measured as the total number of submissions between the solver's first and last submissions). These papers look at submission behavior once a solver has joined a task, and provide insights on how solvers' submission behaviors impact their chances of winning.

Existing research on crowdsourcing contests has focused mainly on solver submission behavior and the design of tasks; there is very little work that looks at the pre-submission stage, where solvers decide on tasks to join. Our paper is related more closely to task selection, as we identify potential tasks for solvers. In this context, Yang et al. (2008) find that solvers on *TaskCN* tend to select tasks with fewer rivals to increase their winning probability; at the same time, solvers also tend to select tasks with higher prizes. However, these two criteria can conflict with each other, as a higher prize will attract more solvers, potentially impeding winning. Therefore, effectively selecting tasks in which to participate remains important yet difficult. In an effort to explain the relationship between task prize and solver participation, DiPalantino and Vojnovic (2009) build an analytical model to examine choices of strategic solvers. They find that solvers with the highest skill levels should select tasks that offer the highest rewards. Schulze et al. (2012) propose a conceptual framework to explore solvers' task selection patterns that include prize money, enjoyment, and time. Our work is different from prior research on task selection. While existing research tries to understand how solvers choose tasks and identify factors that can influence this choice, we focus on finding tasks that platforms can recommend to solvers based on the estimated probability (or the expected payoffs) of their winning the tasks.

Winner Prediction in the Presence of Competition

Predicting winners in the face of competition occurs in several domains. Three areas in particular where winners are chosen after some form of competition are politics, gambling, and sports.

Models to forecast election results are primarily based on economic theory (e.g., Cameron and Crosby 2000; Hummel and Rothschild 2014), opinion polls like ACNielsen (e.g., Erikson and Wlezien 2012), or prediction markets (e.g., Chen et al. 2008; Forsythe et al. 1992). Models that make predictions in a betting context have been studied for a long time (e.g., Edwards 1955). However, most of the prediction models relevant to our context come from the sports literature. Consequently, we focus on winner prediction in the specific context of sports.

Numerous papers have investigated predicting success in sports, with a variety of models used. Winner prediction in horseracing has been examined using multinomial logit models (e.g., Bolton and Chapman 1986). Logistic models have also been effectively used in predicting the outcomes of tennis matches (Clarke and Dyte 2000), boxing bouts (Lee 1997), and football games (Willoughby 2002). To predict the playoff performance of teams in the National Hockey League, Demers (2015) considers support vector machines (SVMs), in addition to logit and probit models, and finds them to outperform the other models in their context. SVMs have also been used successfully in the context of horse racing (e.g., Lessman et al. 2012) and to predict the outcomes of NCAA bowl games (Delen et al. 2012). Naïve Bayes classifiers have been found effective in predicting Cy Young award winners (Smith et al. 2007), while Constantinou et al. (2012) develop a Bayesian network to forecast the outcomes of games in the English Premier League. Loeffelhold et al. (2009) and David et al. (2011) predict the outcomes of basketball and football games respectively using neural networks.

For our framework, any model that can be used to predict winners is a potential candidate for inclusion. We should point out that the models need to be adapted appropriately to our context. For instance, current research on winner prediction in horseracing only considers information on winners and non-winners and ignores other potentially useful rank information among non-winners. We consider variants of the models that can capture the detailed information contained in the ranks provided by seekers.

A Recommendation Framework for Crowdsourcing Contests

Our context is one where a crowdsourcing platform wishes to provide recommendations to a solver when the solver visits the platform's website. We assume that the solver has already registered at the site and provided information as required by the platform. This information typically includes, along with personal identifying information, the skill sets of the solver.

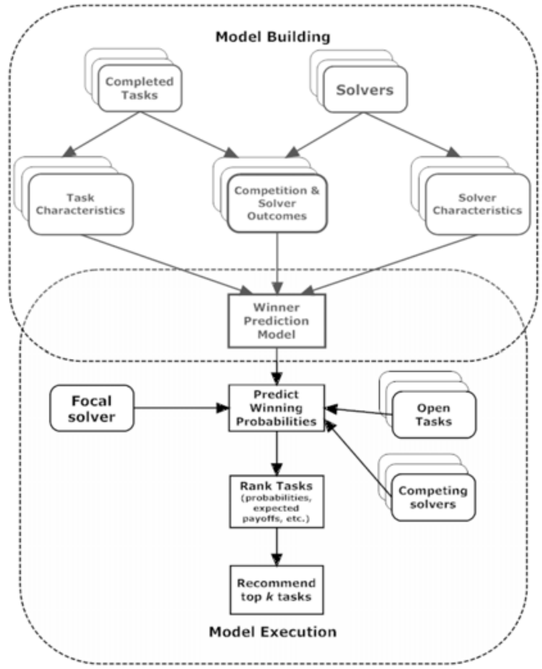

Figure 1. Proposed Framework

For example, a solver interested in designing logos may list formal training in graphic design. When the solver logs on to the site, the platform could provide, along with the default ordering of open tasks, the tasks recommended for the solver based on the predicted probability of her winning those tasks or, alternatively, her expected payoff.

We first establish a strong theoretical foundation for our context. As the success probabilities depend on solver performance, we begin by defining performance. We choose Campbell's (1990) model of performance as the basis for identifying predictors of performance, and reinforce it with theories of learning, motivation, and tournaments. There are three major components to recommendation system design: input, process, and output (Xiao and Benbasat 2007). Our framework incorporates all three components to different extents, and comprises two parts: *model building* and *model execution* (Figure 1). In the model building stage, we rely on the reinforced version of Campbell's model to identify inputs available to a crowdsourcing platform that could help predict the probability that a focal solver will win a specific task. We then leverage the literature on winner prediction to examine various models that could be used to estimate such probabilities, given the inputs available. We show how prediction models may need to be adapted to predict winning probabilities in competitive environments.

The model execution part illustrates how the recommender system would be deployed when a solver logs in. This includes determining, for each open task, the probability that the solver would win that task if she participated in it. Once these probabilities are obtained for all open tasks, the tasks could be ranked based on either the predicted winning probability or the expected payoffs. The system could then recommend the top k ranked open tasks to the target solver.

Model Building: Identifying Inputs

The probability of a solver winning a contest depends on her performance. Campbell et al. (1993) define performance as "goal-relevant actions that are under the control of the individual" (p. 40). Many theories on performance exist (e.g., Campbell 1990; Porter and Lawler 1968). The theory proposed by Campbell stands out, as it is well-established, quite general, and identifies the determinants of performance, along with a comprehensive set of predictors for these determinants. We note that while some variables that could potentially influence winning probabilities have been discussed previously in the literature on crowdsourcing contests, there has been no research to date that systematically assembles the predictors discussed here.

Theoretical Foundations and Determinants of Performance

Campbell views performance as a function of three major determinants: *declarative knowledge*, *procedural knowledge and skill*, and *motivation*. Declarative knowledge represents the individual's understanding of task requirements, while procedural knowledge and skill represents the combination of declarative knowledge with the ability to perform a task (Anderson 1985; Kanfer and Ackerman 1989). In other words, declarative knowledge is knowing what needs to be done, and procedural knowledge is knowing how to do it. According to Campbell, motivation is the combined effect of three "choice behaviors": the choice to perform, the choice of a level of effort to expend, and the choice of how long to expend that level of effort. Campbell's model is particularly convenient as it identifies potential predictors for the determinants. While the basic model focuses on the three main determinants, Campbell recognizes that situational effects can also affect performance.

In Campbell's model, predictors for declarative knowledge and procedural knowledge and skill can be related either to an individual's personal traits or to external factors. Personal traits such as ability, personality, and interests influence an individual's declarative knowledge and procedural knowledge and skill. These intrinsic characteristics can impact performance by influencing an individual's potential to acquire knowledge. On the other hand, both declarative knowledge and procedural knowledge and skill can also be affected by external factors such as an individual's levels of education, training, and previous experience.

While Campbell's model helps identify predictors for the determinants of performance, it does not explain precisely how these predictors influence performance. We supplement this model using theories of learning, motivation, and tournaments to establish the potential reasons behind these relationships. Theories of learning proposed by Cyert and March (1963), and extended by Schwab and Miner (2008) suggest that individuals learn from the outcomes of various activities, repeating activities with positive outcomes and avoiding those whose outcomes are negative. This selective replication of actions based on past performance enhances procedural knowledge and skill, thereby improving future performance. Learning can result from such experience or through training or education, or by being exposed to the actions of others.

Theories of motivation (e.g., Porter and Lawler 1968; Vroom 1964) point to two categories of predictors of motivation: intrinsic and extrinsic rewards. Intrinsically motivated behavior has "no recognizable reward except the activity itself" (Akin-Little et al. 2004, p. 345) and relate to the individual's positive experiences from completing the task, such as satisfaction and a sense of achievement. Extrinsically motivated behavior refers to "behavior controlled by stimuli external to the task" (Akin-Little et al. 2004, p. 345) (e.g., bonuses, pay increases). Intrinsic and extrinsic rewards could influence the effort an individual devotes to a task, and thereby influence performance.

The determinants and predictors of performance are summarized in Table 1. These are general in nature, and have been applied previously to different environments to examine the quality of the predictors (e.g., McCloy et al. 1994). Campbell notes that the measures of these predictors are likely to be different for different environments and, therefore, they need to be evaluated in the relevant environment.

In the following discussion, we consult existing literature to discuss predictors relevant to crowdsourcing contests, and explain how they influence performance. While they may be relevant, crowdsourcing contest platforms, including the one we study in this paper, usually do not (and perhaps cannot) track information on some of the predictors, including personality, interests, education, training, practice, and intrinsic motivation. Therefore, we do not discuss them further.

Ability

Ability has been viewed along two dimensions: intrinsic ability inherent to an individual (Hunter 1986) and expertise acquired through various means (Faraj and Sproull 2000).

Intrinsic Ability. Intrinsic ability refers to talent or general aptitude. Every individual is endowed with some level of ability that can be used at no cost (Terwiesch and Xu 2008). While expertise can be acquired (e.g., by learning design skills and tools), the ability to create a new and innovative artistic product is innate to the individual, and cannot be learned easily. Intrinsic ability enables individuals to learn by linking new information to knowledge already in memory (Hunter 1986). As people with high intrinsic ability can make such connections quickly, intrinsic ability affects how much and how quickly individuals learn (Hunter 1986). Talented individuals master requisite knowledge faster than those with lower intrinsic ability (Simonton 2000). Therefore, higher levels of ability are likely to be associated with higher levels of knowledge and, consequently, performance. The literature on tournaments reinforces this perspective—that ability affects performance—and bolsters its relevance in competitive environments (Dechenaux et al. 2015).

<table>
<tr><th colspan="3">Table 1. Performance Determinants and Predictors</th></tr>
<tr><th colspan="2">Determinants</th><th>Predictors</th></tr>
<tr><td rowspan="7">Procedural Knowledge and Skill</td><td rowspan="6">Declarative Knowledge</td><td>Ability</td></tr>
<tr><td>Personality</td></tr>
<tr><td>Interests</td></tr>
<tr><td>Education</td></tr>
<tr><td>Training</td></tr>
<tr><td>Experience</td></tr>
<tr><td></td><td>Practice</td></tr>
<tr><td colspan="2" rowspan="2">Motivation</td><td>Extrinsic Motivation</td></tr>
<tr><td>Intrinsic Motivation</td></tr>
<tr><td colspan="2">Environment</td><td>Work Conditions</td></tr>
</table>

Expertise. Expertise is context-specific knowledge resulting from patterned interactions and practices in specific scenarios (Faraj and Sproull 2000). It also refers to the proficiency needed to perform a task that is developed through practice: it is competence acquired from prolonged exposure to a specific area. Expertise can lead to superior performance (Ericsson 2006). People with high levels of expertise tend to have more context-specific knowledge, and are likely to be proficient at solving problems in that domain (Doane et al. 1990). Such individuals can also select appropriate actions more consistently than novices as their expert knowledge allows them to identify and maintain access to relevant information easily (Ericsson et al. 2000).

Experience

Experience has many dimensions, and we focus on four key ones: overall (i.e., general) experience, specialized experience, diverse experience, and complexity of experience.

General Experience. Quiñones et al. (1995) define experience as events "in an individual's life that are perceived by the individual" (p. 890). Experience can be good indicator of future performance (McDaniel et al. 1988). Huckman et al. (2009) note that past experiences allow individuals to develop routines to solve problems. Schmidt et al. (1986) find that experience directly impacts job knowledge, which in turn is a strong determinant of performance.

Specialization. Specialization is the accumulation of experience in a specific area. Specialized experience increases an individual's familiarity with the area, and could improve performance. In particular, extensive experience in a domain has been found to be necessary to reach high levels of performance (Ericsson 2006). Specialization in an area has the potential to lead to expertise in that area (Ericsson 2006). From the perspective of learning, specialization makes it easier for individuals to transfer skills or knowledge from prior tasks to other similar tasks (Narayanan et al. 2009). Specialization also allows individuals to complete more repetitions of similar tasks in a given time, and to gain a deeper understanding of the problem domain (Schilling et al. 2003).

Diversity. Diversity is the breadth of experience (i.e., experience in different areas). It has been observed to positively affect learning and, therefore, can influence performance (Cohen and Levinthal 1990). The understanding of one problem domain could provide analogous solutions in a new problem domain (Schilling et al. 2003). Simonton (1999) notes that this transfer of knowledge may occur across domains that appear to have little in common. Diversity can increase the capacity to transfer knowledge across domains as it allows individuals to "recognize the value of new information, assimilate it, and apply it" (Cohen and Levinthal 1990, p. 128).

Complexity. Job complexity is the extent to which a job entails autonomy and allows for decision latitude (Kohn and Schooler 1983). Complex tasks may have multifaceted requirements, encouraging the need to combine knowledge from various sources. Such tasks require more intricate thought processes than do simpler ones (Farr 1990). As task complexity can influence the application of knowledge and the learning process, it can influence performance.

Extrinsic Motivation

Expectancy theory (Vroom 1964) assumes behavior to be a result of choices among alternatives wherein individuals make choices to maximize pleasure and minimize pain. Pleasure comes from rewards intrinsic or extrinsic to the individual (Vallerand 1997). Extrinsic rewards in the form of money are very common (e.g., salaries and bonuses associated with jobs, or prizes associated with sports and other competitions). Individuals who find the extrinsic reward attractive devote more effort, and are likely to achieve better performance (Gneezy et al. 2011). This is reinforced by research on the theory of tournaments, where monetary incentives have been found to induce individuals to work harder in competitive environments (Connelly et al. 2014).

Work Conditions

To better understand individual variation in performance, it is important to examine the nature of the job as well as the environment within which the job is performed, in addition to the characteristics (e.g., knowledge levels) of the individual who is performing the work (Porter and Lawler 1968; Sonnentag et al. 2008). One aspect of work conditions that is particularly relevant to our context is competition, and research on tournaments is very pertinent in this regard as well. This theory suggests that competition intensity could influence performance positively or negatively (Nalebuff and Stiglitz 1983). For instance, while some people might work harder to defeat other participants, others might capitulate in the face of competition and perform poorly.

Model Building: Winner Prediction in the Face of Competition

Competition is a vital factor in crowdsourcing contests, and models that make predictions in that domain need to take that into account. As mentioned earlier, most of the models relevant to our context come from the sports literature (e.g., Bolton and Chapman 1986). Any model that can be used to predict winners is a potential candidate for our framework. We have included many well-established methods in our experiments, namely SVMs, neural networks, Bayesian classifiers (including Bayesian networks), and multinomial logit and its variants. While some of them (e.g., SVMs and neural networks) do not provide probabilities directly, the scores they provide can be converted into probabilities through appropriate transformations (e.g., a logistic transformation).

As discussed earlier, platforms can provide seekers different options for evaluation, and we consider three possible ranking mechanisms. In Scenario I, the seeker selects only a winner, ranking her above all the other solvers as a result. In Scenario II, the seeker ranks every submitted entry. In Scenario III, a seeker is allowed to shortlist solvers before eventually choosing a winner from this list. For example, the site we have analyzed, *99designs*, allows both the first and the third options, while *Topcoder* allows the second option. Several of the models we have considered cannot be applied directly to the problem we are studying; they have to be adapted to fit our context. Depending on the specific scenario, capturing all the information available from the historical task outcomes may lead not only to adapting how a model can be used, but also to different data representations to train some of the models.

The logistic regression family of models can be tailored quite naturally to accommodate each scenario. Therefore, for each scenario, we first discuss how those models could be used, and then explain how the other models can be adapted, along with associated data representations.

Scenario I: Seeker Chooses Only the Winner

When a seeker selects only the winner from all the submitted solutions, the winner is implicitly being ranked above all the other solvers. The multinomial logistic regression model can explicitly capture such information on ranks and competition structure; the probability of the target solver winning an open task i is determined as follows. Let the number of solvers (including the focal solver) in task i be M_i, and the solvers be indexed by j ($j = 1, \ldots, M_i$). X_{ij} are the attribute values of solver j participating in task i. We measure the *performance* of solver j in task i by V_{ij}, and model it as a function of the attribute values X_{ij} (Terwiesch and Xu 2008), that is, $V_{ij} = \alpha X_{ij} + \varepsilon_{ij}$. Here, α is a vector of weights corresponding to the importance of each attribute in a solver's performance, and ε_{ij} is an error term that captures the impact of unobserved factors on a solver's performance (e.g., the solver's effort level). Solver j can win task i only if she performs better than all other participants. Therefore, the winning probability for solver j in task i is

$$P_{ij} = P(V_{ij} > V_{ik}),\ k = 1, \ldots, M_i \tag{1}$$

Assuming that the error term ε_{ij} follows a Type 1 extreme value distribution (Greene 2011), the closed-form expression for P_{ij} is

$$P_{ij} = \frac{\exp(\alpha X_{ij})}{\sum_{k=1}^{M_i} \exp(\alpha X_{ik})} \tag{2}$$

Maximum likelihood methods can be used to estimate α, given historical data on tasks, solvers and outcomes. Once the parameters are obtained, a focal solver's probability of winning an open task given the extant competition can be directly determined using Equation 2.

Unlike the multinomial logistic model, the other models do not allow all the competitors in a task to be evaluated simultaneously. For those models, some extra steps are required to estimate the winning probabilities of solvers participating in a task. We use naïve Bayes as an example to illustrate this process. Solver j's outcome in task i, Y_{ij}, is treated as a binary class label where Y_{ij} is 1 if solver j wins task i, and 0 otherwise. Let X_{wij}($w = 1, \ldots, r$) denote the value for attribute w for solver j participating in task i. Naïve Bayes relies on the assumption of conditional independence for tractability: given the class variable, all the X_{wij} values are independent of each other. Solver j's winning probability is then given by

$$P(Y_{ij} \mid X_{1ij}, X_{2ij}, X_{3ij}, \ldots, X_{rij}) \propto P(Y_{ij}) \prod_{w=1}^{r} P(X_{wij} \mid Y_{ij}) \quad (3)$$

The prior and conditional probabilities in the above model are estimated from the training data. With these parameters, the probability a solver will win an open task can be first estimated ignoring the competition. The process is repeated for each competitor separately, and the estimates obtained for each solver are then normalized to provide the desired probabilities.

Scenario II: Seeker Ranks Every Solver

Next, we consider the case where seekers rank each submission.[4] The multinomial model is adapted as follows. For solver j to be ranked ahead of solver k, her performance must have been better than that of solver k, that is, $V_{ij} > V_{ik}$. If the solvers are ranked 1, 2, …, M_i by the seeker, then we have $V_{i1} > V_{i2} \ldots V_{iM_i}$. The probability of interest in this context is $P(V_{i1} > V_{i2} \ldots V_{iM_i})$. This is analogous to the rank-ordered logit model (Beggs et al. 1981; Chapman and Staelin 1982), for which the probability of observing a particular ranking is

$$P(V_{i1} > V_{i2} \ldots V_{iM_i}) = \prod_{j=1}^{M_i} \left[\frac{\exp(\alpha X_{ij})}{\sum_{k=1}^{M_i} \exp(\alpha X_{ik})} \right] \quad (4)$$

For the other models, the training data used to learn the model parameters has to be represented differently than before. This is because class labels cannot directly capture the full ranking information, and the labels have to be ordered. Instead, we consider a task's outcome as a collection of pair-wise competitions. For example, consider four solvers S_1, S_2, S_3, and S_4 in a contest, with the final ordering $S_1 > S_2 > S_3 > S_4$. To capture rank information among the four solvers, the task can be viewed as six (C_2^4) pair-wise competitions with the following outcomes: $S_1 > S_2$, $S_1 > S_3$, $S_1 > S_4$, $S_2 > S_3$, $S_2 > S_4$, and $S_3 > S_4$. These pair-wise outcomes capture the full ordering. Once the training data has been so reconstructed, the other models can be applied.

[4]A platform may allow solvers to submit multiple solutions for a given task. For such situations, a solver's highest ranked solution can be considered to determine the solver's rank.

Scenario III: Seeker Shortlists Solvers before Choosing the Winner

This is where seekers shortlist a few solvers before eventually choosing a winner. This implies the shortlisted solvers are ranked above the non-shortlisted ones, and the eventual winner is ranked above the other solvers in the shortlist. As there is no other rank-ordering among the shortlisted solvers, these solvers have to be treated as being of equal rank thereby creating ties in the ordering. Not considering the ties properly may lead to estimation bias (Hausman and Ruud 1987), and therefore a model that can account for ties in rank ordered data is needed.

The multinomial logit model can be extended to capture shortlist information explicitly (Allison and Christakis 1994). With ties allowed, a seeker for task i could, for instance, rank the M_i solvers as 1, 2, 3, …, k, $\{k + 1, k + 2, \ldots, k + d\}$, $k + d + 1, \ldots, M_i$, where all the solvers in the range $k + 1$ to $k + d$ have the same rank. One way to handle this is to assume that the seeker had some preference ordering that was not revealed. This means the seeker's ordering could have been any from the set of permutations D of the d solvers who have been given the same rank.

To illustrate this, suppose there are four solvers, three of whom (S_1, S_2, and S_3) are initially short-listed and then S_1 is picked as the winner. The ranking is $S_1 > \{S_2, S_3\} > S_4$, that is, solvers S_2 and S_3 have the same rank. There are two possible preference orderings for these two solvers, $S_2 > S_3$ or $S_3 > S_2$, and thus $|D| = 2$. Following Allison and Christakis (1994), we have

$$\begin{aligned} P(\{V_{i2}, V_{i3}\} > V_{i4}) &= pr(V_{i2} > V_{i3}) + pr(V_{i3} > V_{i2} > V_{i4}) \\ &= \frac{\exp(\alpha X_{i2})}{\exp(\alpha X_{i2}) + \exp(\alpha X_{i3}) + \exp(\alpha X_{i4})} \frac{\exp(\alpha X_{i2})}{\exp(\alpha X_{i2}) + \exp(\alpha X_{i4})} \\ &+ \frac{\exp(\alpha X_{i3})}{\exp(\alpha X_{i2}) + \exp(\alpha X_{i3}) + \exp(\alpha X_{i4})} \frac{\exp(\alpha X_{i2})}{\exp(\alpha X_{i2}) + \exp(\alpha X_{i4})} \end{aligned}$$

Solver S_1 has a unique rank, and therefore her probability expression will have the same form as in Equation 4. Expressions of the above form, along with Equation 4, can be used to express the general form of the likelihood function when there is rank-ordering with ties. If a seeker assigns R distinct ranks ($R \le M_i$) and d solvers are tied for rank $k + 1$, the likelihood for the solvers to be in that specific ranking is

$$P(V_{i1} > \ldots > V_{ik} > \{V_{ik+1} \ldots V_{ik+d}\} > V_{ik+d+1} > V_{iM_i}) =$$
$$\prod_{j=1}^{k}\left[\frac{\exp(\alpha X_{ij})}{\sum_{l=j}^{M_i}\exp(\alpha X_{il})}\right]\left\{\sum_{q\in D}\prod_{r=k+1}^{k+d}\left[\frac{\exp(\alpha X_{ir})}{\sum_{s=r}^{M_i}\exp(\alpha X_{is})}\right]\right\} \quad (5)$$
$$\prod_{g=k+d+1}^{k+d}\left[\frac{\exp(\alpha X_{ig})}{\sum_{c=g}^{M_i}\exp(\alpha X_{ic})}\right]$$

where D is the set of permutations of all the possible preference sequence combinations of the d solvers, and q is an element in that set. The probability Equation 5 has three subparts: $\prod_{j=1}^{k}\left[\frac{\exp(\alpha X_{ij})}{\sum_{l=j}^{M_i}\exp(\alpha X_{il})}\right]$ captures the unique ranking for solvers $V_1, \ldots, V_k$ before the ties; the ties among $V_{ik+1}, \ldots, V_{ik+d}$ are captured by $\sum_{q\in D}\prod_{r=k+1}^{k+d}\left[\frac{\exp(\alpha X_{ir})}{\sum_{s=r}^{M_i}(\alpha X_{is})}\right]$; and the unique ranking for solvers $V_{ik+d+1},\ldots,V_{iM_i}$ is captured by $\prod_{g=k+d+1}^{M_i}\left[\frac{\exp(\alpha X_{ig})}{\sum_{c=g}^{M_i}(\alpha X_{ig})}\right]$. The key idea here is to adapt the model where every solver is assigned a distinct rank in such a way as to capture all the possible ranks among ties.

For the other models, the training data used to learn the model parameters is reconstructed in a manner similar to the full ranking scenario. For example, suppose as before that S_1, S_2, and S_3 are shortlisted, and S_1 eventually chosen as the winner from among four solvers S_1, S_2, S_3, and S_4. The training data will incorporate solver pairs pertaining to solvers with different ranks as in Scenario II. As S_2 and S_3 are ranked the same, they cannot be distinguished from each other; consequently, the corresponding pair is ignored. Therefore, there are five pairs of ordered solvers in this example: $S_1 > S_2$, $S_1 > S_3$, $S_1 > S_4$, $S_2 > S_4$, and $S_3 > S_4$. Once the training dataset has been reconstructed in this manner, the other models can be applied as before.

Operationalizing the Framework

The proposed framework can be operationalized as follows. The set of attributes that are observable by a platform and can be expected to impact a solver's performance for a given task must be identified first. Depending on the scenario being used by the platform, the training data would be prepared accordingly for the models under consideration. It is desirable to try different models, with the model that performs best on the test data set being adopted for making recommendations. This evaluation can be performed by dividing the historical data into two parts, with one part used for training the models and the other for testing. Once a model has been selected, it can be deployed for solvers when they log in to the site.

The framework is computationally quite efficient. The probabilities can be computed easily if the model parameters have been obtained beforehand, which would normally be the case. If the maximum number of open tasks at any point in time is T, and the maximum number of solvers in an open task is S, the worst case computational complexity for making recommendations is $O(TS)$. Given the maximum values likely to be observed for T (in the 1,000s) and S (in the 100s), platforms can determine in real-time the tasks to recommend to a focal solver when she logs on to the site, while accounting for all competitors participating in each of the open tasks.

Data

We validate our framework with data collected from a popular crowdsourcing contest platform, *99designs*. In this section, we describe the data and explain how the theory established earlier helps us identify the variables used to build the prediction models.

Data Collection and Description

99designs is one of the largest online platforms for graphic design in the world. They have eight task categories: (1) logo and identity, (2) website and app, (3) business and advertising, (4) clothing and merchandise, (5) art and illustration, (6) packing and label, (7) book and magazine, and (8) other. We have chosen to work with the logo and identity category as it is the most popular category on the platform, with approximately 60% of open tasks coming from it. Most tasks in this category are posted by firms looking to design logos for official use on business cards, stationary, or web pages. The large number of open tasks makes it difficult for solvers even to browse through them all, let alone evaluate each task effectively for participation. The category is further divided into five subcategories: logo design, logo and business card, business card, stationery, and business identity pack.

We collected data using a crawler written in Perl. Data collection started from the task listing page, which provides information on all open tasks and some of the recently closed (completed) ones. The crawler collected data on all completed tasks that were listed on the site on a specific day. For each such task, it then went to the task landing page and collected all the information available there. This included information such as task title, task number, prize amount, firm (seeker) industry, and logo style. The crawler then visited the corresponding solution page to collect information on all solutions. This included the list of solvers who had submitted entries, the sequence in which entries were submitted, and the winner identified by the seeker. The platform allows seekers to shortlist a subset of solvers, and this information was collected where available. Finally, it collected all available information on each participating solver, including both solver reported information and information on the solver generated by the platform.

Our data set involves 1,917 tasks that were listed on *99designs.com* over a two-week period in August 2013, and about 13,000 solvers. There were 37 solvers on average in each task with a standard deviation of 29. The number of solvers in a task ranged from a minimum of 3 to a maximum of 284. The average prize money was about $367, and the standard deviation $163; prize money ranged from $179 to $1,581. We divided the dataset into two equal parts for analysis, with the first half of the tasks (based on task posting times) being used for training (i.e., parameter estimation) and the second half reserved for testing. As many of the attribute-values evolve over time based on solver participations and task outcomes, this partitioning scheme helps build models that incorporate past information as comprehensively as possible.

Attributes Influencing Winning Probability

Various general determinants and associated predictors of performance were introduced earlier, based on theories of performance, learning, motivation, and tournaments. Next, we discuss how these predictors could influence the performance of solvers in a crowdsourcing context. We also identify various attributes for each predictor based on existing literature on crowdsourcing contests and information available to *99designs*. It is not always easy to identify a single attribute that completely represents each predictor, and platforms usually have different pieces of information that can measure the predictors. We identify as many attributes grounded in the theory outlined previously as we can from *99designs*.

Attributes for Ability

Intrinsic Ability. The category we investigate involves logo design. Davis (1994) notes that artistic activities like graphic design require higher levels of intrinsic ability. Consequently, it is reasonable to expect that intrinsic ability will affect solver performance. Solvers who have performed well historically are likely to be of higher intrinsic ability, therefore intrinsic ability can be captured using variables that describe past performance. Existing research on crowdsourcing contests have used such measures as proxies of intrinsic ability. For example, Archak (2010) used *solver rating*, a platform-provided measure calculated based on past performance, to measure intrinsic ability, while Yang et al. (2011) and Khasraghi and Aghaie (2014) used the number of wins and the number of times a solver has been short-listed for this purpose.

99designs makes available four metrics that reflect each solver's past performance: *points*, *capability level*, *number of wins*, and *number of times short-listed*. These variables (X1–X4) directly relate to past performance, and are potential measures of a solver's intrinsic ability.

Expertise. Research on crowdsourcing suggests that performance is limited by the extent of an individual's expertise or skills (Terwiesch and Xu 2008). For example, in logo design contests, solvers with designing expertise are likely to perform better than solvers without that expertise. Huang et al. (2012) find that higher levels of expertise lead to higher levels of performance in crowdsourcing contests. We develop three metrics to measure a solver's expertise level.

99designs allows solvers to specify their expertise categories, and we use the number of such categories as a measure of expertise (X5). Furthermore, as a solver who is an expert in a task category is more likely to win a task in that category relative to solvers who do not have such expertise, we have a variable that indicates whether a solver has claimed expertise in a specific category or not (X6). We create a similar variable for subcategory levels as well (X7).

Attributes for Experience

General Experience. Solvers can get better at executing existing routines and developing new designs as they gain experience. Solvers who participate in many tasks become more proficient as a result, and that could help improve their performance in future tasks. A solver's experience has been found to positively influence performance in crowdsourcing contests (Archak 2010; Boudreau et al. 2011; Yang et al. 2009). For example, Yang et al. (2009) find that greater experience (as represented by the number of participated tasks) leads to a higher chance of winning a task.

99designs does not provide measures of solver experience. In keeping with prior research (Archak 2010; Yang et al. 2009), we consider the number of participated tasks (X8) as a measure of experience. *99designs* allows solvers to submit multiple solutions for a task. As this may capture additional information about a solver's experience, we use the total number of solutions submitted by the solver in the past (X9) as another measure of experience. In addition, we include another variable associated with experience, *success rate* (X10). Because our data collection is over a two-week period, this variable captures the recent winning experience of a solver. As X10 relates to success, it also captures some aspects of expertise and ability.

Specialization. Specialized experience can improve a solver's performance in crowdsourcing contests. For example, solvers could accumulate designs in their specialized area (e.g., the restaurant industry), making it easier for them to generate designs for new tasks; solvers with existing designs that could be adapted to meet the requirements of a new task will not have to start from scratch. Further, solvers specializing in an area are more likely to be good at assessing the contextual characteristics associated with that area. This would enable them to arrive at designs better suited to the task's target audience, and to the tacit requirements of the seekers.

We create measures of specialized experience at the industry and subcategory levels. X11 captures the number of prior participations by a solver in a specific industry, while X12 captures the fraction of a solver's prior participations in that industry. X13 captures the number of prior wins by a solver in a specific industry, while X14 captures the fraction of a solver's prior wins in that industry. Similarly, we create two variables (X15 and X16) to measure the experience of a solver in each subcategory. We also capture a solver's winning experience in each subcategory using the number of wins and the proportion of wins (X17 and X18) in that subcategory.

Diversity. Solvers with experience in diverse domains approach problems from different perspectives and create novel solutions (Jeppesen and Lakhani 2010). Diversity of experience resulting from solving tasks across industries and design styles could expose solvers to different ideas, which could affect performance. Solvers can choose to diversify across industries and subcategories, and it could be useful to track solvers' experiences in each of these dimensions. Therefore, we count the number of distinct industries and subcategories in which a solver has participated (X19 and X20), along with the number of different industries and subcategories in which the solver has won tasks (X21 and X22).

Complexity. Tasks on crowdsourcing contest platforms can be quite heterogeneous in term of seeker requirements and competition intensity. We capture a solver's complexity of experience using variables that reflect both of these aspects. The prize amounts of tasks reflect seeker requirements, with higher prize amounts usually associated with tasks that have more complex requirements. Task complexity is also reflected in the intensity of competition, as more competitive tasks tend to be harder to win. That is, solvers who have experienced only easy tasks in the past may not find it easy to adapt to difficult new tasks, while solvers who have experience with complex tasks (whether as a result of difficult task requirements, or as a result of intense competition) will be better prepared to compete in difficult tasks in the future.

We develop three metrics to capture complexity of experience. Tasks with more solvers and more submitted solutions will usually be harder to win, as these make the competition more intense. Therefore, we use the average number of solvers per participated task (X23) and the average number of solutions per participated task (X24) as measures of the complexity of a solver's experience. Because there is usually greater competition in tasks with more prize money, we also consider the average listed prize per participated task (X25) as another measure.

Extrinsic Motivation and Work Conditions

While prize amount and task duration are important drivers of extrinsic motivation, we do not include them in our model. These are common across all participants in a task, and their effects will cancel out. In fact, that is explicitly the case for multinomial logistic models, and our experiments using the other models found these attributes to be uninformative as well. However, attributes such as task industry and subcategory (which are also common across all participants) could affect solvers' winning probabilities differently (e.g., X7) because of their historical performance in such tasks. In our context, an important aspect of work conditions is competition, which is captured by considering the predictors and associated variables for all competitors in a task.

In all, we identify 25 attributes (summarized in Table 2). In practice, the variables would be continuously updated as solvers participate in new tasks. We should mention that the variables would be able to capture the pertinent information accurately even for new solvers; if historical data suggests that solvers new to the platform rarely win their first task, the models would learn this from the training data and make appropriate predictions for new solvers in future tasks.

While data has been collected for over 1,900 tasks, we note that these are collected over a relatively short period. Only the platform-provided variables (X1–X4) reflect a solver's entire participation history; all other variables only reflect events

Table 2. Predictors and Associated Variables

Predictors	Variables
Intrinsic Ability	X1: points
	X2: capability level
	X3: # of wins (across all categories)
	X4: # of times shortlisted (across all categories)
Expertise	X5: # of expertise categories
	X6: 1 if solver reports expertise in current task category; 0 if not
	X7: 1 if solver reports expertise in current task subcategory; 0 if not
General Experience	X8: # of tasks already participated in
	X9: aggregate # of solutions submitted by solver
	X10: success rate
Specialization	X11: # of solver's prior participations in industry of current task
	X12: fraction of solver's prior participations in industry of current task
	X13: # of solver's prior wins in industry of current task
	X14: fraction of solver's prior wins in industry of current task
	X15: # of solver's prior participations in subcategory of current task
	X16: fraction of solver's prior participations in subcategory of current task
	X17: # of solver's prior wins in subcategory of current task
	X18: fraction of solver's prior wins in subcategory of current task
Diversity of Experience	X19: # of distinct industries participated in
	X20: # of distinct subcategories participated in
	X21: # of distinct industries with wins
	X22: # of distinct subcategories with wins
Complexity of Experience	X23: average # of competing solvers per participated task
	X24: average # of solutions per participated task
	X25: average listed prize per participated task

of the two-week period. Our objective in conducting experiments on this real-world dataset is to illustrate the viability of our framework. A platform like *99designs* will have at its disposal complete historical data, and would be in a position to estimate models with a richer dataset than the one used in our analysis.

Evaluating the Framework Part I: Comparing Alternative Winner Prediction Models

We have implemented the framework and examined the ability of various models to predict winners. *99designs* allows seekers to either select a winner directly (Scenario I), or after a short-listing stage (Scenario III). We describe the baseline models first, and then evaluate the framework for both scenarios.

Baseline Models

Two benchmarks are used for comparison. In the first, a random solver is projected as the winner; no information is considered when making the prediction. This *random selection* model tells us how often the winner would be correctly identified if a solver was picked randomly from the participants in a task. A systematic method (like the one introduced in this paper) is useful only if it can do better than random selection, as that would establish that it is able to leverage available information in a useful manner. Consequently, this forms a fundamental baseline to evaluate our framework. *99designs* measures solvers' abilities through points, and the second benchmark predicts the solver with the most points as the winner in a task. This *points-based* model makes recommendations based on a platform-developed metric the site itself considers the best measure of solver ability. Therefore, this is a highly discriminating benchmark.

Table 3. Winner Prediction Accuracies: Baseline Models

Model	*n* = 1	*n* = 2	*n* = 3	*n* = 4	*n* = 5
Random Selection	35	83	118	161	196
	3.65%	8.66%	12.32%	16.81%	20.46%
Points-Based	68	144	216	292	342
	7.10%	15.03%	22.55%	30.48%	35.70%

We use *prediction accuracy* (the fraction of correct predictions) as one metric for evaluation: if the projected winner of a task is its actual winner, the prediction is considered correct. We track the number of correct predictions for the 958 tasks in the testing set. The random selection model identifies winners correctly in only 35 of the 958 tasks, an accuracy of about 3.65%. The points-based model identifies winners correctly in 68 tasks. This represents an accuracy level of 7.1%, and the increase over random selection is statistically significant at the 0.01 level.

So far, we have focused on projecting a single winner for each task. Finding the specific winner correctly with well over 30 solvers on average is difficult in general. For instance, there may be multiple strong solvers competing in the same task. To make a correct prediction, the model has to precisely predict that the performance of one of these strong solvers is superior to that of all the others. From this perspective, measuring accuracy based on predicting only one winner is a very stringent requirement. For example, consider a model that predicts the actual winner to have the second highest winning probability for one task, and the lowest winning probability for another task. The prediction quality is quite different for these two tasks, and this aspect is not captured if we consider accuracy based on predicting only a single winner.

Therefore, we consider another more nuanced measure: in addition to predicting a single winner, we rank solvers based on winning probability and predict the top *n* solvers (Hariri et al. 2011; Sieg et al. 2010). When one of the top *n* predicted solvers is the actual winner, the prediction is considered a success. We have used values of 1 to 5 for *n* in our experiments. The results (Table 3) provide a broader perspective of the performance of the two approaches. The points-based model performs significantly better than the random approach for all values of *n*.

Model Comparison: Scenario I

We first consider the scenario where only one winner is chosen (with no additional ranking information) for a task, and examine the performance of the five models discussed earlier: multinomial logit (MNL), naïve Bayes (nB), Bayesian networks (BN), neural networks (NN), and support vector machines (SVM). Each model was estimated from the training dataset using all 25 variables (X1–X25) identified earlier (Table 2). The performance of neural networks often depends on the number of hidden layers and the number of hidden nodes in each layer. We experimented with several values of these parameters.[5] The network with one hidden layer and five nodes performed the best, and the reported results are based on this setting. For SVMs, the polynomial kernel performed best in our experiments.

In addition to the accuracy of the top *n* predictions, we report another measure that considers how far the winner's predicted rank is from her true rank (which is 1). The smaller the distance, the better the prediction. The average distance (reported as mean average error, or MAE) provides another way to compare the predictive ability of the different models.

After training, we predict the winning probability for each solver in each test task.[6] As the multinomial logit, naïve Bayes, and Bayesian network models can predict the winning probability directly, we rank solvers based on the predicted probability for these models. The solvers are ranked based on the predicted scores in the case of NNs and SVMs. The solver with the highest predicted winning probability (or score) is projected as the winner of the task. The results are summarized in Table 4.

All the models except SVM perform better than the benchmarks in terms of prediction accuracy. In terms of MAE, neural networks, Bayesian networks, and MNL outperform both benchmarks, while naïve Bayes and SVM are inferior to

[5]We experimented with networks having up to 2 hidden layers, and with up to 14 nodes in a hidden layer.

[6]Our focus is on the quality of predictions, and not on the impact of specific variables. For completeness, we list the estimated parameters for MNL and ranked-MNL-ties and corresponding significance levels in Appendix A.

Table 4. Winner Prediction: Scenario I

Model	*n* = 1	*n* = 2	*n* = 3	*n* = 4	*n* = 5	*MAE*
Random Selection	35***	83***	118***	161***	196***	17.89***
	3.65%	8.66%	12.32%	16.81%	20.46%	
Points-Based	68***	144***	216***	292***	342***	11.61**
	7.10%	15.03%	22.55%	30.48%	35.70%	
nB	87***	157***	230***	283***	347***	12.49***
	9.08%	16.39%	24.01%	29.54%	36.22%	
BN	114	181***	275*	329***	395**	10.46
	11.90%	18.89%	28.71%	34.34%	41.23%	
NN	111	212	277	331***	381***	11.53**
	11.59%	22.13%	28.91%	34.55%	39.77%	
SVM	51***	97***	142***	184***	230***	14.98***
	5.32%	10.13%	14.82%	19.21%	24.01%	
MNL	123	214	292	372	427	10.17
	12.84%	22.34%	30.48%	38.83%	44.57%	

***, **, and * represent significance levels at the 0.01, 0.05 and 0.1 levels, respectively.
Significance levels show the extent by which the performance of the models is *inferior* to that of MNL.

the points-based benchmark.[7] The improvements in terms of accuracy are significant particularly for Bayesian networks, neural networks and MNL. We find that MNL performs better than the other models consistently for our dataset in this scenario; this is true for both prediction accuracy and MAE.

Table 4 shows whether the performance for a given model is significantly *inferior* to that of MNL. The relative improvement in prediction accuracy of MNL over the points-based benchmark is in excess of 80% when $n = 1$. MNL performs better than naïve Bayes and SVM across all experiments, with the differences being statistically significant at the 0.01 level. The improvements over Bayesian networks are less strong. While MNL performs better, the improvement in prediction accuracy is not significant when n is 1 (they are significant for all other values of n) nor is the improvement significant in terms of MAE. The comparison with neural networks is similar: while MNL performs better in every case, the improvement in accuracy is significant only for $n > 3$.

[7]SVM performs poorly in Scenario 1 because the class variable is very skewed. With an average of around 37 participants per task, only 1 in every 37 training instances would have a positive class. The other models are more robust when dealing with such skewed class distributions. The performance of SVM improves dramatically for Scenario III, where the class variable is no longer skewed.

Model Comparison: Scenario III

We use the models described under Scenario III in the previous section to deal with ranks and ties.[8] No modifications to the dataset are needed to develop the ranked-MNL-ties model. For the other models, a modified dataset is created as described in that section. The results are presented in Table 5.

We find that in terms of prediction accuracy, all models except Bayesian networks perform significantly better than the two benchmarks in every case. The ranked-MNL-ties model consistently performs better than the other models. We also find that for all values of n, the accuracies of ranked-MNL-ties are significantly higher (at different levels of significance). When n is 5, ranked-MNL-ties has an accuracy of 45.62%, that is, it can predict the winner to be in the top 5 positions almost half of the time. In terms of MAE, SVM does the best closely followed by Ranked-MNL-ties, with these two models significantly outperforming all other models.

These experiments demonstrate that information available to the platform can be used to significantly improve the quality of predictions. We reiterate that our data captures roughly two weeks of activity on the site, and we have been able to

[8]Of the 959 tasks in the training set, seekers short-listed multiple solvers before picking a winner in 775 tasks.

Table 5. Winner Prediction: Scenario III

Model	*n* = 1	*n* = 2	*n* = 3	*n* = 4	*n* = 5	*MAE*
Random Selection	35***	83***	118***	161***	196***	17.89***
	3.65%	8.66%	12.32%	16.81%	20.46%	
Points-Based	68***	144***	216***	292***	342***	11.61**
	7.10%	15.03%	22.55%	30.48%	35.70%	
nB	98**	171***	236***	307***	369***	12.36***
	10.23%	17.85%	24.63%	32.05%	38.52%	
BN	93***	162***	218***	268***	312***	14.85***
	9.71%	16.91%	22.76%	27.97%	32.57%	
NN	99***	192***	278***	334***	391***	11.61**
	10.33%	20.04%	29.02%	34.86%	40.81%	
SVM	106***	202***	289**	361**	411***	10.23
	11.06%	21.09%	30.17%	37.68%	42.90%	
Ranked-MNL-Ties	123	225	309	377	437	10.36
	12.84%	23.49%	32.25%	39.35%	45.62%	

***, **, and * represent significance levels at the 0.01, 0.05 and 0.1 levels, respectively.
Significance levels show the extent to which the performance of the models is inferior to that of ranked-MNL-ties.

demonstrate the value of using additional attributes for prediction using just this limited amount of data. A crowdsourcing platform would be able to better fine-tune the predictive model given the data available to it.

Evaluating the Framework Part II: Recommendation System Effectiveness

In this section, we evaluate the effectiveness of the recommendation system from two perspectives. First, given our test data, we examine the ability of our approach to correctly identify the tasks a solver won from among the tasks in which the solver participated. We then evaluate the reliability of recommendations made when only partial competition structures are available. We have used ranked-MNL-ties in these experiments, because of its good performance overall.

Recommendation Quality Within a Solver's Chosen Tasks

In these experiments, we focus on a solver's performance across different tasks (as distinct from the performance of different solvers on a focal task). Specifically, we examine if our approach can identify the tasks a solver actually won from among those in which the solver had chosen to participate. We expect the solver to participate in those tasks that she has a reasonable expectation of winning. We examine, from within these tasks preselected by a solver, how well our approach can identify her ability to win each task relative to the eventual outcome.

The experiment is conducted as follows. For each solver, we identify every task in which she has participated and use ranked-MNL-ties to estimate the probability with which the solver will win each task. These tasks are then ranked by the estimated probability, and the top n tasks in this ranked list are compared with the eventual outcome (where n is the number of tasks the solver has won). These top n tasks can be viewed as *recommendations* that our approach would make to the solver from among the participated tasks. We measure the quality of recommendations by the success rate. The success rate of recommendations made to a solver who wins k of the n recommended tasks is k/n. This is compared with the success rate the solver has experienced on the platform, which is n/m for a solver who has won n tasks from the m in which she has participated. The solver's success rate n/m captures the effectiveness with which a solver has been able to select winnable tasks, and is a natural basis for comparison. If the success rate of the recommendation system is higher than the rate actually experienced by the solver, that is further evidence that the recommendation system can indeed identify tasks the solver can win.

Table 6. Recommendation and Solver Success Rates

Solver Success Rate Range	Number of Solvers	Number of Recommendations Made	Average Number of Tasks Solvers Participate In	Solver Success Rate	Recommen-dation Success Rate	Improvement (%)
Overall	714	869	9.32	22.17%	31.24%	40.91%
[0.6, 1)	10	24	3.00	84.33%	88.33%	4.74%
[0.5,0.6)	84	101	2.41	50.00%	66.07%	32.14%
[0.4, 0.5)	14	31	5.43	40.52%	54.76%	35.14%
[0.3, 0.4)	84	99	3.54	33.33%	44.05%	32.14%
[0.2, 0.3)	165	207	5.43	23.11%	36.08%	56.10%
[0.1, 0.2)	180	212	8.80	13.63%	20.25%	48.55%
(0, 0.1)	177	195	20.17	6.51%	10.22%	56.86%

In order to make meaningful comparisons, we consider only those solvers who have won at least once and further have not won every task in which they have participated (i.e., considering solvers with a success rate of either zero or one in our dataset is not informative). The final dataset analyzed in this set of experiments involved 714 solvers with a total of 869 wins. These solvers participate in 9.32 tasks on average, and have an overall success rate of 22.17%.

The results are presented in Table 6. The first row ("Overall") provides results for the entire dataset. The overall recommendation success rate shows a 40.91% improvement over the success rates currently enjoyed by these solvers on the platform. The abilities of solvers vary, as do their actual success rates. The rest of Table 6 drills down into the overall results to understand the effect of the recommender system for solvers with different success rates. As the results show, the recommender system can help solvers in every range. While the impact is most pronounced for solvers who have low success rates (solvers with success rates below 30%), relatively successful solvers (solvers with success rates over 30%) could also benefit considerably from the recommendations. The improvement in success rates is over 30% in every case except the [0.6, 1) category. There is some improvement even in this category, despite the already high success rates enjoyed by these solvers. These experiments highlight the effectiveness of the recommender system, and show that it can improve the success rates of solvers at all levels, from the relatively unsuccessful to the very successful.

Ranking Consistency over the Lifespans of Tasks

In order to make recommendations to a target solver, the platform would need to estimate her winning probability for all open tasks, and then recommend a predetermined number of tasks based on these probabilities (e.g., rank tasks based either on estimated winning probability or expected payoffs). Naturally, recently posted tasks have fewer participants than tasks that are about to close; therefore, at a given point in time the winning probabilities for newly posted tasks would tend to be higher than those for the other tasks. To make comparisons more meaningful, the platform could provide recommendations to a solver based on different time lines; for instance, tasks for each closing date could be recommended separately, with the recommendations for each closing date sorted by the predicted winning probabilities or expected payoffs.

We conduct a set of experiments to evaluate the robustness of the recommendation system. When recommending open tasks to a solver, the platform will not know the final competition structures of the tasks. Instead, the platform will have to make recommendations based on the competition structure that exists for each open task at the time the recommendation is to be made. An important consideration is the reliability of the task rankings when tasks have received only a fraction of their eventual set of solutions. An interesting question then is whether existing competition structures for different open tasks that are at the same stage in their overall time lines are representative of the eventual competition that the target solver will face at the time the tasks close. In that case, even though the probability that the target solver will win a task changes over time as more participants join in, the likelihood of winning one task relative to another may not be very different when the task closes (i.e., the rank order of the winning probabilities for different open tasks at a similar stage of completion may not change much over time).

To examine if this holds, we conduct the following experiment. We consider tasks at the same level of completion; for example, we consider the competition structures for tasks when they have received only the first 25% of all their eventual solutions. We then compare, using the ranked-MNL-ties model, which of two tasks would be recommended to a target solver if only their current sets of competitors were considered, relative to the recommendation if their final competition structures were known. If the recommendations are consistent at both points in time, it would indicate that it is reasonable to make recommendations using tasks with partial (incomplete) competition structures. We repeat this for all feasible task pairs for each solver for three partial competition structures: specifically, when the first 25%, 50%, and 75% of the competition structures are known. We find that the recommendations stay the same for 89.32% of the task pairs when the first 25% of the competition are known, with the fractions increasing to 93.56% and 96.47% when 50% and 75% of the competition are known, respectively. This indicates that the proposed approach is able to identify tasks to recommend to a solver reasonably well even with incomplete information regarding the final competition structure.

Implications to Stakeholders

While we have designed our recommender system to help solvers find winnable tasks, it has important implications for all stakeholders in a crowdsourcing contest platform. First and foremost, by helping solvers find winnable tasks relatively quickly (i.e., without having to explore all open tasks, or restrict their search to a small subset of open tasks), it should enable them to devote more effort toward solving the tasks. This should lead to solvers producing better solutions, or participating in more contests, or both, and thereby lead to an increase in their chances of winning. The improved quality and/or quantity of solutions should benefit the seekers as well, since they are more likely to find better solutions in the process. We expect that the deployment of such a system will be particularly helpful to seekers whose tasks have somehow not been able to attract a reasonable number of solvers; the recommender system should be able to recognize if a posted task has not received many solutions (which would make it more winnable than other tasks) and recommend that task to other solvers. This should reduce the chances of a task receiving fewer than the minimum acceptable number of solutions, a criterion that several platforms adopt for a task to be considered successful. This will directly benefit the platform in terms of potential revenues from such tasks; for instance, we have observed in *99designs* that seekers have not picked a winner in more than 10% of the posted tasks.

In order to examine the potential impact of a recommender system to the different stakeholders, we conduct experiments on tasks posted and completed on *99designs*. In these experiments, we consider 100 new tasks that have been posted in sequence. These represent tasks that would typically be at a similar stage of completion when a solver visits the site. We collect all pertinent information on these tasks, including information on the solvers who have participated in them (some solvers participate in more than 1 of these 100 tasks), the prize amounts, and the eventual winner, if any. We assume that the first quartile of solvers who submitted solutions for each of these tasks is fixed (this roughly corresponds to tasks that have been open for one day). We then simulate the recommendation process as follows: we assume that the system makes a recommendation to each of the remaining solvers of every task by considering all eligible open tasks from among the 100 tasks; a task is eligible so long as the solver is not among the fixed solvers for that task. The task that maximizes the solver's expected payoff given the current competition structure for each task is recommended; we use expected payoffs as we expect the prize amount to affect the number of participants in a task. When a solver receives a recommendation, the solver may choose to participate in the recommended task or not. To simulate this behavior, we allow a solver to accept a recommendation (i.e., participate in the recommended task) with adoption probability p; the solver is assumed to participate in the task she actually selected otherwise. This process is repeated for all the non-fixed solvers across the 100 tasks. We conduct experiments by varying the adoption probability p.

It is difficult to estimate the potential savings in the search effort of the solvers based on the recommender system. It is also difficult to determine what would be the potential increase in the quality and quantity of solutions from these solvers. We assume that each solver will participate in the same number of tasks in the simulation experiment as they did in the true participation data; this is conservative because the savings in search effort should allow some solvers to participate in more tasks in practice. We then compare the simulated participation level for each of the tasks with the task's true participation level.

The average prize money for the 100 tasks selected for these experiments is \$411. The average number of solvers per task is slightly over 52, with 3,362 distinct solvers. The minimum number of solvers in a task is 3 and the maximum is 223. Importantly, the seekers selected a winner in 86 of the 100 tasks; they did not find a satisfactory solution in the remaining 14 tasks (these are referred to as *failed* tasks). We were particularly interested to see how participation levels changed for failed tasks when the recommender system was simulated. Table 7 displays the actual number of solvers for each of these 14 tasks, and also the number of solvers in the simulated experiments when the adoption probability p was set to 0.1 and 0.2, respectively.

Table 7. Number of Solvers in Failed Tasks: Actual and Simulated

Task ID	Number of Solvers		
	Actual	Simulated	
		p = 0.1	p = 0.2
F1	3	41	48
F2	7	40	49
F3	8	26	34
F4	11	31	39
F5	11	32	41
F6	12	19	31
F7	12	38	47
F8	14	25	31
F9	14	18	27
F10	19	23	32
F11	19	16	22
F12	20	23	31
F13	22	27	39
F14	23	22	27

As we see from the table, the actual number of solvers in the failed tasks ranged from 3 to 23. The number of solvers increases in 12 of these tasks when p is 0.1, and in all 14 tasks when p is 0.2. The average increase in the number of solvers is more than 13 even when p is only 0.1. This is quite substantial, and would result in many of these tasks having a successful outcome. For example, if the threshold for the minimum number of solvers needed to ensure a task will have a winner was 25, eight of these 14 tasks would have successful outcomes. If p was 0.2, then 13 of the 14 tasks would have enough solvers to be considered successful.

Deploying a recommender system could also potentially reduce the number of solvers for a task by diverting some solvers to other, potentially more profitable, alternatives (in fact, given the constraint we have imposed regarding the total number of tasks in which a solver participates, that is guaranteed to occur for some tasks in our simulation experiments). This could adversely affect some of the seekers, as they would have fewer solutions from which to choose. We find from the experiments that the tasks that had a large number of solvers are more likely to end up with fewer solvers after the recommendation system is deployed. This is not surprising given the way the recommender system is designed: once it finds a task has a healthy level of competition, it is unlikely to recommend that task to other solvers. To gain a better understanding about the potentially negative impact on the seekers for such tasks, we identified all tasks that had a net decrease of 10 or more solvers when the adoption probability was 0.1. There were five such tasks. For these tasks, we examined not only the decrease in the number of solvers, but also the potential change in the quality of solvers (as a surrogate for the quality of solutions). We measured the quality of a solver in a task using the variables we had used in our recommendation model. Table 8 lists the changes in the number of solvers and in the average performance level of the solvers for such tasks. Interestingly, there was no change in the performance level of the strongest solver for *any* of these five tasks.

We find that the tasks with the largest decrease in the number of solvers were those that had a large number of solvers to begin with; four of the five tasks had well in excess of 100 solvers participating in them. This is consistent with our expectations. The average performance levels of the solvers increased a little in four of these five tasks. This suggests that these tasks would continue to receive a large number of solutions, and the pool of solvers will be, in aggregate, no worse than the original pool. Consequently, although a seeker for such a task may have fewer solvers participating in the task due to a recommender system, it is unlikely that the eventual quality of the solution will be significantly compromised. Therefore, in the aggregate, the seekers should be more satisfied (i.e., in terms of overall welfare of all seekers).

These experiments also help demonstrate the potential hard benefits of the recommender system to the platform (this is in addition to helping solvers locate tasks more efficiently, there-

Table 8. Statistics for Tasks with a Decrease of 10 or More Solvers (p = 0.1)

	Number of Solvers			Average Solver Performance Level		
Task ID	Actual	p = 0.1	Difference	Actual	p = 0.1	Difference
D1	223	202	–21	3.57	3.65	+0.08
D2	181	161	–20	3.80	3.84	+0.04
D3	162	151	–11	4.02	4.05	+0.03
D4	155	144	–11	3.80	3.79	–0.01
D5	87	77	–10	4.09	4.23	+0.14

by providing better and more solutions to benefit the seekers). By increasing participation in tasks with few solvers, the system should help reduce the number of failed tasks. If we assume 25 solvers are necessary to make a task successful (based on what we have informally observed in *99designs*), then the number of failed tasks would be reduced to less than half the current level with an adoption probability of only 0.1. The corresponding increase in revenue to the platform would be greater than 8% based on the prize amounts associated with the tasks that are now projected to be successful, relative to the total prize amounts associated with the originally successful tasks. This should lead to a commensurate increase in the platform's profits (platforms typically charge a fixed percentage of the task's prize amount if the task is successful). We should point out that part of this additional revenue would be distributed among the solvers winning these tasks, thereby leading to an increase in expected payoffs to the solvers.

Conclusions and Future Research

Our research makes several contributions. We propose a recommendation system framework for crowdsourcing contest platforms, where the presence of competition involving multiple solvers provides an interesting contrast to domains where traditional recommendation methodologies are applied. Traditional systems are appropriate where multiple users can buy the same product, while typically only one of the participants can win in a crowdsourcing contest. This necessitates a recommendation system tailored to the competitive crowdsourcing contest environment.

The probability of a solver winning a task depends on her performance, along with those of her competitors. We select Campbell's theory of performance as the basis for determining predictors of performance, reinforcing it with other well-accepted theories of learning, motivation, and tournaments. This reinforced theory helps identify variables to inform the prediction models utilized in our context. To the best of our knowledge, this is the first paper to propose a theoretical framework for performance in the crowdsourcing contest domain.

We illustrate how existing prediction and classification models can be adapted to solve the problem being studied. We validate our framework using data collected from the platform *99designs*. The performance of the winner prediction models is compared to two benchmarks and shown to work well. The variants of the multinomial logistic models generally perform the best given our data. We should emphasize that several other models also work very well, and practitioners should examine the performance of these (and possibly other) models in their specific environments when determining which model(s) to deploy.

We compare the effectiveness of the proposed approach relative to the choices made by the solvers and show that solvers can be more successful with the help of the recommender system. This is true not just for the weaker solvers, but for solvers who enjoy high success rates as well; the average improvement in success rates is over 40% in our experiments. Our experiments also show that the current competition levels for different open tasks which are at the same stage in their overall time lines are representative of the eventual competition that the target solver will face at the time the tasks close. The relative rankings of tasks for a target solver are remarkably consistent over time even when only 25% of the eventual solvers have participated in those tasks.

Our framework has important implications for all stakeholders in crowdsourcing contest platforms. First, by helping solvers find winnable tasks, it would enable them to devote more effort to solving the tasks. This should also help seekers as they would benefit from the extra effort devoted on their tasks. However, the deployment of such a system could divert some solvers to other, potentially more profitable, tasks. This could hurt some of the seekers. Nevertheless, our simulation experiments show that seekers, on the whole, should also benefit from the use of the system. In particular, the system recognizes that a posted task that has not received many solutions is relatively easier to win, and recommends such tasks, thereby increasing participation for them. This reduces the chances of task failure. The platform would also benefit, both in terms of expected revenues and potential improvement in solvers' and seekers' satisfaction with the site. Finally, by reducing the incidence of failures, the expected winnings of solvers would also increase.

The framework has been developed to make recommendations whenever a solver visits the platform (i.e., in a push mode). However, the models we have developed are very flexible and can also be used in a pull mode by a solver; for instance, a solver can shortlist a set of tasks based on her own private preferences, and then ask the platform to rank these tasks based on her winning probability or expected payoff. Such a feature can be very valuable to a solver; even if a solver were able to identify tasks that match her skill set, she would still find it difficult to evaluate by herself her chances of winning each of those tasks given the potentially large number of different solvers participating in each of those tasks.

There is an interesting issue regarding a platform's choice of metric when comparing tasks to recommend to a solver. That concerns what exactly is presented to the solver along with the recommended task(s) when she logs in. The simplest approach would be to provide the recommendation, along with the criterion used (e.g., probability or expected payoffs). Another would be to provide the estimated probabilities themselves; the solver can use the probabilities along with her own criterion (e.g., expected payoffs or expected utilities) to make the decision. Yet another option is to allow solvers to pick the ranking criterion of their choice, and list recommendations accordingly. Future research could examine the effectiveness of different approaches to rank tasks along with what information to display when making recommendations.

Our work opens up several other areas of future research. An interesting extension would be to predict which solvers are likely to join existing tasks, and then incorporate this information when making recommendations. This would be particularly useful for platforms where the relative competition structures of tasks change differently as the tasks progress. Future research could also involve field or laboratory experiments to evaluate the impact of such recommender systems on solver and/or seeker satisfaction. Another interesting avenue for research would be to help solvers create a portfolio of tasks in which to participate by accounting for interactions across tasks in terms of seekers, industries, time lines, and competitors. Some platforms allow solvers to collaborate; the value of recommending collaboration partners also deserves further study. A limitation of our experiments has been our inability to identify variables corresponding to some important predictors of a solver's performance such as personality, interests, education, etc. It would be worthwhile to examine ways in which platforms could obtain such information. For instance, a platform could obtain information on education or interests explicitly by providing suitable incentives to solvers or implicitly from studying solvers' participation patterns. Finally, the focus of this work is on recommending tasks to which a solver has not made submissions yet. Future research could consider recommending tasks to solvers for which making resubmissions would increase the solvers' chances of success.

Acknowledgments

The authors thank the senior editor, the associate editor, and the review team for their detailed suggestions and comments. They also thank the seminar participants at the Workshop on Information Technologies and Systems (WITS), the University of Texas at Austin, Iowa State University, Southern Methodist University, and the Indian School of Business, for valuable feedback.

References

Adomavicius, G., and Tuzhilin, A. 2001. "Using Data Mining Methods to Build Customer Profiles," *Computer* (34:2), pp. 74-82.

Adomavicius, G., and Tuzhilin, A. 2015. "Context-Aware Recommender Systems," in *Recommender Systems Handbook* (2nd ed.), F. Ricci, L. Rokach, and B. Shapira (eds), New York: Springer.

Aggarwal, C. 2016. *Recommender Systems: The Textbook*, Cham, Switzerland: Springer International Publishing Switzerland.

Akin-Little, K., Eckert, T., Lovett, B., and Little, S. 2004. "Extrinsic Reinforcement in the Classroom: Bribery or Best Practice," *School Psychology Review* (33:3), pp. 344-362.

Allison, P. D., and Christakis, N. A. 1994. "Logit Models for Sets of Ranked Items," *Sociological Methodology* (24), pp.199-228.

Ambati, V., Vogel, S., and Carbonell, J. 2011. "Towards Task Recommendation in Micro-Task Markets," in *Proceedings of the 25th AAAI Workshop on Artificial Intelligence,* San Francisco, pp. 80-83.

Anderson, J. R. 1985. *Cognitive Psychology and its Implications* (2nd ed.), New York: Freeman.

Arazy, O., Kumar, N., and Shapira, B. 2010. "A Theory-Driven Design Framework for Social Recommender Systems," *Journal of the Association for Information Systems* (11:9), pp. 455-490.

Archak, N. 2010. "Money, Glory and Cheap Talk: Analyzing Strategic Behavior of Contestants in Simultaneous Innovation Contests on Topcoder.Com," in *Proceedings of the 19th International Conference on World Wide Web*, Raleigh, NC, pp. 21-30.

Archak, N., and Sundararajan, A. 2009. "Optimal Design of Crowdsourcing Contests," in *Proceedings of the 13th International Conference on Information Systems*, Phoenix, AZ.

Atahan, P., and Sarkar, S. 2011. "Accelerated Learning of User Profiles," *Management Science* (57:2), pp. 215-239.

Balabanovic, M., and Shoham, Y. 1997. "Fab: Content-Based, Collaborative Recommendation," *Communications of the ACM* (40:3), pp. 66-72.

Banerjee, S., Sanghavi, S., and Shakkottai, S. 2016. "Online Collaborative Filtering on Graphs," *Operations Research* (64:3), pp. 756-769.

Becker, B. E., and Huselid, M. A. 1992. "The Incentive Effects of Tournament Compensation Systems," *Administrative Science Quarterly* (37), pp. 336-350.

Beggs, S., Cardell, S., and Hausman, J. A. 1981. "Assessing the Potential Demand for Electric Cars," *Journal of Econometrics* (17), pp. 1-19.

Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. 2013. "Recommender Systems Survey," *Knowledge-Based Systems* (46), pp. 109-132.

Bockstedt, J., Druehl, C., and Mishra, A. 2015. "Problem-Solving Effort and Success in Innovation Contests: The Role of National Wealth and National Culture," *Journal of Operations Management* (36), pp. 187-200.

Bockstedt, J., Druehl, C., and Mishra, A. 2016. "Heterogeneous Submission Behavior and its Implications for Success in Innovation Contests with Public Submissions," *Production and Operations Management* (25:7), pp. 1157-1176.

Bolton, R., and Chapman, R. 1986. "Searching for Positive Returns at the Track: A Multinomial Logit Model for Handicapping Horse Races," *Management Science* (32:8), pp. 1040-1060.

Boudreau, K. J., Lacetera, N., and Lakhani, K. R. 2011. "Incentives and Problem Uncertainty in Innovation Contests: An Empirical Analysis," *Management Science* (57:5), pp. 843-863.

Brabham, D. C. 2010. "Moving the Crowd at Threadless: Motivation for Participation in a Crowdsourcing Application," *Information, Communication & Society* (13:8), pp. 1122-1145.

Breese, J. S., Heckerman, D., and Kadie, C. 1998. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence,* Madison, WI, pp. 43-52.

Cameron, L., and Crosby, M. 2000, "It's the Economy Stupid: Macroeconomics and Federal Elections in Australia," *The Economic Record* (76:235), pp. 354-364.

Campbell, J. P. 1990. "Modeling the Performance Prediction Problem in Industrial and Organizational Psychology," in *Handbook of Industrial and Organizational Psychology*, M. D. Dunnette and L. M. Hough (eds.), Palo Altom CAL: Consulting Psychologists Press, Inc., pp. 687-732.

Campbell, J. P., McCloy, R. A., Oppler, S. H., and Sager, C. E. 1993. "A Theory of Performance," in *Personnel Selection in Organizations*, N. Schmitt and W. C. Borman (eds.), San Francisco: Jossey-Bass, pp. 35-70.

Chapman, R. G., and Staelin, R. 1982. "Exploiting Rank Ordered Choice Set Data within the Stochastic Utility Model," *Journal of Marketing Research* (19:3), pp. 288-301.

Chen, K. M., Ingersoll, Jr., J. E., and Kaplan, E. H. 2008. "Modeling a Presidential Prediction Market," *Management Science* (54:8), pp. 1381-1394.

Chen, L., Chen, G., and Wang, F. 2015. "Recommender Systems Based on User Reviews: The State of The Art," *User Modeling and User-Adapted Interaction* (25:2), pp. 99-154.

Chilton, L. B., Horton, J. J., Miller, R. C., and Azenkot, S. 2010. "Task Search in a Human Computation Market," in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, New York.

Clarke, S. R., and Dyte, D. 2000. "Using Official Ratings to Simulate Major Tennis Tournaments," *International Transactions in Operational Research* (7:6), pp. 585-594.

Cohen, W. M., and Levinthal, D. A. 1990. "Absorptive Capacity: A New Perspective on Learning and Innovation," *Administrative Science Quarterly* (35:1), pp. 128-152.

Connelly, B. L., Tihanyi, L., Crook, T. R., and Gangloff, K. A. 2014. "Tournament Theory: Thirty Years of Contests and Competitions," *Journal of Management* (40:1), pp. 16-47.

Constantinou, A., Fenton, N., and Neil, M. 2012. "Pi-Football: A Bayesian Network Model for Forecasting Association Football Match Outcomes," *Knowledge-Based Systems* (36), pp. 322-339.

Cyert, R. M., and March, J. G. 1963. *A Behavioral Theory of the Firm* (2nd ed.), Malden, MA: Blackwell.

David, J., Pasteur, R., Ahmad, S., and Janning, M. 2011. "NFL Prediction Using Committees of Artificial Neural Networks," *Journal of Quantitative Analysis in Sports* (7:2).

Davis, M. 1994. "Folk Music Psychology," *The Psychologist* (7:12), p. 537.

Dechenaux, E., Kovenock, D., and Sheremeta, R. M. 2015. "A Survey of Experimental Research on Contests, All-Pay Auctions and Tournaments," *Experimental Economics* (18:4), pp. 609-669.

Delen, D., Cogdell, D., and Kasap, N. 2012. "A Comparative Analysis of Data Mining Methods in Predicting NCAA Bowl Outcomes," *International Journal of Forecasting* (28:2), pp. 543-552.

Demers, S. 2015. "Riding a Probabilistic Support Vector Machine to the Stanley Cup," *Journal of Quantitative Analysis in Sports* (11:4), pp. 205-218.

DiPalantino, D., and Vojnovic, M. 2009. "Crowdsourcing and All-Pay Auctions," in *Proceedings of the 10th ACM International Conference on Electronic Commerce*, Innsbruck, Austria, pp. 119-128.

Doane, S. M., Pellegrino, J. W., and Klatzky, R. L. 1990. "Expertise in a Computer Operating System: Conceptualization and Performance," *Human–Computer Interaction* (5:2), pp. 267-304.

Edwards, W. 1955. "The Prediction of Decisions among Bets," *Journal of Experimental Psychology* (50:3), pp. 202-214.

Ericsson, K. A. 2006. "The Influence of Experience and Deliberate Practice on the Development of Superior Expert Performance," *The Cambridge Handbook of Expertise and Expert Performance*, K. A. Ericsson, N. Charness, P. J. Feltovich, and R. R. Hoffman, (eds.), Cambridge, UK: Cambridge University Press, pp. 683-703.

Ericsson, K. A., Patel, V., and Kintsch, W. 2000. "How Experts' Adaptations to Representative Task Demands Account for the Expertise Effect in Memory Recall: Comment on Vicente and Wang (1998)," *Psychological Review* (107:3), 578-592.

Erikson, R., and Wlezien, C. 2012. "Markets vs. Polls as Election Predictors: An Historical Assessment," *Electoral Studies* (31), pp. 532-539.

Faraj, S., and Sproull, L. 2000. "Coordinating Expertise in Software Development Teams," *Management Science* (46:12), pp. 1554-1568.

Farr, J. L. 1990. "Facilitating Individual Role Innovation," in *Innovation and Creativity at Work*, M. A. West and J. L. Farr (eds.), New York: Wiley, pp. 1-13.

Forsythe, R., Nelson, F., Neumann, J., and Wright, J. 1992. "Anatomy of an Experimental Political Stock Market," *The American Economic Review* (82:5), pp. 1142-1161.

Ghoshal, A., Menon, S., and Sarkar, S. 2015. "Recommendations Using Information from Multiple Association Rules: A Probabilistic Approach," *Information Systems Research* (26:3), pp. 532-551.

Gneezy, U., Meier, S., and Rey-Biel, P. 2011. "When and Why Incentives (Don't) Work to Modify Behavior," *The Journal of Economic Perspectives* (25:4), 191-209.

Goldberg, D., Nichols, D., Oki, B., and Terry, D. 1992. "Using Collaborative Filtering to Weave an Information Tapestry," *Communications of the ACM* (35:12), pp. 61-70.

Greene, W. N. 2011. *Econometric Analysis* (7th ed.), Upper Saddle River, NJ: Prentice Hall.

Hariri, N., Mobasher, B., Burke, R., and Zheng, Y. 2011. "Context-Aware Recommendation Based on Review Mining," in *Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, Barcelona.

Hausman, J. A., and Ruud, P. A. 1987. "Specifying and Testing Econometric Models for Rank-Ordered Data," *Journal of Econometrics* (34), pp. 83-104.

Ho, S., Bodoff, D., and Tam, K. 2011. "Timing of Adaptive Web Personalization and Its Effects on Online Consumer Behavior," *Information Systems Research* (22:3), pp. 660-679.

Huang, Y., Singh, P.V., and Mukhopadhyay, T. 2012. "Crowdsourcing Contests: A Dynamic Structural Model of the Impact of Incentive Structure on Solution Quality," in *Proceedings of the 33rd International Conference on Information Systems*, Orlando.

Huang, Z., Zeng, D., and Chen, H. 2007. "Analyzing Consumer-Product Graphs: Empirical Findings and Applications in Recommender Systems," *Management Science* (53:7), pp. 1146-1164.

Huckman, R. S., Staats, B. R., and Upton, D. M. 2009. "Team Familiarity, Role Experience, and Performance: Evidence from Indian Software Services," *Management Science* (55:1), pp. 85-100.

Hummel, P., and Rothschild, D. 2014. "Fundamental Models for Forecasting Elections at the State Level," *Electoral Studies* (35), pp. 123-139.

Hunter, J. E. 1986. "Cognitive Ability, Cognitive Aptitudes, Job Knowledge, and Job Performance," *Journal of Vocational Behavior* (29:3), pp. 340-362.

Jeppesen, L. B., and Lakhani, K. R. 2010. "Marginality and Problem-Solving Effectiveness in Broadcast Search," *Organization Science* (21:5), pp. 1016-1033.

Kanfer, R., and Ackerman, P. L. 1989. "Motivation and Cognitive Abilities: An Integrative/Aptitude-Treatment Interaction Approach to Skill Acquisition," *Journal of Applied Psychology* (74), pp. 657-690

Khasraghi, H. J., and Aghaie, A. 2014. "Crowdsourcing Contests: Understanding the Effect of Competitors' Participation History on Their Performance," *Behaviour & Information Technology* (33:12), pp. 1-28.

Kohn, M. L., and Schooler, C. 1983. *Work and Personality: An Inquiry into the Impact of Social Stratification*, Norwood, NJ: Ablex Publishing.

Komiak, S., and Benbasat, I. 2006. "The Effects of Personalization and Familiarity on Trust and Adoption of Recommendation Agents," *MIS Quarterly* (30:4), pp. 941-960.

Koren, Y., Bell, R., and Volinsky, C. 2009. "Matrix Factorization Techniques for Recommender Systems," *Computer* (42:8), pp. 30-37.

Kucherbaev, P., Daniel, F., and Marchese, M. 2014. "Toward Effective Tasks Navigation in Crowdsourcing," in *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, Como, Italy.

Lacy, S. 2012. "Get Over It, Haters: 99designs Has Tipped" (http://pando.com/2012/01/24/get-over-it-haters-99designs-has-tipped/).

Lakhani, K. R., Jeppesen, L. B., Lohse, P. A., and Panetta, J. A. 2007. "The Value of Openness in Scientific Problem Solving," Working Paper, Harvard Business School, Cambridge, MA.

Lee, C. 1997. "An Empirical Study of Boxing Match Prediction Using a Logistic Regression Analysis," paper presented at the Section Statistics Sports, American Statistical Association, Joint Statistical Meeting. Anaheim, CA.

Lessman, S., Sung, M. C., and Johnson, J. E. 2012. "Adapting Least-Square Support Vector Regression Models to Forecast the Outcome of Horseraces," *The Journal of Prediction Markets* (1:3), pp. 169-187.

Li, S., and Karahanna, E. 2015. "Online Recommendation Systems in a B2C E-Commerce Context: A Review and Future Directions," *Journal of the Association for Information Systems* (16:2), pp. 72-107.

Li, Z., Fang, X., Bai, X., and Sheng, O. 2017. "Utility-Based Link Recommendation for Online Social Networks," *Management Science* (63:6), pp. 1938-1952.

Lin, Z., Goh, K., and Heng, C. 2017. "The Demand Effects of Product Recommendation Networks: An Empirical Analysis of Network Diversity and Stability," *MIS Quarterly* (41:2), pp. 397-426.

Liu, T. X., Yang, J., Adamic, L. A., and Chen, Y. 2014. "Crowdsourcing with All-Pay Auctions: A Field Experiment on TaskCN," *Management Science* (60:8), pp. 2020-2037.

Loeffelhold, B., Bednar, E., and Bauer, K. 2009. "Predicting NBA Games Using Neural Networks," *Journal of Quantitative Analysis in Sports* (5:1), pp. 1-15.

Manning, C., Raghavan, P., and Schutze, H. 2008. *Introduction to Information Retrieval*, Cambridge, UK: Cambridge University Press.

McCloy, R. A., Campbell, J. P., and Cudeck, R. 1994. "A Confirmatory Test of a Model of Performance Determinants," *Journal of Applied Psychology* (79:4), pp. 493-505.

McDaniel, M. A., Schmidt, F. L., and Hunter, J. E. 1988. "Job Experience Correlates of Job Performance," *Journal of Applied Psychology* (73:2), pp. 327-330.

Moldovanu, B., and Sela, A. 2001. "The Optimal Allocation of Prizes in Contests," *American Economic Review* (91:3), pp. 542-558.

Nalebuff, B. J., and Stiglitz, J. E. 1983. "Prizes and Incentives: Towards a General Theory of Compensation and Competition," *The Bell Journal of Economics* (14:1), pp. 21-43.

Narayanan, S., Balasubramanian, S., and Swaminathan, J. M. 2009. "A Matter of Balance: Specialization, Task Variety, and Individual Learning in a Software Maintenance Environment," *Management Science* (55:11), pp. 1861-1876.

Oestreicher-Singer, G., and Sundararajan, A. 2012a. "Recommendation Networks and the Long Tail of Electronic Commerce," *MIS Quarterly* (36:1), pp. 65-83.

Oestreicher-Singer, G., and Sundararajan, A. 2012b. "The Visible Hand? Demand Effects of Recommendation Networks in Electronic Markets," *Management Science* (58:11), pp. 1963-1981.

Porter, L. W., and Lawler, E. E. 1968. *Managerial Attitudes and Performance*, Homewood, IL: R. D. Irwin.

Provost, F., Martens, D., and Murray, A. 2015. "Finding Similar Mobile Consumers with a Privacy-Friendly Geosocial Design," *Information Systems Research* (26:2), pp. 243-265.

Quiñones, M. A., Ford, J. K., and Teachout, M. S. 1995. "The Relationship Between Work Experience and Job Performance: A Conceptual and Meta-Analytic Review," *Personnel Psychology* (48:4), pp. 887-910.

Roth, Y., Petavy, F., and de Matos, M. B. 2016. "The State of Crowdsourcing in 2016," *eYaka 2016 Crowdsourcing Report* (eyaka.com).

Sahoo, N., Singh, P., and Mukhopadhyay, T. 2012. "A Hidden Markov Model for Collaborative Filtering," *MIS Quarterly* (36:4), pp. 1329-1356.

Schilling, M. A., Vidal, P., Ployhart, R. E., and Marangoni, A. 2003. "Learning by Doing Something Else: Variation, Relatedness, and The Learning Curve," *Management Science* (49:1), pp. 39-56.

Schmidt, F. L., Hunter, J. E., and Outerbridge, A. N. 1986. "Impact of Job Experience and Ability on Job Knowledge, Work Sample Performance, and Supervisory Ratings of Job Performance," *Journal of Applied Psychology* (71:3), pp. 432-439.

Schulze, T., Krug, S., and Schader, M. 2012. "Worker's Task Choice in Crowdsourcing and Human Computation Markets," in *Proceedings of the 33rd International Conference on Information Systems*, Orlando.

Schwab, A., and Miner, A. 2008. "Learning in Hybrid-Project Systems: The Effects of Project Performance on Repeated Collaboration," *Academy of Management Journal* (51:6), pp. 1117-1149.

Sieg, A., Mobasher, B., and Burke, R. 2010. "Improving the Effectiveness of Collaborative Recommendation with Ontology-based User Profiles," in *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, Barcelona, pp. 39-46.

Simonton, D. K. 1999. "Creativity as Blind Variation and Selective Retention: Is the Creative Process Darwinian?," *Psychological Inquiry* (10:4), pp. 309-328.

Simonton, D. K. 2000. "Creative Development as Acquired Expertise: Theoretical Issues and an Empirical Test," *Developmental Review* (20), pp. 283-318.

Smith, L., Lipscomb, B., and Simkins, A. 2007. "Data Mining in Sports: Predicting Cy Young Award Winners," *Journal of Computing Sciences in Colleges* (22:4), pp. 115-121.

Sonnentag, S., Volmer, J., and Spychala, A. 2008. "Job Performance," *The Sage Handbook of Organizational Behavior* (Volume 1), J. Barling and C. L. Cooper (eds.), London: SAGE Publications, pp. 427-447.

Terwiesch, C., and Xu, Y. 2008. "Innovation Contests, Open Innovation, and Multiagent Problem Solving," *Management Science* (54:9), pp. 1529-1543.

Vallerand, R. J. 1997. "Toward a Hierarchical Model of Intrinsic and Extrinsic Motivation," in *Advances in Experimental Social Psychology*, M. P. Zanna (ed.), New York: Academic Press, pp. 271-360.

Vroom, V. H. 1964. *Work and Motivation*, New York: John Wiley & Sons, Inc.

Willoughby, K. 2002. "Winning Games in Canadian Football: A Logistic Regression Analysis," *The College Mathematics Journal* (33:3), pp. 215-220.

Xiao, B., and Benbasat, I. 2007. "E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact," *MIS Quarterly* (31:1), pp. 137-209.

Yang, J., Adamic, L. A., and Ackerman, M. S. 2008. "Crowdsourcing and Knowledge Sharing: Strategic User Behavior on TaskCN," in *Proceedings of the 9th ACM Conference on Electronic Commerce,* New York: ACM Press, pp. 246-255.

Yang, Y., Chen, P., and Banker, R. 2011. "Winner Determination of Open Innovation Contests in Online Markets," in *Proceedings of the 32nd International Conference on Information Systems,* Shanghai, China.

Yang, Y., Chen, P., and Pavlou, P. 2009. "Open Innovation: An Empirical Study of Online Contest," in *Proceedings of the 13th International Conference on Information Systems,* Phoenix, AZ.

Yuen, M., King, I., and Leung, K. 2012. "Task Recommendation in Crowdsourcing Systems," in *Proceedings of the 1st International Workshop on Crowdsourcing and Data Mining*, Beijing, China, pp. 22-26.

About the Authors

Jiahui Mo is an assistant professor of Information Technology and Operations Management in the Nanyang Business School, Nanyang Technological University, Singapore. She graduated from the University of Texas at Dallas with a Ph.D. in Management Science. Her current research interests are in crowdsourcing, online labor market, online community, social media, and social network.

Sumit Sarkar is the Charles and Nancy Davidson Chair and Professor of Information Systems in the Naveen Jindal School of Management at the University of Texas at Dallas. His research interests include personalization and recommendation technologies, sponsored search, crowdsourcing, data privacy, information quality, web services, and software release strategies. He has been a visiting faculty member at the Indian School of Business, and a visiting scientist at IBM Research Laboratories. He is a Distinguished Fellow of the INFORMS Information Systems Society.

Syam Menon is professor of Information Systems in the Naveen Jindal School of Management at the University of Texas at Dallas. He received his Ph.D. and MBA degrees from the University of Chicago. His current interests include big data, privacy concerns in shared databases, recommendation system design, and software design. His contributions have appeared in *Management Science*, *Information Systems Research*, *MIS Quarterly*, *Operations Research*, *INFORMS Journal on Computing*, and many other outlets including various IEEE and ACM Transactions.

Appendix A

Parameter Estimates

Our focus is on the quality of predictions, and not on the impact of specific variables. We have listed the parameter estimates and corresponding significance levels (Table A) for completeness. Note that many of the variables show up as significant for at least one of the scenarios despite the fact that we have only two weeks of data. The platform will have data that is much richer than what we have been able to collect. That can be used to develop better models, and to make inferences on the impact of each variable.

Table A. Parameter Estimates for Scenarios I (MNL) and III (Ranked-MNL-Ties)

	Variables	Scenario I	Scenario III
Intrinsic	X1: points	– 0.000 (0.000)***	– 0.000 (0.000)***
	X2: capability level	0.316 (0.026)***	0.208 (0.011)***
	X3: # of wins (across all categories)	0.015 (0.002)***	0.006 (0.001)***
	X4: # of times shortlisted (across all categories)	– 0.012 (0.002)***	– 0.004 (0.001)***
Expertise	X5: # of expertise categories	– 0.240 (0.219)	0.006 (0.003)**
	X6: 1 if solver reports expertise in current task category; 0 if not	0.230 (0.210)	– 0.080 (0.103)
	X7: 1 if solver reports expertise in current task subcategory; 0 if not	0.005 (0.006)	0.0005 (0.099)
General Experience	X8: # of tasks already participated in	– 0.018 (0.033)	-0.029 (0.017)*
	X9: aggregate # of solutions submitted by solver	0.004 (0.002)**	0.004 (0.001)***
	X10: success rate	– 1.155 (0.816)	– 0.161 (0.429)
Specialization	X11: # of solver's prior participations in industry of current task	0.154 (0.050)***	0.0001 (0.028)
	X12: fraction of solver's prior participations in industry of current task	– 0.152 (0.250)	0.255 (0.129)**
	X13: # of solver's prior wins in industry of current task	– 0.523 (0.438)	– 0.026 (0.252)
	X14: fraction of solver's prior wins in industry of current task	1.102 (0.617)*	– 0.042 (0.383)
	X15: # of solver's prior participations in subcategory of current task	– 0.019 (0.030)	0.016 (0.016)
	X16: fraction of solver's prior participations in subcategory of current task	0.078 (0.132)	– 0.120 (0.067)*
	X17: # of solver's prior wins in subcategory of current task	0.187 (0.233)	– 0.048 (0.131)
	X18: fraction of solver's prior wins in subcategory of current task	0.708 (0.775)	0.359 (0.420)
Diversity of	X19: # of distinct industries participated in	– 0.056 (0.033)*	-0.057 (0.015)***
	X20: # of distinct subcategories participated in	– 0.054 (0.090)	– 0.014 (0.046)
	X21: # of distinct industries with wins	– 0.030 (0.294)	0.114 (0.166)
	X22: # of distinct subcategories with wins	0.117 (0.235)	0.001 (0.133)
Complexity of Experience	X23: average # of competing solvers per participated task	– 0.003 (0.001)**	– 0.000 (0.001)
	X24: average # of solutions per participated task	0.014 (0.008)*	0.012 (0.004)***
	X25: average listed prize per participated task	0.000 (0.000)	0.000 (0.000)

Numbers in parenthesis are standard errors.
***, **, and * represent significance levels at the 0.01, 0.05 and 0.1 levels, respectively.