Submitted to
manuscript MS-13-01388

Reputation Transferability in Online Labor Markets

(Authors' names blinded for peer review)

Online workplaces such as oDesk, Amazon Mechanical Turk, and TaskRabbit have been growing in importance over the last few years. In such markets, employers post tasks on which remote contractors work and deliver the product of their work online. As in most online marketplaces, reputation mechanisms play a very important role in facilitating transactions, since they instill trust and are often predictive of the employer's future satisfaction. However, labor markets are usually highly heterogeneous in terms of available task categories; in such scenarios, past performance may not be an accurate signal of future performance. To account for this natural heterogeneity, in this work, we build models that predict the performance of a worker based on prior, category-specific feedback. Our models assume that each worker has a category-specific quality, which is latent and not directly observable; what is observable, though, is the set of feedback ratings of the worker and of other contractors with similar work histories. Based on this information, we provide a series of models of increasing complexity that successfully estimate the worker's quality. We start by building a binomial and a multinomial model under the implicit assumption that the latent qualities of the workers are static. Next, we remove this assumption, and we build linear dynamic systems that capture the evolution of these latent qualities over time. We evaluate our models on a large corpus of over a million transactions (completed tasks) from oDesk, an online labor market with hundreds of millions of dollars in transaction volume. Our results show an improved accuracy of up to 25% compared to feedback baselines, and significant improvement over the commonly-used collaborative filtering approach. Our study clearly illustrates that reputation systems should present different reputation scores, depending on the context in which the worker has been previously evaluated and the job for which the worker is applying.

Key words: Online Labor Markets, Reputation, Bayesian modeling, Linear Dynamical Systems

1. Introduction

In recent years, online marketplaces have experienced (and continue to experience) a significant growth in their transaction volume.[1] As significant new entrants, online labor marketplaces, such as oDesk, Amazon Mechanical Turk, and TaskRabbit, follow this trend as well. More precisely, statistics from oDesk, which has the largest revenue share in online workplaces, show an exponential growth in total hours worked per week since 2004; for 2012, the company was reporting transactions of more than 500,000 hours of work time

[1] http://www.statista.com/topics/871/online-shopping/

billed per week.[2] In addition, the online-worker's annual earnings are expected to grow from $1 billion in 2012 to $10 billion by 2020 (Agrawal et al. 2013). On a similar note, Mechanical Turk receives hundreds of thousands of dollars worth of new jobs every day.[3]

A key difference between online labor markets and other marketplaces is that a work project on the former is mainly an '*experience good*'. This means that it's difficult (if not impossible) to predict the quality of the deliverable in advance (Nelson 1970). To resolve this uncertainty, a key solution would be to implement and use reputation systems. Reputation systems provide signals about the past performance of workers (Dellarocas 2003). Such signals are commonly predictive of the quality of users' future performance, in a wide variety of online communities, e.g., online reviews, 'question and answer' (Q&A) communities and others (Danescu-Niculescu-Mizil et al. 2009, Liu et al. 2008b, Lu et al. 2010). Consequently, it is rational to assume that employers, who have limited knowledge of the skills and abilities of a remote contractor, often consult the history of past transactions to better understand whether a contractor is qualified and suitable for the task at hand.

The implicit assumption of most existing reputation systems is that the past working history, for which a participant has been rated for, is similar to the future tasks in which the participant will engage in. However, in many online marketplaces, the tasks that are completed span across a variety of different categories, for example 'Web Development', 'Writing & Translation', 'Sales & Marketing', and so on. Such an assortment naturally forms a highly heterogeneous workplace environment.

Given this heterogeneity, what happens when, for instance, a worker switches to a new type of task? What happens when a contractor, with a background in web development, decides to work on a graphic design task? What can we say regarding the possible outcome of a programming task, for a worker with a history in technical writing? In general, are reputations transferable across categories and predictive of future performance? How can we estimate task affinity and use past information to best estimate expectations of future performance?

Similar questions also apply to 'offline' work, which increasingly leaves traces in online settings (e.g., through profiles on LinkedIn, or online resumes on Monster). As workers progress in their careers, they often transition from one type of job to another (e.g., an

[2] http://web.archive.org/web/20120501051827/https://www.odesk.com/info/about/

[3] http://mturk-tracker.com/arrivals/

engineer to a managerial position). Being able to understand how past performance in one type of job signals transfers to another can significantly improve our ability to better allocate the right workers to the right positions.

Intuitively, we can assume that employers manually check the reputation of workers across categories, and try to 'guess' how these reputations are mapped to the category at hand. A key contribution of the paper is that it allows existing rating systems to explicitly use the type of task that is associated with past ratings. In particular, we propose a set of predictive models that use Bayesian inference to estimate the future performance of a user, based on category-specific past performance. We assume that the category-specific qualities (or skills) of a user are *latent and not directly observable*. However, these skills are reflected into a set of other *measurable* characteristics, such as employer ratings for past projects. Based on these past ratings, we build models that are capable of connecting past performance *across categories* to predict performance in a new category for which we either have zero or very few past data points. We present models of increasing complexity, starting with the assumption that the latent qualities are static, but then alleviate this assumption, allowing the latent qualities to evolve with time or gained experience. To capture this evolution, we use a linear dynamical system (Bishop et al. 2006), which provides predictions that incorporate the dynamic behavior of latent qualities.

While our work has conceptual similarities with the task of recommender systems, our setting is unique: The worker, who is being rated, has the flexibility of moving across task categories, while the items that are rated in existing recommender system settings (movies, songs, products) are static entities that do not evolve over time. Furthermore, in recommender systems, products are identical when used by different users. In the case of labor markets, the workers are evaluated each time in a different task posted by a different employer, introducing multiple levels of heterogeneity (employer heterogeneity, task heterogeneity); in our work, we attempt to directly address the issue of task heterogeneity. Our setting is much closer to the setting of most reputation systems; the key novelty of our work is the introduction of task types for the past ratings, something that, to the best of our knowledge, has not been used in the past. An additional goal of our approach is to estimate task affinities, and understand what types-of-tasks are related in terms of actual worker performance, as this allows for better organization of the task assignments in labor markets. Finally, our experimental evaluation compares against the existing state of the art

in recommender and reputation systems, and illustrates the benefits of using an approach that targets the peculiarities of labor markets.

For our experimental evaluation we use a unique dataset of *real* transactional oDesk data. In particular, this dataset consists of *over a million real transactions* across six different categories from the oDesk marketplace. These transactions capture histories of hundreds of thousands of different contractors. We build and evaluate our models on this data and clearly demonstrate how different categories are correlated with each other, and whether *past performance in a given category contains predictive information about performance in another*. We next compare our models with the existing baseline of uniformly averaging past reputation, and we show that our models perform significantly better, providing up to 25% improvement over the baseline in terms of mean absolute error. Furthermore, we show evidence that our models outperform the collaborative filtering approach. Finally, to examine the robustness of our models, we run simulations with a set of different input distributions. The simulation results give us further confidence regarding the adaptiveness of our models, as well as very insightful information about the performance and appropriateness of each one of our approaches. In particular, our analysis suggests that our approaches should be employed in scenarios where users present skewed past histories towards certain categories/skills/types-of-tasks. To further justify the generalizability of our framework, we present an additional empirical analysis of reputation transferability on Amazon.com. We finally conclude that reputation schemes stand to benefit significantly if they adjust the feedback scores of the participating users to take into account the type of task that a user is expected to complete (or has already completed), as well as the user's past category-specific performance history.

Our study contributes to managerial decision-making in online and other workplaces, and offers an analytics-based approach that can improve the design of online work marketplaces. In particular, our analysis shows a clear and methodologically sound approach for analyzing the correlations between different task categories, and as a result, we provide a more accurate estimate of a worker's performance in a new category. This information is valuable to employers that participate in online labor markets, allowing them to make safer and better-informed hiring decisions. On a parallel trajectory, our analysis can be also used by these marketplaces as a guideline to reduce friction (Brynjolfsson and Smith 2000), by recommending to contractors to apply for tasks that are seemingly out of their scope, but

for which these contractors are highly likely to provide successful outcomes. Furthermore, the increased availability of digital footprints for offline work allows our approaches to be applicable in offline work as well: job transitions are readily available from online resume sites, and signals about work performance are increasingly available (e.g., promotions within the same job, or when moving to a different job). Our framework can be potentially applied in such settings, offering the benefits of our approach in the offline labor market as well.

2. Related work

Related work can be separated into two streams: studies that focus on online reputation systems and studies that explore online labor markets.

2.1. Research in Online Reputation Systems

There are many studies of online reputation mechanisms and how such mechanisms resolve various information asymmetries (Dellarocas 2003, 2006). Common reputation systems use the average of past performance across all transactions, often adding a time-discounting mechanism, or weighting feedback ratings by the size of the transaction. In our work, we explore how past, *task-specific* reputation can be used to predict future performance on *different types* of tasks. We are not aware of other studies that compartmentalize the past reputation of an agent in a market, in order to better understand the ability of a worker to carry out a specific type of task.

Many significant studies in the past focused on the effectiveness of reputation systems. For instance, Bolton et al. (2004) compared trading in an online marketplace with feedback, to a market without feedback, and to a market in which the same people interact with one another repeatedly (partners market). They concluded that (1) online feedback increases transactions' efficacy and (2) that online feedback and one's own past experience do not perfectly overlap in the feedback market. Standifird (2001) studied the importance of a seller's reputational rating and showed that positive ratings are mildly influential compared to negative ratings ,which are strongly influential and detrimental. Furthermore, Resnick et al. (2006) conducted a randomized experiment to study the value of reputation on eBay, and found that buyers had an 8.1% increase in their willingness to pay, in order to buy from a high reputatable, established seller. Bakos and Dellarocas (2011) studied litigation and reputation both as substitutes and as complements, and they found that only when legal costs are too high or damage awards are too low, reputation mechanisms improve efficiency.

Finally, Aperjis and Johari (2010) studied the value of the seller's ratings within some fixed windows of past transactions, and they showed that mechanisms that use information from a larger number of past transactions tend to provide incentives for patient sellers to be more truthful, but for higher quality sellers to be less truthful.

Two other major streams of research that relate to our work (and to reputation systems, in a more general sense) are research on helpfulness of online reviews and research on community question answering (CQA). Most of the studies on online reviews focus on using different review characteristics to estimate the review helpfulness. For example, Kim et al. (2006) use review length, unigrams and product rating; O'Mahony and Smyth (2010) use readability tests; Otterbacher and Arbor (2009) use the topical relevancy, the believability, and the objectivity of the review; Danescu-Niculescu-Mizil et al. (2009) use the difference of a product evaluation with other evaluations of the same product. Furthermore, Liu et al. (2008b) take into account the reviewer's expertise, the writing style of the review, as well as the timeliness of the review. Lu et al. (2010) include in their predictive feature sets information about the author's identities and their social networks. Lappas and Gunopoulos (2010) propose a framework for capturing the overall consensus of the reviewers, on a given subset of item attributes. Tsaparas et al. (2011) propose algorithms for selecting a comprehensive set of a few, high-quality reviews that cover many different aspects of the reviewed item. Ghose and Ipeirotis (2011) examine how the overall history of the reviewer (along with other textual features of a review, such as its subjectivity and readability levels) affects the helpfulness of a review. Our proposed approach, instead of just using the average past reputation of a user, also exploits the correlation among given topic categories and provides more accurate quality estimates.

Our work is orthogonal and complementary to these efforts: in many settings (e.g., online labor markets), we cannot extract features of the past submitted work, and in settings where we can, these extra features are orthogonal to the idea of creating category-specific features.

As we mentioned before, a lot of research focuses on predicting the quality of online answers in 'Community Question Answering' platforms. Towards this direction (i.e., identifying high quality answers), Jeon et al. (2006) propose a framework that uses non-textual features such as click counts, Agichtein et al. (2008) present a model that exploits community feedback (such as links between items or explicit ratings), Bian et al. (2009) develop a

semi-supervised coupled mutual reinforcement frameworkm and Suryanto et al. (2009) propose a model that considers both the answer's quality and relevance. In addition, Liu et al. (2008a) present a prediction model of customers' satisfaction in the 'Yahoo! Answers' platform. Shah and Pomerantz (2010) use Amazon Mechanical Turk workers to label the quality of the answers, and then, they train classifiers that select the highest quality answers. Our work is conceptually different from all these previous studies because it focuses on the associations among different categories: none of these works studied how user reputation in *CQA* platforms is transferable across different topic-categories.

Finally, Adamic et al. (2008) cluster forum categories according to content characteristics and study patterns of interactions among users. In particular, Adamic *et al.* relate categories based on user participation and estimate the user's interests' entropy values. Using these values, they observe that lower entropy is correlated with receiving higher answer ratings, but only for categories where factual expertise is required. Their work deviates from ours in that it does not use prior, category-specific quality to predict the current user quality, as well as in the fact that the authors correlate categories based on user replies and not on how user participation is associated with the quality of completed tasks.

2.2. Research in Online Labor Markets

Current research in Online Labor Markets (OLMs) spans across a variety of problems. Horton (2010) explores market creators' choices of price structure, price level, and investment in platforms. He further discusses possible productivity and welfare implications that these markets can have. Horton and Chilton (2010) present a model of workers supplying labor to paid crowdsourcing. They find that workers work less when the pay is lower, but they do not work less when the task is more time-consuming.

A different stream of work studies the validity of behavioral experiments in these markets. Rand (2012) discusses how Mechanical Turk can be used as a tool for behavioral experimentation. Similarly, Horton et al. (2011) show that online experiments can be just as valid (both internally and externally) as laboratory field experiments. In addition, Berinsky et al. (2012) assess the internal and external validity of experiments performed using Mechanical Turk.

In a different direction, a lot of work focuses on incentivizing workers as well as finding ways to manage the quality of their outcomes. In particular, Shaw et al. (2011) ran an experiment on Mechanical Turk to measure the effectiveness of social and financial incentive

schemes on outcome quality. One of their main findings was that when workers had to think about responses of their peers, combined with financial incentives, they provided higher quality results. Mason and Watts (2010) studied the effect of compensation on performance in the context of two experiments conducted on AMT, and found that increased financial incentives increase the quantity but not the quality of work performed by participants. They also observed an anchoring effect, where workers who were paid more also perceived the value of their work to be greater, and thus were no more motivated than workers who were paid less. Furthermore, Chandler and Horton (2011) ran a natural field experiment on Amazon Mechanical Turk, and found evidence that the user interface and the cognitive biases of the workers play an important role in OLMs. Sheng et al. (2008) studied repeated-labeling strategies in OLMs. Two of their main findings were that (1) repeated-labeling can improve label quality but not always, and (2) that when processing unlabeled data is not free, even the simple strategy of labeling everything multiple times can give considerable advantage. Ipeirotis et al. (2010) presented algorithms that separate workers' ability errors from errors caused by workers' biases. Finally, Ipeirotis and Horton (2011) discussed the need of standardization of basic building block tasks that could make crowdsourcing more scalable.

In 2003, Snir and Hitt (2003) studied costly bidding in online markets and found that higher value projects attract significantly more bids, with lower quality, and that a greater number of bids raises the cost to all participants, due to costly bidding and bid evaluation. Finally, Pallais (2012) ran an experiment on the oDesk.com platform to study the 'cold start' problem (i.e. hiring inexperienced workers) in an OLM. Her experiment showed that both hiring workers and providing more detailed evaluations substantially improves workers' subsequent employment outcomes.

Similarly to our work, Kokkodis and Ipeirotis (2013) studied what happens when a worker transitions between different task categories. In their study, they provided a static approach for studying reputation transferability across different categories in OLMs. We extend this work and provide a more realistic dynamic framework that accounts for worker evolution. We further compare the dynamic and static approaches and thoroughly discuss the resulting business insights and managerial implications.

3. oDesk Data Set

oDesk is a global job marketplace, with a plethora of tools targeted to businesses that intend to hire and manage remote workers. The company reports more than 500,000 hours of work billed per week, as well as an exponentially growing transaction volume of more than $300 million per year.

3.1. Statistics

For our experiments, we use real oDesk transactional data, collected between September 1st and September 21st of 2012. In particular, we analyze a total of 1,029,024 completed oDesk transactions. An instance in our datasets consists of the worker id, the category of the completed task, and the average feedback score that the specific worker received for that task.

In the oDesk platform specifically, after a user completes a task, the employer supplies feedback scores integers between 0 and 5 in the following six fields: 'Availability' ($f1$), 'Communication' ($f2$), 'Cooperation' ($f3$), 'Deadlines' ($f4$), 'Quality' ($f5$), 'Skills' ($f6$). The average of these scores divided by 5 represents the observed quality of the specific task ($\bar{q}$):

$$\bar{q} = \frac{1}{5}\left(\frac{\sum_{i=1}^{6} f_i}{6}\right), \;\; \bar{q} \in [0,1] \cdot \tag{1}$$

The feedback score distribution in our training set is highly skewed towards high scores, with a mean value of 0.89, i.e., approximately 4.5/5 in a five-star scale. Intuitively, this can be explained by the user survival patterns in online communities: users that receive low feedback scores are unable to get hired again, so they leave the marketplace (or rejoin with different credentials (Jerath et al. 2011)). Thus, the majority of the marketplace users end up having high feedback scores. Notice here that such skewed distributions of ratings are very common across many different marketplaces (Hu et al. 2009).

3.2. Task Categories

In this study we examine tasks in six categories: 'Software Development', 'Web Development', 'Design & Multimedia', 'Writing', 'Administration' and 'Sales & Marketing'.

Figure 1 shows the associative probability of categories in our study. Specifically, a directed edge from node j to node k in the graph represents the portion of workers that

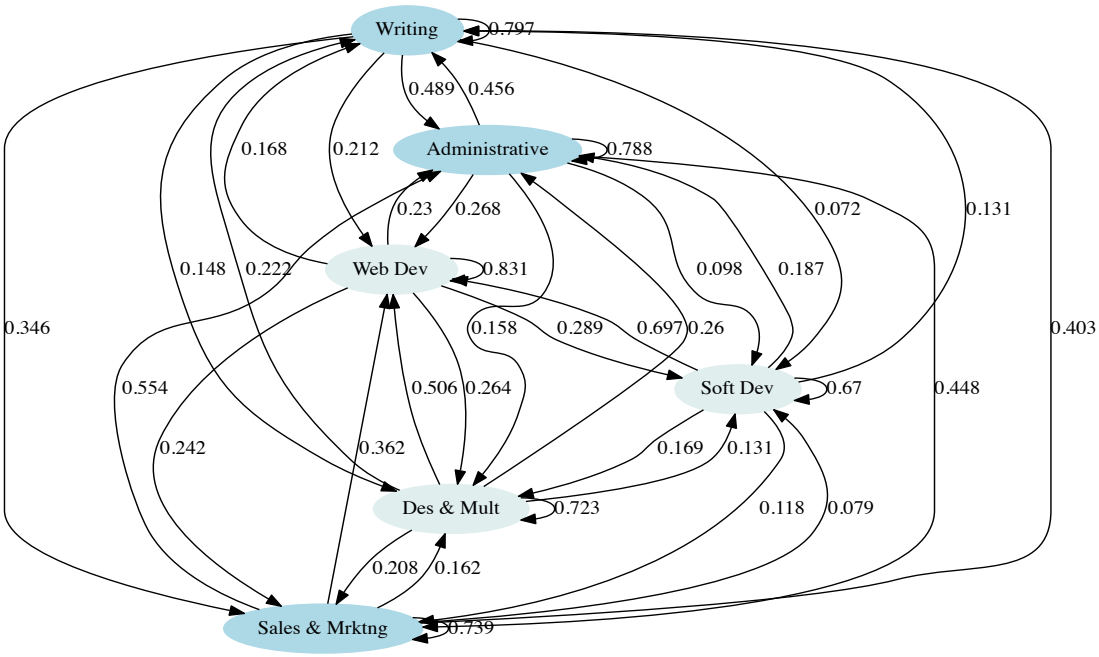


Figure 1 **Associative probabilities across the six categories in our dataset. The graph includes only edges with probabilities greater than 0.05. The weight of an edge from j to k describes the portion of workers that complete a task j who had previously completed at least one task in category k.**

complete a task in category j, given that they have previously completed at least one task in category k. Formally:

$$\text{Weight } (j \to k) = \frac{\#\text{workers in category } j \text{ have previously been in category } k}{\#\text{workers in category } j} \tag{2}$$

The first thing we observe is that users work on the same category more than once, with probabilities close to 0.8 (edges from j to j). For example, the probability of completing at least two tasks in 'Writing' is 0.797, in 'Web Development' is 0.831, *etc.* This is expected and shows a reasonable preference of the workers to keep working on tasks that they are familiar with and build on their expertise. Second, we observe high probabilities in categories that require similar skillsets. For example, from 'Software Development' to 'Web Development' the probability is 0.697, or from 'Sales & Marketing' to 'Writing' the probability is 0.403 *etc.* The final note is that our graph is fully connected, i.e., there is an edge from every node to all other nodes, indicating that the workers in our dataset complete tasks across all available categories with significant probabilities. We will use

this observation to demonstrate in the following sections that properly leveraging past performance data from other categories can significantly improve the prediction of future performance, even when contractors choose to complete a task in the same category.

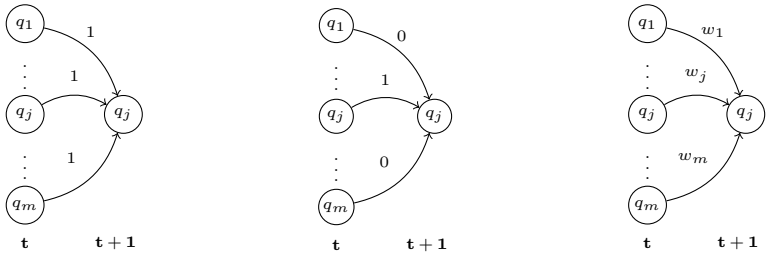
4. Estimating worker's quality

In this section, we present a set of increasingly sophisticated methods for estimating future ratings for a worker, given the past rating history. We initially discuss our latent-variable model, which assumes that each worker has multiple, latent, and potentially correlated qualities across categories, which we try to estimate by observing the ratings received by a variety of users across categories. For the estimation part, we start with a simple binomial Bayesian mode;, which learns the (latent) quality from a user's past ratings in the same category using a binary measurement: whether the feedback will be positive or negative; next, we show how to handle multi-degree ratings using a multinomial model. These two approaches share the assumption that user quality is static. Since workers' quality is potentially dynamic and evolves over time, we then present a linear dynamical system (LDS) approach that captures this evolution. Finally, we extend these approaches by controlling for contractors' specializations, as well as for the development of trust between contractors and employers.

4.1. Model

In all our models, we assume that we have m categories of tasks (e.g., 'Software Development', 'Design & Multimedia', 'Sales & Marketing', *etc.*). We further assume that each user is endowed with a set of m category-specific, latent qualities. We denote with $q_{ij} \in [0,1]$ the quality of a user i in category j ($j \in \{1,\ldots,m\}$). The category-specific quality, q_{ij}, is the probability that, given a task in category j, user i will receive a specific rating for the task. Our goal is to estimate q_{ij} by observing the user's past performance; we are mainly interested in improving the vanilla averaging, mainly in cases where past feedback in a given category is sparse.

In Figure 2, we show a schematic description of the existing baselines and of our approach. In particular, Figure 2(a) shows the existing baseline, which provides an estimation of the next task's quality by uniformly aggregating *all feedback ratings from past tasks*, irrespective of the affinity of past tasks to the current one. Figure 2(b) focuses on estimating the quality of a new task in a specific category, by only using past information from completed tasks in



(a) Feedback from all past tasks treated equal.

(b) Only feedback from tasks in the same category counts.

(c) Adjusting feedback score by affinity.

Figure 2 **Different ways of estimating the quality of a new task in category j at time $t+1$.**

the exact same category, while ignoring feedback from other categories. Finally, our model in Figure 2(c) assigns different weights to each category's feedback, and uses these weights to predict the expected rating for the new task. We discuss this in detail in Section 4.3.

4.2. Learning from past ratings, within category

In the next couple of sections, we describe different methodologies of learning the latent quality of a contractor in a specific category.

4.2.1. Binomial Approach: We start with a very simple setting; we examine the case where a user is performing tasks only within a category j, and the performance rating on these tasks is strictly binary, either 'good' or 'bad'. Given a past history of n tasks within the given category, and assuming that we know the current quality q_{ij} of the worker i in category j, we expect the number x of completed tasks rated as 'good' to follow a binomial distribution:

$$\Pr(x|q_{ij},n) = \binom{n}{x} q_{ij}^x (1-q_{ij})^{n-x}$$

Now, by using basic concepts of Bayesian statistics (Gelman et al. 2004), we can try to infer q_{ij} based on the number of 'good' and 'bad' completed tasks. Specifically, if we assume some prior distribution, $q_{ij} \sim Beta(\alpha,\beta)$, by applying Bayes' theorem we get that:

$$\Pr(q_{ij}|x,n) = \frac{p(x|q_{ij},n)p(q_{ij})}{\int_0^1 p(x|q_{ij},n)p(q_{ij})dq_{ij}}\,.$$

The aforementioned quantity is known to follow the $Beta(\alpha+x, n-x+\beta)$ distrinbution.

Figure 3 shows an example. Assuming a prior distribution $Beta(2,3)$, we show that the resulting probability distribution functions for the two possible outcomes, 'Bad' ($Beta(2,4)$)

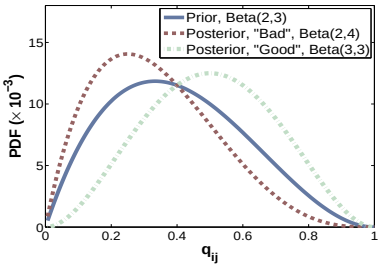


Figure 3 Prior and posterior distributions comparison for both 'Bad' and 'Good' outcomes.

and 'Good' ($Beta(3,3)$). We can observe the shift to the right (i.e., improved quality) when we have a successful outcome, and to the left (i.e., downgraded quality) otherwise.

4.2.2. Multinomial Approach: In reality, binary feedback is typically used for small tasks (e.g., on Amazon Mechanical Turk). For more complex tasks, we often see reputation systems that have multiple grades for feedback (e.g., 5-star ratings are common). To extend the previous model to account for a range of discrete outcomes, we use a multinomial distribution of K possible outcomes (instead of just two):

$$\Pr(\mathbf{x}|\mathbf{q}_{ij}, n) = \binom{n}{x_1, \ldots, x_K} \prod_{k=1}^{K} q_{ij,k}^{x_k} ,$$

where the vector $\mathbf{x} = (x_1, \ldots, x_K)$ encodes the past feedback, with x_k being the number of times that outcome k occurred in the past. The vector $\mathbf{q}_{ij}$ captures the probability that the work of worker i in category j will be of quality k. The conjugate prior for $\mathbf{q}_{ij}$ is the Dirichlet distribution (see Gelman et al. (2004) for more details), with a vector hyperparameter $\boldsymbol{\alpha}$: $\Pr(\mathbf{q}_{ij}|\boldsymbol{\alpha}) \sim \mathcal{D}(\boldsymbol{\alpha})$. Using a Dirichlet prior, and after observing the past feedback $\mathbf{x}$, the posterior distribution becomes:

$$\Pr(q_{ijk}|\mathbf{x}, \boldsymbol{\alpha}) \sim \mathcal{D}(x_k + \alpha_k) \tag{3}$$

In the previous equation, α_k refers to the k dimension of the parameter vector $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_K)'$.

Instead of the approaches presented here, we could also adapt approaches from item response theory (IRT) (Hambleton 1991), for the task at hand. However, most techniques in IRT do not work well with relatively sparse data. IRT models work well for standardized tests, trying to estimate the skills of students that complete houndreds of questions, and identical questions are repeated across thousands of students. Furthermore, there is little focus on inter-task correlations of performance, which is the focus of our work.

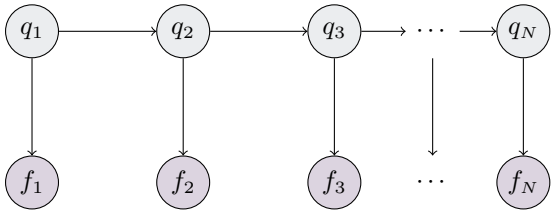


Figure 4 **Graphical model of the linear dynamical system. The figure depicts a series of N observations $\{f_1, ..., f_n\}$ that are a result of the latent qualities of a worker $\{q_1, ..., q_n\}$.**

4.2.3. Linear dynamical system approach: So far, we proposed two static approaches, in the sense that they assign equal weights to past ratings, inherently assuming that the latent qualities are static. In reality, we would expect a more dynamic worker behavior: As users complete more and more tasks, it is sensible to assume that their more recent tasks are more predictive than their initial and older completed tasks. Hence, we will need a dynamic approach that captures this evolutionary worker behavior. In this direction, we propose to use a linear dynamical system (Bishop et al. 2006). For notation simplicity, we drop subscripts and use q to denote the quality of some user i in some category j. For each completed task in a specific category, we observe a feedback score, which we denote as f. The graphical model representation of our approach is shown in Figure 4. We consider that both q and f follow normal distributions, whose means are linear functions of the states of their parents in the graph.

As before, our goal here is to estimate the latent quality q based on the observed feedback f. Assuming that the worker at hand completes N tasks in the same category, the following holds for f and q:[4]

$$p(q_1) = \mathcal{N}(\mu_0, p_0) \tag{4}$$

$$p(q_n|q_{n-1}) = \mathcal{N}(aq_{n-1}, g) \tag{5}$$

$$p(f_n|q_n) = \mathcal{N}(cq_n, r) \tag{6}$$

$$p(q_{n-1}|f_1, ..., f_{n-1}) = \mathcal{N}(\mu_{n-1}, v_{n-1}) \tag{7}$$

We use Equation 4 to initialize our model. By using equations 5 and 6, we predict the next observed outcome. In particular, we use Equation 5 to infer the current quality q_n based on the previous inferred quality q_{n-1}, and then we use Equation 6 to get a distributional

[4] These equations are called *Kalman Filter*.

estimate of the feedback f_n. Finally, with Equation 7 we estimate the quality q_n based on all the feedback observed up to time n.

In equations 4 to 7 we observe two types of parameters: those that are time-independent, and they form a vector of input parameters $\boldsymbol{\theta} = \{a, g, c, r, \mu_0, p_0\}$, and those that are time-dependent $\{\mu_n, v_n\}$ and change at each new observation. For now, we assume that the vector of input parameters ($\boldsymbol{\theta}$) is known, and we concentrate on the estimation of $\{\mu_n, v_n\}$. We recursively estimate these parameters at each state and make quality inferences by the following relations:

$$\begin{aligned} \mu_n &= a\mu_{n-1} + k_n(f_n - ac\mu_{n-1}), \\ v_n &= (1 - ck_n)p_{n-1}, \\ k_n &= \frac{cp_{n-1}}{c^2 p_{n-1} + r}, \\ p_{n-1} &= a^2 v_{n-1} + g \end{aligned}$$

where k_n is known as the *Kalman gain* of the model.

Input parameter estimation: Now that we know how to use our model to make quality predictions, we need to estimate the input parameter vector $\boldsymbol{\theta} = \{a, g, c, r, \mu_0, p_0\}$. To do so, we use expectation maximization (EM). The intuition for our EM algorithm is the following. Assuming that at some particular state of our dynamical system, the parameter vector is $\boldsymbol{\theta}$, we ran the Kalman filter equations to determine the distribution of the latent quality of the worker, $p(q|f, \boldsymbol{\theta})$. For each worker, the complete data log-likelihood is given by:

$$\log L = \ln p(f, q|\boldsymbol{\theta}) = \ln p(q_1|\mu_0, p_0) + \sum_{n=2}^{N} \ln p(q_n|q_{n-1}, a, g) + \sum_{n=1}^{N} \ln p(f_n|q_n, c, r)$$

Our objective function will be the expectation of this log-likelihood w.r.t. $q|\boldsymbol{\theta}$:

$$Q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = E_{q|\boldsymbol{\theta}}(\log L)$$

Assuming that we are estimating our parameters in a set of M sequences of observations of length N, this function becomes:

$$Q(\theta'|\theta) = E_{q|\boldsymbol{\theta}}\left[\sum_{m=1}^{M}\left(\ln p(q_{1,m}|\mu_0, p_0) + \sum_{n=2}^{N} \ln p(q_{n,m}|q_{n-1,m}, a, g) + \sum_{n=1}^{N} \ln p(f_{n,m}|q_{n,m}, c, r)\right)\right]$$

To maximize this function Q, we only need the following sufficient statistics:

$$E[q_n] = \hat{q}_n$$
$$E[q_n q_{n-1}] = \hat{v}_n J_{n-1} + \hat{q}_n \hat{q}_{n-1}$$
$$E[q_n^2] = \hat{v}_n + \hat{q}_n^2$$

where:

$$\hat{q}_n = \mu_n + J_n(\hat{q}_{n+1} - a\mu_n) \tag{8}$$
$$\hat{v}_n = v_n + J_n^2(\hat{v}_{n+1} - p_n) \tag{9}$$
$$J_n = \frac{a v_n}{p_n} \tag{10}$$

Notice that in equations 8 to 10 we include future observations. This set of backward recursions is called Kalman smoother. Now, if we take the derivative of Q w.r.t. to our input parameter vector $\boldsymbol{\theta}$, we get:

$$\mu_0' = \frac{1}{M}\sum_{m=1}^{M} \hat{q}_{1,m} \tag{11}$$
$$p_0' = \frac{1}{M}\sum_{m=1}^{M} \Big(E[q_{1,m}^2] - E^2[q_{1,m}]\Big) \tag{12}$$
$$a' = \frac{\sum_{m=1}^{M}\sum_{n=2}^{N} E[q_n q_{n-1}]}{\sum_{m=1}^{M}\sum_{n=2}^{N} E[q_{n-1}^2]} \tag{13}$$
$$g' = \frac{1}{M(N-1)}\sum_{m=1}^{M}\sum_{n=2}^{N} \Big(E[q_n^2] - 2a'E[q_n q_{n-1}] + a'^2 E[q_{n-1}^2]\Big) \tag{14}$$
$$c' = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} f_n E[q_n]}{\sum_{m=1}^{M}\sum_{n=1}^{N} E[q_n^2]} \tag{15}$$
$$r' = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N} \Big(f_n^2 - 2c'E[q_n]f_n + c'^2 E[q_n^2]\Big) \tag{16}$$

4.3. Learning across categories

In practice, we often have insufficient history within a category, and the distribution of q_{ij} does not provide much information. This results in a q_{ij} distribution with high variance and very uncertain estimates. However, we often have the intuition that even though someone may have no experience in a given category (e.g., in developing Android applications), the past experience in some other, related categories (e.g., iPhone development) can be

predictive of future performance in a new category. Conversely, some categories may give no useful information; for example, past experience as an administrative assistant does not give much information about the ability to carry out a translation task from Chinese to English.

In our model, we assume that the quality of worker i for a category j (q_{ij}) can be estimated based on our knowledge of the history and values q_{ik} for other categories. Since q_{ij} are probability values, we use the method presented by (Clemen and Winkler 1990) to *combine probability estimates from multiple, correlated sources*:

$$\text{logit}(q_{ij}) = \sum_{k=1}^{m} \alpha_{jk} \text{logit}(q_{ik}) + \varepsilon_{ij} \,, \tag{17}$$

where α_{jk}, β_j are data-specific coefficients, ε_{ij} is a random disturbance, and logit is the standard logit function:

$$\text{logit}(x) = \log(\frac{x}{1-x}) \Leftrightarrow \text{logit}^{-1} x = \frac{\exp(x)}{1+\exp x} \,. \tag{18}$$

We compute the parameters of Equation 17 by running linear regression (Greene 2007).

4.4. Estimating quality distributions

We showed before that in the binomial (multinomial) case, $\Pr(q_{ij}|.)$ follows some *Beta* (*Dirichlet*) distribution and that in our linear dynamically system approach, q_{ij} and f_{ij} follow Gaussian distributions. However, to use the regression in Equation 17, we need numeric values for q_{ij} and not distributions. As a result, in order to use the acquired knowledge of the distribution of values of q_{ij} within a framework, that allows only scalar values, we use the following two techniques:

- **Point Estimate (PE)**: We set q_{ij} to be a mean of the user's resulting distribution.

In particular, for the binomial case, for a prior $Beta(\alpha, \beta)$, the value of q_{ij} is:

$$q_{ij} = \frac{x+\alpha}{n+\alpha+\beta}$$

For the multinomial model, with a prior $\mathcal{D}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k, \ldots, \alpha_K)$, the mean value of q_{ij} is:

$$q_{ij} = \frac{1}{K} \sum_{k=1}^{K} k \cdot \frac{x_k + \alpha_k}{n + \sum_{k=1}^{K} \alpha_k} \tag{19}$$

Finally, the point estimates for the normal distributions are their means.

- **Random Sampling (RS)**: With this approach, we instantiate the values q_{ij} by sampling multiple random values from the associated distribution.[5] For the multinomial model, in order to sample from the resulting Dirichlet distribution, we follow the procedure described by Gelman et al. (2004): we draw values $d_1, \ldots, d_K$ from K independent $Gamma(x_k + \alpha_k, x_k + \alpha_k)$ distributions, and then we estimate q_{ijk} as follows:

$$q_{ijk} = \frac{d_k}{\sum_{k=1}^{K} d_k}$$

4.5. User Specificity

Users in OLMs are highly heterogeneous; some of them focus on one category and build an expertise on a specific set of tasks, while others complete tasks that span multiple categories. So far, we have not accounted for this 'user specificity' in our model. Similar to Adamic et al. (2008), we include the entropy of the previously completed tasks' category distribution. In particular, we assume that the entropy of a user i is given by the following:

$$e_i = -\sum_j p(j) \log(p(j)) \,,$$

where $p(j)$ is the probability of worker i to complete a task in category j. When i is a new contractor, we assume that all categories have equal probability (uniform). Intuitively, the lower the entropy, the higher the user specificity in a certain set of categories.

With the inclusion of user-specificity, our regression formula presented in Equation 17 now becomes:

$$\text{logit}(q_{ij}) = \sum_{k=1}^{m} \alpha_{jk} \text{logit}(q_{ik}) + \beta e_i + \varepsilon_{ij} \,.$$

4.6. Developing reliability in the marketplace

As workers complete more and more tasks in the marketplace, their reliability increases; intuitively, a worker who has completed 20 tasks in the marketplace is more trustworthy than a worker that has just joined (also see (Jerath et al. 2011)). In parallel, workers build up their reliability by successfully completing multiple tasks with the same employers. On top of these two observations, the work of Sharara et al. (2011) suggests that highly trusted users are more likely to receive higher ratings. To control for this possibility in our model,

[5] In our work, we sample 40 values from the underlying distribution.

we assume that the number of completed tasks, as well as the number of past collaborations between same worker-employer pairs, are correlated with the expected quality of the worker. Our proposed model now becomes:

$$\text{logit}(q_{ij}(t+1)) = \sum_{k=1}^{m} \alpha_{jk} \text{logit}(q_{ik}(t)) + \beta e_i + \gamma h_i + \delta w_i + \varepsilon_{ij} \,, \tag{20}$$

where h_i is the number of completed tasks of worker i, and w_i is the number of times that worker i has previously collaborated with the employer at hand.

4.7. Increasing robustness

To further improve the robustness of our model, we propose to break down the category-specific quality of each user q_{ij} into the average quality of the category (q_j), as well as the average quality of the user (q_i). The final (extended) version of the proposed approach now becomes:

$$\text{logit}(q_{ij}) = \sum_{k=1}^{m} \alpha_{jk} \text{logit}(q_{ik}) + \beta e_i + \gamma h_i + \delta w_i + \eta q_i + \zeta q_j + \varepsilon_{ij} \,. \tag{21}$$

5. Analysis of oDesk Transactions and Feedback

In this section we build and evaluate our approaches on a real transactional dataset from oDesk.com (also see section 3).[6] Recall that our goal here is to examine whether we can improve the prediction of feedback ratings for contractors that perform a task through oDesk, by incorporating information from other categories.

5.1. Setup

We start our discussion by presenting information about the settings and parameters that we use in our analysis, as well as the experimental procedure that we follow.

5.1.1. Parameters: For the binomial model, we use the threshold θ to discretize the outcome into 'good' and 'bad'. By considering the skewness of the feedback distribution in the oDesk marketplace towards high scores, we choose $\theta = 0.9$.[7] The prior class probabilities under this setting ('bad' vs. 'good') are 24.3% vs. 76.7%.

For the multinomial model, we use the value K to define the number of discrete classes. For our analysis, we set $K = 5$ and uniformly split the $[0, 1]$ interval into five buckets: tasks

[6] The dataset is available, on request, through oDesk.

[7] We also experimented with θ values: $0.6, 0.7, 0.8, 0.9$. In all these experiments, the binomial approach was significantly better than the baseline. However, the best results were achieved with $\theta = 0.9$.

with $\bar{q} \leq 0.2$ fall in bucket 1, tasks with $0.2 < \bar{q} \leq 0.4$ fall in bucket 2, and so on. (See Equation 1 for the definition of $\bar{q}$.) Intuitively, K is a discrete star rating, 1 to 5, assigned to a worker.

Next, for all our models, we use a History Threshold, (η), that represents the worker's minimum number of completed tasks *across all categories* for providing a prediction; on expectation, the higher this threshold, the more accurate our predictions will be. In addition, by varying the η value, we also examine the volatility of the lower bound for observing adequate performance. We evaluate each one of our models for discrete values of $\eta \in \{3, 5, 7, 9, 11, 13, 15, 17, 19\}$.

Finally, for the LDS model, the initialization of our parameters is automatically performed by the EM procedure described before (see Equations 11 to 16).

5.1.2. Procedure: We conduct our experiments as follows: for each of our models, we first use the training data to compute the $logit(q_{ij})$ values for each reviewer i in the set and for each category j, following the point estimate (PE) and random sampling (RS) approaches, described in Section 4.4. We then compute the linear regression coefficients of Equation 17. Finally, we repeat the process for different history threshold values.

Holdout Evaluation: We use holdout evaluation to test our models: we randomly choose 70% of the total workers and their related tasks as our training set, and we consider the remaining 30% of the data to be our test set. In all of our experiments, we build models on the training sets, and evaluate them on the test sets. In this way, we ensure that the resulting performance evaluation metrics are not due to overfitting the data.

Prior Distributions: Our models suggest that we have to choose some reasonable prior distributions. Specifically, for our binomial approach, we assume that $q_{ij} \sim Beta(9, 1)$ (i.e., $\alpha = 9, \beta = 1$). The selection is not random, since it represents a belief that is close to the real prior expectation in the marketplace (which is captured by the feedback scores in our training set).[8] Similarly, for our multinomial approach, we choose a parameter vector $\boldsymbol{\alpha} = (1, 0, 0, 0, 10)$. Again, this selection aims to capture the marketplace's biases, first towards high scores and second towards scores at the extremes of the distribution.

[8] We further experimented with many other priors, including the uniform prior $Beta(1, 1)$. The results were qualitative the same across our evaluations, but slightly worse in comparison with our $Beta(9, 1)$ prior.

5.2. Evaluation Metrics

Our goal here is two-fold: first, we want to have good predictive performance when predicting the quality of a new task; second, we are interested in understanding whether there are significant correlations among different task categories. To estimate the predictive accuracy of our approach, we use the mean absolute error (MAE) across all tasks in our test set, defined as:

$$MAE = \frac{1}{N}\sum_{t=1}^{N}|\hat{q}_t - \bar{q}_t|$$

where N is the total number of tasks in our test set, $\hat{q}_t$ is the predicted quality of task t, and $\bar{q}_t$ is the actual feedback score of task t. We compare our results with two baselines that we discuss next in Section 5.2.1, by computing the MAE percentage *improvement over the baseline*, which we define as follows:

$$Improvement\% = \frac{MAE_{Baseline} - MAE_{model}}{MAE_{Baseline}}$$

We further estimate the information entropies of the resulting error distributions for all our models and the baseline. Intuitively, the entropy of an error distribution represents the uncertainty of the distribution: lower values of entropies indicate more concentrated error distributions (Borda 2011). To compute the entropies, we assume that the error distributions are represented by a random variable $X \in [0,1]$, and we use the following formula:

$$E = -\sum_{i \in D_e} p(X=i)\log p(X=i), \tag{22}$$

where D_e is the resulting error distribution.

5.2.1. Baseline Models We compare the performance of our proposed approaches to two different baselines. The first one averages the past reputation of the workers across categories. In particular:

$$\hat{q}_{ij}(T+1) = \frac{1}{N_i(T)}\sum_{t=1}^{N_i(T)} q_{ij}(t) \tag{23}$$

The second one draws on recommender systems, and predicts the outcome based on workers' similarities (user-user collaborative filtering (CF) (Shapira 2011). As we discussed in the introduction, our setting does not directly map to the commonly-observed recommender

Task (cat)	Contractor 1	Contractor 2	...	Contractor n
Web Dev	1	0.8	...	0.4
Soft Dev	?	0.3	...	1
Writing	?	?	...	0.7
Admin	0.9	0.8	...	0.5
Multimedia	1	?	...	0.4
Sales	?	?	...	1

Table 1 Example of our rating matrix.

systems setting (e.g., the Netflix setting or the Amazon setting). To build a collaborative filtering approach, we assume that contractors are the users, and categories are the items. The ratings are then the received feedback scores for each completed task. An example of the proposed rating matrix is shown in Table 1. An element of this matrix represents the observed average quality of the specific worker (column) on the specific category (row). Question marks ('?') denote that the worker has't completed a task in the respective row category.

User-user Collaborative filtering is based on the premise of finding other users whose past rating behaviors are similar to that of the user at hand. In our case, the algorithm finds workers with past performances, from all available categories, that are similar to that of the worker with whom we want to predict the performance. For example, suppose we are interested in worker w's performance in 'Software Development'. We know that worker w has completed tasks in 'Web Development' and 'Design & Multimedia', with average past performances of 0.8 and 0.9 respectively. User-user CF will find other users (nearest neighbors) that have similar performances in 'Web Development' and 'Design & Multimedia', and use their performances in 'Software Development' to predict the performance of worker w.

User-user CF needs a similarity function to find the nearest neighbors for each user. Multiple similarity metrics are reported in the literature (Ekstrand et al. 2011b). In our scenario, we use the cosine similarity among users:

$$sim(w,z) = \frac{< q_w . q_z >}{||q_w|| ||q_z||}, \tag{24}$$

where q_w (q_z) is the vector of past performance in different categories of worker w (z), and $||.||$ is the L2 norm.

To generate predictions for a worker, we need to compute the worker's neighborhood of neighbors. The size N of this neighborhood is given as input to the algorithm. To select

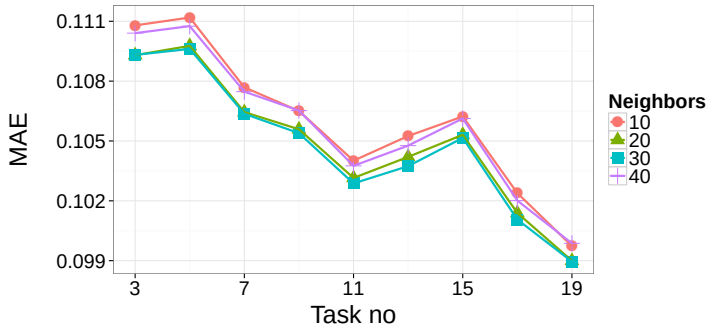


Figure 5 **MAE comparison between different neighborhood sizes.**

the best possible value for N, we evaluate user-user CF in terms of MAE (see 5.2) for $N \in \{10, 20, 30, 40\}$. The results are shown in Figure 5. Better performance is achieved for $N = 30$, which is the neighborhood size that we use in the rest of our analysis.[9]

Finally, for our evaluation, we order our train and test sets by the date of completed tasks, and we retrain our recommender every week, including all the completed tasks of that week.

5.3. Performance Analysis

We start our analysis by discussing the holdout evaluation results, and then, we present the estimated error distribution entropies.

5.3.1. Holdout evaluation: In Figure 6, we show the percentage improvement over the average baseline of our basic approaches (Equation 17, Binomial, Multinomial, and LDS), using the point estimate (PE) (left) and the Random Sampling (RS) (right). On the x-axis we show the number of completed tasks (history η). Note that the baseline is at zero, and **every positive value is an improvement** over the baseline (see Equation 5.2). All our approaches perform better than the baseline, providing an improvement of up to 25%. In addition, all our models show an increasing improvement over the baseline with the history parameter η growth. This behavior is expected, and can be explained by the Bayesian feature of all our approaches (the more input points, the better the posterior distribution estimates). As expected due to its simplicity, the Binomial approach performs worse than the Multinomial, which in turn performs worse than the LDS. Furthermore, all of our approaches perform significantly better than the collaborative filtering approach.

[9] For the implementation of our collaborative filtering approach, we used the Lenskit library (Ekstrand et al. (2011a)).

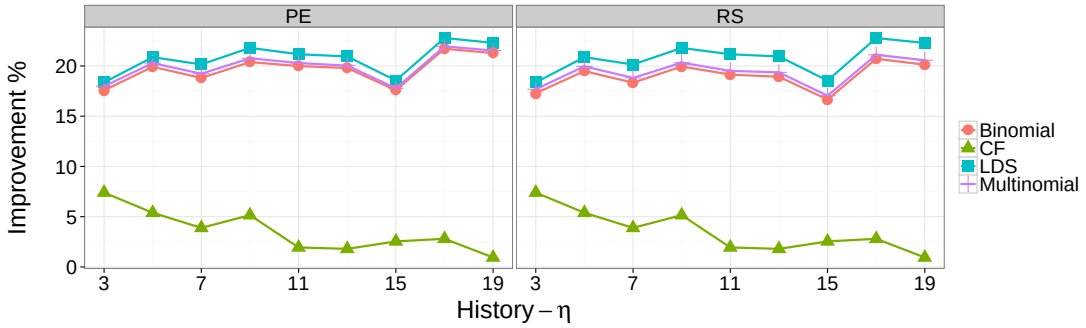


Figure 6 **The holdout improvement of our models (Equation 17) compared to the baselines, as measured by 'mean absolute error' (MAE) , for the point estimate (PE) and random sampling (RS).**

Finally, there are no significant differences between the point estimate and random sampling approaches (left and right figures).

To evaluate how our approaches perform, without aggregating information from other categories, we build single-category models. In particular, for each one of the Binomial, Multinomial, and LDS, we build models that restrict prediction on category-specific history (see 2(b)). In the case where no previous category-specific history is available, we use the across-categories history to estimate performance. The results are shown in Figure 7. The improvement is now up to 3%, significantly lower than the improvement provided by the models that combine information across categories. Second, we can see that LDS learns faster (at 7 observations LDS already performs better than the average baseline) while the Multinomial takes longer (11 observations). The Binomial never outperforms the baseline. It is only fair to point out that this version of our models is not directly comparable to the average baseline, because the latter always accounts for the maximum number of completed tasks (i.e., complete history), while our models only account for the number of tasks that are completed in the category at hand (i.e., category-specific history).

Finally, in Figure 8 we present the performance of the extended version of our models (see Equation 21). The performance of this extended version is very similar to the performance of the basic version of our approaches presented in Figure 6. The main difference is that all three approaches (Binomial, Multinomial and LDS) collapse; this is because of the high impact that all the extra variables (e, h, w, q_i, q_j) have on the quality estimation. We further discuss this in section 5.4, where we review the marginal effects of each variable.

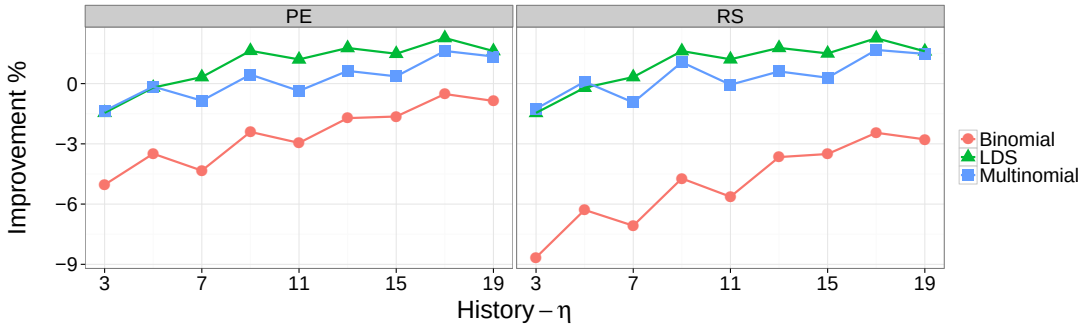


Figure 7 **The holdout improvement of our per-category models compared to the average baselines, as measured by 'mean absolute error' (MAE) , for the point estimate (PE) and random sampling (RS).**

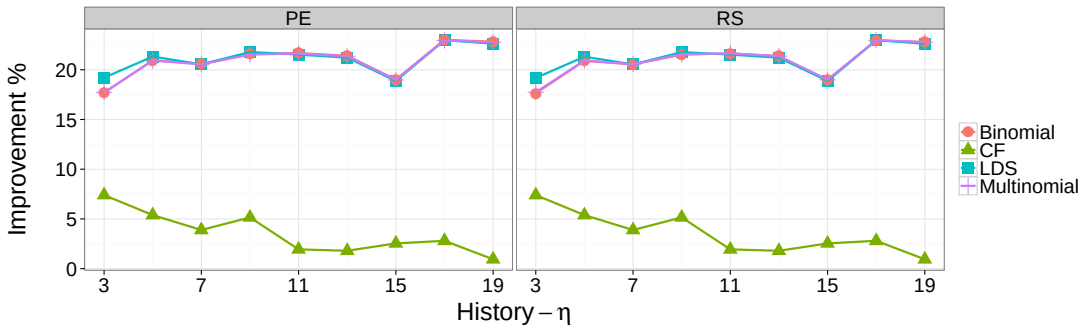


Figure 8 **The holdout improvement of our extended models (Equation 21) compared to the baselines, as measured by 'mean absolute error' (MAE) , for the point estimate (PE) and random sampling (RS).**

5.3.2. Entropies of the error distributions: In Figure 9, we present the information entropies of the error distributions of the basic version of our models (Equation 17), the collaborative filtering approach, and the baseline. As expected, all our models have significantly lower entropy values than the baseline and the collaborative filtering approach, in all histories. Furthermore, all our models have low entropies in the beginning (indicating a good choice of prior values). When the number of completed tasks increases, and up to around seven completed tasks, the entropies slightly increase. This is the adaptation period, where our models try to capture user-specific performances. Beyond that point, and as the number of completed tasks further increases, all our models seem to adapt to the user quality and present lower entropies.

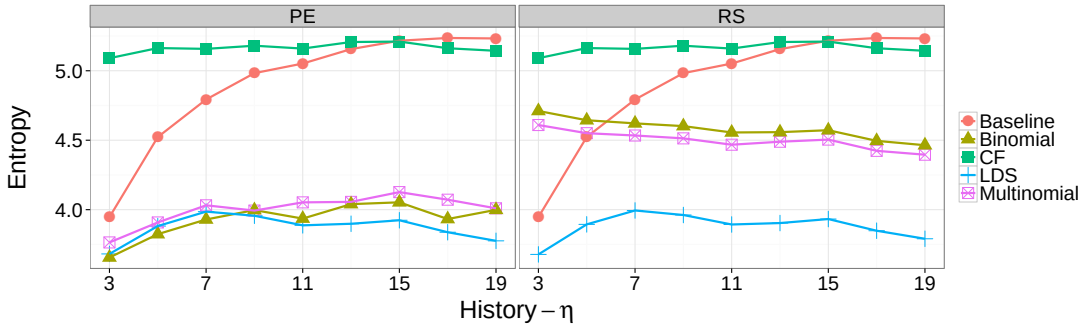


Figure 9 **The entropy values for the resulting error distributions of our basic models (Equation 17) and the baseline, for point estimate (PE) and random sampling (RS).**

5.4. Coefficient analysis: Correlated categories

To study the transferability of each considered task-category, we estimate the marginal effects of the coefficients of Equation 21. In particular, we first solve this Equation w.r.t. q_{ij} (we drop the i index for simplicity):

$$q_j = \frac{\prod_{k=1}^{m}(\frac{q_k}{1-q_k})^{a_{jk}} \cdot \exp(\beta e + \gamma h + \delta w + \eta q_i + \zeta q_j)}{1 + \prod_{k=1}^{m}(\frac{q_k}{1-q_k})^{a_{jk}} \cdot \exp(\beta e + \gamma h + \delta w + \eta q_i + \zeta q_j)} \tag{25}$$

Now we can compute the marginal effects for each q_k, by estimating their partial derivatives w.r.t. to the rest of the categories. In particular we have:

$$\frac{\partial q_j}{\partial q_l} = \frac{\alpha_{jl}(\frac{q_l}{1-q_l})^{\alpha_{jl}}}{q_l - q_l^2} \frac{\prod_{k\neq l}(\frac{q_k}{1-q_k})^{a_{jk}} \cdot \exp(\beta e + \gamma h + \delta w + \eta q_i + \zeta q_j)}{\left(1 + \prod_{k=1}^{m}(\frac{q_k}{1-q_k})^{a_{jk}} \cdot \exp(\beta e + \gamma h + \delta w + \eta q_i + \zeta q_j)\right)^2} \tag{26}$$

For entropy, trust, as well as for the average user quality q_i and the average category performance q_j:

$$\frac{\partial q_j}{\partial h} = \gamma \cdot \frac{\prod_{k=1}^{m}(\frac{q_k}{1-q_k})^{a_{jk}} \cdot \exp(\beta e + \gamma h + \delta w + \eta q_i + \zeta q_j)}{\left(1 + \prod_{k=1}^{m}(\frac{q_k}{1-q_k})^{a_{jk}} \cdot \exp(\beta e + \gamma h + \delta w + \eta q_i + \zeta q_j)\right)^2} \tag{27}$$

We evaluate this formula at the means of the distributions q_k (Greene 2007). Intuitively, for a certain category j, the marginal effect w.r.t. category l (i.e., me_{jl}) implies that if the quality of category l increases by 0.001, and assuming that the qualities of all the other categories remain at their averages, then we would expect on average an increase in the quality of the next task in category j of $0.001 * me_{jl}$. Hence, the higher the marginal effect of category l to category j, the more transferable is the reputation of category l to category j.

	WebDev	SoftDev	Writing	Admin	Des&Mult	Sales	Entropy	# tasks	Rehires	q_i	q_j
Web	0.009***	0.003*	0.002	0.004*	0.004***	0.002	-0.001*	0	0.001***	0.042***	-0.016***
Soft	0.001.	0.011***	0.003	0.006**	0.004.	0.001	-0.001.	-0	0***	0.034***	-0.012****
Writing	0.002.	0.004	0.013***	0	0.006***	0	-0	-0	0***	0.038***	-0.017***
Admin	0.003*	0.007*	0.007***	0.011***	0.001	-0	-0	0	0***	0.041***	-0.015***
Mult.	0.001.	0.006*	0.004*	0	0.01***	-0	-0	-0	0***	0.032***	-0.014***
Sales	0.005***	0.001	0.006**	0.006***	0.019***	0.01***	-0.001	-0	0.001***	0.056***	-0.024***

Table 2 Marginal Effects of the coefficients for the LDS model. Significance codes: '*' 0.001,'**' 0.01, '*' 0.05, '.' 0.01**

The marginal effects of our LDS model are presented in Table 2. An element i, j in Table 2 shows the effect of the j^{th} column category/variable to the i^{th} row category. If we focus on the effects amongst the six categories we consider (first six columns of the table), we observe that for each category, the diagonal effects are all significant and strong. For example, the effect of 'Web Development' on a 'Web Development' task is 0.009 (element (1,1) on the table) and it is the strongest effect on 'Web Development' amongst all categories. The same applies for 'Software Development' (element (2,2)), 'Writing' (element 3,3)), *etc.*

As we mentioned earlier, the higher the effect of one category with another, the more transferable is the reputation. For instance, 'Administrative' tasks have a significant and high coefficient (0.006) on 'Software Development' tasks; Hence, we can say that reputation in 'Administrative' tasks is transferable to 'Software Development' tasks. In Table 2, we observe that all marginal effects in the first six columns are positive. This indicates a positive correlation/transferability between categories. However, not all of the coefficients are significant, and among the significant ones, some have very small effects. Based on the significance and the value of each marginal effect, we suggest the following:

- Reputation in 'Design & Multimedia' transfers to 'Web Development'
- Reputation in 'Administration' transfers to 'Software Development'
- Reputation in 'Design & Multimedia' transfers to 'Writing'
- Reputation in 'Writing' transfers to 'Administration'
- Reputation in 'Web Development' transfers to 'Sales'
- Reputation in 'Administration' transfers to 'Sales'
- Reputation in 'Design & Multimedia' transfers to 'Sales'

If we look at the effects of the rest of the variables, we observe that the average quality of the user (q_i) has a very strong positive effect (between 0.03 and 0.056) in all categories. The entropy effects appear to be very small or insignificant, and not surprisingly, negatively correlated with the expected quality of the outcome (i.e., the higher the user specificity, the higher the expected outcome). The effect of the number of completed tasks appears

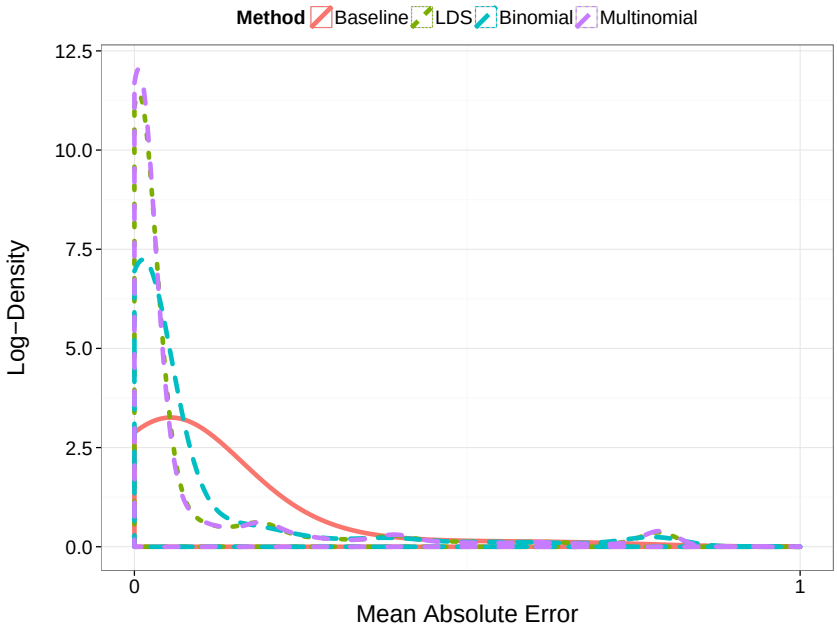


Figure 10 **Mean Absolute Error distributions for all our models and the baseline.**

to be insignificant, while the effect of rehires is positive, but very small. Finally, the average performance in the category at hand (q_j) is negatively correlated with the expected performance.

5.5. Errors broken down by categories

In Figure 10 we show the log-density of the mean absolute error distributions, for all our models and the baseline. We observe that the baseline tends to have higher errors in general, while our proposed models result in error distributions with a close to zero mean. If we compare our models, the LDS and multinomial have similar behavior, while the binomial performs slightly worse. Furthermore, we observe a small peak in the error distributions of the LDS and the Multinomial models concentrated around 0.8. This is due to a few very bad workers that receive systematically low feedback ratings; our models, equipped with priors that reflect the general population, need some time to properly estimate the low scores of these workers. Using an uninformative prior helps in this case, with the trade-off of having a relatively higher error rate for LDS and multinomial, which is still significantly lower than the baseline.

We further analyze the errors of our model by breaking them down by category. In Table 3, we show the improvement of our models over the baseline on transitions between

Transition	Binomial (%)	Multinomial (%)	LDS (%)
Web Dev→Web Dev	22.737	24.845	25.217
Web Dev→Soft Dev	23.721	26.257	25.856
Web Dev→Writing	13.594	14.427	15.090
Web Dev→Admin	21.691	23.837	25.249
Web Dev→Multimedia	22.241	24.792	24.724
Web Dev→Sales	12.328	13.295	13.398
Soft Dev→Web Dev	16.391	18.210	18.600
Soft Dev→Soft Dev	22.715	25.325	25.429
Soft Dev→Writing	23.646	27.203	27.970
Soft Dev→Admin	16.973	18.692	19.665
Soft Dev→Multimedia	38.684	44.095	45.771
Soft Dev→Sales	52.847	59.584	59.010
Writing→Web Dev	10.215	13.213	14.395
Writing→Soft Dev	28.797	34.333	35.069
Writing→Writing	25.900	29.386	29.643
Writing→Admin	23.915	26.743	27.168
Writing→Multimedia	43.897	47.879	49.395
Writing→Sales	15.317	16.432	16.854
Admin→Web Dev	18.320	20.330	20.817
Admin→Soft Dev	42.837	47.725	47.677
Admin→Writing	20.083	22.375	22.430
Admin→Admin	22.850	25.360	25.642
Admin→Multimedia	16.848	18.909	19.971
Admin→Sales	13.938	14.992	15.256
Multimedia→Web Dev	23.002	26.073	26.499
Multimedia→Soft Dev	22.651	25.764	25.834
Multimedia→Writing	25.210	28.801	29.519
Multimedia→Admin	17.592	20.132	21.621
Multimedia→Multimedia	25.383	28.331	28.328
Multimedia→Sales	12.204	13.031	14.936
Sales→Web Dev	13.499	15.038	15.489
Sales→Soft Dev	38.968	40.005	40.143
Sales→Writing	18.225	19.934	20.079
Sales→Admin	19.004	20.903	21.064
Sales→Multimedia	22.554	26.965	27.907
Sales→Sales	16.734	18.404	18.805

Table 3 Improvements broken down by transitions.

categories. For the first block ('Web Development' transitions), we can see that our models perform worse in transitions between 'Web Development' and 'Sales & Marketing', or between 'Web Development' and 'Writing' (improvement between 12% and 16%). In the rest of the transitions the improvement is fairly good, around 24%. Moving to the 'Software Development' transitions, we notice a kind of similar behavior in most of the transitions. However, in the transitions from 'Software Development' to 'Design & Multimedia' and 'Software Development' to 'Sales & Marketing', we observe an improvement of up to 46% and 59% over the baseline, respectively. Similarly, our model significantly improves the predictions for the transitions from 'Admininitration' → 'Software Development' and 'Sales & Marketing' → 'Software Development'.

Feature	Description
Completed Tasks	The number of completed tasks
Entropy	The entropy of the worker, as defined by equation 4.5
Rehire	Whether the instance at hand is a rehire or not

Table 4 Attributes used to investigate when our models fail.

These observations can facilitate a better use of our model: an online labor market can assign different weights on predictions, based on previous evidence derived from such an error-by-category analysis, and hence make inferences about/or merchandise contractors with higher confidence.

5.6. Further insights

To further understand the behavior of our models we propose to build models that capture the probability of providing a wrong prediction. In particular, we create a dataset where each instance has as target variable the prediction error of our approaches, and as feature vector, the attributes shown on Table 4. For each instance in our dataset, we assign an 'Error' label if the error of our prediction was greater than 0.02 [10], and a 'Correct' label otherwise.

Our goal is to predict the probability of having a correct prediction, given the values of our feature sets. We consider Logistic Regression, Naive Bayes, Decision Trees, and Support Vector Machines. We split our data into test and training sets, and perform ten-fold cross-validation. We then evaluate their performances in terms of Accuracy and Area Under the Curve (AUC Provost and Fawcett (2001)). The results are shown in Table 5. We observe that Decision Trees and Logistic Regression have the highest accuracy (71.2%) and AUC scores (0.778).

Of particular interest are the coefficients of Logistic Regression: the 'entropy' has -1.13, the 'completedTasks' 0.09 and the 'workedTogether' 0.473. The marginal effects are -0.281, 0.023, 0.118 respectively. Aligned with intuition, we observe that as the entropy increases, the probability of making a 'correct' prediction decreases. In addition, this probability increases with the number of completed tasks of the worker, as well as with the number of previous collaborations between the same worker-employer pair.

[10] Note that this number was chosen so that we have a balanced dataset, 50% Error instances and 50% Correct instances.

Classifier	Accuracy	AUC
Logistic Regression	0.702	0.778
Support Vector Machines	0.702	0.702
Naive Bayes	0.632	0.765
Decision Trees	0.712	0.774

Table 5 Classification Results.

6. Robustness checks using simulations

While the analysis with the oDesk data indicates that our approach can offer significant improvements in the predictive ability of a reputation system, we also want to examine the robustness of our approach under different settings. For this reason, we present here an analysis with the use of synthetic data, examining the performance of our models with datasets that follow a variety of distributions. In particular, we test the performance of our models in three different scenarios of input distributions:

1. oDesk-like input distribution
2. Uniform input distribution
3. Random input distribution

In the next paragraphs we discuss the data generation and the experimental results for each one of these input distributions.

6.1. Data Generation

In all our synthetic experiments we assume a total of eight categories. The distribution of these categories is defined by a vector $\boldsymbol{c}$. We further use an 8×8 transition matrix M between the eight categories, where an element in the i-th row and j-th column represents the transition probability from category i at time t to category j at time $t+1$. Each user i in our synthetic dataset has a quality vector $\mathbf{q_i} = [q_{i1}, \ldots, q_{i8}]'$ for all available categories. This vector $\mathbf{q_i}$ describes the probability that the user will successfully complete a task in category j. We assume that for each worker i and for a certain category j, the worker's quality follows a normal distribution with mean q_{ij}, and some randomly-defined small variance $\sigma^2 \in (0, 0.2]$. Based on these quality distributions, we sample the performance of a completed task. Finally, each user is assumed to randomly complete between 1 and 40 tasks.

6.1.1. oDesk-like input distribution: In this scenario, we assume that categories form clusters, i.e., their transitional probability from one category to another is higher within the same cluster than across different clusters. We randomly assign probabilities to the

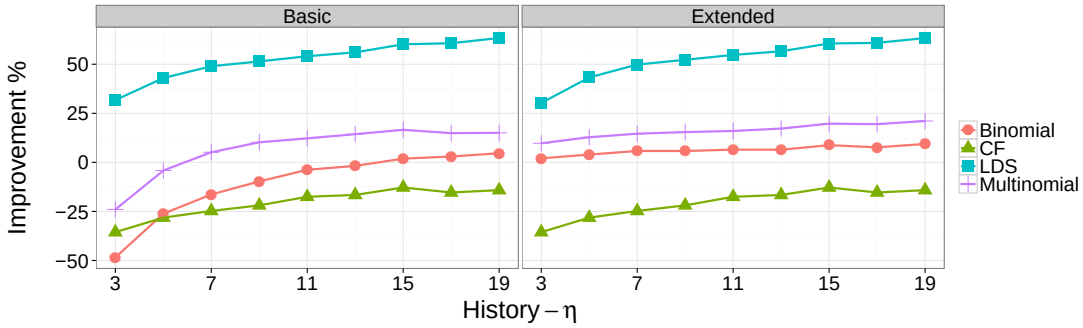


Figure 11 **Synthetic Experiment — oDesk-like inputdistribution**

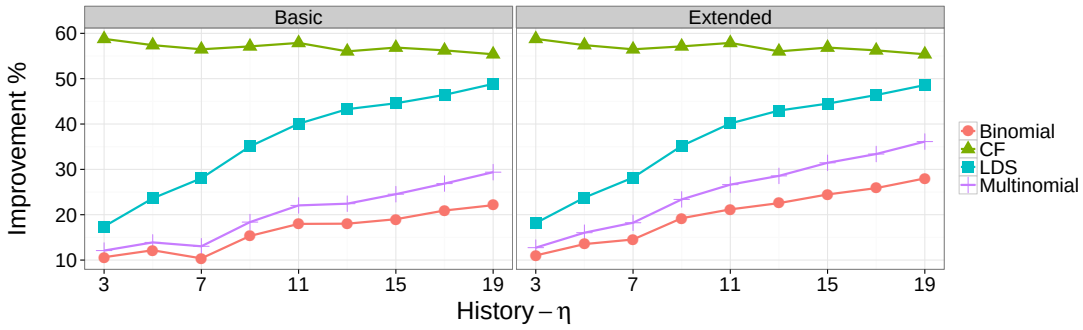


Figure 12 **Synthetic Experiment — Uniform input distribution**

distribution vector $\boldsymbol{c}$. The transition matrix M has low probability values when transitioning happens across clusters (less than 0.05), and high probability values when the worker remains in the same category or when transitioning happens across other categories in the same cluster. Users are assumed to have expertise in one main category (randomly selected) and in a few other similar ones based on the cluster that the main category belongs to.

6.1.2. Uniform input distribution: In this scenario, the transition matrix is uniform (every transition has equal probability = 0.125), the category vector is also uniform, and the user quality vector $\mathbf{q_i}$ is randomly created.

6.1.3. Random input distribution: In this scenario, the transition probabilities are completely random, as is the quality vector for each user and the category vector $\boldsymbol{c}$. Since all qualities are randomly selected in this case, we run our experiments 100 times to get reliable results. We discuss these results next.

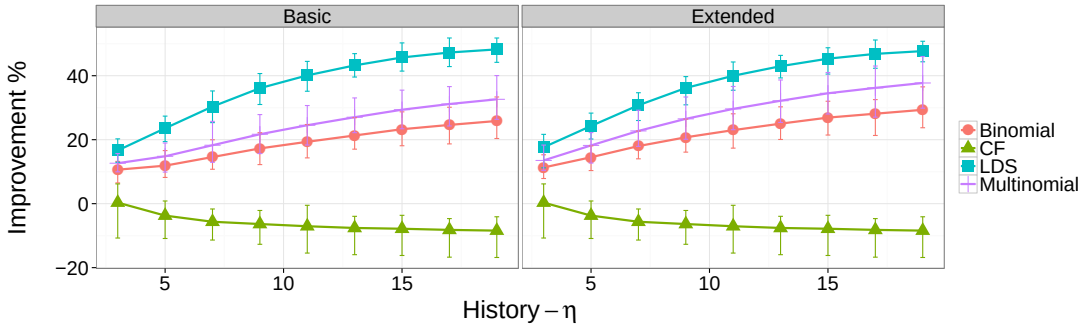


Figure 13 Synthetic Experiment — Random input distribution

6.2. Results on Synthetic Data

After generating the data, we split it into training and test sets, based on users (i.e., the same user cannot be both in the training and test datasets). We use the training sets to build our models and the test sets to evaluate them.

In Figures 11, 12, and 13, we present the results of our simulations. In each figure, we show on the left the performance of our basic model (see Equation 17), and on the right the performance of the extended version of our model (Equation 21).

The first thing we notice is that collaborative filtering performs better than our approaches in the uniform case (Figure 12). To understand this observation, recall that in this scenario, workers don't present a skewed past history towards certain categories: instead, they complete tasks across all categories with equal probability. This results in rating matrices that capture a more accurate average per-category quality of each contractor (see Table 1). This characteristic is crucial to the collaborative approach since it's the base for (1) predicting the quality of a new task and (2) selecting nearest-neighbors that follow similar quality distributions (i.e.,vector $\boldsymbol{q}$) with the worker at hand.

To clarify this, consider the example presented in Table 6. The first row shows the number of completed tasks per category for an oDesk worker. This worker has a preference in tasks of category 1. If he/she chooses to complete a task in category 3, the CF-prediction will be based only on one observation (i.e., uncertain). If the oDesk worker chooses to complete a task in category 2, the CF will find the k-nearest neighbors based on the highly uncertain values – only a single observation – of categories 3 and 4, and of course, on the low-variance estimated quality of category 1. As a result, the CF approach would present quite uncertain estimates and perform poorly (similar to the CF performance in the other two cases of our

	Category 1	Category 2	Category 3	Category 4
oDesk worker	8	0	1	1
Uniform worker	3	2	2	3

Table 6 Example: Number of completed jobs per category

synthetic study, Figures 11 and 13). On the other hand, in the uniform-worker scenario (second row on Table 6), the CF approach will present low-variance predictions based on more data points and hence show an increased performance (similar to Figure 12).

Our proposed approaches perform reasonably well in all cases. In addition, the LDS clearly outperforms the Multinomial model, which in turn outperforms the Binomial approach. Specifically, in the oDesk-like input distribution (Figure 11), LDS provides improvements up to 65%, followed by the Multinomial and the Binomial, which need more observations to provide significant improvements over the average baseline. In the uniform-input distribution, the LDS outperforms the average baseline, however it's not performing as good as the collaborative filtering since, as we explained before, the CF approach is more appropriate for this scenario. Finally, in the Random input distribution, LDS is again a clear winner, followed by the Multinomial and the Binomial approaches.

In conclusion, the synthetic experiments provide evidence that the proposed approaches perform reasonably well independent of the underlying input distribution. Collaborative filtering should be preferred over the proposed approachers only when the users past histories are uniformly distributed across all available categories/skills/types-of-tasks. In the rest, more realistic scenarios, where users present skewed past histories, our approaches – and especially LDS – provide significantly better results. We further support this argument in Appendix A, where we provide an additional analysis on the transferability of Amazon.com reviewers' reputation across different various categories.

7. Managerial Implications, Limitations, and Future Directions

In this study, we presented a variety of models that improve existing reputation systems by predicting a task-specific reputation score based on the past, category-specific reputation history of a worker. We achieved this by assigning different weights to the worker's observed category-specific qualities, which are automatically inferred by analyzing the available reputation ratings. We evaluated our methods by using over a million transactions from oDesk, an online labor market, and we demonstrated that our methods provide more accurate results than existing baselines. Based on our resulting coefficients, we were also

able to infer the affinity of tasks and contractor abilities across different categories of the oDesk marketplace. Finally, by performing a synthetic analysis, we provided evidence that our approaches perform much better than the competing baselines in all realistic scenarios where users present an affinity to certain types of tasks.

Our work has direct implications for the design and scalability of online marketplaces. Consider a real example from the oDesk marketplace. Suppose that a worker has completed a set of 'Sales & Marketing' tasks, and is now applying for a 'Software Development' task. Before, the client would have no accurate information to estimate the performance of this worker, or the client could just use the overall reputation of the worker to get an estimate, with high uncertainty. Our approach limits this uncertainty by 40.1% (see Table 3), hence it provides a significantly more accurate estimate of future performance. As a result, the marketplace builds up trust, increases transaction volume, and creates an environment for better matches and better overall experience for all involved parties.

Furthermore, our approach provides a guideline for many other labor marketplaces. For example, TaskRabbit[11] or LinkedIn[12] can leverage this approach to infer correlations among job types. Even online marketplaces such as Amazon.com can use our approaches to improve the reputation scores displayed for merchants that are active across multiple product categories (e.g., selling photo equipment vs. selling ethnic food), and analyze the abilities of Amazon.com reviewers to provide helpful reviews across different product categories. We include a short analysis of the latter case in appendix A .

Our framework can also be applied to offline marketplaces, if data is available. Since in the offline market workers present skewed past histories towards certain types of jobs, we expect our approaches to perform similar to the online setting. However, the specific observations we made regarding reputation transferability across categories in the oDesk platform cannot be taken as-is to the offline setting. The main reasons are that (1) the definitions of these categories in online labor markets are different than those in the offline market and (2) the tasks in online labor markets are usually short-term, while in the offline setting we frequently deal with long-term employments.

An example of the offline setting that we could deploy our methods is the following: consider an academic department that is responsible to teach a given set of courses. Based

[11] https://www.taskrabbit.com/

[12] http://www.linkedin.com/

on the previous evaluations of the department's professors across the given set of courses, we can build our approaches to estimate the courses' associations. The department then could use this information to perform a better-informed and more efficient course allocation.

An extension of the current line of work is to go beyond categories, and use the 'skill tags' that are used in LinkedIn, oDesk, TaskRabbit, and other marketplaces, to understand affinities of skills and the predictive ability that these skills have when contractors move to new areas. For example, if a contractor knows 'jquery', we may be able to see a good predictive power when transitioning to a skill 'node.js.'[13] Such an analysis can allow for easier filtering and identification of candidates for job openings, even if these candidates do not fully satisfy the requirements of a job opening, therefore significantly improving the efficiency of recruiting processes.

In general, our work provides a clear methodology on how to study whether reputation is transferable across different types of categories, and shows the quantifiable improvements that result from actively using this information to improve current reputation systems. Furthermore, our analysis shows that the proposed approaches can be successfully applied in any situation where users (online or offline) have skewed past histories towards certain types of tasks.

A key limitation that should be mentioned is that our current model is predictive and not necessarily causal. A basic characteristic of predictive models is that they capture the behavior of the existing system, as is. For example, we may predict that a worker who has worked as virtual assistant in the past, with good ratings, is also going to be a good transcriptionist. However, this is a result of a training set in which virtual assistants self-selected and applied for transcription jobs. It is important *not* to assume that *every* virtual assistant will be a good transcriptionist, but this applies to those that self-selected to apply to such jobs. As a result, our methods can best be applied to modify the rating scores shown to the employers when they pick workers, as this 'interference' is not expected to change the self-selection process of applying for jobs much.

Despite the shortcoming listed above, we believe that a multi-category reputation scheme stands to substantially improve the reputation scores, and reputation systems in general, of online (and offline) marketplaces that allow a heterogeneous mix of tasks to be done through

[13] They are both JavaScript-related technologies.

them. LinkedIn, TaskRabbit, oDesk, and even Amazon.com widely host heterogeneous tasks. Past histories can be deceiving when users transition between job categories, engage in a career change, or naturally move into the 'next step' of their career (e.g., from software developer to managing a team of engineers). Our presented framework improves significantly upon the existing reputation systems, and delineates a systematic methodology for improving these systems within a wide variety of settings.

References

Adamic, Lada A., Jun Zhang, Eytan Bakshy, Mark S. Ackerman. 2008. Knowledge sharing and yahoo answers: Everyone knows something. *WWW*.

Agichtein, Eugene, Carlos Castillo, Debora Donato, Aristides Gionis, Gilad Mishne. 2008. Finding high-quality content in social media. *WSDM*.

Agrawal, Ajay, John Horton, Nico Lacetera, Elizabeth Lyons. 2013. *Digitization and the Contract Labor Market: A Research Agenda*. University of Chicago Press. URL http://www.nber.org/chapters/c12988.

Aperjis, Christina, Ramesh Johari. 2010. Optimal windows for aggregating ratings in electronic marketplaces. *Management Science* **56** 864–880.

Bakos, Yannis, Chrysanthos Dellarocas. 2011. Cooperation without enforcement? a comparative analysis of litigation and online reputation as quality assurance mechanisms. *Management Science* **57** 1944–1962.

Berinsky, Adam J, Gregory A Huber, Gabriel S Lenz. 2012. Evaluating online labor markets for experimental research: Amazon. com's mechanical turk. *Political Analysis* **20** 351–368.

Bian, Jiang, Yandong Liu, Ding Zhou, Eugene Agichtein, Hongyan Zha. 2009. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. *WWW*.

Bishop, C.M., et al. 2006. *Pattern recognition and machine learning*, vol. 4. springer New York.

Bolton, Gary E, Elena Katok, Axel Ockenfels. 2004. How effective are electronic reputation mechanisms? an experimental investigation. *Management Science* **50** 1587–1602.

Borda, Monica. 2011. *Fundamentals in information theory and coding*. Springer.

Brynjolfsson, Erik, Michael D Smith. 2000. Frictionless commerce? a comparison of internet and conventional retailers. *Management Science* **46** 563–585.

Chandler, Dana, John Horton. 2011. Labor allocation in paid crowdsourcing: Experimental evidence on positioning, nudges and prices. *Proceedings of the 3rd Human Computation Workshop, HCOMP*, vol. 11.

Clemen, Robert T., Robert L. Winkler. 1990. Unanimity and compromise among probability forecasts. *Management Science* .

Danescu-Niculescu-Mizil, Christian, Gueorgi Kossinets, Jon Kleinberg, Lillian Lee. 2009. How opinions are received by online communities: a case study on amazon.com helpfulness votes. *WWW*.

Dellarocas, Chrysanthos. 2003. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science* .

Dellarocas, Chrysanthos. 2006. Reputation mechanisms. *Handbook on Economics and Information Systems*. Elsevier Publishing, 2006.

Ekstrand, Michael D, Michael Ludwig, Joseph A Konstan, John T Riedl. 2011a. Rethinking the recommender research ecosystem: reproducibility, openness, and lenskit. *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 133–140.

Ekstrand, Michael D, John T Riedl, Joseph A Konstan. 2011b. *Collaborative filtering recommender systems*. Now Publishers Inc.

Gelman, A., J.B. Carlin, H.S. Stern, D.B. Rubin. 2004. *Bayesian Data Analysis*. Chapman & Hall/CRC.

Ghose, Anindya, Panagiotis G. Ipeirotis. 2011. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *TKDE* **23**.

Greene, W.H. 2007. *Econometric analysis*. Prentice Hall.

Hambleton, Ronald K. 1991. *Fundamentals of item response theory*. Sage Publications, Incorporated.

Horton, John J. 2010. *Online labor markets*. Springer.

Horton, John J, David G Rand, Richard J Zeckhauser. 2011. The online laboratory: Conducting experiments in a real labor market. *Experimental Economics* **14** 399–425.

Horton, John Joseph, Lydia B Chilton. 2010. The labor economics of paid crowdsourcing. *Proceedings of the 11th ACM conference on Electronic commerce*. ACM, 209–218.

Hu, Nan, Jie Zhang, Paul A. Pavlou. 2009. Overcoming the j-shaped distribution of product reviews. *Commun. ACM* **52** 144–147. doi:10.1145/1562764.1562800.

Ipeirotis, Panagiotis G, John J Horton. 2011. The need for standardization in crowdsourcing. CHI.

Ipeirotis, Panagiotis G, Foster Provost, Jing Wang. 2010. Quality management on amazon mechanical turk. *Proceedings of the ACM SIGKDD workshop on human computation*. ACM, 64–67.

Jeon, Jiwoon, W. Bruce Croft, Joon Ho Lee, Soyeon Park. 2006. A framework to predict the quality of answers with non-textual features. *SIGIR*.

Jerath, Kinshuk, Peter S Fader, Bruce GS Hardie. 2011. New perspectives on customer death using a generalization of the pareto/nbd model. *Marketing Science* **30** 866–880.

Kim, Soo-Min, Patrick Pantel, Timothy Chklovski, Marco Pennacchiotti. 2006. Automatically assessing review helpfulness. *EMNLP*.

Kokkodis, Marios, Panagiotis G Ipeirotis. 2013. Have you done anything like that?: predicting performance using inter-category reputation. *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 435–444.

Lappas, Theodoros, Dimitrios Gunopoulos. 2010. Efficient confident search in large review corpora. *ECML PKDD*.

Liu, Yandong, Jiang Bian, Eugene Agichtein. 2008a. Predicting information seeker satisfaction in community question answering. *SIGIR*.

Liu, Yang, Xiangji Huang, Aijun An, Xiaohui Yu. 2008b. Modeling and predicting the helpfulness of online reviews. *ICDM*.

Lu, Yue, Panayiotis Tsaparas, Alexandros Ntoulas, Livia Polanyi. 2010. Exploiting social context for review quality prediction. *WWW*.

Mason, Winter, Duncan J Watts. 2010. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter* **11** 100–108.

Nelson, Philip. 1970. Information and consumer behavior. *Management Science.* .

O'Mahony, M. P., B. Smyth. 2010. Using readability tests to predict helpful product reviews. *RIAO*.

Otterbacher, J., A. Arbor. 2009. Helpfulness in online communities : A measure of message quality. *CHI*.

Pallais, Amanda. 2012. Ineffiient hiring in entry-level labor markets. *Available at SSRN 2012131* .

Provost, Foster, Tom Fawcett. 2001. Robust classification for imprecise environments. *Machine Learning* **42** 203–231.

Rand, David G. 2012. The promise of mechanical turk: How online labor markets can help theorists run behavioral experiments. *Journal of theoretical biology* **299** 172–179.

Resnick, Paul, Richard Zeckhauser, John Swanson, Kate Lockwood. 2006. The value of reputation on ebay: A controlled experiment. *Experimental Economics* **9** 79–101.

Shah, Chirag, Jefferey Pomerantz. 2010. Evaluating and predicting answer quality in community qa. *SIGIR*.

Shapira, Bracha. 2011. *Recommender systems handbook*. Springer.

Sharara, Hossam, William Rand, Lise Getoor. 2011. Differential adaptive diffusion: Understanding diversity and learning whom to trust in viral marketing. *ICWSM*.

Shaw, Aaron D, John J Horton, Daniel L Chen. 2011. Designing incentives for inexpert human raters. *Proceedings of the ACM 2011 conference on Computer supported cooperative work*. ACM, 275–284.

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

Snir, Eli M, Lorin M Hitt. 2003. Costly bidding in online markets for it services. *Management Science* **49** 1504–1520.

Standifird, Stephen S. 2001. Reputation and e-commerce: ebay auctions and the asymmetrical impact of positive and negative ratings. *Journal of Management* **27** 279–295.

Suryanto, Maggy Anastasia, Ee-Peng Lim, Aixin Sun Roger H. L. Chiang. 2009. Quality-aware collaborative question answering: Methods and evaluation. *WSDM*.

Tsaparas, Panayiotis, Alexandros Ntoulas, Evimaria Terzi. 2011. Selecting a comprehensive set of reviews. *KKD*.

Appendix A Reputation Transferability on Amazon.com reviews

In this appendix we discuss how our approach can be used in studying the transferability of reputation in a different setting than the one of online labor markets. In particular, we examine how the ability to write helpful reviews on Amazon.com transfers across various product categories. For example, if a reviewer writes great reviews about electronics, what does this say about the reviewer's ability to write similarly helpful reviews for other electronic products, and also, what does it say about the reviewer's ability to write helpful reviews, say, for kitchen appliances? We consider a set of 11,200 reviewers that have reviewed products in five categories: 'Movies', 'Kitchen', 'Video', 'Electronics', and 'Music'. We analyze a total of 78,000 reviews, collected between August 1997 and June 2011.

The metrics and analysis follow the same logic as in Section 5. Figure 14 shows the mean absolute error (MAE) improvements for the extended model (Equation 21), for both point estimate (PE) and random sampling (RS). We observe the same pattern as before: LDS outperforms the Multinomial and the Binomial models, which in turn outperform the Collaborative Filtering approach. Compared to the oDesk case, the overall improvement is lower (but significant), ranging between 2% and 8%. The bad performance of the CF is expected since, as we discussed earlier, Amazon reviewers have skewed histories towards certain categories. Another observation that explains the poor performance of the CF approach is that the input distribution of the Amazon dataset is closer to the oDesk-like distribution of our synthetic experiment (see Figure 15).

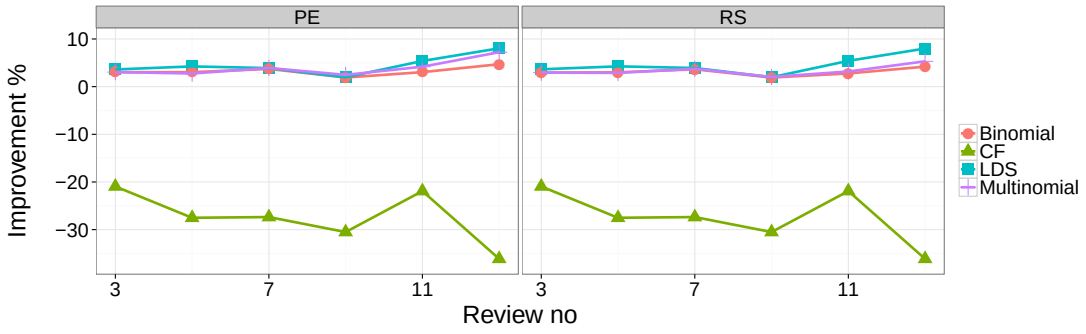


Figure 14 MAE Improvements in the Amazon.com dataset.

The Amazon-reviews scenario presents one main difference compared to the oDesk case. On Amazon, we study the reputation transferability within a 'micro skill' (*review writing*).

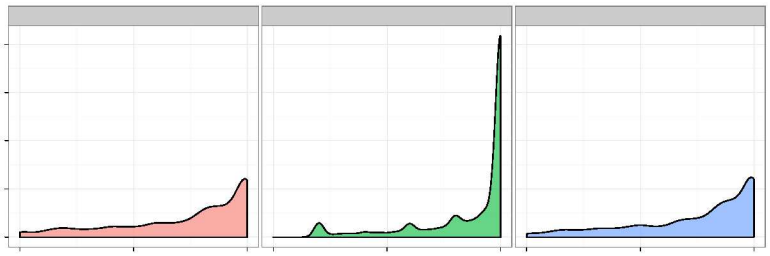


Figure 15 **Input distributions (training sets) of feedback scores/helpfulness for the Amazon,oDesk, and synthetic-oDesk-like scenarios.**

Such a task would fall into the 'Writing' category of oDesk, and would probably be a (potentially small) subcategory. As a result, we consider oDesk as the application of our framework in a 'macro' scale, and we consider Amazon as the application of our framework in a "micro" setting. The proposed approaches perform well in both scenarios, providing us with additional evidence that our work generalizes beyond online labor marketplaces, and in particular, in any setting where users have past performance evaluations, and skewed past histories towards certain types of tasks.