Have you Done Anything Like That? Predicting Performance Using Inter-category Reputation

Marios Kokkodis
mkokkodi@stern.nyu.edu

Panagiotis G. Ipeirotis
panos@stern.nyu.edu

Leonard N. Stern School of Business, New York University
New York, New York 10012, USA
and
oDesk Research
Redwood City, California 94063, USA

ABSTRACT

Online labor markets such as oDesk and Amazon Mechanical Turk have been growing in importance over the last few years. In these markets, employers post tasks on which remote contractors work and deliver the product of their work. 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 future satisfaction of the employer. However, labor markets are usually highly heterogeneous in terms of available task categories; in such scenarios, past performance may not be a representative signal of future performance. To account for this heterogeneity, in our 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 build a multi-level, hierarchical scheme that deals effectively with the data sparseness, which is inherent in many cases of interest (i.e., contractors with relatively brief work histories). We evaluate our models on a large corpus of real transactional data from oDesk, an online labor market with hundreds of millions of dollars in transaction volume. Our results show an improved accuracy of up to 47% compared to the existing baseline.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Online labor markets; Reputation; Task performance prediction; Bayesian modeling

WSDM'13, February 4–8, 2013, Rome, Italy.

1. INTRODUCTION

Online labor markets are growing in importance: Statistics from oDesk show an exponential growth in total hours worked per week since 2004, and the company is currently reporting transactions of more than 500, 000 hours of work-time billed per week.[1] Mechanical Turk,[2] a micro-work site, receives hundreds of thousands of dollars worth of new jobs every day.[3] In these markets, employers post tasks on which contractors work and deliver the product of their work online.

Work is mainly an "*experience good*," meaning it is difficult to observe the quality of the deliverable in advance [19]. A key solution to resolve this uncertainty is the use of online reputation systems, which provide signals about the past performance of workers [6]. Reputation signals are predictive of the quality of users' future performance, in a wide variety of online communities e.g., online reviews, question and answering communities and others [5, 17, 18]. 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 understand better whether a contractor is qualified and suitable for the task at hand.

The implicit assumption of most existing reputation systems is that the past history, for which a participant has been rated for, is similar to the future tasks in which the participant will engage in. However, what happens when, say, a worker switches to a new type of task? For example, 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 history in technical writing? In general, are reputations transferable across categories and predictive of future performance? How can we estimate task affinity and use the information to best estimate expectations of future performance?

As a novel contribution of this paper, 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. Specifically, 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*

[1] http://www.odesk.com
[2] http://www.mturk.com
[3] http://mturk-tracker.com/arrivals/

to predict performance in a new category for which we have either no, or very few, past data points. Since for many category pairs we lack sufficient data to build robust predictors (e.g., from English-Russian translation to web development), we also present a hierarchical model that compensates for the inherent sparseness by using training data from higher-level, more general categories, to compute the cross-category predictive power of past ratings.

For our experimental evaluation, we first present some synthetic experiments that illustrate the effect of various parameters in the performance of our model, showing that our model outperforms existing baselines. Next, we present experiments with *real* data, consisting of hundreds of thousands of real transactions across tens of different categories from the oDesk marketplace, which capture histories of hundreds of thousands of different contractors. We demonstrate how different categories are correlated with each other, and whether past performance in a given category contains predictive information about performance in another. Our results show that our techniques that explicitly take into account category-specific reputation demonstrate significant improvement (of up to 47%) in estimating the feedback score of a worker's next task compared to the existing baseline.

Our bottom line result is clear and simple: 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. For example, the "stars" in the profile of a user bidding for a translation work may be more influenced by past writing tasks compared to past programming tasks. Or even when presenting statistics for the author of a review, the history statistics of the user (e.g., "written 100 reviews, with 98% usefulness") can be adjusted to take into consideration the type of the past reviews (e.g., for a camera review to put more weight on past ratings on digital cameras, and not so much weight on reviews about shoes).

2. BACKGROUND

Reputation systems: There are many studies of online reputation mechanisms and how such mechanisms resolve various information asymmetries [6, 7]. 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 understand better the ability of a worker to carry out a specific type of task.

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*).

Estimating helpfulness of online reviews: Various approaches have been proposed to predict review helpfulness that exploit different sets of features: Soo-Min Kim et al. [14] use review length, uni-grams and product rating; O'Mahony and Smyth use readability tests [20]; Otterbacher and Arbor [21] use the topical relevancy, the believability and the objectivity of the review; and Danescu-Niculescu-Mizil et al. [5] use the difference of a product evaluation with other evaluations of the same product. Furthermore, Liu et al. [17] take into account the reviewer's expertise, the writing style of the review, as well as the timeliness of the review. Lu et al. [18] include in their predictive feature sets information about the author's identities and their social networks. Lappas and Gunopoulos [15] propose a framework for capturing the overall consensus of the reviewers, on a given subset of item attributes. Tsaparas et al. [24] propose algorithms for selecting a comprehensive set of a few, high-quality reviews that cover many different aspects of the reviewed item.

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.

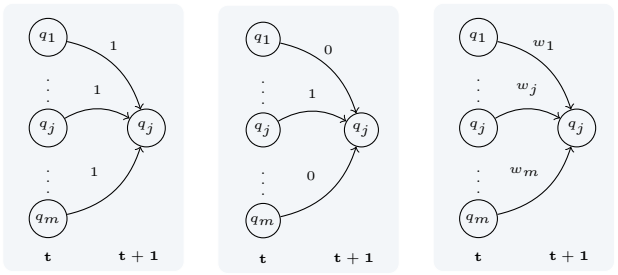
Closely related to our work, Ghose and Ipeirotis examine how the overall history of the reviewer (along with other textual features of a review such as its subjectivity and readability levels) affects the helpfulness of a review [9]. Our proposed approach, instead of just using the average past reputation of a user, exploits also the correlation among given topic categories and provides more accurate quality estimates.

Selecting quality answers in community question answering: A lot of research has been conducted in the past that aims to predict the quality of online answers in "Community Question Answering" platforms. In that direction, Jeon et al. [13] propose a framework that uses non-textual features such as click counts. Liu et al. [16] present a prediction model of customers' satisfaction in the "Yahoo! Answers" platform. Similarly, Agichtein et al. [2] address the task of identifying high quality content in *CQA* platforms by exploiting community feedbacks (such as links between items, and explicit ratings). Furthermore, Bian et al. [3] develop a semi-supervised coupled mutual reinforcement framework for extracting high quality content answers, that requires only relatively few labeled examples. Suryanto et al. [23] propose a different framework to design methods that select high-quality answers from a *CQA* site by considering both the answer's quality and relevance. Shah and Pemrantz [22] use Amazon Mechanical Turk workers to label the quality of the answers, and then they use these labels to train classifiers that select the highest quality answers.

Closely related to our work, Adamic et al. [1] 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.

3. PROBLEM FORMULATION

In this section, we present a set of increasingly sophisticated methods for estimating future ratings for a worker, given the past rating history. We start with a simple binomial Bayesian model that 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. Then we move and discuss our latent-variable model that assumes that each worker has multiple, latent, and interdependent qualities across categories, which we try to estimate by observing the



(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 1: Different ways of estimating the quality of a new task in category j. worker , (b) aggregating only history in the specific weights to

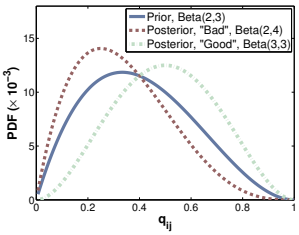


Figure 2: Prior and posterior distributions comparison for both "Bad" and "Good" outcomes.

ratings received by a variety of users across categories. Finally, we propose the creation of hierarchical models that are able to successfully deal with heterogeneous environments and the resulting data sparseness.

In all our models, we assume that we have m categories of tasks (e.g., Software Development, Multimedia & Design, Sales & Marketing). 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; needless to say, we are mainly interested in improving the vanilla averaging mainly in cases where past feedback in a given category is sparse.

In Figure 1 we show a schematic description of the existing baselines and of our approach. In particular, Figure 1(a) shows the existing baseline, which provides an estimation of the next task's quality by uniformly aggregating *all feedback ratings from past tasks*, irrespectively of the affinity of past tasks to the current one. Figure 1(b) focuses on estimating the quality of a new task in a specific category by only using past information from completed tasks in the exact same category, ignoring feedback from other categories. Finally, our model in Figure 1(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 approach in Section 3.2.

3.1 Learning from past ratings, within category

In this case we assume that a user has completed enough tasks in category j and has enough feedback ratings, so that we can use the user's prior performance in this specific category to predict the performance for a new task in this category.

3.1.1 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 [8], 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:

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

which is known to follow $Beta(\alpha+x, n-x+\beta)$.

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

3.1.2 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):

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

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 value q_{ink} 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 be the Dirichlet distribution (see [8] 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:

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

Instead of the approaches presented here, we could adapt approaches from item response theory (IRT) [11], 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 tens and hundreds 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. We present our approach next.

3.2 Learning across categories

In practice, we often have insufficient history within a category and the distribution of $p(q_{ij}|x,n)$ does not provide much information. This results in a q_{ij} distribution with high variance, as well as in very uncertain estimates. A naive approach to estimate q_{ij} would be to simply treat every single past feedback as equal, or, potentially taking a moving average to give higher weight to recent tasks. 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, as well as the overall average quality of the user, q_i, as reflected in the past ratings.[4] Since q_{ij} are probability values, we use the method presented by Clemen and Winkler [4] to *combine probability estimates from multiple, correlated sources*:

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

where α_{jk}, β_j are data-specific coefficients, ε_{ij} is a random disturbance, and logit is defined as follows:

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

Since we can observe the full feedback history of the worker, we use *panel data* [10] and estimate the quality $q_{ij}(t+1)$ at time $t+1$, given the feedback ratings until time t. The resulting equation that we use is the following:

$$\text{logit}(q_{ij}(t+1)) = \sum_{k=1}^{m} \alpha_{jk}\text{logit}(q_{ik}(t)) + \beta_j \text{logit}(q_i(t)) + \varepsilon_{ij} \quad (1)$$

or in matrix notation, for all the available categories that we examine:

$$\begin{bmatrix} \text{logit}(q_{i1}(t+1)) \\ \dots \\ \text{logit}(q_{im}(t+1)) \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \dots & \alpha_{1m} & \beta_1 \\ \dots & \dots & \dots & \dots \\ \alpha_{m1} & \dots & \alpha_{mm} & \beta_m \end{bmatrix} \begin{bmatrix} \text{logit}(q_{i1}(t)) \\ \dots \\ \text{logit}(q_{im}(t)) \\ \text{logit}(q_i(t)) \end{bmatrix} + \begin{bmatrix} \varepsilon_{i1} \\ \dots \\ \varepsilon_{im} \end{bmatrix}$$

The parameters of Equation 1 can be easily computed using regression.

3.3 Estimating quality distributions

We showed before that in the binomial (multinomial) case, $\Pr(q_{ij}|.)$ follows some *Beta* (*Dirichlet*) distribution. However, to use the regression in Equation 1 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:

[4]The q_i factor can be considered as a global smoothing factor in the model, which we know to be correlated with the q_{ik} values.

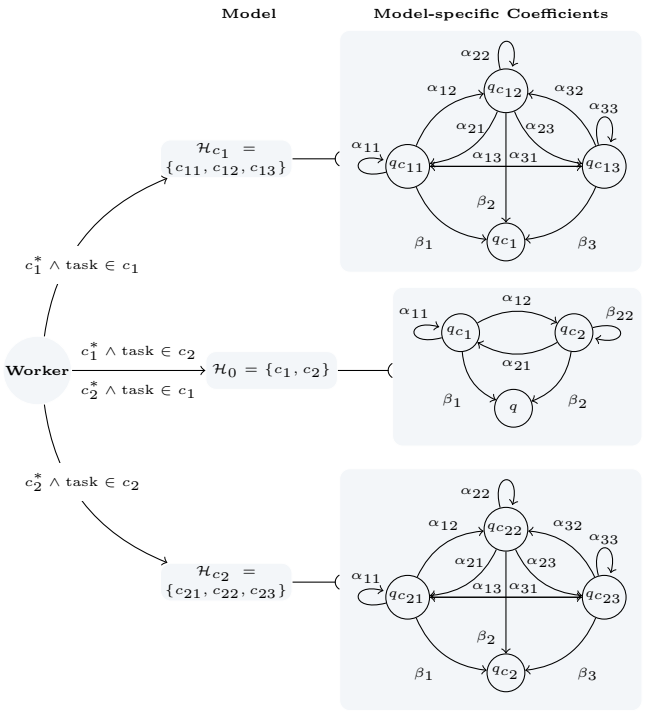


Figure 3: Model selection for a user with the majority of past tasks in category c_l^* and current task in category c_l, where $l = \{1, 2\}$. In the presented example, $L = 2, L_1 = L_2 = 3$.

- **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 $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k, \dots, \alpha_K)$, the mean value of q_{ij} is:

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

- **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. [8]: we draw values $d_1, \dots, d_K$ from K independent Gamma distributions, $Gamma(x_k+\alpha_k, x_k+\alpha_k)$, and we then estimate q_{ijk} as follows:

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

3.4 Dealing with sparseness

So far we have discussed how we can predict the quality for a new task, given the history of a worker across all available categories. However, when the number of categories grows, we

[5]In our work, we sample 30 values from the underlying distribution.

often face problems of data sparseness: there are not enough points to adequately estimate, in a robust manner, the coefficients of Equation 1 (i.e., we end up overfitting the training data and get poor results when dealing with unseen data).

To deal with sparsity, we propose to pair our technique with a hierarchical clustering algorithm. In particular, we cluster categories into L number of abstract groups, for which we have enough "*transitional data points*" to estimate the necessary coefficients. By transitional data points we mean that there is a sufficient number of users that worked in both categories, allowing us to estimate the predictive power of feedback in one category to predict future performance in another.

These categories represent the top level model, $\mathcal{H}_0 = \{c_1, \ldots, c_l, \ldots, c_L\}$. Then, by decomposing each abstract category c_l in the top level, we can create a number of second level models, each one with L_l number of categories. For example, if we decompose category c_1, we will get $\mathcal{H}_{c_1} = \{c_{11}, c_{12}, \ldots, c_{1L_1}\}$; if we decompose category c_2 we will get $\mathcal{H}_{c_2} = \{c_{21}, c_{22}, \ldots, c_{2L_2}\}$, etc. By repeatedly applying this procedure we can reach a desired category-specific detail level, where all categories in a specific set will have enough data points for the necessary coefficients' estimation.

Now that we have different models at different levels, we need to choose which one suits best for each user. To do that, we follow the procedure shown in Figure 3: we first find the abstract category in the highest level model $\mathcal{H}_0$ that contain the majority of the worker's past tasks. Say this category is c_l^*. If the current task that we examine and want to predict its quality is also of the same category c_l^*, we use the coefficients of the model in the lower level that decomposes c_l^*, $\mathcal{H}_{c_l^*}$. Otherwise, we use the coefficients of $\mathcal{H}_0$. We move to lower levels in a similar fashion. The example illustrated in Figure 3 assumes a specific case where $L = 2$, and $L_1 = L_2 = 3$. Note that in this approach, all three models have different coefficients (rightmost column of diagrams in Figure 3).

4. EXPERIMENTAL SETUP

In this section, we discuss the experimental setup we use to evaluate our approaches. We start by discussing the different tuning parameters for our model, and then present the evaluation metrics and the baselines that we use.

4.1 Threshold Parameters

We use three parameters to evaluate our models. For the Binomial Model, we use parameter $\theta \in (0, 1)$ which is a threshold that separates "good" outcomes from the "bad" (i.e., $q > \theta$ is a "good" outcome while $q \leq \theta$ is a "bad" outcome). For the multinomial approach, we use a parameter for the number of discrete classes (K). Finally, for both of our approaches, we use a History Threshold, (η), that represents the minimum number of completed tasks that a worker needs to complete *across all categories* before our model provides a prediction; of course, for workers with very limited history (i.e., one or two tasks) the predictions are going to be worse than with workers with longer histories. By varying the η value, we want to examine what is the lower bound for getting good performance, even for workers with limited number of ratings.

4.2 Evaluation Metrics

Our goal here is twofold: 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 accuracies of our approach, we use the mean absolute error (MAE) across all tasks in our test set, which we define as follows:

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

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

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

4.3 Baseline Methods

The baselines we use to compare our results predict the quality of a new task by uniformly considering the past history of a worker in all categories. For our binomial approach, the predicted baseline quality $\hat{q}_{ij}^{(Bin)}$ of a new task by worker i in category j at time $T + 1$ will be:

$$\hat{q}_{ij}^{(Bin)}(T+1) = \frac{1}{N_i(T)} \sum_{t=1}^{N_i(T)} \mathbf{1}_{\{\bar{q}_{ij}(t) > \theta\}}$$

where $\mathbf{1}_{\{\bar{q}_{ij}(t) > \theta\}}$ is the indicator function for "good" outcomes, and $N_i(T)$ is the total number of completed tasks by worker i at time T. Similarly, for the multinomial approach, the baseline predictions $\hat{q}_{ij}^{(Mult)}$ is:

$$\hat{q}_{ij}^{(Mult)}(T+1) = \frac{1}{N_i(T) \times K} \sum_{k=1}^{K} \sum_{t=1}^{N_i(T)} k \mathbf{1}_{\{\bar{q}_{ij}(t) \in (\frac{k-1}{K}, \frac{k}{K}]\}}$$

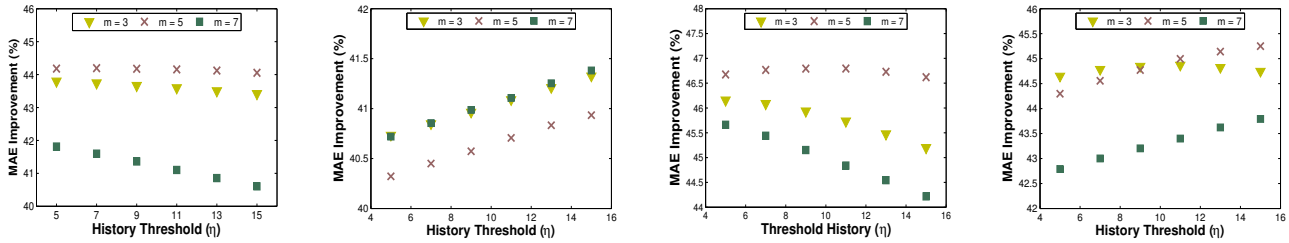
5. SYNTHETIC DATA EXPERIMENTS

In this section, we describe the synthetic experiments we ran to study the effect of various parameters (see Section 4.1) on our algorithms. We conducted the experiments with synthetic data, in addition to the experiments with real data described in Section 6, in order to understand better the performance of our algorithms under different settings, something that is not directly possible with real data.

5.1 Data Generation

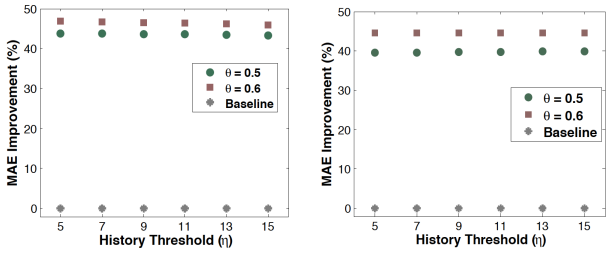
Dense data, simple model: We first generated data that is "dense" enough and for which we can build our models as described in Section 3.2. Specifically, we generated m different categories, where $m \in \{3, 5, 7\}$, and we randomly assigned prior probabilities across these categories. Next, we randomly created an $m \times m$ transition matrix, where the element in the i-th row and j-th column represents the probability that given a completion of a task in category i at time t the next task (at time $t + 1$) will be in category j.

Next, for each user i in our synthetic dataset, we created a quality vector $\mathbf{q_i} = [q_{i1}, \ldots, qim]$ for all available categories. This vector $\mathbf{q_i}$ describes the probability that the user will successfully complete a task in category j. Then, each user is assumed to randomly complete between 10 and 50 tasks. We pick the category distribution as follows: Based on the global category distribution we estimated before, we pick an initial category for each user; then, based on the transition matrix, we draw the next task category for the user and we continue the random walk until exhausting the tasks. Knowing the category of the task, we pick a performance for the task using



(a) $\theta = 0.5$, PE (b) $\theta = 0.5$, RS (c) $\theta = 0.6$, PE (d) $\theta = 0.6$, RS

Figure 4: Synthetic Experiment: the *improvement* of our models compared to the existing baselines, as measured by "mean absolute error" (MAE) for the (a) Binomial model point estimate (PE) and random sampling (RS), for different number of categories ($m = 3, 5, 7$), and different θ-thresholds ($\theta \in \{0.5, 0.6\}$).



(a) PE (b) RS

Figure 5: Synthetic Experiment - Hierarchical Model: the *improvement* of our hierarchical model compared to the baselines, as measured by "mean absolute error" (MAE) for point estimate (PE) and random sampling (RS), for different θ-thresholds.

the latent quality values q_{ij}. For the binomial case, and for a task in category j we conduct a Bernoulli trial with probability of having "good" performance being q_{ij}. Analogously, for the multinomial case.

Sparse data, hierarchical model: To evaluate the robustness of our hierarchical approach, we created three higher-level categories (clusters), which in turn consist of three lower-level categories. In contrast with the dense case, the transition matrix has low probability values when transitioning across clusters (less than 0.05), and high probability to transition in the same category or in other categories within the same cluster. In this way, we simulate a scenario where users have expertise in one category and in a few other similar ones, and they continue to show a preference for tasks from these categories.

5.2 Results for Synthetic Data

After generating our data, we split it into training and test sets, based on users; the same user could not be both in the training and test data sets. Then, we use the training sets to build our models and the test sets to evaluate them.

In the "dense data" case, we build simple models (no hierarchies). Figures 4(a) to 4(d) show the results. We can see that in all different settings we tried (different θ values, different number of categories, different history thresholds η) our approach outperforms the baselines. Note that we report the *improvement* over the baseline, so any value above 0 indicates better performance than the baseline. We further notice that the improvement does not significantly vary with the number of categories (m) or the history threshold (η), which is a good sign: This indicates that our approach scales nicely with an increasing number of categories, and can also work with limited work histories, respectively.

Next, in Figures 5(a) and 5(b) we present the results of our hierarchical model implementation with sparse data. Similarly with before, we observe that our hierarchical approach provides a significant improvement compared to existing baselines. Further, we notice again that this improvement is sustained across the different values of θ-threshold and history threshold η, indicating again that our approach works with limited work histories, even with sparse data.

6. REAL DATA EXPERIMENTS: ODESK

In this section we apply our approaches to real transactional data from the online labor site oDesk.com.[6] We examine whether we can improve the prediction of feedback ratings for contractors that perform a task through oDesk. In addition, we also conducted experiments using data from Amazon.com product reviews; we examined the performance of *reviewers*, measured as the usefulness of the submitted product reviews, but due to space restrictions and due to the qualitatively similar results, we do not include Amazon results in this paper. We focus instead solely on the oDesk dataset, which we believe is inherently more interesting: For product reviews there is already a significant amount of literature (e.g., [14, 17, 20, 21]) that can predict the quality of a review using purely textual analysis; we do not have such an ability for online work contracts so we believe that the usefulness of our models is higher in the oDesk setting.

6.1 Data

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 USD per year.

For all our experiments, we build and test our models on real oDesk transactional data. In particular, we analyze a total of 666,229 completed tasks by 115,436 individual workers, along with their feedback scores. We vertically (i.e., by user) split this data into training and test sets. More specifically, we randomly choose 90% of the total workers and their related tasks as our training set, and we consider the remaining 10% of the data to be our test set. In all of our experiments, we

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

Level	Coeff.	Correlates Categories
Gen. ($\mathcal{H}_0$)	α_{11}	Technical - Technical
	α_{12}	Technical - Non Technical
	β_1	Technical - Overall
	α_{21}	Non Technical - Technical
	α_{22}	Non Technical - Non Technical
	β_2	Non Technical - Overall
Technical ($\mathcal{H}_{c_1}$)	α_{11}	Software developer - Software developer
	α_{12}	Software developer - Web developer
	α_{13}	Software developer - Design & Multimedia
	β_1	Software developer - Overall
	α_{21}	Web developer - Software developer
	α_{22}	Web developer - Web developer
	α_{23}	Web developer - Design & Multimedia
	β_2	Web developer - Overall
	α_{31}	Design & Multimedia - Software developer
	α_{32}	Design & Multimedia - Web developer
	α_{33}	Design & Multimedia - Design & Multimedia
	β_3	Design & Multimedia - Overall
Non Technical ($\mathcal{H}_{c_2}$)	α_{11}	Writing - Writing.
	α_{12}	Writing - Administration
	α_{13}	Writing - Sales & Marketing
	β_1	Writing - Overall
	α_{21}	Administration - Writing
	α_{22}	Administration - Administration
	α_{23}	Administration - Sales & Marketing
	β_2	Administration - Overall
	α_{31}	Sales & Marketing - Writing
	α_{32}	Sales & Marketing - Administration
	α_{33}	Sales & Marketing - Sales & Marketing
	β_3	Sales & Marketing - Overall

Table 1: Interpretation of different coefficients in all three levels.

build models on the training set, and evaluate them on the test set. In this way, we ensure that the resulting performance evaluation metrics are not due to overfitting the data.

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 (in integers of 0 to 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 ($\bar{q}$):

$$\bar{q} = \frac{1}{5}\left(\frac{\sum_{i=1}^{6} f_i}{6}\right), \quad \bar{q} \in [0,1] \cdot \tag{2}$$

The feedback score distribution in the training set is 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). On the other hand, users that receive high feedback scores, tend to continue to use the marketplace. Thus, the majority of the marketplace users end up having high feedback scores. Notice that such skewed distributions of ratings are very common across many different marketplaces [12].

6.2 oDesk hierarchical models

In our experiments, we examine a set of the following six categories: "Software Developer", "Web Developer", "Design & Multimedia", "Writing", "Administration" and "Sales & Marketing". Figure 7 shows the transition probabilities with values above 0.05, across our categories. The graph shows the probability that, given that a user has just completed a task in category j will next complete a task in category k. We can observe

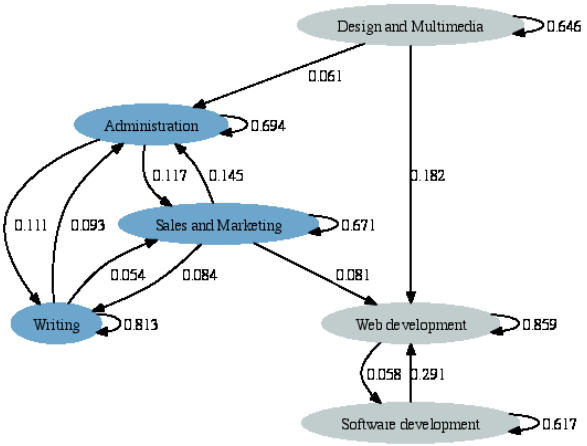


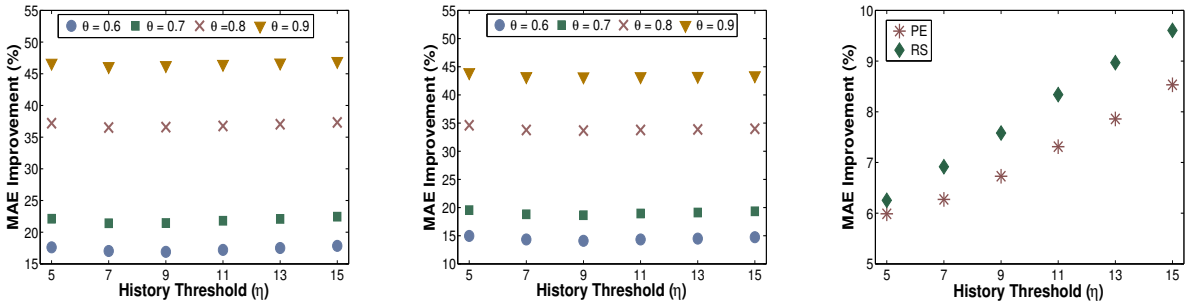
Figure 7: Transition probabilities across different categories in our training dataset. The graph shows only transitions with probability greater than 0.05.

that users, with relatively high probability, choose to complete tasks in the same category (e.g., they choose to remain in web development with probability 0.859). This is intuitively expected, since it shows a reasonable preference of the workers to keep choosing to work on tasks that are familiar with and build on their expertise. A natural, but incorrect, conjecture would be that inter-category performance would not matter much in such a setting, where users stay often within the same category. We will demonstrate, however, that properly leveraging past performance data from other categories can improve significantly the prediction of their future performance.

An important characteristic of the graph is the natural clustering of the nodes in the transition matrix: on the right, we have the "technical" categories, while on the left part of the graph we have a cluster of the "non-technical" categories, that can be characterized as the "Knowledge Process Outsourcing (KPO)" categories. This indicates that our categories are heterogeneous, and as a result, our transitional data points across categories are very sparse. To deal with the sparseness, we use the methodology described in Section 3.4. In particular, we choose $L = 2$, meaning that our top-level model deals with only two abstract categories: $\mathcal{H}_0 = \{$"Technical", "Non-Technical"$\}$. We name this model "**Generic**". Then we decompose each one of the two abstract categories into the following, more fine grained category sets:

- "**Technical**", $\mathcal{H}_{c_1} = \{$"Software developer", "Web developer", "Design & Multimedia"$\}$, and
- "**Non-technical**", $\mathcal{H}_{c_2} = \{$ "Writing", "Administration", "Sales & Marketing" $\}$.

To estimate the quality for worker i, we proceed as follows: We first estimate the necessary coefficients α_{jk} and β_j for three different models: the higher-level model, with $m = 2$, and the categories in $\mathcal{H}_0$, the lower-level technical model, with $m = 3$ and the categories in $\mathcal{H}_{c_1}$, and the lower level non-technical model, with $m = 3$ and the categories in $\mathcal{H}_{c_2}$. Then, we characterize the type of worker i as "Technical" or "Non-technical"



(a) Binomial Model,(PE) (b) Binomial Model,(RS) (c) Multinomial Model

Figure 6: The *improvement* of our models compared to the existing baselines, as measured by "mean absolute error" (MAE) for the (a) Binomial model point estimate (PE), (b) random sampling (RS), and (c) for the Multinomial model approaches.

based on the majority of her past tasks (i.e., $c_l^* \in$ {"Technical", "Non-technical"}). Given the task's category at time $t+1$ and the type of worker, we predict the quality of the task using the most appropriate model as determined by the process in Figure 3. Table 1 shows the coefficients for each model along with the categories that they correlate.

6.3 Experimental Procedure

We conducted 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 3.3. We then use regression with panel data [10] to estimate the coefficients a_{jk} and β_j for the model. We repeat the process for all possible combinations of thresholds θ and η.

Tuning Parameters: By considering the skewness of our training data distribution in our experiments (see also Section 6.1), we use various discrete threshold values for θ. In particular we assume that $\theta \in \{0.6, 0.7, 0.8, 0.9\}$. For each of these thresholds, the prior class probabilities ("bad" - "good") are respectively (11.5% vs. 88.5%), (13.6% vs. 86.4%), (19% vs. 81%), and (24.3% vs. 76.7%). Intuitively, low threshold values, result in higher skews. For the multinomial approach, we choose $K = 5$ and uniformly split the $[0, 1]$ interval into 5 buckets. Tasks 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. Intuitively, K can also be seen as the discrete star rating, 1 to 5. Finally, we evaluate each one of our models for discrete values of $\eta \in \{5, 7, 9, 11, 13, 15\}$.

Prior Distributions: Our models suggest that we have to pick 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). Note here that we further experimented with many other priors, including the uniform prior $Beta(1, 1)$. The results were not significantly different across our evaluations, so we kept the $Beta(9, 1)$ as prior, which reflects our prior knowledge about the marketplace. Similarly, for our multinomial approach, we choose a parameter vector $\boldsymbol{\alpha} = (2, 1, 1, 4, 8)$. Again this selection aims to capture the marketplace's biases, first toward high scores and second toward scores at the extremes of the distribution.

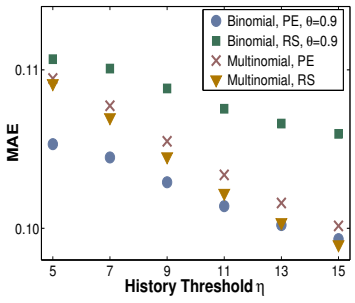


Figure 8: MAE values for each model and each proposed approach, for different values of η.

6.4 Experimental Results

6.4.1 *Performance Analysis*

In Figure 6(a) we show the percentage improvement for our binomial model, using the point estimate (PE) approach, for different threshold θ values. As one might expect, higher θ thresholds lead to better results for our model, as the data set becomes more balanced and training becomes more effective with more data in the minority class ("bad"). Specifically, for $\theta = 0.9$, the case with the more evenly balanced classes and therefore the highest underlying uncertainty, we achieve up to 47% improved performance. In the case with very skewed data (i.e., most cases are "good"), our model results in a MAE improvement of 16%. The history threshold η appears to have a very small effect in the MAE improvement, indicating that our algorithm works well even with very small worker histories.

In Figure 6(b), we show the percentage improvement for the Random Sampling (RS) approach, for the binomial model and for different values of θ. The results are very similar to those of the point estimate approach. Our maximum improvement, compared to the previous approach, is now a bit lower, 43.5%. we achieve this maximum with higher θ threshold. Similar to the point estimate approach, the worst case scenario of our model provides an MAE improvement of 13%. Once again,

Model	Level-Approach		α_{11}	α_{12}	α_{13}	β_1	α_{21}	α_{22}	α_{23}	β_2	α_{31}	α_{32}	α_{33}	β_3
Binomial	Generic ($\mathcal{H}_0$)	PE	-0.902	1.437	-	1.110	-1.165	1.088	-	1.691	-	-	-	-
			(0.200)	(0.188)	-	(0.101)	(0.342)	(0.152)	-	(0.292)	-	-	-	-
		RS	0.433	0.512	-	0.467	0.441	0.468	-	0.492	-	-	-	-
			(0.019)	(0.064)	-	(0.065)	(0.009)	(0.098)	-	(0.039)	-	-	-	-
	Technical ($\mathcal{H}_{c_1}$)	PE	-0.661	1.149	0.782	0.358	-0.362	0.877	0.725	0.433	-0.265	0.650	0.669	0.609
			(0.215)	(0.187)	(0.076)	(0.051)	(0.113)	(0.098)	(0.048)	(0.115)	(0.165)	(0.155)	(0.027)	(0.112)
		RS	0.341	0.412	0.343	0.316	0.353	0.378	0.354	0.323	0.362	0.373	0.327	0.335
			(0.028)	(0.029)	(0.048)	(0.046)	(0.031)	(0.029)	(0.028)	(0.065)	(0.022)	(0.045)	(0.020)	(0.043)
	Non-Technical ($\mathcal{H}_{c_2}$)	PE	-0.439	0.947	0.556	0.537	-0.446	0.642	0.835	0.610	-0.398	0.545	0.588	0.853
			(0.193)	(0.090)	(0.130)	(0.092)	(0.171)	(0.048)	(0.124)	(0.109)	(0.165)	(0.102)	(0.074)	(0.098)
		RS	0.349	0.412	0.329	0.306	0.340	0.351	0.396	0.348	0.320	0.337	0.338	0.409
			(0.035)	(0.037)	(0.068)	(0.062)	(0.022)	(0.048)	(0.048)	(0.059)	(0.030)	(0.066)	(0.052)	(0.042)
Multinomial	Generic ($\mathcal{H}_0$)	PE	-0.477	1.372	-	1.330	-0.111	1.058	-	0.933	-	-	-	-
		RS	0.433	0.325	-	0.286	0.373	0.278	-	0.371	-	-	-	-
	Technical ($\mathcal{H}_{c_1}$)	PE	0.844	-0.021	0.976	0.312	0.297	0.541	0.575	0.829	0.137	0.308	0.938	0.941
		RS	0.525	0.185	0.168	0.144	0.543	0.094	0.265	0.174	0.475	-0.004	0.334	0.307
	Non-Technical ($\mathcal{H}_{c_2}$)	PE	0.339	0.725	0.314	0.692	0.364	0.781	0.567	0.433	0.574	0.497	0.416	0.521
		RS	0.423	0.299	0.115	0.184	0.385	0.232	0.280	0.169	0.402	0.172	0.192	0.256

Table 2: The regression coefficients (resulting from the train set), broken down by model/level/approach. Numbers in parenthesis represent the standard deviations of the means across all θ values.

the history threshold η does not appear to significantly affect the MAE improvement.

Next, in Figure 6(c), we present the results for our multinomial model. Compared to the binomial model, the improvement is lower, but this might be related to the fact that the baseline for this model is more fine-grained than that used in the binomial case. The maximum achieved improvement is 9.6% for the random sampling approach (RS) and for $\eta = 15$, while the minimum MAE improvement is 6%, for $\eta = 5$ and for the point estimate approach (PE). We make two important observations here: first, compared to before, the *RS* approach provides slightly better results than the *PE* approach, and second, η threshold seems to have an increased effect on the achieved accuracies. Intuitively, the latter means that the multinomial model learns better from the outcome of each task than the binomial, and hence, it improves its predictions as history increases.

Finally, in order to give a presentation of integrated results, in Figure 8 we compare the actual MAE values for each of our models, for both approaches PE and RS. Note that for the binomial approach, we choose threshold $\theta = 0.9$ since that is the value with which our models achieve their highest accuracy. The first thing to observe here is that there is a significant difference between the PE and RS approaches of our binomial Model. On the other hand, for the multinomial model, the two approaches give very similar accuracies. Next, we can see that our multinomial approach benefits more from longer worker histories compared to the binomial as performance increases more as the threshold η increases.

6.4.2 Coefficient Analysis

To evaluate our coefficient estimates, we use both a quantitative and a qualitative approach. In particular, for the binomial model, we estimate the standard deviation of the means across all possible θ values for both the point estimate and random sampling approaches. Second, we interpret the meaning of these coefficients with respect to the oDesk environment.

In Table 2 we present all of the coefficients, broken down by model, level, and approach. For all the coefficients of the Binary model we show the mean and its standard deviation (in parentheses) across all the possible values of θ; we notice that the variances are very small among the coefficients of our binomial model. These low variances across different regressions also imply that these coefficients accurately capture correlations across different task categories, meaning they can be used to better interpret inter-category relations in the marketplace.

Finally, we notice that the β coefficients tend to be higher on non-technical task categories. This has an interesting explanation, and also serves as a reason for including the global quality as a factor in the regression The key reason for this is the relative easiness with which a worker can move across non-technical categories: As a rule of thumb, there is no specific training or difficult-to-acquire skill to perform many of the tasks in these categories.[7] For example, a worker who performs an "Administrative" task is likely to have an easier time switching to a "Sales & Management" task than a worker switching from "Software Development" to "Sales & Marketing". As a result, we can argue that the overall quality across the non-technical categories is more informative (higher coefficient) compared to the overall quality across all technical categories.

7. DISCUSSION AND FUTURE WORK

In this work we built different Bayesian hierarchical models that predict the quality of a new task performed by an on-line worker, by assigning different weights to the worker's observed category-specific qualities. We evaluated our methods by using hundreds of thousands of transactions from oDesk, an on-line labor market, and we demonstrated that our methods provide more accurate results than existing baselines. Based on our resulting coefficients, we were further able to note specific relations across different categories of the oDesk marketplace. In the future, we plan to expand the process to even more fine-grained categories, or even to deal with transitions across skills (e.g., "jquery" and "node.js"). In such a setting, with potentially thousands of micro-categories, we intend to study the building of hierarchical models in more detail. For example, we could create hierarchies by clustering categories based on their respective skillsets (e.g., the skillset {java,sql} could be one category, the skillset {php,javascript,ruby} could be a different category etc.).

In our current work we did not account for the effect of time. In our problem definition, we assign the same weight to a prior task, as long as it comes from the same category, independent of the time that has passed since then. This might not be quite accurate, since, in the majority of the tasks, a worker

[7] Translation is a notable exception in this rule of thumb.

acquires more expertise and improves his skills by completing more and more additional tasks. In the future, we intend to extend our models by including timestamps as a component of each completed task.

A key thing to mention is that our current model is predictive and not necessarily causal. A basic characteristic of predictive models such as the one we propose here 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 the workers applied to the jobs they wanted to get. So, such a tool should *not* be used to recommend jobs to workers as this will modify the self-selection process, and by extension the underlying data generation process, potentially rendering our model invalid. Our algorithm 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 much the self-selection process of applying for jobs.

A desirable feature of our proposed algorithms is their broadness: they can be easily adapted for various kinds of marketplaces, which are interested in estimating the quality of their users. For example, as mentioned in Section 6, we have already experimented with predicting the usefulness of product reviews on Amazon.com, by analyzing the usefulness of prior reviews posted by a user, and examining their category distribution. Our results, not included in the current paper for space reasons, are similar in nature with the oDesk results and illustrate the benefits of this approach. We believe that most reputation systems can benefit from using approaches similar to ours, instead of relying on simple average values.

8. REFERENCES

[1] L. Adamic, J. Zhang, E. Bakshy, and M. Ackerman. Knowledge sharing and yahoo answers: everyone knows something. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008.

[2] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. In *Proceedings of the international conference on Web search and web data mining*. ACM, 2008.

[3] J. Bian, Y. Liu, D. Zhou, E. Agichtein, and H. Zha. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceedings of the 18th international conference on World Wide Web*. ACM, 2009.

[4] R. Clemen and R. Winkler. Unanimity and compromise among probability forecasters. *Management Science*, 36(7), 1990.

[5] C. Danescu-Niculescu-Mizil, G. Kossinets, J. Kleinberg, and L. Lee. How opinions are received by online communities: a case study on amazon. com helpfulness votes. In *Proceedings of the 18th international conference on World Wide Web*. ACM, 2009.

[6] C. Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management science*, 49(10), 2003.

[7] C. Dellarocas. Reputation mechanisms. *Handbook on Economics and Information Systems*, 2006.

[8] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 2004.

[9] A. Ghose and P. G. Ipeirotis. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *TKDE*, 23(10), 2011.

[10] W. Greene. *Econometric analysis*. Prentice Hall, 2007.

[11] R. Hambleton. *Fundamentals of item response theory*, volume 2. Sage Publications, Incorporated, 1991.

[12] N. Hu, J. Zhang, and P. A. Pavlou. Overcoming the j-shaped distribution of product reviews. *Commun. ACM*, 52(10), 2009.

[13] J. Jeon, W. Croft, J. Lee, and S. Park. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th annual international conference on Research and development in information retrieval*. ACM, 2006.

[14] S. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti. Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 2006.

[15] T. Lappas and D. Gunopulos. Efficient confident search in large review corpora. *Machine Learning and Knowledge Discovery in Databases*, 2010.

[16] Y. Liu, J. Bian, and E. Agichtein. Predicting information seeker satisfaction in community question answering. In *Proceedings of the 31st annual international conference on Research and development in information retrieval*. ACM, 2008.

[17] Y. Liu, X. Huang, A. An, and X. Yu. Modeling and predicting the helpfulness of online reviews. In *Eighth IEEE International Conference on Data Mining, 2008. ICDM'08.* IEEE, 2008.

[18] Y. Lu, P. Tsaparas, A. Ntoulas, and L. Polanyi. Exploiting social context for review quality prediction. In *Proceedings of the 19th international conference on World Wide Web*. ACM, 2010.

[19] P. Nelson. Information and consumer behavior. *The Journal of Political Economy*, 78(2), 1970.

[20] M. O'Mahony and B. Smyth. Using readability tests to predict helpful product reviews. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, 2010.

[21] J. Otterbacher. 'helpfulness' in online communities: a measure of message quality. In *Proceedings of the 27th international conference on Human factors in computing systems*. ACM, 2009.

[22] C. Shah and J. Pomerantz. Evaluating and predicting answer quality in community qa. In *Proceeding of the 33rd international conference on Research and development in information retrieval*. Citeseer, 2010.

[23] M. Suryanto, E. Lim, A. Sun, and R. Chiang. Quality-aware collaborative question answering: methods and evaluation. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*. ACM, 2009.

[24] P. Tsaparas, A. Ntoulas, and E. Terzi. Selecting a comprehensive set of reviews. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011.