Hiring Behavior Models for Online Labor Markets

Marios Kokkodis
NYU Stern
mkokkodi@stern.nyu.edu

Panagiotis Papadimitriou
Elance-oDesk
papadimitriou@elance-odesk.com

Panagiotis G. Ipeirotis
NYU Stern
panos@stern.nyu.edu

ABSTRACT

In an online labor marketplace employers post jobs, receive freelancer applications and make hiring decisions. These hiring decisions are based on the freelancer's observed (*e.g.* education) and latent (*e.g.* ability) characteristics. Because of the heterogeneity that appears in the observed characteristics, and the existence of latent ones, identifying and hiring the best possible applicant is a very challenging task. In this work we study and model the employer's hiring behavior. We assume that employers are utility maximizers and make rational decisions by hiring the best possible applicant at hand. Based on this premise, we propose a series of probabilistic models that estimate the hiring probability of each applicant. We train and test our models on more than 600,000 job applications obtained by oDesk.com, and we show evidence that the proposed models outperform currently in-use baselines. To get further insights, we conduct an econometric analysis and observe that the attributes that are strongly correlated with the hiring probability are whether or not the freelancer and the employer have previously worked together, the available information on the freelancer's profile, the countries of the employer and the freelancer and the skillset of the freelancer. Finally, we find that the faster a freelancer applies to an opening, the higher is the probability to get the job.

1. INTRODUCTION

Online labor marketplaces (OLMs) such as oDesk.com, and Freelancer.com allow employers to connect with freelancers around the globe to accomplish diverse tasks including web development, writing and translation, accounting, etc. These marketplaces are growing fast and the freelancer annual earnings are expected to grow from \$1 billion in 2012 to \$10 billion by 2020 [2].

Although such platforms have provided the employers with a solution to the scarcity of local talent, they have not really changed the process that employers have to go through to source the ideal candidates for their tasks: an employer needs initially to describe the job opening requirements to which freelancers that are looking for opportunities can apply. Then, the employer has to (1) review all applicants by looking at their online profile information and/or by personally interviewing them, and (2) come up with a hiring decision.

To evaluate an applicant, the employer has to assess both observed and latent characteristics. The observed characteristics usually include a list of skills, the educational background, the work history and the certifications or tests that the applicant has taken. The latent characteristics include the freelancer's quality and true ability on his listed qualifications. The existence of latent characteristics, the heterogeneity that appears in the observed ones as well as the interactions between the two make the matching process a very challenging task; Hiring decisions are based on manually shaped expectations of complicated similarities between job openings, employers and freelancers. These expectations usually come with high uncertainties, since performing a task is an "experience good" [37]: for both the freelancer and the employer, it is practically infeasible to know the outcome of their collaboration in advance.

To minimize the level of uncertainty, most of the online labor marketplaces have developed reputation systems. Freelancers get rated for the tasks they accomplish and these ratings become part of their online resumes. Employers can then get a better picture of the freelancers' past performance and make better informed hiring decisions. However, in online labor markets, as well as in most of the online markets in general, reputation scores are very skewed towards high ratings (J-shape distributions, see Hu *et al.* [22]), and as a result they become almost uninformative. Since reputation systems fail to provide insightful information about the freelancers' quality, how do employers make hiring decisions? What are the characteristics that value the most?

In this work we focus on answering these questions. We propose a series of increasing complexity predictive models that describe employers' hiring decisions. We start our analysis by assuming that employers are rational utility maximizers; Their utility is straightforwardly maximized along with the probability of selecting the best possible applicant for each specific opening. Based on this assumption, we first propose a ranking aggregator that ranks candidates in all the available dimensions and then aggregates these ranks to create a global ranking. Next, we draw on empirical economics and propose a Logit model [46] and finally, we built a probabilistic graphical model (Bayesian network). We compare our models with the vanilla reputation score baseline, where each employer ranks the available applicants based on their previously collected feedback score.

We train and test all proposed approaches on a unique dataset of *real* transactional oDesk data. In particular, this

WSDM'15, February 2–6, 2015, Shanghai, China.

http://dx.doi.org/10.1145/2684822.2685299 .

dataset consists of *more than 600,000 job applications* on *more than 50,000 openings* related to four different task categories. We use different evaluation metrics and show evidence that our models significantly outperform the vanilla reputation baseline. Next, we perform an econometric analysis and observe that the attributes that are strongly correlated with the hiring probability are whether or not the freelancer and the employer have previously worked together, the available information on the freelancer's profile, the countries of the employer and the freelancer and the skillset of the contractor. Finally, our analysis shows that the faster the freelancer applies to an opening, the higher is the probability to get hired.

Our work is the first to study the association of different characteristics with hiring decisions by incorporating a massive amount of observational data. We believe that understanding how hiring decisions are made is beneficial for both the employers and the contractors, and *critical* for the marketplace. In particular, by developing approaches that estimate the applicants' hiring probabilities: (1) employers will be able to make better-informed and faster decisions based on the suggested applicants' rankings, (2) freelancers will save time by not applying to openings that have very low hiring probability and (3) the marketplace might identify weaknesses in freelancers' profiles (*e.g.* skills not reported, the profile description is not sufficient *etc.*) and suggest targeted profile improvements based on how each profile characteristic affect hiring decisions. As a result, our proposed approaches will minimize the friction in the marketplace[1] and increase both the marketplace's transaction volume as well as the overall satisfaction of the freelancers and the employers.

2. BACKGROUND

In this section, we provide background information regarding online labor markets as well as research that dealt with hiring decisions in offline workplaces. We further briefly discuss the key differences between our scenario, and the "learning to rank" framework which is usually found in information retrieval (IR) problems. Finally, we position our study and argue about its uniqueness.

2.1 Online labor markets

Current research in Online Labor Markets (*OLM*s) spans across a variety of problems. A stream of work focuses on the validity of behavioral experiments in these markets and in particular on Amazon Mechanical Turk [3, 20, 42]. The general consensus of these studies is that online experiments appear to be as valid (both internally and externally) as laboratory field experiments. A different group of studies focuses on incentivizing freelancers as well as finding ways to manage the quality of their outcomes [10, 19, 24, 25, 36, 44, 45]. These studies propose and evaluate a set of social and financial incentives and they further provide sophisticated techniques that assure a certain level of outcome quality. Finally, Kokkodis and Ipeirotis focused on how past reputation transfers to new tasks [29], and they further quantified the value of skills in *OLM*s [30].

2.2 Hiring Decisions

Previous studies dealt with several dimensions of the matching problem in offline markets. Lindeboom et. al [33] studied the effectiveness of search and recruitment channels. Audra J. Bowlus [6] focused on the extent to which the level of job mismatching varies over the business cycle and how it is dealt with by the labor market. Finally Yael S. Hadass [18] studied the impact of the spread of online recruiting on the matching of freelancers and firms.

On a different direction, a lot of work has focused on gender and attractevness biases in hiring decisions [11, 12, 16, 16, 28, 34, 38]. The overall consensus of all of these studies is that gender and attractiveness have a strong effect on hiring decisions, but the type of the effect depends on the environment, the position, and the employer.

A lot of work has focused on other characteristics that affect hiring decisions. In particular, Forsythe *et al.* [14] found that female applicants' clothing is an avenue for influencing the selection decision for management positions. Rasa and Carpenter [43] found that the influence of demographics is modest and less important than other variables, such as the applicant's skills and qualifications. Hu [21] found that large firms hire younger applicants than small firms Yakubovich [52] found that a freelancer is more likely to get a job through a weak tie rather than a strong one. Recently Acquisti and Fong [1] found (1) that only a minority of US firms likely search online for candidates' information and (2) evidence of discrimination linked to political party affiliation. Finally, A. Pallais [39] studied the cold-start problem (i.e. hiring inexperienced freelancers) in *OLM*s and showed that both hiring freelancers and providing more detailed evaluations substantially improves freelancers' subsequent employment outcomes.

2.3 Learning to rank

Our problem has conceptual similarities with the "learning to rank" problem (*i.e.* the construction of ranking models), commonly found in information retrieval [47]. State of the art ranking approaches mainly use either a pairwise approach, which reduces ranking to classification on document pairs w.r.t. the same query (see [8, 15, 27, 41, 54]) or a listwise approach, which performs learning directly on document list by optimizing either some information retrieval evaluation measures or by minimizing listwise loss functions (see [9, 49–51, 53]).

We argue that the hiring problem we study has a strong peculiarity that doesn't allow for as-is adaptation of learning-to-rank algorithms: We lack multiple ranks: in our scenario, there is no ground truth in terms of which applicant is better than other. We only observe whether or not an applicant got hired; the rest of the applicants are then getting assigned with the same rank.

The hiring decision problem we study is very close to the "product search problem" [32]. One main difference is that job openings usually have completely independent groups of applicants with each other, while the pool of a certain product alternatives is usually the same. In their work, Li *et al.* propose a Logit model for homogenous consumers (an approach we adapt in this work as well), and a BLP model [4] for heterogeneous consumers. To train the BLP model, they use product-specific data from multiple markets, which is practically impossible to do in our case (*i.e.* acquire employers/contractors data from multiple *OLM*s).

Our work is the first one to study how a series of characteristics of both the employer and the freelancer collectively affect hiring decisions. The emergence of *OLM*s allow us for the first time to perform such a study in a large scale (hundreds of thousands of applications, tens of thousands of hiring decisions, multiple task categories). The premise of *OLM*s (remote freelancers, zero face-to-face interaction) sug-

---

[1] See also Brynjolfsson and Smith [7]

gests that gender/attractiveness biases should have a small (if any) effect in hiring decisions. Hence in this study we focus on features such as the freelancer's skills, reputation, education, certifications, demographics, pay rate, *etc.* We discuss these characteristics next.

3. FEATURES IN HIRING DECISIONS

In this section we describe the set of characteristics that (we believe) affect hiring decisions in *OLM*s. This set includes straightforward features (*e.g.* accumulated reputation score) as well as derived features (*e.g.* a combination of the employer's and the freelancer's countries).

3.1 Freelancer Characteristics

The first cluster of straightforward attributes we believe have an effect on employers' hiring decisions appear on the freelancer's profile; In particular we consider the average feedback score of the freelancer ($feedback$), the time of the application (*i.e.* whether the applicant applies first, second *etc.*) ($time$), the total hours that the freelancer has worked on the oDesk platform ($hours$), the number of jobs that the freelancer has completed ($jobs$), the hourly billing (mentioned on the freelancer's profile ($bill$)) and bidding (bid) pay rate of the freelancer, the number of tests that the freelancer has completed in the past ($tests$), the freelancer's years of experience (self-reported) (exp), and the freelancer's education (edu). We further consider a binary variable that controls whether the freelancer just joined the marketplace (new).

As we discussed in the introduction, hiring decisions come with high uncertainty: it is practically impossible for the employer to know beforehand the performance of a freelancer. Reputation systems usually minimize this uncertainty since they provide some information about the past performance of freelancers. As a result, it is rational to assume that employers, who have limited practical knowledge of the skills and abilities of a remote freelancer, often consult these reputation systems to better understand whether a freelancer is qualified and suitable for the task at hand.

Two other signals that contributes towards minimizing this uncertainty (and as a result we expect to have an effect on hiring decisions) is the freelancer's number of completed jobs and hours worked in the marketplace. In online marketplaces, users that receive low feedback scores usually abandon, and they either create new accounts and re-join or find an alternative marketplace to use (see Jerath *et al.* [26]). As a result, users that accumulate long history on the marketplace are ceteris paribus more trustful than users with no hirstory.

The hourly billing and pay rates of the freelancer also signal the unobserved quality of the freelancer. Intuitively, on average, a freelancer that gets paid more to complete a certain task is better than a freelancer with a lower pay rate for the same task. Similar arguments also apply for the freelancer's education[2], the number of tests that the freelancer has taken[3] as well as the freelancer's self-reported years of experience on similar tasks.

Finally, since new freelancers always join the marketplace, and because we don't want to "unfairly" penalize them for their zero values in "feedback","hours" and "jobs", we include a binary variable to control for the fact that these contractors are "new" to the marketplace.

Rank	**Employer's Country**	**Freelancer's country**	**PMI**
TOP	Nicaragua	Nicaragua	4.95
	Mexico	Sierra Leone	4.79
	Moldova	Argentina	4.66
	South Korea	South Korea	4.66
	Puerto Rico	Panama	4.14
BOTTOM	United Kingdom	India	-1.74
	Australia	India	-1.74
	United States	Bangladesh	-1.82
	United States	Pakistan	-1.85
	United States	India	-2.13

Table 1: Combination of countries that have high PMI.

3.2 Freelancer - Employer characteristics

So far we analyzed characteristics that are straightforwardly extracted from the online profile of the freelancer. Next, we describe how combined information from the freelancer's and the employer's profile might affect hiring decisions.

First, we believe that the combination of freelancer's and employer's locations has an effect on the hiring probability; For example, an employer from the United States might have a strong positive prior for freelancers from the United States, and as a result a strong hiring preference for such freelancers. To quantify and study the effect of location on hiring decisions we use the Point-wise Mutual Information (PMI) of the two countries (see Bouma [5]). PMI is a measure of how much the actual probability of a particular co-occurance of two events (in our case the freelancer's country c_w and the employer's country c_e) differs from what we would expect to see if the two events were independent. Formally:

$$PMI(c_w, c_e) = \log \frac{\Pr(c_w, c_e)}{\Pr(c_w)\Pr(c_e)} \tag{1}$$

PMI takes positive and negative values, it is zero if the two events are independent, and it maximizes when the two events are perfectly associated. To estimate the necessary probabilities we use our training sets (see section 5.1). Some of the combinations that have very high and very low PMI values are shown in Table 1. It's interesting to notice the separation between rare events (for example employers from Nicaragua) and very frequent ones (for example employers from U.S.A. and freelancers from India). In line with intuition, co-location of the freelancer and the employer also appears to have a high PMI score (for instance Korea - Korea and Nicaragua - Nicaragua).

Next, we believe that whether or not the freelancer has previously worked with the same employer (potential re-hire) must have an effect on hiring decisions. Intuitively on average, a freelancer who wants to work again with the same employer must have had an overall successful experience with that employer.

3.3 Freelancer - Opening characteristics

The last set of features we consider originate in the interaction of the freelancer and the opening at hand. Based on the belief that the set of applicant's skills in combination with the set of required skills of the specific job opening must have a strong effect on hiring decisions, we compute the inner product of the two. Intuitively, the higher the in-

---

[2] We consider five levels of education: High school, Diploma, Bachelor, Masters and PhD

[3] *OLM*s make available a list of certification tests to their users. For example, oDesk.com provides the following set of tests `https://www.odesk.com/tests`

	Applicant	jobs	feedback
Opening 1	Anna	3	5
	Michael	4	5
Opening 2	George	1	3
	Mary	1	3

Table 2: Opening-dependent hiring decisions.

ner product the higher is the compatibility of the applicant for the task at hand. Similarly, we further estimate the inner product between the required skills of the opening and the certifications that a freelancer has (exams). For example, let's assume that a freelancer has only one certification in Java, and that the opening requires Java and SQL. Then the inner product of this freelancer and the opening will be 1. The intuition is the same as before.

Finally, we include the percentage of shared information between the opening description and the freelancer's profile. By using a bag of words approach [35] we compute the cosine similarity between an opening description and an applicant's profile. In particular, using bigrams, we create a binary vector representation of the freelancer profile ($\mathbf{w}_w$) and the opening ($\mathbf{w}_o$). We then compute the cosine similarity (m) of the two vectors:

$$m(\mathbf{w}_w, \mathbf{w}_o) = \frac{\mathbf{w}_w \cdot \mathbf{w}_o}{||\mathbf{w}_w||||\mathbf{w}_o||} \ ,$$

where $||\cdot||$ is the $L2$ norm.

3.4 Average pairwise transformation

The introduction of these features in probabilistic models is not straightforward; Each hiring decision is opening-specific; the instances of each opening are strongly dependent and directly comparable within the opening, but not across openings. As a result, we end up having a set of non-comparable instances to build our probabilistic models on. Consider for example two different openings in the same category, and suppose that each one of these two openings has the two applicants shown in Table 2. The set of applicants of the first opening is superior to the set of applicants of the second opening: the feedback scores of all applicants in opening 1 strictly dominate the feedback scores of candidates in opening 2. The same applies to jobs. The employer of each opening however, will choose to hire one of the available applicants[4]; As a result, the hiring decisions for the two openings in Table 2, will be based on different evaluation criteria that use pairwise comparisons of the available applicants.

Since we learn a global model on hiring decisions, we overcome this inconsistency that naturally appears in our data in the following way: for every feature, we first rank all the instances within an opening in descending order. Then, we compute the average pairwise difference of each instance, with all the other instances in an opening. Specifically, for each instance i and for each feature $x_{i,k} \in \boldsymbol{X}$ we have:

$$x_{i,k}^{transformed} = \frac{1}{|o|-1} \sum_{j \neq i} x_{i,k} - x_{j,k} \ , \qquad (2)$$

	$x_{i,jobs}$	$x_{i,jobs}^{transformed}$
Anna	3	$((3-1)+(3-0))/2 = 2.5$
Mary	1	$((1-3)+(1-0))/2 = -0.5$
John	0	$((0-3)+(0-1))/2 = -2$

Table 3: Example of the average pairwise transformation for feature *jobs*. We assume that these are the only three applicants in the opening at hand, hence $|o| = 3$.

where $|o|$ is the number of applications in opening o ranked in descending order of values x_k. To clarify this process, consider an example with three applicants, Mary, Anna and John. In Table 3 we show the number of completed jobs for these three applicants in decreasing order. Their transformed values are shown in the third column. Note that the proposed transformation is a variation of the pairwise transformation, which is commonly used in ranking approaches (*e.g.* [27][5]). We apply this transformation to all our features but "time". The reason is that by definition "time" compares applicants within an opening, while it takes the same values across openings.

4. PROBLEM FORMULATION

Before we describe our approaches, it is important to clearly formulate the problem we study:

Problem Definition: *Given an opening, an employer and a set of applicants with specific characteristics, we are interested in* ***estimating*** *the* ***hiring probability*** *of these applicants.*

In this section we present models that explicitly address this problem. In particular, we first describe a simple ranking model that is based on applicant's reputation; we later use this model as a baseline. Next we present three different approaches of increasing complexity: (1) a ranker aggregator, (2) a Logit Model and (3) a Bayesian Network. For the rest of this work, we assume that all dimensions discussed in section 3 form a feature vector $\boldsymbol{X}$.

4.1 Feedback Score Baseline

Based on the intuition that freelancer's reputation should have a strong effect on hiring decisions, we propose a baseline that ranks applicants based on their feedback score. Specifically in the oDesk platform, after the completion of a task, the employer supplies feedback scores (integers between 0 and 5) to the freelancer in the following six fields: "Availability" ($f1$), "Communication" ($f2$), "Cooperation" ($f3$), "Deadlines" ($f4$), "Quality" ($f5$), "Skills" ($f6$). This vanilla model ranks all available applicants based on their average feedback scores, and assumes that the top applicant for each opening is the one that gets hired. Ties are resolved by the time of application (*i.e.* the fastest applicant gets the job). Because of the simplicity of this model and our belief that hiring decisions draw on a much more complex process, we use this model as a baseline, and we tackle our problem from three different perspectives, which we discuss next.

4.2 Ranker aggregator

Based on the premise that employers take into account multiple dimensions for ranking the available applicants, and

---

[4] Note that this is a peculiarity of our dataset, since we only consider openings that lead to a single hiring decision. In practice, the employer could have chosen not to hire anyone. We further discuss the limitations that our dataset imposes in Section 8.1.

[5] In all our experiments the proposed transformation outperform the pairwise transformation presented in [27].

Feedback	Past Jobs	Skills	Aggregator (median)
Mary	Anna	Anna	Anna (1)
Anna	Mary	Mary	Mary (2)
John	John	John	John (3)

Table 4: Example of Ranker Aggregator

not just their reputation, we propose a ranker aggregator. Our goal is to use information from all the available dimensions we discussed in section 3 and come up with a set of ranked applicants for each opening. To do so, we consider previous work on rank aggregation methods for web documents (see Dwork et al. [13]). We first create separate rankings in each available dimension of the feature vector $\boldsymbol{X}$. Then, for each application in an opening, we compute its median position across all rankings. We finally use these values (median) to rank the available applicants.

To clarify this process, assume that our feature vector consists of only three dimensions, "Feedback" , "Past Jobs" and "Skills", i.e., $\boldsymbol{X}$ = [Feedback, Past Jobs, Skills]. Now suppose that for a specific opening, we have three applicants, Mary, Anna, and John. For each of the three dimensions, we create a ranking of these applicants, shown in Table 4. The proposed aggregator function takes the median of all positions of each applicant across the three rankings, and creates an aggregated ranking, presented in the rightmost column of the table. As a result, Anna with median '1' is ranked first; Mary with median '2' is ranked second, *etc.* Note that the length of our feature vector is $|\boldsymbol{X}| = 15$, and as a result ties are very rare. If however ties occur, we resolve them chronologically, as in the feedback baseline.

4.3 Logit model

The ranker aggregator is an intuitive approach that uniformly assigns equal weights to each dimension of the feature vector $\boldsymbol{X}$. However, in practice, such a uniform assignment is an oversimplification of reality: employers have individual preferences, and as so, they value some features more than others. For example, the skills of an applicant and the number of previous completed jobs might have stronger effect on the employer's hiring decisions than the location of the applicant, or vice a versa. To study this anticipated weight variation across the different features, we draw on empirical economics and propose a Logit binary choice model [17].

In the logit model, the conditional probability of hiring an applicant i given the feature vector $\boldsymbol{X}_i$ is given by the following:

$$\Pr(Y_i = hire|\boldsymbol{X_i}) = \frac{\exp(\boldsymbol{\beta X_i'})}{1+\exp(\boldsymbol{\beta X_i'})} \ , \tag{3}$$

where $\boldsymbol{\beta}$ is the vector of coefficients, and $Y_i \in \{hire, nohire\}$. To estimate the vector of coefficients we use maximum likelihood. In particular, we assume that each instance (observation) in our dataset is i.i.d.[6], and we estimate the likelihood function as follows:

$$\Pr(Y_1 = y_1, ..., Y_n = y_n|\boldsymbol{X}) = \prod_{y_i=hire} \Lambda(\boldsymbol{\beta X'}) \prod_{y_i=nohire} (1-\Lambda(\boldsymbol{\beta X'})) \quad ,$$

where$\Lambda(\ .)$ is the logistic sigmoid [17].

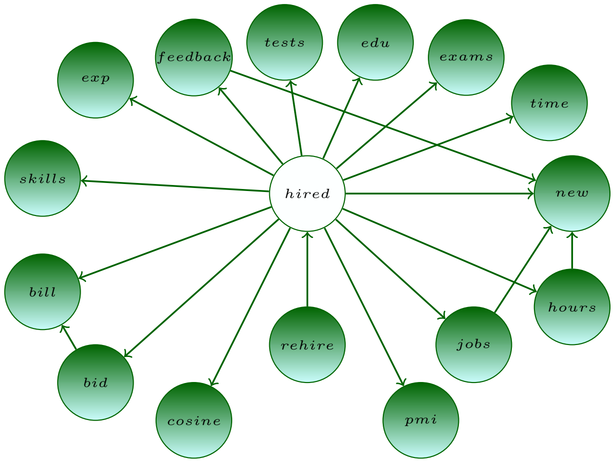

Figure 1: Structure of the proposed Bayesian network.

4.4 Bayesian Network Approach

The nature of our problem suggests that many of the features are correlated with each other. For example, the number of previously completed jobs is strongly correlated with whether or not the freelancer is new. Bayesian networks are able to capture these dependencies, and provide a probabilistic estimation of the target variable by taking into consideration the structure of the network. For our case, we propose the network presented in Figure 1. Intuitively we suggest that if an employer and a freelancer have worked together in the past, then the hiring decision is directly affected by their previous interaction (hired node has only one parent, the "rehire" node). Hiring decisions are correlated with all the other characteristics, in a father - child relationship. We further propose that the pay rate (bid) is correlated with the bill rate (bill) and that whether or not an applicant is new in the marketplace affects the applicant's reputation, hours worked and completed number of jobs[7].

The joint distribution of the proposed network is given by the following equation (see Koller and Friendman [31]):

$$\Pr(Y, \boldsymbol{X}) = \Pr(Y|rehire) \prod_{x_i \in \boldsymbol{X}} \Pr\left(x_i|Pa(x_i)\right) \ , \tag{4}$$

where $Pa(x_i)$ is the parent of the node x_i. Because we are interested in estimating the conditional probability of $\Pr(Y|\boldsymbol{X})$, we get the following:

$$\begin{aligned} \Pr(Y|\boldsymbol{X}) &= \frac{\Pr(Y, \boldsymbol{X})}{\Pr(\boldsymbol{X})} \\ &= \frac{Pr(Y|rehire) \prod_{x_i \in \boldsymbol{X}} \Pr\left(x_i|Pa(x_i)\right)}{\Pr(\boldsymbol{X})} \end{aligned}$$

Next, since all values of vector $\boldsymbol{X}$ are observed, we can simply estimate the conditional probabilities by counting the

---

[6] we discussed this assumption in section 3.4.

[7] Even though we propose this network by intuition, we further experimented with learning the structure of the network through the data by using different scoring functions and learning algorithms (see also [31]). In our evaluation, the intuitive model presented here performed significantly better than all the automatically learned structures.

Opening	Top-2 ranked applications	True Label	Outcome
1	1 2	*nohire* *hire*	Success
2	1 2	*nohire* *nohire*	Failure
3	1 2	*hire* *nohire*	Success

Table 5: Accuracy at top-n toy example.

joined appearances in our training sets:

$$\Pr\left(x_i|pa(x_i)\right) = \frac{N\left(x_i, pa(x_i)\right)}{N\left(pa(x_i)\right)} ,$$

where $N(e)$ is the number that evidence e appears in our training set.

5. EXPERIMENTAL SETUP

In this section we discuss the experimental setup we use to build and evaluate our approaches. We start by presenting characteristics of the oDesk transactional data we used in our analysis, and then we present and reason about the evaluation metrics we used.

5.1 oDesk Data

oDesk is a global job marketplace with a plethora of tools targeted to businesses that intend to hire and manage remote freelancers. In the past decade, the company experienced a consistent exponential growth in transaction volume. At oDesk, people from all over the world can post any type of task and hire freelancers to work and deliver a requested outcome.

In terms of contracts, there are two types of openings: Hourly and Fixed paid. In this study we present results only on Hourly openings[8]. We analyze a total of 630,000 job applications that lead to more than 50,000 hiring decisions. All this data was sampled between September 1st 2012 and December 31st 2013.

Our approaches use time sensitive features. For example, consider the number of tests taken by the freelancer and their scores. At different times, this feature is expected to take different values, since freelancers constantly try to improve their profiles. The same applies to skills, profile information *etc.* To overcome this naturally occurring transience we take snapshots of freelancers' profiles at the time of application.

Our data span across four different task categories: web development, software development, writing and translation and design and multimedia. The reason for choosing more than one task category is to study how the proposed approaches behave, given that employers might possibly follow different category-specific hiring decision processes. For instance, it is possible that the average employer who makes a hiring decision on a writing task opening, follows a completely different selection process than the average employer who hires freelancers on web programming tasks.

We split our data across these categories and we build different category-specific models. We further separate vertically (on openings) our data into training and test sets (67%-33%). To avoid overfitting, we build all our models on the training sets and evaluate them on the respective test sets. We focus on openings that have between 2 and 50 applicants, and that led to a *single hiring decision.* As a result, the distribution of our data is highly skewed towards "no-hires"; 94% of the applications we consider are "no-hires". This makes the evaluation of our models trickier, since using a metric such as the accuracy of our predictions would be naive: a classifier that would always predict "no-hire" would have an accuracy of 94%. We discuss next the established metrics we used to evaluate our models.

5.2 Evaluation Metrics

The first metric we use to evaluate the predictive power of our approaches is the Area Under the Curve[9] (**AUC**). Intuitively, the AUC can be seen as the probability that our model will rank a "positive" instance (e.g. a hire) higher than a "negative" one.

The next metric we use is the **lift** (see also [48]). We define the lift as follows:

$$lift = \frac{\Pr[+|top\ x\%\ ranked]}{\Pr(+)} , \tag{5}$$

where $\Pr[+|top\ x\%\ ranked]$ is the probability of randomly selecting a positive instance (a hire) in the top $x\%$ ranked instances of the test set, and $\Pr(+)$ is the probability of randomly choosing a positive instance across the test set. Intuitively the lift shows how many times better than random our models' rankings are.

These two metrics evaluate the global performance of our models, but they don't capture the per opening performance. To study the per opening performance we define the **accuracy at top-n** (ACC-n) as the probability that our model will rank a true positive (TP) instance in the top n instances of each job opening. For clarity, we present an example on Table 5, where we consider only three openings, and we examine only the top-2 ranked applications for each one of these openings. Our model predicts a true positive (TP) instance for openings 1 and 3. We consider these two openings as successful predictions at top-2. Then, by assuming that these three openings represent our test set, we estimate the ACC-2 by applying the following formula:

$$\text{ACC-n} = \frac{\#\ successes}{\#\ openings} \tag{6}$$

In our example we get $ACC - 2 = 2/3$.

6. EXPERIMENTAL RESULTS

In this section we analyze the performance of our approaches. Following the flow presented in section 5, we first discuss the resulting AUCs and Lifts of our models, and then we present their accuracies at top-n (ACC-n).

In Table 6 we show the area under the curve for all our models and the baseline, for the task categories we study. Recall than a randomized model will give AUC = 0.5 (see Foster and Fawcett [40]). First, we notice that the resulting rankings of the feedback baseline outperforms a random ranker, with AUC values between 0.535 and 0.548. One might argue that this slightly improvement of the feedback baseline over a random classifier is counterintuitive; freelancers' reputation should play a critical role in hiring decisions. There are two observations that explain this result. First, as we discussed earlier, feedback scores in online workplaces tend to form J-shaped distributions (see also Hu et

---

[8] Our approaches have been successfully tested on Fixed price openings as well.

[9] Detailed analysis about the use of AUC can be found in the work by Provost and Fawcett [40].

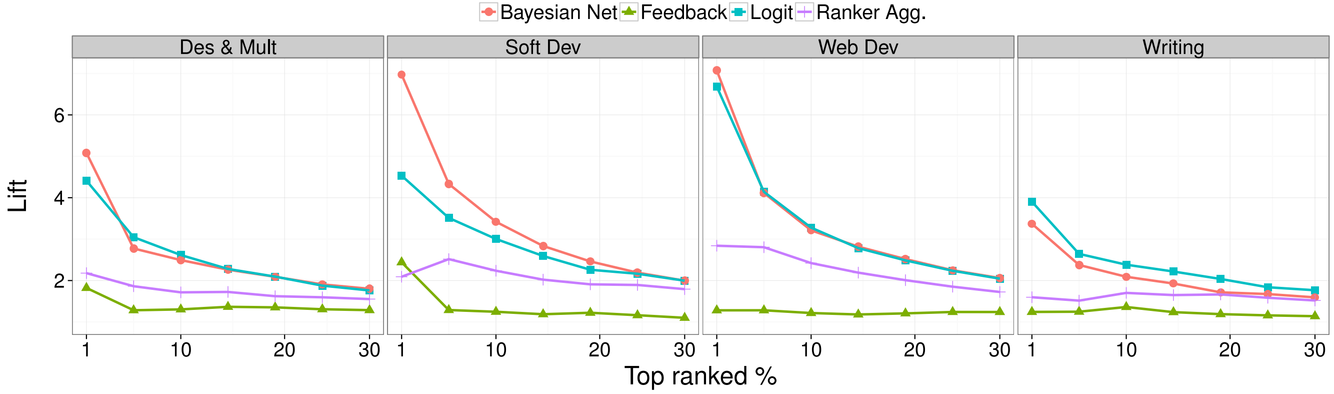

Figure 2: The lift values for all the proposed approaches and the baseline, for the four task categories we study.

al. [23]): most of the freelancers have excellent reputation (5 stars). In our training set the median is 5 and the average 4.7! Furthermore, these J-shaped feedback distributions 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) [26]. Another explanation why the feedback score does not perform well is the fact that amongst the applications we study, roughly 15% originate from freelancers without any previous history in the marketplace, hence no feedback score.

The *AUC* of the ranker aggregator and the two probabilistic models (Logit and Bayesian network) ranges between 0.628 and 0.74, significantly higher than the feedback baseline's values. Recall that intuitively, AUC is the probability of ranking a positive instance (i.e., a hire) higher than a negative one, so we can say that our models rank positive instances on top of negative instances with different probabilities up to 0.74. In the context of predicting hiring decisions, this probability is really high (recall that 94% of the applications are "nohires". If we compare the three proposed approaches, the Bayesian Network slightly outperforms the Logit Model, which in turn outperforms the ranker aggregator; as expected, increasing the algorithm complexity provided better results.

Next, in Figure 2 we present the *lifts* (see equation 5) of all the proposed approaches and the feedback baseline. The results are similar to the AUC scores. First, all three models perform significantly better than the feedback baseline. As with the AUC scores, the Bayesian Network slightly outperforms the Logit model, which in turn outperforms the ranker aggregator. For "Software Development" and "Web Development" lift reaches 7 for the top 1% ranked instances; This means that if we focus on the top-1 percent of our resulting rankings, we are 7 times more likely to randomly select a positive instance than by choosing an instance from the entire testing dataset.

Finally, we compute the *ACC-n* (see Equation 6) for $n = 1, 2, ..., 5$ for all the proposed approaches, the feedback baseline, as well as a random ranker (based on the time of application). In Figure 3 we show the results. The Bayesian approach and the Logit model perform much better than the ranker aggregator and the feedback baseline in all categories. As expected, the Random ranker has the lowest performance. The actual probabilities of predicting a hiring decisions are impressive; In the "Software Development" category, the Bayesian approach ranks the applicants in a way that 28% of the times, the applicant who ranks first is the one who gets hired! At random, this probability is 10%. In other words, we see an improvement of 280%! In "Web Development" this probability is 20%. Similar observations can be made for the other two categories.

7. ECONOMETRIC ANALYSIS

In this section we study the association of each one of the variables on hiring decisions by analyzing the results of our Logit models from an econometric perspective. In particular we focus on the effect of each one of our variables on the hiring probability. This analysis is extremely useful since it provides insights about the behavior of the employers that is tricky (if not impossible) to capture by just looking at the coefficients of the Logit models (or the conditional probabilities of the proposed Bayesian network).

7.1 Marginal Effects Interpretation

The coefficient values of the proposed logit models do not represent the effect of each variable on the hiring probability [17]. To estimate the effect of each one of the variables, we have to calculate their marginal effects; in particular, we estimate the partial derivative of $\Pr(Y|\boldsymbol{X})$ *w.r.t.* each feature x_i:

$$\frac{\partial}{\partial x_i}\left(\frac{\exp(\boldsymbol{\beta X})}{1+\exp(\boldsymbol{\beta X})}\right) = \beta_i \frac{\exp(\boldsymbol{\beta X})}{(1+\exp(\boldsymbol{\beta X}))^2}\ . \tag{7}$$

Recall that $\boldsymbol{\beta}$ is the vector of coefficients of $\boldsymbol{X}$ (see Equation 3). To compute the marginal effects of each coefficient we evaluate the previous equation at the means of the feature distributions in our training sets [17].

Table 7 shows the marginal effects of all the variables broken down by task category. To better understand the interpretation of a marginal effect of a variable consider the following example: Suppose that we are interested on the effect of the feedback score f_i on the hiring probability of an applicant i. Recall that all our features have been transformed by equation 2. The marginal effect that f has on web development tasks is 0.004. This implies that if the transformed feedback score in an opening increases by one, all else equal,

Category	**Feedback**	**Ranker**	**Logit**	**Bayes Net**
Des & Mult	0.545	0.628	0.671	0.670
Soft Dev	0.533	0.675	0.722	0.728
Web Dev	0.548	0.660	0.730	0.738
Writing	0.537	0.634	0.664	0.646

Table 6: The per-category AUC for all the proposed approaches and the baseline.

Feature	**Des & Mult**	**Soft Dev**	**Web Dev**	**Writing**
time	-0.002***	-0.004***	-0.002***	-0.004***
bill	-0.002***	-0.002***	-0.002***	-0.003***
cosine	0.199***	0.281***	0.176***	0.318***
edu	-0.002**	-0.001	-0.002***	0.0
exams	0.002	0.004	0.001	0.003
exp	0.0	0.0	0.0	0.001***
feedback	0.004***	0.003*	0.004***	0.002
hours	0.002***	-0.002**	0.001**	0.0
jobs	0.0*	0.0***	0.0***	0.0**
new	0.012***	0.003	0.012***	0.007
tests	-0.001**	-0.001*	0.0.	0.001*
bid	0.003***	0.002***	0.002***	0.004***
pmi	0.005**	0.013***	0.009***	0.027***
rehires	0.028***	0.056***	0.041***	0.022***
skills	0.005***	0.006**	0.007***	0.009***

Table 7: Marginal Effects of the Logit Models in all categories. Significance codes: '*' 0.001,'**' 0.01, '*' 0.05, '.' 0.01**

the hiring probability of freelancer i, $\Pr(Y_i = hire|\boldsymbol{X}_i)$, will increase on average by $0.004 * 1$. In other words, the higher the positive value of a marginal effect, the stronger is the contribution to the hiring probability.

7.2 How each feature affects hiring decisions

Looking at Table 7, we observe that for all four categories, the signs of all significant coefficients agree[10], which provides an extra verification that these models explain the data well. In all categories the strongest effect comes from "cosine" and "rehires" (0.022 to 0.318). For example, in "Software Development", if an applicant is the only one within an opening who has been previously hired by the employer at hand, gets a boosts in hiring probability of 5.6%. On the other hand, in "Design & Multimedia", if an applicant's profile cosine similarity with the opening is higher than all other applicants in that opening by 0.01, then this applicant's probability will increase by 0.318 * 0.01 = 3.2%. These two observations are expected: in order for a freelancer to apply to an opening of an employer that has already collaborated with in the past, we can rationally assume that their collaboration was successful. Furthermore, given that their collaboration was successful, a risk-minimizer employer would choose to work again with this freelancer; Hence the high marginal effect. Similarly, the more coherent and informative the profile of an applicant is, and the more "similar" to the job opening, the higher is the chance that this applicant is suitable for the job.

The next two features that have a strong effect on hiring probability are the pmi score between the applicant's and the employer's countries and the skills' inner product of the contractor and the opening. For the skills, the strong effect is expected: it's the minimum requirements a freelancer must meet in order to get hired. The fact that most of the applicants satisfy these minimum requirements is the reason this effect is not stronger. For the pmi score, we notice that it has a relatively strong effect on all categories but writing. In writing, the effect is much higher (0.027). By looking in our data, we observed that the most frequent {employer-country, freelancer-country} pair that appears in "Writing" is {"USA",'Philippines"}, with a pmi score of 0.229, three times higher than the average PMI score in "Writing", 0.08. On the contrary, in "Web Development" for example, the most common {employer-country, freelancer-country} pair is {"USA","India"}, with a pmi score of 0.43, only slightly above the average pmi score of "Web Development" which is 0.42. The fact that the most frequent pair has a very high pmi score compared to the category average means that very frequently {"USA",'Philippines"} pairs will have higher transformed scores (Equation 2), which explains the strong effect of pmi scores in "Writing".

For new applicants, we observe that in "Web Development" and "Design & Multimedia" the coefficient is significant and positive. This is counterintuitive: being a new contractor shouldn't contribute at all to the hiring probability. However, recall that the reason we included this binary feature is to control for the zero "feedback", "hours" and "jobs" of a new freelancer (see also Sections 3 and 4.4). In our dataset, 10% of all new applicants' applications result in hires; since for all these hires "feedback","jobs" and "hours" were zero, the binary variable "new" balances out this "unfair" penalization by resulting to a positive effect.

As we discussed in Section 3.4, since the time of application is universal across all openings, we exclude it from the transformation of Equation 2. As a result, the negative effect we observe on table 7 is interpreted as follows: if an applicant moves up one rank in *time* (*e.g.* from 2 to 1), then the effect of this change in "Software Development" hiring probability will be (1-2)*(-0.004) = 0.4%. As a result, all else equal, the earlier a freelancer applies, the higher the hiring probability will be.

Finally "feedback" and the biding price ("bid") appear to have a small but significant positive effect, while the billing price and the level of education have small but significant negative effect.

8. DISCUSSION AND FUTURE WORK

In this work we studied hiring decisions in online labor markets. We proposed three different approaches that rank applicants based on their hiring probabilities. To build and evaluate our approaches, we used real transactional data from oDesk.com, and we showed that all our models perform significantly better over the vanilla feedback baseline. Finally we analyzed the (correlational) effect of each variable on the hiring probability of an applicant and found that the attributes that have the strongest positive effect are whether or not the freelancer and the employer have previously worked together, the available information on the freelancer's profile, the countries of the employer and the freelancer, the skillset of the contractor, as well as the time of application. In the following paragraphs we briefly discuss the limitations of our work and our future directions.

8.1 Limitations

Our first limitation comes from the fact that we built a global model under the assumption that our pool of employers is homogeneous in terms of hiring decisions. This

---

[10] We ignore the non-significant effects (*i.e.* those values that are not followed by at least one "*"). "hours" and "tests" are the only exceptions, but their marginal effects are really weak around 0.001)

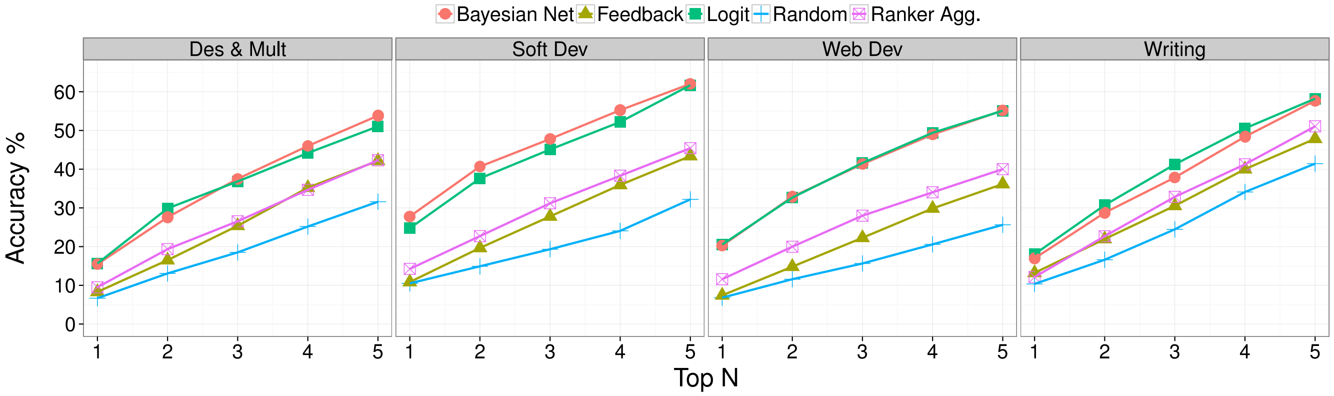

Figure 3: The accuracy at top-n (ACC-n) for all the proposed approaches, the feedback baseline and a random ranker, for all four categories we study.

is generally not true since employers' individual preferences might vary. Our analysis captures an average behavior of employers in the four task categories we examine.

The second important note is that our work does not study causality: we don't imply that the included features of our models have a causal relationship with hiring decisions. We clearly observe that those features can very well explain hiring decisions and that they are highly correlated with the probability of hiring.

The third limitation of our study derives from the dataset that we used. Since our goal was to study how employers make hiring decisions, we only considered openings that lead to a single hire. Our dataset does not capture the behavior of employers who decide (for whatever reason) not to hire anyone of the available applicants. As a result, our proposed approaches should not be used as-is to recommend potential candidates (*i.e.* freelancers that haven't applied yet) to employers. We briefly discuss in Section 8.2 how we can extend this analysis to create a suitable freelancer recommendation framework.

8.2 Future Directions

In the future, we intend to extend this study in three dimensions. First we plan to study "what attracts better applicants". At oDesk.com, at the end of each completed task, both the employer and the freelancer assign feedback to each other. By considering successful hires (positive instances) those that feedback scores were mutually high, and unsuccessful ones (negative instances) those that the pairwise feedback scores were low, we can study the job description characteristics that are correlated with the successful hires.

Second, as we discussed in the limitations section, our models are not ideal for recommending potential good candidates that haven't yet applied to an opening. We intend to extend this work by incorporating in our dataset openings that remain unfilled, and study why employers choose to not hire anyone of the available freelancers. We will then use this information to built a framework that will recommend high quality freelancers to new openings.

Finally, once we have a complete model that provides good estimates of the hiring probability of each *active freelancer* (not applicant) on each *active opening*, we can work towards maximizing the closing rate of openings in the marketplace. In particular, assume that we are given a bipartite graph, where the edges represent current applications from the available freelancers. The question we propose to study is how can we re-allocate these edges in order to maximize not just the filling rate of jobs, but also, the expected payoff (*e.g.* *OLM* revenue) from the available openings.

9. REFERENCES

[1] A. Acquisti and C. M. Fong. An experiment in hiring discrimination via online social networks. *Available at SSRN 2031979*, 2013.

[2] A. Agrawal, J. Horton, N. Lacetera, and E. Lyons. *Digitization and the Contract Labor Market: A Research Agenda*. University of Chicago Press, September 2013.

[3] A. J. Berinsky, G. A. Huber, and G. S. Lenz. Evaluating online labor markets for experimental research: Amazon.com's mechanical turk. *Political Analysis*, 20(3):351–368, 2012.

[4] S. Berry, J. Levinsohn, and A. Pakes. Automobile prices in market equilibrium. *Econometrica: Journal of the Econometric Society*, pages 841–890, 1995.

[5] G. Bouma. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*, pages 31–40, 2009.

[6] A. J. Bowlus. Matching freelancers and jobs: Cyclical fluctuations in match quality. *Journal of Labor Economics*, 13(2):pp. 335–350, 1995.

[7] E. Brynjolfsson and M. D. Smith. Frictionless commerce? a comparison of internet and conventional retailers. *Management Science*, 46(4):563–585, 2000.

[8] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

[9] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

[10] D. Chandler and J. Horton. Labor allocation in paid crowdsourcing: Experimental evidence on positioning, nudges and prices. In *Proceedings of the 3rd Human Computation Workshop, HCOMP*, volume 11, 2011.

[11] S. L. Cohen and K. A. Bunker. Subtle effects of sex role stereotypes on recruiters' hiring decisions. *Journal of Applied Psychology*, 60(5):566, 1975.

[12] H. K. Davison and M. J. Burke. Sex discrimination in simulated employment contexts: A meta-analytic investigation. *Journal of Vocational Behavior*, 56(2):225–248, 2000.

[13] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622. ACM, 2001.

[14] S. Forsythe, M. F. Drake, and C. E. Cox. Influence of applicant's dress on interviewer's selection decisions. *Journal of Applied Psychology*, 70(2):374, 1985.

[15] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of machine learning research*, 4:933–969, 2003.

[16] D. C. Gilmore, T. A. Beehr, and K. G. Love. Effects of applicant sex, applicant physical attractiveness, type of rater and type of job on interview decisions*. *Journal of Occupational Psychology*, 59(2):103–109, 1986.

[17] W. Greene. *Econometric Analysis*. Prentice Hall, 2012.

[18] Y. Hadass. The effect of internet recruiting on the matching of freelancers and employers. *Available at SSRN 497262*, 2004.

[19] J. J. Horton and L. B. Chilton. The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 209–218. ACM, 2010.

[20] J. J. Horton, D. G. Rand, and R. J. Zeckhauser. The online laboratory: Conducting experiments in a real labor market. *Experimental Economics*, 14(3):399–425, 2011.

[21] L. Hu. The hiring decisions and compensation structures of large firms. *Industrial and Labor Relations Review*, pages 663–681, 2003.

[22] N. Hu, J. Zhang, and P. Pavlou. Overcoming the j-shaped distribution of product reviews. *Communications of the ACM*, 52(10):144–147, 2009.

[23] N. Hu, J. Zhang, and P. A. Pavlou. Overcoming the j-shaped distribution of product reviews. *Commun. ACM*, 52(10):144–147, Oct. 2009.

[24] P. G. Ipeirotis and J. J. Horton. The need for standardization in crowdsourcing. CHI, 2011.

[25] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

[26] K. Jerath, P. S. Fader, and B. G. Hardie. New perspectives on customer "death" using a generalization of the pareto/nbd model. *Marketing Science*, 30(5):866–880, 2011.

[27] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[28] K. Kawakami, J. F. Dovidio, and S. van Kamp. Kicking the habit: Effects of nonstereotypic association training and correction processes on hiring decisions. *Journal of Experimental Social Psychology*, 41(1):68–75, 2005.

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

[30] M. Kokkodis and P. G. Ipeirotis. The utility of skills in online labor markets. In *International Conference on Information Systems (ICIS)*, 2014.

[31] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[32] B. Li, A. Ghose, and P. G. Ipeirotis. Towards a theory model for product search. In *Proceedings of the 20th international conference on World wide web*, pages 327–336. ACM, 2011.

[33] M. Lindeboom, J. V. Ours, and G. Renes. Matching employers and freelancers: An empirical analysis on the effectiveness of search. *Oxford Economic Papers*, 46(1):pp. 45–67, 1994.

[34] K. S. Lyness and M. K. Judiesch. Are women more likely to be hired or promoted into management positions? *Journal of Vocational Behavior*, 54(1):158–173, 1999.

[35] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[36] W. Mason and D. J. Watts. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter*, 11(2):100–108, 2010.

[37] P. Nelson. Information and consumer behavior. *The Journal of Political Economy*, 78(2):311–329, 1970.

[38] J. D. Olian, D. P. Schwab, and Y. Haberfeld. The impact of applicant gender compared to qualifications on hiring recommendations: A meta-analysis of experimental studies. *Organizational Behavior and Human Decision Processes*, 41(2):180–195, 1988.

[39] A. Pallais. Ineffiient hiring in entry-level labor markets. *Available at SSRN 2012131*, 2012.

[40] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

[41] C. Quoc and V. Le. Learning to rank with nonsmooth cost functions. *NIPSâĂŹ07*, 19:193, 2007.

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

[43] S. M. Raza and B. N. Carpenter. A model of hiring decisions in real employment interviews. *Journal of Applied Psychology*, 72(4):596, 1987.

[44] A. D. Shaw, J. J. Horton, and D. L. Chen. Designing incentives for inexpert human raters. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pages 275–284. ACM, 2011.

[45] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM, 2008.

[46] K. Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

[47] A. Trotman. Learning to rank. *Information Retrieval*, 8(3):359–381, 2005.

[48] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[49] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199. ACM, 2008.

[50] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 391–398. ACM, 2007.

[51] J. Xu, T.-Y. Liu, M. Lu, H. Li, and W.-Y. Ma. Directly optimizing evaluation measures in learning to rank. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 107–114. ACM, 2008.

[52] V. Yakubovich. Weak ties, information, and influence: How freelancers find jobs in a local russian labor market. *American sociological review*, 70(3):408–421, 2005.

[53] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278. ACM, 2007.

[54] Z. Zheng, K. Chen, G. Sun, and H. Zha. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 287–294. ACM, 2007.