http://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Articles in Advance, pp. 1–21
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Estimating Effects of Incentive Contracts in Online Labor Platforms

Nur Kaynar,[a] **Auyon Siddiq**[b,*]

[a] Samuel Curtis Johnson Graduate School of Management, Cornell University, Ithaca, New York 14853; [b] Anderson School of Management, University of California, Los Angeles, California 90095
*Corresponding author
Contact: nur.kaynar@cornell.edu (NK); auyon.siddiq@anderson.ucla.edu, https://orcid.org/0000-0003-2977-5558 (AS)

Received: August 5, 2019
Revised: September 1, 2020; March 29, 2021; August 23, 2021
Accepted: November 29, 2021
Published Online in Articles in Advance: ■■■■ ■■, 2022

https://doi.org/10.1287/mnsc.2022.4450

Abstract. The design of performance based incentives—commonly used in online labor platforms—can be naturally can be naturally posed as a moral hazard principal-agent problem. In this setting, a key input to the principal's optimal contracting problem is the agent's production function: the dependence of agent output on effort. Although agent production is classically assumed to be known to the principal, this is unlikely to be the case in practice. Motivated by the design of performance-based incentives, we present a method for estimating a principal-agent model from data on incentive contracts and associated outcomes, with a focus on estimating agent production. The proposed estimator is statistically consistent and can be expressed as a mathematical program. To circumvent computational challenges with solving the estimation problem exactly, we approximate it as an integer program, which we solve through a column generation algorithm that uses hypothesis tests to select variables. We show that our approximation scheme and solution technique both preserve the estimator's consistency and combine to dramatically reduce the computational time required to obtain sound estimates. To demonstrate our method, we conducted an experiment on a crowdwork platform (Amazon Mechanical Turk) by randomly assigning incentive contracts with varying pay rates among a pool of workers completing the same task. We present numerical results illustrating how our estimator combined with experimentation can shed light on the efficacy of performance-based incentives.

History: Accepted by Chung Piaw Teo, optimization.
Supplemental Material: The data files and e-companion are available at https://doi.org/10.1287/mnsc.2022.4450.

Keywords: principal-agent model • incentive contracts • estimation • integer programming • online labor platforms

1. Introduction

The extent to which financial incentives increase worker performance is of interest in many employment settings. This question has taken on renewed relevance because of the emergence of online labor platforms, which are used for on-demand jobs like ride-hailing (e.g., Uber, Lyft), delivery (Postmates), freelance work (Upwork), and short, discrete tasks (Amazon Mechanical Turk). Although these platforms support different types of work, they also have common features: workers are hired and compensated on a per-task basis, work is done remotely with limited supervision, and workers may be offered performance-based incentives.[1]

The design of performance-based incentives can be naturally posed as a moral-hazard principal-agent problem, in which an agent's (worker's) effort is hidden from the principal (employer), and the agent's output depends stochastically on their effort (Holmstrom 1979, Grossman and Hart 1983, Sappington 1991). In this setting, the relationship between worker output and effort corresponds to a set of parameters that define agent production. If these parameters are known, then the principal's problem of optimally designing incentives is well defined and potentially convex (Grossman and Hart 1983).

In practice, however, the relationship between worker effort and output is unlikely to be known a priori. Given data on incentives and associated output, this dependence can be inferred by specifying an appropriate agent model and estimating the parameters that govern agent production. Despite the importance of principal-agent models to the analysis of incentive contracts, estimation problems of this nature are scarce in the literature, even for simple agent models. Estimating an agent model from observational or experimental data can be a useful step toward the design of incentive contracts in practice and can also play a role in estimating agent welfare under a given contract.

Our main contribution is to present an estimator for a principal-agent model with hidden actions, along with an algorithm for solving the estimation problem. Our focus is on estimating model parameters that encode agent production, namely, the conditional distribution over output for each effort level. To reflect a moral-hazard setting, we assume no data are available on agent effort, which makes the estimation problem computationally nontrivial. We make two methodological contributions in particular: (1) we provide an estimator that is statistically *consistent* under appropriate conditions, meaning it uncovers the true model parameters as the sample size goes to infinity, and (2) we develop an accompanying solution technique that is computationally efficient and preserves consistency.

The agent model we consider is nonparametric, in that we do not assume functional forms for the dependence of agent output on effort, and we assume both output and effort levels are discrete. This specification has two important consequences. First, it admits a simple and *tractable* formulation of a general optimal contracting problem, which allows us to readily solve for an optimal contract under the estimated agent model. Second, estimating agent models is well known to be challenging because of a need to embed the agent's problem, itself an optimization problem, within the estimator (Bajari et al. 2007). Our modeling approach allows us to express the estimator as an integer program, which admits a structure that supports obtaining estimates quickly using a novel solution technique. In addition to these computational advantages, our nonparametric model naturally handles threshold-based incentives, which commonly arise in practice, and is flexible enough to have strong predictive performance on a variety of datasets without overfitting.

In an empirical study, we show how our estimator can be combined with experimentation to characterize worker output over a class of incentive contracts, which in turn allows us to solve for an optimal contract from the given class. In a randomized experiment, we recruited a pool of 500 workers from a crowdwork platform (Amazon Mechanical Turk), each of whom was asked to complete an identical proofreading task, with output measured by the number of typos identified. We created exogenous variation in payments by randomly generating the parameters of an incentive contract for each worker. We then applied our estimator to the experimental data to investigate the effect of performance-based incentives on worker output. Our results complement existing findings that incentives do increase output in crowdwork, although we observe diminishing returns to output beginning at relatively low payments.

Our model has limitations. The agent model does not include common features of principal-agent problems; in particular, we do not address risk aversion or unobserved agent heterogeneity in this paper. This abstraction arises from our focus on obtaining consistent estimates (potentially for a large number of parameters) while maintaining computational tractability. Generalizing our estimation procedure to accommodate a richer class of agent models may expand its applicability in practice. Furthermore, our nonparametric approach may be unsuitable for settings with limited data, because it may require estimating many parameters if the action or outcome space is large.

The remainder of the paper is organized as follows. Section 2 defines the agent model, presents the associated estimator, and establishes consistency. Section 3 presents an exact formulation of the estimator as an integer program and discusses the computational challenges of the exact representation. Section 4 develops an approximate estimator and an accompanying solution technique, which dramatically improve tractability while preserving consistency of the exact estimator. Section 5 describes the randomized experiment and demonstrates the application of our estimator to experimental data. Section 6 concludes. All proofs are contained in the e-companion.

1.1. Related Literature

Existing work on estimating principal-agent models is relatively limited. Several papers have focused on employee compensation. Ferrall and Shearer (1999) use payroll records of copper mine workers to estimate the cost of employee risk aversion. Paarsch and Shearer (2000) use a tree-planting firm's records to estimate the impact of providing piece-rate compensation over fixed wages, and Shearer (2004) addresses the same question through a field experiment. Duflo et al. (2012) estimate an agent model to assess the impact of financial incentives for schoolteachers and use the model to estimate cost reductions associated with a counterfactual payment scheme. Misra et al. (2005) and Misra and Nair (2011) both estimate agent models based on salesforce compensation and empirically validate the models on out-of-sample data. Gayle and Miller (2015) focus on identifying a general principal-agent model motivated by managerial compensation. Georgiadis and Powell (2022) provide conditions under which a single A/B test can estimate the impact of marginal changes to an incentive contract, using the classical principal-agent model from Holmstrom (1979). Applications beyond employee compensation include agriculture (de Zegher et al. 2019) and healthcare (Vera-Hernandez 2003, Lee and Zenios 2012, Aswani et al. 2019).

Previous work on estimating principal-agent models have used a variety of methods, including least squares (Lee and Zenios 2012), simulated method of moments (Paarsch and Shearer 2000, Misra et al. 2005, Misra and Nair 2011, Duflo et al. 2012), simulation-based maximum likelihood estimation (Ferrall and Shearer 1999, Vera-Hernandez 2003, Aswani et al. 2019), and numerical minimization of a sum-of-squares

criterion (Gayle and Miller 2015). Our approach differs in that we formulate the estimation problem as an integer program, which is made possible by our specification of the agent model, in particular by assuming agent actions and outputs are discrete.

We solve the estimation problem using a column generation algorithm that exploits statistical properties of the formulation. Column generation methods have been successfully applied to solve large-scale linear and integer programs in which an extremely large number of variables is the main obstacle to obtaining optimal solutions (Vanderbeck and Wolsey 1996, Barnhart et al. 1998, Lubbecke and Desrosiers 2005). These methods typically involve solving a tractable *master* problem that restricts attention to a subset of decision variables and selectively introducing variables into the formulation until a certificate of optimality or alternative termination criterion is met. In contrast to existing column generation methods that select columns using dual information, our algorithm uses a series of nonparametric hypothesis tests to identify variables to introduce into the master problem. This approach is viable in our setting because the decision variables are mapped to empirical probability distributions constructed from the data, giving them a clear statistical interpretation. By comparison, existing column generation methods have typically been applied to deterministic settings where the model parameters may not have any statistical meaning (see Lubbecke and Desrosiers 2005 for a review).

The estimation problem we consider is also closely related to a recent line of research on *inverse* optimization, in which optimization model parameters are inferred from (potentially noisy) solution data. Existing approaches to inverse optimization have focused on estimating parameters of linear programs (Chan et al. 2019) or general convex optimization problems (Keshavarz et al. 2011, Bertsimas et al. 2015, Aswani et al. 2018). Similar to our paper, the literature on inverse optimization is often motivated by an interest in estimating a model of agent decision-making from data (Aswani et al. 2018, Esfahani et al. 2018). Our paper differs in that instead of assuming the agent solves a convex optimization problem, we assume they select a utility-maximizing action from a discrete set, which calls for a different solution approach.

1.2. Notation

For convenience, we describe notational conventions here. Sets are denoted by uppercase letters, scalars by lowercase letters, and vectors and matrices by lowercase boldfaced letters. For a $m \times d$ matrix $\mathbf{x}$, let $\mathbf{x}_a$ be the ath row, and let x_{aj} be the entry in the ath row and jth column. For vectors $\mathbf{x}$ and $\mathbf{y}$, let $||\mathbf{x}||_1 = \sum_{a=1}^{m} \sum_{j=1}^{d} |x_{aj}|$ denote the ℓ_1-norm, and let $\mathbf{x} \circ \mathbf{y} = \sum_{a=1}^{m} \sum_{j=1}^{d} x_{aj} y_{aj}$ be the elementwise product. For a matrix of random variables $\mathbf{x}_n$, we use both $\mathbf{x}_n \to \mathbf{x}^0$ and $\text{plim}_{n\to\infty} \mathbf{x}_n = \mathbf{x}^0$ to mean $\mathbf{x}_n$ converges elementwise in probability to $\mathbf{x}^0$ as $n \to \infty$, unless otherwise specified. Define the indicator variable $\mathbb{I}\{\cdot\} = 1$ if the statement $\{\cdot\}$ is true, and zero otherwise. For simplicity, we use $\mathbb{E}(\cdot)$ for all expectations and $\Pr(\cdot)$ for all probabilities throughout the paper.

2. Estimator

In this section, we define the principal-agent model (Section 2.1), formulate the estimator (Section 2.2), and prove its statistically consistency (Section 2.3).

2.1. Principal-Agent Model and Contract Data

Our principal-agent model is a discrete analogue to the canonical model introduced by Grossman and Hart (1983). We choose this model for both its simplicity and generality. The interaction between the principal and agent proceeds as follows. The principal selects a *contract* to offer the agent, which is a mapping of payments to *outcomes* (i.e., agent output). Outcomes depend stochastically on a costly *action* (i.e., effort) taken by the agent. Outcomes are observed by both parties, whereas actions are observed only by the agent.

Let A and J index the set of possible actions and outcomes, respectively, where $|A| = m$ and $|J| = d$. Let ξ be a discrete random variable denoting the outcome, where $\xi \in J$. We denote a contract by $\mathbf{r} \in \mathbb{R}_+^d$, where r_j is the payment to the agent if outcome j is realized. Let $\mathbf{c} \in \mathbb{R}_+^m$ denote action costs, where c_a is the cost to the agent of taking action a. The dependence of outcomes on actions is governed by a parameter matrix $\boldsymbol{\pi} \in \mathbb{R}_+^{m\times d}$, where π_{aj} denotes the probability that action a leads to outcome j. We use $\boldsymbol{\pi}_a \in \mathbb{R}^d$ to denote the probability mass function over outcomes associated with action a.

Given a contract $\mathbf{r}$, the agent selects an action to maximize their expected utility by solving

$$\max_{a\in A} \left\{ \sum_{j\in J} \pi_{aj} r_j - c_a \right\}. \tag{1}$$

We assume that there exists at least one action that yields nonnegative expected utility for the agent. If for each $a \in A$, the distribution $\boldsymbol{\pi}_a$ is known, the principal's problem of selecting a utility-maximizing contract can be formulated as a convex optimization problem (Grossman and Hart 1983). We take an inverted perspective in this paper, by instead supposing that the distributions $\boldsymbol{\pi}_a$, $a \in A$ are unknown, but may be estimated given appropriate data. In particular, suppose we have data from n identical agents,[2]

$$(\mathbf{r}^i, \xi^i),\ i \in I, \tag{2}$$

where I indexes pairs of incentive contracts and outcomes, and $|I| = n$. Let $R \subseteq \mathbb{R}^d$ be the set of all possible values of $\mathbf{r}^i$. Furthermore, we assume the contract set R

is bounded, in that there exists a constant $\bar{r}$ such that $\bar{r} = \sup_{\mathbf{r}\in R} ||\mathbf{r}||_0 < \infty$. The assumption that R is bounded ensures that the contracts $\mathbf{r}^i$ remain bounded as $n \to \infty$.

Next, suppose we have no observations of past agent actions, and only know the agent's action set A and associated costs, $\mathbf{c}$. A natural question in this setting is to predict the distribution of the outcome ξ^{n+1} under a new contract $\mathbf{r}^{n+1}$. If $\boldsymbol{\pi}$ is known, then this prediction task reduces to solving the agent's problem (1) under $\mathbf{r}^{n+1}$, identifying the optimal action a, and taking $\boldsymbol{\pi}_a$ to be the distribution of ξ^{n+1}. Therefore, the matrix $\boldsymbol{\pi}$ is the key model primitive for predicting the outcome associated with $\mathbf{r}^{n+1}$. Our goal is to estimate the parameter $\boldsymbol{\pi}$ from data that takes the form given in (2).

The assumption that agent costs are unknown is relatively mild in our setting, given that agent actions are also hidden. From a model-fitting perspective, it suffices to select $\mathbf{c}$ to cover a range of *possible* costs to the agent. In our numerical study in Section 5, we take a machine learning perspective by treating the number of agent actions m and the set of costs $\mathbf{c}$ as hyperparameters that are tuned prior to fitting the model.

2.2. Estimator Formulation

Next, we formalize the estimator for $\boldsymbol{\pi}$.[3] Let

$$A(\mathbf{r}, \boldsymbol{\pi}) = \operatorname*{argmax}_{a\in A} \left\{ \sum_{j\in J} \pi_{aj} r_j - c_a \right\} \tag{3}$$

denote the set of optimal actions under the contract $\mathbf{r}$ and the model $\boldsymbol{\pi}$. Let $\mathbf{y} \in \{0,1\}^{n\times d}$ be a binary matrix that encodes historical outcomes, where $y^i_j = 1$ if $\xi^i = j$ and $y^i_j = 0$ if $\xi^i \neq j$. For each $i \in I$, let x^i be a decision variable representing the agent action under contract $\mathbf{r}^i$, and let $\boldsymbol{\omega} \in \mathbb{R}_+^{m\times d}$ be a set of auxiliary variables, which will be used to model empirical probabilities. For fixed $\boldsymbol{\pi}$, the loss function $L_n(\boldsymbol{\pi})$ is then given by

$$L_n(\boldsymbol{\pi}) = \operatorname*{minimize}_{\mathbf{x},\boldsymbol{\omega}} \sum_{a\in A}\sum_{j\in J} |\pi_{aj} - \omega_{aj}|, \tag{4a}$$

$$\text{subject to} \quad x^i \in A(\mathbf{r}^i, \boldsymbol{\pi}), \quad i \in I, \tag{4b}$$

$$\omega_{aj} = \frac{1}{|\{i \,|\, x^i = a\}|} \sum_{i\in\{i|x^i=a\}} y^i_j, \quad a \in A, j \in J. \tag{4c}$$

In this formulation, (4b) restricts each x^i to be an optimal action under $\mathbf{r}^i$ and $\boldsymbol{\pi}$, and (4c) defines ω_{aj} to be the empirical probability that action a leads to outcome j. The empirical probability ω_{aj} depends on the cardinality of the set $\{i|x^i = a\}$, which is the implied number of data points for which the action a is optimal for the agent under $\boldsymbol{\pi}$. The objective (4a) then simply measures the error between the model probabilities $\boldsymbol{\pi}$ and the implied empirical probabilities $\boldsymbol{\omega}$.

Next, let Π be a compact set representing the parameter set for $\boldsymbol{\pi}$. The estimate is then attained at a minimizer of the loss function over Π:

$$\text{(PA)} \qquad \hat{\boldsymbol{\pi}}_n \in \operatorname*{arg\,min}_{\boldsymbol{\pi}\in\Pi} L_n(\boldsymbol{\pi}).$$

It will be convenient to interpret the parameter set Π as the Cartesian product of m probability simplices: one for each action $a \in A$.

2.3. Statistical Consistency

Let us now suppose there exists a "true" model parameter $\boldsymbol{\pi}^0$ that is responsible for generating the data $(\mathbf{r}^i, \xi^i)$, $i \in I$. We say an estimator is *statistically consistent* if it produces a sequence of estimates $\hat{\boldsymbol{\pi}}_n$ such that $\hat{\boldsymbol{\pi}}_n \to \boldsymbol{\pi}^0$ as $n \to \infty$. This raises a natural question: Under what conditions, if any, is PA a consistent estimator? In general, whether an estimator is consistent depends on the specification of the loss function. Our main result in this section, Theorem 1, shows that minimizing the loss function $L_n(\boldsymbol{\pi})$ defined in (4) produces an estimate that is indeed consistent.

Before addressing the consistency of PA, we first formalize the statistical model that generates the data. First, we define an important set that is used throughout our analysis:

$$R_a(\boldsymbol{\pi}) = \left\{ \mathbf{r} \in R \,\middle|\, a \in \operatorname*{argmax}_{a\in A} \sum_{j\in J} \pi_{aj} r_j - c_a \right\}, \tag{5}$$

where $R_a(\boldsymbol{\pi})$ represents the subset of the contract set R where action $a \in A$ is optimal for the agent, given the model $\boldsymbol{\pi}$. Next, we impose two assumptions. The first assumption formalizes the data generation process.

Assumption 1 (Data). *The data* $(\mathbf{r}^i, \xi^i)$, $i \in I$, *are independent samples of random variables* $(\mathbf{r}, \xi)$, *where* (i) $(\mathbf{r}, \xi)$ *are jointly distributed with support* $R \times J$, (ii) $\mathbf{r}$ *has continuous marginal density function* $f(\mathbf{r})$, (iii) $\Pr(\mathbf{r} \in R_a(\boldsymbol{\pi})) > 0$ *for all* $a \in A$ *and* $\boldsymbol{\pi} \in \Pi$, *and* (iv) ξ *has conditional mass function* $\pi^0_{aj} = \Pr(\xi = j | \mathbf{r} \in R_a(\boldsymbol{\pi}^0))$, *where* $\boldsymbol{\pi}^0 \in \Pi$.

Assumption 1(iv) states that there exists a "true" parameter, denoted $\boldsymbol{\pi}^0$, that is responsible for generating the outcomes ξ^i, based on the agent model (1). The statements in (ii) and (iii) are regularity conditions that we use to prove convergence of $\hat{\boldsymbol{\pi}}_n$ to $\boldsymbol{\pi}^0$.[4] Our assumption that the data are independent and identically distributed (i.i.d.) is commonly used in the statistical learning literature to obtain similar consistency results.[5]

Next, we consider an additional condition that is important for our main result in Theorem 1.

Assumption 2 (Identifiability). *For every* $\boldsymbol{\pi} \in \Pi$ *such that* $\boldsymbol{\pi} \neq \boldsymbol{\pi}^0$, *there exists an* (a, j) *such that*

$$\pi_{aj} \neq \sum_{b\in A} \pi^0_{bj} \cdot \Pr(\mathbf{r} \in R_b(\boldsymbol{\pi}^0) | \mathbf{r} \in R_a(\boldsymbol{\pi})).$$

Assumption 2 is an *identifiability* condition, which ensures that the unknown parameter $\boldsymbol{\pi}^0$ can be learned

from the data. This assumption implies a one-to-one mapping between the parameter set Π and the joint distribution of the random variables $(\mathbf{r}, \xi)$. In other words, Assumption 2 guarantees that the distribution of $(\mathbf{r}, \xi)$ is unique for each $\boldsymbol{\pi} \in \Pi$. In the absence of model identifiability, there may exist multiple parameters values in Π that generate the same distribution in the observed data; in this case, it is impossible for any estimation procedure to pinpoint the true $\boldsymbol{\pi}^0$. Identifiability conditions like Assumption 2 are commonly imposed to prove consistency of an estimator (Van der Vaart 2000).

We can now present the main result of Section 2, which shows that the estimator PA uncovers the true model parameter $\boldsymbol{\pi}^0$ under Assumptions 1 and 2.

Theorem 1. *Let Assumption 1 hold. Then* $\hat{\boldsymbol{\pi}}_n \to \boldsymbol{\pi}^0$ *for any* $\boldsymbol{\pi}^0 \in \Pi$ *if and only if Assumption 2 holds.*

Theorem 1 states that the estimator PA is statistically *consistent*, which is defined as the convergence of estimates to the true model parameters (Van der Vaart 2000, Casella and Berger 2002, Bickel and Doksum 2015). Despite being an asymptotic property, consistency is valuable in practice, because it guarantees that parameter estimates will generally improve with additional data. Conversely, an inconsistent estimator may produce inaccurate estimates of the unknown parameters, even if data are abundant. In pathological cases, the accuracy of an inconsistent estimator may even decrease with additional data. Therefore, a proof of consistency provides some assurance that parameter estimates will be "reasonable" under moderate sample sizes, and that the accuracy of the estimates will continue to improve with additional data.

Having established that the estimate $\hat{\boldsymbol{\pi}}_n$ behaves desirably, we now shift our attention to solving the estimator PA. In a setting where agent actions are observable, a consistent estimate of $\boldsymbol{\pi}^0$ can be obtained by simply counting the relative frequency of outcomes associated with each action. In contrast, when agent actions are hidden, the estimation problem is nontrivial. At a high level, our approach for solving PA will be to leverage integer programming within a broader solution algorithm. The key challenge we face in solving PA is to develop a solution method that satisfies two criteria: (1) is computationally efficient and (2) preserves the statistical consistency of PA. We note here that an alternative solution approach might be to formulate and solve a convex approximation to PA, although doing so may result in an inconsistent estimator. We will therefore focus on obtaining solutions to PA directly.

3. Exact Integer Programming Formulation

In this section, we present an approach for solving PA exactly using integer programming. We will assume throughout that the parameter set Π is of the form

$$\Pi = \left\{ \boldsymbol{\pi} \in Q_\pi \,|\, \boldsymbol{\pi} \geq 0, \sum_{j \in J} \pi_{aj} = 1 \text{ for } a \in A \right\}, \tag{6}$$

where Q_π is a polyhedron defined by a set of linear inequalities in $\boldsymbol{\pi}$. Assuming that $\boldsymbol{\pi} \in Q_\pi$ permits the formulation of the estimator as a mixed-integer linear program, while also allowing Π to capture various shape constraints on the parameter $\boldsymbol{\pi}$. For example, if

$$Q_\pi = \left\{ \boldsymbol{\pi} | \sum_{k=j}^{d} \pi_{ak} \leq \sum_{k=j}^{d} \pi_{(a+1)k}, a \in \{1, 2, \ldots, m-1\}, j \in J \right\}, \tag{7}$$

then for any $a \in \{1, 2, \ldots, m-1\}$, Π forces the distribution $\boldsymbol{\pi}_{a+1}$ to stochastically dominate $\boldsymbol{\pi}_a$ in the first order, meaning costlier actions taken by the agent are more likely to generate high output. Alternatively, if $Q_\pi = \mathbb{R}^{m \times d}$, then Π permits each $\boldsymbol{\pi}_a$ to be any valid probability mass function over the outcomes J. We will assume throughout that Π satisfies (6) unless otherwise stated.

Although PA is based on an intuitive loss function, a naive formulation of PA as a mathematical program yields nonlinear terms in the objective, because of how the variable $\boldsymbol{\omega}$ enters the loss expression (4a). However, the estimation problem is amenable to mathematical programming approaches under a slight modification. Consider the proxy loss function

$$\begin{aligned} Z_n(\boldsymbol{\pi}) = \underset{\mathbf{x}, \boldsymbol{\eta}, \boldsymbol{\omega}}{\text{minimize}} \; & \frac{1}{n} \sum_{a \in A} \sum_{j \in J} \eta_{aj} |\pi_{aj} - \omega_{aj}| \\ & \eta_{aj} = |\{i | x^i = a\}|, \quad a \in A, j \in J, \\ & (4b) - (4c). \end{aligned} \tag{8}$$

Here, η_{aj} is the number of observations for which action a is implied to be optimal for the agent under $\boldsymbol{\pi}$. The loss function $Z_n(\boldsymbol{\pi})$ can be interpreted as a scaled version of $L_n(\boldsymbol{\pi})$, where the (a, j) component of $L_n(\boldsymbol{\pi})$ is scaled by η_{aj}/n. The proxy estimator is then given by

$$\boldsymbol{\pi}_n^* = \underset{\boldsymbol{\pi} \in \Pi}{\arg\min} \, Z_n(\boldsymbol{\pi}). \tag{9}$$

Next, we show that (9) can be formulated exactly as a mixed-integer linear program. With a slight abuse of notation, let $\mathbf{x} \in \{0,1\}^{m \times n}$ be binary variables, where $x_a^i = 1$ if $a \in A(\mathbf{r}^i, \boldsymbol{\pi})$, and $x_a^i = 0$ if $a \notin A(\mathbf{r}^i, \boldsymbol{\pi})$. Introducing the auxiliary variables z_{aj} to linearize the absolute values in the objective of (8) (Bertsimas and Tsitsiklis 1997) yields the following formulation:

$$\underset{\boldsymbol{\pi}, \mathbf{x}, \mathbf{z}}{\text{minimize}} \quad \sum_{a \in A} \sum_{j \in J} z_{aj} \tag{10a}$$

$$\text{subject to} \quad z_{aj} \geq \frac{1}{n} \sum_{i \in I} (y_j^i - \pi_{aj}) x_a^i \quad a \in A, \; j \in J, \tag{10b}$$

$$z_{aj} \geq \frac{1}{n} \sum_{i \in I} (\pi_{aj} - y_j^i) x_a^i \quad a \in A, \; j \in J, \tag{10c}$$

$$\left(\sum_{j\in J}\pi_{aj}r_j^i - c_a\right)x_a^i \geq \left(\sum_{j\in J}\pi_{bj}r_j^i - c_b\right)x_a^i, \quad i\in I, a\in A, b\in A, \tag{10d}$$

(PA-C)

$$\sum_{j\in J}\pi_{aj} = 1, \quad a\in A, \tag{10e}$$

$$\sum_{a\in A}x_a^i = 1, \quad i\in I, \tag{10f}$$

$$x_a^i \in \{0,1\}, \quad a\in A, \tag{10g}$$

$$\pi_{aj} \geq 0, \quad a\in A, j\in J, \tag{10h}$$

$$\boldsymbol{\pi}\in Q_\pi. \tag{10i}$$

Objective (10a) and Constraints (10b)–(10c) represent the error function $\frac{1}{n}\|\boldsymbol{\eta}\circ(\boldsymbol{\pi}-\boldsymbol{\omega})\|_1$ given in (8). Constraint (10d) ensures that $x_a^i = 1$ only if $a\in A(\mathbf{r}^i,\boldsymbol{\pi})$, that is, only if action a is optimal under contract $\mathbf{r}^i$ and the parameter $\boldsymbol{\pi}$. Constraint (10e) ensures that the probability vector $\boldsymbol{\pi}_a$ sums to one for each $a\in A$, and Constraint (10f) forces exactly one action to be selected as optimal for each contract $i\in I$. Next, we establish an equivalence between the proxy estimator PA-C and the original estimator PA.

Proposition 1. *The estimate $\boldsymbol{\pi}_n^*$ attained at a solution to* PA-C *is* (i) *a minimizer of the proxy loss function $Z_n(\boldsymbol{\pi})$*, (ii) *an asymptotic minimizer of the loss function $L_n(\boldsymbol{\pi})$, $|L_n(\boldsymbol{\pi}_n^*) - L_n(\hat{\boldsymbol{\pi}}_n)| \to 0$, and* (iii) *consistent, $\boldsymbol{\pi}_n^* \to \boldsymbol{\pi}^0$.*

In Proposition 1, (i) establishes that the mathematical program PA-C is equivalent to the proxy estimator (9), (ii) establishes that solving PA-C asymptotically produces an optimal solution to PA, and (iii) confirms that PA-C is also a consistent estimator for $\boldsymbol{\pi}^0$. Based on the equivalence in Proposition 1, we will refer to PA-C as the *exact* estimator in the remainder of the paper.

The intuition behind Proposition 1 is as follows. Note that $Z_n(\boldsymbol{\pi})$ can be interpreted as a reweighted version of $L_n(\boldsymbol{\pi})$, where for each (a, j), the term $|\pi_{aj} - \omega_{aj}|$ is multiplied by the weight η_{aj}/n. As $n\to\infty$, the minimal possible loss for both estimators occurs when $\pi_{aj} = \omega_{aj}$ for all (a, j). Therefore, minimizing $Z_n(\boldsymbol{\pi})$ also minimizes $L_n(\boldsymbol{\pi})$, in the limit.

Next, (10a)–(10d) contains bilinear terms because of the product of the decision variables $\mathbf{x}$ and $\boldsymbol{\pi}$. Because $\mathbf{x}$ and $\boldsymbol{\pi}$ are binary and continuous variables, respectively, these product terms can be linearized exactly using well-known reformulation techniques (Glover 1975, Adams et al. 2004), leading to a mixed-integer linear program. However, a drawback of this approach is that linearizing products of variables is known to yield weak linear programming relaxations (Adams et al. 2004, Luedtke et al. 2012), which can make solving PA-C using off-the-shelf optimization solvers challenging, even for moderately sized data sets. In the next section, we propose an approximation to PA-C that bypasses the linearization step while remaining statistically well behaved.

4. Restricted Estimator and Statistical Column Generation

We begin this section by proposing an approximation of PA-C, which we call PA-D, based on replacing the parameter set Π with a discrete subset $\tilde{\Pi}$ (Section 4.1). We then present a data-driven procedure for constructing the parameter set $\tilde{\Pi}$ and investigate the behavior of the resulting estimates (Section 4.2). Then, to solve PA-D, we present a column generation algorithm based on hypothesis testing and show that the algorithm preserves statistical consistency (Section 4.3). We conclude the section by comparing the numerical performance of the statistical column generation algorithm with off-the-shelf optimization solvers (Section 4.4).

4.1. Restricted Estimator

Our approach to approximately solving PA-C will be to minimize the proxy loss $Z_n(\boldsymbol{\pi})$ over a restricted parameter set $\tilde{\Pi}\subseteq\Pi$ instead of Π. The advantage of this "restricted estimator" is that the agent optimality conditions (10d) can be enforced without introducing bilinear terms into the formulation, which allows us to avoid the computational challenges that often accompany linearization techniques.

Next, we define a set that plays a critical role in our estimation procedure: Let $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{|S|}\}\subseteq\mathbb{R}_+^d$ be a set of vectors indexed by S, where $\sum_{j\in J}v_{sj} = 1$ and $\mathbf{v}_s \geq 0$ for all $s\in S$. We refer to each $\mathbf{v}_s$ as a *candidate distribution*. Next, let the restricted parameter set be defined as

$$\tilde{\Pi} = \{\boldsymbol{\pi}\in\Pi \,|\, \boldsymbol{\pi}_a \in V \text{ for } a\in A\}, \tag{11}$$

and let

$$\tilde{\boldsymbol{\pi}}_n = \operatorname*{arg\,min}_{\boldsymbol{\pi}\in\tilde{\Pi}} Z_n(\boldsymbol{\pi}) \tag{12}$$

be the associated estimate. For each action $a\in A$, the parameter set $\tilde{\Pi}$ restricts the probability distribution $\boldsymbol{\pi}_a$ to lie in the set of candidate distributions V. We assume throughout that $\tilde{\Pi}$ is nonempty.[6]

Similar to the exact estimator (9), the restricted estimator (12) can also be formulated as a mixed-integer linear program. The intuition behind this formulation is to construct the estimate $\boldsymbol{\pi}$ in a row-wise manner by assigning a candidate distribution in V to each row of $\boldsymbol{\pi}$. To that end, let $\mathbf{w}\in\{0,1\}^{m\times S}$, $\mathbf{x}\in\{0,1\}^{n\times S}$ and $\phi\in\{0,1\}^{n\times m\times S}$ be binary variables with the following interpretations: $w_{as} = 1$ if the candidate distribution $\mathbf{v}_s$ is assigned to be the distribution $\boldsymbol{\pi}_a$, $x_s^i = 1$ if the action assigned to candidate distribution $\mathbf{v}_s$ is optimal under contract $\mathbf{r}^i$, and $\phi_{as}^i = 1$ if the candidate distribution $\mathbf{v}_s$ is assigned to distribution $\boldsymbol{\pi}_a$ *and* action a is optimal under $\mathbf{r}^i$ and $\boldsymbol{\pi}$.

Similar to PA-C, let $\mathbf{z} \in \mathbb{R}_+^{d \times S}$ be auxiliary variables used to linearize the absolute values in the loss function $Z_n(\boldsymbol{\pi})$. Then the restricted estimator (12) is equivalent to the following mixed-integer linear program:

$$\underset{\boldsymbol{\pi},\mathbf{w},\mathbf{x},\mathbf{z},\phi}{\text{minimize}} \quad \sum_{s\in S}\sum_{j\in J} z_{sj} \tag{13a}$$

$$\text{subject to} \quad z_{sj} \geq \frac{1}{n}\sum_{i\in I}(y_j^i - v_{sj})x_s^i, \quad s \in S, j \in J, \tag{13b}$$

$$z_{sj} \geq \frac{1}{n}\sum_{i\in I}(v_{sj} - y_j^i)x_s^i, \quad s \in S, j \in J, \tag{13c}$$

$$\sum_{b\in A}\sum_{s\in S}\left(\sum_{j\in J} v_{sj}r_j^i - c_b\right)\phi_{bs}^i \geq \left(\sum_{j\in J} v_{s'j}r_j^i - c_a\right)w_{at}, \quad i \in I, a \in A, s' \in S, \tag{13d}$$

(PA-D)

$$\sum_{s\in S} w_{as} = 1, \quad a \in A, \tag{13e}$$

$$\sum_{a\in A}\sum_{s\in S}\phi_{as}^i = 1, \quad i \in I, \tag{13f}$$

$$x_s^i = \sum_{a\in A}\phi_{as}^i, \quad i \in I, s \in S, \tag{13g}$$

$$\phi_{as}^i \leq w_{as}, \quad i \in I, a \in A, s \in S, \tag{13h}$$

$$\pi_{aj} = \sum_{s\in S} w_{as}v_{sj}, \quad a \in A, j \in J, \tag{13i}$$

$$x_s^i \in \{0,1\}, \quad i \in I, s \in S, \tag{13j}$$

$$w_{as} \in \{0,1\}, \quad a \in A, s \in S, \tag{13k}$$

$$\phi_{as}^i \in \{0,1\}, \quad i \in I, a \in A, s \in S, \tag{13l}$$

$$\boldsymbol{\pi} \in Q_\pi. \tag{13m}$$

Objective (13a) and Constraints (13b)–(13c) together represent the loss function $Z_n(\boldsymbol{\pi})$. Constraint (13d) enforces the agent's optimality conditions by ensuring that $\phi_{as}^i = 1$ only if candidate distribution $\mathbf{v}_s$ is mapped to $\boldsymbol{\pi}_a$ and if action a is optimal for the agent under $\mathbf{r}^i$ and $\boldsymbol{\pi}$. Constraint (13e) forces exactly one candidate distribution in V to be mapped to each distribution $\boldsymbol{\pi}_a$. Constraint (13f) ensures that only one candidate distribution in V and action $a \in A$ is selected for contract $\mathbf{r}^i$. Constraint (13g) forces $x_s^i = 1$ if candidate distribution $\mathbf{v}_s$ is mapped to $\boldsymbol{\pi}_a$ and if action a is optimal under $\mathbf{r}^i$ and $\boldsymbol{\pi}$. Constraint (13h) ensures $\phi_{as}^i = 1$ only if $\mathbf{v}_s$ is mapped to $\boldsymbol{\pi}_a$. Constraint (13i) defines $\boldsymbol{\pi}_a$ as the candidate distribution from V that is assigned by $\mathbf{w}$, and Constraint (13m) represents additional shape constraints imposed by the polyhedron Q_π. The key distinction between PA-D and PA-C is that the discrete nature of the parameter set allows the key decision variables $(\mathbf{w}, \mathbf{x}, \phi)$ to be binary, which allows us to represent the agent's optimality conditions in a way that circumvents the need for product terms.

Note that $Z_n(\tilde{\boldsymbol{\pi}}_n) - Z_n(\boldsymbol{\pi}_n^*)$ represents the error in the loss function that arises from solving the restricted estimator PA-D instead of the exact estimator PA-C. Next, we present a random clustering procedure for constructing the set of candidate distributions V, and provide a finite-sample characterization of the error $Z_n(\tilde{\boldsymbol{\pi}}_n) - Z_n(\boldsymbol{\pi}_n^*)$ under the proposed procedure.

4.2. Construction of Candidate Distributions and Finite-Sample Error

Because PA-C is a consistent estimator of $\boldsymbol{\pi}^0$ (by Proposition 1), we might expect PA-D to also produce a reasonable estimate of $\boldsymbol{\pi}^0$ if the loss function error $Z_n(\tilde{\boldsymbol{\pi}}_n) - Z_n(\boldsymbol{\pi}_n^*)$ is sufficiently small. Additionally, $Z_n(\tilde{\boldsymbol{\pi}}_n)$ is the minimal loss when the restricted parameter set $\tilde{\Pi}$ is substituted for Π. As a result, the magnitude of the gap $Z_n(\tilde{\boldsymbol{\pi}}_n) - Z_n(\boldsymbol{\pi}_n^*)$ depends on the restricted parameter set $\tilde{\Pi}$, and by extension, the set of candidate distributions V. Here, we present a method for constructing $\tilde{\Pi}$, based on leveraging the observed data $(\mathbf{r}^i, \xi^i)$, $i \in I$ to guide the construction of V. Our approach to constructing the candidate distributions V is summarized in Algorithm 1.

Algorithm 1 (Sample-Based Construction of Candidate Distributions)

Input: Data $(\mathbf{r}^i, \xi^i), i \in I$, parameter $\rho > 0$.

1. Randomly sample a subset S from I.
2. **for** each $s \in S$:
   $B_s = \{\mathbf{r} \in R \,|\, ||\mathbf{r}^s - \mathbf{r}||_2 \leq \rho\}$,
   $I_s = \{i \in I \,|\, \mathbf{r}^i \in B_s\}$.
   for each $j \in J$:
   $v_{sj} = \frac{1}{n_s}\sum_{i\in I_s} y_j^i$.

Output: Candidate distributions $V = \{\mathbf{v}_s \text{ for } s \in S\}$.

Algorithm 1 involves selecting subsets of the contract data, computing the empirical mass function over outcomes for each subset, and designating each of these empirical mass functions as a candidate distribution, $\mathbf{v}_s$. The sth candidate distribution is based on the outcomes of all contracts $\mathbf{r}^i$ that fall within a ball $B_s \subseteq R$; accordingly, we shall refer to the collection of data points indexed by I_s as the sth *cluster*. The intuition for constructing the candidate distributions V in this manner is simple: Based on the agent model (1), contracts that are within a small distance of each other are likely to induce the same action from the agent. Therefore, the empirical distribution of outcomes for all contracts that lie within the ball B_s can be assumed to approximate one of the rows of the true parameter matrix $\boldsymbol{\pi}^0$ (although which row it approximates remains unknown).

Next, we show that the error $Z_n(\tilde{\boldsymbol{\pi}}_n) - Z_n(\boldsymbol{\pi}_n^*)$ is well behaved if V is constructed using Algorithm 1. We first impose the following assumption.

Assumption 3 (Clustering Condition). *For each $a \in A$, there exists $s \in S$ such that $B_s \subseteq R_a(\boldsymbol{\pi}^0)$ and $I_s \neq \emptyset$.*

Assumption 3 states that for every action a, Algorithm 1 produces a ball B_s that is entirely inside the

subset of the contract set R that induces action a from the agent, $R_a(\boldsymbol{\pi}^0)$. If $B_s \subseteq R_a(\boldsymbol{\pi}^0)$, then every contract in cluster s induces action a from the agent. This implies that $\mathbf{v}_s$ is an empirical distribution sampled from $\boldsymbol{\pi}^0_a$. Therefore, Assumption 3 implies that for each row of $\boldsymbol{\pi}^0$, there exists at least one candidate distribution in V that is constructed by sampling from that row. Assumption 3 is more likely to hold when S in Algorithm 1 is large (because we construct many balls B_s) and ρ is small (because each ball is smaller).

Our next result shows that if Assumption 3 and an additional condition on Π holds, we can bound the approximation error $Z_n(\tilde{\boldsymbol{\pi}}_n) - Z_n(\boldsymbol{\pi}^*_n)$.

Theorem 2. *Let Assumption 3 hold, and let V be constructed using Algorithm 1. Furthermore, suppose $\Pi = \{\boldsymbol{\pi} \geq 0 | \sum_{j\in J} \pi_{aj} = 1, a \in A\}$. Then there exists $\kappa \in (0,1)$ such that for any $\varepsilon \in (0,1)$,*

$$\Pr(|Z_n(\boldsymbol{\pi}^*_n) - Z_n(\tilde{\boldsymbol{\pi}}_n)| > \varepsilon) \leq O(n^2\kappa^n). \tag{14}$$

We offer a few remarks on Theorem 2. First, observe that the bound is not monotonic because of the n^2 term, which implies that the bound can become looser in n for small n. This occurs because our proof approach depends on constructing a feasible solution $\bar{\boldsymbol{\pi}}$, and bounding the absolute number of observations where the hidden agent action is "misclassified" by $\bar{\boldsymbol{\pi}}$. Thus, the n^2 term reflects the possibility that the number of misclassified actions may increase with the sample size. Furthermore, note that if κ is close to one and n is of moderate size, the bound in Theorem 2 may be vacuous. However, because it is guaranteed that $\kappa \in (0,1)$, it is straightforward to verify that $n^2\kappa^n \to 0$, which implies that the error $Z_n(\tilde{\boldsymbol{\pi}}_n) - Z_n(\boldsymbol{\pi}^*_n)$ eventually vanishes in n.

Second, the rate depends on the constant κ, with lower values of κ leading to faster convergence. Although κ is not particularly interpretable, it can be shown to decrease in ρ and increase in the number of clusters $|S|$. Note from Algorithm 1 that ρ is the radius of the ball B_s, for each cluster $s \in S$. Intuitively, for fixed n, larger values of ρ makes each ball B_s contain a larger number of observations, which leads to faster convergence. Conversely, larger values of $|S|$ will slow convergence, for the following reason: Because the bound depends in part on the cluster that has the fewest observations, large values of $|S|$ will increase the probability that at least one of the clusters has very few data points, which weakens the bound. Therefore, the rate $n^2\kappa^n$ is fastest when ρ is large and $|S|$ is small. However, Assumption 3 is more likely to hold in the opposite case: when ρ is small and $|S|$ is large. Therefore, selecting ρ and $|S|$ requires balancing their effects on κ with ensuring that Assumption 3 holds.

Third, observe that the bound expression is invariant to ε provided $\varepsilon \in (0,1)$. Intuitively, this occurs because the key object of interest in the proof is a sequence of Bernoulli variables (which contribute to the loss function error in a binary manner) that we use to bound the number of times the hidden action is misclassified by a constructed solution $\bar{\boldsymbol{\pi}}$. However, we note that ε does indeed appear in the nondominant terms of the bound, as we would expect (see EC.37 in the proof of Theorem 2).

Theorem 2 is only valid for the case where each $\boldsymbol{\pi}_a$ is permitted to be any valid probability vector (i.e., $Q_\pi = \mathbb{R}^{m\times d}$). This additional condition is imposed on Π because the randomness of the set V can render the solution constructed by our proof approach infeasible for a more general parameter set Π. However, this additional assumption on Π is only needed for the finite-sample characterization of the error in Theorem 2; Proposition 2 shows that the solution from the restricted estimator, $\tilde{\boldsymbol{\pi}}_n$, is asymptotically optimal with respect to the exact estimator PA-C for any Π that satisfies (6).

Proposition 2. *Let Assumption 3 hold. Then* PA-D *is asymptotically optimal with respect to* PA-C*: $|Z_n(\boldsymbol{\pi}^*_n) - Z_n(\tilde{\boldsymbol{\pi}}_n)| \to 0$.*

The asymptotic optimality established in Proposition 2 provides assurance that PA-D is a reasonable approximation to PA-C when n is large, which is precisely the regime where PA-C is likely to be intractable. As a consequence, we should also expect the restricted estimator to produce "good" estimates of $\boldsymbol{\pi}^0$ for larger sample sizes. Having established that PA-D reasonably approximates PA-C, we now focus on developing a solution technique for tackling the mixed-integer program PA-D.

4.3. Statistical Column Generation

Observe that the size of the optimization problem PA-D grows with the number of candidate distributions in V, which can make PA-D computationally intractable if V is large. In this section, we propose a solution algorithm that involves solving PA-D over a subset of V, which we shall call V^+, which dramatically improves the tractability of the estimator PA-D, with minimal degradation in estimation error. Because each candidate distribution in V is mapped to a set of decision variables in PA-D (where the set S indexes the distributions in V), our solution technique can be interpreted as a column generation algorithm.

The key step of our approach is a series of nonparametric hypothesis tests, which identifies a subset V^+ by performing pairwise comparisons of candidate distributions in V. The intuition is as follows. Consider any candidate distribution $\mathbf{v}_s \in V$, and recall from Algorithm 1 that $\mathbf{v}_s$ is the empirical mass function over outcomes associated with the contracts in the sth cluster. If there exists another cluster s' such that all

contracts in clusters s and s' induce the same action from the agent, then $\mathbf{v}_s$ and $\mathbf{v}_{s'}$ can be interpreted as two empirical mass functions that were generated by the same probability distribution (i.e., one of the rows of $\boldsymbol{\pi}^0$). Therefore, our goal will be to apply nonparametric hypothesis tests to identify whether any pairs in V are generated by the same distribution, and to discard those that are effectively "duplicates."

4.3.1. Hypothesis Test Function. A hypothesis test typically consists of four main steps: (1) a null hypothesis is specified that we wish to test, (2) a significance level α (i.e., type I error rate) is specified for the test, (3) a test statistic is computed based on the sample data, and (4) the null hypothesis is rejected if and only if the magnitude of the test statistic exceeds a threshold τ_α, where τ_α depends on α. In the context of our column generation algorithm, the null hypothesis we will test is whether two candidate distributions $\mathbf{v}_s$ and $\mathbf{v}_{s'}$ are generated from the same probability distribution (i.e., the same $\boldsymbol{\pi}^0_a$), for many pairs (s, s').

We first introduce some additional definitions that are required by our algorithm. For each $s \in S$, define a vector $\boldsymbol{\psi}_s \in \mathbb{Z}^d_+$, where the jth entry is the frequency of outcome j in the sth cluster of Algorithm 1. The vector $\boldsymbol{\psi}_s$ is simply a convenient form for representing the candidate distribution $\mathbf{v}_s$ within our hypothesis tests. Let $n_s = |I_s|$ be the number of observations in cluster s, and note $n_s = \sum_{j\in J}\psi_{sj}$ for $s \in S$. Next, for each $s \in S$, by the weak law of large numbers, there exists $\boldsymbol{\nu}_s \in \mathbb{R}^d_+$ such that $||\boldsymbol{\nu}_s - \mathbf{v}_s|| \to 0$ as $n_s \to \infty$. We now define the main ingredient of the algorithm, which is a *test function* that declares whether $\boldsymbol{\psi}_s$ and $\boldsymbol{\psi}_{s'}$ are statistically different at a significance level α.

Definition 1. The function $H_\alpha(\boldsymbol{\psi}_s, \boldsymbol{\psi}_{s'}) : \mathbb{Z}^d_+ \times \mathbb{Z}^d_+ \mapsto \mathbb{R}$ is a test function if $\Pr(H_\alpha(\boldsymbol{\psi}_s, \boldsymbol{\psi}_{s'}) > 0 | \boldsymbol{\nu}_s \neq \boldsymbol{\nu}_{s'}) \to 1$ as $n_s \to \infty$ and $n_{s'} \to \infty$ and $\Pr(H_\alpha(\boldsymbol{\psi}_s, \boldsymbol{\psi}_{s'}) > 0 | \boldsymbol{\nu}_s = \boldsymbol{\nu}_{s'}) \leq \alpha$.

Definition 1 states that the hypothesis test function returns a positive value if and only if the null hypothesis, that the candidate distributions $\mathbf{v}_s$ and $\mathbf{v}_{s'}$ are generated by the same probability distribution, is rejected. This definition subsumes many two-sample, nonparametric hypothesis tests. One example is the Kolmogorov–Smirnov hypothesis test (Massey 1951, Stephens 1974), which is widely used for its ease of implementation. In particular, the test function is given by

$$H_\alpha(\boldsymbol{\psi}_s, \boldsymbol{\psi}_{s'}) = \sup_{j\in J}\left|\frac{\psi_{sj}}{n_s} - \frac{\psi_{s'j}}{n_{s'}}\right| - \tau_\alpha\sqrt{\frac{n_s + n_{s'}}{n_s n_{s'}}},$$

where τ^{KS}_α is the critical value associated with a significance level of α (Smirnov 1948). The Kolmogorov-Smirnov test is known to be conservative for discrete distributions (Slakter 1965, Conover 1972). As a result, selecting τ_α based on Kolmogorov–Smirnov critical values for continuous distributions makes α an upper bound on the true type I error rate in our setting but otherwise does not affect the validity of our algorithm. Other examples of nonparametric tests that fit within our framework are the Anderson-Darling (Anderson and Darling 1952, Scholz and Stephens 1987), chi-squared (Cochran 1952), and the Cramér–von Mises (Anderson 1962) tests.

4.3.2. Algorithm Overview. Let S^+ index the candidate distributions in V^+. We let PA-D (S^+) denote formulation PA-D where S is replaced with the subset S^+, and we let PA-D (S) denote the original formulation with the full set V. Let $V^- = V \setminus V^+$ and $S^- = S \setminus S^+$ denote the omitted distributions and the accompanying index set, respectively. Given a significance level α, we shall say two candidate distributions $\mathbf{v}_s$ and $\mathbf{v}_{s'}$ are *statistically different* if and only if $H_\alpha(\boldsymbol{\psi}_s, \boldsymbol{\psi}_{s'}) > 0$; that is, the null hypothesis that $\mathbf{v}_s$ and $\mathbf{v}_{s'}$ were generated from a common probability distribution is rejected. In each iteration of the main loop of the algorithm, we perform a series of hypothesis tests identify a new candidate distribution to be introduced to V^+, and solve PA-D (S^+) once there does not exist any distribution in V^- that is statistically different from every distribution in V^+ at a significance level of α. Specifically, in each iteration we compute

$$s^* = \operatorname*{argmax}_{s\in S^-} \inf_{s'\in S^+} H_\alpha(\boldsymbol{\psi}_s, \boldsymbol{\psi}_{s'}).$$

Intuitively, $\mathbf{v}_{s^*}$ is the distribution in V^- that is the "most" different from all distributions in V^+, based on the selected test function H_α. The distribution $\mathbf{v}_{s^*}$ is then added to V^+ if and only if

$$\inf_{s'\in S^+} H_\alpha(\boldsymbol{\psi}_{s^*}, \boldsymbol{\psi}_{s'}) > 0. \tag{15}$$

If (15) holds, then $\mathbf{v}_{s^*}$ is statistically different from every distribution in V^+, and is thus added to V^+. If (15) does not hold, then there are no remaining distributions in V^- that are statistically different from all distributions in V^+. In this case, we solve PA-D(S^+), and the algorithm terminates. A summary is given in Algorithm 2.

Algorithm 2 (Statistical Column Generation (PA-D+))

Input: Data $(\mathbf{r}^i, \xi^i)$, $i \in I$, candidate distributions V produced by Algorithm 1, significance level $\alpha > 0$.

Initialize: Set $t = 0$. Select any $s \in S$. Set $S^+ = \{s\}$ and $S^- = S \setminus \{s\}$.

1. Let $s^* = \operatorname{argmax}_{s\in S^-} \inf_{s'\in S^+} H_\alpha(\boldsymbol{\psi}_s, \boldsymbol{\psi}_{s'})$.
   if $\inf_{s'\in S^+} H_\alpha(\boldsymbol{\psi}_{s^*}, \boldsymbol{\psi}_{s'}) \leq 0$ or $S^- = \emptyset$,
   Solve PA-D(S^+) and obtain solution $\boldsymbol{\pi}^+_n$, set $T = t$, and **terminate**.
   else Update $t \leftarrow t + 1$, $S^+ \leftarrow \{S^+, s^*\}$, and $S^- \leftarrow S^- \setminus \{s^*\}$. Return to Step 1.

Output: Parameter estimate $\boldsymbol{\pi}^+_n$, iteration count T.

We will use "PA-D+" to denote the estimator represented by Algorithm 2. There are two main differences between existing column generation methods for large-scale integer programs and the one we propose in Algorithm 2. First, the column generation process in Algorithm 2 involves performing several hypothesis tests, which are fast to compute. By comparison, existing methods for integer programs typically generate columns by solving an auxiliary optimization problem (often called the *pricing* problem because of its use of dual information), which is often an integer program itself and may be difficult to solve (Lubbecke and Desrosiers 2005). Second, Algorithm 2 is not guaranteed to produce an optimal solution to PA-D; in contrast, the purpose of existing column generation methods is to solve the "original" optimization problem exactly. Therefore, Algorithm 2 effectively sacrifices optimality for computational efficiency. However, although Algorithm 2 does not produce optimal solutions to PA-D, it can be shown to produce a consistent estimate of $\boldsymbol{\pi}^0$, which is our main objective in this paper.

4.3.3. Consistency and Iteration Bound. Next, we present the main result of Section 4: Theorem 3 shows that the approximate solution obtained by Algorithm 2 preserves the consistency of the exact estimator PA-C.

Theorem 3. *Let $\boldsymbol{\pi}_n^+$ be the estimate obtained by* PA-D+ *(Algorithm 2). Then*

$$\boldsymbol{\pi}_n^+ \to \boldsymbol{\pi}^0.$$

As a consequence of Theorem 3, we should expect $\boldsymbol{\pi}_n^+$ to provide a reasonable estimate of the unknown parameter $\boldsymbol{\pi}^0$. However, Theorem 3 is an asymptotic result only and that for small n the estimate from PA-D+ may be less accurate than the exact estimate obtained by solving PA-C. We compare the performance of these two approaches numerically in Section 4.4.

Because the termination condition in Algorithm 2 depends on the outcome of a series of hypothesis tests, the total number of iterations, denoted by T, is a random variable. In Theorem 4, we show that $\mathbb{E}[T]$ is bounded by a function of the problem parameters, including the significance level α used in the hypothesis testing step of Algorithm 2.

Theorem 4. *Let Assumption 3 hold. Furthermore, assume that for each $s \in S$, there exists $a \in A$ such that $B_s \subseteq R_a(\boldsymbol{\pi}^0)$. Then*

$$\mathbb{E}[T] \le m[1 + \alpha \cdot |S| \cdot (|S| - m)].$$

The proof of Theorem 4 relies on upper bounding $\Pr(T > m)$: the probability that the number of iterations in Algorithm 2 exceeds the number of agent actions. In particular, we show in the proof of Theorem 4 that $\Pr(T > m) \le \alpha m S$. The intuition for the preceding inequality is as follows. Observe that by construction, the candidate distribution v_s is the empirical distribution over outcomes associated with all contracts $\mathbf{r}^i$ such that $\mathbf{r}^i \in B_s$. Because for each $s \in S$, $B_s \subset R_a(\boldsymbol{\pi}^0)$ for some $a \in A$ (by assumption), there are at most m unique distributions from which the empirical distributions v_s are generated, which are $\boldsymbol{\pi}_a^0$, $a \in A$. Next, in Algorithm 2, a candidate distribution is only added to the set V^+ if the hypothesis testing step finds it to be statistically different from every distribution in V^+. Therefore, the event $\{T > m\}$ implies that a type I error has occurred at some point during Algorithm 2; that is, a candidate distribution was added to $V+$ despite the underlying distribution $\boldsymbol{\pi}_a^0$ already being "represented" in $V+$ by another candidate distribution.

Because α bounds the probability of making a Type I error, smaller values of α will make Algorithm 2 more *conservative* in adding new distributions to V^+, thus increasing the probability of the event $\{T > m\}$. Conversely, if α is large, then it becomes more likely that a given distribution v_s is determined to be statistically different from those in V^+, which leads to more distributions being added to V^+ and thus a greater number of iterations. The dependence on S arises for a similar reason; as S increases, so does the number of omitted distributions V^-, which increases the likelihood that there exists a distribution in V^- that satisfies the inclusion criterion in Step 2 of Algorithm 2.

Additionally, the bound $\mathbb{E}[T] \le |S|$ holds trivially, because $T = |S|$ implies $S_T^- = \emptyset$ by Algorithm 2. As a result, the bound in Theorem 4 may be vacuous if α is large but is made meaningful for an appropriate selection of S and α. It is also straightforward to verify that the assumption in the statement of Theorem 4 implies that $|S| \ge m$, which confirms that the bound on $\mathbb{E}[T]$ is strictly positive for all $\alpha > 0$.

4.4. Numerical Performance

In this section, we compare the performance of three estimation methods using synthetic data. The first two are solving the exact estimator (PA-C) and the restricted estimator (PA-D) directly with optimization software. The third is solving the restricted estimator using the column generation technique outlined in Algorithm 2 (PA-D+). We focus our comparison on the solution times and estimation errors from the three approaches.

4.4.1. Setup. Recall that m and d denote the number of actions and outcomes, respectively. We consider five problem sizes, given by $(m,d) \in \{(2,2),(4,5),(5,10),(10,20),(20,40)\}$. For each of the five problem sizes, we consider three sample sizes, given by n

$\in \{100, 500, 1{,}000\}$. Then for each combination (m, d, n), we randomly generate $\boldsymbol{\pi}^0$ from the appropriately sized parameter set Π given by (6), where Q_π is given by (7). For each (m, d, n), we randomly generate contract data by sampling $\mathbf{r}^i$ uniformly from $[1,10]^d$ for each $i = 1, \ldots, n$, and sampling $\mathbf{c}$ uniformly from $[0,1]^m$. The outcome associated with each $\mathbf{r}^i$ is obtained by solving the agent's problem (1) under the corresponding $\boldsymbol{\pi}^0$. We repeat this procedure for a total of 10 trials for each (m, d, n). To parameterize PA-D, we set $S = 50$ and $\rho = 10 \times d$. For the hypothesis testing step for PA-D+, we use the discrete analogue of the two-sample Anderson–Darling test (Scholz and Stephens 1987), and set $S = 50$, $\rho = 10 \times d$, and $\alpha = 0.05$. We use the optimization solver Gurobi 8.0 to solve PA-C, PA-D, and PA-D+.

4.4.2. Results. Table 1 summarizes the average solution time and estimation errors over 10 trials for the three estimators. In each trial, the error associated with PA-C, PA-D, and PA-D+ is given by $\frac{1}{md}\|\boldsymbol{\pi}^0 - \boldsymbol{\pi}_n^*\|$, $\frac{1}{md}\|\boldsymbol{\pi}^0 - \tilde{\boldsymbol{\pi}}_n\|$, and $\frac{1}{md}\|\boldsymbol{\pi}^0 - \boldsymbol{\pi}_n^+\|$, respectively. In all trials, we set a time limit of 3,600 CPU seconds. Dashes in the table indicate instances where an optimal solution was not found within 3,600 CPU seconds for any of the 10 trials. In many of these trials, no feasible solution was found within 3,600 CPU seconds; we therefore only include errors obtained at optimal solutions to PA-C or PA-D when reporting the average estimation error.

We offer a few observations regarding Table 1. First, for each problem size, the estimation error generally decreases in n, which corroborates our consistency results (Proposition 1 and Theorem 3, respectively). Second, for smaller problem instances (e.g., $m = 4$, $d = 5$, $n = 1{,}000$), PA-C is less computationally expensive than PA-D+, which we posit is a consequence of requiring fewer binary decision variables. However, PA-D+ generally scales more efficiently in the problem and sample size than PA-C and PA-D, with the most notable performance improvement occurring at larger problem instances (e.g., $m = 10$, $d = 20$, $n = 1{,}000$). Third, solving the restricted estimator PA-D directly with Gurobi is less tractable than solving the exact estimator PA-C with Gurobi. This is again likely attributable to PA-D requiring many more binary variables than PA-C, because of how the restricted parameter set is represented in the formulation PA-D. Nonetheless, the results indicate that this intractability can be overcome by (approximately) solving the restricted estimator using the statistical column generation technique, without significantly compromising estimation error. Fourth, observe that larger problem sizes are not necessarily more computationally expensive; for example, the average solution time of the instances (2, 2, 1,000) and (5, 10, 1,000) for PA-C was 245 and 12 seconds, respectively. We conjecture that this is because the larger problem sizes offer the estimator additional degrees of freedom in fitting the agent model to the data (because of containing a larger number of unknown parameters), which allows the optimization problem to more quickly attain the minimal objective function value. Last, the favorable performance of PA-D+ in the larger instances (e.g., $m = 20$, $d = 40$) suggests that our estimator and algorithm can also be used to tractably approximate contracting problems with continuous actions and outcomes through discretization.

Table 1. Solution Time (CPU Seconds) and Normalized Estimation Error of Three Formulations Averaged over 10 Trials

			PA-C		PA-D		PA-D+	
m	d	n	Time	Error	Time	Error	Time	Error
2	2	100	2	0.07	20	0.06	2	0.09
2	2	500	19	0.06	3,432	0.06	4	0.06
2	2	1000	245	0.06	—	—	15	0.06
4	5	100	0	0.05	—	—	4	0.09
4	5	500	2	0.05	—	—	18	0.06
4	5	1000	3	0.05	—	—	66	0.06
5	10	100	1	0.04	—	—	4	0.06
5	10	500	6	0.03	—	—	14	0.04
5	10	1000	12	0.03	—	—	47	0.03
10	20	100	2,404	0.02	—	—	3	0.02
10	20	500	—	—	—	—	15	0.01
10	20	1000	—	—	—	—	26	0.01
20	40	100	—	—	—	—	2	0.01
20	40	500	—	—	—	—	84	0.01
20	40	1000	—	—	—	—	211	0.01

Notes. Instances that did not solve to optimality under 3,600 CPU seconds are omitted when calculating average estimation error. Dashes indicate no instance solved to optimality within 3,600 CPU seconds in any trial.

The purpose of Algorithm 2 is not to generate a provably optimal solution to PA-D, which is typically the case with similar column generation methods. Instead, our primary goal is to generate an estimate of the true parameter $\boldsymbol{\pi}^0$ that is statistically consistent, competitive with solutions from solving the exact estimator, and attainable in a computationally efficient manner. Theorem 3 and the numerical results in Table 1 suggest that Algorithm 2 meets each of these criteria.

5. Empirical Study: Randomizing Incentives in a Crowdwork Platform

In this section, we demonstrate our method by using it to investigate the effect of financial incentives on work quality in an online labor platform. First, we conducted an experiment on a crowdwork platform (Amazon Mechanical Turk) by randomly assigning incentive contracts to a pool of workers completing the same task. We then estimate an agent model from the experimental data, which allows us to characterize

the link between incentives and quality and solve for an optimal incentive contract.

5.1. Background: Incentives and Quality on Amazon Mechanical Turk

Crowdwork platforms are used by businesses that require temporary labor to complete tasks that are typically difficult for computers but simple for humans. Common tasks include audio transcription, classification of images, and data entry. The largest and most well-known crowdwork platform is Amazon's Mechanical Turk ("mTurk"), which has been estimated to have 100,000 unique workers, with 2,000 active at any given time (Difallah et al. 2018).

The mTurk platform allows "requesters" to post tasks, along with a reward to be paid to the worker upon successful completion. Workers can select the tasks they want to complete, typically on a first-come, first-served basis. Requesters have discretion over whether to pay workers for their submissions and can deny payment if the worker's submission is incomplete or low quality. Requesters can also provide bonuses to workers. Workers can be informed of the structure of the bonus payment within the instructions for a task, which offers the requester considerable flexibility in designing incentives.

The question of whether financial incentives improve quality of work in crowdwork platforms has been addressed in multiple studies, with differing conclusions. Mason and Watts (2009) find that incentives improve the quantity, but not quality of work; similarly, Yin et al. (2013) find that the magnitude of the bonus does not affect quality. In contrast, Horton and Chilton (2010) and Harris (2011) both find that quality can improve with worker pay. An important study in this line of research is by Ho et al. (2015), who suggest that for tasks where quality plausibly depends on worker effort (e.g., proofreading), incentives can improve quality.

With respect to experimental design, we underline two differences between our study and the work cited above. First, instead of assigning workers to a finite number of treatments (e.g., bonus or no bonus), we vary incentives in a continuous manner, meaning the parameters of the incentive contract are randomly drawn *for each worker*. This design significantly complicates the implementation of the experiment on mTurk but introduces useful variation for estimating our agent model. Second, we examine how incentives affect the *distribution* of work quality instead of average quality.

5.2. Experimental Setup

5.2.1. Task Design. A major source of observable heterogeneity in the mTurk worker population is location. Approximately 91% of workers are located in two countries: the United States (75%) and India (16%) (Difallah et al. 2018). We collected and analyzed data from both countries separately.

The experiment involved posting two types of tasks on mTurk. First, we posted a recruitment task in which workers were paid $1.00 for agreeing to be notified of future tasks by email. We recruited 250 workers from both the United States and India using this task, for a total worker pool of 500. The recruitment task in each country was made available for one day and reached its maximum number of submissions (250 for each location) within 3 hours of posting. Second, inspired by Ho et al. (2015), we created a proofreading task by inserting 10 typos into a one-page, 500-word excerpt from a newspaper article. The proofreading task required workers to report the line number and correct spelling for each misspelled word in the article (e.g., "5:automobile"). We use a proofreading task because it allows us to objectively measure the quality of each submission (percentage of typos identified). After constructing the worker pool, we posted the proofreading task on mTurk and notified each worker by email of the task's availability. The task was available for 24 hours.

5.2.2. Incentive Structure and Randomization. We next describe how we randomized incentives among workers. The mTurk platform allows requesters to assign "qualification" criteria to tasks, which only allows workers with the required qualifications to view and complete the task. For example, a requester might assign a location or age-based qualification to a task if they wish to target a specific worker population. Requesters can also create and assign custom qualifications to workers. When conducting a randomized experiment, creating and randomly assigning qualifications to workers effectively allows the requester to construct multiple treatment groups, where each qualification represents one treatment.

We use the qualification feature in mTurk to create exogenous variation in worker incentives. We first created 500 unique qualifications and assigned each qualification to a single worker in the pool. We then created 500 tasks where each task was randomly assigned to a qualification. As a result, for each of the 500 tasks, only a single worker in our pool was able to view and complete it.

The payment for the proofreading task consisted of two components: a *base* payment for finding at least 25% of the typos in the document and an additional *bonus* payment for finding at least 75% of typos. For each task (i.e., each worker in the pool), we drew *base* and *bonus* uniformly from the interval [$0.10, $1.00], rounded to the nearest $0.01. We provided the details of the payment structure upfront in the task instructions. Because workers were only able to view the task

assigned to them, workers could not observe the payment offered to others and had no knowledge that payments were randomized. In the context of the proofreading task, worker output corresponds to the fraction of typos corrected, which we also refer to as the task *quality*.

5.2.3. Submissions. We collected a total of 346 submissions, each from a unique mTurk worker. Of these, 215 submissions were from U.S.-based workers and 131 were from India-based workers. We analyze the data from U.S. and India workers separately throughout our study. Figure 1 depicts the distribution of quality scores for workers in each location. A large number of submissions achieve a quality score of zero. Low-quality submissions are a well-known feature of mTurk; because verifying responses manually for a large number of submissions is difficult, workers may submit blank or low-quality responses in the hope of nevertheless receiving a payment (Ipeirotis et al. 2010). Scores of zero may also be because of submissions not being in the correct format, which we specified as a condition for payment in the task instructions.

The mTurk platform provides timestamps for when a worker accepted and submitted a task. The mean completion time (i.e., time between acceptance and submission) was 9.7 minutes, and 95% of workers submitted the task between 1 and 29 minutes after accepting it. Because mTurk allows workers to accept tasks into a queue before working on them, the recorded completion time is an upward-biased measurement of the actual time the worker spent on the task. As a result, completion time may be a poor proxy for true worker effort, because the requester cannot observe how much time the task spent in the worker's queue. We therefore treat effort as fully hidden and do not use completion time data in our study.

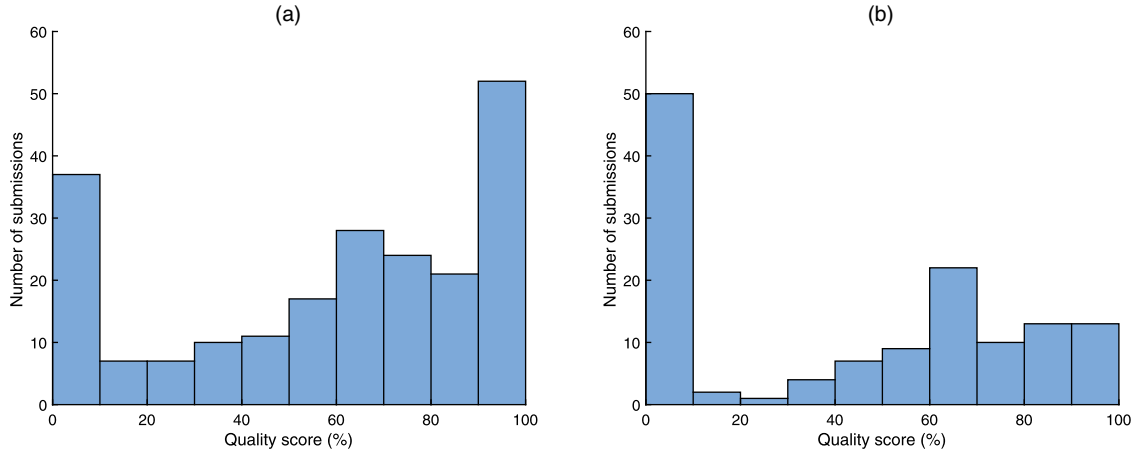
Based on each worker's completion time, we estimated the average wage to be \$14.50/hour for our task (including the guaranteed \$1.00 payment at recruitment). This is likely a conservative estimate of the true average wage because of the queueing behavior described previously.

5.3. Estimation and Validation

Next, we describe the application of our estimation procedure to the experimental data. Putting the results of the experiment in the format required by our estimator is straightforward. Recall that each worker was eligible for three possible payments based on their submission quality: no payment (if they found 0%–25% of typos), a base payment (25%–75%), or both a base and bonus payment (75%–100%). In our framework, this corresponds to $d = 3$ possible performance levels for the worker's outcome ξ^i. Accordingly, the ith worker's incentive contract $\mathbf{r}^i$ has the components $r_1^i = 0$, $r_2^i = base^i$, and $r_3^i = base^i + bonus^i$, where $base^i$ and $bonus^i$ are the randomly generated parameters for that worker. For the PA-D+ algorithm, we set $\rho = 0.5$, $S = 10$, and $\alpha = 0.0001$ throughout all experiments.

5.3.1. Measuring Goodness of Fit. We require a goodness of fit metric for fitting and validating the model. Recall that our estimation procedure generates a prediction of the outcome distribution: Given an estimate $\hat{\boldsymbol{\pi}}$, a contract $\mathbf{r}$, and action costs $\mathbf{c}$, the model's prediction of the outcome distribution under $\mathbf{r}$ is $\hat{\boldsymbol{\pi}}_{a(\mathbf{r})}$, where $a(\mathbf{r})$ is the agent's optimal action under contract $\mathbf{r}$. For ease of interpretation, we measure goodness of fit as the absolute error between the empirical and predicted probability of a given outcome, averaged over

Figure 1. (Color online) Distribution of Quality Scores for Submissions Made by Workers in the United States and India



Notes. (a) United States. (b) India.

all outcomes. Specifically, let $(\mathbf{r}^i, \xi^i)$, $i = 1, \ldots, n$ be the data we wish to measure our model fit against. As before, for each i, let $y^i_j = 1$ if $\xi^i = j$ (if outcome j is observed). Then the mean absolute error (MAE) is given by

$$\text{MAE} = \frac{1}{d}\sum_{j=1}^{d}\left|\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\pi}_{a(\mathbf{r}^i),j} - y^i_j\right)\right|. \tag{16}$$

5.3.2. Setting Cost Parameters. Two hyperparameters in our model are the number of actions, m, and the action costs, $c_1, \ldots, c_m$. We selected these parameters using a standard 10-fold cross-validation procedure, using MAE to measure cross-validation errors. To avoid performing an extremely large number of cross-validation iterations, we imposed additional structure by assuming action costs were of the form $c_a = (a - 1) \cdot \delta$, for $a = 1, \ldots, m$. We used cross-validation to jointly select m and δ from the sets {2, 3, 4} and $\{0.02, 0.05, 0.1, 0.2, 0.5\}$, respectively (units of the latter set are dollars). Results are presented in Table 2. Errors are relatively stable for all values of δ when $m = 2$ or $m = 3$, whereas the model appears to overfit for $m = 4$. We select $(m, \delta) = (3, 0.1)$ for both the U.S. and India data sets, resulting in the cost vector $\mathbf{c} = [0, 0.1, 0.2]$. Last, for Algorithms 1 and 2, we set $\rho = 0.5$, $S = 50$, and $\alpha = 0.01$, and use a chi-squared test for the hypothesis test step of Algorithm 2.

Our handling of action costs is fairly stylized, because they are treated as hyperparameters to be tuned prior to model fitting. Horton and Chilton (2010) estimate the median reservation wage of mTurk workers to be \$1.38/hour. Given that the median completion time for our task was 8 minutes, \$0.00–\$0.20 appears to be a reasonable approximation for the range of effort costs of an mTurk worker. We discuss costs in more detail in Section 5.5.

5.3.3. Bootstrapping. Given our moderately sized data set ($n = 215$ and $n = 131$), we validated our estimation procedure by bootstrapping. For each of 100 repetitions, we sampled n observations with replacement and estimated the model parameters from the sample using Algorithm 2. For each repetition, we assessed model fit using two hypothesis tests: a Chi-squared (χ^2) test, which is appropriate in our setting because outcomes are discrete, and an exact test using MAE as the test statistic, where the sampling distribution is obtained through Monte Carlo simulation. In both hypothesis tests, the null hypothesis is that the empirical distribution of quality outcomes in the out-of-bootstrap data are generated by the fitted model. Accordingly, we interpret large p values as indicating a good model fit.

Table 3 shows the distribution of test statistics and associated p values over the 100 bootstrap repetitions. Both tests produced comparable p values within each worker group. The median p value was above 0.1 for both groups, which suggests the model reasonably fits the joint distribution over $(\mathbf{r}, \xi)$ in the majority of bootstrap iterations.

Table 4 presents the estimated values of $\boldsymbol{\pi}$ and standard errors for both worker groups. Each 3×3 section in the center of Table 4 corresponds to the estimated $\boldsymbol{\pi}$ matrix for the labeled worker group, averaged over 100 bootstrap repetitions. For convenience, we refer to the outcome in which the worker earns the bonus ($\xi^i = 3$) as the "bonus outcome" and the probability that this outcome is realized as the "bonus probability." The highest cost action ($a = 3$) has the highest bonus probability in both worker groups, and the bonus probability is lower in the India worker group compared with the U.S. group for all actions.

Our estimation procedure treats each action a as a latent variable. The solution to the estimation problem produces a clustering where each outcome is assumed to have been generated by one of the m distributions (i.e., agent actions). As a result, for each bootstrap repetition, we can count the number of observations that are assigned to each action by the estimator. The average number of observations mapped to each action are reported in the final column of Table 4.

5.3.4. Predictive Performance. Next, we evaluate the predictive performance of the estimator. For each of the 100 bootstrap models, we compute the prediction error (given in (16)) attained by the fitted model on the out-of-bootstrap observations. We set $S = 10$, $\rho = 0.5$, and $\alpha = 0.0001$. To serve as performance benchmarks, we repeat the bootstrap procedure for standard implementations of multinomial logistic regression (MLR) and classification trees (CT), both of which also generate predictions of the outcome distribution for a given set of contracts.[7] Figure 2 depicts the distribution over prediction errors for the three methods over the 100 bootstrap repetitions. For the U.S. data, the average MAE for PA-D+, MLR, and CT is 0.059, 0.070, and 0.093, respectively; for the India

Table 2. Ten-Fold Cross-Validation Errors (MAE) for U.S. and India Groups, with Varying Number of Actions (m) and Cost Spacing (δ)

		δ				
	m	0.02	0.05	0.1	0.2	0.5
United States	2	0.06	0.06	0.07	0.06	0.06
	3	0.04	0.06	0.04	0.06	0.05
	4	0.18	0.11	0.12	0.12	0.08
India	2	0.07	0.08	0.06	0.06	0.08
	3	0.07	0.05	0.06	0.06	0.08
	4	0.15	0.08	0.06	0.11	0.09

Table 3. Percentiles of Chi-Squared and MAE Test Statistics with Associated p Values over 100 Bootstrap Repetitions

	Test Statistic	5th	25th	Median	75th	95th
United States	χ^2 (p value)	0.20 (0.90)	1.70 (0.43)	3.42 (0.18)	7.36 (0.03)	23.58 (0.00)
	MAE (p-value)	0.02 (0.89)	0.04 (0.42)	0.07 (0.14)	0.10 (0.02)	0.15 (0.00)
India	χ^2 (p value)	0.18 (0.91)	0.99 (0.61)	2.13 (0.34)	5.23 (0.07)	14.33 (0.00)
	MAE (p-value)	0.02 (0.93)	0.05 (0.65)	0.09 (0.35)	0.14 (0.09)	0.19 (0.00)

data, the average errors are 0.072, 0.086, and 0.112. In summary, Figure 2 confirms that the PA-D+ estimator produces sound predictions on the experimental mTurk data and is competitive with well-known benchmark methods. In Section EC.2 of the e-companion, we further compare all three methods on several synthetic instances and find that our estimator continues to perform well.

5.4. Impact of Bonuses on Quality

We now use the estimated model to examine the effect of varying the bonus payment on quality. For a given incentive contract, we form a prediction of the outcome distribution by averaging over the 100 bootstrapped models, which improves stability and reduces overfitting (Breiman 1996). Let $\hat{\boldsymbol{\pi}}^1, \ldots, \hat{\boldsymbol{\pi}}^K$ be the estimates obtained from K bootstrap repetitions. The probability of observing outcome j under the incentive contract $\mathbf{r}$ is then given by $\frac{1}{K}\sum_{k=1}^{K} \hat{\boldsymbol{\pi}}^k_{a^k(\mathbf{r}),j}$, where $a^k(\mathbf{r})$ is the optimal action for contract $\mathbf{r}$ in the kth agent model. To isolate the influence of the bonus payment, we fix the base payment to \$0.10, vary the bonus payment between \$0.10 and \$1.00, and compute the probability of each quality outcome under each bonus amount. We repeat for a base payment of \$1.00.

Figure 3 shows the results for both the U.S. and India groups of workers. For a base payment of \$0.10 (Figure 3, (a) and (b)), the bonus probability (i.e., probability that submission quality is above 75%) increases moderately for both groups as the bonus is increased from \$0.10 to 1.00 (from 0.21 to 0.36 for the U.S. group; from 0.09 to 0.17 for the India group). However, with a base payment of \$1.00 (Figure 3, (c) and (d)), the effect of increasing the bonus payment from \$0.10 to \$1.00 is dampened (bonus probability increases from 0.34 to 0.37 for the U.S. group; 0.18 to 0.22 for the India group). These results suggest that increasing the bonus payment can indeed increase quality, but the effect is significantly diminished when the base payment is already high. A qualitatively similar result can be obtained by fixing the bonus payment and varying the base payment (results not shown).

We shed some light on the mechanics behind Figure 3. Because our predictions are based on the average of 100 different agent models, for a fixed incentive contract, we can count the number of models in which each action is taken. Furthermore, if the bonus payment increases, an agent may find it optimal to "switch" from a low-cost action to a high-cost action, thus increasing the probability of realizing the high-quality outcome. The change in probabilities depicted in Figure 3 is the result of the underlying agent models jumping from one action to the next as the parameters of the contract change.

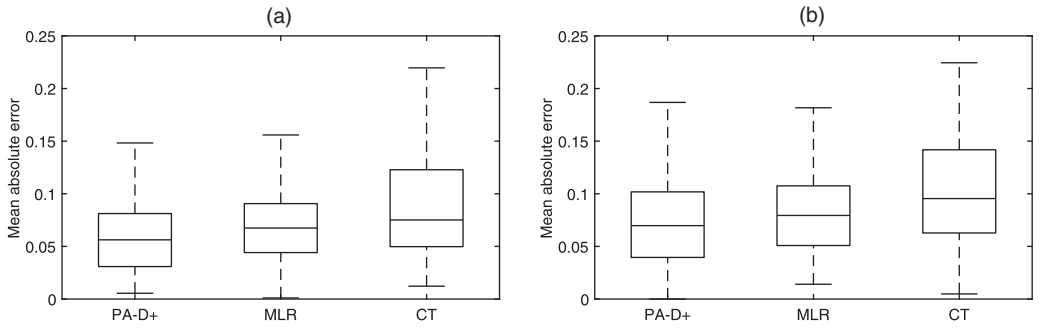
Figure 4 shows the fraction of agent models that take each of the three actions as the bonus is increased from \$0.10 to \$1.00. The four panels in Figure 4 map to the four panels in Figure 3. As expected, when the base payment is \$0.10, increasing the bonus amount from \$0.10 to \$1.00 is associated with agents switching away from the lowest cost action ($a=1$) toward the higher cost actions ($a=2$ and $a=3$). Moreover, the shift toward higher cost actions is more pronounced for the U.S. worker group, where the fraction of agents taking the highest cost action ($a=3$) increases from 0 to 0.69; for the India group, this fraction increases from 0 to 0.17. In parallel with Figure 3, when the base payment is \$1.00, the fraction of agents taking the highest cost action ($a=3$) is higher overall, but the shift toward higher cost actions as the bonus is increased is muted. In other words, the stability in selected actions shown in Figure 4, (c) and (d) explains the stability in outcome

Table 4. Estimated Values of $\boldsymbol{\pi}$ for Both Groups, with Standard Errors in Parentheses

		Outcomes (j)			
	Actions (a)	1	2	3	No. of observations
United States	1	0.46 (0.18)	0.34 (0.18)	0.20 (0.09)	19
	2	0.30 (0.10)	0.42 (0.12)	0.28 (0.08)	27
	3	0.20 (0.07)	0.43 (0.07)	0.37 (0.06)	169
India	1	0.57 (0.12)	0.34 (0.12)	0.09 (0.07)	36
	2	0.45 (0.09)	0.38 (0.11)	0.17 (0.06)	36
	3	0.35 (0.07)	0.42 (0.07)	0.23 (0.06)	59

Note. The final column reports the number of in-bootstrap observations mapped to each action, averaged over 100 bootstrap repetitions.

Figure 2. Comparison of Out-of-Bootstrap Prediction Errors for PA-D +, MLR, and CT on mTurk Data (100 Repetitions)



Notes. (a) U.S. data. (b) India data.

probabilities seen in Figure 3 (c) and (d). We emphasize here that Figure 4 is intended to illustrate the mechanics behind the predictions in Figure 3 and is not necessarily a depiction of worker behavior.

5.5. Solving for an Optimal Incentive Contract

An advantage of our model specification is that it leads to an optimal contracting problem that is highly tractable (see Section EC.1 of the e-companion for details). To illustrate this in the context of our mTurk study, we consider the simple problem of maximizing the bonus probability (i.e., outcome $\{\xi = 3\}$) subject to a budget constraint on the expected payment:

$$\underset{\mathbf{r}}{\text{maximize}}\ \hat{\pi}_{\hat{a}(\mathbf{r}),3} \tag{17a}$$

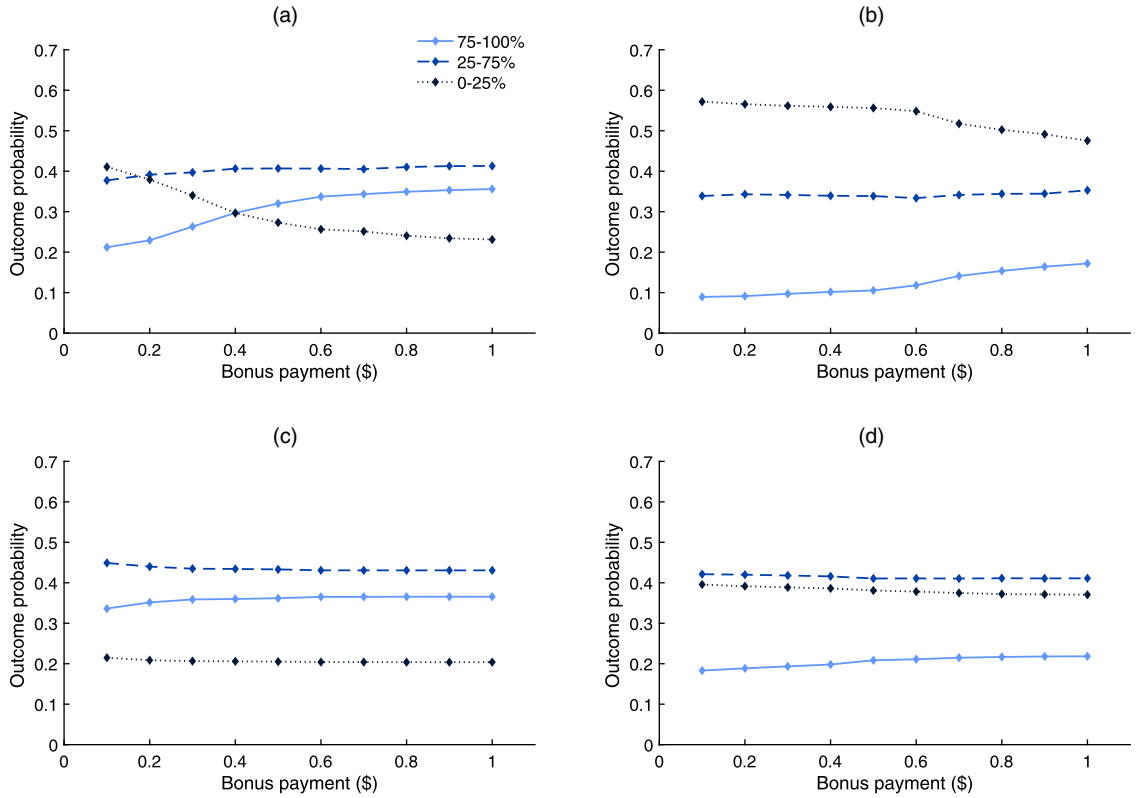
$$\text{subject to } \hat{a}(\mathbf{r}) = \underset{a\in A}{\text{argmax}}\ \mathbf{r}^\top \hat{\boldsymbol{\pi}}_a - c_a, \tag{17b}$$

$$\mathbf{r}^\top \hat{\boldsymbol{\pi}}_{\hat{a}(\mathbf{r})} \le \Gamma, \tag{17c}$$

$$\mathbf{r} \ge \mathbf{0}. \tag{17d}$$

This formulation is a special case of the general optimal contracting problem presented in Section EC.1 of

Figure 3. (Color online) Effect of Varying Bonus Payment on Probability of Each Quality Outcome (0%–25%, 25%–75%, 75%–100%) for U.S. and India Workers



Notes. (a) United States, base = \$0.10. (b) India, base = \$0.10. (c) United States, base = \$1.00. (d) India, base = \$1.00.

the e-companion and can be solved exactly by solving $|A|$ linear programs. An important consequence of the tractability of the optimal contracting problem (17) is that we can easily characterize the performance of the optimal contracts as the budget parameter Γ varies. To do so, we solve (17) for each $\Gamma \in \{0.05, 0.1, \ldots, 1\}$ (for each of the 100 bootstrap estimates) and compute the average bonus probability under each value of Γ.

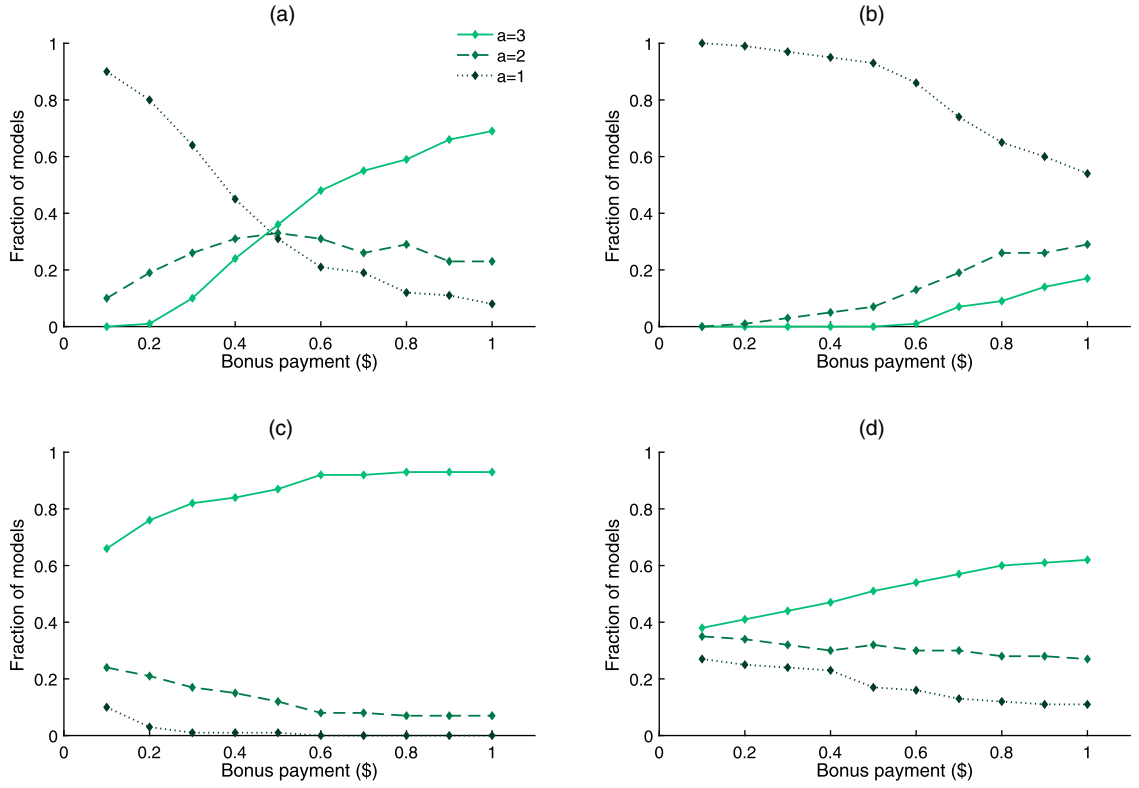
Figure 5 shows the resulting frontiers for both the U.S. and India worker data. Because the curves are obtained by solving the optimal contracting problem (17), they represent estimates of the maximum attainable performance for both worker groups over the entire class of contracts used in the experiment. The value of the budget parameter Γ can be interpreted as the expected payment to the agent under the corresponding optimal contract. Our main finding is that higher payments increase quality modestly: increasing the expected payment from \$0.10 to \$1.00 increases the bonus probability under the optimal contract by 0.08–0.12, depending on the worker group. However, the most striking observation is that returns to quality diminish at fairly low payment levels, with quality improvements leveling off around \$0.30 and \$0.60 for the U.S. and India groups, respectively (we discuss possible explanations in Section 5.7).

Figure 5 also clearly depicts the difference in the performance of optimal contracts between the U.S. and India worker groups. For example, for the U.S. group, attaining a bonus probability of 0.30 requires an expected payment of at least \$0.20; for the India group, a bonus probability of 0.30 is not attainable through higher payments alone. It can also be observed that the bonus probability is approximately 0.10–0.15 higher among U.S. workers across all payment levels.

5.6. Experimental Validation of Contract Performance

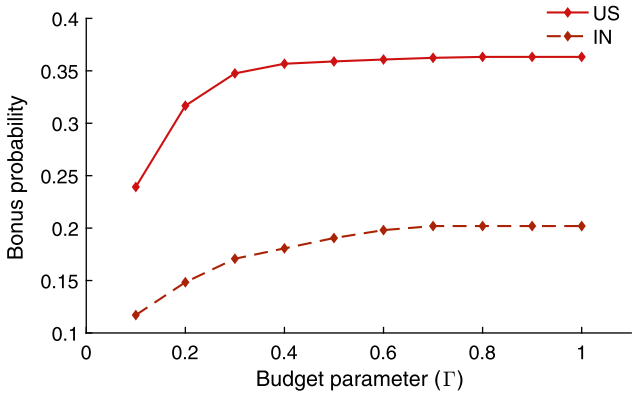
To validate the predicted performance of the optimal contracts shown in Figure 5, we conducted six follow-up experiments on mTurk. First, for each of the 100 bootstrap estimates $\hat{\boldsymbol{\pi}}^1, \ldots, \hat{\boldsymbol{\pi}}^K$, we solved the optimal contracting problem (17) for $\Gamma \in \{0.25, 0.50, 0.75\}$, which corresponds to three different points on the frontiers in Figure 5. We then computed the optimal contract by taking the component-wise average of the 100 solutions to (17). This produced six different testable contracts (i.e., combinations of the *base* and

Figure 4. (Color online) Effect of Bonus Payment on Optimal Agent Actions in 100 Bootstrapped Models



Notes. (a) United States, base = \$0.10. (b) India, base = \$0.10. (c) United States, base = \$1.00. (d) India, base = \$1.00.

Figure 5. (Color online) Frontier of Optimal Bonus Probabilities Under Varying Budget Parameter Γ



bonus parameters), which are shown in Table 5. We implemented each contract on mTurk by recruiting a new pool of 600 unique workers (using the same approach described in Section 5.2) and assigning 100 workers to each of the six contracts. Table 5 summarizes the results from these experiments, including the empirical bonus probability for each contract (i.e., the fraction of submissions with quality above 75%). In Figure 6, we plot the empirical bonus probabilities along with the 95% prediction intervals obtained from the bootstrap.

Figure 6 shows that for each of the six experimentally tested contracts, the empirical bonus probability sits comfortably inside its corresponding prediction interval and is often close to the midpoint of the interval. In general, the prediction intervals are wide, which is unsurprising given that many other factors likely influence submission quality beyond the payment amount, including unobserved worker attributes. Furthermore, validating the predictions from any model through experiments is challenging in general; because the worker population on mTurk is not temporally static (Difallah et al. 2018), the worker population in the validation experiments may be different from the initial experiments used to estimate the model. Nevertheless, our results in Figure 6 suggest that the estimator can reasonably predict experimental outcomes under a given incentive contract.

5.7. Discussion

Our results suggest larger incentives can increase quality on crowdwork platforms, corroborating the results of Ho et al. (2015). Although similar results are reported in the literature, we have taken a complementary approach by characterizing worker performance over a *class* of incentive contracts. Furthermore, the tractability of the optimal contracting problem under our agent model allows us to estimate performance under an optimal contract. In particular, as summarized in Figure 5, we find that increasing the expected worker payment by about $1 increases the probability that a worker crosses the bonus threshold by 0.08–0.12, depending on the worker's location. Most notably, we find diminishing returns to quality at relatively low payments in both worker groups, which may help explain why requesters tend to set low wages on mTurk (Hara et al. 2018).

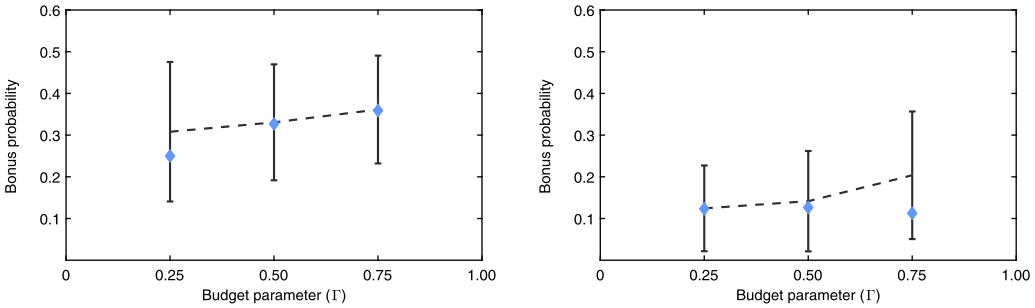
We also observe that quality can depend strongly on the worker's location. In particular, as seen in Figure 5, the bonus probability for the India group at an expected payment $1.00 is comparable to the U.S. group at $0.10. This result aligns with a finding by Shaw et al. (2011), who observe that quality on mTurk is much more strongly associated with worker location than financial incentives. Although we have only focused on worker location in this study, our approach can be readily extended to other worker attributes, provided sufficient data are available.

We highlight some limitations of our study and note directions for future work. First, we have treated agent costs as hyperparameters by tuning them through cross-validation. This makes the costs used in our model a rough approximation of actual worker costs and may limit the interpretability of the resulting agent model. Our agent model also does not capture many of the worker dynamics present in crowdwork platforms. Horton and Chilton (2010) point out that mTurk worker output appears to deviate from what would be predicted by simple, rational agent models, which applies to our model as well. Last, an important aspect of crowdwork not addressed here is worker welfare. In particular, mTurk has been widely criticized for low worker pay, which is often far below the U.S. minimum wage (Hara et al. 2018). Although we

Table 5. Optimal Incentive Contracts Under Three Different Values of Γ and Associated Results from mTurk Experiments

	Budget (Γ)	Base	Bonus	Submissions	> 75%	Empirical bonus probability
United States	0.25	0.16	0.42	20	5	0.20
	0.50	0.23	0.47	52	17	0.33
	0.75	0.41	0.78	39	14	0.36
India	0.25	0.11	0.31	73	9	0.12
	0.50	0.25	0.42	79	10	0.13
	0.75	0.58	0.77	71	8	0.11

Figure 6. (Color online) Empirical Bonus Probabilities and 95% Prediction Intervals of Six Contracts Implemented on mTurk



Notes. (a) United States. (b) India.

did not address worker welfare in this paper, our modeling framework can also be used to characterize welfare over a class of incentive contracts and allows for welfare considerations to be explicitly incorporated into the optimal contracting problem (e.g., by imposing constraints on agent utility). Investigating the tradeoff between worker welfare and quality in crowdwork may be a fruitful direction for future work.

6. Conclusion

We proposed an approach for estimating parameters that govern agent production in a moral-hazard principal-agent model. First, we presented an estimator for a nonparametric agent model, and showed it to be statistically consistent. To avoid computational drawbacks of solving the estimator exactly, we proposed an approximate estimator based on a restricted parameter set and characterized the approximation error both asymptotically and in a finite-sample setting. To solve the restricted estimator, we developed a novel column generation technique that uses hypothesis testing to select variables, which we showed preserves consistency. Numerical results show that the approximation scheme and solution technique produce accurate estimates in a computationally efficient manner. Last, we applied our estimator to data from a randomized experiment on a crowdwork platform to demonstrate how our method can be used to characterize performance over a class of incentive contracts and identify optimal incentives from the estimated model.

We conclude by noting some possible directions for future work. Our estimation procedure is built upon a general but simple moral-hazard agent model; it may be useful to extend our approach to accommodate other common features of principal-agent models, such as unobserved heterogeneity and risk aversion. There may also be fertile ground in generalizing our statistical column generation algorithm to other integer programming problems. In particular, our approach may be relevant to other estimation problems where the parameter space is a very large set of discrete distributions. Last, estimating an agent model from data may be valuable for investigating questions related to worker welfare, which is an issue of increasing prominence in online labor platforms.

Endnotes

[1] For example, Lyft offers drivers bonuses for fulfilling a target number of rides within a predefined time frame (Lyft 2021), and Postmates offers a similar incentive (Postmates 2021). Similarly, freelance platforms Upwork and Amazon Mechanical Turk allow clients to provide workers with bonuses at their own discretion.

[2] We extend our model to accommodate heterogeneous agents in Section EC.3 of the e-companion.

[3] Throughout the paper, we shall use *estimator* to refer to an optimization problem or algorithm and *estimate* to refer to its solutions.

[4] The assumption that the contract data $\mathbf{r}^i$, $i \in I$ is generated by a continuous density function $f(\mathbf{r})$ is important for our technical results. Intuitively, because the $\mathbf{r}^i$ are input data, assuming this continuity provides the estimator with more information, which makes precise inference of $\boldsymbol{\pi}^0$ possible under the identifiability condition in Assumption 2. If the contract data are instead generated by a discrete distribution supported on a subset of R, then a stronger identifiability than Assumption 2 is needed to compensate for the loss of information. We consider such a case in Section EC.4 of the e-companion.

[5] Because we assume the data are generated by n independent agents making decisions simultaneously, which is plausible in online labor platforms, the i.i.d. assumption is not particularly restrictive for our setting. Moreover, this assumption is not strictly necessary to achieve consistency, depending on the problem setup. In Section EC.4 of the e-companion, we consider a variation of the model where $\boldsymbol{\pi}^0$ can be estimated by dynamically selecting the contracts to offer the agent. This breaks the independence assumption on the contracts $\mathbf{r}^i$ but allows for consistent estimation of $\boldsymbol{\pi}^0$ under a different set of assumptions.

[6] The parameter set $\tilde{\Pi}$ may be empty if the requirement that $\boldsymbol{\pi}_a \in V$ for $a \in A$ conflicts with the requirement that $\boldsymbol{\pi} \in Q_\pi$ from (6). In this case, nonemptiness of $\tilde{\Pi}$ can be guaranteed by projecting the candidate distributions contained in V onto the polyhedron Q_π.

[7] Both benchmark methods are implemented using MATLAB's Statistics and Machine Learning Toolbox using default settings.

Acknowledgments

The authors thank the department editor Chung Piaw Teo, the associate editor, and three anonymous referees for insightful and constructive feedback, which significantly improved this paper.

References

Adams WP, Forrester RJ, Glover FW (2004) Comparisons and enhancement strategies for linearizing mixed 0-1 quadratic programs. *Discrete Optim.* 1(2):99–120.

Anderson TW (1962) On the distribution of the two-sample Cramer-von Mises criterion. *Ann. Math. Statist.* 33(3):1148–1159.

Anderson TW, Darling DA (1952) Asymptotic theory of certain goodness of fit criteria based on stochastic processes. *Ann. Math. Statist.* 23(2):193–212.

Aswani A, Shen Z-J, Siddiq A (2018) Inverse optimization with noisy data. *Oper. Res.* 66(3):870–892.

Aswani A, Shen Z-JM, Siddiq A (2019) Data-driven incentive design in the Medicare Shared Savings Program. *Oper. Res.* 67(4): 1002–1026.

Bajari P, Benkard CL, Levin J (2007) Estimating dynamic models of imperfect competition. *Econometrica* 75(5):1331–1370.

Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* 46(3):316–329.

Bertsimas D, Tsitsiklis JN (1997) *Introduction to Linear Optimization*, vol. 6 (Athena Scientific, Belmont, MA).

Bertsimas D, Gupta V, Paschalidis IC (2015) Data-driven estimation in equilibrium using inverse optimization. *Math. Programming* 153(2):595–633.

Bickel PJ, Doksum KA (2015) *Mathematical Statistics: Basic Ideas and Selected Topics, Volumes I-II Package* (Chapman and Hall).

Breiman L (1996) Bagging predictors. *Machine Learn.* 24(2):123–140.

Casella G, Berger RL (2002) *Statistical Inference*, vol. 2 (Duxbury, Pacific Grove, CA).

Chan TCY, Lee T, Terekhov D (2019) Inverse optimization: Closed-form solutions, geometry, and goodness of fit. *Management Sci.* 65(3):1115–1135.

Cochran WG (1952) The Chi-squared test of goodness of fit. *Ann. Math. Statist.* 23(3):315–345.

Conover WJ (1972) A Kolmogorov goodness-of-fit test for discontinuous distributions. *J. Amer. Statist. Assoc.* 67(339):591–596.

de Zegher JF, Iancu DA, Lee H (2019) Designing contracts and sourcing channels to create shared value. *Manufacturing Service Oper. Management.* 21(2):271–289.

Difallah D, Filatova E, Ipeirotis P (2018) Demographics and dynamics of mechanical turk workers. *Proc. 11th ACM Internat. Conf. on Web Search and Data Mining* (Association for Computing Machinery, New York), 135–143.

Duflo E, Hanna R, Ryan SP (2012) Incentives work: Getting teachers to come to school. *Amer. Econom. Rev.* 102(4):1241–1278.

Esfahani PM, Shafieezadeh-Abadeh S, Grani A Hanasusanto DK (2018) Data-driven inverse optimization with imperfect information. *Math. Programming* 167(1):191–234.

Ferrall C, Shearer B (1999) Incentives and transactions costs within the firm: Estimating an agency model using payroll records. *Rev. Econom. Stud.* 66(2):309–338.

George-Levi G, Miller RA (2015) Identifying and testing models of managerial compensation. *Rev. Econom. Stud.* 82(3):1074–1118.

Georgiadis G, Powell M (2022) A/B contracts. *Amer. Econom. Rev.* 112(1):267–303.

Glover F (1975) Improved linear integer programming formulations of nonlinear integer problems. *Management Sci.* 22(4):455–460.

Grossman SJ, Hart OD (1983) An analysis of the principal-agent problem. *Econometrica* 51(1):7–45.

Hara K, Adams A, Milland K, Savage S, Callison-Burch C, Bigham JP (2018) A data-driven analysis of workers' earnings on amazon mechanical turk. *Proc. CHI Conf. on Human Factors in Comput. Systems* (Association for Computing Machinery, New York), 1–14.

Harris C (2011) You're hired! An examination of crowdsourcing incentive models in human resource tasks. *Proc. Workshop on Crowdsourcing for Search and Data Mining at the 4th ACM Internat. Conf. on Web Search and Data Mining* (Association for Computing Machinery, New York), 15–18.

Ho C-J, Slivkins A, Suri S, Vaughan JW (2015) Incentivizing high quality crowdwork. *Proc. 24th Internat. Conf. on World Wide Web.* (Association for Computing Machinery, New York), 419–429.

Holmstrom B (1979) Moral hazard and observability. *Bell J. Econom.* 10(1):74–91.

Horton JJ, Chilton LB (2010) The labor economics of paid crowdsourcing. *Proc. 11th ACM Conf. on Electronic Commerce.* 209–218.

Ipeirotis PG, Provost F, Wang J (2010) Quality management on Amazon Mechanical Turk. *Proc. ACM SIGKDD Workshop on Human Comput.* (Association for Computing Machinery, New York), 64–67.

Keshavarz A, Wang Y, Boyd S (2011) Imputing a convex objective function. *Proc. IEEE Internat. Sympos. on Intelligent Control* (IEEE, Piscataway, NJ), 613–619.

Lee DKK, Zenios SA (2012) An evidence-based incentive system for Medicare's End-Stage Renal Disease Program. *Management Sci.* 58(6):1092–1105.

Lubbecke ME, Desrosiers J (2005) Selected topics in column generation. *Oper. Res.* 53(6):1007–1023.

Luedtke J, Namazifar M, Linderoth J (2012) Some results on the strength of relaxations of multilinear functions. *Math. Programming* 136(2):325–351.

Lyft (2021) Ride challenges. Accessed March 30, 2021, https://help.lyft.com/hc/enca/articles/360001943867-Ride-Challenges.

Mason W, Watts DJ (2009) Financial incentives and the performance of crowds. *Proc. ACM SIGKDD Workshop on Human Comput.* (Association for Computing Machinery, New York), 77–85.

Massey FJ (1951) The Kolmogorov-Smirnov test for goodness of fit. *J. Amer. Statist. Assoc.* 46(253):68–78.

Misra S, Nair HS (2011) A structural model of sales-force compensation dynamics: Estimation and field implementation. *Quant. Marketing Econom.* 9(3):211–257.

Misra S, Coughlan AT, Narasimhan C (2005) Salesforce compensation: An analytical and empirical examination of the agency theoretic approach. *Quant. Marketing Econom.* 3(1):5–39.

Paarsch HJ, Shearer B (2000) Piece rates, fixed wages, and incentive effects: Statistical evidence from payroll records. *Internat. Econom. Rev.* 41(1):59–92.

Postmates (2021) How do bonuses and incentives work? Accessed March 30, 2021, https://support.postmates.com/fleet/articles/228603028-article-How-do-bonuses-and-incentives-work-.

Sappington DEM (1991) Incentives in principal-agent relationships. *J. Econom. Perspectives* 5(2):45–66.

Scholz FW, Stephens MA (1987) K-sample Anderson–Darling tests. *J. Amer. Statist. Assoc.* 82(399):918–924.

Shaw AD, Horton JJ, Chen DL (2011) Designing incentives for inexpert human raters. *Proc. ACM Conf. on Computer Supported Cooperative Work* (Association for Computing Machinery, New York), 275–284.

Shearer B (2004) Piece rates, fixed wages and incentives: Evidence from a field experiment. *Rev. Econom. Stud.* 71(2):513–534.

Slakter MJ (1965) A comparison of the Pearson chi-square and Kolmogorov goodness-of-fit tests with respect to validity. *J. Amer. Statist. Assoc.* 60(311):854–858.

Smirnov N (1948) Table for estimating the goodness of fit of empirical distributions. *Ann. Math. Statist.* 19(2):279–281.

Stephens MA (1974) Edf statistics for goodness of fit and some comparisons. *J. Amer. Statist. Assoc.* 69(347):730–737.

Van der Vaart AW (2000) *Asymptotic Statistics*, vol. 3 (Cambridge University Press, Cambridge, UK).

Vanderbeck F, Wolsey LA (1996) An exact algorithm for IP column generation. *Oper. Res. Lett.* 19(4):151–159.

Vera-Hernandez M (2003) Structural estimation of a principal-agent model: Moral hazard in medical insurance. *RAND J. Econom.* 34(4):670–693.

Yin M, Chen Y, Sun Y-A (2013) The effects of performance-contingent financial incentives in online labor markets. *Proc. AAAI Conf. on Artificial Intelligence* (Association for the Advancement of Artificial Intelligence, Menlo Park, CA), vol. 27.