Manipulation-Proof Machine Learning*

Daniel Björkegren† Joshua E. Blumenstock‡ Samsun Knight§

Brown University U.C. Berkeley Brown University

This version: May 21, 2020
First version: November 30, 2018

Abstract

An increasing number of decisions are guided by machine learning algorithms. In many settings, from consumer credit to criminal justice, those decisions are made by applying an estimator to data on an individual's observed behavior. But when consequential decisions are encoded in rules, individuals may strategically alter their behavior to achieve desired outcomes. This paper develops a new class of estimator that is stable under manipulation, even when the decision rule is fully transparent. We explicitly model the costs of manipulating different behaviors, and identify decision rules that are stable in equilibrium. Through a large field experiment in Kenya, we show that decision rules estimated with our strategy-robust method outperform those based on standard supervised learning approaches.

Keywords: machine learning, manipulation, decisionmaking, targeting

*We are grateful for helpful conversations with Susan Athey, John Friedman, and Jesse Shapiro. This project would not have been possible without the creative work of Chaning Jang, Simon Muthusi, Nicholas Owsley, and the rest of the team at the Busara Center for Behavioral Economics. We thank numerous audiences for helpful feedback. We are grateful for funding from the Brown University Seed Fund, the Bill and Melinda Gates Foundation, and the Digital Credit Observatory. Björkegren thanks the W. Glenn Campbell and Rita Ricardo-Campbell National Fellowship at Stanford University, and Microsoft Research for support. This study was pre-registered with the AEA RCT Registry (AEARCTR-0004649), and approved by the IRBs of UC Berkeley, Brown University, and the Kenya Medical Research Institute.

†dan@bjorkegren.com

‡jblumenstock@berkeley.edu

§samsun_knight@brown.edu

1 Introduction

An increasing number of important decisions are being made by machine learning algorithms. Algorithms determine what information we see online (Perlich et al., 2014); who is hired, fired, and promoted (Brynjolfsson and Mitchell, 2017); who gets a loan (Hand and Henley, 1997), and whether to give bail and parole (Kleinberg et al., 2018). In the typical machine learning deployment, an individual's observed behavior is used as input to an estimator that determines future decisions.

These applications of machine intelligence raise two related problems. First, when algorithms are used to make consequential decisions, they create incentives for people to reverse engineer or 'game.' If agents understand how their behavior affects decisions, they may alter their behavior to achieve the outcome they desire. Second, society increasingly demands a 'right to explanation' about how algorithmic decisions are made (Goodman and Flaxman, 2016; Barocas et al., 2018). For instance, articles 13-15 of the European Union's General Data Protection Regulation mandate that "meaningful information about the logic" of automated systems be made available to data subjects (European Union, 2016). However, such transparency increases the scope for gaming: the more clearly that agents know how their behavior affects a decision, the easier it is to manipulate.

These problems result from a simple core. The standard estimators that are used to construct decision rules assume that the relationship between the outcome of interest and human behaviors is stable. But this assumption tends to be violated as soon as a decision rule is implemented: agents have incentives to change their behavior to achieve more favored outcomes. When decision rules are gamed, they can produce decisions that are arbitrarily poor or unsafe. Lenders' portfolios may be swamped with fraud, social media may be overrun by nefarious actors, self driving cars can be tricked into crashing (Eykholt et al., 2018). This problem can undermine the use of machine learning in critical applications, and is used to justify keeping decisionmaking secret.

There are two common approaches to deal with this problem. The first, familiar to economists, restricts models to predictors that are presumed to have a theoretical or structural relationship to the outcome of interest.[1] This theory-driven approach amounts to having a dogmatic prior that the cost of manipulation is either infinite (for included features) or zero (for excluded features). However, most behaviors are manipulable at some cost. The

[1] An extreme version of this restricts to predictors that causally affect the outcome of interest (Kleinberg and Raghavan, 2019; Milli et al., 2019). This may make manipulation desirable: for example, an exam may induce students to study and learn general knowledge.

second approach, which we refer to as the 'industry approach', keeps decision rules secret, and periodically updates the model to account for changes in the relationship between features and outcomes (Bruckner and Scheffer, 2011). However, such 'security through obscurity' exposes current applications to substantial risk (NIST 2008). It also limits the application of machine learning in settings where secrecy cannot be maintained (e.g., when regulations mandate transparency, or when consumers learn decision rules directly or through third parties) or feedback is noisy or delayed (e.g., it may take years for a social media platform to learn that its content prioritization algorithm was gamed by foreign actors). There is also no guarantee that the back and forth between estimation and agents will reach equilibrium, or if it does, that such an equilibrium will be desirable.

This paper develops a new approach. We explicitly model the costs that agents incur to manipulate their behavior, and embed the resulting game theoretic model within a machine learning estimator. This allows us to derive estimators that anticipate strategic agents, and which produce stable decisions even when the decision rule is fully transparent. We demonstrate, using Monte Carlo simulations, that our 'strategy-robust' estimator performs better than standard models when these costs are known, even if costs are misspecified. We then test the theory in a real world environment, through an incentivized field experiment with 1,557 people in Kenya. We use the experiment to elicit costs of manipulating behavior, and to show that the strategy-robust approach leads to more robust machine decisions.

The paper is organized into two main parts. The first part develops a method to estimate strategy-robust decision rules that are stable under manipulation. We consider a supervised machine learning framework for a policymaker making a decision y_i for each individual i. Each individual prefers a larger decision y_i. We observe a *training* subset of cases that possess both features $\mathbf{x}_i$ and optimal decisions y_i. The policymaker seeks to estimate a decision rule $\hat{y}(\mathbf{x}_i)$ for cases in a *testing* subset where only features $\mathbf{x}_i$ are observed. Standard methods assume that $\mathbf{x}_i$'s are fixed: training and test samples of $(\mathbf{x}_i, y_i)$ are drawn from same distribution. Our method allows individuals to adjust behavior in response to the incentives generated by the decision rule: $\mathbf{x}_i(\hat{y}(\cdot))$ is a function of the decision rule. As a result, while our training samples come from an unincentivized distribution $(\mathbf{x}_i(0), y_i)$; test samples come from $(\mathbf{x}_i(\hat{y}(\cdot)), y_i)$. We assume individuals incur quadratic costs for manipulating behavior $(\mathbf{x}_i)$, and that these costs can be parametrized by a matrix $\mathbf{C}_i$. We describe several methods to estimate this cost matrix, a new object needed to determine how behavior shifts when incentivized.

To sharpen intuition, we derive results for linear decision rules of the form $\hat{y}(\mathbf{x}) = \boldsymbol{\beta}\mathbf{x}$.

The resulting estimator takes a simple nonlinear least squares form. Our method introduces a new notion of fit, which has analogues to other common linear regression approaches. Ordinary least squares (OLS) maximizes fit within sample; two stage least squares (2SLS) sacrifices fit within sample to estimate coefficients that have causal interpretations; penalized least squares (such as LASSO and ridge) sacrifice within-sample fit to better generalize to other samples drawn from the same population. Our method sacrifices fit within sample to maximize equilibrium fit in the counterfactual where the decision rule is used to allocate resources, and agents manipulate against it. Our estimator is an example of a new class of estimator that maximizes *counterfactual fit*–predictive fit in a counterfactual state of the world.

We use Monte Carlo simulations to compare this new strategy-robust approach to common alternatives. OLS can perform extremely poorly when agents behave strategically. The industry approach, which periodically retrains the model, may not converge, or if it does, may do so slowly or to an undesirable equilibrium. By contrast, our method adjusts the model to anticipate manipulation, in a Stackelberg solution. In simulations where agents respond to the decision rule, and manipulation costs are known, our approach exceeds the performance of other estimators. Our approach can exceed the performance of others even if manipulation costs are misspecified for some cases. Under certain parameters, the presence of manipulation can *improve* predictive performance, if it signals unobservables associated with the outcome of interest (in the spirit of Spence, 1973). In these cases, one may wish to use certain features that are manipulable by the types that you want to screen in, but not by those you want to screen out.

In the second part of the paper, we implement and test our method in the context of a field experiment in Kenya. This experiment allows us to compare the performance of the strategy-robust estimator to standard machine learning algorithms in a real-world environment. Specifically, we built a new smartphone app that passively collects data on how people use their phones, and disburses monetary rewards to users based on the data collected. The app is designed to mimic 'digital credit' products that are spreading dramatically through the developing world (Francis et al., 2017). Digital credit products similarly collect user data, and convert it into a credit score using machine learning, based on the insight that historical patterns of mobile phone use can predict loan repayment (Björkegren, 2010; Björkegren and Grissen, 2019). However, as these systems have scaled, manipulation has become commonplace as borrowers learn what behaviors will increase their credit limits

(McCaffrey et al., 2013; Bloomberg, 2015).[2]

This field experiment produces several results. First, consistent with prior work, we show that a person's mobile phone usage behaviors ($\mathbf{x}_i(0)$) can be used to predict characteristics of the phone user, such as income, intelligence (Raven's matrices), and overall activity.[3] Second, through the use of randomly-assigned experiments, we structurally estimate $\mathbf{C}_i$ in our model, i.e., the costs of manipulating a variety of observed behaviors $\mathbf{x}_i$. Our experiments offer financial incentives to participants for altering behaviors that are observed through the app, such as increasing the number of outgoing calls in a given week, or decreasing the number of incoming text messages. The pattern of costs is intuitive: outgoing communications are less costly to manipulate than incoming communications; text messages, which are relatively cheap to send, are more manipulated than calls, which are relatively expensive. We also find that complex behaviors (such as the standard deviation of talk time) are less manipulable than simpler behaviors (such as the average duration of talk time).

Second, we find that 'strategy-robust' decision rules, which account for the costs of manipulation, perform substantially better than standard machine learning algorithms. We make this comparison by offering rewards to people who use their phones like a person of a particular type. For instance, some people receive a message that says, "Earn up to 1000 Ksh if the Sensing app guesses that you are a high income earner, based on how you use your phone," while others receive messages that offer rewards for acting like an "intelligent" person, and so forth. Across a variety of such decision rules, we show that classifications made with the strategy-robust algorithm are more accurate than classifications from standard algorithms.

Finally, we use our method to estimate the equilibrium cost of algorithmic transparency: the loss incurred to the policymaker for disclosing details of the decision rule. In the experiment, we experimentally vary the amount of information subjects have about the decision rule (the model used to predict the outcome), and show that the relative performance of the strategy-robust estimator increases with transparency. While predictive performance decreases by on average 23% under transparency for standard machine learning estimators, the strategy-robust estimator reduces this cost of transparency to approximately 8%. Overall, this suggests that the equilibrium cost of moving from a regime where the decision rules are secret, to one where they are disclosed, to be less than 8% in our setting. Our model allows

[2]A recent survey in Kenya and Tanzania found that one of the top five reasons people report saving money in digital accounts is to increase the loan amount qualified for (FSD Kenya, 2018).

[3]Prior work has used mobile phone data to predict income and wealth (Blumenstock et al., 2015; Blumenstock, 2018), gender (Blumenstock et al., 2010; Frias-Martinez et al., 2010), and employment status (Sundsøy et al., 2016), and loan repayment (Björkegren and Grissen, 2018, 2019), .

policymakers to bound this equilibrium cost of transparency even without disclosing decision rules to the world.

Taken together, the paper develops and tests a new approach to supervised learning when agents are strategic. This relates to papers from a variety of sub-literatures have confronted the notion that agents will act strategically when their actions are used to determine allocations. Our paper aims to integrate these approaches by applying principles of mechanism design to the machine learning setting, where data may have many dimensions and traditional approaches to designing incentive-compatible allocations are not possible. To our knowledge, this is also the first paper to estimate and test a strategy-robust machine learning estimator using data from a field experiment.

1.1 Connection to Literature

The dilemma of manipulation is not new. Goodhart (1975), in what has since become referred to as 'Goodhart's Law', noted that once a measure becomes a target, it ceases to be a good measure. Lucas (1976) also famously observed that historical patterns can warp when economic policy changes. More broadly, our approach connects with literatures in both economics and computer science.

Our problem can be viewed as a mechanism design problem. Canonical signaling models (Spence, 1973) rely on a single crossing condition to allow full revelation of individual types. In our setting, like the settings of Frankel and Kartik (2019, 2020) and Ball (2019), there are two forms of heterogeneity: types $\boldsymbol{\theta}_i$ and the costs of manipulating behavior $\mathbf{C}_i$. Frankel and Kartik (2019) show that unobserved heterogeneity in manipulation costs $\mathbf{C}_i$ 'muddles' the relationship between behavior $\mathbf{x}_i$ and types $\boldsymbol{\theta}_i$, causing the single crossing condition to fail. That paper shows that muddling reduces the information available in a market. Ball (2019) extends that framework to multiple dimensions of behavior, and in a theoretical model similar to ours, characterizes and proves the existence of equilibrium. That paper, as well as Frankel and Kartik (2020), show that committing to a Stackelberg solution like ours can lead to better outcomes than the repeated best response used by what we call the industry approach. Relative to this work, our paper builds a model that can be empirically estimated, which allows us to probabilistically separate types and costs.[4]

[4]In a related setting, Hussam et al. (2017) implement an incentive compatible mechanism that collects peer reports to estimate an individual's entrepreneurial ability. That method requires gathering peer reports from a community during implementation; in contrast, our approach produces stand in replacements for standard machine learning models, which can use arbitrary data on behavior. Also related, Holmström (1979) shows that a principal should use any information that has signal when contracting with an agent. Our method suggests how manipulable information be downweighted. Eliaz and Spiegler (2018) study a

Our paper is also related to the problem in public finance of setting taxes in environments where agents adapt their behaviors. Our method weights predictors by the inverse of the matrix of the costs of manipulating them, in a manner similar to Ramsey (1927). Relatedly, Mirrlees (1971) recommends using proxies when it is not possible to observe the true income earning ability of potential beneficiaries. Niehaus et al. (2013) find that when implementing agents can be corrupted, considering additional poverty indicators can worsen the targeting of benefits, by making it more difficult to verify eligibility.

Finally, our approach relates to existing strands in the computer science literature. The theoretical computer science community has recently considered this problem as one of 'strategic classification' (Hardt et al., 2016; Dong et al., 2018). This literature is focused primarily on obtaining computationally efficient learning algorithms, and how strategic behavior can affect statistical definitions of fairness (Hu et al., 2019; Milli et al., 2019). In computer security, 'adversarial machine learning' considers how strategic adversaries can systematically undermine supervised learning algorithms, typically by injecting erroneous data into the model fitting procedure.[5] Also related is the concept of 'covariate shift', which considers scenarios where a test distribution differs from the training distribution. However, it is common to assume that the conditional distribution $y|\mathbf{x}$ is fixed, and the distribution of $\mathbf{x}$'s changes exogenously (Sayed-Mouchaweh and Lughofer, 2012). The manipulation we consider induces the conditional distribution $y|\mathbf{x}$ to change endogenously when action is taken based on the estimated relationship.

Thus, papers from a variety of sub-literatures have confronted the notion that agents will act strategically when their actions are used to determine allocations. Relative to prior work, our paper makes two main contributions. First, we develop an equilibrium model of manipulation that can be estimated using data, which produces a machine learning estimator that functions well under manipulation even when the decision rule is fully transparent. And second, to our knowledge for the first time in any literature, we design and implement a field experiment that stress-tests such an estimator in a real-world setting with incentivized agents.

---

related problem where a "statistician" is making decisions on behalf of an agent, with two-sided incomplete information: the agent knows his preferred behavior, but the statistician knows the decision rule. They focus on characterizing incentive-compatible estimators, and find that commonly-used regularized linear models create incentive issues.

[5]For instance, Bruckner and Scheffer (2011) study adversarial prediction when the agent acts in response to an observed predictive model, with an application to spam filtering. Dong et al. (2018) model an iterated industry approach where a policymaker observes how agents manipulate in response to previous rules, but does not know their utility functions or costs.

1.2 Applications and Examples

Agents game decision rules in a wide variety of empirical settings. Manipulation has been documented in contexts ranging from New York high school exit exams (Dee et al., 2019) and health provider report cards (Dranove et al., 2003), to pollution monitoring in China (Greenstone et al., 2019), to fish vendors in Chile (Gonzalez-Lira and Mobarak, 2019). In the online advertising industry, firms spend many millions of dollars each year on search engine optimization, manipulating their websites in order to receive a higher ranking from search engine algorithms (Borrell Associates, 2016). A quick Google search suggests over 50 thousand different websites (and 3,000 YouTube videos) contain the phrase "hack your credit score."

We apply our method to an experiment that mimics poverty targeting. In developing countries, where income is difficult to observe, policymakers commonly target program eligibility (y_i) based on easily observable characteristics or behaviors ($\mathbf{x}_i$) (Hanna and Olken, 2018). The policymaker may infer a household's type based on the levels of these variables, or, implicitly, on how they change in response to incentives.[6] There is evidence that such decision rules induce households to manipulate their observable features. For instance, Banerjee et al. (2018) find that adding a question about flat screen TV ownership to a census caused people to underreport ownership by 16% on a follow-up survey, in order to appear less wealthy.[7]

The method we develop is directly relevant to a variety of other settings where a policymaker derives a decision from a prediction (y_i) based on agent behaviors ($\mathbf{x}_i$). These include other supervised settings where it is possible to obtain a ground truth value of y_i for a training sample of individuals. For instance, in credit scoring applications, a decision about whether it is prudent to provide a loan (y_i) is made based on characteristics on the potential borrower (traditional credit scores are based on the borrower's formal credit history, but increasingly the characteristics $\mathbf{x}_i$ include private data like mobile phone usage (Björkegren and Grissen, 2019) and social network structure (Wei et al., 2015)). It also includes settings where no definite ground truth of y_i exists. Search engines, social media, and spam filters attempt to determine the quality of a piece of content (y_i) based on features that can be observed ($\mathbf{x}_i$: keywords, reputation of the sender, inbound links). Manipulating these features may be costly directly, or may undermine the author's intent in distributing the content. Similarly,

[6] Our method thus nests this latter case of self-targeting (Nichols and Zeckhauser, 1982; Alatas et al., 2016), which identifies beneficiaries based on willingness to engage with a costly "ordeal."

[7] In other examples from the development literature, Camacho and Conover (2011) find that after a program eligibility decision rule was made transparent to local officials in Colombia, it was manipulated by an amount corresponding to 7% of the National Health and Social Security budget. They note, "there is anecdotal evidence of people moving or hiding their assets, or of borrowing and lending children."

'report cards' for universities, hospitals, and doctors attempt to determine quality (y_i) based on indicators ($\mathbf{x}_i$: alumni giving rates, endowment size, acceptance rates, graduation rates).[8]

The remainder of the paper is organized as follows. The next section introduces our theory. Section 3 describes estimation. Section 4 describes the results of our field experiment. Section 5 discusses extensions. Section 6 concludes.

2 Theory

This section introduces the model underlying our estimator, and demonstrates the intuition with simulations.

2.1 Model

A policymaker observes a *training* subset of cases that possess both features $\mathbf{x}_i$ and optimal decisions y_i. The policymaker also obtains information on the costs of manipulating features, which will be detailed later. The policymaker would like to estimate the parameters of a decision rule $\hat{y}(\mathbf{x}_i)$ for cases in a *testing* subset where only features $\mathbf{x}_i$ are observed, and may be manipulated.

A policymaker has a preferred action y_i for each ***individual*** i, denominated in units of individuals' utility. The action y_i can be projected onto $i's$ bliss behavior $\underline{\mathrm{x}}_i$ by the equation $y_i = b_0 + \boldsymbol{b}'\underline{\mathrm{x}}_i + e_i$, with $e_i \perp \underline{\mathrm{x}}_i$ representing idiosyncratic preference.

However, the policymaker observes an individual's actual behavior $\mathbf{x}_i$, which may differ from their bliss level $\underline{\mathrm{x}}_i$. It selects a deterministic decision rule of the form[9]:

$$\hat{y}(\mathbf{x}_i) = \beta_0 + \boldsymbol{\beta}'\mathbf{x}_i$$

Individuals can manipulate their behavior $\mathbf{x}_i$ away from their bliss level $\underline{\mathrm{x}}_i$ at some cost. i earns utility from the decision minus these costs:

$$u_i = \hat{y}(\mathbf{x}_i) - c(\mathbf{x}_i, \underline{\mathrm{x}}_i)$$

[8] Our model does not consider behaviors $\mathbf{x}_i$ that have a causal relationship to y_i, where manipulation can be productive (Kleinberg and Raghavan, 2019). It thus would not cover report card variables that directly influence quality, nor the case of a student who 'games' a test by studying ($\mathbf{x}_i \uparrow$), and as a result improves their knowledge ($y_i \uparrow$). The approach could be extended to cover such cases.

[9] Although randomizing a decision rule may make it harder to manipulate, it undermines a major goal of transparency: that people know how they are evaluated.

For simplicity, we consider the case where the utility from the decision exactly coincides with the policymaker's prediction.[10]

Individuals i are heterogeneous in two respects, bliss behaviors $\underline{\mathrm{x}}_i$ and gaming ability γ_i (as in Frankel and Kartik (2019)).

Manipulation costs are quadratic:

$$c(\mathbf{x}_i, \underline{\mathrm{x}}_i) = \frac{1}{2}(\mathbf{x}_i - \underline{\mathrm{x}}_i)' C_i (\mathbf{x}_i - \underline{\mathrm{x}}_i)$$

for matrix C_i:

$$C_i = \frac{1}{\gamma_i} \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{K1} \\ \vdots & \ddots & \vdots \\ \alpha_{1K} & \cdots & \alpha_{KK} \end{bmatrix}$$

Different behaviors may be differentially hard to manipulate, by themselves (the diagonal α_{kk}) or in conjunction with other behaviors (the off diagonals α_{kj}). And different people may find it easier or harder to manipulate (γ_i): for example, people with more technical savvy or lower opportunity cost of time may find it easier to game decision rules.

When i knows the decision rule $\hat{y}(\mathbf{x}_i)$ and receives benefits according to it, first order conditions imply he will manipulate behavior to level:

$$\mathbf{x}_i^*(\boldsymbol{\beta}) = \underline{\mathrm{x}}_i + C_i^{-1}\boldsymbol{\beta}$$

When behavior is not incentivized ($\boldsymbol{\beta} = \mathbf{0}$), optimal behavior equals the bliss level ($\mathbf{x}_i^*(\mathbf{0}) = \underline{\mathrm{x}}_i$). However, as $\boldsymbol{\beta}$ moves away from zero, behavior moves in the same direction, downweighted by the cost of manipulation (as highlighted in blue).

Decision rules. The policymaker faces expected squared loss:

$$L\left(\hat{y}(\cdot)\right) = E_i\left[\left[y_i - \hat{y}\left(\mathbf{x}_i\left(\hat{y}(\cdot)\right)\right)\right]^2 + M(\cdot)\right]$$

The first term represents fit of the model *in the counterfactual* where the model is implemented and agents manipulate behavior. If the policymaker additionally cares about the costs that individuals incur manipulating, this manipulation cost results in additional term $M(\cdot)$.

Our **strategy-robust decision rule** is given by:

[10] That is, we consider the case where the utility of the decision $u(\hat{y}) = \hat{y}$, which holds in our experiment. Under more general functions $u(\cdot)$, our model would represent a linear approximation. One could easily generalize our framework to allow for more general functional forms.

$$\boldsymbol{\beta}^{stable} = \arg\min_{\boldsymbol{\beta}} \left(\frac{1}{N} \sum_i \left[y_i - \beta_0 - \boldsymbol{\beta}'(\underline{\mathrm{x}}_i + C_i^{-1}\boldsymbol{\beta}) \right]^2 + \dots \right) \tag{1}$$

which deviates from ordinary least squares due to the term $C_i^{-1}\boldsymbol{\beta}$ which captures manipulation in response to $\boldsymbol{\beta}$. Additional terms '...' can include any weight $M(\cdot)$ the policymaker places on manipulation costs incurred by agents, and any regularization terms $R_{\lambda^{decision}}(\cdot)$.

Discussion

If the policymaker only cares about targeting performance ($M(\cdot) \equiv 0$) and there are no additional regularization terms ($R(\cdot) \equiv 0$), then ours is a nonlinear least squares estimator. Moment conditions are given by:

$$E\left[\underline{\mathrm{x}}_i \cdot \left(y_i - \beta_0 - \boldsymbol{\beta}'(\underline{\mathrm{x}}_i + C_i^{-1}\boldsymbol{\beta})\right)\right] = -2E\left[C_i^{-1}\boldsymbol{\beta} \cdot \left(y_i - \beta_0 - \boldsymbol{\beta}'(\underline{\mathrm{x}}_i + C_i^{-1}\boldsymbol{\beta})\right)\right]$$

This suggests that the estimator imposes that equilibrium errors in the counterfactual can deviate from orthogonality to individual types $\underline{\mathrm{x}}_i$. This accounts for the fact that $\boldsymbol{\beta}$ induces a marginal incentive to respond. This can depend on, for example, whether the individuals who the policymaker wishes to target for unobserved reasons (e_i) find it easier or harder to manipulate behavior (C_i). When $C_i \equiv \boldsymbol{\infty}$, the resulting estimator corresponds to OLS.

When the policymaker cares about not only the resulting allocation, but also the manipulation costs that individuals incur, this is accompanied by the term $M(\cdot)$, which can take a different form depending on policymaker preferences. An entity that is narrowly concerned with its own objective may thus select different decision rules from those that maximize social welfare (for example, a profit maximizing firm may be satisfied with an equilibrium where all individuals expend welfare gaming a test, where a social planner may not be).[11]

To reduce overfitting in small samples, one may also include common forms of regularization; for example, $R^{LASSO}_{\lambda^{decision}}(\boldsymbol{\beta}) = \lambda^{decision} \sum_{k>0} |\beta_k|$ or $R^{ridge}_{\lambda^{decision}}(\boldsymbol{\beta}) = \lambda^{decision} \sum_{k>0} \beta_k^2$. Hyperparameter $\lambda^{decision}$ can be set with cross validation in the baseline sample. Under these regularization terms, when $M(\cdot) \equiv 0$ and $C_i \equiv \boldsymbol{\infty}$ the resulting estimator corresponds to LASSO, or ridge, respectively.

[11] For example, the policymaker may place weight w on the sum of manipulation costs: $M(\cdot) = w \sum_i c(\underline{\mathrm{x}}_i + C_i^{-1}\boldsymbol{\beta}, \underline{\mathrm{x}}_i)$. The Supplemental Appendix derives a microfounded term for the case of proxy means testing.

2.2 Intuition

We demonstrate the method with Monte Carlo simulations.

We derive desired payments $\mathbf{y}$, from individual types $\underline{\mathrm{x}}$ and payment rule $\mathbf{b}$, with deviations $\boldsymbol{e}$. We then assess decision rules $\hat{y}(\mathbf{x})$ based on observed behaviors $\mathbf{x}$ generated with different estimators. Our strategy-robust estimators anticipate that behaviors $\mathbf{x}$ may change when they are used in a decision rule, factoring in manipulation costs $\mathbf{C}$. This section assumes that manipulation costs are known.

Comparative Statics

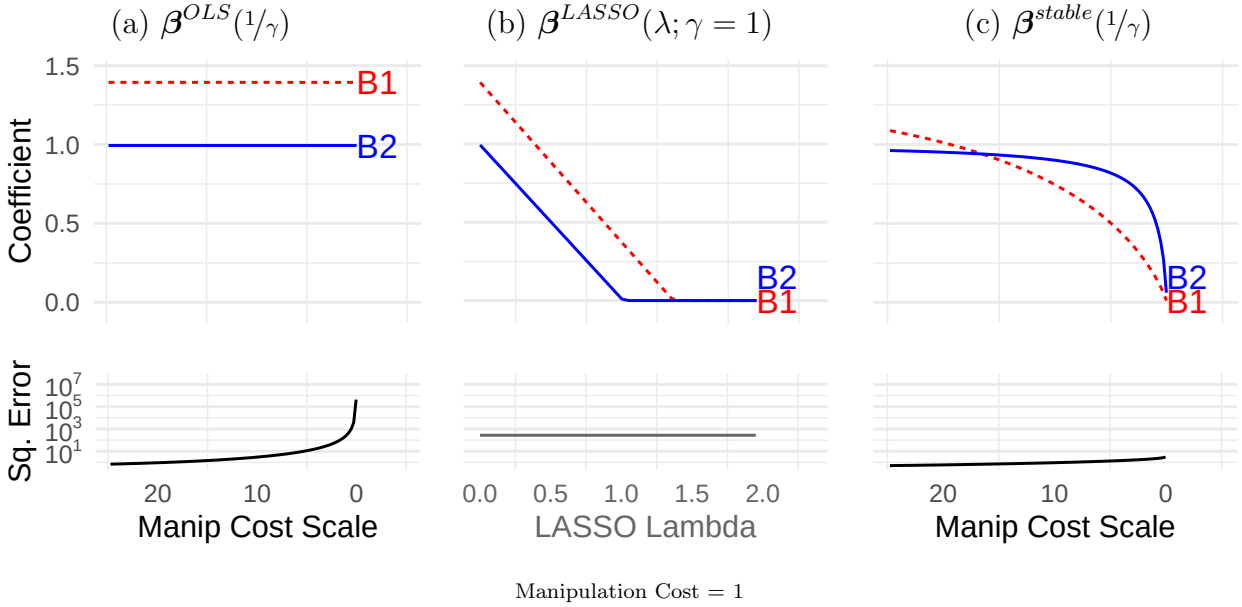
We consider a case where x_1 is more predictive than x_2 in baseline behavior, but would be easily manipulated if used in a decision rule ($b_1 > b_2$ but $\alpha_{11} \ll \alpha_{22}$).

Figure 1 compares our method to OLS and LASSO, which mistakenly place most weight on x_1. OLS maximizes predicted performance within the unincentivized sample $(\mathbf{x}_i(\mathbf{0}), y_i)$; as shown in Figure 1a, it performs poorly as manipulation becomes easier. Figure 1b shows that for a given cost of manipulation, LASSO shrinks these coefficients. However, when LASSO selects variables, it does exactly the wrong thing: it kicks x_2 out of the regression first. In contrast, our method considers how predictive features will be in equilibrium when the decision rule is implemented: $(\mathbf{x}_i(\boldsymbol{\beta}), y_i)$. As shown in Figure 1c, when manipulation costs are high, our method approaches OLS; as manipulation becomes easier, our method substantially penalizes x_1. Our method can also be combined with LASSO or ridge penalization to fine tune out of sample fit.[12]

If each feature is equally costly to manipulate, our method shrinks them together, similar to ridge regression, as shown in Figure A1. If all individuals have the same gaming ability ($\gamma_i \equiv \gamma$), then manipulation shifts behavior uniformly and does not affect predictive performance. However, even though predictive performance is high, individuals may incur substantial costs manipulating. Figure A2 develops this intuition further, by showing how the strategy-robust method penalizes indicators that are easy to shift: Figure A2a shows the effect of scaling the cost of one behavior (x_2). As the cost of manipulating that particular behavior (α_{22}) decreases, it is penalized, and weight is shifted to other behaviors. The method also penalizes indicators that make it easier to shift other predictive indicators (in a manner similar to Ramsey (1927) taxation). Figure A2b shows that the effect of cost interactions: when manipulating x_1 makes it easier to manipulate x_2 (α_{12} sufficiently negative), our method

[12] See Appendix Figure A1 for a comparison to ridge regression, as well as a demonstration of combining our method with ridge penalization.

Figure 1: Common vs. Strategy Robust Estimators



Note: The first behavior is more predictive ($b_1 > b_2$), but is easily manipulable ($\alpha_{11} \ll \alpha_{22}$). **(a)** OLS performance deteriorates substantially when behavior can be manipulated. (**b**) LASSO penalization favors x_1, which will be manipulated as soon as the decision rule is implemented. **(c)** Our method anticipates that x_1 will be manipulated if it is incentivized. It shifts weight to x_2 as behavior becomes manipulable.

$\underline{x}_i \overset{iid}{\sim} N(0,1)$, $\mathbf{b} = [1.4, 1]$, $\mathbf{C}^{het} = \frac{1}{\gamma\gamma_i}\begin{bmatrix} 4 & 0 \\ 0 & 32 \end{bmatrix}$, $\frac{1}{\gamma_i^{het}} \overset{iid}{\sim} Uniform[0,10]$, $e_i \overset{iid}{\sim} N(0, 0.25)$.
Squared error measured on an out of sample draw from the same population, incentivized to that decision rule.

further reduces weight on x_1 .

Performance

Table 1 shows the results of an example Monte Carlo simulation, chosen to demonstrate how standard approaches can fail. In this simulation, type $\underline{x}_1$ has a large weight in the desired payment ($b_1 = 3$) relative to the other two dimensions ($b_2 = b_3 = 0.1$); however, the resulting behavior x_1 is much easier to manipulate ($\alpha_{11} = 1$ vs. $\alpha_{22} = 2$ and $\alpha_{33} = 4$).

In this environment, OLS considers the static relationship in the unmanipulated data. This rule would perform well if behavior were fixed (no manipulation column); however, once consumers adjust to the rule, it makes terrible decisions (manipulation column).

The industry approach would retrain (refresh) this model after this manipulation. If we observe how consumers adjust their behavior and reestimate OLS, we obtain $\beta^{OLS(1)}$, which

Table 1: Manipulation Can Harm Prediction (Monte Carlo)

	Decision Rule				**Performance** (squared loss)	
	β_0	β_1	β_2	β_3	No manip.	Manipulation
Panel A: **Data generating process**						
$\boldsymbol{b}^{DGP}$	0.200	3.000	0.100	0.100	0.267	3745.046
Panel B: **Standard Approaches**						
$\boldsymbol{\beta}^{OLS}$	0.205	3.042	0.061	0.116	0.266	3961.225
'Industry' Approach (estimated cumulatively)						
$\boldsymbol{\beta}^{OLS(1)}$ after $\boldsymbol{\beta}^{OLS}$	-0.798	0.061	2.090	-1.675	3.275	625.762
$\boldsymbol{\beta}^{OLS(2)}$ after $\boldsymbol{\beta}^{OLS}$	-2.174	0.174	0.436	0.143	12.861	8.369
$\boldsymbol{\beta}^{OLS(3)}$ after $\boldsymbol{\beta}^{OLS}$	-1.376	0.165	0.573	0.483	9.343	4.415
⋮						
$\boldsymbol{\beta}^{OLS(100)}$ after $\boldsymbol{\beta}^{OLS}$	-1.619	0.316	0.753	-0.059	8.442	2.105
⋮						
$\boldsymbol{\beta}^{OLS(1000)}$ after $\boldsymbol{\beta}^{OLS}$	-1.854	0.489	0.582	-0.124	9.211	1.959
Panel C: **Strategy Robust Method**						
$\boldsymbol{\beta}^{stable}$	-1.813	0.503	0.536	-0.096	9.155	1.939
If costs are misestimated:						
$\boldsymbol{\beta}^{stable}_{\hat{C}_i=2diag(C_i)}$	-1.566	0.658	0.719	-0.352	6.893	10.826
Followed by Industry Approach (estimated cumulatively):						
$\boldsymbol{\beta}^{OLS(1)}$ after $\boldsymbol{\beta}^{stable}_{\hat{C}_i=2diag(C_i)}$	-2.045	0.800	0.042	0.418	10.891	4.447
$\boldsymbol{\beta}^{OLS(2)}$ after $\boldsymbol{\beta}^{stable}_{\hat{C}_i=2diag(C_i)}$	-2.022	0.558	0.327	0.137	10.685	2.453

Notes: Monte Carlo simulation results. Panel A shows the coefficients that relate the outcome (y) to behaviors ($\mathbf{x}$) under the data generating process (DGP). Panel B shows coefficients from OLS; Panel C shows coefficients estimated with the strategy robust method. Performance is assessed on the same sample of individuals, under behavior without manipulation: $\mathbf{x}_i(\mathbf{0})$, or with: $\mathbf{x}_i(\boldsymbol{\beta})$. Parameters:

$$C = \begin{bmatrix} 1.0 & 0.1 & 0.2 \\ 0.1 & 2.0 & 0.8 \\ 0.2 & 0.8 & 4.0 \end{bmatrix}, \underline{\mathrm{x}} \overset{iid}{\sim} N\left(\mathbf{0}, \begin{bmatrix} 1 & 1 & 0.1 \\ 1 & 2 & 1 \\ 0.1 & 1 & 1 \end{bmatrix}\right), \gamma_i = \begin{cases} 1 & \underline{\mathrm{x}}_{i1} \leq 0.2 \\ 10 & \underline{\mathrm{x}}_{i1} > 0.2 \end{cases}, e_i \overset{iid}{\sim} N(0, 0.25)$$

places negative weight on the manipulated x_1. However, its also makes terrible decisions when consumers respond to it. We can try to do better by repeatedly allowing individuals to best respond, and then reestimating the decision rule. But even with perfect information and no changes in the environment, this process may make poor decisions en route, and may converge to a suboptimal equilibrium, or not at all.

If we estimate $\boldsymbol{\beta}^{OLS(r)}$ using cumulative data from all prior periods $(1, \ldots, r-1)$, it continues to make terrible decisions over several iterations of the algorithm designer announcing decision rules, and consumers reoptimizing behavior. While the performance of these decisions then begins to improve, even after a thousand such reoptimizations, it performs worse than our approach. (See the second set of estimates in Table 1.)

If we instead estimate $\boldsymbol{\beta}^{OLS(r)}$ using only data from the prior period $(r-1)$ (iterative best response), this approach does not reach equilibrium. It alternates between decision rules that place high and low weight on x_1 (see Table A1).

Thus standard approaches can perform poorly even in ideal cases. If there were noise or frictions in learning, the risks of this approach are greater: a system could appear to be making good decisions, only to fail catastrophically when the other side discovers how to exploit it.[13]

In contrast, our strategy-robust estimator ($\boldsymbol{\beta}^{stable}$) anticipates that including a behavior in the decision rule will shift that behavior. It penalizes the easily manipulable behavior x_1, and shifts weight to behaviors that are harder to manipulate (x_2 and x_3). It sacrifices performance in the environment in which it is trained (in sample, no manipulation) for performance in the counterfactual where there is manipulation. When individuals manipulate as described in the model, our estimator exceeds the performance of other estimators.[14]

Our method can reduce risk even if manipulation costs are misestimated. We consider a case with two measurement mistakes: (a) all off diagonal elements are set to zero, and (b) the estimated costs of manipulation are two times too large. Performance deteriorates relative to the case where we know the true cost matrix, but our method still outperforms OLS in the presence of manipulation. One can use our method as a first step towards equilibrium, and then follow it with the industry approach; as shown in the bottom rows, doing so skips the terrible decisions made in the first two iterations of the industry approach.

[13]For example, Gonzalez-Lira and Mobarak (2019) find that increased enforcement of a ban on selling an endangered fish can lead vendors to learn about the decision rule, and more effectively undermine it).

[14]Our method discovers a Stackelberg solution that anticipates manipulation; it differs from a fixed point solution and thus requires a degree of commitment from the policymaker, as noted theoretically by Ball (2019) and Frankel and Kartik (2020).

Manipulation can improve performance

Manipulation can *improve* performance, if ease of manipulation (γ_i) is correlated with the outcome (y_i). In that case, manipulation itself represents a signal of the underlying type, as in Spence (1973), and applications of self-targeting (Nichols and Zeckhauser, 1982; Alatas et al., 2016). An example is shown in Table A2: manipulation improves the performance even of naïve estimators, as shown in the first two rows. Our method *increases* the coefficient on the manipulated behaviors to better exploit the information contained in manipulation, and thus further improves performance as shown in the third row.

3 Estimation

Our model can be fully estimated with experimental data. To estimate manipulation costs, we hire study participants to undermine component parts of the model, and gauge how sensitive these manipulations are to incentives.

We observe multiple time periods. Each period, an individual may desire to deviate from bliss behavior due to manipulation, or shocks that are common ($\boldsymbol{\mu}_t$) or individual specific ($\boldsymbol{\epsilon}_{it}$):

$$u_{it} = \hat{y}(\mathbf{x}_i) - c(\mathbf{x}_i, \underline{\mathrm{x}}_i) + (\boldsymbol{\mu}_t + \boldsymbol{\epsilon}_{it}) \cdot (\mathbf{x}_i - \underline{\mathbf{x}}_i)$$

where both components are mean zero: $E\boldsymbol{\mu}_t = \mathbf{0}$ and $E\boldsymbol{\epsilon}_{it} = \mathbf{0}$. Then, in week t we will observe behavior:

$$\mathbf{x}^*_{it}(\boldsymbol{\beta}) = \underline{\mathrm{x}}_i + \boldsymbol{\mu}_t + \boldsymbol{\epsilon}_{it} + C_i^{-1}\boldsymbol{\beta} \tag{2}$$

We parameterize the inverse of the cost matrix as follows:

$$C_i^{-1} = \gamma_i \cdot C^{-1}$$

with elements of inverse costs defined for convenience as:

$$C^{-1} =: \begin{bmatrix} c_{11} & \cdots & c_{K1} \\ \vdots & \ddots & \vdots \\ c_{1K} & \cdots & c_{KK} \end{bmatrix}$$

Gaming ability includes two types of heterogeneity:

$$\gamma_i = e^{-\boldsymbol{\omega}\mathbf{z}_i} + v_i$$

It is allowed to vary with characteristics $\mathbf{z}_i$ that are observable in the training sample (but need not be observed in an implementation sample; for example, we survey participants on tech savviness). It also includes unobserved heterogeneity $v_i \sim V$ with $Ev_i = 0$, which will enter the model as random effects.

We estimate strategy-robust decision rules in two steps.

3.1 Primitives

We first estimate primitives: types $\underline{\mathrm{x}}$, cost parameters $\boldsymbol{\omega}$ and C^{-1}, and the distribution of unobserved gaming ability V.

Types

We infer types $\underline{\mathrm{x}}$ by observing baseline behavior prior to the implementation of a decision rule. When $\boldsymbol{\beta} = \mathbf{0}$, behavior will not be manipulated. We can estimate types and time period fixed effects with moment conditions derived from the equation:

$$\mathbf{x}^*_{it}(\mathbf{0}) = \underline{\mathrm{x}}_i + \boldsymbol{\mu}_t + \boldsymbol{\epsilon}_{it} \tag{3}$$

including only time periods where $\boldsymbol{\beta} = \mathbf{0}$.

Costs

Our main specification recovers manipulation costs experimentally. Each week we randomly assign individuals to a decision rule $\boldsymbol{\beta}_{it}$. The decision rule may be a control, in which case $\boldsymbol{\beta}_{it} \equiv 0$. Or, it may be a treatment group that incentivizes one behavior $k \in 1...K$, by disclosing a rule that pays incentives for k: $\beta_{itk} > 0$ but not for other behaviors: $\beta_{itj} = 0$ for $j \neq k$. These treatments make it possible to recover the inverse cost matrix (diagonal and off-diagonal elements), as well as heterogeneous gaming ability (observed $\boldsymbol{\omega}$ and unobserved V).

Moment Conditions

We recover all parameters jointly with the following moment conditions.

Incentives are orthogonal to idiosyncratic behavior shocks ($E[\boldsymbol{\beta}_{itk}\epsilon_{itj}] = 0$). For each pair of behaviors jk (including $j = k$) this yields sample moment condition:

$$0 = \frac{1}{N}\sum_{i=1}^{N} \beta_{itk}\left[x_{ijt} - \underline{\mathrm{x}}_{ij} - \mu_{jt} - \beta_{itj}\left(e^{-\boldsymbol{\omega}\mathbf{z}_i} \cdot c_{kj}\right)\right]$$

We also have $E[\epsilon_{itj}] = 0$: for each time period t and behavior k, we obtain:

$$\mu_{kt} = \frac{1}{N}\sum_{i=1}^{N}\left[x_{ikt} - \underline{\mathrm{x}}_{ik} - \boldsymbol{\beta}_{it}\left(e^{-\boldsymbol{\omega}\mathbf{z}} \cdot C^{-1}\right)\right]$$

For each individual i and behavior k, we obtain:

$$\underline{\mathrm{x}}_{ik} = \frac{1}{T}\sum_{t=1}^{T}\left[x_{ikt} - \mu_{kt} - \boldsymbol{\beta}_{it}\left(e^{-\boldsymbol{\omega}\mathbf{z}} \cdot C^{-1}\right)\right]$$

given T observations.

Unobserved heterogeneity is mean zero ($E[v_i] = 0$), yielding:

$$0 = \frac{1}{T}\sum_{i,k,t \text{ where } k \text{ incentivized}}\left[\frac{x_{ikt} - \underline{\mathrm{x}}_{ik} - \mu_{kt}}{C^{-1}\boldsymbol{\beta}_{it}} - e^{-\boldsymbol{\omega}\mathbf{z}_i}\right]$$

Each heterogeneity characteristic $z \in \mathbf{z}$ is orthogonal to unobserved heterogeneity ($E[z_i v_i] = 0$), yielding:

$$0 = \frac{1}{T_z}\sum_{i} z_i \sum_{k,t \text{ where } k \text{ incentivized}}\left[\frac{x_{ikt} - \underline{\mathrm{x}}_{ik} - \mu_{kt}}{C^{-1}\boldsymbol{\beta}_{it}} - e^{-\boldsymbol{\omega}\mathbf{z}_i}\right]$$

These moment conditions jointly identify $\underline{\mathrm{x}}$, C^{-1}, and $\boldsymbol{\omega}$.

Joint Estimation

We jointly solve for the parameters to minimize the squared distance from zero:

$$L(\underline{\mathrm{x}}, C^{-1}, \boldsymbol{\omega}) + R^{\boldsymbol{\lambda}^{costs}}_{costs}(C^{-1}, \boldsymbol{\omega})$$

where $L(\cdot)$ represents the associated general method of moments (GMM) loss function.

Penalization and Cross Validation We make include two adjustments to reduce overfitting of the cost matrix to our limited dataset. First, we impose the constraint that incentivizing a behavior increases it: $c_{jj} > 0$. Second, we regularize the cost estimates:

$$R^{\boldsymbol{\lambda}^{costs}}_{costs}(\cdot) = \left[\lambda^{costs}_{diagonal}\sum_k c^2_{kk} + \lambda^{costs}_{offdiagonal}\sum_{j\neq k} c^2_{jk}\right]\left[\sum_i e^{-2\boldsymbol{\omega}\mathbf{z}}\right]$$

where we allow the possibility of using separate hyperparameters $\boldsymbol{\lambda}^{costs} = \{\lambda^{costs}_{diagonal}, \lambda^{costs}_{offdiagonal}\}$ for diagonal and off diagonal costs. These penalize the cost of manipulation towards infinity (ease of manipulation towards zero), which will tend to penalize our method's estimates towards standard methods (OLS/LASSO/etc).

We jointly solve for parameters $\underline{\mathrm{x}}$, C^{-1}, and $\boldsymbol{\omega}$, and hyperparameters $\boldsymbol{\lambda}^{costs}$ to minimize out of sample prediction error, using cross validation. Then, we impose the optimal $\boldsymbol{\lambda}^{costs}$ and jointly estimate $\underline{\mathrm{x}}$, C^{-1}, and $\boldsymbol{\omega}$ on the full sample.

Unobserved Gaming Ability

After estimating these parameters, we back out the distribution of unobserved gaming ability V in two steps. First we compute whether each individual manipulates more or less than predicted during incentivized weeks:

$$\tilde{v}_i = \frac{1}{K_i}\sum_k \frac{1}{T_i}\sum_{t \text{ where } k \text{ incentivized}}\left[\frac{x_{ikt} - \underline{x}_{ik} - \mu_{kt}}{C^{-1}\boldsymbol{\beta}_{it}} - e^{-\boldsymbol{\omega}\mathbf{z}_i}\right]$$

Second, to reduce the impact of noise and outliers, we shrink and winsorize these backed out shocks. We form the empirical distribution $V = \{\max(\phi \cdot \tilde{v}_i, \underline{\mathrm{v}})\}_i$, where $\underline{\mathrm{v}}$ is the lowest value of $\tilde{v}$ that leads to a nonnegative implied gaming ability.[15] We set the shrinkage factor ϕ to 0.005 so that less than 5% of distribution is winsorized.[16] This yields a distribution of costs C_i.

3.2 Decision Rules

Given these primitives, a strategy robust decision rule is given by:

$$\boldsymbol{\beta}^{stable} = \arg\min_{\boldsymbol{\beta}} E\left[\frac{1}{N}\sum_i \left[y_i - \beta_0 - \boldsymbol{\beta}'(\underline{\mathrm{x}}_i + C_i^{-1}\boldsymbol{\beta})\right]^2 + R^{\lambda^{decision}}_{decision}(\boldsymbol{\beta}, \mathbf{y}, \mathbf{C})\right]$$

taken over expectation over C_i, and given decision rule regularization term $R^{\lambda^{decision}}(\cdot)$. Hyperparameter $\lambda^{decision}$ is set through cross validation in the unmanipulated sample (where

[15] That is, $\underline{\mathrm{v}} = \min_i(\tilde{v}_i | \tilde{v}_i \geq \min_j(e^{-\boldsymbol{\omega}\mathbf{z}_j}))$.

[16] After shrinkage, 4.1% of observations are winsorized.

we can observe ground truth):

$$\lambda^{decision} = c.v. \arg\min_{\lambda^{cv}} \left[\min_{\boldsymbol{\beta}^{naive}} \left[\frac{1}{N} \sum_i \left[y_i - \beta_0^{naive} - \boldsymbol{\beta}^{naive} \underline{\mathrm{x}}_i \right]^2 + R^{\lambda^{cv}}_{decision}(\boldsymbol{\beta}^{naive}, \mathbf{y}, \mathbf{C}) \right] \right]$$

4 Experiment

We designed a field experiment to test the performance of our strategy-robust estimator in a real-world setting. Design started in 2017. Working with the Busara Center for Behavioral Economics in Nairobi, we developed and deployed a new smartphone-based application ('app') to 1,557 research subjects. The app was designed to mimic the key features of the 'digital credit' apps that are quickly transforming consumer credit in developing countries (Francis et al., 2017). In Kenya, at the time of our study, CGAP (2018) estimates that 27% of all adults had an outstanding 'digital credit' loan. These phone-based apps construct an alternative credit score ($\hat{y}_i$) based on how each applicant uses their phone ($\boldsymbol{x}_i$; Björkegren (2010); Björkegren and Grissen (2019)). The app we built similarly collects data on how each subject uses their phone, and uses that data to make cash transfer decisions. This section describes the app and experimental design (Section 4.1); estimates costs of manipulation and derives strategy-robust decision rules using our method; and compares the performance of these new estimators to traditional learning algorithms (Section 4.3). Our design was pre-specified in a pre-analysis plan registered in the AEA RCT registry under AEARCTR-0004649.

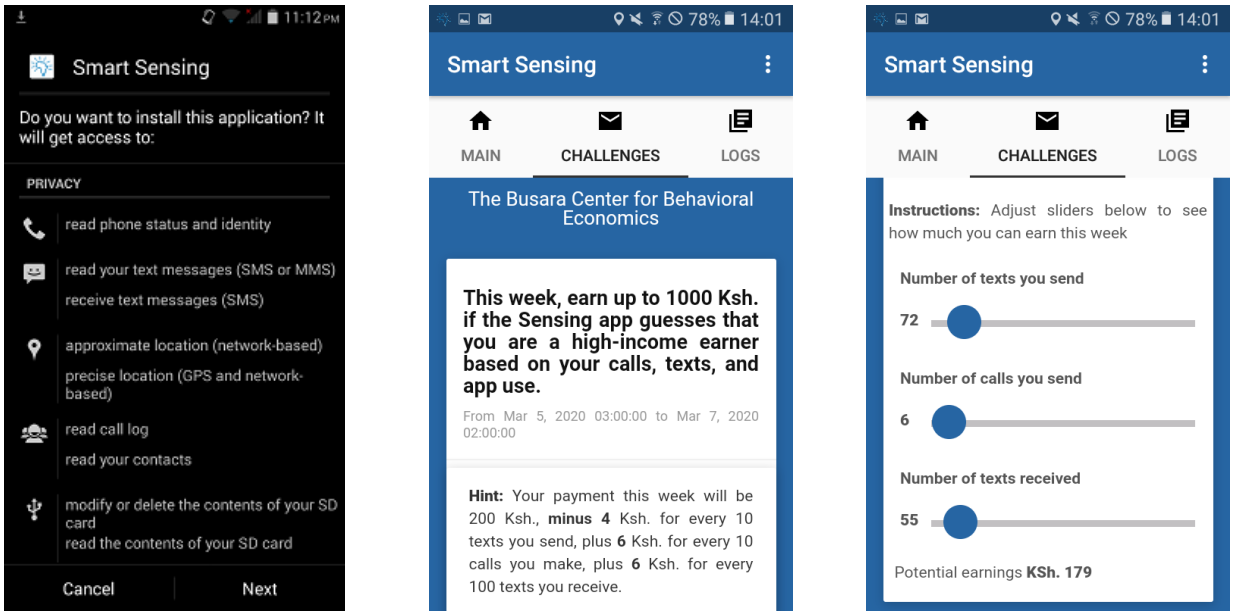
4.1 Experimental design and smartphone app

Our experiment is intended to create an environment with incentives similar to those of a 'digital credit' lending app. These apps run in the background on a smartphone, and collect rich data on phone use (including data on communications, mobility, social media behavior, and much more). Digital credit apps use this information to allocate loans to people who appear creditworthy (i.e., for whom $\hat{y}_i$ exceeds some threshold). Since financial regulations prevented us from actually underwriting loans to research subjects, we instead focused on analogous problems where a decisionmaker wishes to allocate resources to individuals with specific characteristics—for instance, by paying individuals who have a certain income level, or other characteristic (e.g., intelligence, level of activity, education).[17] This allows us to

[17]While these target predictions may bear little resemblance to credit-worthiness, there are many settings where characteristics like these are being inferred by digital traces (for example, welfare programs that target

Figure 2: Smart Sensing App

(a) Installation Screen (b) Challenge with Hint (c) Earnings Calculator



focus on the mechanics of manipulation in a prediction task, which is the same regardless of which outcome is predicted.

Smartphone app

The 'Smart Sensing' app we built has has two key features. First, it runs in the background on the smartphone to capture anonymized metadata on how individuals use their phones, such as when calls or texts are placed, which apps are installed and used, geolocation, battery usage, wifi connections, and when the screen was on. In total, we extract over $\bar{K} > 1,000$ behavioral features — Appendix Figure A3 shows the correlation between 80 different behavioral indicators ("features") collected through the app.[18] Second, the app provides a platform to deliver weekly "challenges" to research subjects (see Figure 2). These challenges appear on the subject's phone, and offer financial incentives based on their behavior. The challenges can be very simple ('You will receive 12 Ksh. for every incoming call you receive this week') or more complex ('Earn up to 1000 Ksh. if the Sensing app guesses you are a high-income earner'). Users are paid a base amount of 100 Ksh. for uploading data, plus any challenge winnings, directly via M-PESA at the conclusion of each week.

unmarried women, or digital advertisers who target college students).

[18]The app is designed to capture this data with minimal impact on battery life and performance. Data is uploaded to secure Busara servers at a set frequency, or can be uploaded manually.

Study population and recruitment

The subject population consists of Kenyans aged 18 years or older who own a smartphone and are able to travel to the Busara center in Nairobi. Participants were recruited through in person solicitations in public spaces in neighborhoods around Nairobi. From this master list of potential participants, every third individual was saved for a 'top up' sample; we drew invited individuals from this list to participate later in the experiment, to form a fresh test sample. The remaining sample was invited at the beginning. All individuals were sequentially invited for an enrollment session at the Busara center. (The center had a capacity to enroll 200 people per week.) During enrollment, participants complete a survey on a tablet on demographics and technology usage. These responses will form the ground truth about users that we seek to infer based on phone usage behavior.

Prospective participants were given the opportunity to install the Sensing App on their phones for about 16 weeks. Participants were told the dimensions of behavior that would be captured and used anonymously, and assured that no content of calls or text messages would be recorded. Participants were given the opportunity to ask questions. Participants showed understanding of the privacy tradeoffs involved, and voiced trust in Busara based on its positive reputation in this community. Participants who opted in to the study were offered help installing the Sensing App, which provided the main interaction of the study. During installation, participants had the opportunity to view the Android permissions required and to decide whether to accept. Our sample includes only participants who opted in. Participants could elect to receive challenges in English, Swahili, or both. 82.6% elected English, 15.9% elected Swahili, and 1.4% elected both.

Weekly rhythm

The study follows a weekly rhythm. Each Wednesday at noon, each user receives a generic notification, 'Opt in to see this week's challenge!', via Android notifications and a text message. When a user opens the app, it will ask them to opt in to a challenge for that week. Only after a user opts in are the details of their challenge for that week revealed (see Figure 2).[19] Challenges are valid until 6pm Tuesday. At the conclusion of the challenge, users have 16 hours to ensure that their data is uploaded (until 10am Wednesday). Busara then computes and sends any payments to users via M-PESA by noon Wednesday, and users receive the next challenge.

[19]To minimize the possibility of differential attrition, the pre-opt-in notification was the same for all users regardless of their assigned challenge.

Each week, participants could attrit in two ways: by not uploading their data, or by not opting in to the challenge.[20] Participants who failed to upload or opt in were sent text message reminders, or called by Busara staff, following an attrition protocol detailed in Appendix A1.2. We include in our analysis only participant-weeks where the participant opted in, and uploaded during the end-of-week upload window.

4.2 Baseline predictions and model estimation

Predicting user characteristics

We begin the experiment with baseline weeks that have no incentives (no active challenges). These baseline weeks allow us to estimate each individual's type in absence of manipulation, $\underline{x}$.[21] We estimate each dimension of type using Equation 3, with week fixed effects to absorb idiosyncratic weekly shocks.

Consistent with prior work (Blumenstock et al., 2015; Björkegren and Grissen, 2019), we find that characteristics of users can be predicted from phone behaviors. Results for several outcomes, based on OLS, are shown in Table 2. For characteristics such as monthly income, intelligence (Ravens Matrices), and overall phone activity, R^2 values range from 0.02 to 0.15. To make these rules easier for participants to interpret, we will focus on three variable decision rules selected via LASSO; the last row of Table 2 shows that these obtain similar R^2 when cross validated.

Evidence that app-based challenges induce manipulation

We will eventually use variation in behavior induced by our randomized experiment to estimate the cost of manipulating different behaviors, $\mathbf{C}(\mathbf{z}_i)$. This exogenous variation comes from weeks when subjects are assigned 'simple' challenges that incentivize modifying a single behavior, of the form, 'We'll pay you M for each additional x_j you do', where amount M and behavior j are assigned randomly. For example, one challenge was, 'You will receive 3 Ksh. for each text you send this week, up to Ksh. 250.' In the long run, individuals may identify new, easier ways to manipulate these indicators. To mimic this, we held focus groups

[20] As some participants may upload data sparsely throughout the week, only those who upload within the 21-hour window at the end of the challenge-week (between 1pm Tuesday and 10am Wednesday) will be counted as having fully uploaded all of their weekly data.

[21] In these 'control' weeks, the subject receives a challenge of the form, 'Dear user, you do not have to do anything for this week's challenge. You will receive an extra Ksh 50 for accepting this challenge.' Our method could also be used without these control weeks, as long as there is variation in incentives between weeks; one would then need to net out the manipulation in estimation.

Table 2: Behavior Predicts Individual Characteristics

	Monthly Income		**Intelligence**		**Activity PCA**	
OLS			(Ravens)			
Average Duration of Workday Calls	-6.877	(0.471)	0.0009	(0.6)	-0.0007	(0.185)
Average Duration of Outgoing Calls	5.746	(0.584)	-0.0005	(0.815)	0.0003	(0.607)
Calls with Non-Contacts	-27.747	(0.005)***	-0.006	(0.001)***	0.0002	(0.649)
# Unique Evening Text Contacts	102.477	(0.129)	0.016	(0.196)	0.003	(0.435)
Incoming Call Count	14.962	(0.065)	0.001	(0.416)	0.005	(0.0)***
Evening Text Count	-5.904	(0.194)	-0.0007	(0.399)	-0.0002	(0.322)
Average Duration of Evening Calls	-1.739	(0.637)	0.0004	(0.614)	0.0007	(0.703)
Minimum Duration of Weekend Calls	2.950	(0.874)	0.003	(0.406)	-0.0008	(0.935)
Outgoing Texts on Weekdays	-7.130	(0.417)	-0.002	(0.225)	-0.0001	(0.791)
Outgoing Text Count	3.666	(0.621)	0.0008	(0.585)	0.001	(0.001)***
Outgoing Call Count	14.556	(0.004)***	-0.001	(0.14)	0.004	(0.0)***
Incoming Text Count	1.762	(0.6)	0.002	(0.013)**	0.001	(0.0)***
Intercept	5259.547	(0.0)***	5.071	(0.0)***	-0.956	(0.0)***
N	1539		1557		1415	
R2	0.0241		0.0223		0.7593	
OTHER MODELS						
LASSO: 3 covariate model, 10-fold CV R2	0.0180		0.0044		0.6173	

Notes: Each column indicates a different prediction target. P-values in parentheses. N represents individuals. 10-fold cross-validated R2 is reported for a LASSO regression where the regularization parameter is set in order to achieve a 3-covariate model.

to identify the most effective ways to manipulate different features, and during onboarding, exposed each participant to a discussion of how one could change different types of behavior (this is similar to hiring 'white hat' hackers to uncover security weaknesses).

People response to these challenges, as anticipated by our theory (Equation 2). For intuition, Table 3 shows how behavior changed in response to simple challenges. Each column shows a regression of an outcome on different incentives (randomly assigned). Individuals manipulate the particular behaviors that were incentivized, as shown by the diagonal, which is positive and significant for these outcomes. Incentivizing one behavior also affects others, as shown in the off diagonal elements. For example, incentivizing missed incoming calls also increases the number of texts sent (presumably requests to contacts to be called). Our method can theoretically exploit these cross elasticities.

Since we have a limited sample on which to estimate costs, our challenges focus on incentivizing a subset of K focal behaviors (from the full set of $\bar{K}$). Specifically, we select behaviors $\underline{\mathbf{x}}^C$ that are useful in predicting the set of user characteristics that form the basis for our 'complex' challenges. To identify this subset, we run LASSO regressions for each $\mathbf{y}$ to induce variable selection, and include the selected variables $\{\underline{\mathrm{x}}_k \,|\beta_k^{naive} \neq 0\}$. For each of these variables, we pair an additional behavior that measures a similar concept but which we anticipate may be differently easy to manipulate (for example, if a naïve regression selects outgoing calls, we will also include the variable incoming calls).[22] Note that by including only a subset of variables, our procedure implicitly assumes that omitted variables are costless to manipulate (and therefore should not be included in any decision rule); we will thus underestimate the performance that could be attained with our method if costs were fully estimated.[23] In Section 5, we evaluate other potential methods to lower the expense of measuring manipulation costs.

[22]We determined "similar" behaviors as those that met at least one of the following conditions: (1) correlated with the primary behavior with a coefficient of at least 0.75; (2) was a 'close cousin' of the primary behavior, in that it was a different transformation of a similar underlying behavior (e.g., for 'weekly number of late-night calls', 'maximum number of late-night calls in a single day' would be considered a close cousin); (3) a cross validated LASSO regression that excluded the principal behavior from the feature set then newly picked out this variable in its optimal set. From this list of similar behaviors, we picked alternates based on our intuition of which behaviors would substitute the best, and which would be the easiest to explain in a challenge.

[23]Note that this procedure will perform poorly if baseline predictiveness and manipulation cost are highly negatively correlated: in that case we may omit a behavior k which is less predictive at baseline but is more predictive in the counterfactual because it is difficult to manipulate.

Table 3: Behavior Changes when Incentivized

	BEHAVIOR OBSERVED				
	# Texts sent	**# Missed calls** (outgoing)	**# Missed calls** (incoming)	**# People called (workday)** (M-F, 9am-5pm)	**# Calls w non-contacts (weekend)**
	change in actions per ¢ of incentive				
BEHAVIOR INCENTIVIZED					
# Texts sent	**24.508**	-0.052	-0.836	-0.305	-0.022
	(0.0)***	(0.929)	(0.337)	(0.161)	(0.953)
# Missed Outgoing Calls	4.16	**0.709**	0.825	0.128	-0.002
	(0.058)*	(0.079)*	(0.167)	(0.391)	(0.995)
# Missed Incoming Calls	-0.206	0.324	**1.187**	0.22	0.502
	(0.942)	(0.536)	(0.126)	(0.255)	(0.126)
# People Called during Workday	2.307	0.156	0.68	**0.497**	0.108
	(0.357)	(0.734)	(0.318)	(0.003)***	(0.708)
# Calls w Non-Contacts on Weekend	-2.022	-0.056	1.234	0.015	**1.233**
	(0.481)	(0.916)	(0.113)	(0.94)	(0.0)***
Week and Individual Fixed Effects	X	X	X	X	X
N (person-weeks)	7976	7976	7976	7976	7976
R2	0.705	0.637	0.552	0.604	0.491

Notes: P-values in parentheses. Bold indicates diagonal: effect on behavior j when behavior j is incentivized. N represents person-weeks when no "incentive challenge" was assigned to the given participant. Individual and weekly fixed effects included, excluding the first week and first individual hash. Each column represents a separate regression, over the full set of covariates assigned; only the first five coefficients reported here. * p < 0.1, ** p < 0.05, *** p < 0.01.

Estimation

Finally, we use the data from all weeks of the experiment to jointly estimate types and manipulation costs (using GMM with the moment conditions outlined in Section 3.1). We allow manipulation cost to differ by behavior, by whether a person reports having high tech skills, and by an unobserved random effect by person.[24] Table 4 summarizes these estimated costs. With our sample size, we find that off diagonal elements are noisily estimated, so we penalize them to zero ($\lambda^{costs}_{offdiagonal} \to \infty$); this results in a diagonal cost matrix $\mathbf{C}$.

Several intuitive patterns can be discerned from the estimated manipulation costs in the top panel of Table 4 (here we present only behaviors selected by models; see Supplemental Appendix for all estimated costs). Outgoing communications are less costly to manipulate than incoming communications. Text messages, which are relatively cheap to send, are more manipulated than calls, which are relatively expensive. We also find that complex behaviors (such as the standard deviation of talk time; estimated but not shown on this summary diagram) are less manipulable than simpler behaviors (such as the average duration of talk time).

Costs are also heterogeneous across people, as shown in the bottom panel of Table 4. On average it is 10%pt easier for individuals who report advanced or higher tech skills to manipulate their mobile phone behaviors. Overall, including unobserved heterogeneity in gaming ability, the 90th percentile finds it 2.5 times easier to game than the 10th percentile.

4.3 Results: Naive vs. Robust Decisions

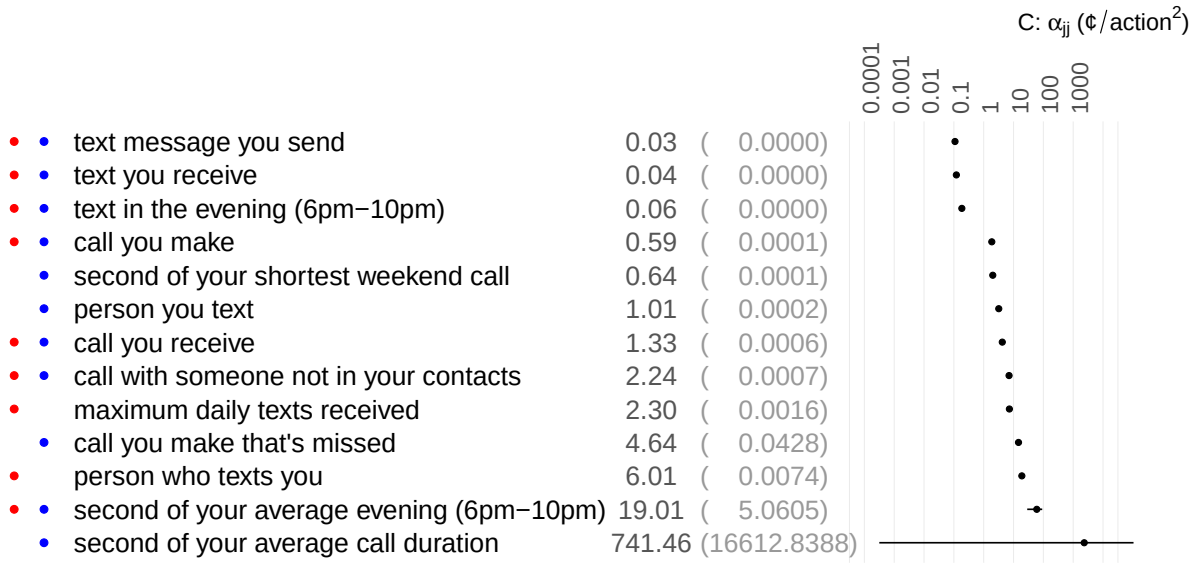
The final and most important stage of the experiment compares decisions made by standard machine learning algorithms to the decisions made by our new strategy-robust estimator that accounts for the cost of manipulating behavior. The robust decision rules can be directly estimated with Equation 1, which relies on the estimates of $\underline{\mathrm{x}}$ and $\mathbf{C}_i$ that come from previous stages of the experiment.

In this final stage, subjects receive complex challenges that reward them for their ultimate classification, of the form 'We'll pay you M if you are classified as $\hat{y}$.' We consider a focal challenge of the form, 'Earn up to 1000 Ksh. if the Sensing app guesses you are a high-income earner.' These challenges are designed to mimic real world applications of machine learning,

[24] We have allowed for a single dimension of observed heterogeneity in costs $\mathbf{z}$; with the rest absorbed into unobserved heterogeneity V. Thus Spence signaling will only be captured in that dimension $\mathbf{z}$. With a larger sample one could estimate a more nuanced functional form for the observable portion, which would better capture the correlations between gaming ability γ_i and bliss behavior $\underline{\mathbf{x}}_i$.

Table 4: Estimated Manipulation Costs

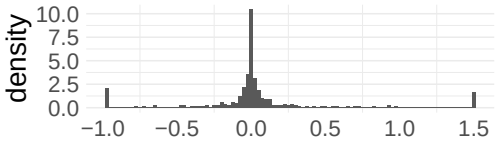
Heterogeneity by Behavior (**C** diagonal; subset of behaviors selected by models)



Heterogeneity by Person (γ_i)

$$\gamma_i = e^{-\omega z_i} + v_i$$

Low tech skills	1.00
High tech skills	1.10



In top panel: Red: used in a LASSO model; blue: used in SR model. Line segment represents standard error. Parameters estimated using GMM. In cost matrix, off diagonal elements α_{jk}; j$\neq k$ regularized to zero ($\lambda^{costs}_{offdiagonal} \to \infty$), diagonal elements regularized with $\lambda^{costs}_{diagonal} = 1.0$, set via cross validation. Standard errors estimated from PD approximation of inverse Hessian. Shown here with v_i winsorized at top and bottom of range; in implementation, only bottom is winsorized, to maintain assumption of non-negative γ_i. Only behaviors selected by models shown in Panel I; for all behaviors see Supplemental Appendix.

where depending on how they are classified, users may receive a loan (digital credit), grant (targeted aid), or other benefits.

Estimating Decision Rules

In order to keep decision rules simple and interpretable for our participants, we consider decision rules of up to three features. We regularize naïve decision rules to three features, selecting $\lambda^{decision} = \max(\lambda^{cv}, \underline{\lambda}^{3var})$, where $\underline{\lambda}^{3var}$ is the smallest hyperparameter that results in a 3 variable LASSO model. We use the same hyperparameter to penalize our strategy robust decision rule, and allow it to select only among three variable models.[25]

Treatments

Participants are randomly assigned into different targets ($\hat{y}$), decision rules (standard: $\boldsymbol{\beta}^{LASSO}$, or robust $\boldsymbol{\beta}^{stable}$), and whether the decision rule is kept opaque or revealed transparently to the user. Under the opaque treatment, users are told only the outcome and the reward. Under the transparent treatment, users see the coefficients of the decision rule, which reveals how much they are rewarded for changing which behaviors. We included an interactive interface that can be used to compute the payments that would result from different behavior (see Figure 2c). Because the transparent treatment reveals information about potential decision rules, after a person has seen a transparent challenge for $\hat{y}$, we do not assign them to an opaque challenge for the same outcome.

Table 5 summarizes the effect of decision rule incentives on behavior. High income people make more outgoing calls, and send fewer texts but receive more. If we pay people to 'act like a high-income earner,' without revealing the decision rule, the response is noisy and often in the wrong direction (participants place fewer calls and send more texts). Participants who are transparently presented with the decision rule change their behavior, closer to the direction incentivized by the algorithm, though the response is still noisy.

Performance of decision rules

We compare performance of naïve vs. robust decision rules in Table 6. The first two columns (under 'Income') show results for the challenge that incentivized participants to use their phones like a high-income earner; the last two columns show the performance averaged across

[25] For a given $\lambda^{decision}$ that selects three variables in a LASSO model, the strategy robust model will tend to select more than three variables, because it induces some penalization on its own. Instead of restricting to three variable models, one could alternately increase $\lambda^{decision}$.

Table 5: Agents Game Algorithms

	# Calls (outgoing)	# Texts (outgoing)	# Texts (incoming)	# Calls w Non-Contacts (incoming + outgoing)	Mean Call Duration (evening, seconds)
Weekly Challenge: ***Use your phone like a high-income earner!***					
Panel I:	**Incentives Generated by Algorithm** (¢/action)				
$\boldsymbol{\beta}^{LASSO}$	0.625	-0.395	0.065	0	0
Panel II: $\mathbf{x}_{it}$					
Assigned to challenge,	-6.5573	14.3701	12.0135	1.1672	-6.8104
algorithm opaque	(9.949)	(16.405)	(20.583)	(3.473)	(7.002)
Assigned to challenge,	11.8231	-15.69	-11.907	0.6706	-4.5744
algorithm transparent	(9.083)	(14.976)	(18.79)	(3.17)	(6.392)
N (Person-weeks)	1664	1664	1664	1664	1664

Notes: The first panel reports the decision rule associated with the challenge. The second reports the results of a regression of behavior on challenge assignment. Regressions estimated based on dummy indicators for complex challenge assignment for participants assigned "income" challenge, over person-weeks when the income challenge was assigned or when no challenge was assigned ("control" weeks). Simple challenge assignment person-weeks, used in estimating costs, are not included. Standard errors in parentheses.

several different challenges. The decision rules and associated manipulation costs are shown in the top panel ("Decision Rule"); the relative performance of the different estimators is shown below (under "Prediction Error"). We note several results.

First, in the top panel, we observe important differences in the decision rules estimated by β^{LASSO} vs. β^{act}. LASSO places weight on the behaviors that were most correlated at baseline: outgoing calls, outgoing texts, and incoming texts. However, the estimated costs of manipulating some of these behavior – and in particular the costs of manipulating text messaging behavior – are low, and therefore likely to be manipulated when incentivized. Thus, our strategy robust decision rule both selects less manipulable behaviors (evening texts rather than incoming texts), and shrinks manipulable behaviors (especially outgoing texts).

We evaluate prediction error using root mean squared error (RMSE), in units of dollars, in the middle panel. The magnitude of error is similar to the average payout, around $3 for a week. The first row shows prediction error in the baseline data: LASSO performs slightly better than our strategy robust estimator when no manipulation is expected. But when people manipulate their behavior, our method is expected to perform better, as shown in the second row.

When actually implemented, our method performs better when the decision rule is

Table 6: Strategy Robust vs. Standard Decision Rules

	Income		*Costs*	**All Outcomes (Pooled)**	
				Income, Intelligence, Activity PCA	
Decision Rule	$\boldsymbol{\beta}^{LASSO}$	$\boldsymbol{\beta}^{stable}$	α_{jj}		
	¢/action		¢/action^2		
# Calls (outgoing)	0.625	0.542	0.591	.	.
# Texts (outgoing)	-0.395	-0.107	0.035	.	.
# Texts (incoming)	0.065	0	0.038	.	.
# Texts (6pm-10pm)	0	-0.121	0.058	.	.
Prediction Error	RMSE ($)			RMSE ($)	
Baseline Data: Control	3.55	3.55		3.70	3.75
Baseline Data: Predicted Transparent	4.66	3.83		4.34	3.85
Implemented: Opaque	3.24	3.23		4.00	3.80
Implemented: Transparent	3.87	3.66		4.93	4.31
Predicted Cost of Transparency		≤0.28			≤0.15
Equilibrium Cost of Transparency		≤0.41			≤0.31
Average Payout ($)	3.30	3.24		3.23	2.98
N (Control Person-Weeks)	3781	3781		3781	3781
N (Treatment Person-Weeks, Opaque)	85	85		230	230
N (Treatment Person-Weeks, Trans.)	91	74		252	216

Notes: The first panel reports the decision rule associated with the challenge, and the costs associated with these behaviors. The second reports the performance of the different models over the groups they were assigned to; on the left, the naive LASSO regression, and on the right, this paper's strategy-robust (SR) model. Performance figures estimated using a regression of model indicators on week-model RMSE, weighted by number of person-weeks. 'Transparent Predicted' RMSE denotes the RMSE that our theoretical model expected, given costs of manipulation and behavioral incentives. 'Predicted Cost of Transparency' denotes the difference between predicted transparent RMSE under the SR model and baseline RMSE under the naive LASSO. 'Equilibrium Cost of Transparency' denotes the difference between implemented transparent SR model RMSE and opaque naive model RMSE. Pooled performance is estimated using this same regression approach, after combining all model-weeks over the three outcomes investigated: a PCA of phone activity, intelligence, and monthly income. Full regression results and standard errors reported in appendix.

transparent (average error \$3.66 instead of \$3.87 for income; or \$4.31 vs. \$4.93 for all outcomes pooled). When the decision rule is opaque, we find that our method performs comparably to or slightly better than LASSO, possibly due to increased shrinkage (\$3.23 vs. \$3.24 for income; \$3.80 vs. \$4.00 for all outcomes pooled). Table A3 reports results for all outcomes.

Even if a policymaker intended to keep the decision rule opaque, using our robust method can reduce systematic risk in the chance that agents discover the decision rule. In practical implementations, policymakers could adaptively tweak the level of robustness to match the level of manipulation. An ad hoc approach could select a convex combination of the naive and robust models; a more nuanced approach could model consumers' uncertainty about the model.

Cost of transparency

Our framework provides a way to bound a key cost of imposing algorithmic transparency (Akyol et al., 2016). Many tech firms argue that imposing transparency would reduce the quality of machine decisions, because rules may perform better if they can rely on opacity to prevent manipulation. Our method allows us to bound this performance cost. We can compare the performance arising from the optimal opaque rule (under the assumption that opacity will prevent it from being manipulated) to the optimal equilibrium transparent rule (factoring in equilibrium manipulation). Because the opaque rule also faces the threat of manipulation, this difference is the upper bound of the performance cost of imposing transparency, arising from increased manipulation.

The most straightforward way to measure this cost of transparency would require disclosing the decision rule to a subset of users, and assessing any drop in performance after a process of equilibration. But for the most consequential decisions, once the decision rule is revealed to some, it can leak out to the entire market. Such disclosure irreversibly tips the market to transparency, and thus is a nonstarter for policy discussions.

Crucially, under the assumptions of our model, this quantity can be estimated without revealing the decision rule: it only requires the estimation of types and costs (the first part of our experiment).[26] Our method makes it possible for regulators or firms to assess the cost that transparency would impose—prior to making their model transparent. Our model based estimates suggest that transparency introduces a performance cost of $\leq$ \$0.28 (8% of baseline

[26] Our method of estimating costs does requires revealing the existence of features to users, but does not require specifying whether those features are included in the model, or with what weights (one could estimate costs for a large set of features, hiding the features critical to the model).

error) for our income targeting rule, or $\leq$ \$0.15 (4%) for all outcomes pooled together. These numbers are shown in the final rows of the middle panel of Table 6.

When we actually implement transparency in our experiment, we find that the performance cost is similar to these model based estimates: $\leq$ \$0.41 (13%) for income, or $\leq$ \$0.31 (8%) for all outcomes pooled together. (To mitigate the problem of leakage, we only assess opaque performance prior to each individual observing a transparent challenge for that outcome. Because our decision rules were not going to be used later in production, we were unconcerned about them leaking out after the experiment.)

5 Extensions

5.1 Alternate methods to estimate manipulation costs

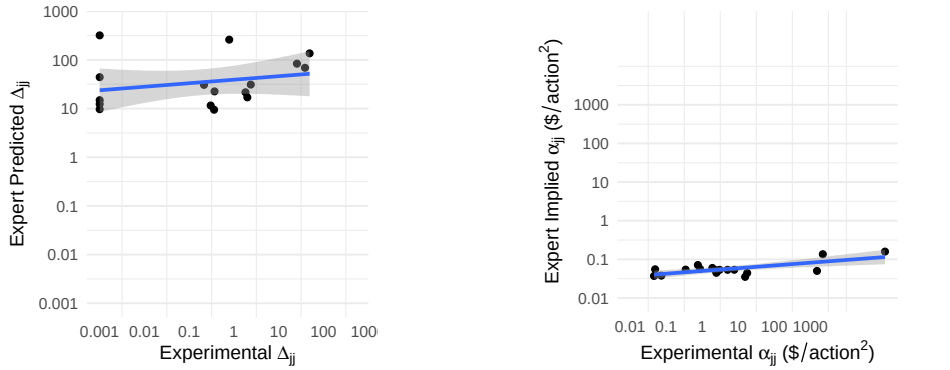
Our method requires estimating C_i, a new object. The experimental approach we use may not be feasible in some settings. We offer suggestions on alternative approaches to measure these costs.

Expert elicitations. We evaluate how well experts can predict the costs of manipulating different behaviors, using a method similar to DellaVigna and Pope (2016). We sent a survey to 177 experts with different backgrounds (PhDs from different fields, research assistants, Busara staff who had not worked on the experiment, and Mechanical Turk workers in the US) to predict how Kenyans would manipulate different phone behaviors when incentivized. Results are shown in Figure 3. In Panel A, we compare the predicted change in behavior from a given incentive to the actual experimental estimate ($\Delta_{jj} \coloneqq x_j(\beta_j) - \bar{x}_j$). In Panel B, we compare the implied structural cost estimates (for predicted costs $\hat{\alpha}_{jj} = \frac{\bar{\gamma} \cdot \beta_j}{\Delta_{jj}}$); although experts predict that costs are too low, the correlation is 0.75. This suggests that it may be possible to use expert elicitations to estimate manipulation costs.

First principles/structural approach. In some cases, it can be straightforward to build up the cost of underlying manipulations from first principles. For example, one can increase the number of noncontacts spoken with by randomly dialing 10 digit numbers and hanging up after the recipient picks up. That costs the call price of \$0.04/minute plus the value of the time to dial a 10 digit number, divided by the fraction of such numbers that are valid and pick up, which can be valued at the going wage. Or, likewise for text messages (with a price of \$0.01/message). One can model different such strategies, and account for the cost to open a new phone number, send a robocall using standard providers, use labor from a service like

Figure 3: Expert Elicited Manipulation Cost Estimates

(a) Reduced Form Shift in Behavior **(b)** Structural Cost Estimates



For structural costs we set $\bar{\gamma} = 1$ and $\hat{\alpha}_{jj} = \frac{\bar{\gamma} \cdot \beta_j}{\max(0.001, \Delta_{jj})}$.

Mechanical Turk, etc.[27] A structural model of costs would allow an implementer to account for changes in these underlying parameters, suggesting how manipulation will change if for example, the telecom reduces the price of calls, or a service emerges that makes it easy to generate incoming calls. This approach can work well for mechanical manipulations, but is complicated for marginal manipulations around the status quo that have social consequences. For example, a marginal call to a contact may be valued by both parties, but additional calls may lead them to annoyance: that annoyance may form an important portion of the cost.

Partially estimated. The costs of behavior k may be related to that of behavior k'. Because of this, we may be able to predict unknown cost α_{kj} based on correlations between types $\underline{x}$ and known costs, for some prediction function: $\hat{\alpha}_{kj} = f(C, \underline{x})$.

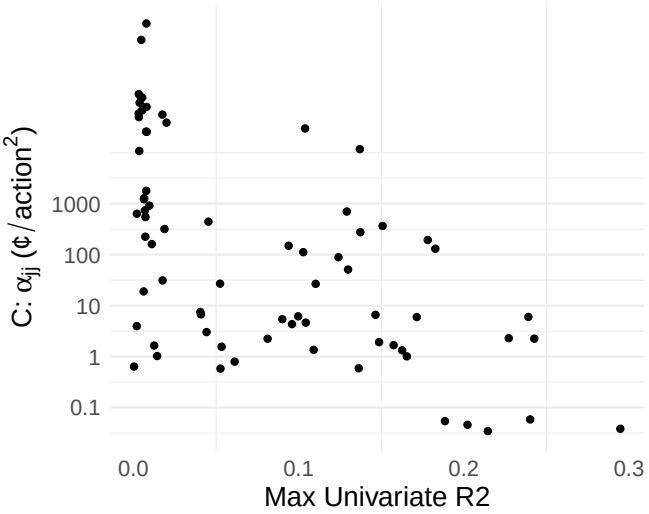
5.2 Nonlinear decision rules

To sharpen intuition, this paper focuses on linear decision rules, but our core insight is also relevant in nonlinear settings. Figure 4 plots the cost of manipulation of each feature against its highest univariate R^2 with one of our outcomes. Features that appear equally predictive in baseline data are differentially manipulable, and thus in equilibrium will be differentially predictive when included in a decision rule subject to manipulation. In fact, in our setting, there is a negative correlation: the features that are most predictive at baseline also appear easiest to manipulate.

Our method demonstrates how to trade these off in a linear setting, but the same insight can be extended to nonlinear settings.

[27] Thanks to Jon Bittner for these ideas.

Figure 4: Manipulation Costs vs. Baseline Predictive Power



Each feature is represented by a dot, showing its highest R^2 across outcomes and estimated manipulation cost.

Discrete outcome variables. If outcomes are binary or discrete, then each agent's incentives to manipulate depend on their distance to the classification threshold. Agents near the threshold have higher incentives to manipulate their behavior. Agents must have some belief not only about the shape of the model, but about how close they are to the threshold.

General nonlinear rules. While many modern machine learned decision rules are nonlinear, agents' beliefs about those rules may be well approximated by linear functions. In such a context, our derivations could be viewed as linear approximations to both these beliefs, and the actual functions. Additionally, it may be that some benefits of extreme nonlinearities that can surface in modern machine learning are lessened when manipulation is taken into account: contract theory suggests that linear decision rules are more robust (Holmstrom and Milgrom, 1987; Carroll, 2015).[28] In nonlinear environments there may also be many equilibria. In such a setting, if iterative learning converges, it may converge to an undesirable equilibrium, whereas an approach like ours could be used to select a global optimum.[29]

5.3 Decision rules that are imperfectly known or nondeterministic

Our model considers the case where agents know the decision rule perfectly. In practice, agents are likely to have noisy beliefs: even when the decision rule is revealed, it can be

[28] With the exception that linear models can be subject to the influence of outliers; one may thus want to tamp down inputs as they approach the boundaries of the distribution of training data.

[29] Thanks to Glen Weyl for this point.

difficult to interpret (Freitas, 2014); and if it is kept secret, agents may still be able to guess some of its properties.[30] Our model can be extended with a model of belief formation; to obtain the result that noise in beliefs lowers manipulation, one would need to add risk aversion to the agent's utility function. In some settings opacity may be optimal (for example, Ederer et al. (2018)). Beliefs will also ultimately depend on how knowledge is transmitted through population, including the ability for middlemen to capitalize on exploits, or fraud to scale.

One could also introduce noise directly into the decision rule in order to lower incentives to manipulate. Randomness undermines some of the goals of interpretability, but may be appropriate in some settings (for example, in enforcement of drunk driving checkpoints Banerjee et al. (2019)).

Even if misunderstood rules generate better predictions, they can still induce manipulation that is costly from a social perspective (which may be factored into the decisionmaker's objective in the term $M(\cdot)$).

5.4 Alternate forms of costs

For simplicity we have modeled the cost of manipulation as having a symmetric quadratic form, which we believe is a reasonable approximation for our setting. In general, manipulation costs may include fixed components (e.g., the cost of setting up a spoofing app), asymmetries (the cost of installing an app differs from that of deleting it), changes over time (seasonality), or be affected by policy themselves (e.g., the cost of calls). A decisionmaker may have some ability to endogenously affect the cost of manipulation.[31]

Additionally, there may be costs of learning the decision rule, which can be separately modeled, and will affect belief formation.

5.5 Manipulations that have a causal effect on outcomes

In some settings, behaviors $\mathbf{x}_i$ may have a causal impact on the outcome of interest y_i (Kleinberg and Raghavan, 2018; Milli et al., 2019). For example, a student who 'manipulates' their grades (x_{i1}) to get into college by studying may become a better candidate (y_i). These causal impacts can differ by behavior: the other hand, manipulating an SAT score (x_{i2}) with

[30] For example, even in our opaque treatments, participants may have been able to guess the relationship between income, calls, and texts.

[31] Additionally, due to power we treat as independent two dimensions of heterogeneity in manipulation cost (by individual and by feature). As suggested by Ball (2019), there may be particular features that have more heterogeneity in cost between individuals. If one extended our specification to allow this it would downweight indicators that have a particular spread in manipulability.

test prep may have a weaker causal effect. Our framework can be extended to such settings by modeling the outcome as a function of behavior: $y_i(\mathbf{x}_i)$.

5.6 Greenfield vs. brownfield implementations

Like our study, new applications of machine learning are typically trained in *greenfield* settings, using baseline data that was not incentivized, and not manipulated. Models trained in new settings can be acutely susceptible to manipulation, as baseline data does not expose evidence of manipulation. In greenfield settings, during training it is possible to infer individual types directly from baseline data (Equation 3).

Our method can also be applied in *brownfield* settings, where an implementation has already been implemented and baseline behavior is already manipulated. One would still need to observe variation in incentives (and corresponding manipulation) to back out costs. One would obtain the underlying types by inverting the observed behavior under the current incentive (this can be done by solving the joint moment conditions, omitting Equation 3). This inversion will be more sensitive to the specification of the model than when unincentivized behavior can be observed directly in training.

6 Conclusion

This paper considers the possibility that the implementation of machine decisions changes the world they describe. We focus on the case where individuals manipulate their behavior in order to game decision rules. Our chief contribution is to derive decision rules that anticipate this manipulation, by embedding a behavioral model of how individuals will respond. This structural approach makes it possible to decompose decision rules into constituent components, and to gather data on how those components can be manipulated. From these components, our structural model allows us to understand how *any* proposed decision rule of a given form would be manipulated. This allows us to compute decision rules that are optimal in equilibrium.

We demonstrate our method in a field experiment in Kenya, by deploying a tailor-made smartphone app that mimics the 'digital credit' loan products that are now commonplace in sub-Saharan Africa. We find that even some of the world's poorest users of technology – who are relatively recent adopters of smartphones and to whom whom the concept of an 'algorithm' is quite foreign (Musya and Kamau, 2018) – are savvy enough to change their behavior to game machine decisions. In this setting, we show that our strategy robust

estimator outperforms standard estimators on average by 13% when individuals are given information about the scoring rule. This framework also allows us to quantify the "cost of transparency", i.e., the loss in predictive performance associated with moving from "security through obscurity" (with a naive decision rule) to a regime of full algorithmic transparency (with our strategy-robust rule). We estimate this loss to be roughly 8% in equilibrium – substantially less than the 23% loss associated with making the naive rule transparent.

Our discussion focuses on the simple case of linear models with a small number of predictor variables, where subjects have either no information or full transparency of the scoring rule. We envision useful extensions to more complex models and more nuanced beliefs. More generally, our approach of embedding a model of behavior within a machine learning estimator may be relevant to a wide range of contexts where machine learning systems face a changing human environment. In this sense, it offers a machine learning interpretation of Lucas (1976), where algorithmic decisions change the context of the systems they model. For example, financial forecasts may affect the underlying financial processes they attempt to describe, personalized news recommendations may change the information seeking behaviors of consumers, and predictions about the intensity of a disease may affect individuals' protective behaviors and thus its realized intensity.

References

Akyol, Emrah, Cedric Langbort, and Tamer Basar, "Price of Transparency in Strategic Machine Learning," *arXiv:1610.08210 [cs]*, October 2016. arXiv: 1610.08210.

Alatas, Vivi, Abhijit Banerjee, Rema Hanna, Benjamin A. Olken, Ririn Purnamasari, and Matthew Wai-Poi, "Self-Targeting: Evidence from a Field Experiment in Indonesia," *Journal of Political Economy*, March 2016, *124* (2), 371–427.

Ball, Ian, "Scoring Strategic Agents," *arXiv:1909.01888 [econ]*, November 2019. arXiv: 1909.01888.

Banerjee, Abhijit, Esther Duflo, Daniel Keniston, and Nina Singh, "The Efficient Deployment of Police Resources: Theory and New Evidence from a Randomized Drunk Driving Crackdown in India," Working Paper 26224, National Bureau of Economic Research September 2019. Series: Working Paper Series.

_ , Rema Hanna, Benjamin A Olken, and Sudarno Sumarto, "The (lack of) Distortionary Effects of Proxy-Means Tests: Results from a Nationwide Experiment in Indonesia," Working Paper 25362, National Bureau of Economic Research December 2018.

Barocas, Solon, Moritz Hardt, and Arvind Narayanan, *Fairness and Machine Learning*, fairmlbook.org, 2018.

Björkegren, Daniel, "'Big data' for development," 2010.

_ and Darrell Grissen, "The Potential of Digital Credit to Bank the Poor," *American Economic Association Papers and Proceedings*, 2018.

_ and _ , "Behavior Revealed in Mobile Phone Usage Predicts Credit Repayment," *The World Bank Economic Review*, 2019.

Bloomberg, "Phone Stats Unlock a Million Loans a Month for Africa Lender," *Bloomberg.com*, September 2015.

Blumenstock, Joshua E., "Estimating Economic Characteristics with Phone Data," *AEA Papers and Proceedings*, 2018, *108*, 72–76.

Blumenstock, Joshua Evan, Dan Gillick, and Nathan Eagle, "Who's Calling? Demographics of Mobile Phone Use in Rwanda," in "2010 AAAI Spring Symposium Series" March 2010.

_ , Gabriel Cadamuro, and Robert On, "Predicting poverty and wealth from mobile phone metadata," *Science*, November 2015, *350* (6264), 1073–1076.

Borrell Associates, "Trends in Digital Marketing Services," 2016.

Bruckner, Michael and Tobias Scheffer, "Stackelberg Games for Adversarial Prediction Problems," in "Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining" KDD '11 ACM New York, NY, USA 2011, pp. 547–555.

Brynjolfsson, Erik and Tom Mitchell, "What can machine learning do? Workforce implications," *Science*, 2017, *358* (6370), 1530–1534.

Camacho, Adriana and Emily Conover, "Manipulation of Social Program Eligibility," *American Economic Journal: Economic Policy*, May 2011, *3* (2), 41–65.

Carroll, Gabriel, "Robustness and Linear Contracts," *American Economic Review*, February 2015, *105* (2), 536–563.

CGAP, "Kenya's Digital Credit Revolution Five Years On," *CGAP*, March 2018.

Dee, Thomas S., Will Dobbie, Brian A. Jacob, and Jonah Rockoff, "The Causes and Consequences of Test Score Manipulation: Evidence from the New York Regents Examinations," *American Economic Journal: Applied Economics*, July 2019, *11* (3), 382–423.

DellaVigna, Stefano and Devin Pope, "Predicting Experimental Results: Who Knows What?," Working Paper 22566, National Bureau of Economic Research August 2016.

Dong, Jinshuo, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu, "Strategic Classification from Revealed Preferences," in "Proceedings of the 2018 ACM Conference on Economics and Computation" EC '18 ACM New York, NY, USA 2018, pp. 55–70.

Dranove, David, Daniel Kessler, Mark McClellan, and Mark Satterthwaite, "Is More Information Better? The Effects of "Report Cards" on Health Care Providers," *Journal of Political Economy*, June 2003, *111* (3), 555–588.

Ederer, Florian, Richard Holden, and Margaret Meyer, "Gaming and strategic opacity in incentive provision," *The RAND Journal of Economics*, 2018, *49* (4), 819–854.

Eliaz, Kfir and Ran Spiegler, "Incentive-Compatible Estimators," 2018.

European Union, "EU General Data Protection Regulation (GDPR)," 2016.

Eykholt, Kevin, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song, "Robust Physical-World Attacks on Deep Learning Models," *arXiv:1707.08945 [cs]*, April 2018. arXiv: 1707.08945.

Francis, Eilin, Joshua Blumenstock, and Jonathan Robinson, "Digital Credit: A Snapshot of the Current Landscape and Open Research Questions," *CEGA White Paper*, 2017.

Frankel, Alex and Navin Kartik, "Muddled Information," *Journal of Political Economy*, August 2019, *127* (4), 1739–1776.

_ **and** _ , "Improving Information from Manipulable Data," *arXiv:1908.10330 [econ]*, April 2020. arXiv: 1908.10330.

Freitas, Alex A., "Comprehensible Classification Models: A Position Paper," *SIGKDD Explor. Newsl.*, March 2014, *15* (1), 1–10.

Frias-Martinez, Vanessa, Enrique Frias-Martinez, and Nuria Oliver, "A Gender-Centric Analysis of Calling Behavior in a Developing Economy Using Call Detail Records," in "2010 AAAI Spring Symposium Series" March 2010.

FSD Kenya, "Tech-enabled lending in Africa," 2018.

Gonzalez-Lira, Andres and Ahmed Mobarak, "Slippery Fish: Enforcing Regulation under Subversive Adaptation," IZA Discussion Paper 12179, Institute of Labor Economics (IZA) February 2019.

Goodhart, Charles, *Monetary Relationships: A View from Threadneedle Street*, University of Warwick, 1975. Google-Books-ID: GKwJMwEACAAJ.

Goodman, Bryce and Seth Flaxman, "European Union regulations on algorithmic decision-making and a "right to explanation"," *arXiv:1606.08813 [cs, stat]*, June 2016. arXiv: 1606.08813.

Greenstone, Michael, Guojun He, Ruixue Jia, and Tong Liu, "Can Technology Solve the Principal-Agent Problem? Evidence from Pollution Monitoring in China," 2019.

Hand, D. J. and W. E. Henley, "Statistical Classification Methods in Consumer Credit Scoring: A Review," *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 1997, *160* (3), 523–541. Publisher: [Wiley, Royal Statistical Society].

Hanna, Rema and Benjamin A. Olken, "Universal Basic Incomes versus Targeted Transfers: Anti-Poverty Programs in Developing Countries," *Journal of Economic Perspectives*, November 2018, *32* (4), 201–226.

Hardt, Moritz, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters, "Strategic Classification," in "Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science" ITCS '16 ACM New York, NY, USA 2016, pp. 111–122.

Holmström, Bengt, "Moral Hazard and Observability," *The Bell Journal of Economics*, 1979, *10* (1), 74–91.

Holmstrom, Bengt and Paul Milgrom, "Aggregation and Linearity in the Provision of Intertemporal Incentives," *Econometrica*, 1987, *55* (2), 303–328.

Hu, Lily, Nicole Immorlica, and Jennifer Wortman Vaughan, "The Disparate Effects of Strategic Manipulation," *Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT* '19*, 2019, pp. 259–268. arXiv: 1808.08646.

Hussam, Reshmaan, Natalia Rigol, and Benjamin N. Roth, "Targeting High Ability Entrepreneurs Using Community Information: Mechanism Design in the Field," November 2017.

Kleinberg, Jon and Manish Raghavan, "How Do Classifiers Induce Agents To Invest Effort Strategically?," *arXiv:1807.05307 [cs, stat]*, July 2018. arXiv: 1807.05307.

_ **and** _ , "How Do Classifiers Induce Agents to Invest Effort Strategically?," in "Proceedings of the 2019 ACM Conference on Economics and Computation" EC '19 ACM New York, NY, USA 2019, pp. 825–844. event-place: Phoenix, AZ, USA.

_ , **Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan**, "Human Decisions and Machine Predictions," *The Quarterly Journal of Economics*, February 2018, *133* (1), 237–293.

Lucas, Robert E., "Econometric policy evaluation: A critique," *Carnegie-Rochester Conference Series on Public Policy*, January 1976, *1* (Supplement C), 19–46.

McCaffrey, Mike, Olivia Obiero, and George Mugweru, "M-Shwari: Market Reactions and Potential Improvements," Technical Report 139 2013.

Milli, Smitha, John Miller, Anca D. Dragan, and Moritz Hardt, "The Social Cost of Strategic Classification," in "Proceedings of the Conference on Fairness, Accountability, and Transparency" FAT* '19 ACM New York, NY, USA 2019, pp. 230–239. event-place: Atlanta, GA, USA.

Mirrlees, J. A., "An Exploration in the Theory of Optimum Income Taxation," *The Review of Economic Studies*, 1971, *38* (2), 175–208.

Musya, Mercy and Grace Kamau, "How do you say "algorithm" in Kiswahili?," December 2018. Library Catalog: medium.com.

National Institute of Standards and Technology, "Guide to General Server Security," *NIST Special Publication*, July 2008, (800-123).

Nichols, Albert L. and Richard J. Zeckhauser, "Targeting Transfers through Restrictions on Recipients," *The American Economic Review*, 1982, *72* (2), 372–377.

Niehaus, Paul, Antonia Atanassova, Marianne Bertrand, and Sendhil Mullainathan, "Targeting with Agents," *American Economic Journal: Economic Policy*, 2013, *5* (1), 206–238.

Perlich, Claudia, Brian Dalessandro, Troy Raeder, Ori Stitelman, and Foster Provost, "Machine learning for targeted display advertising: Transfer learning in action," *Machine learning*, 2014, *95* (1), 103–127.

Ramsey, F. P., "A Contribution to the Theory of Taxation," *The Economic Journal*, 1927, *37* (145), 47–61.

Sayed-Mouchaweh, Moamar and Edwin Lughofer, *Learning in Non-Stationary Environments: Methods and Applications*, Springer Science & Business Media, April 2012. Google-Books-ID: qFWM2nva7xQC.

Spence, Michael, "Job Market Signaling," *The Quarterly Journal of Economics*, 1973, *87* (3), 355–374.

Sundsøy, Pål, Johannes Bjelland, Bjørn-Atle Reme, Eaman Jahani, Erik Wetter, and Linus Bengtsson, "Estimating individual employment status using mobile phone network data," *arXiv:1612.03870 [cs]*, December 2016. arXiv: 1612.03870.

Wei, Yanhao, Pinar Yildirim, Christophe Van den Bulte, and Chrysanthos Dellarocas, "Credit Scoring with Social Network Data," *Marketing Science*, October 2015, *35* (2), 234–258. Publisher: INFORMS.

Appendices

A1 Experimental Design

A1.1 Pre Analysis Plan

This study was pre-registered with the AEA RCT Registry (AEARCTR-0004649) prior to the experiment (September 3, 2019).[32]

Our implementation deviated in several respects from the pre analysis plan: at the start of phase 2 the cloud server account ran out of storage space, and the Busara center was hit by a power outage due to construction on a nearby road. These two events disrupted servers for several hours during the upload window, and caused some participants' phones to become overloaded with records. It took several weeks to recover the affected participants. Because of the disruption, we extended phase 2 and delayed the expert cost surveys.

A1.2 Attrition Management

Attrition in the context of this study had two dimensions: first, there were participants who do not regularly upload data through their app, and second, there were participants who did not participate in the assigned weekly challenges. (As some participants may have uploaded data sparsely throughout the week, only those who uploaded within the 21-hour window at the end of the challenge-week [between 1pm Tuesday and 10am Wednesday] were counted as having fully uploaded all of their weekly data.)

In order to minimize both such types of attrition, participants were sent regular reminders via text to encourage engagement. Every participant in the study was sent a text every Tuesday at 1pm to remind them to upload their data through the Smart Sensing app.

Additionally, on Wednesday, Thursday and Friday, participants who still had not uploaded data or activated their challenge respectively were contacted by phone and surveyed by the Busara team. Specifically, the protocol was as follows:

[32]Prior to the collection of the main outcomes in phase 2, we amended the registration, adding one sentence that specifies that the focal performance measure will be mean squared error (which corresponds with the objective minimized by the method; January 15, 2020). We later noticed that the registration still contained text in another section that appeared to specify that the focal measures would be R^2 or AUC; prior to the completion of phase 2 and prior to analysis of the main outcomes, we amended the registration to delete that sentence (February 4, 2020).

- On Wednesday, participants who had not uploaded any data during the five day period ending on Wednesday at 12pm were contacted and surveyed, as were those who uploaded some data in this period but not during the 'end-of-week upload window' (between 6pm Tuesday and 10am Wednesday)

- On Thursday, participants whose phones showed that they did not receive a challenge by Thursday 12pm were contacted and surveyed, as were participants whose phones show that they did receive a challenge but who had not opted in to accept the challenge.

- On Friday, participants whose phones showed that they still had not received and opted-in to a challenge were contacted and surveyed, as were participants whose phones showed that they did receive a challenge but who had not opted in to accept the challenge.

For all of the above categories, any participant who did not answer a call on the first attempt would be re-contacted once more by the surveyor after the rest of the calls were complete.

Finally, to mitigate the effects of attrition during the analysis stage, any participant-weeks wherein the participant did not opt in and/or did not upload during the end-of-week upload window were dropped from the sample prior to all analysis. During baseline weeks, a single passive challenge was assigned to all participants, offering a flat bonus to upload data within the upload window; in this way, we ensured that our analysis control groups would also be restricted to those who opt in to this passive challenge, and were thus a valid comparison group to the restricted panel during the challenge weeks.

A1.3 Communicating Decision Rules

In focus groups we found that individuals had difficulty understanding decimals or complicated mathematical operations (e.g., standard deviation). We stuck to simple behaviors and formatted decision rules as follows, to make it easier for participants to understand how their marginal behavior affects their payment:

- Each coefficient is rounded to the nearest integer. If the nearest integer is zero, the denominator was inflated by factors of 10 until it became nonzero. (If the unit was seconds or minutes, the denominator was instead inflated by factors of 60.)

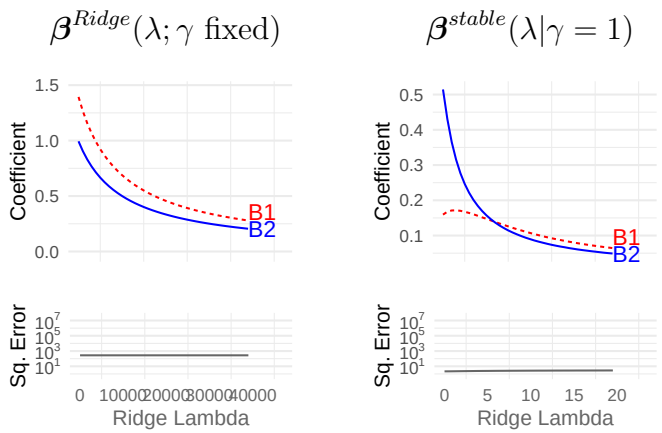
- The order of indicators was randomized between three orderings (ABC, CAB, BCA for indicators A, B, and C).

- The constant term was reported last, unless the first coefficient was negative, in which case the constant was reported first.

A2 Appendix Figures

Figure A1: **Comparative Statics**

C and γ_i heterogeneous
The first behavior is more predictive in the baseline behavior ($b_1 > b_2$), but is easily manipulable ($\alpha_{11} \ll \alpha_{22}$).

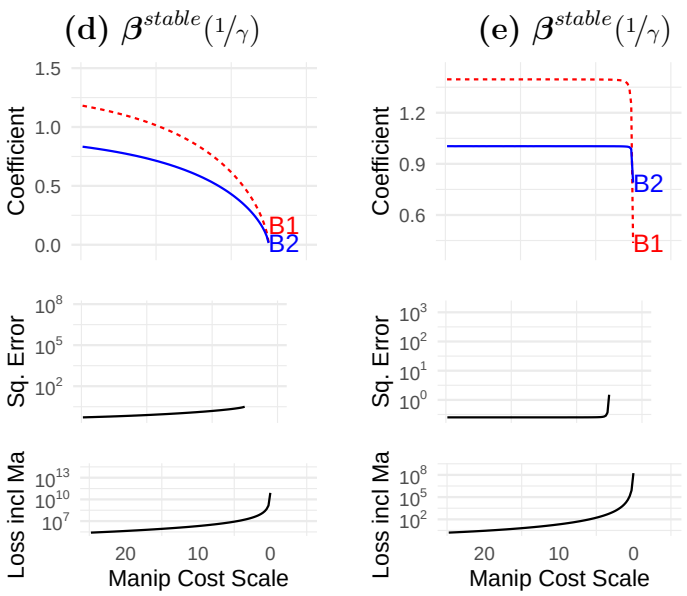


Like LASSO, ridge places more weight on x_1.

Our method can be combined with other forms of penalization (such as ridge shown here), to more finely manage out of sample fit.

C homogenous:
Features equally costly to manipulate

γ_i homogeneous:
Same gaming ability



Manipulation Cost = 1

When features are equally costly to manipulate, our method penalizes in a similar manner to ridge.

When gaming ability is homogenous, everyone shifts behavior equally. Predictive performance remains high, but utility is wasted on manipulation.

$\underset{\sim}{x}_i \overset{iid}{\sim} N(0,1)$, $\mathbf{b} = [1.4, 1]$, $\mathbf{C}^{het} = \frac{1}{\gamma\gamma_i}\begin{bmatrix} 4 & 0 \\ 0 & 32 \end{bmatrix}$, $\mathbf{C}^{hom} = \frac{1}{\gamma\gamma_i}\begin{bmatrix} 8 & 0 \\ 0 & 8 \end{bmatrix}$, $\frac{1}{\gamma_i^{het}} \overset{iid}{\sim} Uniform[0, 10]$, $\gamma_i^{hom} = 5$, $e_i \overset{iid}{\sim} N(0, 0.25)$. Squared error measured on an out of sample draw from the same population, incentivized to that decision rule.

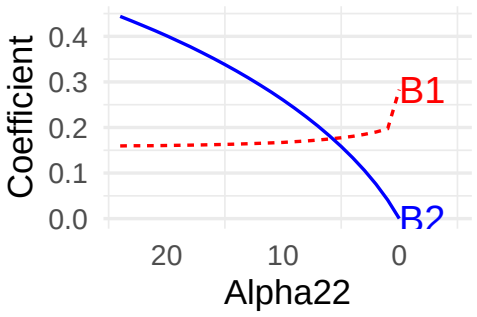
Figure A2: Additional Comparative Statics

Note: The first behavior is more predictive in the baseline behavior ($b_1 > b_2$), but is easily manipulable ($\alpha_{11} \ll \alpha_{22}$). Below panels show weights on coefficients as manipulation costs are scaled for:

(a) x_2

$\boldsymbol{\beta}^{stable}(\alpha_{22})$

$\alpha_{12} = 0$

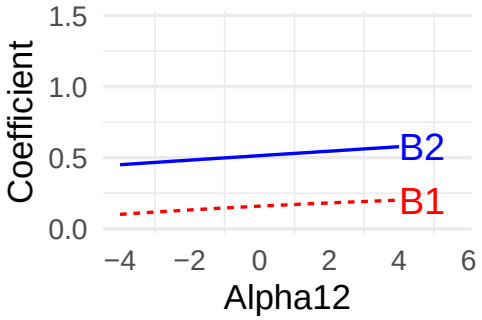


As x_2 becomes cheaper to manipulate (α_{22} decreases), $\boldsymbol{\beta}^{stable}$ places less weight on it, and adjusts the weight placed on x_1.

(b) Interaction x_1, x_2

$\boldsymbol{\beta}^{stable}(\alpha_{12})$

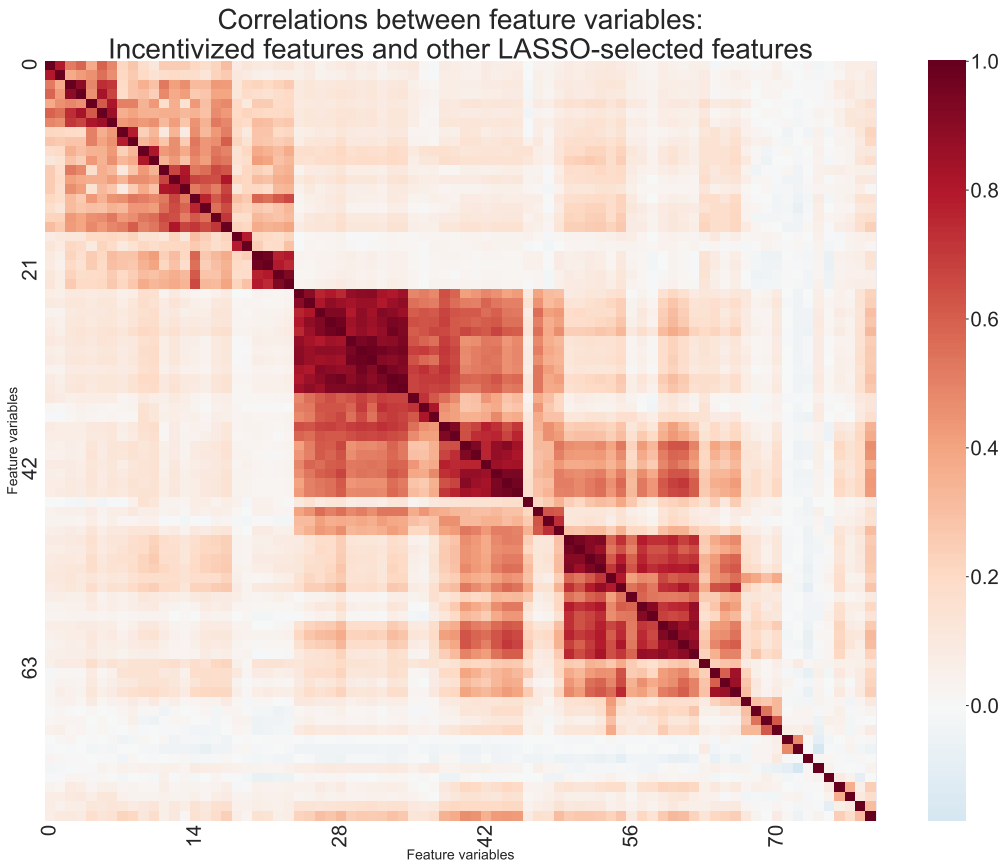
$\alpha_{22} = 32$



If manipulating one variable makes it easier to manipulate the other (α_{12} sufficiently negative), $\boldsymbol{\beta}^{stable}$ reduces weight on both.

$\underline{x}_i \overset{iid}{\sim} N(0,1)$, $\mathbf{b} = [1.4, 1]$, $\mathbf{C} = \frac{1}{\gamma\gamma_i}\begin{bmatrix} 4 & \alpha_{12} \\ \alpha_{12} & \alpha_{22} \end{bmatrix}$, $\frac{1}{\gamma_i} \overset{iid}{\sim} Uniform[0, 10]$, $\epsilon_i \overset{iid}{\sim} N(0, 0.25)$.
Squared error measured on an out of sample draw from the same population, incentivized to that decision rule.

Figure A3



Each row and column represent a feature of behavior. Features are clustered into similar groups. The diagonal indicates that the correlation of a feature with itself is +1.

A3 Appendix Tables

Table A1: Manipulation Can Harm Prediction (Monte Carlo): "Industry Approach"

	Decision Rule				**Performance** (squared loss)	
	β_0	β_1	β_2	β_3	No manip.	Manipulation
Panel A: **Data generating process**						
$\boldsymbol{b}^{DGP}$	0.200	3.000	0.100	0.100	0.267	3745.046
Panel B: **Standard Approaches**						
$\boldsymbol{\beta}^{OLS}$	0.205	3.042	0.061	0.116	0.266	3961.225
'Industry' Approach (estimated with just data from that period)						
$\boldsymbol{\beta}^{OLS(1)}$ after $\boldsymbol{\beta}^{OLS}$	-0.798	0.061	2.090	-1.675	3.275	625.762
$\boldsymbol{\beta}^{OLS(2)}$ after $\boldsymbol{\beta}^{OLS}$	0.172	3.111	-0.040	0.215	0.270	4332.208
$\boldsymbol{\beta}^{OLS(3)}$ after $\boldsymbol{\beta}^{OLS}$	-0.755	0.120	2.077	-1.671	3.071	619.059
⋮						
$\boldsymbol{\beta}^{OLS(1000)}$ after $\boldsymbol{\beta}^{OLS}$	-0.393	3.741	-1.341	1.566	1.375	11611.884
$\boldsymbol{\beta}^{OLS(1001)}$ after $\boldsymbol{\beta}^{OLS}$	-0.404	0.704	1.861	-1.526	1.674	565.383

Notes: Monte Carlo simulation results. Panel A shows the coefficients that relate the outcome (y) to behaviors ($\mathbf{x}$) under the data generating process (DGP). Panel B shows coefficients from OLS, under behavior without manipulation: $\mathbf{x}_i(\mathbf{0})$, or with manipulation: $\mathbf{x}_i(\boldsymbol{\beta})$. Parameters:

$$C = \begin{bmatrix} 1.0 & 0.1 & 0.2 \\ 0.1 & 2.0 & 0.8 \\ 0.2 & 0.8 & 4.0 \end{bmatrix}, \underline{\mathrm{x}} \overset{iid}{\sim} N\left(\mathbf{0}, \begin{bmatrix} 1 & 1 & 0.1 \\ 1 & 2 & 1 \\ 0.1 & 1 & 1 \end{bmatrix}\right), \gamma_i = \begin{cases} 1 & \underline{\mathrm{x}}_{i1} \leq 0.2 \\ 10 & \underline{\mathrm{x}}_{i1} > 0.2 \end{cases}, e_i \overset{iid}{\sim} N(0, 0.25)$$

Table A2: Manipulation Can Improve Prediction (Monte Carlo)

	Decision Rule			**Performance** (squared loss)	
	β_0	β_1	β_2	No manipulation	Manipulation
Panel A: **Data generating process**					
$\boldsymbol{b}^{DGP}$	1.00	0.10	0.01	8.749	8.748
Panel B: **Standard Approach**					
$\boldsymbol{\beta}^{OLS}$	1.014	-0.003	0.130	8.724	8.720
Panel C: **Strategy Robust Method**					
$\boldsymbol{\beta}^{stable}$	1.014	-0.022	0.156	8.725	8.719

Notes: Monte Carlo simulation results. Panel A shows the coefficients that relate the outcome (y) to behaviors ($\mathbf{x}$) under the data generating process (DGP). Panel B shows estimated coefficients from OLS; Panel C shows coefficients estimated with the strategy robust method. Performance is assessed on the same sample of individuals, under behavior without manipulation: $\mathbf{x}_i(\mathbf{0})$, or with: $\mathbf{x}_i(\boldsymbol{\beta})$. Parameters:

$$C = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \ \underset{\sim}{\mathrm{x}}_i \overset{iid}{\sim} N\left(\mathbf{0}, \begin{bmatrix} 2 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right), \quad \begin{array}{ll} \gamma_i & = 0.1u_i - \epsilon_i^3 + B \\ u_i & \sim N(0,1) \\ B & \text{set so } \min \gamma_i = 0.1 \end{array} \quad , \epsilon_i \overset{iid}{\sim} N(0,9)$$

Table A3: Performance of Decision Rules

	Costs	**All outcomes (pooled)**		Income		Ravens (intelligence) above median		Activity PCA	
	α_{jj}			β^{LASSO}	β^{stable}	β^{LASSO}	β^{stable}	β^{LASSO}	β^{stable}
	¢/action^2			¢/action		¢/action		¢/action	
Decision Rule									
call_count_out	0.591	-	-	0.625	0.542			1.978	1.241
text_count_incoming	0.038	-	-	0.065		0.278	0.145	1.154	0.306
text_count_out	0.035	-	-	-0.395	-0.107				
text_count_evening	0.058	-	-		-0.121			0.036	0.35
calls_noncontacts	2.24	-	-			-0.606	-0.575		
call_count_outgoing_missed	4.64	-	-			-0.208			
max_daily_texts_incoming	2.30	-	-				0.324		
Prediction Error									
Baseline Data:		RMSE ($)		RMSE ($)		RMSE ($)		RMSE ($)	
Control		**3.698**	**3.745**	3.553	3.554	5.144	5.158	2.396	2.523
		(0.2)	(0.19)	(0.032)	(0.032)	(0.024)	(0.025)	(0.142)	(0.076)
Predicted Transparent		**4.344**	**3.85**	4.663	3.831	5.148	5.119	3.319	2.592
		(0.479)	(0.622)	(0.0)	(0.0)	(0.0)	(0.0)	(0.0)	(0.0)
Implemented:									
Opaque		**4.002**	**3.803**	3.243	3.232	5.147	5.165	3.655	2.974
		(0.512)	(0.588)	(0.0)	(0.0)	(0.0)	(0.0)	(0.0)	(0.0)
Transparent		**4.933**	**4.308**	3.867	3.655	5.323	5.138	5.762	4.014
		(0.505)	(0.41)	(0.0)	(0.0)	(0.0)	(0.0)	(0.0)	(0.0)
Cost of Transparency		**0.931 (0.719)**	**0.504 (0.717)**	0.624 (0.0)	0.423 (0.0)	0.176 (0.0)	-0.027 (0.0)	2.108 (0.0)	1.04 (0.0)
Eqm.: Predicted		**0.152 (0.654)**		0.278 (0.032)		-0.025 (0.024)		0.196 (0.142)	
Eqm.: Implemented		**0.305 (0.656)**		0.412 (0.0)		-0.009 (0.0)		0.36 (0.0)	
Average Payout ($)		**3.226**	**2.979**	3.295	3.239	3.77	3.757	2.612	1.94
N		114		38		38		38	
N person-weeks		4476		3979		3983		3951	

Notes: The first panel reports the decision rule associated with the challenge, and the costs associated with these behaviors. The below panels report the performance of naive LASSO and our strategy-robust model, by outcome and pooled across outcomes, respectively. Performance metrics estimated using a regression of model indicators on week-model RMSE, weighted by number of person-weeks. Opaque (Training Weeks) represents the average performance of models in control person-weeks, when no behavior was incentivized. Transparent (Model) represents the average expected performance of models given the theoretical model, behavior incentives and estimated costs. Implemented Opaque represents the average performance of models when assigned without transparency hints. Implemented Transparent represents the average performance of models when assigned with transparency hints. Cost of transparency represents the difference between transparent and opaque RMSE for naive LASSO and strategy-robust (SR) models, respectively. 'Eqm.: Predicted' denotes the difference between predicted transparent RMSE under the SR model and baseline RMSE under the naive LASSO. 'Equilibrium Cost of Transparency' denotes the difference between implemented transparent SR model RMSE and opaque naive model RMSE. Average payouts represents the average payout from the assigned challenges associated with the model. N represents the number of model-weeks that the regression is estimated over, N person-weeks represents the number of person-weeks that these model-weeks include, as well as the sum of the weights used in regression. Costs represent structural costs estimated using above procedure. Standard errors in parentheses, clustered at week-outcome level.