Measuring the Completeness of Theories*

Drew Fudenberg† Jon Kleinberg‡ Annie Liang§ Sendhil Mullainathan¶

January 26, 2020

Abstract

To evaluate how well economic models predict behavior it is important to have a measure of how well *any* theory could be expected to perform. We provide a measure of the amount of predictable variation in the data that a theory captures, which we call its "completeness." We evaluate the completeness of leading theories in three applications—assigning certainty equivalents to lotteries, initial play in games, and human generation of random sequences—and show that this approach reveals new insights. We also illustrate how and why our completeness measure varies with the experiments considered, for example with the choice of lotteries used to evaluate risk preferences, and explain how our completeness measure can help guide the development of new theories.

Theories are used for many reasons, so there are many criteria for evaluating them, including parsimony, portability, tractability, and fit with prior intuition. One important criterion is the accuracy of a theory's predictions—all other things equal, we prefer more accurate theories. Thus when the best models for a given behavior

*A one-page abstract of an early version of this project appeared in ACM:EC as "The Theory is Predictive, but is it Complete?" We thank Alberto Abadie, Amy Finkelstein, and Johan Ugander for helpful comments. We are also grateful to Adrian Bruhin, Helga Fehr-Duda, Thomas Epper, Kevin Leyton-Brown, and James Wright for sharing data with us, and National Science Foundation Grants SES 1643517 and

†Department of Economics, MIT

‡Department of Computer Science, Cornell University

§Department of Economics, University of Pennsylvania

¶Department of Economics, University of Chicago

all predict poorly, there is reason to look for improvements. This paper provides a way to determine when a theory "predicts poorly" that takes into account how well we could expect any theory to perform.

Our focus throughout is on *prediction problems*, where there is a vector of measured features or covariates, and an outcome that must be predicted. The *feasible* level of predictive accuracy varies across prediction problems. For some problems, e.g. predicting the movement of astronomical bodies, the measured features are sufficient to make highly accurate predictions. But for many other prediction problems, especially in the social sciences, the available features do not allow for such accurate predictions. Here it is important to use the right benchmark for evaluating the magnitude of a model's errors.

Our view is that a model's out-of-sample prediction error can be decomposed into two parts: (1) intrinsic noise in the problem due to limitations of the feature set, i.e. the *irreducible error*, and (2) regularities in the data that the model does not capture. To understand how much we can improve the predictions of existing theories, holding constant the set of measured features, we should compare prediction errors not against a perfect zero, but rather against the irreducible noise in the problem. When the prediction errors of the best models are much larger than the irreducible error, there is room for improving predictions by using the existing features in new ways (i.e. choosing a different functional form). In contrast, when the achieved prediction errors come close to the irreducible error, then the only way to improve predictive accuracy is to measure or identify new features.

We propose that the predictive success of a theory should be measured as its "completeness," which we define as

$$\frac{\mathcal{E}_{\text{naive}} - \mathcal{E}_{\text{irreducible}}}{\mathcal{E}_{\text{model}} - \mathcal{E}_{\text{irreducible}}},$$

where $\mathcal{E}_{\text{naive}}$ is the out-of-sample prediction error under a naive baseline (e.g. "guess at random"), $\mathcal{E}_{\text{model}}$ is the out-of-sample error of the model, and $\mathcal{E}_{\text{irreducible}}$ is the irreducible error. Thus, a model's completeness is the fraction of *achievable* reduction of prediction error that it achieves. To use this measure, we need an estimate of how much irreducible error is present in the data set. We explain how to do this given data, and estimate the irreducible error on data from three important domains: the evaluation of risk, initial play in games, and human perception of randomness.

Estimation of irreducible error is possible in these domains because we have access to data sets containing a large number of outcome observations for each unique feature combination. This means that we can nonparametrically search the space of possible models, and identify the model that maximizes out-of-sample predictive accuracy for the set of available features. The resulting prediction error is an estimate of the irreducible error.

We next evaluate the completeness of popular theories for these domains relative to the benchmark of irreducible error. This comparison allows us to derive new insights in each of the domains we consider. For example, we find that the best model we use for experimental subjects trying to generate random sequences is only 24% complete, while Cumulative Prospect Theory is 95% complete for predicting certainty equivalents, despite having a mean-squared prediction error of 67.38. We find also that the Poisson Cognitive Hierarchy model is more complete for predicting play on some classes of games than it is on others. These, and the subsequent observations we make in Sections 2.1–2.3, are informative about the problem domains and how much room there is for improving the predictions of their leading models without obtaining new sorts of data.

It is perhaps striking that estimation of irreducible error is possible at all. Part of our contribution is to show that this estimation is feasible in three domains of economic interest. Irreducible error can be estimated in other domains, but not in all of them, and we discuss various limitations to our approach and its interpretation throughout the paper. The range of applications in the paper suggests that irreducible noise can be estimated in a broader range of settings than one might initially suspect.

Finally, we emphasize that our completeness measure depends on a specified set of features and is evaluated on a given data set. If we change either the underlying feature set or the data, we would expect the measurement of completeness to change, as we discuss in Section 3.2. It is important to keep in mind that the completeness measure depends on the data sets used. Moreover, as we show in Sections 2.2, the way that the completeness of a model varies across data sets is of independent interest, as it can shed light on the domains in which the model performs well or performs poorly.

1 Problem and Approach

1.1 Prediction Problems

In a prediction problem, there is an *outcome* Y whose realization is of interest, and *features* X that are statistically related to the outcome. The goal is to predict the outcome given the observed features. Some examples include predicting an individual's future wage based on childhood covariates (city of birth, family income, quality of education, etc.), or predicting a criminal defendant's flight risk based on their past record and properties of the crime (Kleinberg et al., 2017). We focus on three prediction problems that emerge from experimental economics:

Example 1 (Risk Preferences). Can we predict the valuations that people will assign to various money lotteries?

Example 2 (Predicting Play in Games). Can we predict how people will play the first time they encounter a new simultaneous-move game?

Example 3 (Human Generation of Random Sequences). Given a target random process—for example, a Bernoulli random sequence—can we predict the errors that a human will make while mimicking this process?

Formally, suppose that the observable features belong to some space $\mathcal{X}$ and the outcome belongs to $\mathcal{Y}$. There is a true but unknown joint distribution P over $\mathcal{X} \times \mathcal{Y}$. A map $f : \mathcal{X} \to \mathcal{Y}$ from features to outcomes is a *(point) prediction rule*.[1] Many economic models can be described as a family of prediction rules $\mathcal{F}_\Theta$ indexed by an interpretable parameter set Θ. For example, the model class may impose a linear relationship $f(x) = \langle x, \theta \rangle$ between the outcome and a set of features x, in which case the parameter $\theta \in \Theta$ defines a vector of weights applied to each feature. In Section 2.1, one specification of $\mathcal{F}_\Theta$ is a family of utility functions $u(z) = z^\theta$ over dollar amounts, where the parameter θ reflects the degree of risk aversion.

1.2 Accuracy and Completeness

We suppose that our prediction problem comes with a *loss function*, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, where $\ell(y', y)$ is the error assigned to a prediction of y' when the realized outcome

[1] Note that a prediction of a probability distribution over $\mathcal{Y}$ can be cast as the prediction of a point in the space $Y' = \Delta(\mathcal{Y})$ of distributions on $\mathcal{Y}$.

is y. The commonly used loss functions *mean-squared error* and *classification loss* correspond to $\ell(y', y) = (y' - y)^2$ and $\ell(y', y) = \mathbb{1}(y' \neq y)$ respectively.

Definition. *The* ***expected error*** *(or* ***risk****) of prediction rule f on a new observation $(x, y) \sim P$ is*[2]

$$\mathcal{E}_P(f) = \mathbb{E}_P[\ell(f(x), y)]. \tag{1}$$

The prediction rule in the parametric class $\mathcal{F}_\Theta$ that minimizes the expected prediction error is the one associated with the parameter value

$$f^*_\Theta = \arg\min_{f \in \mathcal{F}_\Theta} \mathcal{E}_P(f).$$

The expected error of this "best" rule in $\mathcal{F}$ is $\mathcal{E}_P(f^*_\Theta)$.

In Section 1.3, we discuss how to estimate $\mathcal{E}_P(f^*_\Theta)$ on finite data; here we discuss how to interpret it. To understand a model's error, it is helpful to distinguish between two very different sources of error.

First, if the conditional distribution $Y \mid X$ is not degenerate, then even the ideal prediction rule

$$f^*(x) = \arg\min_{y' \in \mathcal{Y}} \mathbb{E}_P\left[\ell(y', y) \mid x\right]$$

does not predict perfectly.

Definition. *The* ***irreducible error*** *in the prediction problem is the expected error*

$$\mathcal{E}_P(f^*) = \mathbb{E}_P\left[\ell(f^*(x), y)\right] \tag{2}$$

of the ideal rule on a new test observation.

The irreducible error is a lower bound on the error in predicting Y using the features in X.

A different source of prediction error is the specification of which prediction rules are in the class $\mathcal{F}_\Theta$. Typically the best possible model will not be an element of $\mathcal{F}_\Theta$, as most model classes are at least slightly misspecified. If $\mathcal{F}_\Theta$ leaves out an important

[2]Different loss functions are typically used when predicting distributions, see e.g. Gneiting and Raftery (2007).

regularity in the data, there may be models outside of $\mathcal{F}_\Theta$ that give much better predictions.[3]

These two sources of prediction error have very different implications for how to generate better predictions. If the model's prediction error is substantially higher than the irreducible error, it may be possible to identify new regularities and incorporate them into new models that improve prediction given the same feature set. New models might be preferable if they do not involve too great an increase in complexity or in the number of parameters. Conversely, if the model's prediction error is close to the irreducible error for the current feature set, the priority should be to identify additional features that will allow for better predictions.

We propose the ratio of the reduction in prediction error achieved by the model, compared to the *achievable* reduction, as a measure of how close the model comes to the best achievable performance. We call this ratio the model's *completeness*. To operationalize this measure, we provide a lower bound for the "worst case" prediction accuracy using a naive rule $f_{\text{naive}} : \mathcal{X} \to \mathcal{Y}$ suited to the prediction problem, e.g. "predict uniformly at random."

Definition. *The* **completeness of the parametric model class $\mathcal{F}_\Theta$ is**

$$\frac{\mathcal{E}_P(f_{\textbf{naive}}) - \mathcal{E}_P(f^*_\Theta)}{\mathcal{E}_P(f_{\textbf{naive}}) - \mathcal{E}_P(f^*)}. \tag{3}$$

Note that the completeness measure depends on the underlying distribution P. We expect the conditional distribution $P(y \mid x)$ to be a fixed distribution describing the dependence of the outcome on the specified set of features, but the marginal distribution on X will depend on how the data is generated. In experimental economics (where the data is laboratory data), the distribution over X is typically chosen by the analyst—e.g. which games to ask laboratory participants to play. As we show in Section 2.2, changing this marginal distribution can lead to different measures of completeness for the same model. Ideally, we would like the chosen distribution over features to be the one that is most economically relevant; in practice, we may not know the most relevant distribution.

[3]On the other hand, expanding the model class risks overfitting, so more parsimonious model classes can lead to more accurate predictions when data is scarce (Hastie et al., 2009). As we discuss in Sections 1.3 and 1.4, all of the data sets we consider here are large relative to the number of features.

Note also that we define a "model" to be a map from features to the prediction of interest, which isn't the typical meaning of the word. For this reason, the completeness of a model in our sense depends on the specified prediction problem: With the same features, a model of the effect of a price cut on sales might be able to predict the aggregate effect (e.g. a 5% increase in sales) very well but unable to predict which consumers would increase their purchases.

1.3 Evaluating Completeness from Data

The quantities $\mathcal{E}_P(f_{\text{naive}})$, $\mathcal{E}_P(f^*_\Theta)$, and $\mathcal{E}_P(f^*)$ from (3) are not directly observable. To estimate them from data, we use the following standard procedure. First, we estimate model parameters on a set of training observations. Then, we use test data to evaluate the *out-of-sample* prediction error of the chosen model. The quantities $\mathcal{E}_P(f_{\text{naive}})$, $\mathcal{E}_P(f^*_\Theta)$, and $\mathcal{E}_P(f^*)$ can be directly estimated in this way, noting that $\mathcal{E}_P(f_{\text{naive}})$ corresponds to the special case $\mathcal{F} = \{f_{\text{naive}}\}$, while $\mathcal{E}_P(f^*_\Theta)$ corresponds to $\mathcal{F} = \mathcal{F}_\Theta$, and $\mathcal{E}_P(f^*)$ corresponds to $\mathcal{F} = \mathcal{X}^\mathcal{Y}$ (the unrestricted set of all possible maps from features in $\mathcal{X}$ into outcomes in $\mathcal{Y}$).

We now describe the estimation procedure and its theoretical guarantees in more detail.

Training and Testing. Let Z denote a typical data set of observations (x, y) (which may include repetitions of the same pair). The *in-sample* performance of f for predicting the observations in Z is

$$e(f, Z) = \frac{1}{|Z|} \sum_{(x,y)\in Z} \ell(f(x), y)).$$

This is a sample analog of the expected prediction error in (1).

We will momentarily describe how the analyst splits the available data into *training observations* Z_{train} and *test observations* Z_{test}; for now suppose these sets are given. Let

$$\hat{f} \in \arg\min_{f\in\mathcal{F}} e(\hat{f}, Z_{\text{train}}) \tag{4}$$

be a model from $\mathcal{F}$ that best predicts the training observations in Z_{train}.[4] The selected

[4]When there are multiple functions satisfying (4), choose between them randomly.

model $\hat{f}$ is subsequently evaluated on the *disjoint* set of test observations in Z_{test}. Its out-of-sample prediction error is $e(\hat{f}, Z_{\text{test}})$.

Note that in the special case where $\mathcal{F}$ includes all possible mappings from $\mathcal{X}$ to $\mathcal{Y}$, a solution to (4) is a function $\hat{f}_{LT}$ such that

$$\hat{f}_{LT}(x) \in \arg\min_{y \in \mathcal{Y}} \sum_{(x,y') \in Z_{\text{train}}} \ell(y, y') \qquad \forall x \in X.$$

Such a function can be described as a lookup table that maps different elements $x \in X$ to the best prediction for that feature realization. For example, when the loss function is mean-squared error, $\hat{f}_{LT}$ maps each x to the average outcome for that feature in the training data. When the loss function is the misclassification rate, $\hat{f}_{LT}$ maps each x to the modal observed outcome for that feature.

Cross-Validation. The standard way of creating training and testing data is *K-fold cross-validation*: First, all of the available data is randomly split into K equally-sized disjoint subsets $Z_1, \ldots, Z_K$. In each iteration $1 \leq i \leq K$ of the procedure, the subset $Z^i_{\text{test}} \equiv Z_i$ is identified as the *test data* and the remaining subsets $Z^i_{\text{train}} \equiv \cup_{j \neq i} Z_j$ are used as *training data*. The best-fit model for the i-th training set is

$$f_i \equiv \arg\min_{f \in \mathcal{F}} e(f, Z^i_{\text{train}}).$$

This model's out-of-sample performance on the i-th test set is

$$\text{CV}_i = e(f_i, Z^i_{\text{test}}). \tag{5}$$

The average out-of-sample error across the K test sets is

$$\text{CV}(\mathcal{F}, \{Z_i\}_{i=1}^K) = \frac{1}{K} \sum_{i=1}^{K} \text{CV}_i. \tag{6}$$

Estimated Completeness. Define

$$\begin{aligned} \widehat{\mathcal{E}}_{\text{naive}} &\equiv \text{CV}(\{f_{\text{naive}}\}, \{Z_i\}_{i=1}^K) \\ \widehat{\mathcal{E}}_{\Theta} &\equiv \text{CV}(\mathcal{F}_{\Theta}, \{Z_i\}_{i=1}^K) \\ \widehat{\mathcal{E}}_{\text{best}} &\equiv \text{CV}(\mathcal{X}^{\mathcal{Y}}, \{Z_i\}_{i=1}^K). \end{aligned}$$

In place of the theoretical completeness measure described in (3), we compute the empirical ratio

$$\frac{\widehat{\mathcal{E}}_{\text{naive}} - \widehat{\mathcal{E}}_{\Theta}}{\widehat{\mathcal{E}}_{\text{naive}} - \widehat{\mathcal{E}}_{\text{best}}} \tag{7}$$

from our data. The tables we report in the subsequent applications in Sections 2.1-2.3 are structured as follows:

	Error	Completeness
Naive Benchmark	$\widehat{\mathcal{E}}_{\text{naive}}$	0%
Economic Model	$\widehat{\mathcal{E}}_{\Theta}$	$100 \times \left(\widehat{\mathcal{E}}_{\text{naive}} - \widehat{\mathcal{E}}_{\Theta}\right) / \left(\widehat{\mathcal{E}}_{\text{naive}} - \widehat{\mathcal{E}}_{\text{best}}\right)$ %
Irreducible Error	$\widehat{\mathcal{E}}_{\text{best}}$	100%

The out-of-sample prediction error for the economic model, $\widehat{\mathcal{E}}_{\Theta}$, is evaluated relative to our estimate for the naive performance error, $\widehat{\mathcal{E}}_{\text{naive}}$, and our estimate for the irreducible error, $\widehat{\mathcal{E}}_{\text{best}}$. In the main text, we refer to these estimates simply as *prediction errors*, understanding that they are finite-data estimates. Our estimate for the completeness of the model is the ratio of the difference between the naive error and the model's error, $\widehat{\mathcal{E}}_{\text{naive}} - \widehat{\mathcal{E}}_{\Theta}$, and the difference between the naive error and the irreducible error, $\widehat{\mathcal{E}}_{\text{naive}} - \widehat{\mathcal{E}}_{\text{best}}$.

Theoretical Guarantees. The empirical quantities $\widehat{\mathcal{E}}_{\text{naive}}$, $\widehat{\mathcal{E}}_{\Theta}$, and $\widehat{\mathcal{E}}_{\text{best}}$ are consistent estimators for $\mathcal{E}_P(f_{\text{naive}})$, $\mathcal{E}_P(f^*_{\Theta})$, and $\mathcal{E}_P(f^*)$, respectively (Hastie et al., 2009), and the empirical estimate of completeness in (7) is a consistent estimator for (3).

These estimates are good approximations for the theoretical quantities when the number of observations is sufficiently large. In particular, for $\widehat{\mathcal{E}}_{\text{best}}$ to be a good approximation of the irreducible noise $\mathcal{E}_P(f^*)$, the analyst must have access to a sufficiently large number of observations *for each* distinct $x \in \mathcal{X}$. This can be a demanding criterion. To evaluate whether we have "enough" data in our applications, we report for each model class the *standard error of the cross-validated prediction errors*, which is

$$\sqrt{\frac{1}{K}\text{Var}(\text{CV}_1, \ldots, \text{CV}_K)},$$

with CV_i as defined in (5). Loosely speaking, if the quantity of data is small, then the empirical out-of-sample error will be very sensitive to which observations are used as training data and which are used as testing data. We thus expect substantial variation in the out-of-sample error across the different training-testing iterations. For all of the data sets we look at, the standard errors are small relative to the magnitudes of the prediction errors. This suggests that our empirical estimates, in particular our estimate for irreducible error, are close to their infinite-sample counterparts. See Appendix A.1 for more detail.[5]

In general, the condition that the data includes many observations per feature is easier to satisfy in experimental settings, where the experimentalist has control over the structure of the data and can choose to acquire a large number of observations for each of a fixed set of feature values. For example, in the data sets that we consider, there is an average of 179 observations per unique x (Section 2.1), 50 observations per unique x (Section 2.2), and 164 observations per unique x (Section 2.3).

1.4 Relationship to Literature

Irreducible error is an old concept in statistics and machine learning, and a large amount of work has focused on further decomposing this error into *bias* (reflecting error due to the specification of the model class) and *variance* (reflecting sensitivity of the estimated rule to the randomness in the training data). Depending on the quantity of data available to the analyst, it may be preferable to trade off bias for variance or vice versa.[6] This paper abstracts from these concerns, as well as the related concern of overfitting. We work exclusively with data sets where there is enough data that

[5]Besides looking at standard errors, we consider two additional test for whether $\widehat{\mathcal{E}}_{\text{best}}$ is a good approximation for the irreducible error $\mathbb{E}_P(f^*)$. First, we compare the performance of the lookup table function $\hat{f}_{LT}$ with a machine learning algorithm that is better suited to smaller data sets (bagged decision trees). The out-of-sample performances are comparable, but $\hat{f}_{LT}$ has a lower error for all of our applications (see Appendix A.2). Second, we investigate whether the out-of-sample performance of $\hat{f}_{LT}$ has converged by evaluating its performance on subsamples of our data. The prediction errors using just 70% of the data are very close to those using all of our data. These analyses suggest that our estimate for irreducible error is a reasonable approximation in each of our applications.

[6]For example, given small quantities of data, we may prefer to work with models that have fewer free parameters, leading to higher bias but potentially lower variance.

the best feasible out-of-sample prediction accuracy is well approximated by searching across the unrestricted space of mappings from $\mathcal{X}$ into $\mathcal{Y}$ (see Appendix A).

A related literature compares the performance of specific machine learning algorithms to the performance of existing economic models. The closest of these papers to our work is Peysakhovich and Naecker (2017), which studies choices under uncertainty and under ambiguity, and compares the performance of economic models to the performance of regularized regression algorithms.[7] This paper makes a similar conceptual point to ours, but regularized regression algorithms are themselves potentially incomplete. Thus, these algorithms provide a lower bound for the best achievable predictive accuracy, where the degree to which the algorithms are incomplete is a priori unknown. In many experimental contexts, it can be possible to directly estimate the quantity of irreducible noise and use that as a benchmark.[8]

Erev et al. (2007) define a model's *equivalent number of observations* as the number n of prior observations such that the mean of a data set of n random observations has the same prediction error as the model. We expect that models with larger numbers of equivalent observations will be more complete by our measure.

Finally, an alternative measure of a model's performance is the proportion of the variance in the outcome that it explains, i.e. the model's R^2. This measure plays an analogous role to the measures of predictive accuracy that we consider in this paper. Crucially, the achieved R^2 of a given model should be evaluated not against a perfect benchmark of $R^2 = 1$, but against the *best achievable* R^2 for the data set and the given features. This best achievable R^2 cannot be directly inferred from the R^2 of any existing model, but we can estimate it using a lookup table as described in Section 1.3. One could use this approach to define an analogous notion of completeness based on the ratio of the achieved improvement in R^2 (over a naive baseline) compared to the achievable improvement.

[7] In addition, Plonsky et al. (2017), Noti et al. (2016), and Plonsky et al. (2019) develop algorithmic models for predicting choice, Camerer et al. (2018) uses machine learning to predict disagreements in bargaining, and Bodoh-Creed et al. (2019) uses random forests to predict pricing variation. The improvements achieved by these algorithms are sometimes modest, perhaps due to intrinsic noise, as Bourgin et al. (2019) point out. We show how this noise can be quantified.

[8] Although the error of regularized regression models need not approximate the irreducible error, they (and other more scalable machine learning methods) may serve as effective substitutes when there is not enough data for nonparametric estimation to perform well.

2 Three Applications

2.1 Domain #1: Assigning Certain Equivalents to Lotteries

Background and Data. An important question in economics is how individuals evaluate risk. In addition to Expected Utility models (von Neumann and Morgenstern, 1944; Savage, 1954; Samuelson, 1952), one of the most influential models of decision-making under risk is Cumulative Prospect Theory (Tversky and Kahneman, 1992). This model provides a flexible family of risk preferences that accommodates certain behavioral anomalies, including reference-dependent preferences and nonlinear probability weighting.

A standard experimental paradigm for eliciting risk preferences, and thus for evaluating these models, is to ask subjects to report *certainty equivalents* for lotteries—i.e. the lowest certain payment that the individual would prefer over the lottery. We consider a data set from Bruhin et al. (2010), which includes 8906 certainty equivalents elicited from 179 subjects, all of whom were students at the University of Zurich or the Swiss Federal Institute of Technology Zurich. Subjects reported certainty equivalents for the same 50 two-outcome lotteries, half over positive outcomes (e.g. gains) and half over negative outcomes (e.g. losses).

Prediction Task and Models. In this data set, the outcomes are the reported certainty equivalents for a given lottery, and the features are the lottery's two possible monetary prizes z^1 and z^2, and the probability p of the first prize. A prediction rule is any function that maps the tuple (z^1, z^2, p) into a prediction for the certainty equivalent, i.e. a function $f : \mathbb{R} \times \mathbb{R} \times [0, 1] \to \mathbb{R}$. We use mean-squared error as the loss-function: In a test set of n observations $\{(z_i^1, z_i^2, p_i; y_i)\}_{i=1}^n$—where (z_i^1, z_i^2, p_i) is the lottery shown in observation i and y_i is the reported certainty equivalent—the performance of f is

$$\frac{1}{n} \sum_{i=1}^{n} \left(f(z_i^1, z_i^2, p_i) - y_i \right)^2 .$$

We evaluate two prediction rules that are based on established models from the literature. Our *Expected Utility* (EU) rule sets the agent's utility function to be $u(z) = z^\alpha$, where α is a free parameter that we train. The predicted certainty equivalent is $p \cdot (z^1)^\alpha + (1 - p) \cdot (z^2)^\alpha$.

Second, our *Cumulative Prospect Theory* (CPT) rule predicts

$$w(p)v(z^1) + (1 - w(p))v(z^2)$$

for each lottery, where w is a probability weighting function and v is a value function. We follow Bruhin et al. (2010) in our choice of functional forms:

$$v(z) = \begin{cases} z^\alpha & \text{if } z > 0 \\ -(-z^\beta) & \text{if } z \leq 0 \end{cases} \qquad w(p) = \frac{\delta p^\gamma}{\delta p^\gamma + (1-p)^\gamma}. \tag{8}$$

This model has four free parameters: $\alpha, \beta, \delta, \gamma \in \mathbb{R}_+$.

Finally, as a naive benchmark, we predict the *expected value* of the lottery, which is $pz^1 + (1-p)z^2$.[9]

Results. The following table reveals that both models are predictive, as their out-of-sample prediction error improves upon the Expected Value benchmark:[10]

	Error
Naive Benchmark	103.81
	(4.00)
Expected Utility	99.67
	(4.50)
CPT	67.38
	(4.49)

Table 1: Both models are predictive.

The improvement of CPT over the naive benchmark is larger than that of Expected Utility, but even CPT's performance is substantially worse than perfect prediction.

[9] This naive benchmark is arguably less naive than the naive benchmarks we use for the other prediction problems. Replacing our naive benchmark with, for example, an unconditional mean, would result in even higher completeness for CPT than we find in Table 2.

[10] The parameter estimate for EU is $\alpha = 0.98$, and the parameter estimates for CPT are $\alpha = 1.024, \beta = 0.975, \delta = 0.5$, and $\gamma = 0.525$.

It is not surprising that these models do not achieve perfect prediction, as we expect different subjects to report different certainty equivalents for the same lottery, and thus a model that provides the same prediction for each (z^1, z^2, p) input cannot possibly predict every reported certainty equivalent.

But based on Table 1 alone, it is difficult to interpret the gap between CPT's accuracy and perfect prediction. In particular, another source of prediction error is the functional form assumptions that we made in (8). Could a different (potentially more complex) specification for the value function or probability weighting function lead to large gains in prediction? Relatedly, might there be other features of risk evaluation, yet unmodelled, which lead to even larger improvements in prediction?

To separate these sources of error, we need to understand how CPT's error compares to the irreducible error for this data. We estimate the irreducible error in this problem using a lookup table, where each of the 50 unique lotteries is mapped to the average certainty equivalent for that lottery in the training data. With 179 observations for each of the lotteries, we are able to approximate the mean certainty equivalent for each lottery using the training data, thus (approximately) minimizing the out-of-sample prediction error. We report the estimated irreducible error in Table 2.

	Error	Completeness
Naive Benchmark	103.81	0%
	(4.00)	
Expected Utility	99.67	11%
	(4.50)	
CPT	67.38	95%
	(4.49)	
Irreducible Error	65.58	100%
	(3.00)	

Table 2: CPT is nearly complete for prediction of our data.

Table 2 shows that the CPT prediction error is almost as low as the irreducible error—CPT achieves 95% of the feasible reduction in prediction error over the naive

baseline. Thus this data suggests that there is no reason to try to construct more predictive theories that use only the features (z^1, z^2, p).[11] To further reduce error, we would need to expand the set of variables on which the model depends. For example, as we discuss in Section 9, we could group subjects using auxiliary data such as their evaluations of other lotteries or response times, and make separate predictions for each group.

We note that our completeness measure does not imply that *in general* CPT is a nearly-complete model for predicting certainty equivalents, since the completeness measure we obtain is determined from a specific data set, so its generalizability depends on the extent to which that data is representative. That said, it is suggestive that Peysakhovich and Naecker (2017) find that CPT approximates the performance of regularized regression models for the prediction of a data set of certainty equivalents on 3-outcome lotteries.[12]

2.2 Domain #2: Initial Play in Games

Background and Data. In many game theory experiments, equilibrium analysis is a poor predictor of the choices that people make when they encounter a new game. This has led to models of initial play that depart from equilibrium theory, for example the level-k models of Stahl and Wilson (1994) and Nagel (1995), the Poisson Cognitive Hierarchy model (Camerer et al., 2004), and the related models surveyed in Crawford et al. (2013). These models represent improvements over the equilibrium predictions, but we do not know how substantial these improvements are. Do these models exhaust the important regularities in initial play?

We suspect that the answer to this question depends on the kinds of games we choose to study. To test this, we compare the performance of PCHM on three subsamples of a data set from Fudenberg and Liang (2019). Our full data set consists of 23,137 total observations of initial play from 486 3×3 matrix games, where observa-

[11] It is hard to know whether the high completeness of CPT (in the specified functional form) comes from its good match to actual behavior or because it is flexible enough to mimic most functions in $\mathcal{X}^{\mathcal{Y}}$. We leave the exploration of this question to future work.

[12] The specification of CPT in Peysakhovich and Naecker (2017) sets $\delta = 1$ and thus has one fewer free parameter, so its model error may be higher.

tions are pooled across all of the subjects and games.[13,14]

The first subsample, *Game Set A*, consists of the 16,660 observations of play from the 359 games with no strictly dominated actions.[15] *Game Set B* consists of the 7,860 observations of play from the 161 games in which the profile that maximizes the sum of the players' payoffs is much larger (at least 20% of the largest row player payoff in the game) than the highest sum from the level-k actions for any k.[16] For example, in the game below (which is included in Game Set B), the action profile (a_2, a_2) leads to a payoff sum of 160, but the largest payoff sum using level-k actions is 120. The difference, 40, is more than 20% of the max row player payoff in this game, 100.[17]

	a_1	a_2	a_3
a_1	$40, 40$	$10, 20$	$70, 30$
a_2	$20, 10$	$80, 80$	$0, 100$
a_3	$30, 70$	$100, 0$	$60, 60$

Finally, *Game Set C* consists of the 9,243 observations of play from the 175 games where the level 1 action's expected payoff against uniform play is much higher than the expected payoff of the next best action (specifically, it is larger by at least 1/4 of the max row player payoff in the game).

The analysis we perform for these three subsamples can be conducted for arbitrary sets of games.

[13]This data is an aggregate of three data sets: the first is a meta data set of play in 86 games, collected from six experimental game theory papers by Kevin Leyton-Brown and James Wright, see Wright and Leyton-Brown (2014); the second is a data set of play in 200 games with randomly generated payoffs, which were gathered on MTurk for Fudenberg and Liang (2019); the third is a data set of play in 200 games that were "algorithmically designed" for a certain model (level 1) to perform poorly, again from Fudenberg and Liang (2019).

[14]There was no learning in these experiments—subjects were randomly matched to opponents, were not informed of their partners' play, and did not learn their own payoffs until the end of the session.

[15]Specifically, we consider games where no pure action is strictly dominated by another pure action.

[16]Following Stahl and Wilson (1995) and Nagel (1995), level-0 corresponds to uniform play, and each level-k action is the best response to level-$(k-1)$ play.

[17]In this game, action a_3 is level 1, since it yields the highest expected payoff against uniform play, and action a_1 is level 2, since it is a best response against play of a_1. Because (a_1, a_1) is a pure-strategy Nash equilibrium, action a_1 is level-k for all $k \geq 2$.

Prediction Task and Models. In the prediction problem we consider here, the outcome is the action that is chosen by the row player in a given instance of play, and the features are the 18 entries of the payoff matrix. A prediction rule is thus any map $f : \mathbb{R}^{18} \to \{a_1, a_2, a_3\}$ from 3×3 payoff matrices to row player actions.

For each prediction rule f and test set of observations $\{(g_i, a_i\}_{i=1}^n$—where g_i is the payoff matrix in observation i, and a_i is the observed row player action—we evaluate error using the *misclassification rate*

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{1} \left(f(g_i) \neq a_i \right).$$

This is the fraction of observations where the predicted action was not the observed action.

As a naive baseline, we consider guessing uniformly at random for all games, which yields an expected misclassification rate of $2/3$. Additionally, we consider a prediction rule based on the *Poisson Cognitive Hierarchy Model* (PCHM), which supposes that there is a distribution over players of differing levels of sophistication: The *level-0* player randomizes uniformly over his available actions, while the *level-1* player best responds to level-0 play (Stahl and Wilson, 1994, 1995; Nagel, 1995). Camerer et al. (2004) defines the play of level-k players, $k \geq 2$, to be the best response to a perceived distribution

$$p_k(h) = \frac{\pi_\tau(h)}{\sum_{l=0}^{k-1} \pi_\tau(l)} \qquad \forall\ h \in \mathbb{N}_{<k} \tag{9}$$

over (lower) opponent levels, where π_τ is the Poisson distribution with rate parameter τ.[18] We can derive a predicted distribution over actions by supposing that the proportion of level-k players in the population is proportional to $\pi_\tau(k)$. Assuming this is the true distribution of play, the misclassification rate is minimized by predicting the mode of this distribution. We define the PCHM prediction to be that mode.

Results. Below we report the estimated irreducible error and associated completeness measures for each of the three sets of games.

[18] Throughout, we take τ to be a free parameter and estimate it from the training data.

	Game Set A		Game Set B		Game Set C	
	Error	Completeness	Error	Completeness	Error	Completeness
Naive Benchmark	0.66	0%	0.66	0%	0.66	0%
PCHM	0.49	68%	0.44	68%	0.28	97%
	(0.006)		(0.009)		(0.004)	
Irreducible Error	0.41	100%	0.34	100%	0.27	100%
	(0.005)		(0.006)		(0.005)	

Table 3: Comparison of the completeness of PCHM across the three sets of games.

Our estimate for the irreducible error is derived using a lookup table, where each game is mapped to the action most commonly chosen in that game in the training data. Since we have on average 50 observations per game, the modal action in the training data is a good approximation for the modal action in the test data. High irreducible error means that there is substantial heterogeneity in play, so predicting the mode still leads to a high rate of incorrect classification. Low irreducible error means that play across subjects is more coordinated on a single action. We find that the estimated irreducible error is largest—and hence, there is the most heterogeneity in play—in Data Set A, which includes only games where there are no strictly dominated actions, and smallest in Data Set C, which includes only games where the level-1 action has by far the highest expected payoff against uniform play.

Next we use the estimated irreducible errors as a benchmark to evaluate the completeness of PCHM on the three data sets. Although the PCHM achieves a better *absolute* prediction error for predicting play in the games in Game Set A than in Game Set B, its completeness is approximately 68% on both data sets. In contrast, the PCHM achieves 97% of the feasible reduction in prediction error in Game Set C. This means that PCHM captures essentially all of the predictable variation in games where the level 1 action clearly has the largest expected value against uniform play, while there is additional structure beyond the PCHM in Game Sets A and B. We leave to future work the question of what additional properties of the game influence the completeness of the PCHM.

2.3 Domain #3: Human Generation of Random Sequences

Background and Data. Extensive experimental and empirical evidence suggests that humans misperceive randomness, for example expecting that sequences of coin flips "self-correct" (too many Heads in a row must be followed by a Tails) and are balanced (the number of Heads and Tails are approximately the same) (Bar-Hillel and Wagenaar, 1991; Tversky and Kahneman, 1971). These misperceptions are significant not only for their basic psychological interest, but also for the ways in which misperception of randomness manifests itself in a variety of contexts: for example, investors' judgment of sequences of (random) stock returns (Barberis et al., 1998), professional decision-makers' reluctance to choose the same (correct) option multiple times in succession (Chen et al., 2016), and people's execution of a mixed strategy in a game (Batzilis et al., 2016).

A common experimental framework in this area is to ask human participants to generate fixed-length strings of k (pseudo-)random coin flips, for some small value of k (e.g. $k = 8$), and then to compare the produced distribution over length-k strings to the output of a Bernoulli process that generates realizations from $\{H, T\}$ independently and uniformly at random (Rapaport and Budescu, 1997; Nickerson and Butler, 2009). Following in this tradition, we use the platform Mechanical Turk to collect a large dataset of human-generated strings designed to simulate the output of a *Bernoulli(0.5) process*, in which each symbol in the string is generated from $\{H, T\}$ independently and uniformly at random. To incentivize effort, we told subjects that payment would be approved only if their (set of) strings could not be identified as human-generated with high confidence.[19,20] After removing subjects who were clearly not attempting to mimic a random process, our final data set consisted of 21,975 strings generated by 167 subjects.[21]

[19] In one experiment, 537 subjects each produced 50 binary strings of length eight. In a second experiment, an additional 101 subjects were asked to each generate 25 binary strings of length eight.

[20] Subjects were informed: "To encourage effort in this task, we have developed an algorithm (based on previous Mechanical Turkers) that detects human-generated coin flips from computer-generated coin flips. You are approved for payment only if our computer is not able to identify your flips as human-generated with high confidence."

[21] Our initial data set consists of 29,375 binary strings. We chose to remove all subjects who repeated any string in more than five rounds. This cutoff was selected by looking at how often each subject generated any given string and finding the average "highest frequency" across subjects. This

Prediction Task, Performance Metric, and Models. We consider the problem of predicting the probability that the eighth entry in a string is H given its first seven elements. Thus the outcome here is a number in $[0, 1]$—a distribution on $\{H, T\}$—and the feature space is $\{H, T\}^7$ (note that as in the previous examples we fit a representative-agent model and do not treat the identity of the subject as a feature).

Given a test data set $\{(s_i^1, \ldots, s_i^8)\}_{i=1}^n$ of n binary strings of length-8, we evaluate the error of the prediction rule f using mean-squared error

$$\frac{1}{n}\sum_{i=1}^{n}\left(s_i^8 - f(s_i^1, \ldots, s_i^7)\right)^2$$

where $f(s_i^1, \ldots, s_i^7)$ is the predicted probability that the eighth flip is 'H' given the observed initial seven flips $s_i^1, \ldots, s_i^7$, and s_i^8 is the actual eighth flip.[22] Note that the naive baseline of unconditionally guessing 0.5 guarantees a mean-squared prediction error of 0.25. Moreover, if the strings in the test set were truly generated via a Bernoulli(0.5) process, then no prediction rule could improve in expectation upon the naive error.[23] We expect that behavioral errors in the generation process will make it possible to improve upon the naive baseline, but do not know *how much* it is possible to improve upon 0.25.

In this task, the natural naive baseline is the rule that unconditionally guesses that the probability the final flip is 'H' is 0.5. We compare this baseline to prediction

turned out to be 10% of the strings, or five strings. Thus, our selection criteria removes all subjects whose highest frequency was above average. This selection eliminated 167 subjects and 7,400 strings, yielding a final dataset with 471 subjects and 21,975 strings. We check that our main results are not too sensitive to this selection criteria by considering two alternative choices in Appendix C.2—first, keeping only the initial 25 strings generated by all subjects; second, removing the subjects whose strings are "most different" from a Bernoulli process under a χ^2-test. We find very similar results under these alternative criteria.

[22] Alternatively we could have defined the outcome to be an individual realization of H or T, so that prediction rules are maps $f : \{H, T\}^7 \to \{H, T\}$, and then evaluated error using the misclassification rate (i.e. the fraction of instances where the predicted outcome was not the realized outcome). We do not take a stand on which method is better, but note that the completeness measure can depend on which approach is used. In Appendix C.1 we show that the completeness measures are very similar using this alternative formulation.

[23] Due to the convexity of the loss function, it is possible to do *worse* than the naive baseline, for example by predicting 1 unconditionally.

rules based on Rabin (2002) and Rabin and Vayanos (2010), both of which predict negatively autocorrelated sequences.[24] Our prediction rule based on Rabin (2002) supposes that subjects generate sequences by drawing sequentially *without replacement* from an urn containing $0.5N$ '1' balls and $0.5N$ '0' balls. The urn is "refreshed" (meaning the composition is returned to its original) every period with independent probability p. This model has two free parameters: $N \in \mathbb{Z}_+$ and $p \in [0, 1]$.

Our prediction rule based on Rabin and Vayanos (2010) assumes that the first flip $s_1 \sim \text{Bernoulli}(0.5)$ while each subsequent flip s_k is distributed

$$s_k \sim \text{Ber}\left(0.5 - \alpha \sum_{t=0}^{k-2} \delta^t (2 \cdot s_{k-t-1} - 1)\right),$$

where the parameter $\delta \in \mathbb{R}_+$ reflects the (decaying) influence of past flips, and the parameter $\alpha \in \mathbb{R}_+$ measures the strength of negative autocorrelation.[25]

Results. Table 4 shows that both prediction rules improve upon the naive baseline. The need for a benchmark for achievable prediction is starkest in this application, as the best improvement is only 0.0008, while the gap between the achieved prediction errors and a perfect zero is large. This is not surprising—since the data is generated by subjects attempting to mimic a fair coin, we naturally expect substantial variation in the eighth flip after conditioning on the initial seven flips.

[24] Although both of these frameworks are models of mistaken *inference* from data, as opposed to human attempts to generate random sequences, they are easily adapted to our setting, as the papers explain.

[25] We make a small modification on the Rabin and Vayanos (2010) model, allowing $\alpha, \delta \in \mathbb{R}_+$ instead of $\alpha, \delta \in [0, 1)$.

	Error
Naive Benchmark	0.25
Rabin (2002)	0.2494
	(0.0007)
Rabin and Vayanos (2010)	0.2492
	(0.0007)

Table 4: Both models improve upon naive guessing, but the absolute improvement is small.

For this problem, we can approximate the irreducible error by learning the empirical frequency with which each length-7 string is followed by 'H' in the training data. Although there are 2^7 unique initial sequences, with approximately 21,000 strings in our data set we have (on average) 164 observations per initial sequence.

	Error	Completeness
Naive Benchmark	0.25	0
Rabin (2002)	0.2494	10%
	(0.0007)	
Rabin & Vayanos (2010)	0.2492	14%
	(0.0007)	
Irreducible Error	0.2441	100%
	(0.0006)	

Table 5: The feasible reduction in prediction error over the naive baseline is small in this problem.

We find that irreducible error in this problem is 0.2441, so that naively comparing achieved prediction error against perfect prediction (which would suggest a completeness measure of at most 0.4%) grossly misrepresents the performance of the models. The existing models produce up to 14% of the achievable reduction in prediction error. This suggests that although negative autocorrelation is indeed present

in the human-generated strings and explains a sizable part of the deviation from a Bernoulli(0.5) process, there is additional structure that could yet be exploited for prediction.

3 Extensions

3.1 Subject Heterogeneity

So far, we have evaluated the completeness of "representative agent" models that implement a single prediction across all subjects. When we evaluate models that allow for subject heterogeneity, the question of what is the largest achievable reduction in prediction error is still relevant, and the irreducible error for the new expanded feature set can again help us determine the size of potential error reductions. As a simple illustration, we return to our evaluation of risk preferences and demonstrate how to construct a predictive bound for certain models with subject heterogeneity.

The models that we consider extend the Expected Utility and Cumulative Prospect Theory models introduced in Section 2.1 by allowing for three groups of subjects. To test the models, we randomly select 71 (out of 171) subjects to be test subjects, and 45 (out of 50) lotteries to be test lotteries. All other data—the 100 training subjects' choices in all lotteries, as well as the test subjects' choices in the 5 training lotteries—are used for training the models.

We first use the training subjects' responses in the *training* lotteries to develop a clustering algorithm to separate subjects into three groups.[26] This algorithm can assign a group number to any new subject based on their choices in the five training lotteries. Second, we use each group's training subjects' responses in the *test* lotteries to estimate free model parameters—that is, the single free parameter of the EU model, and the four free parameters for CPT. This yields three versions of EU and CPT, one per group.

Out of sample, we first use the clustering algorithm to assign groups to the test subjects, and then use the associated models to predict each group's certainty equivalents in the test lotteries. We measure accuracy using mean-squared error, as in

[26] We use a simple algorithm, k-means, which minimizes the Euclidean distance between the vectors of reported certainty equivalents for subjects within the same group.

Section 2.1, and we again report the Expected Value prediction as a naive baseline.

	Prediction Error
Naive Benchmark	104.17
	(12.95)
Expected Utility	86.68
	(10.69)
CPT	57.14
	(7.17)

Table 6: Prediction Errors Achieved by Models with Subject Heterogeneity

What we find from Table 6 is very similar to what we observed in Section 2.1: Both models improve upon the naive baseline, but we do not know how complete these improvements are. To better evaluate the achieved improvements, we need a benchmark that tells us the best feasible prediction.

Our approach here for estimating the irreducible error is to learn the mean response of training subjects in each group for each lottery, and predict those means. With sufficiently many training subjects, this method approximates the best possible accuracy. We find that although the CPT error is substantially different from zero, the model is again nearly complete.

	Prediction Error	Completeness
Naive Benchmark	104.17	0%
	(12.95)	
Expected Utility	86.68	36%
	(10.69)	
CPT	57.14	96%
	(7.17)	
Irreducible Error	55.45	100%
	(6.26)	

We note that because the same clustering method is used in all of the approaches, the gap between irreducible error and the prediction errors does not shed light on how much predictions could be improved by better ways of grouping the subjects.[27]

3.2 Comparing Feature Sets

In the main text, we considered a fixed feature set $\mathcal{X}$, and evaluated the completeness of different models for prediction given this feature set. We can alternatively compare irreducible error across different feature sets as a way of contrasting the predictive limits of those features. We illustrate this comparison by revisiting our problem from Section 2.3—predicting human generation of randomness—and considering three feature sets.

The first feature set, $\mathcal{X}_{1:7}$, is our main feature set, which consists of the initial seven flips. Define $\mathcal{X}_{4:7} = \{H, T\} \times \{H, T\} \times \{H, T\}$ as the set with only the values of flips 4–7, and $\mathcal{X}_H = \{0, 1, 2 \ldots, 7\}$ as the number of 'H' realizations in the first seven flips. Interpreted as lookup tables, these new feature sets correspond to "compressed" lookup tables built on different properties of the initial seven flips, where strings are partitioned based on certain properties. We can estimate irreducible error by predicting the average continuation probability of 'H' among all strings in the same partition element.

Table 7: Comparison of the value of various feature sets.

	Error	Completeness
Naive Benchmark	0.25	0%
Irreducible Error for $\mathcal{X}_{4:7}$	0.2478	36%
	(0.0010)	
Irreducible Error for $\mathcal{X}_H$	0.2464	59%
	(0.0009)	
Irreducible Error for $\mathcal{X}_{1:7}$	0.2441	100%
	(0.0006)	

[27] A comparison of the irreducible error under clustering, 55.45, with the irreducible error from Section 2.1, 65.58, sheds light on the size of predictive gains achieved by the present method for clustering.

We find that the feature sets $\mathcal{X}_{4:7}$ and $\mathcal{X}_H$ achieve large fractions of the achievable improvement from using $\mathcal{X}_{1:7}$. For example, using only the number of Heads as a feature, it is possible to achieve 59% of the achievable reduction of the full structure of the initial flips. Using only the most recent three flips achieves 36% of the reduction from using all seven initial flips. On the other hand, the gap between irreducible error for $\mathcal{X}_{4:7}$ and for $\mathcal{X}_{1:7}$ demonstrates that there is predictive content in flips 1–3 beyond what is captured in flips 4–7.

The feature set $\mathcal{X}_{1:7}$ could be expanded to create richer feature sets, and it would be interesting to consider what additional features might significantly improve predictive accuracy, for example "neuroeconomic" data such as the speed with which the strings were entered, or demographic data such as age or education.[28] The exercise in Section 3.1, in which we used subject types (determined based on choices in auxiliary problems), illustrates yet another way to expand the feature set. As we have shown above, comparing irreducible error across different feature sets is one potentially useful approach for measuring the predictive value of those features.[29]

4 Conclusion

When evaluating the predictive performance of a theory, it is important to know not just whether the theory is predictive, but also how complete its predictive performance is. Thus we should compare the prediction errors achieved by our models against the irreducible error for that problem. What is perhaps striking is that irreducible error can be feasibly computed in certain problems of interest. We demonstrate three domains in which completeness can help us evaluate the performance of existing models.

This paper focuses on predictiveness. The completeness of a theory's prediction is not meant to be the final word on its value. Instead, the purpose of the completeness measure is to guide the evaluation of the predictive content of a theory. Occasionally,

[28] As another example: recent work by Bernheim et al. (2019) test how well a model of Cumulative Prospect Theory that is trained on two-outcome lotteries predicts certainty equivalents for three-outcome lotteries. It finds that these "cross-domain" predictions can be improved using additional non-choice features (e.g. survey responses).

[29] Note that the value of individual features will in general depend on what other features are available.

as we saw in Section 2.1, a model that is only weakly predictive may nevertheless be nearly complete for its feature set.

We conclude with a brief discussion of our completeness measure, its limitations, and possibilities for extension.

Interpretability. In many applications, researchers may prefer to sacrifice some predictive power and completeness to use a model that is easier to interpret, for example using a model of preferences to predict choice as opposed to a black box, or adding a risk aversion parameter to level-l models as in Fudenberg and Liang (2019). With an interpretable model it can be easier to see how to extend predictions to a new domain. It is also easier to bring intuition to bear, and when the model fits with intuition it may be more robust to different data selection and collection procedures.

Counterfactuals. Economic models are often used to provide counterfactual predictions about the impact of new policies. Of course, if there is no data about such policies, these counterfactual predictions rely on untested intuitions about the robustness of various forces that drive behavior. Suppose for example that the price variation in our data only comes from price changes by firms, and we want to predict the effect of a sales tax. We might conjecture that the price effects are the same as before, but in some cases consumers might be either more or less willing to accept a price increase imposed by the government. With or without an economic theory, any attempt to extrapolate from data in settings without sales taxes to the effects of sales taxes requires an untested hypothesis. And if we *do* have representative data on the past effect of sales taxes, the prediction problem does not involve a substantive counterfactual.[30]

Experimental Data. Experimental economists have a degree of control over the scope of their data that is not available in field studies. In particular, the experimentalist can choose to acquire a large number of observations for a fixed input space, so that nonparametric estimation of irreducible error for those inputs is feasible. Thus estimating completeness for laboratory data is feasible in many instances, as illustrated in the three applications in this paper. The main tradeoff is between gathering more instances of observations for a given set of feature values, versus ranging over a larger set of feature values. With a sufficiently large budget size, both may be

[30]Except in the trivial sense that any extrapolation from past data to future outcomes requires some form of inductive hypothesis.

possible.

Alternative Measures of Completeness. This paper estimates irreducible error nonparametrically, which is feasible when the number of data observations is large relative to the number of inputs. When this is not the case, econometric methods such as splines, sieves, and lasso regression can potentially be used as substitutes. [31].

In some cases, it may be possible to indirectly evaluate irreducible noise. For example, an interesting analogy to our approach to completeness is found in the literature on inheritability. Biologists have discovered a gap between two different methodologies for discovering how much of a particular outcome (say propensity to have a disease) is heritable, dubbed the 'missing heritability problem' (Manolio et al. (2009)). Traditional methods of measuring heritability, such as through carefully controlled twin studies, do not attempt to isolate individual genes. Newer measurement techniques instead allow us to postulate individual genes as the carrier of heritability. Yet for many outcomes, the explanatory power of individual genes has proven far smaller (sometimes by an order of magnitude) than overall measures of heritability suggest. This gap has motivated further theorizing and measurement to isolate where the "missing heritability" may lie. Roughly speaking, the aggregate measures of heritability are in effect being used as an analog of our completeness metric for the specific gene-based theories.

Measuring Portability. One important question for future work is how to compare the transferability of models across domains. Indeed, we may expect that economic models that are outperformed by machine learning models in a given domain have higher transfer performance outside of the domain. In this sense, within-domain completeness may provide an insufficient measure of the "overall completeness" of the model, and we leave development of such notions to future work.

References

Bar-Hillel, M. and W. Wagenaar (1991): "The Perception of Randomness," *Advances in Applied Mathematics*.

[31] These methods may have better finite-sample performance when suitable regularity assumptions apply, but those assumptions may not be directly testable.

Barberis, N., A. Shleifer, and R. Vishny (1998): "A Model of Investor Sentiment," *Journal of Financial Economics*.

Batzilis, D., S. Jaffe, S. Levitt, J. A. List, and J. Picel (2016): "How Facebook Can Deepen our Understanding of Behavior in Strategic Settings: Evidence from a Million Rock-Paper-Scissors Games," Working Paper.

Bernheim, D., C. Exley, J. Naecker, and C. Sprenger (2019): "The Model You Know: Generalizability and Predictive Power of Models of Choice Under Uncertainty," Working Paper.

Bodoh-Creed, A., J. Boenhke, and B. Hickman (2019): "Using Machine Learning to Explain Price Dispersion," Working Paper.

Bourgin, D. D., J. C. Peterson, D. Reichman, T. L. Griffiths, and S. J. Russell (2019): "Cognitive Model Priors for Predicting Human Decisions," *CoRR*, abs/1905.09397.

Bruhin, A., H. Fehr-Duda, and T. Epper (2010): "Risk and Rationality: Uncovering Heterogeneity in Probability Distortion," *Econometrica*.

Camerer, C. F., T.-H. Ho, and J.-K. Chong (2004): "A cognitive hierarchy model of games," *The Quarterly Journal of Economics*, 119, 861–898.

Camerer, C. F., G. Nave, and A. Smith (2018): "Dynamic unstructured bargaining with private information: theory, experiment, and outcome prediction via machine learning," *Management Science*.

Chen, D., K. Shue, and T. Moskowitz (2016): "Decision-Making under the Gambler's Fallacy: Evidence from Asylum Judges, Loan Officers, and Baseball Umpires," *Quarterly Journal of Economics*.

Crawford, V. P., M. A. Costa-Gomes, and N. Iriberri (2013): "Structural models of nonequilibrium strategic thinking: Theory, evidence, and applications," *Journal of Economic Literature*, 51, 5–62.

Domingos, P. (2000): "A Unified Bias-Variance Decomposition and its Applications," *Proc. 17th International Conf. on Machine Learning*.

Erev, I., A. E. Roth, R. L. Slonim, and G. Barron (2007): "Learning and equilibrium as useful approximations: Accuracy of prediction on randomly selected constant sum games," *Economic Theory*, 33, 29–51.

Fudenberg, D. and A. Liang (2019): "Predicting and Understanding Initial Play," *American Economic Review*.

Gneiting, T. and A. E. Raftery (2007): "Strictly Proper Scoring Rules, Prediction, and Estimation," *Journal of the American Statistical Association*.

Hastie, T., R. Tibshirani, and J. Friedman (2009): *The Elements of Statistical Learning*, Springer.

Kleinberg, J., H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan (2017): "Human Decisions and Machine Predictions," *The Quarterly Journal of Economics*.

Manolio, T. A., F. S. Collins, N. J. Cox, D. B. Goldstein, L. A. Hindorff, D. J. Hunter, M. I. McCarthy, E. M. Ramos, L. R. Cardon, A. Chakravarti, et al. (2009): "Finding the missing heritability of complex diseases," *Nature*, 461, 747.

Nagel, R. (1995): "Unraveling in Guessing Games: An Experimental Study," *American Economic Review*, 85, 1313–1326.

Nickerson, R. and S. Butler (2009): "On Producing Random Sequences," *American Journal of Psychology*.

Noti, G., E. Levi, Y. Kolumbus, and A. Daniely (2016): "Behavior-Based Machine-Learning: A Hybrid Approach for Predicting Human Decision Making," *CoRR*, abs/1611.10228.

Peysakhovich, A. and J. Naecker (2017): "Using Methods from Machine Learning to Evaluate Behavioral Models of Choice Under Risk and Ambiguity," *Journal of Economic Behavior and Organization*.

Plonsky, O., R. Apel, E. Ert, M. Tennenholtz, D. Bourgin, J. Peterson, D. Reichman, T. Griffiths, S. Russell, E. Carter, J. Cavanagh, and I. Erev (2019): "Predicting human decisions with behavioral theories and machine learning," *CoRR*, abs/1904.06866.

Plonsky, O., I. Erev, T. Hazan, and M. Tennenholtz (2017): "Psychological forest: Predicting human behavior," *AAAI Conference on Artificial Intelligence*.

Rabin, M. (2000): "Risk Aversion and Expected-utility Theory: A Calibration Theorem," *Econometrica*, 68, 1281–1292.

——— (2002): "Inference by Believers in the Law of Small Numbers," *The Quarterly Journal of Economics*.

Rabin, M. and D. Vayanos (2010): "The Gambler's and Hot-Hand Fallacies: Theory and Applications," *Review of Economic Studies*.

Rapaport, A. and D. Budescu (1997): "Randomization in Individual Choice Behavior," *Psychological Review*.

Samuelson, P. (1952): "Probability, Utility, and the Independence Axiom," *Econometrica*.

Savage, L. (1954): *The Foundations of Statistics*, J. Wiley.

Stahl, D. O. and P. W. Wilson (1994): "Experimental evidence on players' models of other players," *Journal of Economic Behavior and Organization*, 25, 309–327.

——— (1995): "On players' models of other players: Theory and experimental evidence," *Games and Economic Behavior*, 10, 218–254.

Tversky, A. and D. Kahneman (1971): "The Belief in the Law of Small Numbers," *Psychological Bulletin*.

——— (1992): "Advances in Prospect Theory: Cumulative Representation of Uncertainty," *Journal of Risk and Uncertainty*, 5, 297–323.

von Neumann, J. and O. Morgenstern (1944): *Theory of Games and Economic Behavior*, Princeton University Press.

Wright, J. R. and K. Leyton-Brown (2014): "Level-0 meta-models for predicting human behavior in games," *Proceedings of the fifteenth ACM conference on Economics and computation*, 857–874.

Appendix

A How Good is our Estimate of Irreducible Error?

In the main text, we present an approach for estimating irreducible error, where we estimate a "lookup table" function $\hat{f}_{LT}$ on training data (see (4) and the discussion following). Below we investigate whether the data sets we study are large enough for this to be a good approximation.

We first review some results from the machine learning and statistics literatures, which explain why the cross-validated standard errors that we report in the main text are informative about the quality of this approximation (Section A.1).

In Section A.2, we compare the out-of-sample performance of the lookup table with that of bagged decision trees, an algorithm that works better on smaller quantities of data. We find that in each of our prediction problems, the two prediction errors are similar, and the lookup table weakly outperforms bagged decision trees. Finally, in Section A.3, we study the sensitivity of the lookup table's performance to the quantity of data. The predictive accuracies achieved using our full data sets are very close to those achieved using, for example, just 70% of the data. This again suggests that only minimal improvements in predictive accuracy are feasible from further increases in data size.

A.1 Cross-Validated Standard Error

Suppose the loss function is mean-squared error: $\ell(y', y) = (y' - y)^2$. (Similar arguments apply for the misclassification rate; see e.g. Domingos (2000).) Let

$$f^*(x) = \mathbb{E}_P[y \mid x]$$

be the idealized prediction rule discussed in Section 1.2, which assigns to each x its expected outcome y under distribution P. Write $\hat{f}_{LT}[Z]$ for the *random* lookup table prediction rule that has been estimated from a set Z of n i.i.d. training observations. The expected mean-squared error of $\hat{f}_{LT}$ on a new observation $(x, y) \sim P$ can be

decomposed as follows (Hastie et al., 2009):

$$\mathbb{E}[(\hat{f}_{LT}[Z](x) - y)^2] = \underbrace{\mathbb{E}[(f^*(x) - y)^2]}_{\text{irreducible noise}} + \underbrace{\left(\mathbb{E}[\hat{f}_{LT}[Z](x)] - f^*(x)\right)^2}_{\text{bias}} + \underbrace{\mathbb{E}[(\hat{f}_{LT}[Z](x) - \mathbb{E}[\hat{f}_{LT}[Z](x)])^2]}_{\text{sampling error}}$$

where the expectation is both over the realization of the training data Z used to train $\hat{f}_{LT}$, and also over the realization of the test observation (x, y).

The first component is the *irreducible noise* introduced in (2). The second component, *bias*, is the mean-squared difference between the *expected* lookup table prediction and the prediction of the ideal prediction rule f^*. The final component, *sampling error*, is the variance of the lookup table prediction (reflecting the sensitivity of the algorithm to the training data).

Since $\hat{f}_{LT}$ is unbiased, the second component is zero. Thus, irreducible noise is the difference between the expected lookup table error and the sampling error of the lookup table predictor. As described in Section 1.3, we follow the standard procedure of using the *variance of the cross-validated prediction errors* to estimate the sampling error (Hastie et al., 2009). That is,

$$\mathbb{E}[(\hat{f}_{LT}[Z](x) - \mathbb{E}[\hat{f}_{LT}[Z](x)])^2] \approx \frac{1}{K}\text{Var}(\{CV_1, \ldots, CV_K\})$$

where CV_i is the prediction error for the i-th iteration of cross-validation. The right-hand side of the display is the square of the cross-validated standard errors reported in the main text; thus, we have from Tables 2, 3, and 5:

	Estimate of Irreducible Error	Sampling Error
Risk Preferences	65.58	9
Predicting Initial Play, Data Set A	0.41	<0.0001
Predicting Initial Play, Data Set B	0.34	<0.0001
Human Generation of Random Sequences	0.2441	<0.0001

A.2 Comparison with Scalable Machine Learning Algorithms

An alternative way to evaluate whether the out-of-sample performance of the lookup table approximates the best possible prediction accuracy is to compare it with the performance of other machine learning algorithms. Below we compare its performance with **bagged decision trees** (also known as *bootstrap-aggregated* decision trees). This algorithm creates several bootstrapped data sets from the training data by sampling with replacement, and then trains a **decision tree** on each bootstrapped training set. Decision trees are nonlinear prediction models that recursively partition the feature space and learn a (best) constant prediction for each partition element. The prediction of the bagged decision tree algorithm is an aggregation of the predictions of individual decision trees. When the loss function is mean-squared error, the decision tree ensemble predicts the average of the predictions of the individual trees. When the loss function is misclassification rate, the decision tree ensemble predicts based on a majority vote across the ensemble of trees.

Table 8 shows that for each prediction problem, the error of the bagged decision tree algorithm is comparable to and slightly worse than that of the lookup table. These results again suggest that our estimate of irreducible error is a reasonable approximation.

	Risk	Games A	Games B	Games C	Sequences
Bagged Decision Trees	65.65	0.45	0.36	0.29	0.2442
	(0.10)	(0.004)	(0.005)	(0.004)	(0.0005)
Lookup Table $\hat{f}_{LT}$	65.58	0.41	0.34	0.27	0.2441
	(3.00)	(0.005)	(0.006)	(0.005)	(0.0006)

Table 8: The lookup table outperforms Bagged Decision Trees in each of our prediction problems.

A.3 Performance of the Lookup Table on Smaller Samples

Finally, we report the lookup table's cross-validated performance on random samples of $x\%$ of our data, where $x \in \{10, 20, \ldots, 100\}$. For each x, we repeat the proce-

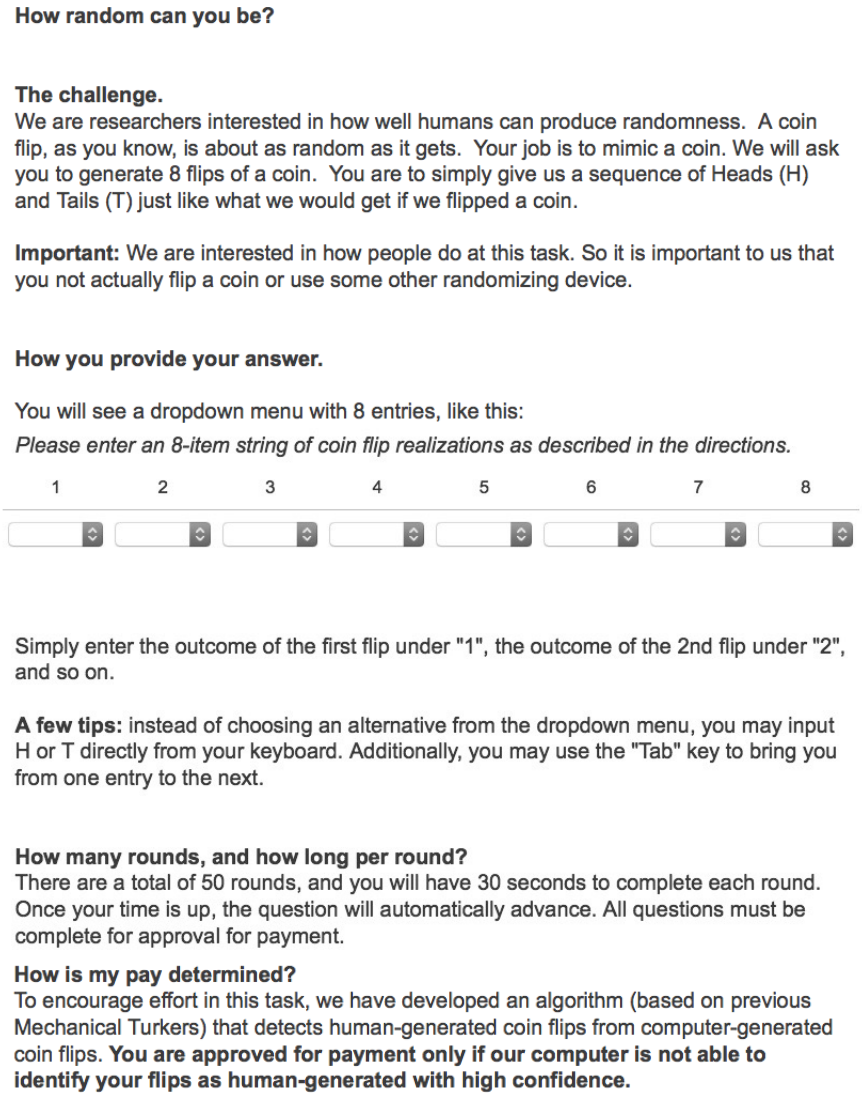
dure 1000 times, and report the average performance across iterations. We find that performance error flattens out for larger values of x, suggesting that the quantity of data we have is indeed large enough that further increases in the data size will not substantially improve predictive performance.

$x\%$	Risk	Games A	Games B	Games C	Sequences
10%	69.47	0.4191	0.3473	0.2729	0.2592
	(11.13)	(0.012)	(0.018)	(0.0015)	(0.0034)
20%	67.13	0.4183	0.3476	0.2718	0.2504
	(7.95)	(0.0018)	(0.024)	(0.0020)	(0.0018)
30%	66.28	0.4178	0.3472	0.2714	0.2479
	(6.51)	(0.0022)	(0.0029)	(0.0025)	(0.0014)
40%	66.25	0.4169	0.3470	0.2708	0.2464
	(5.65)	(0.0024)	(0.0032)	(0.0028)	(0.0011)
50%	65.68	0.4157	0.3459	0.2703	0.2458
	(4.59)	(0.0025)	(0.0036)	(0.0032)	(0.0010)
60%	65.68	0.4141	0.3449	0.2691	0.2452
	(4.24)	(0.0027)	(0.0040)	(0.0035)	(0.0008)
70%	65.68	0.4131	0.3435	0.2682	0.2448
	(3.95)	(0.0031)	(0.0045)	(0.0037)	(0.0007)
80%	65.68	0.4119	0.3427	0.2677	0.2445
	(3.95)	(0.0034)	(0.0046)	(0.0040)	(0.0007)
90%	65.66	0.4109	0.3416	0.2672	0.2443
	(3.71)	(0.0034)	(0.0047)	(0.0042)	(0.0007)
100%	65.58	0.4100	0.3404	0.2668	0.2441
	(3.00)	(0.0036)	(0.0051)	(0.0045)	(0.0006)

Table 9: Performance of Lookup Table $\hat{f}_{LT}$ using $x\%$ of the data, averaged over 100 iterations for each x

B Experimental Instructions for Section 2.3

Subjects on Mechanical Turk were presented with the following introduction screen:



How random can you be?

The challenge.
We are researchers interested in how well humans can produce randomness. A coin flip, as you know, is about as random as it gets. Your job is to mimic a coin. We will ask you to generate 8 flips of a coin. You are to simply give us a sequence of Heads (H) and Tails (T) just like what we would get if we flipped a coin.

Important: We are interested in how people do at this task. So it is important to us that you not actually flip a coin or use some other randomizing device.

How you provide your answer.

You will see a dropdown menu with 8 entries, like this:

Please enter an 8-item string of coin flip realizations as described in the directions.

1 2 3 4 5 6 7 8

Simply enter the outcome of the first flip under "1", the outcome of the 2nd flip under "2", and so on.

A few tips: instead of choosing an alternative from the dropdown menu, you may input H or T directly from your keyboard. Additionally, you may use the "Tab" key to bring you from one entry to the next.

How many rounds, and how long per round?
There are a total of 50 rounds, and you will have 30 seconds to complete each round. Once your time is up, the question will automatically advance. All questions must be complete for approval for payment.

How is my pay determined?
To encourage effort in this task, we have developed an algorithm (based on previous Mechanical Turkers) that detects human-generated coin flips from computer-generated coin flips. **You are approved for payment only if our computer is not able to identify your flips as human-generated with high confidence.**

C Supplementary Material to Section 2.3

C.1 Robustness

Here we check how our results in Section 2.3 change when the outcome space and error function are changed so that prediction functions are maps $f : \{H,T\}^7 \to \{H,T\}$ and the error for predicting the test data set $\{(s_i^1,\ldots,s_i^8\}_{i=1}^n$ is defined to be

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(s_i^8 \neq f(s_i^1,\ldots,s_i^7)),$$

i.e. the misclassification rate. We use as a naive benchmark the prediction rule that guesses H and T uniformly at random; this is guaranteed an expected misclassification rate of 0.50.

For this problem, we can estimate irreducible error by using a lookup table that learns the modal continuation for each sequence in $\{0,1\}^7$. We find that the completeness of Rabin (2002) and Rabin (2000) relative to this benchmark are respectively 19% and 9%.

	Error	Completeness
Naive Benchmark	0.50	0
Rabin (2002)	0.45	19%
	(0.003)	
Rabin & Vayanos (2010)	0.475	9%
	(0.01)	
Irreducible Error	0.23	1
	(0.002)	

C.2 Different Cuts of the Data

Initial strings only. We repeat the analysis in Section 2.3 using data from all subjects, but only their first 25 strings. This selection accounts for potential fatigue in generation of the final strings, and leaves a total of 638 subjects and 15,950 strings. Prediction results for our main exercise are shown below using this alternative selection.

	Error	Completeness
Naive Benchmark	0.25	0
Rabin & Vayanos (2010)	0.2491	5%
	(0.0008)	
Irreducible Error	0.2326	100%
	(0.0030)	

Removing the least random subjects. For each subject, we conduct a Chi-squared test for the null hypothesis that their strings were generated under a Bernoulli process. We order subjects by p-values and remove the 100 subjects with the lowest p-values (subjects whose generated strings were most different from what we would expect under a Bernoulli process). This leaves a total of 538 subjects and 24,550 strings. Prediction results for our main exercise are shown below using this alternative selection.

	Error	Completeness
Naive Benchmark	0.25	0
Rabin & Vayanos (2010)	0.2491	12%
	(0.0005)	
Irreducible Error	0.2427	100%
	(0.0016)	