

Level-0 Models for Predicting Human Behavior in Games

James R. Wright

*Computing Science Department,
University of Alberta
Edmonton, AB, Canada T6G 2E8*

JAMES.WRIGHT@UALBERTA.CA

Kevin Leyton-Brown

*Computer Science Department,
University of British Columbia,
Vancouver, BC, Canada V6T 1Z4*

KEVINLB@CS.UBC.CA

Abstract

Behavioral game theory seeks to describe the way actual people (as compared to idealized, “rational” agents) act in strategic situations. Our own recent work has identified iterative models, such as quantal cognitive hierarchy, as the state of the art for predicting human play in unrepeated, simultaneous-move games. Iterative models predict that agents reason iteratively about their opponents, building up from a specification of nonstrategic behavior called *level-0*. A modeler is in principle free to choose any description of level-0 behavior that makes sense for a given setting. However, in practice almost all existing work specifies this behavior as a uniform distribution over actions. In most games it is not plausible that even nonstrategic agents would choose an action uniformly at random, nor that other agents would expect them to do so. A more accurate model for level-0 behavior has the potential to dramatically improve predictions of human behavior, since a substantial fraction of agents may play level-0 strategies directly, and furthermore since iterative models ground all higher-level strategies in responses to the level-0 strategy. Our work considers models of the way in which level-0 agents construct a probability distribution over actions, given an arbitrary game. We considered a large space of alternatives and, in the end, recommend a model that achieved excellent performance across the board: a linear weighting of four binary features, each of which is general in the sense that it can be computed from any normal form game. Adding real-valued variants of the same four features yielded further improvements in performance, albeit with a corresponding increase in the number of parameters needing to be estimated. We evaluated the effects of combining these new level-0 models with several iterative models and observed large improvements in predictive accuracy.

1. Introduction

It is well known that the standard game-theoretic assumption that agents will adopt *Nash equilibrium* strategies—where each agent responds optimally to all the others—is often a poor predictor of actual human behavior (e.g., Goeree & Holt, 2001; Camerer, 2003; Kagel & Roth, 2016). This is a particular problem for researchers building artificially intelligent systems to interact with humans in strategic settings, such as randomized security protocols (e.g., Pita et al., 2008; Jain et al., 2010; Yin et al., 2012; Yang et al., 2013) or negotiation (e.g., Gal & Pfeffer, 2007; Wolpert & Bono, 2013). The field of *behavioral game theory* aims to develop models that more accurately describe human strategic behavior, as

evaluated using experimental data (Camerer, 2003). Our own recent work has identified one particular model, *quantal cognitive hierarchy*—an extension of the cognitive hierarchy model of Camerer, Ho, and Chong (2004)—as the state of the art behavioral model for predicting human play in unrepeated, simultaneous-move games (Wright & Leyton-Brown, 2012, 2017). The quantal cognitive hierarchy (QCH) model has two key components. The first component is *quantal response*; that is, agents respond stochastically to their incentives—playing high utility actions with high probability and low utility actions with low probability—rather than best responding. This expresses the intuition that two actions that yield roughly equal utilities have a roughly equal chance of being chosen. The second component is *iterative reasoning*; that is, agents do not reason arbitrarily deeply about their opponents’ beliefs about beliefs about beliefs, but instead start from a simple nonstrategic¹ (*level-0*) behavior, and then reason for some fixed number of iterations about responses to that starting point (e.g., a *level-2* agent quantally best responds to the combined behaviors of level-1 and level-0 agents).

Thus, in order to make use of a quantal cognitive hierarchy model one must first commit to a specification of level-0 behavior. Indeed, this is true of iterative models in general, such as cognitive hierarchy (Camerer et al., 2004) and level- k (Stahl & Wilson, 1994; Nagel, 1995; Costa-Gomes, Crawford, & Broseta, 2001). There are two key reasons why it is important to get this specification right. First, there is growing evidence that a substantial fraction of human players do act nonstrategically (e.g., Burchardi & Penczynski, 2014). Second, the level-0 model also drives predictions that are made at higher levels: higher-level agents are assumed to act by responding strategically to lower-level agents’ behavior, grounding out at level-0. Almost all work in the literature that uses iterative models adopts the specification that level-0 agents play a uniform distribution over actions. (In Section 5 we discuss the few exceptions of which we are aware, each of which is based on an explicitly encoded intuition about a specific setting of interest.) The uniform-distribution approach has the advantage that it does not require insight into a game’s structure, and hence can be applied to any game. However, in many games it is not plausible that an agent would choose an action uniformly at random, nor that any other agent would expect them to do so. For example, consider a dominated action that always yields very low payoffs for all players.

In this paper we consider the question of how to do better. Specifically, we investigate general rules that can be used to induce a level-0 specification from the normal-form description of an arbitrary game. In the next section we formally define our setting, and describe the data, methods, and model that we used in our work. In Section 3 we define the space of richer level-0 models that we considered. We searched this space in two different ways: manually via exhaustive evaluation and automatically via Bayesian optimization. We explain this methodology in more detail and analyze the resulting models in Section 4. Notably, we found that richer level-0 models improved performance across all of the iterative models that we considered, even though they were selected based only on their impact on the performance of the Poisson-QCH model. We briefly survey some related work in Section 5 before concluding in Section 6.

1. In this work, we consider “strategic” agents to be those whose actions can be described as responses to beliefs about the behavior of other agents; we call all other agents “nonstrategic.” Please see Section 3.2 for a detailed discussion of this distinction. Nonstrategic should not be taken as a synonym for thoughtless; indeed, some of the level-0 behavior that we describe below is rather sophisticated.

2. Framework

We focus on unrepeated, simultaneous-move *normal-form games*. A normal-form game G is defined by a tuple (N, A, u) , where $N = \{1, \dots, n\}$ is a finite set of *agents*; $A = A_1 \times \dots \times A_n$ is the set of possible *action profiles*; A_i is the finite set of *actions* available to agent i ; $u = \{u_i\}_{i \in N}$ is a set of *utility functions* $u_i : A \rightarrow \mathbb{R}$, each of which maps from an action profile to a utility for agent i . Let $\Delta(X)$ denote the set of probability distributions over a finite set X . Overloading notation, we represent the expected utility of a profile of mixed strategies $s \in S = \Delta(A_1) \times \dots \times \Delta(A_n)$ by $u_i(s)$. By a_{-i} we refer to the joint actions of all agents except i . Our objective is to find a behavioral model that maps from a game G and a (human) agent i to a probability distribution over i 's action set A_i that predicts i 's behavior in G . In this section we first describe a model that we have shown to achieve state-of-the-art performance in predicting human behavior, Poisson-QCH (Wright & Leyton-Brown, 2017). We then describe the dataset and methods that we used to learn parameters and evaluate the performance of extensions to Poisson-QCH.

2.1 Quantal Cognitive Hierarchy

The first key component of the quantal cognitive hierarchy model is *quantal best response*, in which errors become less likely as they become more costly. Like most of the behavioral game theory literature, we use the logit specification of this concept.

Definition 1 (Quantal best response). Let $u_i(a_i, s_{-i})$ be agent i 's expected utility when playing action $a_i \in A_i$ against mixed strategy profile s_{-i} in game G . Then a *quantal best response* $QBR_i(s_{-i}; G, \lambda)$ by agent i to s_{-i} is a mixed strategy s_i such that

$$s_i(a_i) = \frac{\exp[\lambda \cdot u_i(a_i, s_{-i})]}{\sum_{a'_i \in A_i} \exp[\lambda \cdot u_i(a'_i, s_{-i})]}, \quad (1)$$

where λ (the *precision*) represents agents' sensitivity to utility differences. For a mixed strategy profile s we sometimes write $QBR(s; G, \lambda)$ to represent a profile of strategies s' with $s'_i = QBR_i(s_{-i}; G, \lambda)$. We suppress reference to G when it is clear from context. \square

Unlike classical best response, which is a set-valued function, quantal best response always returns a single mixed strategy. When $\lambda = 0$, quantal response mixes uniformly over all of the agents' actions; as $\lambda \rightarrow \infty$, quantal best response approaches best response.²

The second key component of the quantal cognitive hierarchy model is *iterative response*, in which higher-level agents reason about and respond to lower-level agents. Describing the distribution of different levels in the population of agents is a crucial decision. We use the single-parameter Poisson distribution to specify this distribution, as in the work of Wright and Leyton-Brown (2017).

The Poisson-QCH model is a QCH model where the distribution of agent types follows a Poisson distribution.

2. More precisely, quantal best response approaches best response when best response returns a singleton set. When best response returns a set containing multiple elements, quantal best response approaches a mixed strategy that uniformly randomizes among those elements.

Definition 2 (Poisson-QCH model). Let $\pi_{i,m} \in \Delta(A_i)$ be the distribution over actions predicted for an agent i with level m . Let

$$\pi_{i,0:m} = \sum_{\ell=0}^m \frac{\text{Poisson}(\ell; \tau)}{\sum_{\ell'=0}^m \text{Poisson}(\ell'; \tau)} \pi_{i,\ell}$$

be the truncated distribution over actions predicted for an agent conditional on that agent's having level $0 \leq \ell \leq m$. Let $\pi_{-i,0:m}$ be the truncated distribution over the action profiles predicted for agents other than i , conditional on those agents' having level $0 \leq \ell \leq m$. Now we can define the $\pi_{i,m}$ predicted by the Poisson-QCH model:

$$\begin{aligned} \pi_{i,0}(a_i) &= |A_i|^{-1} \\ \pi_{i,m}(a_i) &= QBR_i(\pi_{-i,0:m-1}; \lambda). \end{aligned}$$

Notice that $\pi_{i,0}$ is just the uniform distribution; that is, level-0 agents randomize uniformly among their actions. In Sections 3 and 4, we will construct more plausible definitions of level-0 behavior by replacing $\pi_{i,0}$ with richer distributions. This baseline model thus has two parameters: the mean of the Poisson distribution τ and the precision λ . The overall predicted distribution of actions is a weighted sum of the distributions for each level:

$$\Pr(a_i | G, \tau, \lambda) = \sum_{\ell=0}^{\infty} \text{Poisson}(\ell; \tau) \pi_{i,\ell}(a_i). \quad \square$$

2.2 Data

We analyzed data from the ten experimental studies summarized in Table 1.

Several studies (Stahl & Wilson, 1994, 1995; Haruvy, Stahl, & Wilson, 2001; Haruvy & Stahl, 2007; Stahl & Haruvy, 2008) paid participants according to a randomized procedure in which experimental subjects played normal-form games for points representing a 1% chance (per game) of winning a cash prize. In the work of Costa-Gomes, Crawford, and Broseta (1998), each payoff unit was worth 40 cents, but participants were paid based on the outcome of only one randomly-selected game. In the remaining studies (Goeree & Holt, 2001; Cooper & Van Huyck, 2003; Costa-Gomes & Weizsäcker, 2008; Rogers, Palfrey, & Camerer, 2009), game payoffs were worth a deterministic number of cents. We summarize the expected value of payoff points in the “Units” column of Table 1.

Goeree and Holt (2001) presented 10 games in which subjects' behavior was close to that predicted by Nash equilibrium, and 10 other small variations on the same games in which subjects' behavior was *not* well-predicted by Nash equilibrium. We included the 10 games that were in normal form. In the work of Cooper and Van Huyck (2003), agents played the normal forms of 8 games, followed by extensive form games with the same induced normal forms; we include only the data from the normal-form games. The remaining studies consisted exclusively of normal-form games.

We represent the data for each game G_r as a pair $(G_r, \{a_{rj}\})$ containing the game itself and a set of observed actions in the game. All games had two players, so each single play of a game generated two observations. We built one such dataset for each study, as listed in Table 1. We also constructed a combined dataset, dubbed ALL10, containing data from all

Table 1: Names and contents of the datasets we analyzed. The “Units” column gives the expected value of each payoff point in US dollars; in the ALL10 dataset, earnings are all normalized to \$0.01 per point. The “Earnings Range” column gives the range of possible earnings for each subject. The “Obs.” column gives the number of observations. The “Games” column gives the number of games we included from the study, with the number of included games that were symmetric in parentheses.

Name	Source	Games (sym.)	Obs.	Units	Earnings Range
SW94	Stahl and Wilson (1994)	10 (10)	400	\$0.025	\$0.25 – \$25.00
SW95	Stahl and Wilson (1995)	12 (12)	576	\$0.02	\$0.00 – \$24.00
CGCB98	Costa-Gomes et al. (1998)	18 (0)	1566	\$0.022	\$7.84 – \$36.16
GH01	Goeree and Holt (2001)	10 (4)	500	\$0.01	\$-1.02 – \$23.30
CVH03	Cooper and Van Huyck (2003)	8 (0)	2992	\$0.10	\$0.80 – \$4.80
HSW01	Haruvy et al. (2001)	15 (15)	869	\$0.02	\$0.00 – \$30.00
HS07	Haruvy and Stahl (2007)	20 (20)	2940	\$0.02	\$1.05 – \$17.40
CGW08	Costa-Gomes and Weizsäcker (2008)	14 (0)	1792	\$0.0107	\$1.83 – \$14.13
SH08	Stahl and Haruvy (2008)	18 (18)	1288	\$0.02	\$0.00 – \$17.55
RPC09	Rogers et al. (2009)	17 (12)	1210	\$0.01	\$2.31 – \$13.33
ALL10	Union of above	142 (91)	13863	\$0.01	n/a

the datasets. The datasets contained very different numbers of observations, ranging from 400 (Stahl & Wilson, 1994) to 2992 (Cooper & Van Huyck, 2003). To ensure that each fold had approximately the same population of subjects and games,³ we evaluated ALL10 using *stratified* cross-validation, ensuring that the number of games from each source dataset was approximately equal in each partition element (e.g., Murphy, 2012). We also renormalized all games so that their payoffs were expressed in expected cents. This is important because the precision parameter for quantal response is not *scale invariant*: the correct value of λ can differ depending upon the units in which payoffs are expressed.

2.3 Methods

Models cannot be evaluated unless their parameters are instantiated. We set parameters to values that maximized the likelihood the model assigned to a *training dataset*. We performed this likelihood maximization using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithm (Hansen & Ostermeier, 2001). We then scored a given model’s performance by the likelihood it assigned to a *test dataset*, consisting entirely of games (and, hence, observations) that were not used for estimating parameters.

3. Each fold had only an approximately equal number of games under this scheme because the number of games in most source datasets was not an exact multiple of 10.

We randomly divided our data into training and test datasets using *10-fold cross-validation*. Specifically, for each round, we randomly ordered the games and then divided them into 10 equal-sized parts. For each of the 10 ways of selecting 9 parts from the 10, we computed the maximum likelihood estimate of the model’s parameters based on the observations associated with the games of those 9 parts. We then determined the likelihood of the remaining part given the prediction. We call the average of this quantity across all 10 parts the *cross-validated likelihood*. Randomly dividing our experimental data into training and test sets introduces variance into the prediction score, since the exact value of the score depends partly upon the random division. To reduce this variance, we performed 10 rounds of 10-fold cross-validation, and report the average of these 10 rounds (the *average cross-validated likelihood*). This allows us to estimate the variance that the division into training and test sets introduced, in a way that simply performing a single round of 100-fold cross-validation would not. The average cross-validated likelihood is distributed according to a Student’s-*t* distribution (e.g., Witten & Frank, 2000). We compare the predictive power of different behavioral models on a given dataset by comparing the average cross-validated likelihood of the dataset under each model. We say that one model predicted significantly better than another when the 95% confidence intervals for the average cross-validated likelihoods do not overlap.

It can also be desirable to analyze and interpret the values taken by parameters in a model. Point estimates—such as values that maximize likelihood—can be unstable and/or misleading. We advocate instead for computing posterior distributions over a model’s parameters, estimated via a set of samples. In this paper (see Section 4.3) we used the *Metropolis–Hastings* algorithm, as implemented by the PyMC software package (Patil, Huard, & Fonnesbeck, 2010), to generate such samples. The Metropolis–Hastings algorithm is a Markov Chain Monte Carlo (MCMC) algorithm (e.g., Robert & Casella, 2004) that computes a series of values from the support of a distribution. MCMC algorithms (and related techniques such as annealed importance sampling (Neal, 2001)) are useful for estimating multidimensional distributions for which a closed form of the density is unknown. We used a flat (i.e., uniform) prior for all parameters. For precision parameters, another natural choice might have been to use a flat prior on the log of precision. In this work, we wanted to avoid artificially preferring precision estimates closer to zero, since it is common for iterative models to assume agents best respond nearly perfectly to lower levels (that is, that they have infinitely high precisions). Our posterior precision estimates tended to be concentrated near zero regardless.

3. Level-0 Model

In this section we present components from which we can construct models for computing level-0 distributions of play. We begin by precisely defining “nonstrategic” behavior. We then describe specific nonstrategic features computed for each of an agent’s actions, followed by options for combining feature values to obtain a level-0 prediction.

3.1 Nonstrategic Behavior

Most applications of iterative models specify that level-0 agents choose their actions uniformly at random, thus implicitly equating nonstrategic behavior with uniform randomiza-

tion. This paper is motivated by the observation that nonstrategic behavior need not be uniform. How then might a nonstrategic agent behave?

We refer to the process of responding to explicit beliefs about the behavior of other agents as *strategic* behavior, and refer to all other behavior as *nonstrategic*. In QCH, agents of level 1 or higher are strategic in this sense; they quantally best respond to their beliefs about the actions of lower-level agents.

Nonstrategic behavior may take account of payoffs to varying degrees. At the most extreme, it is easy to see that uniform behavior is nonstrategic, as it depends neither on an agent’s own payoffs, nor those of the other agents. Behavior that depends on only an agent’s own payoffs—such as the maxmax payoff feature of Section 3.2—is also nonstrategic, since the agent cannot plausibly form beliefs about the other agents’ behavior without referencing their payoffs. Even some behavior that depends on the payoffs of all agents may be nonstrategic, when the payoffs are not used for forming and responding to explicit beliefs about the other agents’ actions. Here, however, it is necessary (and nontrivial) to show that behavior that is not couched in terms of responding to explicit beliefs furthermore cannot be represented as though it did respond to some such beliefs. For example, the maxmax payoff feature can be expressed as a best response to a particular probabilistic belief (namely, that the other agents will play actions such that the maxmax action will actually yield its maximum payoff).

Formalizing nonstrategic behavior is thus trickier than it might seem; doing so rigorously is the subject of a companion paper (Wright & Leyton-Brown, 2018). We summarize some key take-aways here. In that work we define as nonstrategic any behavior that can be computed via a finite combination of *elementary agent models*—that is, of functions that score each action available to an agent based upon a *single* scalar for each action profile, rather than a full tuple of separate payoffs for each agent. This outcome score may be the agent’s own payoff or some combination of all agents’ payoffs; however, it must not be possible in general to recover the original tuple of payoffs from the outcome score.⁴ The essential intuition captured by this definition is that a nonstrategic agent does not independently reason about the actions of the other agents as distinct from the agent’s own actions.

Definition 3 (Elementary agent model). An *agent model* for agent i is a function $f_i(G)$ that maps from a normal-form game G to a vector of reals with dimension $|A_i|$ (i.e., one real number for each of i ’s actions). An agent model is *elementary* if it can be computed as $f_i(G) = h_i(\Phi(G))$, where:

- $\Phi(G)_a = \varphi(u(a))$ for every action profile a ,
- $\varphi(u(a)) = w^T u(a)$. That is, $\varphi(u(a))$ is computed by taking a linear combination of the players’ utilities at pure action profile a , with the weights defined by a vector $w \in R^n$.⁵

4. Here, let us assume that outcome scores are linear combinations of each outcome’s payoffs. More broadly, we can generalize beyond linearity.

5. In the companion paper, we use a weaker condition on φ that generalizes linear combinations. However, every feature that we consider here can be represented as a linear combination, so we use this more restrictive definition of elementary agent models.

- h is an arbitrary function.

Clearly any behavior that can be computed using only the agent’s own utilities (such as the maxmax payoff feature) can be computed by an elementary agent model with $w = (1, 0, \dots, 0)$. However, some behavior that takes account of other agents’ utilities can also be computed—e.g., the maxmax welfare feature, which recommends actions leading to the highest total payoffs among all agents, can be calculated using only the sum of all agents’ utilities, corresponding to an elementary agent model with $w = (1, 1, \dots, 1)$.

The key result of our companion paper is that restricting attention to behavior that can be computed by elementary agent models is a substantive restriction. Of particular relevance to our discussion here, we prove that it is impossible to define QCH using only finite combinations of elementary agent models (Wright & Leyton-Brown, 2018). This justifies our calling level-0 behaviors nonstrategic and level- $k > 0$ behaviors strategic: the mathematical structure of QCH ensures that this will be true in general.

3.2 Level-0 Features

The models we considered are driven by the idea of simple rules (*features*) that recommend one or more actions to greater or lesser degrees. We restrict our attention to features that can be computed directly from the normal form of the game and that do not depend on presentation details such as the units in which payoffs are expressed or the order in which actions are presented. This allows for more accurate analysis of strategic models, even when details of the presentation are unknown or not yet known. We do not claim that the features that we investigated comprise an exhaustive list of factors that could influence nonstrategic agents’ actions. Of course, we also restrict ourselves to features that are nonstrategic in the sense of not being responses to explicit beliefs about the other agents’ behavior. Following the argument just given, we formalize this restriction as requiring that each feature be an elementary agent model as in Definition 3.

For each feature, we briefly describe its motivation and then give a formal definition. Many of our features have been investigated in both the classical and behavioral game theory literature in other contexts. To give one example, the maxmax payoff, maxmin payoff, and maxmax welfare features correspond to the *Optimistic*, *Pessimistic*, and *Altruistic* nonstrategic types in the work of Costa-Gomes et al. (2001). We obtained other features, such as *max-symmetric*, via introspection about paradigmatic games such as the Traveler’s Dilemma.

For each feature, we define both a binary version with range $\{0, 1\}$ and a real-valued version with range \mathbb{R}^+ . Binary features recommend one or more actions that maximize some criterion. In contrast, the corresponding real-valued feature recommends each action to the *degree* that it maximizes the same criterion. This addresses the intuition that two very high payoff actions may both be attractive, even if one offers marginally higher payoff. We indicate real-valued features with the modifier $[R]$. Some real-valued features represent quantities that an agent would wish to minimize, rather than maximize. We apply the inv transformation to these features, where inv is defined differently depending upon how features will be combined: if feature values will be combined linearly, then $\text{inv}(x) = 1/x$; if feature values will be combined with a logit function, then $\text{inv}(x) = -x$.

Maxmin payoff. A maxmin action for agent i is the action with the best worst-case guarantee. That is,

$$f^{\text{maxmin}}(a_i) = \begin{cases} 1 & \text{if } a_i \in \arg \max_{a'_i \in A_i} \min_{a_{-i} \in A_{-i}} u_i(a'_i, a_{-i}), \\ 0 & \text{otherwise.} \end{cases}$$

This is an action that is weakly safer to play against a hostile agent than any other individual action; there may be multiple such actions.⁶ The real-valued version of this feature returns the worst-case payoff for an action:

$$f^{\text{min[R]}}(a_i) = \min_{a_{-i} \in A_{-i}} u_i(a_i, a_{-i}).$$

Maxmax payoff. In contrast, a maxmax action for agent i is the best action in the best case. That is,

$$f^{\text{maxmax}}(a_i) = \begin{cases} 1 & \text{if } a_i \in \arg \max_{a'_i \in A_i} \max_{a_{-i} \in A_{-i}} u_i(a'_i, a_{-i}), \\ 0 & \text{otherwise.} \end{cases}$$

An agent who aims to maximize his possible payoff will play a maxmax action. The real-valued version of this feature returns the best-case payoff for an action:

$$f^{\text{max[R]}}(a_i) = \max_{a_{-i} \in A_{-i}} u_i(a_i, a_{-i}).$$

Minimax regret. Savage (1951) proposed the *minimax regret* criterion for making decisions in the absence of probabilistic beliefs. In a game theoretic context, it works as follows. For each action profile, an agent has a possible *regret*: how much more utility could the agent have gained by playing the best response to the other agents' actions? Each of the agent's actions is therefore associated with a vector of possible regrets, one for each possible profile of the other agents' actions. A minimax regret action is an action whose maximum regret (in the vector of possible regrets) is minimal. That is, if

$$r(a_i, a_{-i}) = \max_{a_i^* \in A_i} u_i(a_i^*, a_{-i}) - u_i(a_i, a_{-i})$$

is the regret of agent i in action profile (a_i, a_{-i}) , then

$$f^{\text{mmr}}(a_i) = \begin{cases} 1 & \text{if } a_i \in \arg \min_{a'_i \in A_i} \max_{a_{-i} \in A_{-i}} r(a'_i, a_{-i}), \\ 0 & \text{otherwise.} \end{cases}$$

The real-valued version of this feature returns the worst-case regret for playing an action:

$$f^{\text{mmr[R]}}(a_i) = \text{inv} \left[\max_{a_{-i} \in A_{-i}} r(a_i, a_{-i}) \right].$$

We apply the *inv* transformation to model a preference for lower maximum regret.

6. Often, a mixed strategy will be safer still against a hostile agent. However, in this application we are not actually trying to find a safest strategy for the agent. Rather, we are trying to specify features of individual actions that might make them attractive to nonstrategic agents.

Maxmax fairness. Concern for the fairness of outcomes is a common feature of human play in strategic situations, as has been confirmed in multiple behavioral studies, most famously in the Ultimatum game (Thaler, 1988; Camerer & Thaler, 1995). Let the unfairness of an action profile be the difference between the maximum and minimum payoffs among the agents under that action profile:

$$d(a) = \max_{i,j \in N} u_i(a) - u_j(a).$$

Then a *fair* outcome minimizes this difference in utilities. The *maxmax fairness* feature selects every action which is part of a maximally fair action profile.

$$f^{\text{fair}}(a_i) = \begin{cases} 1 & \text{if } a_i \in \arg \min_{a'_i \in A_i} \min_{a_{-i} \in A_{-i}} d(a'_i, a_{-i}), \\ 0 & \text{otherwise.} \end{cases}$$

The real-valued version of this feature returns the maximum fairness that could result from playing a given action:

$$f^{\text{fair[R]}}(a_i) = \text{inv} \left[\min_{a_{-i} \in A_{-i}} d(a_i, a_{-i}) \right].$$

We apply the inv transformation to translate from unfairness to fairness.

Max symmetric. In a symmetric game, one simple way to guess how other agents would act is to assume that they will act in the same way as oneself. We say that a utility function u is symmetric if all agents' action sets are the same size, and if when two agents swap actions under any action profile for the other agents, they also swap utilities. Formally (restricting to the case of two agents to simplify notation), we define the proposition

$$\text{Symm}(u) \Leftrightarrow \forall i, j \in N, |A_i| = |A_j| \wedge \forall a_i, a_j \in A_i, u_i(a_i, a_j) = u_j(a_j, a_i).$$

A max-symmetric action is then the agent's part of the highest-utility profile of identical actions:

$$f^{\text{maxsymm}}(a_i) = \begin{cases} 1 & \text{if } \text{Symm}(u) \wedge a_i \in \arg \max_{a'_i \in A_i} u_i(a'_i, \dots, a'_i), \\ 0 & \text{otherwise.} \end{cases}$$

Note that if a game is not symmetric, this feature is still well defined, but evaluates to zero across all actions.

The real-valued version of this feature returns the symmetric payoff of an action for symmetric games, or 0 otherwise:

$$f^{\text{symm[R]}}(a_i) = \begin{cases} 0 & \text{if } \neg \text{Symm}(u), \\ u_i(a_i, \dots, a_i) & \text{otherwise.} \end{cases}$$

Maxmax welfare. Finally, one reason that a nonstrategic agent might find an action profile desirable is that it produces the best overall benefit to the agents collectively. The

maxmax welfare feature selects every action that is part of some action profile that maximizes the sum of utilities:

$$f^{\text{welfare}}(a_i) = \begin{cases} 1 & \text{if } a_i \in \arg \max_{a'_i \in A_i} \max_{a_{-i} \in A_{-i}} \sum_{j \in N} u_j(a'_i, a_{-i}), \\ 0 & \text{otherwise.} \end{cases}$$

The real-valued version of this feature returns the maximum welfare that could result from playing a given action:

$$f^{\text{welfare[R]}}(a_i) = \max_{a_{-i} \in A_{-i}} \sum_{j \in N} u_j(a_i, a_{-i}).$$

3.3 Combining Feature Values

Once a set of features has been computed for each of a set of actions, their values must be combined to yield a single distribution over actions. We considered two functional forms, inspired by linear regression and logistic regression. Both specifications accept a set of features and a set of weights. Let F be a set of features mapping from an action to \mathbb{R}^+ . For each feature $f \in F$, let $w_f \in [0, 1]$ be a weight parameter with $\sum_{f \in F} w_f \leq 1$; let $w_0 = 1 - \sum_{f \in F} w_f$.

Our first functional form produces a level-0 prediction over actions for a given agent by taking a weighted sum of feature outputs for each action and then normalizing to produce a distribution.

Definition 4 (Weighted linear level-0 specification). The *weighted linear level-0 specification* predicts the following distribution of actions for level-0 agents:

$$\pi_{i,0}^{\text{linear},F}(a_i) = \frac{w_0 + \sum_{f \in F} w_f f(a_i)}{\sum_{a'_i \in A_i} [w_0 + \sum_{f \in F} w_f f(a'_i)]}.$$

The second functional form assigns a level-0 probability proportional to the exponential of a weighted sum of feature values.

Definition 5 (Logit level-0 specification). The *logit level-0 specification* predicts the following distribution of actions for level-0 agents:

$$\pi_{i,0}^{\text{logit},F}(a_i) = \frac{\exp(w_0 + \sum_{f \in F} w_f f(a_i))}{\sum_{a'_i \in A_i} \exp(w_0 + \sum_{f \in F} w_f f(a'_i))}.$$

3.4 Feature Transformations

In addition to two functional forms for combining the feature values, we also evaluated two transformations to feature values. If employed, these transformations are applied to each feature value before the features are weighted and combined.

The first transformation zeroes out features that have the same value for every action, which we call *uninformative*. The intuition behind this transformation is that informative features should have a greater influence on the prediction precisely when the other features are less informative. This transformation is useful only when working with the linear

functional form: it has no effect under the logit functional form, since it corresponds to a constant translation by the magnitude of the zeroed-out feature, and the logit functional form is invariant to translations.

Definition 6 (Informativeness feature transformation). A feature f is *informative* in a game G if there exists $a'_i, a''_i \in A_i$ such that $f(a'_i) \neq f(a''_i)$. The informativeness transformation $I(f)$ of a feature f returns the feature’s value when it is informative, and zero otherwise:

$$I(f)(a_i) = \begin{cases} f(a_i) & \text{if } \exists a'_i, a''_i \in A_i : f(a'_i) \neq f(a''_i), \\ 0 & \text{otherwise.} \end{cases}$$

The hypothesis represented by this transformation is that people will ignore features that do not distinguish between actions, even if those features have high weights. When a feature is uninformative due to not recommending any action, the informativeness transformation has no effect. However, when a feature is uninformative due to recommending *every* action, the informativeness transformation allows the model to give higher relative weight to informative features. To see why, consider two features, one with a weight of 0.6 and one with a weight of 0.3 (meaning that the noise weight $w_0 = 0.1$). If the high-weight feature recommends both actions of a two-action game and the low-weight feature recommends only the first action, then the untransformed linear model would assign probabilities $\frac{0.6+0.3+0.1}{1.7} = 0.59$ and $\frac{0.6+0.1}{1.7} = 0.41$ to the two actions. When the informativeness transformation is applied, however, the linear model would assign much less noisy probabilities: $\frac{0.3+0.1}{0.5} = 0.8$ and $\frac{0.1}{0.5} = 0.2$.

The second transformation normalizes feature values to $[0, 1]$. This limits the degree to which one real-valued feature can overwhelm other features.

Definition 7 (Normalized activation feature transformation). The normalized activation transformation $N(f)$ constrains a feature f to take nonnegative values that sum to 1 across all of a game’s actions:

$$N(f)(a_i) = \frac{f^+(a_i)}{\sum_{a'_i \in A_i} f^+(a'_i)},$$

where

$$f^+(a_i) = f(a_i) - \min\{0, \min_{a''_i \in A_i} f(a''_i)\}.$$

4. Model Selection

We took two approaches to constructing models from the candidate features, functional forms, and transformations described in the previous section. First, we exhaustively evaluated all subsets of binary features using the linear functional form and the informativeness transformation (Section 4.1). We chose this functional form based on a manual evaluation we performed in the experiments described by the conference version of this paper (Wright & Leyton-Brown, 2014), in which a linear functional form and normalized activation- and informativeness-transformed binary features yielded good performance. Second, although the full space of candidate features, functional forms, and transformations was too large to permit us to test every combination, we used a Bayesian optimization technique to seek

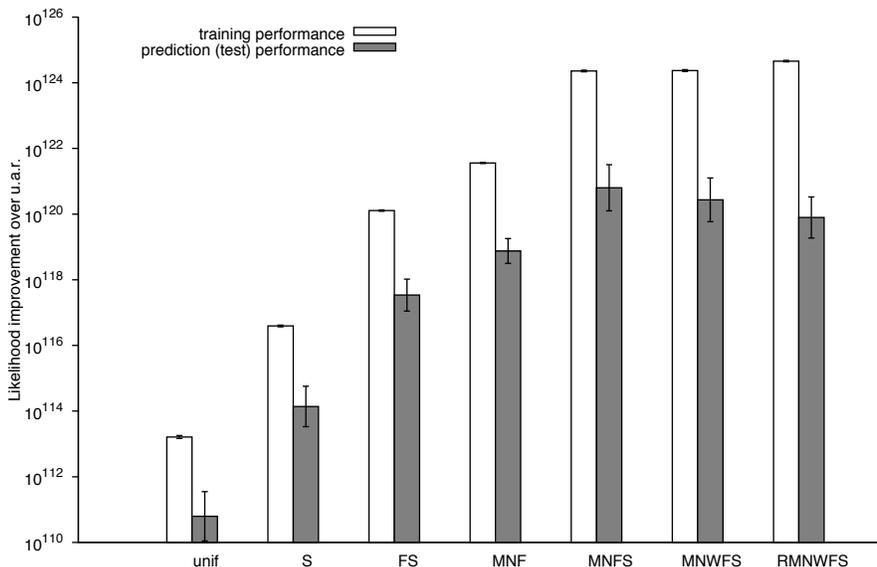


Figure 2: Training and prediction performance with 95% confidence intervals for Poisson-QCH extended by binary features, for the best performing set of features chosen by exhaustive evaluation at each number of features. Feature sets are specified by the following code: (M) maxmax payoff; (N) maxmin payoff; (R) minmax regret; (W) maxmax welfare; (F) maxmax fairness; (S) max symmetric.

the likelihood of a uniform random prediction.⁷ All else being equal, a model with higher performance is more desirable, as is a model with fewer parameters. Figure 1 includes an *efficient frontier* of those models that achieved the best performance for each given number of parameters or fewer.⁸

The best performing linear model contained four features: maxmax payoff, maxmin payoff, maxmax fairness, and max symmetric; we dub this model `linear4`. Adding further features did not improve prediction performance. Two obvious hypotheses could explain why adding extra features harmed test performance: *overfitting* (our model was overly flexible and so fit noise in the training dataset that was not duplicated in the test dataset) and *underfitting* (our optimization procedure failed to effectively optimize the more complex models). Figure 2 shows the training and test performance for the best-performing model at each number of features. Notice that training performance increased with every additional feature, whereas test performance increased for the first four features and then decreased. This supports the hypothesis that overfitting was the cause of our observed performance decrease.

7. The numbers on this axis are very large, as the likelihood ratios are exponential in the number of observations in the dataset.

8. The max-symmetric feature was selected first by the exhaustive search, and was included in every model on the efficient frontier. Clearly this would not have happened in a dataset that did not include symmetric games. However, this does not mean that the selected model fails to generalize to asymmetric games; in such games, the model acts as if the max-symmetric feature was not included.

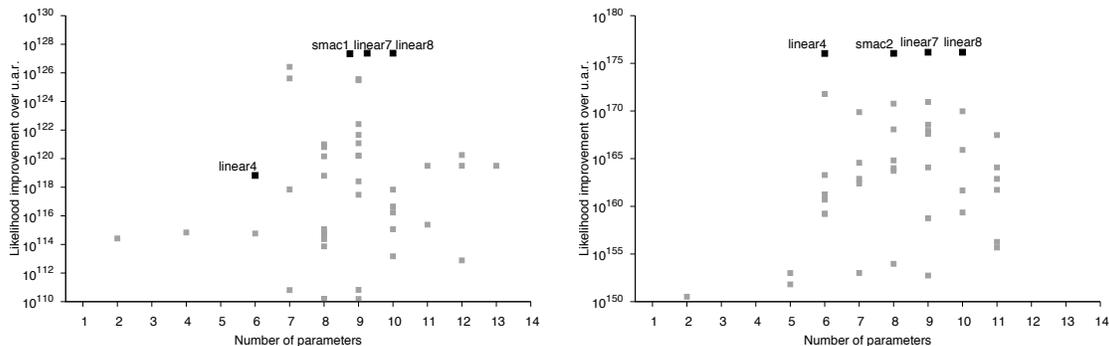
4.2 Bayesian Optimization

We performed Bayesian optimization using SMAC (Hutter, Hoos, & Leyton-Brown, 2010, 2011, 2012). While SMAC was originally designed to optimize the configuration of algorithms, it has since proven effective for a variety of other black-box function optimization tasks (Hutter, Hoos, & Leyton-Brown, 2013; Thornton, Hutter, Hoos, & Leyton-Brown, 2013). SMAC evaluates each configuration on a randomly-chosen *instance* (i.e., input to the algorithm); it then updates a random forest model of predicted performance for configurations. It determines which configurations to evaluate based on this performance model using the *expected improvement* criterion. A model’s expected improvement is its expected performance, conditional on its performance exceeding that of the current best model. This provides a natural balance between exploration and exploitation, as models that are very likely to have high performance are favored, but so are models whose performance is very uncertain.

During Bayesian optimization, we need to train and evaluate candidate models on separate data. This evaluation must not involve any of our held-out test data, to ensure that our evaluation of final models remains unbiased. For the experiments described in this section, we randomly selected 10% of the ALL10 dataset as a held-out test set. This test set was used only for generating the final performance estimates that we report in this paper. The remaining 90% of the data was used as a training data set (80% of the original data) and a validation set (10% of the original data). Each model selected by SMAC for evaluation was first trained on the training set and then evaluated on the validation set. The results of the validation were reported to SMAC. When evaluating the results of the run to generate figures, we retrained each model using the combination of the training and validation sets, and evaluated the model’s performance on the held-out test set.

A consequence of this setup is that we do not obtain error bounds for each model, unlike the procedure we used in Section 2.2. (Recall that the latter procedure obtained error bounds by cross-validating results from multiple divisions of the data.) We used a different procedure for our SMAC experiments for two reasons. First, the use of a single data division allowed SMAC to rapidly search a much larger number of models. Second, we needed separate validation and test sets: the first to allow SMAC to estimate generalization performance, and the second to evaluate the results of the Bayesian optimization. To get some sense of what variation was due to training/validation/test set splits, we did repeat the procedure using two independent random divisions. More specifically, we ran 22 parallel SMAC processes for 380 hours each. Half of these processes operated on one random split of the data (Figure 3(a)), and half on another random split (Figure 3(b)). In addition to the incumbent models, Figure 3 includes the following models for comparison: `linear4`, `linear7` (defined in Section 4.4), and `linear8` (defined later in this section).

Out of the 1766 models evaluated by SMAC, there were 42 unique “incumbents”—models that were considered the best by SMAC at some point in the search. Figure 3 shows the predictive performance achieved by each of these incumbent models in the divisions in which they were found, again grouped by number of parameters. The best-performing model found by SMAC in the first random division (henceforth `smac1`) was a 9-parameter model that linearly combined the following features: `max payoff[R]`, `min payoff[R]`, `fairness[R]`, `symmetric payoffs[R]`, `maxmax payoff`, `maxmax welfare`, and `max symmetric payoffs`. In the



(a) First random training/validation/test set split. (b) Second random training/validation/test set split.

Figure 3: Prediction performance for Poisson-QCH extended by features, functional form, and feature transformations selected by Bayesian optimization. We show models that were at any point “incumbent” (i.e., the best found by SMAC at some point in time), plus `linear4`, `linear8`, and the `linear7` model of Section 4.4, which is `linear8` with the `fairness[R]` feature omitted. Each incumbent is shown only in the division in which SMAC evaluated it. The best-performing incumbent in division 1 is labeled `smac1`, and the best-performing incumbent in division 2 is labeled `smac2`.

second division, SMAC’s best-performing incumbent (henceforth `smac2`) was an 8-parameter model that linearly combined the same features, except that `maxmax welfare` was replaced by `maxmax fairness`, and `fairness[R]` was not included.

Interestingly, the best-performing incumbent model for every number of parameters combined its features linearly, with no exceptions in the first division, and only two exceptions in the second division. One might have expected the logit specification to perform better, since it has a tunable precision. However, the logit combination tended to produce higher-entropy (i.e., noisier) predictions than the linear combination; this might explain its worse performance. We note that this explanation is consistent with the selection of the informativeness transformation, which also reduces predictions’ entropy.

Many of SMAC’s incumbents contained both real-valued and binary-valued versions of the same features. We hypothesized that `linear4` augmented by real-valued versions of its four binary features only (i.e., excluding the `welfare` and `max-regret` features) would perform better than `linear4`. This model, which we refer to as `linear8`, was not checked by SMAC, and so we checked it manually. As shown in Figure 3, `linear8` performed better than any other model SMAC checked—including `smac1`—in the first division, and just as well as `smac2` in the second division. For the remainder of the paper, we thus focus our attention on the `linear4` and `linear8` models.

4.3 Parameter Analysis

We now examine and interpret some of the parameters of the Poisson-QCH model combined with the `linear4` and `linear8` level-0 specifications. As described in Section 2.3, we take

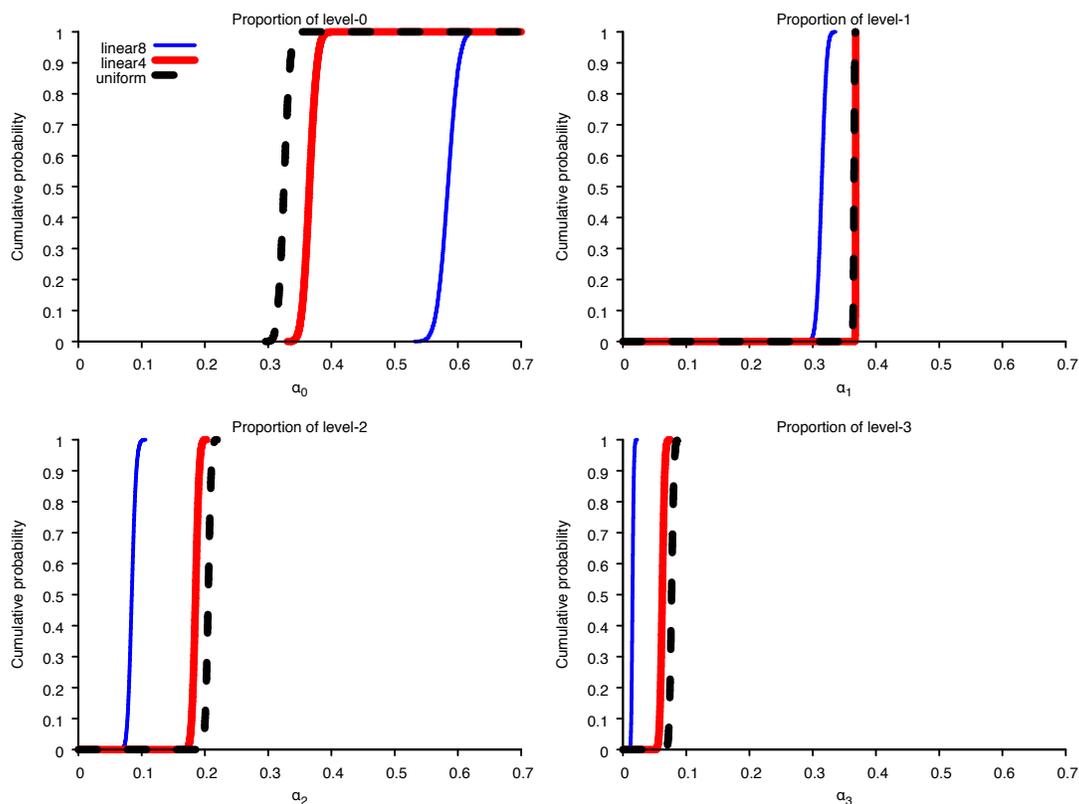


Figure 4: Marginal cumulative posterior distributions of levels of reasoning in the ALL10 dataset, for Poisson-QCH with `linear8`, `linear4`, and `uniform` specifications.

the Bayesian approach of analyzing the posterior distribution of the parameters given the data. The variation in these plots is over the parameters themselves, rather than over the data.⁹ We present the posterior distributions as cumulative marginal distributions; i.e., for every parameter, we plot the cumulative density function (CDF)—the probability that the parameter should be set less than or equal to a given value—averaging over values of all other parameters. Plotting cumulative density functions allows us to visualize an entire continuous distribution without having to estimate density from discrete samples, thus sparing us manual decisions such as the width of bins for a histogram. Plotting marginal distributions allows us to examine intuitive two-dimensional plots about multi-dimensional distributions. Interaction effects between parameters are thus obscured. We revisit the issue of possible interaction effects at the end of this subsection.

Figure 4 shows the marginal posterior distribution for the proportions of each level in the population (up to level 3), for each of the `linear4`, `linear8`, and `uniform` specifications. The possible proportions are given on the x axis; the probabilities of the proportion having that value or less (i.e., the cumulative probabilities) are given on the y axis. Note that the

9. That is, we hold the data fixed and report the posterior probability of different parameter values.

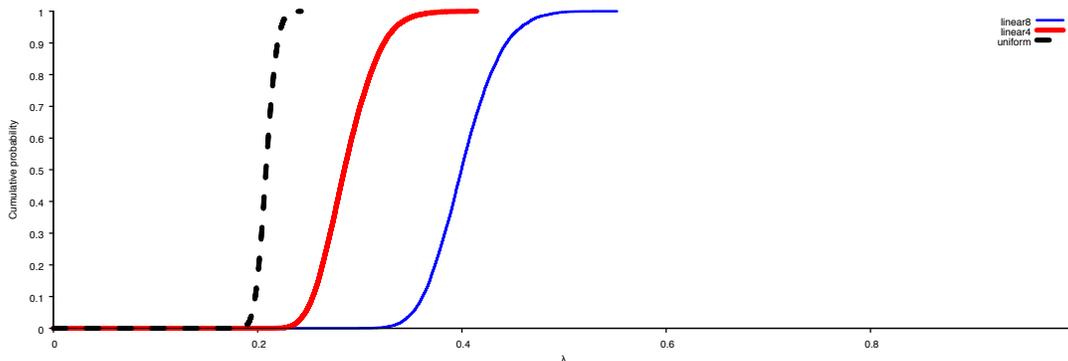


Figure 5: Marginal cumulative posterior distributions of precision (λ) in the ALL10 dataset, for Poisson-QCH with `linear8`, `linear4`, and `uniform` specifications.

distributions of proportions are computed from the distribution of the τ parameter: i.e., each value of τ implies a specific proportion for each level.

Generally, we observe that richer level-0 specifications gave rise to higher proportions of level-0 agents. This was especially striking under the `linear8` specification, which gave rise to a median estimate of over half the agents (58%) being level 0. For the `linear4` and `uniform` specifications the corresponding medians were 0.37 and 0.32 respectively. The `linear8` and `linear4` specifications therefore also led to predictions of fewer higher-level agents than the `uniform` specification.

Figure 5 shows the posterior distributions of the λ parameter for each of the `linear4`, `linear8`, and `uniform` specifications. All three specifications had similar estimates for this parameter. The posterior distributions of λ for the `linear4` and `linear8` specifications had overlapping 95% credible intervals, and hence were not significantly different. The 95% credible intervals for the posteriors of both the `linear4` and `linear8` specifications were disjoint from and larger than that of the `uniform` specification. Quantal response serves two purposes: it accounts for errors of reasoning that people actually make, and it provides an error structure for the model itself. Thus, higher precision estimates for the richer specifications may simply reflect their improved accuracy.

Figures 6 and 7 show the marginal posterior distributions for the weights of the `linear4` and `linear8` models respectively on the ALL10 dataset. As with the distribution over levels, the posterior distributions on nearly all of the weight parameters had modes with very narrow supports, indicating that the data argued consistently for specific ranges of parameter values. The `fairness[R]` feature of the `linear8` specification was the exception to this rule, with an flat distribution over values that indicates a lack of identification. In Section 4.4, we will analyze a version of the `linear8` model that omits this feature (called `linear7`), and find that it performs just as well as `linear8`.

First, let us consider binary features. These had qualitatively similar weights in both the `linear4` and `linear8` specifications: the `fairness` feature had by far the highest median posterior weight, the `maxmax` feature had the second-highest weight, and the `max-symmetric` and `maxmin` features both had small and essentially identical weights, with very overlap-

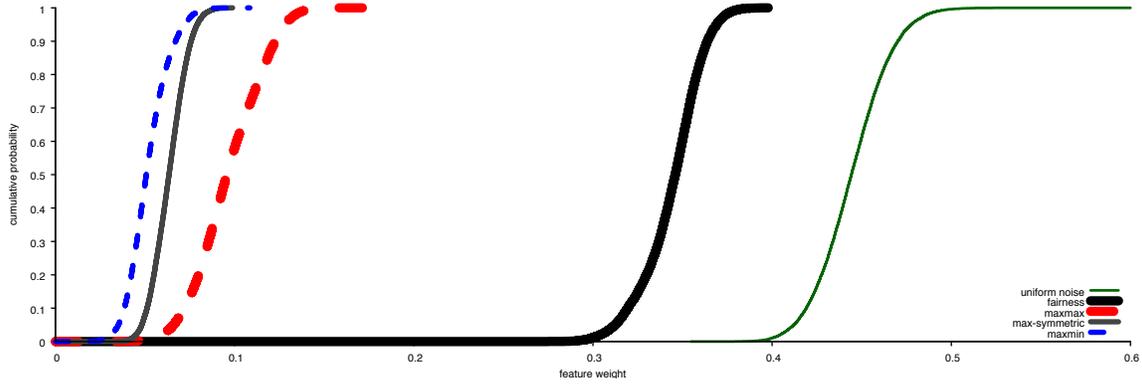


Figure 6: Marginal cumulative posterior distributions over weight parameters of the `linear4` specification, for Poisson-QCH on the ALL10 dataset. The uniform noise weight is defined implicitly by the other four weights as $w_0 = 1 - \sum_{f \in F} w_f$.

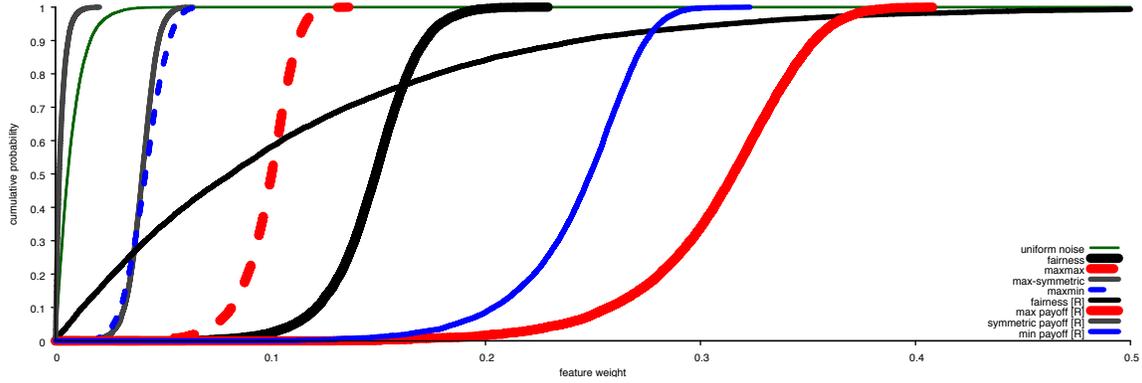
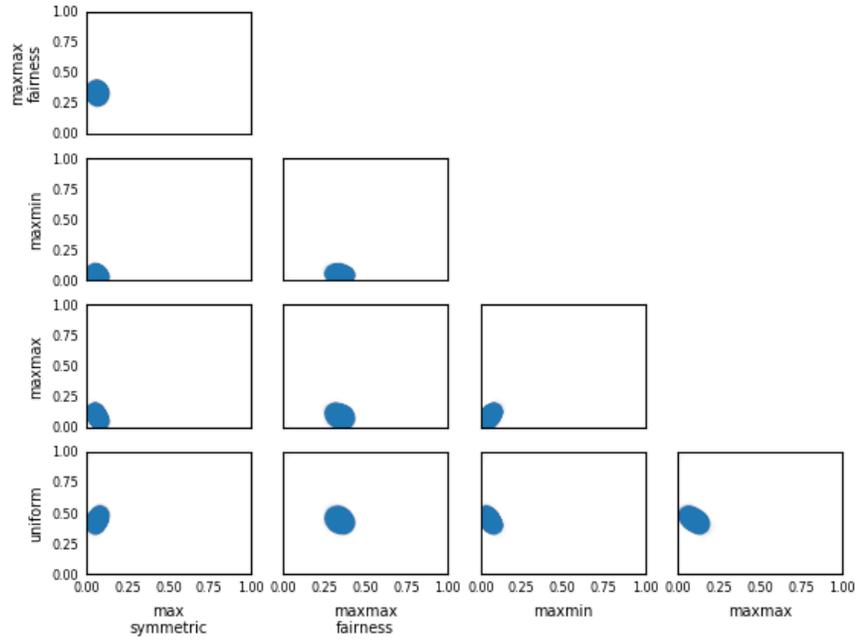


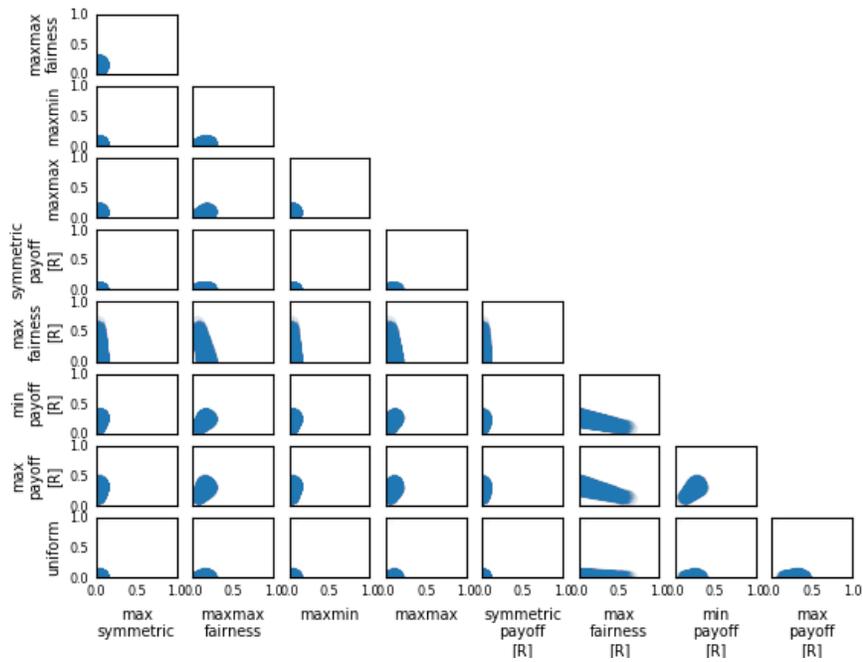
Figure 7: Marginal cumulative posterior distributions over weight parameters of the `linear8` specification, for Poisson-QCH on the ALL10 dataset. The uniform noise weight is defined implicitly by the other eight weights as $w_0 = 1 - \sum_{f \in F} w_f$.

ping posterior distributions. Interestingly, even though the fairness feature was the highest weighted, it was not selected first by the exhaustive evaluation procedure (max symmetric was selected first). This likely indicates that fairness is more predictive than other features when it is present, but that it is predictive in fewer games than max symmetric.

Now let us consider real-valued features. The max-payoff[R] and min-payoff[R] features had similar posterior weights in the `linear8` specification, with overlapping posterior supports. These were the highest-weighted features in the `linear8` specification. The fairness[R] feature was not well identified. This could indicate that it is not an important feature; alternatively, it could indicate interaction effects. We present additional evidence in Section 4.4 that the fairness[R] feature is simply not an important feature. The symmetric-payoff[R] feature was well identified and had a very small weight; evidently, the action with the highest symmetric payoff is somewhat salient, but the actual value of the payoff is not.



(a) linear4



(b) linear8

Figure 8: Two-dimensional projections of posterior distributions over weight distributions for the `linear4` and `linear8` models, for Poisson-QCH on the ALL10 dataset.

The weight allocated to uniform randomization between the `linear8` and `linear4` specifications was very different; the `linear4` specification allocated nearly half of its weight to uniform randomization, whereas for the `linear8` specification uniform randomization played almost no part. This, combined with the strong similarity in the weighting of binary features between the two specifications, suggests that the real-valued features (especially the max and min payoff features) played a genuine role in reducing uncertainty.

Figure 8 shows scatter plots of samples from the posterior distribution of the weights of the `linear4` and `linear8` models on the ALL10 dataset, marginalizing out all but each *pair* of features. This lets us check for interactions between pairs of weights. The plots for `linear4` give little indication of interaction effects. The modes of the projected distributions are roughly circular ellipses, rather than the long ridges that would arise given a strong interaction effect. The ellipses are slightly diagonal, possibly indicating a mild correlation introduced by the normalized activation feature transformation.

In the `linear8` model, the maxmin, maxmax, and symmetric payoff features appear to interact with the maxmax fairness feature. The max payoff[R] and min payoff[R] features also appear to have a mild interaction. This may simply reflect that the games in our dataset have slightly different payoff ranges depending on the underlying study, such that the maximum payoff is mildly predictive of the minimum payoff, and vice versa. The scatter plots for the `linear8` model that involve the fairness[R] feature extend either horizontally or vertically across the plot, consistent with the poor identification of the fairness[R] feature that is apparent from Figure 7.

4.4 Extended Model Performance

We now investigate the extent to which our results generalize beyond Poisson-QCH models, comparing to Poisson Cognitive Hierarchy (Camerer et al., 2004) and Level- k (Costa-Gomes et al., 2001) models. We revisited the specifications of level-0 behavior discussed so far: uniform randomization, `linear4` from Section 4.1, and `linear8` from Section 4.2. In light of the lack of identification of the fairness[R] feature discussed in the previous section, we also considered a version of `linear8` that omitted fairness[R]; we call this specification `linear7`. The results are displayed in Figure 9. Overall, the `linear4` specification yielded a large performance improvement, both on Poisson-QCH and on the two other iterative models. In fact, the other two models benefited disproportionately from the improved level-0 specifications. This is especially interesting given that the level-0 models were selected based solely on the degree to which they improved Poisson-QCH’s performance. Poisson-QCH performed better than the other two models under all level-0 specifications, but the three models had much more similar (and improved) performance under the `linear4` and `linear7/linear8` specifications than under the uniform specification.

The `linear7` and `linear8` specifications both yielded a performance improvement over `linear4`, but their performance did not differ significantly from each other. This provides further evidence that the fairness[R] feature is not predictive.

5. Related Work

Almost every study that employs iterative reasoning models based on either level- k or cognitive hierarchy architectures assumes a uniform distribution of play for level-0 agents.

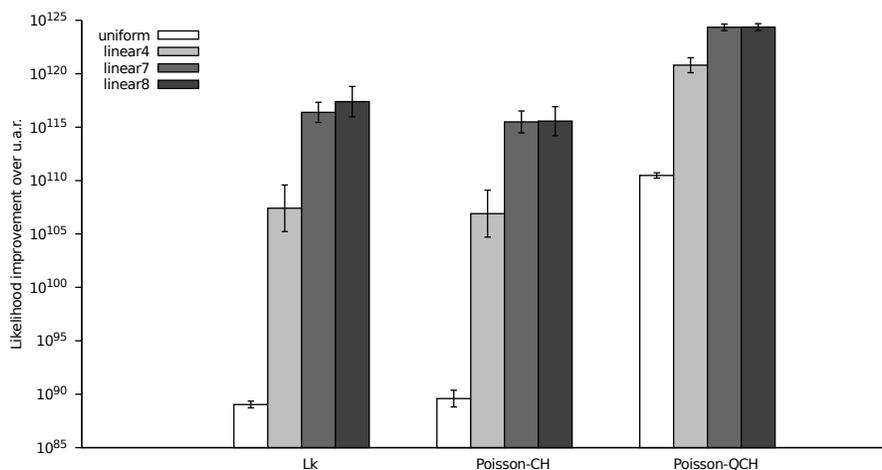


Figure 9: Average likelihood ratios of model predictions to random predictions, with 95% confidence intervals. Results are shown for three different iterative models (Poisson Cognitive Hierarchy (Camerer et al., 2004), Level- k (Costa-Gomes et al., 2001), and Poisson Quantal Cognitive Hierarchy (Wright & Leyton-Brown, 2017)) using four different level-0 specifications (uniform randomization, `linear4` from Section 4.1, `linear8` from Section 4.2, and `linear7`, which is `linear8` with the unidentified fairness[R] feature omitted).

However, there are a few exceptions. Crawford and Iriberri (2007a) manually designated certain actions as “salient” (based on visual features such as “leftmost”) in a hide-and-seek game. They then estimated an iterative model with a level-0 specification in which level-0 agents play salient actions, with the strengths of each action’s salience estimated under the assumption that no agent truly plays a level-0 distribution. Crawford and Iriberri (2007b) specified truth telling as the single salient action in first-price auctions. Arad and Rubinstein (2009) specified a single action of reinforcing all battlefields equally in a Colonel Blotto game as the sole level-0 action. Arad and Rubinstein (2012) specified the highest possible request as the level-0 action in a money-request game where players receive the amount of money they request, but also receive a relatively large bonus for requesting exactly 1 shekel less than the other player. Arad (2012) manually specified two “anchor” strategies for a Colonel Blotto-like game in which players simultaneously assign four representatives to four separate contests in order of the representatives’ ability.

In spite of the crucial dependence of iterative models upon the specification of the level-0 distribution, few studies have empirically investigated level-0 play. Agranov, Caplin, and Tergiman (2010) incentivized subjects to choose an action quickly (and then to revise it after thinking) by imposing a randomized time limit when playing the beauty-contest game of Nagel (1995). They hypothesized that early actions represent level-0 choices and that later actions represent higher-level choices. Based on this assumption, they found that level-0 behavior did not differ significantly from a uniform distribution. In contrast, Burchardi and Penczynski (2014) incentivized players to reveal their reasoning by allowing

a one-time simultaneous exchange of messages between teammates playing a beauty-contest game. Teams of two simultaneously sent each other a single message containing arguments in favor of a given action, and then simultaneously chose an action, with the team’s action being chosen randomly from the two choices. Burchardi and Penczynski then classified each argument according to level of reasoning, and extracted the level-0 behavior from both level-0 and higher-level arguments. They found that level-0 play was significantly different from uniform. Intriguingly, they also found that the level-0 behavior hypothesized by higher-level players was very similar to the level-0 behavior that was actually proposed.

Hargreaves Heap, Rojo Arjona, and Sugden (2014) evaluated the transferability of level-0 specifications between three games in which all of the actions are strategically equivalent: a coordination game, a discoordination game, and a hide-and-seek game. They argue that any level-0 specification based only on the strategic structure of the game must produce an identical level-0 behavior for each type of game, since in each game each action is strategically equivalent to every other action. Based on experimental data, they reject a joint hypothesis that includes an identical distribution of levels for each game and an identical level-0 action in each game.¹⁰ This may indicate that framing effects, in addition to strategic considerations, play a role in determining level-0 behavior. It may also indicate that the population distribution of levels varies between games; we are studying this latter possibility in ongoing work.

The logit level-0 specification of Section 3.3 is widely used for modeling choice behavior based on features of the possible actions (e.g., Train, 2009). Nguyen, Yang, Azaria, Kraus, and Tambe (2013) achieved good performance in the security domain with the SUQR model, which models choice behavior as a logit combination of features such as the coverage of a target and the penalty/reward arising from attacking a specific target.

6. Conclusions

This paper introduced two specifications of level-0 behavior that dramatically improve the performance of the level- k , cognitive hierarchy, and quantal cognitive hierarchy models. These specifications depend only upon the payoffs of the game, and are thus generally applicable to any domain, even ones in which human intuition gives little guidance about the level-0 specification. A linear weighting of four nonstrategic binary features—maxmax payoff, maxmin payoff, maxmax fairness, and max symmetric—improved all three models’ performances, with the weaker models (level- k and cognitive hierarchy) improving the most. We named this specification `linear4`. Including either or both of the remaining two binary features caused models to overfit and prediction performance to decline. Fairness was the feature with by far the greatest weight. Including real-valued versions of the binary features further improved prediction performance for all three models. We called this augmented specification `linear8`. The weight of the `fairness[R]` feature was not well identified in this model, and omitting it (to produce the model we call `linear7`) did not significantly decrease prediction performance.

Conventional wisdom in the economics literature is that level-0 agents exist only in the minds of higher level agents; that is, that a level-0 specification acts solely as a starting

10. They initially assume that no level-0 agents exist as part of their joint hypothesis. However, their results are robust to the existence of level-0 agents.

point for higher level agents' reasoning (e.g., Crawford & Iriberri, 2007b). Our results argue against this point of view: both the uniform and `linear4` specifications estimated that more than a third of agents were level 0, and the `linear8` specification estimated that over half the population was level 0. These results are strong evidence that nonstrategic behavior is an important aspect of human behavior, even in strategic settings. Further refining our understanding of nonstrategic behavior is an important direction for future work, both in terms of capturing factors that are considered by nonstrategic agents and in terms of incorporating these factors into predictive behavioral models.

Acknowledgements

An early version of this work appeared as a conference publication (Wright & Leyton-Brown, 2014). This work expands upon the early version in several ways. The Bayesian optimization of Section 4.2 is entirely new, as are the real-valued versions of the features in Section 3.2 and all discussion of the `linear7` and `linear8` models. In addition, we performed an entirely new analysis on an expanded dataset, switching to a stricter cross-validation methodology in which all testing was performed on held-out *games* rather than held-out *observations*.

This work was funded in part by a Canada Graduate Scholarship from the Natural Sciences and Engineering Research Council of Canada, a Four Year Fellowship from the University of British Columbia, an NSERC E.W.R. Steacie Fellowship, and an NSERC Discovery Grant. Part of this work was done while the first author was a postdoctoral researcher at Microsoft Research New York.

References

- Agranov, M., Caplin, A., & Tergiman, C. (2010). The process of choice in guessing games. Social science working paper 1334r, California Institute of Technology. Available at <http://www.wordsmatter.caltech.edu/SSPapers/sswp1334R.pdf>, accessed Feb. 9, 2014.
- Arad, A. (2012). The tennis coach problem: A game-theoretic and experimental study. *The BE Journal of Theoretical Economics*, 12(1).
- Arad, A., & Rubinstein, A. (2009). Colonel Blotto's top secret files. Working paper. Available at <http://www.dklevine.com/archive/refs4814577000000000432.pdf>, accessed Feb. 9, 2014.
- Arad, A., & Rubinstein, A. (2012). The 11–20 money request game: A level- k reasoning study. *The American Economic Review*, 102(7), 3561–3573.
- Burchardi, K. B., & Penczynski, S. P. (2014). Out of your mind: Eliciting individual reasoning in one shot games. *Games and Economic Behavior*, 84, 39–57.
- Camerer, C., Ho, T., & Chong, J. (2004). A cognitive hierarchy model of games. *Quarterly Journal of Economics*, 119(3), 861–898.
- Camerer, C., & Thaler, R. H. (1995). Anomalies: Ultimatums, dictators and manners. *The Journal of Economic Perspectives*, 9(2), 209–219.

- Camerer, C. F. (2003). *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press.
- Cooper, D., & Van Huyck, J. (2003). Evidence on the equivalence of the strategic and extensive form representation of games. *Journal of Economic Theory*, *110*(2), 290–308.
- Costa-Gomes, M., Crawford, V., & Broseta, B. (2001). Cognition and behavior in normal-form games: An experimental study. *Econometrica*, *69*(5), 1193–1235.
- Costa-Gomes, M., Crawford, V., & Broseta, B. (1998). Cognition and behavior in normal-form games: an experimental study. Discussion paper 98-22, University of California, San Diego.
- Costa-Gomes, M. A., & Weizsäcker, G. (2008). Stated beliefs and play in normal-form games. *The Review of Economic Studies*, *75*(3), 729–762.
- Crawford, V., & Iriberri, N. (2007a). Fatal attraction: Salience, naivete, and sophistication in experimental “hide-and-seek” games. *American Economic Review*, *97*(5), 1731–1750.
- Crawford, V., & Iriberri, N. (2007b). Level- k auctions: Can a nonequilibrium model of strategic thinking explain the winner’s curse and overbidding in private-value auctions?. *Econometrica*, *75*(6), 1721–1770.
- Gal, Y., & Pfeffer, A. (2007). Modeling reciprocal behavior in human bilateral negotiation. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, pp. 815–820. AAAI Press.
- Goeree, J. K., & Holt, C. A. (2001). Ten little treasures of game theory and ten intuitive contradictions. *American Economic Review*, *91*(5), 1402–1422.
- Hansen, N., & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, *9*(2), 159–195.
- Hargreaves Heap, S., Rojo Arjona, D., & Sugden, R. (2014). How portable is level-0 behavior? a test of level- k theory in games with non-neutral frames. *Econometrica*, *82*(3), 1133–1151.
- Haruvy, E., & Stahl, D. (2007). Equilibrium selection and bounded rationality in symmetric normal-form games. *Journal of Economic Behavior and Organization*, *62*(1), 98–119.
- Haruvy, E., Stahl, D., & Wilson, P. (2001). Modeling and testing for heterogeneity in observed strategic behavior. *Review of Economics and Statistics*, *83*(1), 146–157.
- Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2010). Sequential model-based optimization for general algorithm configuration (extended version). Tech. rep. TR-2010-10, University of British Columbia, Department of Computer Science. Available online: <http://www.cs.ubc.ca/~hutter/papers/10-TR-SMAC.pdf>.
- Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In *Proc. of LION-5*, p. 507–523.
- Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2012). Parallel algorithm configuration. In *Proc. of LION-6*, pp. 55–70.

- Hutter, F., Hoos, H., & Leyton-Brown, K. (2013). An evaluation of sequential model-based optimization for expensive blackbox functions. In *Proceedings of the ACM Conference on Genetic and Evolutionary Computation (GECCO)*, pp. 1209–1216. ACM.
- Jain, M., Tsai, J., Pita, J., Kiekintveld, C., Rathi, S., Tambe, M., & Ordóñez, F. (2010). Software assistants for randomized patrol planning for the LAX airport police and the federal air marshal service. *Interfaces*, 40(4), 267–290.
- Kagel, J. H., & Roth, A. E. (2016). *The Handbook of Experimental Economics, Volume 2*. Princeton University Press.
- Murphy, K. (2012). *Machine learning: a probabilistic approach*.
- Nagel, R. (1995). Unraveling in guessing games: An experimental study. *American Economic Review*, 85(5), 1313–1326.
- Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2), 125–139.
- Nguyen, T. H., Yang, R., Azaria, A., Kraus, S., & Tambe, M. (2013). Analyzing the effectiveness of adversary modeling in security games.. In *AAAI*.
- Patil, A., Huard, D., & Fonnesebeck, C. (2010). PyMC: Bayesian stochastic modelling in python. *Journal of Statistical Software*, 35(1).
- Pita, J., Jain, M., Marecki, J., Ordóñez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., & Kraus, S. (2008). Deployed armor protection: the application of a game theoretic model for security at the los angeles international airport. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial track*, pp. 125–132. International Foundation for Autonomous Agents and Multiagent Systems.
- Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods*. Springer Verlag.
- Rogers, B. W., Palfrey, T. R., & Camerer, C. F. (2009). Heterogeneous quantal response equilibrium and cognitive hierarchies. *Journal of Economic Theory*, 144(4), 1440–1467.
- Savage, L. (1951). The Theory of Statistical Decision. *Journal of the American Statistical Association*, 46(253), 55–67.
- Stahl, D., & Wilson, P. (1994). Experimental evidence on players’ models of other players. *Journal of Economic Behavior and Organization*, 25(3), 309–327.
- Stahl, D., & Haruvy, E. (2008). Level- n bounded rationality and dominated strategies in normal-form games. *Journal of Economic Behavior and Organization*, 66(2), 226–232.
- Stahl, D., & Wilson, P. (1995). On players’ models of other players: Theory and experimental evidence. *Games and Economic Behavior*, 10(1), 218–254.
- Thaler, R. H. (1988). Anomalies: The ultimatum game. *The Journal of Economic Perspectives*, 2(4), 195–206.
- Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013). Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 847–855. ACM.

- Train, K. (2009). *Discrete Choice Methods with Simulation*. Cambridge University Press.
- Witten, I. H., & Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.
- Wolpert, D. H., & Bono, J. W. (2013). Predicting behavior in unstructured bargaining with a probability distribution. *Journal of Artificial Intelligence Research*, 46, 579–605.
- Wright, J. R., & Leyton-Brown, K. (2012). Behavioral game-theoretic models: A Bayesian framework for parameter analysis. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, Vol. 2, pp. 921–928.
- Wright, J. R., & Leyton-Brown, K. (2014). Level-0 meta-models for predicting human behavior in games. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation (EC'14)*, pp. 857–874.
- Wright, J. R., & Leyton-Brown, K. (2017). Predicting human behavior in unrepeated, simultaneous-move games. *Games and Economic Behavior*, 106, 16–37.
- Wright, J. R., & Leyton-Brown, K. (2018). A formalization of the boundary between strategic and nonstrategic reasoning. Arxiv preprint arXiv:1812.11571.
- Yang, R., Kiekintveld, C., Ordóñez, F., Tambe, M., & John, R. (2013). Improving resource allocation strategies against human adversaries in security games: An extended study. *Artificial Intelligence*, 195, 440–469.
- Yin, Z., Jiang, A. X., Tambe, M., Kiekintveld, C., Leyton-Brown, K., Sandholm, T., & Sullivan, J. P. (2012). TRUSTS: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine*, 33(4), 59.