UTMD Working Paper

The University of Tokyo
Market Design Center

UTMD-033

Using Big Data and Machine Learning to Uncover How Players Choose Mixed Strategies

Toshihiko Hirasawa
UCLA

Michihiro Kandori
University of Tokyo

Akira Matsushita
University of Tokyo

September 26, 2022

Using Big Data and Machine Learning to Uncover How Players Choose Mixed Strategies[*]

Toshihiko Hirasawa
UCLA

Michihiro Kandori
University of Tokyo

Akira Matsushita
University of Tokyo

September 26, 2022

Abstract

How do humans behave in a situation where (i) one needs to make one's own behavior unpredictable and (ii) one needs to predict an opponent's behavior? This is an important class of strategic situations, formulated as games with a mixed strategy equilibrium. If humans are put in such a situation, it is obvious that, rather than calculating the mixed equilibrium strategy, they use their hunches and some heuristics to achieve the aforementioned goals (i) and (ii). Exactly what kind of mechanisms are employed has not been fully understood. To address this issue, we use our unique big experimental data set about a game with a mixed strategy equilibrium, which has about 75,000 observations, and compare conventional behavioral economics models with some leading machine learning models. The use of big data enables us to examine the *external validities* of those models, i.e., compare the predictive powers of those models in data sets that are *not* used for parameter estimation. We found that machine learning models, most notably a version of the deep learning model LSTM, substantially outperform the leading behavioral model (EWA), and this happens only when the size of the data set for parameter estimation is sufficiently large. Finally, we try to improve the EWA model by incorporating the insights gained from the machine learning models.

[*] Acknowledgments: We would like to thank Hisashi Kashima for his helpful comments and discussion. The first author acknowledges financial support from the Yoshida Scholarship Foundation, Japan. The second author acknowledges financial support under JSPS KAKENHI Grants JP20H00609, JP21K01399, and JP21H04979.

1 Introduction

The central research question of this article concerns how humans behave in a situation where (i) one needs to make one's own action unpredictable and (ii) at the same time one needs to predict which action an opponent will take. Examples abound and include tax auditing, terrorist attacks vs. airport security guards (Tambe, 2011), tennis serves (Walker and Wooders, 2001), and penalty kicks in soccer games (Palacios-Huerta, 2003; Chiappori, Levitt and Groseclose, 2002).

Such a situation can be formalized as a game with a mixed strategy equilibrium. A mixed strategy equilibrium is an ideal state where players' random actions constitute the mutual best reply. Previous research (e.g., the aforementioned papers as well as O'Neill, 1987; Camerer, 2011) has revealed that the concept of a mixed strategy equilibrium describes human behavior in field and lab data to some reasonable extent, while it has also been shown that humans do not exactly follow a mixed strategy equilibrium (e.g., Brown and Rosenthal, 1990). The latter point makes good sense; we would like the reader to reflect for a moment on how one might choose the directions of tennis serves or actions in the Rock-Paper-Scissors game. It is rather clear that one uses intuition, hunches, and some kind of limited reasoning to make one's action unpredictable and at the same time to predict the opponent's action. Exactly what kind of cognitive processes are employed has not yet been fully uncovered. The purpose of our research is to address this issue using machine learning models and big data.

We use a unique data set that one of the authors has collected about a two-player game that has a non-trivial mixed strategy equilibrium. It is a card game invented by O'Neill (1987), where each player, the "red" and the "black" player, has four cards, K, 1, 2, and 3, and chooses one of those cards at the same time as the opponent. The winner is determined by a rather complicated set of rules, and as a result, unlike the Rock-Paper-Scissors game, the equilibrium probability distribution over the four cards is not uniform, and the black player has a higher equilibrium win rate. The data set was collected in a Coursera online course on game theory, and it covers more than 5,000 participants. Each pair played the game 30 times, which provides roughly 75,000 observations (for each player's role). To the best of our knowledge, this is one of the largest data sets for a single treatment in economic laboratory experiments.

To uncover how players behave in a game with a mixed strategy equilibrium, we employ some of the leading machine learning models as well as the conventional behavioral economics models. Machine learning refers to a class of models in artificial intelligence that are used to detect various empirical regularities in big data. It received much public attention when a deep learning model gave a sensational performance in a leading image recognition contest, the ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al., 2015). The

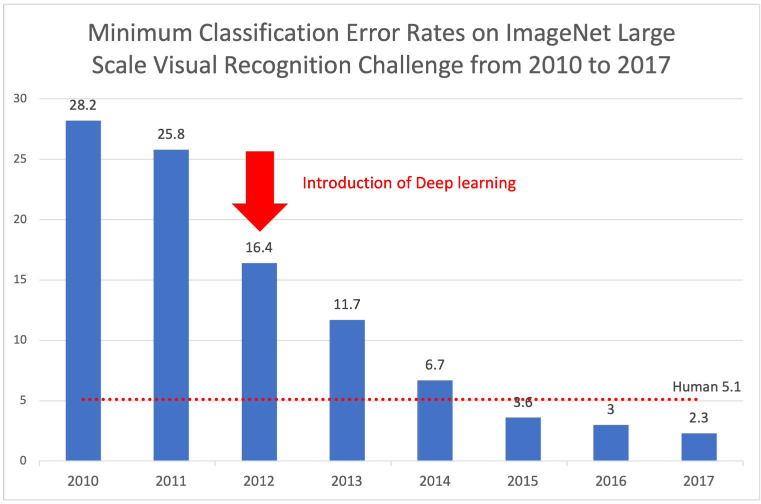
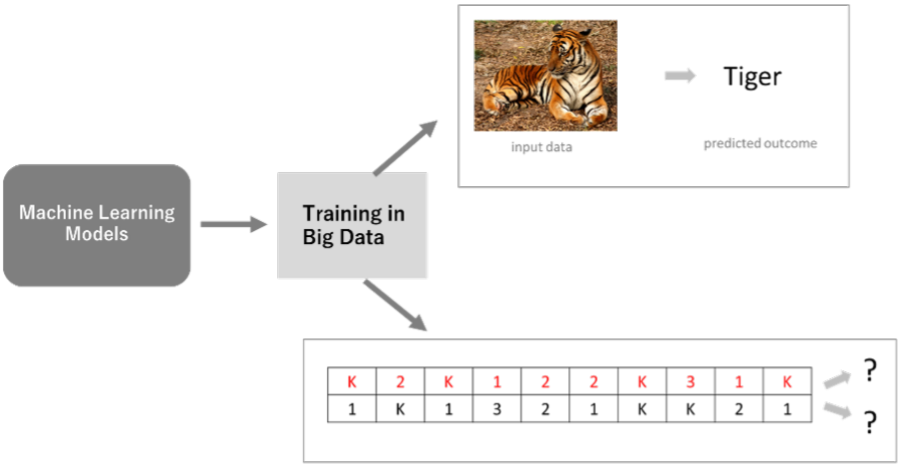


Figure 1. The graph is created based on data on Russakovsky et al. (2015) and ImageNet `https://www.image-net.org/`. Note that for 2012 and 2013, we selected the minimum classification error rates by a team not using outside training data. Also, note that the human error rate is based on test data from 2012 to 2014.

task was to recognize the object (such as a tiger) in an image (such as a picture of a tiger). Figure 1 provides the error rates of the winners of the competition over time. Previously, the best error rate was somewhere above 20%, with gradual improvements made each year, but the introduction of a deep learning model in 2012 resulted in a sudden drop of around 10 percentage points. In the contest, the contestants "trained" their models (i.e., performed parameter fitting) on a large data set of about 1.2 million images, and the actual competitive image recognition was performed on a separate "test" data set.



K	2	K	1	2	2	K	3	1	K
1	K	1	3	2	1	K	K	2	1

Figure 2. The motivation of our research. The image of the tiger is adapted from `https://commons.wikimedia.org/wiki/File:2012_Suedchinesischer_Tiger.JPG` (J. Patrick Fischer, 2011).

Figure 2 illustrates the motivation of our research: If machine learning is capable of recognizing some patterns in an image, such as the overall shape of the object and its local pattern of black stripes, to conclude that the object in the image is a tiger, it might also recognize patterns in the past history of play in the card game to predict how players choose their current actions. We have a unique big data set that enables us to explore that possibility.

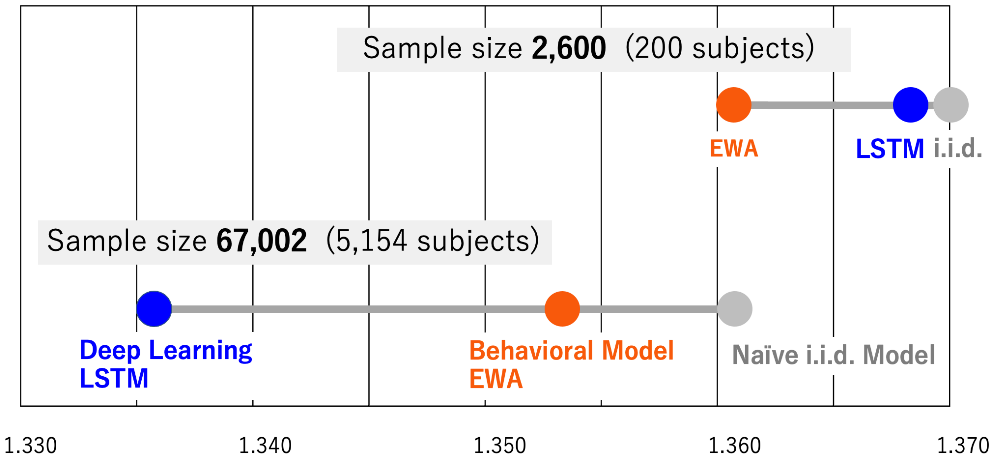


Figure 3. Prediction error rates (Kullback-Leibler divergence) of alternative models (of the red player) and the sample size.

Figure 3 summarizes our main findings. The figure shows the predictive powers of alternative models of the subjects' behavior. We conducted parameter estimation/fitting in 80% of the data (the "training" data) and utilized the remaining 20% of data for the performance comparison of the models (the "test" data) in terms of the prediction error rates. Figure 3 shows the results with the full sample size of our unique big data (below) as well as with an artificially reduced size (above) that is comparable to those in conventional in-person laboratory experimental studies. The full sample results show that the best machine learning model we analyzed, a version of deep learning called LSTM, substantially improved upon the conventional behavioral model (Experience-Weighted Attraction model: EWA). The leading behavioral model (EWA) achieves only 30.9% of the error rate reduction of the best machine learning model relative to a naive benchmark in the form of an i.i.d. mixture model. In contrast, if we only had a smaller sample size comparable to those in conventional in-person laboratory experiments (the upper-right part in Figure 3), the differences in the error rates of alternative models would be much smaller and the notable dominance of the machine learning model would not manifest itself.

We must stress that what Figure 3 shows is *not* the goodness of fit of various models to the data set that is used for parameter estimation. Given that the machine learning model has a much larger number of parameters than the behavioral model, it would not be surprising that the former fits the data better than the latter. Instead, what Figure 3 shows is the "external validity," i.e., the predictive power of the models in a data set that is not used for parameter estimation. The superiority of the machine learning model shows that it captures the regularities that are common in the training and test data, i.e., the regularities of human subjects' behavior. In summary, thanks to our unique big data set, we were able to detect, by means of a machine learning model, that there are certain regularities in the subjects' behavior that have not been captured by the existing models.

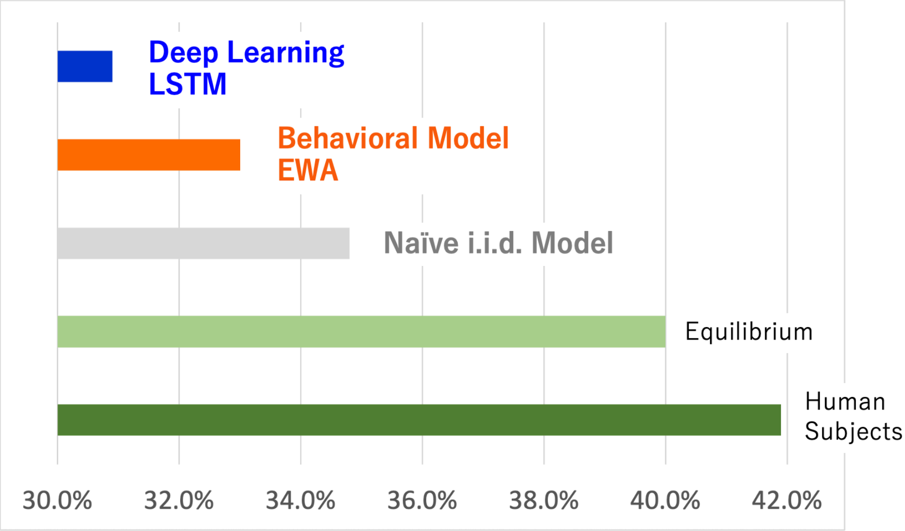


Figure 4. Strategic error rates: how the loss rate of the black player is reduced by utilizing the estimated/fitted models of the opponent.

An important implication is that, even though the subjects try to be unpredictable, their behavior patterns are detectable to some extent by a machine learning model. Figure 4 shows the extent to which the models capture the predictable patterns of the subjects' behavior. The figure shows, for each model, a novel measure of prediction errors we introduce, *strategic error rate*, which measures the average loss rate of a player if she/he utilized the estimated model of the opponent to make the best reply in any given single period. Note that, if a model predicts the action of the focal player for sure, the opponent can surely win and the strategic error rate of the model is zero. While the actual loss rate of the black player is 41.9%, which is close to the equilibrium loss rate, the black player can reduce the loss rate by more than 10 percentage points, down to 30.9% by utilizing the best machine learning model (LSTM) of the opponent.

We also tried to obtain insights into the nature of players' behavior from several machine learning models we employed. This is because machine learning models are often "black boxes" in that the meaning of the fitted parameters is not immediately clear. The best machine learning models in our study successfully replicate how players choose mixed strategies, but what mechanisms they capture remains a crucial open research question. This is in line with the recent trend in artificial intelligence research to open the black box and to gain insights or to provide accountability about what is being done inside the machine, something that is called XAI (explainable artificial intelligence) (c.f. Adadi and Berrada, 2018, for the survey). By examining the machine learning models, we incorporated additional mechanisms into the conventional behavioral model (EWA), and the modified model captures 85.3% and 71.4% of the error rate reduction for the red and black player, respectively, as opposed to the 30.9% captured by EWA.

Finally, our paper suggests the following research program, which can be phrased as *Capture and Decode* and proceeds in three steps.

1. Collect Big Data: Collect a data set that is large enough to clearly distinguish the relative performance of models. Our analysis suggests that a number of observations in the order of tens of thousands would be a reasonable target.

2. Capture: Use machine learning models to see if there are systematic regularities in the data that have previously not been discovered. The regularities are "captured" by a machine learning model, and they are encoded in the fitted parameters of the model. The term "encoded" reflects the fact that the meaning of the parameters of machine learning models is often not immediately clear.

3. Decode: "Open" the black box of the machine learning model to identify and understand how the regularities are generated. This is in line with the XAI research mentioned above. The model and fitted parameters that captured the regularities are to be disclosed in the public domain so that researchers can collaborate on this task.

A previous contribution, Peysakhovich and Naecker (2017), also suggests and follows the steps 2 and 3 in the above procedure for experimental data on decision-making under ambiguity, although they use substantially fewer observations (about 3,000 observations in each treatment) than us.

We have success in the capture stage, but work remains to be done on the decoding stage. How to fully uncover the nature of the regularities captured by our machine learning models remains a challenging open question.

1.1 Related Literature

There is a large body of literature on laboratory experiments of games with a unique mixed strategy equilibrium (c.f. Chapter 3 in Camerer, 2011). O'Neill (1987) is one of the most influential works in the literature. He proposed a new experimental design, which we explain in Section 2, to overcome some difficulties of testing the minimax strategy in the previous literature. His finding is that although deviations from the minimax strategy are observed on an individual level, the overall distributions of play and winning rates are very close to the predicted ones from the minimax strategy. In contrast, Brown and Rosenthal (1990) reexamined O'Neill's result and showed that there is less evidence of a minimax strategy than O'Neill (1987) indicates. In particular, they pointed out that people's behavior depends on their own and opponents' behavior, implying that people do not follow an independently mixed Nash strategy. Many studies appearing after these papers (e.g., Rapoport and Boebel, 1992) have tested a mixed strategy under different situations (for example, with a focus on learning in Mookherjee and Sopher, 1997). One of the authors of this paper examines the replicability of O'Neill's results in Kandori (2018). Overall, the literature indicates a modest deviation from a mixed strategy equilibrium.

Our paper is also related to the literature on learning in games that asks how an equilibrium arises in a game. Leading models include reinforcement learning (e.g., Roth and Erev, 1995; Erev and Roth, 1998) and belief learning such as fictitious play (Brown, 1949). Camerer and Ho (1999) propose an influential generalized model incorporating both reinforcement learning and belief learning, called the experience-weighted attraction (EWA) model. We use the EWA model as a benchmark of economic theories, and thus we explain it in detail in Section 4.3.

Finally, our paper is positioned in the literature on machine learning and economics. In recent years, an increasing number of economics studies have been conducted using machine learning approaches (c.f. Athey and Imbens, 2019; Mullainathan and Spiess, 2017). Behavioral economics and experimental economics are no exception. As Camerer (2019) discusses, behavioral economics and artificial intelligence may interact in future research, and these fields are trying to incorporate the machine learning approach into their analysis. In the literature, our paper is most closely related to Peysakhovich and Naecker (2017) and Fudenberg et al. (2021). Peysakhovich and Naecker (2017) show that machine learning tools can improve out-of-sample predictive power over economic models in the domain of ambiguity. Fudenberg et al. (2021) show that machine learning models outperform the out-of-sample predictive power of economic models such as the Level-K model in the initial play of matrix games. What both papers and ours have in common is that they try to improve the economic theory, focusing on the out-of-sample prediction performance and using a measure to quantify the performance. Peysakhovich and Naecker (2017) use regularized regression (including LASSO, explained in Section 5.3), as a benchmark, and compare the out-of-sample mean squared error from regularized regression with that from economic models. Through the comparison, they consider how much out-of-sample variance is explainable. Fudenberg et al. (2021) defined the notion of completeness to evaluate how much out-of-sample prediction errors are explained by economic theories. Completeness is defined as the fraction of out-of-sample reducible prediction errors that the model could reduce, where the reducible prediction errors are the difference between prediction errors from the best model minus those from a naive benchmark. We use a version of that concept in the performance comparison of our models.

2 O'Neill's Game

In this paper, we consider *O'Neill's game*, introduced by O'Neill (1987). Our game is a normal form game with two players, a red (R) player and a black (B) player. Each player $i \in I \equiv \{\text{R, B}\}$ chooses one of four cards, $a_i \in C \equiv \{1, 2, 3, \text{K}\}$ (Ace, Two, Three, and King), as their action. The payoff matrix is shown in Table 1. In words, the red player wins if and only if

1. both players choose K, or
2. both players choose numbers (1, 2, or 3), and those two numbers are different,

and the black player wins if and only if

1. exactly one player chooses K, or
2. both players choose the same number.

	The black player				
		1	2	3	K
The red player	1	0, 1	1, 0	1, 0	0, 1
	2	1, 0	0, 1	1, 0	0, 1
	3	1, 0	1, 0	0, 1	0, 1
	K	0, 1	0, 1	0, 1	1, 0

Table 1. The payoff matrix of the stage game.

This game was designed to be the simplest possible one with a non-trivial mixed strategy equilibrium. More precisely, O'Neill (1987) shows that the game is the unique normal form game that satisfies the following conditions.

1. There are binary payoffs for each player.
2. Neither player has two identical strategies.
3. Neither player has a dominant strategy.
4. The game is not completely symmetrical in strategies.
5. Any other game satisfying the conditions above has at least as many strategies for each player.

This is the game played by our subjects, and the labeling of pure strategies described above is slightly different from the one originally used in O'Neill's experiment[*1]. The game has a

[*1] In O'Neill's experiment, the card that played the role of K in the above description of the game was

unique mixed strategy Nash equilibrium: both players play K with a probability of 0.4 and play 1, 2, and 3 with a probability of 0.2, respectively. In the equilibrium, the red player wins with a probability of 0.4, and the black player wins with a probability of 0.6. Hence, unlike the Rock-Paper-Scissors game, the mixed strategy is not trivial. The complexity or non-triviality of the mixed strategy allows us to test and understand how people choose a mixed strategy.

3 Data

3.1 Data Collection

We collected our data in a Coursera online course offered by one of the authors (M. Kandori), "Welcome to Game Theory"[*2]. The course started in February 2015 and is still available. In the first week, students are asked to play this game with someone else for 30 periods and submit the results on the course web page before taking lectures on Nash equilibrium and mixed strategy equilibrium. The rules of the game are explained in one of the lectures, and the students are told, "This is a perfect example of a strategic situation because what is best for you depends on what the other player is going to do, and each player is trying to do his or her best against the opponent. By experiencing this game, you can personally see the nature of strategic thinking." The written instructions say the following: "We encourage everyone to give it a shot, . . . Please note that no grade will be given to this activity. Your participation in this activity should be on a completely voluntary basis, and your answers will have no bearing on your course grades."

Hence, our data set is different from the usual lab data in the following respects. First, the experiments were unsupervised. We asked the students to find a partner, play the game, record the results, and report back. Second, we do not know the identities of the students' partners, who we expect to be their friends or family members. Third, no financial rewards were given to the subjects. We expect that these features do not pose a serious problem for the following reasons. First, we asked the students to play this game with a deck of cards, and we are naturally motivated to win when we play a card game, even if no monetary payment is made. Second, we made it clear that participation is completely voluntary and has nothing to do with the course grade, and participation is costly in terms of time and effort. This fact is likely to discourage attempts to submit fake data, as the cost of doing this exceeds the benefit. A person who submits fake data incurs the time and effort of reading and understanding the instructions, cooking up data for 30 rounds, and uploading the file, even though no reward is given. Our view is that only those who are curious about experiencing a strategic situation are likely to have participated in the experiments and that they will have

Joker.

[*2] https://www.coursera.org/learn/game-theory-introduction

done this for the fun of playing the game.

Our data set here contains 2781 pairs (5562 participants) of play in total, submitted from February 2015 to April 2021. It contains 192 pairs of data in which a player's action was missing in at least one period. We eliminated these pairs and used the remaining 2589 pairs (5178 participants) of data.

3.2 The uniqueness of Our "Big" Data

The uniqueness of our data set lies in the number of participants, and especially the number of observations. As indicated in Section 3.1, we collected data from 2589 pairs (5178 participants). Each pair played the stage game for 30 rounds. Hence, the total numbers of observations for the red player and the black player are $77670 = 2589 \times 30$, respectively[*3]. To the best of our knowledge, the number of observations in our data set is one of the largest for a single treatment.

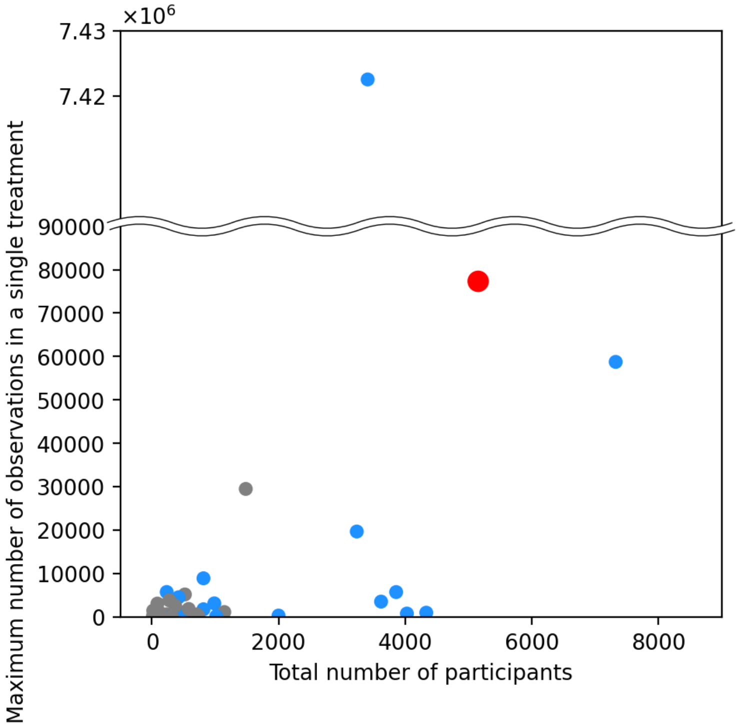


Figure 5. The total number of participants and the maximum number of observations in a single treatment in each paper published in 2020 and 2021. The red point corresponds to our paper. The gray points represent laboratory experiments, and the blue ones correspond to papers employing "artefactual field experiments," defined by Harrison and List (2004), which include experiments conducted by Amazon Mechanical Turk (MTurk).

[*3] In Section 3.3, we eliminate 12 pairs of data. Therefore, the number of observations for each player's role becomes $77310 = 2577 \times 30$.

To substantiate our claim about how big our data set is, we have checked how many participants and observations recent experimental papers obtained. Nunnari, Congiu and Emiliano (2022) create a list of all experimental papers with a lab component published in Top 5 journals[*4] between 2010 and 2021. Checking the papers in the list published in 2020 and 2021, we found that the number of observations in our data is more than those in all those papers except for Augenblick and Rabin (2021), which analyzes artefactual field data[*5]. The results are shown in Figure 5, which plots the total number of observations and the maximum number of observations in a single treatment in each paper. Although there is another paper in Figure 5 (Rees-Jones and Taubinsky, 2020) whose maximum number of observations is close to ours, it is clear that the number of observations in our paper is one of the largest.

3.3 Data Cleaning

Since our data were collected by the participants themselves in an uncontrolled environment, there are concerns about data credibility in some cases, in the sense of whether the submitted data are based on actual play. For example, if the black player in a pair plays $a_B = 2$ for 30 periods and the red player plays $a_R = 1$ for 30 periods, this results in a complete win for the black player. We exclude a fairly small number of such "outliers" through the following procedure. First, we list the pairs

- whose win rate for the red or the black player is in the top 1%, or
- in which either player repeats the same card more than 15 times.

We obtained 53 pairs through this process. We then manually checked them and finally eliminated 12 pairs of obviously suspicious data. In the end, we conducted our analysis using the remaining 2577 pairs (5154 participants) of data.

3.4 Descriptive Statistics

In this section, we briefly describe some features of our data based on summary statistics. Hereafter, we use the cleaned data for all analyses: 2577 pairs × 30 periods.

First, we observe that the choice probabilities of the actual subjects are very close to the

[*4] American Economic Reviews, Econometrics, Quarterly Journal of Economics, Journal of Political Economy, and Review of Economic Studies.

[*5] They collected three data sets, whose number of observations are 593,218, 926,083, and 7,422,530. The first data set consists of probability assessments from thousands of forecasters solicited via email. The second one is not about human subjects and consists of probability assessments from an algorithm that makes dynamic probabilistic predictions about baseball games. The third one consists of the "market average probability assessments" backed up by the high-frequency market data of Betfair, a prediction market in the UK. Thus, the paper concerns more of a field experiment than a lab experiment. This is indicated by the fact that Nunnari, Congiu and Emiliano (2022) classify the paper as a paper employing *artefactual field experiments*, as defined by Harrison and List (2004).

(a) Distribution of the action profiles of the subjects in our data

Red \ Black	**1**	**2**	**3**	**K**	**Marginal Distribution**
1	.059 (.040)	.051 (.040)	.048 (.040)	.081 (.080)	.238 (.200)
2	.051 (.040)	.047 (.040)	.043 (.040)	.073 (.080)	.214 (.200)
3	.045 (.040)	.041 (.040)	.044 (.040)	.070 (.080)	.199 (.200)
K	.080 (.080)	.064 (.080)	.064 (.080)	.140 (.016)	.348 (.400)
Marginal Distribution	.235 (.200)	.202 (.200)	.198 (.200)	.364 (.400)	

(b) Distribution of action profiles of the subjects in O'Neill (1987)

Red \ Black	**1**	**2**	**3**	**K**	**Marginal Distribution**
1	.044 (.040)	.043 (.040)	.043 (.040)	.091 (.080)	.221 (.200)
2	.046 (.040)	.038 (.040)	.038 (.040)	.092 (.080)	.215 (.200)
3	.049 (.040)	.032 (.040)	.037 (.040)	.085 (.080)	.203 (.200)
K	.086 (.080)	.065 (.080)	.051 (.080)	.158 (.016)	.362 (.400)
Marginal Distribution	.226 (.200)	.179 (.200)	.169 (.200)	.426 (.400)	

Table 2. Distribution of the action profiles played by all the subjects (a) in our data and (b) in O'Neill's data (the statistics are summarized in Brown and Rosenthal (1990)). The numbers in parentheses are the distribution of action profiles in the Nash equilibrium. O'Neill collected 2625 observations of strategy profiles (25 pairs times 105 periods).

Nash equilibrium in the aggregate level. Table 2a shows the empirical distribution of action profiles. The numbers in parentheses represent the theoretical distribution of profiles under the Nash equilibrium. One can see that the actual distribution and the NE distribution are very close, except that there is an "Ace bias" for both players. The original result in O'Neill (1987) exhibits similar patterns as shown in Table 2b, and our data largely replicate what O'Neill found.

Table 3 is the win rate distribution for both players. Here, the average win rate (0.419 for red players, 0.581 for black players) is also very close to that in the Nash equilibrium (0.4 for red players, 0.6 for black players).

Figure 6 represents the distribution of the pairwise K ratio of each pair over 30 periods. We see that the points are gathered around the mixed NE or the average point.

However, the time series of the choices of the subjects cast doubt on the hypothesis that

	count	mean	std	min	5%	25%	50%	75%	95%	max
Red	2577	0.419	0.091	0.100	0.267	0.367	0.433	0.467	0.567	0.767
Black	2577	0.581	0.091	0.233	0.433	0.533	0.567	0.633	0.733	0.900

Table 3. Summary statistics of win rates among all pairs in our data.

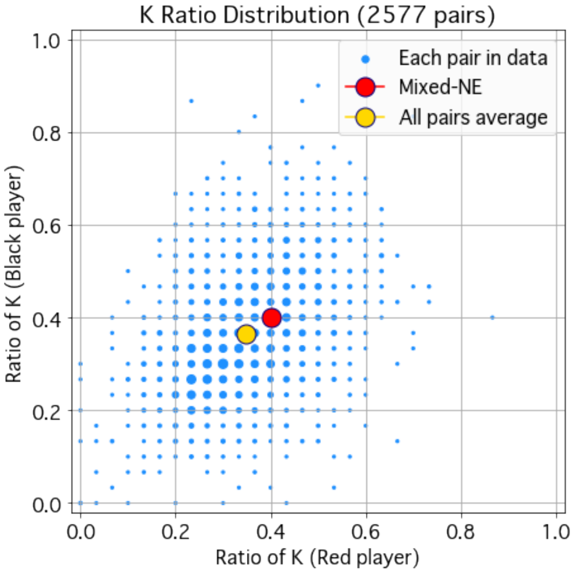


Figure 6. Pairwise distribution of the K ratio. We plot the ratio of K that the red and the black player in each pair play in 30 periods. The size of the markers represents the number of pairs at that point (a larger marker implies more pairs). The red marker indicates the NE (0.4, 0.4) and the yellow marker represents the average of all subjects, (0.348, 0.364).

the subjects play the mixed Nash equilibrium independently every period. The two graphs in the upper half of Figure 7 illustrate the trajectories of the average choice frequencies of all the red and black players in each period. The two graphs in the lower half show the same trajectories in the simulations where the same number of agents play the NE independently. We observe larger variations in the actual data than in the i.i.d. simulations.

Table 4 shows how the card frequencies depend on the previous action of the focal player. The rows in this table should be similar if the subjects played the i.i.d. mixed strategy, but they are rather different. This shows that the choice of the current action depends at least on the previous action, a violation of i.i.d. choice. In fact, Brown and Rosenthal (1990) rigorously showed that the null hypothesis that the subjects in O'Neill's data played the i.i.d. mixture is statistically rejected. What mechanisms guide the subjects' behavior? This is the question we address in this paper.

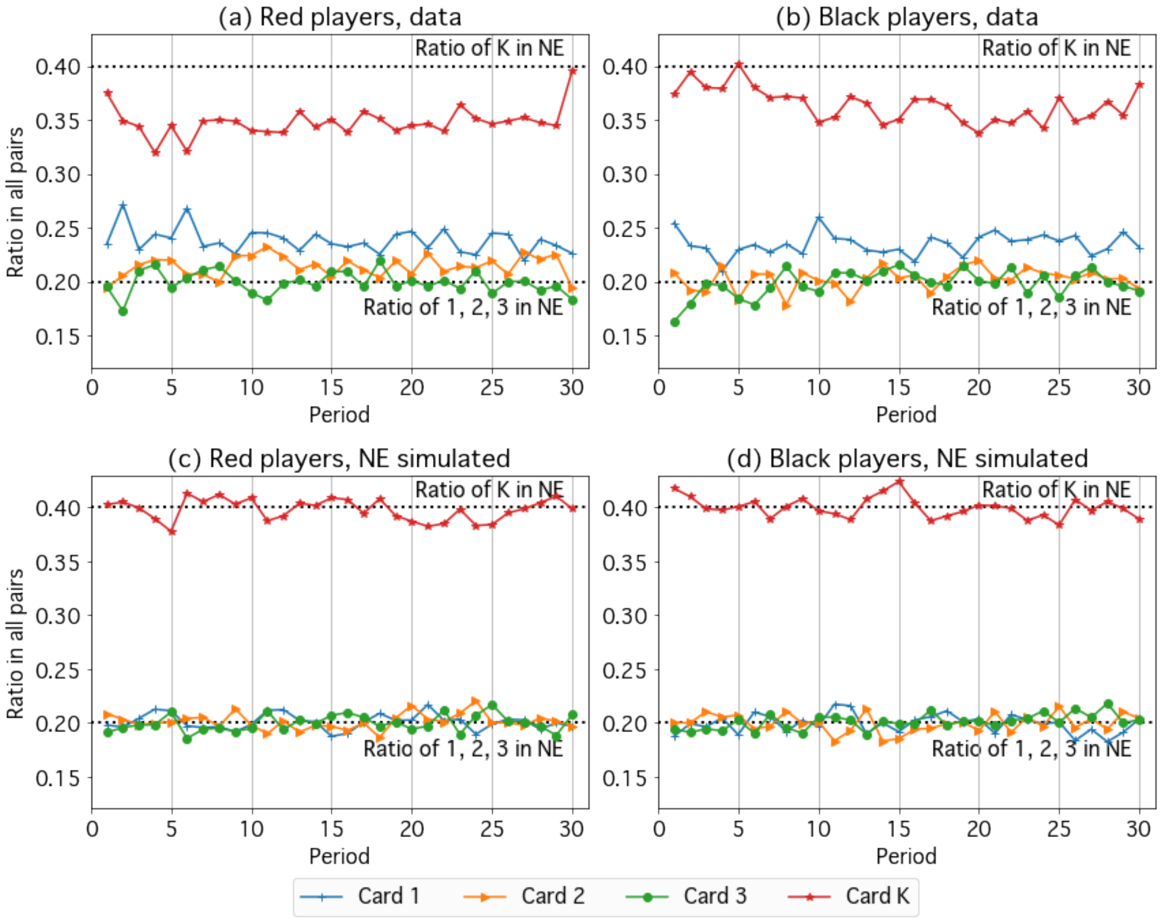


Figure 7. The transition of the average choice probabilities among all 2577 pairs. The upper two figures show the period transition of the actual choice probabilities (left: red players, right: black players). The lower two figures show the transition of the simulated choice probabilities when 2577 agents play the mixed NE independently (left: red players, right: black players).

4 Econometric Models

We adopt two types of models to emulate actual human behavior: traditional economic models and machine learning models. The economic models include major behavioral models such as reinforcement learning (RL), belief learning (BL, also known as fictitious play), and the EWA model. The machine learning models are the decision tree, LASSO, and deep learning models. In this section, we explain the economic models that we adopted. We also discuss the estimated parameters using all the sample data. The final performance comparison with the test data will be discussed in Section 7.

Before discussing the models, we first introduce some notation. In every period $t \in T \equiv \{1, 2, \dots, 30\}$, each player $i \in I = \{R, B\}$ simultaneously chooses one of the four cards,

(a) The red player

		Current action			
		1	2	3	K
Last action	1	0.183	0.242	0.217	0.358
	2	0.264	0.160	0.220	0.356
	3	0.262	0.230	0.148	0.360
	K	0.247	0.221	0.204	0.327
Average		0.238	0.214	0.199	0.348

(b) The black player

		Current action			
		1	2	3	K
Last action	1	0.178	0.223	0.217	0.382
	2	0.253	0.153	0.223	0.371
	3	0.260	0.224	0.149	0.367
	K	0.246	0.204	0.204	0.346
Average		0.235	0.202	0.198	0.364

Table 4. Transition matrix of (a) the red player's actions and (b) the black player's actions. Each line indicates the empirical probability distribution over current action ($a_i^t = 1, 2, 3, K$) given a certain previous actions (a_i^{t-1}).

$a_i^t \in C \equiv \{1,\ 2,\ 3,\ \mathrm{K}\}$. We write the set of action profiles in each stage game as $A \equiv C^2$. We denote the set of histories of action profiles before period t by $H^t \equiv A^{t-1}$ and its elements by $h^t \equiv ((a_{\mathrm{R}}^1, a_{\mathrm{B}}^1), (a_{\mathrm{R}}^2, a_{\mathrm{B}}^2), \ldots, (a_{\mathrm{R}}^{t-1}, a_{\mathrm{B}}^{t-1}))$. For notational convenience, we define the initial dummy history $h^0 \equiv \emptyset$ and let $H^1 \equiv \{\emptyset\}$.

Each pair in our data is indexed by $s = 1, 2, \ldots, S$. For the estimation of the parameters of the economic models we report in this section, we use all the data we have ($S = 2577$). In the performance comparison discussed later in this paper, on the other hand, we randomly split our data into "training data" and "test data," and estimate model parameters using the training data only. This procedure will be explained in detail in Section 5.1.

4.1 Multinomial logit

We first introduce a logistic regression model. Using the dummy variables of the history of action profiles, the probability that a subject in the role of player $i \in I$ chooses action $a \in C$ at period t is given by

$$P(a_i^t = a \mid h^t) = \frac{\exp\left\{\beta_{i,a}^{\mathrm{T}} x_i(h^t)\right\}}{\sum_{c \in C} \exp\left\{\beta_{i,c}^{\mathrm{T}} x_i(h^t)\right\}} \tag{1}$$

where $x_i(h^t) \in \{0,1\}^m$ is an m-dimensional vector of dummy variables (including the constant term) that depends on the history $h^t \in H^t$, where m is the number of dummy variables used in the model. For example, if a model includes the dummies of the history of action profiles in the last two periods, the choice probability is given by

$$P(a_i^t = a \mid h^t) \propto \exp\left\{\sum_{\boldsymbol{c} \in A^2} \beta_a^{\boldsymbol{c}} \cdot 1\left\{((a_{\mathrm{R}}^{t-1}, a_{\mathrm{B}}^{t-1}), (a_{\mathrm{R}}^{t-2}, a_{\mathrm{B}}^{t-2})) = \boldsymbol{c}\right\}\right\}.$$

We start by estimating the following three classes of simple "baseline" models (for the red and the black player). The number m in parentheses indicates the dimension of dummy variables used in a model.

1. Constant only ($m = 1$)
2. History of action profiles in the previous one and two periods ($m = 16$ and 256)
3. History of "king or number" profiles in the previous one, two, three, and four periods ($m = 4$, 16, 64, 256, and 1028)

Here a "king or number" profile (we call it a *K-profile* henceforth) is a summarized action profile in a round that takes one of four values (king, king), (king, number), (number, king), (number, number).

The above models have at most $m = 1028$ variables, and estimating similar models with longer histories of $m > 1028$ variables is infeasible for the following reason. There is, for example, no three-period history of action profiles in the data that takes $((a_{\mathrm{R}}^{t-1}, a_{\mathrm{B}}^{t-1}), (a_{\mathrm{R}}^{t-2}, a_{\mathrm{B}}^{t-2}), (a_{\mathrm{R}}^{t-3}, a_{\mathrm{B}}^{t-3})) = ((1,2),(2,2),(2,2))$. The three-period sample histories lack seven data points out of 4096 patterns, the four-period sample histories lack 30473 data points out of 65536 patterns, and the six-period histories of K-profiles lack 88 data points out of 4096 patterns.

We estimate the coefficients using conditional maximum likelihood estimation. Note that when a model includes dummies of $\underline{t}$-period histories, a maximum of $30 - \underline{t}$ periods of each pair's data can be used for estimation. The log-likelihood function is given by

$$LL_i(\beta_i) = \sum_{s=1,\ldots,S} \sum_{t=\underline{t},\ldots,30} \sum_{a\in C} \beta_{i,a}^{\mathrm{T}} x_i(h_s^t) - \ln\left(\sum_{c\in C} \exp\left\{\beta_{i,c}^{\mathrm{T}} x_i(h_s^t)\right\}\right)$$

for each player $i = \mathrm{R}, \mathrm{B}$, where $\beta_i \equiv (\beta_{i,c})_{c\in C}$ are the coefficient vectors of the model.

In specifications 1, 2, and 3, by construction, only one element in a dummy variable vector $x_i(h^t)$ takes the value 1 at any given time. Let $H^{(k)}$ be the subset of histories for which the k-th element of a dummy variable vector $x_i^{(k)}(h)$ takes the value 1. The maximum likelihood estimator of the coefficients of the dummy variable $\beta_i^{(k)} = \left(\beta_{i,1}^{(k)}, \beta_{i,2}^{(k)}, \beta_{i,3}^{(k)}, \beta_{i,K}^{(k)}\right)$ is known to have the property that the estimated choice probability after the histories $H^{(k)}$ matches the empirical frequencies of the actions after $H^{(k)}$ [*6]. That is, for any $h \in H^{(k)}$,

[*6] To see why, first denote the left-hand side of equation (2) by $p^{(k)}(a)$ for $a = 1, 2, 3, K$. The likelihood function for the model (1) is equal to

$$\prod_k \left(\prod_a p^{(k)}(a)^{n^{(k)}(a)}\right)$$

where $n^{(k)}(a)$ is the number of occurrences of action a in the data after histories in $H^{(k)}$. Therefore, the log-likelihood function is equal to

$$\sum_k N^{(k)} \left(\sum_a \frac{n^{(k)}(a)}{N^{(k)}} \log p(a)\right),$$

$$P(a_i = a \mid h) = \frac{e^{\beta^{(k)}_{i,a}}}{e^{\beta^{(k)}_{i,1}} + e^{\beta^{(k)}_{i,2}} + e^{\beta^{(k)}_{i,3}} + e^{\beta^{(k)}_{i,K}}} \tag{2}$$

$$= (\text{The empirical frequency of card } a \text{ after histories in } H^{(k)}).$$

Note that the four coefficients in (2) are not identified, because if we replace $\beta^{(k)}_{i,a}$ with $\beta^{(k)}_{i,a} + b$ for some constant b, the right-hand side of (2) remains unchanged. Because of this, when we estimate the coefficients, we normalize the coefficients of card K to 0. The validity of that procedure can be seen more directly in equation (4) in the next subsection.

4.2 The Challenge of Model Selection and Non-Parametric Estimation Using Our Big Data Set

Our goal is to find the model that best describes the subjects' behavior. Note that any model that provides a full-support probability prediction of the current action of a player given the history of play can be written as equation (1) for the following reason. Since probabilities of actions (cards) add up to 1, the ratios of probabilities (odds), such as $P(a^t_i = a \mid h^t)/P(a^t_i = K \mid h^t)$, $a = 1, 2, 3$, uniquely determine the probabilities of actions. Hence, any full-support model can be written as

$$\frac{P(a^t_i = a \mid h^t)}{P(a^t_i = K \mid h^t)} = g_a(h^t), \tag{3}$$

while equation (1) boils down to

$$\frac{P(a^t_i = a \mid h^t)}{P(a^t_i = K \mid h^t)} = \exp\Big\{(\beta_{i,a} - \beta_{i,K})^{\mathrm{T}} x_i(h^t)\Big\}. \tag{4}$$

If we let $x_i(h^t)$ be the vector of indicator functions for histories $(\cdots, \mathbf{1}_{\hat{h}^t}, \cdots)$, where $\hat{h}^t$ runs over all possible histories, and denote the element of vector $\beta_{i,a}$ that corresponds to the coefficient of $\mathbf{1}_{\hat{h}^t}$ by $\beta_{i,a,\hat{h}^t}$, then the left-hand side of (4) is simply equal to $\exp\Big\{\beta_{i,a,h^t} - \beta_{i,K,h^t}\Big\}$. Therefore, by setting $\beta_{i,a,h^t} - \beta_{i,K,h^t} = \log g_a(h^t)$, any full-support model is represented by

where $N^{(k)}$ is the number of occurrences of histories in $H^{(k)}$ in the data. Choosing $\beta^{(k)}_i$ to maximize the log-likelihood is equivalent to

$$\max_{p^{(k)}} \sum_a \frac{n^{(k)}(a)}{N^{(k)}} \log p^{(k)}(a) \quad \text{s.t.} \quad \sum_a p^{(k)}(a) = 1$$

for each k because there always exists $\beta^{(k)}_i$ that satisfies (2) for any $p^{(k)}$. The Lagrangian is $\mathcal{L} = \sum_a \frac{n^{(k)}(a)}{N^{(k)}} \log p^{(k)}(a) - \lambda^{(k)} \left(\sum_a p^{(k)}(a) - 1\right)$, and the first-order condition is

$$\frac{\partial \mathcal{L}}{\partial p(a)} = 0 \implies \frac{n^{(k)}(a)}{N^{(k)}} / p^{(k)}(a) = \lambda^{(k)}.$$

The constraint is satisfied with $\lambda^{(k)} = 1$, and therefore $p^{(k)}(a) = n^{(k)}(a)/N^{(k)}$, the empirical frequency of card a after the histories in $H^{(k)}$.

(1). Equation 4 also shows that $(\beta_{i,a} - \beta_{i,K})$ is identified but $\beta_{i,a}$ and $\beta_{i,K}$ are not. As we explained, in the baseline models, we employed normalization $\beta_{i,K} = 0$.

Given that (1) encompasses all models, our quest for the best one boils down to (i) the selection of variables $x_i(h^t)$ and (ii) possible parametrization of the coefficient vector $\beta_{i,a}$ by a smaller number of parameters, all within the ambient model (1).

If we have enough data in the sense that we have enough data points for each history h^t, we can perform non-parametric estimation, by using indicator functions of all histories $x_i = (\cdots, \mathbf{1}_{\hat{h}^t}, \cdots)$. This is infeasible because the number of complete histories is astronomical. Non-parametric estimation is feasible, however, if we postulate that subjects have limited memory. For example, if we postulate that current action only depends on the history of outcomes (action profiles) in the past two periods, the number of two-period histories is $(4 \times 4)^2 = 256$, and our data set with 74733 observations provides at least 66 observations of actions after *each* two-period history, which is large enough for a reliable non-parametric estimation of action distribution after each history. The following figure shows the distribution of the number of observations (occurrences) of two-period histories.

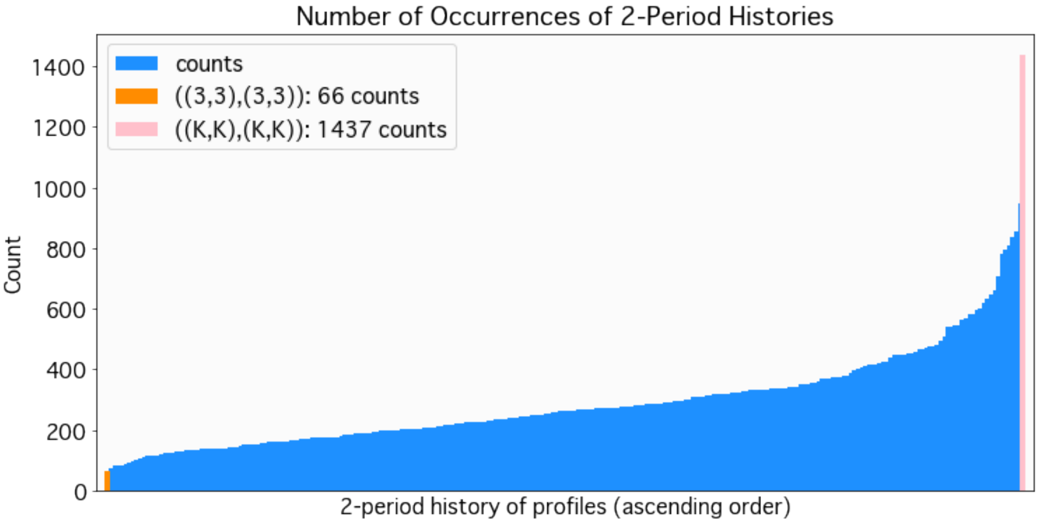


Figure 8. The number of occurrences of each two-period history $((a^t_{\mathrm{R}}, a^t_{\mathrm{B}}), (a^{t-1}_{\mathrm{R}}, a^{t-1}_{\mathrm{B}}))$ out of 74733 two-period histories in the data. The average count is 291.9. The most frequently observed history is $((K, K), (K, K))$, which occurs 1437 times in the data. The history of least frequent observation is $((3, 3), (3, 3))$, which still appears 66 times in the data.

Therefore, one of the models in the previous subsection, the logit model with two-period history dummies, would provide a reliable non-parametric estimation of the subjects' behavior provided that they have two-period memories.

Non-parametric models may not be the best choice for the following well-known reasons. If, for example, if the true data generation process is given by equation (1) with a small number of variables on the right-hand side, such as $x_i = (\mathbf{1}_{H^0}, \mathbf{1}_{H^1})$ for some subsets of histories H^0

and H^1, estimating this specification of the model is better (provides a smaller prediction error) than estimating a non-parametric model. How should we decide which variables to include in the model (1)? This is a classical statistical problem of model selection, and the statistics theory recommends that we use information criteria such as AIC and BIC. According to those methods, we must estimate each possible specification of the model and calculate a number called the AIC or BIC value, and then select the model with the minimum such value. This guarantees the optimal selection of the model in the appropriate sense as the number of data points goes to infinity.

The challenge of model selection we face is that the statistically optimal procedure is computationally infeasible. Even when we confine our attention to the selection of dummy variables of subsets of two-period history of play, the number of non-empty subsets is equal to $2^{256} - 1$ and the number of feasible combinations of those dummy variables is even larger. It is clearly impossible to compute and compare such a large number of AIC or BIC values. The machine learning models we introduce in Section 5 can be viewed as a practical way of selecting a good model (or as having been shown to select good models in real-life applications) when statistically optimal model selection is infeasible.

4.3 Experience-Weighted Attraction (EWA) Model

Next, we study economic learning models commonly used in the literature. Although various kinds of models have been invented, many of (not all of) them are categorized into two groups: reinforcement learning (RL) and belief learning (BL). Reinforcement learning is based on the idea that, if an agent chose an action and the outcome was good, that action is "reinforced" in the sense that the agent is more willing to choose it. This can be formulated as a model in which an agent has (unobserved) attraction (propensity) to an action, which embodies how many payoffs it gained in the past. Belief learning, also known as fictitious play, is a model in which an agent forms a belief on the choice probability of the opponent's action based on the weighted average of the past actions taken by the other player and plays the best response to the belief.

Camerer and Ho (1999) developed the experience-weighted attraction (EWA) learning model, which is a hybridization of the reinforcement learning model and the belief learning model. In their original EWA model, each player $i \in I$ chooses action $a \in C$ at period t with probability,

$$P_i(a_i^t = a \mid h^t) = \frac{\exp\left\{\lambda_i A_i^a(t-1)\right\}}{\sum_{c \in C} \exp\left\{\lambda_i A_i^c(t-1)\right\}}, \tag{5}$$

where

$$N_i(t) = \rho_i N_i(t-1) + 1, \text{ and} \tag{6}$$

$$A_i^a(t) = \frac{\phi_i N_i(t-1) A_i^a(t-1) + \left[\delta_i + (1-\delta_i) 1\left\{a_i^t = a\right\}\right] \pi_i(a, a_{-i}^t)}{N_i(t)}. \tag{7}$$

Here $A_i^a(t)$ is player i's attraction to action a at period t. The agent chooses actions according to the logit of these attractions as expressed in (5).

We have 14 parameters to estimate in total: for each player $i \in I$, λ_i, ρ_i, ϕ_i, δ_i, $(A_i^1(0), A_i^2(0), A_i^3(0))$, and $N_i(0)$; we normalize $A_i^K(0) = 0$. Here λ expresses how accurately the choice probability reflects the attractions. $\lambda_i \to +\infty$ means that the agent plays the action that has the highest attraction with probability one, while $\lambda_i = 0$ implies that the agent plays all actions with equal probability irrespective of the attractions. ρ_i, ϕ_i are the discount factors and δ_i represents the ratio between RL and BL. Thus, naturally, we assume $\lambda_i \in [0, \infty)$, $\rho_i, \phi_i \in [0, 1]$, and $\delta_i \in [0, 1]$.

When $\delta_i = \rho_i = 0$ and $N_i(0) = 1$, the model (5)-(7) reduces to a reinforcement learning model since the agent updates their attractions only if they actually take action j,

$$\begin{aligned}
&N_i(t) = 1 \text{ for all } t, \text{ and}\\
&A_i^j(t) = \phi A_i^j(t-1) + 1\left\{a_i^t = j\right\}\pi_i(j, a_{-i}^t).
\end{aligned}$$

On the other hand, when $\delta_i = 1$, $\rho_i = \phi_i$ and $N_i(0) = 1$, the model becomes a belief learning model:

$$\begin{aligned}
N_i(t) &= \frac{1-\rho_i^t}{1-\rho_i}, \text{ and}\\
A_i^j(t) &= \frac{\rho N_i(t-1) A_i^j(t-1) + \pi_i(j, a_{-i}^t)}{N_i(t)}.
\end{aligned}$$

To see why this is a belief learning model, define player i's belief that the other player plays $c \in C$ as a weighted empirical frequency,

$$\mu_i^c(t) = \frac{\sum_{\tau=1}^t \rho_i^{\tau-1} \cdot 1\left\{a_{-i}^\tau = c\right\}}{\sum_{\tau=1}^t \rho_i^{\tau-1}}.$$

Then we can show that $A_i^j(t)$ is an expected payoff (with respect to μ_i) when player i plays action j, that is,

$$\begin{aligned}
A_i^j(1) &= \pi_i(j, a_{-i}^1) = \sum_{c \in C} \pi_i(j, c)\mu_i^c(1)\\
A_i^j(2) &= \frac{\rho_i \pi_i(j, a_{-i}^1) + \pi_i(j, a_{-i}^2)}{1+\rho} = \sum_{c \in C} \pi_i(j, c)\mu_i^c(2)\\
&\vdots\\
A_i^j(t) &= \frac{\sum_{\tau=1}^t \rho_i^{t-\tau} \pi_i(j, a_{-i}^\tau)}{\sum_{\tau=1}^t \rho_i^{t-\tau}} = \sum_{c \in C} \pi_i(j, c)\mu_i^c(t).
\end{aligned}$$

Therefore, an agent following BL plays the best (logit-smoothed) response with respect to their belief $\mu_i(t)$.

In O'Neill's game, the roles of number cards are symmetric, and therefore the subject may first choose between K and number cards, and when the number cards option is chosen, the subject further considers which number card to choose. Such a decision rule can be captured by the following nested logit version of the EWA model (see, for example, Train (2009)). The nested logit model has

$$P_i(a_i^t = a \mid h^t) \propto \begin{cases} e^{\lambda_i A_i^a(t-1)/\eta_i} \left(\sum_{c=1,2,3} e^{\frac{\lambda_i}{\eta_i} A_i^c(t-1)}\right)^{\eta_i - 1} & \text{if } a = 1,2,3 \\ e^{\lambda_i A_i^a(t-1)} & \text{if } a = K. \end{cases} \tag{8}$$

When $\eta_i = 1$, then (8) reduces to the original logit EWA model. In the limit when η_i tends to $+\infty$, the model effectively chooses K or number cards, and when the number cards option is chosen, each number card is chosen with an equal probability. Our estimation result below shows somewhere in between those polar cases.

4.3.1 Estimates of the Parameters

We estimate the parameters using maximum likelihood estimation using all sample data. Table 5 presents the estimated model parameters and the standard errors of the four specifications. We also report the likelihood ratio test statistic comparing the original EWA and the other models [*7].

The LR test clearly indicates that the original EWA model fits the data better than the BL and the RL. The estimates of the mixing parameters in EWA, $\delta_{\mathrm{R}} = 0.450$ and $\delta_{\mathrm{B}} = 0.000$, suggest that the red player plays the mixture of RL and BL, while the black player plays the complete RL.

The difference between the original EWA and the nested EWA is also significant, but the difference is smaller than the difference between the EWA and (RL or BL). The estimated discount factors and the mixing parameters are very similar between the two models.

As expected, we find that the estimated initial attractors of the numbers are all negative (compared to $A_{\mathrm{R}}^K(0) = A_{\mathrm{B}}^K(0) = 0$). Thus, both red and black players play K with a higher probability than 1, 2, 3 from the beginning.

5 Machine Learning Models

In this section, we provide a brief overview of machine learning in comparison to conventional econometric models. Machine learning refers to a class of models that make predictions or decisions based on observable data. For our purposes, we focus on machine learning models

[*7] For BL and RL, we report

$$\text{LR statistic} = 2 \times \Big((\text{Log-likelihood of EWA}) - (\text{Log-likelihood of RL or BL})\Big).$$

For nested EWA, we report

$$\text{LR statistic} = 2 \times \Big((\text{Log-likelihood of nested EWA}) - (\text{Log-likelihood of EWA})\Big).$$

Each test statistic asymptotically follows the chi-squared distribution whose degree of freedom is equal to the difference in the number of parameters between the two specifications.

Table 5. Estimation results of EWA-variant models

Models	Logit			Nested logit
	EWA	RL	BL	Nested EWA
Discount factors				
ϕ_{R}	1.014	1.034	2.235	1.002
	(0.006)	(0.005)	(0.022)	(0.000)
ϕ_{B}	1.059	1.025	4.189	1.001
	(0.009)	(0.003)	(0.136)	(0.001)
ρ_{R}	0.870	$\underline{0.000}$	$\underline{= \phi_{\mathrm{R}}}$	0.901
	(0.081)			(0.021)
ρ_{B}	1.003	$\underline{0.000}$	$\underline{= \phi_{\mathrm{B}}}$	0.956
	(0.016)			(0.008)
Mixing parameters				
δ_{R}	0.450	$\underline{0.000}$	$\underline{1.000}$	0.451
	(0.030)			(0.006)
δ_{B}	0.000	$\underline{0.000}$	$\underline{1.000}$	0.000
	(0.027)			(0.003)
Accuracy parameters				
λ_{R}	0.480	0.084	0.370	0.961
	(0.255)	(0.004)	(0.004)	(0.194)
λ_{B}	0.894	0.094	0.333	2.237
	(0.113)	(0.003)	(0.010)	(0.335)
Nest parameters				
η_{R}	$\underline{1.000}$	$\underline{1.000}$	$\underline{1.000}$	5.777
				(0.276)
η_{B}	$\underline{1.000}$	$\underline{1.000}$	$\underline{1.000}$	9.038
				(0.503)
Initial values				
$A^1_{\mathrm{R}}(0)$	−0.835	−6.413	−1.737	−5.411
	(0.472)	(0.692)	(0.029)	(1.136)
$A^2_{\mathrm{R}}(0)$	−1.110	−8.520	−2.278	−6.080
	(0.627)	(0.772)	(0.037)	(1.276)
$A^3_{\mathrm{R}}(0)$	−1.270	−8.366	−2.625	−6.510
	(0.715)	(0.758)	(0.042)	(1.365)
$A^1_{\mathrm{B}}(0)$	−0.665	−2.285	−1.692	−3.843
	(0.107)	(0.571)	(0.063)	(0.604)
$A^2_{\mathrm{B}}(0)$	−0.893	−5.863	−2.261	−4.443
	(0.142)	(0.636)	(0.084)	(0.698)
$A^3_{\mathrm{B}}(0)$	−0.932	−8.836	−2.343	−4.507
	(0.145)	(0.714)	(0.087)	(0.708)
$N_{\mathrm{R}}(0)$	5.258	$\underline{1.000}$	$\underline{1.000}$	9.619
	(2.748)			(1.916)
$N_{\mathrm{B}}(0)$	5.162	$\underline{1.000}$	$\underline{1.000}$	20.378
	(0.740)			(2.981)
No. of Observations	77310	77310	77310	77310
No. of Parameters	16	10	10	18
Log Likelihood	-208940.8	-209391.3	-209637.0	-208670.4
LR Statistic		901.0***	1464.4***	519.2***

Notes: Maximum likelihood estimates of the parameters of EWA variant models. Standard errors are in parentheses. Underlined values are determined by the model restrictions and are not estimated. The LR Statistic is the likelihood ratio test statistic comparing the EWA and the other models. *p<0.1; **p<0.05; ***p<0.01.

to make a prediction:

$$y = f(x|\beta),$$

where x is the input data, y is the predicted outcome, and β is the vector of parameters. This looks exactly the same as the conventional econometric models, but there are some notable differences.

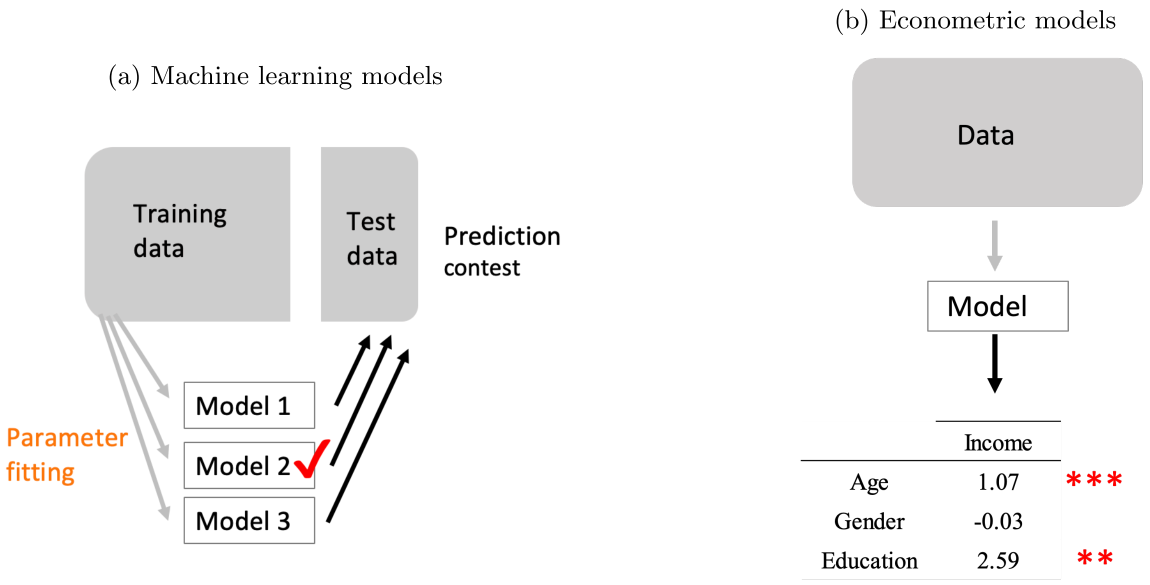
First, machine learning offers new functional forms that have not been employed in conventional econometrics. Leading examples include deep learning, decision trees, and LASSO, which we will explain and use in later sections.

Second, the main objectives of machine learning and econometrics are different. Machine learning is all about selecting the best model to make a prediction (Figure 9a). The data set at hand is partitioned into “training data” and “test data”, and the parameter values of models are determined so that the goodness of fit is higher on the training data. Then a prediction contest is conducted using the test data to select the model that makes the best prediction. The main concern is how to make the best prediction, and the calculated parameter values are oftentimes not paid much attention. This reflects the fact that some leading machine learning models are “black boxes” and the meaning of parameters is not immediately clear. (We will come back to this issue in Section 6).

In contrast, the main concern of econometrics is parameter estimation (Figure 9b). Here, the model is given a priori (such as linear regression or logit), and assuming that the true data-generating mechanism is within the model, the parameter values are estimated to infer the true values. The magnitude, sign, and statistical significance of the estimated parameter values are the main concerns of the researchers so as to understand how data is generated.

Third, machine learning is data-driven while econometrics is driven by theory. The ultimate goal of machine learning is to make good predictions in real-life data sets. Machine learning models are not derived by finding the optimal one in a certain class of functional forms for a certain class of tasks, but instead by constantly running prediction contests on real-life data to select and improve the existing models. As a result, the reason why certain models perform well for certain tasks is often not well understood. In contrast, econometric methods are derived from the statistical theory of optimal estimation.

In this paper, we adopt the machine learning paradigm of conducting a prediction contest to compare conventional behavioral economics models and machine learning models in our unique “big” data set derived from our mixed strategy experiment. We split our data into training data and test data as illustrated in Figure 9a to see which models perform well. When we find that machine learning models make good predictions, we try to “open” the black boxes to gain insight into what factors are important in explaining the subjects’ behavior in our experiment. This is in line with the recent trend in machine learning known as XAI (explainable artificial intelligence). Before moving on to the description of the models, we explain in the next subsection how data sets are split in the machine learning approach.


(a) Machine learning models

(b) Econometric models

Figure 9. Difference between econometric models and machine learning models. The left panel 9a describes machine learning models, and the right panel 9b describes econometric models.

5.1 Cross Validation

If we split the data set at hand as in Figure 9a, we can conduct a prediction contest only once. In the machine learning approach, however, prediction contests are generally conducted multiple times on the given data set. This is called cross-validation (CV)[*8], and works as follows. First, the whole data set is randomly split into K subsets of an (almost) equal size, with the typical value of K being 5. Since our data contains 2577 pairs, we created two subsets of 516 pairs and three subsets of 515 pairs. Call them $D_1, \dots, D_5$. Second, the first subset D_1, which accounts for 20% of the whole data set, is set aside as test data, and parameter fitting is conducted on the remaining 80% of training data $D_2 \cup \cdots \cup D_5$. Then we evaluate the performance of the trained model using the test data D_1. In general, we set aside D_k as test data, conduct parameter fitting on the remaining training data $\bigcup_{h \neq k} D_h$, and then measure its prediction performance using D_k. This process is repeated for $k = 1, \dots, 5$. In this way, even though we have a single set of data, we can conduct prediction contests five times.

We use CV to compare the out-of-sample prediction performance of machine learning models and conventional behavioral economics models. The performance measures we adopted and the other details are discussed in Section 7.

[*8] Specifically, this process is called leave-one-out cross-validation.

CV is also used to determine the hyperparameters of machine learning models. In this case, we split the training data (in each CV) into sub-training data and sub-test data, and choose the hyperparameters of a model so that it minimizes the average prediction error (KL divergence between the prediction and the actual choices) in the sub-test data. Details of the procedure will be explained when we describe the machine learning models that we consider in the following subsections.

5.2 Decision Trees

One of the most commonly used machine learning models is the **decision tree**. The decision tree model partitions the space of "features" of data recursively according to a tree structure, and it returns predictions, one for each subset of features finally obtained by the tree (Figure 10). Figure 10 shows a tree with depth 2. The depth of the tree is defined to be the maximum length of paths from the root to the terminal nodes of the tree.

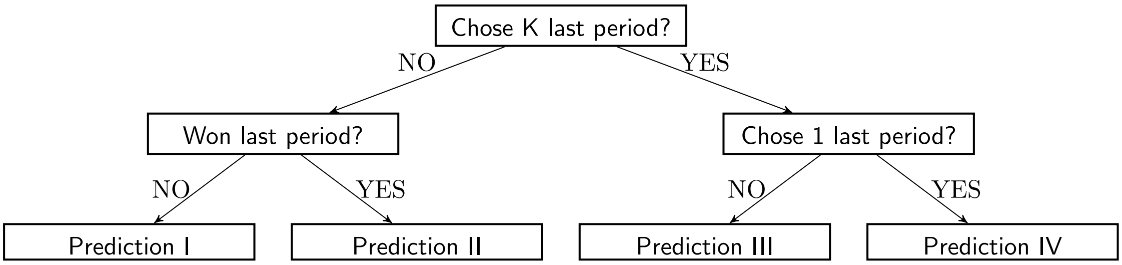


Figure 10. An example of a decision tree.

In this paper, we predict the choice probability of each action using a classification tree. The decision tree automatically classifies past histories of play according to the similarity of current action choices. We need, however, to provide the set of "features" of past history of play that the decision tree utilizes to perform its classification. We use 893 features that are binary variables based on the history of the past three periods [*9]. We do not have enough data points to go beyond three-period histories, as we discussed in Section 4.1. The 893 features of history of play (action profile) in the past three periods form a set $F = \{f_1, \ldots, f_{893}\}$, where each feature f_n is a property of the three-period history whose occurrences can be answered by Yes or No. An example of a feature is whether the focal player won in the last period or not. Note that a feature f corresponds to a subset of three-period histories where f is true. By choosing our feature set F, we are effectively looking at subsets of three-period histories that have a clear meaning (in some informal, intuitive sense). Features such as "the numbers of the cards chosen in the past three periods add up to 8" are a priori excluded

[*9] The set of 893 features consists of all features that are used in LASSO and based on a history of up to three periods.

from our feature set, although such features may possibly play a role in the focal player's choice and this fact could possibly be detected by other machine learning models.

The input data for the decision tree comes in pairs (x, y) consisting of a feature vector $x = (x_1, \ldots, x_{893})$ and an outcome label y, where x_i is equal to 1 if the history associated with the data has the feature f_i and otherwise 0, and $y = (y_1, y_2, y_3, y_K)$ is the degenerate probability vector of the current action of the focal player. For example, if the focal player chooses K in the current period, we have $y = (0, 0, 0, 1)$.

Each node m of a decision tree is associated with a subset of data $Q(m)$, and two outgoing branches Yes and No. The node m uses a feature $f(m)$ to classify $Q(m)$, and we let $Q(m, Y)$ be the subset of $Q(m)$ that has the feature $f(m)$ and $Q(m, N)$ be the subset where the feature does not hold. The successor node of m after the Yes (No) is associated with $Q(m, Y)$ ($Q(m, N)$).

The decision tree model chooses the feature $f(m)$ to minimize the prediction errors of the current choice of action in $Q(m, Y)$ and $Q(m, N)$, which are defined in the following way. Let y_a^{*j} be the prediction of probability of action a in the data subset $Q(m, j), j = Y, N$. The associated prediction error is given by the Kullback-Leibler (KL) divergence. The KL divergence between two probability distributions p and q is defined as $\sum_a p_a \log \left(\frac{p_a}{q_a} \right)$, and it is minimized at $q = p$. The KL divergence in the data subset $Q(m, j), j = Y, N$ is

$$\sum_{(x,y) \in Q(m,Y)} \sum_a y_a \log \left(\frac{y_a}{y_a^{*j}} \right) = |Q(m, j)| \sum_a \overline{y}_a^j \log (\frac{\overline{y}_a^j}{y_a^{*j}}), \tag{9}$$

where $\overline{y}_a^j$ is the average frequency of action a in the subset $Q(m, j)$. Hence, the optimal prediction to minimize the right-hand side is $y_a^{*j} = \overline{y}_a^j$. In other words, the optimal prediction in data subset $Q(m, j)$ is given by its empirical frequencies of actions. Given this and the convention for the KL divergence that $y_a \log y_a$ is defined to be zero when $y_a = 0$, the prediction error in terms of the KL divergence (9) reduces to

$$H(Q(m, j)) = -|Q(m, j)| \sum_a \overline{y}_a^j \log(\overline{y}_a^j),$$

which is known as the Shannon entropy in the data subset $Q(m, j)$. In summary, our decision tree chooses the feature $f(m)$ to minimize the total prediction error in terms of the KL divergence, which is equal to the sum of the Shannon entropies

$$H(Q(m, Y)) + H(Q(m, N)).$$

The decision tree branches in this way until the prespecified depth is reached[*10]. The subsets associated with the final nodes of the tree partition the whole data set, and the

[*10] If we cannot further classify the past three-period history before reaching the prespecified depth, branching from node m is suppressed. For example, when node m is reached after "$f = (K, K)$ is played in t-1, Yes", "$f' = (1, 2)$ was played in t-2, Yes", and "$f'' = (K, 1)$ was played in t-3", that uniquely pins down the past three-period history and there is no room for further classification.

prediction of choice probability in each subset is given by the average of the choice vector y, which is equal to the vector of empirical frequencies of the cards in the subset.

Now, how is the maximum depth of the tree determined? The depth is called a hyperparameter, which determines the overall structure of the model, and it has to be chosen before the parameter fitting (training) of the model is performed. The choice of hyperparameter is made using the "nested" cross-validation method, and it works as follows. In cross-validation, we split the data into five subsets of equal size $D_1, ..., D_5$ and let D_i be the test data and $\bigcup_{h \neq i} D_h$ be the training data for $i \in \{1, \ldots, 5\}$. Next, each training data subset $\bigcup_{h \neq i} D_h$ is again partitioned into subsets of equal size $d_1^i, \ldots, d_5^i$. And then, for each depth $k = 1, 2, \ldots, 12$, we fit the decision tree model with training data $\bigcup_{h \neq j} d_h^i$ and derive the prediction performance using the test data d_j^i for each $j \in \{1, \ldots, 5\}$. The prediction performance is measured by the Kullback-Leibler divergence of the action actually chosen in the test data from the prediction. Finally, we select the maximum depth of the tree that achieves the best average prediction based on five test data subsets $d_1^i, \ldots, d_5^i$. In this way, we can select the maximum depths of the tree for training data $\bigcup_{h \neq i} D_h$ for each $i \in \{1, \ldots, 5\}$. This is the cross-validation process for selecting the hyperparameter, and the other machine learning methods in this paper also select their hyperparameter in this way.

The above decision tree model can be regarded as a procedure for selecting the right-hand side variables in the multinomial logit model (1) to be estimated by maximum likelihood. Let us illustrate this fact by the example of a decision tree in Figure 10. Let $D_{K(-1)}, D_{W(-1)}$, and $D_{1(-1)}$ be the dummy variables (indicator functions) for "the focal player chose K in the last period", "the focal player won in the last period", and "the focal player chose 1 in the last period." The decision tree in Figure 10 is equivalent to the maximum likelihood estimation of the multinomial logit model 1 where the right-hand side variables are

$$x_i = \begin{pmatrix} D_{K(-1)} \times D_{W(-1)} \\ D_{K(-1)} \times (1 - D_{W(-1)}) \\ (1 - D_{K(-1)}) \times D_{W(-1)} \\ (1 - D_{K(-1)}) \times (1 - D_{W(-1)}) \end{pmatrix}.$$

Maximum likelihood estimation of the coefficients of $D_{K(-1)} \times D_{1(-1)}$, one for each card, makes the estimated choice probability when $D_{K(-1)} \times D_{1(-1)} = 1$ equal to the empirical frequencies of actions in the subset of data where $D_{K(-1)} \times D_{1(-1)} = 1$, as we explained in Section 4.1. In summary, the above-mentioned version of a decision tree provides a procedure for selecting the right-hand side variables of the multinomial logit model, where the variables take the form of the product of history dummies and (1−history dummy)'s.

5.2.1 What we can learn from the decision tree model

We fit the trees using all of the samples. The chosen maximum depth of the trees for the red player is 4 in four trees and 3 in one tree, and those for the black player, 4 in three trees and 3 in two trees. The decision trees are depicted in Appendix A.1.

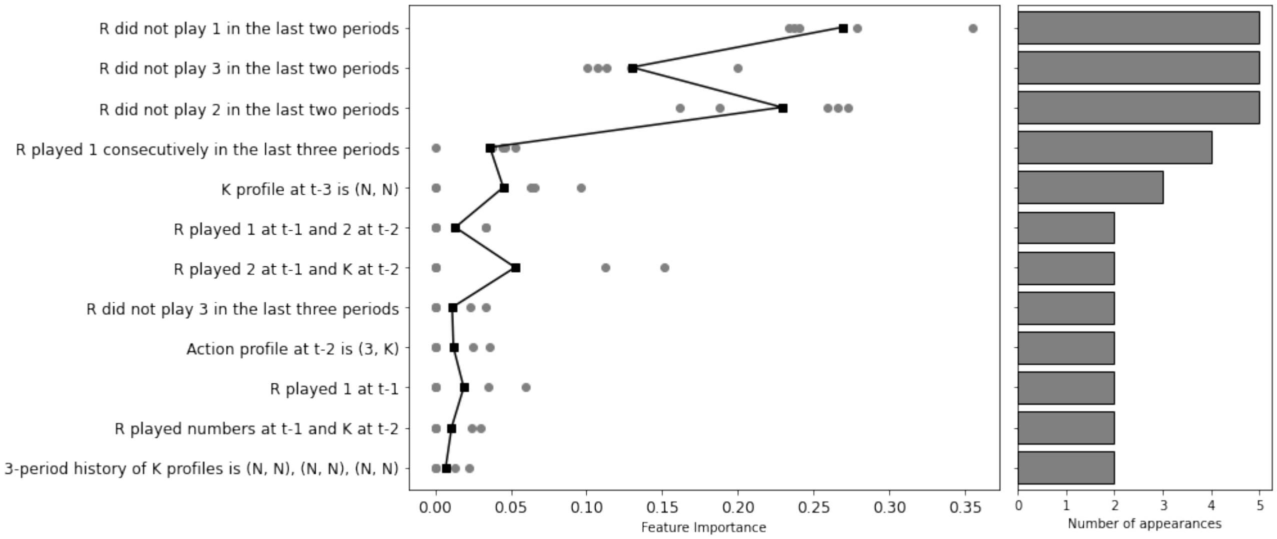


Figure 11. Feature importance and the number of occurrences (duplicates within a single tree do not count) in five trees for the red player. The gray dots correspond to feature importance in each CV and the black dots are the average of them. Note that we plot the importance of the features that appear at least twice. For the importance of the features that appear only once, see Appendix A.1.

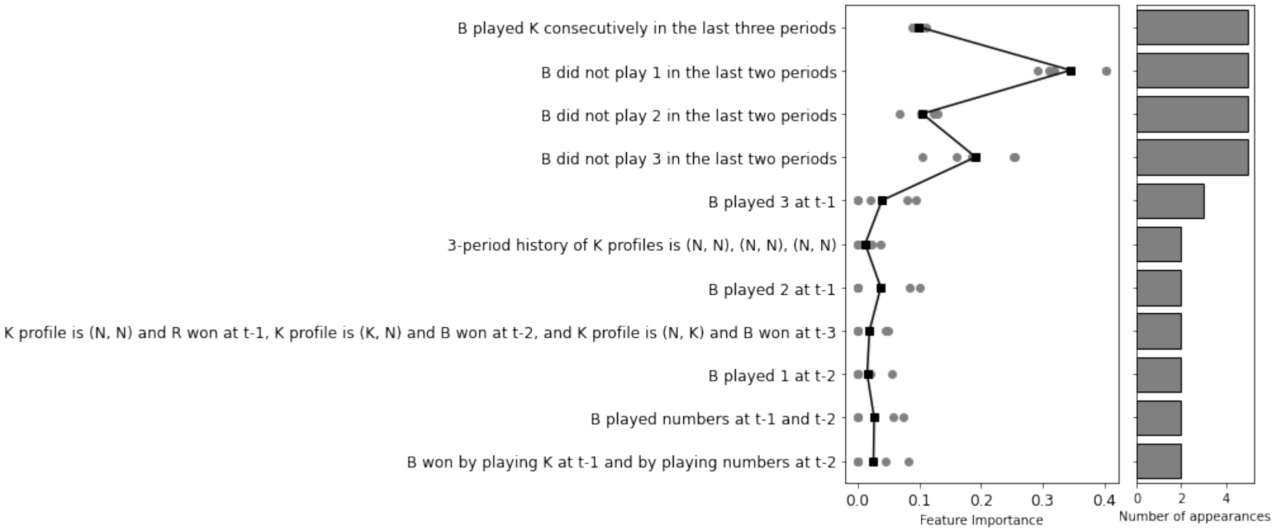


Figure 12. Feature importance and the number of occurrences (duplicates within a single tree do not count) in five trees for the black player. The gray dots correspond to feature importance in each CV and the black dots are the average of them. Note that we plot the importance of the features that appear at least twice. For the importance of the features that appear only once, see Appendix A.1.

Figures 11 and 12 summarize the feature importance and how many of the five trees used each variable for branching for the red player's and black player's trees. The importance of feature f, denoted by $I(f)$, is defined as follows. First, for any subset of data Q, consider predicting the probabilities of current action by means of the empirical frequencies in the subset Q. The resulting Shannon entropy is $H(Q) = -|Q| \sum_a \overline{y}_a \log(\overline{y}_a)$, where $\overline{y}$ is the vector of average frequencies of current action in subset Q. Second, define the feature importance by

$$I(f) = \frac{\sum_{m \in M(f)} \Delta H_m}{\sum_m \Delta H_m},$$

where $M(f)$ is the set of nodes associated with feature f and

$$\Delta H_m = H(Q(m)) - H(Q(m, Y)) - H(Q(m, N))$$

is the reduction of Shannon entropy at node m. Intuitively, the feature importance of f indicates how much the use of feature f improves the prediction of the model. Looking at Figures 11 and 12, we can see that for both players, the choice probability is affected when the player did not take the action in the past two periods. In particular, the decision tree shows that actions not taken in the past two periods are more likely to be taken in the current period.

5.3 LASSO

Another machine learning model that is commonly used in the literature is LASSO (Least Absolute Shrinkage and Selection Operator), which enables us to perform an automatic selection of variables. We include $m = 1731$ variables (for each card $c \in C$) as potential covariates of the multinomial logit model and add an L1-penalty term into the log-likelihood. That is, the LASSO estimator $\hat{\beta}$ is the solution to the following penalized log-likelihood maximization problem

$$\max_{\beta} LL(x \mid \beta) - \lambda ||\beta||_1, \tag{10}$$

where LL is the log-likelihood function

$$LL_i(x \mid \beta) \equiv \sum_{s=1}^{S} \sum_{t=1}^{30} \sum_{a \in C} \beta_{i,a}^{\mathrm{T}} x_i(h_s^t) - \ln \left(\sum_{c \in C} \exp\left\{ \beta_{i,c}^{\mathrm{T}} x_i(h_s^t) \right\} \right)$$

for each $i = \mathrm{R}, \mathrm{B}$. Here $x_i(\cdot) \in \mathbb{R}^m$ is the candidate covariates computed by the histories.

The 1,731 candidate covariates that we employ are selected according to the following criteria. First, we include all one- and two-period history dummies. Second, since we do not have enough observations of longer histories, we focus on the reduced histories such as histories of K-number profiles and histories of win-lose patterns to incorporate the associated dummies in the last four periods. Finally, we also incorporated non-dummy variables that count how many times the same action as in the previous period was consecutively chosen

(= the maximum number n such that a given action was played in the last n periods). Since LASSO minimizes the sum of the absolute values of the estimated parameters, the size of covariates should be normalized. We normalize those non-dummy variables to a mean of 0 and a standard deviation of 1. All of the variables we used are briefly explained in Table 6 and fully listed in Table 22 in the Appendix.

Constant
Period specific constant
R's action in t-1, t-2, t-3, t-4
B's action in t-1, t-2, t-3, t-4
Action profile in t-1, t-2
R's action was a number or K in t-1, t-2, t-3, t-4
B's action was a number or K in t-1, t-2, t-3, t-4
Action profile (number or K) in t-1, t-2, t-3, t-4
History of R's actions in the last 2, 3, 4 periods
History of B's actions in the last 2, 3, 4 periods
History of action profiles in the last 2 periods
History of action profiles (number or K) in the last 2, 3, 4 periods

R played 1 (or 2, 3, K) consecutively in the last three periods
B played 1 (or 2, 3, K) consecutively in the last three periods
R did not play 1 (or 2, 3, K) in the last three periods
B did not play 1 (or 2, 3, K) in the last three periods
R played K (or numbers) in t-1, ..., t-n consecutively (n=4, 5, ..., 29)
B played K (or numbers) in t-1, ..., t-n consecutively (n=4, 5, ..., 29)
R won or lost in periods t-1, t-2, t-3, t-4
History of winners in the last 2, 3, 4 periods
History of actions (number or K) and winners in t-1, t-2, t-3, t-4
History of action profiles (number or K) and winners in t-1, t-2, t-3, t-4

Table 6. The list of variables included in LASSO. Here t is the current period, that is, the LASSO model predicts the action in period t using the variables listed above.

The Lagrange multiplier of the LASSO optimization (10), λ (> 0), is a hyperparameter that determines how much the model penalizes the nonzero coefficients. As λ increases, the estimated model tends to have fewer nonzero coefficients. We determined the value of λ by the "nested cross-validation" method that we explained for our decision tree model. In particular, we performed cross-validation *within* the training data; each training data set of the cross-validation are randomly partitioned into four subsets of an equal size, and we repeated the following process for various candidate values of λ:

1. Choose one subset (1/4 of the training data) as the "subtest data," and use the union of the remaining subset (3/4 of the training data) as the "subtraining data."

2. Estimate the LASSO model with a given value of λ in the subtraining data.
3. Calculate the KL divergence between the mixed actions predicted by the estimated LASSO model and the actual choices.

We repeated this for four times for the four possible choices of the subtest data. We then choose the best λ that minimizes the average KL divergence over the four rounds of the nested cross-validation that perform items 1-3 above.

5.3.1 Estimates of the parameters

Table 7. Number of Coefficients Selected by LASSO

Player	Coefficient	No. of Selected Variables (average of five CVs)	No. of Commonly Selected Variables in all five CVs
Red	$\beta_{\mathrm{R},1}$	68.0	12
Red	$\beta_{\mathrm{R},2}$	65.0	21
Red	$\beta_{\mathrm{R},3}$	68.8	18
Red	$\beta_{\mathrm{R},K}$	113.2	44
Black	$\beta_{\mathrm{B},1}$	83.6	30
Black	$\beta_{\mathrm{B},2}$	79.6	31
Black	$\beta_{\mathrm{B},3}$	71.8	21
Black	$\beta_{\mathrm{B},K}$	111.0	45

Since the number of variables selected by LASSO is so large that we cannot fully present the estimated parameters in the main body of the paper, we show in Table 22 in Appendix A.2 the parameters estimated for the red player using the entire data set (instead of presenting five estimation results of cross-validation). The chosen LASSO hyperparameter is $\lambda^* = 0.0439$, and we get 317 variables in total (the sum of the numbers of nonzero coefficients of the four cards). One way to gain some insight from such a large number of variables selected by LASSO is to focus on the subset of variables that are commonly selected in the five rounds of cross-validation for performance testing. The next subsection examines those "key" variables selected by LASSO.

5.3.2 What we can learn from LASSO

Since the five rounds of cross-validation choose different values of λ and utilize different training data, the selected variables differ across the five rounds of cross-validation. To gain some insight from LASSO, we examined the variables commonly selected in the five rounds.

Among $m = 1731$ variables, the average number of selected variables is 315 for the red player and 346 for the black player in total. On the other hand, the number of commonly selected variables is 95 for the red player and 127 for the black player. The details of the numbers are shown in Table 7.

We present the list of those variables in Table 8 (the red player, card 1) and Table 9 (the red

Table 8. Parameters that LASSO does not eliminate in all five train-test splits (the red player, card 1)

	$\beta_{\text{R},1}$	
	Value	Count
Constant	-0.084	53586
R played 1 at t-1	-0.132	12781
R played 2 at t-2	0.068	12772
R consecutively played 1 in the last 2 periods	0.097	2378
R played 1 at t-1 and 2 at t-2	-0.046	3011
R did not play 1 in the last 2 periods	0.275	30411
R consecutively played 1 in the last 3 periods	0.618	574
R did not play 1 in the last 3 periods	0.071	22266
B did not play 1 in the last 3 periods	0.081	22533
R consecutively played 1 and lost in the last 2 periods	0.139	771
Period Constant (t=6)	0.065	2061
Period Constant (t=27)	-0.035	2061

Notes: The value column indicates the point estimates. The count column specifies the number of histories in which each dummy variable should be equal to 1.

player, card K). Other tables are relegated to Appendix A.2. The values of the parameters are estimated using all the sample data. The complete estimation table is shown in Table 22 in Appendix A.2.

5.4 Deep Neural Network (DNN)

Deep neural networks (DNNs) are the most successful machine learning model and have many applications such as image recognition, natural language processing, drug discovery, and self-driving cars. For the estimation of behavioral models, show that their DNN model outperforms a reinforcement learning and a fictitious play model in terms of the prediction of outcomes in 2 × 2 repeated game experiment data.

We created a five-layer neural network composed of one input layer, one output layer, and three hidden layers described in Figure 13. To allow us to make the time series prediction, we use the four-period history of (the action profile, the payoffs to the focal player) as input and then estimate the choice probabilities of the current period[*11]. That is, each input is a vector of $(16 + 1) \times 4 = 68$ dummies across four periods[*12], and each output is a four-dimensional probability vector. The number of cells in the three hidden layers are hyperparameters that are determined by the nested cross-validation, where the number of cells in each hidden layer is optimally selected from the candidate set $\{10, 30, 50\}$. Each cell in the hidden and output layers is "densely connected" in the sense that it is connected to all cells in the previous

[*11] Even if the payoff to the focal player can be deduced from the action profile, including the payoff data helps to reduce the prediction error of the model.

[*12] For each period, we used 16-dimensional dummy variables of action profiles and 1-dimensional payoff of the red player as input data.

Table 9. Parameters that LASSO does not eliminate in all five train-test splits (the red player, card K)

	$\beta_{\mathrm{R},K}$	
	Value	Count
Constant	0.595	53586
B played K at t-2	0.014	19378
B played numbers at t-2	−0.011	34208
R played K at t-3	0.007	18621
R played numbers at t-3	−0.006	34965
B played K at t-3	0.060	19447
B played numbers at t-3	−0.032	34139
R played 2 at t-4	−0.021	11394
R played K at t-4	0.029	18695
R played numbers at t-4	−0.024	34891
B played K at t-4	0.119	19513
B played numbers at t-4	−0.027	34073
Action profile at t-2 is (1, 3)	0.042	2577
K profile at t-2 is (N, K)	0.006	11895
K profile at t-2 is (N, N)	−0.006	23063
K profile at t-3 is (K, K)	0.002	7503
K profile at t-3 is (N, N)	−0.130	23021
Three-period K history is ((N, K), (K, N), (N, K))	0.084	601
Three-period K history is ((N, N), (K, N), (N, K))	−0.144	1179
K profile at t-4 is (K, K)	0.021	7515
K profile at t-4 is (N, N)	−0.131	22893
R played K at t-1 and t-2	0.109	6169
R played K at t-1 and numbers at t-2	−0.044	12468
R played numbers at t-1 and K at t-2	−0.044	12459
R played numbers at t-1 and t-2	0.007	22490
B played K at t-1 and numbers at t-2	−0.099	12642
R consecutively played K in the last 3 periods	0.231	2252
B consecutively played K in the last 3 periods	0.168	2592
R consecutively played numbers in the last 3 periods	0.113	14039
R consecutively played numbers in the last 4 periods	0.058	8698
R consecutively played numbers in the last 7 periods	−0.096	2223
R consecutively played numbers in the last 8 periods	−0.118	1525
R consecutively played numbers in the last 10 periods	−0.115	793
B consecutively played numbers in the last 4 periods	0.034	8404
B consecutively played numbers in the last 6 periods	−0.066	3476
B consecutively played numbers in the last 8 periods	−0.260	1659
R won by playing 3 at t-1	0.076	4609
R won by playing K at t-1 and won by playing a number at t-2	−0.138	2222
B won by playing a number at t-2	−0.020	19206
B won by playing a number at t-1 and lost by playing K at t-2	−0.036	2473
K profile at t-2 is (N, N) and B won	−0.017	8061
Period constant (t=6)	−0.072	2061
Period constant (t=23)	0.058	2061
Period constant (t=30)	0.197	2061

Notes: The value column indicates the point estimates. The count column specifies the number of histories in which each dummy variable should be equal to 1.

layer.

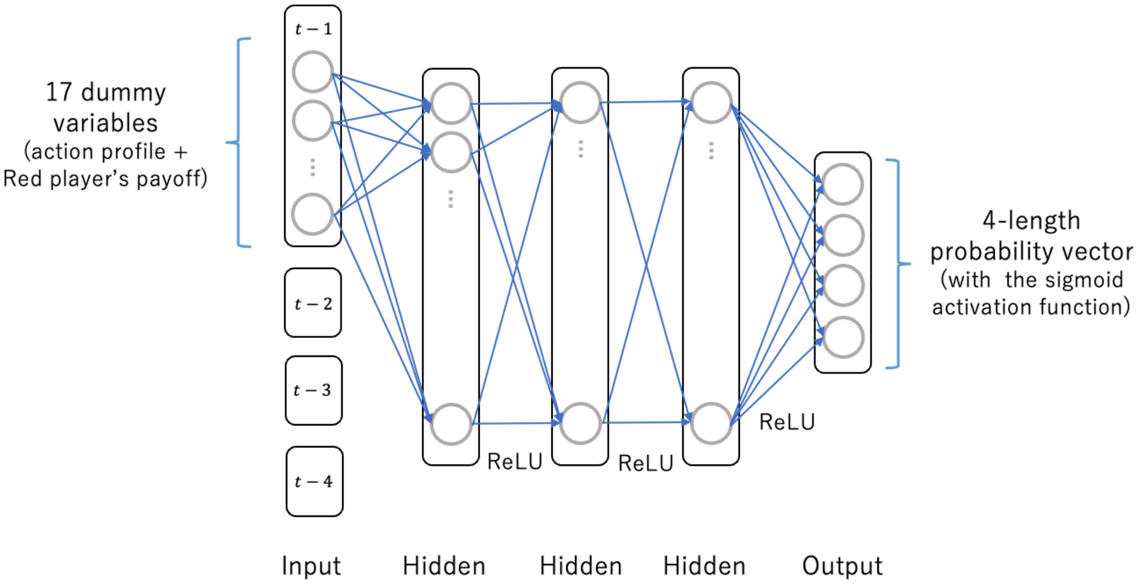


Figure 13. An illustration of a DNN model.

DNN recursively determines the value of each cell in the following way. Let j be a cell in the hidden or output layer and let $i = 1, ..., I$ be the cells in the previous layer. The branch from i to j is associated with parameter (or "weight") w_{ij}, and given the values x_i of previous cells $i = 1, ..., I$, the "input" to cell j is determined as $\sum_i w_{ij}x_i$ and the value of j, denoted x_j, is determined by "activation function" f as $x_j = f(\sum_i w_{ij}x_i)$. We adopt the ReLU activation function $f(x) = \max\{0, x\}$ for all cells except those in the output layer. Cells in the output layer use the softmax activation function (multinomial logit). The parameters w_{ij} are selected to provide the best fit of the DNN choice probabilities to the actual choice frequencies in the training data in terms of Kullback-Leibler divergence.

We used the Keras TensorFlow implementation of a standard DNN. Our loss function is the Kullback-Leibler divergence (cross-categorical entropy) between the estimated choice probability vector and the 0-1 indicator vector. To avoid overfitting, we used early stopping and dropouts when training the model. We trained our model with the Adam optimizer on Google Colaboratory.

5.5 Long-Short Term Memory (LSTM)

Another commonly used neural network model for sequential data is the recurrent neural network (RNN). Unlike a DNN, an RNN has a closed circuit that can maintain states inside its network. Long-Short Term Memory (LSTM) is a specific type of RNN architecture developed by Hochreiter and Schmidhuber (1997), which solves the drawback of simple RNNs only being able to sustain states for several periods. It is widely used in applications such as Google and Facebook machine translation systems, Google's voice recognition application, and autocorrect on iOS.

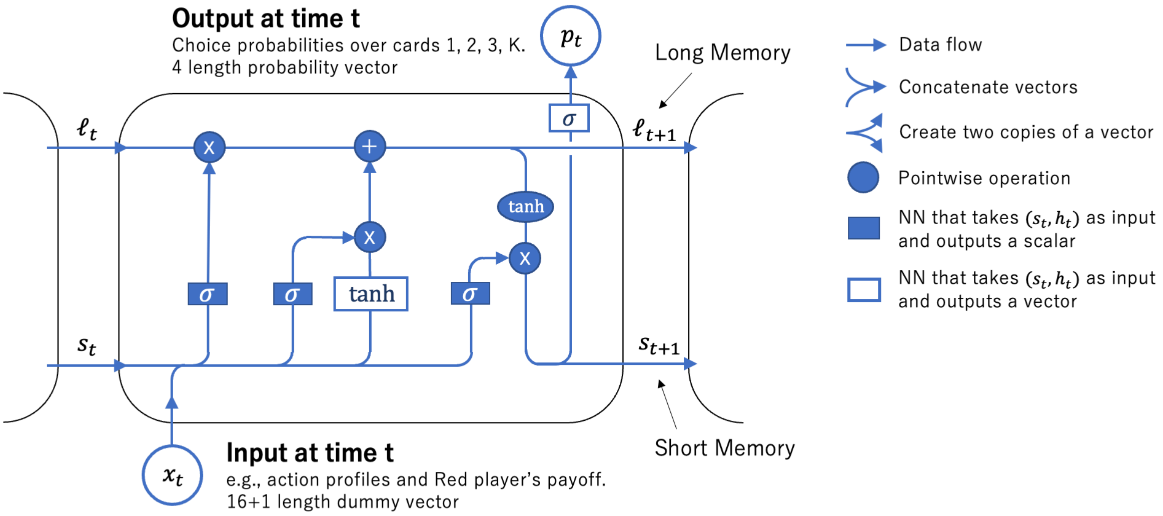

Output at time t
Choice probabilities over cards 1, 2, 3, K.
4 length probability vector
p_t
Long Memory
ℓ_t
ℓ_{t+1}
s_t
s_{t+1}
X
+
X
X
tanh
tanh
σ
σ
σ
σ
σ
Short Memory
x_t
Input at time t
e.g., action profiles and Red player's payoff.
16+1 length dummy vector
Data flow
Concatenate vectors
Create two copies of a vector
Pointwise operation
NN that takes (s_t, h_t) as input and outputs a scalar
NN that takes (s_t, h_t) as input and outputs a vector

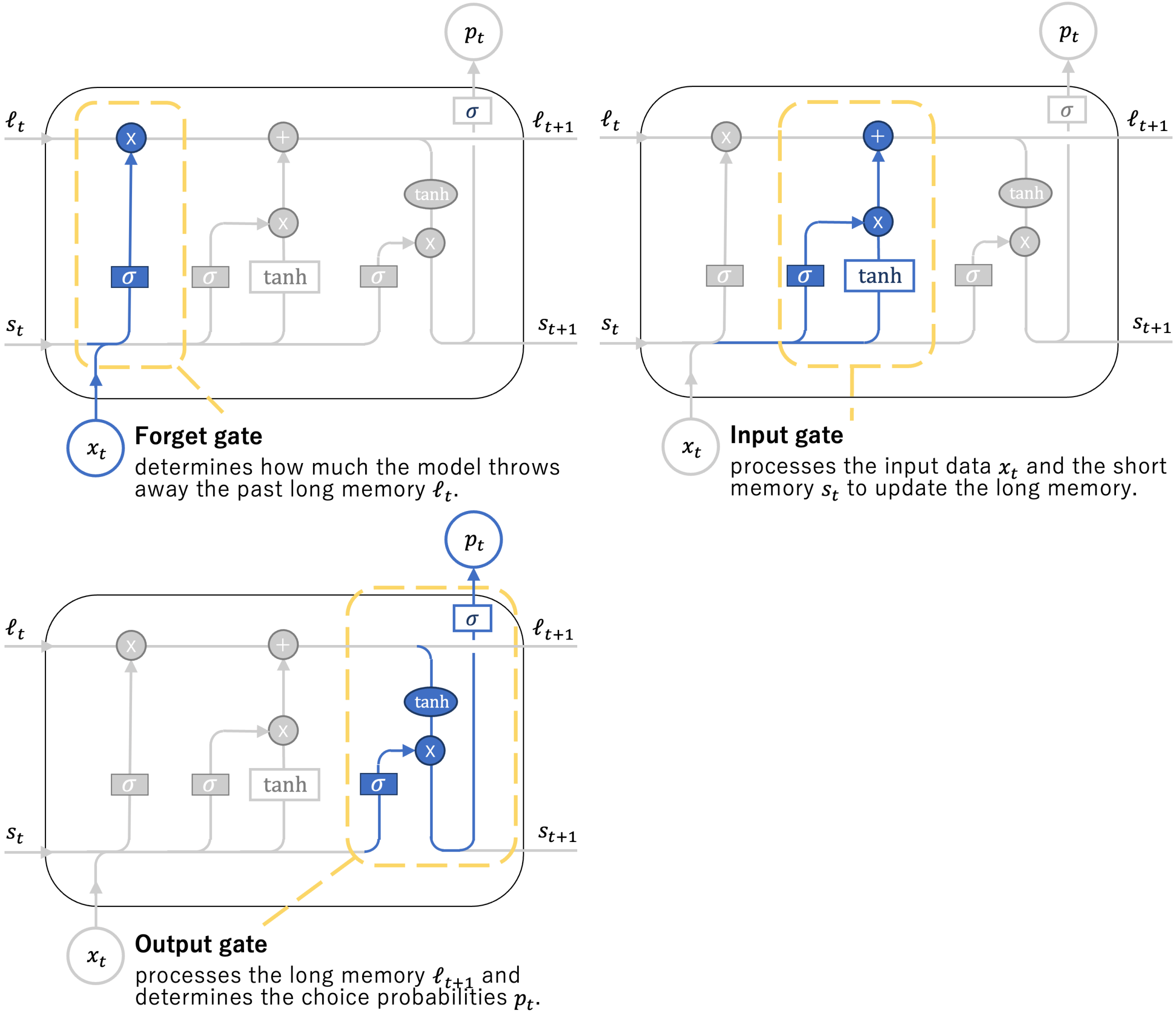
Figure 14. An unfolded illustration of the structure of our LSTM model architecture based on Gers, Schmidhuber and Cummins (2000). A repeating module at time t (specified by a rounded rectangle) takes two state vectors (a long memory and a short memory) and a new data vector as inputs, and outputs new state variables (a new long memory and short memory) and estimated choice probabilities at time t. A detailed explanation of a module is given in Figure 15. This figure is drawn by the authors based on Olah (2015).

We use the Keras TensorFlow implementation of LSTM, which is a standard network introduced in Gers, Schmidhuber and Cummins (2000). Our LSTM model is illustrated in Figure 14. This model includes two state vectors, short memory s_t and long memory ℓ_t, both of which are numerical vectors of dimension L. Here L is a hyperparameter of the model and is determined by the cross-validation procedure.

In each period t, a repeating module (specified by a rounded rectangle in Figure 14) takes as input a dummy vector of an action profile and the red player's payoff in period $t-1$, along with short memory s_t and long memory ℓ_t. Then it updates both memories as s_{t+1} and ℓ_{t+1}, and then outputs the choice probability at time t, p_t.

The detailed structure of a module is illustrated in Figure 15. It consists of three different sections: a **forget gate**, an **input gate**, and an **output gate**. The forget gate is a filter that determines how much it transmits the previous long memory based on the short memory and the new input vector. Formally, a three-layer network (consisting of an input layer, one hidden layer, and an output layer) takes as input (s_t, x_t) and outputs a single scalar in $[0, 1]$ with a softmax activation function. That is,

$$f_\sigma(W_{\text{forgetf}} \cdot (s_t, x_t)^{\mathrm{T}})$$



Forget gate
determines how much the model throws away the past long memory ℓ_t.

Input gate
processes the input data x_t and the short memory s_t to update the long memory.

Output gate
processes the long memory ℓ_{t+1} and determines the choice probabilities p_t.

Figure 15. The detailed structure of our LSTM model. A module of each period t consists of three different sections; a forget gate, an input gate, and an output gate. The forget gate is a filter that determines how the module transmits the previous long memory based on the short memory and the new input vector. The input gate processes the short memory and the new input vector to update the long memory. Finally, the output gate generates the short memory from the updated long memory and outputs the choice probabilities of this period. The figures are drawn by the authors based on Olah (2015).

where W_{forgetf} is a $1 \times (L + 17)$ weight matrix and $f_\sigma(v) = \frac{e^v}{1+e^v}$. Then the previous long memory is multiplied by this value. If the filter value is 1, it fully transfers the long memory. If the value is 0, it completely discards the long memory of the past.

Next, the input gate processes the short memory and the new input vector to update the long memory. A three-layer network takes as input (s_t, x_t) and outputs an L-dimensional new memory with activation function $\tanh(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}} \in [-1, 1]$, whose graph is an upward-sloping S-shaped curve. Another three-layer network generates a filter that takes as input (s_t, x_t) and outputs a scalar in $[0, 1]$ with a softmax activation function, and the former vector is multiplied by the latter scalar:

$$f_\sigma(W_{\text{inputf}} \cdot (s_t, x_t)^{\text{T}}) \times f_{\tanh}(W_{\text{input}} \cdot (s_t, x_t)^{\text{T}})$$

where W_{input} and W_{inputf} are $L \times (L + 17)$ and $1 \times (L + 17)$ weight matrices, and $f_{\tanh}((v_1, \ldots, v_L)) = (\tanh(v_1), \ldots, \tanh(v_L))$. The updated long memory ℓ_{t+1} is

$$\ell_{t+1} = f_\sigma(W_{\text{forgetf}} \cdot (s_t, x_t)^{\text{T}}) \times \ell_t + f_\sigma(W_{\text{inputf}} \cdot (s_t, x_t)^{\text{T}}) \times f_{\tanh}(W_{\text{input}} \cdot (s_t, x_t)^{\text{T}}).$$

Finally, the output gate generates the short memory from the updated long memory and outputs the choice probabilities for this period. Here, we also use a three-layer network with a softmax activation function as a filter. The new short memory is generated as

$$s_{t+1} = f_\sigma(W_{\text{outputf}} \cdot (s_t, x_t)^{\text{T}}) \times f_{\tanh}(\ell_{t+1}),$$

where W_{outputf} is a $1 \times (L + 17)$ weight matrix, and then the attractors are

$$(A_i^1(t), A_i^2(t), A_i^3(t), A_i^K(t)) = f_\sigma(W_{\text{output}} \cdot s_{t+1}),$$

where W_{output} is a $1 \times (L+17)$ weight matrix. The softmax function is applied element-wise. The final choice probabilities are as follows.

$$P(a_i^t = a \mid h^t) = \frac{A_i^a(t)}{\sum_{c \in C} A_i^c(t)}.$$

Our loss function is the Kullback-Leibler divergence (cross-categorical entropy) between the estimated choice probability vector and the 0-1 indicator vector. To avoid overfitting, we use early stopping and dropouts when training the model. As a result of cross-validation, we adopt $L = 10$ to 40 from the candidates $\{5, 10, 15, \ldots, 50\}$. We trained our model using the Adam optimizer on Google Colaboratory. On average, it takes 3.4 minutes to train one model in each train-test split.

6 Modified Economic Models

6.1 Modified EWA

Our next task is to incorporate what is captured by the machine learning models into the leading behavioral economics model EWA. The deep learning models DNN and LSTM, however, are complete "black boxes". This is because the parameters of those models are the weights attached to the branches in their network structure, so the meaning of those parameters is not immediately clear. Therefore, we focus on the lessons we can learn from the more "explainable" machine learning models, the decision tree and LASSO. The decision tree model shows that important factors in determining current action choice are (1) whether a certain action was consecutively played (or not played) in the last two periods, and (2) the choice of the previous action (Figure 11 and 12). Those factors also appear in the list of the variables commonly selected in the five cross-validations of the LASSO model (Table 8). Given those observations, we add three terms (A), (B), and (C) into the attractions of the original EWA as in (11).

$$\begin{aligned} P_i^j(t) \propto \exp\Bigg\{ \lambda \times \Bigg(& A_i^j(t-1) + \underbrace{\gamma_i^j 1\{a_i(t-1)=j\}}_{\text{(A)}} + \\ & \underbrace{\beta_i^j 1\{a_i(t-1)=a_i(t-2)=j\}}_{\text{(B)}} + \underbrace{\alpha_i^j 1\{a_i(t-1)\neq j \text{ and } a_i(t-2)\neq j\}}_{\text{(C)}} \Bigg) \Bigg\}. \end{aligned} \tag{11}$$

(A) reflects the fact that agents tend to avoid the number cards that they chose in the previous period. (B) captures the effect of a certain action being consecutively played twice in a row. (C) captures the effect of a certain action that was not played in the last two periods.

We estimate parameters in five different specifications. Modified EWA (1) to (3) only include the term (A) with different parameter restrictions. Modified EWA (4) and (5) include all terms (A), (B), and (C).

Modified EWA (1) is the formulation under the assumption that all gammas are the same: $\gamma_i^j = \gamma_i$ for all $j \in C$. Modified EWA (2) allows the gammas to differ between the numbers and K but uses the same gamma for each of the numbers. Modified EWA (3) adopts the fully flexible form of gamma. Specification (1), (2), and (3) have 9, 10, and 12 parameters to be estimated for each player $i = \mathrm{R}, \mathrm{B}$.

Modified EWA (4) and (5) incorporate all terms (A), (B), and (C). Modified EWA (5) imposes no further restrictions. Modified EWA (4) is similar to Modified EWA (2) in the sense that we impose restrictions saying that number cards are treated equally. That is,

- $\gamma_i^1 = \gamma_i^2 = \gamma_i^3$
- $\beta_i^1 = \beta_i^2 = \beta_i^3$
- $\alpha_i^1 = \alpha_i^2 = \alpha_i^3$.

We have 16 parameters in total (for each player $i = \mathrm{R}, \mathrm{B}$) with the specification (4). Modified EWA (5) estimates all 20 parameters separately.

6.1.1 Estimates of the parameters

We estimate the parameters using all the sample data using maximum likelihood estimation. Table 10 presents the in-sample performance comparison among the original EWA and the five modified EWAs. We see that specifications (4) and (5) exhibit particularly high performance compared to the original.

Table 11 and 12 present the point estimates of the parameters and the standard errors. We show the estimates of the new parameters (γ, β, α) only because the other parameters are not very different from those of the original EWA model. In Table 11, we find that all the

Table 10. In-sample Performance Comparison of Estimated Modified EWA Models

Models	Logit					
	EWA	(1)	(2)	(3)	(4)	(5)
#Pairs	2577	2577	2577	2577	2577	2577
#Observations	77310	77310	77310	77310	77310	77310
#Parameters	16	18	20	24	32	40
Log Likelihood	-208930.0	-207885.0	-207710.7	-207712.4	-206879.2	-206867.0
LR Stat. (EWA)		2090.0***	2438.7***	2435.2***	4101.7***	4126.1***

Notes: Performance comparison among estimated modified EWA models. We estimate all the models by maximum likelihood estimation using all sample data. The estimated parameters are shown in Table 11, 12. The LR Stat. represents the likelihood ratio test statistic that compares each model to the original EWA. *p<0.1; **p<0.05; ***p<0.01.

estimates of gamma are negative, which implies that subjects tend to avoid their previous action. This effect is stronger for the number cards than for the king. We can also see this feature in the transition matrix in Table 4. The significance of this feature is reflected in the fact that the difference between the log-likelihood of Modified (1) and those of Modified (2) and (3) is relatively large, compared to the small difference between those of (2) and (3).

Table 11. Estimated Parameters of Modified EWA (1), (2), and (3)

Models	Logit		
	Modified (1)	Modified (2)	Modified (3)
Avoid previous actions			
γ_{R}^{1}	-0.514	-0.670	-0.645
	(0.062)	(0.060)	(0.054)
γ_{R}^{2}	$\underline{=\gamma_{\mathrm{R}}^{1}}$	$\underline{=\gamma_{\mathrm{R}}^{1}}$	-0.666
			(0.056)
γ_{R}^{3}	$\underline{=\gamma_{\mathrm{R}}^{1}}$	$\underline{=\gamma_{\mathrm{R}}^{1}}$	-0.652
			(0.055)
γ_{R}^{K}	$\underline{=\gamma_{\mathrm{R}}^{1}}$	-0.200	-0.191
		(0.018)	(0.016)
γ_{B}^{1}	-0.160	-0.310	-0.436
	(0.006)	(0.010)	(0.012)
γ_{B}^{2}	$\underline{=\gamma_{\mathrm{B}}^{1}}$	$\underline{=\gamma_{\mathrm{B}}^{1}}$	-0.398
			(0.011)
γ_{B}^{3}	$\underline{=\gamma_{\mathrm{B}}^{1}}$	$\underline{=\gamma_{\mathrm{B}}^{1}}$	-0.401
			(0.011)
γ_{B}^{K}	$\underline{=\gamma_{\mathrm{B}}^{1}}$	-0.077	-0.096
		(0.003)	(0.003)

Notes: Estimates of the parameters of the modified EWA model (1), (2), and (3) by maximum likelihood estimation. Standard errors are in parentheses. We show only the estimates of the new parameters (γ_i) in the table because the other estimates are not very different from the original EWA model. Underlined values are determined by the model restrictions and are not estimated.

We also observe the avoidance of previous actions in modified EWA (4) and (5) in Table 12. In addition, the estimates of α show that the players tend to choose a card more frequently when it was not played in the last two periods.

We need to be more careful when we interpret the coefficients of β, which measure the effect of playing the same card twice in a row. Although all the coefficients of β are positive, it does not mean that a player tends to play a certain card with a higher probability than usual when she has played it consecutively in the last two periods. If a player played 1 twice in a row, for example, then it means she did not play the other cards 2, 3, and K in the last two periods. Therefore, the attraction of 2, 3, and K increase by α_i^2, α_i^3, and α_i^K at the same time the attraction of 1 increases by $\gamma_i^1 + \beta_i^1$. Thus, in Table 13, we calculate the overall effects on Modified EWA (4) attractions when a player has played a certain card (1) in the previous period or (2) twice in a row. We chose to show the effects in EWA (4) because it is the best one out of the modified EWA models in our performance comparison in Section 7. The effects in the table are computed in the following way. For example, if a red player played card K in the last period (case (1) in the table), then the log odds between playing K and 1, $\log \frac{P^K}{P^1}$, changes by -0.030 points, which is equal to $\lambda_{\mathrm{R}}\gamma_{\mathrm{R}}^K$. If a red player played card K consecutively in the last two periods (case (2) in the table), in contrast, the log odds between playing card K and card 1 changes by -0.112 points, which is equal to the overall effect $\lambda_{\mathrm{R}}(\gamma_{\mathrm{R}}^K + \beta_{\mathrm{R}}^K - \alpha_{\mathrm{R}}^1)$. The overall effect is negative, even though the value of β_{R}^K is positive. Case (2) in Table 13 shows that the same is also true for all cards. If a certain card is played twice in a row, the choice probability of that card decreases, even though the estimated values of betas are all positive.

We observe that when a player has played K consecutively in the last two periods, she will play K less frequently compared to when she has only played K once in a row (i.e., only in the previous period). This is in line with our intuition that a naive player would avoid choosing the same card repeatedly. In contrast, when a player has played 1 (or 2, 3) twice in a row, her choice probability of the same card decreases, but the magnitude of this negative effect is smaller compared to the case of playing it only once in a row. Although this effect is somewhat puzzling, it is also observed in the LASSO estimates in Table 8.

6.2 Discussion on the additional terms

Literature on psychology has shown that people are poor at generating an i.i.d. random sequence (c.f., Bar-Hillel and Wagenaar, 1991). The common finding is that people tend to generate a sequence with more alternations and fewer repetitions than in an i.i.d. sequence. This kind of sequence matches a naive decision-maker's expectations under the gambler's fallacy, which maintains that if a certain outcome occurred (or did not occur) in the previous period, the outcome is less (or more) likely to occur in the current period. This is an implication of the law of small numbers (Rabin, 2002), reflecting a naive decision-maker's

Table 12. Estimated Parameters of Modified EWA (4) and (5)

Models	Logit			
	Modified (4)		Modified (5)	
	Value	Std	Value	Std
Avoid previous actions				
γ^1_{R}	−0.426	0.097	−0.503	0.132
γ^2_{R}	$\underline{= \gamma^1_{\mathrm{R}}}$		−0.403	0.106
γ^3_{R}	$\underline{= \gamma^1_{\mathrm{R}}}$		−0.375	0.099
γ^K_{R}	−0.051	0.012	−0.049	0.014
γ^1_{B}	−0.117	0.007	−0.140	0.008
γ^2_{B}	$\underline{= \gamma^1_{\mathrm{B}}}$		−0.124	0.007
γ^3_{B}	$\underline{= \gamma^1_{\mathrm{B}}}$		−0.088	0.005
γ^K_{B}	−0.041	0.003	−0.042	0.003
Play the same card twice in a row				
β^1_{R}	0.649	0.147	0.687	0.180
β^2_{R}	$\underline{= \beta^1_{\mathrm{R}}}$		0.683	0.178
β^3_{R}	$\underline{= \beta^1_{\mathrm{R}}}$		0.586	0.154
β^K_{R}	0.371	0.084	0.374	0.098
β^1_{B}	0.151	0.008	0.183	0.010
β^2_{B}	$\underline{= \beta^1_{\mathrm{B}}}$		0.152	0.009
β^3_{B}	$\underline{= \beta^1_{\mathrm{B}}}$		0.114	0.007
β^K_{B}	0.121	0.007	0.124	0.007
Did not play the same card in the last two periods				
α^1_{R}	0.507	0.114	0.447	0.117
α^2_{R}	$\underline{= \alpha^1_{\mathrm{R}}}$		0.568	0.147
α^3_{R}	$\underline{= \alpha^1_{\mathrm{R}}}$		0.543	0.141
α^K_{R}	0.444	0.101	0.452	0.119
α^1_{B}	0.139	0.007	0.147	0.007
α^2_{B}	$\underline{= \alpha^1_{\mathrm{B}}}$		0.116	0.006
α^3_{B}	$\underline{= \alpha^1_{\mathrm{B}}}$		0.162	0.008
α^K_{B}	0.080	0.005	0.082	0.005

Notes: Point estimates and standard errors of the parameters of the modified EWA models using maximum likelihood estimation. We show only the estimates of the new parameters $\gamma_i, \beta_i, \alpha_i$ in the table because the other estimates are not very different from the original EWA model. Underlined values are determined by the model restrictions and are not estimated.

Table 13. Overall Effects of Modified EWA Models

		(1) $a(t-1)=j$				(2) $a(t-1)=a(t-2)=j$			
		$\log\frac{P^j}{P^1}$	$\log\frac{P^j}{P^2}$	$\log\frac{P^j}{P^3}$	$\log\frac{P^j}{P^K}$	$\log\frac{P^j}{P^1}$	$\log\frac{P^j}{P^2}$	$\log\frac{P^j}{P^3}$	$\log\frac{P^j}{P^K}$
Red	$j=1$		-0.255	-0.255	-0.255		-0.170	-0.170	-0.132
	$j=K$	-0.030	-0.030	-0.030		-0.112	-0.112	-0.112	
Black	$j=1$		-0.260	-0.260	-0.260		-0.234	-0.234	-0.103
	$j=K$	-0.092	-0.092	-0.092		-0.132	-0.132	-0.132	

Notes: Overall effects of the (A), (B) and (C) terms in Modified EWA (4) when a player plays a certain card (1) in the previous period or (2) twice in a row. For example, when a red player has played card K in the previous period, then the log odds between playing 1 and K decreases by 0.030 points, which is equal to γ_{R}^{K}. Another example is when a red player has played card 1 consecutively in the last two periods, the log odds between playing card 1 and card 2 decreases by 0.170 points, which is equal to $\lambda_{\mathrm{R}}(\gamma_{\mathrm{R}}^{1}+\beta_{\mathrm{R}}^{1}-\alpha_{\mathrm{R}}^{2})$.

tendency to expect that the frequencies of outcomes in a short i.i.d. sequence would match the probability distribution of the outcomes. Fudenberg et al. (2021) compares the models of Rabin (2002) and Rabin and Vayanos (2010) and non-parametric estimation models and shows that the latter fare better.

Similar tendencies are also observed in the strategic situation, i.e., in the setting of a repeated two-person zero-sum game with a unique mixed strategy equilibrium (Brown and Rosenthal, 1990; Rapoport and Boebel, 1992; Budescu and Rapoport, 1994). These papers estimate logit models as Equation (1) with indicator variables for each player's choices up to the past two periods, and show that certain outcomes in the past affect the choice probability of current action. In Brown and Rosenthal (1990) and Rapoport and Boebel (1992), both one's own past choices and the opponent's choices influence players' current choices. In contrast, Budescu and Rapoport (1994) shows that the dependence on the other player's past action is largely insignificant. Our modified EWA models (4) and (5) can be regarded as more elaborate versions of those observations, which incorporate the effects of a certain action chosen (or not chosen) twice in a row. Such variables are not included in those papers.

7 Performance Comparison

All of our models predict the probability distribution of the current action in each period for each player. In the terminology of machine learning, our prediction comparison is categorized as supervised learning, and each model f is a function that assigns to each set of features X of the past history of play a vector of probabilities $\hat{P}=(\hat{P}(1),\hat{P}(2),\hat{P}(3),\hat{P}(K))$, where each $\hat{P}(c)\geq 0$ is a predicted probability of choosing the action $c\in C$ and $\sum_{c\in C}\hat{P}(c)=1$. We conduct performance comparison in terms of external validities, in the sense that we compare the predictive powers of the models in the data set that is *not* used for parameter estimation.

7.1 Performance Measures

We evaluated the predictive powers of the models using loss functions. The value of a loss function represents the accuracy of the prediction. The smaller the loss, the better the prediction. We adopt four measures in total: L1 norm, L2 norm, Kullback-Leibler (KL) divergence, and our own novel **strategic error rate**.

Given a set $\{(X_n, P_n)\}_{n=1}^{N}$ of N sets of features and degenerate probability distributions that indicate which action was actually selected [*13], we calculate the average loss L of a trained model f as

$$L = \frac{1}{N}\sum_{n=1}^{N} l(\hat{P}_n, P_n), \tag{12}$$

where $\hat{P}_n = f(X_n)$ and l is a given loss function. Each loss function is defined as:

$$\begin{aligned} \text{L1} &= \sum_{c\in C} |\hat{P}_n(c) - P_n(c)|, && (13)\\ \text{L2} &= \sqrt{\sum_{c\in C}(\hat{P}_n(c) - P_n(c))^2}, && (14)\\ \text{KL divergence} &= \sum_{c\in C} P_n(c)\cdot \log\left(\frac{P_n(c)}{\hat{P}_n(c)}\right), \quad \text{and} && (15)\\ \text{Strategic Error Rate} &= 1 - \pi_{-i}(\text{BR}_{-i}(\hat{P}_n), P_n), && (16) \end{aligned}$$

where $\text{BR}_{-i}(\hat{P}_n)$ is a best response of the opponent player against the predicted mixed actions $\hat{P}_n$ of the focal player i,

$$\text{BR}_{-i}(\hat{P}_n) \in \arg\max_{c_{-i}\in C} \sum_{c_i\in C} \hat{P}_n(c_i)\cdot \pi_{-i}(c_i, c_{-i}),$$

and $\pi_{-i}(\text{BR}_{-i}(\hat{P}_n), P)$ is the potential payoff to the opponent player if she played a best response to the opponent's mixed action that is predicted by the model. It is equal to the hypothetical probability of winning because the payoff is 1 if the player wins and otherwise 0. Note that for a generic choice of $\hat{P}_n$, there is a unique best reply, and this was indeed the case for our estimated models.

The L1, L2, and the KL divergence [*14] are standard ways of measuring prediction errors,

[*13] $P(c) = 1$ if c is chosen, and $P(c) = 0$ otherwise.

[*14] Here the KL divergence is equal to the negative log-likelihood: if the agent selects action $c \in C$, then

$$\begin{aligned} \text{KL divergence} &= 1 \times \log\left(\frac{1}{\hat{P}(c)}\right) + \sum_{c'\neq c} 0 \times \log\left(\frac{0}{\hat{P}(c')}\right)\\ &= -\log \hat{P}(c)\\ &= -\sum_{c'\in C} P(c')\log(\hat{P}(c')) = \text{Log-likelihood}. \end{aligned}$$

but the values of these measures are somewhat difficult to understand. This is because what is being measured is the “distance” between a probability distribution of action (prediction of a model) and actual action taken (degenerated probability distribution), and therefore even if the model accurately predicts the choice probability, the distance is not equal to zero. For that reason, when we are told that the KL divergence of a model is 1.36, we do not have a feel for the accuracy of the model's prediction. Note also that this difficulty does not arise if the prediction and data are both scalars. For example, if models predict the dollar income of subjects and the error rate is .92, we can understand that the prediction error is on average 92 percents. In contrast, the meaning of the value of KL divergence is not immediately clear. To address that problem, we introduce a novel performance measure, the *strategic error rate*. It is defined as the hypothetical loss probability when the opponent of the focal player chooses the best response to the estimated choice probabilities of the focal player. The meaning of this measure is illustrated as follows. In the mixed strategy equilibrium, the black player's loss rate is 0.4, and if model X of the red player has a strategic error rate 0.38, it means that when the black player utilizes this model to predict the red player's behavior in any given single period and makes the best reply, on average (across all single periods in the sample), she can reduce the loss rate by 0.02 relative to the mixed strategy equilibrium loss rate. We stress that the strategic error rate concerns the usefulness of the estimated model in any given *single* period. In principle, if the back player in the previous story makes the best reply to the estimated model of the red player in one period, this has two effects: (i) the loss rate in the current period changes, and (ii) the future behavior of the red player changes, which affects the future loss rates of the black player. Our notion of strategic error rate only measures the former effect.

As we mentioned in Section 5.1, we compare the model performance using leave-one-out cross-validation. We first randomly divided the entire 2577 pairs in our data into five subgroups $D_1, \ldots, D_5$. Then, for each $k = 1, \ldots, 5$, we trained (estimated) each model using its training data $\cup_{\ell \neq k} D_\ell$ and computed their out-of-sample losses using its test data D_k. We call this pair of training data and test data a CV-k split. The final loss score of a model is the average of the losses in all five CV splits.

In particular, we compute the average loss of the trained model f_k for the red player in the CV-k split as follows. For each pair in its test data $s \in D_k$,

- We first estimate the choice probability at period 5 using the initial four period history as features, $\hat{P}^5_{\mathrm{R}_s} = f_k(h^4_{\mathrm{R}_s})$, and compute the period loss $l(\hat{P}^5_{\mathrm{R}_s}, P^5_{\mathrm{R}_s})$,
- Next we estimate the choice probability at period 6 using all past histories as features, $\hat{P}^6_{\mathrm{R}_s} = f_k(h^5_{\mathrm{R}_s})$, and compute the period loss $l(\hat{P}^6_{\mathrm{R}_s}, P^6_{\mathrm{R}_s})$, and
- We repeat this process until $t = 30$.

Note that we define $0 \times \log(0) = \lim_{x \to +0} x \log x = 0$ as usual in the literature.

Then we take the average of $|D_k|$ pairs × 26 periods. That is,

$$(\text{Average loss in CV-}k) = \frac{1}{|D_k| \times 26} \sum_{s\in D_k} \sum_{t=5}^{30} l(P^t_{i(s)}, f_k(h^{t-1}_s)).$$

We can use at most 26 periods for each pair because LASSO uses a four-period history of action profiles as its explanatory variable.

7.2 Results

Performance comparison results are shown in Table 14 (for the red player models) and Table 15 (for the black player models). Here L1, L2, KL, SER, and RC are the abbreviations for the L1 loss, the L2 loss, the KL-divergence, the strategic error rate, and the relative completeness defined below. We also counted the number of parameters we used in each CV split. Each performance score and the number of parameters are the averages of the five CV splits.

Both Table 14 and Table 15 show that the LSTM model exhibits outstanding performance on all performance measures [*15]. Given that, we measure the relative predictive powers of the models in terms of the error rate reduction from a naive benchmark in the form of an i.i.d. mixture model (MLE const) to the best model LSTM. This leads to the notion of *relative completeness* (RC) defined as

$$(\text{RC of a model}) = \frac{(\text{KL divergence of the model}) \text{ - } (\text{KL divergence of MLE const})}{(\text{KL divergence of LSTM}) \text{ - } (\text{KL divergence of MLE const})},$$

where KL divergence is an average of the five CV splits. This is a version of the completeness measure introduced by Fudenberg et al. (2021). If, in the above definition of RC, we replace LSTM, the best model we have obtained, with the best one among all specifications of the model equation (1), we obtain the completeness measure.

We observe that EWA, a leading learning model in behavioral economics, has relative completeness 0.309 (0.206) for the red (black) player. This means that EWA achieves only 30.9% (20.6%) for the red (black) player of the predictive power of the state-of-the-art LSTM model (in terms of the error rate reduction from the naive benchmark i.i.d. mixture model (the MLE const). This result suggests that the EWA model fails to capture some important aspects of actual human behavior in this game. The strategic error rates, which measure predictive power in an intuitive way, of LSTM, EWA and the naive benchmark i.i.d. mixture model ("MLE const" in Table 14) for the red player are presented in Figure 4 in the introduction.

[*15] However, traditional econometric criteria of model selection, such as AIC and BIC in the training sample, do not select LSTM due to its large parameter size. When we measure the performance of singular models like DNN and LSTM, these traditional information criteria are known to be not appropriate in the machine learning literature. We list the average AIC and BIC of the representative models in Appendix B.3.

Table 14. Prediction Peformance Comparison (Red Players)

	Red Players					
	L1	L2	KL	SER	#Params	RC
Simple Logit						
MLE Constant (Baseline)	1.473	0.855	1.361	0.348	4	0.000
MLE 1 profile	1.467	0.853	1.355	0.347	64	0.246
MLE 2 profile	1.460	0.852	1.355	0.344	1024	0.219
MLE 1 K-profile	1.472	0.855	1.360	0.348	16	0.010
MLE 2 K-profile	1.470	0.853	1.359	0.346	64	0.083
MLE 3 K-profile	1.466	0.852	1.357	0.337	256	0.142
MLE 4 K-profile	1.462	0.851	1.359	0.336	1024	0.063
EWA-variants						
RL	1.469	0.853	1.357	0.337	5	0.135
BL	1.473	0.854	1.359	0.348	5	0.056
EWA	1.464	0.850	1.353	0.330	8	0.310
Nested EWA	1.462	0.849	1.352	0.328	9	0.346
Modified EWA (1)	1.457	0.848	1.346	0.326	9	0.580
Modified EWA (2)	1.456	0.848	1.345	0.324	10	0.623
Modified EWA (3)	1.456	0.848	1.345	0.324	12	0.623
Modified EWA (4)	1.450	0.845	1.339	0.314	16	0.853
Modified EWA (5)	1.450	0.845	1.339	0.313	20	0.849
Tree Algorithms						
Decision Tree	1.463	0.851	1.351	0.340	14.4	0.372
LASSO						
LASSO	1.455	0.847	1.343	0.324	317.0	0.710
Deep Learning Models						
DNN	1.464	0.851	1.349	0.334	10324.0	0.483
LSTM	1.444	0.842	1.336	0.309	7212.0	1.000
Human				0.419		

Notes: Average prediction performance scores measured by the test data. We first randomly split the data into five disjoint groups. Then for each group, we train a model using the data of the remaining four groups, and test the model's performance with it. Hyperparameters of each model are optimized by cross-validation within the training data. We take the averages of these five train-test splits as the performance scores above. Here L1, L2, KL, SER, and RC are the abbreviations for L1 loss, L2 loss, KL-divergence, strategic error rate, and relative completeness.

Table 15. Prediction Peformance Comparison (Black Players)

	Black Players					
	L1	L2	KL	SER	#Params	RC
Simple Logit						
MLE Constant (Baseline)	1.465	0.852	1.354	0.564	4	0.000
MLE 1 profile	1.458	0.849	1.347	0.540	16	0.308
MLE 2 profile	1.451	0.847	1.346	0.537	1024	0.326
MLE 1 K-profile	1.464	0.851	1.353	0.563	16	0.056
MLE 2 K-profile	1.462	0.850	1.351	0.562	64	0.112
MLE 3 K-profile	1.460	0.849	1.351	0.558	256	0.128
MLE 4 K-profile	1.457	0.849	1.355	0.561	1024	-0.034
EWA-variants						
RL	1.477	0.856	1.365	0.578	5	-0.446
BL	1.465	0.851	1.354	0.559	5	0.017
EWA	1.460	0.849	1.349	0.555	8	0.201
Nested EWA	1.458	0.848	1.347	0.552	9	0.292
Modified EWA (1)	1.453	0.846	1.343	0.529	9	0.472
Modified EWA (2)	1.453	0.846	1.342	0.526	10	0.513
Modified EWA (3)	1.453	0.846	1.342	0.526	12	0.513
Modified EWA (4)	1.447	0.844	1.337	0.520	16	0.714
Modified EWA (5)	1.447	0.844	1.337	0.520	20	0.714
Tree Algorithms						
Decision Tree	1.456	0.848	1.345	0.534	12.8	0.358
LASSO						
LASSO	1.448	0.845	1.337	0.526	348	0.704
Deep Learning Models						
DNN	1.456	0.848	1.342	0.533	10544	0.515
LSTM	1.438	0.840	1.330	0.508	7908	1.000
Human				0.581		

Notes: Average prediction performance scores measured by the test data. We first randomly split the data into five disjoint groups. Then for each group, we train a model using the data of the remaining four groups, and test the model's performance with it. Hyperparameters of each model are optimized by cross-validation within the training data. We take the averages of these five train-test splits as the performance scores above. Here L1, L2, KL, SER, and RC are the abbreviations for L1 loss, L2 loss, KL-divergence, strategic error rate, and relative completeness.

Note that the non-parametric model that depends on two-period histories of action profiles (MLE 2-profile) fares much worse (RC is 21.9% for the red player) than EWA (for the red player) and the machine learning models. Given that we have a sufficiently large data set for non-parametric estimation (see Figure 8), this is a reliable indication that subjects have a longer memory than two periods.

Finally, we find that LASSO performs relatively well, with performance being 71.0% (70.4%) of the LSTM performance for the red (black) players. In contrast to the best model LSTM, which is a complete black box in the sense that the parameters of the model are hard to interpret, LASSO is easier to interpret. We can look at the explanatory variables commonly selected in the five rounds of cross-validation for the LASSO model. As we explained in Subsection 6.1, from the LASSO model as well as the decision tree model, we deduced that (i) the subjects tend to avoid choosing the same card consecutively, and (ii) the subject's action is influenced by whether a certain action was (or was not) chosen consecutively in the last two periods. By adding those components to the original EWA model (the Modified EWA models in Table 14 and Table 15), we succeeded in improving the performance from 30.9% to 85.3% for the red player (from 20.1% to 71.4% for the black player). This is illuminating because the two building blocks of EWA, belief learning and reinforcement learning, are general-purpose learning rules, but they do not capture the subject's desire to make their own behavior unpredictable. As we elaborated on in Subsection 6.1, the added components can be interpreted as naive attempts to choose actions over time that "look like" a random sequence, and adding those components substantially improves the performance of EWA. Behavioral economics literature emphasizes the importance of identifying the learning rules that are commonly applicable to a wide range of problems, but our result suggests the importance of context-specific learning rules (our context is a situation where one wants to make one's own behavior unpredictable). Lastly, note that there is still a 15% to 29% difference between the modified EWA and the LSTM models, so we still have room for improvement if we are to fully understand the learning mechanisms of the human subjects.

7.3 The Role of Big Data

In this section, we show that we obtained quite a reliable performance comparison of various models thanks to our big data. To substantiate our claim, we show what happens if we artificially reduce the data size for our parameter estimation and performance comparison. This exercise shows that we need a data set which is an order of magnitude larger than the conventional laboratory experiments to detect the superiority of machine learning models.

First, we show that the relative performance of the models is stable across the five rounds of cross-validation (Figure 16).

Next, we show how the figure for the red player changes if we conduct parameter fitting *and* performance comparison in artificially reduced data sets (Figure 17). When we use all

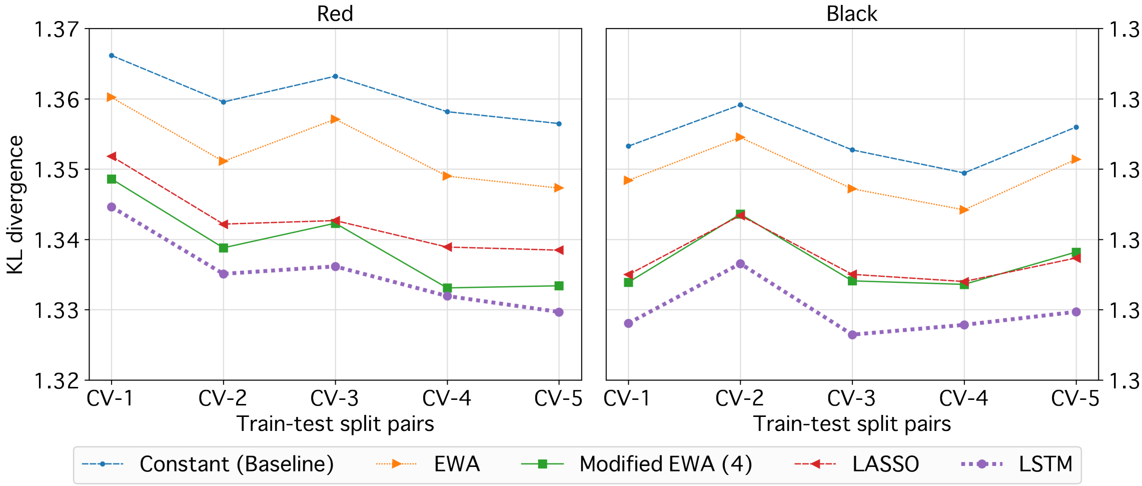


Figure 16. Comparison of KL divergence of five representative models in each CV split (left: red players, right: black players). We observe that LSTM outperforms the other models in all splits. Precise KL divergence numbers are shown in Table 23 and 24 in the Appendix.

of the data with sample size equal to 67,002, the relative performance of models is clearly distinguishable in the sense that (i) there are wide margins in the error rates of different models and (ii) the ranking of the models is stable (unchanged) across the five rounds of cross-validation. If we reduce the sample size to 26,000, the performance of the top two models, LSTM and modified EWA, becomes quite similar. If we further reduce the sample size to 13,000, the ranking of the models differs across the five rounds of cross-validation, and most importantly, the dominance of LSTM disappears. These features are more prominent if the sample size is reduced to 2,600 with 100 pairs, which is comparable to the sample sizes of many in-person laboratory experiments.

Note that reducing the data size has two effects. First, the estimation of parameters becomes less accurate. Second, the comparison of the error rates of the models becomes less reliable. Since Figure 17 captures both effects, we now present those two effects separately.

We first show what happens when we use the full set of data for performance comparison but artificially reduce the training data for parameter fitting (Figure 18), which is probably one of the most important findings we obtained. The figure shows that the performance of the traditional models, Constant, EWA, and Modified EWA, do not improve if we increase the training data size beyond 200 pairs. In contrast, the error rates of the machine learning models, LASSO and LSTM, keep on decreasing as the training data size increases, and when the training data size reaches 1,200, LSTM just overtakes Modified EWA. The figure thus clearly shows that we need at least 1,400 pairs (2,800 players) to see that the machine learning model LSTM substantially dominates the conventional behavioral model. Conventional laboratory experiments usually have at most hundreds of subjects (Figure 5), and our result suggests that we need a subject pool that is an order of magnitude larger than the

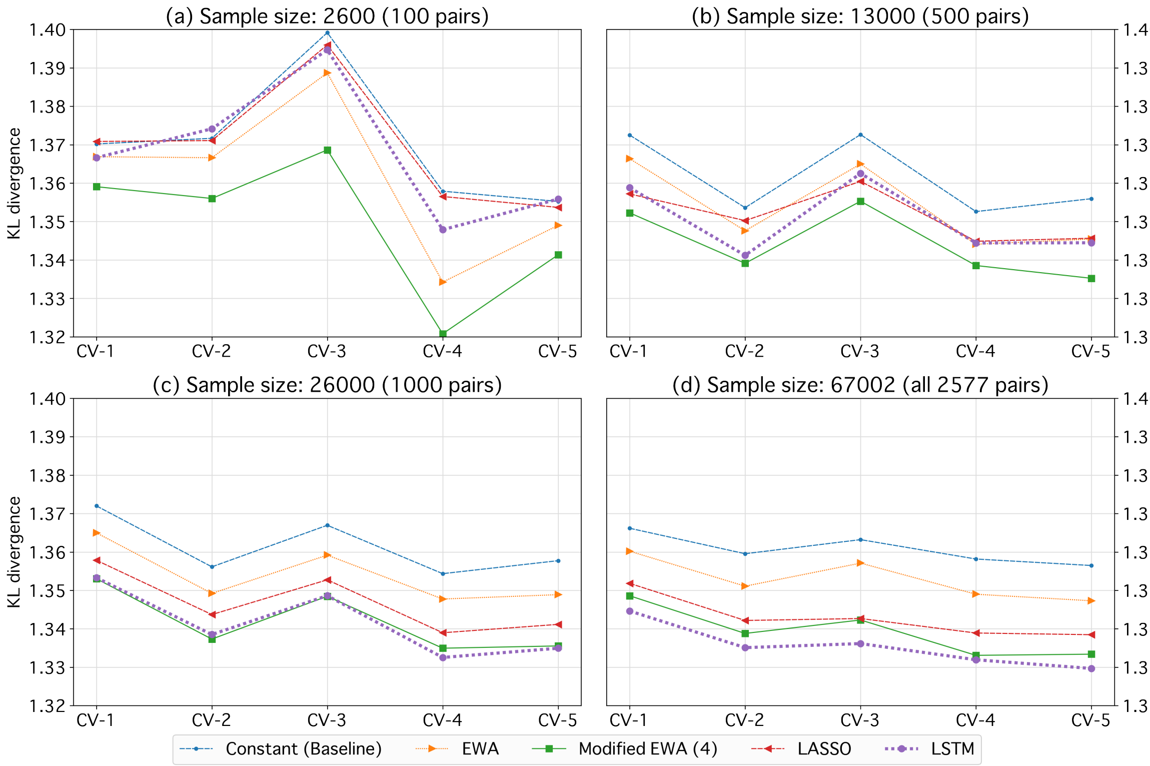


Figure 17. Performance comparison of five representative algorithms for the red player when we artificially reduce **both training data and test data**. In the upper right panel, for example, we randomly picked up 400 (100) pairs from the original training (test) data in each CV and used it as training (test) data. Precise KL divergence numbers are shown in Table 23 in the Appendix.

conventional ones if we want to see the full potential of machine learning models.

Second, we use the full data set for parameter fitting but artificially reduce the test data size for performance comparison (Figure 19). This figure shows that a test data set of 20 pairs (40 players) does not effectively reveal the true predictive powers of the models, but that with more than 100 pairs (200 players) is good enough. In summary, the improved accuracy of performance comparisons as we increase the data size, which we saw in Figure 17, is due to more accurate parameter estimation (especially for the machine learning models), and it is largely *not* due to higher resolution of performance comparison.

8 Conclusion

Our results are summarized in Figure 20, which shows the relative rate of reduction in prediction errors from the naive benchmark i.i.d. mixture model (MLE const), or more precisely, the relative completeness. It shows that the machine learning models outperform the conventional behavioral model (EWA), and most notably a version of deep learning model

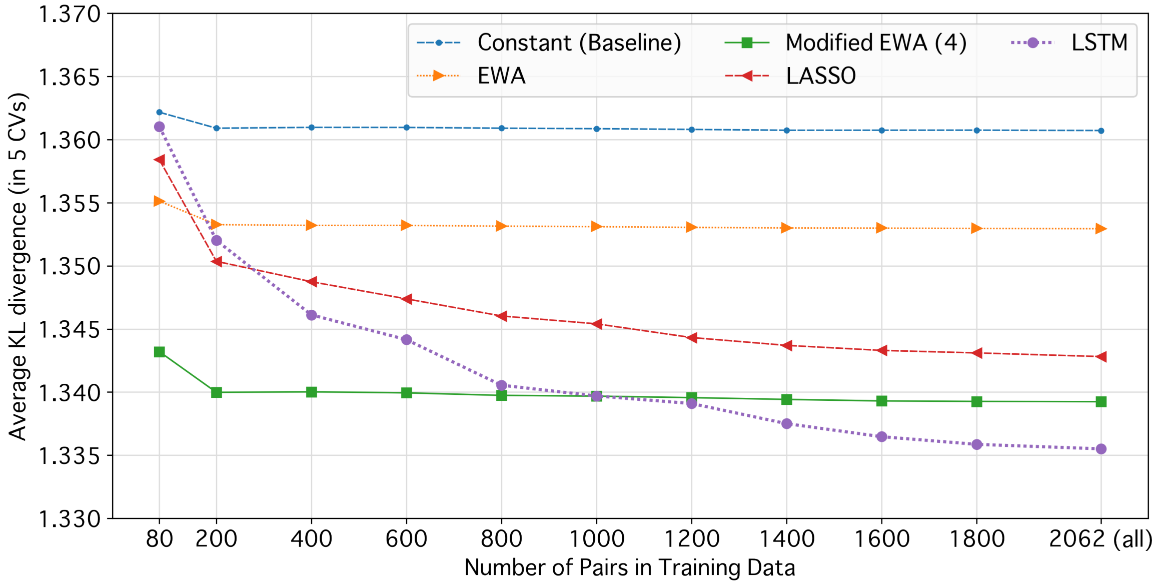


Figure 18. Average performance (in five CVs) of the representative algorithms for the red player when we artificially reduce **training** data. An x-axis value of 800 means that we randomly sampled 800 pairs from the original training data in each CV split, and conducted parameter fitting with the 800 pairs only. We computed the KL divergence figures using the original test data (we do not decrease test data). Detailed numbers are shown in Table 25 in the Appendix.

of LSTM fares substantially better.

This figure also illustrates the "capture and decode" research program explained in the Introduction. The shaded area represents the regularities or mechanisms of the subjects' behavior that are not present in the conventional behavioral model EWA but "captured" by the best machine learning model we consider, LSTM. Those regularities are encoded in the estimated parameters of the LSTM model. Since these parameters are the weights attached to the branches of the network structure of the LSTM model, their role and meaning are not immediately clear. Hence, we need to open the "black boxes" of the machine learning model to understand what the model captures. This is the "decoding" stage, and we tried to gain insights from the more "interpretable" machine learning models, the decision tree, and LASSO. This led us to incorporate certain additional terms into the EWA model, which represent the subjects' naive attempt to create a sequence of actions that "look like" a random sequence. The resulting modified EWA model successfully "decodes" approximately 70% to 85% of what is captured by the best machine learning model, LSTM. The remaining 15-30% of the captured regularities is yet to be decoded or explained, and this poses a challenging future research agenda.

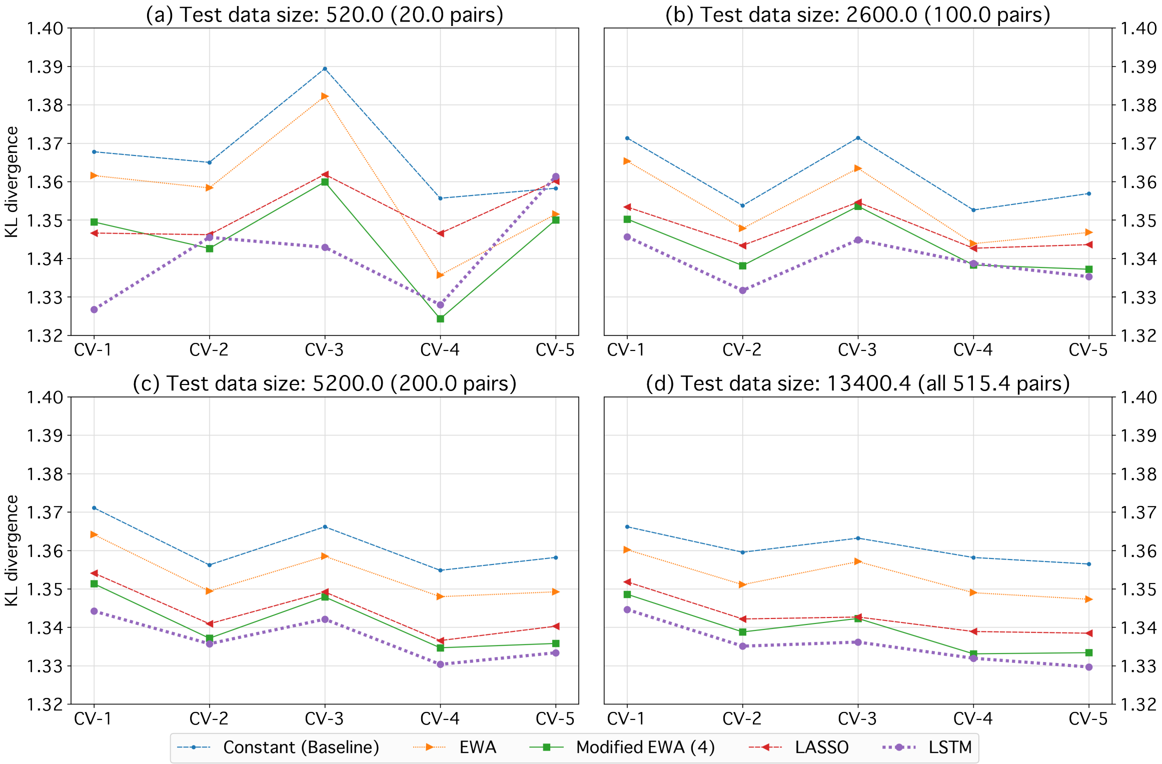


Figure 19. Test data resolution: performance comparison of five representative algorithms for the red player when we artificially reduce **test** data. In the upper right panel, for example, we randomly sampled 100 pairs from the original test data in each CV. We trained the models with the original training data and evaluated performance with smaller test data. Panels (a)-(d) in this figure correspond to those in Figure 17. Precise KL divergence numbers are shown in Table 26 in the Appendix.

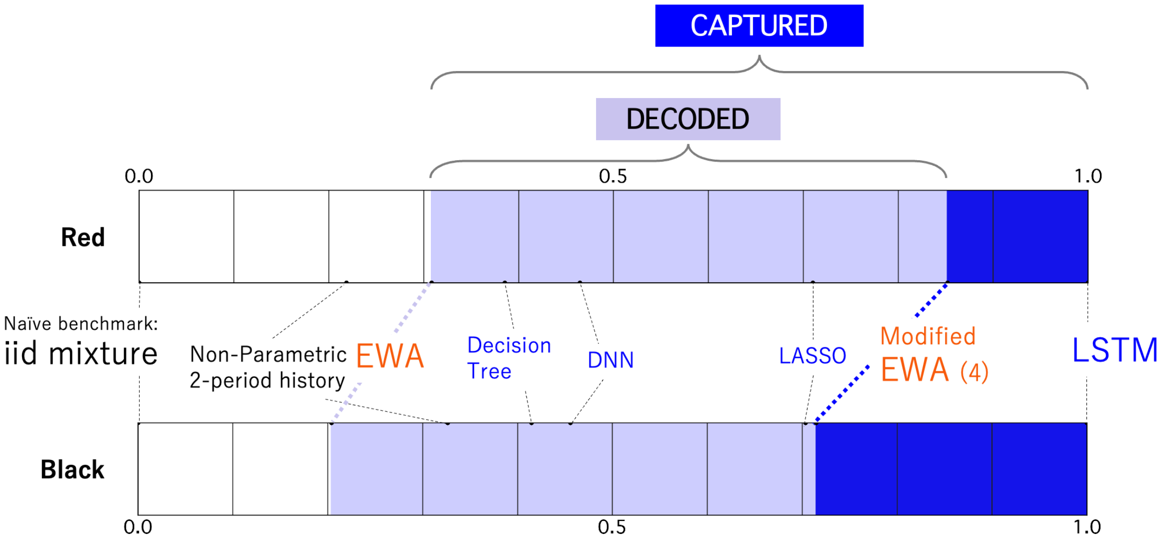


Figure 20. The relative rate of prediction error reduction from the naïve benchmark model (relative completeness) for the red player. Blue text indicates to machine learning models.

References

Adadi, Amina, and Mohammed Berrada. 2018. “Peeking inside the black-box: a survey on explainable artificial intelligence (XAI).” *IEEE access*, 6: 52138–52160.

Athey, Susan, and Guido W Imbens. 2019. “Machine Learning Methods that Economists Should Know About.” *Annual Review of Economics*, 11: 685–725.

Augenblick, Ned, and Matthew Rabin. 2021. “Belief movement, uncertainty reduction, and rational updating.” *The Quarterly Journal of Economics*, 136(2): 933–985.

Bar-Hillel, Maya, and Willem A Wagenaar. 1991. “The perception of randomness.” *Advances in applied mathematics*, 12(4): 428–454.

Brown, George W. 1949. “Some Notes on Computation of Games Solutions.” *Report*, 78.

Brown, James N, and Robert W Rosenthal. 1990. “Testing the minimax hypothesis: A re-examination of O’Neill’s game experiment.” *Econometrica*, 58(5): 1065–1081.

Budescu, David V, and Amnon Rapoport. 1994. “Subjective randomization in one-and two-person games.” *Journal of Behavioral Decision Making*, 7(4): 261–278.

Camerer, Colin, and Teck Hua Ho. 1999. “Experience-weighted attraction learning in normal form games.” *Econometrica*, 67(4): 827–874.

Camerer, Colin F. 2011. *Behavioral game theory: Experiments in strategic interaction.* Princeton university press.

Camerer, Colin F. 2019. “Artificial intelligence and behavioral economics.” *The economics of artificial intelligence: An agenda*, 587–608.

Chiappori, P-A, Steven Levitt, and Timothy Groseclose. 2002. “Testing mixed-strategy equilibria when players are heterogeneous: The case of penalty kicks in soccer.” *American Economic Review*, 92(4): 1138–1151.

Erev, Ido, and Alvin E. Roth. 1998. “Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique, Mixed Strategy Equilibria.” *The American Economic Review*, 88(4): 848–881.

Fudenberg, Drew, Jon Kleinberg, Annie Liang, and Sendhil Mullainathan. 2021. “Measuring the Completeness of Economic Models.” *Journal of Political Economy*, 130(4): 956–990.

Gers, Felix A, Jürgen Schmidhuber, and Fred Cummins. 2000. “Learning to forget: Continual prediction with LSTM.” *Neural computation*, 12(10): 2451–2471.

Harrison, Glenn W, and John A List. 2004. “Field experiments.” *Journal of Economic literature*, 42(4): 1009–1055.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long short-term memory.” *Neural computation*, 9(8): 1735–1780.

Kandori, Michihiro. 2018. “Replicability of experimental data and credibility of economic theory.” *The Japanese Economic Review*, 69(1): 4–25.

Mookherjee, Dilip, and Barry Sopher. 1997. "Learning and decision costs in experimental constant sum games." *Games and Economic Behavior*, 19(1): 97–132.

Mullainathan, Sendhil, and Jann Spiess. 2017. "Machine learning: an applied econometric approach." *Journal of Economic Perspectives*, 31(2): 87–106.

Nunnari, Salvatore, Luca Congiu, and Sandrim Emiliano. 2022. "Database of laboratory experiments in top economics journals." *https://docs.google.com/spreadsheets/d/1434ApdJsvdtRNIpDR_03h-kmUffcL24_cJj7BXyCezs/edit#gid=491277435*.

Olah, Christopher. 2015. "Understanding LSTM Networks." https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

O'Neill, Barry. 1987. "Nonmetric test of the minimax theory of two-person zerosum games." *Proceedings of the National Academy of Sciences*, 84(7): 2106–2109.

Palacios-Huerta, Ignacio. 2003. "Professionals play minimax." *The Review of Economic Studies*, 70(2): 395–415.

Peysakhovich, Alexander, and Jeffrey Naecker. 2017. "Using methods from machine learning to evaluate behavioral models of choice under risk and ambiguity." *Journal of Economic Behavior & Organization*, 133: 373–384.

Rabin, Matthew. 2002. "Inference by believers in the law of small numbers." *The Quarterly Journal of Economics*, 117(3): 775–816.

Rabin, Matthew, and Dimitri Vayanos. 2010. "The gambler's and hot-hand fallacies: Theory and applications." *The Review of Economic Studies*, 77(2): 730–778.

Rapoport, Amnon, and Richard B Boebel. 1992. "Mixed strategies in strictly competitive games: A further test of the minimax hypothesis." *Games and Economic Behavior*, 4(2): 261–283.

Rees-Jones, Alex, and Dmitry Taubinsky. 2020. "Measuring "schmeduling" ." *The Review of Economic Studies*, 87(5): 2399–2438.

Roth, Alvin E, and Ido Erev. 1995. "Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term." *Games and economic behavior*, 8(1): 164–212.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. "ImageNet Large Scale Visual Recognition Challenge." *International Journal of Computer Vision (IJCV)*, 115(3): 211–252.

Tambe, Milind. 2011. *Security and game theory: algorithms, deployed systems, lessons learned.* Cambridge university press.

Train, Kenneth E. 2009. *Discrete choice methods with simulation.* Cambridge university press.

Walker, Mark, and John Wooders. 2001. "Minimax play at Wimbledon." *American Economic Review*, 91(5): 1521–1538.