A HORSE RACE AMONG MODELS OF STRATEGIC THINKING ACROSS SIMILAR GAMES

İBRAHİM EMİRAHMETOĞLU

BOĞAZİÇİ UNIVERSITY

2015

A HORSE RACE AMONG MODELS OF STRATEGIC THINKING ACROSS SIMILAR GAMES

Thesis submitted to the

Institute for Graduate Studies in Social Sciences

in partial fulfillment of the requirements for the degree of

Master of Arts

in

Economics

by

İbrahim Emirahmetoğlu

Boğaziçi University

2015

A Horse Race among Models of Strategic Thinking across Similar Games

The thesis of İbrahim Emirahmetoğlu

has been approved by:

Assist. Prof. Mehmet Yiğit Gürdal ____________________

(Thesis Advisor)

Assist. Prof Sinan Ertemel ____________________

(External Member)

Assist. Prof. Tolga Umut Kuzubaş ____________________

June 2015

DECLARATION OF ORIGINALITY

I, İbrahim Emirahmetoğlu, certify that

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;
- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institutions;
- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature.............................

Date.....................................

ABSTRACT

A Horse Race among Models of Strategic Thinking across Similar Games

Human behavior generally deviates from equilibrium in one-shot games. For this reason, a number of strategic thinking models which relax one or more assumptions of equilibrium have emerged. A natural extension to the emergence of these models is to compare their predictive and explanatory powers. In this study we have made a full-fledged comparison of eight prominent models (QRE, Lk, CH, NI, SLk, SCH, GCH and Lm) through a new game and its variations. We have analysed their performances in two ways. First, we out-of-sample predicted experimental results by these models and compared them by calculating the mean of squared distances between predictions and the observed data. Secondly we estimated the models for all games together and compared their log-likelihood values to determine their performance in explaining subjects' behaviors. We found that models with payoff dependent noise had consistently better predictive performances than those without noisy behavior. Our main contribution is to show that a little modification on game structure might lead to drastically different results in the predictive performances and statistical fits of the models. Even across very similar games, there were significant changes on the performances of the models.

ÖZET

Benzer Oyunlar Üzerinden Stratejik Düşünme Modellerinin Karşılaştırılması

Tek seferlik oyunlarda insan davranışları genellikle Nash dengesinden sapmaktadır. Bu sebepten denge varsayımlarından bir ya da birkaçını gevşeten bazı stratejik düşünme modelleri ortaya çıkmıştır. Bu modellerin doğuşunun doğal bir uzantısı, onların tahmin ve açıklama güçlerini karşılaştırmaktır. Bu makalede tanınmış sekiz modeli (QRE, Lk, CH, NI, SLk, SCH, GCH ve Lm) yeni bir oyun ve onun varyasyonları üzerinden karşılaştırmaktayız. Bu modellerin performanslarını iki şekilde analiz ettik. İlk olarak deney sonuçlarını bu modelleri kullanarak örneklem dışı tahmin ettik ve onları, tahminler ile gözlenen veriler arasındaki mesafelerin karelerinin ortalamasını hesaplayarak karşılaştırdık. İkinci olarak modelleri bütün oyunlar için birlikte tahmin ettik ve onların denek davranışlarını açıklamadaki performanslarını belirlemek için olabilirlik değerlerini karşılaştırdık. Gördük ki ödemeye bağlı hata içeren modeller, içermeyenlere göre istikrarlı bir şekilde daha iyi tahmin performanslarına sahipler. Bizim temel katkımız, oyun yapısındaki küçük değişikliklerin, modellerin tahmin performansları ve istatistiksel uyumlarını ciddi oranda değiştirebildiğini göstermektir. Çok benzer oyunlar arasında bile modellerin performansları önemli ölçüde değişiklik göstermiştir.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Dr. Mehmet Yiğit Gürdal, for his excellent guidance, caring, and patience, and for providing me with an excellent atmosphere for doing research.

I would also like to thank Dr. Türkmen Göksel for his help and support on our experimental design.

Next, I would like to thank Dr. Sinan Ertemel and Dr. Tolga Umut Kuzubaş for their valuable comments. Special thanks go to Dr. Lawrence Choo, who was always willing to help and make valuable suggestions.

I would like to acknowledge my gratitude to Gökçen Cangüven and Mehmet Nazım Tamkoç for their support during this entire process.

Finally, thanks go to my beloved family for their support and understanding under all conditions, especially to a swallow.

CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

For repeated games there is no need to make much effort to explain human strategic behavior. Because players can learn about others' decisions and beliefs and coordinate with them along with these repeated games. This eventually leads to the convergence of players' behaviors to the equilibrium. However, in one-shot games, human behavior generally deviates from the equilibrium. It is for this reason that a number of strategic thinking models which relax one or more assumptions of the equilibrium have emerged. In one-shot interactions, these models are expected to perform better in describing human behavior than the Nash equilibrium.

A natural extension to the emergence of behavioral models is to compare them in terms of their predictive powers or statistical fit. Therefore we wanted to determine which strategic thinking model was best suited to predicting or explaining human behavior in one-shot games. When we conducted a literature survey to see to what extent this question has been answered, we realized that there is no comprehensive comparison of the models. Of the few existing studies, most deal with behavioral models, comparing only one model with another one - generally the Nash equilibrium - and focus on an explanation of a single model in detail. There are a few studies comparing more than two models of strategic thinking. Inspired by this point, we made a full-fledged comparison of the models through a new game and its variations. We saw from previous works that the performance of the models varies from game to game. This observation raises the question of whether there is a superiority of any one model over other behavioral models, regardless of the structure of the game. We designed our experiment to determine if the degree to which the model fits the experimental data depends on the structure and content of the game.

For this study, we compared the models in two ways. First we took out-of-sample predicted human behavior by different models and compared them by calculating the mean of squared distances between predictions and the observed data. Secondly, we estimated the models for all games together and compared their log-likelihood values to determine their performance to explain the subjects' behaviors. As expected, these two comparison methods gave different results for the models because the out-of-sample prediction was employed to choose the most accurate model, while the in-sample fitting was used to determine the most flexible one. Our paper is unique in that it makes both of these comparisons together.

There are two prominent concepts that underlie these models: payoff-dependent noise and bounded iterated reasoning. For simplicity, we can classify models into three groups: models with payoff-dependent noise, models with bounded iterated reasoning, and hybrid models with both concepts. Six of our eight models possess a bounded iterated reasoning concept. For these models, level-0 is the anchoring element of the model and is of great importance. But there is no consensus among behavioral economists on level-0 behavior. Another contribution of our work is related to the specification of level-0 behavior. We have introduced a new parameter to the models, combining two outstanding assumptions about level-0 behavior. Thanks to this parameter, we have increased the predictive and explanatory powers of the models with iterative thought processes.

When we began to examine studies comparing multiple models, we encountered the following studies. Costa-Gomes, Crawford and Iriberri (2009) compare the prominent models of strategic thinking in Van Huyck, Battalio and Beil's (1990,1991) coordination games. To our knowledge, theirs is the first study to compare four leading behavioral models, which are QRE, Lk, CH and NI. They compare in-sample fits of these models and find that Lk and CH usually fit better than QRE and NI.

Goeree, Louis and Zhang (2013) apply the NI model developed by Goeree and Holt to Arad and Rubinstein's (2012) game and use estimated parameters to out-of-sample predict behaviors and beliefs in other game variations. They compare QRE, Lk, NI and NE in these games, and report that the Lk model underperforms compared to the NI, which assumes the common knowledge of noise. Furthermore, the NE predicts no worse than the Lk in these games.

Wright and Leyton-Brown (2013) analyse four strategic thinking models (QRE, Lk, CH, and QLk) in unrepeated, simultaneous-move games. They perform meta-analysis of these models, using nine different data sets, and evaluate the predictive performance of the models. They conclude that the QLk model of Stahl and Wilson (1995) consistently yields the best performance. However, the estimated parameters of the QLk are not consistent with their economic intuitions. Therefore, they create a new model family which is a variation of QLk and gives a better performance with fewer parameters.

Choo and Kaplan (2014) replicate Arad and Rubinstein's (2012) "11-20" game, which is well-designed for observing players' cognitive levels for models based on an iterated thinking process. For this reason, it is a popular choice for research that compares models. They use this game and its two variants, and achieve different predicted behavior by the Lk model. Then they allow subjects to best respond noisily in their SK model (which is the same with SLk in our work), and then they compare the SK model with QRE and SCH in terms of their statistical fits. Choo and Kaplan deduce that SK and SCH have better performance than QRE. Allowing for stochastic best response in Lk improves the explanatory power of the model.

This paper is organized as follows. The next chapter reviews the basic games related to the models. Chapter **??** explains the prominent models of strategic thinking. Chapter **??** introduces the experimental design, and Chapter **??** presents the experimental results. Chapter **??** explains the estimation process and comparison criteria, and discusses the estimation results. Chapter **??** concludes, and the Appendix presents the experimental instructions and the estimation codes.

CHAPTER 2

BASIC GAMES RELATED TO MODELS

There are several games that are closely associated with strategic thinking models and have an impact on the process of constructing these models. One of the most famous games is Keynes's beauty contest. Keynes (1936) introduces the following:

> ... professional investment may be likened to those newspaper competitions in which the competitors have to pick out the six prettiest faces from a hundred photographs, the prize being awarded to the competitor whose choice most nearly corresponds to the average preferences of the competitors as a whole; so that each competitor has to pick, not those faces which he himself finds prettiest, but those which he thinks likeliest to catch the fancy of the other competitors, all of whom are looking at the problem from the same point of view. It is not a case of choosing those which, to the best of one's judgement, are really the prettiest, nor even those which average opinion genuinely thinks the prettiest. We have reached the third degree where we devote our intelligences to anticipating what average opinion expects the average opinion to be. And there are some, I believe, who practice the fourth, fifth and higher degrees. (p. 156)

Nagel (1995) Ho, Camerer and Weigelt (1998), and Camerer, Ho and Chong (2004) formed symmetric n-person guessing games with inspiration taken from Keynes's beauty contest. In this guessing game, players are asked to choose numbers between lower and upper limits, and the player who guesses the closest to p-times the average wins a prize. There were 15-18 subjects in Nagel (1995), 3-7 subjects in Ho et al. (1998), and 24 subjects in Camerer et al. (2004). The lower and upper limits were 0-100 in Nagel (1995) and Camerer et al.(2004), 0-100 or 100-200 in Ho et al. (1998). The p-value was 1/2, 2/3 or 4/3 in Nagel (1995); 0.7, 0.9, 1.1 or 1.3 in Ho et al. (1998) and 2/3 in Camerer et al. (2004). Equilibrium theory predicts that all players guess their lower limit when $p < 1$ and upper limit when $p > 1$. If limits are 0-100 and p-value is 2/3 (the most well-known case), then the equilibrium will be 0. However, subjects rarely made equilibrium guesses in the first round. The guess distributions have peaks at some choices, which points out an iterated thinking process. Nagel (1995) and Camerer et al. (2004) applied the Lk and CH models, respectively, to these games to explain subjects behavior, and got a particularly high performance because in beauty contest games, it is natural to use an iterative thought process.

Another suitable game for models with bounded iterated reasoning is Arad and Rubinstein's (2012) game involving two players simultaneously requesting an amount of money between 11 and 20 shekels (integers), which they are certain to receive. A player also receives an additional amount of 20 shekels if her choice is exactly one shekel less than that of the other player. Arad and Rubinstein used this ”11-20” game to explain subjects' behaviors with the Lk model and found that players did not use more than three steps of thinking. On the other hand, since levels are unambiguously obvious in ”11-20” game, there are two other studies using this game to estimate four different models. First, Goeree et al. (2013) employed ”11-20” game to out-of-sample predict the experimental results via Lk, QRE and NI models. They introduced that NI and QRE performed better to predict human behavior than Lk and NE. Choo and Kaplan (2014) replicated the ”11-20” game to show that the Lk and CH models outperform the QRE explaining behaviors when they allow players to best respond stochastically (we refer to them as SLk and SCH).

QRE has advantages in some types of games due to its random component of noise. In many varieties of two-person zero-sum games that have a unique NE (for example, Lieberman, 1960; O'Neill, 1987; Rapoport & Boebel, 1992), QRE predictions are found to have fit the data consistently better than both the random and Nash predictions. Behaviors in the following games were well explained by the QRE model: centipede game (McKelvey & Palfrey, 1992), all-pay auctions (Anderson, Goeree & Holt, 1998), traveler's dilemma (Capra, Goeree, Gomez & Holt, 1999), alternating-offer bargaining games (Goeree & Holt, 2000), private-value first price auctions (Goeree, Holt & Palfrey, 2002) and matching pennies (Goeree, Holt & Palfrey, 2003).

The CH model holds the whip hand in dominance solvable games as well as in coordination games. Camerer et al. (2004) state that thinking steps establish a connection with iterated deletion of dominated strategies. Thanks to the possibility of incorrect beliefs, CH also makes a precise prediction in coordination games. Stag hunt is a type of coordination game which characterizes a trade-off between safety and social cooperation. CH can predict a significant effect of group size in stag hunt games. Additionally, a magical coordination is performed by CH in market entry games where players decide simultaneously whether to enter or stay out of a market (Camerer et al., 2004). If the number of players who entered the market is less than or equal to the market capacity, the entrants all gain a given positive profit; but if the number of players who enter the market is more than the market capacity, all get a given negative profit. Staying out gives zero profit, regardless of how many subjects enter. Another game, the battle of the sexes, is a simplified version of a two-person market entry game with a capacity of one. All the above games can be adequately explained by level-k analysis. (Crawford, 2007). Lastly, Camerer et al. (2004) assert that CH can account reasonably well for the pattern in speculation and zero-sum betting, and money illusion games.

There are numerous applications of the Lk model to analyse human strategic behavior. These include hide and seek games with non-neutrally framed locations (Crawford & Iriberri, 2007), overbidding in independent private-value and common-value auctions (Crawford & Iriberri, 2007), coordination via Schelling-style focal points (Crawford, Gneezy & Rottenstreich, 2008), and optimal auction games (Crawford, Kugler, Neeman & Pauzner, 2009).

CHAPTER 3

MODELS OF STRATEGIC THINKING

In this section, we review the prominent behavioral models of strategic thinking in which we will compare these models to each other and to a model based on the Nash equilibrium. We also discuss their assumptions and cognitive requirements.

3.1 Quantal Response Equilibrium

McKelvey and Palfrey (1995) put forward the notion of Quantal Response Equilibrium (QRE) to capture the cost-sensitive deviations from equilibrium. Players' decisions are noisy with a specified distribution, which is logit in almost all applications, adjusted by a precision parameter. Players make their choice on the basis of relative expected utility, taking the noisiness of others' decisions into consideration. Then a QRE is defined as a fixed point in the strategy set. Basically, this model relaxes the assumption of perfectly maximizing behavior: Players best respond noisily, rather than with certainty.

Consider a two-player (three-player for the second treatment) symmetric game with finite set of actions, A. Let $\pi_i^e(a_i, s_{-i})$ be player i's expected payoff of choosing $a_i \in A$ against strategy profile s_{-i}. Adopting the familiar logit formulation, the quantal best response by player i to s_{-i} is a mixed strategy $s_i : [0,1]^{|A|} \longrightarrow [0,1]^{|A|}$ with components

$$s_i(a_i) = \frac{\exp(\lambda.\pi_i^e(a_i, s_{-i}))}{\sum_{a' \in A} \exp(\lambda.\pi_i^e(a_i', s_{-i}))} \qquad \forall a_i \in A \tag{1}$$

The precision parameter, $\lambda \geq 0$, determines how sensitive the response function is with respect to expected payoffs, with $\lambda = 0$ corresponding to uniform randomization and $\lambda \to \infty$ corresponding to best response. In applications, this precision parameter is estimated statistically. For our case, it is calibrated from the analysis performing on three baseline games. With estimated precision, we out-of-sample predict the choice distributions in all game variations.

McKelvey and Palfrey (1995) prove that there exists a QRE with non-negative precision for any normal form game. Unlike existence, the uniqueness is not guaranteed. However, there is a unique QRE with the specified precision for all of our games.

From the point of view of explaining strategic thinking, QRE assumes homogeneity on players' levels of strategic sophistication. In many experiments, the behavior of the players, however, is observed to be quite heterogeneous. This leads to the emergence of particular class of models to allow for different cognitive levels among players.

3.2 Levek-k

The leading model allowing for heterogeneity is Level-k. In contrast to QRE, the Level-k model relaxes the assumption of mutually consistent beliefs, while the assumption of perfectly maximizing behavior remains valid.

The main idea behind the Level-k model is based on separating players into levels according to their sophistication of strategic thinking and proposing a particular structure on players' beliefs about others' decisions.

There are different versions of the Level-k model such as Nagel (1995), Stahl and Wilson (1995), Costa-Gomes, Crawford and Broseta (2001), and Costa-Gomes and Crawford (2006). They have different assumptions about types of players, the number of parameters, the accuracy of best responses, the belief distribution, and the specification of level-0 behavior.

In this study we consider the Level-k model of Nagel (1995) with some modifications. The model consists of an iterative decision mechanism for players performing k steps of reasoning. The iterative process starts with a non-strategic level-0 player who is assumed to choose according to some common knowledge probability distribution. In general, it is taken as uniformly random over all strategies. A level-k player, for $k \geq 1$, assumes that all other players are level-(k-1) and best responds to others' strategies according to this belief. In other words, a level-1 player best responds to a level-0 player, a level-2 player best responds to a level-1 player, and so forth. If a level-k player has multiple best responses, he/she uniformly randomizes over them.

The exogenous specification of the level-0 behavior is of great importance, because level-0 player is the anchoring element of the recursive structure. There are two main assumptions about this specification in the literature. The first assumption is the uniform randomization of level-0 behavior. The second one is that level-0 player picks a salient action, if it exists, over a strategy set. The salient action refers to the most obvious choice in the feasible strategy space. By combining these two assumptions, we have proposed more attentive specification of level-0 behavior in order to increase the explanatory power of the model. Level-0 type plays uniformly random across all strategies with probability $r \in [0, 1]$ or chooses the most obvious strategy (18 TL for our case) with probability $1 - r$.

To avoid entering into a vicious cycle, we assume that all players belong to levels 0, 1, or 2. f(0), f(1), and f(2) are their relative frequencies respectively. This frequency distribution and r parameter, 4 parameters in total, are estimated using experimental data via maximum likelihood estimation.

3.3 Cognitive Hierarchy

Camerer et al. (2004) proposed a new structural model which is a variant of the Level-k. The only difference between Cognitive Hierarchy and Level-k is an assumption of players' belief distributions. Unlike in the Level-k model, in the CH model, level-k players best respond to their opponents that are distributed from level-0 to level-(k-1), instead of to level-(k-1) players only. The model assumes that a level-k player's belief about the relative frequencies of lower level players is equal to normalized actual proportions.

Let the relative frequency of level-k player be f(k), and a level-k player's belief of proportion of level-h be $g_k(h)$. The CH model posits that

$$g_k(h) = \begin{cases} \frac{f(h)}{\sum_{l=0}^{k-1} f(l)} & \text{if } h < k \\ 0 & \text{if } h \geq k \end{cases} \tag{2}$$

We focus the CH model on two-player (three-player for the second treatment), symmetric games with a finite set of actions, A. Let $\pi_i(s_i^j, s_{-i}^{j'})$ be player i's payoff of choosing s_i^j against his/her opponent's strategy profile $s_{-i}^{j'}$. The expected payoff of strategy s_i^j for a level-k player, given his/her beliefs, is as follows:

$$E_k(\pi_i(s_i^j)) = \sum_{j'=1}^{|A|} \pi_i(s_i^j, s_{-i}^{j'})\{\sum_{h=0}^{k-1} g_k(h).P_h(s_{-i}^{j'})\} \tag{3}$$

where $P_k(s_i^j)$ denotes the probability that player i choosing strategy s_i^j according to decision mechanism of the level-k:

$$P_k(s_i^*) = \begin{cases} \frac{1}{|S_i^*|} & \text{if } s_i^* \in S_i^* \quad \& \quad S_i^* \equiv \arg\max_{s_i^j} E_k(\pi_i(s_i^j)) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

In applications, we assume that the frequency distribution of players follows a Poisson distribution, which is described by a single parameter τ (both its mean and variance).

$$f(k) = \frac{\tau^k.e^{-k.\tau}}{k!} \tag{5}$$

The specifications of the level-0 behavior are the same as those in the level-k model. In the Poisson-CH model, only two parameters, τ and r, are estimated, which makes it easier to work with this model statistically.

3.4 Noisy Introspection

Models using the equilibrium concept, such as NE and QRE, explain behavior successfully in repeated games where players have the opportunity to learn about others' decisions and beliefs and adapt to them gradually. However, in one-shot interactions as in our games, there are systematic deviations from equilibrium. Goeree and Holt (2004) introduce a structural model of noisy introspection designed to explain human strategic behavior in unrepeated games. To obtain more realistic results, they relax the equilibrium requirement which is based on consistency of beliefs and actions.

The critical part of the NI model is the notion of the common knowledge of noise: Players' behaviors are noisy, but they are aware of the noisiness of players' decisions and act accordingly. In the NI model there is an assumption of heterogeneity in levels of strategic sophistication, as in the Level-k model. Unlike the Level-k model, Goeree and Holt inject noise into the iterated thinking process about others' actions and beliefs. That is to say, level-1 player makes a noisy best response to level-0, level-2 player makes a noisy best response to level-1, so on.

Consider a two-player (three-player for the second treatment), symmetric game with finite set of actions, A. Denote the expected payoff from choosing $a \in A$ when a player's belief about the other's play is q by $\pi^e(a, q)$. q is a probability distribution over A. Using the well-known logit choice function, we can define a player's better response mapping
$\phi_\mu : [0, 1]^{|A|} \longrightarrow [0, 1]^{|A|}$ in this manner:

$$\phi_\mu^a(q) = \frac{\exp(\pi^e(a, q)/\mu)}{\sum_{a' \in A} \exp(\pi^e(a', q)/\mu)} \tag{6}$$

The noise parameter associated with a player's decision, μ, is a reciprocal of the precision parameter in the QRE. As μ goes to zero, the NI model reduces to the Level-k model, to which players make a best response. As μ goes to infinity, players act uniformly random.

Players' higher order beliefs are formed by using better response mapping iteratively. The following composition of better response mappings converges to the unique noisy introspection prediction as n goes to infinity:

$$\phi = \lim_{x \to \infty} \phi_{\mu_0} \circ \phi_{\mu_1} \circ \ldots \circ \phi_{\mu_n}(q) \tag{7}$$

The model assumes that noise parameters associated with higher levels of iterated thinking construct a non-decreasing sequence ($\mu_0 \leq \mu_1 \leq ... \leq \mu_\infty$), since every additional iteration makes the thought process more complex. This non-decreasing sequence diverges to infinity as the number of iterations increases. Better response mapping for $\mu_\infty = \infty$ maps any initial point for the process of iterated reasoning to a uniform probability distribution over a set of actions. Therefore, the initial belief probability, q, can be chosen arbitrarily.

In applications, to allow for a broad range of thinking levels, noise parameters are assumed to grow geometrically with each iteration:

$$\mu_k = t^k \mu_0 \quad \text{such that} \quad t > 1 \tag{8}$$

t is referred to as the telescoping parameter which determines the geometric growth rate of the noise parameter. Only these two parameters, μ_0 and t, are estimated in the NI model.

Goeree and Holt prove that there exists a unique noisy introspection prediction with a sequence of increasing and non-negative noise parameters that diverge to infinity. They guarantee that the limit sequence in the equation **??** converges to a unique point independent of the initial belief probability, q, for the iterated reasoning process.

3.5 Stochastic Level-k / Quantal Level-k

Stahl and Wilson (1995) describe the Level-k model with stochastic best response. That is why we designate their Level-k model as a Stochastic Level-k or Quantal Level-k model (henceforth known as SLk). They relax the perfectly maximizing behavior assumption by injecting random noise into the model via logistic response function as in QRE and NI. The SLk model maintains all but the strictly best response assumption in the Lk model.

Consider a two-player (three-player for the second treatment) symmetric game with a finite set of actions, A. $\pi_i(a_i, a_{-i})$ denotes player i's payoff of choosing strategy a_i against the other's strategy a_{-i}. The expected payoff for player i choosing strategy a_i is calculated as

$$\pi_i^e(a_i) = \sum_{a_{-i} \in A} \pi_i(a_i, a_{-i}) \{ \sum_{h=0}^{k-1} g_k(h).P_h(a_{-i}) \} \tag{9}$$

where $g_k(h)$ denotes a level-k player's belief about the proportion of level-h players, which is assumed in the SLk to be

$$g_k(h) = \begin{cases} 1 & \text{if } h = k-1 \\ 0 & \text{if } h \neq k-1 \end{cases} \qquad \forall k > 0 \tag{10}$$

and $P_k(a_i)$ denotes the probability of a level-k player i choosing strategy a_i is assumed as follows:

$$P_k(a_i) = \frac{\exp(\lambda.\pi_i^e(a_i))}{\sum_{a_i' \in A} \exp(\lambda.\pi_i^e(a_i'))} \qquad \forall a_i \in A \quad \& \quad \forall k > 0 \tag{11}$$

The frequency distribution of levels, the precision parameter λ, and the parameter r resulting from the specification of level-0 are estimated in the SLk model.

3.6 Stochastic Cognitive Hierarchy

Rogers, Palfrey and Camerer (2009) find an intuitive connection between two different structural models, CH and QRE. They generalize QRE by including heterogeneity in levels of reasoning (referred to as Heterogeneous Quantal Response

Equilibrium, HQRE) so that CH becomes a special case of the truncated HQRE. Motivated by these considerations, we introduce noise into the CH model to construct a more comprehensive model including two important concepts, payoff magnitude effect and limited strategic thinking.

The SLk and SCH approaches are similar, except for the assumption about players' belief distributions. The only difference from the above structure in the SLk model is the function $g_k(h)$. Like the CH model, the SCH assumes that a level-k player knows the actual relative proportion of lower level-h player:

$$g_k(h) = \begin{cases} \dfrac{f(h)}{\sum_{l=0}^{k-1} f(l)} & \text{if } h < k \\ 0 & \text{if } h \geq k \end{cases} \qquad \forall k > 0 \tag{12}$$

In the SCH model, three parameters, τ, r, and λ, are estimated statistically. It can be easily realized from the number of the estimated parameters that the SCH model is less flexible than the SLk model.

3.7 Generalized Cognitive Hierarchy and Level-m

Chong, Ho and Camerer (2014) introduce a generalization of the CH model, and demonstrate that the CH and a special version of the Lk are members of the same family. They integrate a new parameter, α, to the CH model, which reflects a stereotype bias that is a well-known phenomena in social psychology. In the GCH model, a level-k player's belief about the relative proportion of lower level players is namely

$$g_k(h) = \begin{cases} \dfrac{f(h)^\alpha}{\sum_{l=0}^{k-1} f(l)^\alpha} & \text{if } h < k \\ 0 & \text{if } h \geq k \end{cases} \qquad \forall k > 0 \tag{13}$$

where $\alpha \geq 1$ is the parameter of capturing stereotype bias. When $\alpha = 1$, the GCH reduces exactly to the CH model. When $\alpha > 1$, a level-k player's belief about players who are using less than k-steps of reasoning is focused on levels that occur more frequently. As α goes to ∞, a level-k player believes that other players are only of the modal lower rule. This special case of the GCH is called as the Level-m model. In the Lm model, players best respond to the most frequently occurring level players, unlike in the Lk. If this most frequently occurring level is level-(k-1), then these two models become identical.

The GCH model makes two new assumptions about level-0, which are the minimum-aversion tendency and the compromise effects. The model posits that level-0 players have a tendency to choose dominant strategies more frequently than dominated strategies. In our games, there is no strictly dominant or dominated strategy. For this reason, the contribution of this generalization of the CH model arises from the introduction of the parameter reflecting stereotype bias for our games. In applications, the GCH has four parameters, namely, τ, r, α, and β, but for our case we do not need β parameter.

CHAPTER 4

EXPERIMENTAL DESIGN

The experiment was composed of two treatments. We used a between-subjects design to avoid carry-over effects. Each subject participated in only one of these two treatments. Both treatments consisted of ten normal-form games having similar structures. In the first treatment, games involved two players, whereas in the second one, there were three. The two-player games were explained to the subjects as follows:

> The experiment consists of ten rounds. In each round, you will be matched with an anonymous partner to play the following game. For all ten rounds, there is no player in the game other than you and your matched partner. On your screen you will see a series of boxes containing various amounts of money. You are expected to choose one of these boxes. The monetary payoff you receive in each round is calculated as follows:
>
> $$\frac{\text{(The amount of money in the box you choose)}}{\text{(The number of players choosing this box including yourself)}}$$

As for three-player games, they were depicted below:

> The experiment consists of ten rounds. In each round, you will be matched with two anonymous partners to play the following game. For all ten rounds, there is no player in the game other than you and your matched partners. On your screen you will see a series of boxes containing various amounts of money. You are expected to choose one of these boxes. The monetary payoff you receive in each round is calculated as follows:
>
> $$\frac{\text{(The amount of money in the box you choose)}}{\text{(The number of players choosing this box including yourself)}}$$

Players did not learn their payoffs or the identity of their partners until the end of the experiment. They were assigned to a new partner at the beginning of each round.

We gave the following examples to ensure that they understood the games. For two-player games, the example was as follows:

> Example:
>
> 1.box: 9 TL
> 2.box: 6 TL
> 3.box: 3 TL
>
> In the above example, suppose that both you and your partner choose the first box. Then the first box will be chosen by two players including yourself, and your monetary payoff will be 9 TL / 2 = 4.5 TL
>
> Note that in each round you are matched with only one player. Therefore in each game, the number of players choosing any box can be no more than two.

As regards three-player games, the example was as follows:

> Example:
>
> 1.box: 9 TL
> 2.box: 6 TL
> 3.box: 3 TL
>
> In the above example, suppose that you choose the first box. Also suppose that one of your partners chooses the first box, and the other one picks the second box. Then the first box will be chosen by two players including yourself, and your monetary payoff will be 9 TL / 2 = 4.5 TL
>
> Note that in each round you are matched with only two players. Therefore in each game, the number of players choosing any box can be no more than three.

In some cases, the amounts of money in the boxes could be same. We did not give players an opportunity to choose the one that they wanted from boxes containing the same amount of money. Instead, the computer selected randomly one of the boxes for the players. Using this method, we prevented the possible biased results among boxes containing the same amounts. The below example is for such cases. A screenshot of the experiment is shown in Appendix A. The experiment instructions in English and Turkish are shown in Appendix B and Appendix C, respectively.

Example:

1.box: 9 TL
2.box: 6 TL
3.box: 6 TL
4.box: 3 TL

Which box do you choose?:

- 1.box
- Randomly selected one of 2.box and 3.box
- 4.box

If you choose the option ”Randomly selected one of 2.box and 3.box”, one of these two boxes will be selected by the computer for you in a completely random way.

In each treatment, subjects played ten games, as shown in the figures below, in random order. To avoid order effects, we shuffled games for each session. In other words, the order of games was randomized uniformly for every new session. Two-player games and three-player games are displayed in Figure 1 and Figure 2, respectively. In these figures, the number of boxes in each game and the amounts of money in the boxes are indicated under the name of the game.

Fig. 1. Two-player games.

Fig. 2. Three-player games.

In Figure 1 and Figure 2, three blue-colored games are our baseline games. We estimate the model parameters that maximize the likelihood of the observed choices in three baseline games and then use these estimated parameters to out-of-sample predict the choice distributions in all game variations. We first designed three blue-colored games. After that we constructed seven variations on these three games by just replicating some of the boxes in them. These three games are not actually special. We chose them as the baseline due to the fact that others originated from them. Designating different games as the baseline will not affect our results significantly, since the important thing is how many boxes there are with different amounts of money.

After the subjects finished playing ten games, we sent them a questionnaire consisting of a number of demographical questions. Then subjects were asked to explain their decision-making processes. At the end of the experiment, one game out of ten that subjects played was randomly chosen to determine the subject's earnings. Adding 10 TL as a participation payment to this earning, subjects' total monetary payoffs from the experiment were determined. The average monetary payoff that subjects gained was 20.53 TL.

A total of 161 subjects participated in 12 experimental sessions. Six sessions were conducted for both treatments. Two-player games and three-player games had 80 and 81 subjects, -respectively. We conducted the experiment in the Finance Lab of Boğazici University using Z-Tree. Subjects were undergraduate students chosen from various departments at Boğazici University.

CHAPTER 5

EXPERIMENTAL RESULTS

Table 1 shows the distribution of choices made by 80 subjects in two-player games. Table 2 demonstrates the choice distributions in three-player games, where 81 subjects played. There are ten games in both treatments. The choosing percentages and frequencies of the boxes for each game are indicated in these tables. The percentages and frequencies of the boxes containing the same amount of money are given together due to our experimental design.

Not surprisingly, the frequency of choosing the box containing 18 TL (henceforth, x TL represents the box containing x TL) generally decreases as the number of alternative boxes in the game increases. The subjects in the two-player games chose 18 TL more frequently than those in the three-player games. As far as we can see from the players' responses about their decision making processes, in two-player games subjects' responses are generally separated into two groups. The first group follows the minimax strategy until incentives attract them. These subjects usually chose 18 TL after they compared their potential minimum payoffs, choosing 18 TL with their potential maximum payoffs by choosing the alternative box. They may deviate from 18 TL if the number of alternative boxes to 18 TL is larger than two, and/or the alternative box is 14 TL. On the other hand, the second group follows the level-k reasoning. They think about what others choose and select their strategies accordingly. Twenty-six of 80 subjects clearly acted according to the level-k decision rule. One's response among them is as follows:

> In general I try to choose a box other than 18 TL, since I think that most people tend to pick 18 TL. So I prefer to get 10 TL rather than getting half of 18 TL. Similarly if there are several alternative boxes to 18 TL, I choose 18 TL because I suppose that they can deviate from 18 TL to alternative boxes.

Another subject, whose response was in line with the iterated decision rule, can be seen below:

> I assume my opponent is a robot that chooses randomly. Then I act to maximize my expected payoff accordingly, so I pick 18 TL. But if my opponent thinks like me, we can meet at the same box, unfortunately. My calculations lead me to the high amount.

As for three-player games, 33 of 81 subjects followed the level-k thought process. A typical example of level-k reasoning is as follows:

> In the games with more boxes containing 12 TL, I guess that everyone picks one of 12 TL boxes, so I choose 18 TL. On the other hand, in the games with more boxes containing 9 TL, I choose 12 TL with the assumption that everyone tends to pick 18 TL. Similarly, in the games with three boxes containing 18 TL, 12 TL and 9 TL/6 TL, I again choose 18 TL using the same assumption as above.

For iterated reasoning, the complexity of the thought process increases with every additional iteration. The response of a higher level subject is as follows:

> Thanks to keyboard sounds that I am hearing right now, I realized that everybody has thought a lot. First, I decided which box I would choose, regardless of what I think others will do. Then I modified my decision, considering the possibility that others have thought just like me. Finally, I became paranoid, figuring that these guys are clever, and they have thought the same thing as I have.

Original texts of these responses of players are demonstrated in Appendix D.

Table 1. Observed Distributions of Choices in Two-Player Games

Game 1			Game 2		
Boxes	Percentages	Frequencies	Boxes	Percentages	Frequencies
18 TL	80.00	64	18 TL	70.00	56
12 TL	20.00	16	12 TL	30	24
			12 TL		
Game 3			**Game 4**		
Boxes	Percentages	Frequencies	Boxes	Percentages	Frequencies
18 TL	53.75	43	18 TL	60.00	48
12 TL	46.25	37	12 TL	40.00	32
12 TL			12 TL		
12 TL			12 TL		
			12 TL		
Game 1			**Game 2**		
Boxes	Percentages	Frequencies	Boxes	Percentages	Frequencies
18 TL	86.25	69	18 TL	77.50	62
10 TL	13.75	11	10 TL	22.50	18
			10 TL		
Game 7			**Game 8**		
Boxes	Percentages	Frequencies	Boxes	Percentages	Frequencies
18 TL	71.25	57	18 TL	68.75	55
10 TL	28.75	23	14 TL	31.25	25
10 TL					
10 TL					
10 TL					
Game 9			**Game 10**		
Boxes	Percentages	Frequencies	Boxes	Percentages	Frequencies
18 TL	61.25	49	18 TL	37.50	30
14 TL	38.75	31	14 TL	62.50	50
14 TL			14 TL		
			14 TL		
			14 TL		

Table 2. Observed Distributions of Choices in Three-Player Games

Game 1

Boxes	Percentages	Frequencies
18 TL	76.54	62
12 TL	20.99	17
6 TL	2.47	2

Game 2

Boxes	Percentages	Frequencies
18 TL	59.26	48
12 TL	30.86	25
9 TL	9.88	8

Game 3

Boxes	Percentages	Frequencies
18 TL	71.60	58
12 TL	28.40	23
12 TL		

Game 4

Boxes	Percentages	Frequencies
18 TL	51.85	42
12 TL	48.15	39
12 TL		
12 TL		

Game 5

Boxes	Percentages	Frequencies
18 TL	43.21	35
12 TL	56.79	46
12 TL		
12 TL		
12 TL		

Game 6

Boxes	Percentages	Frequencies
18 TL	44.44	36
12 TL	55.56	45
12 TL		
12 TL		
12 TL		
12 TL		

Game 7

Boxes	Percentages	Frequencies
18 TL	55.56	45
12 TL	32.10	26
9 TL	12.34	10
9 TL		

Game 8

Boxes	Percentages	Frequencies
18 TL	55.56	45
12 TL	24.69	20
9 TL	19.75	16
9 TL		
9 TL		

Game 9

Boxes	Percentages	Frequencies
18 TL	60.49	49
12 TL	17.28	14
12 TL		
9 TL	22.22	18
9 TL		

Game 10

Boxes	Percentages	Frequencies
18 TL	66.67	54
12 TL	33.33	27
12 TL		
9 TL	0	0

CHAPTER 6

ESTIMATIONS

To compare the models of strategic thinking, we have used two approaches that were explained in detail in the following two subsections. These approaches were the evaluation of out-of-sample prediction performance of models and the comparison of sample fits of models. The former approach was used to answer to the question of which strategic thinking model is best suited to predicting human behavior in one-shot games, while the latter one was appropriate for determining which model we should prefer to explain human strategic behavior.

In both approaches, we have estimated model parameters. These estimated parameters were derived from maximum likelihood estimation. To construct a log-likelihood function, we denoted $f_m(a)$ as the observed frequency of choosing box a from the strategy set A in game m, ($m \in M$), and $p_a(x_1, ..., x_n)$ as the corresponding predicted probability from the model consisting of parameters x_j's. Then the log-likelihood function was

$$\text{log}L(x_1, ..., x_n) = \sum_{m \in M} \sum_{a \in A} f_m(a).\text{log}(p_a(x_1, ..., x_n)) \tag{14}$$

where M was equal to $\{1, 5, 8\}$ for the first treatment, and $\{1, 2, 3\}$ for the second one as the baseline. Estimation codes for QRE and SLk are given in Appendix E as an example.

To maximize the log-likelihood function, we made use of the optimization toolbox of MATLAB. In this toolbox, there is a ”fminsearch” function that finds the minimum of a scalar function of several variables. This function works only for unconstrained nonlinear optimization. But we had constraints for some models, so we used a ”fminsearchbnd” function, which was its converted version, to a constrained nonlinear optimization. The fminsearchbnd function uses the Nelder-Mead simplex algorithm (Nelder & Mead, 1965) as described in Lagarias, Reeds, Wright and Wright (1998). It is suitable for providing rapid results; however, we needed to implement one more step to find the global minimum because this function converges to a local minimum instead of the global one. For this reason, another useful function, ”rmsearch”, helped us to achieve the global minimum by using simple framework: It automatically generates random samples for us, tests which result in the best initial points, then starts our chosen optimizer at that set of points, and finally compiling the results. Using the rmsearch function, we repeated the estimation process, considering different starting points at least 100 times for each model. For some models, we tried with 10000 different starting points for the model parameter to ensure that our estimates were the global minimum. Lastly, these optimization functions were designed to find the minimum of an objective function. However, our aim was to maximize the log-likelihood function. As a result we looked for the minimum of the negative log-likelihood, which gave us the maximum of the log-likelihood function.

While estimating models, we made the substantial assumption about specification of level-0 behavior. As we described in the level-k part of Chapter **??**, we introduced a new parameter, r, about the level-0 specification, to the models founded on an iterative thought process. Level-0 type played uniformly random across all strategies with probability r, or chose the salient action, which was the box including 18 TL for all games in this experiment, with probability 1-r. We have discussed the comparison of models in two parts, in terms of parameter r. In the first case we set the parameter r as zero, which means that level-0 players are sure to choose the salient action 18 TL. In the other case, we considered the parameter r as an endogenous variable and estimated it along with other parameters. The results of both parts can be seen in the following subsections.

The estimation of the models also required a prior arbitrary specification of the highest level type, L_k, that exists in the data. In the Lk, SLk, and Lm models, we estimated the log-likelihoods for successive L_k's starting from $L_k = 2$ until there was no statistically significant increase in the log-likelihood values for them with an additional increase in L_k. Then we set this L_k value as the highest level type for the related model. This highest level type was $L_k = 2$ in the Lk, and $L_k = 3$ in the Lm. In the SLk model, L_k could be equal to 2, 3, 4 or 5 on a case by case basis. The CH, SCH, and GCH were estimated with an arbitrary high $L_k = 10$. This arbitrary high type L_k was set 20 in the NI model.

For both level-0 behavior and the model based on random behavior, the randomization process was generated over all boxes in a game rather than options on the screen. To illustrate, if there are three boxes with one 18 TL and two with 12 TL in the game and we assume uniformly random distribution, then each box is chosen with a probability of 1/3. When estimating the model based on the Nash equilibrium, we have mixed uniformly over pure equilibrium strategies to determine predicted probabilities.

In three-player games, when r was set as zero, there were choosing strategies (boxes with 6 TL and 9 TL) that were predicted to have zero probability for the Lk, Lm, CH, and GCH models, then the product of all the likelihoods was zero. To overcome the zero-likelihood problem, we did not include terms coming from the strategy for 9 TL and 6 TL in the estimation process. In other words, we have made estimations among choices that were predicted by the models with other than zero probability. But this was a problem for comparisons of the statistical fits of the models, since the log-likelihood values of these models became significantly higher and were therefore unfair. To make a fair comparison, we chose the mean of squared distances (MSD) as a measure of fitting for this case.

6.1 Out-of-sample predictions

All of the models mentioned in Chapter **??** have been proposed to explain human behaviors in one-shot interactions. The next step will naturally be to try to predict these behaviors. Although there are several studies in the literature about the comparison of models in terms of their explanatory power, comparing the predictive power of these models has not received much attention. But we may face the danger of overfitting while explaining the data. As stated in Wright and Leyton-Brown (2013), in such cases we may have chosen the most flexible model rather than the most accurate one. In light of these issues, we have adopted an out-of-sample prediction which follows the procedure of using the estimated parameters from three baseline games to predict the choice distributions in all game variations. To measure the predictive performance of these models, we computed the mean of the squared distances (MSD) between the predictions and the observed data. We computed the MSD using percentages instead of probabilities in three different ways.

6.1.1 MSD for all ten games

We computed the MSD for all ten games and achieved the results shown in Table 3. We made estimations for both cases where r is equal to zero or estimated. In the case of $r = 0$, QRE predicted human behavior very well for both two-player and three-player games. However, when we considered r as a free parameter and estimated it, the SLk model improved significantly and worked better than all other models in tracking the observed data. Another noteworthy point is that in $r = 0$ case, the performance of the CH was substantially changed in a negative direction when we moved on to three-player games from two-player ones. Although the Lk model performed better than NE, it did not have a sufficiently good performance to surpass the models with payoff dependent noise. We can say that the QRE, SLk, and NI models were generally clustered in the upper rows of the ranking in terms of MSD for ten games. The NE performed poorly compared to the behavioral models.

Table 3. Means of Squared Distances for Ten Games

r=0				r estimated			
2-player games		3-player games		2-player games		3-player games	
Model	MSD-10	Model	MSD-10	Model	MSD-10	Model	MSD-10
QRE	62.3	QRE	162.6	SLk	20.8	SLk	145.4
NI	78.2	SLk	168.6	QRE	62.3	QRE	162.6
CH	84.1	NI	192.7	NI	78.2	NI	192.7
GCH	84.1	SCH	275.5	SCH	84.8	Lk	244.5
SCH	96.6	Lk	411.8	CH	94.4	CH	244.5
SLk	240.1	Lm	411.8	GCH	94.4	GCH	244.5
Lk	423.0	CH	577.6	Lk	142.2	Lm	244.5
Lm	423.0	GCH	577.6	Lm	142.2	SCH	275.5
NE	795.9	NE	764.0	NE	795.9	NE	764.0
Random	1801.1	Random	1703.3	Random	1801.1	Random	1703.3

6.1.2 MSD for seven game variations

To observe the pure prediction performance of the models, we computed the MSD for only seven game variations, as shown in Table 4. After excluding three baseline games from the calculation of MSD, the MSD value of NE decreased for both two-player and three-player games in both $r = 0$ and r estimated cases. At the same time, behavioral models performed worse in terms of the MSD for ten games than for seven game variations. One of two exceptions was the QRE model. In three-player games, the predictive performance of QRE for these seven games was better than its performance for all ten games. The other one was the CH performance in two-player games of $r = 0$ case. In general, QRE had the best or nearly the best performance for seven games among all models in total. As for the Lk model, it was outperformed by even NE in one case.

Table 4. Means of Squared Distances for Seven Games

r=0				r estimated			
2-player games		3-player games		2-player games		3-player games	
Model	MSD-7	Model	MSD-7	Model	MSD-7	Model	MSD-7
QRE	85.2	QRE	118.3	SLk	29.7	QRE	118.3
NI	111.1	SLk	204.5	QRE	85.2	SLk	186.4
CH	116.4	NI	229.7	NI	111.1	NI	229.7
GCH	116.4	SCH	355.9	SCH	120.9	Lk	315.0
SCH	136.8	CH	383.6	CH	133.8	CH	315.0
SLk	333.9	GCH	383.6	GCH	133.8	GCH	315.0
NE	404.0	Lk	538.9	Lk	197.5	Lm	315.0
Lk	559.4	Lm	538.9	Lm	197.5	SCH	355.9
Lm	559.4	NE	721.2	NE	404.0	NE	721.2
Random	1840.0	Random	1519.4	Random	1840.0	Random	1519.4

6.1.3 MSD for seven game variations among the percentages of the choice of 18TL

In this part, we looked for the MSD for seven game variations among the percentages of subjects choosing the box with 18 TL, as shown in Table 5. In our games, there were different numbers of boxes. To avoid giving different weights to the games in the calculation process of the MSD, we took just the percentage of subjects choosing boxes with 18 TL into account. The model which was most affected by this regulation was the CH, which rose to second place just after the QRE in two-player games of $r = 0$ case. There was no further change in the rankings of models from those of the previous subsection.

Table 5. Means of Squared Distances for Seven Game Variations among the Percentages of the Choice of 18TL

r=0				r estimated			
2-player games		3-player games		2-player games		3-player games	
Model	MSD-7-18	Model	MSD-7-18	Model	MSD-7-18	Model	MSD-7-18
QRE	64.9	QRE	49.6	SLk	21.4	QRE	49.6
NI	85.3	CH	94.5	QRE	64.9	SLk	101.2
CH	90.2	GCH	94.5	NI	85.3	NI	143.7
GCH	90.1	SLk	125.6	SCH	93.3	Lk	226.9
SCH	104.8	NI	143.7	CH	103.2	CH	226.9
SLk	259.6	SCH	248.7	GCH	103.2	GCH	226.9
NE	286.4	Lk	388.8	Lk	152.4	Lm	226.9
Lk	431.4	Lm	388.8	Lm	152.4	SCH	248.7
Lm	431.4	NE	389.9	NE	286.4	NE	389.9
Random	1347.7	Random	1086.4	Random	1347.7	Random	1086.4

6.2 Comparing statistical fit

In the literature, the most common method of comparing the behavioral models is to check their log-likelihood values. To measure the predictive power of the models, we estimated the parameters of the models for three baseline games. In this part, we estimated them for all ten games together, and compared their log-likelihood values to determine their performances in explaining the observed data pattern. The log-likelihood values of the models are shown in Table 6. As we explained in Chapter **??**, to solve the zero-likelihood problem, the MSD was chosen as a measure of fitting in three-player games when r was set as zero.

Three different models, NI, SLk, and GCH, fit the data most accurately in three different cases. The most interesting result is that the GCH model had an excellent performance on describing subjects' behavior in three-player games when r was estimated, in spite of its poor fit in the other three cases. Moreover, the three-player games when r was estimated was the only case where the GCH estimates gave different results from the CH estimates out of all estimations. The fitting performance of the QRE was relatively inconsistent. When r was estimated, the QRE estimates fit the data well enough in two-player games. However, those in three-player games had the worst fit to the observed data among all models. With regard to the Lk model, it once again underperformed compared to the models considering payoff dependent noise. Not only the Lk, but also the other models assuming perfectly maximizing behavior performed much worse than those relaxing this assumption.

Table 6. Log-Likelihood Values

r=0				r estimated			
2-player games		3-player games		2-player games		3-player games	
Model	LL	Model	MSD	Model	LL	Model	LL
NI	-715.7	SLk	101.9	SLk	-715.6	GCH	-905.1
QRE	-717.5	NI	113.1	NI	-715.7	SLk	-906.2
SCH	-717.8	SCH	154.6	QRE	-717.5	SCH	-906.8
SLk	-719.7	QRE	163.9	SCH	-717.6	CH	-907.8
CH	-724.3	Lk	273.5	CH	-722.1	NI	-908.1
GCH	-724.3	CH	273.5	GCH	-722.1	Lk	-909.4
Lk	-746.3	GCH	273.5	Lk	-724.0	Lm	-911.5
Lm	-746.3	Lm	273.5	Lm	-724.0	QRE	-915.0

6.3 Results and discussion

In this section, we discuss the main results of the estimations, and their rationales.

Finding 1: *The Nash Equilibrium underperformed in explaining human play compared to the models of strategic thinking.*

Many experimental studies have already suggested that human behaviors in one-shot games systematically deviate from equilibrium, as in this paper. The basic conditions for equilibrium are correct beliefs and perfectly maximizing behavior. But these conditions are rather demanding for one-shot games, since players have not any opportunity to learn about others' decisions and beliefs in one-shot interactions. For this reason it is not surprising that NE is outperformed by the models which relaxed one or both of these assumptions.

Finding 2: *The specification of level-0 behavior had a significant effect on the predictive and explanatory power of the models that were based on an iterative thought process.*

The specification of level-0 behavior was crucial in higher level players' actions. Two different assumptions about level-0 behavior resulted in a considerably different performance of the models in capturing the observed data. Instead of assuming that level-0 player will certainly choose the salient choice, introducing the new parameter to the models in order to give level-0 player the chance to choose randomly or in line with the most obvious choice improved the performance of the models to explain human behavior.

Finding 3: *The models with payoff dependent noise had a consistently better predictive performance than those without noisy behavior.*

The models with probabilistic best response such as QRE, NI, SLk, and SCH generally outperformed the models that assumed perfectly maximizing behavior such as Lk, CH, GCH, and Lm. For one-shot games, the assumption that players strictly best respond to others was not realistic in general. Injecting some noise into the models led to improved predictions for the aggregate choice distribution. It can be said that the Stochastic Level-k was roughly the most accomplished model to predict human behavior in our games.

Finding 4: *Predicting out-of-sample behavior yielded different results in explaining in-sample behavior.*

By explaining in-sample behavior, we found the most flexible model. Therefore, the models with more free parameters had an advantage in capturing the observed data. It is for this reason that the NI performed better than the QRE in all cases, unlike their performance in predicting out-of-sample behavior.

Finding 5: *A little modification on game structure might lead to drastically different results in the predictive performance and statistical fit of the models.* The most important contribution of our work was to determine how well the model fit or predict the experimental data depends on the structure and concept of the game. No model was superior to other behavioral models, regardless of game structure. Even across very similar games, there could be significant changes in the predictive performance of models. The most striking evidence for this finding came from the statistical fits of the QRE and GCH when r was estimated. In two-player games, QRE performed quite well while GCH performed poorly. However, when we estimated these models in three-player games which had a structure very similar to that of the two-player ones, GCH worked surprisingly better than all other models while QRE became the worst among all models. The only difference between these games that resulted in such a drastic change on statistical fits of models was simply the number of players in the game.

CHAPTER 7

CONCLUSION

In this study, we performed an exhaustive comparison of eight models of strategic thinking. To our knowledge, this is the first study that includes eight behavioral models. We designed a new game with variations, taking into account the characteristics of the models to avoid favoring a particular family of models.

The main question that we aimed to answer was whether there was a consistent superiority of any one model over other behavioral models. But having calculated estimations, we realized that performances of the models were directly related to the structure of a game. The more interesting result was that the performance of the models can fluctuate just by adding an additional player to the game.

In fact, when comparing the models, all our assumptions had a significant effect on estimation results. Introducing a new parameter, r, had a profound effect on the performance of some models. The method we choose to compare these models also had an impact on their performances. The model which predicted out-of-sample behavior very well could be inadequate to explain in-sample behavior, or vice versa.

The structural models of strategic thinking considered in this study, models with payoff dependent noise had better performance than those without noise parameter to predict human behavior. The SLk model in particular, which combines payoff dependent noise with iterated thinking process, gave the closest predictions to the observed data. These results are in line with Wright and Leyton-Brown (2013) and Choo and Kaplan (2014).

APPENDIX A

SCREENSHOT OF THE EXPERIMENT

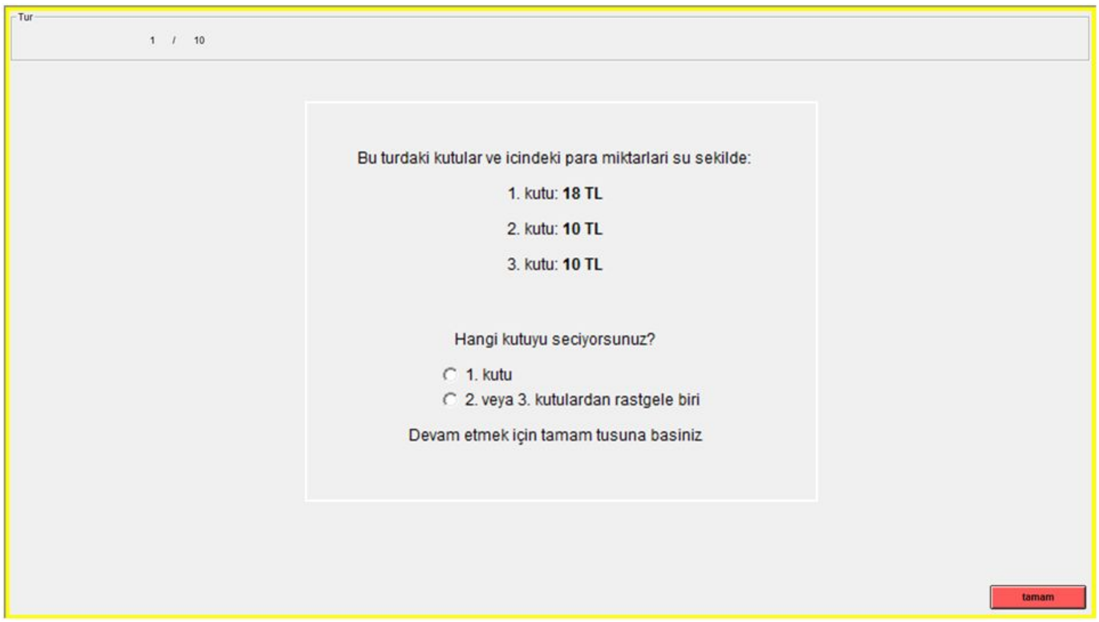

Fig. 3. Screenshot of the experiment.

APPENDIX B

EXPERIMENT INSTRUCTIONS IN ENGLISH

Welcome!

Thank you for your participation. The aim of this study is to understand how people decide in certain cases. From now on, the participants are forbidden to talk to each other. Violation of this rule requires termination of the experiment. If you have any question, please raise your hand and ask your question. We will come to you to answer your question.

The experiment will be conducted via computer and all decisions that you make will be transmitted via computer. You will receive the monetary payoff as a result of the game in this experiment. Your payoff depends on your decisions and those of the other players. In addition to this payoff, a participation fee for completing the experiment will be paid you in cash at the end of the experiment. Now we explain the game to be played during the experiment.

The game:

The experiment consists of ten rounds. In each round, you will be matched with an anonymous partner to play the following game. For all ten rounds, there is no player in the game other than you and your matched partner. On your screen you will see a series of boxes containing various amounts of money. You are expected to choose one of these boxes. The monetary payoff you receive in each round is calculated as follows:

$$\frac{\text{(The amount of money in the box you choose)}}{\text{(The number of players choosing this box including yourself)}}$$

Example 1:

1.box: 9 TL

2.box: 6 TL

3.box: 3 TL

In the above example, suppose that both you and your partner choose the first box. Then the first box will be chosen by two players including yourself, and your monetary payoff will be 9 TL / 2 = 4.5 TL

Note that in each round you are matched with only one player. Therefore in each game, the number of players choosing any box can be no more than two.

In some cases, the amounts of money in the boxes could be the same. The example below illustrates for such cases.

Example 2:

1.box: 9 TL

2.box: 6 TL

3.box: 6 TL

4.box: 3 TL

Which box do you choose?:

- 1.box
- Randomly selected one of 2.box and 3.box
- 4.box

If you choose the option ”Randomly selected one of 2.box and 3.box”, one of these two boxes will be selected by the computer for you in a completely random way.

At the end of the experiment, one game out of ten games that subjects played, was randomly chosen to determine subject's earnings. All rounds have the same probability of being selected. Adding 10 TL as a participation payment to this earning, subjects' total monetary payoffs from the experiment were determined.

APPENDIX C

EXPERIMENT INSTRUCTIONS IN TURKISH

Hoş geldiniz!

Katılımınız için teşekkür ederiz. Bu çalışmanın amacı, insanların belli durumlarda nasıl kararlar aldıklarını anlamaktır. Şu andan itibaren katılımcıların birbirleri ile konuşması yasaktır. Bu kuralın ihlali, deneyi sonlandırmamızı gerektiriyor. Eğer sorunuz varsa lütfen elinizi kaldırıp sorunuz. Yanınıza gelerek sorunuzu cevaplandıracağız.

Deney bilgisayar üzerinden gerçekleştirilecektir ve aldığınız bütün kararları bilgisayar üzerinden ileteceksiniz. Deney esnasında oynanacak oyun sonucunda bir para ödülü kazanacaksınız. Kazancınız, sizin ve diğer oyuncuların kararlarına bağlıdır. Bu kazanç ve buna ek olarak deneye katılım ücreti size deneyin sonunda nakit olarak ödenecektir. Şimdi deney esnasında oynayacağınız oyunu anlatmaya başlıyoruz.

Oyun:

Deney 10 tur sürecek ve her turda sizin dışınızdaki bir katılımcı ile eşleşerek aşağıda anlatacağımız oyunu oynayacaksınız. O turdaki oyunda, siz ve eşleştiğiniz katılımcı dışında bir oyuncu bulunmamaktadır. Size, her tur için, bir dizi kutu ve içlerindeki para miktarları gösterilecek. Sizden yapmanızı istediğimiz bu kutulardan birini seçmeniz.

O turda kazanacağınız para ödülü ise şu şekildedir:

$$\frac{\text{(Seçtiğiniz kutudaki para)}}{\text{(Siz dahil o kutuyu seçen kişi sayısı)}}$$

Örnek 1:

1.kutu: 9 TL

2.kutu: 6 TL

3.kutu: 3 TL

Yukarıdaki örnekte, sizin 1. kutuyu seçtiğinizi düşünelim. Diğer oyuncu da 1. kutuyu seçmiş olsun. O halde 1. kutuyu siz dahil 2 kişi seçmiş olacak ve sizin kazancınız = 9 TL / 2 = 4.5 TL olacak.

Not: Her tur için yalnızca 1 diğer katılımcı ile eşleşmektesiniz. Dolayısıyla, oynadığınız oyunda herhangi bir kutuyu seçen kişi sayısı en fazla iki olabilir.

Bazı durumlarda kutulardaki para miktarı aynı olabilir. Aşağıdaki örnek böyle bir durumu gösteriyor.

Örnek 2:

1.kutu: 9 TL

2.kutu: 6 TL

3.kutu: 6 TL

4.kutu: 3 TL

Hangi kutuyu seçiyorsunuz:

- 1.kutu
- 2. veya 3. kutulardan rastgele biri
- 4.kutu

Yukarıda ”2. veya 3. kutulardan rastgele biri” seçeneğini işaretlerseniz, bilgisayar tamamen rastgele şekilde, bu iki kutudan bir tanesini sizin için seçecektir.

Oynayacağınız 10 turdan biri rastgele seçilecek ve o turdaki kazancınız deneydeki gerçek kazancınız olacak. Bütün turların seçilme şansı aynıdır. Son olarak bu kazanca 10 TL'lik katılım ücretini ekleyerek toplam kazancınızı hesaplayacağız.

APPENDIX D

ORIGINAL TEXTS OF THE PLAYERS' RESPONSES

Original texts of the players' responses translated in Chapter **??** are as follows, in order of their appearance in this study:

> Genellikle miktarın az olduğu kutuyu seçmeye çalıştım. Bunun sebebi ise genel ortalamanın yüksek meblaya yöneleceğini düşünmemdi. Yani 18 TL'nin ikiye bölünmesinden ise 10 TL tek başına daha kazançlı geldi. Ayrıca aynı miktarın birkaç kutuda olduğu kısımlarda ise yüksek meblayı seçtim. Çünkü o zaman da aralarından birinin çıkma ihtimaline karşı onun işaretlenebileceğini düşündüm.

> Karşımdaki kişinin rastgele seçim yapan bir robot olduğunu düşünerek şıkları seçerken olasılık olarak kazanabileceğim miktarları karşılaştırdım. Yüksek olanı seçtim. Fakat şu var ki karşımdaki kişi de bu şekilde düşündüyse ikimiz de birbirimizin mantıklı düşüncesini bozmuş olabiliriz. Hesaplamalarım genelde beni yüksek miktara yönlendirdi.

> 12 TL'nin fazla olduğu kutulu turlarda herkesin 12 TL alacağını düşünerek 18 TL'yi seçtim. 9 TL fazlayken de insanların 18 TL'ye yöneleceğini düşünerek 12 TL'yi seçtim. Eşit sayıda kutular varken de insanların 12 TL'yi seçmeye meyilleri olacağınıdüşündüğüm için 18 TL'yi seçtim.

> Şu an çatur çutur gelen klavye seslerinden anlıyorum ki herkes temiz düşünmüş. Önce diğerlerinin ne düşündüğünü düşünmeden neyi seçeceğime karar verdim. Sonra başkaları böyle düşünür diyerek kararımı modifiye ettim. Sonra bu çocuklar zeki, onlar da bu kadarını düşünmüştür diyerek paranoyaklaşmaya başladım.

APPENDIX E

ESTIMATION CODES FOR QRE AND SLK

Out-of-sample prediction codes for QRE in two-player games are as follows.

QRE12 function:

```
function F = QRE12(input, lambda)
syms p18 p12 a
A = [18; 12];
B = [p18; p12];
p = [a; a];
x = 1;
TempPay2 = [a; a];
while (x<=length(A))
        y = 1;
        TempPay = [a; a];
        while(y<=length(A))
        TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
        y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end
S = solve(p18==p(1,1), p12==p(2,1));
p18calc = S.p18;
p12calc = S.p12;
if input == A(1,1)
F = p18calc;
elseif input == A(2,1)
F = p12calc;
end
end
```

QRE10 function:

```
function F = QRE10(input, lambda)
syms p18 p10 a
A = [18; 10];
B = [p18; p10];
p = [a; a];
x = 1;
TempPay2 = [a; a];
while (x<=length(A))
        y = 1;
        TempPay = [a; a];
        while(y<=length(A))
        TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
        y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end
S = solve(p18==p(1,1), p10==p(2,1));
p18calc = S.p18;
p10calc = S.p10;
if input == A(1,1)
F = p18calc;
elseif input == A(2,1)
F = p10calc;
end
end
```

QRE14 function:

```
function F = QRE14(input, lambda)
syms p18 p14 a
A = [18; 14];
B = [p18; p14];
p = [a; a];
```

```
x = 1;
TempPay2 = [a; a];
while (x<=length(A))
    y = 1;
    TempPay = [a; a];
    while(y<=length(A))
    TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
    y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end
S = solve(p18==p(1,1), p14==p(2,1));
p18calc = S.p18;
p14calc = S.p14;
if input == A(1,1)
F = p18calc;
elseif input == A(2,1)
F = p14calc;
end
end
```

MLEBASE function:

```
function loglike = MLEBASE(v)
lambda = v;
loglike = -( 64*log(QRE12(18, lambda)) + 16*log(QRE12(12, lambda)) +...
69*log(QRE10(18, lambda)) + 11*log(QRE10(10, lambda)) +...
55*log(QRE14(18, lambda)) + 25*log(QRE14(14, lambda)) ) / 80;
end
```

Payoff function:

```
function F = Payoff(x, y)
if x == y
```

```
        F = 1/2;
elseif x  = y
        F = 1;
end
end
```

RunMLE function:

```
% Run this file to do the MLE estiamtes
clear;
clc;
opts = optimset('fminsearch');
opts.Display = ('iter');
opts.TolX = 1.e-4;
opts.MaxFunEvals = 10000;
opts.MaxIter = 10000;
LB = [0];
UB = [inf];
x0 = [1];
[x,fval,exitflag,output] = fminsearchbnd(@MLEBASE,x0,LB,UB,opts);
```

msd function:

```
lambda = x;

p12_obs = [64; 16] * (100/80);
p1212_obs = [56; 12; 12] * (100/80);
p121212_obs = [43; 37/3; 37/3; 37/3] * (100/80);
p12121212_obs = [48; 8; 8; 8; 8] * (100/80);
p10_obs = [69; 11] * (100/80);
p1010_obs = [62; 9; 9] * (100/80);
p10101010_obs = [57; 23/4; 23/4; 23/4; 23/4] * (100/80);
p14_obs = [55; 25] * (100/80);
p1414_obs = [49; 31/2; 31/2] * (100/80);
p14141414_obs = [30; 50/4; 50/4; 50/4; 50/4] * (100/80);

%-----------------------------------------------------------------

syms p18 p12 a
```

```
A = [18; 12];
B = [p18; p12];

p = [a; a];

x = 1;
TempPay2 = [a; a];
while (x<=length(A))
        y = 1;
        TempPay = [a; a];
        while(y<=length(A))
        TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
        y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end

S = solve(p18==p(1,1), p12==p(2,1));

pcalc12 = [S.p18; S.p12] * 100;

%------------------------------------------------------------

syms p18 p12_1 p12_2 a

A = [18; 12; 12];
B = [p18; p12_1; p12_2];

p = [a; a; a];

x = 1;
TempPay2 = [a; a; a];
while (x<=length(A))
y = 1;
```

```
TempPay = [a; a; a];
while(y<=length(A))
TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end

S = solve(p18==p(1,1), p12_1==p(2,1), p12_2==p(3,1));

pcalc1212 = [S.p18; S.p12_1; S.p12_2] * 100;

%-----------------------------------------------------------

syms p18 p12_1 p12_2 p12_3 a

A = [18; 12; 12; 12];
B = [p18; p12_1; p12_2; p12_3];

p = [a; a; a; a];

x = 1;
TempPay2 = [a; a; a; a];
while (x<=length(A))
y = 1;
TempPay = [a; a; a; a];
while(y<=length(A))
TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
```

```
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end

S = solve(p18==p(1,1), p12_1==p(2,1), p12_2==p(3,1), p12_3==p(4,1));

pcalc121212 = [S.p18; S.p12_1; S.p12_2; S.p12_3] * 100;

%-------------------------------------------

syms p18 p12_1 p12_2 p12_3 p12_4 a

A = [18; 12; 12; 12; 12];
B = [p18; p12_1; p12_2; p12_3; p12_4];

p = [a; a; a; a; a];

x = 1;
TempPay2 = [a; a; a; a; a];
while (x<=length(A))
y = 1;
TempPay = [a; a; a; a; a];
while(y<=length(A))
TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end

S = solve(p18==p(1,1), p12_1==p(2,1), p12_2==p(3,1), p12_3==p(4,1),
p12_4==p(5,1));

pcalc12121212 = [S.p18; S.p12_1; S.p12_2; S.p12_3; S.p12_4] * 100;
```

```
%--------------------------------------------------

syms p18 p10 a

A = [18; 10];
B = [p18; p10];

p = [a; a];

x = 1;
TempPay2 = [a; a];
while (x<=length(A))
y = 1;
TempPay = [a; a];
while(y<=length(A))
TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end

S = solve(p18==p(1,1), p10==p(2,1));

pcalc10 = [S.p18; S.p10] * 100;

%--------------------------------------------------

syms p18 p10_1 p10_2 a

A = [18; 10; 10];
B = [p18; p10_1; p10_2];

p = [a; a; a];
```

```
x = 1;
TempPay2 = [a; a; a];
while (x<=length(A))
y = 1;
TempPay = [a; a; a];
while(y<=length(A))
TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end

S = solve(p18==p(1,1), p10_1==p(2,1), p10_2==p(3,1));

pcalc1010 = [S.p18; S.p10_1; S.p10_2] * 100;

%-----------------------------------------------------------

syms p18 p10_1 p10_2 p10_3 p10_4 a

A = [18; 10; 10; 10; 10];
B = [p18; p10_1; p10_2; p10_3; p10_4];

p = [a; a; a; a; a];

x = 1;
TempPay2 = [a; a; a; a; a];
while (x<=length(A))
y = 1;
TempPay = [a; a; a; a; a];
while(y<=length(A))
TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
y = y + 1;
```

```
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end

S = solve(p18==p(1,1), p10_1==p(2,1), p10_2==p(3,1), p10_3==p(4,1),
p10_4==p(5,1));

pcalc10101010 = [S.p18; S.p10_1; S.p10_2; S.p10_3; S.p10_4] * 100;

%-----------------------------------------------------

syms p18 p14 a

A = [18; 14];
B = [p18; p14];

p = [a; a];

x = 1;
TempPay2 = [a; a];
while (x<=length(A))
y = 1;
TempPay = [a; a];
while(y<=length(A))
TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
```

```
end

S = solve(p18==p(1,1), p14==p(2,1));

pcalc14 = [S.p18; S.p14] * 100;

%----------------------------------------------------------------------

syms p18 p14_1 p14_2 a

A = [18; 14; 14];
B = [p18; p14_1; p14_2];

p = [a; a; a];

x = 1;
TempPay2 = [a; a; a];
while (x<=length(A))
y = 1;
TempPay = [a; a; a];
while(y<=length(A))
TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end

S = solve(p18==p(1,1), p14_1==p(2,1), p14_2==p(3,1));

pcalc1414 = [S.p18; S.p14_1; S.p14_2] * 100;

%----------------------------------------------------------------------

syms p18 p14_1 p14_2 p14_3 p14_4 a
```

```
A = [18; 14; 14; 14; 14];
B = [p18; p14_1; p14_2; p14_3; p14_4];

p = [a; a; a; a; a];

x = 1;
TempPay2 = [a; a; a; a; a];
while (x<=length(A))
y = 1;
TempPay = [a; a; a; a; a];
while(y<=length(A))
TempPay(y,1) = Payoff(x,y)*A(x,1)*B(y,1);
y = y + 1;
end
TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
x = x + 1;
end
t = 1;
while (t<=length(A))
p(t,1) = TempPay2(t,1) / sum(TempPay2);
t = t + 1;
end

S = solve(p18==p(1,1), p14_1==p(2,1), p14_2==p(3,1), p14_3==p(4,1),
p14_4==p(5,1));

pcalc14141414 = [S.p18; S.p14_1; S.p14_2; S.p14_3; S.p14_4] * 100;

%---------------------------------------------------------

MSD10 = ( ( (p12_obs(1)-pcalc12(1))^2 + (p12_obs(2)-pcalc12(2))^2 ) + ...
( (p1212_obs(1)-pcalc1212(1))^2 + (p1212_obs(2)-pcalc1212(2))^2 +
(p1212_obs(3)-pcalc1212(3))^2 ) + ...
( (p121212_obs(1)-pcalc121212(1))^2 + (p121212_obs(2)-pcalc121212(2))^2 +
(p121212_obs(3)-pcalc121212(3))^2 + (p121212_obs(4)-pcalc121212(4))^2 ) + ...
( (p12121212_obs(1)-pcalc12121212(1))^2 +
(p12121212_obs(2)-pcalc12121212(2))^2 +
(p12121212_obs(3)-pcalc12121212(3))^2 +
```

```
(p12121212_obs(4)-pcalc12121212(4))^2 +
(p12121212_obs(5)-pcalc12121212(5))^2 ) + ...
        ( (p10_obs(1)-pcalc10(1))^2 + (p10_obs(2)-pcalc10(2))^2 ) + ...
        ( (p1010_obs(1)-pcalc1010(1))^2 + (p1010_obs(2)-pcalc1010(2))^2 +
(p1010_obs(3)-pcalc1010(3))^2 ) + ...
        ( (p10101010_obs(1)-pcalc10101010(1))^2 +
(p10101010_obs(2)-pcalc10101010(2))^2 +
(p10101010_obs(3)-pcalc10101010(3))^2 +
(p10101010_obs(4)-pcalc10101010(4))^2 +
(p10101010_obs(5)-pcalc10101010(5))^2 ) + ...
        ( (p14_obs(1)-pcalc14(1))^2 + (p14_obs(2)-pcalc14(2))^2 ) + ...
        ( (p1414_obs(1)-pcalc1414(1))^2 + (p1414_obs(2)-pcalc1414(2))^2 +
(p1414_obs(3)-pcalc1414(3))^2 ) + ...
        ( (p14141414_obs(1)-pcalc14141414(1))^2 +
(p14141414_obs(2)-pcalc14141414(2))^2 +
(p14141414_obs(3)-pcalc14141414(3))^2 +
(p14141414_obs(4)-pcalc14141414(4))^2 +
(p14141414_obs(5)-pcalc14141414(5))^2 ) ) / 10;

        MSD7 = ( ( (p1212_obs(1)-pcalc1212(1))^2 + (p1212_obs(2)-pcalc1212(2))^2
+ (p1212_obs(3)-pcalc1212(3))^2 ) + ...
        ( (p121212_obs(1)-pcalc121212(1))^2 + (p121212_obs(2)-pcalc121212(2))^2 +
(p121212_obs(3)-pcalc121212(3))^2 + (p121212_obs(4)-pcalc121212(4))^2 ) + ...
        ( (p12121212_obs(1)-pcalc12121212(1))^2 +
(p12121212_obs(2)-pcalc12121212(2))^2 +
(p12121212_obs(3)-pcalc12121212(3))^2 +
(p12121212_obs(4)-pcalc12121212(4))^2 +
(p12121212_obs(5)-pcalc12121212(5))^2 ) + ...
        ( (p1010_obs(1)-pcalc1010(1))^2 + (p1010_obs(2)-pcalc1010(2))^2 +
(p1010_obs(3)-pcalc1010(3))^2 ) + ...
        ( (p10101010_obs(1)-pcalc10101010(1))^2 +
(p10101010_obs(2)-pcalc10101010(2))^2 +
(p10101010_obs(3)-pcalc10101010(3))^2 +
(p10101010_obs(4)-pcalc10101010(4))^2 +
(p10101010_obs(5)-pcalc10101010(5))^2 ) + ...
        ( (p1414_obs(1)-pcalc1414(1))^2 + (p1414_obs(2)-pcalc1414(2))^2 +
(p1414_obs(3)-pcalc1414(3))^2 ) + ...
        ( (p14141414_obs(1)-pcalc14141414(1))^2 +
(p14141414_obs(2)-pcalc14141414(2))^2 +
(p14141414_obs(3)-pcalc14141414(3))^2 +
```

```
(p14141414_obs(4)-pcalc14141414(4))^2 +
(p14141414_obs(5)-pcalc14141414(5))^2 ) ) / 7;

    MSD7_18 = ( (p1212_obs(1)-pcalc1212(1))^2 + ...
    (p121212_obs(1)-pcalc121212(1))^2 + ...
    (p12121212_obs(1)-pcalc12121212(1))^2 + ...
    (p1010_obs(1)-pcalc1010(1))^2 + ...
    (p10101010_obs(1)-pcalc10101010(1))^2 + ...
    (p1414_obs(1)-pcalc1414(1))^2 + ...
    (p14141414_obs(1)-pcalc14141414(1))^2 ) / 7;
```

Out-of-sample prediction codes for SLk in two-player games are as follows.

SK10 function:

```
function F = SK10(input, a0, a1, a2, lambda,r, L)
A = [18; 10];
f = [a0; a1; a2];
p0 = zeros(length(A),1);
p0(1,1) = (1-r) + r/(length(A));
i = 2;
while (i<=length(A))
    p0(i,1) = r / (length(A));
    i = i + 1;
end
p = [p0 zeros(length(A),L)];
k = 1;
while (k<=L)
    x = 1;
    TempPay2 = zeros(length(A),1);
    while (x<=length(A))
        y = 1;
        TempPay = zeros(length(A), 1);
        while(y<=length(A))
        TempPay(y,1) = Payoff(x,y)*A(x,1)*p(y,k);
        y = y + 1;
    end
    TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
    x = x + 1;
```

```
end
t = 1;
while (t<=length(A))
        p(t,k+1) = TempPay2(t,1) / sum(TempPay2);
        t = t + 1;
end
k = k + 1;
end
ChoiceProb = p*f;
if input == A(1,1)
F = ChoiceProb(1,1);
elseif input == A(2,1)
F = ChoiceProb(2,1);
end
end
```

SK12 function:

```
function F = SK12(input, a0, a1, a2, lambda,r, L)
A = [18; 12];
f = [a0; a1; a2];
p0 = zeros(length(A),1);
p0(1,1) = (1-r) + r/(length(A));
i = 2;
while (i<=length(A))
        p0(i,1) = r / (length(A));
        i = i + 1;
end
p = [p0 zeros(length(A),L)];
k = 1;
while (k<=L)
        x = 1;
        TempPay2 = zeros(length(A),1);
        while (x<=length(A))
                y = 1;
                TempPay = zeros(length(A), 1);
                while(y<=length(A))
                TempPay(y,1) = Payoff(x,y)*A(x,1)*p(y,k);
                y = y + 1;
        end
```

```
        TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
        x = x + 1;
end
t = 1;
while (t<=length(A))
        p(t,k+1) = TempPay2(t,1) / sum(TempPay2);
        t = t + 1;
end
k = k + 1;
end
ChoiceProb = p*f;
if input == A(1,1)
F = ChoiceProb(1,1);
elseif input == A(2,1)
F = ChoiceProb(2,1);
end
end
```

SK14 function:

```
function F = SK14(input, a0, a1, a2, lambda,r, L)
A = [18; 14];
f = [a0; a1; a2];
p0 = zeros(length(A),1);
p0(1,1) = (1-r) + r/(length(A));
i = 2;
while (i<=length(A))
        p0(i,1) = r / (length(A));
        i = i + 1;
end
p = [p0 zeros(length(A),L)];
k = 1;
while (k<=L)
        x = 1;
        TempPay2 = zeros(length(A),1);
        while (x<=length(A))
                y = 1;
                TempPay = zeros(length(A), 1);
                while(y<=length(A))
                TempPay(y,1) = Payoff(x,y)*A(x,1)*p(y,k);
```

```
            y = y + 1;
        end
        TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
        x = x + 1;
    end
    t = 1;
    while (t<=length(A))
        p(t,k+1) = TempPay2(t,1) / sum(TempPay2);
        t = t + 1;
    end
    k = k + 1;
    end
    ChoiceProb = p*f;
    if input == A(1,1)
    F = ChoiceProb(1,1);
    elseif input == A(2,1)
    F = ChoiceProb(2,1);
    end
    end
```

MLEBASE function:

```
function loglike = MLEBASE(v)
a0 = v(1);
a1 = v(2);
a2 = v(3);
l = v(4);
r = 0;
L = 2;
%%
loglike = -( 64*log(SK12(18, a0, a1, a2, l, r, L)) + 16*log(SK12(12, a0, a1, a2, l, r, L)) +...
69*log(SK10(18, a0, a1, a2, l, r, L)) + 11*log(SK10(10, a0, a1, a2, l, r, L)) +...
55*log(SK14(18, a0, a1, a2, l, r, L)) + 25*log(SK14(14, a0, a1, a2, l, r, L)) ) / 80;
end
```

RunMLE function:

```
clear;
```

```
clc;
opts = optimset('fminsearch');
opts.Display = ('iter');
opts.TolX = 1.e-6;
opts.MaxFunEvals = 10000;
opts.MaxIter = 5000;
lb = [0 0 0 0];
ub = [1 1 1 inf];
A = [1 1 1 0];
b = [1.000];
x0 = [0.0521 0.8459 0.0639 0.14];
n = [];
[x,fval,exitflag,output] = fminsearchcon(@MLEBASE,x0,lb,ub,A,b,n,opts);
```

msd function:

```
L = 2;
r = 0;
a0 = x(1);
a1 = x(2);
a2 = x(3);
lambda = x(4);
f = [a0; a1; a2];
p12 = [64; 16] * (100/80);
p1212 = [56; 12; 12] * (100/80);
p121212 = [43; 37/3; 37/3; 37/3] * (100/80);
p12121212 = [48; 8; 8; 8; 8] * (100/80);
p10 = [69; 11] * (100/80);
p1010 = [62; 9; 9] * (100/80);
p10101010 = [57; 23/4; 23/4; 23/4; 23/4] * (100/80);
p14 = [55; 25] * (100/80);
p1414 = [49; 31/2; 31/2] * (100/80);
p14141414 = [30; 50/4; 50/4; 50/4; 50/4] * (100/80);
A_cell = {[18; 12],...
[18; 12; 12],...
[18; 12; 12; 12],...
[18; 12; 12; 12; 12],...
[18; 10],...
[18; 10; 10],...
[18; 10; 10; 10; 10],...
```

```
[18; 14],...
[18; 14; 14],...
[18; 14; 14; 14; 14] };
pcalc_cell = cell(1,10);
%------------------------------------------------------------
for l = 1:length(A_cell)

      A = A_cell{l};

      p0 = zeros(length(A),1);
      p0(1,1) = (1-r) + r/(length(A));
      i = 2;
      while (i<=length(A))
            p0(i,1) = r / (length(A));
            i = i + 1;
      end
      p = [p0 zeros(length(A),L)];

      k = 1;
      while (k<=L)
            x = 1;
            TempPay2 = zeros(length(A),1);
            while (x<=length(A))
                  y = 1;
                  TempPay = zeros(length(A), 1);
                  while(y<=length(A))
                  TempPay(y,1) = Payoff(x,y)*A(x,1)*p(y,k);
                  y = y + 1;
            end
            TempPay2(x,1) = exp(lambda*(sum(TempPay(:))));
            x = x + 1;
      end
      t = 1;
      while (t<=length(A))
            p(t,k+1) = TempPay2(t,1) / sum(TempPay2);
            t = t + 1;
      end
      k = k + 1;
end
pcalc_cell{l} = p*f * 100;
```

```
end
%------------------------------------------------------------

pcalc12 = pcalc_cell{1};
pcalc1212 = pcalc_cell{2};
pcalc121212 = pcalc_cell{3};
pcalc12121212 = pcalc_cell{4};
pcalc10 = pcalc_cell{5};
pcalc1010 = pcalc_cell{6};
pcalc10101010 = pcalc_cell{7};
pcalc14 = pcalc_cell{8};
pcalc1414 = pcalc_cell{9};
pcalc14141414 = pcalc_cell{10};

%------------------------------------------------------------

MSD10 = ( ( (p12(1)-pcalc12(1))^2 + (p12(2)-pcalc12(2))^2 ) + ...
    ( (p1212(1)-pcalc1212(1))^2 + (p1212(2)-pcalc1212(2))^2 +
(p1212(3)-pcalc1212(3))^2 ) + ...
    ( (p121212(1)-pcalc121212(1))^2 + (p121212(2)-pcalc121212(2))^2 +
(p121212(3)-pcalc121212(3))^2 + (p121212(4)-pcalc121212(4))^2 ) + ...
    ( (p12121212(1)-pcalc12121212(1))^2 + (p12121212(2)-pcalc12121212(2))^2
+ (p12121212(3)-pcalc12121212(3))^2 + (p12121212(4)-pcalc12121212(4))^2 +
(p12121212(5)-pcalc12121212(5))^2 ) + ...
    ( (p10(1)-pcalc10(1))^2 + (p10(2)-pcalc10(2))^2 ) + ...
    ( (p1010(1)-pcalc1010(1))^2 + (p1010(2)-pcalc1010(2))^2 +
(p1010(3)-pcalc1010(3))^2 ) + ...
    ( (p10101010(1)-pcalc10101010(1))^2 + (p10101010(2)-pcalc10101010(2))^2
+ (p10101010(3)-pcalc10101010(3))^2 + (p10101010(4)-pcalc10101010(4))^2 +
(p10101010(5)-pcalc10101010(5))^2 ) + ...
    ( (p14(1)-pcalc14(1))^2 + (p14(2)-pcalc14(2))^2 ) + ...
    ( (p1414(1)-pcalc1414(1))^2 + (p1414(2)-pcalc1414(2))^2 +
(p1414(3)-pcalc1414(3))^2 ) + ...
    ( (p14141414(1)-pcalc14141414(1))^2 + (p14141414(2)-pcalc14141414(2))^2
+ (p14141414(3)-pcalc14141414(3))^2 + (p14141414(4)-pcalc14141414(4))^2 +
(p14141414(5)-pcalc14141414(5))^2 ) )/10;

MSD7 = ( ( (p1212(1)-pcalc1212(1))^2 + (p1212(2)-pcalc1212(2))^2 +
(p1212(3)-pcalc1212(3))^2 ) + ...
```

```
    ( (p121212(1)-pcalc121212(1))^2 + (p121212(2)-pcalc121212(2))^2 +
(p121212(3)-pcalc121212(3))^2 + (p121212(4)-pcalc121212(4))^2 ) + ...
    ( (p12121212(1)-pcalc12121212(1))^2 + (p12121212(2)-pcalc12121212(2))^2
+ (p12121212(3)-pcalc12121212(3))^2 + (p12121212(4)-pcalc12121212(4))^2 +
(p12121212(5)-pcalc12121212(5))^2 ) + ...
    ( (p1010(1)-pcalc1010(1))^2 + (p1010(2)-pcalc1010(2))^2 +
(p1010(3)-pcalc1010(3))^2 ) + ...
    ( (p10101010(1)-pcalc10101010(1))^2 + (p10101010(2)-pcalc10101010(2))^2
+ (p10101010(3)-pcalc10101010(3))^2 + (p10101010(4)-pcalc10101010(4))^2 +
(p10101010(5)-pcalc10101010(5))^2 ) + ...
    ( (p1414(1)-pcalc1414(1))^2 + (p1414(2)-pcalc1414(2))^2 +
(p1414(3)-pcalc1414(3))^2 ) + ...
    ( (p14141414(1)-pcalc14141414(1))^2 + (p14141414(2)-pcalc14141414(2))^2
+ (p14141414(3)-pcalc14141414(3))^2 + (p14141414(4)-pcalc14141414(4))^2 +
(p14141414(5)-pcalc14141414(5))^2 ) )/7;

    MSD7_18 = ( (p1212(1)-pcalc1212(1))^2 + ...
    (p121212(1)-pcalc121212(1))^2 + ...
    (p12121212(1)-pcalc12121212(1))^2 + ...
    (p1010(1)-pcalc1010(1))^2 + ...
    (p10101010(1)-pcalc10101010(1))^2 + ...
    (p1414(1)-pcalc1414(1))^2 + ...
    (p14141414(1)-pcalc14141414(1))^2 ) / 7;
```

REFERENCES

Anderson, S. P., Goeree, J. K., & Holt, C. A. (1998). Rent seeking with bounded rationality: An analysis of the all-pay auction. *Journal of Political Economy, 106(4)*, 828-853.

Arad, A., & Rubinstein, A. (2012). The 1120 money request game: a level-k reasoning study. *The American Economic Review, 102(7)*, 3561-3573.

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

Capra, C. M., Goeree, J. K., Gomez, R., & Holt, C. A. (1999). Anomalous behavior in a traveler's dilemma? *American Economic Review*, 678-690.

Chong, J. K., Ho, T. H., & Camerer, C. (2014). *A Generalized Cognitive Hierarchy Model of Games.* Retrieved June 14, 2015, from http://faculty.haas.berkeley.edu/hoteck/papers/Chong-Ho-Camerer.pdf

Choo, L. C., & Kaplan, T. R. (2014). *Explaining Behavior in the "11-20" Game.* Retrieved June 14, 2015, from http://mpra.ub.uni-muenchen.de/52808/

Costa-Gomes, M., Crawford, V. P., & Broseta, B. (2001). Cognition and behavior in normal-form games: An experimental study. *Econometrica, 69(5)*, 1193-1235.

Costa-Gomes, M. A., & Crawford, V. P. (2006). Cognition and behavior in two-person guessing games: An experimental study. *The American economic review*, 1737-1768.

Costa-Gomes, M. A., Crawford, V. P., & Iriberri, N. (2009). Comparing models of strategic thinking in Van Huyck, Battalio, and Beil's coordination games. *Journal of the European Economic Association*, 7(2-3), 365-376.

Crawford, V. P., & Iriberri, N. (2007). Fatal attraction: Salience, naivete, and sophistication in experimental" Hide-and-Seek" games. *The American Economic Review*, 1731-1750.

Crawford, V. P., & Iriberri, N. (2007). Level-k Auctions: Can a Nonequilibrium Model of Strategic Thinking Explain the Winner's Curse and Overbidding in Private-Value Auctions?. *Econometrica, 75(6)*, 1721-1770.

Crawford, V. P. (2007). *Let's talk it over: Coordination via preplay communication with level-k thinking.* Retrieved June 14, 2015, from http://econweb.ucsd.edu/ v2crawford/LetsTalk13Aug07.pdf

Crawford, V. P., Gneezy, U., & Rottenstreich, Y. (2008). The power of focal points is limited: even minute payoff asymmetry may yield large coordination failures. *The American Economic Review*, 1443-1458.

Crawford, V. P., Kugler, T., Neeman, Z., & Pauzner, A. (2009). Behaviorally optimal auction design: Examples and observations. *Journal of the European Economic Association, 7(2-3)*, 377-387.

Goeree, J. K., & Holt, C. A. (2000). Asymmetric inequality aversion and noisy behavior in alternating-offer bargaining games. *European Economic Review, 44(4)*, 1079-1089.

Goeree, J. K., Holt, C. A., & Palfrey, T. R. (2002). Quantal response equilibrium and overbidding in private-value auctions. *Journal of Economic Theory, 104(1)*, 247-272.

Goeree, J. K., Holt, C. A., & Palfrey, T. R. (2003). Risk averse behavior in generalized matching pennies games. *Games and Economic Behavior, 45(1)*, 97-113.

Goeree, J. K., & Holt, C. A. (2004). A model of noisy introspection. *Games and Economic Behavior, 46(2)*, 365-382.

Goeree, J. K., Louis, P., & Zhang, J. (2013). *Noisy introspection in the "11-20" game.* Retrieved June 14, 2015, from https://www.uts.edu.au/sites/default/files/cpmd-GoereeLouisZhang2014.pdf

Ho, T. H., Camerer, C., & Weigelt, K. (1998). Iterated dominance and iterated best response in experimental" p-beauty contests". *American Economic Review*, 947-969.

Keynes, J. M. (1936). *The general theory of interest, employment and money.*

Lagarias, J. C., Reeds, J. A., Wright, M. H., & Wright, P. E. (1998). Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM Journal on optimization, 9(1)*, 112-147.

Lieberman, B. (1960). Human behavior in a strictly determined 3x3 matrix game. *Behavioral Science, 5(4)*, 317-322.

McKelvey, R. D., & Palfrey, T. R. (1992). An experimental study of the centipede game. *Econometrica: Journal of the Econometric Society*, 803-836.

McKelvey, R. D., & Palfrey, T. R. (1995). Quantal response equilibria for normal form games. *Games and economic behavior, 10(1)*, 6-38.

Nagel, R. (1995). Unraveling in guessing games: An experimental study. *The American Economic Review*, 1313-1326.

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal, 7(4)*, 308-313.

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

Rapoport, A., & Boebel, R. B. (1992). Mixed strategies in strictly competitive games: a further test of the minimax hypothesis. *Games and Economic Behavior, 4(2)*, 261-283.

Rogers, B. W., Palfrey, T. R., & Camerer, C. F. (2009). Heterogeneous quantal response equilibrium and cognitive hierarchies. *Journal of Economic Theory, 144(4)*, 1440-1467.

Stahl, D. O., & Wilson, P. W. (1995). On players' models of other players: Theory and experimental evidence. *Games and Economic Behavior, 10(1)*, 218-254.

Van Huyck, J. B., Battalio, R. C., & Beil, R. O. (1990). Tacit coordination games, strategic uncertainty, and coordination failure. *The American Economic Review*, 234-248.

Van Huyck, J. B., Battalio, R. C., & Beil, R. O. (1991). Strategic uncertainty, equilibrium selection, and coordination failure in average opinion games. *The Quarterly Journal of Economics*, 885-910.

Wright, J. R., & Leyton-Brown, K. (2013). *Predicting Human Behavior in Unrepeated, Simultaneous-Move Games.* Retrieved June 14, 2015, from http://arxiv.org/abs/1306.0918