Available online at www.sciencedirect.com

ScienceDirect

Games and Economic Behavior 57 (2006) 93–122

GAMES and Economic Behavior

www.elsevier.com/locate/geb

An initial implementation of the Turing tournament to learning in repeated two-person games [☆]

Jasmina Arifovic [a,*], Richard D. McKelvey [†], Svetlana Pevnitskaya [b]

[a] *Department of Economics, Simon Fraser University, 8888 University Drive, Burnaby, BC V5A 1S6, Canada*
[b] *Department of Economics, Florida State University, Tallahassee, FL 32306-2180, USA*

Received 25 March 2003

Available online 5 June 2006

Abstract

We report on a design of a Turing tournament and its initial implementation to learning in repeated 2-person games. The principal objectives of the tournament, named after the original Turing Test, are (1) to find learning algorithms (*emulators*) that most closely simulate human behavior, (2) to find algorithms (*detectors*) that most accurately distinguish between humans and machines, and (3) to provide a demonstration of how to implement this methodology for evaluating models of human behavior. In order to test our concept, we developed the software and implemented a number of learning models well known in the literature and developed a few detectors. This initial implementation found significant differences in data generated by these learning models and humans, with the greatest ones in coordination games. Finally, we investigate the stability of our result with respect to different evaluation approaches.

JEL classification: D83; C63; C92; H41

Keywords: Turing tournament; Emulators; Detectors; Learning models; Repeated games

[☆] The journal is grateful to the authors for contributing this article in honor of Richard McKelvey to *Games and Economic Behavior*. It was submitted originally for publication in the special issue in honor of Richard D. McKelvey (Vol. 51, Issue 2, May 2005) and was under review when the special issue went to press.

[*] Fax: +1 (604) 291 5944.
E-mail addresses: arifovic@sfu.ca (J. Arifovic), spevnitskaya@fsu.edu (S. Pevnitskaya).

[†] We regret to say that Richard D. McKelvey passed away in April 2002. All errors in completing the work are ours.

0899-8256/$ – see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.geb.2006.03.013

1. Introduction

Studying the dynamic of human learning and adaptation is an area of social science that has generated a lot of research interest. There are many models attempting to explain how individuals learn in various game theoretic settings. We can begin with two of the classical models—*Fictitious Play* and *Cournot Best Reply* (see Boylan and El-Gamal, 1993, for an experimental evaluation of these models). Crawford (1991, 1995) considers *Evolutionary models*. Stahl (1996, 1999) explores boundedly rational rules. Roth and Erev (1995, 1998, 1999) and others use a *Reinforcement Learning* model to explain learning in repeated stage games. Camerer and Ho (1999a, 1999b) develop the *Experience Weighted Attraction* (*EWA*) models. All of the above models have been evaluated by using standard econometric methods (maximum likelihood or grid search) to fit the models to experimental data. Using these methods, one can get estimates of the parameters of the models, test various hypotheses about these parameters and compare models to each other.

Arifovic and McKelvey (2002) describe a methodology for evaluating models of human behavior in social sciences based on the idea of the Turing test (Turing, 1950).[1] They propose a two sided tournament, called a *Turing Tournament*, which would solicit machine algorithms, called *emulators*, that mimic human behavior and machine algorithms, they call *detectors*, that are designed to distinguish between human behavior and that generated by the emulators.

In the tournament, the emulators generate data sets with information on actions of computer agents in a given environment. The human behavior is represented by data sets generated in the experiments with human subjects in the same environment. The detectors are then presented with all the data sets, both those generated by emulators and by experiments with human subjects, and try to distinguish between machine and human data sets. They do so by assigning a probability that a given data set is human rather than machine generated. Each detector gets a score based on how close its decisions are to the true state. The detector that obtains the highest score is a winner of the tournament. The winning emulator is a machine algorithm to which the winning detector assigns the highest probability of being human.

The tournament is designed along the lines of the original Turing test. Both tests try to answer the question "Man or machine?" However, the major difference in Arifovic and McKelvey's modification is in the evaluation stage. Researchers in the field of experimental economics and (or) behavioral economics who deal both with human and machine generated data would probably acknowledge that they can often tell the difference between human and machine data when they see the data charts. The way to implement this knowledge, and the real challenge of the tournament, is to develop good performing detectors. That is why in the tournament's design, instead of humans trying directly to assess whether some output is generated by the machine or human, this role is assigned to detectors.

Arifovic and McKelvey suggest a number of environments where the implementation of the Turing tournament could be interesting. In this paper we describe an implementation of the Turing tournament to a class of repeated games with full information. Since this is a new concept

[1] In a famous paper in 1950, Turing (1950) addressed the question of determining when machines can "think." His proposal was to replace this question with the more manageable question of when a machine can mimic human behavior. Turing's answer was the so-called *Turing Test*: a machine is sufficiently human when a third party can not distinguish between the behavior of the machine and a human. In Turing's version, the third party is a human interrogator who is allowed to ask whatever questions he or she wants to both the machine and a human. Both the machines and humans have their answers put on tape for the interrogator to read.

of evaluation of models' performance, we first test our design of the tournament. In this paper, we report the initial results that we have obtained.

We *submitted* to the tournament the source code for the programs of several emulators (a number of well-known learning algorithms that have been extensively studied in the literature). We also *submitted* the source code for several relatively simple detectors. These detectors compute various measures using presented data sets, such as closeness to Nash equilibrium, closeness to payoff dominant outcome, changes in players' payoffs over time etc. Based on the values of these measures, detectors assign a probability that a particular data set is human. For this initial tournament, we used experimental data collected by McKelvey and Palfrey (2002) under various information and matching conditions. For our purposes, we used the data generated in full information, fixed matching treatment. We choose a fixed matching protocol in order to move the research agenda a step further, towards development of the algorithms that behave well in the repeated game theoretic framework. The development of the new algorithms better suited for repeated games has taken place simultaneously with the initial implementation described in this paper. Camerer et al. (2002) introduced Sophisticated EWA model. McKelvey and Palfrey (2002) developed strategic learning models for repeated games and introduced Strategic EWA learning. Hanaki et al. (2005) developed algorithm for learning in repeated games that was directly motivated by results of this implementation of the Tournament.

The games we consider are: Ochs Game, Stag Hunt, Ultimatum Game, Centipede Game, Prisoner's Dilemma, Battle of Sexes and the Game of Chicken. We then generated machine data sets for the above games using the programs developed for various emulators. The main emulators that we implemented include Fictitious Play, Cournot Best Reply, Adjusted Reinforcement, and Experience Weighted Attraction Learning. We also had several variants of mixed models where players were using different emulators to make their decisions. Our initial simulations (for specific parameter values of learning algorithms) show that there are often significant differences between human and machine generated data. These results serve as further motivation to conduct the Tournament and invite submissions of sophisticated emulators and detectors.

When very accurate detectors are established through the Tournament, evaluation of the learning models would not require presence of human data under identical conditions as in statistical evaluation. The performance of the models can be studied under various conditions prior to collecting human data and, in fact, used in developing new experimental designs. In addition, since many detectors use a particular *logic* in their decision (for example, the ability to coordinate), results from a particular detector would allow evaluation of specific features of the model.

We would like to emphasize that this implementation of emulators and detectors was for primarily for testing purposes. While we tried to implement a number of well-known learning algorithms, we used the parameter sets reported in the literature, but did not try to get the *optimal* sets for each of the games. In addition, our detectors represent just an initial attempt to tackle the problem of developing those types of algorithms. The objective of the paper is to present the methodology that can be used to test the models of human behavior in a setup of repeated 2-person games. The results that we present here are obtained based on a single run of the tournament. In the subsequent stages, the tournament will run iteratively until, in statistical terms, significant detector and emulator are identified.

We give a description of the Turing tournament design as we apply it to repeated games' environments in Section 2. A description of an initial implementation of the Tournament is given in Section 3. We report the results of the tournament and some comparison between human and machine generated data in Section 4. Concluding remarks are given in Section 5.

2. Details of the Turing tournament

In this section, we describe an implementation of the Turing tournament methodology to repeated normal form games. We have two categories of entries, both of which are computer programs. Emulators are computer programs that simulate human players' behavior in a class of repeated games. Detectors are programs that assign probabilities of whether a given data set that they encounter is machine or human generated.

2.1. Definitions

Define a *two person normal form game* to be a file containing an $S_1 \times S_2 \times 2$ array of real numbers u_{mlp} $(m = 1, \dots, S_1;\ l = 1, \dots, S_2;\ p = 1, 2)$. This represents the stage game in an N round repeated 2-person game, where player p has S_p strategies, and u_{mlp} is the payoff to player p if player 1 adopts strategy m and 2 adopts l.

Define a *data set* to be a file containing a $Q \times N$ matrix of integers. This represents the result of running an N round repeated game experiment with Q subjects, matched in pairs (1 matched with 2, 3 with 4, etc.). The (q, n)th entry in the data set is the strategy choice of player q in round n.

2.2. Valid entries

An *emulator* is a computer program that takes as input a normal form game, and generates as output a data set.

A *detector* is a computer program that takes as input a normal form game, together with a set of D data sets, and gives as output a vector r of length D, $r = (r_1, , \dots, r_d, \dots, r_D)$, where each component r_d is between 0 and 1 and represents the probability that the detector assigns to a data set $d \in \{1, D\}$ being human.

2.3. Stages of the tournament

Stage 1. After all entries have been submitted, a set of J normal form games is selected. For each normal form game, H data sets are generated based on human play (by running H experiments with human subjects).

Stage 2a. For each normal form game, each of the submitted emulators generates one data set. If there are E emulators, this gives a total of $D = H + E$ data sets.

Stage 2b: For each selected normal form game, each detector k is run with input consisting of the normal form game j, and the set of D data sets, one at the time. In other words, a detector works with a single data set, assesses probability and is then presented with another data set.[2] This yields an output vector r^{kj} of length D for each detector k and normal form game j. For $1 \leqslant j \leqslant J$ define z_d^{kj} to be r_d^{kj} if the d^{th} data set is human, and $1 - r_d^{kj}$ otherwise. The score for detector k is

$$v_k = \frac{1}{DJ} \sum_{d,j} \log\bigl(z_d^{kj}\bigr).$$

[2] It does not work with a combination of data sets.

The score of emulator e (which generates one of D data sets for each game) against detector k is

$$v_k^e = \frac{1}{J} \sum_j \log\left(z_e^{kj}\right).$$

Stage 3. We repeat Stage 2 (a and b) T times,[3] obtaining average scores for each detector, $\bar{v}_k$, and emulator, $\bar{v}_k^e$:

$$\bar{v}_k = \frac{1}{T} \sum_t v_{kt},$$

where v_{kt} is the value of v_k obtained on the tth trial, and

$$\bar{v}_k^e = \frac{1}{T} \sum_t v_{kt}^e,$$

where v_{kt}^e is the value of v_k^e on the tth trial. The value of T depends on the results reported at each iteration t of the tournament. We stop the tournament at the iteration T when the best detector and the best emulator are significantly different (at 95% confidence level) from the second placed detector and the second placed emulator.

Stage 4. The detector k^* with the highest score $\bar{v}_k$ is declared to be the winning detector. The winning emulator is defined to be the emulator e^* that minimizes $\bar{v}_{k^*}^e$.[4]

We chose the logarithmic scoring rule described above because it is a strictly proper scoring rule, i.e. one in which a forecaster maximizes (optimizes) by forecasting/revealing exactly his or her true beliefs about the situation.[5] In addition, logarithmic scoring rule has increasing "punishment" for marginal error, compared to scoring rules based on absolute deviations from the true state. In the Tournament, if a detector makes a big error with one of the data sets, it received a very large negative score (logarithm of almost zero) and therefore is unlikely to become the winner. With scoring rules based on absolute deviations, a detector can perform poorly in one case, but still receive a very high score by performing well with other data sets (for example other types of games).

Prior to the beginning of the Tournament each entrant knows the details of the tournament as described above, including scoring rules for detectors and emulators. Specifically, each entrant knows that there will be a series of two person normal form games with given N and Q. Entrants do not know how many human data sets, H (determined exogenously by the researcher), or emulator generated data sets, E (determined endogenously based on submissions), there will be in the Tournament. The complete instructions for submissions can be found in the Supplementary Appendix.

[3] In the initial implementation that is described in this paper, we ran the Tournament only once, i.e. $T = 1$.

[4] In the unlikely event of a tie among two or more detectors, in terms of their expected scores, at 0.05 significance level, the detectors will share the prize. Similarly, if there is a tie among two or more emulators in terms of their expected scores against the winning detector, at 0.05 significance level, the prize will be shared among them. Detailed Tournament instructions can be found as electronically available supplementary material.

[5] For a thorough discussion and proofs of proper scoring rules we refer the reader to many articles in the literature like Savage (1971), O'Carroll (1977), Winkler (1969) and others. There are also many examples of applications of proper scoring rules in statistics, economics and meteorology, for example, Gneiting and Raftery (2004), Hanson (2002), Staël von Holstein (1970) and others.

2.4. Evaluation approaches

The purpose of the Tournament is to identify an emulator that most closely resembles human behavior and find a detector that is most accurate in distinguishing humans from algorithms. We next comment on different approaches for identifying the best detector and the best emulator based on the scores they generate. While theorems in social choice theory argue that there is no perfect aggregation rule, we find it instructive to report ranking of emulators and detectors using a few different weighting rules.

2.4.1. Scoring of detectors

Our working scoring scheme, the *baseline scoring scheme*, described earlier is to assign each detector an average score over all human and emulator entries (the emulator/human scores are in turn averaged across games). The advantage of this approach is that no data on performance is wasted. The disadvantage is that the score becomes sensitive to outliers. Given the logarithmic score rule used, all scores have an upper bound of zero and a lower bound of minus infinity.[6] Therefore when a detector makes one big error (e.g. assigns a probability 1 of being human to an emulator), it is sufficient to significantly decrease the detector's score even if it performed well in other cases.

Depending on the objectives in selecting the best detector, the disadvantage described above may serve as a rationale for another scoring scheme. Suppose that we are interested in the best *worst case* detector, i.e. we adopt a minimax choice strategy. The best detector then is the one that makes the minimum worst error across all data sets. In a way this scoring scheme ensures against large deviations.

A third scoring scheme we consider to determine the best detector is based on a median game performance. A *median game* rule finds the average score for each game for a detector and then picks the median of the games' average scores for a given detector to find its "median game score."[7] This approach eliminates disadvantages of the first approach—sensitivity to outliers. It also eliminates the effect of the particularly good score in situations when a detector performs exceptionally well in just one or two specific games. On the other hand, a drawback of this scoring scheme is that most of data on performance is not utilized. Many other weighting schemes could be applied. However, the above described when taken together should give a good characterization of detectors' performance as they allow us to identify sensitivity of detectors to outliers, as well as their average and median game performance.

2.4.2. Scoring of emulators

Our *working* approach in evaluating the emulators in based on the scores they receive against the best detector. The best emulator then is the one that the best detector assigns the highest probability of being human. An advantage of this scoring scheme is that the most accurate detector is used. Note that since we know the true state of a data set (human or emulator generated), the detectors are evaluated objectively based on the "true state of nature." It therefore seems advantageous to rank emulators based on the most accurate rule. However this scoring scheme may have a disadvantage. If a detector uses a specific "logic" in deciding on the probability of being human, then an emulator with a the same "logic" would be more likely to win.

[6] In the computer implementation of our scoring rule, if $z_d^{kj} = 0$, then we assign a large, in absolute terms, negative constant as a score for a detector for a given data set.

[7] In case of the even number of games the rule takes the average of two games.

An alternative scoring scheme, *unweighted mean*, is to use all detectors equally, so that the best emulator is the one that has the best average score across all detectors. However in this case even “poor” detectors participate equally in assigning the winner, which biases the outcome.

An approach that decreases this problem, *weighted mean*, is weighting detectors according to their performance. We implement weighting procedure in the following way. Suppose that each detector k gets a score v_k, where $v_{\min}$ is the score of the worst detector. Then each detector k is assigned the following weight

$$w_k = \frac{v_k - v_{\min}}{\sum_k (v_k - v_{\min})}.$$

The worst detector is assigned a weight of zero and all other detectors are assigned weights proportional to their performance. Poor detectors still participate in the scoring, but their impact is smaller than in the second scheme. Although the relation of this scoring scheme to our working scheme is less trivial, there are cases when the working scheme is clearly superior to this weighting. For example suppose that there are three categories of detectors. One detector performs poorly and has a score $v_{\min}$, the second category is a number of detectors C, each with the score v_2 (close to $v_{\min}$) and the third category is one detector with the highest score v_3. Then emulator i is assigned a score

$$\frac{C(v_2 - v_{\min})}{C(v_2 - v_{\min}) + v_3 - v_{\min}} v_2^i + \frac{(v_3 - v_{\min})}{C(v_2 - v_{\min}) + v_3 - v_{\min}} v_3^i.$$

For $C > (v_3 - v_{\min})/(v_2 - v_{\min})$, greater weight is assigned to less accurate detectors which decreases the accuracy of evaluation. Therefore, when the number of poor detectors is large, a better evaluation method is to determine the winning emulator based on the best detector.

The fourth approach we study is the *median game* approach, which finds the mean score for each game for an emulator and then picks the median of the game scores to determine the winner. We provide empirical illustration of these evaluation approaches in Section 4 of the paper.

The design of the Tournament is such that evaluation of emulators relies on the detection ability of detectors. Since detectors are evaluated objectively based on the “true state” (human or machine) of all data sets, human data indirectly affects scores given to emulators. If all submitted detectors are not accurate in distinguishing human and machine data, the scores of emulators have little use for conclusions since emulators are evaluated against the best detector. To avoid this situation we need to introduce a reference score that the best detector needs to exceed to be the winner. In this implementation our reference is the score of a detector that assigns all data sets probability 0.5 of being human. This detection is equivalent to “I do not know” conclusion and we call this detector “Random.” If the best submitted detector receives a higher than this threshold score, its detection ability is reasonably accurate and therefore emulators’ scores against the best detector are insightful.

3. A preliminary version and test

We report the results of the initial implementation of the Turing Tournament conducted at the California Institute of Technology. During the initial phase, we set up a tournament using experimental data collected by McKelvey and Palfrey (2002) on repeated two-person games. Eight two-person games were used. Table 1 gives a listing of the games, and Table 2 provides a listing of the set of all Nash Equilibria for each game.

Table 1
List of stage games used in the first phase test

Game 1
Ochs Game

	L		R	
U	21	3	3	5
D	3	5	5	3

Game 2
2 × 2 Stag Hunt

	L		R	
U	3	3	3	1
D	1	3	6	6

Game 3
3 × 3 Stag Hunt

	L		M		R	
U	3	3	3	2	3	1
M	2	3	4	4	4	3
D	1	3	3	4	5	5

Game 4
2 × 2 Ultimatum

	L		M		R	
U	3	3	3	3	3	3
M	4	2	4	2	0	0
D	5	1	0	0	0	0

Game 5
3 × 3 Centipede

	L		M		R	
U	4	1	4	1	4	1
M	2	8	16	4	16	4
D	2	8	8	32	64	16

Game 6
Prisoner’s Dilemma

	L		R	
U	8	8	1	9
D	8	8	1	9

Game 7
Battle of Sexes

	L		R	
U	18	6	3	3
D	3	3	6	18

Game 8
Chicken

	L		R	
U	5	5	2	6
D	6	2	1	1

We used only the experiments corresponding to full information repeated games. This consisted of two experiments with Pasadena City College (PCC1 and PCC2) subjects and one experiment with California Institute of Technology (CIT) subjects. There were 16 subjects in each experiment (except one with 14 subjects). Each experiment consisted of eight sessions, in each of which subjects were matched in pairs and played one of the eight games above for 50 rounds with the same player (subjects made moves simultaneously). At the beginning of the session, subjects were told the game matrix. After each round, subjects were told the choice of the subject they were matched with, and the payoffs received by each subject. In each new session, a different game was used, and subjects were re-matched using a *zipper* design, i.e. each subject played with the same partner only once (except in the experiment with 14 subjects, in which it was necessary to rematch once the players that had already played in one of the sessions.)

3.1. Emulators

We started the evaluation of the existing learning models by implementing the following algorithms as *emulators*: *Random*, *Cournot*, *Fictitious Play*, *Adjusted Reinforcement* (Roth and Erev, 1998, 1999) *EWA* (Camerer and Ho, 1999a, 1999b). We also created a number of emulators consisting of *mixed* models where different players made decisions according to different learn-

Table 2
Nash equilibria of the games

	Game	Nash equilibria		Payoffs	
		Row	Column	Row	Column
1	Ochs	$(\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{10}, \frac{9}{10})$	$\frac{24}{5}$	4
2	2 × 2 Stag Hunt	$(1, 0)$	$(1, 0)$	3	3
		$(0, 1)$	$(0, 1)$	6	6
		$(\frac{6}{10}, \frac{4}{10})$	$(\frac{6}{10}, \frac{4}{10})$	3	3
3	3 × 3 Stag Hunt	$(1, 0, 0)$	$(1, 0, 0)$	3	3
		$(0, 1, 0)$	$(0, 1, 0)$	4	4
		$(0, 0, 1)$	$(0, 0, 1)$	5	5
		$(\frac{1}{2}, \frac{1}{2}, 0)$	$(\frac{1}{2}, \frac{1}{2}, 0)$	3	3
		$(\frac{1}{2}, 0, \frac{1}{2})$	$(\frac{1}{2}, 0, \frac{1}{2})$	3	3
		$(0, \frac{1}{2}, \frac{1}{2})$	$(0, \frac{1}{2}, \frac{1}{2})$	4	4
4	3 × 3 Ultimatum	$(1, 0, 0)$	(q_1, q_2, q_3)[a]	3	3
		$(0, 1, 0)$	(q_1, q_2, q_3)[b]	4	2
		$(0, 0, 1)$	$(1, 0, 0)$	5	1
5	3 × 3 Centipede	$(1, 0, 0)$	(q_1, q_2, q_3)[c]	4	1
6	Prisoner's Dilemma	$(0, 1)$	$(0, 1)$	2	2
7	Battle of Sexes	$(1, 0)$	$(1, 0)$	18	6
		$(0, 1)$	$(0, 1)$	6	18
		$(\frac{5}{6}, \frac{1}{6})$	$(\frac{1}{6}, \frac{5}{6})$	$\frac{11}{2}$	$\frac{11}{2}$
8	Chicken	$(1, 0)$	$(1, 0)$	2	6
		$(0, 1)$	$(0, 1)$	6	2
		$(\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, \frac{1}{2})$	$\frac{7}{2}$	$\frac{7}{2}$

[a] $(q_1, q_2, q_3) \in Co[(0, 0, 1), (\frac{3}{5}, \frac{3}{20}, \frac{1}{4}), (\frac{3}{5}, 0, \frac{2}{5}), (0, \frac{3}{4}, \frac{1}{4})]$.

[b] $(q_1, q_2, q_3) \in Co[(0, 1, 0), (\frac{4}{5}, \frac{1}{5}, 0)]$.

[c] $(q_1, q_2, q_3) \in Co[(1, 0, 0), (\frac{6}{7}, \frac{1}{7}, 0), (\frac{30}{31}, 0, \frac{1}{31}), (\frac{6}{7}, \frac{6}{49}, \frac{1}{49})]$.

ing models. We included two new algorithms in the Tournament, *Alg1*—a coordination based algorithm written by Svetlana Pevnitskaya, and *Alg2*—a Quantal Response Equilibrium, *QRE* (McKelvey and Palfrey, 1995) based algorithm written by Brian Rogers.

We would like to emphasize that (with the exception of Alg1 and Alg2 that were written while we were developing the tournament software) the learning models that we implemented were not specifically developed for repeated games with full information. Camerer et al. (2002) Sophisticated EWA model as well as models by McKelvey and Palfrey (2002) and Hanaki et al. (2005) would be excellent entrants to the Tournament since they allow for repeated games. However, these models were not available at the time of this initial implementation.

The *emulators* were implemented as follows:

Random Play: Each choice available to the player is assigned the same probability. For $n \times n$ game, the probability of choosing j is $p_j = \frac{1}{n}$ for any j. There is no updating.

Fictitious Play: Players choose a best response to the historical average of past play of their opponents. The first round choice is random.

Cournot Best Reply: A player chooses best response to the previous move of the other player. The first round choice is random.

Adjusted Reinforcement: Players adopt mixed strategies so that the probability that player i will play strategy j at time t is

$$p_j^i(t) = \frac{q_j^i(t)}{\sum_j q_j^i(t)},$$

where $q_j^i(t)$ is the attraction of strategy j to player i at time t. The updating rule for attractions is

$$q_j^i(t+1) = (1-\varphi)q_j^i(t) + E_k(j, x - x_{\min}),$$

where φ is the "forgetting" (or recency) parameter, x is received payoff, $x_{\min}$ is the smallest possible payoff and

$$E_k(j, x - x_{\min}) = \begin{cases} (x - x_{\min})(1-\varepsilon) & \text{if } j = k, \\ (x - x_{\min})\varepsilon\frac{1}{m-1} & \text{otherwise} \end{cases}$$

is a function which determines how the experience of playing strategy k and receiving reward $x - x_{\min}$ is generalized to update each strategy j. Here m is the number of pure strategies.

We defined initial attraction of strategy j, $q_j(1)$, to be the expected value of playing j assuming that the other player chooses randomly. In the *pure* Adjustment Reinforcement algorithm, we used $\varphi = 0.1$ and $\varepsilon = 0.1$. These values are based on Roth and Erev's (1998) results where they estimated $\varphi = 0.1$ and $\varepsilon = 0.2$ and the acceptable range of parameter values as $0 < \varphi < 0.2$ and $0.02 < \varepsilon < 0.3$.[8]

Experience Weighted Attraction: There are two main variables that are updated after each round of experience: $N(t)$—the number of *observation-equivalents* of past experience; and $A_i^j(t)$—player i's attraction of strategy s_i^j after period t has taken place, $N(t)$ and $A_i^j(t)$ begin with some prior values, $N(0)$ and $A_i^j(0)$. These prior values can be thought of as reflecting pre-game experience. The experience weight starts at $N(0)$ and is updated according to

$$N(t) = \rho N(t-1) + 1, \quad \text{for } t \geqslant 1$$

where ρ is a depreciation rate or retrospective discount factor. The attraction of a strategy j for player i at time t is updated according to

$$A_i^j(t) = \{\phi N(t-1)A_i^j(t-1) + [\delta + (1-\delta)I(s_i^j, s_i(t))]\pi_i(s_i^j, s_{-i}(t))\}/N(t).$$

The factor ϕ is a discount factor or decay rate, which depreciates previous attraction and $\pi_i(s_i^j, s_{-i}(t))$ is a payoff to player i of choosing j, given the choice of other player(opponent) at time t. Finally, $I(s_i^j, s_i(t))$ is an indicator function that equals 1 when $s_i^j = s_i(t)$ and 0 otherwise. The probabilities are updated using the logistic form

$$p_i^j(t+1) = \exp[\lambda A_i^j(t)] \Big/ \sum_k \exp[\lambda A_i^k(t)].$$

[8] The values of the parameters were not estimated for our class of games. Some data generated by this implementation of the Adjusted Reinforcement Learning model might not reflect the performance of the model with optimized parameters.

Besides the payoff matrix, the inputs of this emulator are: $N(0)$, $A^j(0)$, ρ, ϕ, δ, and λ. The initial attractions $A^j(0)$ are determined in the same way as in the Adjusted Reinforcement model. For the "pure" version of the emulator we used $N(0) = 10$, $\rho = 0.95$, $\phi = 0.99$, $\delta = 0.2$, $\lambda = 0.35$. In mixed models we also used the following values of the parameters: $\rho = 0.95$, $\phi = 1$, $\delta = 0.6$, $\lambda = 0.2$, $N(0) = 15$ and $\rho = 0.94$, $\phi = 1$, $\delta = 0.4$, $\lambda = 0.4$, $N(0) = 16$. These values are based on Camerer and Ho (1999b) estimation results where they obtain for different games $\rho = \{0.961, 0.946, 0.935, 0.926\}$, $\phi = \{1.040, 1.005, 0.986, 0.991\}$, $\delta = \{0, 0.73, 0.413, 0.547\}$, $\lambda = \{0.508, 0.182, 0.646, 0.218\}$, $N(0) = \{19.63, 18.391, 15.276, 9.937\}$.[9]

Alg1: [10]The players first "identify" a game, analyzing the payoffs of the opponent as well as their own. The main objective is to look for coordination games, and symmetric, Pareto superior payoffs. Then players try to coordinate. If the opponent deviates, the player uses "punishment" to try to enforce coordination.

Alg2: [11]This algorithm works in the following way. If there is a unique Nash equilibrium in mixed strategies, then play according to the Quantal Response Equilibrium (QRE).[12] Otherwise, if maximum payoffs for both players are in the same cell, they play this with probability, γ which is chosen randomly, from the uniform distribution, in the interval [0.8, 0.99].

We also implemented a number of emulators consisting of *mixed* models—i.e., emulators where different players played according to decisions rules specified by different learning models. It is worthwhile pointing out that in the design of the Tournament, there is no restriction on the parameter values that emulators are allowed to use for any particular setup. In other words, once they are presented with the game payoff matrix, emulators are allowed to set (and change) the parameter values of the algorithms they are using.

We generated the emulators' data sets in the following way. For each emulator, we generated 16 robots (this corresponds to the number of human subjects who participated in the experiments) who used a particular emulator to play the game. The matching of robots who played a repeated game and their re-matching after each game was performed following the experimental *zipper* design described above. After each round of a given repeated game, robots received the same information as the human subjects did, i.e. what the strategy choice of the robot they were matched with was. A separate data file was generated for each game.

3.2. Detectors

Detectors reported in this section serve only as an initial attempt to identify the differences in human and emulator behavior. We implemented the following six *detectors*:

[9] Again as in the case of the Adjusted Reinforcement Learning, the values of the parameters were not estimated for our class of games. Some data generated by this implementation of EWA model might not reflect the performance of the model with optimized parameters.

[10] Due to S. Pevnitskaya.

[11] Due to B. Rogers.

[12] Gambit program (http://www.hss.caltech.edu/gambit/Gambit.htm) is used to solve a for QRE. The QRE parameter λ is chosen randomly, from the log-normal distribution, with the distribution parameters chosen such that the mode of the distribution is equal to 2.

Random (Zero Information): Each data set is equally likely to be human or machine—i.e., assigned a probability 0.5.

Dates: Data sets are split in two groups according to the time-stamp they have. The smaller groups are assigned to be human with high probability.[13]

Fast Convergence: This detector assumes that humans converge to a particular play faster that emulators. It assigns a probability of $\rho = 0.7$ that a data set is human if both players choose the same move more than 90% of the time. Otherwise, it assigns a probability of $\rho = 0.3$.

Payoff Change: This detector looks for payoff differences between the beginning and end of playing the game: humans are more likely to have greater difference in payoffs in the beginning and end of the game than machines. We find the average of pair i's payoffs in the first and last 12 rounds and compute the difference d_i. Then compute average of d_i over all pairs for data set k, D_k. The ranking of D_k's is mapped into probability interval of being a human.

High Payoffs: This detector exploits the difference between human and machine generated data in terms of the average payoffs earned during a given game. Humans are more likely to get higher payoffs. The detector finds average subject's payoff for each data set and ranks them. This ranking is mapped into the probability interval.

Coordination Detector:[14] This test identifies a coordination game and counts the number of alternations between optimal cells, *Count*. If the total possible number of alternations is J, then the probability that a data set is human is $1 - \frac{|Count - \varepsilon J|}{J}$.[15]

4. Results

In this section we first present the comparison between the human and emulator generated behavior. We then report the results of the Initial Turing Tournament.

4.1. Overview of human and emulator behavior

Figures 1–8 compare the human to the machine data for four of the *emulators:* Cournot Best Reply, Fictitious Play, Adjusted Reinforcement, and EWA. Polygons in these figures represent the possible per stage average payoff space. The row player's payoffs are on the horizontal axis, and the column player's payoffs are on the vertical axis. Each data point represents the average payoff over the course of the 50 rounds for a pair of subjects who are matched together. The red dots represent the human data, and the black dots represent the *emulator* data. In each figure, pink circles represent Nash equilibria of a stage game.

[13] This detector was added just for the purpose of testing how our software worked. In the current version of the Tournament, this detector would not be successfully implemented as all the data sets, machine and human, that detectors look at have the same data stamp.

[14] Due to S. Pevnitskaya.

[15] This test appears to be accurate in distinguishing humans from machines in coordination games. In the initial Tournament, each file contained data for all eight games and this test was the winner of the Tournament with the score of 16.5 out of 18. However it looked primarily at coordination game and ignored the rest of the data (since the detectors were not required to assign a probability for each game). This result motivated us to change the structure of the Tournament such that a *detector* has to assign probability to data from each game.

The figures reveal that the Cournot Best Response and the Fictitious Play usually get to one of the equilibria of a given game. This rarely coincides with the data points generated by humans. At the same time, Adjusted Reinforcement and EWA have scattered data points that are not close in many cases to the clusters of human data observations. They also rarely get close to the region of the payoff space where equilibrium points are located. In order to investigate this further, we introduce a statistical measure that we call a *center estimate*. This is the point, $(\bar{x}, \bar{y})$ in the payoff space which minimizes the sum of Euclidean distances to all data points for the emulator or human data

$$\min_{\bar{x},\bar{y}} \sum_i \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2}$$

where x_i and y_i are payoffs to the row and the column players (who are matched together in a pair i) respectively and $(\bar{x}, \bar{y})$ is the *center estimate*. The center estimate is shown as a yellow dot in each polygon. It indicates the center of the cluster of observations generated for specified emulator or human data. Figures 1–8 illustrate that data generated by some emulators have practically no deviation and are clustered around a single point, while observations generated by other emulators are scattered over the whole polygon. We obtain the Euclidean standard deviation for each data set and illustrate them on the graphs as circles, red for human data and black for emulators around the *center estimates*. The radius of a circle is equal to the standard deviation.[16] Numerical values of the center estimates and standard deviations are shown in Table 3.

We find that the center estimate clearly illustrates information about the emulators and the human data. In the Ochs game (Fig. 1), Cournot Best Reply has the center estimate almost coinciding with the center estimate of the human data, but human data standard deviation is greater. Fictitious Play does a relatively good job, as its center estimate is close to the human data, and its standard deviation (although the circles do not overlap) is greater, and thus closer to the standard deviation of the human data. Adjusted Reinforcement does really well in this game. Its center estimate is close to the human data center estimate, and the standard deviation, although smaller, is within the circle depicting human data dispersion. EWA's center estimate is fairly far away from the human data center estimate. It also generates a standard deviation greater than the human data one, with some overlap.

In the 2 × 2 Stag Hunt game (Fig. 2), all of the emulators' data points and their center estimates are concentrated around a Pareto inferior equilibrium. The center estimate of human data is close to the Pareto superior equilibrium. We obtain a very similar result in 3 × 3 Stag Hunt (Fig. 3) which has 3 equilibria that can be Pareto ranked. It is clear that the center estimate for the human data is very close to the Pareto optimal equilibrium, while the same estimates for all of the four emulators are very close to the equilibrium with the lowest stage-game payoff.

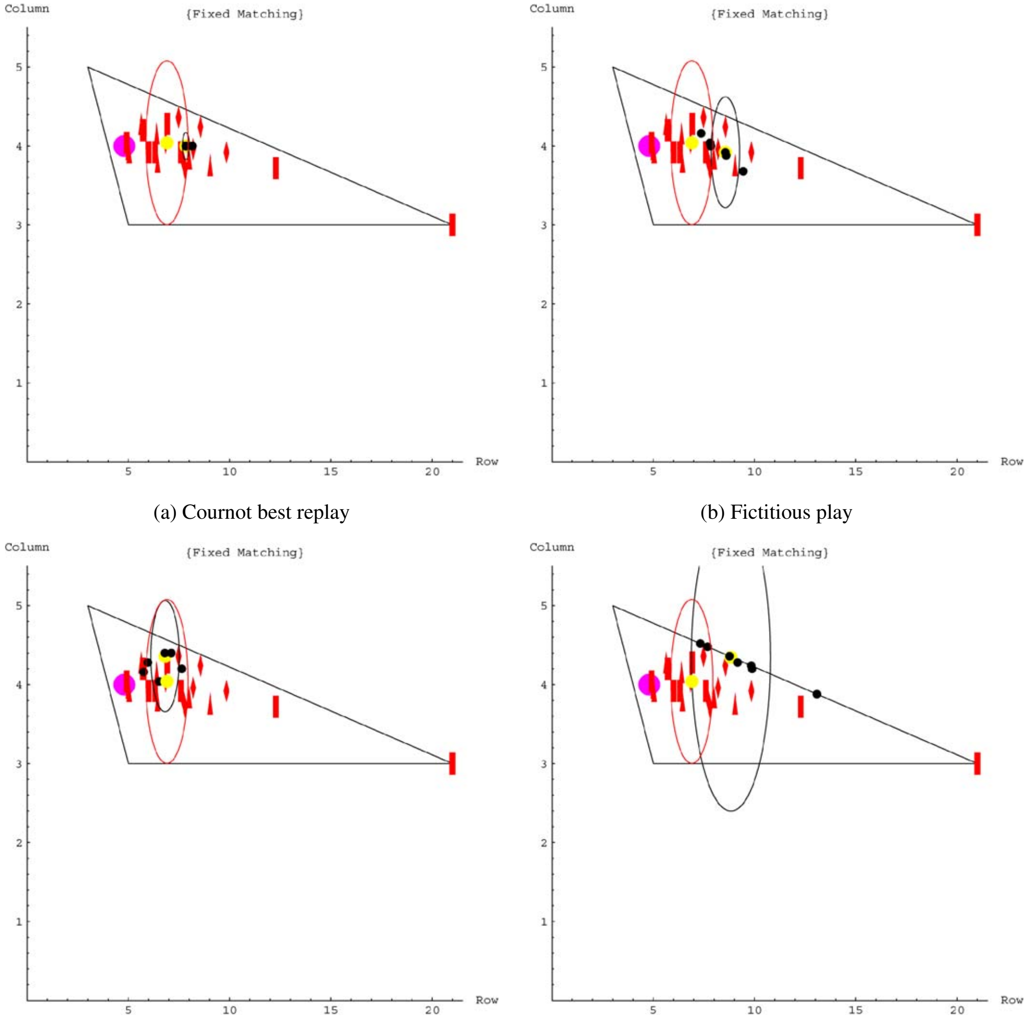
In the 3 × 3 Ultimatum game (Fig. 4), Cournot Best Reply and Fictitious Play are far away, with their center estimates, from the human data. However, Adjusted Reinforcement does a much better job. Its center estimate is very close to the human data center estimate, and its circle is within the boundaries of the human data circle. EWA does not do as well as Adjusted Reinforcement, but still does a pretty good job. Its center estimate is closer to the human data center estimate than are the center estimates for Cournot Best Reply and Fictitious Play.

[16] Note that, for a given game, the human data presented in each figure are the same. Also, in each figure, two center estimates for both emulator and human data are presented with yellow dots. However, the circles that indicate deviations are presented in black for emulators, and with red for human data. The larger the circle, the more dispersed the data points are.

Table 3
Center estimates coordinates and standard deviations

Game	Data	$\bar{x}$	$\bar{y}$	$\sigma_{\bar{x},\bar{y}}$
Ochs	Cournot	3	3	3.08
	Fictitious	3.46	4.51	3.48
	AdjReinf	3.73	3.62	2.41
	EWA	3.79	3.78	2.18
	human data	4.93	4.94	2.41
2 × 2 Stag	Cournot	9.88	9.88	8.57
	Fictitious	10.20	4.45	2.62
	AdjReinf	7.44	7.98	5.01
	EWA	6.07	3.91	2.96
	human data	10.86	10.73	4.91
3 × 3 Stag	Cournot	1.98	2.14	0.71
	Fictitious	5.80	1.59	1.84
	AdjReinf	5.10	4.60	3.67
	EWA	3.15	4.38	2.66
	human data	7.51	7.65	3.32
3 × 3 Ultimatum	Cournot	4.10	1.23	2.39
	Fictitious	21.83	17.80	5.24
	AdjReinf	16.80	14.52	7.92
	EWA	8.49	31.44	2.80
	human data	11.53	11.69	7.11
Centipede	Cournot	4.96	1.04	0.59
	Fictitious	2.91	2.65	1.83
	AdjReinf	3.84	2.74	1.96
	EWA	2.75	1.20	1.32
	human data	2.46	2.81	0.74
Prisoner’s Dilemma	Cournot	2.89	2.90	2.52
	Fictitious	3.24	3.27	1.22
	AdjReinf	3.15	3.13	1.73
	EWA	3.16	3.20	1.40
	human data	4.96	4.97	1.01
Battle of Sexes	Cournot	3	3	2.91
	Fictitious	3.53	3.58789	2.10
	AdjReinf	3.34	3.9	2.17
	EWA	3.31	3.30733	1.90
	human data	5.79	5.80	1.1
Chicken	Cournot	7.84	4	0.8
	Fictitious	8.84	4.34	3.31
	AdjReinf	13.05	3.88	2.68
	EWA	7.47	4.15	2.98
	human data	6.90	4.04	1.04

For 3 × 3 Centipede game (Fig. 5), the Fictitious Play and the Adjusted Reinforcement center estimates are the closest to the human data. However, Fictitious Play’s standard deviation is very small compared to humans, while Adjusted Reinforcement standard deviation indicates dispersion of the data more in line with the human data. Cournot Best Reply’s data points are all concentrated around the Nash equilibria, while EWA’s points are in a different corner of the payoff polygon.



(a) Cournot best replay

(b) Fictitious play

(c) Adjusted reinforcement

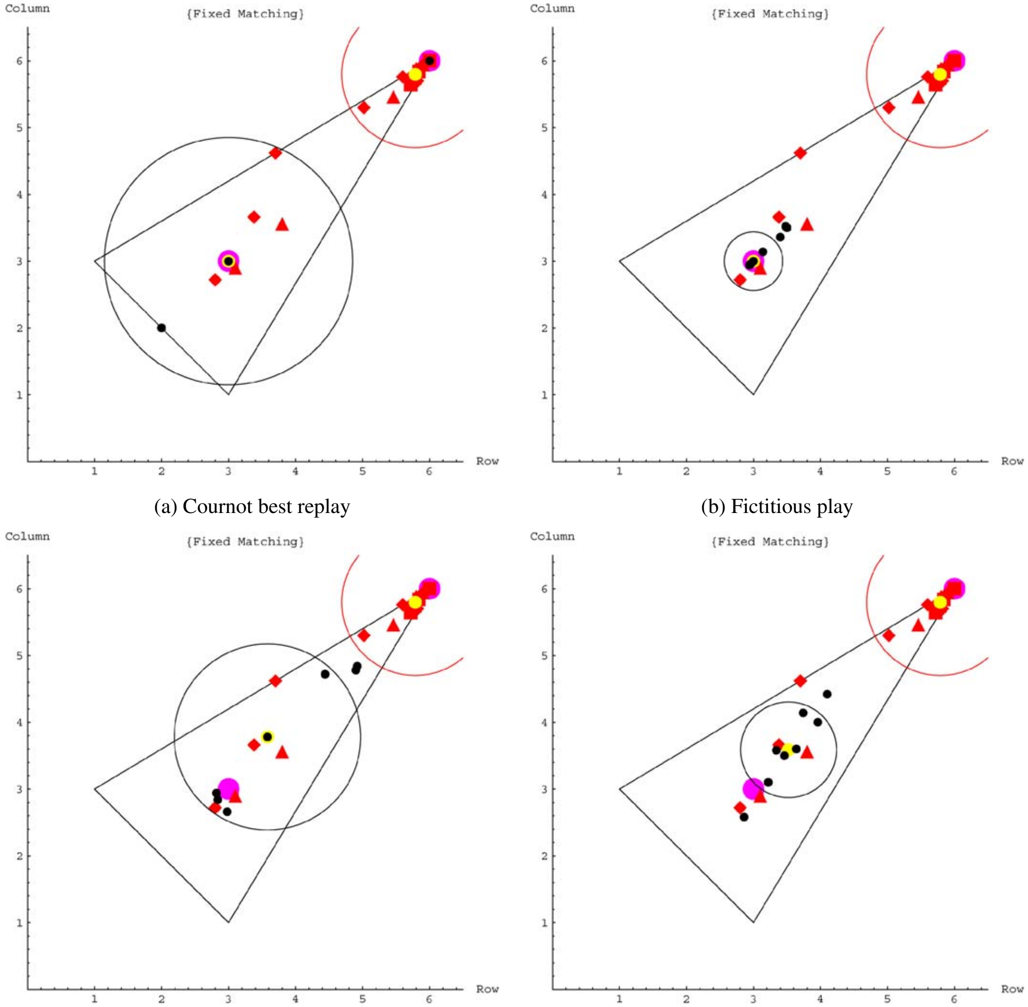
(d) EWA

Fig. 1. Ochs Game (each token gives the payoffs to a matched pair of subjects). *Note*: Must be viewed in color. Red = Human, Black = Machine. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In case of Prisoner's Dilemma (Fig. 6), the center estimate for human data is close to the cooperative outcome with a fairly large standard deviation. However, center estimates of all of the emulators are far away from the human data one.

When it comes to Battle of Sexes game (Fig. 7), Cournot Best Reply has its center estimate relatively close to the human data, but the standard deviation is a lot larger as some of its data points are concentrated in pure strategy, stage game equilibria. The center estimates of the remaining three emulators are far away from the human data center estimate.

Finally, in case of the Game of Chicken (Fig. 8), the center estimates of all of the emulators are far away from the human data center estimate.



(a) Cournot best replay

(b) Fictitious play

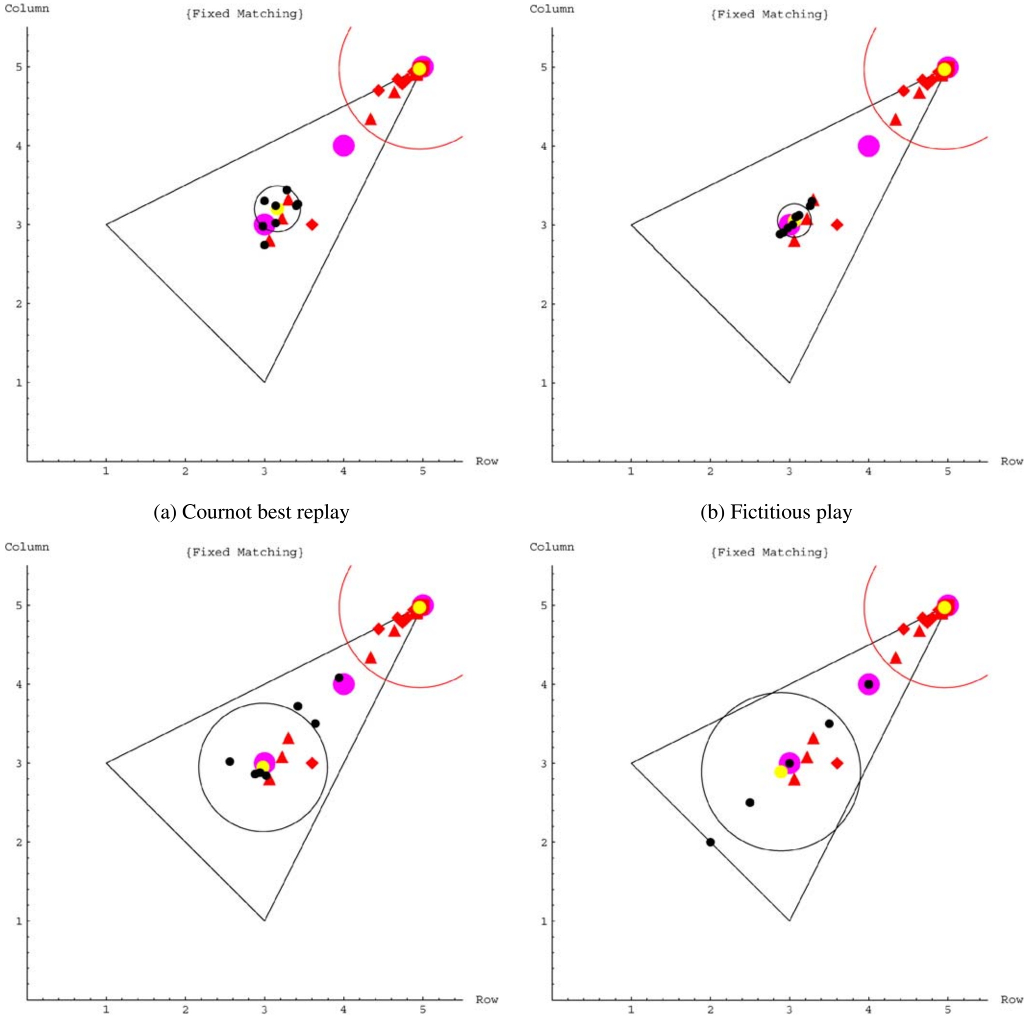
(c) Adjusted reinforcement

(d) EWA

Fig. 2. 2 × 2 Stag Hunt (each token gives the payoffs to a matched pair of subjects). *Note*: Must be viewed in color. Red = Human, Black = Machine. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Two features observed in the figures representing the Prisoner's Dilemma and Battle of the Sexes games are worth pointing out. First, in the Prisoner's Dilemma game (Fig. 6), all the data points generated by Cournot Best Response and Fictitious Play are close to the Nash equilibrium. The data points generated by RL and EWA are scattered mostly in the lower left side of the payoff polygon, not reaching either the Nash equilibrium or Pareto optimal cooperative outcome. However, most of the data points that resulted from human interactions are concentrated close to the cooperative outcome (although there is some noticeable dispersion).

Secondly, in the Battle of the Sexes Game (Fig. 7), none of the four presented emulators were able to pick up the coordination that occurred in the human data. In the human data, subjects would frequently achieve more than could be achieved by independent randomization by alter-



(a) Cournot best replay

(b) Fictitious play

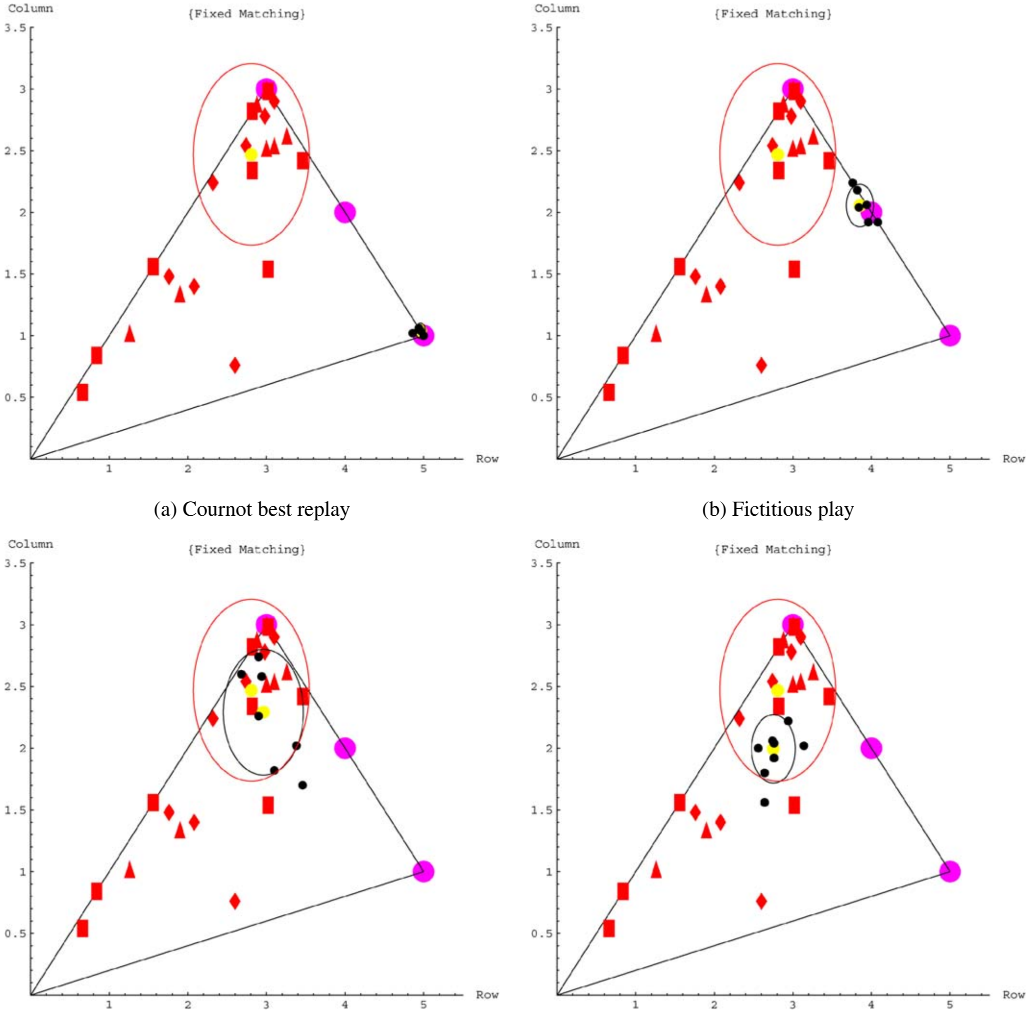
(c) Adjusted reinforcement

(d) EWA

Fig. 3. 3 × 3 Stag Hunt (each token gives the payoffs to a matched pair of subjects). *Note*: Must be viewed in color. Red = Human, Black = Machine. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

nating back and forth between the pure strategy equilibria. Thus, on odd moves, they would go to the equilibrium preferred by one of the players, and on even moves to the equilibrium preferred by the other. At the same time, Cournot Best Responses and Fictitious Play generated points very close to one of the pure strategy Nash equilibria, and RL and EWA generated data points scattered around, without getting close to the point $(5, 5)$ of the polygon's payoff space.

Next, in Tables 4–11 we report mean values and standard deviations of the payoffs of row and column players as well as the total average payoff per each of the eight games for the above discussed emulators (plus Alg1 and Alg2). A number of observations regarding game specific results are worth pointing out.



(a) Cournot best replay

(b) Fictitious play

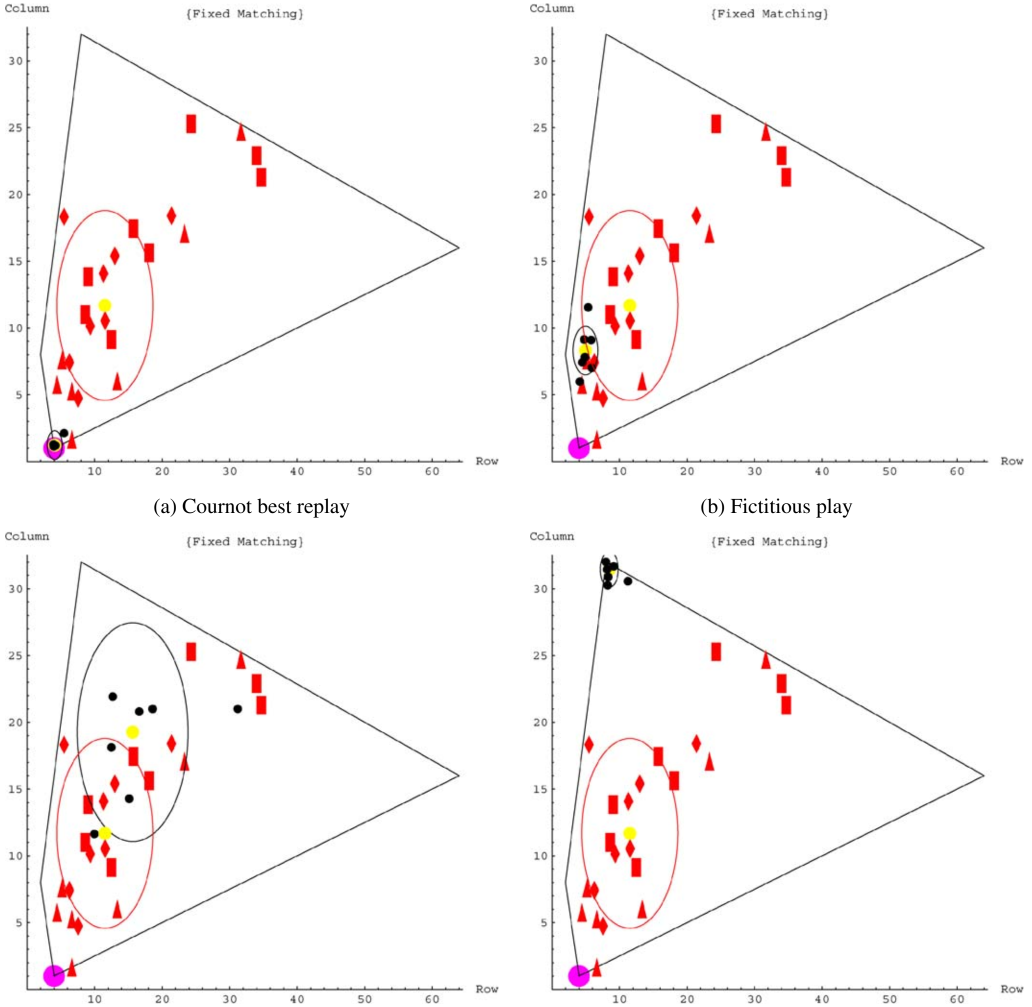
(c) Adjusted reinforcement

(d) EWA

Fig. 4. 3 × 3 Ultimatum (each token gives the payoffs to a matched pair of subjects). *Note*: Must be viewed in color. Red = Human, Black = Machine. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The data recorded in the 2 × 2 Stag Hunt game show that there is not much difference in terms of average payoffs of row and column players in the observations generated by humans and the observations generated by our emulators. Relatively high payoffs reflect the fact that both humans and emulators frequently played the payoff dominant equilibrium.

However, in the 3 × 3 Stag Hunt, humans get payoffs significantly higher than most of the emulators. The difference is that humans continue to frequently play the payoff dominant Nash equilibrium while the emulators frequently choose the equilibrium with the lowest payoffs for both players. The exceptions are emulators Alg1 and Alg2. Alg1 achieved a highest possible



(a) Cournot best replay

(b) Fictitious play

(c) Adjusted reinforcement

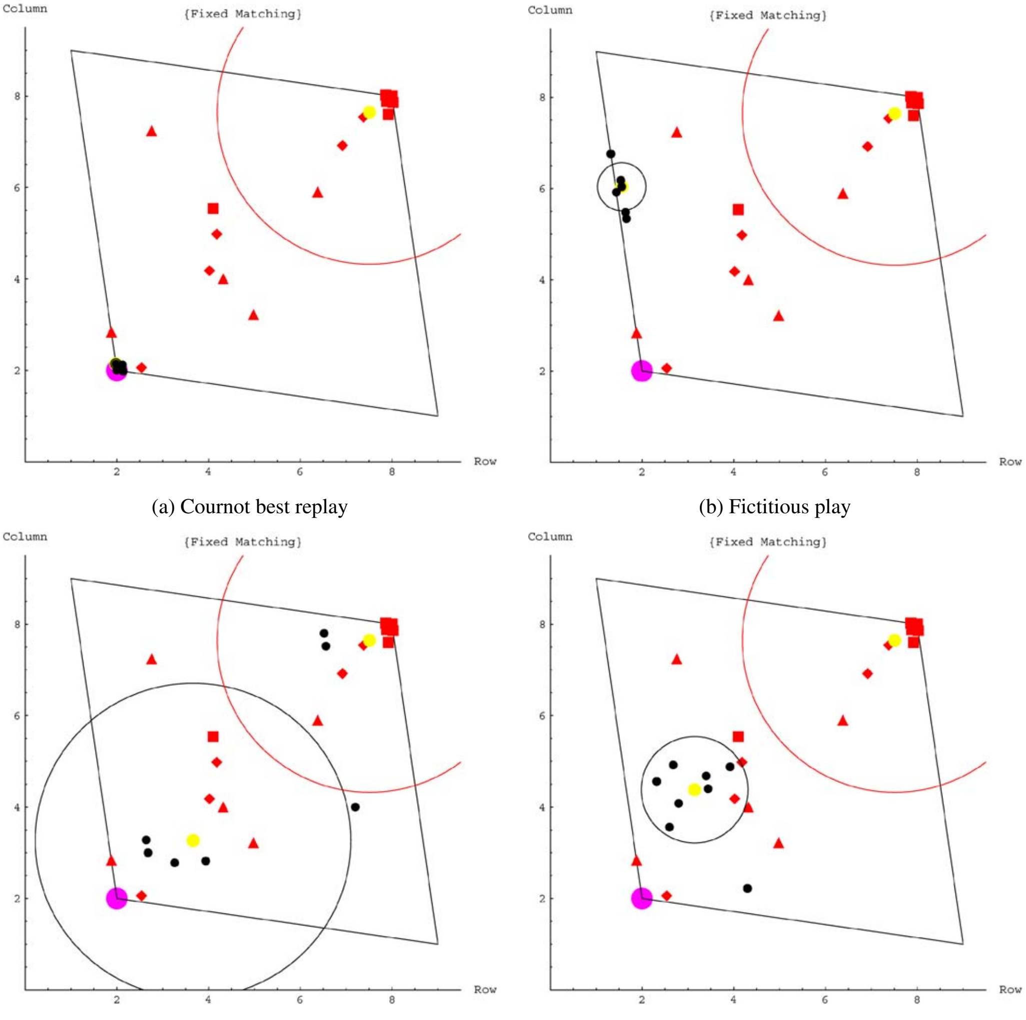
(d) EWA

Fig. 5. 3 × 3 Centipede (each token gives the payoffs to a matched pair of subjects). *Note*: Must be viewed in color. Red = Human, Black = Machine. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

average payoff for both column and row players $(5, 5)$. This means that Alg1 players always played the payoff dominant equilibrium.

In the Centipede game, Alg1 again gets the higher payoff than human subjects or any other emulator by alternating between two cells with the highest payoff for each player.

Finally, in the Prisoner's Dilemma game, human data reveals on average higher payoffs for both row and column players than the emulator generated data. The exception is Alg1 which has a higher average payoff for each player as well as a higher total average score than any of the human data sets. This score reveals cooperation outcome for each round of the games, as the standard deviation for both the row and the column player is 0. This is a higher level of cooperation than any of the ones exhibited by human data. The CIT data set does have values



(a) Cournot best replay

(b) Fictitious play

(c) Adjusted reinforcement

(d) EWA

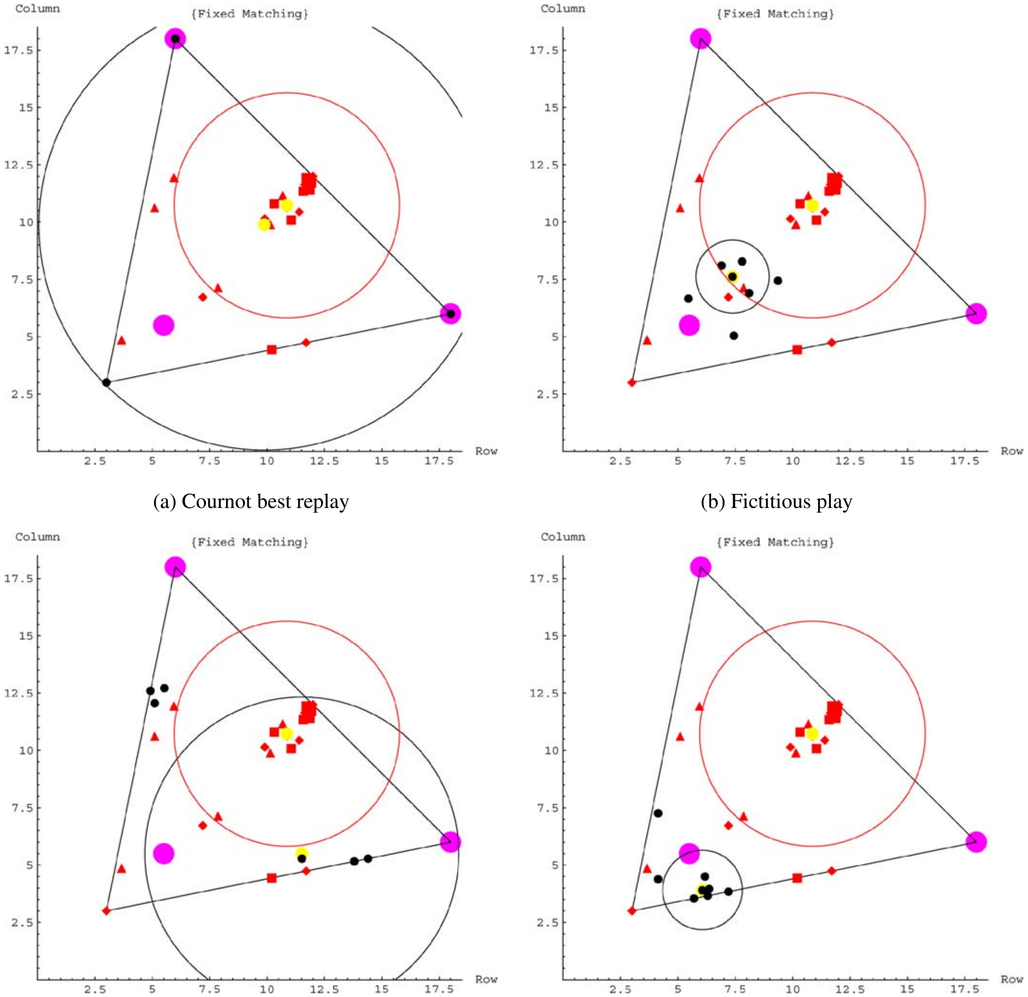
Fig. 6. Prisoner's Dilemma (each token gives the payoffs to a matched pair of subjects). *Note*: Must be viewed in color. Red = Human, Black = Machine. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

close to the cooperative outcome, 7.46 (1.27 standard deviation) for row player and 7.60 (0.79 standard deviation) for the column player.

In the 3 × 3 Ultimatum game, human row players get lower payoffs than machine row players, while the reverse is true of the column players. Human subjects tend to divide the pie more evenly than the emulators.

4.2. *Results of the initial tournament*

The overview of the human and emulator behavior provides insight to the scores of individual emulators and detectors that we obtained in the initial implementation of the Tournament.



(a) Cournot best replay

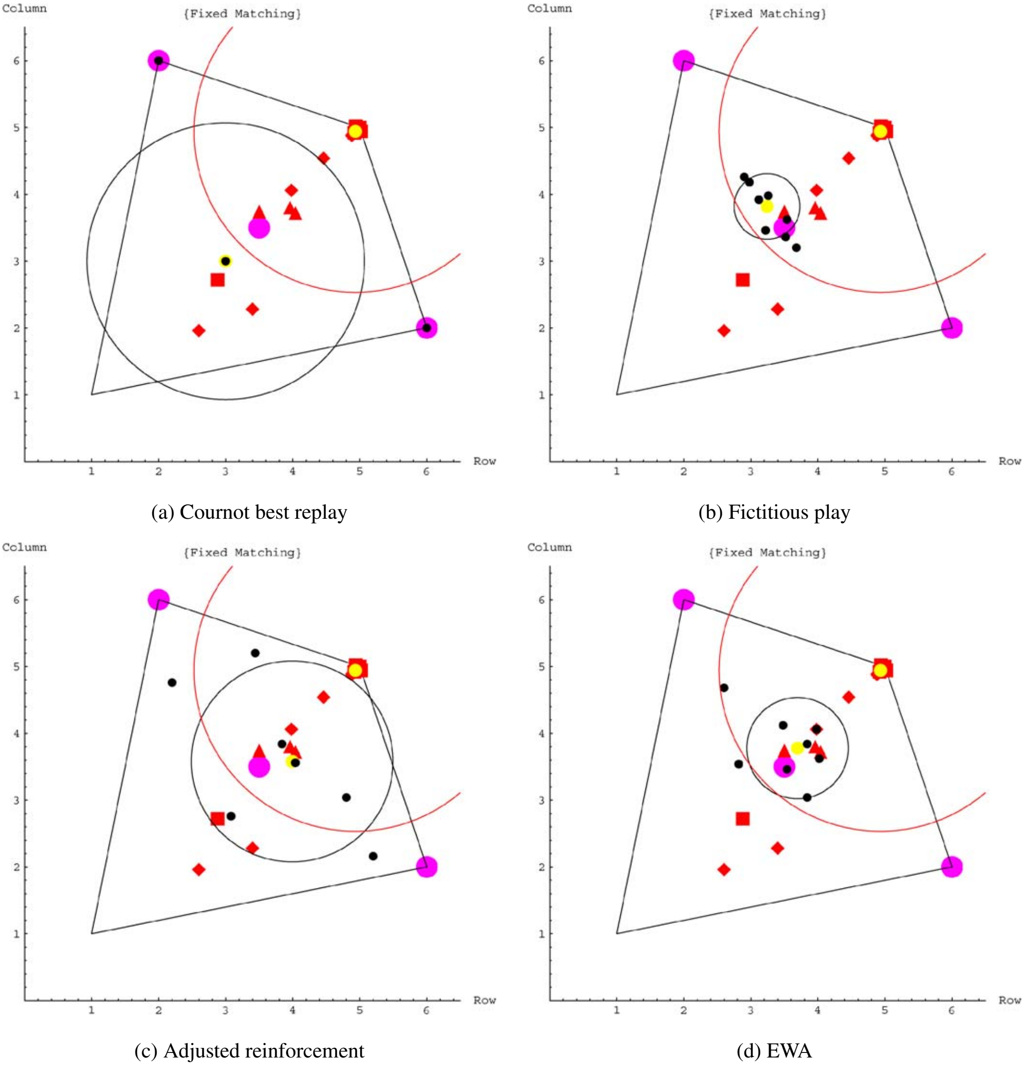
(b) Fictitious play

(c) Adjusted reinforcement

(d) EWA

Fig. 7. Battle of the Sexes (each token gives the payoffs to a matched pair of subjects). *Note*: Must be viewed in color. Red = Human, Black = Machine. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 12 gives the results of a run of the tournament. The first column lists the detectors that were implemented and the second column gives their overall scores against all of the data sets (using baseline scoring scheme). The following three columns give the scores of the detectors for each of the human data sets, followed by columns presenting the scores of the detectors against each of the emulators that were implemented. In this run of the tournament, the Coordination detector won with the lowest negative score of −0.684 and the emulator that had the best score against this detector is Alg1. The detector that comes in the second place is the Random detector. Tables 13 and 14 show, respectively, how the detectors performed across the games, and how they performed on each data set for the Battle of the Sexes game.



(a) Cournot best replay

(b) Fictitious play

(c) Adjusted reinforcement

(d) EWA

Fig. 8. Chicken (each token gives the payoffs to a matched pair of subjects). *Note*: Must be viewed in color. Red = Human, Black = Machine. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 13 reveals that the Coordination detector gets the highest score, −0.28, in the data related to the Battle of the Sexes (Coordination) game. This detector was mainly designed to detect coordination in the Battle of Sexes game that occurs in the human data where subjects alternated between the two Nash equilibria. Looking back at Table 12, we see that Alg1, again designed to emulate this type of coordination has the best score against the winning detector. This is largely due to its good performance in the Battle of the Sexes game where it was hard for the coordination detector to detect that it was facing machine generated data.

Another notable observation is that the High payoff detector gets the best score in the Chicken game and the worst in the Ultimatum game. This means that human subjects are able to realize

Table 4
Ochs Game

Data Set	*n*	Row		Column		Total payoff
		Mean	Std. Dev.	Mean	Std. Dev.	
CIT	16	8.85	5.06	3.88	0.37	12.72
PCC1	14	6.51	1.41	3.96	0.18	10.47
PCC2	16	8.08	0.83	4.01	0.19	12.08
Random	14	8.14	0.84	4.02	0.15	12.16
Cournot	16	7.96	0.15	4.00	0.00	11.96
Fictitious	16	4.53	0.13	4.39	0.13	8.92
AdjReinf	14	6.76	1.21	4.26	0.16	11.02
EWA	16	8.49	3.27	4.36	0.34	12.84
Alg1	16	12.00	0.00	4.00	0.00	16.00
Alg2	12	4.87	0.80	4.31	0.11	9.17

Table 5
2 × 2 Stag Hunt

Data Set	*n*	Row		Column		Total payoff
		Mean	Std. Dev.	Mean	Std. Dev.	
CIT	16	5.95	0.10	5.94	0.12	11.88
PCC1	14	5.12	1.08	5.07	1.19	10.19
PCC2	16	4.75	1.17	4.94	1.10	9.68
Random	14	3.24	0.35	3.13	0.25	6.37
Cournot	16	4.63	1.80	4.63	1.80	9.25
Fictitious	16	5.59	0.98	5.59	0.98	11.17
AdjReinf	14	4.04	0.84	4.01	0.85	8.05
EWA	16	4.00	0.73	4.03	0.58	8.03
Alg1	16	6.00	0.00	6.00	0.00	12.00
Alg2	16	5.23	0.17	5.24	0.20	10.47

Table 6
3 × 3 Stag Hunt

Data Set	*n*	Row		Column		Total payoff
		Mean	Std. Dev.	Mean	Std. Dev.	
CIT	16	4.98	0.03	4.98	0.03	9.96
PCC1	14	4.07	0.79	4.03	0.87	8.10
PCC2	16	4.64	0.43	4.64	0.63	9.28
Random	14	3.01	0.19	3.08	0.20	6.09
Cournot	16	2.69	0.83	2.69	0.83	5.38
Fictitious	16	3.23	0.45	3.23	0.45	6.46
AdjReinf	14	3.37	0.50	3.49	0.34	6.86
EWA	16	3.23	0.29	3.26	0.16	6.49
Alg1	16	5.00	0.00	5.00	0.00	10.00
Alg2	16	4.57	0.09	4.58	0.12	9.15

Table 7
3 × 3 Ultimatum

Data Set	*n*	Row		Column		Total payoff
		Mean	Std. Dev.	Mean	Std. Dev.	
CIT	16	2.28	1.02	1.88	0.84	4.16
PCC1	14	2.63	0.69	2.27	0.72	4.90
PCC2	16	2.55	0.43	2.12	0.75	4.67
Random	14	2.62	0.29	1.61	0.20	4.23
Cournot	16	4.96	0.03	1.04	0.03	6.00
Fictitious	16	4.92	0.05	1.08	0.05	6.00
AdjReinf	14	3.33	0.35	1.93	0.25	5.26
EWA	16	2.73	0.28	1.80	0.18	4.53
Alg1	16	4.00	0.00	2.00	0.00	6.00
Alg2	16	2.80	0.28	1.79	0.18	4.59

Table 8
3 × 3 Centipede

Data Set	*n*	Row		Column		Total payoff
		Mean	Std. Dev.	Mean	Std. Dev.	
CIT	16	19.63	9.73	17.06	5.38	36.69
PCC1	14	13.04	9.79	9.73	7.54	22.77
PCC2	16	10.75	4.73	12.38	4.68	23.13
Random	14	12.31	3.33	8.54	1.11	20.86
Cournot	16	4.01	0.11	1.21	0.09	5.21
Fictitious	16	6.54	4.61	4.49	2.84	11.03
AdjReinf	14	17.20	9.71	18.96	6.00	36.16
EWA	16	8.22	0.36	31.68	0.38	39.90
Alg1	16	36.00	0.00	24.00	0.00	60.00
Alg2	16	19.48	3.54	16.14	1.81	35.61

Table 9
Prisoners Dilemma

Data Set	*n*	Row		Column		Total payoff
		Mean	Std. Dev.	Mean	Std. Dev.	
CIT	16	7.46	1.27	7.60	0.79	15.05
PCC1	14	5.17	2.21	5.60	2.08	10.77
PCC2	16	6.03	1.97	6.17	2.05	12.20
Random	14	4.88	0.74	5.29	0.59	10.17
Cournot	16	2.05	0.06	2.01	0.04	4.06
Fictitious	16	2.05	0.06	2.01	0.04	4.06
AdjReinf	14	4.22	1.23	4.45	1.78	8.66
EWA	16	3.97	1.49	4.03	1.25	8.01
Alg1	16	8.00	0.00	8.00	0.00	16.00
Alg2	16	5.53	0.64	5.39	0.64	10.92

Table 10
Battle of the Sexes

Data Set	n	Row		Column		Total payoff
		Mean	Std. Dev.	Mean	Std. Dev.	
CIT	16	11.28	0.64	10.38	2.31	21.66
PCC1	14	7.91	2.91	9.66	2.49	17.57
PCC2	16	9.71	2.93	8.66	3.17	18.38
Random	14	7.59	0.71	7.25	0.99	14.85
Cournot	16	6.75	4.44	12.75	6.83	19.50
Fictitious	16	10.05	5.25	13.41	5.93	17.21
AdjReinf	14	8.16	3.43	9.05	3.88	23.46
EWA	16	4.82	0.95	5.60	2.34	10.41
Alg1	16	12.00	0.00	12.00	0.00	24.00
Alg2	16	7.24	0.97	7.09	1.10	14.33

Table 11
Chicken

Data Set	n	Row		Column		Total payoff
		Mean	Std. Dev.	Mean	Std. Dev.	
CIT	16	4.70	0.69	4.69	0.75	9.40
PCC1	14	4.49	0.59	4.46	0.61	8.95
PCC2	16	4.27	0.82	4.07	1.16	8.34
Random	14	3.46	0.43	3.67	0.30	7.13
Cournot	16	3.75	1.30	2.75	0.43	6.50
Fictitious	16	3.75	1.30	2.75	0.43	6.50
AdjReinf	14	3.71	1.01	3.95	0.84	7.65
EWA	16	3.60	0.65	3.40	0.62	7.00
Alg1	16	5.49	0.05	3.52	0.18	9.02
Alg2	16	3.80	0.42	3.73	0.42	7.54

Table 12
The results of a run of the tournament (entries in cells are scores)

Detectors	Score	CIT	PCC 1	PCC 2	Random	Cournot	Fictitious	AdjReinf.	EWA	Alg1	Alg2
Random	−.693	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69
Dates	−.969	−0.57	−0.57	−0.57	−0.92	−0.92	−1.26	−1.09	−1.26	−0.92	−0.92
NE and PO	−.830	−1.61	−1.44	−1.26	−1.61	−0.40	−0.22	−0.92	−0.92	−0.40	−1.09
Payoff change	−.742	−1.15	−0.77	−0.46	−0.62	−0.02	−0.33	−0.67	−0.50	−0.02	−1.81
High payoffs	−.908	−0.57	−0.85	−0.75	−0.31	−0.87	−1.44	−0.60	−0.36	−4.40	−0.81
Coordination	−.684	−0.36	−0.37	−0.35	−0.50	−0.77	−0.77	−0.67	−0.56	−1.96	−1.10

Note: Scores do not add up because only 10 of 20 data sets are displayed.

that playing symmetric payoff cell $(5, 5)$ gives greater payoffs than alternating between Nash equilibria. This behavior is hard to capture with the existing learning models. On the other hand, in the Ultimatum game, humans do not receive as high payoffs as many emulators do. All human data sets show significantly lower difference in the payoffs between column and row players (for human data sets, the range is between 0.36 and 0.43 while for emulator data sets, the range is between 0.93 and 3.92).

Table 13
Detectors' scores in games

Detectors	Average	Ochs	Stag 2	Stag 3	Ult	Cent	PD	Coord.	Chicken
Random	−0.693	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69
Dates	−0.969	−0.43	−1.40	−1.40	−0.50	−0.85	−0.36	−1.40	−1.40
Closeness to NE and PO	−0.830	−0.85	−0.78	−0.92	−0.92	−0.78	−0.85	−0.78	−0.78
Payoff change	−0.742	−1.07	−1.03	−0.63	−0.44	−0.71	−0.70	−0.65	−0.70
High payoffs	−0.908	−0.86	−0.83	−0.56	−2.06	−0.91	−0.68	−1.13	−0.25
Coordination	−0.683	−0.69	−1.45	−0.57	−0.69	−0.69	−0.39	−0.28	−0.69

Table 14
Turing test scores in Battle of the Sexes

Tests	Score	CIT	PCC 1	PCC 2	Random	Cournot	Fictitious	AdjReinf.	EWA	Alg1	Alg2
Dates	−1.401	−0.22	−0.22	−0.22	−1.61	−1.61	−1.61	−1.61	−1.61	−1.61	−1.61
Random	−0.693	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69	−0.69
Fast convergence	−0.777	−1.61	−1.61	−1.61	−1.61	−0.22	−0.22	−0.22	−0.22	−1.61	−1.61
Payoff change	−0.652	−0.32	−0.22	−0.32	−0.70	−0.01	−0.35	−0.22	−0.38	−0.01	−1.28
High payoffs	−1.127	−0.11	−0.87	−0.81	−0.37	−1.62	−3.43	−0.52	−0.01	−4.61	−0.33
Coordination	−0.279	−0.01	−0.06	−0.02	−0.21	−0.04	−0.04	−0.12	−0.06	−3.22	−0.78

4.3. *Stability of results to different evaluation approaches*

We next investigate the impact of the alternative ways of scoring the detectors and emulators. Our *baseline* scoring is sensitive to outliers. Note however, that the actual tournament includes a large number of iterations that should take care of this problem. We report below the scores and the ranking of the detectors and emulators, using additional scoring rules as described earlier in Section 2.4.

4.3.1. *Detectors*

Table 15 reports detectors' scores for different scoring rules, the mean value (our *baseline* average rule), the highest worst score obtained (minimax), and the median game score. The mean column shows the tournament results according to our baseline rule that have been discussed in the previous section. Table 16 reports the detectors' ranking according to these different scoring rules.

When we use the worst score rule, the Random detector comes in first. This is due to the fact that each of the other 5 detectors had some of their scores lower than those generated by the Random detector. The Random detector always assigns probability 0.5 and therefore always makes a 0.5 deviation from the true state. That is why it always receives a score of −0.693. However, every other detector may make a worse error by assigning a probability greater than 0.5 to the emulator generated data or lower than 0.5 to human data. Payoff Change, High Payoffs and Coordination detectors made relatively large mistakes at least once.

Looking at the scores according to the median game scoring rule, Coordination detector and Random detector have exactly the same score and are both assigned the same place, 2, in the table. Coordination detector was mainly designed to detect coordination games and games with Pareto dominant payoffs and distinguish between human and emulator data based on the level of coordination observed in the players' decisions. However, this detector does not include a sophisticated response to games that do not contain a coordination element or Pareto dominant

Table 15
Detector scores

	Mean	Minimax	Median game
Random	−0.693	−0.693	−0.693
Dates	−0.968	−1.609	−1.124
Ninety %	−0.830	−1.609	−0.812
Payoff change	−0.742	−4.605	−0.670
High payoffs	−0.908	−4.60	−0.843
Coordination	−0.684	−3.219	−0.693

Table 16
Detector ranking

	Mean	Minimax	Median game
Random	2	1	2
Dates	6	2	6
Ninety %	4	2	4
Payoff change	3	5	1
High payoffs	5	5	5
Coordination	1	4	2

outcome. Thus, when this detector is uncertain about the type of the game, it just assigns a probability 0.5 that the data file is human. This feature lowered the score of this detector, and resulted in a median game score of −0.693, which is the score of the Random detector. The fact that its median score is equal to the *Random* median score comes from the number of such unidentified games presented to Coordination detector in the Tournament. This can be seen in Table 13 that reports the scores that detectors received for each individual game. Coordination receives a score of −0.69 (which is the score received for giving a probability of 0.5) for Ochs, Ultimatum, Centipede, and Chicken games; the games that do not contain the coordination element that this detector was designed for. We could have increased the score for this detector if, for example, we combined it with the elements of the Payoff Change detector, which performed relatively well in a median game. However, our efforts were primarily focused on developing the methodology and the tournament itself, rather than designing the best detector.

Two alternative scoring schemes are based on specific single scores (minimum or median) generated by the detectors. We find that detector that performs best using the baseline approach still shows good performance based on the alternative scoring schemes. The baseline rule also has advantages as described in Section 2.4.1 and illustrated empirically in this section. For example, Random detector that always makes a fixed error comes best using minimax rule. Payoff Change detector comes first using Median game scoring scheme, however performs worst based on the Minimax scoring scheme, since it makes large mistakes in some data sets.

4.3.2. *Emulators*

In the Tournament design, we have scores of each emulator against each detector. The winning emulator is chosen based on scores against the most accurate detector. In this section, we present results of several other scoring schemes (as described in Section 2.4.2) and report the results in Table 17.

We investigate the impact of four scoring schemes. The first is the *Winning detector* scoring scheme used in the Tournament implementation. The reported scores are those that each

Table 17
Emulators' scores

	Rand.	Cournot	Fict.	Reinf.	EWA	Alg1	Alg2
Winning detector	−0.50	−0.76	−0.76	−0.66	−0.56	−1.956	−1.100
Unweighted mean	−0.77	−0.61	−0.79	−0.77	−0.72	−1.40	−1.070
Weighted mean	−0.72	−0.51	−0.64	−0.75	−0.69	−0.79	−1.18
Median game	−0.78	−0.56	−0.67	−0.76	−0.76	−1.28	−1.14

Table 18
Emulators' ranking

	Rand.	Cournot	Fict.	Reinf.	EWA	Alg1	Alg2
Winning detector	7	3	3	5	6	1	2
Unweighted mean	4	7	3	5	6	1	2
Weighted mean	4	7	6	3	5	2	1
Median game	3	7	6	5	4	1	2

emulator obtained versus the winning detector. The second scoring scheme is the *Unweighted mean* that uses scores generated by all detectors for all emulators and finds average by assigning equal weight to all detectors. The emulator's score is thus a simple average of its scores against each one of the detectors. The third scoring scheme is the *Weighted mean* where the emulator's score is given as a weighted average of the scores against each of the detectors as described in Section 2.4.2. The fourth is the *median game* scoring scheme in which each emulator's median game score against the detector is reported. Table 18 reports emulators' ranking when the above evaluation schemes are used.

The Random emulator that assigns the same probability to each choice available to the player is ranked 7th according to our Baseline scoring scheme. However, it has a 3rd rank using the Median game scoring scheme. Cournot's performance is better with the Winning detector scoring scheme (the Baseline) than any other. In fact, while it scores 3rd place with this method, it takes low 6th or 7th place with other scoring schemes. The ranking of Fictitious is similar to that of Cournot, i.e. relatively high ranking with the Winning detector scoring scheme, 3rd place again, while 6th place with all the other schemes. Despite changes in ranks of some emulators as described above, the performance of these emulators does not change significantly for different scoring rules. All these emulators have scores relatively similar to each other and changes in ranks are caused by small deviations. The scores of the winning emulators (Alg1 and Alg2) on the other hand are consistently larger than the rest and are sometimes twice as large as the scores of the rest of the emulators (for example *Unweighted mean* scores in Table 17).

The winning emulator based on the Winning detector scoring scheme, Alg1, comes consistently first using all but one scoring scheme (where it comes second). Alg1 and Alg2 which were designed for the repeated game setting using the features observed in the experimental data, take first or second place when either one of the scoring schemes is used. The results of the Tournament are robust to different scoring schemes. The above considerations and analysis that we conducted provide support for the use of our baseline scoring approach in the Tournament.

5. Conclusion

We presented the design and an initial implementation of the Turing Tournament to learning in two person repeated games. We developed a program that implements the tournament. In addition, we used a number of emulators (mainly, well known algorithms from the literature on learning) and developed a few detectors in order to test how the tournament performs. These emulators and detectors were then treated as submissions to the tournament. For human data, we used the experimental results from McKelvey and Palfrey (2002) and created human data sets that were then shuffled with data sets generated by emulators. Finally, our detectors evaluated the data sets. We studied and illustrated results of different scoring schemes for determining the winning emulator and detector. Based on theoretical considerations and empirical evidence of stability of results, we find support for the baseline approach of the current design.

The analysis of our results shows that there are often significant differences between data sets generated by humans and those generated by emulators, at least for the versions of the emulators we implemented and parameter values that we used. Our results show that there is room for improvement in developing new emulators or more appropriate and better implemented versions of the existing emulators. In addition, the differences between human and machine behavior demonstrate that there is room for development of good detectors, which is a new and complex task. Once the best detector(s) has been objectively established in the Tournament, it can be used to test performance of learning models prior to collecting human data. Future research can address performance of detectors and emulators under different matching and information conditions.

Acknowledgments

We acknowledge support of California Institute of Technology for a grant to fund initial development and of the National Science Foundation (Grant #SES-0079301) for help in funding the experiments. We also acknowledge support of the Social Science Experimental Laboratory for providing use of its facilities for program development and running of the experiments. We thank Brian Rogers for his research assistance, Chris Crabbe, Mayanka Prasad, Melissa Todd and Walter Yuan for their help in programming the tournament. We thank Colin Camerer, Ido Erev, John Ledyard, Tom Palfrey, Charles Plott, participants at the ESA meetings, 2001, in Tucson and two anonymous referees for helpful comments.

Supplementary Appendix

Supplementary material associated with this article can be found, in the on-line version, at doi: 10.1016/j.geb.2006.03.013.

References

Arifovic, J., McKelvey, R.D., 2002. The Turing tournament: A method for evaluation of social science theories. Working paper.

Boylan, R., El-Gamal, M., 1993. Fictitious play: A statistical study of multiple economic experiments. Games Econ. Behav. 5, 205–222.

Camerer, C.F., Ho, T., 1999a. Experience-weighted attraction learning in games: Estimates from weak link games. In: Budescu, D., Erev, I., Zwick, R. (Eds.), Games and Human Behavior: Essays in Honor of Amnon Rapoport. Erlbaum, pp. 31–51.

Camerer, C.F., Ho, T., 1999b. Experience-weighted attraction in games. Econometrica 67, 827–874.

Camerer, C.F., Ho, T., Chong, 2002. Sophisticated EWA learning and strategic teaching in repeated games. J. Econ. Theory 104 (1), 137–188.

Crawford, V., 1991. An 'evolutionary' interpretation of Van Huyck, Battalio, and Beil's experimental results on coordination. Games Econ. Behav. 3, 25–59.

Crawford, V., 1995. Adaptive dynamics in coordination games. Econometrica 63, 103–143.

Gneiting, T., Raftery, A.E., 2004. Strictly proper scoring rules, prediction, and estimation. Technical report No. 463. Department of Statistics, University of Washington.

Hanaki, N., Sethi, R., Erev, I., Peterhansl, A., 2005. Learning strategies. J. Econ. Behav. Organ. 56, 523–542.

Hanson, R., 2002. Logarithmic market scoring rules for modular combinatorial information aggregation. Working paper. George Mason University.

McKelvey, R.D., Palfrey, T.R., 1995. Quantal response equilibria for normal form games. Games Econ. Behav. 10, 6–38.

McKelvey, R.D., Palfrey, T.R., 2002. Playing in the dark: Information, learning, and coordination in repeated games. Working paper. Caltech.

O'Carroll, F.M., 1977. Subjective probabilities and short-term economic forecasts: An empirical investigation. Appl. Statist. 26 (3), 269–278.

Roth, A.E., Erev, I., 1995. Learning in extensive-form games: Experimental data and simple dynamic model in the intermediate term. Games Econ. Behav. (Special Issue: Nobel Symposium) 8, 164–212.

Roth, A.E., Erev, I., 1998. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. Amer. Econ. Rev. 88 (4), 848–881.

Roth, A.E., Erev, I., 1999. On the role of reinforcement learning in experimental games: The cognitive game theory approach. In: Budescu, D., Erev, I., Zwick, R. (Eds.), Games and Human Behavior: Essays in Honor of Amnon Rapoport. Erlbaum, pp. 53–77.

Savage, L.J., 1971. Elicitation of personal probabilities and expectations. J. Amer. Statist. Assoc. 66 (336), 783–801.

Staël von Holstein, C.-A., 1970. A family of strictly proper scoring rules which are sensitive to distance. J. Appl. Meteorol. 9 (3), 360–364.

Stahl, D.O., 1996. Boundedly rational rule learning in a guessing game. Games Econ. Behav. 16, 303–330.

Stahl, D.O., 1999. Evidence based rules and learning in symmetric normal form games. Int. J. Game Theory 28, 111–130.

Turing, A., 1950. Computing machinery and intelligence. Mind 59, 433–460.

Winkler, R.L., 1969. Scoring rules and the evaluation of probability assessors. J. Amer. Statist. Assoc. 64 (327), 1073–1078.