Peer-Design Agents for Reliably Evaluating Distribution of Outcomes in Environments Involving People

Moshe Mash, Raz Lin and David Sarne
Department of Computer Science
Bar-Ilan University
Ramat-Gan, Israel 52900
{mashmos,linraz,sarned}@cs.biu.ac.il

ABSTRACT

In many domains, an autonomous agent needs to reliably predict the distribution of behaviors of a population rather than the behavior of a single agent. For example, when playing the ultimatum game against several unknown opponents from a large known population, the agent can perform better by extracting its best-response strategy based on the distribution of the acceptance value in that population. In this paper, we demonstrate the efficacy of Peer-Designed-Agents (PDAs) for producing a distribution of behaviors that highly resembles the distribution of actual behaviors of a specific population of interest. This is obtained through extensive experiments with more than 700 different individuals and 132 PDAs, using eight game variants from three different domains and two different statistical tests. The analysis of the results demonstrates that PDAs' technology is an effective means for generating a reliable distribution of behaviors of a population of interest, as long as the similarity between the group of PDAs' developers and the latter population is sufficiently high. Moreover, a comprehensive comparison with the results of Elicited-Strategy-Agents (ESAs) shows that there is much more to PDA technology than simply an expression of strategy.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems

General Terms

Experimentation

Keywords

PDAs, Strategy elicitation, Agent-based analysis of human interactions

1. INTRODUCTION

Reliably modeling and predicting the behavior of autonomous agents in a multi-agent system (MAS) is a key capability sought both by system and agent designers (e.g., see [17]). System designers depend on these capabilities for evaluating and tuning the systems and mechanisms they design (e.g., using simulations [27]). Agent designers take advantage of these capabilities to design better strategies for their agents, whenever the outcomes of the actions taken and choices made by the agent are influenced by the behavior of others in its environment [14, 16]. While the behavior of fully rational and computationally unbounded agents can be extracted numerically or analytically using optimization and game theoretic principles, when it comes to bounded rational agents, and particularly people, alternative approaches should be considered. This is mainly because people's behaviors are known to be affected by a range of cognitive influences [1, 13]. People often use rules of thumb or adopt simple heuristics either due to their bounded computational capabilities or because of poor problem solving capabilities [10, 29]. Recent research has been arguing in favor of and extensively advocating the use of peer-designed agents (PDAs) — computer agents developed by human subjects — as an effective means for modeling the behavior of MAS whereby the acting individuals are people [6, 8]. The advantage of using PDAs in that sense (as opposed to "expert-designed agents", for example) is that it allows the generation of a rich set of strategies with a substantially small overhead, in a timely manner and with much parallelism in the process [6]. The underlying assumption in these works is that PDAs capture people's behavior adequately, and therefore the PDA-based system, if equipped with enough PDAs to reliably capture the diversity in people's behavior, is likely to represent a collective behavior similar to when populated with people [5]. As such, much work has been dedicated to empirically investigating the level of similarity observed between PDAs and people. The results reported are not conclusive — some works suggest a relatively strong correlation between the behaviors of the two (e.g., [5]), while other works report on PDAs that act to some extent in a different manner than people [9, 23]. In an effort to resolve the conflict, recent work has shown that the success of using PDAs as a substitute of people is both setting- and measure-dependent, therefore the preference of PDAs for mechanism evaluation must be re-evaluated whenever switching to a new setting or using a different measure [8].

While prior literature dedicated to evaluating the suitability of PDAs as reliable representatives of people is substantial, to the best of our knowledge, it is all based on either: (a) a comparison of individual decisions made by a PDA and the person who designed its strategy, in similar decision situations [5]; or (b) a comparison of the average or other stylized facts at the system level [8, 16, 2, 18] in order to determine similarity between PDA-based and people-based behaviors. None of these works has explored the possibility of comparing the distribution of behaviors emerging in a given setting when using the two approaches. Nevertheless, in many settings the "average" behavior does not capture enough information and an agent's design can be substantially improved if supplied with the distribution of choices made by the population of other agents in a specific scenario. For example, when designing an agent for ultimatum games or sealed bid auctions, there is only trivial benefit from knowing the average acceptance value (cross-opponents) or the average bid made by others. In this case the agent's best-response strategy is derived based on the probability distribution of values [7, 14]. Therefore, comparing the distribution of choices made by PDAs and individuals of the population of interest, in a given envi-

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

ronment, is a better and more reliable measure for determining the level of similarity between the two.

Furthermore, none of the prior work has attempted to supply any evidence to the benefit of requesting individuals to program PDAs rather than simply expressing their strategy. In the latter case, a professional programmer could have been used to do the coding. This issue may have substantial implications since PDAs programming requires some basic programming knowledge and skills. The constraint of using programmers may suggest, in some settings, that the PDAs used are programmed by individuals from a population that does not adequately represent the population of interest (behaviors from the population the designer is trying to extract). Therefore, if the requirement for actually programming the strategy can be waived, a more adequate set of PDAs can be produced.

Motivated by PDA-related research and practice, this paper makes an important leap forward in advancing PDA research by: (a) providing empirical evidence of its success in assessing the strategic behavior of a population in the form of the resulting distribution of outcomes; and (b) comparing the performance of the PDA-based method to an alternative approach that uses ESAs — agents whose strategies were expressed by individuals of the population of interest and then programmed by a programmer. The evaluation reported in the paper encompasses three different games of different domains, overall with eight different problem settings. It relies on experimenting with 708 human subjects that generated $5,599$ different observations (which, to the best of our knowledge, is more than an order of magnitude greater than any former PDA research), 132 PDAs, each, within a given domain, designed by a different subject, and 132 ESAs whose strategies were described, rather than programmed, by subjects.

The analysis of our results indeed shows that PDAs can be used effectively to assess a population's strategic behavior, whenever the similarity between the population who programmed the PDAs and the population of interest is sufficiently high. Since ESAs have been found to perform worse than PDAs, it is evident that there is much added benefit to PDA developers programming the agent. In addition, we demonstrate the advantage of using PDAs for different variations of a given game. Thus, even though acquiring a PDA incurs some overhead, the ability to adapt and extend it to various settings of the environment, reduces the total cost and makes the method an appealing research tool.

2. RELATED WORK

Many approaches have been taken over the years to generate a reliable set of agent behaviors for a given environment, including the use of expert designed agents, modeling based on statistical data [30], strategy development based on a pre-defined set of events and reactions [21], the construction of complex behaviors based on a set of simple ones [31], a combination of rules and finite state machines to control an agent's behavior [32], and participatory simulations [12]. While these approaches have yielded many interesting results their main weakness is in generalizing the behaviors to situations different from those used to collect the real data which the strategies constructed were based. For example, it was shown that the resemblance between people and agents reported for the double auction environment does not hold once the value of one of the market parameters slightly changes [4]. In this sense, PDA technology offers a great promise for system and agent designers if indeed it is capable of reliably capturing the distribution of people's behaviors in a given domain. The method is inspired by the "strategy method" paradigm from behavioral economics [26] in the sense of eliciting people's strategy. Nevertheless, while in the strategy method people state their action for every possible situation that may arise in their interaction (i.e., a state-machine-like description) with PDAs people are actually required to program their (not-necessarily-state-based) strategy into an agent.

As discussed in more detail in the previous section, the efficacy of using PDAs is inconclusive. Furthermore, there is vast evidence in prior work, originating in a variety of domains, of discrepancies between actual and reported human behavior, in particular in metacognition research [11]. Examples of such discrepancies include over-reporting of political participation [3] and contrasting results between self-reported and performance-based levels of physical limitations [15]. Indeed, part of the PDA-based literature uses the technology per-se and does not attempt to make any claims regarding the similarity between the agents and some population of interest (e.g., in TAC [25]). Yet, much of the PDA literature tends to assume that people can successfully (to some extent) capture their real-life strategy in a given domain when programming an agent [23]. Recently, Elmalech and Sarne [8] showed that the success of PDAs in reliably capturing the average behavior in one setting cannot be trivially generalized to others, which perhaps is essentially the main drawback of PDAs.

Regardless of the nature of the results reported in prior work, the experimental methodology used was limited to comparing a specific measure's outcome or some stylized facts (e.g., [16]) or to comparing individual decisions of PDAs and their generators in similar situations as a means of determining the PDAs' success. None of the prior work has actually attempted to compare the collective set of behaviors as a whole within the two populations, e.g., in the form of comparing the resulting distribution of outcomes, as performed in this paper. In this sense, it is possible that with the experimental and analytic methodologies used in prior PDA research, the PDA technology can be a suitable replacement for people in a given environment, though: (a) the PDAs produce a set of behaviors that are similar to people's behavior on average, however substantially different individually; and/or (b) despite a substantial overlap between the behavior of PDA developers and of the PDAs they produce, the population of PDA developers does not adequately represent the population of interest. The methodology used in this paper attempts to overcome these two shortcomings.

3. EVALUATION METHODOLOGY

In this section we describe our multi-game/multi-variation evaluation methodology. We first outline the goals and scope set for the experiments. Then, we describe the different populations from which individuals were recruited to take part in the experiments. We continue with a description of the three games used and their specific problem setting variations, followed by a description of the software infrastructures used and the experiments conducted. Finally, we describe the experimental methods and the statistical methodology that was used to analyze the results obtained.

3.1 Goals and Scope

Two primary goals were set for the experiments. The first was to demonstrate the feasibility of using PDAs for capturing the distribution of exhibited behaviors, as a whole, in a given population. This, as opposed to estimating some average or any other stylized facts. As discussed in the introduction, in many domains the capability of reliably capturing the behaviors distribution of a population is critical to provide the best response. Second, we aimed to compare the accuracy of PDA-based and ESA-based generated distributions. The purpose of this comparison was two fold. First, since ESAs are a natural choice and use subjects from the actual population of interest (compared to PDAs that can be programmed only by a small portion of the population, i.e., those with programming skills) they are a useful benchmark for comparison. Second, since both methods are based on capturing one's stated strategy, the difference between the two may be attributed to the fact that with PDAs one needs to actually program its strategy rather than merely express it.

3.2 Populations and Samples

Our experiments considered two primary populations: a general one, and a more specific one (subset of the general one, comprising individuals that can program agents).

General Population.

The population for which our experiments attempted to replicate its distribution of behaviors is the general population of subjects one may find in the Amazon Mechanical Turk framework (AMT).[1] The AMT framework allows the publishing of tasks designated for people all around the world. Empirical findings [22] suggest that AMT can serve as a viable alternative for data collection, as subjects from AMT exhibit the classic heuristics and biases and pay attention to directions at least as much as subjects from traditional sources. In order to construct the distribution of behaviors which characterizes this population, we recruited a large set of participants from the AMT platform.[2] These participants were also used to gather the strategies upon which the ESAs developed were based.

PDA Developers.

Naturally, the population of PDAs' developers includes only people who were capable of designing and expressing their strategy in a way that could be programmed into a computer agent. Therefore, this population was based on computer science and engineering students. Indeed, as evidenced in our results section, expressing one's strategy to a level that can be programmed into an agent, is not trivial for the general population.

3.3 Games

Three games of three different domains were used for our experiments: the *joint shopping* game, *Blackjack* and the *centipede game*. For each game, a varying number of different problem settings was used. The reason for using several games of different domains, and different variations within each, was to strengthen the validly and the ability of generalizing the results obtained. Furthermore, the use of more than a single problem instance enabled reasoning about the magnitude of influence the similarity between the population who programmed the PDAs and the population of interest has on the success of the PDAs in modeling the population of interest.

Joint shopping. This game involves a two-player coordination game. Each player represents a shopper, interested in buying a specific product, that can be found in several different stores. A player can check the posted price in any of the stores while incurring a cost for each store checked. Once both players have completed their individual price-checking process, they meet and share prices so that they both can use the minimum price found by either of them. While the specific price at each store is a priori unknown, the players are familiar with the distribution from which prices are derived. The goal of each player is to minimize her overall expense, measured as the price paid for the product plus the accumulated costs individually incurred along her price-checking process. An agent's best-response strategy in this case depends on the distribution of the minimum value returned as a result of the other player's price-checking process. Given the probability distribution function of this latter parameter, denoted $f_{other}(x)$, the agent needs to calculate the threshold r which satisfies the following equation (see, for example, [19]):

Parameter / Settings	**1**	**2**	**3**
Maximum stores	10	7	4
Store visit cost	3	5	7
Product's price range	0-100	100-200	200-300

Table 1: Three different settings used in the joint shopping game.

$$c = \int_{y=-\infty}^{r} f(y) \int_{x=-\infty}^{\infty} (min(r,x) - min(y,x)) f_{other}(x) dx dy \qquad (1)$$

where c is the cost of checking a price in a store and $f(y)$ is the probability distribution function of prices in the stores available to the agents. The agent should keep checking prices in the stores, as long as the best (lowest) price it individually found so far is greater than r. Therefore, the ability to model the distribution $f_{other}(x)$ in this game is crucial for an agent that wishes to minimize its expected overall expense. Three different settings of the game were used, varying in the maximal number of stores each agent can check, the cost c of visiting a store and the price range of the product prices (taking the distribution of prices to be uniform within that range). The different settings and their parameters are listed in Table 1.

Blackjack. This game considers a simplified version of the classic Blackjack game.[3] The game involves the person playing and the dealer. Both the player and the dealer initially receive two cards. The player can either *Hit*, to obtain an additional card, or *Stand* to end the game. The dealer has to hit until her cards total 17 or more points. At the end of the game the player has to outrank the dealer without busting (hand of cards that exceeds 21). Our simplification of the game suggests that the deck of cards is re-shuffled at the beginning of each game and that the player's reward function assigns 1 for winning the game and 0 for losing it, i.e., no stakes are involved, thus the player's aim is simply to win the game.

In this game, from the casino's point of view, there is a great advantage in knowing the proportion of winnings when players come from a specific population, for example to determine an entrance fee. With the PDA-generated data, the probability of the dealer winning in a given game can be calculated using the probability distribution of the sum of cards the player receives, denoted $p_{people}(y)$, as follows:[4]

$$\sum_{y=2}^{21} p_{people}(y) \sum_{x=max(y,17)}^{21} p_{dealer}(x) + (1 - \sum_{y=2}^{21} p_{people}(y)) \qquad (2)$$

where $P_{dealer}(x)$ is the probability distribution function of sum of the dealer's cards at the end of the game when following the "hit until cards total 17 or more points" rule. The first term in Equation (2) relates to cases where the player ends up with a sum of 21 or below, hence the dealer can win only if her cards total a greater sum, up to 21. The second term relates to the case where the player has busted, hence the dealer wins. The probability of having the player win is the complementary probability of the above.

Centipede game. The centipede game, first introduced by Rosenthal [24], is a two-player extensive form game. In this game, each player, on her turn, can either "Take" a larger share of an increasing pot, or "Pass" the pot to the other player. The best response action in this game depends on the probability the other player will choose "Take" for the first time on turn i, denoted $p_{people}(i)$. The agent's expected-benefit-maximizing turn to first choose "Take" is

[1] For a comparison between AMT and other recruitment methods see [22].

[2] Since each participant in AMT has a unique ID, connected to a unique bank account, it is possible to block the same ID from participating more than once in a given experiment.

[3] http://wizardofodds.com/blackjack

[4] Of course one could have used the average number of winning directly from the PDAs' data. Yet, the determination that the PDAs-generated data appropriately represent the population of interest, which is the condition for using this data in the first place, is stronger when comparing the distribution of cards sums rather than merely the winnings.

Parameter / Settings	1	2	3	4
Maximum turns	6	6	4	4
Starting player	P	O	P	O

Table 2: Four different settings of the centipede game used in the experimentation. P/O stands for the tested player or its opponent, respectively.

thus determined according to:

$$\underset{k}{\operatorname{argmax}} \sum_{i=k+1}^{T} p_{people}(i) \cdot U(k) \qquad (3)$$

where T is the total number of turns in the game and $U(i)$ is the amount received if stopping on turn i. The equation essentially calculates for each turn k the product of the player's gain if the game is terminated this turn and the probability the players will actually reach that turn (i.e., the probability the other player will choose "Take" later than turn k). This valuable information can assist an automated agent in the decision of whether to choose "Take" in a given stage, when matched with people from that population.

In this game we also experimented with several different settings of the game. The initial pot in all settings was set at 5 and was doubled each turn until the maximum number of possible turns was reached. On any given turn, the player whose turn it is to choose received $\frac{3}{4}$ of the pot if she chooses "Take". The four settings used differ in the number of turns and the identity of the player who goes first in the game (the experimenting subject or her opponent). The different settings of the game are listed in Table 2 and are the same as the ones tested with people in [20]. The theoretical Nash equilibrium for this game is to always choose "Take". If this strategy is used the game ends in the very first round. Still, there is vast experimental evidence that people do not adopt the equilibrium strategy in this game [20].

These three games offer a variety of features. For example, in the joint shopping game and the centipede game one must consider the strategy of the other player, which is a priori unknown. In Blackjack, the strategy of the dealer is a priori known, however the underlying optimization problem is complex for people in general. The centipede game has an equilibrium that does not require any calculation, whereas the equilibrium in joint shopping game is more complex to derive. While these do not encompass the full set of different features one might encounter in real life, the three games seem to offer a decent range.

3.4 Experiment Infrastructure

To administrate the experiments using the AMT framework we implemented a designated application for each game. The application enabled the presentation of a clear description of the situation needing a decision, at each point, and receipt of the participant's corresponding decision. One implementation detail that required considerable care was the modeling of the opponent against whom the people played. While pairing people for games was possible, this approach had a significant drawback in the form of possible influence from their experiences, and consequently their belief concerning their counterparts' strategies, on the strategy they will use in subsequent games. Having a person play only once was infeasible since participants had to undergo a practice session to ensure they understood the game rules. While this issue was completely avoided in the BlackJack game, as the dealer's strategy is a priori defined and known to the user, a solution was required for the other two games. Therefore, in the joint shopping game, we bypassed this problem by telling participants that the results obtained by the other player, in all games played, would be revealed only at the end of the experiment, and only then their performance in the game would be calculated. For the centipede game, we had to reveal the opponent's actions to the participants in real time. Therefore, we used the distribution of first "Take" decisions according to the empirical findings reported in [20] for the specific setting we used.

To allow the generation of PDAs for each game, we generated a skeleton PDA, one for each game, according to the common practices used in recent PDA literature [8]. These skeleton agents were equipped with all the functionalities needed, such as communication and observation of the environment. They only lacked the strategy that determines which actions they will take based on the inputs received. Computer science students were given this skeleton in order to develop PDAs, hence requested to program and debug only the agent's strategy. We also used these skeleton agents to implement the ESAs by transferring the strategies described by the AMT players into codes.

3.5 Flow of the Experiment

All participants in our experiments were given detailed written instructions, explaining the rules of the game, and their individual goal according to which their performance would be calculated, for the specific game of their experiment. In order to qualify to participate in the experiment, the participants also had to undergo a short multiple choice test, verifying that they carefully read the instructions and understood the rules and the method of measuring their performance. In all three games a participant's performance was calculated according to the average "score" she achieved in the game instances played: (a) in the joint shopping game the score in a game was considered the individual expense; (b) in the Blackjack game the score was 1 if the player won the game and otherwise zero; and (c) in the centipede game, the score was the amount received upon a "Take". The performance measure was used to compensate participants linear to their performance.

After reading the game instructions and passing the qualification test, all participants (both AMT and CS/Engineering students) were requested to play several practice games (3-5 mandatory practice games, depending on the game; each subject could extend the practice stage until she felt confident in her ability to succeed). The games in the practice games included all game variants (whenever applicable). Once the participants felt ready, they were requested to play several instances of the specific game in which they were participating. The instances were of all variants of the specific game, and were presented to the participant in a random order to avoid bias. Then, the participants were requested to either express the strategy they use in this game, in free form text (AMT participants), or develop a PDA based on a skeleton supplied, using the strategy they use for this game (CS/Engineering students).[5] Each valid (i.e., programmable) strategy that was expressed by an AMT participant was used by us to program an ESA. Nonetheless, in order to keep conditions equal, we stopped developing ESAs once we reached the same number of valid PDAs received for each game.

Both the PDA-developers and AMT participants expressing their strategy were asked to use a general strategy that does not apply to a specific game variant of the game used (e.g., in the joint shopping application PDAs were requested to use as an input the number of stores that can be visited, any store visit cost, and any interval on which the price-distribution is defined. In the same sense AMT participants were requested to set their strategy capable of handling these parameters). The purpose of this requirement was to allow the use of any of the PDAs/ESAs in all the variants tested. While this suggested a slight compromise in the level of accuracy that can be obtained with the use of the PDAs/ESAs, it served our purpose well, i.e. to show that PDAs encompass the benefit of being able

[5] This followed the common practice used in PDAs development, to ensure that the programmed strategy resembles the one the PDA developer uses in real life (e.g., see [6]).

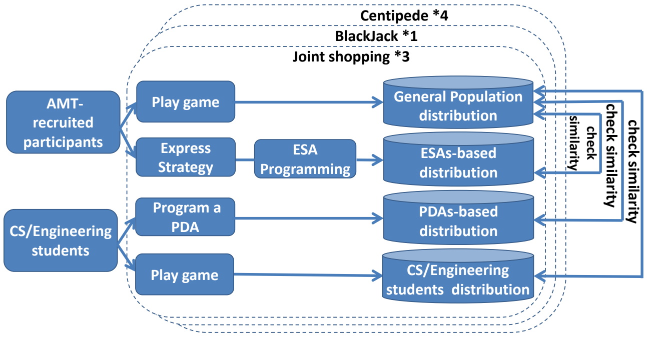

Figure 1: The experimental methodology.

to be used for a wide range of settings without having to be reprogrammed, and the failure of ESAs as a general modeling tool.

The following data was stored for each game played, either by a person, a PDA or an ESA:

- Joint shopping - the lowest price found by the player throughout the game.
- Blackjack - The sum of card values the player ended up with.
- Centipede game - the turn the game ended.

Therefore, after completing the experiments we had the following datasets for each game variant in the three domains (see Figure 1):

- General population distribution - the set of observations collected from the games played by people from AMT (i.e., corresponding to the general population).
- ESA-based distribution - the set of observations collected from the set of ESAs that played the specific game variant several times.
- PDA-based distribution - the set of observations collected from the set of PDAs that played the speciic game variant several times.
- Strategy designers' distribution - the set of observations collected from the games played by the PDA developers themselves (i.e., corresponding to the PDA developer population).

The first dataset among the above four is in fact the distribution that needs to be modeled. The PDA-based and ESA-based datasets are the distributions one is likely to obtain with the two distributions that generate the methods evaluated in this paper. These can be potentially utilized in agent design to extract the agent's best-response strategy for the applicable game, as explained and exemplified above. Therefore, the main comparisons that were carried out were between the distribution represented by the first dataset and the latter two datasets. The last dataset was used to evaluate the level of similarity between CS/Engineering students and the general population, therefore it was compared with the first dataset.

3.6 Statistical Measures

To facilitate a comprehensive validation and understanding of the results, several measures and comparisons were used. First, we ran the Kolmogorov-Smirnov statistical test ([28], Chapter 2) based on the population of interest (General population dataset) and each of the two other datasets (ESA- and PDA-based datasets). The Kolmogorov-Smirnov test (hereafter denoted K-S) is a nonparametric test that compares the cumulative distributions of two datasets. Its null hypothesis is that the data in the two datasets are from the same distribution. Therefore, this test is a natural candidate for identifying the cases where the population of interest and the sample produced with any of the methods were not taken from the same population (i.e., when the null hypothesis is rejected).

An additional complementary measure that was calculated to support the analysis is the similarity between the PDA developers and the general population of interest. This was measured based on the $p\text{-}value$ obtained by comparing the first and fourth datasets. This measure was calculated both when taking the complete datasets, and also for subsets of varying sizes taken from these two datasets. The first was used as the ultimate measure for the similarity between the two populations (programmers versus people in general) and the second was used to evaluate the overhead required for reasoning about the similarity between the two populations in general, as explained in more detail in the following section.

In order to reduce the dependence of the findings on the specific statistical test used, we repeated the analysis of the results, this time, however, using the Wilcoxon rank sum test ([28], Chapter 2). The Wilcoxon rank sum test is a non-parametric test, designated to test the difference between two samples by comparing the two population medians.

4. EXPERIMENTS AND RESULTS

In this section we report the results of the analysis carried out. Due to space considerations, the results reported are those obtained using the K-S statistical test. Towards the end of the section we briefly discuss the differences between these results and those obtained with the Wilcoxon rank-sum test.

Overall, there were 135 different PDA developers (CS and Engineering students) and 573 different AMT human participants who took part in the experiments. Of the 135 PDA developers, 62, 40 and 33 took part in the joint shopping, Blackjack and centipede games, respectively. Of the 573 AMT human participants, 160, 247 and 166 took part in the joint shopping, Blackjack and centipede games, respectively. The number of games each student or AMT participant played was: 3, 2 and 3 (for variants 1-3 of the joint shopping game), 5 (for the BlackJack game) and 3 (for each variant of the centipede game). Table 3 summarizes the number of observations collected from the PDA developers' games (denoted "CS"), the number of functional PDAs and ESAs developed (denoted "PDAs" and "ESAs", respectively) and the number of observations collected from the AMT participants' games. Note that the number of PDAs listed in the table for each game is slightly smaller than the number of students that took part in the game, because the table lists only the functional agents that were produced. The table also includes, for each game and setting, the p-value of the K-S test comparing the AMT population and the last dataset (CS-played games) as a measure of the similarity between the computer science subjects and the population gathered from AMT. Based on the value received, it is apparent that CS/Engineering students play BlackJack differently than AMT participants and play the other two games quite similar to them. The only exception in this sense is the similarity between the two groups in the third setting of the joint shopping game (which is substantially low). This can be explained by the relatively low number of stores and the relatively high cost of price checking characterizing this setting. With these values, the optimal strategy for the agent is to check only one store, regardless of the strategy of the other player, which might be easier for CS students to grasp.

One notable result from our attempts to produce ESAs is that the process is associated with substantial overhead. While participants were specifically requested to describe a strategy that we could then program into a computer agent, only a very small percentage of the strategies received were actually "programmable". This can be seen from the ESA column in Table 3, where the numbers in the parenthesis represent the numbers of strategies we had to review until we reached the same number of valid strategies as the number of PDAs obtained. The numbers given in the column corresponds to 38%, 46% and 19% programmable strategies specified in the joint shopping, Blackjack, and centipede, respectively. For the Blackjack game the results are more encouraging, mainly due to the fact that the game is simpler and the threshold for hitting or standing was

Game and Settings		CS	PDAs	ESAs	AMT	Similarity
Joint Shopping	1	186	60 (62)	60 (160)	480	0.54
	2	124	60 (62)	60 (160)	320	0.52
	3	186	60 (62)	60 (160)	480	0.02
Blackjack	1	200	40 (40)	40 (86)	1,235	0.00
Centipede	1	99	32 (33)	32 (166)	498	0.78
	2	99	32 (33)	32 (166)	498	0.58
	3	99	32 (33)	32 (166)	498	0.62
	4	99	32 (33)	32 (166)	498	0.51

Table 3: Number of observations (CS and AMT), PDAs and ESAs in the different settings, as well as the similarity of computer science (CS) subjects to the same population populated from AMT in the different games and settings. Similarity was measured using the Kolmogorov-Smirnov test.

written clearly in the people's strategy. Nevertheless, even in this game many of the strategies received were too obscure, vague or nonspecific enough to allow the generation of an agent, e.g., a strategy stating that if the sum of cards is "close" to 21 then stand, and otherwise hit. This strategy is not specific enough to program an ESA. This tendency to use subjective conditions as part of the expressed strategy was reflected to a greater extent in the other two games. For example, in the joint shopping game we frequently ran into strategies that advocated browsing until a price is found which is "significantly below the other prices or not worth to continue browsing". In contrast to ESAs, PDAs were generally valid, and the only problem encountered was with agents that got stuck in some states or crashed. Still, the number of such cases is negligible (as can be seen by the difference between the overall number of PDAs received, in parenthesis in the PDA column, and the number of valid PDAs).

Figure 2 presents the K-S p-value as a function of the agent-based dataset size used, for each of the different variants of the three games. The different curves differ in the size of the pool of agents (either PDAs or ESAs) that were used for generating the data, and each data point is the average of 10,000 $p\text{-}value$ results of a specific combination of these different parameters values. For example, in the most upper left graph in Fig. 2 each data point of the curve *35 PDAs* is the average of the 10,000 $p\text{-}values$ obtained when comparing a subset of size x values (according to the horizontal axis) generated using 35 random PDAs from the total 60 available. The purpose of the graphs is to demonstrate that the $p\text{-}values$ we use for the analysis, which are those obtained with datasets of size 10,000, are those the system has converged to, and none of the conclusions made result from using a dataset of insufficient size. The idea in presenting the different curves, corresponding to different agent pool sizes, is to show the effect of the number of agents used, overall, over the level of similarity achieved. Intuitively, the greater the number of PDAs or ESAs programmed, the greater the chance that, if the method is indeed effective in terms of modeling the behavior of the population of interest, the agents will be capable of exhibiting a richer set of behaviors, therefore performing more closely to the population of interest. The fact that each time we re-sampled the pool of PDAs/ESAs from the full repository, assures that the results are not biased due to running into a poor or favorable set of strategies.

Figure 2 demonstrates that only with one setting of the joint shopping (setting 3) and in the Blackjack game the null hypothesis, whereby the PDAs-generated samples and the sample taken from the population of interest were taken from the same populations, can be rejected (e.g., with $\alpha = 0.05$). The statistical tests executed for all other cases with PDAs yielded $p\text{-}values$ that are substantially greater than what is typically used to reject the null hypothesis (e.g., greater than 0.2 and in some cases even 0.5 and 0.8). The only factor that differentiates the two cases, whereby the null hypothesis is rejected, from the other cases is in which the level of similarity between the PDA developers and the population of interest, as reflected in the K-S $p\text{-}value$ for the two (see the last column in Table 3), is relatively low (0.02 for the third setting of the joint shopping, and 0 for Blackjack). This explanatory factor is quite straightforward — if the similarity between the PDA developers and the population of interest is a priori low, then modeling the latter using strategies designed by the first is likely to be futile.

Interestingly, even with a moderate number of PDAs (e.g., with 10 or with 20), we were able to obtain good results, making it very difficult to reject the null hypothesis regarding the similarity between the two populations. This is in contrast to results obtained with the ESAs, where substantially worse results were revealed. In most cases the statistical tests for the maximum number of ESAs yielded $p\text{-}values$ below 0.05, thus rejecting the null hypothesis that ESAs-generated samples and the sample taken from the population of interest were taken from the same populations. Only in three settings (one of the joint shopping and two of the centipede games) the $p\text{-}values$ obtained were higher than 0.05. Still, in those cases the $p\text{-}values$ obtained with the PDAs were substantially greater than those attained with ESAs (for the centipede game) or were initially insignificant (for the joint shopping variant), indicating that the method was not suitable for these variants in the first place.

One important phenomena reflected in the different graphs relates to the benefit in increasing the pool of PDAs/ESAs used for generating the datasets. Whenever the similarity between the population was high an increase in the number of agents generated higher $p\text{-}values$, whereas in the other cases a reverse phenomenon was revealed, indicating that the method is useless. This is because, as discussed above, when PDA/ESA technology is generally suitable for a given setting, having more agents is generally beneficial, as it enables the emulation of behaviors that are even less common in the population of interest. However, when the approach is a priori not suitable for a given setting (i.e., due to the initial low resemblance between the PDA-developers and the population of interest) the influence of agents that do correspond well to some individuals from the population of interest becomes more apparent with a small pool of agents. This is the case because even if these agents are more likely to be initially picked, the portion of the observations they will produce will be substantial, resulting in relatively high $p\text{-}value$ whenever picked. With a large pool, however, even though they will be initially picked more often, their influence in each run for the most part will be almost insignificant.

As the strength and efficacy of PDAs is shown to rely highly on the similarity between those who programmed the PDAs and the target population, we administrated an additional battery of experiments to assure that the similarity can be easily measured based on a small sample of the population. In these experiments we incrementally compared the subjects who programmed the PDAs to the target population from AMT using the K-S test. The results, depicted in Fig. 2 (bottom right graph), show the average $p\text{-}value$ over 10,000 trials for each population size, i.e., each time taking two subsets of a similar size from the two populations. It can be seen that even when a small amount of subjects is tested a rather accurate conclusion can be reached regarding the nature of the similarity between the two populations; thus the usefulness of the PDAs in this case can be determined.

A comparable analysis based on the Wilcoxon rank-sum test yielded similar qualitative results. Table 4 summarizes the qualitative similarities and differences in the results when the analysis is performed based on K-S compared to the results based on the Wilcoxon rank sum test. The comparison is with respect to the decision of whether to accept or reject the null hypothesis (with $\alpha = 0.05$), denoted "Acceptance", and the determination regarding the similarity between PDA developers and the population of interest, based on small samples as given in bottom right graph in Fig. 2, denoted "Trendline".

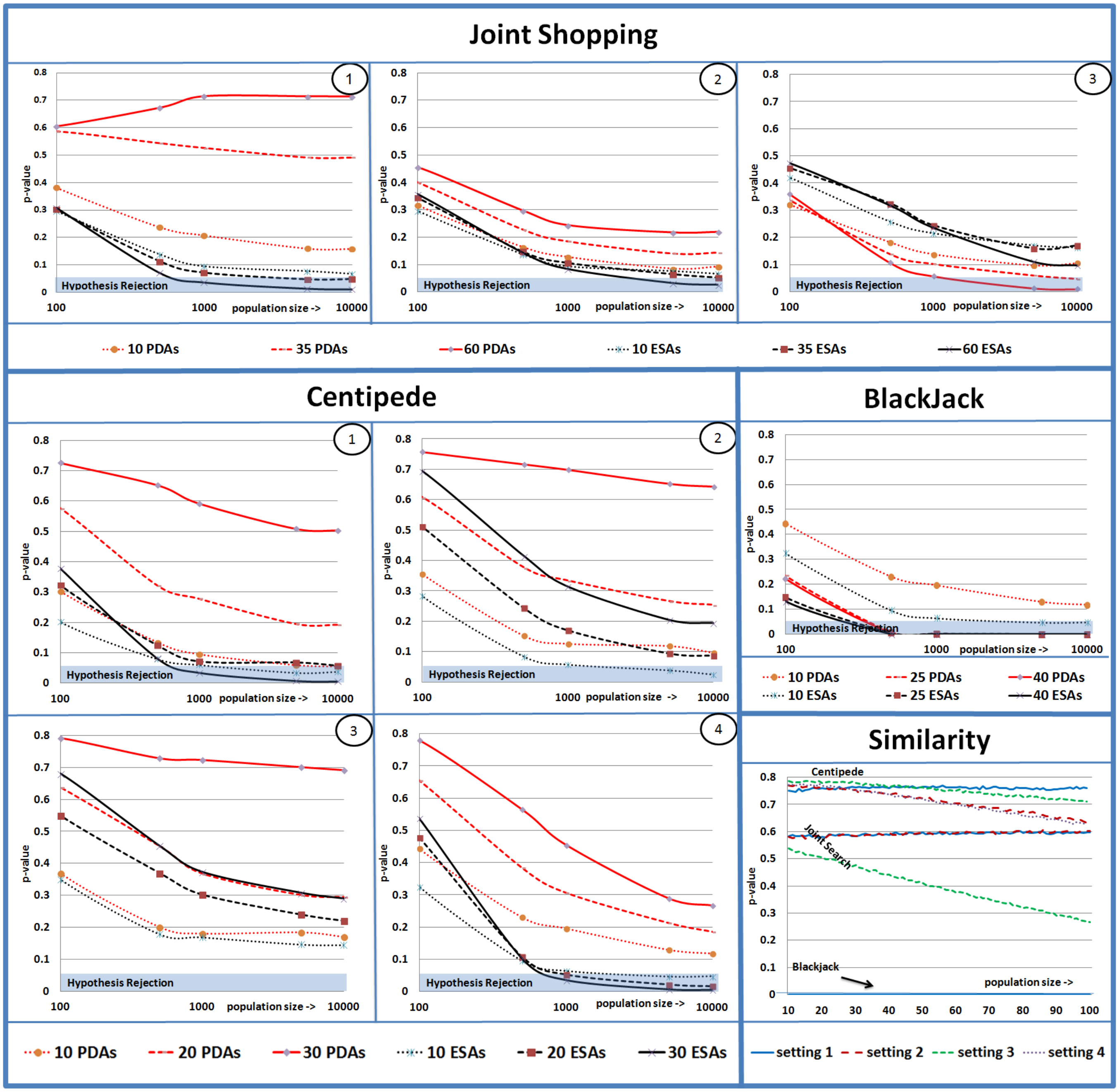

Figure 2: Results for the joint shopping, Blackjack and the centipede game using Kolmogorov-Smirnov test and similarity analysis between computer science students and the target population from AMT.

Game and Settings		PDAs		ESAs	
		Acceptance	Trendline	Acceptance	Trendline
Joint Shopping	1	=	≠	=	=
	2	=	≠	≠	=
	3	=	=	=	=
Blackjack	1	=	=	=	=
Centipede	1	=	=	=	=
	2	=	=	=	=
	3	=	=	=	=
	4	=	=	=	=

Table 4: Qualitative similarities and differences in the results of PDAs and ESAs when the analysis is performed based on K-S compared to the analysis based on the Wilcoxon rank sum test for all games and settings, with respect to acceptance/rejection of the null hypothesis (with $\alpha = 0.05$) and the trendline of the similarity based on the population size.

5. CONCLUSIONS

The results reported in the paper are encouraging in the sense that whenever the behavior of PDA' developers and individuals of the population of interest is sufficiently similar, PDA technology is an efficient means for extracting an accurate distribution of behaviors found in the population of interest. In all the experimental settings, where the statistical test indicated a reasonable similarity between the two populations, even with a moderate number of PDAs, we revealed distributions whereby the K-S $p\text{-}value$, when compared with the population of interest, was far from a rejection of the null hypothesis that the two samples are from the same population. The fact that the same set of PDAs was proved to be useful in more than a single setting of a given game further strengthens our findings that this method is an efficient alternative to other methods.

Indeed, before using PDAs the level of similarity obtained between the population of PDA developers and the population of interest needs to be checked, since this parameter has a critical effect on performance. This holds when using the same PDAs for different variants of the same problem setting. Naturally, this raises the question of whether the additional overhead associated with such a comparison does not make the whole process futile. Our results show that one can correctly determine whether or not the two populations are alike even with relatively small samples. Therefore, the method can and should be used with a relatively small overhead of extracting small samples from both populations by carefully checking the similarity and then proceeding with PDAs whenever a level of similarity considered to be reasonable is found.

The comparison between PDAs and ESAs in this sense offers several important insights. First, we observe that PDAs, whenever applicable, performed substantially better than ESAs. This is in spite of the fact that the ESAs are produced by a set of individuals that better represent the population of interest. This finding suggests that there is much more to PDA technology than simply expressing one's strategy. Another possible explanation for the success of PDAs compared to ESAs is that the PDA developer population is better at generalizing and expressing their strategy than the general population. We note that ESAs are not only less efficient, but also, as reported in the preceding section, more difficult to produce due to the substantial overhead in the form of the exceptionally low ratio of strategies that can actually be programmed from of the total number of strategies collected.

Future work warrants investigation of the possible tradeoff between the amount of distinguished PDAs required to efficiently substitute people in modeling the distribution of behaviors of a given population. Another venue is to better understand the limitations of elicited strategies and use PDAs as a tool to better elicit strategies from people.

6. REFERENCES

[1] D. Ariely. *Predictably Irrational*. Harper Perennial, 2010.

[2] A. Azaria, A. Richardson, A. Elmalech, and A. Rosenfeld. Automated agents' behavior in the trust-revenge game in comparison to other cultures. In *AAMAS*, 2014.

[3] M. Bertrand and S. Mullainathan. Do people mean what they say? implications for subjective survey data. *American Economic Review*, 91(2):67–72, 2001.

[4] P. Brewer, M. Huang, B. Nelson, and C. Plott. On the behavioral foundations of the law of supply and demand: Human convergence and robot randomness. *Experimental economics*, 5(3):179–208, 2002.

[5] M. Chalamish, D. Sarne, and S. Kraus. Programming agents as a means of capturing self-strategy. In *AAMAS*, pages 1161–1168, 2008.

[6] M. Chalamish, D. Sarne, and R. Lin. The effectiveness of peer-designed agents in agent-based simulations. *MAGS*, 8(4):349–372, 2012.

[7] M. Chhabra and S. Das. Learning the demand curve in posted-price digital goods auctions. In *AAMAS*, pages 63–70, 2011.

[8] A. Elmalech and D. Sarne. Evaluating the applicability of peer-designed agents in mechanisms evaluation. In *IAT*, pages 374–381, 2012.

[9] Y. Gal, B. Grosz, S. Kraus, A. Pfeffer, and S. Shieber. Agent decision-making in open-mixed networks. *Artificial Intelligence*, 174(18):1460–1480, 2010.

[10] G. Gigerenzer and R. Selten. *Bounded rationality: The adaptative toolbox*. Mit Press, 2002.

[11] C. Harries, J. Evans, and I. Dennis. Measuring doctors' self-insight into their treatment decisions. *Applied Cognitive Psychology*, 14:455–477, 2000.

[12] T. Ishida, Y. Nakajima, Y. Murakami, and H. Nakanishi. Augmented experiment: Participatory design with multiagent simulation. In *IJCAI*, pages 1341–1346, 2007.

[13] D. Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

[14] R. Katz and S. Kraus. Efficient agents for cliff edge environments with a large set of decision options. In *AAMAS*, pages 697–704, 2006.

[15] G. Kempen, M. Van Heuvelen, R. Van den Brink, A. Kooijman, M. Klein, P. Houx, and J. Ormel. Factors affecting contrasting results between self-reported and performance-based levels of physical limitations. *Age and Ageing*, 25(6):458–464, 1996.

[16] R. Lin, Y. Gal, S. Kraus, and Y. Mazliah. Training with automated agents improves people's behavior in negotiation and coordination tasks. *DSS*, 2013.

[17] R. Lin and S. Kraus. Can automated agents proficiently negotiate with humans? *CACM*, 53(1):78–88, 2010.

[18] R. Lin, S. Kraus, Y. Oshrat, and Y. Gal. Facilitating the evaluation of automated negotiators using peer designed agents. *Proc. of AAAI*, 2010.

[19] M. Mash, I. Rochlin, and D. Sarne. Join me with the weakest partner, please. In *WI-IAT*, pages 17–24, 2012.

[20] R. D. McKelvey and T. R. Palfrey. An experimental study of the centipede game. *Econometrica*, 60(4):803–836, 1992.

[21] S. Musse and D. Thalmann. Hierarchical model for real time simulation of virtual human crowds. *IEEE TVCG*, 7(2):152–164, 2001.

[22] G. Paolacci, J. Chandler, and P. G. Ipeirotis. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5), 2010.

[23] A. Rosenfeld and S. Kraus. Modeling agents based on aspiration adaptation theory. *JAAMAS*, 24(2):221–254, 2012.

[24] R. W. Rosenthal. Games of perfect information, predatory pricing and the chain-store paradox. *Economic Theory*, 25:92–100, 1981.

[25] L. Schvartzman and M. Wellman. Learning improved entertainment trading strategies for the tac travel game. *In Proc. of AMEC/TADA*, 59:195–210, 2010.

[26] R. Selten, M. Mitzkewitz, and G. Uhlich. Duopoly strategies programmed by experienced players. *Econometrica*, 65(3):517–555, 1997.

[27] I. Sharpanskykh and J. Treur. Abstraction relations between internal and behavioural agent models for collective decision making. *WIAS*, 10(4):465–484, 2012.

[28] S. Siegel. *Non-Parametric Statistics for the Behavioral Sciences*. McGraw-Hill, 1956.

[29] H. A. Simon. Theories of bounded rationality. *Decision and organization*, 1:161–176, 1972.

[30] T. Takahashi, S. Tadokoro, M. Ohta, and N. Ito. Agent based approach in disaster rescue simulation - from test-bed of multiagent system to practical application. In *RoboCup 2001: Robot Soccer World Cup V*, pages 102–111, 2002.

[31] D. Terzopoulos, X. Tu, and R. Grzeszczuk. Autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life*, 1(4):327–351, 1994.

[32] B. Ulicny and D. Thalmann. Towards interactive real-time crowd behavior simulation. *Computer Graphics Forum*, 21(4):767–775, 2002.