Predicting Plans and Actions in Two-Player Repeated Games

Najma Mathema, Michael A. Goodrich, and Jacob W. Crandall
Computer Science Department
Brigham Young University
Provo, UT, USA

Abstract

Artificial intelligence (AI) agents will need to interact with both other AI agents and humans. Creating models of associates help to predict the modeled agents' actions, plans, and intentions. This work introduces algorithms that predict actions, plans and intentions in repeated play games, with providing an exploration of algorithms. We form a generative Bayesian approach to model S#. S# is designed as a robust algorithm that learns to cooperate with its associate in 2 by 2 matrix games. The actions, plans and intentions associated with each S# expert are identified from the literature, grouping the S# experts accordingly, and thus predicting actions, plans, and intentions based on their state probabilities. Two prediction methods are explored for Prisoners Dilemma: the Maximum A Posteriori (MAP) and an Aggregation approach. MAP ($\approx$ 89% accuracy) performed the best for action prediction. Both methods predicted plans of S# with $\approx$ 88% accuracy. Paired T-test shows that MAP performs significantly better than Aggregation for predicting S#'s actions without cheap talk. Intention is explored based on the goals of the S# experts; results show that goals are predicted precisely when modeling S#. The obtained results show that the proposed Bayesian approach is well suited for modeling agents in two-player repeated games.

When agents interact, it is useful for one agent to have an idea of what other agents are going to do, what their plans are, and what intentions guide both their plans and actions. This work creates agent models that allow utilizing the observations from their past interactions to predict the modeled agent's actions, plans, and intentions to develop algorithms that: (a) predict actions of another agent and (b) identify their plans and intent.

The main concept of the work is based on the perspective from the literature in which *intentions* are the reasons behind *actions*, and that *plans* are the means for mapping intentions to actions (Perner 1991; Bratman 1987; Malle and Knobe 1997). The overarching objective of the research is to infer another agent's plans, goals, intentions and predict behavior (actions) based on these inferences and using observations of their actions in an environment. If an AI agent is able to predict future actions, the agent can plan ahead for appropriate actions and hence be able to make better decisions for the future.

This work makes predictions in the context of Repeated Games (RGs). Game theory has been applied in numerous ways to understand human/agent behavior, relationships, and decision-making. RGs in game theory provide an environment for understanding and studying the relationship between agents because the game construct requires each agent to account for the consequence of its action on the future action of the other agent. The dilemma of whether to cooperate or to compete with each other has been extensively studied in the game Prisoners Dilemma in the literature of psychology, economics, politics and many other disciplines. Hence, the same game has been used for this study. Prior work (Crandall et al. 2018; Crandall 2014) introduced the S# algorithm, which is designed as a robust algorithm that learns to cooperate with its associate in many 2 by 2 matrix games. S# is built on top of S++ (Crandall 2014) with the ability to share costless, non-binding signals called "cheap talk" indicating its experts' intentionality. For better expert-selection, in each round prior to taking an action, the players get an opportunity to communicate by sharing their plans via cheap talk. This paper presents a model for predicting actions, plans, and intents assuming the agent to be modeled is an S# agent. S# is studied because it is a highly effective algorithm in RGs and it uses explicit models of planners (called "experts") that are motivated by specific designer intentions (Crandall et al. 2018; Oudah et al. 2018). In the context of modeling S#'s behavior in RGs, we use a generative Bayesian model, which assumes that agents have a number of internal states defining the "state of mind" used to select what action they would want to take given the observations they see. The observations are (a) the speech acts/proposals via cheap talk that the players share with each other prior to taking their action and (b) the actions taken by both the S# agent and the agent with whom S# is interacting. Table 1 shows a few interactions of S# against a human player ABL in Prisoners Dilemma.

The generative Bayesian model provides a probability distribution over the S# agent's internal states. This probability distribution can be used as input to algorithms that predict the most likely state, the most likely action, the most likely

Round	Player	Speech acts	Action	Payoff
35	S#	None	B	20
35	ABL	You betrayed me. Curse you.	D	20
36	S#	None	B	100
36	ABL	None	C	0
37	S#	In your face! I forgive you. Let's alternate between AC and BC. This round, let's play BC. Do as I say or I will punish you.	B	100
37	ABL	Let's always play BD.	C	0
38	S#	Excellent. This round, let's play AC.	A	60
38	ABL	None	C	60

Table 1: S# vs human player ABL in the Prisoner's Dilemma

plan, and the most likely intention. Additionally, this kind of Bayesian model is not dependent on the type of RGs, and thus could be used for many two-player RGs. Since the model is based on observations in the environment and employs Bayesian reasoning, it does not require a huge dataset to train the model for better performance. Two types of algorithms for translating the distribution into predictions will be explored: (a) a Maximum A Posteriori (MAP) estimate of the most likely state, which implicitly identifies an action, plan, and intention; and (b) estimates of actions, plans, and intentions that aggregate probability over related states.

Related Literature

A variant of RGs implemented by Crandall et al. (Crandall et al. 2018; Oudah et al. 2018), called RGs with "cheap talk", allows each player to share messages/ proposals with the other player before actions are taken.

Intentional Action

Consider the motivation of using the notion of "intentionality" from folk psychology as the basis for modeling other agents. As per (de Graaf and Malle 2017), people regard "Autonomous Intelligent Systems" as intentional agents; people, therefore, use the conceptual and psychological means of human behavior explanation to understand and interact with them. Folk Psychology suggests that it is the belief, desire, and intention of humans that control human behavior and that our intention is the precursor to the action we take (Perner 1991). Hence, inferring the intent behind a particular action allows a human to infer the plans and goals of the agent. In this context, *intent* is associated with the commitment to achieve a particular goal through a plan (Bratman 1987). Once an intent is formed and a plan is selected to achieve the desire, an "intentional" action is one that is derived as an instrumental means for moving towards the intent.

M. de Graaf et al. mention that so-called "Autonomous Intelligent Systems" exhibit traits like planning and decision making, and hence are considered "intentional agents" (de Graaf and Malle 2017). They further claim that the behaviors of intentional agents can be explained using the human conceptual framework known as *behavior explanation*.

Related to the literature on intentional agents is work in "folk psychology" (Perner 1991; Bratman 1987; Malle and Knobe 1997), in which agent *beliefs*, *desires*, and *intentions* are used to explain how and why agents choose actions towards reaching their goal. Baker et al. (2011) presented a Bayesian model of human Theory of Mind based on inverse planning to make joint inferences about the agents' desires and beliefs about unobserved aspects of the environment. Similar to our work, they model the world as a kind of Markov Decision Process and use the observations in the environment to generate posterior probabilities about the environment. Additionally, by inverting the model, they make inferences about the agents' beliefs and desires.

Modeling Other Agents

Seminal work on modeling agents in the field of game theory was presented in (Axelrod and Hamilton 1981). Axelrod's models allow strategies to play against each other as agents to determine the winning strategy in Prisoners Dilemma tournaments. Early work on agent modeling tended to focus on equilibrium solutions for games and has now extended to various fields of computer science like (Lasota et al. 2017; Stone and Veloso 2000; Kitano et al. 1997).

One modeling approach is to predict action probabilities for the modeled agent, an early example which is the Fictitious Play algorithm (Brown 1951). In contrast to the simple empirical probability distributions of fictitious play, other authors have worked on making action predictions by learning the action probabilities of the opponent conditioned on their own actions (Sen and Arora 1997; Banerjee and Sen 2007).

Similar to our research objective, which is to be able to predict the next moves of the opponent, Gaudesi et al. (2014) worked on an algorithm called Turan to model the opponent player using finite state machines. The work in (Deng and Deng 2015) studies Prisoners Dilemma as a game with incomplete information and using Bayes rule and past interaction history to form a possibility distribution table for each players' choice to predict the players' choices. Park et al. (2016) assert that building precise models of players in Iterated Prisoners Dilemma requires a good dataset, so they use a Bootstrap aggregation approach to generate new data randomly from the original dataset. Also, an observer uses active learning approach to model the behavior of its opponent.

Inferring Intent

There has not been much research in predicting or inferring intents of agents in RGs, but there has been previous work in predicting the intent of agents in various other fields. Intent in prior work relates to goals and plans. Kuhlman et al. (1975) talk about the goals of agents in mixed-motive games by identifying their motivational orientation (cooperative, individualistic, or competitive) based on their choice behavior in decomposed games. Thus, knowing the motive of the

subject, they use it to predict actions for Prisoners Dilemma. The work in (Cheng, Lo, and Leskovec 2017) linked intent with goal specificity and temporal range when predicting intents in online platforms. Very recent research work uses deep-learning models for intent prediction (Qu et al. 2019; Pîrvu et al. 2018). Rabkina et al. (2013) used a computational model based on analogical reasoning to enable intent recognition and action prediction. Other methods that use Bayesian models for intent prediction include (Mollaret et al. 2015; Tavakkoli et al. 2007; Rios-Martinez et al. 2012).

Modeling Framework

Bayesian Graphical Model

A Bayesian graphical model is used to model S#. The model begins with a prior probability distribution over possible S# agent states and then propagates that distribution using observations of actions and speech-acts/proposals.

The structure of the Bayesian model is illustrated in Figure 1. The model makes it evident that future predictions are based on the present state and immediate observations. Understanding this model is made easier by comparing it to a Hidden Markov Model (HMM). An HMM is a five-tuple

$$\text{HMM} = \Big(S, O, p(s_0), p(s_{t+1}|s_t), p(o_t|s_t)\Big),$$

where S is a finite set of (hidden) states, O is a finite set of observations, $p(s_0)$ is the initial state distribution (i.e., the distribution over states at time $t = 0$), $p(s_{t+1}|s_t)$ represents the transition probability function that describes how states change over time, and $p(o_t|s_t)$ is the emission probabilities (i.e., the probability that a given observation o_t is generated by hidden state s_t). An HMM is one of the most simple dynamic Bayesian models because it describes how states change over time. A common application of HMMs is to try to infer a most likely hidden state from a series of observations.

Like an HMM, our model is also a dynamic model, but the inference task is slightly different and so are the model elements. The proposed model differs from the traditional HMM in two ways: First, the Bayesian model makes two state transitions at a single time step, that is, there are two hidden states at each time step. Second, there is an external input to the model. Figure 1 illustrates the proposed model. In the figure, the player being modeled is denoted by a subscript '-i' whereas the player's associate (in the game) is denoted by a subscript 'i'.

The Bayesian model is a tuple with seven elements,

$$\begin{aligned} BModel \;=\; \Big(&S, O, \Sigma, p(\hat{s}_{-i}(t)|s_{-i}(t), z_{-i}(t), z_i(t)),\\ &p(s_{-i}(t+1)|\hat{s}_{-i}(t), a_i(t), a_{-i}(t)),\\ &p(z_{-i}(t)|s_{-i}(t)), p(a_{-i}(t)|\hat{s}_{-i}(t))\Big). \end{aligned}$$

As with the HMM, S represents the set of states and O represents the set of observations. The set of states has two different kinds of states, $s_{-i} \in S$, which represent propositional states (states from which speech acts are generated) and $\hat{s}_{-i} \in S$, which represent action states (states from which game actions are chosen). The set of observations O has two different kinds of observations: (1) the speech acts/proposals, $\{z_{-i}\} \in O$, shared by the player being modeled and (2) the action, $\{a_{-i}\} \in O$, taken by the player being modeled. Σ is the set of exogenous inputs to the model, which consists of the observed actions and speech acts of the other player in the game, represented by a_i and z_i, respectively.

As mentioned earlier, the model of agent $-i$ is based on a time series with two types of hidden states, $s_{-i}(t)$ and $\hat{s}_{-i}(t)$. The proposed model takes two state transitions at a single time step. For a single time step, the first state transition occurs from $s_{-i}(t)$ to $\hat{s}_{-i}(t)$ based on the observation of what proposals are shared. $\hat{s}_{-i}(t)$ is a kind of temporary state for S#, from which it generates its aspiration level to choose the expert to play the game further. The next state transition from $\hat{s}_{-i}(t)$ to $s_{-i}(t+1)$ takes place based on the observation of the actions of the players. This state transition gives the prediction for the state at the next time step, which is then utilized in predicting the action of the modeled player for the next time step.

$S(t)$ is the set of states available to the modeling agent i at time t, and is given by the union of all the states in each expert's state machine,

$$S(t) = \cup_{j \in \mathcal{J}} S_{\phi_j}.$$

Conditional Probabilities

The Bayesian model makes use of the priors and conditional probabilities to find the posterior probability of the states after each observation. The priors about the states of the experts represent the agent's beliefs about the states. Ideally, prior probabilities should be fairly close to the true probabilities in real scenarios; prior probabilities affect the future computations and the predictions to be made. For computing the priors, the initial knowledge about S# and how the experts are formed and then selected is utilized.

Based on the game, S# generates a set of experts, which are essentially strategies that employ learning algorithm to select actions and generate and respond to speech acts, based on the state it is in. To select an expert to take an action, the "expert potential" needs to meet a specified aspiration level and the expert also needs to carry out plans which are congruent with its partner's last proposed plan. Thus, the priors for models of S# agents are the probabilities that S# selects a particular expert in the first round based on the expert's potential and uniform distribution over the aspiration level, with most of the probability assigned to the start state of the expert.

The following conditional probability elements describe the necessary components for designing the model. The notation used in the conditional probabilities is given by:

$$a : action,\ z : proposal,\ i : Partner,\ -i : S\#.$$

1. Sensor Model 1 (Speech): Given the current state of the agent, the sensor model provides the probability of seeing a particular proposal.

$$p(z_{-i}(t) \mid s_{-i}(t))$$

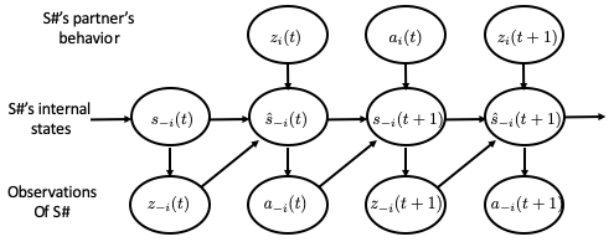

Figure 1: Modeling S#

2. Transition Model 1 (Reflection): For the same time step, the transition model is used for transitioning to a new state after the proposals are observed.

$$p(\hat{s}_{-i}(t) \mid s_{-i}(t), z_i(t), z_{-i}(t))$$

3. Sensor Model 2 (Action): Given the state of the agent, the sensor model provides the probability of seeing a particular action.

$$p(a_{-i}(t) \mid \hat{s}_{-i}(t))$$

4. Transition Model 2 (Update): Once the actions are taken by both the players, the update model encodes how a state transition occurs to a new state for the next time step.

$$p(s_{-i}(t+1) \mid \hat{s}_{-i}(t), a_i(t), a_{-i}(t))$$

The S# algorithm has Finite State Machines (FSMs) for each of the experts which define what speech acts are to be generated based on the internal states of the experts and the events in the game (Crandall et al. 2018). The events in the game could involve the event of selecting an expert or the events that affect individual experts. These FSMs have state transition functions that map the events, internal states, and speech outputs. Hence, for the game, the above conditional probabilities have been determined based on how S# acts in each event and also adding some uncertainty to make sure that other possible transitions are non-zero.

Updating the Probability Distribution – Bayes Filter Algorithm

An algorithm is needed to aggregate observations into a distribution over the hidden states. Since the proposed model is a dynamic Bayesian model, a Bayes Filter is an appropriate algorithm (Thrun, Burgard, and Fox 2005). The Bayes Filter is a general algorithm for estimating an unknown probability density function over time given the observations. The beliefs are computed recursively and are updated at each time step with the most recent observations. The algorithm presented in Algorithm 1 is the Bayes Filter Algorithm modified from the model in (Thrun, Burgard, and Fox 2005) to reflect the two hidden states.

Predicting Actions, Plans, and Intents

The Bayesian model and the Bayes Filter algorithm yield a probability distribution over hidden states in the model. From this distribution, one of the main tasks in this paper is to predict actions, plans, and intents. For each prediction, we will explore two methods: a MAP estimate and a more complete aggregation method. This subsection addresses how actions, plans, and intent can be predicted.

Algorithm 1: Bayes Filter Algorithm

function Bayes Filter ();
$\overline{bel}(s_0)$;
$bel(s_0) = \eta P(z_{-i}(0) \mid \overline{bel}(s_0))$;
for t **to** $len(observations)$ **do**
  for $s_t \in states$ **do**
    $\overline{bel}(\hat{s}_t) = \sum_{s_t} P(\hat{s}_t \mid z_i(t), z_{-i}(t), s_t) bel(s_t)$;
    $bel(\hat{s}_t) = \eta P(a_{-i}(t) \mid s_t) \overline{bel}(\hat{s}_t)$;
  end
  for $s_t \in states$ **do**
    $\overline{bel}(s_{t+1}) = \sum_{\hat{s}_t} P(s_{t+1} \mid a_i(t), a_{-i}(t), \hat{s}_t) bel(\hat{s}_t)$;
    $bel(s_{t+1}) = \eta P(z_{-i}(t) \mid s_{t+1}) \overline{bel}(s_{t+1})$;
  end
end

Predicting an Action The results presented in this paper have been obtained for the Prisoners Dilemma game, so the set of possible actions contains Cooperate, Defect.

MAP Estimate for Action Prediction The MAP estimate takes the maximum of all the probabilities over the actions available to predict an action, $\hat{a}_{\text{MAP}}$. The action probabilities are calculated by aggregating over all states as:

$$\hat{a}_{\text{MAP}} = \arg\max_a \sum_{s \epsilon S} P(s) P(a|s)$$

Aggregation Method for Action Prediction Each expert ϕ_j has different states $s \in S_{\phi_j}$ with the probability distribution over each of these states $P(s_{\phi_j})$ generated by the Bayesian model. Summing probabilities for all the states that belong to a given expert is done for each expert giving $p(\phi_j) = \sum_{s \in S_{\phi_j}} P(s)$. The expert with the maximum probability is identified, and the action is selected with the equation below:

$$\hat{a}_{\phi} = \arg\max_a \sum_{s \epsilon S_{\phi_j}} P(s) P(a|s)$$

Predicting a Plan We can categorize each expert as having a *follower type* or a *leader type*, as per the categorization of plan in (Littman and Stone 2001). A "leader type" creates strategies that will influence its partner to play a particular action by playing a trigger strategy that induces its partner to comply or be punished. Trigger strategies are the ones where a player begins by cooperation but defects for a predefined period of time when the other player shows a certain level of defection (opponent triggers through defection). The experts have a punishment stage in their state diagrams. The punishment phase is the strategy designed to minimize the partner's maximum expected payoff. The punishment phase persists from the time the partner deviates from the offer until the sum of its partner's payoffs (from the time of the deviation) is below what it would have obtained had it not deviated from the offer (Crandall et al. 2018). Hence the partner's optimal strategy would be to follow the offer.

A "follower type" expects its partner to do something and it plays the best response to their move. A follower assumes that its partner is using a trigger strategy. That means it assumes that its partner will propose an offer which is expected to be followed or else it might be punished. Following an offer may require the player to play fair (both getting the same payoff), to bully (demanding higher payoff than its associate) or to be bullied (accept lower payoffs than its associate).

Two approaches can be used to estimate the leader/follower plan being used: MAP and Aggregation.

MAP Estimate for Plan Prediction Let $\theta(\phi_i) \in \{$ leader, follower$\}$ indicate the "type" of expert ϕ_i. The plan is then the most probable type. For this we first identify the MAP estimate for which expert is most likely and then select that expert's type,

$$\hat{\theta} = \theta\Big(\arg\max_{\phi_i} \sum_{s \in S_{\phi_i}} P(s) \Big).$$

Aggregation Method for Plan Prediction Similar to how the probabilities of actions could be aggregated across states, we can aggregate probabilities across plan types and then choose the most likely type as follows:

$$\hat{\theta} = \arg \max_{\text{type} \in \{\text{lead, foll}\}} \sum_{\theta(\phi_i) == \text{type}} \sum_{s \in S_{\phi_i}} P(s).$$

Predicting Intent Each expert can be categorized by the goal it seeks to achieve by adopting its strategy. This is similar to categorizing agents by plan type, but the categorization is by intent type. We identify the intent using the Bayesian model of S# using the simple rule: the intent of S# is the intent of the expert it uses to achieve its goal. S# experts fall into two goal types: "Maximizing Payoff" and "Maximizing Fairness" (by minimizing the difference between the two player's payoffs).

Intent can be predicted by identifying the intent type of the most likely expert using the MAP estimate, or by aggregating probabilities over intent types and then selecting the most probable type. These two prediction methods are exactly analogous to predicting plan type from the previous subsection so details are omitted.

Experiments, Results and Discussion

Data preparation

The dataset used in this work is from previous work by Crandall et al. (2018) on RGs with cheap talk. Interaction logs have been recorded for human-human interaction and human interaction with S#. Two players play Prisoners Dilemma against each other, each game lasting 51 rounds. For each round, each player gets an opportunity to share messages before taking their actions (which could include their plan to play a particular action, or anything they would want to say to their opponent). Interaction logs are formed based on those game logs, consisting of payoffs, cheap talk and actions played by the players in each gameplay. There are a total of 24 interactions, 12 human-human games and 12 human-S# games, lasting 51 rounds each.

Another dataset is formed by having the strategies shown in Table 2 play against both the S# and human players. This dataset is used to compare the predictions of the proposed graphical Bayesian Model to evaluate its performance.

Predicting Intent and Plans

Two approaches were used for predicting the intent and the plan of the players for the repeated Prisoners Dilemma game: the MAP and aggregation method. In our experiments, both the methods for predicting intent predicted "Maximize payoff" as the intent of the players for all interactions. For predictions for S#, the models' predictions comply with the actual intent of the experts of S#. This is because the experts of S# were designed in (Crandall et al. 2018) with the intent to Maximize Payoff, except the Bouncer strategy which is never initialized for the Prisoners Dilemma game (Bouncer is relevant for other repeated games).

For validating the plan prediction, the interaction history was run through S# to see which of the experts were selected during each interaction, and hence the corresponding plan followed by the expert was considered as the true plan followed.

When used for humans, the model also predicts the intent to be "Maximizing Payoff". Unfortunately, we do not have measures to evaluate the intent prediction for humans for this game, which could be considered one of the limitations in our work. The intent prediction is based on the intent of the experts of S#. It would have been interesting to evaluate the intent of the players from a different perspective like with respect to their personalities or motivational orientation as in the work (Kuhlman and Marshello 1975), where the goal of cooperative, competitive, and individualistic agents is to achieve joint gain, relative gain, and own gain respectively.

Both MAP and Aggregation approaches achieved an accuracy of $\approx$88% for predicting plans for S#. Paired T-test shows that the difference in average performance between the MAP approach and Aggregation for plan prediction is not big enough to be statistically significant (p = 0.0997). But for predictions without cheap talk, paired T-test show that the difference in average performance between the MAP

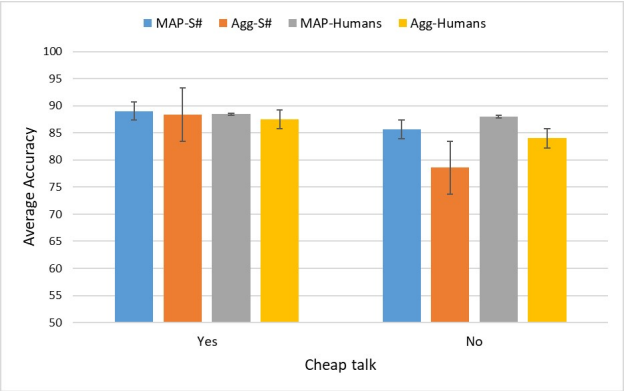

Figure 2: Average action prediction accuracy.

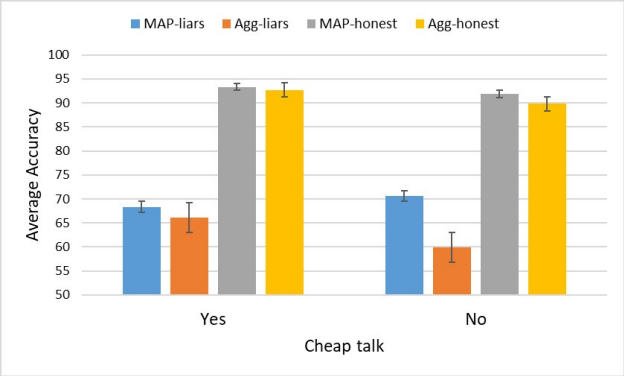

Figure 3: Action prediction comparison for players who lie.

approach and Aggregation is statistically significant (p = 0.0328), MAP being better.

Predicting actions

Average accuracy for action predictions The average accuracy for predicting actions using the MAP and Aggregation was calculated for modeling S#. Considering humans have similar internal states, the ability to form intentions, and plans to take actions, the same model was then used to model humans. MAP performed better than Aggregation and was able to predict the actions 89.05% of the time for S#, and 88.45% of the time for humans. We also tried experimenting on predicting the actions without using cheap talk and achieved an accuracy of 85.62% for S# and 88.02% for humans. Figure 2 summarizes the results. Paired T-test shows that the difference in average performance between the MAP and Aggregation approaches is not big enough to be statistically significant (p = 0.338). However, without cheap talk, the paired T-test shows that MAP performs significantly better than Aggregation for predicting actions(p = 0.0328).

There were 7 predictions where the accuracy was less than 80%. Looking at the data, we found that this was because of players who lie (≈51% of the time on average). Lying refers to proposing a particular action but taking a different

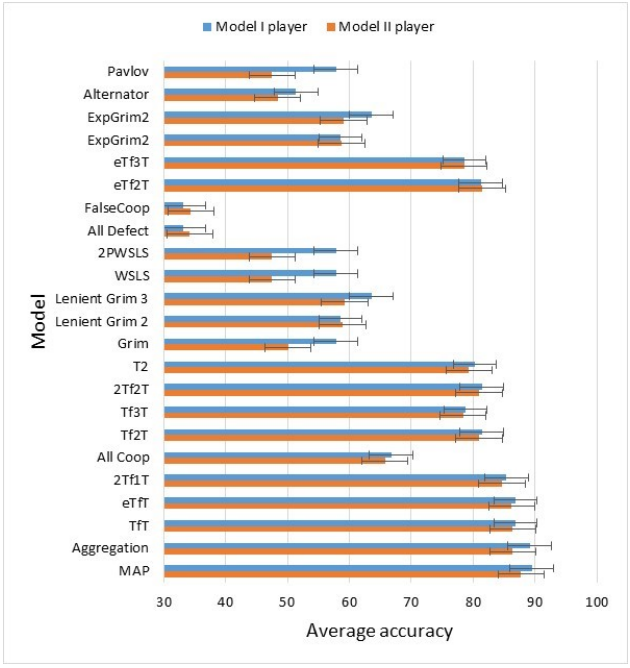

Figure 4: Comparison of action predictions for modeling Player 1 and 2 (Our model vs Others).

one during his/her turn. If we omit such interactions, MAP was 93.31% accurate for modeling humans who lie less frequently (18.6% of the time on average), and when we ignored the cheap talk, it was 92.2% accurate.

We see that the predictions were always better with cheap talk(without lying) as it provided more information about the interaction. It was interesting to see that the only time the accuracy bumped up when not using cheap talk, was when we modeled humans who lie. In this case, the accuracy of our model increased from 68.35% to 70.59% without cheap talk (for MAP approach). Thus, with this observation, we realize that our model performs well for modeling both S# and humans except for the agents who lie. The results are presented in Figure 3.

Comparing MAP predictions to fixed models

The action predictions from our model were compared with predictions from the fixed models presented in (Fudenberg, Rand, and Dreber 2012). The performance of the MAP and aggregation were comparable. The Bayesian model outperformed the fixed strategies. However, it was interesting to see that Tit for tat performed nearly as well as our model for the action predictions. Also, the Exploitive Tit for Tat also performed very close to that of Tit for Tat.

Figure 4 shows how our model performed in modeling players 1 and 2 vs other strategies. The player number simply indicates the player who goes first in each game. The following subsection presents how our model performs better in modeling dynamic behavior in agents as compared to the fixed strategy models.

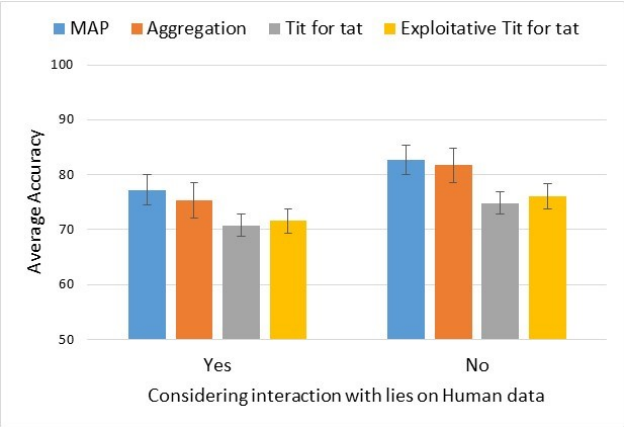

Figure 5: Performance with <=25% continuous repetition in actions.

Comparing MAP predictions excluding consistent interactions

Each of the interactions between all the players were observed carefully. Out of 48, 24 of the players had taken the same action repeatedly for more than 75% of the rounds. For fixed strategies like Tit for tat, it is easier to make correct predictions in such cases, as interactions were consistent for at least 75% of the rounds. So, another analysis was performed where we considered only those interactions which had more variance in its actions in each of the rounds. More specifically, interactions with only 25% or fewer continuous repetition of the same action were taken. The results are presented in Figure 5. Previously, we observed that Tit for tat and exploitative Tit for tat performed very close to our models. But as we compare the interactions with more variance in actions across the rounds, our models were able to perform significantly better as given by paired T-test ($p = 0.0160$). In addition, as we disregarded the interactions including lies, the performance of MAP was increased to $\approx$82%, and that of Aggregation increased to $\approx$81%.

Conclusion

This paper presented a graphical generative Bayesian model that models the S# algorithm for two-player repeated games. The highlight of the model is its ability to model the internal states of S#, considering each observation to calculate the posterior state probabilities, which could also be used to model humans. The other benefit of using this kind of model is that it is game independent, so it could be used for any two-player repeated game.

In comparison with other strategy models, the MAP approach on the Bayesian model performed the best in predicting actions for the Prisoners Dilemma game. It could better model the dynamic actions of players as compared to the other fixed strategy models. Also, for both plan and action prediction, MAP performed significantly better without cheap talk, i.e. when it is an ordinary repeated game. Additionally, obtaining a high accuracy for plans and intent prediction using the different approaches based on the same model, we can say that this Graphical Bayesian Model shows promise for modeling agents in two-player repeated games.

Strategy	Description
Always Cooperate	Always play C
Tit-for-Tat (TFT)	Play C unless partner played D last round
TF2T	C unless D played in both last 2 rounds
TF3T	C unless D played in both last 3 rounds
2-Tits-for-1-Tat	Play C unless partner played D in either of the last 2 rounds (2 rounds of punishment if partner plays D)
2-Tits-for-2-Tats	Play C unless partner played 2 consecutive Ds in the last 3 rounds (2 rounds of punishment if D played twice in a row)
T2	Play C until either player plays D, then play D twice and return to C
Grim	Play C until either plays D, then play D
Lenient Grim 2	Play C until 2 consecutive rounds occur in which either played D, then play D
Lenient Grim 3	Play C until 3 consecutive rounds occur in which either played D, then play D
Perfect TFT/Win-StayLose-Shift	Play C if both players chose the same move last round, otherwise play D
Perfect Tit-for-Tat with 2 rounds of punishment	Play C if both players played C in the last 2 rounds, both players played D in the last 2 rounds, or both players played D 2 rounds ago and C last round. Otherwise play D
Always Defect	Always play D
False cooperator	Play C in the first round, then D forever
Expl. Tit-for-Tat	Play D in the first round, then play TFT
Expl. Tit-for-2-Tats	Play D in the first round, then play TF2T
Expl. Tit-for-3-Tats	Play D in the first round, then play TF3T
Expl. Grim2	Play D in the first round, then play Grim2
Expl. Grim3	Play D in the first round, then play Grim3
Alternator	DCDC ...
Pavlov	Start with C, Always play C if partner does not play D

Table 2: Existing strategies for Prisoners Dilemma.

However, the model had some limitations. It was not able to detect its partners lying in the game and hence did not perform very well in such situations. A future enhancement could include creating experts for S# having the ability to deal with lies in the game. Also, further exploration of the intent of players, based on other dimensions is necessary. It would have been interesting to study intent from a different perspective such as based on the personality of the players, and how the intention of players change over time.

Acknowledgements

This work was supported in part by the U.S. Office of Naval Research under Grant #N00014-18-1-2503. All opinions, findings, conclusions, and recommendations expressed in this paper are those of the author and do not necessarily reflect the views of the Office of Naval Research.

References

Axelrod, R., and Hamilton, W. D. 1981. The evolution of cooperation. *science* 211(4489):1390–1396.

Baker, C.; Saxe, R.; and Tenenbaum, J. 2011. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, volume 33.

Banerjee, D., and Sen, S. 2007. Reaching pareto-optimality in prisoner's dilemma using conditional joint action learning. *Autonomous Agents and Multi-Agent Systems* 15(1):91–108.

Bratman, M. 1987. *Intention, plans, and practical reason*, volume 10. Harvard University Press Cambridge, MA.

Brown, G. W. 1951. Iterative solution of games by fictitious play. *Activity analysis of production and allocation* 13(1):374–376.

Cheng, J.; Lo, C.; and Leskovec, J. 2017. Predicting intent using activity logs: How goal specificity and temporal range affect user behavior. In *Proceedings of the 26th International Conference on World Wide Web Companion*, 593–601. International World Wide Web Conferences Steering Committee.

Crandall, J. W.; Oudah, M.; Ishowo-Oloko, F.; Abdallah, S.; Bonnefon, J.-F.; Cebrian, M.; Shariff, A.; Goodrich, M. A.; Rahwan, I.; et al. 2018. Cooperating with machines. *Nature communications* 9(1):233.

Crandall, J. W. 2014. Towards minimizing disappointment in repeated games. *Journal of Artificial Intelligence Research* 49:111–142.

de Graaf, M., and Malle, B. 2017. How people explain action (and ais should too). In *Proceedings of the Artificial Intelligence for Human-Robot Interaction (AI-for-HRI) fall symposium*.

Deng, X., and Deng, J. 2015. A study of prisoner's dilemma game model with incomplete information. *Mathematical Problems in Engineering* 2015.

Fudenberg, D.; Rand, D. G.; and Dreber, A. 2012. Slow to anger and fast to forgive: Cooperation in an uncertain world. *American Economic Review* 102(2):720–49.

Gaudesi, M.; Piccolo, E.; Squillero, G.; and Tonda, A. 2014. Turan: evolving non-deterministic players for the iterated prisoner's dilemma. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, 21–27. IEEE.

Kitano, H.; Tambe, M.; Stone, P.; Veloso, M.; Coradeschi, S.; Osawa, E.; Matsubara, H.; Noda, I.; and Asada, M. 1997. The robocup synthetic agent challenge 97. In *Robot Soccer World Cup*, 62–73. Springer.

Kuhlman, D. M., and Marshello, A. F. 1975. Individual differences in game motivation as moderators of preprogrammed strategy effects in prisoner's dilemma. *Journal of personality and social psychology* 32(5):922.

Lasota, P. A.; Fong, T.; Shah, J. A.; et al. 2017. A survey of methods for safe human-robot interaction. *Foundations and Trends® in Robotics* 5(4):261–349.

Littman, M. L., and Stone, P. 2001. Leading best-response strategies in repeated games. In *In Seventeenth Annual International Joint Conference on Artificial Intelligence Workshop on Economic Agents, Models, and Mechanisms*. Citeseer.

Malle, B. F., and Knobe, J. 1997. The folk concept of intentionality. *Journal of experimental social psychology* 33(2):101–121.

Mollaret, C.; Mekonnen, A. A.; Ferrané, I.; Pinquier, J.; and Lerasle, F. 2015. Perceiving user's intention-for-interaction: A probabilistic multimodal data fusion scheme. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. IEEE.

Oudah, M.; Rahwan, T.; Crandall, T.; and Crandall, J. W. 2018. How ai wins friends and influences people in repeated games with cheap talk. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Park, H., and Kim, K.-J. 2016. Active player modeling in the iterated prisoner's dilemma. *Computational intelligence and neuroscience* 2016:38.

Perner, J. 1991. *Understanding the representational mind.* The MIT Press.

Pîrvu, M. C.; Anghel, A.; Borodescu, C.; and Constantin, A. 2018. Predicting user intent from search queries using both cnns and rnns. *arXiv preprint arXiv:1812.07324*.

Qu, C.; Yang, L.; Croft, W. B.; Zhang, Y.; Trippas, J. R.; and Qiu, M. 2019. User intent prediction in information-seeking conversations. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, 25–33. ACM.

Rabkina, I., and Forbus, K. D. 2013. Analogical reasoning for intent recognition and action prediction in multi-agent systems.

Rios-Martinez, J.; Escobedo, A.; Spalanzani, A.; and Laugier, C. 2012. Intention driven human aware navigation for assisted mobility.

Sen, S., and Arora, N. 1997. Learning to take risks. In *AAAI-97 Workshop on Multiagent Learning*, 59–64.

Stone, P., and Veloso, M. 2000. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots* 8(3):345–383.

Tavakkoli, A.; Kelley, R.; King, C.; Nicolescu, M.; Nicolescu, M.; and Bebis, G. 2007. A vision-based architecture for intent recognition. In *International Symposium on Visual Computing*, 173–182. Springer.

Thrun, S.; Burgard, W.; and Fox, D. 2005. *Probabilistic robotics*. MIT press.