

Using HMM in Strategic Games

Mario Benevides Isaque Lima Rafael Nader Pedro Rougemont

Systems and Computer Engineering Program and Computer Science Department

Federal University of Rio de Janeiro, Brazil

In this paper we describe an approach to resolve strategic games in which players can assume different types along the game. Our goal is to infer which type the opponent is adopting at each moment so that we can increase the player's odds. To achieve that we use Markov games combined with hidden Markov model. We discuss a hypothetical example of a tennis game whose solution can be applied to any game with similar characteristics.

1 Introduction

Game theory is widely used to model various problems in economics, the field area in which it was originated. It has been increasingly used in different applications, we can highlight their importance in political and diplomatic relations, biology, computer science, among others. The study of game theory relies on describing, modeling and solving problems directly related to interactions between rational decision makers. We are interested in matches between two players that choose their actions in the best way, and know that each other do the same.

In this paper we propose a model which maps opponent players behavior as a set of states (types), each state having a pre-defined payoff table, in order to infer opponents next move. We address the problem in which the states cannot be directly observed, but instead, they can be estimated from the observations of the players actions.

The rest of this paper is organized as follows. In section 2 we show related works and the motivation to our study. In section 3, we present the necessary background in Hidden Markov Model. In section 4 we introduce our model and in section 5 we illustrate it using a tennis game example, whose solution can be applied to any game with similar characteristics. In appendix A, we provide a Game Theory tool kit for the reader not familiar with this subject.

2 Related Work

In this paper we try to solve a particular case of a problem known as Repeated Game with Incomplete Information, first introduced by Aumann and Maschler in 1960 [1], described below:

The game G is a Repeated Game where in the first round the state of Nature is chosen by a probability p and only player 1 knows this state. Player 2 knows the possible states, but doesn't know the actual state. After each round, both players know the action of each one, and play again.

In [8] it is studied Markov Chain Games with Lack of Information. In this game, instead of a probability associated to an initial state of Nature, we have a Markov Chain Transition Probability Distribution that changes the state through time. Both players know the action of each other at each round, but only one player knows the actual and past states and the payoff associated with those actions. The other player knows the Transition Probability Distribution. This game is a particular case of Markov Chain Game or

Stochastic Game [10], where both players know the actual state (or the last state and the transition probability distribution).

[8] presents some properties and solutions for this class of games, using the recurrence property of a Markov Chain and the Belief Matrix B .

Our problem is a particular case of Renault's game, where the lack of information is worse and the unaware player does not know the Transition Probability Distribution of the Markov Chain Game. With that, we can describe formally our problem as below:

The game G is a Repeated Game between two players where the Player 1 has a type that changes through the time. Each type has a probability distribution to other types that is independent of past states and actions. Each type introduces a different game with different payoffs. Player 1 knows his actual type, player 2 knows the types that player 1 can be, but does not know the actual type nor the transition between types. Both players know the action of each round.

As mentioned before, our game is a particular case of a Markov Chain Game, so the transition between types of player 1 follows the Markov Property. But as the player 2 doesn't know the state of player 1 we can't solve this as a Markov Chain Game (or Stochastic Game), and as the player 2 does not know the transition between the types we can not use the Markov Chain Game with Lack of Information.

To solve this problem we propose a model to this game and a solution that involves Hidden Markov Model. We compare our results with other ways to solve a problem like that.

3 Hidden Markov Model (HMM)

A Markov model is a stochastic model composed by a set of states to describe a sequence of events. It has the Markov property, which means that the process is memoryless, i.e., the next state depends only on the present state. A hidden Markov model [6, 7] is a Markov model in which the states are partially observable.

In a hidden Markov model we have a set of hidden states called hidden chain, that is exactly like a Markov chain, and we have a set of observable states. Each state in the hidden chain is associated to the observable states with a probability. We do not know the hidden chain, neither the actual state, but we know the actual observation. The idea is that after a series of observations we can get information about the hidden chain.

An example to illustrate the use of hidden Markov model is a three state weather problem (see figure 1). We are interested in help an office worker that is locked in the office for several days to know about the weather outside. Here we have three weather possibilities: rainy, cloudy and sunny, that changes through the time depending only on the actual state. The worker observes people coming to work with an umbrella sometimes and associates a probability to each weather based on that. With a series of umbrella observations, we would like to know about the weather, the hidden state.

Definition 1 A Hidden Markov Model $\Lambda = \langle A, B, \Pi \rangle$ consists of

- a finite set of states $Q = \{q_1, \dots, q_n\}$;
- a transition array A , storing the probability of the current state be q_i given that the previous state was q_j ;
- a finite set of observation states $O = \{o_1, \dots, o_k\}$;
- a transition array B , storing the probability of the observation state o_i be produced from state q_j ;
- a $1 \times n$ initial array Π , storing the initial probability of the model be in state q_i .

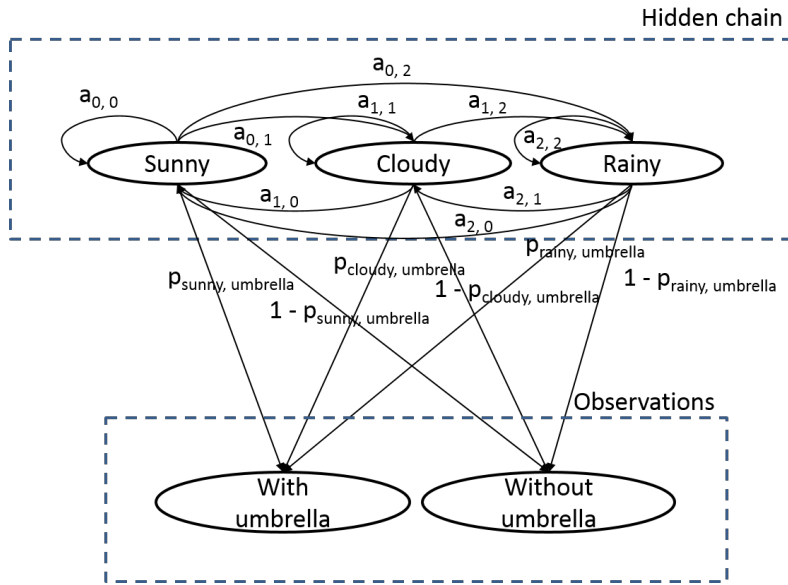


Figure 1: HMM Example

Based on a given set of observations y_i , and a predefined set of transitions b_{ij} , the Hidden Markov Models framework allows us to estimate the hidden transitions, labeled a_{ij} in the figure 2.

4 Hidden Markov Game (HMG)

In this section, we describe the Hidden Markov Game (HMG), which is inspired by the works discussed in section 2. We first define it for many player and then restrict it to two players.

Definition 2 A Hidden Markov Game G consists of

- A finite set of players $P = \{1, \dots, I\}$;
- Strategy sets S_1, S_2, \dots, S_I , one for each player;
- Type sets T_1, T_2, \dots, T_I , one for each player;
- A transition functions $\mathcal{T}_i : T_i \times S_1 \times \dots \times S_I \mapsto \mathcal{P}\mathcal{D}(T_i)$, one for each player;
- A finite set of observation state variables $\mathbf{O} = \{\mathcal{O}_1, \dots, \mathcal{O}_k\}$;
- Payoff functions $u_i : S_1 \times S_2 \times \dots \times S_I \times T_i \mapsto \mathfrak{R}$, one for each player;
- Probability Distribution $\pi_{i,j}$, representing the player i prior belief about the type of his opponent j , for each player j .

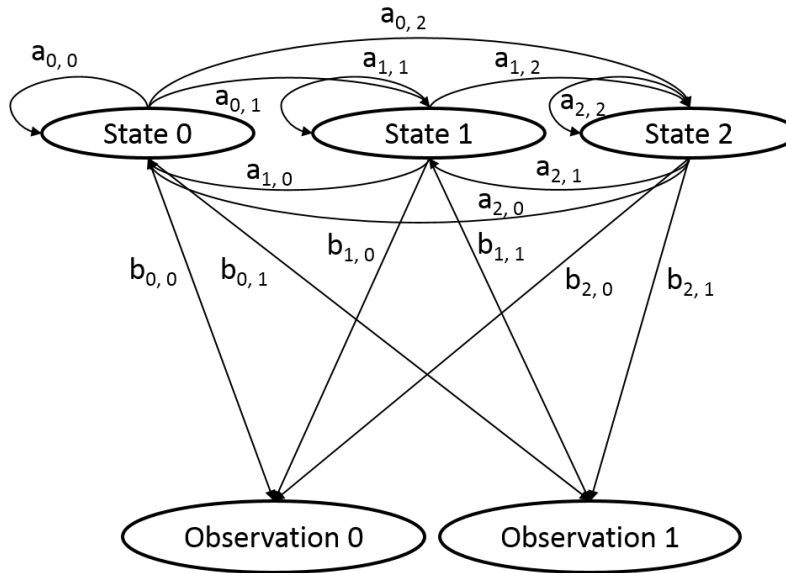


Figure 2: HMM Definition

Where $\mathcal{P}\mathcal{D}(T_i)$ is a probability distribution over T_i .

We are interested in a particular case of the HMG where we have two players and player 1 does not know his transition function \mathcal{T}_i . And the set of observation state variables $\mathbf{O} = S_2$

We are interested in a particular case of the HMG:

- two players
- player 1 does not know the opponent transition function
- The observation state variables \mathbf{O} for player 1 are equal to the set of strategies of the opponent

Problem: Given a sequence of observations on time of the observation state variables O^{t_0}, \dots, O^{t_m} , we want to know the probability, for player i , that $O^{t_{m+1}} = \mathcal{O}_j$, for $1 \leq j \leq k$.

In fact, a Hidden Markov Game can be seen as a Markov Game where a player does not know the probability distributions of the transitions between types. Instead, he knows a sequence of observations on time of the observable behavior of each opponent.

Due to this particularity, we cannot use the standard Markov Game tools to solve the game. The aim of our model is to provide a solution for the game partitioning the problem. In order to achieve that we infer the Markov chain at each turn of the game and then play in accordance to this Markov Game. In fact, if we had a one turn game, then we would have a Bayesian Game and we could calculate the Bayesian Nash Equilibrium and play according to it. This is what has been done in [9].

The inference of the transitions of the Markov chain is accomplished using a Hidden Markov Model HMM. The probability distribution associating each state of the Markov chain to the observable states are given by the Mixed Equilibrium of each matrix in each type. A Baum Welch Algorithm is used to infer the HMM.

Our Solution: Given a Hidden Markov Game G and a sequence of observations on time of the observation state variables O^{t_0}, \dots, O^{t_m} , we want to calculate the probability, for player i , that $O^{t_{m+1}} = \mathcal{O}_j$, for $1 \leq j \leq k$. This is accomplished following the steps below:

1. Represent the Hidden Markov Game G as a Hidden Markov Model each for player i as follows:
 - (a) *the finite set of states* $Q = \{q_1, \dots, q_n\}$ is the set the set $T_i \times S_1 \times \dots \times S_I$;
 - (b) *the transition array* A , *storing the probability of the current state be* q_j *given that the previous state was* q_i *is what we want to infer*;
 - (c) *the finite set of observation states* $O = \{o_1, \dots, o_k\}$ *is the set of observation state variables* $\mathbf{O} = \{\mathcal{O}_1, \dots, \mathcal{O}_k\}$;
 - (d) *the transition array* B , *storing the probability of the observation state* \mathcal{O}_h *be produced from state* $q_j = \langle T_j, s_1, \dots, s_I \rangle$ *is obtained calculating the probability of player* i *using strategy* $\langle s_1, \dots, s_I \rangle$ *in the Mixed Nash Equilibrium of game* T_j *and adding the probabilities, for each profile* $\vec{s}^{-i} \in f_i(\mathcal{O}_h)$;
 - (e) *the* $1 \times n$ *initial array* Π , *storing the initial probability of the model be in state* q_j *is the* $\pi_{i,j}$.
2. Use the Baum Welch algorithm to infer the matrix A ;
3. Solve the underline Markov Game and find the must probable type each opponent is playing;
4. If it is one move game then choose the observable state with the greatest probability. Else, play according to the Mixed Nash Equilibrium.

5 Application and Test

In order to illustrate our frame work we present an example of a tennis game. In this example, we have two tennis players, and we are interested in the server versus receiver situation. Here, we can see the possible set of strategies for each one as:

- Server can try to serve a central ball (center) or an open one (open).
- Receiver will try to be prepared to one of those actions.

We have different server types: aggressive, moderate or defensive, what means that the way he chooses the strategy will vary. For each type, the player evaluate each strategy in a different way. The player's type changes through the time as a Markov chain.

We are interested in helping the receiver to choose which action to take, but we need to know about opponent actual type.

We assume that the transitions between the hidden states and the observations are fixed, i.e., they are previously computed using the payoff matrix of each profile. We do that to reduce the number of loose variables. We compare ours results with the ones obtained by Bayesian game using the same trained HMM to compute the Bayesian output. We also compare our results with some more naive approaches like Tit-for-Tat (that always repeat the last action made by the opponent), Random Choice and More Frequently (choose the action that is more frequently used by the opponent).

	Open	Center
Open	0.65,0.35	0.89,0.11
Center	0.98,0.02	0.15,0.85

Table 1: Payoff Matrix (Aggressive Profile)

	Open	Center
Open	0.15,0.85	0.80,0.20
Center	0.90,0.10	0.15,0.85

Table 2: Payoff Matrix (Moderate Profile)

We used the following payoff matrixes, with two players: player 1 (column) and player 2 (row). We compute the mixed strategy to each profile, and consequentially compute the values of B. Next we present some scenarios that we use.

5.1 Scenarios

In this section we present some of the scenarios that we use.

The original HMM is only used to generate the observations of the game. We generate 10.000 observations, for each scenario, and after every 200 observations we compute the result of the game.

We used the metric proposed in [7] to compare the similarity of two Markov models.

5.1.1 Scenario 1 Aggressive Player

In order to illustrate the behavior of an aggressive player we model a HMM that has more probability to stay in the aggressive state.

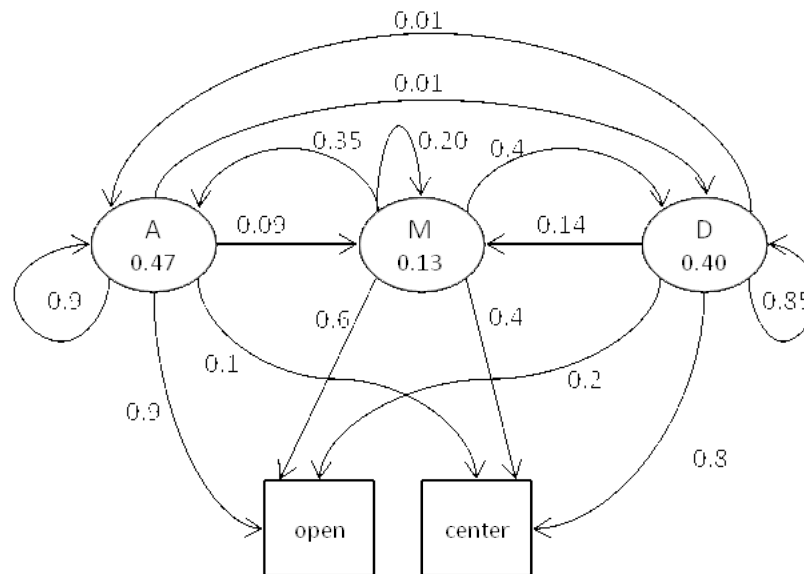


Figure 3: Original HMM (Aggressive Player)

	Open	Center
Open	0.10,0.90	0.55,0.45
Center	0.85,0.15	0.05,0.95

Table 3: Payoff Matrix (Defensive Profile)

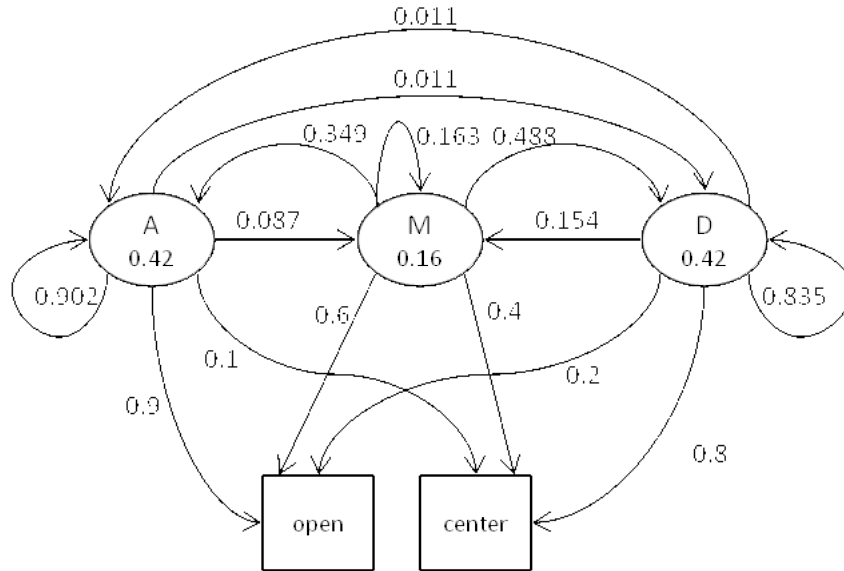


Figure 4: Trained HMM (Aggressive Player)

The difference between the original HMM and de trained HMM was 0,0033, i.e, the two HMM are very close, as we can empiric see in the picture below. As we can see on the table below, the proposed algorithm increase the player odds.

	Proposed Model	Bayesian Game	Random	More Frequently	Tit-for-Tat
hit rate	0,78	0,58	0,496	0,582	0,532

Table 4: Agressive Scenario Hit Rate

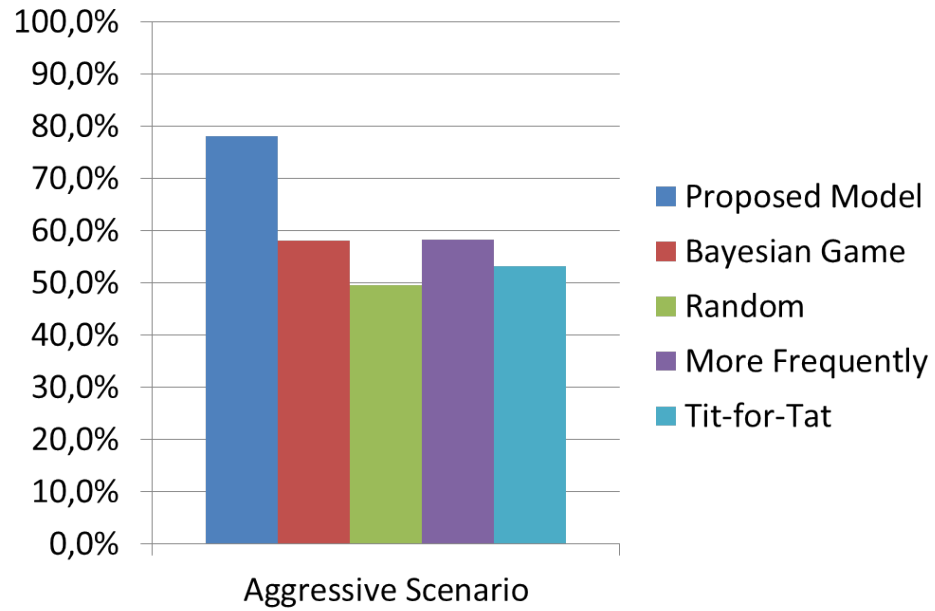


Figure 5: Agressive Scenario Graphic

5.1.2 Scenario 2 Defensive Player

In order to illustrate the behavior of a defensive player we model a HMM that has more probability to stay in the defensive state.

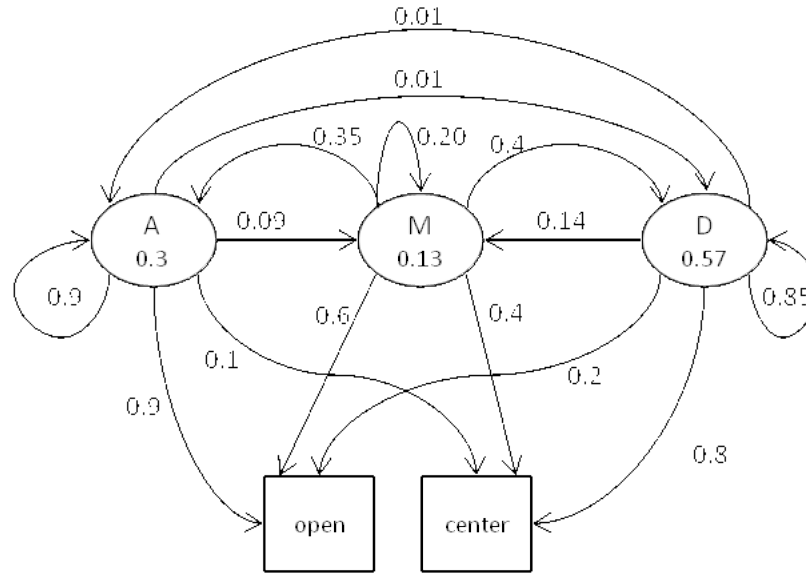


Figure 6: Original HMM (Defensive Player)

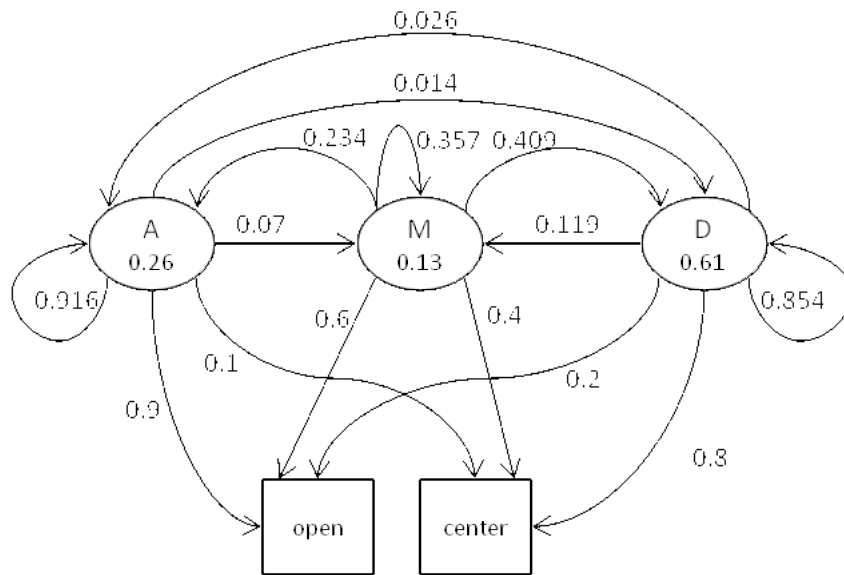


Figure 7: Trained HMM (Defensive Player)

Once again the proposed algorithm increase the player odds and the trained HMM is very close to the original HMM.

	Proposed Model	Bayesian Game	Random	More Frequently	Tit-for-Tat
hit rate	0,71	0,50	0,492	0,53	0,498

Table 5: Defensive Scenario Hit Rate

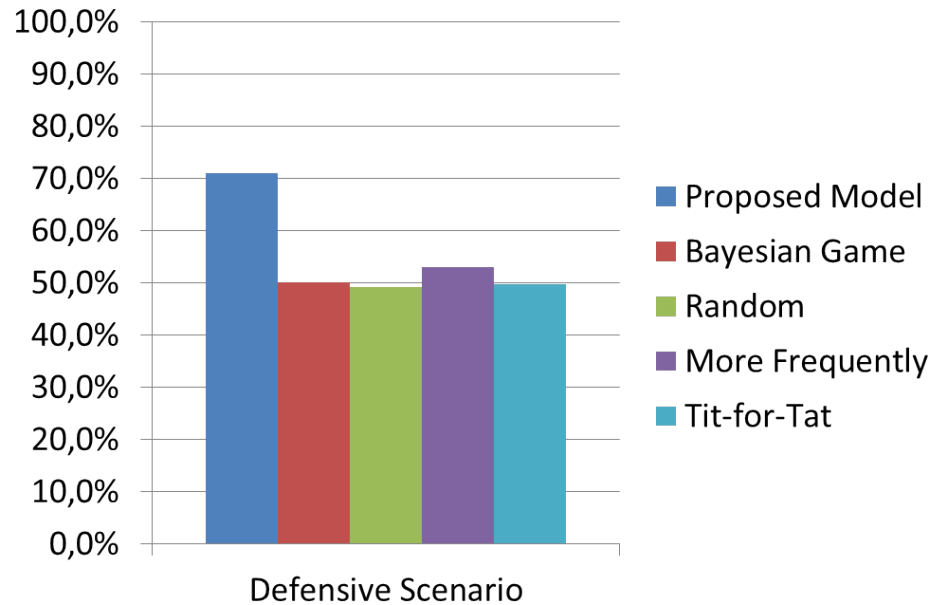


Figure 8: Defensive Player Graphic

The gain in these two scenarios are due to the fact that the proposed algorithm take into account the transitions between hidden states, so we have more information that help us to choose a better strategy.

6 Conclusions

In this work we have introduced a novel class of games called Hidden Markov Games, which can be thought as Markov Game where a player does not know the probability distributions of the transitions between types. Instead, he knows a sequence of observation on time of the observable behavior of each opponent. We propose a solution to our game representing it as a Hidden Markov Model, We use Baum Welsh algorithm to infer the probability distributions of the transitions between types. Finally, we solve the underline Markov Game.

In order to illustrate our approach, we present a tennis game example and solve it using our method. The experimental results indicates that our solution is quite good.

In this work we present a solution for a special case of Markov Chain Games or Stochastic Games with the lack of information and where the unaware player does not know the Transition Probability Distribution of the Markov Chain Game. We provide a solution based on Hidden Markov Models which

is computational and quite efficient. The drawback of our approach is the fact that we use the Nash Equilibrium of each type in a ad hoc way.

As future work we would like to prove that playing with the Nash equilibrium of actual type is an optimal choice or at least an equilibrium in the repeated game. Another problem to solve is to show that the choice of the uninformed player in playing with an equilibrium of the inferred type is an optimal choice or at least a Nash equilibrium of whole game.

A Game Theory Tool Kit

In this section, we present the necessary background on Game theory. First we introduce the concepts of Normal Form Strategic Game and Pure and Mixed Nash Equilibrium. Finally, we define Bayesian Games and Markov Games.

Definition 3 A Normal (Strategic) Form game G consists of

- A finite set of players $P = \{1, \dots, I\}$;
- Strategy sets S_1, S_2, \dots, S_I , one for each player;
- Payoff functions $u_i : S_1 \times S_2 \times \dots \times S_I \mapsto \mathfrak{R}$, one for each player.

A strategy profile is a tuple $s = s_1, s_2, \dots, s_I$ such that $s \in \mathcal{S}$, where $\mathcal{S} = S_1 \times S_2 \times \dots \times S_I$. We denote s_{-i} as the profile obtained from s removing s_i , i.e., $s_{-i} = s_1, s_2, \dots, s_{i-1}, s_{i+1}, \dots, s_I$

The following game is an example of normal (strategic) form game with two players Rose (Row) and Collin (Column) and both have two strategies s_1 and s_2 .

	s_1	s_2
s_1	3,3	2,5
s_2	5,2	1,1

Table 6: Normal Form Strategic Game

Definition 4 A strategy profile s^* is a pure strategy Nash equilibrium of G if and only if

$$u_i(s^*) \geq u_i(s_i, s_{-i})$$

for all players i and all strategy $s_i \in S_i$, where u is the payoff function.

Intuitively, a strategy profile is a Nash Equilibrium if for each player, he cannot improve his payoff changing his strategy alone.

The game presented in table 6 has two Nash Equilibrium (s_1, s_2) and (s_2, s_1) . Which one they should play? A possible answer could be to assign probabilities to the strategies, i.e., Rose could play s_1 with probability p and s_2 with $1 - p$, and Collin could play s_1 with probability q and s_2 with $1 - q$. This game has a mixed Nash equilibrium that is $p = 1/3$ and $q = 1/3$. We can calculate the expected payoff for both player and check that they cannot improve their payoff changing their strategy alone, i.e., this a Nash Equilibrium.

Other class of static games are strategic games of incomplete information also called Bayesian Games [4, 2]. The intuition behind this kind of game is that in some situations the player knows his payoff function but is not sure about his opponents function. He knows that his opponent is playing according to one type in a finite set of types with some probability.

Definition 5 A Static Bayesian game G consists of

- A finite set of players $P = \{1, \dots, I\}$;
- Strategy sets S_1, S_2, \dots, S_I , one for each player;
- Type sets T_1, T_2, \dots, T_I , one for each player;
- Payoff functions $u_i : S_1 \times S_2 \times \dots \times S_I \times T_i \mapsto \mathfrak{R}$, one for each player.
- Probability Distribution $P(t_{-i} | t_i)$, denoting the player i belief about his opponents types t_{-i} , given that his type is t_i .
- Probability Distribution π_i , representing the player i prior belief about the types of his opponents.

Markov Games [10, 5, 3] can be thought as a natural extension of Bayesian games, where we have transitions between types and a probability distributions over these transitions.

A Markov game can be defined as follows

Definition 6 A Markov game G consists of

- A finite set of players $P = \{1, \dots, I\}$;
- Strategy sets S_1, S_2, \dots, S_I , one for each player;
- Type sets T_1, T_2, \dots, T_I , one for each player;
- A transition functions $\mathcal{T}_i : T_i \times S_1 \times \dots \times S_I \mapsto \mathcal{P}\mathcal{D}(T_i)$, one for each player;
- Payoff functions $u_i : S_1 \times S_2 \times \dots \times S_I \times T_i \mapsto \mathfrak{R}$, one for each player;
- Probability Distribution π_i , representing the player i prior belief about the types of his opponents.

Where $\mathcal{P}\mathcal{D}(T_i)$ is a probability distribution over T_i .

References

- [1] R. J. Aumann & M. B. Maschler (1995): *Repeated Games with Incomplete Information*. MIT Press, Cambridge, MA. With the collaboration of Richard E. Stearns.
- [2] Robert Gibbons (1992): *A primer in game theory*. Harvester Wheatsheaf.
- [3] Michael L. Littman (1994): *Markov Games as a Framework for Multi-Agent Reinforcement Learning*. In William W. Cohen & Haym Hirsh, editors: *Proceedings of the Eleventh International Conference on Machine Learning*, Morgan Kaufmann, pp. 157–163.
- [4] M.J. Osborne & A. Rubinstein (1994): *A Course in Game Theory*. MIT Press.
- [5] M. Owen (1982): *Game Theory*. Academic Press. Second Edition.
- [6] L. R. Rabiner (1985): *A Probabilistic Distance Measure for Hidden Markov Models*. *AT&T Tech. J.* 64(2), pp. 391–408.
- [7] L. R. Rabiner (1989): *A Tutorial on Hidden Markov Models and Selected Applic.* IEEE. Speech Recognition.
- [8] Jérôme Renault (2006): *The Value of Markov Chain Games with Lack of Information on One Side*. *Math. Oper. Res.* 31(3), pp. 490–512. Available at <http://dx.doi.org/10.1287/moor.1060.0199>.
- [9] E. Waghabir (2009): *Applying HMM in Mixed Strategy Game*. Master's thesis, COPPE-Sistemas.
- [10] J. Van Der Wal (1981): *Stochastic dynamic programming*. M. Kaufmann. In *Mathematical Centre Tracts*, 139.