A Game Theoretic Approach to Maximize chances of Victory in Two Player Sports

Ambareesh Ravi
ambareesh.ravi@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

Anchit Nagwekar
ahnagwek@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

Atharva Gokhale
asgokhal@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

ABSTRACT

Game Theory concepts have been applied successfully in a wide variety of domains in the past decade. Sports and games are among the most popular areas of game theory application due to its merits and benefits in solving complex scenarios. With recent advancements in technology, the aid available to players before match, during game-play and after the match in the form of post match analysis for any kind of sport has improved to a great extent. In this paper, we propose three novel approaches towards the development of a tool which can assist the players by providing detailed analysis of decisions so that the player is well prepared with the most appropriate strategy which would produce optimum result for a given opponent's strategy. We also describe how the system changes when we consider real time game-play wherein the history of the opponent's strategies in the current rally is also taken into consideration while suggesting.

1 INTRODUCTION

There have been various advancements in the way in which technology is used in sports. The aid provided to players through various technological means has developed at a rapid pace over the past few years. Match analysis results in a high volume of statistical data which in turn is used by players for match preparation. In this paper we propose to use game theoretic approach to develop a tool which can potentially help players in any two player sports to thoroughly prepare a strategic plan to counter a specific opponent. To develop an approach to assist players in sports, we have chosen badminton. Our deep understanding of the game and common passion for the sport was the motivation for this choice. Badminton is a racket sport in which two players alternately hit a shuttle-cock until a point is scored. There are three models which we have proposed in this paper. All the models use match data as the input to provide suggestions and recommendations for a player. Our first model called the recommendation system takes into consideration the various shots that the player and the opponent have played throughout their career. The main purpose of this system is to help the players get an understanding of the different shots that the opponent plays and to gain knowledge of the best possible shots that he/she can play such that the chances of gaining a point is maximum. This model should be used by the players and coaches before going into a match and is not intended to be used during match play. Our second model which we like to call the Simulation model is an extension of the recommendation system but it takes into consideration the history of the match when it is in use. We use a reward system to determine the best shots for the players. This model is intended to provide match practice to the players against opponents so that they can simulate match situation and gain experience before heading into the actual match. Also, we intend to utilize some of the recent revisions in Badminton laws, one of which allows for coach intervention during the match. The proposed tool can prove to be very handy here as the coach can influence the player and guide him/her in the middle of the match through the use of this tool by quickly analysing the match up till that particular point in the game play.

1.1 Related Works

Game theoretic approach has been used to study various strategic sports in the past. Most of the work done so far has been towards studying specific parts of a sport and not the entire match or game. In soccer, the penalty kick has been modelled as a strategic game with imperfect information because of uncertainty about the kicker's type [1]. Bayesian equilibrium concept was used and it was found out that the kickers adopt a mixed strategy equilibrium depending based on their strong foot. In cricket, a normal form game was modelled between the batsman and the bowler [2]. The strategies of the batsmen depended upon the type of shot played and the bowler's strategies were the different types of deliveries that he can bowl. The utility values were derived based on the probability of the player to take a particular strategy. The study revealed that the probability distribution followed by the players in adopting different strategies in real world cricket is very close to the Nash equilibrium conditions.

Alpha Go, an artificial intelligence entity that can play the game of Go which was developed by Deep-Mind was able to defeat the best player in the world by 5 games to 0. This intelligent program uses a combination of Monte Carlo Simulation with value and policy networks [3] with the concept of Markov Decision Processes (MDP) [4][3][5] as its base. Inspired from this work, we have modelled our third approach that formulates the game of badminton for two robotic agents based on MDP. The goal of this model is to make the agents follow an optimal policy which would lead to a *long term benefit* as the match progresses for the agent under consideration. This system however, is limited to the level of modelling as far as the scope of this paper is concerned.

1.2 Motivation and Contribution

Each one of us has been interested in sports since our childhood. Our common passion for the game of Badminton has played a large part in our desire towards contributing to the sport. After witnessing several badminton matches over the years and having played the sport at various levels, we identified a very practical area in the game for the application of game theory concepts, **the digital assistance available to players**. The best of the players in the world learn at a very rapid pace as they progress in their professional career. However, there are sometimes a few areas where this natural learning process doesn't prove to be very effective. In such cases, computers come into the picture and convert this slow time consuming process into a rapid and results oriented one. In our literature survey, we came across many instances where game theory was used to solve problems pertaining to sports. One such case traces back to 2012 in the Olympics encounter between Yu Yang and Wang Xiaoli of China and South Korean pairs Jung Kyun-eun/Kim Ha-na and Ha Jung-eun/Kim Min-jung (doubles). The chinese pair tried to lose on purpose in this group stage encounter to avoid playing against their teammates Tian Qing and Zhao Yunle so that China is assured both the gold as well as the silver medals. [6] presents a detailed analysis on badminton match throwing using this example through game theory. The study reveals that the reason for this kind of match throwing basically lies in the loopholes of the format that the competition adopts and any rational player would adopt this strategy in the interest of the team. Besides this there have been various other similar cases which drove us towards using game theory for our problem. In [7], game theoretic approach is used to determine the optimal time during the match to play a risky serve and how the surprise factor plays a part, also studying how it affects the outcome for the player under consideration. Apart from that, it is found that it is beneficial for the player to play a risky serve during the critical points of the match rather than the less important ones. Highly motivated with the past work along these lines, through this project, we intend to contribute towards the game of badminton and develop a highly effective tool for player assistance with the aid of game theory concepts.

In summary our contributions are:

(1) We present a Recommendation tool which suggests the best shots for all possible shots of the opponent using the concept of *best strategy from game theory.*
(2) A simulator model, which is our novel approach where we consider the history up to two shots while determining the favorable shot to be played at a particular stage. Here, we model the approach as *an approximated finite non-zero sum extensive from game*
(3) An MDP approach, that comes into picture when we consider two intelligent (robotic) agents playing the game of Badminton. The approach determines a decision which ensures maximum long term return. This will essentially be a *perfect information zero sum game*

1.3 Data Collection

Data is essential to model the capability and choice of players which is vital in sports. Badminton is not like board games where predefined moves or strategies can consistently help you win. Humans tend to think differently when it comes to physical capabilities especially in sports. We can't expect a player to play the same shot with same accuracy every time; rather, we understand that a player's capability, stability and mentality changes during the course of a game. But to best model these factors, data plays a key role. Our data was manually collected by going through several full match videos of the players. We considered matches between two best badminton players in the world – Lin Dan from China and Lee Chong Wei from Malaysia so that the inefficiency of the players won't tamper the final results. Also, we had the variety of a left handed and a right handed player in our data. The matches we recorded are spanned over a period of 8 years (2011 - 2019) so that we cover the changing game plan and shot selection over a considerable period . The data was manually annotated shot by shot with their outcomes in terms of points and sets won. This format is essential to calculate the necessary parameters for the proposed models.

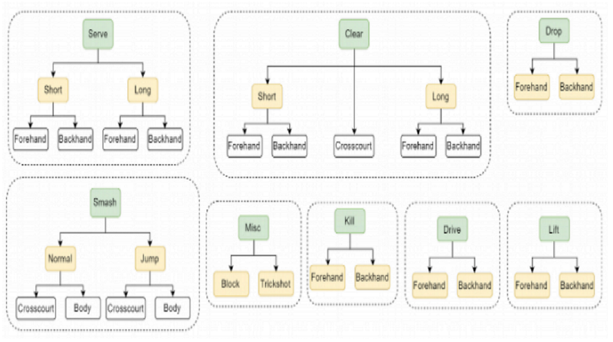


Figure 1: Badminton Shots under consideration

The types of shots which we have considered while collecting our data contribute to efficient results. Figure 1 presents the scope of the shots for our paper. We have tried to incorporate the position of the player on the court in terms of the type of shots. Also, there are certain exceptions to the types of shots that can be played against a particular shot of the opponent. None of the shots can be responded with a serve which is quite obvious. Also, a smash and a block cannot be returned by a smash and a block respectively. Drops are usually difficult to smash. We have ignored factors like agility, fatigue for the players as it is very difficult to model them.

2 METHODOLOGY

2.1 Recommendation Tool

The recommendation tool considers the player's choice and capabilities based on his history with a particular opponent to offer him suggestions to each shot of the same opponent. The concept of best response from game theory is adopted. This will help the player to be match ready with the best and safe shots to play given any shot from the opponent so that he can maximize his chances of winning each point thereby winning the match. We model the recommendation tool as a *normal form game*. The reason being that we are only worried about the player under consideration; meaning, we only care about his strategy and how to maximize his chances of winning the match and not the opponent. So it is enough to consider the game on a shot-by-shot basis rather than a sequential game. At any stage of the game, for a given shot of the opponent s_{-i}, player P_1 will have set of probable shots (strategies) S to play and the recommendation tool outputs those best shot s_i* from the available shots S which is the best response to s_{-i}. The game can be modelled as a tuple representing a normal form game:

$$G =< I, (s_i)_{i \in I}, (u_i)_{i \in I} > \text{ where } s_i, s_{-i} \in S$$

where $I = \{P_1, P_2\}$, $S = \{Set\ of\ Badminton\ shots\}$

$$u_i\left(s_i, s_{-i}\right) = P\left(s_i | s_{-i}\right) * P_{success}\left(s_i | s_{-i}\right) \tag{1}$$

where $P(s_i|s_{-i})$ is the probability of playing a shot s_i for a given shot of the opponent s_{-i}, $P_{success}(s_i|s_{-i})$ is the success rate of a shot s_i for a given shot of the opponent s_{-i}.These probabilities are calculated taking into account the data of the previous matches between the same two players. We consider the number of times a particular shot has been played by the player under consideration to calculate the probability of playing that shot including the instances where the shot yielded a point, resulted in loss of a point or continued the rally. Within these instances, we consider the number of times that particular shot has yielded a point for the player while calculating the probability of success for that shot. This is done specifically for every shot played by the opponent. For calculating the best response for a particular shot of the opponent, we find the shot s_i for the player which yields the maximum value of the utility according to the equation 1.

Algorithm for recommendation system:

(1) *Once an opponent has played a shot, the player has to make his next move.*
(2) *The recommendation system calculates the utilities for all the possible strategies (shots) of the player given the opponent's shot using Equation (2)*
(3) *It then picks the* ***strategy (shot) which yields the maximum utility*** *for the player and suggests the shot*
(4) *The player plays the shot and the opponent returns.*
(5) *The same process is then repeated from step 2*

This recommendation can be a return to a type of service or any other shot during the rally. It will help the player be prepared to face his opponent with confidence and clearly rule out few shots which have resulted in an immediate point loss in his history. Now, we can extend this tool further where we consider the opponent as our primary player and make all the computations to find his best possible shots for all the shots of player P_1 based on the same data set. This will return a set of most probable shots of a particular for each shot of the player P_1. Now, this will be helpful for predicting the return shot of the opponent for a particular shot s_i of player P_1. Accurate prediction of the type of serve or the type of return of the opponent for a particular shot s_i of player P_1 can prove to be very important in winning a point in crucial situations like dues or first point of the set. This approach is the base for our other two methods and prove to be a vital one in real world scenarios. We can observe many cases where a player rightly predicts a return of the opponent and surprises him with a trick shot and wins a point.

2.2 Simulator

The simulator is an extension to the recommendation tool but modelled as an *extensive form game* instead of a normal form game. The most important value addition in this method is that the history of shots between the players is taken into consideration. In badminton, it may not be always the case that the last shot results in winning or losing a point; the earlier shots played during the rally can also be responsible for an outcome. We observed from the collected data that this dependency on the history needs to be considered maximum for two earlier shots for best results. We have introduced reward system for incorporating the history. We consider four types of rewards as follows:

(1) A high positive reward (Rhp) when a shot of a player results in a direct point. For instance, a smash resulting in a direct point; Rhp = +5
(2) A medium positive reward (Rmp) when a shot of a player induces a poor return from the opponent and thereby yielding a point. For instance, a good lift to the back making the opponent make a poor clearance helping the player kill immediately and gain a point; Rmp = +2
(3) A medium negative reward (Rmn) for a poor shot of the player which the opponent takes advantage of making the player lose a point; Rmn = -2
(4) A low negative reward (Rln) when a shot results in a direct point loss; Rln = -5

The rewards are hence considered with a history up to two shots which will help the algorithm suggest the best possible outcome for the player. The total reward for a shot of player P_1 given an opponent's shot is stated as follows:

$$\begin{aligned} R_T(s_i, s_{-i}) = (R_{hp}(s_i, s_{-i})) + (R_{mp}(s_i, s_{-i}, s_i : t-1)) \\ +(R_{mn}(s_i, s_{-i}, s_i : t-1)) + (R_{ln}(s_i, s_{-i})) \end{aligned} \tag{2}$$

The purpose of simulator is to help the player by predicting the result of a rally up to pre-defined number of steps through the match. It will help him emulate the sequence of rallies in different ways for him to practice with another person before the match. Though the game of badminton is a perfect information zero sum game, the simulator is modelled as a *a perfect information infinite non-zero sum extensive form game* as the end utilities according to equation (2) won't be the same for both the players. It is not possible to solve an infinite game. Hence we make a few modifications to the above defined game into a finite extensive form game so as to be able solve it up to a predefined number of steps. We call it ***the approximated sequential representation*** of the original game. Here we restrict the number of sequences to a predefined number depending on the type of sport we are applying to. In badminton, it is enough for a player to think about his next two moves with one move of the opponent in between as he could rectify his mistakes within that else it would result in a point gain or loss within that. The scenario is illustrated in the figure 4

We introduce a reward system which is inspired from reinforcement learning [8] [9] that helps us calculate the favorable outcome for the player while taking into consideration the opponent's moves. The model of the game can be represented as follows:

$$G =< I, S, H, Z, \alpha, \beta_i, \rho, u_i >$$

where $I =$ P1, P2 is the set of agents,$S =$ Set of Badminton shots,H = choice nodes, Z = terminal nodes,α agent function, β action function, ρ successor function,

$$u_i(s_i, s_{-i}) = P(s_i|s_{-i}) * P_{success}(s_i|s_{-i}) * R_T(s_i, s_{-i}) \tag{3}$$

where u_i is the utility of agent i, $P(s_i|s_{-i})$ is the probability of playing a shot s_i for a given shot of the opponent s_{-i}, $P_{success}(s_i|s_{-i})$ is the success rate of a shot s_i for a given shot of the opponent s_{-i},

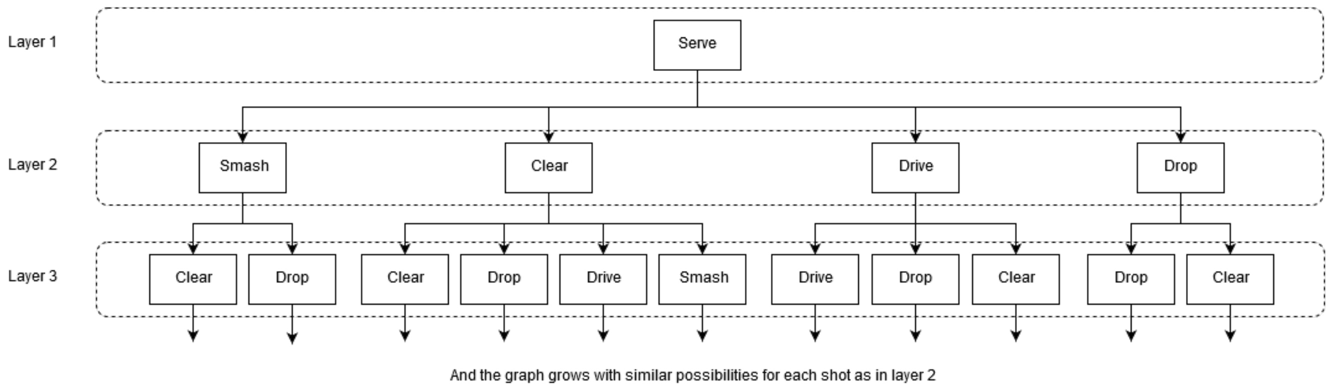


Figure 2: Extensive form representation of the game

$R_T(s_i, s_{-i})$ is the reward for playing s_i for s_{-i}. By approximating the infinite extensive game into a finite extensive form game, based on the rewards and utilities of players, we can predict the progress in the game after each shot.

Algorithm for simulator:

(1) *Once an opponent has played a shot, the player has to make his next move.*
(2) *The simulator expands a tree with the next 3 moves i.e. 2 for the player and 1 for the opponent, incorporating all possible combinations.*
(3) *It then calculates the accumulated utilities for each layer using Equation 3.*
(4) *It then uses backward induction to determine the favorable shots for a player and discard the rest.*
(5) *By solving for all three layers, the simulator then reduces the options to one shot which gives the maximum favorable outcome for the player.*
(6) *The simulator the tunes its values after this shot and has an option to incorporate real-time data from an ongoing match.*
(7) *The player plays a suggested shot and gets a return from the opponent and the process gets repeated from step 2 until the maximum number predictions is reached.*

The simulator then recommends the best favorable way the game could progress with the shots the player has to play along with the ones the opponent is expected to play. Though this model has the capability of producing very good results, it is often very hard to model the cognitive process of humans. In sports, humans tend to think differently, a move by a badminton player will involve a lot of factors like fatigue, ability, confidence, ability, condition in game, precision of opponents shot and even gut feeling

3 RESULTS

In this section, we discuss the results of our models. As there is no established metric to verify the results of our proposed models, an expert opinion or domain knowledge is the only way to check the correctness. As the model operates on the history of two players from the data, the results are only relevant to those players and will be different for others. Also, it is possible to create a generic model to focus on one player completely, given the data of the player with different opponents. The accuracy of the model depends on the size and consistency of data available.

Opponent's shot	suggestion 1	suggestion 2
backhand_short_serve	forehand_drop	backhand_lift
backhand_long_serve	forehand_drop	backhand_drop
forehand_drop	forehand_drop	backhand_lift
backhand_drop	backhand_drop	forehand_lift
forehand_kill	forehand_lift	backhand_lift
backhand_kill	forehand_long_clear	forehand_drop
jump_crosscourt_smash	block	forehand_lift
normal_smash	forehand_drop	block
jump_smash	block	backhand_lift
forehand_long_clear	forehand_drop	backhand_lift
backhand_short_clear	jump_crosscourt_smash	forehand_long_clear
backhand_long_clear	forehand_lift	normal_crosscourt_smash
crosscourt_clear	jump_smash	forehand_lift
forehand_drive	forehand_drive	backhand_lift
backhand_drive	forehand_drop	forehand_lift
forehand_lift	forehand_drop	jump_smash
backhand_lift	forehand_drop	normal_smash

Table 1: Recommendations for Player 1

3.1 Results of the Recommendation System

The recommendation system suggests the best response for an opponent's shot without considering the history of shots in the rally and the condition of the player in the game. The results, i.e. the recommendations of the model for player 1 are shown in Table 1 and that for player 2 are shown in Table 2.

It can be seen from the data that the model has successfully identified the best shots for given shots and that it has discarded the shots that are not playable for the opponent's shot successfully. The number of suggestions can be increased and we have fixed 2 best suggestions for giving the player an alternative. Also, from the suggestions above, it can be seen that the shot $forehanddrop$ has been repeated the most. This coincides with the fact that both the players are successful and capable in playing forehand drop shot without any error which was evident from the matches. Using

Opponent's shot	suggestion 1	suggestion 2
backhand_short_serve	forehand_drop	forehand_lift
backhand_long_serve	normal_smash	forehand_drive
forehand_drop	forehand_drop	backhand_lift
backhand_drop	forehand_drop	backhand_drop
forehand_kill	backhand_lift	forehand_drop
backhand_kill	backhand_lift	forehand_lift
normal_crosscourt_smash	backhand_lift	block
jump_crosscourt_smash	forehand_drop	forehand_lift
normal_smash	block	forehand_drop
jump_smash	backhand_lift	block
body_smash	block	forehand_lift'
forehand_short_clear	forehand_drop	backhand_drop
forehand_long_clear	normal_smash	forehand_long_clear
backhand_short_clear	normal_crosscourt_smash	forehand_drive
backhand_long_clear	jump_smash	backhand_drop
crosscourt_clear	forehand_drop	normal_smash
forehand_drive	forehand_drive	forehand_long_clear
backhand_drive	forehand_drive	backhand_drive
forehand_lift	forehand_drop	forehand_lift
backhand_lift	forehand_drop	forehand_drive

Table 2: Recommendations for Player 2

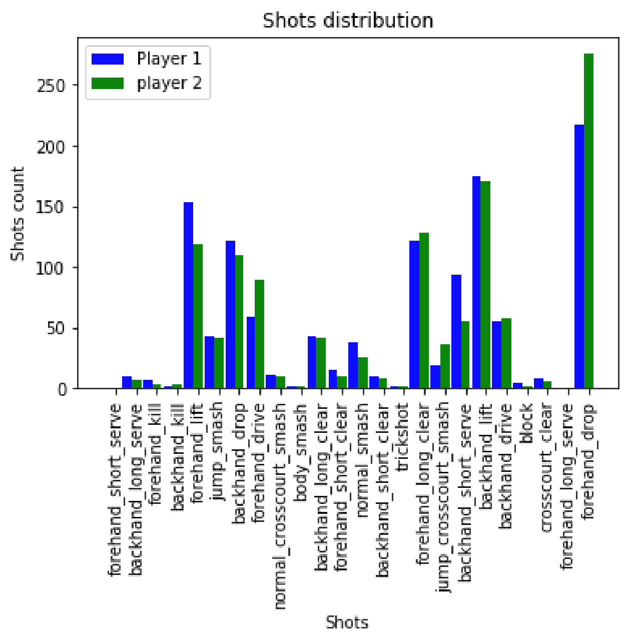
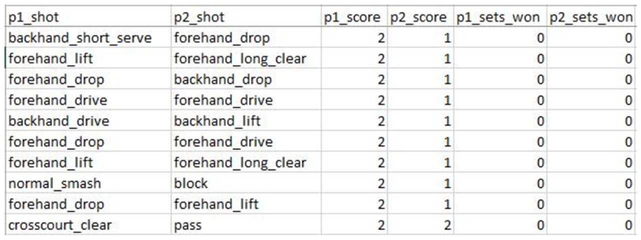


Figure 3: Frequency of shots for Player P_1 and player P_2

this model, the player can be mentally prepared on what shot to play to a given shot of the opponent that can either lead him to a point or keep him alive in the rally to avoid losing a point. As this is modelled directly from the capability and behavior of the players, this information will be of value to the players before the match.

3.2 Results of the simulator

The purpose of the simulator is to take the history of the ongoing match into consideration and to model the game strategy for the player based on his condition in the match, to identify the feasible sequence of shots that has lead him to points and to avoid the poor shots. The simulator can be fed with seed shots i.e. few inputs in the start on how the game should proceed and it predicts the next few shots. Since there is no information about a point gain or loss to the simulator, it will infinitely predict the sequence of the shots in the game. It is accurate when compared to following only the best strategy since in reality, the player should think 2-3 steps ahead in badminton to realize the after-effect of playing a shot as there is always the possibility of a poor shot leading to a point loss during the consecutive shots played. Hence, at any point of time, the simulator builds a tree for the next three shots (2 for the player and 1 for the opponent), does backward induction on the utility values according to equation (3), arrives at the most favorable shot for the player, discards the rest of the tree and the process is repeated. To check the working of the simulator, we seeded it with a few shots from a match's data which was not used for modelling. The actual sequence of shots from the match (table 6.3) and the sequence of predicted shots from the simulator is given below (table 6.4).



p1_shot	p2_shot	p1_score	p2_score	p1_sets_won	p2_sets_won
backhand_short_serve	forehand_drop	2	1	0	0
forehand_lift	forehand_long_clear	2	1	0	0
forehand_drop	backhand_drop	2	1	0	0
forehand_drive	forehand_drive	2	1	0	0
backhand_drive	backhand_lift	2	1	0	0
forehand_drop	forehand_drive	2	1	0	0
forehand_lift	forehand_long_clear	2	1	0	0
normal_smash	block	2	1	0	0
forehand_drop	forehand_lift	2	1	0	0
crosscourt_clear	pass	2	2	0	0

Figure 4: Data from an actual match between Lin Dan and Lee Chong Wei

Shot number	Player 1's (Lin Dan) shot	Player 2's (Lee Chong Wei) shot
1	backhand_short_serve	forehand_drop
2	forehand_lift	forehand_short_clear
3	forehand_drop	backhand_drop
4	forehand_drive	forehand_drop
5	forehand_drop	forehand_drive
6	backhand_drop	forehand_drop
7	forehand_drop	forehand_long_clear
8	jump_smash	forehand_drop
9	forehand_drop	backhand_short_clear
10	jump_crosscourt_smash	forehand_drop

Table 3: Output from the Simulator

It can be seen from the tables that the results of simulator are closer to the actual match. The results mostly differ only in the sub-category of shots and the actual type of shots are the same. This tool can help the players understand how the match will proceed after a shot is played which is crucial to analyze the repercussions of playing a shot. The simulator model is likely to work better when the data used for modelling is larger. Fig. 3 shows the distribution of shots for both the players in the data collected. There is an unequal distribution of shots which affects the accuracy of predictions of the models. We assume that data from at least 20 full matches is required to precisely model the behavior as in our case with 3 matches, the frequency of most of the shots is very low. Almost equal distribution of shots is required to ensure the reliability of the results from the model.

4 FUTURE WORK

In this section, we discuss the future works for our project.

4.1 An approach using Markov Decision Process

In the era of robots trying to play complex sports like table tennis, we tried to build a model where the robots can make decisions to opt for a particular shot to maximize their long term return based on MDP [3][5]. We consider this game as a *perfect information zero sum game*. We have to determine the optimal policy which states appropriate selection of shots dynamically while playing. The policy can be deterministic or stochastic and is specific for a particular state. We have considered deterministic policy as we have a deterministic environment where actions taken determine the outcome. Also, we will consider only one action per state for a particular agent. A finite MDP is defined by a tuple:

$$M =< X, S, P_t, R, \gamma >$$

where $X = \{X_1, X_2, W, L\}$ is finite state space as in Fig.6, S is finite action space as in Fig.1, P_t is transition probability, R is reward function and γ is discount factor. State X_1 is the one where agent A_1 plays a shot, state X_2 is the one where agent A_2 plays a shot, state W is where the rally ends with a point gain for an agent and state L is where rally ends with a point loss for an agent. S is the finite action space of all badminton shots in a particular state which we often refer as a strategy. Here, each action results into a transition of the agent from state x' to some other probable state, given by a probability distribution as shown in Fig. 6. P_t is the probability of the agent to transit from state x to x'. This probability depends upon the reward associated with each state. Each agent will try to transit from state x' to state W which ensures the maximum outcome. In cases where the agent is not in a position to transit to state W, he opts for state X_2 where the opponent gets a chance to play a shot and the rally continues. Every agent targets to force his opponent to choose that particular strategy which will result into his transition from state x' to state L to win a point and eventually win the match. The rewards [5] assigned to each state are such that we can associate the real time scenario of the game. Transition from state X_1 to state X_2 has no incentive as a long rally increases the fatigue of the agent eventually reducing his efficiency. But, it is always better to transit from state X_1 to state X_2 instead of landing to state L as the agent gets lowest return at state L. Lastly, discount factor γ signifies the value of time in the game. The agent should transit to state W as soon as possible for maximum returns.

If we consider the game as an extensive form game as considered in the simulator model, it will result in a higher number of states increasing the time and computational complexity. It is our target to determine the optimal policy dynamically in the course of the game. Thus, we modeled the game with limited number of states as in Fig. 5. We can see that each action in a particular state for each player can result into three possible outcomes, either go to state $P2$,

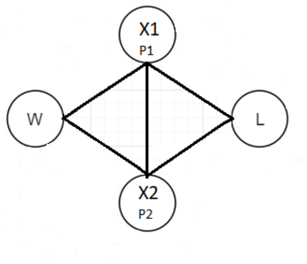


Figure 5: State Transition for any given action of an agent

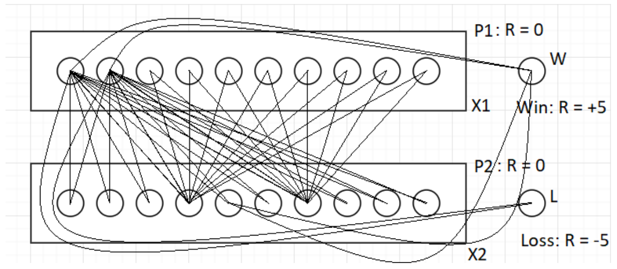


Figure 6: Finite State Space with corresponding Rewards

win a point W or loose a point L. We have to determine an optimum policy for each agent to get maximum long term returns.Thus, we use the action value and state value functions to determine whether or not to be in a particular state or opt an action given that the policy is followed leading to optimal results in the long term for the player. Here we know the state value function of W and L state are +5 and −5 which are the rewards associated with the state.

$State\ Value\ function$:
State value function $v_\pi(x)$ in MDP is expected return with x being the initial state and the policy π is followed by both agents.

$$V_\pi(x) = \sum_{s \in S} \sum_{x \in X} P_t \left[R_{x'} + \gamma * v\left(x'\right) \right] \tag{4}$$

$Action\ Value\ function$:
Action value function $q_\pi(x, s)$ in MDP is expected return with x being the initial state and the policy π is followed by both agents.

$$q_\pi(x, s) = \sum_{x \in X} P_t \left[R_{x'} + \gamma * v\left(x'\right) \right] \tag{5}$$

We calculate the state value function for each state using recursion and later solve the equations simultaneously with respect to equation 4. Once we have state value function for all states we can then determine the corresponding action value function as per equation 5 . Thus, if an agent takes a particular action when in a particular state he can determine his long term outcome given both the players follow the policy. Also, if the agent finds himself anywhere in between the rally, still the model can determine the best policy for maximum returns. Yet, there is a shortcoming in the model as we haven't considered the history for a particular outcome. In this case, an outcome is only subjective to the most recent shot and ignores the contribution of the earlier shots (a point gain or point loss). Due to the shortage of time and computational complexity involved this approach has to be taken as a separate research thread as a intersection between the fields of Game theory and Q-Learning in the near future.

4.2 Other works

We also plan to extend our first two models to incorporate dynamic data from ongoing matches for real time suggestions. One way is to recognize actions and outcomes from the match feed leveraging the power of computer vision and combine the results with our models to get a comprehensive solution. This dynamic behavior can help players correcting their mistakes and improve. The proposed models are capable of performing well with respect to other two-player sports like lawn tennis and table tennis readily.

5 CONCLUSION

In summary, we have successfully developed two novel approaches for the development of an assistance tool for the game of badminton based on the concepts of Game Theory. Our recommendation tool takes in match data for the player under consideration against a particular opponent and gives out the best possible set of strategies (shots) which the player can use. The simulator model is a generalized and robust extension of this recommendation tool which considers the history of shots played in the ongoing match along with match history to suggest the favorable strategy for the players. This is a complete and well-rounded system. We could validate our results with the actual shot selection process for the game of badminton. In the future, we propose to develop an algorithm based on Q-Learning for better results and advanced assistance to the players.

REFERENCES

[1] G. Coloma, “The penalty-kick game under incomplete information,” *SSRN Electronic Journal*, 05 2012.

[2] S. Ahmed, “Game theory in cricket,” *Honors Theses. Paper 918*, 2019.

[3] M. C. Silver D., Huang A., “Mastering the game of go with deep neural networks and tree search,” *Nature 529*, pp. 484–489, 2016.

[4] A. N. Burnetas and M. N. Katehakis, “Optimal adaptive policies for markov decision processes,” *Math. Oper. Res.*, vol. 22, pp. 222–255, Feb. 1997.

[5] M. van Otterlo and M. Wiering, *Reinforcement Learning and Markov Decision Processes*, pp. 3–42. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[6] Y. Hong-zhi and L. Wei, “Analysis of the badminton match-throwing in london olympics,” *Procedia Computer Science*, vol. 17, pp. 1222 – 1230, 2013. First International Conference on Information Technology and Quantitative Management.

[7] T. B. Anthony Bedford and M. Ladds, “Risk taking in badminton to optimize in-the-run performance,” *Proceedings of the 10th Australasian Conference on Mathematics and Computers in Sport*, pp. 20 – 26, 2010.

[8] C. Watkins, “Learning from delayed rewards,” 01 1989.

[9] L. P. Kaelbling, M. L. Littman, and A. P. Moore, “Reinforcement learning: A survey,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.