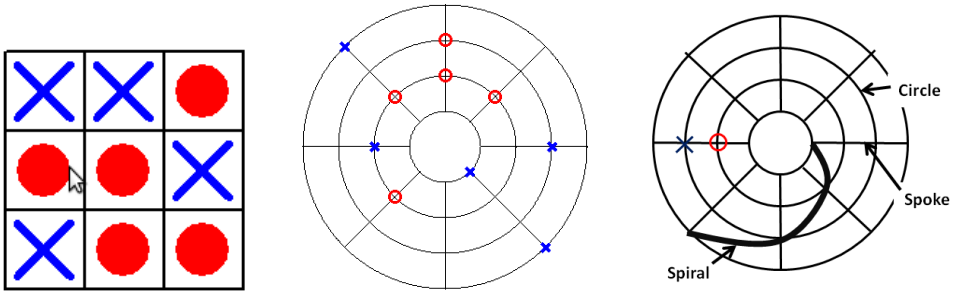
EECS 545 Project
Learning in non-cooperative games: Polar tic-tac-toe

Ashwin Kashyap, Ayoung Kim, Gaurav Pandey,
and Sai Sankalp Arrabolu

2009.12.17

1 The Problem



(a) Regular Tic Tac Toe (b) Polar Tic Tac Toe (c) Notations

Figure 1: Regular 3 x 3 Tic Tac Toe (TTT) and polar Tic Tac Toe (PTTT) board. Notations can be found in (c)

In this report, we explore reinforcement learning methods in solving non-cooperative games with perfect information. The game on the focus in a variant of Tic Tac Toe (Figure1(a)) which is a two player, sequential (players take turns to move), constant sum (rewards at terminal nodes have the same sum for every node) game with perfect information (there is no uncertainty in the state of the game).

The game under consideration in this report is called the polar tic-tac-toe (PTTT) as in Figure 1(b). PTTT is a variant of the 3-by-3 tic-tac-toe game (henceforth referred to as TTT), and which differs from tic-tac-toe in the shape of the board. The game ends when a connected chain of length 4 is formed by either of the players or when there are no further moves possible on the board, in which case it is declared a draw.

1.1 Notation

We define notation for the board configuration as in Figure 1(c). The board consists of four concentric *circles* with different radii. The first and second players are denoted by red (circle) and blue (cross) respectively. *Spokes* indicate the number of radial lines in a board and the *spiral* configuration is as indicated in Figure 1(c). We use a term *square* for all possible positions for a play.

1.2 Previous work

PTTT is relatively unexplored in the literature when compared to investigations on 3 by 3 regular TTT. For regular TTT, [2, 3, 6] show that reinforcement learning (RL) could be used as a relevant approach to find optimal strategies. The idea is to use rewards from terminal nodes and statistics on visited states to improve the quality of play as opposed to using known rational strategies. In fact, one of the goals of the RL agent would be to learn

such strategies through experience. In general, most board games can be approached through reinforcement learning because of similarity how children learns a board game [1]. A seriously limiting factor is the state space of the game. Agents playing games such as chess would not be effective if they use reinforcement learning as the state space becomes countably infinite after a sequence of moves. Further, nuances such as positional play cannot be learnt through artificial intelligence and this is the criticism that some of the best chess playing computers suffer from. A game such as PTTT presents an enormous simplification in its nature of play, moves and objectives. The state space is not too large and there is a possibility to work out locally optimal strategies, that we call base strategies by understanding the game.

2 Game Theory

We use concepts from game theory to analyze PTTT. Please refer to the appendix for an overview about terminology and definitions from game theory. PTTT is a dynamic game†[1] with perfect information†. Hence, the number of subgames† in PTTT is very large. We present a result which shows that there is an equilibrium† in which the first player can guarantee himself a win or a draw, regardless of how the second player plays.

Lemma 1 ***(The first player advantage)*** *Assume that both the players are rational. The first player has at least one set of strategies which can guarantee a forced win or a forced draw.*

Proof Zermelo's theorem [4] states that there is a backward induction equilibrium† which guarantees a forced win, a forced draw or an assured loss for the first player for dynamic games with perfect information with perfectly rational players. However, it does not shed any light on what such a strategy is. In other words, it is merely an existence result. Since PTTT is a game with reducing state space with the goal of connecting pieces, we could argue that the first player invariably has an advantage. We prove the above result by contradiction.

Let us suppose that there is a backward induction equilibrium which guarantees the second player a win. If this were true, the first player could transform himself to the position of a second player by making some irrelevant move which does not belong to the supposedly existing backward induction equilibrium. If this statement were not true, every single square the first player plays on should result in a loss for him through some backward induction equilibrium. Further, this should hold for all board dimensions as we did not require any specific board structure. Clearly, this is not possible. This completes the proof that no backward induction equilibrium can result in a guaranteed win for the second player. This, together with Zermelo's theorem proves our result. □

This proof technique can be used only with games which require the winner to be the first to establish a board pattern. An informal proof would hinge on "first player advantage".

Any finite game with perfect information can be theoretically solved using backward induction. As a consequence, PTTT can be solved completely by backward induction which

[1]Terms marked with † are defined in appendix

makes the game of PTTT seem trivial from the solution standpoint. However, it is not computationally feasible to directly use backward induction on this game because of two reasons: (i) a large state space which grows exponentially as the number of spokes increases and (ii) the existance of multiple Nash equilibria† due to the fact that terminal nodes have non-unique payoffs†.

3 Base strategies

Regular TTT has a set of prioritized rules which would guarantee a draw. PTTT differs significantly from TTT in the connectivity of squares along a circle. Squares on spokes and on spirals correspond to vertical and diagonal squares on a corresponding TTT board of the same dimension. However, playing along a circle has the unique property of allowing more winning states through extended connectivity. The caveat is that every square would belong to exactly four possible winning combinations across a circle, two local winning combinations along spirals and exactly one winning combination along spokes. This argument reinforces the fact that playing along a circle should be given higher priority when compared to playing across spokes or spirals. Another important observation is that a sequence of three continuous pieces of a kind on a circle would guarantee a win for the player if squares adjacent to the ends of this connected triple are empty (This creates a fork and hence cannot be countered). This is an example of a fork that can be created with the smallest number of connected pieces. Creating a fork along other orientations such as spirals or spokes would require at least six pieces.

We move on to formulate a set of base strategies for the agent. We derive base strategies from the concept of subgame perfectness, which has been defined in the appendix. The length of a path from node x to node y is defined as the number of edges between x and y in the game tree. Consider all subgames whose shortest path to any terminal node (where payoffs are realized) from the root node is of length one. A subgame perfect equilibrium† for all these subgames would correspond to the action of taking the win or a draw. A loss is possible only on an opponent's information set. Now, consider all subgames whose shortest path to a terminal node is of length two. All such subgames have the property of permitting a win for the opponent for every possible action of the agent, except one which blocks this win. Now, the equilibrium induced would be a block to prevent the opponent from winning. It is possible to show that subgame perfect equilibria for subgames with shortest path lengths of three and four would be to create a fork if possible or to block a fork from an opponent if such a threat is a possibility, respectively.

These strategies are optimal because any deviation from this equilibrium strategy will make the deviating player worse off. The list of base strategies are listed in their order of importance. Of course, all the strategies listed below are optimal only if such strategies are available to the respective player at any decision node. (For example, a win might not be possible at some decision node for player 1 and as a consequence, making a winning move is no longer an available strategy for him at that node).

1. Win

2. Block
3. Create a fork
4. Block a potential fork from an opponent
5. Connect along a circle
6. Connect along a spiral
7. Connect across a spoke

The last three strategies are derived in the order of their ability to provide winning chances for the agent.

4 Reinforcement learning as an analytical tool

Reinforcement learning has been used extensively to design agents to play games with large state spaces. It relies on the fact that a large number of instances of the game combined with supporting statistics regarding observed states would invariably lead us to the backward induction equilibrium when the agent is trained for a sufficiently large number of games. This leads us to the question of using a backward induction procedure (the so-called Zermelo's algorithm [4]) to determine an optimal decision rule. As mentioned earlier in the report, the state space and the possibility of exploring all possible edges makes an explicit computation of an equilibrium computationally hard. We also established that the backward induction equilibrium is not unique and this makes the task of computing every possible backward induction equilibrium based on the opponent's moves an insurmountable task.

Reinforcement learning through the use of statistics approximates the backward induction equilibrium and eliminates the need of using backward induction to solve PTTT. However, the downside is the memory required to record states and associated statistics for every observed state along with the computational burden of looking up a state when needed. The unique nature of a reducing state-space (at every stage, there will be a reducing number of possibilities to make a move) and radial symmetry make the problem of recording the statistics of the states and looking them up a tractable task. Since we have developed base strategies based on the nature of the game in the previous section, it is not necessary to store every observed state. We can fix an upper-bound on the number of states whose winning or drawing frequencies we expect to extract and thereby decrease the latency in processing a move. This scheme is used in most RL agents to balance between playing thorough experience (the essence of reinforcement learning) and making locally optimal moves based on the nature of the game. We point out that it is trivial to build a game tree following the realizations as it would involve picking the edge corresponding to an action *after* the players have made their moves. The state of the board at time t is actually a combination of all actions of both players until time $t-1$. We use the property of path independence for PTTT

(every path that leads to the same state is equivalent to another such path) to argue that it is not necessary to store the sequence of moves but just the current state of the board.

We now turn our attention to Selten's model [5] of extensive form games which allows for the possibility of slight mistakes or deviations from equilibrium strategies or perfect play. Learning algorithms try to find equilibrium strategies through repeated play against a training agent. It is possible to formally extend Selten's Nobel-prize winning concept of perfect equilibrium to this problem. The result that is shown below holds for all non-cooperative games.

Theorem 1 *A sequence of games $G_1, G_2, ...$ with a decreasing probability of error at decision nodes will possess the property that the equilibria of $G_1, G_2, ...$ will tend to the equilibrium point of the original game G asymptotically. Such a limit equilibrium point is known as a perfect equilibrium of the game G.*

Every training data point (a game in our problem) will equip the reinforcement learning agent with more information about *some* set of actions that lead to *some* physical outcome. If this process is repeated for many games, it leads to a sequence of games with the reinforcement learning agent reducing its possibility of mistakes through observations of actions and their respective outcomes, along with the use of some base strategies. As a result, the strategies at every decision node would converge to a perfect equilibrium of the game G in the limit. Such an equilibrium is robust to the possibility of slight deviations from perfect rationality.

We illustrate the above concept through an example from PTTT. Let us say that the RL agent missed making a winning move for a sequence of games, at the same state. As long as its actions were penalized through the opponent winning or drawing later in the game, it would avoid actions which led to the opponent's win or a draw. If the state under question is observed by the RL agent a large number of times, it would eventually make the winning move and will stick to it whenever the state is seen. Please note that the opponent's actions are important for this process to converge. If the training agent deviates strongly from rationality, then it is not possible to achieve a perfect equilibrium for the RL agent. To put this in context, it does not make sense to train the RL agent with a purely randomizing agent as its opponent and we are required to design an agent which follows at least a partial set of base strategies that we have outlined in the previous section. The idea is to train the RL agent to emulate the actions of the training agent to maximize the possibility of winning in the long-run.

We use the posterior probability of winning for a state as the metric to choose an optimal action, when we are not using base strategies. Every realized state in the game tree is updated based upon the payoff obtained at a terminal node. The posterior probability is a belief of a player that he would win the game if he lands on a particular state. Naturally, this belief changes with the number of training data points. Asymptotically, this probability measure would converge to steady-state equilibrium probabilities over the set of available actions. If one realizes *every* possible path in the game starting with the initial node (an empty board) and ending at all possible terminal nodes, and enumerates the number of wins acquired over

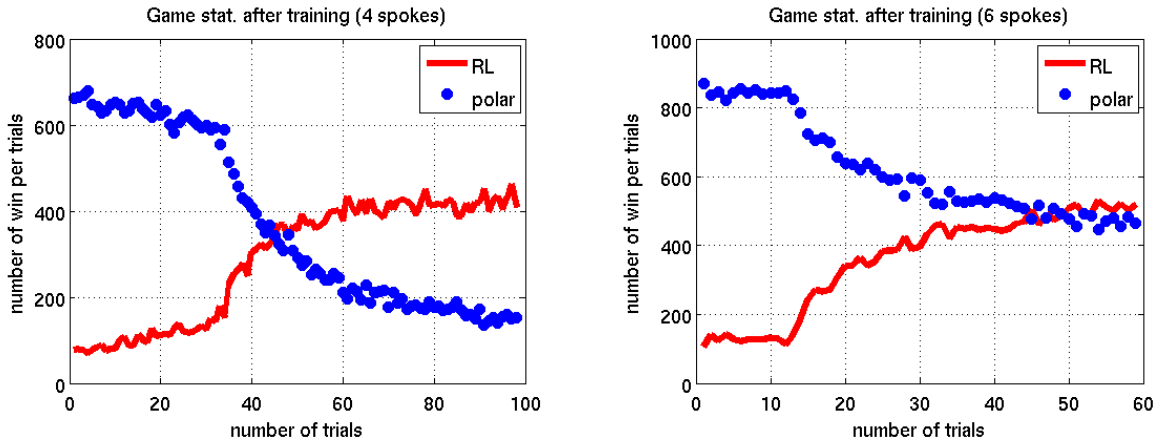
every path which contains a vertex x and the total number of paths which contain x, then the true posterior probability of winning for x is the ratio of these quantities. The accuracy with which we calculate the posterior probability in our game depends on the realization of paths. We will not derive the evolution of this posterior probability or prove rigorously that it is optimal to choose the action which maximizes the posterior probability of winning in this report. We simply use the arguments about convergence to equilibria to argue that it is indeed optimal to play a strategy which maximizes the posterior probability of winning.

5 Experiments & Results

We designed an experiment to capture the learning curves of the RL agent under different conditions. Every trial consisted of $1,000$ games played between the RL agent and the training agent. 100 such trials were carried out, which resulted in a total of $100,000$ games being played. The results we obtained provided conclusive evidence that the RL agent was improving its ratio of wins to losses every trial.

In the following section, we show training results from four, six and eight spoke PTTT. For the purpose of training, we use an agent which plays two of the prioritized base strategies (win and block).

5.1 Training and Validating



(a) Learning curve of 4-spoke PTTT.

(b) Learning curve of 6-spoke PTTT.

Figure 2: Training and validating 4-spoke and 6-spoke PTTT (Each trial consists of $1,000$ games)

Figure 2 shows the learning curve of the RL agent for 4 and 6 spokes respectively. After a significant number of games, the RL agent performed significantly better than the training agent, which was programmed to play a fixed set of base strategies. Every sequence of games was characterized by a decrease in the probability of error for the RL agent and

hence, we could expect an equilibrium point to be reached through this training sequence. This provides empirical proof that the RL agent has the feature of developing all the features of the training agent, and also has the ability to perform better through the exploration of strategies which fall outside the set of strategies that a training agent employs.

5.2 Learning Rate

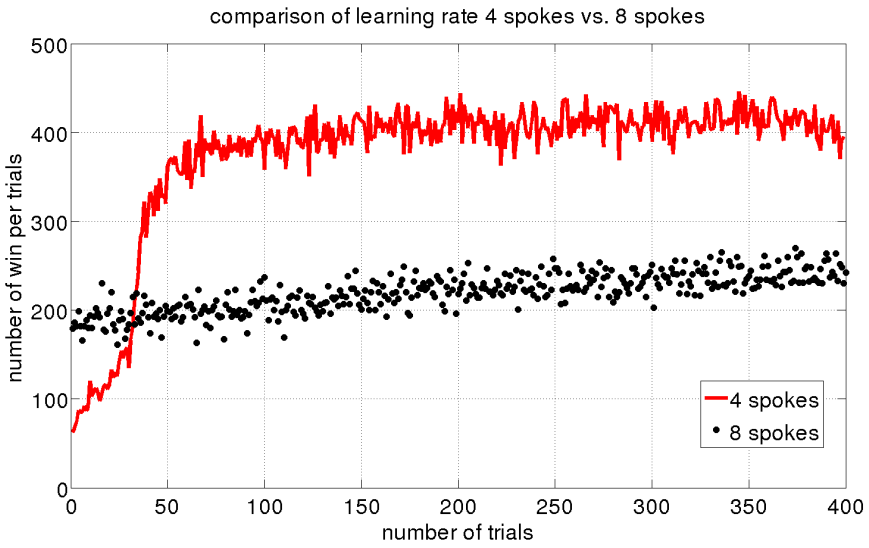


Figure 3: Comparison of learning rates between 4 spoke PTTT and 8 spoke PTTT

As the number of spokes increases, the learning rate (i.e., the time to reach the steady state, defined as the time when the fraction of winning or drawing games does not change significantly) increases. Figure 5.2 shows the comparison between 4-spoke PTTT and 8-spoke PTTT in terms of learning rate. One could attribute this to an exponentially increasing state space as the number of spokes increase. In addition, it would take significantly more time to gather statistical training data that is meaningful for states of the game. The average time to revisit a state through a different instance of the game also increases with the number of spokes. Using this information, we can conclude that it is better to use locally optimal base strategies over choosing actions from experience in a game with a very large state space.

6 Challenges

The biggest challenge from the theory perspective was to learn a large body of material on dynamic games and equilibrium analysis and connect some of the key results in game theory to validate the asymptotic behaviour of the training process. Equilibrium analysis also helped in figuring out the right set of base strategies for training the agent.

The biggest challenge from the perspective of implementation was to generate and maintain the statistics for all the training games. This posed a problem of dealing with large data

sets and conventional methods such as linear search would introduce significant latency in the program. In order to overcome this, we used a Hash Map to index states and update the states contained in the game tree. The key property of a Hash Map is $\mathcal{O}(1)$ look-up time. This decreased the processing time necessary to train the RL agent. Another challenge was the length of time involved in training the RL agent. As the number of spokes increased, the training rate was slower and we required eight hours to obtain evidence that the RL agent was improving steadily for a game with eight spokes. Most of the empirical evidence presented in this report required several rounds of fine-tuning. Reproducing results took a significant amount of time and computational power.

References

[1] Kevin Crowley and Robert S. Siegler. Flexible strategy use in young children's tic-tac-toe. *Cognitive Science*, 17(4):531–561, 1993.

[2] Johannes Fürnkranz and Miroslav Kubat, editors. *Machines that learn to play games*. Nova Science Publishers, Inc., Commack, NY, USA, 2001.

[3] Imran Ghory. Reinforcement learning in board games. Technical Report CSTR-04-004, Department of Computer Science, University of Bristol, May 2004.

[4] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, June 1995.

[5] R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4(1):25–55, March 1975.

[6] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Appendix: Definitions from Game Theory

- **Payoff function** A payoff function u_i for player i is defined as a mapping from the space of actions of all players to a real number. For a two player game such as PTTT, a payoff function for either player maps the Cartesian product of action spaces of both players to a real number.

- **Pure strategies and randomized strategies** A mixed strategy profile (or a randomized strategy profile) is defined as a probability distribution over the set of all possible actions for a player i. A pure strategy profile can be defined as a degenerate case of a mixed strategy profile where a particular action is played with probability 1.

- **Nash equilibria** A strategy profile (s_1, s_2) constitutes a two player pure strategy Nash equilibrium if,

$$u(s_1, y) \geq u(s_1', y)$$
$$u(x, s_2) \geq u(x, s_2')$$

$\forall s_1' \in S_1, y \in S_2$ and $\forall s_2' \in S_2, x \in S_1$

Intuitively, a Nash equilibrium is a mutual best response set. A mixed strategy Nash equilibrium can be defined on similar lines over a mixed strategy profile.

- **Dynamic games** A dynamic game for two players consists of a set of alternating moves between players. In general, the game consists of terminal and non-terminal nodes and payoffs are assigned to only terminal nodes. We use the term decision node for all non-terminal nodes.

- **Perfect information** A dynamic game with perfect information is one in which there is no uncertainty regarding the state of the game for all players at all points in time. Chess, PTTT and checkers are some examples. Perfect information should not be confused with complete information, which generally refers to known payoffs, action spaces, game rules and so on.

- **Information set** Information set for player i is defined as all the possible states that player i could be in at one point in time t. For dynamic games with perfect information, the information set at time t consists of exactly one state/decision node as it is clear to all players what the state of the game is.

- **Subgames** A subgame is defined as any part of a dynamic game that begins with one decision node x and contains all nodes and edges that come after x. Here, after refers to all the states which are connected to x and which cannot occur prior to x in the game. In PTTT, there are an exponential number of subgames. We do not need to worry about information sets with cardinality greater than one as PTTT is a game with perfect information.

- **Perfect equilibria** Selten defines the concept of perfect equilibrium as an equilibrium which is robust to a (small) deviation from Nash strategies by players and is a result of convergence of equilibrium points of all perturbed games (games with possibilities of mistakes) with a decreasing probability of deviations from perfect rationality. It quantifies the stability of Nash equilibria in dynamic games.

Contributions

Ashwin worked on the theoretical aspects of the project, implementing the training agent with base strategies, drafting the proposal and the final report. Ayoung worked on the implementation of the training agent, the progress report, the final report and the lovely figures that appear in all reports. Gaurav worked on the reinforcement learning agent and complex implementation aspects related to large data sets in addition to working on the final report. Sai Sankalp worked on the implementation of the reinforcement learning agent and the blocking agent for the progress report. All the team members met regularly to discuss about the project and the effort was nearly equal for all team members.