Rock Paper Scissors

Playing the Game with Markov Chains

Neil Pomerleau
Worcester Polytechnic Institute
Worcester, MA 01609
npomerleau@wpi.edu

***Abstract*—Though rock-paper-scissors is usually a game of pure chance in which moves are selected at random, a player can predict the opponent's next move based on previous rounds of play. An online version of the game played over 13,000 rounds against human competitors and used this data to calculate transition probabilities in a 10th-order Markov chain. The virtual player was able to win 3.05% more of the rounds that did not end in a draw than it would have won using the naïve method of randomly selecting moves.**

I. Introduction

The rules of rock-paper-scissors are simple: in each round, two opponents simultaneously choose rock, paper, or scissors. Paper beats rock, rock beats scissors, and scissors beats paper. If the moves are chosen completely at random, neither player should have an advantage and the probabilities of either player winning or the round resulting in a tie are all equally likely.

However, a player could use previous rounds of play to gain insight into the opponent's strategy and attempt to predict their next move, giving the player an advantage when selecting his or her own move. In fact, after many rounds of play, a player could learn transition probabilities from the opponent's previous moves to their choice for the next move.

Learning the subtle correlations between current state and next state in what are essentially random choices is no easy task. This strategy of examining transition probabilities would not work against a player who makes truly random choices, since each possible next state would be equally likely and all transition probabilities would be the same. To say that current state tells us anything about what to expect for a player's next move is to say that random human choices are probabilistic and predictable to some extent.

Because these correlations are so subtle, the only way to learn the transition probabilities and test the predictability of these random choices is to accumulate substantial training data through many rounds of play against real human players. This data was collected by establishing a website that plays rock-paper-scissors against people looking to test their skills against a virtual player. Over the course of two weeks, the virtual player had observed and stored over 13,000 rounds of play.

The virtual player then treats this entire database as a 10th-order Markov chain, in which the current state is defined as up to the previous ten rounds of play. The transition probability is determined from the moves selected in previous rounds matching the current state. The extent to which a previous round's state matches the current state increases the weight given to that move in the calculation of the transition probability. A previous round that matches all ten prior rounds of the current state will be weighted much more strongly than a previous round that matches just two or three prior rounds of the current state.

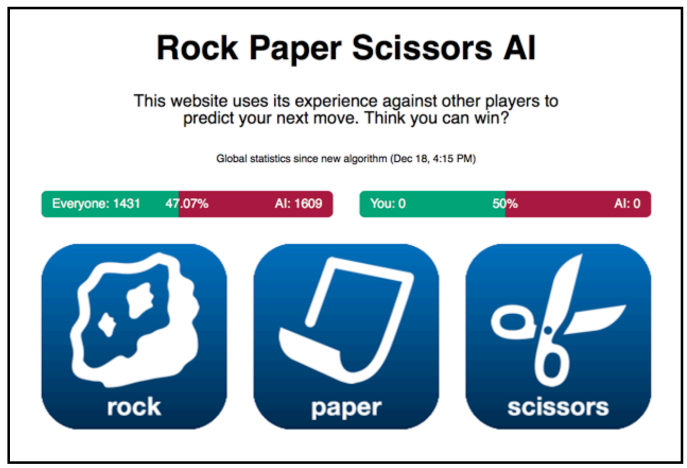

Fig. 1. Screenshot of the website interface

Once the virtual player collected enough training data to calculate transition probabilities on demand, it was able play rounds against human players in which it attempted to predict their next move. In 2,933 rounds of play that did not end in a draw, the virtual player was able to win 53.05% of the time.

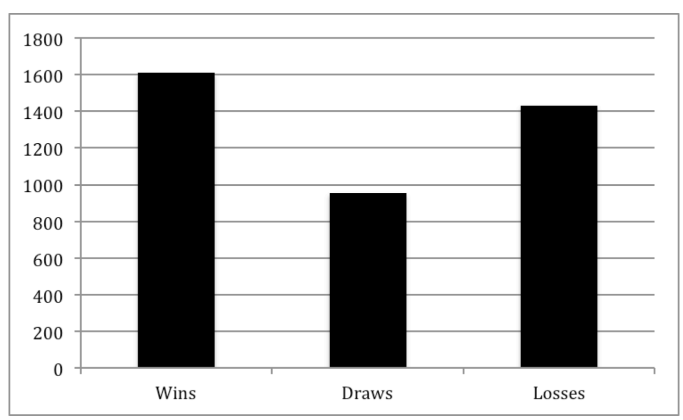

Fig. 2. Final results for the AI (after algorithm adjustment)

If the website had been using the naïve method of randomly selecting moves, it would have won approximately 50% of the time. This mean that the virtual player, using the transition probabilities learned from the first 13,000 rounds of play, was able to gain a 3.05% advantage over pure chance.

The next section describes Markov chains in general and their applications, as well as other attempts to predict moves in rock-paper-scissors using artificial intelligence. After the problem is formally defined, the methodology of this project is described to show how the work in this project builds upon prior developments. Finally, the complete results of this project are provided.

II. Related Work

The basis of the theory for this project is the Markov assumption, which claims that the current state only depends on a finite number of previous states. Processes that satisfy the Markov assumption are called Markov processes and can be described as a Markov chain [1]. In the case of this project, the game of rock-paper-scissors is considered to be a Markov process, and the Markov chain is built over many rounds of play.

The simplest form of a Markov process is a first-order Markov process, which claims that the current state only depends on the previous state and no more. Though it is possible for the first-order assumption to be exactly true, it is often used as an approximation [1]. For example, the probability of rain today (the current state) can be approximated based on whether it rained yesterday (the previous state).

To improve upon the accuracy of the approximation from a first-order Markov process, one can either increase the order of the Markov process to consider more previous states or increase the set of state variables [1]. For the purposes of this project, the former is more practical.

This project is certainly not the first attempt to play rock-paper-scissors using a virtual player. In fact, rpscontest.com serves as an ongoing programming competition to see who can develop the most effective algorithm for playing rock-paper-scissors against human players. The website's homepage cites the game as "fundamental to the fields of machine learning, artificial intelligence, and data compression" and even claims that it might be "essential to understanding how human intelligence works" [2]. The top-ranked algorithms have winning percentages over 70-80%.

Two specific prior works have attempted to solve the problem via data collection similar to that in this project. First, the New York Times published an interactive online game that has over 200,000 rounds of experience and an undisclosed winning percentage [3]. After five rounds, it matches this exact round history against rounds with other competitors to guess your next move. In essence, it treats the game as a fifth-order Markov process.

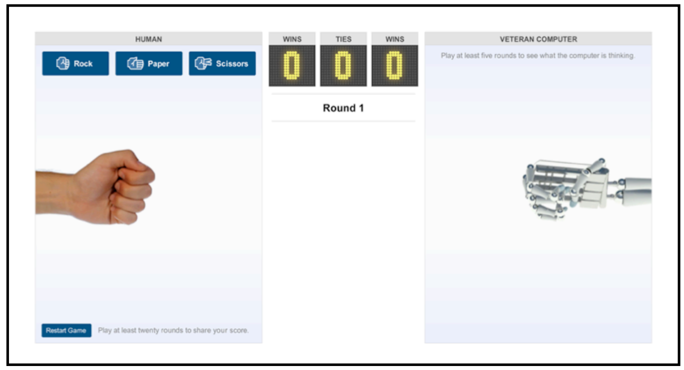

Fig. 3. Screenshot of the New York Times implementation [3]

Second, essentially.net hosts an online game that has collected 497,933 rounds to date and has a winning percentage of 59.82% [4], well above pure chance.

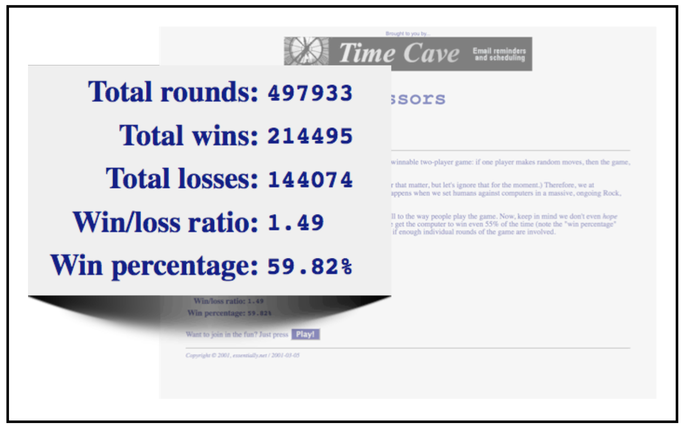

Fig. 4. Screenshot of statistics from the essentially.net implementation [4]

What the two prior works have in common is that they only take into account exact matches when examining historical data, and they both treat the game as a fifth-order Markov process only. Though this project works from fewer rounds of data, it improves upon the prior algorithms by allowing partial matches in historical data and weighting this information accordingly.

III. Problem Statement

It is difficult to predict an opponent's next move in a game of rock-paper-scissors because moves are typically chosen at random. This project aims to determine transition probabilities from previous rounds of play to an opponent's next move, exposing the predictability of human behavior in game play.

For the purposes of this paper, r and R represent a move of rock for the human and AI players, respectively. Similarly, p and P represent paper, and s and S represent scissors. A round is represented as a pair of moves, e.g. (p, S). A game is represented as a sequence of rounds, e.g. [(r, P), (p, S), (s, R)].

IV. METHODOLOGY

An online game of rock-paper-scissors was created, allowing visitors to play the game against a virtual player. The AI uses its experience against previous players to attempt to predict the current opponent's next move.

Fig. 5. Screenshot of the website interface

Each time a human player makes a move, the website checks their move history and looks for moves that other players have made after similar sequences of rounds. Since the website treats the game as a 10th-order Markov process, it considers up to the previous ten rounds of play. The moves for matching sequences are weighed based on the extent that the round history matches the current player's round history, resulting in the transition probability for the Markov chain. This player's move is then added to the gameplay database for use in future rounds and games.

For each move the human player makes, the database stores the sequence of up to ten previous rounds as the key to an entry. The value of the entry is the round with the moves made by the player and the AI.

TABLE I. EXAMPLE DATABASE ENTRIES (CONSECUTIVE ROUNDS)

History	*Move*
[(r, R), (r, S), (p, P), (r, P), (p, P), (r, R), (s, R)]	(r, R)
[(r, R), (r, R), (r, S), (p, P), (r, P), (p, P), (r, R), (s, R)]	(r, R)
[(r, R), (r, R), (r, R), (r, S), (p, P), (r, P), (p, P), (r, R), (s, R)]	(r, P)

When the AI needs to decide what move to play against the user, it searches the database for matches of different history lengths. Simulated rounds revealed that the AI's move should be included in the history for improved accuracy of the probability calculation, and the weighting function of 3^*length* was tuned via simulated rounds. These simulated rounds were run by taking individual entries in the database and choosing moves with variations of the algorithm examining the remaining entries in the database. The results of these simulated rounds are described in the results, and the core algorithm is summarized next.

Algorithm 1: ProbableNextMove(*allHistory*, *userHistory*)

inputs: *allHistory* maps previous histories to previous moves
userHistory the opponent's history
returns: the opponent's most probable next move
local: *weights* array that stores weighted likelihood of next move, initially 0 for all moves
length the extent to which a previous round matches this one
begin
for each (*history*, *move*) ∈ *allHistory* **do**
length = 0
while (*history*[*length*] == *userHistory*[*length*]) **do**
length ← *length* + 1
weights[*move*] ← *weights*[*move*] + 3 ^ *length*
return index of max(weights)
end

This algorithm expands upon prior work with rock-paper-scissors games that learn from rounds against real human players. Existing implementations only consider up to five previous rounds, and they ignore rounds that don't perfectly match all five previous rounds [3][4]. This rejection of imperfect histories works when the collection of observed rounds is large enough to cover the possible states (~200,000 - 500,000), but too many entries would be rejected in a relatively small database (~10,000). This is especially true when the number of previous states considered doubles to ten. In the same way that likelihood weighting improves upon rejection sampling when the number of particles and the likelihood of any particular state are both relatively small [5], the algorithm used in this project improves upon the algorithms used in existing rock-paper-scissors implementations when the number of observed rounds and the probability of any particular sequence of rounds are both smaller.

V. RESULTS

Prior works have established that experience against human players can give an AI an advantage in future rounds of play. After almost 500,000 rounds of play, an implementation using a 5th-order Markov chain and no weighting for was able to win almost 60% of rounds against humans that did not end in a draw. With this project, the question of whether a similar advantage can be achieved after fewer rounds of experience is addressed.

The initial 13,000 rounds of play used an experimental weighting of 2^*length* in the ProbableNextMove algorithm and ignored the AI's own moves. Human players quickly learned that by using the AI's move in the immediate next round gave players a slight advantage, winning 50.77% of rounds that did not end in a draw.

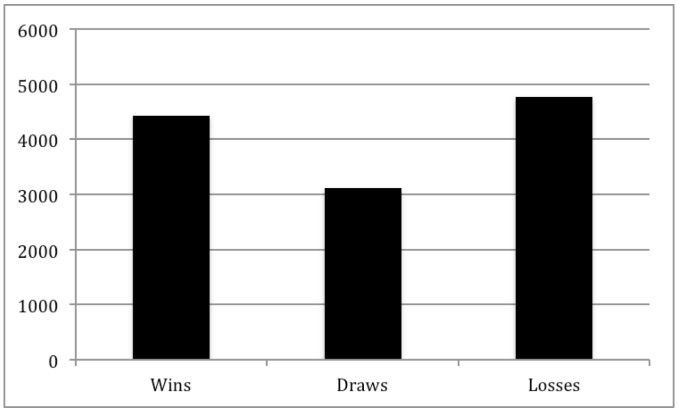

Fig. 6. Real results for the AI after the first 13,000 rounds

The next iteration of the algorithm stored pairs of moves (rounds) rather than just the player's own sequence of moves, taking the adversarial nature of the game into account. The weighting was also adjusted to 3^*length*. Simulated rounds showed that these changes would have resulted in a higher win percentage for the AI.

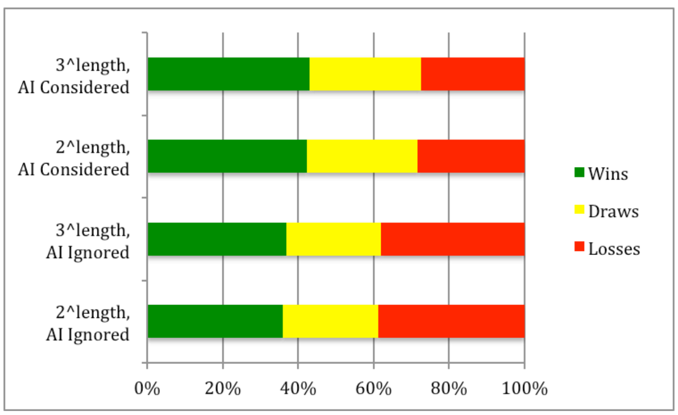

Fig. 7. Simulated results for the AI, testing different algorithm variations

Simulated rounds showed that the AI would win approximately 60% of rounds that did not end in a draw when weighting was adjusted to 3^*length* and the AI's moves were considered as a part of the round history. The revised algorithm was implemented and additional rounds were played against real human players to see if the simulated results translated into a real advantage. In the 3,993 rounds of play since the algorithm was adjusted, the virtual player won 53.05% pof rounds that did not end in a draw, giving it a 3.05% advantage over pure chance.

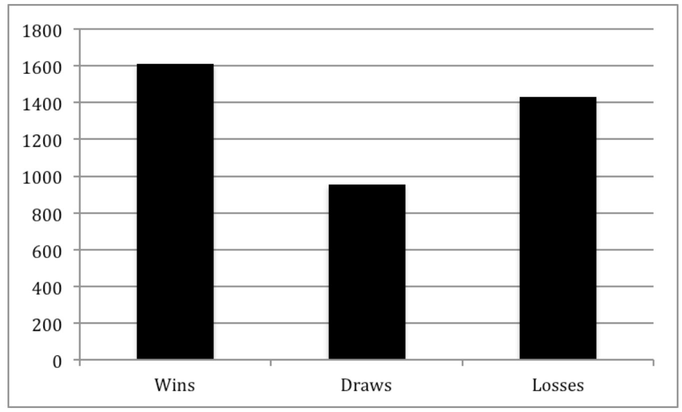

Fig. 8. Real results for the AI after algorithm adjustment

An unexpected result was that the AI avoided draws with the user much more than it would with random play, so most rounds had a decisive winner. Though the AI fared well considering the relatively small round observation pool compared to prior works, it would probably benefit from additional rounds of play against more human players.

VI. Conclusion

After adjusting the algorithm to optimize weighting and account for the AI's moves in the round history, the virtual player was able to achieve a winning percentage of 53.05% in rounds that did not end in a draw. Since this percentage persisted afted thousands of rounds against human players, it is likely statistically significant. Other AI implementations have achieved better results after more rounds of play against humans, but it is possible that this implementation would also benefit from additional play. This project builds on prior works that have shown that human behavior in a random game is not entirely random, and it is predictable to the extent that an AI can begin to predict non-deterministic decisions through experience. Through the game of rock-paper-scissors, this AI implementation has gained some insight into human thought processes and decision making – a field that will surely be explored with future AI developments in the years to come.

Acknowledgment

Professor Dmitry Berenson and his Fall 2013 Artificial Intelligence class at WPI (CS 534) were invaluable for their guidance when deciding how to adjust the algorithm.

References

[1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River: Prentice Hall, 1995, pp. 568-570.

[2] B. Knoll. (2011, May). *Rock Paper Scissors Programming Competition* [Online]. Available: http://www.rpscontest.com/

[3] G. Dance and T. Jackson. (2010, Oct 7). *Rock-Paper-Scissors: You vs. the Computer* [Online]. Available: http://www.nytimes.com/interactive/science/rock-paper-scissors.html

[4] S. Bayern. (2001, Mar 5). *Rock, Paper, Scissors: Humans against AI* [Online]. Available: http://www.essentially.net/rsp/index.jsp

[5] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River: Prentice Hall, 1995, pp. 532-533.