Artificial intelligence has begun to uprise as a powerful tool used throughout all sorts of areas. The most popular use of AI are on websites such as ChatGPT that allow users to command the AI to answer questions or do tasks such as writing. ChatGPT runs off constructed data retrieved from the internet. In this research lab we attempted to apply AI to a simple game of rock-paper-scissors, except our AI would be getting all its data off the active human player's moves. Through the use of various algorithms the AI will organize the data in order to achieve a higher rate of AI wins against a human player over a sequence of game rounds.

**Research Impact**

The objective of our research was to learn more about "online learning." Online learning is a modeled framework of interactions between human and bot users. The impact set by our research is the startup of an outline for a number of real-world applications such as online advertising, gaming, market design, etc.

---

**Methods & Results**

**Follow-the-Leader (FTL)**

The algorithm calculates scores for each possible move (rock, paper, scissors) as if it had been played in all previous rounds. It then chooses the move with the highest score for the current round.

**Score Updating**

After each round, updates scores based on the user's move:

- If user plays Rock: Paper score +1, Scissors score -1
- If user plays Paper: Scissors score +1, Rock score -1
- If user plays Scissors: Rock score +1, Paper score -1

**HEDGE**

The algorithm is an improvement over the Follow-the-Leader (FTL) algorithm. It allows for a more nuanced and adaptive strategy compared to FTL, potentially leading to better performance against a wider range of opponents.

**AdaNormalHedge**

The algorithm is very similar to HEDGE, but it uses a more sophisticated approach than basic HEDGE. AdaNormalHedge also learns from the previous rounds, but an advantage is that it is parameter-free, so it doesn't require any parameter tuning as in Hedge. AdaNormalHedge calculates the best possible move by using the equation:

\[ w(R,C) = \frac{1}{2} (\Phi(R + 1, C + 1) - \Phi(R - 1, C + 1)) \]

**Tree-Expert**

In this algorithm, we maintain a tree where each node acts as an expert and makes its predictions using AdaNormalHedge. We maintain a context string that encodes the current moves of the user. Whenever the user plays a move the following predictions and updates are made: (a) the move for the computer is decided by combining the predictions of the nodes (traversed along the current context string); (b) each node in the tree updates it's predictions by AdaNormalHedge based on the current user move; (c) the move is appended to the beginning of the context string and a new node at a new level is appended to the tree.

**Game Flow**

**First round:** The algorithm plays a random move.

**Subsequent rounds:** It chooses based on past performance.

In our quest to develop an AI capable of consistently outperforming human players, **Tree-Expert** resulted as the most effective algorithm among those we tested. While more sophisticated games typically require more intricate algorithmic approaches, even seemingly simple games like rock-paper-scissors can benefit from advanced online learning techniques. The key to designing a successful algorithm for such games lies in recognizing how to apply and potentially modify fundamental online learning concepts, such as Follow-the-Leader (FTL) or Hedge algorithms.

---

**Citations**


All images PC: Samantha Jauregui

---

**Acknowledgements**

I want to give a big thank you to my SHINE lab mentors Spandan Senapati and Soumita Hait for guiding me through the tricky path of coding with algorithms. I am also very grateful for our SHINE sponsors and mentors for coordinating and allowing me to attend a career impactful program without the worry of tuition so I can keep focus on my ambitions. Thank you all!