Predicting Human Action Using An Online Learning Algorithm

Justin Jang | justin25679@gmail.com
Haipeng Luo, Mengxiao Zhang, Chung-Wei Lee
West Ranch High School, Class of 2020
USC Viterbi Department of Computer Science, SHINE 2019

USC Viterbi School of Engineering

SHINE Summer High School Intensive in Next-Generation Engineering

Introduction

Online learning is a method of data analysis that takes in human sequential input and updates itself to make appropriate decisions in response to those inputs.

- Humans are psychologically geared to make specific patterns in their actions, especially in game applications.
- Online learning algorithms can be utilized to tap into this data and then further improve its adaptability and functionality.

Objective & Impact of Professor's Research

Professor Luo's research is trying to develop an **online learning algorithm** that will most accurately and efficiently predict human actions based off of the human action history. He is doing so in game applications, specifically in **Rock Paper Scissors**.

- This is done through **experts** $(e_1, e_2, \dots e_n)$, which are beings that give advice on what action the computer should play next against the human.
- Expert e_i has a set **probability** $p_{t,i}$ at time t that dictates how likely the computer will select that expert's prediction.
- Expert e_i also experiences a **loss** $\ell_{t,i}$ at time t, which is determined based off whether or not the expert correctly predicts the human action. Loss is gained if the expert predicts incorrectly.
- The goal is for the computer to suffer the least amount of accumulated loss starting from time 1 to time T.

Skills Learned

Fixed Expert Hedge Algorithm

This algorithm has a fixed number of experts through which the accumulated loss of each expert is maintained in order to create a probability distribution among the experts at time t. The computer then has a higher probability of selecting the prediction of the expert that maintains a smaller accumulated loss.

$$L_{t,i} = \sum_{i=1}^{N} \ell_{t,i}$$

$$P_{t,i} = \frac{e^{-\eta L_{t,i}}}{\sum_{j=1}^{N} e^{-\eta L_{t,j}}}$$

L is accumulated loss
$\eta > 0$ is a hyperparameter

Fixed Tree Expert Algorithm

There are a fixed number of experts with a probability distribution, and each expert is a tree. The probability distribution is updated similar to Hedge, but the decision that each expert makes is dependent on previous human action history.

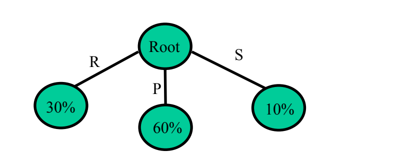


Figure 1. This is a tree expert with depth 1 (only one human action will be accounted for in prediction)

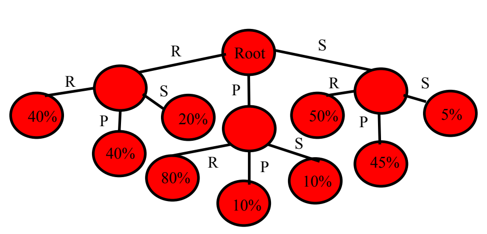


Figure 2. This is a tree expert with depth 2 (two actions will be accounted for in prediction)

Sleeping Hedge Algorithm

There is a tree constructed of experts at each node that are either asleep or awake. The probability distribution across each expert is updated the same way as fixed expert hedge, but only the probabilities of the experts that are on the path of the human history (awake) are updated.

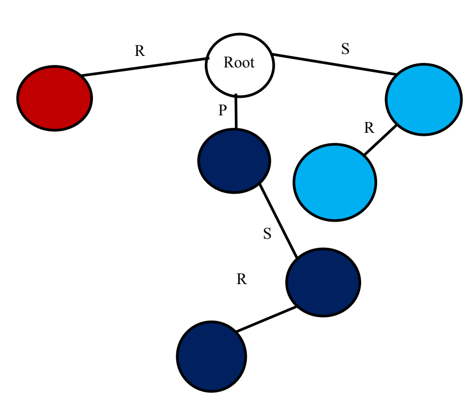


Figure 3. This is a tree with three different paths: R, SR, and PSR. Each action is recorded and reverse-traced in order to make a better prediction based off of patterns.

Sleeping AdaNormalHedge Algorithm

There is a tree construction of experts similar to sleeping hedge. The only key difference is that AdaNormalHedge is a parameter-free algorithm that avoids the uncertainty from the hyper-parameter η in sleeping hedge.

$$R_{t,i} = \sum_{i=1}^{N} \widehat{\ell_{t,i}} - \ell_{t,i}$$

$$C_{t,i} = \sum_{i=1}^{N} |\widehat{\ell_{t,i}} - \ell_{t,i}|$$

$$\omega(R, C) = \frac{1}{2}\left(e^{\max(R+1,0)^2/3(C+1)} - e^{\max(R-1,0)^2/3(C+1)}\right)$$

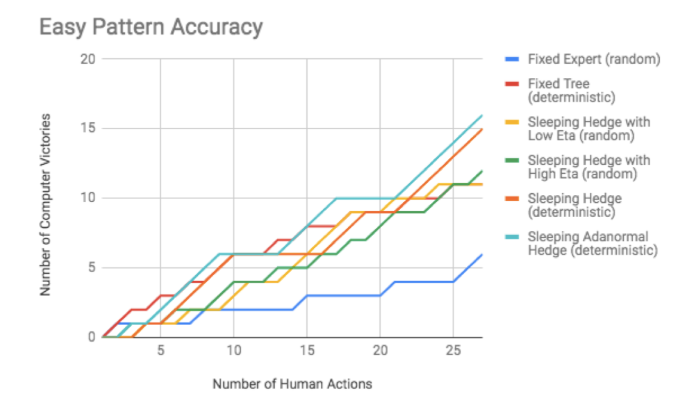
$$P_{t,i} = \frac{\omega(R_{t,i}, C_{t,i})}{\sum_{j=1}^{N} \omega(R_{t,i}, C_{t,i})}$$

N is the number of awake experts,
R is regret, C is absolute regret, and ω is weight

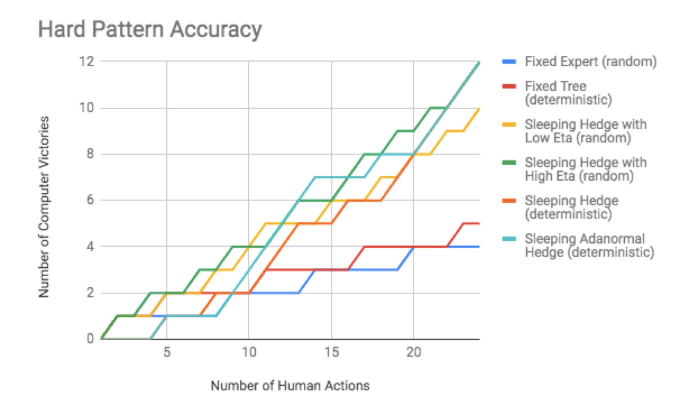
Trials

I have performed two different trials to test the proficiency of each algorithm in making the right predictions:

1. **Easy Patterns** (RPS X 3, SPR X 3, PRS X 3)



2. **Hard Patterns** (RPSSPR X 2, SPSRSR X 2)



Conclusions

- In both graphs, the **fixed hedge expert** performed the **worst** due to there still being a possibility of selecting an action that has a reduced probability.
- The **fixed tree expert** performed **well** with the **easy** pattern, but **failed** to perform as well when presented with a **difficult** pattern due to the limited depths of each tree.
- **All sleeping expert algorithms** did a **sufficiently better job** at learning and predicting due to their inherent ability to limit the scope of experts through which the computer selects a prediction. By limiting the scope of selected experts only to those who are awake, the computer is better able to make a more accurate prediction based off of previous human history.
- A **deterministic** approach, which automatically selects the expert with the highest probability, **outperforms** a **random** approach, which still has a chance of selecting experts with lower probabilities.

Figure 4. I used an interface platform called Qt to develop a playable Rock-Paper-Scissors Artificial Intelligence.

Figure 5. I used an Integrated Development Environment called C++ to initially code my different algorithms.

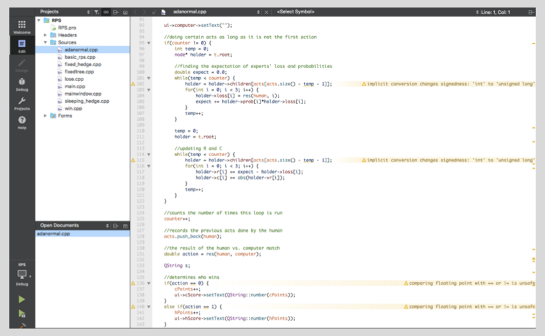


Figure 6. This is an example of some of the code where I integrated the algorithm from C++ into the Qt developer in order to test my algorithms' efficiencies.

Next Steps After Having Done this Research

I believe that my future plans will first involve making these algorithms more applicable, through which there can be more than 3 actions that are considered. After having done so, I hope to apply the skills and concepts that I learned in this program to other engineering pursuits that can hopefully reduce the net suffering of our world. That is my passion and I hope that I can use Artificial Intelligence to do so.

Acknowledgements

I would like to thank Professor Haipeng Luo for giving me the opportunity to work here this summer. I would also like to thank my Ph. D mentor, Mengxiao Zhang, for teaching me the necessary skills to understanding Artificial Intelligence, my lab mate Alec Bernardi for working with me, and the SHINE team for providing me with this amazing experience.