

Introduction

Machine learning is the concept of creating programs that are able to “learn” from inputs and experience resulting in different behaviors without explicitly being programmed to have those behaviors. What I worked on during this program was how to implement machine learning into creating a program that could beat humans in rock paper scissors. The idea here is that humans have patterns and can’t be truly random, so a program can catch on to those patterns after a couple rounds, and eventually start predicting the human’s next move more and more consistently. For my final project, I created a program using a sleeping expert tree that used AdaNormalHedge which could pretty consistently beat anyone in rock paper scissors in a game to 100 rounds.

Impact of Professor Luo’s Research

Professor Luo’s research is mainly about the theoretical and applied machine learning. An example use of machine learning would be when a website such as Netflix creates a recommended tab for a consumer based off of what they have previously watched. Netflix uses programs that predict what shows someone will like based off of shows they have previously watched. The He created Adaptive Normal Hedge (AdaNormalHedge), which is an algorithm used for machine learning. During the SHINE program, I learned about AdaNormalHedge and how to implement it properly into my final project. The field in artificial intelligence and machine learning is growing more and more, research done in these fields could lead to very advanced technologies and AIs for the future.

Skills Learned

The most fundamental thing I have learned from this program is how to program well in C++. Previously, I would have considered myself a very inexperienced programmer, but I believe I have greatly improved over this summer, and I can now program more advanced ideas such as data trees. Some of the more specifics on what I’ve learned would be specifically about machine learning. I have learned about numerous algorithms such as Hedge and AdaNormalHedge. Hedge is a formula that can be used to adjust the probability of a specific “expert” which is used to predict incoming patterns. The one downside of Hedge is that there is a constant, η , which isn’t perfected, and different values of η can change the resulting probability. However, with AdaNormalHedge, there is no constant, which is why it is theoretically better, but it is much more complicated.

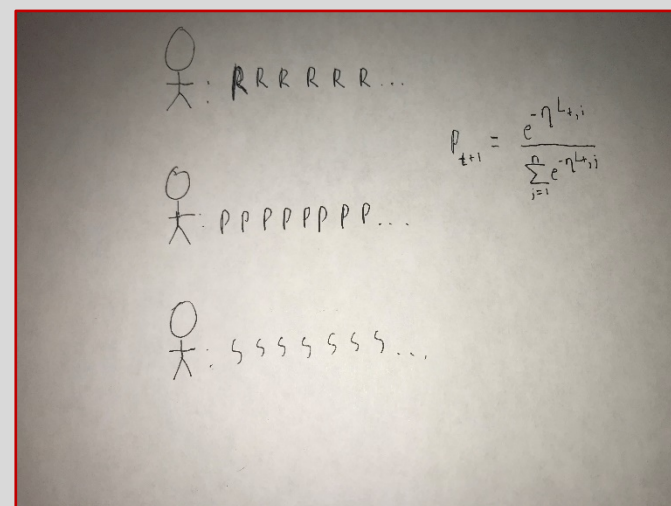
Figure One

Initially: Set $R_{i,0} = 0$, $p_{i,1} = 1/N$ for each i .
For $t = 1, 2, \dots$

1. Each action i incurs loss $\ell_{i,t}$.
2. Learner incurs loss $\ell_{A,t} = \sum_{i=1}^N p_{i,t} \ell_{i,t}$.
3. Update cumulative regrets: $R_{i,t} = R_{i,t-1} + (\ell_{A,t} - \ell_{i,t})$ for each i .
4. Find $c_t > 0$ satisfying $\frac{1}{N} \sum_{i=1}^N \exp\left(\frac{(|R_{i,t}| +)^2}{2c_t}\right) = e$.
5. Update distribution for round $t + 1$: $p_{i,t+1} \propto \frac{|R_{i,t}| +}{c_t} \exp\left(\frac{(|R_{i,t}| +)^2}{2c_t}\right)$ for each i .

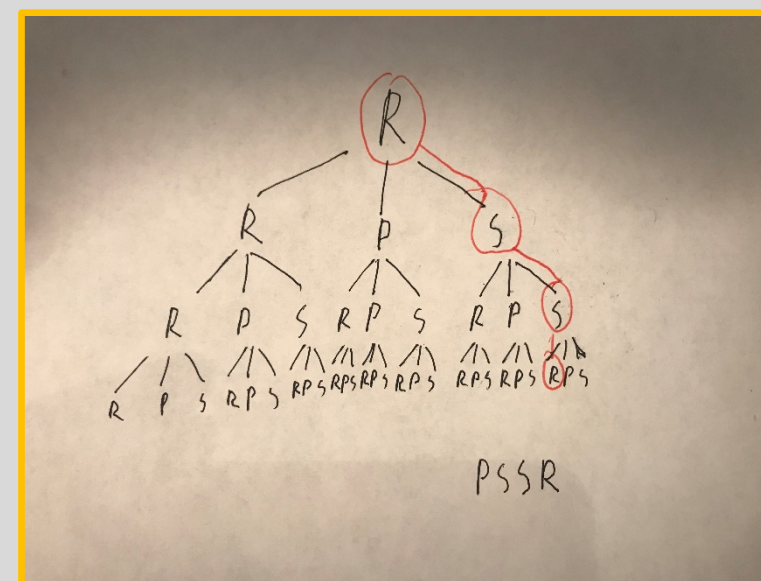
Here is the algorithm for AdaNormalHedge. This algorithm is more complicated than normal Hedge, but it doesn’t have the parameter η , which is what makes this algorithm better.

Figure Two



This is an example of the first concept we worked on, which is machine learning used fixed experts. The idea here is that the program uses three “experts”, with each predicting either rock, paper, or scissors. The probability of the program picking each expert changes based on hedge, which is the formula in the top left.

Figure Three



Here is an example of a tree for our rock paper scissors AI. The tree goes in reverse, so that the human’s last input is the start of the tree, and the tree will be infinite as time goes on.

Relation to STEM Coursework

One of the courses I will be taking next year as a senior is Intro to Programming. A major benefit of this Shine program is how much I learned as a programmer, and I am confident this experience will help me with my class this upcoming school year.

```
187 return pow(e, (((y) * (y))/(3 * CC)));
188 }
189 double rcWeight(double RR, double CC) {
190     return (0.5 * ((rcFunction(RR + 1, CC + 1) - (rcFunction(RR - 1, CC + 1)))));
191 }
192 void updateAll() {
193     for (int k = 0; k < awakeExperts.size(); k++){
194         if (awakeExperts[k] -> prediction == humInput) {
195             awakeExperts[k] -> expertLoss = awakeExperts[k] -> expertLoss + 1;
196             awakeExperts[k] -> instLoss = 1;
197         }
198         else if ((awakeExperts[k] -> prediction == S && humInput == R) or (awakeExperts[k] -> prediction == P
199             && humInput == S) or (awakeExperts[k] -> prediction == R && humInput == P)){
200             awakeExperts[k] -> expertLoss = awakeExperts[k] -> expertLoss + 1;
201             awakeExperts[k] -> instLoss = 1;
202         }
203         else {
204             awakeExperts[k] -> expertLoss = awakeExperts[k] -> expertLoss;
205             awakeExperts[k] -> instLoss = 0;
206         }
207         awakeExperts[k] -> expertProb = rcWeight(awakeExperts[k] -> RR, awakeExperts[k] -> CC) /
208             expertWeightSum;
209         ltxpectation = ltxpectation + awakeExperts[k] -> expertProb * awakeExperts[k] -> instLoss;
210     }
211     for (int h = 0; h < awakeExperts.size(); h++){
212         awakeExperts[h] -> RR = ltxpectation - awakeExperts[h] -> instLoss;
213         awakeExperts[h] -> CC = abs(ltxpectation - awakeExperts[h] -> instLoss);
214     }
215 }
216 void chooseExpert() {
217     double expprobabilityNumber = distribution(expertGen);
218     cout<<expprobabilityNumber;
```

Advice for Future SHINE Students

I was really interested by the topics that I learned about throughout this program. My only advice to future SHINE students is that you shouldn’t be afraid to ask your mentor questions, because they are happy to help, and they are very good at understanding the problem and helping out.

Acknowledgements

I would like to thank Professor Luo for giving me this opportunity to learn this very advanced research. I would also like to thank my mentor Chung-Wei Lee for helping me out every time I needed him during my project. Furthermore, I would like to thank Justin for going through this lab with me and helping me as well. Finally, thank you SHINE for giving me this great experience.