Tyler Tracey

Adrian Sanders

Ty Downs

December 4, 2013

Chuan

Intro to AI

Probabilistic Reasoning

The goals of our project were simple; we wanted to develop a psychic agent that could predict choices a player, or human, would make. We wanted to do this using three different methods, basic probability, advanced pattern recognition, and machine learning (or some way for the agent to remember past results and analyze them for patterns). Our true goal is to create an agent that can guess the player’s choices at a higher rate than just randomly guessing would.

Our project is simple on the outside. We give a player some number of arbitrary choices (for the final version of the project we gave them three choices). Next, we have the agent “guess” what the player will choose. The player then makes a choice, and we compare it with the agent’s decision and reveal whether or not the AI is correct.

Our initial findings were interesting. We decided, early on, that it is much more difficult to create an intelligent agent for this sort of environment than we thought. Adding even one extra choice to the list of choices makes the agent’s job exponentially harder. We also found that simple probability based on the players past choices is a very weak method of making decisions. We decided to add a psychological aspect to the program that searched the players past choices and basically looked for a pattern or “motive.” Utilizing machine learning is far more difficult than either (so difficult in fact that we were not able to implement it).

When working with simple probability we made two assumptions. The first assumption is that every time a player chooses something, it increases the probability that they will later choose it again. This is wrong because the player’s choices are completely independent. There is no weight to the choices a player makes, so no choices, either past or planned future choices, will impact the choice that a player is making currently. The second assumption we made is that if the player has yet to choose one of the options, they will eventually choose it at least once. This is also wrong. In fact, the player may never choose an option. They might choose only one option the whole time, or choose only two (or n-1) and keep alternating. In summary, we found that simple probabilities are not a good tool for this project.

Next we went on to implement Psychology. Basically we wanted to look at the choices of the player and try to determine patterns. This turned out to not be very easy to do, but it was effective. Our most effective method was to try to discover patterns manually. We decided that, should we implement machine learning, this would be unnecessary as the pattern recognition would come naturally to the agent.

We realized, during testing, that our agent was “trickable.” His patterns were noticeable by players who understood how the agent worked, or what the agent’s goal was. We let five players play the game with different knowledge (see the table below). It became apparent to us that players who knew about the agent and its goal were more likely to “trick” the agent, or attempt to.

|  |  |  |
| --- | --- | --- |
| Player | Knowledge of AI? | Agent’s success rate |
| 1 | Yes | 34% |
| 2 | No | 49% |
| 3 | Yes | 21% |
| 4 | No | 51% |
| 5 | Yes | 27% |

In response to this knowledge we tried to implement a reverse psychology aspect. During the agent’s execution, it would keep track of the player’s choices and at each step try to decide if they player was going to trick them. If the agent believes that the player is working to trick them it will let the player keep going, and attempt to pick up what “tricks” the player is using. It would then use this knowledge against the player and expose their tricks by picking their choices successfully. This approach did not work well, because it is difficult to make the agent as adaptive as a human player. Even if the agent catches on to a player’s deceptive patterns, the human can changes patterns or “tricks” much more quickly than the agent can identify them. In the end we changed the real-time output of the program to try to mask the agent’s decisions from them as much as possible.

Our results were actually pretty satisfying. We were able to develop a system that correctly guessed what the player would choose roughly 42% of the time. However, this is if there are only three choices. Adding even just a fourth choice decreases our rate of success to 23% (lower than randomly guessing). We realize that machine learning, or some way for the agent to remember all past results and analyze those results, is necessary to the project. The agent does better the more data it is, so some storage of past data would ultimately be powerful for the agent. We believe that, with machine learning, our agent could become close to sentient (with regards to this arbitrary decision-making process). One of our members plans on fleshing out the project in the future to include machine learning and also some kind of weighting on decisions so that the agent can be applied to realistic situations.