ADAPTIVE POKER BOT BASED ON PLAYER-MODELLING

Bjrn Hasager Vinther & Nicolai Guldbk Holst
bhas@itu.dk ngul@itu.dk

A thesis submitted for the degree of

Bachelor Softwaredevelopment

May 2014

Abstract

this is abstract

Preface

This is a bachelor thesis consisting of 15 ETCS points. The thesis has been written exclusively by Nicolai Guldbk Holst and Bjrn Hasager Vinther, both studying *Software development* at the IT University of Copenhagen. It has been written in the spring semester 2015. Our supervisor was Kasper Sty.

Glossary and abbreviations

Contents

List of Figures

Part I
Theory

1 Glossary

Poker terms Neural network terms

2 Problem statement

3 Theory

3.1 Bayes theorem

3.2 Monte Carlo simulation

We make decisions every day and before during that, whether conscious or subconscious we are calculating risks of doing just that. It is impossible to predict what the future holds for us, but the Monte Carlo simulation will let you see which possible outcomes you can expect from the different decisions. A Monte Carlo Simulation will let you know all possible outcomes which can occur and how possible it is that that outcome will be the one. Basicly it is like saying what if thousands of times and seeing what happens and how often something happens. If we were to roll a dice and find out what the possibility is that two dices rolls 8. We would throw the dices 100 times, and for each time we would record the throw. That way we have how many times we roll each of the outcomes and therefore we can predict the possibility of that outcome. Lets say we rolled 8 12 times out of a 100 rolls. The possibility of rolling 8 is 18In poker this could maybe be used to predict what cards are popping up in the flop and so what are our chances of having a hand that is good. Another alternative could be to determine whether a player is an experienced player or a beginner. If we were to look how often a player wins versus how many times you win, it could be a guideline to see how big a chance you have of winning the game.

3.3 Nash equilibrium

"Nash equilibrium refers to a condition in which every participant has optimized its outcome, based on the other players expected decision."

This theory works with the assumption that a party knows the other parties strategies. If all parties have an optimized strategy compared to the other parties strategies then this situation is said to be in Nash equilibrium.

In poker this is useful in a situation where your opponent plays really aggressive and goes all-in a lot. Depending on the size of your stack compared to the big blind different hands is profitable to call in order to win the big blind. For this purpose a chart has been created to show which hands are profitable to call depending on the size of your stack. Likewise a

3.4 Neural network

Neural networks is models which are somewhat inspired by biological neural networks. Neural networks is considered to be one of the best methods to classify input-patterns. Neural network is for example used to recognize and reading handwriting and speech recognition. Humans are very good at recognizing visual patterns, but if we were to write a program that could do just that, it becomes alot more difficult. We have simple intuitions when we are trying to recognize different shapes. But to explain to a program how to recognize for example a Y would be something like a straight line with two lines pointing out from it at the top to each side at a 35 degree. When we try to make rules like this to let the program recognize letters we can quickly become lost in what we expect it to look like and the special cases. Neural networks has a different approach to the program which is much more suitable than describing each letter in some mathematical algorithm formula. We give the neural network a very large amount of handwritten letters, which

shall be used in the training of the neural network.

The neural network will then use the examples to automatically determine

some rules for reading the handwritten letters. Of course this example doenst have a lot of different types of the letter A so if we want to let the neural network become better at reading the handwritten letter A we would have to give it an example with many more examples of the letter. It would improve the accuracy, therefore it is better to provide the neural network with a thorough example.

"...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs. A neural network consists of neurons which can send signals to each other. Each neuron processes its input signals and determines if the signal needs to be sent further. A neural network is not being fed with rules but instead examples that it can make rules from. With that a neural network is able to learn new skills, which is something that the traditionel computer cant do.

3.5 What did we use?

4 Method

5 Implementation

5.1 Poker game system

5.2 Implementation of the neural network

6 Test

7 Conclusion