Ray Goh Xin Zhen & Son Le
2/5/2018
CSC 180 Intelligent Systems
Due: 2/23/18

Blackjack Strategy Intelligent System: Project Report

Introduction and Background

Blackjack is a game where a player would compete against a dealer by getting card values to 21 without going over, using multiple strategies as HIT, STAND, DOUBLE DOWN, a player can use multiple strategies to win the dealer or hoping for the dealer to get a burst. There are also strategies to DOUBLE DOWN to take advantage when odds of winning is in your favor to In a game of blackjack a player is often dealing with ambiguity regarding which move to make. However, although luck plays a big role in blackjack, having strategies derived from calculated probability does help reducing the odds of losing. Due of one of the fact that it is the most widely played casino game in the world, our purpose and goal of this project is to design an artificial intelligence that would give advice to a player on the highest probability winning move. If a person does not understand the rules well or plan his strategies, blackjack is a game which a player could easily lose from. Hence we decided to design an intelligent system to overcome that problem.

Disclaimer: This artificial intelligence program **DOES NOT** guarantee wins in a match. However, The intention of this program is to advise on the move that a player has more odds of winning. This program does not correlate with counting cards or any illegal gambling methods,

it only calculates the winning odds and chooses the best move.

Knowledge Engineering Extraction

Our expert, Sin Eng Joo, is an uncle of Ray Goh is a retired electrical engineer from Intel who after retirement has been very regular going to the casino over many years and based on his experience and knowledge we are confident on having him as our expert as a blackjack strategy advisor. He was able to spend time to tell us on how he would have a set of rules that determines his actions based on the situation he, he was also able to plot out a simple graph that shows the right moves to take based on the situation. However the simple figure he provided wasn't sufficiently detail oriented comparing to our demands for our project . We had to seek further help through online research in order to figure out on a more accurate manner that which moves actually contain more winning odds.

Expert System Design

In a blackjack game, each player and dealer is being dealt with two cards, which the player has options to draw a card, stand or double down(hit while doubling their bet). A player has to decide whether the next card that he/she hit would get him/her as close to 21 as possible, or hoping that a dealer would draw a card and burst (Over 21).

Judging based on our interview with Eng Joo and our understanding of the game, we had listed down the different possible scenarios and extracted our expert's strategy he would take given that specific occasion. We can then decided to map the moves that were determined in that situation. For example a scenario when a player gets dealt with a hand of a 20, the strategy would

always be to STAND. As the probability of drawing a 1 and not busting the hand will likely be less than 7%. Apart from the situation that have a specific strategy, there are certain situations where the player would have to face a dilemma because it could be a 50/50 chance. We figured that the only way to deal with a situation like this is to consider external factors. Judging between the percentage of high or low cards that have been dealt during previous hands, we can slightly estimate the chance of hitting the next card and not getting burst by a small percentage increase. This is where the fuzzy logic steps in, by determining by the number of high cards that came out during previous matches which are cards that consists of high cards like (10, J ,Q, K,A). For example, a player kept track by counting the high cards, he estimated about 80 high cards that came out and there was only about 2 desks(108) left on the table (Assume 5 desk total = 270 cards). that mean the chances of the remaining cards being a low count would be very high (only 100 out of 270 high cards), so when facing a 50/50 situation on whether to HIT/STAND. We can allocate the move which consist of a slightly higher percentage of winning chance. Although given that the percentage increase isn't too significant, maximising the winning odds is still the best bet.

Matrix table that indicates move to make given each conditional situation

	Dealer's display card									
	2	3	4	5	6	7	8	9	10	A
4	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit
5	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit
6	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit
7	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit
8	Hit	Hit	Hit	DD	DD	Hit	Hit	Hit	Hit	Hit
9	DD	DD	DD	DD	DD	Hit	Hit	Hit	Hit	Hit
10	DD	DD	DD	DD	DD	DD	DD	DD	Hit	Hit
11	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD
12	Stand/Hit	Stand/Hit	Stand/Hit	Stand/Hit	Stand/Hit	Hit	Hit	Hit	Hit	Hit
13	Stand/Hit	Stand/Hit	Stand/Hit	Stand/Hit	Stand/Hit	Hit	Hit	Hit	Hit	Hit
14	Stand/Hit	Stand/Hit	Stand/Hit	Stand/Hit	Stand/Hit	Hit	Hit	Hit	Hit	Hit
15	Stand/Hit	Stand/Hit	Stand/Hit	Stand/Hit	Stand/Hit	Hit	Hit	Hit	Hit	Hit
16	Stand/Hit	Stand/Hit	Stand/Hit	Stand/Hit	Stand/Hit	Hit	Hit	Hit	Surrender	Surrender
17	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand
18	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand
19	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand
20	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand

*Blue region indicates the **fuzzy logic** area.

Fuzzy Logic Determining Graphs:

In diagram 1 of High cards, we have 5 decks of cards totalling 270 cards and 100 approximate count of high cards. We set a positive line from 50 to 100 counts in blue line, zero cards is from 25-75 counts in black and negative is from 0 to 50 counts in red.

Number of High Cards (10-J-K-Q-A)
Negative (Stand) = (25,1) , (50,0)
Zero (Hit/Stand) = (25,0) , (50,1) ,(75,0)
Positive (Hit) = (50,0) , (75,1)

Diagram 1- High Cards

In diagram 2 of Decks, we have 5 decks representing in x-axis, we set a positive line from 3 to 5 decks in blue line, zero cards is from 2-4 decks in black and negative is from 0 to 3 decks in red.

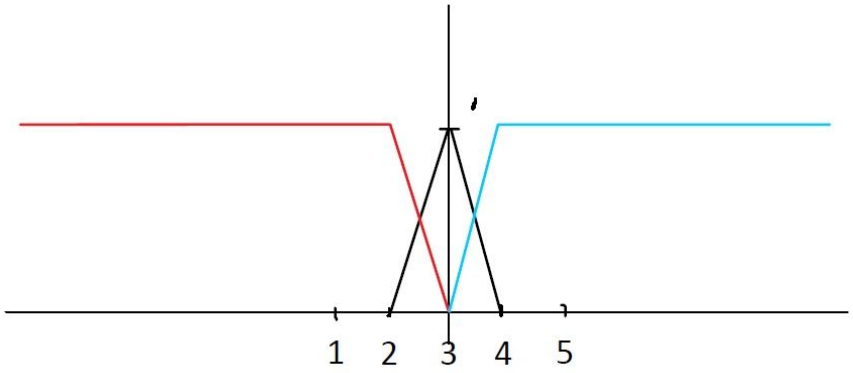
Number of Decks (Scale from 1 to 5 deck)

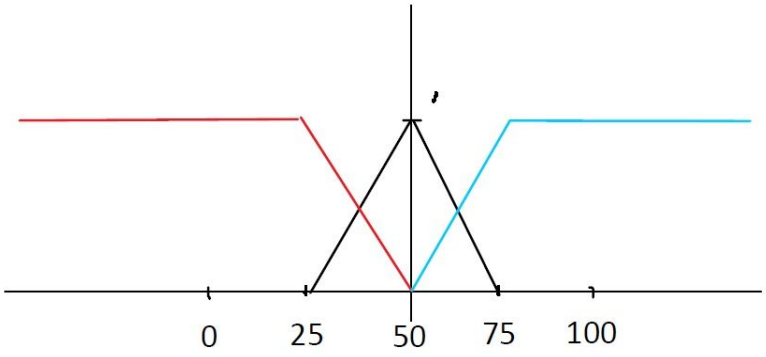
Negative (Stand) = (2,1) , (3,0)

Zero (Hit/Stand) = (2,0) , (3,1) ,(4,0)

Positive (Stand) = (3,0) , (4,1)

Diagram 2 - Desk



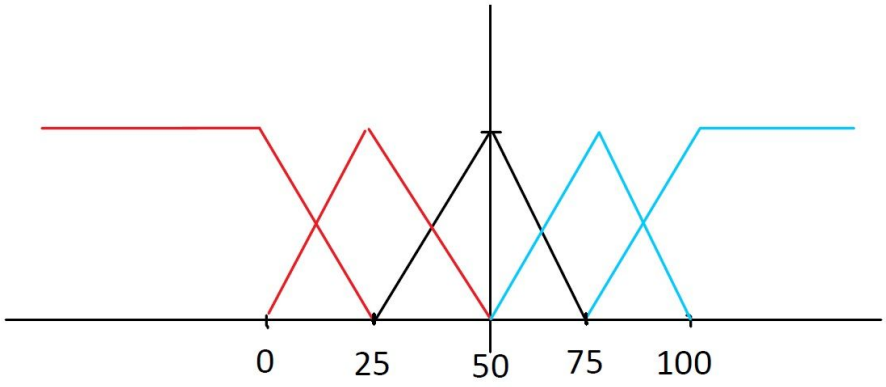


In diagram 3 of Decision, we have 0 to 100 percent chance of winning on the X-axis, we set a big positive line from 75% to positive infinity in blue line, a positive line from 50% to 100% in blue line, zero is from 25% to 75% in black , negative is from 0% to 50% in red and big negative is from negative infinity to 25% in red.

Decision Diagram:

Big Negative (Stand) = (0,1) , (25,0)
Negative (Stand) = (0,0) , (25,1),(50,0)
Zero (Hit/Stand) = (25,0) , (50,1) ,(75,0)
Positive (Hit) = (50,0) , (75,1), (100,0)
Big Positive (Hit) = (75,0), (100,1)

Diagram 3 - Decision



Based on Fuzzy Associative Matrix (FAM):
We have a table of Matrix

Desk \ High Card	Positive	Zero	Negative
Positive	**BP**	**P**	**Z**
Zero	**P**	**Z**	**N**
Negative	**Z**	**N**	**BN**

BP and Positive indicates the confidence level to HIT.
Zero indicates balance between HIT and STAND.
Negative and BN indicates the confidence level to STAND.

So, we can test our fuzzy logic given a particular scenario
For example: If a player has a card value of 13 total in his hand and the dealer's face up card is 5, then he/she has to answer two questions in order for the expert system to make a decision.

1: Estimate how many high Cards (..9-10-J-Q-K) have he/she seen?

In this case, User inputs: 90

2: Estimate how many deck of cards are left on the table ?

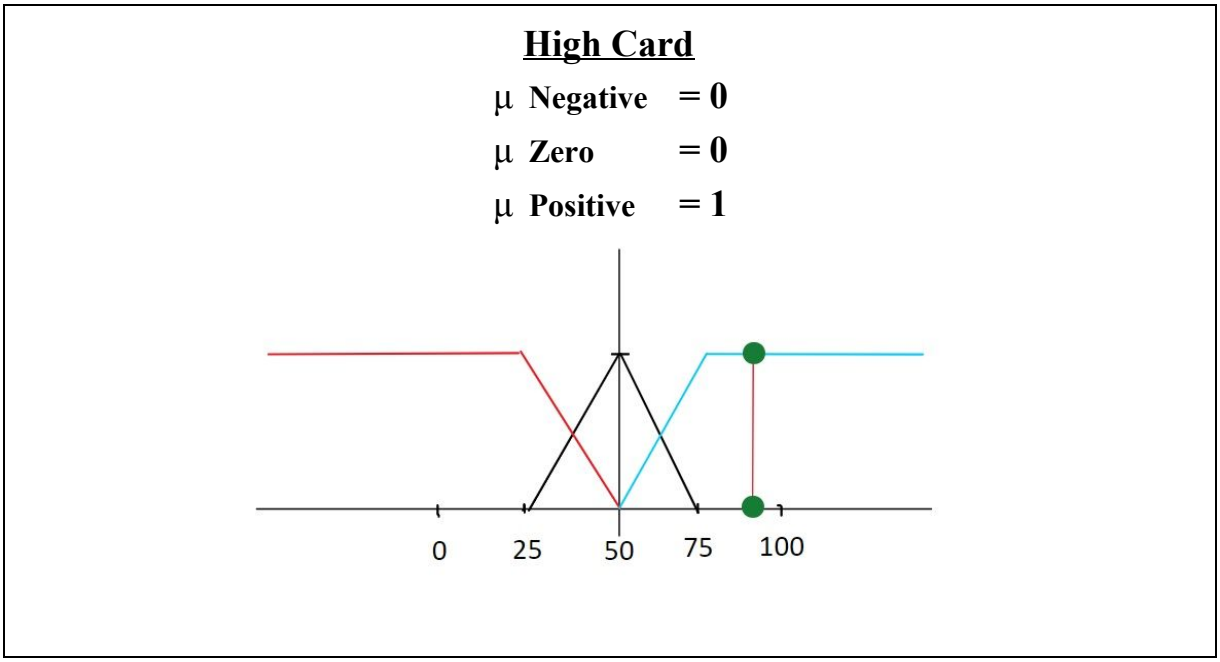
User input : 3

So we now have 2 input values for step 1 fuzzification

High Card= 90

Decks = 3

Step 1: Fuzzification At 90 counts, we have positive 1 in red dot

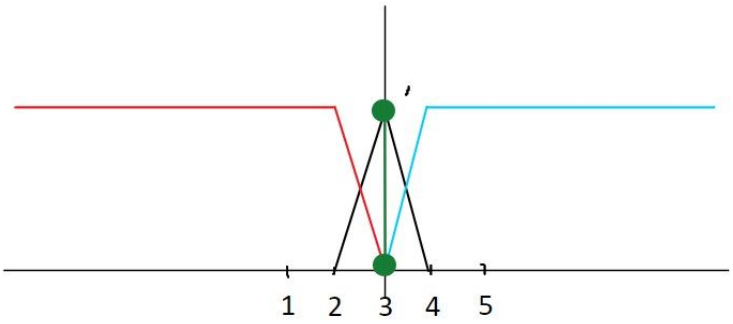


High Card

μ Negative = 0

μ Zero = 1

μ Positive = 0



Step 2: Rule Evaluation: Now we evaluate every rule in the FAM. The rule evaluation is done using the fuzzy definition of AND.

X AND Y = MIN (μ(X),μ(Y))

If HighCard = P (1) **AND Deck = P (0)** **Then Decision= BP (0)**	**If HighCard = P (1)** **AND Deck = Z (1)** **Then Decision= P (1)**	**If HighCard = P (1)** **AND Deck = N (0)** **Then Decision= Z (0)**
If HighCard = Z (0) **AND Deck = P (0)** **Then Decision= P (0)**	**If HighCard = Z (0)** **AND Deck = Z (1)** **Then Decision= Z (0)**	**If HighCard = Z (0)** **AND Deck = N (0)** **Then Decision= N (0)**
If HighCard = N (0) **AND Deck = P (0)** **Then Decision= Z (0)**	**If HighCard = N (0)** **AND Deck = Z (1)** **Then Decision= BP (0)**	**If HighCard = N (0)** **AND Deck = N (0)** **Then Decision= BN (0)**

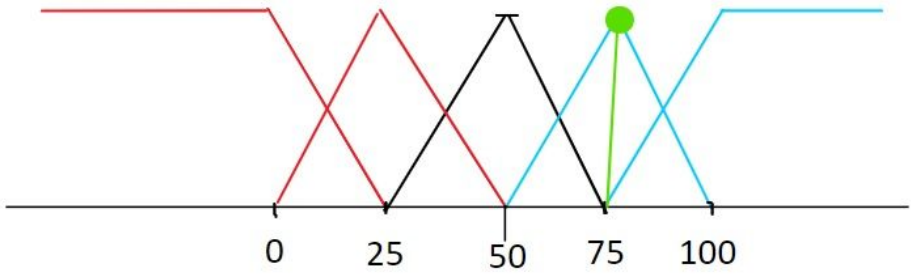
In this case we only have one non-zero output membership, so this gives us the exact value on X-axis.

BP and Positive indicates the confidence level to HIT.

Zero indicates balance between HIT and STAND.

Negative and BN indicates the confidence level to STAND.

Step 3:Aggregation At Decision Diagram Y-axis P = 1 , we have X-axis as = 75 %, which is a good amount of lower cards left in the remaining decks to be drawn and the player will have more chance of drawing a low value card and not exceeding 21, since he/she has a count of 13 and the number of high cards are significantly lower now.. This is the fuzzy result !



Summary: In this game, Assume that we have 5 decks of card on the table and each deck has 5 high card (10-J-Q-K and A), each high card has 4 cards(Hearts-Diamonds-Clubs-Spades), We have 100 of High Cards in 5 Desks. If the player can count or estimate how many high cards have been on the table and how many desk left on the table, In this case, He counted 90 high cards and 3 desk left on the table, So the dealer only has 10 high cards left in 3 desk (10/156 cards). So the player has very high chance of hitting a low card and win the game since he has a score of 13 and dealer has a face up card of 5.

Conclusion

Overall, the approach on plotting a graph and mapping specifich and the intelligent system works according to our expectations. We even conducted tests setting up our own table blackjack game inviting a few people while using the advice our artificial intelligent system provides to play. The result of strategies was exactly as how we expected and designed.
However, there were a couple of issues that we were not able to solve, the ambiguity between having an ACE which in blackjack could be 1 or 10 in the first hand. Our program does not detect when a player's hand contain an ACE card or not. The player has to decide between considering his ACE value as 1 or 10 and enter the corresponding count value as input into our system, as ACE can be counted as both 1 and 10 in blackjack.

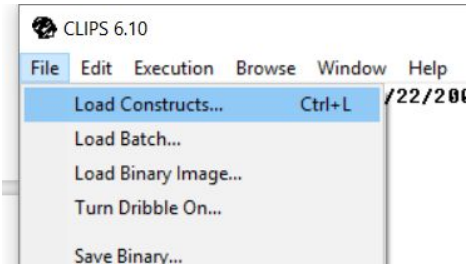
APPENDIX A (installation guide)

Download link for Clips: http://awesom.eu/~quentin/FuzzyCLIPS/fzclp610dWin.zip

Once you have Clips installed in your system, unzip the file named **BlackJackAI.zip** that was given and you should be able to see the files :
BlackJackAI_CLIPS.clp

Step one: Launch the CLIPS executable program corresponding to which operating system you are using.

Step two: Export the **BlackJackAI.clp** file into clips by going into file>load construct>BlackJack.clp *where you stored the clip file



You should get an output that prints TRUE at the end after loading constucts:

```
Defining defrule: NN =j+j
Defining deffacts: startup
TRUE
```

Step three: Enter the following commands accordingly
: (reset)
: (run)

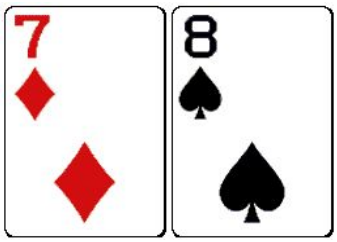
Voila! If you have followed all the steps above you should be good to go and look at Appendix B for a more detailed user guide on how the program works.

APPENDIX B (user's guide)

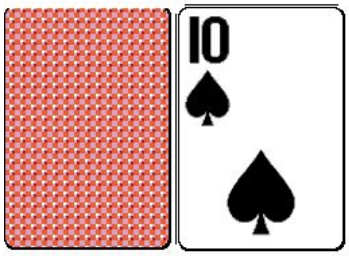
How this program works (In a nutshell):
A user enters the total count value of his hand, and enters the dealer's faced up card value. Then the system decides which move a player should make. When dealing with an ambigous situation where both moves (HIT or STAND) contains similar odds, user is prompt to estimate the number of high cards that have came out in previous hands and also how many decks are left with the dealer, the user could also select NO if he does not know the estimate and our program will spit out a base case decision.

A couple of examples will be illustrated as below:

Situation 1:
Given faced with this situation at a typical table (Set rule without fuzzy logic):
Players hand (You):



Dealers hand:



Use of our program:

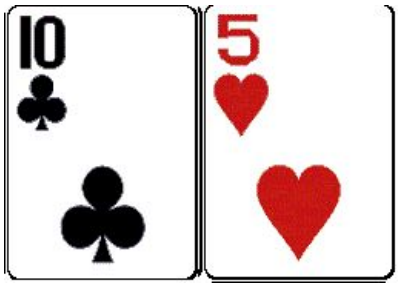
```
What is the total count value of both cards in your hand? 15
What is the dealer's face up card value ? 10
HIT
```

Our program then decides to tell the player to HIT, as it calculates that if the dealer has a 10, chances of the face down card being above 5 would be more likely, so the best option is to HIT a card.

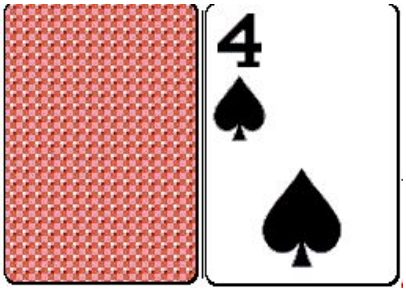
Situation 2:

Given faced with this situation at a typical table (Fuzzy logic situation):

Players hand (You):



Dealers hand:



Use of our program:

```
 What is the total count value of both cards in your hand?15
 What is the dealer's face up card value ?4
Could you roughly estimate how many high cards (10-J-Q-K) have you seen
otherwise enter NO : 80
Could you estimate how many deck of cards are left to be dealt? : 3
Your chance of winning is 75.0 % it favors you to HIT
```

Since a good amount of high value cards (10 counts) have already been dealt, the amount of lower value cards still in the deck will be higher, thus, the system calculates the probability from plotting on the fuzzy logic graph and determines the result. In this specific situation, it favors the player if he/she choose to HIT.

References

Our expert: 'Expert blackjack player and analyst' - *Sin Eng Joo - Uncle of Ray Goh*, a veteran casino player that has been a regular blackjack player for many years. He was very generous to share his expertise and strategies with us to help us complete this project.

Ideas for plotting a fuzzy logic graph :
Fuzzy Logic. [Class lecture and slides]. *Scott, Gordon, California State University, Sacramento,* attended Spring 2018, Retrieved from <http://athena.ecs.csus.edu/~gordonvs/180/180ref.html>.

Reference of blackjack strategies:
'Comparing AI Archetypes and Hybrids Using Blackjack' *Robert Edward Noonan Minnesota State University*- Mankato, viewed February 2018, <https://cornerstone.lib.mnsu.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1335&context=etds>.