

**Abstract:**

The Rule Learning AI is a program based of a previous card game playing program which allows one to play a simple game extensible through loading Rule objects which can allow players to play cards when they previously shouldn't, or punish them for playing specific cards under certain conditions. The AI is meant to form a rough estimate of what a rule is through observing anomalies in the game state and when these anomalies choose to pop up. In the end, it is able to play the game while approaching or avoiding rules which may or may not be beneficial to the player who triggers them.

Carleton University

# **Rule Learning AI**

Ian Jaffe

Supervisor: Dr. Michel Barbeau

Submitted April 9<sup>th</sup>, 2008

## **Acknowledgments:**

Thank you, Guillaume, for putting up with me through the original project, and all these years in university.

## **Table of contents:**

Subject	Page Number
1. Introduction	5
2. Background Information	6
3. Description of Data Collection	7
4. Description of AI	8
4.1 Method of Operation	8
4.2 Final Analysis	9
4.3 Reasons for Limitations	9
5. Testing of the AI	11
6. Conclusion	12
Appendix A: Sample Test Data	13
Appendix B: Program Operation	16

## **1. Introduction:**

The Rule Learning AI is built upon an existing program created by myself and Guillaume Couture-Levesque. The program is a distributed card game similar to crazy eights. By default, each player takes turns playing a card of the same suit or number of the previously played card, or draws a card if they cannot. However, the game is extensible with the ability to add “Rule” type objects. The Rule objects have 3 functions; one which modifies what is acceptable play in the card game, one which triggers from certain cards being played under certain circumstances, and the third which changes the state of the game upon the circumstances defined in the trigger function being met. A Rule has access to any of the state variables of the game, no hard limit is set on what one can program them to do. The Rule objects and what they entail are meant to be unknown to all. It is the responsibility of the player to determine what the rules entail by experimenting with playing different cards.

The purpose of the Rule Learning AI is to be able to distinguish what actions are a result from new rules being added, and what those rules are.

## 2. Background information:

The foundation program sends a copy of each Rule over as it is added, but nothing is done with it; instead, the AI keeps track of the state that the server gives out, and makes its deductions from there. The state information sent each turn is as follows:

- the player whose turn it is
- the hand of the client
- the list of all other players and the size of their hands
- what cards the last player attempted to play
- what the actual top card is

Note that while the top card is derived from what each player attempts to play, it cannot be derived from what one attempted to play due to the fact that the clients allow players to attempt to play invalid input, in the off chance that one of the rules allows such input to be played.

### **3. Description of Data Collection:**

The server, upon each turn being played, sends out the five signals as described in the background information. Whose turn it is, the list of all players and their hand sizes, the cards last attempted to play, and the top card are all broadcast, while a separate hand is sent out to each player. This information is enough for a single player to play, but not enough to run an entire game. Therefore, rather than using a copy of the game object, a smaller State object is used in this program.

The state contains all the objects sent through signals with the exception of the hand. Rules have the ability to directly manipulate hands, but due to a player's ability to only see their own hand and the logistics of this being a simulation of a card game, such actions are discounted. In a real card game, which the base program attempts to simulate, the only hand manipulation which could be carried out would be plainly visible to all players, which the program as of now has no way to emulate. Therefore, the AI has to determine hand changes such as this by comparing the number of cards in each player's hand.

With the four remaining signals, since all the data is sent as Strings to ease for display and transfer, the State object has methods to parse the data from them. As well, the simplified logic to determine what is expected from playing given cards on the game state is done here; with this data one can determine whether an anomalous action has taken place. If a Rule cannot be detected from observing differences in the hand sizes, the turn order of players, or cards being played in unusual circumstances, then it cannot be detected by the AI.

## **4. Description of AI:**

### **4.1 Method of operation:**

For every card in your hand, the AI first checks if a card is playable under the standard rules. If it is, the AI then continues to check if it is undesirable to play the card. Cards can be considered undesirable for three different reasons. If it is a joker, it is undesirable to play, because one would have to part with it. If it has the same value as another card, you also want to save that, as you can play that card as the last of its suit and follow with a new suit. And finally, if a card is similar to the cards that previously triggered adverse anomalies, and the effect of these anomalies has been considered negative, then it is undesirable as well. Similar is defined as sharing either a suit or a value with all of the recorded triggering cards.

To determine if a rule's effects are negative, or if it has even been triggered, the AI stores a copy of the state of the game after every round of signals is sent, as described above. When it senses an anomaly, it compared the state of the game with what it believes the state should be. Any differences in hand sizes are noted; if the hand of the player who triggered it grows in size, or the hands of other players shrink, it is viewed as undesirable, while the opposite is considered desirable. More weight is given to the cards of the player's hand; if one draws one card and a single opponent does as well while other opponents do not, it is clearly not worth the sacrifice.

If a card is playable, but undesirable, it will remember the card, yet continue checking for more desirable cards. It will only play an undesirable card if no others seem desirable. Inversely, if the AI deems the triggering of the rule to be beneficial, it will stop looking through the hand as soon as it finds a rule that triggers it, and play that card immediately.

#### **4.2 Final Analysis:**

Once a game has come to a finish, the AI gives one more look through all the anomalies to offer a final analysis of the cards played during anomalies and the cards played upon. If a suit appears more than 50% of the time, or a value more than 20%. Unrelated to any rule triggering, each suit should appear only 25% of the time, and each value about 7.5%, so large percentages signify a good chance of a relation of that value to the rule that gets triggered. In cases of 100% appearance, it is very likely that the value is required to be such to trigger the rule. Furthermore, if no value is significantly present in the triggerings, we can tell that they are either based on values such as hand sizes, or mathematical functions that change the value of the trigger as time goes on.

#### **4.3 Reasons for Limitations:**

As it stands, the AI is incapable of playing with rules that are functions more complex than having a constant value as the trigger. With the nature of the problem of determining the Rules from their effects on gameplay, initial versions of the AI could not work, due to the inability to determine what was the triggering condition. It is not surprising that complex polynomial equations could obviously not be determined by the AI, however the problem is

that the rules must be much simpler than expected for the AI to be able to pick up what is happening. Due to the factors of hand sizes, played card numbers and suits, the suit and number of the top card, the possibility of signals, and possible reliance on constants, any relation between two elements would be lost in a torrent of information. Furthermore, attempts to create functions that encompass the data sets would delay the AI by a noticeable amount. Playing with only paying attention to immediate constants is able to capture the most common variety of rules as well as operating in a relatively speedy manner that does not interfere with the game for the other players.

## 5. Testing of the AI

The AI, in testing, can determine which card to play in time unnoticeable to the casual player. With extremely large data sets for rule triggers, it begins to slow down, but the server log has shown to run out of memory before it became an issue that could affect gameplay.

Rules such as playing twos causing a player to draw two cards can be picked up rather quickly, while rules that require signals would not be easily caught by the AI alone. Due to the competitive nature of an AI, sending signals would be counterproductive, as if you send the signal with the incorrect input, or if the signal is not needed at all, then the odds are that one will pick up more cards than save from the rules triggering. The exception is when it has no cards to play; it randomly chooses which card to send along with a random chance of it sending the signal; the odds of this working is rather low, however. Yet the AI is still able to observe signal based rules if other players trigger them.

See appendix for some sample output from tests run using two different rules that are within the AI's bounds to understand, and one which is not.

## **6. Conclusion:**

Despite its inability to determine non-constant function based rules, the Rule Learning AI is able to play as if it is learning from what happens, taking caution to avoid playing cards associated with bad happenings while at the same time playing by the base rules well enough to provide as much of a challenge as one can in a game as highly random as this.

## Appendix A: Sample Output

This output was the end of the output captured from using a Queen Of Spades Rule, where the next player will draw 5 cards if you play a Queen of Spades. Since testing was done in single player mode, the next player is still the AI, thus this is determined to be undesirable.

1 Hearts is on top of the pile.  
2 Clubs were played.  
It is your turn.

1 Hearts is on top of the pile.  
1 Spades were played.  
It is your turn.

1 Spades is on top of the pile.  
2 Clubs and a QueenOfSpadesRule signal were played.  
It is your turn.

1 Spades is on top of the pile.  
1 Joker were played.  
It is your turn.

1 Joker is on top of the pile.  
2 Clubs were played.  
It is your turn.

2 Clubs is on top of the pile.  
6 Clubs were played.  
You won!

100% of the cards played when the rule was triggered were Spades.  
100% of the cards that were played upon when the rule was triggered were Spades.  
100% of the cards played when the rule was triggered were 12.  
100% of the cards that were played upon when the rule was triggered were 6.  
The rule does not require signals to trigger.  
Triggering this rule is undesirable.

This is the last 6 turns of output that was produced with a rule where the next player draws two cards when a two is played, once again being detrimental to the AI in single player mode.

5 Clubs is on top of the pile.  
13 Clubs were played.  
It is your turn.

13 Clubs is on top of the pile.  
11 Clubs were played.  
It is your turn.

11 Clubs is on top of the pile.  
7 Clubs were played.  
It is your turn.

7 Clubs is on top of the pile.  
7 Diamonds were played.  
It is your turn.

7 Diamonds is on top of the pile.  
8 Diamonds were played.  
It is your turn.

8 Diamonds is on top of the pile.  
8 Clubs were played.  
You won!

No noticeable ratio involving the suits of the cards played was found.  
No noticeable ratio involving the suits of the cards played upon was found.  
100% of the cards played when the rule was triggered were 2.  
33% of the cards that were played upon when the rule was triggered were 9.

The following data is from a single player game with the Sames rule, where one can play different cards with the same value at the same time. The AI cannot detect the rule as it only ever tries one card at once.

4 Hearts is on top of the pile.  
12 Hearts were played.  
It is your turn.

12 Hearts is on top of the pile.  
12 Clubs were played.  
It is your turn.

12 Clubs is on top of the pile.  
3 Clubs were played.  
It is your turn.

3 Clubs is on top of the pile.  
3 Diamonds were played.  
It is your turn.

3 Diamonds is on top of the pile.  
7 Diamonds were played.  
It is your turn.

7 Diamonds is on top of the pile.  
7 Spades were played.  
You won!  
No anomalies detected.  
The rule does not require signals to trigger.

## Appendix B: Program Operation

Requirements: A computer able to run Java programs.

To begin, run the build-server.bat in the Server folder and the build-client.bat in the AI Client folder. If you would like to play as well, you may run the build-client.bat in the Old Server folder, but keep in mind that multiple connections from the same machine are not supported. Next, run the run-server.bat and run-client.bat from their respective folders. When using the old client, you must open it after the server has been run, and if you wish to play distributed, you must add the parameter -ip followed by the destination IP to the run-client.bat. To play distributed from the AI client, just enter the IP into the bottom text field and hit the IP button to confirm. To change your name, write your name in this field and hit the name button. Once this is done, hit the Connect button to connect to the server (the old Client connects automatically, thus the changing parameters from the .bat). Once all the players have joined, from the server, press the Start button, and to add a Rule, select it from the list and press the Select Rule. Then the game is already to go!

To play as the AI, press either the Move button to play what it determines, or the Random button to play a random card. To play as the old client, select a card from the list by clicking on it, and when you're done selecting cards you may click a rule to send its associated signal. If you make a mistake, click the clear button to start over, and when you are done, click the Send button.