Experimentation on the Development of a Learning Agent Given Rules to a Game

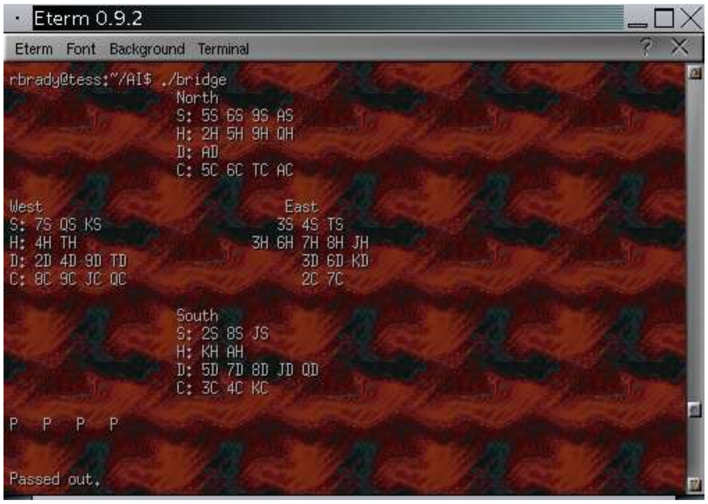
by

Robert Brady

TJHSST Computer Systems Research Lab 2004-2005

Abstract

The goal of this project was to create a learning agent for the game of bridge. I think my current agent, which knows the rules, plays legally, and finds some basic good plays, is a step in the right direction. This agent could and will be improved upon over the course of the year and will become smarter and learn faster throughout the year.



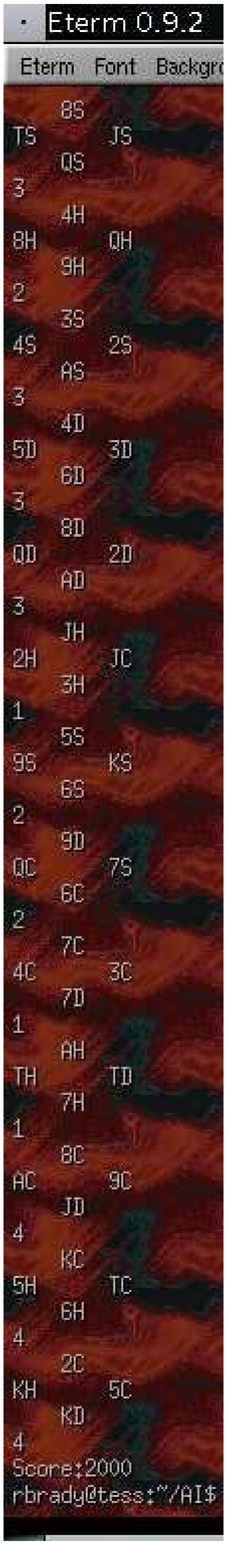
```
rbrady@tess:~/AI$ ./bridge
                North
                S: 5S 6S 9S AS
                H: 2H 5H 9H QH
                D: AD
                C: 5C 6C TC AC

West                        East
S: 7S QS KS                3S 4S TS
H: 4H TH                3H 6H 7H 8H JH
D: 2D 4D 9D TD                3D 6D KD
C: 8C 9C JC QC                2C 7C

                South
                S: 2S 8S JS
                H: KH AH
                D: 5D 7D 8D JD QD
                C: 3C 4C KC

P  P  P  P

Passed out.
```

Bidding

To the left, you can see the computer has failed at arriving at a contract. It has determined that its best option is to pass for each player. This certainly isn't the best decision because either north or south could and should have opened the bidding and therefore arrived at a successful contract. These mistakes are ones that a human player certainly wouldn't fall for, but the computer did. it shows how different the "minds" of computers and humans are.

Playing

To the right we can see a sample output of the program after it has reached a final contract. Here, the opponent to the left of the declarer leads the first card and you must follow suit if possible. The output here is of legal plays and the result of the hand is a score of 2000 points for North/South. After each trick is displayed, there is a number corresponding to who won the trick. The number one represents North and the other numbers follow in a clockwise rotation: two=East; three=South; four=West. This may not be very well played by any of the four players based upon the starting hands so the program needs some work, but it is progress based upon getting the program to learn the rules of the game and play somewhat accurately.



```
   8S
TS    JS
   QS
3
   4H
8H    QH
   9H
2
   3S
4S    2S
   AS
3
   4D
5D    3D
   6D
3
   8D
QD    2D
   AD
3
   JH
2H    JC
   3H
1
   5S
9S    KS
   6S
2
   9D
QC    7S
   6C
2
   7C
4C    3C
   7D
1
   AH
TH    TD
   7H
1
   8C
AC    9C
   JD
4
   KC
5H    TC
   6H
4
   2C
KH    5C
   KD
4
Score:2000
rbrady@tess:~/AI$
```

Background & Information

Machine learning has been researched in the past and has dealt with bridge before. This area is new, though, and anything from intelligent agents for games to the traveling salesman problem counts as part of it. An algorithm used for one problem can be applied in a similar manner to another such as the minimax algorithm or the backtracking search. To build on current work, there would have to be some sort of improvement on current bridge-playing agents such as Bridge Baron or GIB. Both of these programs play at a moderate level, but none of them can compare to an expert player. I hope to improve on that.

The reason why an intelligent bridge-playing agent has been hard to program in the past is that bridge is a closed state environment. This means that only part of the environment is observable. In games such as chess, or checkers, the agent could conceivably come up with the best solution (given enough time to think about it) because it knows where everything is. In bridge, there are certain cards that haven't been played yet and you may be able to guess where they are, but you may not be able to determine where they are certainly. This is why it is hard to program a bridge-playing algorithm because you can't make a tree for the possibilities in the environment since you don't know what the environment is.