International Journal of New Technology and Research (IJNTR)
ISSN:2454-4116, Volume-1, Issue-7, November 2015 Pages 01-05

Study on the Evaluation Function Parameters of the Checkers Game Program on Weka Platform

Li Shuqin , Xue Weiming,Yuan Xiaohua

Abstract—**Based on the machine learning algorithm provided by the Weka platform, this paper performs two studies on the evaluation function parameters of checkers game program. The first one is that, through selecting six numbers, which include the number of the black, the red, the king of black, the king of red, the black in danger, and the red in danger, assigning different weight to each number, then inputting these weighted numbers as factors into the machine learning algorithm, after tested, picking the important numbers as the evaluation parameters, thus, we optimize the parameters of evaluation function for checkers game. The second one is that, by analyzing the influence of removing the weighted numbers of the king of black and the king of red from the program, we obtain the conclusion that the two parameters have large influence, and cannot be removed from the evaluation function.**

Index Terms—**computer game; checkers; evaluation function**

I. Introduction

With the development of computer, games based on and derived from computer technology have made rapid progress, researchers on computer game system of various games have made quite fruitful results, and the related computer game systems become more and more perfect. A computer game system mainly includes four parts [1], that are the board representation, the moving generation, search algorithm, and the situation assess. The board representation describes the game situation, including the encoding and storage of the chessboard, chessman, obstacles, spaces, and chess going. The moving generator refers to generate the effective going according to the current situation. As the core part of the computer game, the search algorithm is to find the optimal moving, the common algorithms include the max-minimum algorithm, the α-β pruning algorithm and etc. The situation evaluation access the current game situation, provide the premise for the search algorithm, thus it is important to the performance of the whole system.

The game called as the Checkers in America, known as the Draughts by the English, is one old chess game, as far as the ancient Egyptian Pharaoh, and is mainly popularized in English speaking countries. In this game, pieces with darker color is the black side, the ones with lighter color represent the white side. The rule is that the side which eat up or blocked all the pieces of the opponent win the game, so the key factors for win is to carefully generate the attack or the defense moving. By now, the Checkers has already taken a relative important position in computer game, many a few related researches have been reported. In which, the researchers led by Jonathan Schaeffer of the University of Alberta have developed an invincible checkers program of 8×8 chessboard, name as Chinook[2-3]; Besides, the Checkers program of 10×10 chessboard is attracting more and more researches; Paper[4] proposed the representation of bit-board for the Checkers. Paper [5] proposed to use the genetic algorithm to solve the problem of game situation evaluation, the method take the six values, which are the pieces number of the black, the red, the king, and the one in danger, as the evaluation parameters, each assign one different weight, thus to form one linear evaluation function; the weights correspond to the chromosome string and are encoded wtih real numbers, the local climbing process of the genetic algorithm and the mathematical probability and statistics are used to filter the child generation, thus to realize the self evolution. After many a few times of evolution and selection, a new chromosome string is obtained, and a better evaluation function is formed. However, when the evaluation parameters of the two playing sides are not the same, if still taking method of [5], then the wining probability difference between the two sides will not be obvious.

This paper will study the parameters of evaluation function of the Checkers with a 10×10 chess board, on the foundation of paper [5], we propose to adopt the j48 machine learning algorithm on the Weka platform to generate the decision tree [6], and further analyze the implicated relationship among the parameters, and the influent degree of the parameters on the final result, thus to optimize the parameter selection, and to improve the performance of the evaluation function.

In section 2, we will summarize the rules of the Checkers.

II. Rules of the Checkers

The composition and rules of play checkers are that.

1)The Board. The board of the checkers is a 10 x 10 board with brown and light lattices, which represent the black and white sides respectively. The board locates in the middle of the two players. At the lower right corner of each player, there are white pieces. This is shown in Figure 1.

2) Pieces: Each of the two sides (the black and the white) have 20 flat and cylindrical pieces, and which side be the black will be decided by throw the coin.

3) Position: the black lattices are taken as the reasonable chess position.

4) Start: at the start, each side of the chess needs to put its pieces only in the first black rows near itself, as shown in Figure 2. And according to the rules, the black side take the first-hand, after that, each side move its piece in turn.

Li Shuqin, School of Computer Beijing Information Science& Technology University ,Beijing, P.R. China,
Xue Weiming, School of Computer Beijing Information Science& Technology University ,Beijing, P.R. China,
Yuan Xiaohua, Information college, Shanghai Oceanic University,Shanghai, P.R.China,

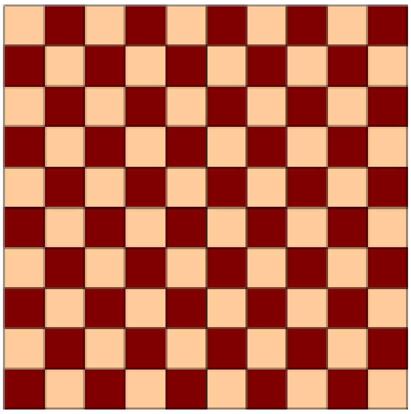


Fig.1. A 10*10 board of the Checkers

4) The start. ：at the start, both sides put their pieces only in the first 4 rows of black lattices of theriself, just as shown in Figure 2. And according to the rules, the black side is the first-hand, then both side move its piece in turn.

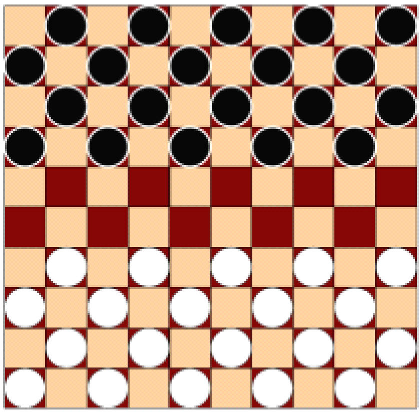


Fig.2. The layout at the Start

5) Target: during all the game, the pieces must stay in the black lattice and move along a diagonal line. And the target of one side in the game is either eat up all the pieces of the opponent, or make a situation in which the pieces of the opponent cannot move again.

6) Eating and jumping: as long as in the black lattices adjacent along the diagonal line there is piece of the opponent, and the lattice behind this lattices is empty, then the piece of own-side can eat that piece of the opponent, and jump to that space lattice behind.

7) If there is no eating and jumping, then the piece has to forward one lattice along the diagonal line.

8) Coronation. Any piece arriving and stay at the bottom line of the opponent will be coronated at once, and become a king from now on. At this time, there need to put another piece of own-side, thus to show its difference from other common piece.

9) Continuous Jumping. Many times of Eating and jumping can continuously take place. If the condition of continuous jumping is meet, then there must to do so. The Continuous Jumping can end only if the condition is not meet again, or the jumping piece is a uncoronated one and has arrived at the bottom line of the opponent.

10) The uncoronated pieces can only move forward, but when in Jumping and continuous Jumping, the piece can either move forward of backward, of the combination of these two.

11) Only piece stop at the bottom line of the opponent can be coronated. So, if in the Jumping one piece arrive at the bottom line and not stay there but leave from, then this piece cannot be coronated.

12) One king can jump any more empty lattices. And similarly, during the Jumping, a king can jump over any more empty lattices before or after one piece of the opponent, so a king is more powerful and more precious. Even so, a common piece can eat one king is at suitable conditions.

13) Only after one move ended, the eaten pieces can be removed from the board. The pieces that eaten but not removed can be not jumped through, so they can be used to form the barrier.

14) In a jumping, if there are more than one move can be selected, there must to select the move which will eat the most pieces of the opponent, if more than pieces or more than one lines can jumping and eat the same number of pieces, then, then move side can select any of the pieces or the lines.

15) During the game, the game can be draw by mutual consent. If one side refuses to draw, then it needs to win within the 40 moves after, or to show some obvious advantages than the opponent.

III. Design ideals of the evaluation parameters of the Checkers on Weka platform

For a computer game, the most important modules are the search and the evaluation. In brief, the search is equivalent to the pre-judgment in the real chess, while the situation assessment is to evaluate which side will be favored under current situation, this will determine how to play. Since in the Checkers, pieces are relative more, the rules are relative more flexible, so in a game, the search will not search through much more layers. For example, in the national computer game contest, the time of each game will not exceed 40 minutes, so the thinking time for each side will not exceed 20 minutes, and that for each move cannot exceed 30 seconds, thus the search depth is limited, which cannot be more than 20 layers. So for a program of the Checkers, its performance will largely depends on its evaluation module.

A chess evaluation function will usually considers the piece number, the force value of one go, the flexibility, the chess position, and the graph formed by the pieces, and etc. As to the Checkers, there are four kinds of pieces numbers, which are, the numbers of the white piece, the black pieces, the white king, the black king. There two kinds of force value, that are of the kings or the common pieces. There two kinds of flexibilities, that are, if can move, if will be eaten after moving. The two kinds of position are that, the number pieces at the bottom line, and that at the lateral line. And at last, there are two kinds of graphs, the number of pieces that form a diamond shape, or a direct line.

In this paper, we select 6 parameters for the evaluation function, which are, the number of the black pieces, of the red pieces, of the black king, of the red king, of the black in danger, of the red in danger. And adopt the genetic algorithm recommended in paper [5] to evaluate the game situation, and besides the genetic algorithm, we also use the J48 decision tree algorithm in Weka's classifier to analyze the parameters. J48 is implemented based on C4.5, which include a series of classification algorithms that used in machine learning and data mining. For a given dataset, in which each element can be described by a set of fields, and each element belong to one of the mutually exclusive classes, then as a supervised learning algorithm, the goal of C4.5 is that through learning, to find a mapping relationship from the attribute value to the class, and this mapping can be used to classify a new entity whose class is unknown before. The algorithms in C4.5 will form a decision tree through learning, and naturally, each layer of the tree represents one attribute value, and the last leave node give determined class. After the decision tree is generated, survey the tree from the top to the bottom and analyze, we can obtain the rules, and the implicit relationship among the parameters.

IV. Experiments and Analysis

Experiment 1: Classify the parameters in the evaluation function

We denote the six parameters which include the numbers of the black, the red, the black king, the red king, and the black and the red in danger, as blackcheckers, redcheckers, blackings, redKings, threatenedblack, threatenedred, respectively. After 100 games, we input the recorded data of the six parameters into the arff file, and load the file into Weka platform to analyze.

Usually the classifiers include decision tree, logistic regression, naive Bayesian, neural network, and etc., in this experiment, we adopt J48 algorithm, which is a decision tree, and for the test mode of J48, we selelct 10 set cross validation, that is, divide the test data into 10 set , each of the same size, and the test will iterate 10 times. In each iteration, 9 data set in turn will be used for training, and one set left for valid. The average correct rate of the 10 test is used to evaluate the algorithm, and the classified results stored as parameter blackwin(which is the seventh one). The process and results are shown in Figures from 3 to 5.

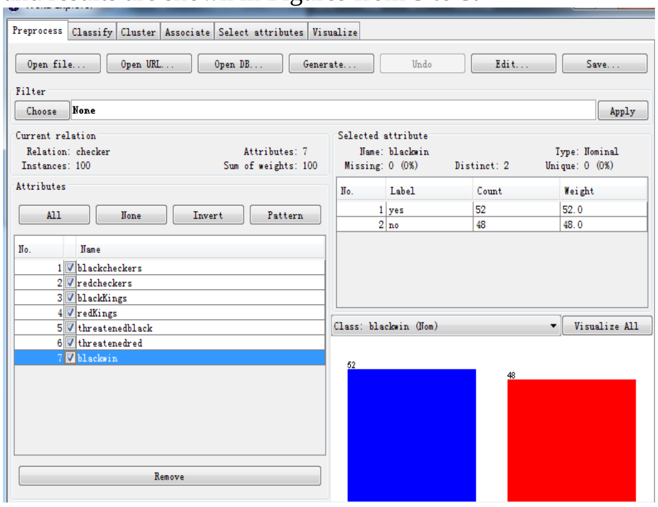


Fig. 3. The seven parameters selected for the classification

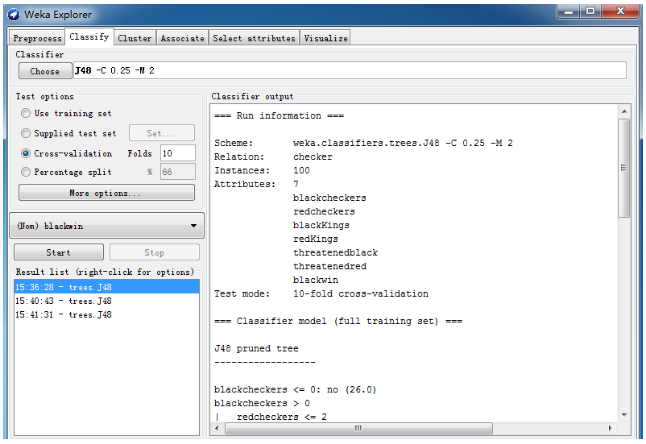
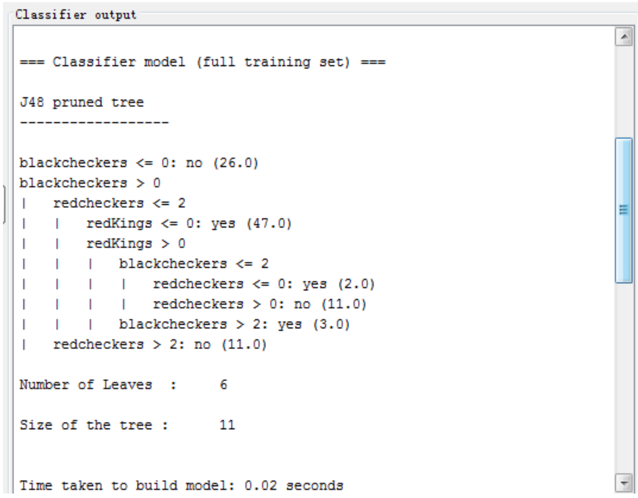
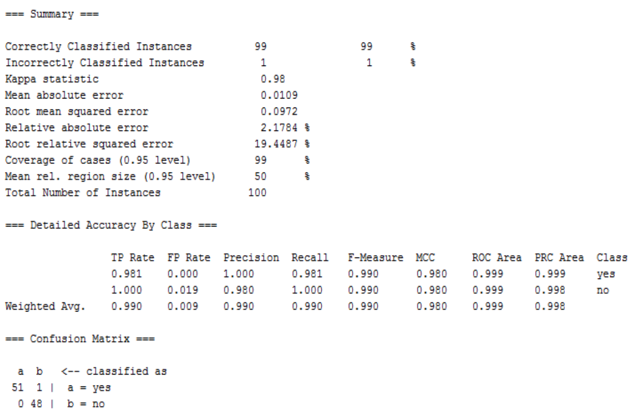


Fig. 4. The accuracy of the 10-times cross validation for the classification



```
Classifier output

=== Classifier model (full training set) ===

J48 pruned tree
------------------

blackcheckers <= 0: no (26.0)
blackcheckers > 0
|   redcheckers <= 2
|   |   redKings <= 0: yes (47.0)
|   |   redKings > 0
|   |   |   blackcheckers <= 2
|   |   |   |   redcheckers <= 0: yes (2.0)
|   |   |   |   redcheckers > 0: no (11.0)
|   |   |   blackcheckers > 2: yes (3.0)
|   redcheckers > 2: no (11.0)

Number of Leaves  :     6

Size of the tree :      11


Time taken to build model: 0.02 seconds
```

Fig. 5. The classification results by J48 on Weka platform



```
=== Summary ===

Correctly Classified Instances          99               99      %
Incorrectly Classified Instances         1                1      %
Kappa statistic                          0.98
Mean absolute error                      0.0109
Root mean squared error                  0.0972
Relative absolute error                  2.1784 %
Root relative squared error             19.4487 %
Coverage of cases (0.95 level)          99      %
Mean rel. region size (0.95 level)      50      %
Total Number of Instances              100

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.981    0.000    1.000      0.981    0.990      0.980    0.999     0.999     yes
               1.000    0.019    0.980      1.000    0.990      0.980    0.999     0.998     no
Weighted Avg.  0.990    0.009    0.990      0.990    0.990      0.980    0.999     0.998

=== Confusion Matrix ===

  a  b   <-- classified as
 51  1 |  a = yes
  0 48 |  b = no
```

Fig. 6. The final learning result

Just as shown in Figure 6, that after inputting the related six parameters of the chess board situations of the 100 game turns, and learned by the J48 algorithm in Weka platform, the classification correct rate is up to 99%, this can testify the accurate of the algorithm.

In order to clearly explain the results shown in Figure 6, we will display the related decision rules in a tree mode, just as shown in Figure 7.

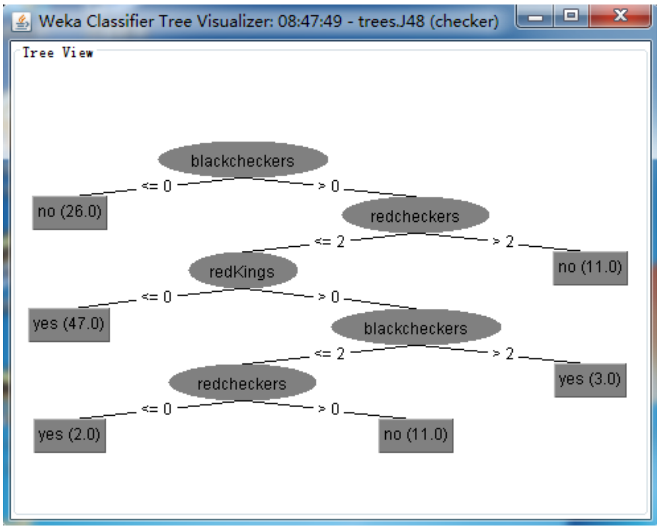


Fig.7. Tree mode of decision tree

From Figure 6 and Figure 7, we can clearly see the decision rate of each parameter, and we can obtain the rules as.

(1) When the final number of the black on the board is 0, then it is possible to lose;

(2) If the final number of the black on the board is not 0, the number of the red is equal or less that 2, and there is no red king, then there is half the possibility to win.

(3) On the final board, there is red king, and the number of the black is near 2, then the win possibility for the black is low.

Experiment 2: Select the important parameters in the evaluation function

After adopting the learning algorithm of J48 decision tree, we can well understand the relative relationship among the 6 parameters, and learn the key conditions which decide the win and defeat, thus help to efficiently make or avoid some special game situations. As to the six parameters, in order to decide if some ones are useless, in this experiment, we delete the related parameters, then perform the learning of J48 again, and compare the new correct rate with that using all the six parameters in the experiment 1. The results are shown in Figure 8.

```
Classifier output

Correctly Classified Instances          94               94      %
Incorrectly Classified Instances         6                6      %
Kappa statistic                          0.8794
Mean absolute error                      0.0707
Root mean squared error                  0.2248
Relative absolute error                 14.1465 %
Root relative squared error             44.9654 %
Coverage of cases (0.95 level)          97      %
Mean rel. region size (0.95 level)      59      %
Total Number of Instances              100
```

Fig.8. Learning result after removing parameters blackings and redkings

From Figure 8, we can see that, after removing the two parameters of blackings and redkings, the correct rate reduce to 94%, this is far less than that using all the six parameters as shown in Figure 6, so we can say that, the two parameters of blackings and redkings have large effect on the game, and cannot be removed from the evaluation function.

V. Conclusion

Artificial intelligence has become the key techniques of computer game, and as the perfectly complement of it, the techniques of machine learning and data wining can surely be used in computer game. By the machine learning algorithm on the Weka platform, we can more efficiently select out the parameters that can determine the win and lose, and obtain the parameters and their relationships, all these can help to pre-judge the game result, thus by the greatest extent to exert the effect of evaluation function, and to lay a good foundation for the development of computer game programming with higher intelligence.

Acknowledgment

This paper is jointly sponsored by Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research (ICDD201507). by the Education and Teaching Project for undergraduate scientific research plan of Education Committee of the Beijing Municipality. (PXM2015_014224_000050),and by Sensing & Computation Intelligence Joint Laboratory,and by The Project of Construction of Innovative Teams and Teacher Career Development for Universities and Colleges Under Beijing Municipality(IDHT20130519).

References

[1] Xiaochun Wang. PC game programming[M]. Chongqing: Chongqing University press, China. 2002:1-27.

[2] Jonathan Schaeffer, "Checkers Is Solved", SCIENCE, vol. 317, September, 2007

[3] Desheng Li(translated). Another leap in artificial intelligence [J]. World science. 2007, (10):31-32.

[4] Yue Wu, ShuQin Li. Research on the chessboard representations of checkers. 2013 International Forum on Computer and Information Technology (IFCIT 2013)

[5]]Su Zhe,Li shuqin.JiaYun Lu;,ZhengLi Wei,Fan Si Cong.The realizati on of genetic algorithm in terms of checkers evaluation function. Appl ied Mechanics and Materials, 1979-1985

[6] Meiyu Yuan(editor). Data mining and machine learning- Weka Application technology and Practice [M]. Beijing : Tsinghua University press, China, 2014-07-01.

[7] Chuan LI(translated). Practical machine learning tools and techniques for data mining [M]. The third version. Beijing: Machinery Industry Press, China, 2014.4.

Shuqin Li received a B.S. degree in computer Science from the Universityof ShanXi in 1985, and the M.S. andPh.D. degrees in Computer applicationfrom the Nanjing University of Science and Technology in 1988 and 2006. She has been with the Department of computer science at Beijing Information& Science Technology University, where she is a Professor. She is also a Technical Committee of China Computer Games Commission and a member of China Association for Artificial Intelligence (CAAI).Her major study is machine study and artificial intelligence.

Xue Weiming received a B.S. degree in computer Science from Beijing Information& Science Technology University. Now he is a Master student at Beijing Information& Science Technology University. His research interest is machine study and computer game.

Yuan Xiaohua received the B.S. degree from the South-East Normal University,Chongqing, in 1991, the M.S. degree from the Chinese Academy of Sciences,Beijing, in 1999, and the Ph.D.

degreefrom Nanjing University of Science andTechnology, Nanjing, in 2006. She iscurrently an associate professor in the Department of Information, Shanghai Ocean University. Her research interests include Image Processing and Artificial Intelligence..