ISSN: 0975-766X
CODEN: IJPTFI
Available Online through *Research Article*
www.ijptonline.com

USING MACHINE LEARNING IN PEG SOLITAIRE STRATEGIC GAME

Dhinesh Babu L.D., YashswiniVarangaonkar, Krishna Shah, NiravKagathara
School of Information Technology and Engineering, VIT University, Vellore, India
Email: lddhineshbabu@gmail.com

Received on 25-10-2016 *Accepted on 02-11-2016*

Abstract

Machine learning is an important research area of computer science with rapid development. This is due to the growth in the database industry, the advancement of data analysis, research and the resulting market needs for methods that are capable of extracting valuable knowledge from large data stores. Using the concept of Machine Learning, we have tried to apply it to a strategic game Peg Solitaire. We have used adecisiontree, backtracking and pagoda function after analysis. Patterns are discovered, and further moves are taken in the game. It shows that ML is a perfect approach for a strategic game.

Introduction

Artificial intelligence (AI) plays avital role in computer evolution. From past few decades, large research work is being done on AI and games. Automated programs now exist for games, which perform very well in classic board games such as Chess, Scrabble, Pentagon, Othello,etc. This research impacts about remarkable progress in machine learning methods, search algorithmsand computer hardware[1]. From past few years, complex strategy simulation games are being focused by AI researchers, which offers new challenges including asynchronous gameplay, partially observable environments, comparatively large decision spaces with changing abstraction levels, and also with the necessity of resource management processes[2]. Games with smaller decision spaces require less sophisticated representations and reasoning capabilities than large decision space games.[5] Most single-player/multi-player video games are widely distributed with inbuilt artificial intelligence player that permits humans to play against the computer. The task of building such players is complicated because the AI player has to be challenging, but the significant fact is that the game still has to be winnable by the player. Creating an AI system for a game require contributions from many different research areas like machine learning to produce a realistic system rather than hardware resources. It is used as a platform for the human level AI System[3]. "The Machine Learning field evolved

from the broad field of *Artificial Intelligence*, which aims to imitate intelligent abilities of humans by machines." The important issue in the field of *Machine Learning* is how to make machines able to "learn." Machine learning tasks are typically classified into three broad categories. Supervised Learning, Unsupervised Learning and Reinforcement Learning[4]. Here we are dealing with reinforcement learning. It is a computer program interacts with a dynamic environment in which it must perform a certain goal without precise information which shows is it close to its target or not[5]. Game playing is a leading application area of research in AI widespread and Machine learning specifically. So here, we are focusing on classic board game Pegsolitaire with applying the concept of machine learning.

Rules for the Game

Peg Solitaire game is a well-known puzzle, which can show difficult despite in its simplerules. Pegs are arranged on board; one `hole' remains which shows the initial state of the game[6]. By making strategic wise moves, pegs are gradually removed until no further moves are possible or some goal state is achieved. This article considers the English variant, consisting of a board in a cross shape with 33 holes(32 pegs).[7]

Pegsolitaire game is played on board, which is an n-by-m grid of holes, from which each hole can be either valid or invalid as shown in figure 1. A soundhole may contain a peg. If a hole contains a peg, then it is filled, and otherwise, it is empty. A configuration of a board depends on some subset of its filled holes. Thus, a board with H holes has2^H possible configurations.

In a peg solitaire game, a player changes the configuration from one board to another via jumps. A leap involves 3 (horizontally or vertically) consecutive holes, from which the first hole is known as the starting hole for the jump. To be more specific, for a given board, a jump can be represented as a triplet (p, q,dir), where p and q indicate the row and column on the board. The jump's starting hole (indexed from 0), and direction of the jump (either EAST, NORTH, SOUTH, or WEST).[6] A legal jump on board for a particular configuration is if the jump involves three consecutive holes in which the first and second holes are filled, and the third hole is empty. [9]

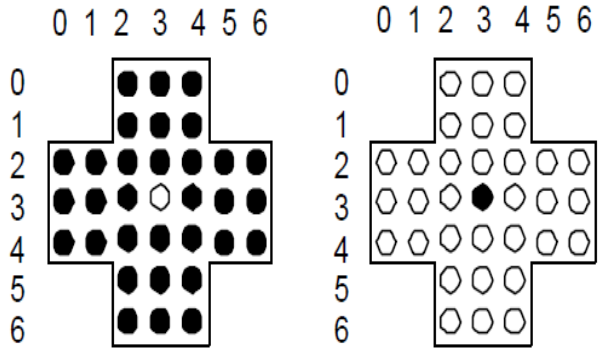


Figure 1: The Solitaire board and first and last states of central Solitaire.

Related Work

Nowadays, computer game and its automation is the most demanding field which requires advancement time to time. It is a stable area of application and thus may see a more intellectually deep impact from evaluated interface specifications[7]. Several exceptionalworksare available on machine learning and Reinforcement Learning (RL) in gaming.

Machine learning apparently RL in games is a popular area of application for AI research. RL has been used in a variety of different types of AI research bothpopular games and in video games. They describe the implementation of a simple, very specialized RL-based method. This method is Micro-managing combat, which focuses on the size of the state, space limits, and the efficiency of the learning agent[8].

Strategy games are an important and challenging subclass of

Video games. ML is significantly used in a turned based strategic game civilization IV. In this competition, the player has to choose a leader amongst given options. Each leader has this strategic play, ML provides a powerful tool which allows different strategy. The player has to take the decisions so as to win the game. He builds cities, train workers, and military units, and interact with other human or AI players. According to the situations, the player has to change his strategy and also leaders. The aim of these games is to make the best use of limited resources to defeat the opposing players. Here in this strategic play, ML provides a powerful tool which allows agents to adapt the strategies and takes actions on it quickly, even in critical scenarios when a player is about to lose the game. This paper used a Markov Decision Processes as the general framework[9].

To win a strategic game, what requires is a great reasoning capability, strong decision power, and techniques to overcome challenging states. This needs an algorithm that gives the optimal environment through experience. Here for a solution they have developed a lattice that is a kind of decision tree for representing and interacting abstract states in a real-time strategy game which can be used incorporated in many board games.

Case-Based Reasoning(CBR) is also a popular approach in strategic games[10]. Several decision techniques like Markov Decision Processes, integrated scheduling, have been implemented in many game's logic. CBR technique is used to select several action sequences. However, thisapproach limits the set of actions that can be chosen for a new state.

Pentagon is a board game which is deterministic, strategic, and unsolved[11]. Therefore it is suitable for Machine Learning. It explores the power of computing of machine learning algorithms. This game is similar to the Peg

solitaire game. The ability of a computer to solve the problems with different techniques like Brute Force approach is remarkable. Games like Pentagon often contain too many choices at each step. So it becomes a challenging job for ML to calculate the best move in a reasonable amount of time.

Peg Solitaire is a well-known puzzle which can prove difficult despite its simple rules. Modeling Peg Solitaire presents a considerable challenge. Jefferson and Terrific solve the Peg Solitaire using CP and OR techniques[12]. This problem is interesting as it is highly symmetric. AI planning apparently ML tackles its sequential nature more usually. It includes solving the problem with unknown parameters, It explores the unmatched ability of ML to develop a complex system which applies different decision strategies for a large variety of cases in the game.

Solving Game

In this game of peg solitaire to reach the completion of a game state, the player has to go through many rules, and for winning the game,he has to make some strategy. The initial state is a state which shows the primary stage of the game where all the pegs are at their places and just one hole is empty. A description of possible actions that canbe taken during the gameplayis stated with different data and rule runners. The set of measures is often a function of the current situation of a search agent, which is traversing the problem space for further moves. This is accomplished by a Decision tree. After removing the pegs one by one, many different possibilities will be there. Each position will be evaluated for further different moves. Various patterns can be possible for a single peg position. Few of them lead towards a solution. We check the moves given by player against the decision tree. Also, we are using the concept of backtracking here. Backtracking refers to go back to the previous state due to some undesirable conditions. The collection of state-action pairs is known as the state transition model. This field gives adescription about the success or failure of the game after each move. Last state or final state is a goal state which determines that only one peg will remain in the hole.

For solving this problem algorithmically and for the optimal solution, we are considering Pagoda Function. With the help of Pagoda function, we can get the solution with fewer moves. We are using this for guiding the players. In this, we give a value to each position on the board. The values are provided in a linear way such that three positions x, y, z are adjacent and in one line. The condition x+y>=z should be satisfied.[12] The pagoda value is calculated for each function at the given state. This is provided by adding the values of the positions on the board(x, y, z) where there is a peg. One more condition is the middle value must be greater than the third value i.e.,.Y>z. We know that the moves are like if x, y, z are in a line then to make a move, z should be empty and x Andy should not. Peg at position x is

removed and placed at position z. this shows that value of Pagoda function will never increase.[15]There are infinitely many pagoda functions exist in the algorithm. Here, the bulk of experimental evaluation is concerned with the Solitaire reversals. That is, beginning with a particular single hole position on the board and ending with only a single peg at the same situation.[4]

Peg Solitaire game is a deterministic game so, after applying algorithm, we can say that player will win or lose the game.

Architecture and Algorithm

Here we have proposed an algorithm for our system which uses concepts of the decision tree, backtracking and pruning approach to solving a peg solitaire problem. This algorithm searches for a solution from a given start configuration by making a legal jump J from theoutset, after which it recursively searches for a solution from the new configuration. The peg makes a move when a legal jump is possible otherwise it backtracks. Since the number of pegs (filled holes) is reduced after every jump on the board by one, the algorithm stops recursing. At that time, both start and final have the same number of pegs. We develop a decision tree according to it and prune it for a better search space. This tree is stored in the repository.

Find Solution:

Findsolution(Start,Final,Path)

1 .if start.numbpeg<= final.numbpeg

Return(Start=Final)

2 .else

Jumpj(a*b*{N,E,S,W})

3 .if j is a legal jump for start

Then makeMove(j)

Path.push(j)

Else

Illegal jump

4 .makeReverseMove(j)

5 .repeat step 3,4untill

Iffinal.pegposition=Inverse(start.pegposition)

Return true

State Win

Else

Return false

State Loose

Here we are proposing a general architecture for strategic rule based games which later we used as a reference for our architecture.

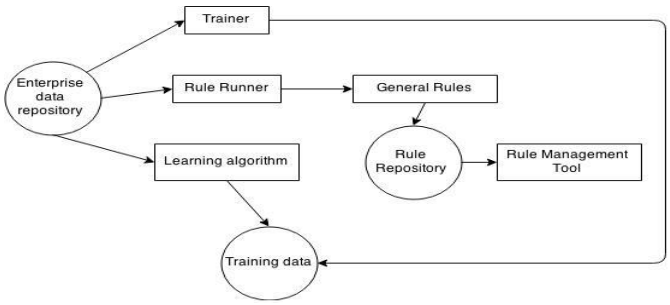


Fig. 3 General gaming architecture.

This is a most widely used architecture for games. It starts with a data repository where all the data like game implementation programs, systems libraries, and all the informationis stored here. It has a training session for the new players under Trainer field. This will give trial games for training sessions for new players. A rule runner is provided for keeping track of standards in the games. For every move, the rules are considered. If anything goes wrong, the rule runner checks the rules against the rule repository. This rule repository contains all the rules of the game. Next to it is an algorithm for asetto be executed. Also, rule management tool is there for the generated rules. This is a general game architecture for any strategic play. As for our Peg Solitaire game we have given the following architecture.[9]

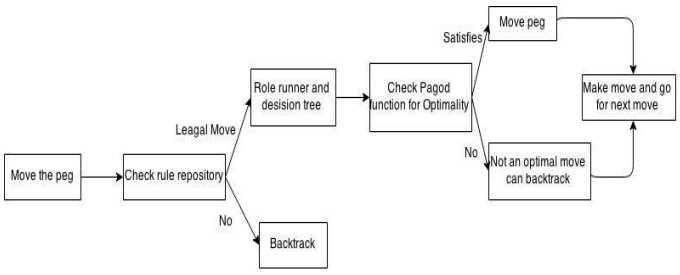


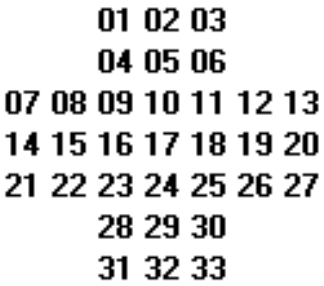
Fig 4: Architecture for peg solitaire.

Our system will work for a move as given in this architecture. A sequence of actions is taken after making a move. The user is unknown of it. Here machine learning works. The reinforcement algorithm is carried out here. According to it

A peg is moved. The rule runner will check whether it is a valid move or not. If it is a valid move then it will go to next step otherwise, that move is backtrackedi.e., that move is rejected. If the move is valid, then it is given to rule runner. Rule runner at first check the move against the decision tree and then it is checked for the Pagoda function as discussed above. If it satisfies both, then the move is an optimal move and can lead to a successful game. If not then it is not an optimalmove, and it may or may not take you to a successful play. We can backtrack from such move. At times decisions are made by the player, the algorithm is modified for the specific move as per Machine Learning. Machine Learning can be summed up with Representation, Evaluation, and Optimization. At first, the move is represented then evaluated with a decision tree and then optimized with Pagoda function.

V. Possible Solutions

Weplay this game without considering any strategy, so we can not repeat that moves later. By seeing this games its looks like very easy to play but considering all rules it is it is a very hard strategic play.[14]



Solution1:

15--17, 28--16, 21--23, 07--21, 16--28, 31--23, 24--22, 21--23, 26--24, 23--25, 32—24--26, 33--25, 26--24, 12--26, 27--25, 13--27, 24--26, 27--25, 10--12, 25--11, 12--10, 03--11, 10--12, 08--10, 01--09--11,02--10,17--05,12--10, 05--17.

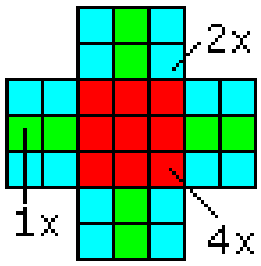
It is a 33peg solitaire game, which is having 32 pegs so we need 31 jumps to reach its solution because we must have to remove 31 pegs. It also counts double jumps as single jump like (32--24--26), so it is possible to have less than correct 31 moves. The board is symmetrical vertically, horizontally or diagonally, therefore, there are more solutions possible.

Solution2**:**

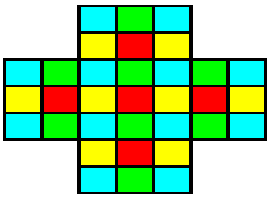
Smallest solution of this problem is with 18 moves which made a World Record.

15--17, 28--16, 21--23, 24--22, 26--24, 33--25, 18--30, 31--33--25, 09--23, 01--09, 06--18--30--28--16--04, 07--21--23--25, 13--11, 10--12, 27--13--11, 03--01--09, 08--10--12--26--24--10, 05--17. [16]

Peg Mobility:

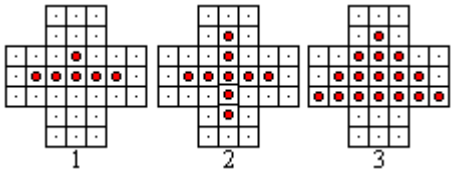


As shown in figure(3 colored class classification) if thepeg is at any position in the red part,jumps are possible in all directions. Two possible jumps can be from the blue field, one possible jump from the green field. There are 76 jumps may possible together.



As shown in figure(4 colored class classification) also defined. In which after completion of any move peg will always remainin the same colored class.

Rule Runner for beginners:



Most frequently repeated patterns are as shown in thefigure. In all cases, only one peg must leave over. These shown problems are very useful for beginners to learn this game.

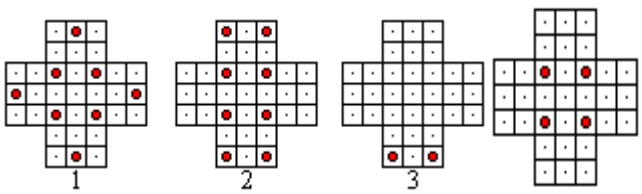
Patter Solutions:

1) Submarine: 10--24, 15--17, 24--10, 19--17, 10--24.

2) Greek Cross: 10--02, 24--10, 19--17--05, 02--10, 15--17, 10--24, 29--17.

3) Pyramid-16: 18--06--04, 26--12, 09--11, 12--10, 24--26, 27--25, 23--09, 04--16, 10--24, 25--23, 16--28, 21--23, 28--16, 15--17.

Dead End Patterns:



Game starting with 33 pegs and on the way to completion of this game we may get these types of patterns which lead to the dead end of problem solution.

Solutions:

2 squares: 19--17, 16--18, 14--16, 05--17--15, 04--16--14, 29--17, 18--16, 06--18, 07--09, 13--11, 18--06, 16--04, 01--09, 03--11, 30--18, 28--16, 21--23, 27--25, 16--28, 18--30, 33--25, 31--23.

Avenue: 19--17, 30--18, 17--19, 20--18, 27--25, 15--17, 04--16, 17--15, 07--09, 14--16, 06--04, 18--06, 03--11, 01--03, 28--30, 16--28, 25--23, 33--25, 31--33, 23--31, 21--23, 09--01, 11--09, 13--11.

Pair: 19--17, 30--18, 17--19, 06--18, 13--11, 27--13, 26--12, 18--06, 03--11, 01--03, 04--06, 16--04, 11--09, 13--11, 08--10, 11--09, 03--11, 22--08--10, 11--09, 04--16, 24--22, 21--23, 07--21, 32--24--22, 31--23, 16--28, 21--23--31.

4 squares: 29--17, 26--24, 17--29, 32--24, 23--25, 31-23, 22-24, 25--23, 33--25, 08--22--24, 10--08, 07--09, 21--07, 02--10--08, 07--09, 12--10, 03--11, 24--26, 27--25, 19--17, 10--24, 13--27, 24--26, 27--25, 09--23, 01--09.

Worst(Shortest) possible pattern:

It will reach toa dead end within only six moves.

Pattern:4--16,23--9,14--16,17--15,19--17,31--23

Conclusion

Machine learning is an important research area of computer science. It has a rapid development. This is due to the growth in the database industry, the advancement in data analysis and research and the resulting market needs for methods that are capable of extracting valuable knowledge from large data stores. With the concept of ML, we have tried to solve a strategic game Peg Solitaire. We have used a decision tree, backtracking and pagoda function after analysis. Patterns are discovered,andfurther moves are taken in the match. It shows that ML is a perfect approach for strategic play.

References

1. M. D. Feith and F. Pasiouras, “Assessing bank efficiency and performance with operational research and artificial intelligence techniques: A survey,” *Eur. J. Oper. Res.*, vol. 204, no. 2, pp. 189–198, 2010.
2. M. A. Huselid, “The impact of human resource management practices on turnover, productivity, and corporate financial performance,” *Acad. Manag. J.*, vol. 38, no. 3, pp. 635–672, 1995.
3. R. Dodhiawala, N. Sridharan, P. Raulefs, and C. Pickering, “Real-Time AI Systems: A Definition and An Architecture.,” *I can*, pp. 256–261, 1989.

4. L. D. D. Babu and P. V. Krishna, "An execution environment oriented approach for scheduling dependent tasks of cloud computing workflows," *Int. J. Cloud Comput.*, vol. 3, no. 2, pp. 209–224, 2014.
5. E. D. Raj and L. D. D. Babu, "A fuzzy adaptive resonance theory inspired overlapping community detection method for online social networks," *Knowledge-Based Syst.*, vol. 113, pp. 75–87, 2016.
6. C. Jefferson, A. Miguel, I. Miguel, and S. A. Tarim, "Modelling and solving English Peg Solitaire," *Comput. Oper. Res.*, vol. 33, no. 10, pp. 2935–2959, 2006.
7. J. P. Nivash, E. Deni Raj, L. D. D. Babu, and M. Nirmala, "A neural network based framework for apache YARN," *2014 Int. Conf. Inf. Commun. Embed. Syst. ICICES 2014*, vol. 2, no. 978, 2015.
8. E. D. Raj and L. D. Dhinesh Babu, "A firefly swarm approach for establishing new connections in social networks based on big data analytics," *Int. J. Commun. Networks Distrib. Syst.*, vol. 15, no. 2–3, pp. 130–148, 2015.
9. "<Markov Decision Processes.pdf>." .
10. A. A. & P. E, "Case-Based Reasoning: Foundational Issues, Methodological\rVariations and System Approaches. ," *AICom- Artif. Intell. Commun.*, vol. 7(1), no. IOS Press, pp. 39–59, 1994.
11. J. Méhat and T. Cazenave, "A Parallel General Game Player," *Künstliche Intelligenz*, vol. 25, no. 1, pp. 43–47, 2011.
12. B. M. Smith, "Caching Search States in Permutation Problems," *Proc. 11th Int. Conf. Princ. Pract. Constraint Program. - CP 2005*, pp. 637–651, 2005.