

CS181 Artificial Intelligence I:

Final Project : A Minesweeper Using Inference, CSP and CNN

Chenyang Zhang, Chunxu Guo, Jiahao Huang, Qianyu Liu, Tianyi Zhang

.@shanghaitech.edu.cn

Abstract

Minesweeper is a single-player puzzle computer video game written by Robert Donner and Curt Johnson which include in Microsoft Windows® in 1991, whose goal is to clear a rectangular board containing hidden 'mines' or bombs without detonating any of them. This game originates from 1960s and has derived different variants on many computing platforms. In fact, Minesweeper is proved to be Turning Complete and in a class of mathematically difficult problems known as co-NP-complete as well. Exploring algorithms to solve Minesweeper game may inspire us to work out other related problems.

1. Introduction

Minesweeper game is basically defined in a customized square grid board with a certain number of mined. The most classical and standard size of mine board is 9×9 (beginner), 16×16 (intermediate), 16×30 (advanced), and the standard mine density of advanced one is 20.63

There are 2 main competitive direction for Minesweeper players, one is speed and the other one is correctness. Algorithm model in this report is tend to revealing safe cells as much as possible without limited time. As it is proved to be an NP-hard problem in 2000, no such algorithm could solve it in linear time complexity so far. Besides, Minesweeper game might sometimes become a non-deterministic problem, which is roughly regarded as a classical models of probability now. (Suppose that all the mine matrices are randomly generated in the beginning, satisfying the uniform distribution.) Under this circumstance, neither people and computer could definitely reveal a risk-free cell.

Playing Minesweeper game requires logic, arithmetic and probabilistic reasoning knowledge based on spatial relationships on the mine board. Up to now, multiple artificial Intelligent algorithms have been proposed to solve minesweeper game, such as Inference, Constraint Satisfaction Problem (CSP) with backtracking, Markov Decision Process (MDP) or Partially Observable Markov Decision Process (POMDP), SAT solver, neural network and

so on. Considering their performance as well as related with course CS181, we design an identical algorithm model combining logic, CSP and convolution neural network (CNN). The most complicated case among our test condition 16×30 with 20.83% mine ratio.

2. Related works

pass

3. Basic Idea for Minesweeper

In a Minesweeper game, players will be given a certain sized grid board and the number M of unknown mines. Initially, all the sub-squares or the cells are blank, hidden among them are M mines distributed uniformly at random. To clarify the subsequent description, we raise some useful concepts here.

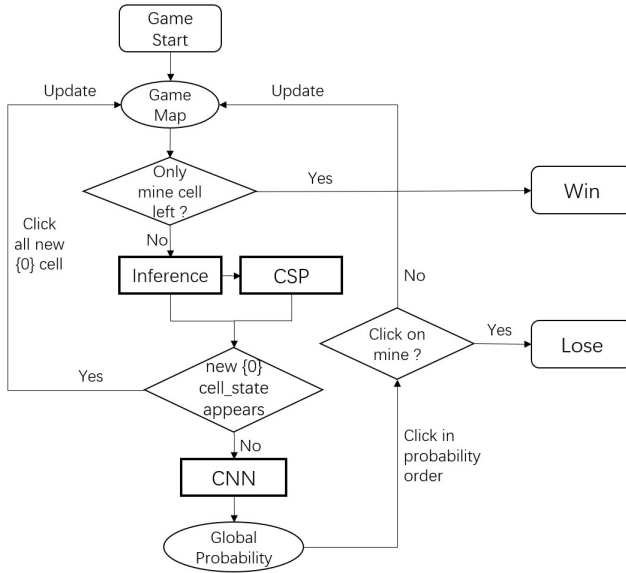
- **Neighbors**: unrevealed one in 8 surrounding blocks are all the neighbors of a given block.
- **Set** : A group of unmarked neighbors around a certain number block.
- **Game Map** : Real world interaction graph, a visual mine board with cells blank or filled with integer n . n varies from 0 to 8.
- **Mine Map** : Grand truth, the real mine distribution that we do not know.
- **Cell State** : At the beginning of game, all cells are signed with a initial state as 1,0, which means not sure about whether a cell has a mine. If a cell has been inferred as safe definitely, update the cell state as 0. Else if a cell has been inferred with mine definitely, update the cell state as 1. Otherwise stick to the original cell state.
- **Transferred Map** : A local graph used in CSP phase, each cell in this graph are related to a cell state.
- **Turn** : Every time the player operates once is a turn.

At each turn, a real world player can select three possible actions: to flag a cell as mine, to unmark a flagged cell, and to reveal a cell. But for AI robots, it would only maintain the transformed map corresponding to game map and click on safe cell with cell state = 0, unless no such cell is definitely safe. Under that uncertain circumstance, clicking depends on the descending order of global probability.

By clicking, player will safely reward a number behind (sometimes a large area will also be revealed fortunately) or lose the game immediately after detonating a mine. Until all the safe cells are revealed our lucky player wins.

4. Our Strategy

The main pipeline of our algorithm model in this report is shown below (Figure 1), involving junior and senior logic reasoning method and probabilistic calculation.



(a) Figure 1

Junior logic reasoning, the fastest and simplest cell assignment stage in our model, which is placed at the most front, only uses the adjacent 3×3 patches' constraints to figure out deterministic safe and mine-behind cells. Over half of situation could be solved after this stage. While sometimes it is not enough only considering adjacent information, so that a senior logic reasoning method called CSP followed. The magic power of CSP is embodied in cell constraint merge and division. In other words it create some new constraints on nonfixed shaped patches, and advancedly compare all the patch pairs instead of adjacent ones. The complexity of this method grows almost as exponential as the scale of the mine board due to the rapid in-

crease of constraints space. Backtracking may help us pruning a lot but it is still much slower than simple inference. CSP is also used to figure out deterministic safe and mine-behind cells.

Either after Inference or CSP stage, if AI robot could click on some safe cells, Game Map and Transferred Map will be maintained and reveal some new mine information which could be used for constraint update. Inference and CSP are able to run as far as possible with such regenerative feedback. Unavoidably, sometimes mine situation will fall into a 'dead corner' and bring about a 'guess' turn. None of the remaining unclicked cells can be definitely assigned with a deterministic state, and the only choice left is to click on the most likely cells without mines. CNN, the last module of our workflow in Figure 1 helps us to build a global probability table. Fine grid kernel helps to worked out neighbors contribution, while the coarse grid kernel provides a greater field of reception, combination of two of them enables local constraint relationships to transform into global probabilities.

4.1. Inference

Inference is the most basic step in our algorithm applying 4 basic rules as followed:

- Theorem 1 : If $\#(\text{items})$ in set is equal to number on block minus mine neighbours, then all the blocks in set can be marked as mines
- Theorem 2 : For 2 number blocks **a** and **b**, suppose **A** and **B** are their corresponding block set, if $A \subseteq B$ and number on **a** minus $\#(\text{marked neighbours})$ is equal to number on **b** minus $\#(\text{marked neighbours})$, then blocks in complement of **A** refer to **B** can be all revealed safely.

• eliminate

• confirm

4.2. Constraint Satisfaction Problem (CSP) with backtracking

pass

4.3. Convolutional Neural Networks (CNN)

pass

5. Experiment and Result

Approach	9x9 Map, 10 Mines	16x16 Map, 40 Mines
Random pick	0%	0%
Logic	70.7%	31.4%
CSP	45%	1%
Net	32.8%	0%
Logic + CSP	84.5%	48.7%
CSP + Net	77%	5.8%
Logic + CSP + Net	95%	63.8%

(b) Figure 2

6. Discussion

References

- [1] Dvaidd J Becerra. Algorithm approaches to playing minesweeper. *Harvard College*, 2015.
- [2] Mordechai Ben-Ari. Miniesweeper is np-complete. *Weizmann Institute of Science, Department of Science Teaching*, 2018.