Suyu Huang, Cong Deng, Bowei Feng

CS520 INTRO TO AI

26 October 2018

MineSweeper

Representation:

We create a two-dimensional array to represent the board, for each element, -1 means bomb, and other values mean the number of bombs around its 8 elements, and we have three level board:

- 9 * 9 with 10 mines
- 16 * 16 with 40 mines
- 30 * 16 with 99 mines

```
========== Real Matrix ===============
  0  0  0  0  1  *  2  *  3  2  2  *  2  *  2  *  2  2  2  1  0  1  *  1  1  2  *  *  1  0
  0  0  0  0  1  1  2  2  *  *  4  3  3  3  3  3  4  *  *  2  1  1  1  1  1  *  4  3  2  0
  0  1  1  1  0  0  1  2  4  *  *  2  *  2  *  2  *  *  5  *  3  1  1  0  2  2  3  *  1  0
  1  2  *  1  0  0  1  *  2  3  4  4  2  3  2  3  2  2  3  *  3  *  1  1  3  *  3  1  2  1
  *  2  1  1  0  0  1  1  1  1  *  *  2  2  *  2  1  1  1  2  3  3  2  2  *  *  3  0  1  *
  1  1  0  1  1  1  0  0  0  1  2  2  2  *  2  2  *  1  0  1  *  3  *  3  4  *  2  0  1  1
  0  0  0  2  *  3  1  1  0  1  1  2  2  2  1  1  2  2  1  1  2  *  3  *  2  1  1  0  0  0
  0  0  1  3  *  3  *  1  0  1  *  3  *  2  1  0  1  *  2  1  1  1  2  1  1  0  0  0  0  0
  0  0  1  *  4  4  2  1  0  1  2  *  3  *  1  1  3  4  *  3  2  1  0  0  0  0  0  0  0  0
  1  1  1  2  *  *  1  1  1  2  2  2  2  1  2  2  *  *  4  *  *  1  1  1  2  1  1  0  0  0
  *  2  1  2  2  2  1  1  *  2  *  1  0  1  2  *  3  3  *  3  2  1  2  *  3  *  1  0  1  1
  1  3  *  3  1  0  1  2  2  2  1  1  1  2  *  3  3  3  2  2  1  1  2  *  5  3  3  1  2  *
  1  3  *  *  1  0  2  *  2  0  0  0  1  *  2  2  *  *  1  1  *  1  2  3  *  *  2  *  2  1
  2  *  4  3  2  1  2  *  2  0  0  0  1  1  1  1  3  4  3  3  2  2  2  *  4  2  3  2  2  0
  2  *  2  2  *  2  1  2  2  1  0  0  1  1  1  1  2  *  *  2  *  1  2  *  2  1  2  *  2  1
  1  1  1  2  *  2  0  1  *  1  0  0  1  *  1  1  *  3  2  2  1  1  1  1  1  1  *  2  2  *
==================================
```

For robot, we create a two-dimensional array to represent its current knowledge about the the board, -1 means mines, infinite number means unsure square and other values mean the

number of mines around its 8 elements. **we replace -1 with "*" and infinite number with "?" when showing the matrix**)\

```
==========  Knowledge Matrix ============
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  2  2  2  1  0  1  *  1  1  2  *  *  1  0
  ?  ?  ?  ?  ?  ?  2  ?  ?  ?  ?  ?  ?  ?  ?  4  *  *  2  1  1  1  1  1  *  4  3  2  0
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  *  5  *  3  1  1  0  2  2  3  *  1  0
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  2  3  *  3  *  1  1  3  *  3  1  2  1
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  1  1  2  3  3  2  2  *  *  3  0  1  *
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  1  0  1  *  3  *  3  4  *  2  0  1  1
  ?  ?  ?  ?  ?  ?  ?  1  ?  ?  ?  ?  ?  ?  ?  ?  ?  2  1  1  2  *  3  *  2  1  1  0  0  0
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  *  2  1  1  1  2  1  1  0  0  0  0  0
  ?  ?  ?  ?  ?  ?  2  ?  ?  1  ?  ?  ?  ?  ?  ?  ?  4  *  3  2  1  0  0  0  0  0  0  0  0
  ?  ?  ?  ?  ?  ?  ?  ?  1  ?  ?  ?  ?  ?  ?  ?  ?  ?  4  *  *  1  1  1  2  1  1  0  0  0
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  *  3  2  1  2  *  3  *  1  0  1  1
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  3  2  2  1  1  2  *  5  3  3  1  2  *
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  1  1  *  1  2  3  *  *  2  *  2  1
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  3  3  2  2  2  *  4  2  3  2  2  0
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  *  *  2  *  1  2  *  2  1  2  *  2  1
  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  3  2  2  1  1  1  1  1  1  *  2  2  *
=======================
```

Probability Algorithm: For each cell whose value is confirmed, we know the number of bombs and we find the block_white and block_banner, we could get each cells surrounding a probability to be a bomb.

Inference:

When we collect a new clue:

In probability algorithm, if it is a bomb, we will find all revealed cells surrounding the bomb, and update these cells' clue since we find one more bomb which make clue clearer; if it is a value, we would update surrounding uncovered cells' probability. The program could deduce everything it can from a given clue before continuing since each time we will calculate all revealed clue.

Decisions:

We use two ALGORITHMs to make decision.

Gaussian elimination algorithm

Strategy 1: it is a straightforward strategy. When robot uncovers a square, it will calculate the number of surrounding covered squares and then consider following case:

- If robot uncover a square which returns 0, then all surrounding squares must be clear. Then robot will push those safe squares into a list for exploring later and update its knowledge matrix.
- For each square, if the number of remaining mines = the number of surrounding covered squares + marked mines, each one of covered mines must be a mine. Then robot will update its knowledge matrix with -1.
- For each square, if the number of surrounding mines = the number of marked mines, all remaining surrounding squares must be clear. Then robot will push those safe squares into a list for exploring later.

Strategy 2: we use gaussian elimination to combine more information in order to solve problem. For each uncovered squares which has value from 1 to 8, we can use this clue to generate a matrix row vector to represent the adjacent uncovered squares and remaining mines. After gaussian elimination,for each matrix row, we calculate the number of positive values and the number of negative values in coefficient matrix. If the number of positive values is equal to the the remaining mines, then all uncovered squares corresponding to the positive value must be mines and all uncovered squares corresponding to the negative value must be clear. If the number of negative values is equal to the the remaining mines, then all uncovered squares corresponding to the positive value must be safe and all uncovered squares corresponding to the negative value must be mine.

Probability Algorithm

Each time we we would choose the cell which has least probability to be bomb. And there is a risk that we may have several cells with same probability, and at this time, we would choose a random cell to cover.

Performance:

For probability algorithm, since there may be several cells with same probability, program could not make sure which could be cover at first and would choose a cell randomly, although these cells have same probability, for people, some cells are easier to make sure it is not a bomb. For some time, for people, some cells are hard to make sure it is not a bomb, program could give us a surprise decision. Program could only follow the probability which sometimes shows a surprising decision but sometimes we could not agree with.

Performance:

Based on the 16 * 30 board:

Gaussian elimination algorithm

When there are 120(0.25) mines, the success rate is nearly 1.8%, since with the increment of mines, it is more difficult for program to determine a certain cell and it would choose a random cell to reveal, and the number of mine is large which means it is more possible to reveal a mine.

Probability Algorithm

When there are 150(0.31) mines, the success rate is nearly 1.2%, since with the increment of mines, the value of each revealed cell is larger than less number mines, which means the probability of each uncovered cell is not far away, even though we still choose the least probability cell, the probability of failure increased.

Improvements:

The number of the rest mines sometimes will give us important information to make a decision in some ambiguous conditions. A lower right corner of a board is shown below as an example.

```
==========  Knowledge Matrix ===========
  3  2  2  1
  *  *  3  1
  *  *  ?  ?
  2  3  ?  ?
============================
```

Initially, there are four possibilities for the four unknown squares:

*	1	4	*	5	*	*	2
2	0	*	0	*	*	3	*

Absolutely, by applying the algorithm like CSP, it has to make a guess that which one is clear or not. But if algorithm could know how many mines are remaining, this complicated situation would become much easier. We can infer from the four possibilities that when number of the rest mines equals to 1 or 3, there is only one situation fits the number and we can make it. It will still be ambiguous when number of the rest mines equals to 2, but win rate will increase from 25% to 50%. So it also helps improve.

In general, algorithm has a 50% chance of failing in this ambiguous situation, which is almost inevitable on expert difficulty. But human could do better in some situations like this (and in other situations even human being can only make a guess and pray) because we are able to use the information of number of mines. It's not too hard to apply it in algorithm. This method will improve the successful rate in this kind of situation.

Bonus 4:

1.Based on your model and implementation, how can you characterize and build this chain of influence? Hint: What are some 'intermediate' facts along the chain of influence?

For each clue square, we build a matrix row vector to represent the adjacent uncovered squares and remaining mines. Every new cell will generate a matrix row vector to the entire matrix. When robot use gaussian elimination to solve this matrix, it combine all information that it can get from its knowledge matrix.

2.What influences or controls the length of the longest chain of influence when solving a certain board?

The size of board determines the dimension of matrix that will be executed gaussian elimination. When doing gaussian elimination. The length of the longest chain of influence will be four elements in a row vector of the matrix, since the maximum number of adjacency squares shared by two squares will be four and when two squares share same adjacency squares, it will be eliminated by gaussian elimination.

3.How does the length of the chain of influence influence the efficiency of your solver?

Suppose we have x*y board, then we need create a (x*y) * (x*y + 1) matrix. Suppose we uncovered a cell at (i, j), then it creates a row vector at row i * y + j. If the board is large, then we

need to create a larger matrix for gaussian elimination, which increases time complexity and space complexity, reducing the efficiency of my solver.

4.Experiment. Can you find a board that yields particularly long chains of influence? How does this vary with the total number of mines?

Since we build a matrix to represent the chain of influence and use gaussian elimination to solve it, the dimension of this matrix is determined by the size of board. Even the total number of mines is small, the robot still need to create a large matrix if the board is large.

5.Experiment. Spatially, how far can the influence of a given cell travel?

Since we build a matrix to represent the chain of influence, considering that when we use the gaussian elimination, if a element can be eliminated, it happens when it has the same column as the other one. Back to the question that how far can the influence of a given cell travel, if two cells share same adjacency uncovered squares, then the chain of influence will take place.

6.Is solving minesweeper hard?

It depends on the size of board and the number of mines.

For beginner, 9*9 board and expected 10 mines(we use probability to generate mines):

Play games 1000 times, it solves game successfully 779 times, which is 77.9%

For intermediate, 16*30 board and expected 40 mines:

Play games 1000 times, it solves game successfully 575 times, which is 57.5%

For expert, 16*30 board and expected 99 mines:

Play games 1000 times, it solves game successfully 178 times, which is 17.8%

Bonus 5

1.When a cell is selected to be uncovered, if the cell is 'clear' you only reveal a clue about the surrounding cells with some probability.

When we cover a cell which does not reveal a clue, we get the number of mines left, and the number of block_white left, calculate the probability of each block_white being a mine. At same time, we go through the surrounding cells, and get the number of block_white, block_banner, we use lower bound of block_white * probability + block_banner as the value of the cell and continue.

2.When a cell is selected to be uncovered, the revealed clue is less than or equal to the true number of surrounding mines (chosen uniformly at random).

Since we have applied Gaussian Elimination to determine the mines based on the accurate information of the surrounding mines. When the revealed clue is less than or equal to the true number of surrounding mines, we will get linear inequality equations.By applying a positive constraint variable M to the linear inequality equations, we can transfer it to linear equality equations and then use Gaussian Elimination to solve it.

3.When a cell is selected to be uncovered, the revealed clue is greater than or equal to the true number of surrounding mines (chosen uniformly at random).

Similar with the method we apply in previous problem, we need to use a positive constraint variable M then apply $-M$ to transfer it to linear equality equations and then use Gaussian Elimination to solve it.