CS 440: Assignment 2.5 - Decision Making

16:198:440

This project is intended to be an extension of the Minesweeper project in Assignment 2. At this point, you should have an inference engine capable of taking clues from a minesweeper board, and performing inferences to determine information about the remaining squares on the board. (For the purpose of this assignment, we assume the total number of mines is unknown.) When there was nothing more that could be inferred, or there were no squares that were obviously safe to collect, the agent you built should've selected randomly from the available squares. For this assignment, we want to consider the problem of how to select the next square to click more intelligently - what is the best place to click next?

To consider this problem, we first need to devise an operational definition of 'best' - what are we actually hoping to accomplish? Using the paradigm from the first assignment that clicking on a mine doesn't end the game (it simply incurs a one point cost), we could quantify best in at least one of two ways:

- **Minimizing Cost:** The best choice of square is the one that minimizes the expected number of mines stepped on.
- **Minimizing Risk:** The best choice of square is the one that minimizes the expected number of squares you have to unknowingly step on.

In the first case, this leads to a general trend of avoiding squares that are likely to be mines. In the second case, this leads to a general trend of stepping on squares that provide a lot of information about the future. In both cases though, we are dealing with decisions about an uncertain future, which will need to be quantified with probability.

Assessing the Future

Given a partially revealed minesweeper board, the previous project focused on figuring out if any unrevealed squares were mines or safe. In this case however, we need to consider *how likely a square is to be a mine or safe.* To do this, we need to be able to assess the probability of a given square being a mine. This is easy to do wrong.

Consider the minefield in the following example, with the clues revealed and the unknown squares given by the variables A, B, C, D, E, F:

```
[2][A][1]
[B][C][D]
[E][3][F]
```

In the previous project, you implemented a system that should be able to determine that B is a mine and D is safe, but the rest of the board is undetermined. What, then, is the likelihood that A is a mine? Or that E is safe? It's tempting to argue that the probability that A is a mine is $2/3$, based on the clue-value of 2 and there being three adjacent squares A, B, C. However, you would then also argue that the probability that A is a mine is $1/3$, based on the clue-value of 1 and the three adjacent squares A, C, D. Given that B is a mine, you might even say further that the probability that A is a mine is actually $1/2$, since one of A and C must be a mine to satisfy the clue-value 2. What is the truth?

If you work it out, you'll find that the only satisfying assignments are B a mine, D as safe, and one of the three following combinations:

- C is a mine, A is not a mine, E is a mine, F is not a mine
- C is a mine, A is not a mine, E is not a mine, F is a mine
- C as not a mine, A is a mine, E is a mine, F is a mine

Because any one of these assignments is equally likely, we can then assess the probability of a given square being a mine as

```
[___][1/3][___]
[3/3][2/3][0/3]
[2/3][___][2/3]
```

From the above, we see that the least risky square to click on (in terms of stepping on a mine) is D, but the second least risky is in fact A.

In this way, given a partially revealed board, we can assess the possible states of unknown squares, and in doing so determine the probability of any event. (For instance, the probably that A and F are the same thing is 2/3.)

Computational Comments:

- *You should be able to use your inference engine from Assignment 2 to help bootstrap the computation of these probabilities. How?*
- *Consider dividing the board into 'active unknowns', the squares that are touching some clue, and 'inactive unknowns', the squares that are touching no clues. For any inactive unknown, that square is equally likely to be a mine as it is to be safe. Why?*

In your writeup, describe in detail how you compute these probabilities and what you utilized from Assignment 2 to do so. If you approximate or estimate anything (not strictly necessary), be clear as to what you approximated, why, and why you feel it was valid.

The Basic Agents

In each of the following subsections, you will compare your agent from Assignment 2 (Pure Random Decision Making) with a (slightly) improved agent.

Minimizing Cost

Consider the following slightly improved agent, built off your agent from Assignment 2:

- Given the current state of the board, use your inference engine from Assignment 2 to figure out as much as possible about the remaining squares.
- If any square can be identified as a mine, flag it to not select it in the future.
- If any square can be identified as safe, click on it, add the clue to your knowledge base, and continue to infer until nothing definite remains.
- Once nothing definite (mines or safe) can be determined, assess for each unrevealed square how likely it is to be a mine.

- Click on the square with the lowest probability of being a mine (breaking ties at random), add the result to your knowledge base.
- Repeat (restarting the inference loop) until the board is cleared.
- The final total cost is taken to be the total number of mines stepped on.

Generate data and plot 'mine density' vs 'average cost' for both the pure random decision agent from Assignment 2, and the basic cost minimizing agent above. How does the slightly improved agent compare? Why? How frequently is the 'improved' decision making necessary?

Minimizing Risk

Consider the slightly improved agent, built off your agent from Assignment 2:

- Given the current state of the board, use your inference engine from Assignment 2 to figure out as much as possible about the remaining squares.
- If any square can be identified as a mine, flag it to not select it in the future.
- If any square can be identified as safe, click on it, add the clue to your knowledge base, and continue to infer until nothing definite remains.
- Once nothing definite (mines or safe) can be determined, assess for each unrevealed square how likely it is to be a mine.
- For each square, determine *the expected number of squares that can be worked out if you clicked on it*:
  - Let q be the likelihood that the square is a mine, and let R be the number of squares that you could work out if it is a mine.
  - Let $1-q$ be the likelihood that the square is safe, and let S be the number of squares that you could work out if it is safe.
  - The expected number of squares that could be worked out is $qR + (1-q)S$.
- Click on the square with the highest expected number of solvable squares (breaking ties at random), add the result to your knowledge base.
- Repeat (restarting the inference loop) until the board is cleared.
- The final total risk is taken to be the total number of squares stepped on *without knowing what they are.*

Generate data and plot 'mine density' vs 'average risk' for both the pure random decision agent from Assignment 2, and the basic risk minimizing agent above. How does the slightly improved agent compare? Why? How frequently is the 'improved' decision making necessary?

Bonus

How does the risk minimizing agent compare to the cost minimizing agent when it comes to minimizing cost? How does the cost minimizing agent compare to the risk minimizing agent when it comes to minimizing risk? Generate data. Discuss. (Points for clarity and thoroughness.)

The Improved Agent

For one of either minimizing cost or minimizing risk, build an improved decision making agent that beats the simple agent described in the previous section (that isn't just the other simple agent!). Plot the relevant data for your original Assignment 2 agent, the slightly improved agent (from the previous section), and your improved agent. Describe in detail, and justify, the logic and construction of your agent. Is your improved agent significantly better than the other two? How frequently does your improved agent make different decisions than the slightly improved agent? Is your improved agent as good as it can be? How could it be improved?

Bonus

Construct and analyze an improved agent for the other metric you didn't already do.