The Resistance: Markov Models

Joel Milligan

October 21, 2021

Contents

1 The Resistance

The Resistance is a party game of social deduction. It is designed for five to ten players, lasts about 30 minutes, and has no player elimination. At the start of a game, each player is assigned one of two teams, the resistance or the spies. Each game lasts for three to five rounds. In each round, the "Leader" picks players to go on a mission. Everyone then publicly votes on whether or not to approve the mission. Once a vote passes, the each of the chosen players secretly decide to either support or sabotage the mission. After each team member has decided, the mission result is determined. If there is at least one sabotage, the mission fails, otherwise it will succeed. When three missions succeed, the resistance have won the game, and when three missions fail, the spies win. [1]

The Resistance has a focus on hidden information and deception, with each team dealing with hidden information in different ways. The Resistance's goal is to find out the hidden information, being which players are the spies. The Resistance then attempts to use this information to make sure no spies go on missions. On the other hand, the spies have perfect information from the beginning of the game and are thus attempting to obfuscate who the spies are, so the resistance approve them to go on missions, where they will have a chance to sabotage them.

2 The Question

Do agents using Monte Carlo Tree Search perform better as the resistance or as spies in The Resistance?

This question is interesting because it can give insight on how MCTS performs in environments with perfect information (as the spies) versus environments with imperfect information (as the resistance). This is especially interesting in The Resistance, because the resistance may have less information, but they are greater in number than the spies and therefore have more influence as a team over the state of the game due to the game's democratic voting style.

I intend to answer the question by creating an agent that implements MCTS for both the resistance and spy teams. I will then run many simulations with varying ratios of spies vs resistance to see how well the agents perform against one another.

2.1 Data Collected

1. TODO

What the data tell us about the question?

3 Literature Review of Techniques

3.1 Bayes' Rule

Baye's Theorem (1) is useful for probability.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

3.2 Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) is a way to apply the Monte Carlo method to trees, especially in game playing. It consists of evaluating actions for a given state by randomly playing out the state produced by said action until you reach a terminal state, which in the case of The Resistance would be either the spies or resistance winning. This result would then be back propagated to the initial node, with the value of the terminal state. This is then iterated on with the new values, with the agent repeatedly running the MCTS over and over, improving results. Once time have run out, one of the action will have a value higher than the others and this action will be taken. This process repeats for every state until the game is over.

MCTS is considered desirable in games with high branching factors as an alternative to Minimax. This is because it requires a lot less compute time and memory than minimax due to not exhaustively expanding every node, only randomly sampling until time is up.

There are 4 basic steps to

Balancing Exploration and Exploitation

Exploration is trying lesser known actions to see if something better can be found. Exploitation is taking the best known actions. Balancing exploration and exploitation is necessary for an effective MCTS. Most implementations of MCTS are based on some variant of a formula called UCT (2). The idea of the formula is to choose the node that maximises the value of the formula. [2]

$$\frac{w_i}{n_i} + c\sqrt{\frac{\ln N_i}{n_i}} \tag{2}$$

The first term in the formula, $\frac{w_i}{n_i}$, represents exploitation, adding value for nodes with high win rates, while the second term, $c\sqrt{\frac{\ln N_i}{n_i}}$, adds value for nodes that have not been visited often.

3.3 Markov Models

A sequential decision problem is . . .

Markov Models are models used to model pseudo-randomly changing systems. There are 4 common Markov models used in different situations, depending on whether every sequential state is observable and if the agent can take actions that influence the state.

As The Resistance is a game of hidden information and our agent can take various actions that influence state, such as through voting and influencing mission outcome, the Markov Model that would apply in this game would be the partially observable Markov decision process.

The Bellman Equation (3) underpins both sequential decision problem algorithms.

$$U_i = R_i + \max_a \sum_j M_{ij}^a U_j \tag{3}$$

- RELATE THIS TO THE RESISTANCE

- M_{ij}^{a} is the probability that doing Action a in State i leaves the agent in State j. This represents the transition model.
- $\sum_j M_{ij}^{a} U_j$ is the weighted sum of all possible outcomes of doing Action a in State i.
- $\max_a \sum_j M_{ij}^{a} U_j$ is the expected outcome of the best action to do in State i.
- $R_i + \max_a \sum_j M_{ij}^{a} U_j$ is the cost of being in State i, plus the optimal cost from then on.

The transition model is crucial to the Bellman Equation, so coming up with a transition model can be hard.

Value Iteration

Iterate on the utility on state.

Policy Iteration

In situations where the utility of a state is hard to determine, it can be a more effective idea to iterate on the policy itself.

- Start with an arbitary policy π
- Compute with utilities U of π, by value determination
- Update π according to U, by action determination
- Repeat until no change in π

4 Selected Techinques

References

[1] BoardGameGeek. The resistance. https://boardgamegeek.com/boardgame/41114/resistance.

[2] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, pages 282–293, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.