Werewolf Game - Group 26

Catarina Mendonca
75381
cvd.mendonca@gmail.com

Luis Borges
78349
luispb7@hotmail.com

Paulo Ritto
78929
paulo_230@hotmail.com

ABSTRACT

This paper was written for the AASMA course at Instituto Superior Tecnico and presents an approach for playing the famous *Werewolf* game (also know as *Mafia*). Our approach is based on villager attributes like intelligence or laziness and strategies they can follow according to their attributes (personality). All this will be described in a detailed manner in the following sections.

1. INTRODUCTION

In this project we decided to implement our approach for playing the *Werewolf* game. For that we used Python 3.6 and the *numpy* library to make use specially of the random choices, so that all our strategies could be implemented.

The basic idea behind the *Werewolf* Game is that we have a village where there are Villagers and some Werewolves. At night, the werewolves coordinate and kill a villager; during the day, the villagers know who died and proceed to vote in the villager that they think can be a werewolf. The werewolves, because of their anonymity during the day are considered villagers and can vote as a normal villager, not voting for other werewolves, tho. Those decisions will be based on the agent's attributes and the agents will learn throughout the game which strategies have given them a better reward, and those that haven't.

In our implementation we decided that only Villagers and Werewolves would be too simple, so we introduced two new categories of Villagers, the Seers and the Doctors. The Seers, when asked for advice, can suggest who they know it's a werewolf for sure, since they have the ability to know if someone is a werewolf or not (always voting for that person during the day). However the villagers don't know exactly who the seers are, and this makes the game more difficult but also more interesting. The Doctors can choose a villager to save during the following night and if a werewolf tries to kill that person it might fail, depending on the doctor's skill. However, just like the Seers, the doctors aren't known, making it a guessing game between the villagers and the werewolves.

Villagers win the game if all the werewolves are dead and werewolves win the game if there are as many villagers as there are werewolves.

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

2. OUR WEREWOLF IMPLEMENTATION

Villagers, Seers, Doctors and Werewolves are implemented as Python classes, with *Villager* being the superclass of all of them. Those classes have several attributes that define their *personality*. Since the villager's identities aren't known, all of them must have a *belief* regarding who's who. Our implementation is then based on a *belief* for every villager, which is coded as a dictionary with the remaining villagers as keys, and a list containing the belief that that villager is a Werewolf, the belief that that villager is a Seer, the belief that that villager is a Doctor and the belief that that villager is just a normal Villager. Our strategies will essentially modify those beliefs, and when called upon voting, every villager will vote according to their belief on who's probably the werewolf. Note that it's not guaranteed a villager will vote for whom he has a higher belief is the werewolf, he just has a higher probability of doing so. As for werewolves of course, they will have other voting strategies, as they are trying to cover up their crimes and identity, and they can't vote for other werewolves. Let's describe this.

2.1 Main Cycle and Game Modes

In our implementation of the game there are two modes.

The first mode is the *History Mode* were the player only chooses the number of villagers and werewolves and watches as the game unfolds without control over the actions of the agents. The game is presented as a story, and we're able to see the impact of the change of the number of villagers and werewolves on the game.

The second mode is the *Player Mode*. In this mode the player also chooses the number of villagers and werewolves. Next, the player can insert the name that they want for their own agent and can choose to be a villager or a werewolf. Afterwards, the player will be able to vote, to kill, to ask for advice or to give advice, if he is asked, and the main loop is the same.

The main loop consists on the *night* phase, where the werewolves will reason about who to kill, vote to kill and kill someone, and the *day* phase, where every villager will reason about who to vote according to several strategies and then vote according to their beliefs of who's the werewolf. If all werewolves are dead, villagers win. If there are as many werewolves as there are villagers, werewolves win.

2.2 Game Agents

For our implementation four types of agents were necessary, as explained before: the Villagers, the Werewolves, the Seers and the Doctors. These are contained in several lists

of agents that change over time, whenever they are killed.

The villagers can advise if they are asked, reason (think) about who might be the werewolf (being able to ask for advice as well) and finally, vote according to that reasoning.

The werewolves can vote to kill a person and do all the actions of the villagers above because they are also villagers.

The seers can do the same regular villagers can. However, they are able to see whether someone is a werewolf or not. If the seers predict someone as being a werewolf, they will always vote for him.

The doctors can choose a villager to try to save during the night and do everything like a normal villager.

2.2.1 *Agents' personality*

So that the behavior of our agents wouldn't be linear with added traits quantified in a 1-100 scale and created randomly. The agent's actions are dependent of those traits, creating like a personality for each agent. These attributes are going to be responsible for the strategy each agent is likely to pick. The traits for each villager are:

- *Intelligence*: villagers with high intelligence will more likely choose more intelligent strategies.
- *Respect*: villagers with high respect may not be as voted as others, since they are respected within the community.
- *Laziness*: lazy villagers might not want to spend lots of time thinking: they may just vote randomly instead...
- *Faith*: villagers with a higher faith on other people will more likely ask for advice.
- *Stubbornness*: when faced with the decision to chose the strategy that he find better versus the strategy that has gotten the villager a higher reward, the stubborn villager will choose his strategy instead.
- *Fear*: If a werewolf is usually very scared of everything, he will be more likely to vote to kill someone who voted for him in the previous day voting or to vote to kill someone who has voted him the most.
- *Veracity*: the villagers with a higher veracity will have a higher probability of telling the truth when asked for advice.
- *Skill*: This is a *doctor exclusive* attribute: doctors with a higher skill will have a higher probability of saving the villager they chose to save during the day.

2.3 Strategies used by Villagers to vote

So that the voting for the possible werewolf wouldn't be linear we decided to implement strategies for the voting. Every time a strategy is executed, the villager's belief is updated and normalized (so that the sum of the probabilities in the belief network is 1).

2.3.1 **Ask Someone**

Villagers with a higher faith will probabily ask someone for their advice regarding who's the werewolf. However, the other villager might lie...

2.3.2 **Random**

Lazy villagers won't spend too much time thinking about who to vote, they'll just have a higher probability of voting randomly.

2.3.3 **Dead Last Vote**

This is the most obvious strategy: villagers with a lower intelligence will vote for whoever the villager killed in the night voted last.

2.3.4 **Dead Most Voted**

A not so obvious strategy will be voting for whoever the dead villager voted the most in all the voting, not just for who he voted the last time.

2.3.5 **Least Respected**

Villagers with a higher respect among the community will more likely vote for villagers who aren't much respected, suspecting they're the werewolves.

Werewolves will also follow the same logic, except they will never vote for other werewolves. If a werewolf chooses a Dead Most Voted strategy but the most voted is also a werewolf, he will then choose Dead Last Vote. If Dead Last Vote is also a werewolf, he will choose randomly someone to vote for. Same thing happen for other strategies: for example, if a werewolf chooses to vote for the Least Respected, but the least respected is a werewolf, he will vote randomly.

2.4 Strategies used by Werewolves for kill voting

When the night arrives, werewolves must vote for who they want to kill in order to reach an agreement. However, they may use different strategies for that voting, which will now be described.

2.4.1 **Kill a Seer**

For a werewolf, killing a seer is the smartest option, since Seers are able to predict for sure whether someone is a werewolf or not and is a constant threat. Therefore, this strategy will be chosen most likely by werewolves with higher intelligence.

2.4.2 **Kill a Doctor**

Another smart (i.e. intelligent) option is to kill the doctor, since the doctor will be able to save a villager and therefore invalidate a night of work.

2.4.3 **Kill randomly**

If a werewolf has a high laziness, it won't bother to think about who to kill; just kill someone randomly.

2.4.4 **Kill the most respected**

For werewolves, in case they are also respected themselves, killing a respected villager is also a good idea.

2.4.5 **Kill who didn't vote me**

Killing someone who didn't vote for a werewolf doesn't make much sense, since that villager never really had a strong belief regarding who's a werewolf, therefore, smaller intelligences will have a high probability of choosing this strategy.

2.4.6 Kill who voted me

When a villager votes for a werewolf during the day, that werewolf (if he survives) might get scared and vote to kill that villager on the next night. Therefore, werewolves with a higher fear will choose this strategy more probably.

2.4.7 Kill who voted me the most

Werewolves who aren't that afraid (low fear) will have such an impulse for killing who voted them the last time, but instead will reflect and vote probably for whoever voted for him the most.

2.5 Learning

We also implemented a learning algorithm so our agents would be able to distinguish from experience which strategies have resulted better for them. Every agent has a Q dictionary that maps a day voting strategy to a number. A werewolf will have an additional Q dictionary, mapping kill voting strategies to numbers, as well. We now need to define *rewards*. At the end of the day voting, a villager is killed and it's revealed whether that villager was a werewolf or not. If the dead was a regular villager, all villagers except werewolves will receive a reward of -10, because their chosen strategy lead something against their goal. If the dead was a Werewolf, that's what they want and they'll receive a reward of +10. As for werewolves, if the dead is also a werewolf, they will get a -20 reward, since there are few werewolves and one of them just died. If the dead is a villager, they'll receive a +10 reward because that's their goal.

After receiving their rewards, each agent must update their Q's according to the *Q-Learning* formula:

$$Q(s) = Q(s) + \alpha(reward + \gamma maxQ(s) - Q(s)) \quad (1)$$

for $\alpha = intelligence/100$ and $\gamma = 0.9$. Therefore, when choosing a strategy to vote, an agent will have to decide whether to go for the strategy he got according to his personality versus the strategy that he knows has given him a better reward throughout the game. This will depend on the agent's stubbornness, as it's been already explained, and allows for an *exploration* vs *exploitation* dilemma.

When a game ends, the player is asked if he wants to play again. If he chooses to do so, the learning of the previous game is passed onto the next one. Our implementation will consider the surviving villagers and the surviving werewolves from last game. Then, every new villager and werewolf will start with a Q not mapping every strategy to 0, but instead to the average of the Q's from the surviving villagers and werewolves.

3. CONCLUSIONS

After playing the game, we have observed that werewolves are essentially very strong, which is actually normal, specially when there are many of them. After all, werewolves do not vote for other werewolves and kill a villager at night, being able to disguise as normal villagers during the day. Besides, on a random community, chances are that villagers might not be that intelligent and vote poorly, giving werewolves another chance at killing two villagers and reaching the same number as regular villagers faster.

Regarding the project itself, all members contributed equally, since the Python code, report and video editing were made when all three elements were present.