My 'IAGO' Agent

Rotem Rousseau
Department of Computer Science, Bar-Ilan University, Israel
rotemrousseau@gmail.com

Abstract

The agent in 'IAGO' learns about the player preferences quite impressively in initial negotiations, but it doesn't use the information learned about the player in previous games to get better results. We have developed an agent that adapts itself to the player in the second game according to the player's behavior in the first game, and adapts itself to the player in the third game according to the player's behavior in the two previous games (More weight is given to the player's behavior in the second game). We Characterized the player's behavior according to two main characteristics: (1) His cooperation, (2) His behavior (polite, threatening...). The agent we developed "punishes" the player for non-cooperation / rudeness and rewards him for cooperation / politeness. (*important note - this method isn't necessarily the best method to develop an agent whose goal is to learn player preferences in back-to-back negotiations. nonetheless, it is the method we have decided to research in the upcoming competition).

1 Introduction

Negotiation is the focus of a great deal of research, both within the traditional business and (more recently) in artificial intelligence. In recent years there has been an impressive progress in learning player's preferences while negotiating. But in reality, we won't necessarily negotiate with someone only once. We may have to negotiate with him multiple times. Take, for example, the country in which I live, Israel. Israel often negotiates with the Palestinian Authority on various issues. In the past, these negotiations were usually unsuccessful, but little by little, each party had learned the other party preferences and thus it was easier for them to negotiate. Negotiations in which both of the parties benefited. Therefore, we would like to develop an agent that can negotiate with a player multiple times and little by little will learn his preferences. We want this agent to use those preferences in order to gain the best results possible from the negotiations conducted with the player. There are many ways in which one can learn about a player's preferences in negotiations, we have decided to learn about the preferences of the player with whom we are negotiating, according to the way he cooperates with the agent and the way he behaves in that negotiation. The more fairly a player negotiates with the agent, the more fairly the agent will negotiate back. On the other hand, the more treacherous a player is, the less the agent will trust him in future negotiations. There may even be a situation where the agent decides to be treacherous towards the player if he reckons that the player is trying to take advantage of him. We do not claim that the way we have decided to learn about a player's preferences is the best, only that it's what we have decided to research in the upcoming competition.

2 Agent design

The core behavior of the agent will be the same as the core behavior of the default IAGO agent except for the following differences:

- Our agent will analyze information about the player's behavior - whether he lies to the agent during the negotiations (by contradicting himself during the negotiations). whether he cooperates with the agent - answers questions asked by the agent, especially questions about the player's preferences. Is he polite to the agent, Is he threatening him, is he angry at the agent, etc.
- The agent will be more strategic. He will usually cooperate and answer the player's questions, but if he'll notice that the player does not share information about himself, he will stop sharing information until the players shares.
- The agent won't be tolerant of lies. If he suspects that the player is lying (for example, if the player contradicts himself), he will no longer trust the player. This trust will be difficult to restore but a player may be able to try to do so in the next negotiation games.

The software main algorithm is the 'RewardPenalty algorithm' (section 2.1). It is called when the first or second negotiation ends. In this algorithm, the agent analyzes the information that has been accumulated on the player so far, decides on a 'penalty' / 'reward' that the player will receive in the next negotiation game and changes his game strategy accordingly.

2.1 RewardPenalty algorithm

The algorithm is called when one of the negotiation games ends. The agent analyzes the information he has stored on

the player so far:

- The agent's cooperation in recent games is analyzed - according to the previous games final transactions, full offers, according to the number of times the agent suspected the player lied to him and according to the number of times the player answered the agent's questions compared to the number of times the agent answered the player's questions.
- The agent analyzes the player's behavior throughout the recent negotiation games - The number of times the player was angry with the agent and the number of times the player threatened or spoke rudely to the agent.

The agent analyzes this information and reduces it to three individual parameters:

- 'Player-cooperation' value - one of the 5 following values is chosen: 'SUPER-LIAR', 'LIAR', 'NOT-COOPERATIVE', 'NEUTRAL' and 'COOPERATIVE'. The player will be 'SUPER-LIAR' if and only if he lied in the previous two games. The player will be 'LIAR' only if he isn't 'SUPER-LIAR' and he lied in the last negotiation game. If the player isn't 'LIAR' too, then one of three remaining values will be chosen. It will be chosen according to the number of times the player answered the agent's questions compared to the number of times the agent answered the player's questions.
- 'Player-Behavior' value - one of the 4 following values: VERY-POLITE, POLITE, NOT-POLITE, RUDE. If the player has always been polite (that is, the player neither threatened the agent nor spoke rudely to him nor was angry with him), then the chosen value will be VERY-POLITE. If the player isn't VERY-POLITE, then one of the three remaining values will be chosen. It will be chosen according to the amount of threats and the number of times the player was angry with the agent. The Player is "punished" more for threats and impudence than for expressing anger (using emoticons).
- 'Good-Final-transactions' value - True if the agent received in the previous negotiations games transactions that are great for him (the number of points the agent received in relation to the number of possible points was very high)

The agent analyzes the three individual parameters and gives a score to the player (low score = "penalty", high score = "reward"). This score will affect the agent in the next negotiation games by the following way:

- Relatively high score - The agent will offer deals that are good for the player.
- Above average score - The agent will tell information about his preferences to the player without expecting anything in return (even if it could be nice) .
- Average score - The agent will use the same strategy from the previous negotiation game
- Below-average score - The agent will become more 'strategic'. He won't volunteer information about his preferences to the player without getting anything in return.
- Low Score - The agent will only accept deals that are good for him.

3 Conclusions

We have developed an agent who 'imitates' the player's behavior. That is, an agent who changes his strategy in negotiations games according to the player's behavior. This is probably just one of many possible agents whose goal is to learn user preferences in back-to-back negotiations. I am positive that in the future it will be possible to combine the functionality of those agents with the same goal and thus create an agent who learns the user preferences in a most impressive way.