CITS3001: Love Letter Research Report

Students: Ashling Charles (22260738) and Paul O'Sullivan (21492328)

Unit: Artificial Intelligence

Unit Code: CITS3001

Due Date: 23/10/2019

Word Count: 2240

Ashling Charles 22260738
Paul O’Sullivan 21492328

Love Letter: AI Tournament

Introduction

Love Letter is a strategic card game for two to four players. The goal is to get your love letter to the Princess whilst deflecting letters from competing suitors. The deck is of sixteen cards, where each player starts with one card unknown to others. On a turn, players draw one card and then play one card. To win, players must use their cards to knock others out of the game.

Literature Review

There are many suitable techniques to approach the implementation of an agent for *Love Letter*. *Love Letter* is a turn-based imperfect information game. Full knowledge of the game state is extremely unlikely and only possible in the final rounds of a game. Consequently, an advanced player must be able to operate without complete information; keeping track of possible world scenarios and applying strategies for success.

When dealing with the bounded rationality of agents for specific games, one such method that requires a pre-set rule basis is the heuristic approach. This approach offers a close to optimum but not necessarily optimal strategy. Heuristics can be designed for many purposes, such as generating optimal counteroffers, predicting information about an opponent, or finding optimal agendas. Heuristics contain key underlying ideas, goals, and main results (Shaheed, 2015).

Cutright (2019) uses the game GNaT as a testbed for AI heuristic strategies. He builds the decision-making process of the computer by identifying the problem a computer tries to solve and breaks it into component problems. These can then be tested and optimal approaches determined. This was implemented in the form of modules based on heuristics. The efficacy of the modules was experimentally evaluated, with the agent comprised of the best-performing strategies.

Heuristics can be important and powerful, but are restricted to a certain game and require strong understanding and thorough implementation. It is here that reinforcement learning can be powerful when finding strategies through trial and error, and can be run over many trials to utilise the best actions of the agent. Agents can either learn policies that previously yielded themselves high reward or opponents’ strategies that resulted in high reward for opponents (Dawson, 2018). These techniques can be applied to many aspects of artificial intelligence, with a *Love Letter* agent being one of these.

Sampling algorithms can be used in artificial intelligence to determine the likelihood of an agent's success. They utilise tree structures where nodes are used to represent the state of the world. As the game plays out in these searches the agent can choose the nodes that are expected to give the optimum outcome. Omarov implements four AI agents for a two-player game of *Love Letter*: Single Observer Information Set Monte Carlo Tree Search (SO-ISMCTS) Algorithm, Perfect Information Monte Carlo (PIMC), Knowledge-based agent and Determinized Minimax. MCTS is an effective algorithm for an agent's decision process in incomplete information games. However, these agents were found to be ineffective and did not yield stable results due to the randomness of *Love Letter* (Omarov, 2018).

Bayes Theorem is useful in maintaining probability distribution for games that have incomplete information. The underlying concept of this theorem can be used to find the probability of events with uncertain knowledge through inductive and deductive logic.
This theorem states: $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

Literature Review (Cont.)

Inductive and deductive logic can be used to infer probabilities of states of the world. A deductive argument is valid in the instance whereby the truth of its premises guarantees the truth of its conclusions and is invalid otherwise (Gauch Jr 2012). Love Letter game rules can provide a foundation for a deduction. Inductive arguments, on the other hand, refers to reasoning from particular instances to general conclusions (Gauch Jr 2012). Gauch Jr (2012) provides examples of both arguments.

A Valid Deductive Argument:

- Premise 1. Every mammal has a heart.
- Premise 2. Every horse is a mammal.
- Conclusion. Every horse has a heart.

A Strong Inductive Argument:

- Premise 1. Every observed horse has had a heart
- Conclusion. Every horse has a heart.

Probabilities can be inferred through induction in Love Letter assuming opponents play rationally to maximise their strategy most of the time. These concepts will be further discussed under the discussion surrounding implementation.

In respect to artificial intelligence, the underlying concept of this theorem can be used to update probabilities of the state of the world. Through this, agents can utilise the probability distributions to select which actions optimise their rewards.

Selected Technique

For investigation purposes, three types of agents have been selected to analyse and compare. These agents include: a random agent that makes decisions randomly although attains to the rules of the game, a knowledge-based agent that maps a single card probability distribution to each to each player and gets updated based on any discarded cards and a Bayesian agent that maps the probability distributions of unseen cards to individual opponents and then uses inductive and deductive logic to update these distributions. These will be updated through conditional probabilities which are calculated based on the events that took place during the most recent round, with the objective of narrowing down which card an opponent possesses.

We chose this technique as we were creating an agent for *Love Letter,* a tactic based game of imperfect knowledge. This was interpreted to be a game that could be based on heuristics and statistical probabilities. For this reason reinforcement learning seemed overcomplicated for an agent for this game and would require training against a range of heuristically implemented base agents. It is expected that these approaches will yield superior results when compared to the random agent.

Implementation

A base probability is initially mapped to each player during the knowledge and Bayesian agent's move. This is calculated through basic card counting and incorporates the probability distribution of the deck or unseen cards at the time of the test agent's move. A copy of this is made and then mapped to each player. The knowledge-based agent will simply follow this probability distribution, however, the Bayesian agent will perform further analysis and ultimately form a probability distribution for each opponent.

Implementation (Cont.)

Improving the Bayesian agent's knowledge of the state of the world becomes a bit more complex. Most in-game actions can give an inference of the player's card and its target's card. We can use inductive and deductive logic to distinguish these inferences. Before updating probabilities through inductive and deductive logic, we must first make some assumptions. For the purpose of this experiment it can be assumed that opponents will play rationally more than they play irrationally. An example of rational play in *Love Letter* may include playing a Guard and guessing a card that has a possible chance of being possessed by the target. Irrational play in the same situation can be observed when a player guesses a card that has no possibility of being a correct guess. This is also referred to as bluffing and through inductive reasoning, we can assume this occurs how often we expect an opponent to bluff. Given that learning the opponent's playing style is not possible and that neither of the test agents incorporate bluffing, we will assume that a potential opponent may bluff very seldom or 10% of the time. This is also known as a Bayesian Game that incorporates the underlying concept of Bayes theorem. For example, using inductive logic, given that a player is unlikely to possess a specific card(s), we can now choose a more superior strategy.

Deduction, on the other hand, revolves more around certainty. Love Letter game rules can provide a foundation for deduction. For example, in a situation where a Baron is played on an opponent, the Bayesian agent can utilise deductive logic to infer probabilities. It is certain that the winner of this duel must possess a card higher in value than the losing card. Consequently, we can update the probability that the winning player possesses a card lower than the losing card to zero. This allows the agent to narrow down the range of cards an opponent may have and ultimately provide a significant advantage. Further inductive and deductive inferences can be observed in *Appendix 2*.

Both the knowledge-based agent and the more advanced agent operate on the heuristics listed in *Appendix 1*. Each agent will have access to the same heuristics. Removing any uncontrolled variables or unfair advantages between agents will uphold the validity of the results.

Both agent's strategies for picking a target is split into three tiers of priority. In the first tier, the target is chosen based on whether or not the agent is certain of their card. For example, if a Priest is played against an opponent in a previous round and it is known that they possess a Baron. The second tier of target decision making is entered on that basis that a target cannot be identified from the first tier of decision making. This involves analysing the varying probability distributions of each player and then choosing who to target based on the combination of cards in the knowledge-based or Bayesian agent's hand. The implementation of attacking and defensive decision-making functions allow the Bayesian agent to decide the most optimal strategy given the probability distributions of opponents.

The attacking function is called if the agent possesses an attacking card (Guard, Baron and Prince). Both agents will always target the opponent with the most amount of hearts. However, it is very likely that there could be a tie between opponents for the most amount of hearts. In this instance, the knowledge-based agent will choose randomly from this group due to the fact that each opponent's probability distribution is identical. The Bayesian agent will, however, opt for a more superior analysis through analysing each of the opponent's probability distributions, thus allowing it to determine which opponent has the highest likelihood of being eliminated. In the instance that the agent possesses two attacking cards, it will analyse the varying probability distributions of threatening players to determine which card is most effective. The defensive function is called if the agent possesses two non-attacking cards. For example, if the knowledge-based or Bayesian agent possess a priest and a King, it will play the King and analyse the probability distributions of all active players in order to determine the target with the highest likelihood of possessing a card greater than a priest.

Validation Tests

A tournament was conducted to test the performance of both the Knowledge and Bayesian agents.

The results in Table 1 show the average and standard deviation of the win rate of the test agent, with the agent varying in its position within the game, in a four-player game, iterated over 100,000 games. This eliminates any bias to agent position and reduces the variation of win rates in smaller game iterations. Table 2 shows a tournament between all three types of agents, iterated over 100,000 games.

	Opponents (100,000 games)		
Agent	3 RandomAgents	3 KnowledgeAgents	3 BayesianAgents
RandomAgent	25% (S.D. 0.85%)	10.15% (S.D. 1.05%)	7.48% (S.D. 1.66%)
KnowledgeAgent	52.7% (S.D. 0.90%)	25% (S.D. 1.75%)	17.65% (S.D. 3.28%)
BayesianAgent	61.175% (S.D. 0.74%)	34.58% (S.D. 1.27%)	25% (S.D. 4.29%)

Table 1: Win-rates of the agents (based off Appendices 3 and 4)

	Tournament (100,000 games)			
Agents:	BayesianAgent	KnowledgeAgent	RandomAgent	RandomAgent
Winrate:	48.5%	30.4%	11.3%	9.7%

Table 2: Variation of agents used in tournament

Performance Analysis

As predicted, it can be seen that both Knowledge agent and Bayesian agent significantly outperforms the Random agent, while the Bayesian agent outperforms Knowledge agent. This is supported in both Table 1 where the test agent of competing against three of each agent, and Table 2 where the tournament consists of Bayesian, knowledge, and random agents. This indicates that utilising different probability approaches can lead to a significant improvement in performance.

A noticeable difference in the win rate depending on the position of the agent was observed. This was found to be biased towards later positions. Slots three and four resulted in better win rates than slots one and two. This is likely due to the fact that later positions have more information about the state of the world than beginning positions. Slot three had the most significant advantage, most likely due to the fact that slot three has the advantage of knowing more information about the state of the world while also playing early enough in the game to not be targeted by every opponent. This variable is uncontrollable and would be expected to be observed when playing four of the same agents.

It was found that the Bayesian and Knowledge agent could be improved in their decision-making functions. These functions are split between attacking and defensive cards however the attacking function does not take into consideration the benefits of playing a defensive card over an attacking card. Analysis of the game log found that the agents played a sub-optimal strategy resulting in eliminating an opponent in a round to being eliminated in the following round. This outcome could have been avoided through improved analysis. Implementing an analysis function to weigh up the costs and benefits of playing an attacking or defensive card can be expected to improve both agent's performance as this would determine the most effective strategy that minimises elimination while at the same time maximises eliminating a target.

Appendices

	Heuristics
1	The probabilities of cards are deducted through evaluating the discarded cards and the cards the player is holding currently, through this the agent accesses the relative probability of the cards another player may hold
2	The agent never targets a player protected by the handmaiden, ourself (unless necessary), or eliminated players
3	The agent will always target a player via the guard if the card is known and is not a guard, and only via a baron if the known card is less than the agent's other card
4	The agent will always target a play via the priest if that card is not yet known
5	The agent plays a prince against a target with a known high card target
6	The agent plays a king against a target with a known higher card value than them
7	If information of players cards are unknown, the agent will target the biggest threat (winning opponent)
8	The agent will play their lower card (to retain their higher card), unless any of the above clauses override this

Appendix 1: Heuristics applied in the knowledge and Bayesian agents

Card played by an opponent	**Inductive logic**	**Deductive logic**
Prince and King	80% likelihood opponent obtains a card greater than the value of the card played	Swapping probability distributions if King is played
Baron	n/a	Card in winning hand must be greater than the losing card
Countess	90% likelihood opponent possesses a card no less than a Prince	n/a
Guard	90% likelihood that the opponent does not possess the guessed card only if there is one of this card type remaining in the probability distribution of the remaining cards	If the guess is incorrect, the probability of the target possessing the guessed card is zero
Any card	70% likelihood that lowest card was played	n/a

Appendix 2: Improving probabilities for our Bayesian agent through inductive and deductive logic

Ashling Charles 22260738
Paul O’Sullivan 21492328

	3 RandomAgents	**3 KnowledgeAgents**	**3 BayesianAgents**
RandomAgent	Expected: 25% (control) Results: RandomAgent 0: 24.9% RandomAgent 1: 23.8% RandomAgent 2: 25.8% RandomAgent 3: 25.3 %	Results: RandomAgent 0: 10.1% KnowledgeAgent 1: 30.3% KnowledgeAgent 2: 30.4% KnowledgeAgent 3: 29.0% KnowledgeAgent 0: 31.6% RandomAgent 1: 11.1% KnowledgeAgent 2: 30.7% KnowledgeAgent 3: 26.5% KnowledgeAgent 0: 30.7% KnowledgeAgent 1: 31.2% RandomAgent 2: 10.7% KnowledgeAgent 3: 27.4% KnowledgeAgent 0: 29.3% KnowledgeAgent 1: 39.3% KnowledgeAgent 2: 31.6% RandomAgent 3: 8.7%	Results: RandomAgent 0: 5.5% BayesianAgent 1: 26.5% BayesianAgent 2: 31.8% BayesianAgent 3: 36.3% BayesianAgent 0: 27.3% RandomAgent 1: 6.7% BayesianAgent 2: 32.2% BayesianAgent 3: 33.8% BayesianAgent 0: 28.4% BayesianAgent 1: 29.4% RandomAgent 2: 9.0% BayesianAgent 3: 33.3% BayesianAgent 0: 27.4% BayesianAgent 1: 27.5% BayesianAgent 2: 36.3% RandomAgent 3: 8.7%
KnowledgeAgent	KnowledgeAgent 0: 52.4% RandomAgent 1: 15.7% RandomAgent 2: 17.3% RandomAgent 3: 14.6% RandomAgent 0: 15.6% KnowledgeAgent 1: 52.0% RandomAgent 2: 17.4% RandomAgent 3: 15.0%	Expected 25% (control) Results: KnowledgeAgent 0: 25.6% KnowledgeAgent 1: 26.2% KnowledgeAgent 2: 25.8% KnowledgeAgent 3: 22.4%	KnowledgeAgent 0: 14.1% BayesianAgent 1: 26.7% BayesianAgent 2: 30.1% BayesianAgent 3: 29.2% BayesianAgent 0: 20.5% KnowledgeAgent 1: 15.7% BayesianAgent 2: 32.1% BayesianAgent 3: 31.7%

	RandomAgent 0: 15.4% RandomAgent 1: 15.2% KnowledgeAgent 2: 54.0% RandomAgent 3: 15.4% RandomAgent 0: 16.0% RandomAgent 1: 15.5% RandomAgent 2: 16.0% KnowledgeAgent 3: 52.3%		BayesianAgent 0: 21.8% BayesianAgent 1: 27.4% KnowledgeAgent 2: 21.0% BayesianAgent 3: 29.8% BayesianAgent 0: 23.2% BayesianAgent 1: 27.5% BayesianAgent 2: 29.5% KnowledgeAgent 3: 19.8%
BayesianAgent	BayesianAgent 0: 61.5% RandomAgent 1: 11.8% RandomAgent 2: 13.6% RandomAgent 3: 13.2% RandomAgent 0: 12.0% BayesianAgent 1: 60.1% RandomAgent 2: 14.1% RandomAgent 3: 13.8% RandomAgent 0: 12.0% RandomAgent 1: 12.8% BayesianAgent 2: 61.8% RandomAgent 3: 13.5% RandomAgent 0: 12.1% RandomAgent 1: 12.7% RandomAgent 2: 13.9% BayesianAgent 3: 61.3%	BayesianAgent 0: 34.5% KnowledgeAgent 1: 21.4% KnowledgeAgent 2: 22.7% KnowledgeAgent 3: 21.4% KnowledgeAgent 0: 20.4% BayesianAgent 1: 36.1% KnowledgeAgent 2: 23.6% KnowledgeAgent 3: 19.8% KnowledgeAgent 0: 20.4% KnowledgeAgent 1: 22.8% BayesianAgent 2: 34.7% KnowledgeAgent 3: 22.0% KnowledgeAgent 0: 19.9% KnowledgeAgent 1: 23.1% KnowledgeAgent 2: 24.1% BayesianAgent 3: 33.0%	Expected 25% (control) Results: BayesianAgent 0: 20.9% BayesianAgent 1: 22.2% BayesianAgent 2: 30.3% BayesianAgent 3: 26.6%

Appendix 3: Win rates of agents in a four player game, alternating the player position given games start with agent 0

Ashling Charles 22260738
Paul O’Sullivan 21492328

%	Bayesian Agents	Knowledge Agents	Random Agents	Random Agent (vs.3 Knowledge Agents)	Knowledge Agent (vs 3 Random Agents)	RandomAgent (v. 3 Bayesian Agents)	Knowledge Agent (v. 3 BayesianAgents)	Bayesian Agent (v. 3 Knowledge Agents)	Bayesian Agent (v. 3 Random Agents
	20.9	25.6	24.9	10.1	52.4	5.5	14.1	34.5	61.5
	22.2	26.2	23.8	11.1	52	6.7	15.7	36.1	60.1
	30.3	25.8	25.8	10.7	54	9	21	34.7	61.8
	26.6	22.4	25.3	8.7	52.3	8.7	19.8	33	61.3
Average	25	25	24.95	10.15	52.675	7.475	17.65	34.575	61.175
S.D.	4.293405796	1.751190072	0.8504900548	1.05039675	0.8995369179	1.666083231	3.278719262	1.268529332	0.7455423082

Appendix 4: Calculation of the mean and standard deviation for win rates in Appendix 4 for the agent being observed

References:

Cutright, W. (2019). *Developing Artificial Intelligence Agents for a Turn-Based Imperfect Information Game*. Retrieved from: https://digitalcommons.liberty.edu/cgi/viewcontent.cgi?article=1972&context=honors

Dawson, M. Liang, R. Turner, A. *Learning to Play Love Letter with Deep Reinforcement Learning.* MIT. Retreived from: http://web.mit.edu/xbliang/www/pdf/6867-final-paper.pdf

Hugh G. Gauch Jr (2012) 'Deductive logic', in Scientific Method in Brief. [Online]. Cambridge University Press. pp. 112–130. Retrieved from: https://www-cambridge-org.ezproxy.library.uwa.edu.au/core/books/scientific-method-in-brief/043721250F8937F0EFDA933D00D67AFF

Joyce, J. (2003). *Bayes' theorem*. Retrieved from: https://stanford.library.sydney.edu.au/entries/bayes-theorem/

Omarov, T. Aslam, H. Brown, J. Reading, E. (2018). *Monte Carlo Tree Search for LoveLetter*. Retrieved from: https://www.researchgate.net/publication/327679828_Monte_Carlo_Tree_Search_for_Love_Letter

Shaheed, F. Sarit, K. Michael, W. (2015). *Principles of automated negotiation.* Cambridge: Cambridge University Press. Retreived from: https://www-cambridge-org.ezproxy.library.uwa.edu.au/core/services/aop-cambridge-core/content/view/673AC10C692434CE0AB5F258CD21F4CB/9780511751691c9_p157-175_CBO.pdf/heuristic_approaches.pdf