Sample Efficient Learning of Strategic Dialogue Policies

Wenshuo Tang, Zhuoran Wang, Verena Rieser, and Oliver Lemon
Interaction Lab
Heriot-Watt University
Edinburgh, UK
Email: wt92@hw.ac.uk Website: www.macs.hw.ac.uk/InteractionLab

Abstract

This work aims to learn strategic dialogue policies which estimate the (hidden) state of the opponent, using extensions of Partially Observable Markov Decision Processes. As a first step towards this goal, we present results of batch Reinforcement Learning (LSTD), which needs only 20% of the training data needed by SARSA(λ). This result now puts us in the position to tackle more computationally intensive partially observable environments .

1 Introduction

Strategic dialogue behaviour includes cooperative as well as non-cooperative actions and the ability to choose amongst these actions dependent on the current context, which includes your long-term goal and current state, as well as the goal and state of your interaction partner (also known as the "opponent"). In this research we investigate opponent modelling for optimising strategic dialogue using models based on Partially Observable Markov Decision Processes (POMDPs), following an initial proposal by (Rieser et al., 2012). Recent work has shown that the ability to reason about each other's beliefs (in terms of states and goals) using Decentralised POMDPs, enables agents to evolve cooperative behaviour (Vogel et al., 2013a), resolve implicatures (Vogel et al., 2013b), and reason about acceptable actions towards a human collaborator (Kamar et al., 2013). We hypothesise that this ability will also allow us to learn strategic dialogue policies which reason about the opponent's state. However, these types of extended POMDP models (and POMDPs in general) are intractable for more complex domains and approximate models are used in practise. Furthermore, they require efficient training algorithms to solve the underlying POMDP. Previous work on non-collaborative dialogue has found that it takes about **100k** of training games to learn a policy that can beat a rule-based opponent in a fully supervised MDP setting with a state space size of 16k (Efstathiou and Lemon, 2014). This previous work has used a online Reinforcement Learning algorithm called SARSA(λ). Current research on POMDPs for statistical dialogue management investigates more sample efficient algorithms such as GPTD (Gasic and Young, 2014), KTD (Daubigney et al., 2012) or LSPI (Pietquin et al., 2011).

In the following, we explore a combination of function approximation methods and offline learning, using batch Least-Squares Temporal Difference (LSTD) approximation. We evaluate this approach against previous work by (Efstathiou and Lemon, 2014) using the same experimental setup within a strategic trading game.

2 The Testbed Trading Game

Taikun is a 2-player, sequential, non-zero-sum game with imperfect information designed to investigate non-cooperative dialogue in a controlled environment (Efstathiou and Lemon, 2014). The goal of the game is for each participant to collect resources (Rock, Wheat and Sheep) via trading or by random game update. In the trading phase a player proposes a 1-for-1 trade of resources and the other player accepts or rejects the proposed trade. In the game update phase the environment randomly modifies the resources of each player by adding two or subtracting one. This information is hidden to the other player. The setup also includes a challenging rule-based adversary which wins 66% games against a random policy. Further details on the adversary's policy can be found in (Efstathiou and Lemon, 2014). The goal state of the Learning Agent (LA) and adversary are predefined and partially overlapping, as shown in Table 1, which motivates trading.

	Wheat	Rock	Sheep
LA	4	5	0
Adv.	4	0	5

Table 1: Goal state for Learning Agent and Adversary

3 Experiment setup and results

We now test different parameterisations of a sample efficient reinforcement learning algorithm called Least-Squares Temporal Difference (LSTD) (Bradtke and Barto, 1996), which is an offline function approximation approach. We evaluate these algorithms using the same setup as (Efstathiou and Lemon, 2014), where we formulate the problem as Markov Decision Process (MDP). The state is represented by the LA's set of resources (only) and the actions are 7 different trading offers (do nothing, trade X resource for Y resource). The long term reward is $+1000$ for winning a game, $+500$ for a draw and -100 for losing. We evaluate the learnt policies on 50k test games. The results are summarised in Table 2.

LA Policy	LA	Adversary	Draws	# games
SARSA(λ)	49.23%	45.62%	5.15%	100k
LSTD	44.5%	51.32%	4.18%	5k
Batch LSTD	46.31%	50.76%	2.93%	17k
Batch LSTD*	48.82%	48.03%	3.05%	20k

Table 2: Winning rates for Learning Agents (LA) trained with different algorithms.

First experiment : LSTD learning agent. For the first experiment we experiment with a 'vanilla' version of LSTD using the same state space factorisation as (Efstathiou and Lemon, 2014). The offline training data is generated by a random policy interacting against the rule-based adversary. In this experiment, the adversary outperforms the LA with 51.32% winning rate. The learning curve shows that the LSTD plateaus after 5k training games. We attribute this early convergence towards a non-optimal policy to the fact that LSTD learns from random data. In other words, since off-line algorithms do not have the capability to explore and exploit, the algorithms does not "see" enough instances of the optimal policy.

Second experiment : Batch learning. In a second experiment we use batch reinforcement learning to enhance exploitation, i.e. interleaving a piece-wise online data collection with offline learning (Lange et al., 2012). That is, it combines the policy-search efficiency of policy iteration with the data efficiency of LSTD. We start from an initial policy (LSTD policy trained on 1k games) interacting with our rule based adversary. We then iterate policy learning and data collection, where we use the latest policy to generate new training data for the next learning phase. The results show that batch LSTD has better sample efficiency and reaches higher performance than vanilla LSTD but still falls behind the adversary. We hypothesise that this is due to the insufficient representation of discriminative state features. For example, a required resource will consistently have a positive contribution to the "trade-in" action regardless of whether its amount already exceeds the goal.

Third experiment : Batch policy learning with non-linear state factorisation. We now experiment with a different state space factorisation to (Efstathiou and Lemon, 2014), where we represent the distance from the goal state. In particular, the state contains 6×3 binary variables, recording each individual resource for each of the 3 types as 0/1 up to a maximum of 5 (which is the max in the goal state). The 6th variable indicates whether the agent holds more than 5 of a given resource. The results for the re-factored Batch LSTD* show that the learned policy now performs equal to the challenging rule-based adversary and reaches a similar performance to SARSA(λ) after only 20k games, rather than 100k games[1].

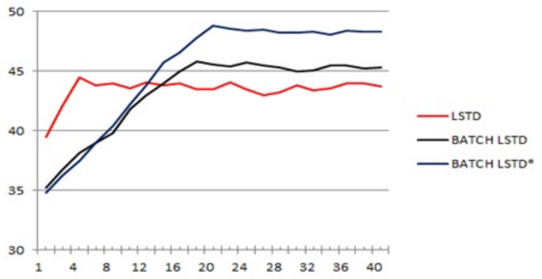

Figure 1: Learning curve: Reward over training data.

4 Discussion and future work

In this paper we have shown that it is possible to learn strategic dialogue policies which can reach a similar performance to a challenging rule-based adversary from a (relatively) small amount of training data (20k games). This now puts us in the position to tackle a more challenging problem where we account for the uncertainty in adversary's state by modelling the problem as a Partially Observable Markov Decision Process.

[1] In future work we will establish statistical significance between winning rates.

Acknowledgments

The research leading to this work has received funding from the European Community's ERC programme under grant agreement no. 269427 (STAC). `http://www.irit.fr/STAC/`

References

Steven J Bradtke and Andrew G Barto. 1996. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3):33–57.

L. Daubigney, M. Geist, and O. Pietquin. 2012. Off-policy learning in large-scale pomdp-based dialogue systems. In *Proceedings of the 37th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pages 4989 – 4992, Kyoto, Japan. IEEE.

Ioannis Efstathiou and Oliver Lemon. 2014. Learning non-cooperative dialogue behaviours. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 60–68, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.

M. Gasic and S. Young. 2014. Gaussian processes for pomdp-based dialogue manager optimization. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22(1):28–40, Jan.

Ece Kamar, Ya'akov (Kobi) Gal, and Barbara J. Grosz. 2013. Modeling information exchange opportunities for effective human-computer teamwork. *Artificial Intelligence*, 195(0):528 – 550.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. 2012. Batch reinforcement learning. In Marco Wiering and Martijn van Otterlo, editors, *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pages 45–73. Springer Berlin Heidelberg.

Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, and Hervé Frezza-Buet. 2011. Sample-efficient batch reinforcement learning for dialogue management optimization. *TSLP*, 7(3):7.

Verena Rieser, Oliver Lemon, and Simon Keizer. 2012. Opponent modelling for optimising strategic dialogue. In *The 16th workshop on the semantics and Pragmatics of Dialogue (SeineDial'12)*.

Adam Vogel, Max Bodoia, Christopher Potts, and Dan Jurafsky. 2013a. Emergence of gricean maxims from multi-agent decision theory. In *Proceedings of NAACL 2013*.

Adam Vogel, Christopher Potts, and Dan Jurafsky. 2013b. Implicatures and nested beliefs in approximate Decentralized-POMDPs. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August. Association for Computational Linguistics.