Mitigating the Negative Side Effects of Reasoning with Imperfect Models: A Multi-Objective Approach

Extended Abstract

Sandhya Saisubramanian
University of Massachusetts Amherst

Ece Kamar
Microsoft Research

Shlomo Zilberstein
University of Massachusetts Amherst

ABSTRACT

Agents often operate using imperfect models of the environment that ignore certain aspects of the real world. Reasoning with such models may lead to *negative side effects* (NSE) when satisfying the primary objective of the available model, which are inherently difficult to identify at design time. We examine how various forms of feedback can be used to learn a penalty function associated with NSE during execution. We formulate the problem of mitigating the impact of NSE as a multi-objective Markov decision process with lexicographic reward preferences and slack. Empirical evaluation of our approach on three domains shows that the proposed framework can successfully mitigate NSE.

KEYWORDS

Negative side effects; Multi-objective reasoning; Agent adaptation

ACM Reference Format:
Sandhya Saisubramanian, Ece Kamar, and Shlomo Zilberstein. 2020. Mitigating the Negative Side Effects of Reasoning with Imperfect Models: A Multi-Objective Approach. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

Agents acting in the open world typically operate based on imperfect models of the environment in which they are situated [3, 5, 6]. In general, models are carefully designed and tested with respect to a given *primary* objective, but some details in the environment that are unrelated to the agent's primary objective are ignored. In this work, we consider an *imperfect model*, denoted by $\tilde{M}$, which does not fully represent the real world but is sufficient to achieve the agent's primary assigned objective. Consequently, the agent may not be aware that its actions may result in *negative side effects* (NSE) in some states [1]. How could deployed agents respond to feedback about NSE and learn to avoid them?

A naive approach to avoiding NSE is to entirely redesign the agent's model every time negative side effects are identified, which requires exhaustive evaluation before redeploying the agent. Prior works on minimizing NSE in a model-based reasoning setting assume that the agent is aware of the model incorrectness [2] and that all negative side effects are avoidable [1, 2, 4, 9]. Real-world situations, however, tend to violate these assumptions.

We introduce a multi-objective formulation that optimizes the agent's primary objective, while mitigating NSE. The agent's model is augmented with a secondary reward function that represents the penalty for NSE. The problem is formulated as a multi-objective Markov decision process with lexicographic reward preferences (LMDP) and slack [8]. The agent's primary objective o_1 is to achieve its assigned task, while the secondary objective o_2 is to minimize NSE. The slack denotes the acceptable deviation from the optimal expected reward of o_1, in order to minimize NSE. The agent does not have knowledge about the NSE initially. Our solution framework utilizes a three-step approach to detect and mitigate NSE.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

2 PROBLEM FORMULATION

Consider an agent reasoning using its acquired model, an MDP $\tilde{M} = \langle \tilde{S}, \tilde{A}, \tilde{T}, \tilde{R} \rangle$ with a single objective, which is the primary task of the agent. The agent, however, is situated in a more complex environment, denoted by M^*, which is an extension of $\tilde{M}$ with an additional secondary objective, initially unknown to the agent. The two objectives in M^* are: assigned task (o_1) and mitigate NSE (o_2), with $o_1 \succ o_2$. We assume that an optimal policy of $\tilde{M}$ for o_1 is also optimal in M^* with respect to o_1. A **primary policy** is an optimal policy for $\tilde{M}$, optimizing o_1 defined by $\tilde{R}$.

Executing a primary policy may result in NSE in some states, since it optimizes for o_1 alone. Let $\Omega : \tilde{S} \times \tilde{A} \rightarrow \mathbb{R}$ denote the severity of the expected NSE produced by executing $\tilde{a}$ in $\tilde{s}$. The agent does not have any prior knowledge about o_2, which reflects the NSE and its associated penalty denoted by R_N. The agent may not be able to observe the NSE except for the penalty, which is proportional to the severity of the NSE, provided by the feedback mechanism. The effectiveness of our approach to avoid NSE will depend on the fidelity of $\tilde{S}$. While perfect information is not required, we assume in this work that the occurrence of NSE correlates with the features in $\tilde{S}$. This allows the agent to learn R_N using its current model $\tilde{M}$.

Given $\tilde{M}$ and feedback data regarding side effects, the agent is expected to compute a policy that optimizes o_1, while avoiding NSE, subject to a slack value. Our formulation can hence handle settings with both avoidable and unavoidable negative side effects. To achieve this, the corresponding LMDP of the agent's $\tilde{M}$ is defined by augmenting it with objective o_2 and a penalty function for NSE.

Definition 2.1. The **augmented MDP** of a given simplified model $\tilde{M}$ is a lexicographic MDP, denoted by $M = \langle S, A, T, \boldsymbol{R}, o, \delta, \gamma \rangle$ with state space $S = \tilde{S}$; action space $A = \tilde{A}$; a transition function $T = \tilde{T}$; reward vector $\boldsymbol{R} = [\mathrm{R}_1, \mathrm{R}_2]$ with $\mathrm{R}_1 = \tilde{R}$ denoting the reward associated with o_1 and $\mathrm{R}_2 = R_N$ denoting the penalty for NSE; $o = [o_1, o_2]$ denotes the objectives in the LMDP with o_1 denoting the primary objective of the agent and o_2 denotes minimizing NSE in the lexicographic order $o_1 \succ o_2$; $\delta \geq 0$ indicates the allowed deviation from the optimal expected reward for o_1 in order to minimize the NSE; and $\gamma \in [0, 1)$ is the discount factor.

Our framework for minimizing the NSE involves the following: (1) since the agent is unaware of the side effects of its actions, it collects data about NSE through various forms of oracle feedback or by exploring the environment; (2) a predictive model is trained using the gathered data to generalize the agent's observations to unseen situations, represented as a reward function R_N; and (3) the agent then computes a policy by solving the augmented MDP optimally with the given δ and learned R_N.

Slack Estimation. The slack denotes the maximum allowed loss in the expected reward of the agent's primary objective in order to minimize NSE. Typically the slack value is based on user tolerance towards NSE. We present an approach to determine the minimum slack required to avoid NSE altogether, when feasible (once knowledge about NSE is obtained). The slack is determined as the difference between the expected reward of the model before and after disabling all the actions that lead to NSE (Algorithm 1). If no solution exists after the actions are disabled, $\delta = \infty$ is returned, indicating that it is impossible to completely avoid NSE and still accomplish the task. Slack values are specified by the user when NSE are unavoidable or the δ estimated is beyond the user tolerance.

3 LEARNING NEGATIVE SIDE EFFECTS

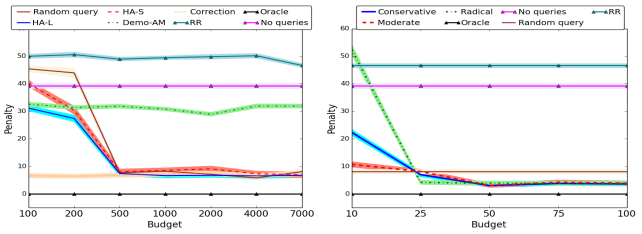
We consider the following forms of feedback that correlate with features in $\tilde{S}$ to learn about NSE.

Learning from Random Queries The agent randomly selects an (s, a) pair for querying an oracle, given a budget and without replacement, and receives the exact penalty, $R_N(s, a)$. The agent can hence learn the true underlying R_N as the budget for querying increases. Despite the benefits offered by this type of feedback, it is often unrealistic to expect exact penalty specification for randomly selected (s, a). Hence we also consider the following feedback mechanisms where the oracle provides signals to identify acceptable actions. Since such feedbacks do not provide the exact penalty for NSE, feedbacks indicating acceptable actions are mapped to zero penalty and others are mapped to a fixed, non-zero penalty denoted by k, thereby producing R_N.

Human Approval (HA) The agent randomly selects (s, a), without replacement, to query the human, who in turn either approves or disapproves the action in a state. The agent assigns $R_N(s, a) = 0$ to approved actions and $R_N(s, a) = k$ for disapproved actions. We consider two types of human approval: *strict* (HA-S) and *lenient* (HA-L). Strict feedback disapproves all actions that result in NSE.

Algorithm 1 `Slack Estimation` $(\tilde{M}, \Omega)$

$\delta \leftarrow \infty$
$\tilde{V}_1^*(s_o) \leftarrow$ Solve $\tilde{M}$ optimally with respect to o_1
Compute NSE-free transition $(\hat{T})$ by disabling all actions that result in negative side effects, $\forall(\tilde{s}, \tilde{a}, \tilde{s}')$:
$\hat{T}(\tilde{s}, \tilde{a}, \tilde{s}') \leftarrow \begin{cases} \tilde{T}(\tilde{s}, \tilde{a}, \tilde{s}'), & \text{if } \Omega(\tilde{s}, \tilde{a}) = 0 \\ 0, & \text{otherwise} \end{cases}$
if solution exists for $\langle \tilde{S}, \tilde{A}, \hat{T}, \tilde{R} \rangle$ with respect to o_1 **then**
    $\hat{V}_1^*(s_o) \leftarrow$ Solve $\langle \tilde{S}, \tilde{A}, \hat{T}, \tilde{R} \rangle$ optimally for o_1
    $\delta \leftarrow |\tilde{V}_1^*(s_o) - \hat{V}_1^*(s_o)|$
return δ;



(a) Human feedback (b) Exploration

Figure 1: Effect of various feedback techniques on NSE.

Lenient approval only disapproves actions with severe NSE. The severity threshold for HA-L is a tunable parameter.

Corrections (C) In this form of feedback, the agent performs a trajectory of its primary policy, with the oracle monitoring the process. If the oracle observes an unacceptable action at any state, it stops the agent and specifies an acceptable action to execute in that state. If all actions in a state lead to NSE, then the oracle specifies an action with the least NSE. The agent proceeds until the goal is reached or until interrupted again.

Demo-action mismatch (D-AM) In D-AM, the human provides limited demonstrations, which are trajectories from start to the goal. The agent collects these trajectories and compares them with its primary policy. For all states in which there is an action mismatch, the agent assumes its policy leads to NSE and assigns $R_N(s, a) = k$.

Learning from Exploration This feedback type uses ϵ-greedy action selection—the agent either follows its primary policy or explores a random action to learn about NSE. The agent executes an action and observes the corresponding NSE penalty, $R_N(s, a)$. We consider three exploration strategies, each with varying degrees of exploration (ϵ): *conservative*—action exploration with probability 0.1, *moderate*—action exploration with probability 0.5, and *radical*—the agent predominantly explores with probability 0.9, allowing it to possibly identify more NSE than the other exploration strategies.

4 RESULTS AND CONCLUSION

Figure 1 plots the results on a modified puddle world [7] domain. The agent's o_1 is to minimize the expected cost of navigation from start to a goal. The agent can move in all 4 directions at low or high speeds, with costs 2 and 1 respectively. Navigating over a puddle at high speed spatters water, which is undesirable. If puddles have pedestrians in the vicinity, splashing water on them results in severe NSE with a penalty of 10 and a mild NSE with a penalty of 5, otherwise. States are represented by $\langle x, y, l, p, h \rangle$ where x, y denote the agent's location, l is the speed, p, h indicate the presence of a puddle or a pedestrian in the vicinity. The agent is trained on a 15×15 grid using $\langle l, p, h \rangle$ for NSE model learning. Five test instances are generated by randomly varying the start, goal, puddle, and pedestrian locations. Values averaged over 100 trials of planning and execution, along with their standard errors, are reported. Overall, our approach significantly reduces the occurrence of NSE.

ACKNOWLEDGMENTS

Support for this work was provided in part by the Semiconductor Research Corporation under grant #2906.001.

REFERENCES

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565* (2016).

[2] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J. Russell, and Anca Dragan. 2017. Inverse reward design. In *Advances in Neural Information Processing Systems*.

[3] Subbarao Kambhampati. 2007. Model-lite planning for the web age masses: The challenges of planning with incomplete and evolving domain models. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*.

[4] Victoria Krakovna, Laurent Orseau, Miljan Martic, and Shane Legg. 2019. Penalizing side effects using stepwise relative reachability. In *IJCAI AI Safety Workshop*.

[5] Ramya Ramakrishnan, Ece Kamar, Debadeepta Dey, Julie Shah, and Eric Horvitz. 2018. Discovering Blind Spots in Reinforcement Learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*.

[6] Sandhya Saisubramanian and Shlomo Zilberstein. 2019. Adaptive outcome selection for planning with reduced models. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[7] Richard S. Sutton. 1996. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*.

[8] Kyle Hollins Wray, Shlomo Zilberstein, and Abdel-Illah Mouaddib. 2015. Multi-objective MDPs with conditional lexicographic reward preferences. In *Proceedings of the 29th Conference on Artificial Intelligence*.

[9] Shun Zhang, Edmund H. Durfee, and Satinder P. Singh. 2018. Minimax-regret querying on side effects for safe optimality in factored Markov decision processes. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*.