Learning Algorithms for Regenerative Stopping Problems with Applications to Shipping Consolidation in Logistics

Kishor Jothimurugan[1], Matthew Andrews[2], Jeongran Lee[2], Lorenzo Maggi[2]

[1] University of Pennsylvania
kishor@seas.upenn.edu
[2] Nokia Bell Labs
{matthew.andrews,jeongran.lee,lorenzo.maggi}@nokia-bell-labs.com

Keywords : *Reinforcement Learning, Shipping Consolidation, Imitation Learning*

1 Scenario

We focus on a classic regenerative stopping problem where, as long as a stopping decision is not made, costs are accumulated over time and the state continues to evolve. Both costs and state evolution are governed by a stochastic data arrival process. When the controller decides to stop, then an immediate cost is incurred and the state is reset to the start state. The goal is to minimize the long-run average cost.

A prominent instance of this class of stopping problems arises in logistics. We consider a transportation hub that sends truckloads of goods to different destinations. There is a cost for shipping a truck to a destination and the goods have delay requirements on their delivery. Our goal is to decide when to send a truck to a destination. If we delay sending a truck then more goods might arrive for the destination, meaning that we can better *consolidate* goods and send fewer trucks. On the other hand, waiting may cause disruptions to the end-to-end delivery process. Our goal is to manage this trade-off according to a cost function that includes both shipping and delay costs. The stochastic input data process represents orders being placed by a customer, while the state describes the orders waiting to be shipped.

2 Our solution approaches

Assuming that the statistical properties of the input arrival process are unknown, in this work we study the efficacy of learning-based solutions for regenerative stopping problems. In particular we compare the following three techniques.

Model-based approach. Here we employ a *direct* problem solution and we are just learning the input. More formally, we use the Markov Decision Process (MDP) solution based on an estimate of the model parameters. Future input arrival statistics are predicted, the associated approximate MDP is solved, the solution is applied for a few steps, and the process is repeated. As time goes by, the parameter estimate improves and hence the solution quality does too.

Deep reinforcement learning (DRL). In the DRL approach the solution itself is learned and we do *not* resort to solving the underlying model. Specifically, we run a deep value-based or policy-based RL algorithm such as Deep Q-Network (DQN) [2] or Proximal Policy Optimization (PPO) [4]. These algorithms learn the optimal action for any given state without explicitly learning the model. The term "deep" implies that a neural network is used to approximate the value function and/or the policy.

Imitation learning (IL). IL is a *hybrid* approach, in between model-based and DRL: the solution is learned and we do solve the underlying model, but only for past samples. More specifically, we run a hindsight optimization algorithm for each time step in the past to determine what the best decision would have been assuming that we know the future. We then

apply the Imitation Learning algorithm from [3] to imitate this hindsight optimal solution in real-time.

We show that the specific nature of the problem at hand makes it appealing to use techniques that exploit the problem structure, such as the model-based approach and IL, thanks to the efficiency of the respective solutions. Indeed, the MDP in the model-based approach can be efficiently solved via an iterative technique involving finding the root of a Lagrangian-type function, as studied by [1]. On the other hand, we prove that in our scenario the hindsight optimal solution in IL can be solved in polynomial time.

3 Evaluation on real-world data

We evaluate the performance of our algorithms on real-world data. We consider the shipping consolidation problem faced by a North American company who has one transportation hub in the United States. In three recent quarters, it processed nearly $5K$ orders with a total weight of $7M+$(kg) and around 800 different destination cities in the US. The maximum capacity of a full truck it used was a total weight of $L = 22K$ (kg).

Our experiments show that, by directly learning the optimal actions from the input data without constructing an explicit prediction for future inputs, one can better adapt to changes in the input distribution. Two key questions are addressed for each of the investigated policy learning approaches (DRL and IL). First, we define the state space in a way that is rich enough to enable learning. Second, most learning algorithms require a significant amount of training data. For most practical problems a single pass through the available data may not be sufficient to learn a good policy. We therefore present ways in which we can reuse the available training data via multiple passes.

4 Conclusions

Model-based solutions rely on an efficient technique by [1] to solve the underlying MDP. Yet, (i) their performance depend greatly on the ability to predict future inputs and (ii) show high run-time complexity, since a new MDP has to be solved whenever the prediction is updated. On the other hand, deep learning approaches (i) can adapt naturally to changes in input distribution as they directly learn a policy from historical data, and (ii) only require a NN inference at run-time.

References

[1] Bruce L Miller. Countable-state average-cost regenerative stopping problems. *Journal of Applied Probability*, 18(2):361–377, 1981.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[3] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.