Actor Based Simulation for Closed Loop Control of Supply Chain using Reinforcement Learning

Extended Abstract

Souvik Barat, Harshad Khadilkar, Hardik Meisher, Vinay Kulkarni, Vinita Baniwal, Prashant Kumar, Monika Gajrani
Tata Consultancy Services Research, India
souvik.barat@tcs.com, harshad.khadilkar@tcs.com

ABSTRACT

Reinforcement Learning (RL) has achieved a degree of success in control applications such as online gameplay and robotics, but has rarely been used to manage operations of business-critical systems such as supply chains. A key aspect of using RL in the real world is to train the agent before deployment, so as to minimise experimentation in live operation. While this is feasible for online gameplay (where the rules of the game are known) and robotics (where the dynamics are predictable), it is much more difficult for complex systems due to associated complexities, such as uncertainty, adaptability and emergent behaviour. In this paper, we describe a framework for effective integration of a reinforcement learning controller with an actor-based simulation of the complex networked system, in order to enable deployment of the RL agent in the real system with minimal further tuning.

KEYWORDS

Reinforcement learning; Simulation of complex systems; Model based simulation

ACM Reference Format:

1 INTRODUCTION

Business-critical systems need to continually make decisions to stay competitive and economically viable in a dynamic environment. Reinforcement Learning (RL) [9, 11] is a class of machine learning algorithms that can be used for controlling such complex systems in an adaptive and flexible manner. The goal of the system controller (also called RL agent) is to learn to take the best possible control actions in each possible state of the system, in order to maximise long-term system objectives. A crucial aspect of RL is the computation of next state and associated rewards for the chosen action(s), in a closed loop to enable learning. The setup is illustrated in Figure 1. This paper argues that the use of analytical expressions for modelling the environment is infeasible for complex systems, and advocates an agent/actor based modelling abstraction [1, 8] as an effective modelling aid to understand the dynamics of such complex systems. We present a framework that uses RL for exploring policies and deciding control actions, and actor-based simulation for performing accurate long-term rollouts of the policies, in order to optimise the operation of complex systems. We use the domain of supply chain replenishment as a representative example.

2 PROBLEM FORMULATION

We illustrate the generic reinforcement learning problem in the context of supply chain replenishment, which presents well-known difficulties for effective control [7, 10]. The scenario is that of a grocery retailer with a network of stores and warehouses served by a fleet of trucks for transporting products. The goal of replenishment is to regulate the availability of the entire product range in each store at all times, subject to the spatio-temporal constraints imposed by available stocks, labour capacity, truck capacity, transportation times, and available shelf space for each product in each store. A schematic of the flow of products is shown in Figure 2.

From operational perspective, each store stocks unique varieties of products, each with a maximum shelf capacity $c_{i,j}$ where $j \leq n$ is the index of the store. Further, let us denote by $x_{i,j}(t)$ the inventory of product $i$ in store $j$ at time $t$. The replenishment quantities (actions) for delivery moment $d$ are denoted by $a_{i,j}(t_d)$, and are to be computed at time $t_d$ where $\Delta$ is the lead time. The observation $O(t_d - \Delta)$ consists of the inventory of each product in each store at the time, the demand forecast for each product between the next two delivery moments, and metadata such as unit volume and weight, and shelf life. The inventory $x_{i,j}(t)$ depletes between two delivery moments $(d-1)$ and $d$, and undergoes a step increase by amount $a_{i,j}(t_d)$ at time $t_d$.

The reward $r(t_{d-1})$ is a function of the actions $a_{i,j}(t_{d-1})$ and the inventory $x_{i,j}(t)$ in $t \in [t_{d-1}, t_d]$. Two quantities are of particular interest: (i) the number of products that remain available throughout the time interval $[t_{d-1}, t_d]$, and (ii) the wastage of any products

![Figure 1: Interaction of RL agent with an environment.](image-url)
A reinforcement learning problem is described by a Markov Decision Process (MDP) [11] represented by a tuple $(S, \mathcal{A}, \mathcal{R}, P, \gamma)$. Here, $S$ is the set of states of the system, $\mathcal{A}$ is the set of control actions, $\mathcal{R}$ is the set of possible rewards, $P$ is the (possibly stochastic) transition function from $(S, \mathcal{A}) \rightarrow S$, and $\gamma$ is a discount factor for future rewards. In several cases, the agent is unable to observe the state space entirely, resulting in a partially-observable MDP or POMDP [11]. Observations $O$ are derived from $S$ to represent what the agent can sense. The RL agent should compute a policy $\pi : \mathcal{A} \rightarrow \mathcal{O}$ that maximises the discounted long-term reward. We use a form of RL known as A2C [6] to compute the actions. The Critic evaluates the goodness of the current system state, while the Actor chooses an action that maximises the improvement in value in the next state.

We propose an actor based simulation framework [4] for training the RL agent in a synthetic environment as shown in Figure 3. The proposed framework contains two control loops: (i) a model centric loop for mapping $\mathcal{A} \rightarrow O$ based on the actions of the RL agent and their effect on the system, and (ii) a real time control loop. We consider an extended form of actor model [3] to closely mimic the

$$r(t_d) = 1 - \frac{\text{count}(x_{i,j} < \rho)}{k} - \frac{\sum_{i=1}^{k} \sum_{j=1}^{n} w_{i,j}(t_{d-1})}{\sum_{i=1}^{k} \sum_{j=1}^{n} X_{i,j}}$$

where $\text{count}(x_{i,j} < \rho)$ is the number of products that run out of inventory (drop below fraction $\rho$) at some time $t \in [t_d, t_{d+1})$, $w_{i,j}(t_{d-1})$ is the number of units of product $i$ in store $j$ that had to be discarded in the time interval because they exceeded their shelf lives, and $X_{i,j}$ is the shelf capacity for product $i$ in store $j$.

3 METHODOLOGY

A reinforcement learning problem is described by a Markov Decision Process (MDP) [11] represented by a tuple $(S, \mathcal{A}, \mathcal{R}, P, \gamma)$. Here, $S$ is the set of states of the system, $\mathcal{A}$ is the set of control actions, $\mathcal{R}$ is the set of possible rewards, $P$ is the (possibly stochastic) transition function from $(S, \mathcal{A}) \rightarrow S$, and $\gamma$ is a discount factor for future rewards. In several cases, the agent is unable to observe the state space entirely, resulting in a partially-observable MDP or POMDP [11]. Observations $O$ are derived from $S$ to represent what the agent can sense. The RL agent should compute a policy $\pi : \mathcal{A} \rightarrow \mathcal{O}$ that maximises the discounted long-term reward. We use a form of RL known as A2C [6] to compute the actions. The Critic evaluates the goodness of the current system state, while the Actor chooses an action that maximises the improvement in value in the next state.

We propose an actor based simulation framework [4] for training the RL agent in a synthetic environment as shown in Figure 3. The proposed framework contains two control loops: (i) a model centric loop for mapping $\mathcal{A} \rightarrow O$ based on the actions of the RL agent and their effect on the system, and (ii) a real time control loop. We consider an extended form of actor model [3] to closely mimic the...
REFERENCES


