Poster Abstract - AdaPool: An Adaptive Model-Free Ride-Sharing Approach for Dispatching using Deep Reinforcement Learning

Marina Haliem
mwadea@purdue.edu
Purdue University

Vaneet Aggarwal
vaneet@purdue.edu
Purdue University

Bharat Bhargava
bbshail@purdue.edu
Purdue University

ABSTRACT

Deep Reinforcement Learning (RL) suffer from catastrophic forgetting due to being agnostic to the timescale of changes in the distribution of experiences. Although, RL algorithms are guaranteed to converge to optimal policies in Markov decision processes, this only holds in the presence of static environments. However, this assumption is very restrictive. In many real world problems like ridesharing, traffic control, etc., we are dealing with highly dynamic environments, where RL methods yield only sub-optimal decisions. In this paper, we introduce an adaptive model-free deep reinforcement approach that can recognize diurnal patterns in the ridesharing environment. To achieve this, we (1) adopt a change point detection algorithm to detect the changes in the distribution of experiences, then (2) we develop a Deep Q Network (DQN) agent that is capable of recognizing diurnal patterns and making informed dispatching decisions according to the changes in the underlying environment. Based on the demand, our DQN approach re-balances idle vehicles by dispatching them to the areas of anticipated high demand using Deep Reinforcement Learning. This approach can be adopted in various domains through tuning the RL agent's objective function, where it will still capture the changes in the corresponding underlying environment. Our framework is validated using the New York City Taxi public dataset. Experimental results show the effectiveness of our approach in real-time and large scale settings.

CCS CONCEPTS

• **Computer systems organization → Real-time systems**; • **Computing methodologies → Multi-agent planning**.

KEYWORDS

Deep Reinforcement Learning, Neural Networks, Car Pooling, Mobility on Demand, Multi-agent, Intelligent Transportation

ACM Reference Format:
Marina Haliem, Vaneet Aggarwal, and Bharat Bhargava. 2020. Poster Abstract - AdaPool: An Adaptive Model-Free Ride-Sharing Approach for Dispatching using Deep Reinforcement Learning. In *The 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '20), November 18–20, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3408308.3431114

1 INTRODUCTION

In Q learning, there is a tight coupling between the learning dynamics (probabilty of choosing an action) and underlying execution policy (the effective rate of upating the Q value associated with that action). This coupling can cause performance degradation in dynamic noisy environments [1]. As the RL agent continues to build on its experiences in order to learn increasingly complex tasks, it should be able to quickly adapt while maintaining its acquired knowledge. However, once the i.i.d assumption is violated, artificial neural networks have been shown to suffer from *catastophic forgeting* [4]. In literature, most approaches that address catastrophic forgetting focus on sequential learning of distinct tasks, where they rely on the awareness of task boundaries [7], This is not practical because in many situations the data distribution evolve gradually over time during training, and thus can not be discretized into separate tasks. We address this problem in the ride-sharing environemnt, where the data distribution can change at multiple and unpredictable timescales while the training the agent to learn one single task (i.e, making dispatching decisions). This can arise due to the fact that (i) states are correlated in time, (ii) the dynamics of the agent's environment are non-stationary [3].

Thus, a robust framework is needed to identify various diurnal patterns and recognize the changes in the underlying environmnent, when the environment dynamics or rewards change with time, and thus quickly adapt its policy to maximize the long-term cumulative rewards collected and ensure efficient system operation as well. This paper utilizes the dispatch of idle vehicles using a Deep Q-learning (DQN) framework as in [2], and we add the profit term in the reward function so that the output expected discounted rewards (Q-values) associated with each action, becomes a good reflection of the expected earnings gained from perfroming this action. We identify the following as our major contributions:

- We propose a model-free RL algorithm for handling non-stationary environments, where we adapt Deep Q-learning (QL) to learn optimal policies for different environment models. This approach is built on top of a distributed model-free approach for matching and dispatching vehicles in large-scale systems, DeepPool [2].
- We adopt a change point algorithm to detect the changes in data distribution, and thus identify different diurnal patterns within the day. This model utilizes data samples collected during training and it leverges a novel detection algorithm [8].
- Using results of change detection, the RL agent switches between models, and estimates policy for the new model or improves the policy learnt, if the model had been previously experienced. In this manner, our method avoids *catastrophic forgetting* [4].
- Through experiments using real-word dataset of New York City's taxi trip records [5] (15 million trips), we simulate the ride-sharing system. We show that the optimization problem of our

BuildSys '20, November 18–20, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-8061-4/20/11.
https://doi.org/10.1145/3408308.3431114

novel AdaPool framework is formulated such that it enhances the overall accepatance rate, increases the profit margins of the fleet, minimizes the extra travel distance and the average idle time, when compared to non-adaptive RL approaches.

2 METHODOLOGY

We consider the scenario where the environment changes between models 1, 2, ..., n dynamically. The implication of the non-stationary environment is this: when the agent exercises a control a_t at time t, the next state s_{t+1} as well as the reward r_t are functions of the *active* environment model dynamics. In our approach, we assume the knowledge of the pattern of change in the environemnt models $M_1, M_2, ...M_n$. However, neither the context information of each model nor the change points $T_1, T_2, ...$, etc., when these model changes occur, are known to the RL agent. In this case, the agent can collect experience tuples while simultaneously following a model-free learning algorithm to learn an approximately optimal policy. Instead of assuming any specific structure, our model-free approach learns the Q-values dynamically using convolutional neural networks. Our method works in tandem with a change point detection algorithm, to get information about the changes in the environemnt. Then, it updates Q-values of the relevant model whenever a change is detected and does not attempt to estimate the transition and reward functions for the new model. Additionally, if the method finds that samples are obtained from a previously observed model, it updates the Q values corresponding to that model. Thus, in this manner, the information which was learnt and stored earlier (in the form of Q-values) is not lost.

The learning begins by obtaining experience tuples E_t according to the dynamics and reward function of current active model M_{θ_c} . The state and reward obtained are stored as experience tuples, since model information is not known. The samples can be analyzed for context changes in batch mode or online mode. If a change gets detected, then the counter c is incremented, signalling that the agent believes that context has changed. We adapt the online parametric Dirichlet changepoint (ODCP) detection algorithm proposed in [8] for data consisting of experience tuples. This algorithm transforms any discrete or continuous data into compositional data and utilizes Dirichlet parameter likelihood testing to detect change points. Multiple changepoints are detected by performing a sequence of single changepoint detections. Although ODCP requires the multivariate data to be i.i.d samples from a distribution. The justification in [6] explains the utilization of ODCP in the Markovian setting, where the data obtained does not consist of independent samples.

At every time step t, the DQN agent obtains a representation for the environment, $s_{t,n}$, and calculates a reward r_t associated with each dispatch-to location in the action space $a_{t,n}$ according to the dynamics and reward function of current active model M_{θ_c}. Based on this information, the agent takes an action that directs the vehicle to different dispatch zone where the expected discounted future reward is maximized. In our algorithm, we define the reward r_k as a weighted sum of different performance components that reflect the objectives of our DQN agent. The reward will be learnt from the environment for individual vehicles and then leveraged by the agnet/optimizer to optimize its decisions.

We define the overall objectives of the dispatcher, where our dispatch policy aims to (1) minimize the supply-demand mismatch: (diff$_t$), (2) minimize the dispatch time: T_t^D (i.e., the expected travel time of vehicle V_j to go zone m at time step t), (3) minimize the extra travel time a vehicle takes for car-pooling compared to serving one customer: Δt, (4) maximize the fleet profits P_t, and (5) minimize the number of utilized vehicles: e_t. We capture this by minimizing the number of vehicles that become active from being inactive at time step t. The DQN overall reward function is represented as a weighted sum of these terms for individual agents/vehicles:

$$r_{t,n} = r(s_{t,n}, a_{t,n}) = \beta_1 C_{t,n} + \beta_2 T_{t,n}^D + \beta_3 T_{t,n}^E + \beta_4 \mathbb{P}_{t,n} + \beta_5 [\max(e_{t,n} - e_{t-1,n}, 0)] \quad (1)$$

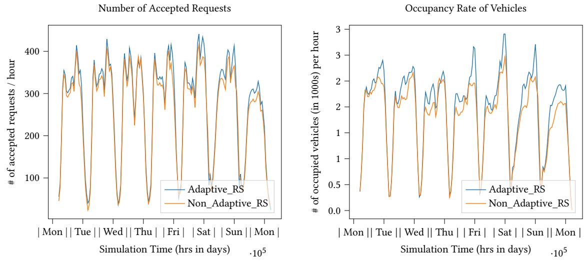

Figure 1: Evaluation Metrices for AdaPool and the baseline

3 EXPERIMENTAL RESULTS

Our simulator is created based on real public dataset of taxi trips in Manhattan, New York city [5]. We consider the data of June 2016 for training, and one week from July 2016 for evaluations. We trained our DQN neural networks for 10000 epochs and used the most recent 5000 experiences as a replay memory. We compare our adaptive RL approach to a baseline non-adaptive approach that only learns one model throughout the training.

Figure 1 shows that AdaPool improves the overall acceptance and occupancy rates. Over a week long of simulation, AdaPool consistently shows a significantly larger number of utilized vehicles (≈ 800 extra vehicles) in the fleet as well as an approx. 10 % higher acceptance rate for ride requests. This comes at the cost of only a slight increase in the average travel distance of the fleet. This can be explained by the additional number of requests served, which will -in turn- increase the average profits of the fleet as well. This is a positive outcome that points towards the viability of our proposed approach to learn diurnal patterns and adapt in a timely manner.

REFERENCES

[1] Sherief Abdallah and Michael Kaisers. 2016. Addressing environment non-stationarity by repeating Q-learning updates. *The Journal of Machine Learning Research* 17, 1 (2016), 1582–1612.

[2] A. Alabbasi, A. Ghosh, and V. Aggarwal. 2019. DeepPool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. In *EEE Transactions on Intelligent Transportation Systems*, Vol. 20.12.

[3] Christos Kaplanis, Murray Shanahan, and Claudia C. 2019. Policy consolidation for continual reinforcement learning. *arXiv preprint arXiv:1902.00255* (2019).

[4] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. 2017. Measuring catastrophic forgetting in neural networks. *arXiv preprint arXiv:1708.02072* (2017).

[5] NYC.gov. 2019. NYC Taxi and Limousine Commission-Trip Record Data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[6] Sindhu Padakandla, Shalabh Bhatnagar, et al. 2019. Reinforcement learning in non-stationary environments. *arXiv preprint arXiv:1905.03970* (2019).

[7] Paul Ruvolo and Eric Eaton. 2013. ELLA: An efficient lifelong learning algorithm. In *International Conference on Machine Learning*. 507–515.

[8] Nitin Singh, Pankaj Dayama, Vinayaka Pandit, et al. 2019. Change Point Detection for Compositional Multivariate Data. *arXiv preprint arXiv:1901.04935* (2019).