Learning Traffic Signal Control from Demonstrations

Yuanhao Xiong
Zhejiang University
xiongyh@zju.edu.cn

Guanjie Zheng
The Pennsylvania State University
gjz5038@ist.psu.edu

Kai Xu
Shanghai Tianrang Intelligent Technology Co., Ltd
kai.xu@tianrang-inc.com

Zhenhui Li
The Pennsylvania State University
jessieli@ist.psu.edu

ABSTRACT

Reinforcement learning (RL) has recently become a promising approach in various decision-making tasks. Among them, traffic signal control is the one where RL makes a great breakthrough. However, these methods always suffer from the prominent exploration problem and even fail to converge. To resolve this issue, we make an analogy between agents and humans. Agents can learn from demonstrations generated by traditional traffic signal control methods, in the similar way as people master a skill from expert knowledge. Therefore, we propose DemoLight, for the first time, to leverage demonstrations collected from classic methods to accelerate learning. Based on the state-of-the-art deep RL method Advantage Actor-Critic (A2C), training with demos are carried out for both the actor and the critic and reinforcement learning is followed for further improvement. Results under real-world datasets show that DemoLight enables a more efficient exploration and outperforms existing baselines with faster convergence and better performance.

ACM Reference Format:
Yuanhao Xiong, Guanjie Zheng, Kai Xu, and Zhenhui Li. 2019. Learning Traffic Signal Control from Demonstrations. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3357384.3358079

1 INTRODUCTION

Traffic congestion has been affecting people's daily lives nowadays. A significant amount of time on commute can be spent due to bad traffic conditions. To alleviate this issue, one of effective ways is a more efficient traffic signal control system.

With the development of computing resources and learning technologies, researchers begin to study reinforcement learning (RL) techniques for traffic signal control. The strength that RL learns an optimal policy by interacting with the environment makes it suitable for real-world traffic signal control. Currently, many RL methods have been proposed. For example, DRL [8] and IntelliLight [12] use DQN to search the policy. In addition, FRAP [15] follows the phase competition principle to design the network and IA2C [1] selects actor-critic with LSTM to improve and stabilize performance.

Although methods mentioned above do achieve relatively satisfactory performance in traffic signal control, they still suffer from the key challenge in RL: large exploration space. A naive trial-and-error approach in such space leads to slow convergence and bad performance. In fact, to reduce unnecessary exploration, inspirations can be obtained from humans. When humans attempt to master a skill, they often refer to expert knowledge, speeding up the learning process. Such knowledge can also work as catalysts in traffic signal control. However, there are two questions to be answered first: (1) Is there any expert knowledge in traffic signal control? (2) How can such knowledge be used to benefit exploration?

For the first question, we seek expert knowledge from classical methods in transportation field. There are several effective methods proposed by transportation researchers. For instance, Self-Organizing Traffic Light control (SOTL) [2] is one of such methods that adjust timing plans according to traffic dynamics. These transportation methods are often based on assumptions of traffic models, so they may not work well under certain scenarios. But they can serve as a strong baseline and can be treated as expert knowledge to benefit RL methods, without making any pre-defined assumptions.

As to utilization of expert knowledge, we can leverage it in the form of demonstrations, i.e., expert-like trajectories in decision-making tasks. Recent works have put forward various algorithms learning from demonstrations, and showed that they can tackle the exploration problem in RL efficiently. DQfD [3] accelerates learning with a small set of demonstrations on Atari games by modified loss functions while DDPGfD [10] makes an extension to robotic control problems. For our problem, traffic situations and corresponding signal plans generated by traditional methods can be regarded as demos, and then be incorporated to improve performance.

To the best of our knowledge, we are the first to propose a method integrating demonstrations into RL in traffic signal control. Our work bridges the gap between fields of transportation and machine learning. Our main contributions are summarized as follows:

- We exploit demonstrations collected from a traditional traffic signal control method to accelerate our actor-critic algorithm.
- We train the actor and the critic with demonstrations respectively to guarantee an expert-like initialization.
- Extensive experiments are performed to show efficiency and effectiveness of our algorithm under guidance of demonstrations.

2 RELATED WORK

Traffic signal control. Recently, reinforcement learning methods have shown superior performance in traffic signal control problems,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '19, November 3–7, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358079

compared to the traditional transportation approaches [11]. However, the essential challenge of RL methods is that they may suffer from severe exploration difficulties when facing large state/action space. *Despite plenty of human prior knowledge from transportation theories and traffic policy about controlling the traffic signal, there have never been attempts in utilizing them to reduce the exploration cost for reinforcement learning methods.*

Learning from demonstrations. Several studies have been proposed to reduce the exploration cost in RL methods by learning from demonstrations. These methods integrate the two parts, mimicking the demonstrations, and learning by exploration. They are usually implemented with a modified loss design. Although they have achieved success in such fields as Atari games [3], there are no attempts in applying this idea in traffic signal control.

Imitation learning methods (e.g., GAIL [4]) and inverse reinforcement learning methods (e.g., EAIRL [6]) aim to train an agent from "perfect" expert trajectories. They do not discuss how to achieve further improvement. Since only using expert knowledge will not bring a perfect traffic signal control policy, such methods are not applicable to our problem and will not be compared.

3 BACKGROUND

3.1 Reinforcement Learning

Our traffic signal control problem can be considered as a standard Markov Decision Process (MDP) with an agent choosing optimal actions. We assume that traffic states at the intersection are fully observed. The goal of the agent is to learn a policy for operating the signals which optimizes travel time. Typically, the MDP problem is formulated by a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$. Given the state set $\mathcal{S}$, action set $\mathcal{A}$, the reward function $\mathcal{R}$ is a function of $\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. The discounted return is denoted by $G_t = \sum_{l=t}^{T} \gamma^{(l-t)} R(s_t, a_t)$ where T is the horizon and γ is a discount factor for future rewards.

The agent aims to learn a stochastic policy $\pi(A_t = a | S_t = s)$, which is a mapping from states to action probabilities, so that the following expected discounted return $J(\pi)$ is maximized:

$$J(\pi) = \mathbb{E}_\pi[G_0] = \mathbb{E}_{(s_0, a_0, s_1, \ldots)} \left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t) \right] \qquad (1)$$

where $(s_0, a_0, s_1, \ldots)$ is a trajectory generated by policy π. Note that state transition matrix $\mathcal{P}$ is not described in model-free methods. For traffic signal control, our RL agent is defined the same as [15].

3.2 Actor-Critic

Numerous algorithms have been developed to solve the RL problem, such as Q-learning and policy gradient. Most of them involve constructing an estimate of the expected return using either value function or action-value function, which can be written as follows:

$$V^\pi(s_t) = \mathbb{E}_\pi[G_t | s_t], \quad Q^\pi(s_t, a_t) = \mathbb{E}_\pi[G_t | s_t, a_t] \qquad (2)$$

As a recent work has demonstrated that actor-critic outperforms Q-learning and pure policy gradient in traffic signal control [1], we select the actor-critic framework for our method, which bridges the gap between policy-based and value approximation methods in RL. The actor and the critic are represented by two neural networks with parameters θ_π and θ_Q respectively. The fundamental idea of actor-critic is that training an action-value function (**critic**) while simultaneously updating the policy parameters (**actor**) in the direction suggested by the critic. Mathematically, the critic parameter θ_Q are trained to minimize the following loss of TD error:

$$y = R(s, a) + \gamma Q(s', a'), \quad L(\theta_Q) = \frac{1}{2}\left(y - Q(s, a | \theta_Q)\right)^2 \qquad (3)$$

where the next state s' is determined by the environment after taking action a and the next action a' is sampled by $a' \sim \pi(a|s')$. For the actor, the gradient is in the form of:

$$g = \mathbb{E}_\pi[Q(s, a) \nabla_{\theta_\pi} \log \pi(a|s)] \qquad (4)$$

In fact, demonstrations can also improve DQN models. However, considering superior performance of actor-critic and limited space, we will not discuss DQN with demonstrations in this paper.

4 METHOD

4.1 Demonstration Collection

How to collect demonstrations is essential for the problem. Unlike Atari games whose demos are obtained from human experts, there is no real expert who can provide optimal trajectories in traffic signal control. However, a variety of traditional signal control algorithms have been proposed and achieved relatively good performance. Among these algorithms, we choose SOTL as the expert because it is an adaptive method based on traffic volume. Using SOTL for demonstration collection, our method can quickly learn a better initialization, i.e., to allocate much more time of green signal to movements with higher traffic. Since SOTL controls the traffic signal by some threshold parameters, we just tune parameters to find best performance and then collect demonstrations.

4.2 Modified Action-Value AC

In practice, directly applying the algorithm in Section 3.2 will suffer from high variance of trajectories. A widely used strategy is to subtract a baseline from the discounted return to reduce the variance of gradient estimation while keeping the bias unchanged. For example, a common way is to subtract state-value from action-value, and if applied, we would use an advantage function $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$ in the gradient ascent update called A2C. However, approximating a new value function $V^\pi(s)$ will introduce additional errors and biases. Since our target is to learn a good initialization for the action-value function, we can replace $V^\pi(s)$ using the following equation:

$$V^\pi(s_t) = \mathbb{E}_\pi[Q^\pi(s_t, a) | s_t] = \sum_a \pi(a|s) Q^\pi(s, a) \qquad (5)$$

Then the advantage function can be written as:

$$A^\pi(s, a) = Q^\pi(s, a) - \sum_a \pi(a|s) Q^\pi(s, a) \qquad (6)$$

where Q^π and π are parameterized by θ_Q and θ_π respectively.

4.3 Actor Training from Demonstrations

First, we put forward training the actor from demos. As stochastic policies are modeled and learned by neural networks in our problem, it is hard to backpropagate gradients. It poses a great challenge in initializing the policy network in the case of deep actor-critic methods, which requires a differentiable computation graph. To tackle this problem, we follow the re-parameterization mechanism, which enables computing the derivatives of stochastic models, to estimate gradients of categorical stochastic elements.

In our traffic signal control scenario with the discrete action space, the idea of categorical re-parameterization with Gumbel-Softmax [5] is widely applied. It is based on the Gumbel-Max trick [5], which provides a simple way to draw sampled actions from a categorical distribution with class probabilities $\pi(\cdot|s)$:

$$\mathbf{a}_{\text{hard}} = \text{one-hot}(\text{argmax}_i[g_i + \pi(a_i|s)]) \tag{7}$$

where $\mathbf{g} = -\log(-\log(\mathbf{u})), \mathbf{u} \sim \text{Uniform}(0, 1)$.

As *argmax* is not differentiable, we just replace the operation with a $softmax$ version:

$$\mathbf{a}_{\text{soft}} = \text{softmax}\big((g_i + \pi)/\tau\big) \tag{8}$$

Under the Gumbel-Softmax distribution, the previous sampling process becomes a deterministic function which allows us to compute its gradient estimation with low variance. Then regarding the training as a classification problem, we can easily define the loss of the policy network where $\mathbf{a}_D$ is the action of demos:

$$L_{\text{pre}}(\theta_\pi) = \text{Cross-Entropy}\big(\mathbf{a}_{\text{soft}}, \mathbf{a}_D\big) \tag{9}$$

4.4 Critic Training from Demonstrations

In this section, we introduce how to train the critic through behavior cloning losses, i.e., supervised losses with special designs.

Following DQfD [3], behavior cloning losses are composed of a 1-step TD loss for the action-value function, an n-step TD loss, a large margin classification loss, and an L2 regularization loss.

The 1-step TD loss is described in Equation 3 while the n-step TD loss is a decomposition of the target value using n-step returns. Taking n-step loss into account benefits expert's information flowing to earlier states and makes the best use of demonstrations.

The margin classification loss is crucial for training with demonstrations. Those demonstrations are only a small part of the whole exploration space. Despite their higher Q values than other state-action pairs, we cannot guarantee the training process will take place as expected. The Q function cannot evaluate unobserved state-action pairs and it is likely that the network would update towards the highest value of those pairs rather than the demonstrated one. Therefore, a large margin classification loss is necessary:

$$L_{\text{margin}}(\theta_Q) = \max_a \big[Q(s, a) + l(a_D, a)\big] - Q(s, a_D) \tag{10}$$

In this equation, $l(a_D, a)$ is a margin function that is set to 0.8 when $a \neq a_D$ and 0 otherwise. With this loss, we constrain the Q value of the demonstrator's action to be a margin higher than other actions.

Finally an L2 regularization loss is included to alleviate the overfitting issue due to the relatively small number of demonstrations.

5 EXPERIMENT

5.1 Experiment Settings

In this paper, we conduct extensive experiments in a new simulator CityFlow [14]. The agent can obtain states of the environment like the number of vehicles through flexible APIs. In the meanwhile, the simulator can execute actions from the agent to control traffic signals accordingly. Each green signal is followed by a three-second yellow signal and two-second all red time by convention. Codes of this paper are provided at https://github.com/xyh97/DemoLight.

5.2 Datasets

We use real-world datasets of three different cities. Raw data are collected from surveillance cameras in various intersections. We process them to extract one-hour traffic flows for a single intersection, which is randomly sampled from traffic-jammed or clear ones with probability 0.5. In a dataset, each vehicle is described as (o, t, d), where o is the origin, t is the departing time, and d is the destination. o and d are both on the road network.

5.3 Training Details

5.3.1 *Parameter Selection.*

In our experiments, Adam is chosen with learning rate 10^{-3} determined by the grid search. The discount factor γ is set to be 0.8. 1800 demonstrations are collected from SOTL to initialize the actor and the critic. Specifically, thresholds of SOTL are selected by brute force and we pick five sets of parameters that perform best to generate demonstrations. Function approximators Q and π are deep neural networks with ReLU activations and they are updated every simulation step. The temperature τ in the actor and the L2 regularization coefficient in the critic are 1 and 10^{-5} respectively. In addition, to avoid changing traffic light frequently, we constrain each phase remains at least ten seconds. It is for drivers to respond and stop and for the algorithm to choose phases flexibly.

5.3.2 *Network Architecture.*

For both networks of the actor and the critic, we use a 2-layer structure with 20 hidden neurons per layer. In addition, we use a target network [9] when training the critic to stabilize performance. Prioritized experience replay [7] is applied in the training process of the critic with demonstrations.

5.4 Results

5.4.1 *Effectiveness.*

We compare DemoLight with a conventional traffic control method SOTL [2], a deep Q-learning method LIT [16], and a state-of-the-art RL method advantage actor-critic (A2C) [13]. In accordance with existing studies in the transportation field, **travel time** is selected as an evaluation metric. This is defined as average time vehicles spend passing the intersection and is frequently used in the problem of traffic signal control.

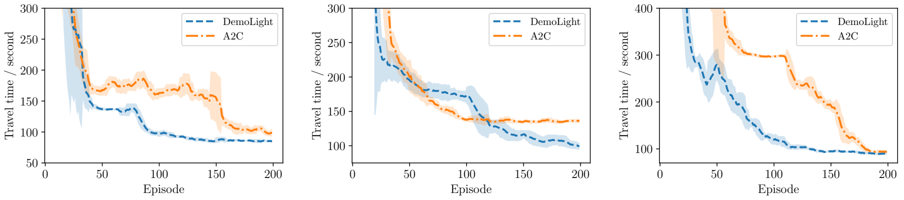
Overall results are reported in Table 1. As is expected, performance of LIT is inferior to that of actor-critic, reconfirming rationality of choosing A2C as our basis. In addition, from this table we can see clearly that our algorithm DemoLight achieves best performance in travel time for each intersection in three cities. Specifically, under some circumstances such as #1 and #5 in the city B where vehicle volume is quite large, learning directly even cannot beat the tradition method SOTL. But with demonstrations incorporated, performance can be improved significantly by over 60%. On the other hand, not much improvement is achieved in some intersections with relatively small traffic. This phenomenon mainly results from different traffic volumes. Specifically, in the City B, traffic volume at #2 is 1671 per hour while the number is over 2000 at #3. Extra vehicles will trap the agent in a congested state frequently, making exploration much harder.

5.4.2 *Efficiency.*

Contrary to training, traffic volume hardly influences time consumption of demo collection and initialization

Table 1: Overall performance. Travel time is reported in the unit of second.

Model	City A			City B					City C		
	1	2	3	1	2	3	4	5	1	2	3
SOTL [2]	102.76	179.41	248.73	248.12	153.43	165.38	123.73	269.64	89.36	149.49	72.11
LIT [16]	103.31	122.88	154.77	346.30	146.88	139.83	104.84	551.65	93.95	130.11	48.43
A2C [1]	85.48	98.36	136.01	380.64	92.14	135.93	77.56	517.77	76.38	96.74	46.83
DemoLight	**76.64**	**85.16**	**85.93**	**116.10**	**92.02**	**97.88**	**76.39**	**183.70**	**72.44**	**91.26**	**43.04**
Improvement	10.34%	13.42%	36.82%	69.50%	0.13%	27.99%	1.51%	31.87%	5.30%	5.66%	8.09%



(a) City A #2 (c) City B # 3 (e) City C #2

Figure 1: Convergence speed comparison among five intersections.

and they can be considered as an ignorable constant cost. It guarantees scalablity of our method to process scenarios with large traffic. Hence, we only focus on time cost of training. Despite the same computational complexity of DemoLight as A2C, our method greatly reduces exploration space, thus accelerating training. We take three intersections as representative examples to show efficiency of our method. In Figure 1, our method always outperforms A2C. We can observe that curves of DemoLight descends faster than those of A2C and our method converges to a lower travel time.

5.4.3 Learnt Policy. As shown in Figure 2, the percentage of vehicles in four directions synchronizes well with green time ratio of DemoLight. It follows an intuitive principle of a good policy that more green time should be allocated to larger traffic.

5.5 Ablation Study

In this section, we perform an ablation study to measure the importance of both training the actor and the critic with demonstrations.

We evaluate results from only initializing actor and only initializing critic. From Figure 3, these two ablations perform better than learning from scratch. But improvement of actor with demos is not obvious compared with that of critic. One possible explanation is that transforming a stochastic policy into a deterministic one may cause training errors. Besides, it is still training both components under guidance of demonstrations that achieves best travel time.

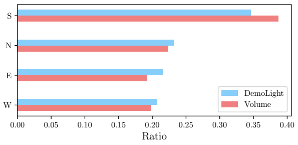


Figure 2: Learnt policy.

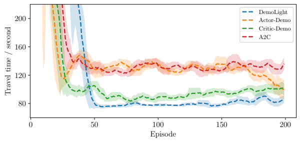


Figure 3: Ablation study.

6 CONCLUSION

In this paper, we proposed DemoLight to further improve RL in traffic signal control with demonstrations. It speeds up training and converges to a better result. For future work, extension to complex traffic scenarios and an off-policy structure are potential directions.

ACKNOWLEDGMENTS

The work was supported in part by NSF awards #1652525 and #1618448. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

REFERENCES

[1] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. 2019. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems* (2019).

[2] Seung-Bae Cools, Carlos Gershenson, and Bart D'Hooghe. 2013. Self-organizing traffic lights: A realistic simulation. In *Advances in applied self-organizing systems*. Springer, 45–55.

[3] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, et al. 2018. Deep Q-learning from demonstrations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[4] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*. 4565–4573.

[5] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[6] Ahmed H. Qureshi, Byron Boots, and Michael C. Yip. 2019. Adversarial Imitation via Variational Inverse Reinforcement Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HJlmHoR5tQ

[7] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952* (2015).

[8] van der Pol et al. 2016. Coordinated Deep Reinforcement Learners for Traffic Light Control. NIPS.

[9] Hado van Hasselt, Arthur Guez, and David Silver. 2015. Deep Reinforcement Learning with Double Q-learning. *arXiv preprint arXiv:1509.06461* (2015).

[10] Matej Večerík, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. 2017. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817* (2017).

[11] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2019. A Survey on Traffic Signal Control Methods. *arXiv preprint arXiv:1904.08117* (2019).

[12] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2496–2505.

[13] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.

[14] Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. 2019. CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. In *The World Wide Web Conference*. ACM, 3620–3624.

[15] Guanjie Zheng, Yuanhao Xiong, Xinshi Zang, Jie Feng, Hua Wei, Huichu Zhang, Yong Li, Kai Xu, and Zhenhui Li. 2019. Learning Phase Competition for Traffic Signal Control. *arXiv preprint arXiv:1905.04722* (2019).

[16] Guanjie Zheng, Xinshi Zang, Nan Xu, Hua Wei, Zhengyao Yu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019. Diagnosing Reinforcement Learning for Traffic Signal Control. *arXiv preprint arXiv:1905.04716* (2019).