DCT: Dual Channel Training of Action Embeddings for Reinforcement Learning with Large Discrete Action Spaces

Extended Abstract

Pranavi Pathakota
TCS Research
Mumbai, India
p.prانavи@tcs.com

Hardik Meisheri
TCS Research
Mumbai, India
hardik.meisherи@tcs.com

Harshad Khadilkar
TCS Research
IIT Bombay
Mumbai, India
harshad.khadilkar@tcs.com
harshадk@iitb.ac.in

ABSTRACT
The ability to learn robust policies while generalizing over large discrete action spaces is an open challenge for intelligent systems, especially in noisy environments that face the curse of dimensionality. In this paper, we present a novel framework to efficiently learn action embeddings that simultaneously allow us to reconstruct the original action as well as to predict the expected future state. We describe an encoder-decoder architecture for action embeddings with a dual channel loss that balances between action reconstruction and state prediction accuracy. We use the trained decoder in conjunction with a standard reinforcement learning algorithm that produces cleaner action embeddings, and the improved representations help learn better policies with earlier convergence.

KEYWORDS
Reinforcement Learning; Self-Supervised learning; Planning and Navigation; Recommender Systems

ACM Reference Format:

1 INTRODUCTION
Reinforcement learning (RL) has had significant recent success in applications such as games and robotics [7, 10]. However, real-world problems that involve a large number of discrete action choices are still very challenging for traditional RL algorithms. Examples include scenarios such as recommendation systems [1], supply chains [9], complex high fidelity games [2, 13] resource management at scale in data centers [5, 8], investment management [6], where large action spaces are handled indirectly using pre- or post-processing heuristics. The key challenge is with exploring large action spaces sufficiently well to arrive at optimal policies. Furthermore, hand-crafted heuristics for mapping RL outputs to actions become intractable as the number of actions increases.

Recently, the success of state embeddings for complex state spaces has inspired studies on the use of action embeddings along similar lines [4]. The key idea is to learn the RL policy not over raw actions, but over action representations in a low dimensional embedding space. If actions with similar effects are grouped close together in the embedding space, the efficiency of exploration is greatly improved. It stands to reason that the better the action representations, the better the chance of reaching good policies.

In this paper, we present an architecture to efficiently learn action embeddings in low dimensional space. We force the embeddings to be rich by imposing the dual task of learning the effect of actions as well as predicting future states. We show experimentally that this helps the RL agents learn better policies in scenarios with large action spaces. We build upon work of Chandak et al. [3] and Pritz et al. [11] and provide a generalized framework for learning embeddings which is not only efficient in encoding transition dynamics between states but also helps in decoding those actions (Fig. 1).

The main contributions of our work are as follows:

- We propose a new architecture for an action encoder-decoder model which results in a better representation of action embeddings by jointly training encoder and decoder for action reconstruction and next state prediction.
- We present extensive experimentation over a noisy maze environment with up to 212 unique actuator actions to validate our model and compare it with previous work and a traditional off policy RL algorithm (DQN).
- We also demonstrate the effectiveness of our algorithm in recommender systems, outperforming baselines on a real-world fashion e-commerce dataset.

2 METHODOLOGY
In this section, we propose a model to efficiently learn action embeddings using an encoder-decoder architecture with Dual Channel Training (DCT). We focus on the explanation of action embeddings $E_t$ from Fig. 1, but an analogous method can be used to train state embeddings $X_t$. Following this step, we can use any off-the-shelf model-free RL algorithm to train the internal policy $\pi_i$.

1For state embeddings, we only use the gradient from the next-state prediction loss.
We use DDPG [12] for most experiments in this paper. The encoder-decoder model is jointly trained using DCT, with loss gradients flowing through both $f$ and $g$. The generic loss function is given by,

$$L = L_1(g(X_t, E_t), S_{t+1}) - \eta \times \frac{1}{N} \times \log P(A_t|f, E_t),$$

(1)

where $L_1$ is a metric to measure the state prediction loss, $N$ is the number of actions, and $P(A_t|f, E_t)$ is the softmax probability of decoding the embedding $E_t$ to the correct action $A_t$, as parameterised by $f$. The multiplier $\eta$ is a hyperparameter used for trading off the importance between the two loss terms. Complete details can be found in the paper.²

3 RESULTS: NAVIGATION IN 2D MAZE

We present results on a 2-D Maze environment, which an agent with a number of directional actuators is expected to navigate.

3.1 Training Results

Figure 2 presents the training results for $2^{11}$ actions (11 actuators). We can observe that DCT outperforms all the other baseline algorithms, converging earlier and reaching a higher reward.

Table 1 presents results over various actions. We can see that even with just 500 episodes (Fig. 2), DDPG over DCT embeddings is able to learn consistently across the actions.

<table>
<thead>
<tr>
<th>Action</th>
<th>DQN</th>
<th>PG-RA</th>
<th>JSAE</th>
<th>DCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^6$</td>
<td>97.38</td>
<td>97.46</td>
<td>98.76</td>
<td>72.69</td>
</tr>
<tr>
<td>$2^{10}$</td>
<td>4.05</td>
<td>0.46</td>
<td>0.33</td>
<td>54.03</td>
</tr>
<tr>
<td>$2^{12}$</td>
<td>-22.03</td>
<td>45.91</td>
<td>82.25</td>
<td>98.39</td>
</tr>
</tbody>
</table>

4 RESULTS: RECOMMENDER SYSTEMS

We present results from a recommender system task as a second experiment aiming to suggest meaningful products that result in actual purchases for the user.

4.1 Baselines and RL training

Figure 3: Training curves for the proposed method (DCT) and two baselines, over 5 random seeds.

5 CONCLUSION

Finally, we can conclude from the experiments that DCT is able to learn across a different number of actions consistently. This is validated across 2 diverse environment of navigation and recommender systems. As a part of the investigation, we have also looked at how loss coefficient $\eta$ affect the structure of embedding.
REFERENCES


