Paper Critique

Anirud N, CE21B014

Course: DA7400, Fall 2024, IITM
Paper: Deep Laplacian-based Options for Temporally-Extended Exploration
Date: 4 September 2024

Make sure your critique addresses the following points:

1. The problem the paper is trying to address
2. Key contributions of the paper
3. Proposed algorithm/framework
4. How the proposed algorithm addressed the described problem

Note: Be concise with your explanations. Unnecessary verbosity will be penalized. Please don't exceed 2 pages.

---

1 Problem Statement

The paper addresses the problem of selection of proper/good action for better learning. It aims to address the question of temporally extended options : How should the options be discovered. The paper looks specifically at diffusion models to encode the flow of information in the environment, Recent Laplacian methods have been limited to assumptions such as :

- Tabular domains where Laplacian matrix was given or could be estimated
- Performing eigen decomposition is computationally traceable.
- Value function could be learnt exact

But these are not scalable

2 Key Contributions

- This paper extends the Laplacian-based options framework to a deep function approximation setting
- Validates the approximation of laplacian objectives for option discovery - reduced computation
- Fully online algorithm for discovering options
- Deep Covering eigen values to deal with option termination conditions

,

3 Proposed Algorithm/Framework

This paper uses Double DQN network - to prevent overestimation

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \alpha \left[Y_t^{(n)} - Q_{\boldsymbol{\theta}_t}(S_t, A_t) \right] \nabla_{\boldsymbol{\theta}_t} Q_{\boldsymbol{\theta}_t}(S_t, A_t) \tag{1}$$

$$Y_t^{(n)} = R_{t+1}^{(n)} + \gamma^n Q_{\boldsymbol{\theta}_t^-}(S_{t+n}, \arg\max_{a' \in \mathcal{A}} Q_{\boldsymbol{\theta}_t}(S_{t+n}, a')),$$

where $R_{t+1}^{(n)} \sum_{i=1}^{n-1} \gamma^i R_{t+i+1}$, and $\boldsymbol{\theta}_t^-$ denotes the parameters of a duplicate network, which are updated less often for stability purposes.

3.1 Covering Eigneoptions

Uses representation driven option discovery. First the agent collect samples from the environment and use it to built a representation. Using this, intrinsic rewards are defined, and options are learnt to maximise the intrinsic rewards. Laplacian encodes the environment's graph - nodes as states and edges are transition. Eigenfunctions help in better representation of the environment. $r^{\mathbf{e}_i}(s, s') = \mathbf{e}_i(s') - \mathbf{e}_i(s)$. INtrinsic rewards The option terminates in every state s where $q_\pi^{\mathbf{e}_i}(s, a) \leq 0$ for all $a \in \mathcal{A}$, where $q_\pi^{\mathbf{e}_i}$ is defined w.r.t. $r^{\mathbf{e}_i}(\cdot, \cdot)$.

3.2 Approximate Laplacian Options

$$\min_{\boldsymbol{f}_1, \ldots \boldsymbol{f}_d} \quad \sum_{i=1}^{d} c_i \boldsymbol{f}_i^\top L \boldsymbol{f}_i \qquad\qquad \text{s.t.} \quad \boldsymbol{f}_i^\top \boldsymbol{f}_j = \delta_{ij} \forall i, j$$

$$G(f_1, ..., f_d) = \frac{1}{2\pi} \left[\sum_{i=1}^{d} \sum_{k=1}^{i} \big(f_k(s) - f_k(s')\big)^2 \right] + \beta \sum_{i=1}^{d} \sum_{j=1}^{i} \sum_{k=1}^{i} \Big({}_\pi\big([f_j(s) f_k(s) - \delta_{jk})\big] \Big)^2 \tag{2}$$

where β is the Lagrange multiplier and the coefficients are defined as $c_i = d - i + 1$, $\{\boldsymbol{f}_i\}_{i=1}^d$ are approximations to the d smallest eigenfunctions of the Laplacian, c_i are their associated coefficients.

3.3 Two phase

Two phases - agent interaction with the environment for learning the representation and then fixing options for an agent to explore and maximize return. The option termination is defined as uniformly random with a probability of $1/D$ where D is the expected length of the option. It uses DDQN with n-step targets to maximize rewards. The algorithm is evaluated on pixel-based versions of the environment.

3.4 Single Cycle

Random initialization of the laplacian representation, set of options, and DDQN learner. The options maximises the random intrinsic reward. Then we go on to adjust these, and Laplacian becomes more accurate correspondingly.

4 Conclusions and Results

This paper extended the tabular approach into a completely online approach with a deep function approximation. The algorithm performs better than many SOTA baselines in different environments. It shows many benefits in 3D navigation tasks - it could learn consistenly and accumlate positive rewards. The algorithm was also tested on nonstationary environments, where it gave good results. There is scope for improvements, such as Credit assignment-based learning of option value functions. Use of options for planning. Incorporating auxiliary task effect for representation learning.