A Laplacian Framework for Option Discovery in Reinforcement Learning

Marlos C. Machado, Marc G. Bellemare, Michael Bowling

***Abstract*—The paper addresses the task of Option Discovery in RL using Proto-Value functions. They introduce the concept of *eigenpurposes* for finding options that act at different time scales, and traverses the state space in principle directions, which also aids in exploration. Their options are discovered independent of the reward distribution which makes them suitable for transfers.**

I. Terminology

1) *Diffusion Model* : L : Captures Information flow in a graph. It can be defined as :
   - *Combinatorial graph Laplacian matrix*, $L = D - A$
   - *Normalized graphLaplacian*,$L = D^{-\frac{1}{2}}(D-A)D^{-\frac{1}{2}}$

   A = Adjacency matrix
   D : Diagonal matrix with entries = row sum of A
2) *Proto-value functions (PVF)* : $\boldsymbol{e} \in \mathbb{R}^{\mathcal{S}}$: Learned representations capturing large scale temporal properties of the environment. They are defined as the eigen vectors obtained after the eigen decomposition of L.
3) *Eigenpurpose* : $r_i^{\boldsymbol{e}}(s, s')$: Intrinsic reward function of PVF. It is given by, $r_i^{\boldsymbol{e}}(s, s') = \boldsymbol{e}^T(\boldsymbol{\phi}(s') - \boldsymbol{\phi}(s))$
   $\boldsymbol{\phi}(s)$: Feature representation of state s
4) *MDP* : $\mathcal{M}_i^{\boldsymbol{e}}$: MDP associated with the eigenpurpose given by, $\mathcal{M}_i^{\boldsymbol{e}} = \langle \mathcal{S}, \mathcal{A} \cup \{\perp\}, r_i^{\boldsymbol{e}}, p, \gamma \rangle$
   $\perp$: Action terminate
5) *Eigenbehaviour* : $\chi^{\boldsymbol{e}} : \mathcal{S} \to \mathcal{A}$: Optimal policy with respect to $r_i^{\boldsymbol{e}}$. It is given by, $\chi^{\boldsymbol{e}}(s) = argmax_{a \epsilon \mathcal{A}} q_*^{\boldsymbol{e}}(s, a)$
   Note : $q_\chi(s, \perp) = 0 \ \forall \ \boldsymbol{e}$
6) *Eigenoption* : $o =< \mathcal{I}_o, \pi_o, \mathcal{T}_o >$: Option defined by $r_i^{\boldsymbol{e}}$ with corresponding eigenbehaviour (policy) $\chi^{\boldsymbol{e}}$.
   - $\mathcal{T}_o$: $q_\chi^{\boldsymbol{e}}(s, a) \leq 0 \ \forall \ a \epsilon \mathcal{A}$
   - $\mathcal{I}_o$: $\forall$ state in which $\exists \ a \epsilon A$ s.t $q_\chi^{\boldsymbol{e}}(s, a) > 0$

   Option policy : $\pi^{\boldsymbol{e}}(s) = argmax_{a \epsilon A \cup \{\perp\}} q_\pi^{\boldsymbol{e}}(s, a)$
7) *Diffusion Time* : Expected number of steps required to navigate between 2 states randomly chosen in the MDP during a random walk.

II. Essence of the paper

PVFs are task independent (does not depend on the reward distribution) and can capture symmetries and bottlenecks. They can be interpreted as basis functions and are defined over the entire state space.

In order to construct the MDP using adjacency matrix, trajectories are sampled from the environment. The unseen transitions are added to *incidence matrix* T as the difference between the current and previous observations ($\phi(s') - \phi(s)$, where $\phi(s)$ is one-hot encoded in the tabular case). Using SVD, $T = U \Sigma V^T$, where the columns of V are the right-eigenvectors of T which are used to generate the *eigenpurposes*. In the tabular case, orthonormal eigenvectors of L are the columns of V^T = eigenvectors of $T^T T$ (= $2L$). Hence, *eigenoptions* can be generalised to linear function approximation.

According to PVF theory, the "smoothest" eigenvectors, corresponding to the smallest eigenvalues. Since the same logic applies to eigenpurposes and eigenoptions (since $L \propto D - A$), the smallest eigen value is considered for the new MDP. $\mathcal{M}_i^{\boldsymbol{e}}$ (refer terminology)is defined with intrinsic reward function (*eigenpurpose*) $r_i^{\boldsymbol{e}}(s, s')$. Now we have a familiar RL problem where *eigenpurpose* defines the intrinsic reward function; giving a *purpose* to the agent to maximise their discounted sum. The state-value function $v_\pi^{\boldsymbol{e}}(s)$ is defined as the expected cumulative discounted intrinsic reward obtained by starting at state s and following policy π until termination and the action-value function $q_\pi^{\boldsymbol{e}}(s, a)$ as expected cumulative discounted intrinsic reward obtained by starting at state s with an action a and following policy π until termination. Policy iteration is used for finding the optimal policy $\pi_*^{\boldsymbol{e}}$ otherwise known as *eigenbehavior*. In function approximation setting, the matrix of transitions does not contain all possible transitions and hence, $a' = argmax_{b \epsilon \mathcal{A}} \int_{s'} p(s'|s, b) r_i^{\boldsymbol{e}}(s, s')$.

In a finite state space MDP, $\mathcal{T}_o$ is a non-empty set implying that for any *eigenoption*, there is always at least one state in which it terminates. Also, *eigenoptions* operate at different timescales.

Advantages of *eigenoptions* :

- Improved Exploration:Reduce the expected number of steps for navigating between states
- Faster accumulation of rewards:Options speed up learning

It was noticed that fewer options might degrade the agents performance(than a random walk using primitive actions) while enough number of options enhance learning by reducing the diffusion time considerably. These options operate in a higher level of abstraction, and given that the agent has access to several options and they are available in most parts of the state space, they enhance and improve exploration. Since multiple options are available in every state, they can be easily sequenced. Fewer options may reduce the deviation from the options trajectory(Eg:by undoing primitive actions), and can hurt exploration. Naively adding just bottleneck options can also hinder exploration strategies. The ideal number of options for the agent is a model selection problem.

References

[1] Marlos C Machado, Marc G Bellemare, and Michael Bowling. "A laplacian framework for option discovery in reinforcement learning". In: *arXiv preprint arXiv:1703.00956* (2017).