A Distributional Perspective on Value Function Factorization Methods for Multi-Agent Reinforcement Learning

Wei-Fang Sun[1], Cheng-Kuang Lee[2], and Chun-Yi Lee[1]

Elsa Lab, Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan[1]

NVIDIA Corporation and NVIDIA AI Technology Center (NVAITC) [2]

Abstract

Value function factorization methods (e.g., VDN, QMIX) for multi-agent reinforcement learning (MARL) offer promising performance in the StarCraft Multi-Agent Challenge (SMAC). In MARL settings, the environments are highly stochastic due to the partial observability of each agent and the continuously changing policies of the other agents. In order to deal with the above issues, distributional reinforcement learning (RL) is a potential solution that has been empirically proven successful in a wide range of single-agent domains. However, distributional RL has not been applied to value function factorization methods in MARL domains to decompose the joint return distribution. In this work, we bridge the gap between distributional RL and value function factorization by proposing the Distributional Value Function Factorization (DFAC) framework as a practical implementation to generalize expected value function factorization methods to their distributional variants. DFAC extends the individual utility functions from deterministic variables to random variables, and models the quantile function of the total return as a quantile mixture.

Background

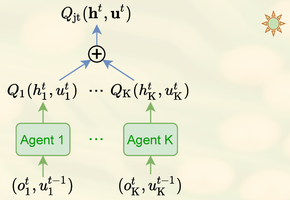
Value-based Methods for Fully Cooperative MARL

Independent Q-Learning (IQL) is the simplest value-based learning method for MARL, where each agent attempts to maximize the total rewards separately. This causes unstationarity due to the changing policies of the other agents and may not converge. Thus, value function factorization methods are introduced to enable centralized training of factorizable tasks based on the IGM (Individual-Global-Max) condition, where optimal individual actions result in the optimal joint action of the group of agents:

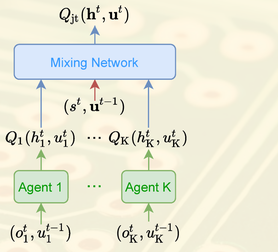
$$\operatorname{argmax}_{\mathbf{u}} Q_{\text{jt}}(\mathbf{h}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u_1} Q_1(h_1, u_1) \\ \vdots \\ \operatorname{argmax}_{u_K} Q_K(h_K, u_K) \end{pmatrix}$$

The previous VDN and QMIX methods assume additional premises: additivity and monotonicity, respectively, to simplify the tasks:

$$\text{(Additivity) } Q_{\text{jt}}(\mathbf{h}, \mathbf{u}) = \sum_{k=1}^{K} Q_k(h_k, u_k)$$
$$\text{(Monotonicity) } Q_{\text{jt}}(\mathbf{h}, \mathbf{u}) = M\big(Q_1(h_1, u_1), \dots, Q_K(h_K, u_K)\big)$$
$$\text{where } \frac{\partial M}{\partial Q_k} \geq 0, \forall k \in \{1, \dots, K\}$$



(VDN) Sunehag et al., 2018



(QMIX) Rashid et al., 2018

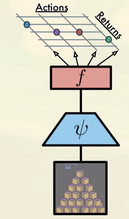
Distributional Reinforcement Learning

Distributional RL methods have been proved empirically to outperform expected RL methods in various single-agent RL (SARL) domain. The distributional Bellman operator T^π is proved to have a contraction in p-Wasserstein distance $W_p, \forall p \in [1, \infty)$:

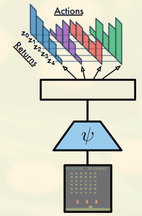
$$T^\pi Z(s,u) \overset{D}{=} R(s,u) + \gamma Z(s',u')$$
$$W_p(X,Y) = \left(\int_0^1 |F_X^{-1}(\tau) - F_Y^{-1}(\tau)| d\tau\right)^{1/p}$$

Implicit Quantile Network (IQN) is by far the most light-weight distributional RL algorithm. It models the quantile function of the return distribution, and can efficiently approximate the expectation by inverse distribution sampling $[\tau_i \sim U([0,1])]_{i=1}^N$, as follows:

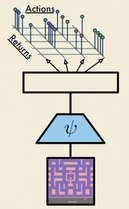
$$Q(s,u) = \mathbb{E}[Z(s,u)]$$
$$= \int_0^1 F^{-1}(s,u;\tau) d\tau \approx \frac{1}{N} \sum_{i=1}^{N} F^{-1}(s,u;\tau_i)$$



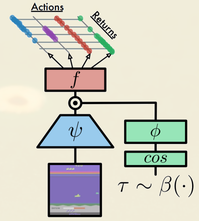
(DQN) Mnih et al., 2015



(C51) Bellemare et al., 2017



(QR-DQN) Dabney et al., 2018



(IQN) Dabney et al., 2018

Methodology

DFAC Framework and Mean-Shape Decomposition

The naive generalization of the distributional form of IGM does not satisfy IGM in general. Thus, we introduced the mean-shape decomposition to separate the approximation of the mean and the shape of the return distribution:

$$Z_{\text{jt}} = \mathbb{E}[Z_{\text{jt}}] + (Z_{\text{jt}} - \mathbb{E}[Z_{\text{jt}}]) = Z_{\text{mean}} + Z_{\text{shape}}, \text{ where}$$

- $Z_{\text{mean}} = \Psi\big(s, Q_1(h_1,u_1), \dots, Q_K(h_K,u_K)\big)$
- $Z_{\text{shape}} = \Phi\big(s, Z_1(h_1,u_1), \dots, Z_K(h_K,u_K)\big)$

$$= Z_{\text{state}}(s) + \sum_{k=1}^{K} \beta_k(s)(Z_k(h_k,u_k) - Q_k(h_k,u_k))$$

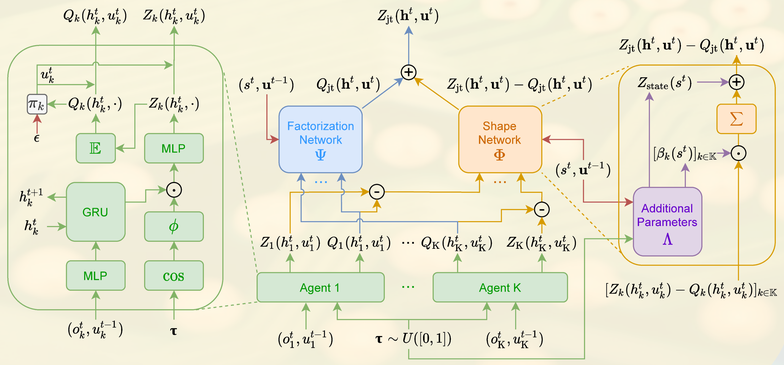
A Practical Implementation with Quantile Mixture

The factorization network Ψ can be any expected value function factorization method, while the shape network Φ can be approximated by a quantile mixture:

$$Z_{\text{shape}}(\tau) = F^{-1}_{\text{state}}(s;\tau) + \sum_{k=1}^{K} \beta_k(s)(F_k^{-1}(h_k,u_k;\tau) - Q_k(h_k,u_k))$$

The Architecture of the Distributional Value Factorization (DFAC) Framework

The Factorization Network Ψ can be any differentiable factorization function (e.g., VDN, QMIX), while the Shape Network Φ if defined by a quantile mixture:

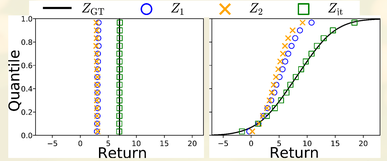


(The architecture of the DFAC framework)

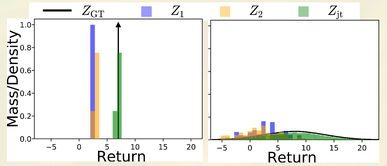
Experimental Results

Factorization of a toy problem

We demonstrated DFAC's ability to factorize the stochastic return of a toy problem with different implementation of the Shape Network Φ.



(Factorization by Quantile Mixture)



(Factorization by 1D Convolution)

Performance on the StarCraft Multi-Agent Challenge (SMAC)

All experiments are conducted on NVIDIA V100 GPUs. We generalize the two baselines: VDN and QMIX to their distributional variant: DDN and DMIX, respectively. The results showed that DDN and DMIX can achieve outstanding performance in SMAC:

Super Hard Map	IQL	VDN	QMIX	DIQL	DDN	DMIX
6h_vs_8z	0.00%	0.00%	8.81%	0.00%	**83.52%**	68.75%
3s5z_vs_3s6z	7.67%	90.91%	65.06%	29.83%	**94.60%**	90.62%
MMM2	69.32%	87.78%	92.33%	83.52%	**97.44%**	95.17%
27m_vs_30m	1.70%	64.20%	86.08%	12.50%	**94.60%**	86.08%
corridor	83.10%	85.23%	4.26%	92.05%	**95.45%**	90.06%
6h_vs_8z	13.96	15.49	14.02	14.98	**19.32**	17.81
3s5z_vs_3s6z	15.48	19.77	20.06	17.42	20.68	**20.78**
MMM2	17.47	19.32	19.45	19.21	**21.06**	19.69
27m_vs_30m	13.95	18.49	19.46	15.16	**19.72**	19.40
corridor	19.30	19.38	13.44	19.57	**19.97**	19.61

(Median win rates and average scores in SMAC)

Questions?
j3soon@gapp.nthu.edu.tw
cylee@cs.nthu.edu.tw

Demo Video QR-code