Reinforcement Learning for Adaptive Routing

Siliang Zeng

CUHK-Shenzhen
Statistical Science

116010279@link.cuhk.edu.cn

April 6, 2019

Overview

Multi-Agent Reinforcement Learning for Adaptive Routing: A Hybrid Method using Eligibility Traces

Siliang Zeng[1] , **Xingfei Xu**[1] , **Yi Chen**[1,2]
[1]The Chinese University of Hong Kong, Shenzhen
[2]Shenzhen Research Institute of Big Data
{116010279, 116010252}@link.cuhk.edu.cn, yichen@cuhk.edu.cn

Overview

1. Value-based RL for Network Routing
    1. Q-routing [Boyan and Littman, 1994]
2. Policy-based RL for Network Routing
    1. Online optimization of the average reward: OLPOMDP [Tao *et al.*, 2001]
    2. Gradient Ascent Policy Search [Peshkin and Savova, 2002]
    3. Multi-Agent Hybrid of the Q-learning and the actor-critic thinking [Our work, 2019]

Notation

1. The cumulative discounted reward:

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$$

2. Q-function: $Q^{\pi}(s, a) = E_{\pi}[G_t | s_t = s, a_t = a]$
3. Bellman Equation: $Q^*(s_t, a) = E[r_t + \gamma \max_{a'} Q^*(s_{t+1}, a')]$

Problem Formulation: Network Routing

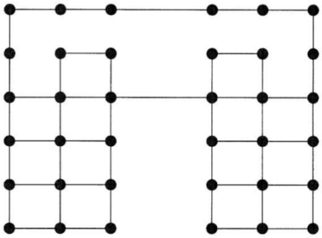


Figure 1: The irregular 6 × 6 grid topology

- Communication Networks: a set of nodes (routers) and links
- Routing: directs data packets from their source nodes toward their destination nodes through some intermedia nodes.

Problem Formulation: Network Routing

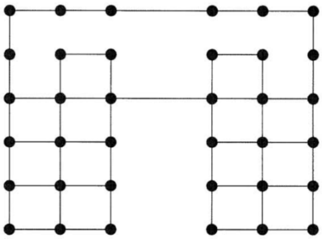


Figure 1: The irregular 6×6 grid topology

- Our objective: efficiently utilize the communication paths and minimize average packet delivery time.
- Packet delivery time: transmission delay and queue delay.

Problem Formulation: Network Routing

1. Single-Agent Reinforcement Learning for Network Routing
    1. Consider each router as an independent agent
    2. Each router in some sense behave selfishly to maximize its own profit without cooperation.
2. Multi-Agent Reinforcement Learning for Network Routing
    1. Consider the network system as a whole agent and update each router through distributed optimization.
    2. Multi-agent cooperation and coordination.

We consider network routing as a multi-agent, partially observable Markov decision process (POMDP).

Q-routing

- Fixed a router/agent, the state s is the destination of the first packet in its waiting buffer (queue) and the action a is one of its outgoing links.
- Supposing at a time step t, agent i chooses to send a packet with destination s through outgoing link a to next agent j, we use u_t^i to denote the queue delay, and use v_t^i to denote the transmission delay between two routers.
  Reward of agent i at time t: $r_t^i = -(u_t^i + v_t^i)$

Q-routing

- Each router maintains a two-dimensional lookup table, called Q-table, for all pairs of the outgoing link and the destination node.
- For the agent i, its Q-value $Q^i(s, a)$ is updated through

$$Q^i_{t+1}(s,a) = Q^i_t(s,a) + \alpha(r^i_t + \gamma \max_{a'} Q^j_t(s,a') - Q^i_t(s,a))$$

- The Q-routing scheme: each agent uses its Q-table to execute greedy action (greedy policy)

Q-routing: drawbacks

1. Q-routing is a deterministic policy:
   causes traffic congestion at high loads and doesn't distribute incoming traffic across the available links.
2. The lack of exploration and ϵ-greedy policy isn't suitable
   1. the network is continuously changing, thus the initial period of exploration never ends; and more significantly
   2. more significantly, random traffic has an extremely negative effect on congestion

Due to the drawbacks of value-based methods, we further consider policy-based reinforcement learning methods.

A hybrid of Q-learning and actor-critic thinking

1. Each router still maintains a Q-table as before. But actions are executed according to the parametrized policy.
2. For an agent, we use parameter $\theta_{sa} \in R$ to denote the preference for a state-action pair (s, a).
   The stochastic policy of an agent is parameterized by $\boldsymbol{\theta}$.

$$\pi(a|s, \boldsymbol{\theta}) := \frac{\exp(\theta_{sa})}{\sum_{a'} \exp(\theta_{sa'})}$$

Hybrid Method: How to update the policy parameters $\boldsymbol{\theta}$

1. Objective function: $J(\boldsymbol{\theta}) = \sum_s \mu(s) \sum_a Q^\pi(s,a)\pi(a|s,\boldsymbol{\theta})$
2. Policy Gradient Theorem:

$$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a Q^\pi(s,a) \nabla_{\boldsymbol{\theta}} \pi(a|s,\boldsymbol{\theta})$$

3. Generalized policy gradient theorem:

$$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a (Q^\pi(s,a) - b(s)) \nabla_{\boldsymbol{\theta}} \pi(a|s,\boldsymbol{\theta})$$

$$\sum_s \mu(s) \sum_a b(s) \nabla_{\boldsymbol{\theta}} \pi(a|s,\boldsymbol{\theta}) = \sum_s \mu(s) b(s) \sum_a \nabla_{\boldsymbol{\theta}} \pi(a|s,\boldsymbol{\theta}) = 0$$

Supplement: Proof of the Policy Gradient Theorem

With just elementary calculus and re-arranging terms we can prove the policy gradient theorem from first principles. To keep the notation simple, we leave it implicit in all cases that π is a function of $\boldsymbol{\theta}$, and all gradients are also implicitly with respect to $\boldsymbol{\theta}$. First note that the gradient of the state-value function can be written in terms of the action-value function as

$$\begin{aligned}
\nabla v_\pi(s) &= \nabla \left[\sum_a \pi(a|s) q_\pi(s,a) \right], \quad \text{for all } s \in \mathcal{S} && \text{(Exercise 3.15)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \nabla q_\pi(s,a) \Big] && \text{(product rule)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \nabla \sum_{s',r} p(s',r|s,a) \big(r + v_\pi(s') \big) \Big] \\
& && \text{(Exercise 3.16 and Equation 3.2)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) \nabla v_\pi(s') \Big] && \text{(Eq. 3.4)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) && \text{(unrolling)} \\
&\qquad \sum_{a'} \big[\nabla \pi(a'|s') q_\pi(s',a') + \pi(a'|s') \sum_{s''} p(s''|s',a') \nabla v_\pi(s'') \big] \Big] \\
&= \sum_{x \in \mathcal{S}} \sum_{k=0}^{\infty} \Pr(s \to x, k, \pi) \sum_a \nabla \pi(a|x) q_\pi(x,a),
\end{aligned}$$

Supplement: Proof of the Policy Gradient Theorem

$$
\begin{aligned}
\nabla J(\boldsymbol{\theta}) &= \nabla v_\pi(s_0) \\
&= \sum_s \left(\sum_{k=0}^{\infty} \Pr(s_0 \rightarrow s, k, \pi) \right) \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&= \sum_s \eta(s) \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&= \left(\sum_s \eta(s) \right) \sum_s \frac{\eta(s)}{\sum_s \eta(s)} \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&\propto \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s,a). \qquad \text{Q.E.D.}
\end{aligned}
$$

Hybrid Method: How to update the policy parameters $\boldsymbol{\theta}$

1. Generalized policy gradient theorem:

$$\nabla J(\boldsymbol{\theta}) \propto \sum_{s} \mu(s) \sum_{a} (Q^{\pi}(s, a) - b(s)) \nabla_{\boldsymbol{\theta}} \pi(a|s, \boldsymbol{\theta})$$

2. Replace $Q^{\pi}(s, a)$ by $G_{t:t+1} = r_t + \gamma \max_{a'} \hat{Q}_t(s_{t+1}, a')$ and choose $\max_a \hat{Q}_t(s_t, a)$ as the baseline term $b(s_t)$
3. The update rule:

$$\Delta\boldsymbol{\theta}_t = (G_{t:t+1} - \max_{a} \hat{Q}_t(s_t, a)) \nabla_{\boldsymbol{\theta}} \ln \pi(a_t|s_t, \boldsymbol{\theta})$$

Hybrid Method

1. To be specific, at time step t, the policy-table(policy parameters $\boldsymbol{\theta}^i$) and Q-table $\boldsymbol{\theta}^i$ of the agent i are updated as follows:

$$\boldsymbol{\theta}_{t+1}^i = \boldsymbol{\theta}_t^i + \beta \nabla \ln \pi(a_t|s_t, \boldsymbol{\theta}^i)\Big(r_t^i + \gamma \max_{a\prime} Q_t^j(s_t, a\prime) - \max_{a} Q_t^i(s_t, a)\Big)$$

$$Q_{t+1}^i(s,a) = Q_t^i(s,a) + \alpha(r_t^i + \gamma \max_{a'} Q_t^j(s,a') - Q_t^i(s,a))$$

2. According to the softmax rule, we have

$$\frac{\partial \ln \pi(a|s, \boldsymbol{\theta}^i)}{\partial \theta_{\dot{s}\dot{a}}^i} = \begin{cases} 1 - \pi(\dot{a}|\dot{s}, \boldsymbol{\theta}^i) & \text{if } \dot{s} = s, \dot{a} = a, \\ -\pi(\dot{a}|\dot{s}, \boldsymbol{\theta}^i) & \text{if } \dot{s} = s, \dot{a} \neq a, \\ 0 & \text{if } \dot{s} \neq s. \end{cases}$$

Multi-Agent Hybrid Method: Motivation

1. In Hybrid method, since each agent learns its policy by a local reward, all agents in some sense behave selfishly to maximize its own profit without cooperation.
2. We further develop the multi-agent hybrid method for multiagent systems. Provided a global feedback signal (global reward), the agents act independently but are able to learn cooperative behavior through limited information exchange.

Multi-Agent Hybrid Method: Motivation

Through introducing the eligibility traces and utilizing a global reward, we are able to handle the delayed reward and design a algorithm for the multi-agent system

Multi-Agent Hybrid Method: Algorithm Analysis

1. Eligibility: $\boldsymbol{e}_t = \nabla \ln \pi(a_t | s_t, \boldsymbol{\theta})$
2. Eligibility traces: $\boldsymbol{z}_t = \sum_{\tau=0}^{t} \rho^{t-\tau} \boldsymbol{e}_\tau$
   where ρ is a discount factor.
3. $\boldsymbol{z}_t$ is used to keep track of the past updates.

We first present our algorithm in the form of the single agent and then generalize it to multi-agent systems later.

Multi-Agent Hybrid Method: Algorithm Analysis

1. The update rule:

$$\begin{aligned}\Delta\boldsymbol{\theta}_t &= \Big(G_{t:t+1} - \max_a \hat{Q}_t(s_t, a)\Big)\,\mathbf{z}_t \\ &= \Big(r_t + \gamma \max_{a\prime} \hat{Q}_t(s_{t+1}, a\prime) - \max_a \hat{Q}_t(s_t, a)\Big)\,\mathbf{z}_t.\end{aligned}$$

2. The eligibility traces are updated as

$$\mathbf{z}_t = \rho\mathbf{z}_{t-1} + \mathbf{e}_t = \rho\mathbf{z}_{t-1} + \nabla_{\boldsymbol{\theta}} \ln \pi(a_t|s_t, \boldsymbol{\theta})$$

Multi-Agent Hybrid Method: Algorithm Analysis

To conduct the analysis of this algorithm, we first assume $\rho = \gamma$. Then the sum of $\Delta\boldsymbol{\theta}_t$ over time can be written as:

$$
\begin{aligned}
&\sum_{t=0}^{\infty} \Delta\boldsymbol{\theta}_t \\
=&\sum_{t=0}^{\infty} \left(r_t + \gamma \max_{a\prime} \hat{Q}_t(s_{t+1}, a\prime) - \max_{a} \hat{Q}_t(s_t, a)\right) \mathbf{z}_t \\
=&\sum_{t=0}^{\infty} (r_t + \gamma \max_{a\prime} \hat{Q}_t(s_{t+1}, a\prime) - \max_{a} \hat{Q}_t(s_t, a))(\sum_{\tau=0}^{t} \gamma^{t-\tau} \mathbf{e}_\tau) \\
=&\sum_{t=0}^{\infty} \mathbf{e}_t \sum_{\tau=t}^{\infty} \gamma^{\tau-t} (r_\tau + \gamma \max_{a\prime} \hat{Q}_\tau(s_{\tau+1}, a\prime) - \max_{a} \hat{Q}_\tau(s_\tau, a)) \\
=&\sum_{t=0}^{\infty} \mathbf{e}_t \left((\sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau) - \max_{a} \hat{Q}_t(s_t, a) \right) \\
=&\sum_{t=0}^{\infty} \mathbf{e}_t (G_t - \max_{a} \hat{Q}_t(s_t, a))
\end{aligned}
$$

Multi-Agent Hybrid Method: Algorithm Analysis

Assuming the policy converges, at time t the expected value $E_\pi[G_t]$ is deterministic given the policy parameters $\boldsymbol{\theta}$. Hence, we have

$$
\begin{aligned}
&\mathbb{E}_\pi[\mathbf{e}_t(G_t - \max_a \hat{Q}(s_t, a))] \\
=&\sum_{a_t} \pi(a_t|s_t, \boldsymbol{\theta}) \nabla \ln \pi(a_t|s_t, \boldsymbol{\theta})(G(s_t, a_t) - \max_a \hat{Q}(s_t, a)) \\
=&\sum_{a_t} \nabla \pi(a_t|s_t, \boldsymbol{\theta})(G(s_t, a_t) - \max_a \hat{Q}(s_t, a)) \\
=&\sum_{a_t} \nabla \pi(a_t|s_t, \boldsymbol{\theta}) G(s_t, a_t) \\
=&\nabla \sum_{a_t} \pi(a_t|s_t, \boldsymbol{\theta}) G(s_t, a_t) \\
=&\nabla \mathbb{E}_\pi(G_t)
\end{aligned}
$$

where $G(s_t, a_t)$ denotes the long-term return from time t after the agent executes action a_t at state s_t.

Multi-Agent Hybrid Method: Algorithm Analysis

1. From the above analysis which is based on the condition $\rho = \gamma$, we see that the policy of the agent is updated in a unbiased direction to increase the expectation of the discounted cumulative reward.
2. If the discount factor ρ equals 0, the policy parameters $\boldsymbol{\theta}$ are updated in the direction of the estimated gradient of the discounted cumulative reward. (lower variance)

When $\rho \in (0, \gamma)$, ρ controls the tradeoff between bias and variance of the estimated gradient.

Apply to Communicaton Network

1. Definition:
   1. S_t and A_t to denote the state and the joint action of the network (i.e., all the agents) at time t, respectively.
   2. Let $\mathcal{I}_t$ denote the set of active routers which have packets in their waiting buffers at time t.
      The global reward: $R_t = \sum_{i \in \mathcal{I}_t} r_t^i$.
   3. The joint action-value function which estimates the total delivery time of the packets being transmitted at time t is approximated by

$$\hat{Q}_t(S_t, A_t) := \sum_{i \in \mathcal{I}_t} \hat{Q}_t^i(s_t^i, a_t^i)$$

   4. We define the global feedback signal at time t:

$$\delta_t = R_t + \gamma \max_{A'} \hat{Q}_t(S_{t+1}, A') - \max_{A} \hat{Q}_t(S_t, A)$$

Apply to Communication Network

For each agent, say, agent i, with the global feedback signal δ_t and eligibility traces $\boldsymbol{z}_t^i$, the policy parameters are updated according to

$$\boldsymbol{\theta}_{t+1}^i = \boldsymbol{\theta}_t^i + \beta \boldsymbol{z}_t^i \delta_t$$

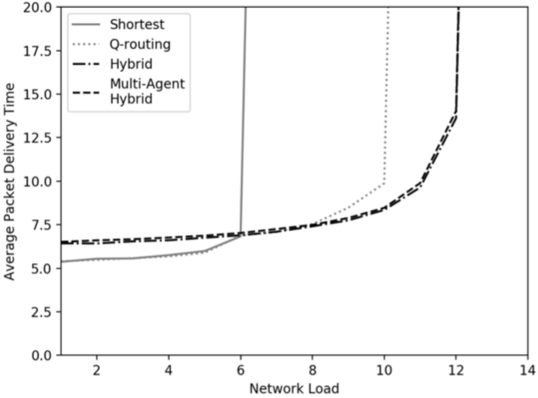
where β is the learning rate of policy parameters $\boldsymbol{\theta}$.

Experiment Results

1. We test our two RL algorithms, Hybrid and Multi-Agent Hybrid, on two network topologies, including an irregular 6×6 grid and a 116-node LATA telephone network.
2. We compare our two algorithms with those of three other algorithms:

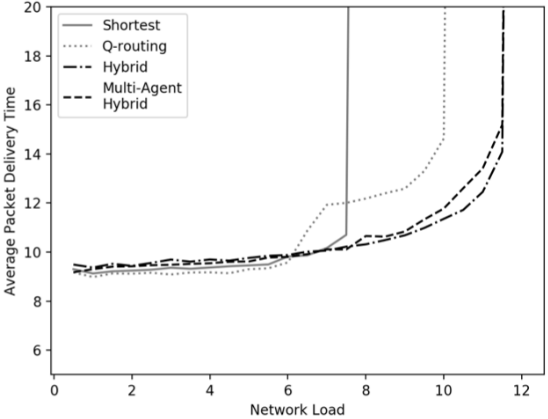
   1) Shortest Paths, which is a static routing scheme and is optimal when the network load is low
   2) Q-routing [Boyan and Littman, 1994], which is a value-based RL scheme
   3) GAPS [Peshkin and Savova, 2002], which is a policy-based RL scheme

Experiment Results



(a) Performance on the irregular 6×6 grid topology

Experiment Results



(b) Performance on the 116-node LATA network

Conclusion

1. Adaptability to dynamically changing network load
2. Affordable load
3. Scalability

References

1) Boyan, Justin A., and Michael L. Littman. "Packet routing in dynamically changing networks: A reinforcement learning approach." *Advances in neural information processing systems*. 1994.

2) Tao, Nigel, Jonathan Baxter, and Lex Weaver. "A multi-agent, policy-gradient approach to network routing." *In: Proc. of the 18th Int. Conf. on Machine Learning*. 2001.

3) Peshkin, Leonid, and Virginia Savova. "Reinforcement learning for adaptive routing." *In: Proc. of the 2002 International Joint Conference on Neural Networks.* Vol. 2. IEEE, 2002.

4) Baxter, Jonathan, and Peter L. Bartlett. "Infinite-horizon policy-gradient estimation." *Journal of Artificial Intelligence Research* 15 (2001): 319-350.