RL theory project

Yuan-Yu Wu (0856022)

June 2020

1 Introduction

In off-policy learning, we use data drawn from behavior policy to update target policy, it shows the mismatch of distributions. To deal with this issue, they first combines basic and weighted version of important sampling with Q-learning.

With the perspective that the weight on reward shouldn't depends on the future, the authors rewrite the formulation of important-sampling Q function as a per-decision version. Per-decision version of important-sampling Q estimator can also be written in a weighted version. In these two per-decision Q estimator, the authors combine eligibility-trace technique, which provides a connection between Monte Carlo and TD method.

The four methods mentioned above need to know the behavior policy in order to know the action probability. Moreover, some state-action pairs are not visited again, which makes it fail to calculate the estimator. To solve this issue, the authors purpose tree-backup method , which calculate the value function with target policy and also combine with eligibility-trace.

To summary the contributions of this paper, (1) they combine important-sampling and eligibility-trace with off-policy learning method; (2)They empirically rank the methods and show the improvement of each method.

2 Problem Formulation

Below shows some basic notations and assumptions behind the theorem.

- MDP Notation
  For each episode, the first state $s_0 \in S$ is chosen from the fixed distribution.At each step t, state of environment denotes as $s_t \in S$, action from agent denotes as $a_t \in A$. In response to action a_t, the environment would produce reward $r_{t+1} \in R$, and next state s_{t+1}. Assume S, A are finite and environment is characterized by one-step transition probability $P^a_{ss'}$ and one-step reward, r^a_s, for all $s, s' \in S$ and $a \in A$.

- Episodic framework
  Agent reacts with environment in a sequence of episodes, numbered $m = 1, 2, ...$, each episode is consist of finite step, $t = 1, 2, ...T_m$

- Agent policy and Q value
  Probability of mapping state to taking each action, denote $\pi : S \times A \to [0, 1]$
  The value of taking action a on s under π denotes $Q^\pi(s, a)$, which is considered as discounted future reward starting in s, taking a and following π, with $0 \leq \gamma \leq 1$.

  $$Q^\pi(s, a) = E_\pi\{r_1 + \gamma r_2 + ... + \gamma^{T-1} r_T | s_0 = s, a_0 = a\}.$$

  The problem of estimating Q^π for an arbitrary target policy π, and all data is generated by behavior policy b, which $b(s, a) > 0, \forall s \in S, a \in A$.

- Important sampling and weighted important sampling
  Classical important sampling is as below:

  $$E_d = \{x\} = \int_x d(x)dx = \int_x x\frac{d(x)}{d'(x)}dx \approx \frac{1}{n}\sum_{i=1}^{n} x_i \frac{d(x_i)}{d'(x_i)}$$

  Because the denominator could be small and cause large variation of coefficient, it comes a weighted version of important sampling. The estimator is shown as below.

  $$\frac{\sum_{i=1}^{n} x_i \frac{d(x_i)}{d'(x_i)}}{\sum_{i=1}^{n} \frac{d(x_i)}{d'(x_i)}}$$

To estimate the action value $Q(s, a)$, there are first-visit version and per-decision version with two types of important sampling, in total four methods to estimate $Q(s, a)$. First, let t_m be the first time when $(s_t, a_t) = (s, t)$ in mth episode.

- Q^{IS}: first-visit important sampling estimator

  $$Q^{IS} = \frac{1}{M}\sum_{m=1}^{M} R_m w_m.$$

  where R_m is discounted reward fallowing (s, a) in episode m,

  $$R_m = r_{t_m+1} + \gamma r_{t_m+2} + ... + \gamma^{T_m - t_m - 1} r_{T_m}.$$

  w_m is important sampling weight in episode m,

  $$w_m = \frac{\pi_{t_m+1}}{b_{t_m+1}} \frac{\pi_{t_m+2}}{b_{t_m+2}} ... \frac{\pi_{T_m-1}}{b_{T_m-1}}.$$

- Q^{ISW}: first-visit weighted important sampling estimator

  $$Q^{ISW} = \frac{\sum_{m=1}^{M} R_m w_m}{\sum_{m=1}^{M} w_m}.$$

- Q^{PD}: per-decision important sampling estimator

$$Q^{PD} = \frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{T_m-t_m}\gamma^{k-1}r_{t_m+k}\prod_{i=t_m+1}^{t_m+k-1}\frac{\pi_i}{b_i}.$$

- Q^{PDW}: per-decision weighted important sampling estimator

$$Q^{PDW} = \frac{\sum_{m=1}^{M}\sum_{k=1}^{T_m-t_m}\gamma^{k-1}r_{t_m+k}\prod_{i=t_m+1}^{t_m+k-1}\frac{\pi_i}{b_i}}{\sum_{m=1}^{M}\sum_{k=1}^{T_m-t_m}\gamma^{k-1}\prod_{i=t_m+1}^{t_m+k-1}\frac{\pi_i}{b_i}}.$$

In the tree-backup method, it forms new target using old value for actions that were not taken, and new value for action that was taken. It can process for n-step as n-step tree-backup estimator

- $Q_n^{TB}(s,a)$: n-step tree-backup estimator

$$\begin{aligned}Q_n^{TB} =& \frac{1}{M}\sum_{m=1}^{M}\gamma^n Q(s_{t_m+n},a_{t_m+n})\prod_{i=t_m+1}^{t_m+n}\pi_i \\ &+\sum_{k=t_m+1}^{t_m+n}\gamma^{k-t_m+1}\prod_{i=t_m+1}^{k-1}(r_k+\gamma\sum_{a\neq a_k}\pi(s_k,a)Q(s_k,a))\end{aligned}$$

In next section, there are some theorems to be proved.

- Theorem 1: Q^{PD} is consistent unbiased estimator of Q^{π}
- Theorem 2: Algorithm 1 with offline updating converges $w.p.1$ to Q^{π}, under the usual step-size conditions on α.
  It is under a assumption that behavior policy is soft, which means $b(s,a) > 0, \forall s \in S, a \in A$
- Theorem 3: For any non-starving behavior policy, the offline version of algorithm 2 converges $w.p.l$ to Q^{π}, under the usual step-size conditions on α.

3 Theoretical Analysis

Two algorithms are shown in Appendix to help the proof of theorem 2,3.

- Theorem 1: Theorem 1: Q^{PD} is a consistent unbiased estimator of Q^{π}
  Extend the formulation of Q^{IS}, it would be

$$E\{(\sum_{k=1}^{T-t}\gamma^{k-1}r_{t_k})\prod_{i=t+1}^{T-1}\frac{\pi_i}{b_i}|s_t=s,a_t=a,b\}$$

Expectation of k-th term can be extented as

$$E\{\gamma^{k-1} r_{t+k} \frac{\pi_{t+1}}{b_{t+1}} ... \frac{\pi_{t+k-1}}{b_{t+k-1}} | s_t, a_t, ..., s_{t+k-1}, a_{t+k-1}, b\}.$$
$$E\{\frac{\pi_{t+k}}{b_{t+k}} ... \frac{\pi_{T-1}}{b_{T-1}} | s_t, a_t, ..., s_{t+k}, a_{t+k}, b\}$$

Due to the environment is under assumption of MDP, second factor can be written as

$$E\{\frac{\pi_{t+k}}{b_{t+k}} ... \frac{\pi_{T-1}}{b_{T-1}} | s_t, a_t, b\} = 1$$

Therefore, the expectation of Q^{IS} can be written as

$$E\{(\sum_{k=1}^{T-t} \gamma^{k-1} r_{t_k}) \prod_{i=t+1}^{T-1} \frac{\pi_i}{b_i} | s_t = s, a_t = a, b\} =$$
$$E\{\sum_{k=1}^{T-t} \gamma^{k-1} r_{t_k} \prod_{i=t+1}^{t-k+1} \frac{\pi_i}{b_i} | s_t = s, a_t = a, b\}$$

It leads to the formulation of Q^{PD}. Since Q^{IS} is known to be consistent unbiased of Q^{π}, so is Q^{PD}.

- Theorem 2: Algorithm 1 with offline updating converges $w.p.1$ to Q^{π}, under the usual step-size conditions on α.
In this part, focusing on eligibility-trace for state-action pair (s, a), it can be written as

$$e_t(s, a) = \gamma^{t-t_m} \prod_{l=t_m+1}^{t} \frac{\pi_l}{b_l}.$$

$$\sum_{k=1}^{n} e_{t+k-1}(s, a)\delta_{t+k-1}(s, a) = \sum_{k=1}^{n} \gamma^{k-1} (\prod_{l=t+1}^{t+k-1} \frac{\pi_l}{b_l})(r_{t+k} + \gamma \frac{\pi(s_{t+k}, a_{t+k})}{b(s_{t+k}, a_{t+k})} Q(s_{t+k}, a_{t+k}) -$$
$$Q(s_{t+k-1}, a_{t+k-1}))$$
$$= \sum_{k=1}^{n} \gamma^{k-1} r_{t+k} \prod_{l=t+1}^{t+k-1} \frac{\pi_l}{b_l} \gamma^n Q(s_{t+n}, a_{t+n}) \prod_{l=t+1}^{t+n-1} \frac{\pi_l}{b_l} -$$
$$Q(s_{t+k-1}, a_{t+k-1})$$
$$= R_t^{(n)} - Q(s_t, a_t)$$

Plug back and will find it converge to correct Q function.

- Theorem 3: For any non-starving behavior policy, the offline version of algorithm 2 converges $w.p.l$ to Q^{π}, under the usual step-size conditions on α.

Focusing on eligibility-trace for state-action pair (s, a), it can be written as

$$e_{t+k}(s,a) = \gamma^k \prod_{l=t+1}^{t+k} \pi(s_l, a_l).$$

$$\begin{aligned} Q(s_t,a_t) + \sum_{k=1}^{n} e_{t+k-1}\gamma^{k-1} \prod_{l=t+1}^{t+k-1} \pi(s_l,a_l)(r_{t+k} \\ + \gamma \sum_{a\in A} \pi(s_{t+k},a_{t+k})Q(s_{t+k},a_{t+k}) - Q(s_{t+k-1},a_{t+k-1})) \\ = Q(s_t,a_t) + \sum_{k=1}^{n} e_{t+k}(s,a)\delta_{t+k} \end{aligned}$$

Plug back and will find it converge to correct Q function.

4 Conclusion

In this paper, it provide different methods combined with important sampling and eligibility-trace under offpolicy training.

In four different important sampling estimator, there is a assumption that behavior policy has to be soft. However, in practice, the policy might not be soft during training. For example, consider policy is a Gaussian distribution, which is commonly used on continuous action space, σ of the distribution sometimes rapidly becomes zero under some situation. It seems to drop into local minimum at the very beginning. At this moment, the policy is not soft anymore, and might lead the training fail.

To solve this, one straight forward method is to add entropy of policy distribution into reward, which is used in Soft Actor-Critic (SAC) method and Proximal Policy Optimization (PPO).This simple method can make policy soft during training and make it follow the theorem.

5 Appendix

Algorithm 1 Online, Eligibility-Trace Version of Per-Decision Importance Sampling

1. Update the eligibility traces for all states:

$$\begin{aligned} e_t(s,a) &= e_{t-1}(s,a)\,\gamma\lambda\frac{\pi(s_t,a_t)}{b(s_t,a_t)}, \qquad \forall s,a \\ e_t(s,a) &= 1, \text{iff } t = t_m(s,a), \end{aligned}$$

   where $\lambda \in [0,1]$ is an eligibility trace decay factor.

2. Compute the TD error:

$$\delta_t = r_{t+1} + \gamma\frac{\pi(s_{t+1},a_{t+1})}{b(s_{t+1},a_{t+1})}Q_t(s_{t+1},a_{t+1}) - Q_t(s_t,a_t)$$

3. Update the action-value function:

$$Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \alpha\, e_t(s,a)\,\delta_t, \qquad \forall s,a$$

Algorithm 2 Online, Eligibility-Traces Version of Tree Backup

1. Update the eligibility traces for all states:

$$\begin{aligned} e_t(s,a) &= e_{t-1}(s,a)\gamma\lambda\pi(s_t,a_t), \qquad \forall s,a \\ e_t(s,a) &= 1 \text{ iff } t = t_m(s,a) \end{aligned}$$

   where $\lambda \in [0,1]$ is an eligibility trace decay parameter.

2. Compute the TD error:

$$\delta_t = r_{t+1} + \gamma \sum_{a \in A} \pi(s_{t+1},a) Q(s_{t+1},a) - Q(s_t,a_t)$$

3. Update the action-value function:

$$Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \alpha e_t(s,a)\,\delta_t, \qquad \forall s,a$$