Off-policy policy estimation

Bochao Li
Department of Computer Science
University of Illinois, Champaign and Urbana
Champaign,IL 61820
bochao2@illinois.edu

Abstract

This paper is a review paper on several papers on some research on Reinforcement Learning with rich observation. To solve this problem a new RL model Contextual Decision Proces is proposed. And a PAC guaranteed algorithm is given.

1 Introduction

In Reinforcement learning, there is one group of problem where observation is rich, such as images or texts. In these problems, sophesicated exploration is needed to get sounding result. In RL theory, this problem already is a solved problem [Brafman and Tennenholtz, 2003][Kearns and Singh, 2002].However, those result is not appliable in practice. This may because those complicated problem are not MDP problems (for example, Atari games)[Mnih et al.,2015]. In this series of papers , we propose a new RL setting Which is called CDP. In Nan's paper, they have proved that most current RL setting including MDP, POMDP, PSR and LQR can be regarded as special case of CDP. And this serieis of paper develop PAC guarantee for learning in CDP environment.

This paper can be devided into three parts. First we introduce some basic definitions on terms used in CDP. And second we shows how some current model setting like MDP or POMDP can be generalized to CDP. The third part is to gives a PAC algorithm to learn in CDP. The structure of this paper is: In section 2, we define the terms used in CDP, and make an comparison on related terms in MDP model. In section 3 we gives 2 examples on how to generalize MDP to CDP model, and how to set corresponding parameter in CDP model. In section 4 we gives a PAC algorithm for learning in CDP, under relatively strong assumption. In section 5 we shows the possibility to relieve some assumptions. And section 6 is a summary and discussion on some related work and possible future work.

2 Model

First of all we need to define the model we study. To make content more understood, we will first introduce a Markov Decision Process(MDP) model and make comparision upon it.

2.1 Markov Decision Process

Definition 1 (Markov Decision Process(MDP))**.** A (finite-horizon) contextual Decision Process(CDP) defined as a tuple$(\mathcal{S}, \mathcal{A}, H, P, r, \gamma)$. $\mathcal{X}$ is the state space, $\mathcal{A}$ is the finite action space, H is horizon of problem and $P = (P_\emptyset, P_+).P_\emptyset \in \Delta(\mathcal{S})$ is distribution over initial contexts and $P_+ : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ while $P_+(x, a, y) = Pr[s_{h+1} = y|s_h = x, a_h = a]$ the transition function. r is the reward $r(x, a, y)$ is the reward received by taking action a at state x and transit to state y. And γ is a decade factor.

Statistical Learning Theory (ECE543 Spring 2019), Champaign, IL,USA.

Although there are MDP model without the decade factor γ and use average reward $\frac{1}{H}\sum\limits_{h=1}^{H} r_h$ or cumupated reward $\sum\limits_{h=1}^{H} r_h$ instead with assumption the cumulated reward is finite, but we just need one model to make comparision.
The value funciton is defined as

$$V^{\pi} = \mathbb{E}_P[\sum_{h=1}^{H} \gamma^{h-1} r_h | a_{1:H\sim\pi}] \tag{1}$$

and Q-value funciton is defined as

$$Q(x, a) = r(x, a, y) + \gamma V^*(y) P_+(x, a, y) \tag{2}$$

while V^* is the optimal value defined as

$$V^* = \max_{\pi} V^{\pi} \tag{3}$$

2.2 Contextual Decision Process

Definition 2 (Contextual Decision Process(CDP))**.** A (finite-horizon) contextual Decision Process(CDP) defined as a tuple$(\mathcal{X}, \mathcal{A}, H, P, r)$. $\mathcal{X}$ is the context space, $\mathcal{A}$ is the finite action space, H is horizon of problem and $P = (P_\emptyset, P_+).P_\emptyset \in \Delta(\mathcal{X})$ is distribution over initial contexts and $P_+ : (\mathcal{X} \times \mathcal{A})^* \times \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})$ while $P_+(x_1, a_1 \cdots x_h, a_h, y) = Pr[x_{h+1} = y | x_h = x_h, a_h = a_h \cdots x_1 = x_1, a_1 = a_1]$ the transition function.

The main difference between CDP and MDP given above are (1)CDP does not have the decade factor, and (2) In CDP we do not possess the Markov property. Which means the future is decided by current and past, not just current
Because we will use cumulated reward in following problem we concern, we add assumption $\sum\limits_{h=1}^{H} r_h \leq 1$ for arbitrarily chosen action sequence as an regularization on the problem. Then we start to define the term in CDP model that corresponding to Bellman equation for MDP.

Definition 3 (Average Bellman error)**.** we define

$$\varepsilon(f, \pi, h) = \mathbb{E}[f(x_h, a_h) - r_h - f(x_{h+1}, a_{h+1}) | a_{1:h} \sim \pi, a_{h:h+1} \sim \pi_f] \tag{4}$$

The meaning of this average Bellman error is that we follows policy π at level 1 to $h-1$, and follows policy at level h to $h+1$.

Notice that in MDP setting, if a Q-value funciton is optimal for Bellman operator, then it has zero Bellman average error for $\forall h \in \{H\}$ and $\forall \pi$. Although in MDP setting we have to redefine the average Bellman error because usually a decade factor γ is considered.

In the CDP we defined above, we make no assumption on the cardinality of context space, so to learn in this environment, approximation is necessary. We use value-based RL with function approxination. We assume we have access to a set of funciton $\mathcal{F} \subset \mathcal{X} \times \mathcal{A} \to [0, 1]$, which has same form as Q-learning function. without loss of generality, we define $f(x_{H+1}, a) \equiv 0$ (this is to simplify the form of Bellman equation). Like typical value-based RL, our goal of learning in this environment is to get a optimal $f^* \in \mathcal{F}$ with respect to a particular Bellman equation defined on CDP, and the optimal f satisfy that the reward follows the greedy policy of this policy $\pi_{f^*}(x) = \arg\max_{a\in\mathcal{A}} f(x, a)$ is largest.

Definition 4 (Bellman equations and validity of f)**.** Given an tuple (f, π, h), a Bellman equation means $\varepsilon(f, \pi, h) = 0$. We define $f \in \mathcal{F}$ is valid if $\varepsilon(f, \pi_{f'}, h) = 0$ holds for every $f' \in \mathcal{F}, h \in [H]$. We define $f \in \mathcal{F}$ is θ-valid if the Bellman equation on $(f, \pi_{f'}, h) \leq \theta$ holds for every $f' \in \mathcal{F}, h \in [H]$

Definition 5 (optimal value)**.** We define optimal function as $f^* = \arg\max_{f\in\mathcal{F}:f\text{is valid}} V^{\pi_f}$, the optimal value $V^*_{\mathcal{F}} = V^{\pi_{f^*}}$ And we define θ-optimal function as $f^*_\theta = \arg\max_{f\in\mathcal{F}:f\text{is}\theta\text{-valid}} V^{\pi_f}$, the θ-optimal value $V^*_{\mathcal{F},\theta} = V^{\pi_{f^*_\theta}}$

If we consider problem in MDP setting, we do not need the definition of validity. Because in MDP setting with decade factor, the Bellman operator is a contractive mapping and the existence of an optimal Q-function is guaranteed by fixed point Theorem [Baser, 2019].In MDP problem, if both state space $\mathcal{X}$ and action space $\mathcal{A}$ are finite, we can calculate the optimal Q-funciton simpliy by Bellman operator iteration. And in MDP problem with infinite state space $\mathcal{X}$, we still can calculate the optimal Q-funciton this way if we define The Bellman operator into integral form [Melo, 2008]. However, in CDP problem, the existence of such optimal not guaranteed. And that is why we need to define the validity (and θ-validity) of problem.

Definition 6 (Bellman factorization and Bellman rank)**.** CDP$(\mathcal{X}, \mathcal{A}, H, P)$, and $\mathcal{F} \subset \mathcal{X} \times \mathcal{A} \rightarrow [0, 1]$. admit Bellman factorization with Bellman rank M and norm parameter ζ, if there exist $\nu_h : \mathcal{F} \rightarrow \mathbb{R}^M$,$\xi_h : \mathcal{F} \rightarrow \mathbb{R}^M$ for each $f, f' \in \mathcal{F}, h \in [H]$.
$\varepsilon(f, \pi_{f'}, h) = \langle \nu_h(f'), \xi_h(f) \rangle$ and $\max\limits_{f,f' \in \mathcal{F}}(||\nu_h(f')||_2 \cdot ||\xi_h(f)||_2) = \zeta < \infty$
We say a CDP and a $\mathcal{F}$ admit η-approximate Bellman factorization with Bellman rank $M.$ norm parameter ζ if there exist $\nu_h : \mathcal{F} \rightarrow \mathbb{R}^M$,$\xi_h : \mathcal{F} \rightarrow \mathbb{R}^M$ for each $f, f' \in \mathcal{F}, h \in [H]$.
$|\varepsilon(f, \pi_{f'}, h) - \langle \nu_h(f'), \xi_h(f) \rangle| \leq \eta$ and $\max\limits_{f,f' \in \mathcal{F}}(||\nu_h(f')||_2 \cdot ||\xi_h(f)||_2) = \zeta < \infty$

We can regard the Bellman rank M as number of "hidden" state (but it is not the "hidden" state we usually consider in POMDP). And the Bellman factorization is quiet tricky here. We do not directly explain why we decompose average Bellman rank this way here (although it is crucial), but leave it to seciton 5.

3 Example of generalization

In this section we show how MDP can be generalized to CDP.

3.1 generalize MDP to CDP

For the MDP setting we consider in section 2.1. Let $(\mathcal{X}, \mathcal{A}, H, P)$ be the CDP induced by the MDP model, with $\mathcal{X} = \mathcal{S} \times [H]$. It is easy to tell that we can generalize it to a CDP with Bellman rank $|M| = |\mathcal{X}|$. And For MDP, the average Bellman rank can be factor into $[\nu_h(f)]_x = Pr[x_h = (x, h)|a_{1:h-1} \sim \pi_f]$ and $[\xi_h(f)]_x = \mathbb{E}[f(x_h, a_h) - r_h - f(x_{h+1}, a_{h+1})|x_h = (s, h), a_{h"h+1} \sim \pi_f]$ $\forall x \in \mathcal{X}$. And because ν_h is a probability, we have $||\nu_h(\cdot)||_1 \leq 1$. Because ξ_h is a expectation of reward, and the cumulated reward is bounded by 1, then the difference of reward between two time step is bounded by 2. Then we have $||\xi_h(\cdot)||_\infty \leq 2$. Then we have $||\nu_h(\cdot)||_2 \leq 1$ and $||\xi_h(\cdot)||_2 \leq 2\sqrt{M}$. And then we have $\zeta = 2\sqrt{M}$[1]

4 PAC algorithm for CDP

Our goal is to find an RL algorithm with no dependence on the number of observations $|\mathcal{X}|$, and a poly nomial dependence on the number of actions K, Bellman rank M, horizon H, and size of function set $|\mathcal{F}| = N$.
And we define the Probably approximately correct(PAC) for CDP problem as: Given $\mathcal{F}$ two parameter $\epsilon, \delta \in (0, 1)$,$\exists$ algorithm product $\hat{\pi}$, so that $V^{\hat{\pi}} \geq V^*_{\mathcal{F}} - \epsilon$ with probability at least $1 - \delta$

4.1 assumptions

It is hard to study problem in general CDP model, because context space usually is very large or infinite, and usually is unavoidable, so to make the problem practice, some assumption is needed. Here we first introduce a algorithm with relatively strong assumption, and we will see how to relieve some of them in chapter 5. The assumption we need are:
(1) $\sum\limits_{h=1}^{H} r_h \leq 1$ for arbitrarily chosen action sequence (policy).
(2) we assume the Bellman rank M is known and finite

[1] because of the page constrain, and the composition of the paper, we only show one example of how to generalize other model to CDP here. More examples can be find in the origin paper [Jiang et al., 2017]

(3) we assume the cardinality of Q-value funciton is finite $|\mathcal{F}| < \infty$
(4) the optimal function exist $f^* \in \mathcal{F}$
(5) the function class $\mathcal{F}$ we use admit Bellman factorization with Bellman rank M.

Algorithm 1 OLIVE $(\mathcal{F}, M, \zeta, \epsilon, \delta)$

```
 1: Collect $n_{est}$ trajectories, actions taken arbitrarily, and save initial contexts$\{x_1^{(i)}\}_{i=1}^{n_{est}}$
 2: Estimate the predictied value for each $f \in \mathcal{F} : \hat{V}_f = \frac{1}{n_{est}} \sum_{i=1}^{n_{est}} f(x_i^{(i)}, \pi_f(x_1^{(i)}))$
 3: $\mathcal{F}_0 \leftarrow \mathcal{F}$
 4: for $t = 1, 2 \cdots$ do do
 5:     Choose policy $f_t = \arg\max_{f \in \mathcal{F}_{t-1}} \hat{V}_f, \pi_t = \pi_{f_t}$
 6:     Collect $n_{eval}$ trajectories$\{(x_1^{(i)}, a_1^{(i)}, r_1^{(i)} \cdots, x_H^{(i)}, a_H^{(i)}, r_H^{(i)})\}_{i=1}^{n_eval}$ by following policy $\pi_t$
 7:     Estimate $\hat{\varepsilon}(f_t, \pi_t, h) = \frac{1}{n_{eval}} \sum_{i=1}^{n_{eval}} [f_t(x_h^{(i)}, a_h^{(i)}) - r_h^{(i)} - f_t(x_{h+1}^{(i)}, a_{h+1}^{(i)})]$ for $\forall h \in [H]$
 8:     if $\sum_{h=1}^{H} \hat{\varepsilon}(f_t, \pi_t, h) \leq 5\epsilon/8$ then
 9:         Terminate and output $\pi_t$
10:     end if
11:     Pick any $h_t \in [H]$ for which $\hat{\varepsilon} \geq 5\epsilon/8H$
12:     Collect trajectories$\{(x_1^{(i)}, a_1^{(i)}, r_{1^{(i)}} \cdots x_H^{(i)}, a_H^{(i)}, r_H^{(i)}))\}_{i=1}^{n}$ following $\pi_t$ for all $h \neq h_i$ and
        choose $a_{h_i}^{(i)}$ randomly
13:     Estimate $\hat{\varepsilon}(f, \pi_t, h_t) = \frac{1}{n} \sum_{i=1}^{n} \frac{\mathbb{1}[a_{h_t}^{(i)} = \pi_f(x_{h_t}^{(i)})]}{1/K} (f_t(x_h^{(i)}, a_h^{(i)}) - r_h^{(i)} - f_t(x_{h+1}^{(i)}, a_{h+1}^{(i)}))$
14:     Learn $\mathcal{F}_t = \{f : \mathcal{F}_{t-1} : |\hat{\varepsilon}(f, \pi_t, h_t)| \leq \phi\}$
15: end for
```

4.2 idea of algorithm

First we produce n_{est} traojectories while action taken in an arbitrary manner. and we use these datas to given an estimation of each $f \in \mathcal{F}$. By concentration inequality(we simply use Holder inequality here), when n_{est} is large enough, we can have an approximated value funciton $\hat{V}_f$ for $\forall f \in \mathcal{F}$(line 1,2). Then we choose the optimal f and evaluate its average Bellman rank. If we use large enough trajectories n_{eval}, we can have an approximate value to certain accuracy we want. If the optimal f is valid, then it is done. If the optimal f is not optimal, there must be at least one horizon h_t, the average Bellman error is large. Then we re-estimate the average Bellman error under a different policy at the horizon h_t(because we already know it cannot be optimal policy). And then elimite those impossible funciton in our funciton set. And during the elliminaiton, we use the ellipsoid method. Ellipsoid method usually is a little costly, but it will keep the convexity of the area we optimizing, and keep an exponential optimize rate.

Theorem 1. *For any $\epsilon, \delta \in (0, 1)$, any Contextual Decision Process, function class $\mathcal{F}$ that admits a Bellman factorization with parameters M, ζ run OLIVE with following parameters: $\phi = \frac{\epsilon}{12H\sqrt{M}}$,$n_{est} = \frac{32}{\epsilon^2} \log(\frac{6N}{\delta})$,$n_{eval} = \frac{288H^2}{\epsilon^2} \log(\frac{12H^2M\log(\frac{6H\sqrt{M}\zeta}{\epsilon})}{\delta})$,$n = \frac{4608H^2MK}{\epsilon^2} \log(\frac{12NHM\log(\frac{6H\sqrt{M}\zeta}{\epsilon})}{\delta})$. With probability at lest $1 - \delta$, OLIVE halts and return a policy $\hat{\pi}$, $V^{\hat{\pi}} \geq V_{\mathcal{F}}^* - \epsilon$ and the number of episodes required is at most $\tilde{O}(\frac{M^2H^3k}{\epsilon^2} \log(\frac{N\zeta}{\delta}))$, whle $\tilde{O}$ notation ssuppress poly-logarithmic dependence on parameters except N and δ.*

5 Relieve on assumptions

5.1 relief of assumption (2)

We assume the Bellman rank M is known when running the OLIVE algorithm. However, that is unrealistic. Also, directly assume Bellman rank to be certain value also may cause problem. For example, if the Bellman rank is too small, it means our model is over-simplified and model may have large bias error, which cause the algorithm cannot converge. We use the algorithm GuessM here to solve this problem. The way to solve this problem is kinds of straight forward. We try $M = 1$ at first, if it does not work, then try with $M = 2$ and then $M = 4$... And when running the i-th time of OLIVE in the algorithm, we give parameter $\frac{\delta}{i(i+1)}$ to the algorithm. The reason why we set the parameter this way is because we want the cumulated probability to be at least $1 - \delta$, and $\sum\limits_{i=1}^{\infty} \frac{\delta}{i(i+1)} < \delta$. And because we run the algorithm with exponentially increasing M', the algorithm halt at $M' \leq 2M$. Then the total number of calls is bounded by $\log_2 M + 1$. Then the time we need is at most $\log(M) \times$ OLIVE time cost, which have same order of complexity under $\tilde{O}$ notion.

Algorithm 2 GUESSM $(\mathcal{F}, \zeta, \epsilon, \delta)$

```
for $i = 1, 2 \cdots$ do
    $M' \leftarrow 2^i$
    Call OLIVE($\mathcal{F}, M', \epsilon, \frac{\delta}{i(i+1)}$), with parameters specified on Theorem 1.
    Terminate the subroutine when $t > HM' \frac{\log(\frac{6H\sqrt{M'}\zeta}{\epsilon})}{\log(5/3)}$ in Line 4
    if  a policy $\pi$ is returned from OLIVER then
        Then return $\pi$
    end if
end for
```

Theorem 2. *For any $\epsilon, \delta \in (0, 1)$, any CDP and any funciton class $\mathcal{F}$ that admit a Bellman factorization with parameter M, ζ, in we run GuessM($\mathcal{F}, \epsilon, \delta$), then with probability at least $1 - \delta$, OLIVE halts and returns a policy that satisfies $V^{\{}hat\pi \geq V_{\mathcal{F}}^{*} - \epsilon$. And the number of episodes required at most $\tilde{O}(\frac{M^2 H^3 k}{\epsilon^2} \log(\frac{N\zeta}{\delta}))$.*

5.2 relief of assumption (4) (5)

At first we can relieve the assumption (4) and (5). We already defined θ-validity (optimal function) and η-approximate factorization in previous chapter. Instead of assume optimal function exist, we can just assume an θ optimal funciton exist. And instead of assume a perfect Bellman factorization exist, we assume a η-approximate factorization exist. And we have an alternative algorithm called OLIVER The idea of this algorithm is similar to OLIVE. Only difference is that we loose the bound and eliminate "bad" funciton as a slower rate.

Theorem 3. *For any $\epsilon, \delta \in (0, 1)$, any COntextual Decision Process and funciotn class $\mathcal{F}$ that adimits a η-approximate Bellman factorizaiton, with parameter M, ζ, η, suppose we run OLIVER with any $\theta \in [0, 1]$ and other parameter the same as THeorem 1. With probability at least $1 - \delta$, OLIVER halts and returns a policy π that is at most $\epsilon + 8H\sqrt{M}(\theta + \eta)$ suboptimal compared to $V_{\mathcal{F},\theta}^{*}$, the θ-optimal value, the number of episodes required is at most $\tilde{O}(\frac{M^2 H^3 k}{\epsilon^2} \log(\frac{N\zeta}{\delta}))$.*

5.3 reilef of assumption (2)

It is possible to havve practice algorithm in CDP with infinite cardinality $|\mathcal{F}| = \infty$. If we review our OLIVE algorithm, we can find that in the algorithm, only two term relevant to function f is used in algorithm OLIVE. The first thing we use is the optimal policy for f π_f. And the second think we consider is a mapping $g_f : x \rightarrow f(x, \pi_f(x))$, which is a V-value function. And each function $f \in \mathcal{F}$ can be decomposed to a pair of funciton (π_f, g_f). This gives us possible way to solve problem with infinite cardinality Q-value funciton $\mathcal{F}$. Because we actually do not use all property of f in our algorithm (basically we only use the optimal funciton). The policy space Π and V-value funciton

Algorithm 3 OLIVER $(\mathcal{F}, \theta, M, \zeta, \eta, \epsilon, \delta)$

```
$\epsilon' = \epsilon + 2H(3\sqrt{M}(\theta + \eta) + \eta)$
**Collect** $n_{est}$ trajectories, actions taken arbitrarily, and save initial contexts$\{x_1^{(i)}\}_{i=1}^{n_{est}}$
**Estimate** the predicted value for each $f \in \mathcal{F} : \hat{V}_f = \frac{1}{n_{est}} \sum_{i=1}^{n_{est}} f(x_i^{(i)}, \pi_f(x_1^{(i)})$
$\mathcal{F}_0 \leftarrow \mathcal{F}$
**for** $t = 1, 2 \cdots$ **do do**
    **Choose policy** $f_t = \arg\max_{f \in \mathcal{F}_{t-1}} \hat{V}_f, \pi_t = \pi_{f_t}$
    **Collect** $n_{eval}$ trajectories$\{(x_1^{(i)}, a_1^{(i)}, r_1^{(i)} \cdots, x_H^{(i)}, a_H^{(i)}, r_H^{(i)})\}_{i=1}^{n_eval}$ by following policy $\pi_t$
    **Estimate** $\hat{\varepsilon}(f_t, \pi_t, h) = \frac{1}{n_{eval}} \sum_{i=1}^{n_{eval}} [f_t(x_h^{(i)}, a_h^{(i)}) - r_h^{(i)} - f_t(x_{h+1}^{(i)}, a_{h+1}^{(i)})]$ for $\forall h \in [H]$
    **if** $\sum_{h=1}^{H} \hat{\varepsilon}(f_t, \pi_t, h) \leq 5\epsilon'/8$ **then**
        Terminate and output $\pi_t$
    **end if**
    Pick any $h_t \in [H]$ for which $\hat{\varepsilon} \geq 5\epsilon'/8H$
    **Collect** trajectories$\{(x_1^{(i)}, a_1^{(i)}, r_{1^{(i)}} \cdots x_H^{(i)}, a_H^{(i)}, r_H^{(i)}))\}_{i=1}^{n}$ following $\pi_t$ for all $h \neq h_i$ and choose $a_{h_i}^{(i)}$ randomly
    **Estimate** $\hat{\varepsilon}(f, \pi_t, h_t) = \frac{1}{n} \sum_{i=1}^{n} \frac{\mathbb{1}[a_{h_t}^{(i)} = \pi_f(x_{h_t}^{(i)})]}{1/K} (f_t(x_h^{(i)}, a_h^{(i)}) - r_h^{(i)} - f_t(x_{h+1}^{(i)}, a_{h+1}^{(i)}))$
    **Learn** $\mathcal{F}_t = \{f : \mathcal{F}_{t-1} : |\hat{\varepsilon}(f, \pi_t, h_t)| \leq \phi + \theta\}$
**end for**
```

space $\mathcal{G}$ might be finite. To properly define the dimension of function class π_f and g_f, we need to review some basic and advance dimension measure in statistical learning.

Definition 7 (VC-dimension)**.** Define $\mathcal{H} \subset \mathcal{X} \to \{0, 1\}$, we define $\mathcal{H}_X = \{h(x_1), \cdots, h(x_{|X|} : h \in \mathcal{H}\}$. The VC-dimension for $\mathcal{H}$, $VC - dim(\mathcal{H})$ is defined as the maximal cardinality of a set $X = \{x_1, \cdot, x_n\} \subset \mathcal{X}$, $|\mathcal{H}_X| = 2^{|X|}$ (or we say X is shattered by $\mathcal{H}$)

Definition 8 (Pseudo-shattering)**.** Define $\mathcal{H} \subset \mathcal{X} \to \mathbb{R}$. We say a series of feature $x = (x_1, x_2 \cdots x_m)$, $x_i in \mathcal{X}$ for all i is Pseudo-shattered by $\mathcal{H}$ if there exists a vector $\xi = (\xi_1, \xi_2 \cdots \xi_m)$, $\xi_i in \mathbb{R}$ for all i(called "witness"), s.t. for all $b \in \{0, 1\}^m = (b_1, b_2 \cdots b_m)$, there exist $h_b \in \mathcal{H}$, such that $\mathbb{1}[h_b(x_i) - r_i] = b_i$ for all i.

Definition 9 (Pseudo dimension)**.** $\mathcal{H} \subset \mathcal{X} \to \mathbb{R}$, the Pseudo-dimension $Pdim(\mathcal{H})$ is the cardinality of the largest set that can pseudo-shattered by $\mathcal{H}$.

Pseudo dimension can be regarede as an extension of VC-dimension. Especially, if we define $\mathcal{H}^+ = \{(x, \xi) \to \mathbb{1}[h(x) > \xi] : h \in \mathcal{H}\} \subset \mathcal{X} \times \mathbb{R} \to \{0, 1\}$, then we have $Pdim(\mathcal{H}) = VC - dim(\mathcal{H}^+)$.

Definition 10 (Natarajan dimension)**.** Suppose $\mathcal{X}$ is a feature space and $\mathcal{Y}$ is a finite label space. Given Hypothesis class $\mathcal{H} \subset \mathcal{X} \to \mathcal{Y}$, its Nataranjan dimension is defined as maximum cardinality of a set $A \subset \mathcal{X}$, which satisfy: there exist $h_1, h_2 : A \to \mathcal{Y}$ (1) $\forall x \in A, h_1(x) \neq h_2(x)$. and (2) $\forall B \subset A, \exists h \subset \mathcal{H}, \forall x \in B, h(x) = h_1(x)$ and $\forall x \in A \setminus B, h(x) = h_2(x)$

With these definiton, it is easy to see that we can bound the dimension of funciton π_f using Nataranjan dimension and bound the dimension of funciotn g_f using Pseudo dimension. And notice that in the time complexity we calculate for OLIVER algorithm, the dependence on $|\mathcal{F}| = N$ is $\log(N)$. This way, we can simplify it to sum of $\log(d_\Pi)$ which is the log of dimension of policy space and $\log(d_\mathcal{G})$, which is the dimension of the V-value funciton.

Theorem 4. $\Pi \subset \mathcal{X} \to \mathcal{A}$,*with* $Ndim(\Pi) \leq d_\Pi < \infty$, $\mathcal{G} \subset \mathcal{X} \to [0, 1]$, *with* $Pdim(\mathcal{G}) \leq d_\mathcal{G} < \infty$. *For any* $\epsilon, \delta \in (0, 1)$, *any Contextual Decision Process with policy space* Π *and function space* $\mathcal{G}$, $(\Pi, \mathcal{G})$ *admit a Bellman factorization with parameter*M, ζ, *if we run OLIVER with appropriate parameters, with probability at least* $1 - \delta$, *OLIVER halts and return a policy*$\hat{\pi}$, $V^{\hat{\pi}} \geq V_\mathcal{F}^* - \epsilon$ *and the number of episodes required is at most* $\tilde{O}(\frac{M^2 H^3 K^2}{\epsilon^2}(d_\Pi + d_\mathcal{G} + \log(\frac{\zeta}{\delta}))$

6 Summary

This paper basically review the result in Nan's paper about Contextual Decision Model [Jiang, 2017], a general model that can include main model in Reinforcement Learning like MDP, POMDP and PSR. The term Contextual Decision Model is first proposed by Akshay [Krishnamurthy et al.,2016]. However, the first CDP actually is a tree-form POMDP. It keeps the assumption of Markov property, and assume that the transition between state is determistic. Several work has been done on these series of work. Christoph proves that the OLIVE/OLIVER algorithm proposed in Nan's paper is not Oracle efficient, and propose an oracle efficient algothm that based on Akshay's CDP setting[Dann et al.,2018]. And further work has been done on ivestigate an Model-based problem in CDP setting [Sun et al.,2019]. This paper's CDP is a little different from all current model. Basically it keeps the Markov property, but remove the assumption that the transition need to be deterministic.

Generally, no concrete work has been done in this field. Only primal result has been given and more work needs to be done in the future. Nan's paper gives a relatively statistical efficient algorithm to learn in CDP, but that is the only work done on this particular problem, and its efficientless in oracle has been proved in later work. While oracle efficient result usually relies on more strict assumption. Also we have to notice that although learning in rich observation environment is generally hard, work in it is worthy. Efficient decompsition will make cost to learning in rich observation environment decrease sharply. For example, if we consider an Mario game. If we learn to play it using image or video, we need to analyze thousands of data every frame. However, if we learn the game play of the game (which is the "hidden" state of this problem), we can know that there are only dozens of move Mario can make and it will decrease the hardness of learning rapidly. And maybe this decomposition is the reason why human can learn much faster than Machine.

Possible future work may be combining Information Theory knowledge to RL problem. No matter which exactly model is used. When considering RL problem with rich observation, usually "hidden" states or similar structure need to be concerned (like discussed in previous section, the Bellman rank proposed in Nan's paper also can be regarded as a different form of "hidden" states). And to effectively learning in this environment, we need sophesiticate method to transfer information we get from observation samples to information in "hidden" states. Apart from this, kernel related method may be useful. Because in RL problem with rich observation, we have enough data sample to generate sounding kernel, and one advantage of kernel related method is it build model irrelevant to the size of observation, which is large or even infinite. Recent work on applicaiton of Reproducing Kernel Hilber Space(RKHS)[Fukumizu et al.,2013] has shows that probability can be viewed as kernel mean(point) on RKHS. And Relation between probability can be expressed as product kernel covariance. A work on adapt Kernel Bayes' Rule on POMDP may provide inspiration on future work in RL with rich observation problem [Nishiyama et al.,2012]. In addition to the above two math base consideration, consider the Mario game example. An intuition to solve this problem may be relying on large memory. When human learn to play new game or face new task, usually human can learn the whole rule within a few attempts. And the reason why human learn this fast may be because human has large memory and can compare the current task with large number of past relevant experiences.

Reference

Krishnamurthy, Akshay, Alekh Agarwal, and John Langford. "PAC reinforcement learning with rich observations." Advances in Neural Information Processing Systems. 2016.

R. I. Brafman and M. Tennenholtz. R-max – a general polynomial time algorithm for near-optimal reinforcement learning. JMLR, 2003.

Dann, Christoph, et al. "On Oracle-Efficient PAC RL with Rich Observations." Advances in Neural Information Processing Systems. 2018.

M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. MLJ, 2002.

Fukumizu, Kenji, Le Song, and Arthur Gretton. "Kernel Bayes' rule: Bayesian inference with positive definite kernels." The Journal of Machine Learning Research 14.1 (2013): 3753-3783.

Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529.

Jiang, Nan, et al. "Contextual decision processes with low Bellman rank are PAC-learnable." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.

Sun, Wen, et al. "Model-Based Reinforcement Learning in Contextual Decision Processes." arXiv preprint arXiv:1811.08540 (2018).

Nishiyama, Yu, et al. "Hilbert space embeddings of POMDPs." arXiv preprint arXiv:1210.4887 (2012).