Distributed training of finite horizon ADP by ADMM algorith

Jiaxin Gao, Yang Guan, Shengbo Eben Li*, Wenyu Li, Fei Ma

***Abstract*—"To be completed"**

***Index Terms*—"To be completed"**

I. Introduction

"To be completed"

II. Problem Formulation and Preliminaries

A. Problem Formulation

In this section, we will define basic reinforcement lenrning concepts, following standard textbook definitions [**?**]. Reinforcement learning addresses the problem of learning to control a dynamical system, in a general sense. The dynamical system is fully defined by a fully-observed or partially-observed Markov decision process(MDP). In this article we will use the fully-observed formalism.

Definition 2.1 (Markov decision process). The Markov decision process is defined as a tuple $\mathcal{M} =< \mathcal{S}, \mathcal{A}, \mathcal{P}, d_0, r, \gamma >$, where $\mathcal{S}$ is a set of states $s \in \mathcal{S}$, which may be either discrete or continuous (i.e., multi-dimensional vectors), $\mathcal{A}$ is a set of actions $a \in \mathcal{A}$, which similarly can be discrete or continuous, $\mathcal{P}$ defines a conditional probability of the form $\mathcal{P}(s_{t+1}|s_t, a_t)$ that describes the dynamics of the system. d_0 defines the initial state distribution $d_0(s_0)$, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ defines a reward function, and $\gamma \in (0, 1]$ is a scalar discount factor.

The final goal in a reinforcement learning problem is to learn a police, which defines a distribution over actions conditioned on states, $\pi(a_t|s_t)$. From the definitions, we can derive the trajectory distribution. The trajectory is a sequence of states and actions of length T, given by $\tau = (s_0, a_0, ..., s_T, a_T)$, where T may be infinite. The trajectory distribution p_π for a given MDP $\mathcal{M}$ and police π is given by

$$p_\pi(\tau) = d_0(s_0) \prod_{t=0}^{T} \pi(a_t|s_t) P(s_t + 1|s_t, a_t). \tag{1}$$

This work is supported by International Science & Technology Cooperation Program of China under 2019YFE0100200, Beijing Natural Science Foundation with JQ18010 and National Key Research and Development Program of China (Grant No. 2018YFC0810500). This work is also partially supported by Tsinghua EE Xilinx AI Research Fund.

J. Gao, F Ma are with the School of Mechanical Engineering, University of Science and Technology Beijing, Beijing, 100083, China. (gaojiaxin2017@163.com, yeke@ustb.edu.cn).

S. Li, Y. Guan and W. Li are with the State Key Lab of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University, Beijing, 100084, China. (lishbo@tsinghua.edu.cn, liwenyu@mail.tsinghua.edu.cn).

The corresponding author is Shengbo Eben Li. All questions about this paper should be sent to the email lishbo@tsinghua.edu.cn.

The reinforcement learning objective, $\mathcal{J}(\pi)$, can then be written as an expectation under this trajectory distribution:

$$\mathcal{J}(\pi) = \mathbb{E}_{\tau \backsim p_\pi(\tau)}[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)]. \tag{2}$$

B. Introduction of ADMM Algorithm

ADMM algorithm is used to solve consensus optimization problem. ADMM is an algorithm that solves problem in the following form (See [**?**], especially in Chapter 3 and 7):

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \\ &\text{s.t.} \quad A\mathbf{x} + B\mathbf{z} = \mathbf{c} \end{aligned} \tag{3}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^m \to \mathbb{R}$, both are closed proper convex functions, and $\mathbf{x} \in \mathbb{R}^n, \mathbf{z} \in \mathbb{R}^m, A \in \mathbb{R}^{p\times n}, B \in \mathbb{R}^{p\times m}, \mathbf{c} \in \mathbb{R}^p$. A and B are matrices that represent consensus constraint between local variable $\mathbf{x}$ and global variable $\mathbf{z}$. The augmented Lagrangian of (3) is

$$\begin{aligned} L(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^\top(A\mathbf{x} + B\mathbf{z} - \mathbf{c}) \\ + \frac{\lambda}{2}\|A\mathbf{x} + B\mathbf{z} - \mathbf{c}\|_2^2 \end{aligned} \tag{4}$$

$\mathbf{y}$ is the dual variable and $\lambda > 0$ is the penalty parameter. ADMM consists of the following iterations:

$$\begin{aligned} \mathbf{x}^{k+1} &:= \arg\min_{\mathbf{x}} L(\mathbf{x}, \mathbf{z}^k, \mathbf{y}^k) \\ \mathbf{z}^{k+1} &:= \arg\min_{\mathbf{z}} L(\mathbf{x}^{k+1}, \mathbf{z}, \mathbf{y}^k) \\ \mathbf{y}^{k+1} &:= \mathbf{y}^k + \lambda(A\mathbf{x}^{k+1} + B\mathbf{z}^{k+1} - \mathbf{c}) \end{aligned} \tag{5}$$

In each iteration the algorithm minimizes $\mathbf{x}^{k+1}$ firstly, then it minimizes $\mathbf{z}^{k+1}$ and updates dual variable $\mathbf{y}^{k+1}$. By adopting $\mathbf{r}^{k+1}$, the primal residual, and $\mathbf{s}^{k+1}$, the dual residual, the termination criterion can be described as follows:

$$\begin{aligned} \|\mathbf{r}^{k+1}\|_2 &= \|A\mathbf{x}^{k+1} + B\mathbf{z}^{k+1} - \mathbf{c}\|_2 \leqslant \epsilon^{\text{pri}} \\ \|\mathbf{s}^{k+1}\|_2 &= \|\lambda A^\top B(\mathbf{z}^{k+1} - \mathbf{z}^k)\|_2 \leqslant \epsilon^{\text{dual}} \end{aligned} \tag{6}$$

where $\epsilon^{\text{pri}} > 0$ and $\epsilon^{\text{dual}} > 0$ are tolerance error of the primal and dual feasibility conditions, and can be chosen according to the absolute and relative criteria, which implies

$$\begin{aligned} \epsilon^{\text{pri}} &= \sqrt{p}\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \max\{\|A\mathbf{x}^k\|_2, \|B\mathbf{z}^k\|_2, \mathbf{c}\} \\ \epsilon^{\text{dual}} &= \sqrt{n}\epsilon^{\text{abs}} + \epsilon^{\text{rel}}\|A^\top \mathbf{y}^k\|_2 \end{aligned} \tag{7}$$

where $\epsilon^{\text{abs}} > 0$ is an absolute tolerance and $\epsilon^{\text{rel}} > 0$ is a relative tolerance.

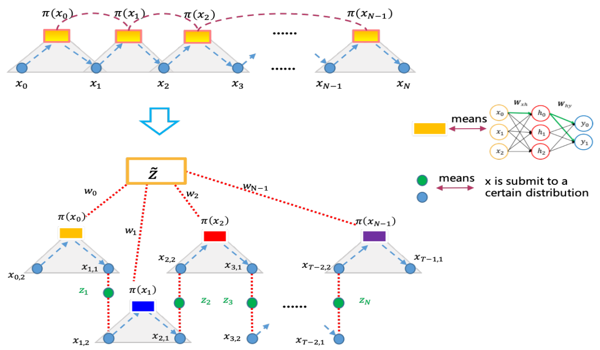

Fig. 1. Distributed ADP diagram.

III. Distributed training of finite horizon ADP

Consider a finite horizon approximate dynamic programming (ADP):

$$\begin{aligned} &\min_{\theta} \mathbb{E}_{x_0\sim\mathcal{U}(\mathcal{B})}\{\sum_{j=0}^{T-1} l(x_j,\pi_\theta(x_j))\} \\ s.t.\quad & x_{j+1}=f(x_j,\pi_\theta(x_j)) \\ & j=0,...,T-1 \end{aligned} \tag{8}$$

where $\mathcal{B}=\{x_0^0,\cdots,x_0^{N-1}\}$ We assume that there are only a finite number of initial values, and N represents the number of possible initial values., and $\mathcal{U}$ is the uniform distribution over $\mathcal{B}$. where d_0 defines the initial state distribution $d_0(s_0)$, the state space dimension is $|S|$, the action space dimension is $|A|$, and the parameter space dimension is $|\theta|$, $l(x_j,\pi_\theta(x_j))$ is a cost function.

Due to dynamic equation $x_{j+1}=f(x_j,\pi_\theta(x_j))$, there is coupling in the transition of state sequence before and after. In this paper, the relaxation variables z_i are introduced to solve the coupling between adjacent states by expanding the dimension, so that each state is independent and can be solved in parallel. Relaxation variables z_i to every state x_i except x_0 and x_{T-1} are introduced to split problem 8. In addition, a relaxation variables z_θ is introduced into the policy network, as shown in Fig.1

Rewrite (8) as:

$$\begin{aligned} &\min_{\theta_j,z_\theta,x_{j,1}^i,x_{j,2}^i,z_j^i} \frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{T-1} l(x^i_{(j,2)},\pi_{\theta_j}(x^i_{(j,2)})) \\ s.t.\quad & x^i_{(j+1,1)}=f(x^i_{(j,2)},\pi_{\theta_j}(x^i_{j,2})) \\ & z_j^i=x_{j,1}^i \\ & z_j^i=x_{j,2}^i \\ & z_\theta=\theta_j \\ & j=0,...,T-1 \\ & i=0,...,N-1 \end{aligned} \tag{9}$$

Define indicator function

$$\begin{aligned} & I_j(\mathbf{x})=\begin{cases} 0,\text{if} \quad x_{j+1,1}^i=f(x_{j,2}^i,\pi_{\theta_j}(x_{j,2}^i)) \\ \infty,\text{else}\end{cases} \\ & i=0,\cdots,N-1 \\ & j=0,\dots,T-1 \end{aligned} \tag{10}$$

where

$$\begin{aligned} &\mathbf{x}=[\theta_0,\cdots,\theta_{T-1},x_{0,1}^i,\cdots,x_{T-1,1}^i,x_{0,2}^i,\cdots,x_{T-1,2}^i] \\ &\mathbf{z}=[z_\theta,\cdots,z_\theta,z_0^i,\cdots,z_{T-1}^i,z_0^i,\cdots,z_{T-1}^i] \\ &i=0,\cdots N-1 \end{aligned} \tag{11}$$

$\mathbf{x}$ is ordered expansion vector of θ and x, $\mathbf{z}$ is ordered expansion vector of z_θ and z

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{z}} \frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{T-1} l(x^i_{(j,2)},\pi_{\theta_j}(x^i_{(j,2)}))+\sum_{j=0}^{T-1} I_j(\mathbf{x}) \\ s.t.\quad & \mathbf{x}-\mathbf{z}=0 \end{aligned} \tag{12}$$

Define lagrange variable:

$$\begin{aligned} &\mathbf{y}=[y_{\theta,0},\cdots,y_{\theta,T-1},y_{0,1}^i,\cdots,y_{T-1,1}^i,y_{0,2}^i,\cdots,y_{T-1,2}^i] \\ &i=0,...,N-1 \end{aligned} \tag{13}$$

where $\mathbf{y}$ is ordered expansion vector of y_θ and y. Introduce $y_{j,1}^i, y_{j,2}^i$ and $y_{\theta,j}$, and rewrite (12) as augment lagrange function $L(\mathbf{x},\mathbf{z},\mathbf{y})$ as:

$$\begin{aligned} & L(\mathbf{x},\mathbf{z},\mathbf{y}) \\ = \; & \frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{T-1} l(x^i_{j,2},\pi_{\theta_j}(x^i_{j,2}))+\sum_{j=0}^{T-1} I_j(\mathbf{x}) \\ + \; & \sum_{i=0}^{N-1}\sum_{j=0}^{T-1}(y_{j,1}^i)^\top(z_j^i-x_{j,1}^i)+\sum_{i=0}^{N-1}\sum_{j=0}^{T-1}(y_{j,2}^i)^\top(z_j^i-x_{j,2}^i) \\ + \; & \sum_{j=0}^{T-1}(y_{\theta,j})^\top(z_\theta-\theta_j)+\frac{\rho}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{T-1}\|z_j^i-x_{j,1}^i\|^2 \\ + \; & \frac{\rho}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{T-1}\|z_j^i-x_{j,2}^i\|^2+\frac{\rho}{2}\sum_{j=0}^{T-1}\|z_\theta-\theta_j\|^2 \end{aligned} \tag{14}$$

Further more, we expand formulation (14) as :

$$\begin{aligned} & L_0(\mathbf{x},\mathbf{z},\mathbf{y}) \\ = \; & \frac{1}{N}\sum_{i=0}^{N-1} l(x^i_{0,2},\pi_{\theta_0}(x^i_{0,2}))+I_0(\mathbf{x})+\sum_{i=0}^{N-1}(y_{1,1}^i)^\top(z_1^i-x_{1,1}^i) \\ + \; & y_{\theta,0}^\top(z_\theta-\theta_0)+\frac{\rho}{2}\sum_{i=0}^{N-1}\|z_1^i-x_{1,1}^i\|^2+\frac{\rho}{2}\|z_\theta-\theta_0\|^2 \end{aligned} \tag{15}$$

$$\begin{aligned} & L_j(\mathbf{x},\mathbf{z},\mathbf{y}) \\ = \; & \frac{1}{N}\sum_{i=0}^{N-1} l(x^i_{j,2},\pi_{\theta_j}(x^i_{j,2}))+I_j(\mathbf{x}) \\ + \; & \sum_{i=0}^{N-1}(y_{j+1,1}^i)^\top(z_{j+1}^i-x_{j+1,1}^i)+\sum_{i=0}^{N-1}(y_{j,2}^i)^\top(z_j^i-x_{j,2}^i) \\ + \; & (y_{\theta,j})^\top(z_\theta-\theta_j)+\frac{\rho}{2}\sum_{i=0}^{N-1}\|z_{j+1}^i-x_{j+1,1}^i\|^2 \\ + \; & \frac{\rho}{2}\sum_{i=0}^{N-1}\|z_j^i-x_{j,2}^i\|^2+\frac{\rho}{2}\|z_\theta-\theta_j\|^2 \end{aligned} \tag{16}$$

$$
\begin{aligned}
& L_{T-1}(\mathbf{x},\mathbf{z},\mathbf{y}) \\
= \quad & \frac{1}{N}\sum_{i=0}^{N-1} l(x^i_{T-1,2}, \pi_{\theta_{T-1}}(x^i_{T-1,2})) + I_{T-1}(\mathbf{x}) \\
+ \quad & \sum_{i=0}^{N-1} (y^i_{T-1,2})^\top (z^i_{T-1} - x^i_{T-1,2}) + (y_{\theta,T-1})^\top (z_\theta - \theta_{T-1}) \\
+ \quad & \frac{\rho}{2}\sum_{i=0}^{N-1} \|z^i_{T-1} - x^i_{T-1,2}\|^2 + \frac{\rho}{2}\|z_\theta - \theta_{T-1}\|^2
\end{aligned}
\tag{17}
$$

Using ADMM update scheme, sub-problems are parallel solved.

$$
\begin{aligned}
\mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} L(\mathbf{x},\mathbf{z}^k,\mathbf{y}^k) \\
\mathbf{z}^{k+1} &= \arg\min_{\mathbf{z}} L(\mathbf{x}^{k+1},\mathbf{z},\mathbf{y}^k) \\
\mathbf{y}^{k+1} &= \mathbf{y}^k + \rho(\mathbf{x}^{k+1} - \mathbf{z}^{k+1})
\end{aligned}
\tag{18}
$$

IV. Conclusion

”To be completed”