Artificial Intelligence and Robotics Research 人工智能与机器人研究, 2019, 8(2), 46-56
Published Online May 2019 in Hans. http://www.hanspub.org/journal/airr
https://doi.org/10.12677/airr.2019.82006

Hans汉斯

A New Data-Driven Neural Dynamic Programming Algorithm

Xingke Li, Xuesong Chen

School of Applied Mathematics, Guangdong University of Technology, Guangzhou Guangdong
Email: chenxs@gdut.edu.cn

Received: Feb. 27th, 2019; accepted: Mar. 13th, 2019; published: Mar. 20th, 2019

Abstract

A new data-driven neural dynamic programming method for model-free discrete-time nonlinear dynamic system is proposed in this paper. The residual of the Q-function and the control strategy are operated to be zero with the basis function through the inner product. Then the coefficients of the neural network are updated by the offline trained data and the online data. Finally the optimal control strategy is obtained and the convergence of this algorithm is proved.

Keywords

Optimal Control, Neural Dynamic Programming, Q-Function, Neural Network

一种新的基于数据驱动的神经动态规划方法

李星科，陈学松

广东工业大学应用数学学院，广东 广州
Email: chenxs@gdut.edu.cn

收稿日期：2019年2月27日；录用日期：2019年3月13日；发布日期：2019年3月20日

摘　要

为了实现无模型离散时间非线性动态系统的最优控制，提出了一种新的基于数据驱动的神经动态规划方法。该方法利用Q函数的残差与基函数的内积为零，同时控制策略的残差与基函数的内积也为零，从而得到控制方程。接着使用离线数据集与在线数据来迭代更新神经网络的系数，从而得到近似最优的控制策略，本文还证明了该算法是收敛的。

文章引用：李星科，陈学松．一种新的基于数据驱动的神经动态规划方法[J]．人工智能与机器人研究，2019，8(2): 46-56. DOI: 10.12677/airr.2019.82006

关键词

最优控制，神经动态规划，*Q*函数，神经网络

1. 引言

无论是在控制理论还是工程领域，最优化控制问题都是一个重难点。最优控制问题的求解依赖于求解 HJB 方程，由于系统一般是非线性的，导致对于求解 HJB 方程很困难，难以得到解析解[1]。

Bellman 在 1957 年首次提出了动态规划方法。动态规划对于求解最优控制问题是一个行之有效的方法。该方法适应广泛：离散系统，连续系统，非线性系统，以及随机系统等等。但是该方法是一种逆向计算方法，随着解的时间维度增加，且由于控制变量维数大容易造成“维数灾难”问题。于是近年来，针对未知的复杂非线性系统的最优控制问题，自适应动态规划应运而生。与传统的动态规划相比，由于其采用函数近似结构来逼近系统模型，评价指标[2]，和控制策略，避免了因时间逆向计算造成的“维数灾难”问题。自适应动态规划可用于大规模非线性系统[3]，其应用也相当广泛；例如：机器人运动规划[4]，无人驾驶汽车[5]，污水处理系统[6]等等。

LIU 等采用值迭代的自适应动态规划，并证明其收敛条件是迭代性能指标函数初始化为任意半正定函数[7]。根据此收敛条件，提出了一种基于自适应动态规划的协同优化算法[8]。该协同优化算法令迭代的残差快速减小，大幅提高了自适应动态规划的收敛速度。LIU 等对于离散的非线性系统采用基于策略迭代的自适应动态规划方法[9]，证明了该迭代控制律可以使系统稳定，并说明其是收敛的。LUO 等则分别对于连续时间非线性系统采用基于策略迭代的自适应动态规划方法[10]，该策略迭代的实现使用基于神经网络的最小二乘方法并证明该方法是收敛的。文献[11]针对离散时间非线性系统，提出了策略梯度自适应动态规划算法，并证明了该算法中 Q 函数能收敛到最优值。

由于神经网络良好的自适应与自学习等特点[12]，本文采用神经网络与动态规划结合的自适应动态规划对无模型的最优控制问题进行求解。对于神经网络的权重系数采用离线数据集与在线数据结合进行更新。

2. 问题描述

本文所描述的离散时间非线性系统为：

$$x_{k+1} = f\left(x_k + u_k\right), k = 0,1,2,\cdots \tag{1}$$

其中 $x_k \in R^n$ 为状态向量，$u_k \in R^m$ 为控制向量。此外 X 和 U 为一个完备集合，$D = \left\{\left(x,u\right) \mid x \in X, u \in U\right\}$。系统(1)在 X 上稳定，函数 $f\left(x,u\right)$ 在集合 D 上连续，且 $f\left(0,0\right) = 0$，$x = 0$ 是系统(1)的一个稳定状态。本文考虑的是无模型的最优控制问题，即函数 $f\left(x,u\right)$ 的具体解析式是不知道的。优化目标是找到一个稳定的反馈控制 $u_k = u\left(x_k\right)$，使得性能指标函数(2)最小。

$$V_u\left(x_0\right) = \sum_{l=0}^{\infty} P\left(x_l, u_l\right). \tag{2}$$

其中 $P\left(x,u\right) = S\left(x\right) + W\left(u\right)$，$S\left(x\right)$ 和 $W\left(u\right)$ 为正定函数。

3. 策略梯度自适应动态规划

对于上述最优控制问题，由于该系统是非线性的且系统模型 $f(x,u)$ 的解析式并不知道。为了克服这些困难，引入了策略梯度自适应动态规划来求解该最优控制问题。

定义 1 对于系统(1)，如果 $u(x)$ 在 X 上连续，且 $\forall x \in X$，$V_u(x) < \infty$，则称控制 $u(x)$ 是可控的，记 $u(x) \in U(X)$。基于自适应动态规划理论，定义值函数 $V_u(x_k)$ 为：

$$V_u(x_k) = \sum_{l=k}^{\infty} P(x_l, u_l). \tag{3}$$

易知：

$$V_u(x_k) = P(x_k, u_k) + V_u(x_{k+1}). \tag{4}$$

记最优值函数为：

$$V^*(x) = \min_u V_u(x). \tag{5}$$

对应的最优控制为：

$$u^*(x) = \arg\min_u V_u(x_k). \tag{6}$$

根据自适应动态规划，为了更好求解该最优控制问题，下面引入 Q 函数：

$$Q_u(x_k, a) = P(x_k, a) + \sum_{l=k+1}^{\infty} P(x_l, u_l). \tag{7}$$

且易知 $Q_u(x_k, u) = V_u(x_k)$，Q 函数也可以表示为：

$$Q_u(x_k, a) = P(x_k, a) + Q_u(x_{k+1}, u) = P(x_k, a) + V_u(x_{k+1}). \tag{8}$$

对应最优控制 $u^*(x)$ 的最优 Q 函数为：

$$Q^*(x_k, a) = Q_{u^*}(x_k, a) = P(x_k, a) + V^*(x_{k+1}). \tag{9}$$

相应的最优控制 $u^*(x)$ 为：

$$u^*(x) = \arg\min_u V_u(x) = \arg\min_a Q^*(x, a). \tag{10}$$

为了求解该最优控制问题，通过迭代的思想，希望利用 Q 函数的梯度信息来更新控制策略 u，然后把控制策略 u 带入 Q 函数。策略梯度自适应动态规划算法的具体步骤如下：

步骤 1：给定初始控制策略 $u^{(0)} \in U(X)$ 和允许误差 ε，且令 $i = 0$。

步骤 2：估计 Q 函数：

$$Q^{(i)}(x_k, a) = P(x_k, a) + Q^{(i)}(x_{k+1}, u^{(i)}). \tag{11}$$

步骤 3：更新控制策略 u：

$$u^{(i+1)}(x) = u^{(i)}(x) - \alpha \nabla_a Q^{(i)}(x, a)\big|_{a=u^{(i)}}. \tag{12}$$

其中 α 为常数。

步骤 4：若 $\left\| Q^{(i)}(x_k, a) - Q^{(i+1)}(x_k, a) \right\| \le \varepsilon$，则输出 Q 函数和控制策略，否则令 $i = i + 1$，返回步骤 2。

对于系统(1)，首先定义其 Hamiltonian 函数为：

$$\begin{aligned} H(x_k, u, V) &= V(x_{k+1}) - V(x_k) + P(x_k, u) \\ &= V(f(x_k, u)) - V(x_k) + P(x_k, u). \end{aligned} \tag{13}$$

且 $V^{(i)}\left(x_k\right)=V_{u^{(i)}}\left(x_k\right)=\sum_{l=k}^{\infty}P\left(x_l,u^{(i)}\left(x_l\right)\right)$。

引理 1 [11] 给定 $u^{(0)}\left(x\right)\in U\left(X\right)$，根据策略梯度自适应动态规划算法得到控制序列 $\left\{u^{(i)}\left(x\right)\right\}$。假设 $\nabla_a V^{(i)}\left(f\left(x_k,a\right)\right)$，$\nabla_{aa}V^{(i)}\left(f\left(x_k,a\right)\right)$，$\nabla_a W\left(a\right)$ 和 $\nabla_{aa}W\left(a\right)$ 存在，且 $A=\nabla_u^{\mathrm{T}}H\nabla_{uu}H\nabla_u H$，$B=\nabla_u^{\mathrm{T}}H\nabla_u H$，$C=h$。$\underline{\alpha}=\dfrac{B-\sqrt{B^2-AC}}{A}$，$\bar{\alpha}=\dfrac{B+\sqrt{B^2-AC}}{A}$。如果对于 $\forall i,x,a$，$B^2-AC\geq 0$ 且 $\underline{\alpha}\leq\alpha\leq\bar{\alpha}$，则：

1) $H\left(x_k,u^{(i+1)},V^{(i)}\right)\leq 0$，

2) $u^{(i)}\in U\left(X\right)$。

证明：1) 将 Hamiltonian 函数 $H\left(x_k,u,V^{(i)}\right)$ 在 $u^{(i)}$ 处进行二阶泰勒展开：

$$\begin{aligned}H\left(x_k,u,V^{(i)}\right)&=H\left(x_k,u^{(i)},V^{(i)}\right)+\left(u-u^{(i)}\right)^{\mathrm{T}}\nabla_u H\left(x_k,u^{(i)},V^{(i)}\right)\\&\quad+\frac{1}{2}\left(u-u^{(i)}\right)^{\mathrm{T}}\nabla_{uu}H\left(x_k,u^{(i)},V^{(i)}\right)\left(u-u^{(i)}\right)+h\left(u\right).\end{aligned}\tag{14}$$

根据式(8)和(13)易知：

$$\nabla_a Q^{(i)}\left(x_k,a\right)=\nabla_a V^{(i)}\left(f\left(x_k,a\right)\right)+\nabla_a W\left(a\right)=\nabla_u H\left(x_k,u,V^{(i)}\right).\tag{15}$$

$$\nabla_{aa} Q^{(i)}\left(x_k,a\right)=\nabla_{aa} V^{(i)}\left(f\left(x_k,a\right)\right)+\nabla_{aa} W\left(a\right)=\nabla_{uu} H\left(x_k,u,V^{(i)}\right).\tag{16}$$

由于 $\nabla_a V^{(i)}\left(f\left(x_k,a\right)\right),\nabla_{aa}V^{(i)}\left(f\left(x_k,a\right)\right),\nabla_a W\left(a\right),\nabla_{aa}W\left(a\right)$，对于 $\forall i$ 都存在，因此 $\nabla_u H,\nabla_{uu}H$ 都存在。由式(12)和(14)~(16)易知：

$$\begin{aligned}&H\left(x_k,u^{(i+1)},V^{(i)}\right)\\&=H\left(x_k,u^{(i)},V^{(i)}\right)+\left(u^{(i+1)}-u^{(i)}\right)^{\mathrm{T}}\nabla_u H\left(x_k,u^{(i)},V^{(i)}\right)\\&\quad+\frac{1}{2}\left(u^{(i+1)}-u^{(i)}\right)^{\mathrm{T}}\nabla_{uu}H\left(x_k,u^{(i)},V^{(i)}\right)\left(u^{(i+1)}-u^{(i)}\right)+h\left(u^{(i+1)}\right)\\&=H\left(x_k,u^{(i)},V^{(i)}\right)-\alpha\nabla_a^T Q^{(i)}\left(x_k,u^{(i)}\right)\nabla_u H(x_k,u^{(i)},V^{(i)})\\&\quad+\frac{1}{2}\alpha^2\nabla_a^{\mathrm{T}}Q^{(i)}\left(x_k,u^{(i)}\right)\nabla_{uu}H\left(x_k,u^{(i)},V^{(i)}\right)\nabla_a Q^{(i)}\left(x_k,u^{(i)}\right)+h\left(u^{(i+1)}\right)\\&=H\left(x_k,u^{(i)},V^{(i)}\right)-\alpha\nabla_a^{\mathrm{T}}H\left(x_k,u^{(i)},V^{(i)}\right)\nabla_u H\left(x_k,u^{(i)},V^{(i)}\right)\\&\quad+\frac{1}{2}\alpha^2\nabla_a^T H\left(x_k,u^{(i)},V^{(i)}\right)\nabla_{uu}H\left(x_k,u^{(i)},V^{(i)}\right)H\left(x_k,u^{(i)},V^{(i)}\right)+h\left(u^{(i+1)}\right).\end{aligned}$$

令 $A=\nabla_u^{\mathrm{T}}H\nabla_{uu}H\nabla_u H,B=\nabla_u^{\mathrm{T}}H\nabla_u H,C=h$，且 $A\geq 0$，则：

$$H\left(x_k,u^{(i+1)},V^{(i)}\right)=H\left(x_k,u^{(i)},V^{(i)}\right)+\frac{1}{2}A\alpha^2-B\alpha+C.\tag{17}$$

如果 $B^2-AC\geq 0$，当 $\underline{\alpha}\leq\alpha\leq\bar{\alpha}$ 时，其中 $\underline{\alpha}=\dfrac{B-\sqrt{B^2-AC}}{A}$，$\bar{\alpha}=\dfrac{B+\sqrt{B^2-AC}}{A}$，则：

$$\frac{1}{2}A\alpha^2-B\alpha+C\leq 0.\tag{18}$$

即 $H\left(x_k,u^{(i+1)},V^{(i)}\right)\leq H\left(x_k,u^{(i)},V^{(i)}\right)$。根据 Hamiltonian 函数定义知：$H\left(x_k,u^{(i)},V^{(i)}\right)=0$。

故：$H\left(x_k,u^{(i+1)},V^{(i)}\right)\leq 0$。

2) 假设 $u^{(i)}(x)\in U(X)$，对于系统 $x_{k+1}=f\left(x_k,u^{(i+1)}\right)$，则对于 Lyapunov 函数 $V^{(i)}(x_k)$ 有：

$$\begin{aligned}\Delta V^{(i)}(x_k)&=V^{(i)}\left(f\left(x_k,u^{(i+1)}\right)\right)-V^{(i)}(x_k)\\&=V^{(i)}\left(f\left(x_k,u^{(i+1)}\right)\right)-V^{(i)}(x_k)+P\left(x_k,u^{(i+1)}\right)-P\left(x_k,u^{(i+1)}\right)\\&=H\left(x_k,u^{(i+1)},V^{(i)}\right)-P\left(x_k,u^{(i+1)}\right).\end{aligned}\tag{19}$$

由 $H\left(x_k,u^{(i+1)},V^{(i)}\right)\le 0$，知：$\Delta V^{(i)}(x_k)\le -P\left(x_k,u^{(i+1)}\right)\le 0$。即：$V^{(i)}\left(f\left(x_k,u^{(i+1)}\right)\right)\le V^{(i)}(x_k)<\infty$。

对于所有 $x,u\ne 0$，根据定义 1 知：$u^{(i+1)}(x)\in U(X)$。由数学归纳法知，当 $u^{(0)}(x)\in U(X)$，有 $u^{(i)}(x)\in U(X)$。

引理 2 [11] 对于所有 $(x,a)\in \mathrm{D}$，根据策略梯度自适应动态规划算法得到序列 $\left\{Q^{(i)}(x,a)\right\}$ 和 $\left\{u^{(i)}(x)\right\}$ 满足：1) $Q^{(i)}(x,a)\ge Q^{(i+1)}(x,a)\ge Q^{*}(x,a)$，

2) $\lim\limits_{i\to\infty}Q^{(i)}(x,a)=Q^{*}(x,a)$。

证明 1)由式(4)和(8)知：

$$\begin{aligned}Q^{(i)}\left(x_k,u^{(i+1)}\right)&=P\left(x_k,u^{(i+1)}\right)+V^{(i)}\left(f\left(x_k,u^{(i+1)}\right)\right)\\&=P\left(x_k,u^{(i+1)}\right)+V^{(i)}\left(f\left(x_k,u^{(i+1)}\right)\right)-V^{(i)}(x_k)+V^{(i)}(x_k)\\&=H\left(x_k,u^{(i+1)},V^{(i)}\right)+V^{(i)}(x_k)\\&\le H\left(x_k,u^{(i)},V^{(i)}\right)+V^{(i)}(x_k)\\&=V^{(i)}(x_k).\end{aligned}\tag{20}$$

且：

$$\begin{aligned}V^{(i+1)}(x_k)&=P\left(x_k,u^{(i+1)}\right)+V^{(i)}(x_{k+1})-V^{(i)}(x_{k+1})+V^{(i+1)}(x_{k+1})\\&=Q^{(i)}\left(x_k,u^{(i+1)}\right)-V^{(i)}(x_{k+1})+V^{(i+1)}(x_{k+1})\\&\le V^{(i)}(x_k)-V^{(i)}(x_{k+1})+V^{(i+1)}(x_{k+1})\\&=P\left(x_k,u^{(i)}\right)+V^{(i+1)}(x_{k+1})\\&\le P\left(x_k,u^{(i)}\right)+P\left(x_{k+1},u^{(i)}\right)+V^{(i+1)}(x_{k+2})\\&\le \sum_{l=k}^{\infty}P\left(x_l,u^{(i)}\right)=V^{(i)}(x_k).\end{aligned}\tag{21}$$

则对于所有 $(x_k,a)\in \mathrm{D}$，有：

$$\begin{aligned}Q^{(i+1)}(x_k,a)&=P(x_k,a)+V^{(i+1)}(x_{k+1})\\&\le P(x_k,a)+V^{(i)}(x_{k+1})\\&=Q^{(i)}(x_k,a).\end{aligned}\tag{22}$$

由于式(9)知：

$$\begin{aligned}Q^{(i)}(x_k,a)&=P(x_k,a)+V^{(i)}(x_{k+1})\\&\ge P(x_k,a)+V^{*}(x_{k+1})\\&=Q^{*}(x_k,a).\end{aligned}\tag{23}$$

故可得：$Q^{(i)}(x,a)\ge Q^{(i+1)}(x,a)\ge Q^{*}(x,a)$。

2) 由式(12)知，当 i 趋于 ∞ 时有：

$$u^{\infty}=u^{\infty}-\alpha\nabla_a Q^{\infty}\left(x,a\right)\big|_{a=u^{\infty}}. \tag{24}$$

即：

$$\alpha\nabla_a Q^{\infty}\left(x,a\right)\big|_{a=u^{\infty}}=0. \tag{25}$$

由式(8)知：

$$\begin{aligned}Q^{\infty}\left(x_k,a\right)&=P\left(x_k,a\right)+Q^{\infty}\left(x_{k+1},u^{\infty}\right)\\&=P\left(x_k,a\right)+V^{\infty}\left(x_{k+1}\right)\\&=P\left(x_k,a\right)+V^{\infty}\left(f\left(x_k\right)\right).\end{aligned} \tag{26}$$

则有：

$$\nabla_a W\left(a\right)\big|_{a=u^{\infty}}+\nabla_a V^{\infty}\left(f\left(x_k,a\right)\right)\big|_{a=u^{\infty}}=0. \tag{27}$$

又由式(4)知：

$$V^{*}\left(x_k\right)=P\left(x_k,u^{*}\right)+V^{*}\left(x_{k+1}\right). \tag{28}$$

上式对 u 求导得：

$$\nabla_u W\left(u\right)\big|_{u=u^{*}}+\nabla_u V^{*}\left(f\left(x_k,u\right)\right)\big|_{u=u^{*}}=0. \tag{29}$$

易知式(27)和(29)是一样的，由唯一性得：

$$V^{\infty}\left(x\right)=V^{*}\left(x\right). \tag{30}$$

由式(26)得：

$$\begin{aligned}Q^{\infty}\left(x_k,a\right)&=P\left(x_k,a\right)+V^{\infty}\left(x_{k+1}\right)\\&=P\left(x_k,a\right)+V^{*}\left(x_{k+1}\right)\\&=Q^{*}\left(x_k,a\right).\end{aligned} \tag{31}$$

由结论 1)知，序列 $\left\{Q^{(i)}\left(x,a\right)\right\}$ 是单减序列，记 $\lim\limits_{i\to\infty}Q^{(i)}\left(x,a\right)=Q^{\infty}\left(x,a\right)$，则有：$\lim\limits_{i\to\infty}Q^{(i)}\left(x,a\right)=Q^{*}\left(x,a\right)$。

4. 基于数据驱动的神经动态规划及其实现

首先定义一个数据 $\left(x,a,\tilde{x}\right)$ 其中 $\tilde{x}$ 为当前状态 x 执行控制动作 a 后得到状态向量。$\tilde{x}$ 的获得是通过实际系统输入控制动作 a 后得到，而不是使用系统模型 f 的数学解析式。如图 1，是策略梯度自适应动态规划算法的结构。其中包括两个部分：离线数据集和在线数据。离线数据集 $S_M=\left\{\left(x_l,a_l,\tilde{x}_l\right)\middle|\left(x_l,a_l\right)\in \mathrm{D},\tilde{x}_l\in X\right\},l=1,2,\cdots,M$。$M$ 为数据的数量，其中离线数据集 S_M 可以通过实际系统随机采样获得，其结构如图 2。在线数据 $s_k=\left(x_{k-1},u_{k-1},x_k\right)$，其分别是在时刻 $k-1$ 和 k 的在线状态和控制信息，其获得的结构图如图 3。图 1 的大概流程是：首先，给定一个初始控制 $u^{(0)}$，根据离线数据集 S_M 通过策略梯度自适应动态规划算法步骤 2，得到 $Q^{(0)}\left(x,a\right)$。其次，将初始控制 $u^{(0)}$ 通过实际系统作用于当前状态 x_0，得到下一状态 x_1，即得到在线数据 s_1。然后，将 $Q^{(0)}\left(x,a\right)$ 和 $u^{(0)}$ 通过策略梯度自适应动态规划算法步骤 3，得到 $u^{(1)}$。最后，将 s_1 加入离线数据集 S_M 作为新的离线数据集，以此重复循环，则可得到在线数据 $s_2,s_3,\cdots$，控制序列 $u^{(2)},u^{(3)},\cdots$ 和 Q 函数序列 $Q^{(2)}\left(x,a\right),Q^{(3)}\left(x,a\right),\cdots$。

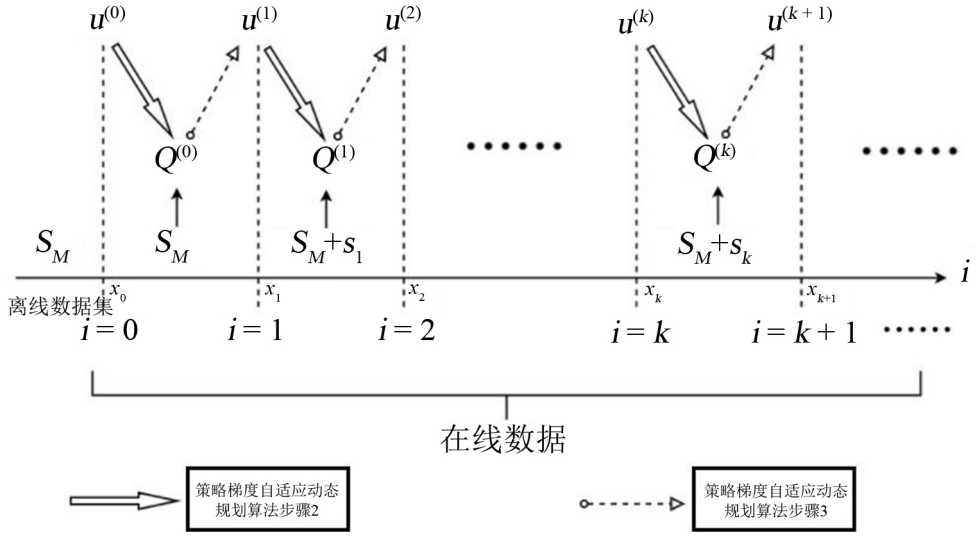

Figure 1. Policy gradient adaptive dynamic programming algorithm
图 1. 策略梯度自适应动态规划算法

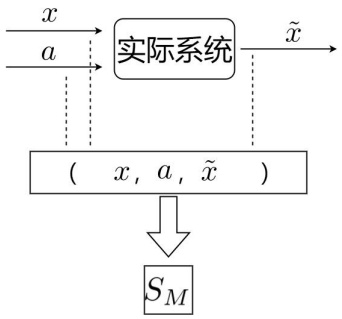

Figure 2. Collect offline data sets
图 2. 收集离线数据集

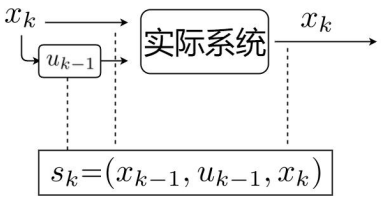

Figure 3. Collect online data
图 3. 收集在线数据

4.1. 神经动态规划的实现设计

由于根据策略梯度自适应动态规划算法，要求去计算未知的 Q 函数 $Q^{(i)}(x,a)$ 和控制策略 $u^{(i)}(x)$，为了实现该过程，下面引入执行－评价神经网络。其中用执行网络来逼近控制策略 $u(x)$，用评价网络来逼近 Q 函数 $Q(x,a)$。首先介绍俩组线性无关的基函数：$\Phi(x)=\{\phi_j(x)\}_{j=1}^{\infty}$，$\Psi(x,a)=\{\psi_j(x,a)\}_{j=1}^{\infty}$。其中 $\phi_j(0)=0, \psi_j(0,0)=0$。根据代数理论，控制策略 $u(x)$ 和 Q 函数 $Q(x,a)$ 可以如下线性表示：

$$u(x)=\sum_{j=1}^{\infty}\upsilon_j\phi_j(x). \tag{32}$$

$$Q(x,a)=\sum_{j=1}^{\infty}\theta_j\psi_j(x,a). \tag{33}$$

根据函数逼近理论，Q 函数 $Q^{(i)}(x,a)$ 和控制策略 $u^{(i)}(x)$ 可以被有限维基函数近似表示：

$$\bar{u}^{(i)}(x)=\sum_{j=1}^{L_1}\upsilon_j^{(i)}\phi_j(x)=\Phi_L^{\mathrm{T}}(x)\upsilon^{(i)}. \tag{34}$$

$$\bar{Q}^{(i)}(x,a)=\sum_{j=1}^{L_2}\theta_j^{(i)}\psi_j(x,a)=\Psi_L^{\mathrm{T}}(x,a)\theta^{(i)}. \tag{35}$$

其中 $\upsilon^{(i)}=\left[\upsilon_1^{(i)}\cdots\upsilon_{L_1}^{(i)}\right]^{\mathrm{T}}$ 和 $\theta^{(i)}=\left[\theta_1^{(i)}\cdots\theta_{L_2}^{(i)}\right]^{\mathrm{T}}$ 分别为执行网络和评价网络的权重系数向量，但其是未知的。$\Phi_L(x)=\left[\phi_1(x)\cdots\phi_{L_1}(x)\right]^{\mathrm{T}}$ 和 $\Psi_L(x,a)=\left[\psi_1(x,a)\cdots\psi_{L_2}(x,a)\right]^{\mathrm{T}}$ 分别为执行网络和评价网络的激活函数向量。则该神经网络的输出可以表示为：

$$\hat{u}^{(i)}(x)=\sum_{j=1}^{L_1}\hat{\upsilon}_j^{(i)}\phi_j(x)=\Phi_L^{\mathrm{T}}(x)\hat{\upsilon}^{(i)}. \tag{36}$$

$$\hat{Q}^{(i)}(x,a)=\sum_{j=1}^{L_2}\hat{\theta}_j^{(i)}\psi_j(x,a)=\Psi_L^{\mathrm{T}}(x,a)\hat{\theta}^{(i)}. \tag{37}$$

其中 $\hat{\upsilon}^{(i)}=\left[\hat{\upsilon}_1^{(i)}\cdots\hat{\upsilon}_{L_1}^{(i)}\right]^{\mathrm{T}}$ 和 $\hat{\theta}^{(i)}=\left[\hat{\theta}_1^{(i)}\cdots\hat{\theta}_{L_2}^{(i)}\right]^{\mathrm{T}}$ 分别为 $\upsilon^{(i)}$ 和 $\theta^{(i)}$ 的近似估计。由于神经网络有误差，在用 $\hat{u}^{(i)}(x)$ 和 $\hat{Q}^{(i)}(x,a)$ 估计 $u^{(i)}(x)$ 和 $Q^{(i)}(x,a)$ 时会产生残差，定义 Q 函数残差为：

$$\begin{aligned}\sigma_Q^{(i)}(x,a,\tilde{x})&=\hat{Q}^{(i)}(x,a)-\hat{Q}^{(i)}\left(\tilde{x},\hat{u}^{(i)}\right)-P(x,a)\\&=\Psi_L^{\mathrm{T}}(x,a)\hat{\theta}^{(i)}-\Psi_L^{\mathrm{T}}\left(\tilde{x},\hat{u}^{(i)}\right)\hat{\theta}^{(i)}-P(x,a)\\&=\Psi_L^{\mathrm{T}}(x,a)\hat{\theta}^{(i)}-\Psi_L^{\mathrm{T}}\left(\tilde{x},\Phi_L^{\mathrm{T}}(\tilde{x})\hat{\upsilon}^{(i)}\right)\hat{\theta}^{(i)}-P(x,a).\end{aligned} \tag{38}$$

下面的目标是，在满足残差趋于 0 的条件下，基于数据驱动来计算 $\hat{\upsilon}^{(i)}$，$\hat{\theta}^{(i)}$。任意 $s_1(x,u)\in U$ 和 $s_2(x,u)\in U$，定义内积 $\left\langle s_1(x,u),s_2(x,u)\right\rangle_D=\int_D s_1(x,u),s_2(x,u)\mathrm{d}(x,u)$。令：

$$\left\langle\Psi_L(x,a),\sigma_Q^{(i)}(x,a,\tilde{x})\right\rangle_D=0. \tag{39}$$

将(38)式带入(39)式得：

$$\begin{aligned}&\left[\left\langle\Psi_L(x,a),\Psi_L^{\mathrm{T}}(x,a)\right\rangle_D-\left\langle\Psi_L(x,a),\Psi_L^{\mathrm{T}}\left(\tilde{x},\Phi_L^{\mathrm{T}}(\tilde{x})\hat{\upsilon}^{(i)}\right)\right\rangle_D\right]\hat{\theta}^{(i)}\\&-\left\langle\Psi_L(x,a),P(x,a)\right\rangle_D=0.\end{aligned} \tag{40}$$

则可得：

$$\begin{aligned}\hat{\theta}^{(i)}=&\left[\left\langle\Psi_L(x,a),\Psi_L^{\mathrm{T}}(x,a)\right\rangle_D-\left\langle\Psi_L(x,a),\Psi_L^{\mathrm{T}}\left(\tilde{x},\Phi_L^{\mathrm{T}}(\tilde{x})\hat{\upsilon}^{(i)}\right)\right\rangle_D\right]^{-1}\\&\times\left\langle\Psi_L(x,a),P(x,a)\right\rangle_D.\end{aligned} \tag{41}$$

在计算 $\hat{\theta}^{(i)}$ 时，其中涉及许多积分，根据蒙特卡洛积分方法，令：$I_{\mathrm{D}}=\int_{\mathrm{D}}\mathrm{d}(x,a)$。首先基于离线数据集 S_M 来计算 $\hat{\theta}^{(0)}$：

$$\begin{aligned}\left\langle\Psi_L(x,a),\Psi_L^{\mathrm{T}}(x,a)\right\rangle_D&=\int_D\Psi_L(x,a)\Psi_L^{\mathrm{T}}(x,a)\mathrm{d}(x,a)\\&=\frac{I_D}{M}\sum_{l=1}^{M}\Psi_L(x_l,a_l)\Psi_L^{\mathrm{T}}(x_l,a_l)\\&=\frac{I_D}{M}\eta_0.\end{aligned} \tag{42}$$

其中 $\eta_0=\sum_{l=1}^{M}\Psi_L(x_l,a_l)\Psi_L^{\mathrm{T}}(x_l,a_l)$。同理可得：

$$\left\langle \Psi_L(x,a), \Psi_L^{\mathrm{T}}\left(\tilde{x}, \Phi_L^{\mathrm{T}}(\tilde{x})\hat{\upsilon}^{(0)}\right)\right\rangle_D = \frac{I_D}{M}\rho_0. \tag{43}$$

$$\left\langle \Psi_L(x,a), P(x,a)\right\rangle_D = \frac{I_D}{M}\xi_0. \tag{44}$$

且 $\rho_0 = \sum_{l=1}^{M} \Psi_L(x_l,a_l)\Psi_L^{\mathrm{T}}\left(\tilde{x}_l, \Phi_L^{\mathrm{T}}(\tilde{x}_l)\hat{\upsilon}^{(0)}\right)$，$\xi_0 = \sum_{l=1}^{M} \Psi_L(x_l,a_l)P(x_l,a_l)$。则可得：

$$\hat{\theta}^{(0)} = (\eta_0 - \rho_0)^{-1}\xi_0. \tag{45}$$

如图 4，基于离线数据集 S_M 可以计算出 $\hat{\theta}^{(0)}$，当 $i=k$ 时，对于在线数据 s_k，此时把在线数据 s_k 加入到离线数据集 S_M 作为新的离线数据集 $S_M + s_k$，且用其来计算 $\hat{\theta}^{(k)}$：

$$\left\langle \Psi_L(x,a), \Psi_L^{\mathrm{T}}(x,a)\right\rangle_D = \frac{I_D}{M+1}\eta_k. \tag{46}$$

$$\left\langle \Psi_L(x,a), \Psi_L^{\mathrm{T}}\left(\tilde{x}, \Phi_L^{\mathrm{T}}(\tilde{x})\hat{\upsilon}^{(k)}\right)\right\rangle_D = \frac{I_D}{M+1}\rho_k. \tag{47}$$

$$\left\langle \Psi_L(x,a), P(x,a)\right\rangle_D = \frac{I_D}{M+1}\xi_k. \tag{48}$$

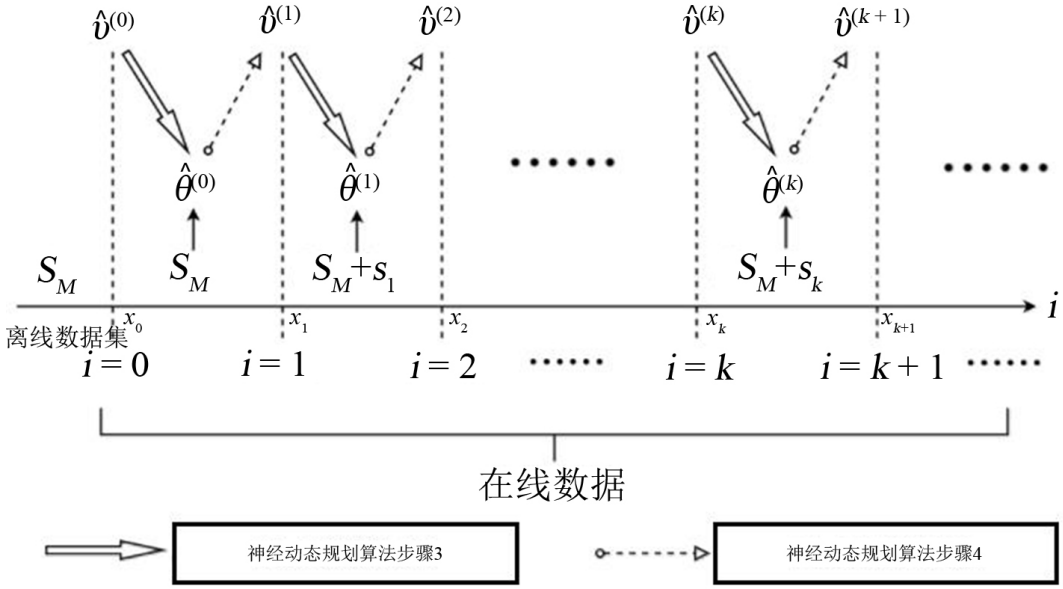

Figure 4. Neural dynamic programming algorithm
图 4. 神经动态规划算法

其中 $\eta_k = \eta_0 + \Psi_L(x_{k-1},a_{k-1})\Psi_L^{\mathrm{T}}(x_{k-1},a_{k-1})$，

$$\rho_k = \sum_{l=1}^{M} \Psi_L(x_l,a_l)\Psi_L^{\mathrm{T}}\left(\tilde{x}_l, \Phi_L^{\mathrm{T}}(\tilde{x}_l)\hat{\upsilon}^{(k)}\right) + \Psi_L(x_{k-1},a_{k-1})\Psi_L^{\mathrm{T}}(x_k,u_k).$$

则由(41)式知，当 $i=k$ 时：

$$\hat{\theta}^{(k)} = (\eta_k - \rho_k)^{-1}\xi_k. \tag{49}$$

接着需要计算执行网络的权重系数 $\hat{\upsilon}^{(i)}$，由神经动态规划算法步骤 3 更新控制策略时，用 $\hat{u}^{(i)}(x)$ 替换 $u^{(i)}(x)$ 会产生误差，定义控制策略残差为：

$$\begin{aligned}\sigma_u^{(i)}(x) &= u^{(i+1)}(x) - u^{(i)}(x) + \alpha\nabla_a Q^{(i)}(x,a)\Big|_{a=u^{(i)}} \\ &= \Phi_L^{\mathrm{T}}(x)\hat{\upsilon}^{(i+1)} - \Phi_L^{\mathrm{T}}(x)\hat{\upsilon}^{(i)} + \alpha\nabla_a \Phi_L^{\mathrm{T}}\left(x, \Phi_L^{\mathrm{T}}(x)\hat{\upsilon}^{(i)}\right)\hat{\theta}^{(i)}.\end{aligned} \tag{50}$$

同理在计算 $\hat{\upsilon}^{(i)}$ 时要满足控制策略残差趋于 0：

$$\left\langle \Phi_L(x), \sigma_u^{(i)}(x)\right\rangle_X = 0. \tag{51}$$

即：

$$\begin{aligned}&\left\langle \Phi_L(x), \Phi_L^{\mathrm{T}}(x)\right\rangle_X \hat{\upsilon}^{(i+1)} - \left\langle \Phi_L(x), \Phi_L^{\mathrm{T}}(x)\right\rangle_X \hat{\upsilon}^{(i)} \\ &+ \alpha\left\langle \Phi_L(x), \nabla_a \Phi_L^{\mathrm{T}}\left(x, \Phi_L^{\mathrm{T}}(x)\hat{\upsilon}^{(i)}\right)\right\rangle_X \hat{\theta}^{(i)} = 0.\end{aligned} \tag{52}$$

则可得：

$$\hat{\upsilon}^{(i+1)} = \hat{\upsilon}^{(i)} - \alpha\left\langle \Phi_L(x), \Phi_L^{\mathrm{T}}(x)\right\rangle_X^{-1} \times \left\langle \Phi_L(x), \nabla_a \Phi_L^{\mathrm{T}}\left(x, \Phi_L^{\mathrm{T}}(x)\hat{\upsilon}^{(i)}\right)\right\rangle_X \hat{\theta}^{(i)}. \tag{53}$$

根据蒙特卡洛积分方法，令：$I_X = \int_X \mathrm{d}x$。基于离线数据集 S_M，当 $i = k$ 时，对于在线数据 s_k，此时把在线数据 s_k 与离线数据集 S_M 结合，且用其来计算 $\hat{\upsilon}^{(k)}$：

$$\left\langle \Phi_L(x), \Phi_L^{\mathrm{T}}(x)\right\rangle_X = \frac{I_X}{M+1}\Gamma_k. \tag{54}$$

$$\left\langle \Phi_L(x), \nabla_a \Psi_L^{\mathrm{T}}\left(x, \Phi_L^{\mathrm{T}}(x)\hat{\upsilon}^{(k)}\right)\right\rangle_X = \frac{I_X}{M+1}F_k. \tag{55}$$

其中 $\Gamma_k = \Gamma_0 + \Phi_L(x_k)\Phi_L^{\mathrm{T}}(x_k)$，且 $\Gamma_0 = \sum_{l=1}^{M}\Phi_L(x_l)\Phi_L^{\mathrm{T}}(x_l)$，

$F_k = \sum_{l=1}^{M}\Phi_L(x_l)\nabla_a \Psi_L^{\mathrm{T}}\left(x_l, \Phi_L^{\mathrm{T}}(x_l)\hat{\upsilon}^{(k)}\right) + \Phi_L(x_k)\nabla_a \Psi_L^{\mathrm{T}}\left(x_k, \Phi_L^{\mathrm{T}}(x_k)\hat{\upsilon}^{(k)}\right)$。则由(53)可得：

$$\hat{\upsilon}^{(k+1)} = \hat{\upsilon}^{(k)} - \alpha\Gamma_k^{-1}F_k\hat{\theta}^{(k)}. \tag{56}$$

4.2. 神经动态规划算法

步骤 1：收集离线数据集 S_M，计算 η_0, ρ_0, ξ_0。

步骤 2：给定一个初始允许误差 ε 和初始控制策略 $\hat{u}^{(i)}(x) \in U(X)$，并令 $i = 0$。

步骤 3：使用 S_M 和 s_{i+1} 计算 η_i, ρ_i, ξ_i。并根据(49)式计算 $\hat{\theta}^{(i)}$。

步骤 4：使用离线数据集 S_M 和在线状态数据 x_i，计算 Γ_i, F_i，并根据(56)式计算 $\hat{\upsilon}^{(i+1)}$。

步骤 5：若 $\left\|\hat{\theta}^{(i)} - \hat{\theta}^{(i+1)}\right\| \le \varepsilon$，则输出 $\hat{\theta}^{(i+1)}$ 和 $\hat{\upsilon}^{(i+1)}$；否则令 $i = i+1$，返回步骤 3，继续循环。

5. 结束语

本文提出了一种基于数据驱动的神经动态规划方法。该方法不依赖于系统的数学解析式，采用神经网络与动态规划结合的方式对最优控制问题进行求解。其分别利用 Q 函数的残差和 Q 函数的基函数做内积为零，控制策略的残差与控制策略的基函数做内积为零；并使用离线数据集与在线数据来迭代更新神经网络的系数，最后得到所需的控制策略。该方法能将离线数据与在线数据有效结合，使得系数更新更加完善。并且证明该算法是收敛的；且收敛到最优值。

基金项目

广东省自然科学基金项目(No.2018A030313505)，广东省科技计划项目(No.2017B010124003, No.2017B090909001)。

参考文献

[1] 张化光, 张欣, 罗艳红, 杨珺. 自适应动态规划综述[J]. 自动化学报, 2013, 39(4): 303-311.

[2] 林小峰, 丁强. 基于评价网络近似误差的自适应动态规划优化控制[J]. 控制与决策, 2015, 30(3): 495-499.

[3] Lakovos, M., Simone, B., Elias, B.K. and Petros, A.L. (2017) Adaptive Optimal Control for Large-Scale Nonlinear Systems. *IEEE Transactions on Automatica Control*, **62**, 5567-5577. https://doi.org/10.1109/TAC.2017.2684458

[4] 赵金刚, 戈新生. 基于动态规划的机器人运动规划最优控制[J]. 控制工程, 2017, 24(11): 2374-2379.

[5] 田涛涛, 侯忠生, 刘世达, 邓志东. 基于无模型自适应动态规划的无人驾驶汽车横向控制方法[J]. 自动化学报, 2017, 43(11): 1931-1940.

[6] 乔俊飞, 王亚清, 柴伟. 基于迭代 ADP 算法的污水处理过程最优控制[J]. 北京工业大学学报, 2018, 44(2): 200-206.

[7] 刘毅, 章云. 基于值迭代的自适应动态规划的收敛条件[J]. 广东工业大学学报, 2017, 34(5): 10-14.

[8] 刘毅, 章云. 一种基于自适应动态规划的协同优化算法[J]. 广东工业大学学报, 2017, 34(6): 15-19.

[9] Liu, D.R. and Wei Q.L. (2014) Policy Iteration Adaptive Dynamic Programming Algorithm for Discrete-Time Nonlinear Systems. *IEEE Transactions on Neural Networks Learning Systems*, 2014, **25**, 621-634. https://doi.org/10.1109/TNNLS.2013.2281663

[10] Luo, B., Wu, H.N., Huang, T.W. and Liu, D.R. (2014) Data Based Approximate Policy Iteration for Affine Nonlinear Continuous-Time Optimal Control Design. *Automatica*, **50**, 3281-3290. https://doi.org/10.1016/j.automatica.2014.10.056

[11] Luo, B., Liu, D.R., Wu, H.N., Wang, D. and Lewis, F.L. (2017) Policy Gradient Adaptive Dynamic Programming for Data-Based Optimal Control. *IEEE Transactions on Cybernrtics*, **47**, 3341-3354. https://doi.org/10.1109/TCYB.2016.2623859

[12] 王鼎, 穆朝絮, 刘德荣. 基于迭代神经动态规划的数据驱动非线性近似最优调节[J]. 自动化学报, 2017, 43(3): 366-375.