Estimating passive dynamics distributions in linearly solvable Markov decision processes from measured immediate costs in reinforcement learning problems

Mauricio A. P. Burdelis (PY), and Kazushi Ikeda

1 Graduate School of Information Science, Nara Institute of Science and Technology
E-mail: \{mauricio-b, kazushi\}@is.naist.jp

Abstract— The Bellman equation in reinforcement learning for Markov decision processes becomes linear under specific conditions, where the passive dynamics is crucial but unknown in general. We propose a method to estimate it from observed immediate costs, by generating systems of linear equations.

Keywords— Reinforcement Learning; Markov Decision Process; Linear Bellman Equation; Passive Dynamics Estimation

1. Introduction

A reinforcement learning (RL) problem is usually formulated on a discrete-time Markov decision process (MDP) with stochastic dynamics, and results in a dynamic programming problem that requires a heavy computational load. Recent works [1] [2] have presented a new class of MDPs, named “Linearly Solvable Markov Decision Processes” (LMDPs), where the Bellman equation can become linear, and optimal solutions can be obtained in an explicit form. The conditions for the class are as follows:

• At any given current state, the agent can specify the state transition probabilities to each of the possible future states directly. The controlled transition probabilities will be represented as \( u(x'|x) \), where \( x' \) is a possible future state, and \( x \) is the current state.

• The total cost incurred in a state transition must be expressed as:

\[
I(x,u) = q(x) + KL(u(\cdot|x) \| p_d(\cdot|x)) \tag{1}
\]

where \( q(x) \) is the state cost of state \( x \) (indicating how undesirable the state \( x \) is) and the second term (also called the action cost) is the Kullback–Leibler (KL) divergence from the controlled state transition distribution \( u(x'|x) \) to the passive dynamics \( p_d(x'|x) \) (which corresponds to the state transition distribution of the system in the absence of controls). Note that the divergence measures the difference between these distributions and it requires \( u(x'|x)=0 \) whenever \( p_d(x'|x)=0 \), in order to keep the KL divergence well-defined and avoid impossible state transitions.

Once the conditions above are met the Bellman equation can be written as:

\[
z(x) = e^{-(q(x))} \sum_{x'} p(x'|x) z(x') \tag{2}
\]

where \( z \) is called the desirability function, and is defined as:

\[
z(x) \equiv e^{-(v(x))} \tag{3}
\]

where \( v \) is the value function, which is the statistical expectation of the total accumulated cost when starting from the argument state \( x \). It is important to notice that the equation (2) is linear in \( z \), and this property reduces the reinforcement learning problem in question to a linear problem, making the estimation (or calculation) of the \( z \) function (and consequently the value function) more efficient than in traditional MDPs.

Usually, the passive dynamics distribution of a given system (to be modeled as an LMDP) is not known, and therefore needs to be estimated by trial and error. In the present work we propose a method for obtaining the passive dynamics distribution of a given system in reinforcement learning problems (to enable modeling as LMDPs) by generating and solving systems of linear equations based on the action cost equation (1).

2. Passive dynamics estimation

Consider a system with a discrete state space \( X \) with cardinality \(|X|=\mathcal{N}\) (i.e. there are \( \mathcal{N} \) possible states).

If each possible transition probability under the passive dynamics distribution for such a system can be considered as a variable, then in the state space there are \( \mathcal{N}^2 \) transition probabilities to be calculated, in order to fully determine the passive dynamics distribution.

Suppose a policy \( u_t \) is given, where no restrictions are assumed except that its transition probabilities are all known, and that for all cases in which \( p_d(x|x')=0 \), \( u_t(x|x') \neq 0 \) (i.e., \( \forall x, \exists x' : p_d(x|x') = 0 \)) then \( u_t(x|x') \neq 0 \) as well. The action cost \( L_t(u_t,x) \) incurred when following the policy \( u_t \) from an arbitrary state \( x_i \) is given by the second term of equation (1), which is the KL divergence from the controlled dynamics distribution \( u_t(x|x') \) to the passive dynamics distribution \( p_d(x|x') \), both starting from state \( x_i \). It can be written as:

\[
L_t(u_t,x) = KL(u_t(\cdot|x) \| p_d(\cdot|x)) = \\
= u_t(x_i|x) \log \left( \frac{u_t(x_i|x)}{p_d(x_i|x)} \right) + ... + u_t(x_N|x) \log \left( \frac{u_t(x_N|x)}{p_d(x_N|x)} \right) \tag{4}
\]

If the action cost \( L_t(u_t,x) \) can be measured and if we apply the logarithm quotient identity to equation (4), we have a linear equation in the variables \( \log(p_d(x|x')) \):

\[
AC_{ui} = u_t(x_i|x) \log p_d(x_i|x) = \sum_{x'} u_t(x_i|x) \log p_d(x_i|x) + ... + u_t(x_N|x) \log p_d(x_N|x) \tag{5}
\]

where the measured value of \( L_t(u_t,x) \) has been denoted as \( AC_{ui} \). Rearranging the terms:

\[
\sum_{x'} u_t(x_i|x) \log p_d(x_i|x) = \sum_{x'} u_t(x_i|x) \log u_t(x_i|x) - AC_{ui} \tag{6}
\]

It is possible to produce up to \( \mathcal{N} \) equations still using the policy \( u_t \), but starting from each existing state \( x_i \) and measuring the incurred action cost \( L_t(u_t,x_i) \) in the transition from the starting state \( x_i \).
Repeating the procedure for $N$ different policies $u_k$ ($k \in \mathbb{N}$, $1 \leq k \leq N$), it is possible to produce $N^2$ different equations and the system of linear equations can finally be solved.

To extend the idea to use total costs, we subtract the state cost $q(x)$, which becomes a new unknown in the system, on both sides of equation (6). By using (1) we have:

$$\sum_{x} u(x, i, j) \log p_d(x, | i, j) - q(x) = \sum_{x} u(x, i, j) \log u(x, i, j) - TC_{ui}$$  

(7)

Where $TC_{ui}$ denotes the measured total cost starting from state $x_i$ and following the policy $u_i$. Unfortunately the obtained system is underdetermined even if we obtain $k+1$ equations. However, it can be solved by using the gradient descent algorithm and normalizing the obtained probabilities to sum up to one at each step.

1. An initial solution for the system is obtained by taking Moore-Penrose pseudoinverse $A^+$ of the matrix $A$:

$$x_{i0} = A^+B$$

2. The probabilities are normalized to sum up to 1:

$$x_i = \frac{x_i}{\sum_{j} x_j \exp(-\gamma x_j)}$$

3. One step of the gradient descent algorithm is taken:

$$x_{i+1} = x_i + \frac{1}{\gamma} \left( \nabla \log \frac{p(x, i, j)}{p(x, i, j)} - A^t(B - A)x_i \right)$$

4. Repeat from step 2 until convergence.

Figure 1: the gradient descent algorithm with probability normalization

3. Computational experiments
3.1. Environment

All experiments were conducted using a 10x10 size two-dimensional grid world as shown in figure 1.

Figure 2: A two-dimensional “grid world” (size 10x10)

Figure 3: Modeling (a) inertia and collisions - reflexive (b) and absorptive (c) - in the passive dynamics

The agent cannot occupy obstacle positions, nor move through obstacles (i.e. “jump” obstacles). The goal of the agent is to reach the goal position with minimum accumulated cost.

States were not modeled as single positions but as adjacent position pairs, including the current position of the agent and its previous position, in order to enable modeling inertia and collisions.

In this way, in a 10x10 size grid world like the one in figure 2, instead of having a total of $N = 100$ states ($N = 86$ when not considering the obstacle positions) the model will have $N = 784$ states ($N = 575$ when not considering impossible states, like states which have an obstacle position in one or both of its positions, or states which have the goal as its previous position and a different position than the goal as the current position).

Inertia and collisions were modeled by assigning a highest probability $hp$ value to the most probable future state driven by inertia (or by a collision) given the current state, and equally sharing the remaining $(1 - hp)$ probability among the remaining adjacent states, as illustrated in figure 3.

3.2. Results

In each experiment, the agent was placed at each and every possible state (one at a time), and for each state arbitrary policies were generated in equal quantity to the number of adjacent states. The total costs for each of the policies in each of the states were measured, the corresponding systems of linear equations were created and solved using the gradient descent algorithm with probability normalization (as shown in figure 1) with the parameter $\gamma = 10$. All the passive dynamics transition distributions and state costs were successfully calculated, showing results equal to the original values used in the simulated environment.

4. Conclusion and future work

A consistent method for calculating the passive dynamics distribution from measured action costs and/or total costs has been proposed. In essence, it is necessary to know all possible states and, for each state, a well-defined number of different arbitrary policies and to measure the action costs and/or total costs when leaving the state following these policies. In future works, it is important to consider the time/computational cost involved in the passive dynamics and state costs calculation, and verify if this is still faster than traditional reinforcement learning methods.

References


