ECE 598: Interplay between Control and Machine Learning **Fall 2020**

Lecture 6

Markov Chains and MDPs

Lecturer: Bin Hu, Date:09/22/2020

In this lecture, we briefly review Markov chains and Markov decision processes (MDPs).

6.1 Markov Chains

A discrete-time stochastic process is a collection of random variables $\{X_t\}_{t=0}^{\infty}$. Suppose $X_t \in \mathcal{X}$ for all t. Then the set $\mathcal{X}$ is called the state space.

Definition 1 (Markov Chain). *A discrete-time stochastic process $\{x_t\}_{t=0}^{\infty}$ sampled from a countable state space $\mathcal{X}$ is a Markov Chain if*

$$P(x_{t+1} = j | x_t = i, x_{t-1} = i_{t-1}, \dots, x_0 = i_0) = P(x_{t+1} = j | x_t = i), \ \forall k.$$

For a Markov chain, once the value of the current state is known, the distribution of the next state becomes independent of the past information.

A Markov Chain is time-homogeneous if the transition probabilities $P(x_{t+1} = j | x_t = i)$ do not depend on t. Then we can denote such transition probability as P_{ij}. The transition matrix P is defined to be a matrix whose (i, j)-th entry is P_{ij}. Obviously, we have $P_{ij} \geq 0$ for all (i, j). The matrix P is right stochastic since the entries in each row of P sum to one.

Let the distribution of x_t be a row vector whose i-th entry is equal to $p_t(i) = P(x_t = i)$. Then we have $p_t = p_{t-1}P = p_0 P^t$.

Two useful facts are now reviewed below:

- A finite state irreducible Markov Chain has a unique stationary distribution π, which satisfies $\pi = \pi P$.
- For an irreducible and aperiodic finite state Markov chain, p_t always converges to π at a geometric rate described by the spectral gap of P.

Markov chains in control. The above finite state space Markov chain model is useful for many tasks in computer science. For control, we typically look at the case where $\mathcal{X} = \mathbb{R}^n$. In this case, we have a continuous state variable x_t. The Markov property means the conditional probability density function for x_{t+1} is completely determined once x_t is observed, i.e.

$$p(x_{t+1} | x_t, x_{t-1}, \dots, x_0) = p(x_{t+1} | x_t).$$

In control engineering, we typically look at the state-space model:

$$x_{t+1} = f(x_t, w_t) \tag{6.1}$$

where $x_t \in \mathbb{R}^n$ is the state and w_t is some random noise. If w_t is IID, then $\{x_t\}$ forms a Markov chain on $\mathbb{R}^n$. Sometimes w_t is also correlated and generated by a time series model itself, i.e. $w_t = g(w_{t-1}, e_t)$ where e_t is IID, then the augmented variable $\{(x_t, w_t)\}$ forms a Markov chain. If we have a model in the form of $x_{t+1} = f(x_t, x_{t-1}, w_t)$, then we need to augment a new state $y_t = \begin{bmatrix} x_t \\ x_{t-1} \end{bmatrix}$ and $\{y_t\}$ forms a Markov chain.

Linear systems as Markov chains. Let's look at more concrete examples. The following linear state-space model has been widely used in control applications:

$$x_{t+1} = Ax_t + Bw_t$$

Here, w_t is an IID Gaussian process and x_t is the state. Then $\{x_t\}$ forms a Markov chain. By induction, one can show x_t is Gaussian for all t. For simplicity, we assume $w_t \sim \mathcal{N}(0, W)$. Suppose x_t is known, then x_{t+1} becomes a Gaussian variable sampled from the distribution $\mathcal{N}(Ax_t, BWB^\mathsf{T})$. Clearly, the distribution of x_t is completely determined once x_t is observed. Hence $\{x_t\}$ is a Markov chain. In addition, since a Gaussian variable is completely determined by its mean and variance, the statistics of $\{x_t\}$ can be actually determined by the following iteration:

$$\begin{aligned} \mu_{t+1} &= A\mu_t \\ Q_{t+1} &= AQ_tA^\mathsf{T} + BWB^\mathsf{T} \end{aligned}$$

where μ_t is the mean value of x_t, and Q_t is the covariance of x_t.

Mixed continuous/discrete state space. Let's look at another example here. Consider the Markovian jump linear system $x_{t+1} = A_{i_t}x_t + B_{i_t}w_t$ where i_t is the switching parameter and w_t is IID. This system is not a time-homogeneous Markov chain under arbitrary switching. However, if $\{i_t\}$ is a Markov chain itself, we can augment $\{(x_t, i_t)\}$ to obtain a Markov chain. In this case, the augmented state (x_t, i_t) involves a mixture of continuous variable (x_t) and discrete variable (i_t).

6.2 Markov Decision Processes (MDPs)

A Markov decision process (MDP) can be viewed as a Markov process with feedback control. Formally, a MDP is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the transition kernel, R is the reward, and γ is the discount factor. Let s_t be the state at step t. At every step, we are allowed to choose an action $a_t \in \mathcal{A}$ to "control" the system. Once the action a_t is applied, the probability distribution for s_{t+1} is completely determined by the transition kernel $P^a_{ss'} := P(s_{t+1} = s'|s_t = s,\, a_t = a)$, and a reward $R(s_t, a_t)$ is received to measure the performance of the control action. The goal is to choose the action sequence $\{a_t\}$ to maximize the total accumulated rewards $V(s_0) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k)\middle|s_0\right]$.

How to solve MDPs? If the transition model P is known, then one can solve the MDP using dynamic programming. When P is unknown, one can solve the MDP by applying reinforcement learning methods. In general, reinforcement learning refers to a collection of data-driven techniques that can be used to solve MDPs when the transition model is unknown. We will talk about reinforcement learning in the next few lectures.

Applications in computer science and control. Many tasks such as game playing and Go can be viewed as MDPs with discrete spaces. More guarantees can be obtained for such setups. In contrast, control tasks are mostly formulated as MDPs with continuous state/action variables. Let's look at two examples here.

Example 1: Linear Quadratic Regulator (LQR) without process noise. We start with a simple setup. Consider the following linear dynamical system

$$x_{t+1} = Ax_t + Bu_t \tag{6.2}$$

where A is the state matrix, B is the input matrix, u_t is the control action, and x_t is the system state. The objective is to choose $\{u_t\}$ to minimize the following cost

$$\mathcal{C} = \mathbb{E}_{x_0 \sim \mathcal{D}} \sum_{t=0}^{\infty} (x_t^\mathsf{T} Q x_t + u_t^\mathsf{T} R u_t) \tag{6.3}$$

where Q and R are positive definite matrices. There is an initial distribution $\mathcal{D}$ where x_0 is sampled from. Since there is no process noise, the only randomness stems from $\mathcal{D}$. The choices of Q and R reflect the conflicting design objectives in control: We want to achieve small tracking error by using small control inputs. Since there is no process noise w_t, we can set the discount factor γ to be 1 and the cost $\mathcal{C}$ is still finite. The above LQR problem can still be viewed as a MDP on a continuous state space. A few key features are summarized as follows.

1. Continuous state space: x_t is a real vector and hence can take any values in $\mathbb{R}^x$.
2. Continuous action space: u_t is also a real vector.
3. Transition dynamics: Given x_t and u_t, then x_{t+1} is also known due to (6.2). The transition dynamics can be viewed a stochastic kernel centering at $(Ax_t + Bu_t)$ with probability 1 .

4. Additive structure of cost function: $\mathcal{C}$ is a sum of cost values at different t. The one-step cost depends on both the state and the input at that step. The cost and the reward are somehow equivalent concepts. Specifically, we can think our reward function as $R(x, u) = -(x^\mathsf{T} Q x + u^\mathsf{T} R u)$.

5. The discount factor γ is set to be 1.

Given an initial condition x_0, we denote the state value function as $V(x_0) = \sum_{t=0}^{\infty}(x_t^\mathsf{T} Q x_t + u_t^\mathsf{T} R u_t)$. Therefore, we have $\mathcal{C} = \mathbb{E}_{x\sim\mathcal{D}} V(x)$.

Example 2: LQR with process noise. In this case, the dynamics become

$$x_{t+1} = A x_t + B u_t + w_t \tag{6.4}$$

where w_t is an IID Gaussian noise. When there is the process noise term w_t, the cost in (6.3) is never finite for $\gamma = 1$ due to the fact that x_t does not converge to 0. Hence we need to set $\gamma < 1$. Now we consider the cost function

$$\mathcal{C} = \mathbb{E} \sum_{t=0}^{\infty} \gamma^t (x_t^\mathsf{T} Q x_t + u_t^\mathsf{T} R u_t). \tag{6.5}$$

Again, this is a MDP problem. A key fact is that the probability distribution of x_{t+1} is completely known if x_t and u_t are seen. This is due to the IID nature of w_t. When w_t is Gaussian, the transition density will also be Gaussian. For simplicity, let's assume $w_t \sim \mathcal{N}(0, W)$. Once (x_t, u_t) is given, then the distribution of x_{t+1} is completely determined as $\mathcal{N}(A x_t + B u_t, W)$. Therefore, the above LQR problem is exactly a MDP problem. Given the information of (A, B, Q, R), we can apply model-based control methods such as Riccati equation or LMIs to solve this MDP. If (A, B, Q, R) is unknown, we can apply reinforcement learning to solve this problem. We will talk more about this in the next few lectures.