Lecture 10

Linear Function Approximation

We've spent a good deal of time understanding how to generate control policies for a large class of robotic systems, assuming a model exists. In learning-based control for robotics, we assume that parametric function approximation is a core element of the solution, where the function approximator is used to approximate the scalar "value" function, the policy, the dynamics model, or some combination of the three. These function approximators can take many different forms, and can be broken down into two different classes: linear function approximators and nonlinear function approximators. Before exploring the function approximator structure, we will first explore approaches for fitting the parameters.

10.1. Maximum Likelihood Estimation

Consider the case of a special coin that lands heads up with an unknown probability:

$$P(Heads) = \theta \quad P(Tails) = 1 - \theta \tag{10.1}$$

Flips are independent and identically distributed (i.i.d.). Assume you get some data sequence $D = \{H, H, T, H, T...\}$

$$p(D|\theta) = \theta^k (1-\theta)^{n-k} \tag{10.2}$$

where you have k heads out of n flips.
Maximum Likelihood Estimation (MLE) maximizes the probability of the observed data according to:

$$\hat{\theta} = \arg\max_{\theta} p(D|\theta) \tag{10.3}$$

Since the logarithm is monotonically increasing, this is equivalent to:

$$\hat{\theta} = \arg\max_{\theta} \log p(D|\theta) \tag{10.4}$$

Take the derivative and set equal to zero:

$$\frac{d}{d\theta} \log p(D|\theta) = 0. \tag{10.5}$$

This results in:

$$\hat{\theta} = \frac{k}{n} \quad \mathbb{E}[\hat{\theta}] = \theta^*. \tag{10.6}$$

Maximum likelihood estimate is unbiased.
Consider the case where you have a Gaussian random variable:

$$p(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(x-\mu)^2}{2\sigma^2}. \tag{10.7}$$

Remember the properties of Gaussian distributions:

$$X \sim \mathcal{N}(\mu, \sigma^2) \tag{10.8}$$

where

$$Y = aX + b \implies Y \sim \mathcal{N}(a\mu + b, a^2\sigma^2). \tag{10.9}$$

A sum of a Gaussians is a Gaussian...

If

$$X \sim \mathcal{N}(\mu_x, \sigma_x^2) \tag{10.10}$$

$$Y \sim \mathcal{N}(\mu_y, \sigma_y^2) \tag{10.11}$$

$$Z = X + Y \implies Z \sim \mathcal{N}(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2) \tag{10.12}$$

Maximum Likelihood Estimation for Gaussian (i.i.d. samples)

$$p(D|\mu, \sigma) = p(x_1 ... x_n|\mu, \sigma) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \prod_{i=1}^{n} \exp \frac{-(x_i - \mu)^2}{2\sigma^2} \tag{10.13}$$

$$\log(p(D|\mu, \sigma)) = -n \log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2\sigma^2} \tag{10.14}$$

Maximum likelihood for mean is

$$\frac{d}{d\mu} \log(p(D|\mu, \sigma)) = \frac{d}{d\mu}\left[- n \log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2\sigma^2}\right] \tag{10.15}$$

$$= \frac{d}{d\mu}\left[\sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2\sigma^2}\right]. \tag{10.16}$$

Setting equal to zero, we have:

$$\sum_{i=1}^{n} \frac{(x_i - \mu)}{\sigma^2} = 0 \tag{10.17}$$

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} \mu \tag{10.18}$$

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{10.19}$$

For the variance we have

$$\frac{d}{d\sigma} \log P(D|\mu, \sigma) = \frac{d}{d\sigma}\left[- n \log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2\sigma^2}\right] \tag{10.20}$$

$$= -\frac{n\sqrt{2\pi}}{\sqrt{2\pi}\sigma} + \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{\sigma^3} \tag{10.21}$$

$$= -n\sigma^2 + \sum_{i=1}^{n} (x_i - \mu)^2 \tag{10.22}$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})^2. \tag{10.23}$$

You will notice that the estimate of the variance is biased, since errors in $\hat{mu}$ will always contribute a "positive" offset to the variance estimate.

10.2. Linear Least Squares

$$y_i = x_i \theta + \epsilon_i \tag{10.24}$$

$$\min_\theta \sum_{i=1}^{n} (y_i - x_i \theta)^2 \tag{10.25}$$

Multiple parameters:

$$y_i = x_i^T \theta + \epsilon_i. \tag{10.26}$$

Multiple measurements give

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \tag{10.27}$$

where

$$Y = X\theta + \epsilon. \tag{10.28}$$

Let's minimize the least-squares error:

$$\begin{aligned} \hat{\theta} &= \arg\min_\theta \sum_{i=1}^{n} (y_i - x_i^T \theta) && (10.29) \\ &= \arg\min_\theta ||Y - X\theta||_2^2 && (10.30) \\ &= \arg\min \left[(Y - X\theta)^T (Y - X\theta) \right] && (10.31) \\ &= (X^T X)^{-1} X^T Y. && (10.32) \end{aligned}$$

10.3. Linear Least Squares and MLE

Why is $\sum_{i=1}^{n} \epsilon_i^2$ a good objective to minimize?

$$\begin{aligned} \hat{\theta} &= \arg\max_\theta \log p(D|\theta, \sigma) && (10.33) \\ &= \arg\max_\theta -n \log \left(\sqrt{2\pi}\sigma \right) - \sum_i^n \frac{(y_i - x_i^T \theta)^2}{2\sigma^2} && (10.34) \\ &= \arg\min_\theta \sum_i^n (y_i - x_i^T \theta)^2 && (10.35) \end{aligned}$$

Assume

$$y_i = x_i^T \theta + \epsilon_i \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2) \tag{10.36}$$

and

$$Y = X\theta + \epsilon \tag{10.37}$$

so that

$$\hat{\theta} = (X^T X)^{-1} X^T Y. \tag{10.38}$$

Then

$$\hat{\theta} = (X^T X)^{-1} X^T (X\theta + \epsilon) \tag{10.39}$$

$$\begin{aligned} Cov(\hat{\theta}) &= \mathbb{E}\Big[(\hat{\theta} - \mathbb{E}(\hat{\theta}))(\hat{\theta} - \mathbb{E}(\hat{\theta}))^T\Big] && (10.40) \\ &= (X^T X)^{-1}\sigma^2 && (10.41) \end{aligned}$$

10.4. Linear Function Approximators and Robot Dynamics

Linear function approximation may seem a restrictive representation, but it is actually quite a powerful approach. Consider the nonlinear dynamics of the form

$$\dot{x} = f(x, u) + F(x, u)\theta \tag{10.42}$$

If we can measure $\dot{x}$, then we have

$$\dot{x} - f(x, u) = F(x, u)\theta \tag{10.43}$$

which we can rewrite as

$$y_i = x_i^T \theta \tag{10.44}$$

y_i is known as "equation error".

10.5. Nonlinear Basis Functions

Using nonlinear basis functions also provide a powerful means of representing arbitrary functions. Consider a function with the following structure

$$y = \Phi(x)^T \theta \tag{10.45}$$

Here $\Phi(x)$ could be any vector of nonlinear functions. For our cost-to-go, we could write

$$\hat{J}^*(x) = \Phi(x)^T \theta. \tag{10.46}$$

For our policy, we could write

$$\hat{\pi}^*(x) = \Phi(x)^T \theta. \tag{10.47}$$

Or for our dynamics model, we could write

$$\dot{x} = f(x, u) + \Phi(x)^T \theta. \tag{10.48}$$

These nonlinear basis functions can take many different forms. There are some classic forms. One such form is referred to as Radial Basis Functions, which are given as:

$$\phi_i(x) = \exp \frac{-||x - \mu_i||^2}{2\sigma^2} \tag{10.49}$$

where $i = 1..N$ and $\Phi(x)^T = \begin{bmatrix}\phi_1(x) & \phi_2(x) & ...\phi_N(x)\end{bmatrix}$. These are also called "kernels".
Other common basis functions are trigonometric basis functions such as

$$\Phi(x)^T = \begin{bmatrix}\sin(x) & \cos(x) & \sin(x)\cos(x)\ldots\end{bmatrix} \tag{10.50}$$

or polynomial basis functions such as

$$\Phi(x)^T = \begin{bmatrix}x & x^2 & x^3\ldots\end{bmatrix}. \tag{10.51}$$

Bibliography

[1] Russ Tedrake. *Underactuated Robotics.* 2023.

[2] Dimitri Bertsekas. *Reinforcement Learning and Optimal Control.* Athena Scientific, 2019.