Reinforcement Learning and Optimal Control

IFT6760C, Fall 2021

Pierre-Luc Bacon

October 25, 2021

Stochastic Optimization

General problem of the form:

$$\text{minimize} \quad J(\theta) = \mathbb{E}_{X \sim P_\theta}\left[h(X, \theta)\right] \quad .$$

This is an unconstrained optimization problem, which we want to solve by gradient descent.

The dependency on θ can be:

1. Distributional if $X \sim P_\theta$
2. Structural if θ appears inside the expectation
3. Structural and distributional (as in the above)

Structural Dependency

Let $X \sim P$ and the distribution of X doesn't depend on θ, we have:

$$DJ(\theta) = \mathbb{E}_{X \sim P}\left[D_2 h(X, \theta)\right] \quad ,$$

where D_2 denotes the partial derivative of h with respect to the argument θ.

We then get a derivative estimator, call it $\hat{D}J(\theta)$ by taking a sample average over N samples $\{x_1, \ldots, x_N\}$:

$$\hat{D}J(\theta) \triangleq \frac{1}{N} \sum_{i=1}^{N} D_2 h(x_i, \theta) \quad .$$

The fact that we can push the derivative inside the expectation relies on the fact that we can interchange the order of integration and differentiation. We rely on Leibniz integral rule for that.

Generally speaking, you want to make sure that the **derivative exists at all points** and that all the values are **bounded** inside the expectation.

Empirical Risk Minimization

Let θ denote the parameters of a predictive model (hypothesis) f, we define the *risk* R as:

$$R(\theta) \triangleq \mathbb{E}_{X,Y\sim P}\left[L(f(X;\theta),Y)\right] \quad .$$

Vapnik's Empirical Risk Minimization Principle (ERM) says that instead we ought to work with the *empirical risk*:

$$\hat{R}(\theta) \triangleq \frac{1}{N}\sum_{i=1}^{N} L(f(x_i;\theta),y_i) \quad .$$

where $\{(x_1,y_1),\ldots,(x_N,y_N)\}$ is a dataset of input-output pairs drawn from the joint over X and Y.

- Changing θ won't affect the distribution of examples in the world: the dependency is structural only

ERM and Stochastic Counterpart

$\hat{R}(\theta)$ is an *empirical* loss, and because the depency is structural:

$$D\hat{R}(\theta) = \hat{D}R(\theta) \quad .$$

The notation above is subtle, but highlights an important point: the hat symbol is first over R, then over D. On the left-hand-side, we are taking the exact derivative of an approximate loss; the right-hand-side, the approximate derivative of an exact loss.

The ERM can be viewed as an instance of Rubinstein's (1968) *Stochastic Counterpart*.

Sample Average Approximation (SAA)

In the SAA (or stochastic counterpart), the idea is to transform a stochastic optimization problem into a deterministic one via the Monte Carlo method: ie. to approximate the objective by a sample average estimate.

- This is in contrast to Stochastic Approximation, in which we spend very little time approximating the objective

SAA and SA

Let's highlight the dependency on the number of samples used in the two kinds of estimates:

1. Derivative estimation: $\hat{D}^{(N)}J(\theta) \triangleq \frac{1}{N}\sum_{i=1}^{N} D_2 h(x_i, \theta)$
2. Objective estimation: $D\hat{J}^{(N)}(\theta) = \frac{1}{N}\sum_{i=1}^{N} D_2 h(x_i, \theta)$.

The SAA approach for unconstrained optimization by gradient descent is of the form:

$$\theta^{(k+1)} = \theta^{(k)} - \eta D\hat{J}^{(N)}(\theta^{(k)}) \quad ,$$

whereas the SA counterpart is:

$$\theta^{(k+1)} = \theta^{(k)} - \eta D\hat{J}^{(1)}(\theta^{(k)}) = \theta^{(k)} - \eta \hat{D}^{(1)}J(\theta^{(k)}) \quad .$$

Distributional Dependency

$$J(\theta) = \mathbb{E}_{X \sim P_\theta} \left[h(X, \theta) \right]$$

Assuming that we can change the order of differentiation and integration:

$$\begin{aligned} DJ(\theta) &= \int \left(D_2 h(x, \theta) P(x; \theta) + h(x, \theta) D_2 P(x; \theta) \right) dx \\ &= \mathbb{E}_{X \sim P_\theta} \left[D_2 h(X, \theta) \right] + \int h(x, \theta) D_2 P(x; \theta) dx \quad . \end{aligned}$$

There is nothing wrong with the above expression. But we are now facing a computational challenge. How to deal with the second term?

Change of measure

Let $\rho(x, \theta, q) \triangleq p(x; \theta)/q(x)$, then:

$$J(\theta) = \mathbb{E}_{X \sim p_\theta}\left[h(X, \theta)\right] = \mathbb{E}_{X \sim q}\left[h(X, \theta)\rho(X, \theta, q)\right] \quad .$$

The derivative of J becomes:

$$DJ(\theta) = \mathbb{E}_{X \sim q}\left[D_2 h(X, \theta)\rho(X, \theta, q) + h(X, \theta)D_2\rho(X, \theta, q)\right] \quad .$$

Likelihood ratio (LR) estimator

The LR estimator is given by:

$$\hat{D}^{\mathrm{LR}} J(\theta) \triangleq \frac{1}{N} \sum_{i=1}^{N} D_2 h(x_i, \theta) \rho(x_i, \theta, q) + h(x_i, \theta) D_2 \rho(x_i, \theta, q) \quad .$$

Note that the same dataset $\{x_1, \ldots, x_N\}$ can be used throughout optimization. That is:

$$\theta^{(k+1)} = \theta^{(k)} - \eta \hat{D}^{\mathrm{LR}} J(\theta^{(k)}) \quad .$$

and the only thing that changes is where we evaluate our derivative.

LR as SAA

From the SAA perspective, LR amounts to using the following "surrogate" objective:

$$\hat{J}^{\mathrm{LR}}(\theta) \triangleq \frac{1}{N} \sum_{i=1}^{N} h(x_i, \theta) \rho(x_i, \theta, q)$$

and form the sequence:

$$\theta^{(k+1)} = \theta^{(k)} - \eta D \hat{J}^{\mathrm{LR}}(\theta^{(k)}) \quad .$$

Once again, the dataset $\{x_1, \ldots, x_N\}$ gets "bound" inside the function $\hat{J}^{\mathrm{LR}}$, and we can evaluate the surrogate objective everywhere.

Score Function Estimator

The score function is obtained for the choice $q = p_\theta$. We then get:

$$DJ(\theta) = \mathbb{E}_{X \sim p_\theta} \left[D_2 h(X, \theta) \rho(X, \theta, p_\theta) + h(X, \theta) D_2 \rho(X, \theta, p_\theta) \right] \quad .$$