Why is there a minus in the Lagrangian but a plus in the Hamiltonian?

September 29, 2017

In the Lagrangian formalism, the action $S[q] = \int L(q, \dot{q})dt$ with the Lagrangian $L(q, \dot{q}) = K(\dot{q}) - \Pi(q)$ is considered to be a functional of the trajectory $q(t)$, whereas in the Hamiltonian formalism, the action $A[q, p] = \int H(q, p)dt$ with the Hamiltonian $H(q, p) = K(p) + \Pi(q)$ is a functional of the primal variables $q(t)$ and dual variables $p(t)$. When we want to control a system, we usually come up with a cost function—which is a kind of Hamiltonian for the system comprised of the plant and the controller—and then we optimize it with respect to controls. The question is: Why do we always use plus in the definition of the cost? The answer is that both L and H are just convenience functions for deriving the equations of motion. Once the kinetic and potential energies are chosen, one can derive the resulting equations of motion. The cost function in a control problem, on the other hand, is something different, since it tries to impose a certain trajectory by changing or modulating the system dynamics. For example, we can set the length of a pendulum and observe how fast it oscillates. Then we may change the length and observe it again. In both cases, the Hamiltonian gives us the equations of motion specific to the choice of the length. However, it tells us nothing as to what length we should prefer. This knowledge should come from outside, i.e., from a cost function. Indeed, you might say that you want to have a pendulum that swings once per minute. Then you get an optimization problem over the control parameter l that modulates the dynamics of the system. The Hamiltonian, however, is indifferent to the cost function since it treats l as a constant parameter and returns the equations of motion for whatever l you specify. Thus, although superficially related concepts of the system energy and cost are actually two different things.

1 Lagrangian and Hamiltonian formalisms

The Hamiltonian principle of least action says that a physical system follows a stationary trajectory $q(t)$ of the action

$$S[q] = \int L(q, \dot{q})dt \tag{1.1}$$

with the Lagrangian $L(q, \dot{q}) = K(\dot{q}) - \Pi(q)$. For simplicity the system is assumed time invariant. The equations of motion follow from the necessary condition for optimality

$$\frac{d}{dt} L_{\dot{q}} = L_q.$$

Note that action S in (1.1) is a function of $q(t)$ only. Although $\dot{q}$ appears in the right-hand side, it is not an independent variable. For example, for the harmonic oscillator we get

$$L(q) = \frac{\dot{q}^2}{2} - \frac{q^2}{2} \quad \Rightarrow \quad \ddot{q} = -q.$$

If, however, we reduce the second order system of differential equations to the first order system by the standard trick of treating $L_{\dot{q}}$ as an independent variable p, we arrive at the Hamiltonian

$$H(q, p) = \frac{p^2}{2} + \frac{q^2}{2}.$$

The Hamiltonian allows one to derive the equations of motion by equating its differential to zero,

$$dH = H_p dp + H_q dq = 0. \tag{1.2}$$

From here we obtain the famous equations

$$\begin{aligned} \dot{q} &= H_p, \\ \dot{p} &= -H_q, \end{aligned}$$

which can also be stated in vector form as

$$\dot{\begin{bmatrix} q \\ p \end{bmatrix}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} H_q \\ H_p \end{bmatrix}.$$

One can also cook up an action-like integral with the Hamiltonian inside

$$A[q, p] = \int H(q, p) dt,$$

but since we are treating q and p as independent variables equating its variation to zero is equivalent to (1.2). In any case, for the harmonic oscillator one obtains the following equations

$$\dot{\begin{bmatrix} q \\ p \end{bmatrix}} = \begin{bmatrix} p \\ -q \end{bmatrix}.$$

What is important here is that L is a function of only q, really; whereas H is a function of (q, p). The equations of motion are obtained in both cases by variating the corresponding action functional.

2 Relation to optimal control

Thinking about the pendulum, we can imagine that nature actively decides how to move it. In the equation of motion

$$\ddot{q} = -q, \tag{2.1}$$

we can imagine that the right-hand side is a forcing function that nature applies to the point mass to move it. How does nature decide? Perhaps, first we should determine what "decide" means in this context. Nature decides what terms to put into the Hamiltonian because this in turn determines the evolution of the system. Indeed, consider the hamiltonian

$$H(q,p) = \frac{p^2}{2} + \int_0^q u(x)dx. \tag{2.2}$$

The equation of motion derived from it is the forced double integrator

$$\ddot{q} = -u(q).$$

So, for the pendulum, nature choses $u(q) = q$. However, in general (2.2) describes any dynamical system consisting of a particle acted upon by a conservative force.

Ok, we figure out that "decision" in this context refers to the choice of the Hamiltonian (or to the choice of the forcing function in the right-hand side of (2.1)). How does nature decide what forcing function to apply? This question cannot be answered based on any principle of mechanics because the forcing function is given by the gradient of a potential field (in this case gravitational field). Nevertheless, if we imagine ourselves having control over what the field should be, we can impose any forcing function we like. What would the criteria be? Perhaps, we would base our choice on the resulting trajectory $q(t)$. If we assume that we can only set the system parameters once in the beginning of an experiment and then observe a trajectory $q(t)$, then it makes sense to describe what we want from the trajectory somehow, maybe in terms of an objective function. If we succeed in describing the "goal" of our game in terms of an objective function, we can formalize the search as an optimization problem.

3 General remarks on the RL problem formulation

The standard problem formulation in RL seems to be unsatisfactory in several regards.

1. Agent should be considered part of the world. Conventionally, one thinks of an agent as an immaterial being that controls something in the world. At best, it is an Atari player that pushes some buttons behind the scenes.

It is better to think of the agent and the world as one joint system, dynamics of which the agent can modulate. It is even better to just imagine a single composite dynamical system with some tunable parameter exposed. The game is then to pick the parameters that drive the system to a desired state.

2. Goal $\neq$ cost function. First, reward being external to the agent does not make sense at all. Reward should be computed internally by the agent. Reward (or cost function) by itself is just a proxy that helps in shaping the system dynamics to achieve a goal. More concretely, it seems to be unrelated to both the real world and to engineering disciplines to assume zero reward everywhere and one scalar value at the end of an episode. The agent should have access to the mechanism that computes the reward. Truly, we have no idea of how one picks a goal. However, for a machine, we can safely assume that a goal is given. Then the question is merely how to achieve it. In this view, the agent could shape the reward if it is beneficial for achieving the goal. The cost function in some sense acts as a potential. Thus, we are shaping the system dynamics using potentials.

Taking these points into account, one can identify two main challenges that need to be solved to enable task completion.

1. Develop cost functions consistent with the goal. To be more precise, if we assume that the agent changes system parameters at a low frequency, thus exploiting as much as possible natural dynamics of the world, then it can adjust the cost function between every parameter change. This makes sense if optimization of the cost with respect to parameters is fast. Then the main tunable parameter is the cost function.

2. In the whole discussion above, we silently assumed that system dynamics is known to the agent. However, this is not so. And this is, actually, the only reason why we need real world at all (given that goals are provided; otherwise, real world experiences could trigger goals). We need real world to test hypotheses. All the reward shaping and optimization can be done in simulation. (I discern between simulation, virtual environment, and real world. Simulation refers to using analytic models such as $\dot{x} = f(x; \theta)$. This can be called internal rehearsal or imagination. Virtual environment and real world can be used interchangeably and they refer to the ground truth about the system dynamics. If the agent is supposed to live in a virtual environment such as Facebook, for example, then it is its real world.)

To conclude, one needs to have an algorithm that takes a goal as input. Based on the goal, a sequence of cost functions should be generated. If system dynamics is not known a priori, it must be estimated from experience in the real world.

4 Bandit problem as dynamics estimation

Bandit problem is usually formulated as if there were no dynamics involved. However, the cumulative reward is nothing else but the state of the system. As well as the number of times each arm was played and what payoff was observed. All these data are part of the system state. Decision as to what arm to play can be considered as state feedback. The goal may be to maximize the cumulative return. As before, goal is not equal to reward function. Reward function may include exploration bonus and such kind of things, but the goal is still return maximization. Why do bandits do exploration? To estimate the dynamics. The dynamics tells the agent what reward to expect from each arm.

Maybe a sequence of arms gives a better return? More concretely, arms may be non-commutative. For example, showing some ads may not work on its own, but then showing another ad afterwards may bring better result. Imagine seeing an ad of a subscription to FT for $100 every day for a week, and then suddenly it is just $10. Of course, you are more inclined to buy. This is the dynamics behind reward generation. If the bandit agent only observes the reward, it is hard for her to tell what led to such outcome. However, it can figure out the dynamics that leads to good returns from playing arms.