COS 435/ECE 433 Week 1 Precept Notes

January 31, 2025

1 Probability Review

In this precept, we are going to go over key probability and random variable objects and definitions. We'll finish off with problems on (1) calculating and interpreting entropy, (2) expected value, and (3) decomposing probabilities.

1.1 Where does randomness show up in RL?

Reinforcement Learning (RL) studies decision-making over time. Consider an RL problem such as driving from home to a supermarket. To ground ourselves,

- What are the states/observations of the system?
- What are the actions of the system?
- What are possible rewards of the system?

Given the **states, actions, and rewards** of the system, **what are possible sources of randomness in the autonomous driving problem?** (Discuss.)

As you have found, there are many possible sources of randomness, from stochasticity in transitions from state to state, to stochasticity in rewards, to stochasticity in learning dynamics. Thus, we must be comfortable with the language of probability and random variables before diving into the RL problem.

1.2 Expected Value

A random variable X randomly takes values with certain probabilities. The **expected value** of a random variable $X \sim p(x)$ is defined as:

$$\mathbb{E}[X] \triangleq \int p(x)x\,dx$$

We will also work with **conditional expectation values**:

$$\mathbb{E}[X \mid Y] \triangleq \int x\,p(x \mid y)\,dx.$$

In words, this is the "average" value we expect random variable X to take on *given* a different random variable Y.

1.3 Entropy

Definition 1.1 (Entropy)**.** The entropy S of a random variable X is defined as

$$S(X) \triangleq -\int p(x) \log p(x)\, dx \tag{1}$$

$$= -\mathbb{E}_{p(x)}[\log p(x)]. \tag{2}$$

In this class, we assume that log is base 2, and thus quantify $S(X)$ in terms of **bits**. We can interpret $S(X)$ as the number of (Yes/No) questions (binary questions!) one has to ask in *expectation* to figure out the value of X. This is equivalent to the expected number of bits needed to encode X, which gives a clue that $S(X)$ describes both uncertainty and information. ◇

Entropy is a measure of (1) uncertainty of a random variable's value or (2) information contained in a random variable. While these definitions may sound contradictory, let's consider an example that bridges these two concepts.

For example, consider random variable B_{bag} which is the number of books in my backpack. There is not much uncertainty in this random variable – however, there isn't much *information* contained in the random variable, either! The random variable is, most likely, the fairly obvious 0: you don't need much information to make a fairly good guess at the value of B.

However, consider random variable $B_{\text{Firestone}}$, which is the number of books in Firestone. There is a lot of uncertainty in this random variable and, correspondingly, we need a lot of information to figure out the true value of $B_{\text{Firestone}}$. Thus, these concepts of *information* and *uncertainty* are related, not contradictory.

In this class, we assume that log is base 2, and thus quantify $S(X)$ in terms of **bits**. We can interpret $S(X)$ as the number of (Yes/No) questions (binary questions!) one has to ask in *expectation* to figure out the value of X. This is equivalent to the expected number of bits needed to encode X, which gives a clue that $S(X)$ describes both uncertainty and information.

1.4 Cross Entropy

We can also define cross-entropy, which you may have seen before in classification tasks:

Definition 1.2 (Cross Entropy)**.** The entropy cross entropy $H(p, q)$ is defined as

$$H(p, q) \triangleq -\int p(x) \log q(x)\, dx. \tag{3}$$

The interpretation for cross entropy is a bit fuzzier than entropy: it is the expected number of bits needed to communicate X when assuming an *incorrect* distribution $q(x)$ on a true probability distribution $p(x)$. This quantity is closely related to the Kullback-Liebler divergence, which is a measure of "distance" between two probability distributions. ◇

1.5 Preview of Week 2

With these probability tools, we introduce the formalized RL problem.

Definition 1.3 (Markov Decision Process, Finite-Horizon Setting)**.** A *finite-horizon MDP* is a tuple $(\mathcal{S}, \mathcal{A}, p, r, T)$, where

- $\mathcal{S}$ is the set of *states*,

- $\mathcal{A}$ is the set of *actions*,
- p is the *transition dynamics*, such that $p(s'|s,a)$ is the distribution over the next state s', given the agent took action a from state s.
- r is the *reward function*, such that $r(s,a)$ is the reward from taking action a from state s.
- T is the *rollout horizon*, i.e. the number of actions an agent (i.e. a robot) takes in every rollout. $\diamond$

(Ask what is the optimized objective.) The objective that is optimized is
Let's go through the components of an MDP step-by-step, beginning with the word "Markov".

1.5.1 A note on the Markovian Assumption

We assume that dynamics are Markovian:

Definition 1.4 (Markovian dynamics)**.** A dynamical system is Markovian if the next-time-step statistics of our state ($p(s_{t+1})$) is entirely a function of our current state and action ((s_t, a)).

That is,

$$p(s_{t+1} \mid (s_0, a_0), (s_1, a_1), ..., (s_t, a_t)) = p(s_{t+1} \mid s_t, a_t).$$

$\diamond$

Let's make this definition concrete. Say that we are at some state s_0 (i.e. "Go" in Monopoly). We take some action a (i.e. rolling a die). Then, our distribution over the next tile we reach is **a function of the current state and our action**: our dynamics, when our states are tiles on the board, are Markovian. The multi-step history of our previous actions and states does not provide any *additional* information on our future state.

But wait... aren't there a lot of decision problems where the full history of my actions determines my future state? Note that, in these cases, the specific parametrization of the **state** is important, and one may need to add dimensions to the state to capture the salient effects of "historical" actions. For example, let's continue considering Monopoly. Let's say that your state is now your and your opponents' locations on the board *and* money. **Is this system Markovian?** (Discuss.) Answer: No! You losing money is also a function of your opponents' *historical* states, as their previously visited states likely dictate which tiles they have/do not have houses and hotels.

If you parameterize the state as:

- positions of all players,
- amount of money of all players,
- *and* the cards, titles, houses, and hotels of all players

then note that the system is now, again, Markovian: the current state entirely describes the next-timestep statistics. **Can we turn any non-Markovian system into a Markovian system?** Answer: Yes, trivially, by parameterizing the entire history of states and one's current state. However, this is not desireable from a state-complexity standpoint – by definition, our state space is going to scale $|S| \propto \exp(T)$, which is bad.

1.5.2 Stochasticity of transition dynamics and rewards

By definition, the transition dynamics are described by a probability distribution, which can be a source of stochasticity/noise in the system.

Similarly, the reward function or signal can *also* be stochastic: we can let reward be a random variable R, where $\mathbb{E}[R(s,a)] = r(s,a)$, and $r(s,a)$ is some deterministic function of (s,a). We will consider this source of stochasticity in the multi-arm bandit setting, which will be introduced in lecture next week.

1.5.3 Horizon T

Finally, the horizon tells us the timescale of our overall problem/task. So far, we have used notation like s_t (state at time t) fairly loosely, but we generally consider our eventual RL optimization problem over some **finite horizon**: how well can we complete task X/achieve reward Y from time $t = 0$ to time $t = H$?

1.5.4 The objective

We can write our final objective to optimize as

$$\mathcal{F}(\pi) = \mathbb{E}_{s_1 \sim p_0(s_1), a_t \sim \pi(a_t|s_t), s_{t+1} \sim p(s_{t+1}|s_t,a_t)} \left[\sum_{t=1}^{H} \gamma^t r(s_t, a_t) \right] \tag{4}$$

$$= \mathbb{E}_{a_t \sim \pi(a_t|s_t), s_{t+1} \sim p(s_{t+1}|s_t,a_t)} \left[\sum_{t=0}^{H} \gamma^t r(s_t, a_t) \mid s_0. \right] \tag{5}$$

. (Explain the components of this objective.)

2 Discussion Problem

1. (Practice with expectation values) Joe picks a random two digit integer. What is the expected value of the sum of the digits of his number? **Answer:** We use **linearity of expectation**. The expected value of the first digit d_1 is $\mathbb{E}_{\text{two digit numbers}}[10d_1] = 50$. The expected value of the second digit d_2 is $\mathbb{E}_{\text{two digit numbers}}[d_2] = 4.5$. By linearity of expectation, $\mathbb{E}[d] = \mathbb{E}[d_1 + d_2] = 54.5$.

2. (Optimization) Consider cross entropy objective $\mathcal{L}(q) = H(p,q) \triangleq -\int p(x) \log q(x)\, dx$. Here, distribution $p(x)$ is fixed and we are optimizing over $q(x)$. What is the optimal $q(x)$ that minimizes $\mathcal{L}(p)$? **Answer: We take the functional derivative with respect to some** *function* $p(x')$ **(x' to not overload notation) and use Lagrange multipliers. Evaluating, this is** $\frac{\delta L(q)}{\delta q} = \frac{\delta[H(p,q)+\lambda(\int q(x)\,dx - 1)]}{\delta q} = -\int p(x) \frac{1}{q(x)} \delta(x - x') + \lambda \int \delta(x - x')\, dx) = 0 \rightarrow \frac{p(x)}{q(x)} = \lambda$**.** For λ, $\frac{\partial L(q)}{\partial \lambda} = \frac{\partial[H(p,q)+\lambda(\int q(x)\,dx - 1)]}{\partial \lambda} = \int q(x)\, dx - 1 = 0 \rightarrow \int \frac{p(x)}{\lambda}\, dx = 1 \rightarrow \lambda = 1$. Thus, $q^*(x) = p(x)$.