Markov Decision Processes

Csaba Szepesvári

Monday 6th February, 2023

1 Warming up

MDPs stand for "Markov Decision Processes." Markov's name is there because he invented what is now known as Markov chains in an argument to convince his fellow mathematician, one of the many Bernoullis, that the "strong law of large numbers" (look it up!) holds beyond i.i.d. (independent, identically distributed) random sequences (like what we get from coin flips).

The word "Decision" is there in the phrase because we want to talk about how to act, which is about decisions. Contentious question: Do we even make decisions, ever? Do we need to discuss free-will to understand what's going on? Let's skip this for now and just go with that; in any case it may be useful to think that we (or agents) can choose between actions. Is it not beautiful that in math, we can just posit that there is free-will? As opposed to the boring world where everything is predetermined. Why do this? Because this allows us to do good things, like designing a robot to do my house chores. And we are predetermined to think this way :).

Processes come from probability theory. An infinite sequence of random variables is called a *stochastic process*. So many names. Our stochastic process is like this:

$$S_0, A_0,\ R_1, S_1, A_1,\ R_2, S_2, A_2,\ \dots . \tag{1}$$

Here, every letter is upper case: We use upper case to denote random variables. What is a random variable? It is a map whose domain is a *sample space*. We denote the sample space, usually, by Ω. You may have noticed that people who write text that uses random variables and probability almost never mention the sample space. As mentioned in Lecture 1, this is the big (open) secret of probability theory.[1] By the way, at this level, probability theory is not so much a theory in the sense of how we would understand a scientific theory. Rather, it is a mathematical formalism that helps us to talk about a random phenomenon. But this was supposedly covered earlier in other courses. Anyhow, if you want to brush up your knowledge of (elementary) probability theory (how to setup a probabilistic model of some problem), consult the first three chapters of Evans and Rosenthal [2010].

So, if Ω is the sample space, $S_0, A_0, R_1, S_1, A_1, \dots$ are all maps from Ω to their respective ranges. So what are the ranges of these random variables? The range of random variables $S_0, S_1, S_2, \dots$ is $\mathcal{S}$, the *set of states*, which we also call the *state space*. The range of the random variables $A_0, A_1, A_2, \dots$ is $\mathcal{A}$, the *set of actions*, which we also call the *action space*. And the range of $R_1, R_2, \dots$ is $\mathcal{R} \subset \mathbb{R}$, where $\mathcal{R}$ is the set of rewards and $\mathbb{R}$ stands for the set of reals.

For simplicity we will assume that $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{R}$ are *finite* (they are all nonempty!). Note that many of the statements concerning reinforcement learning extend beyond this finite case. The finite case is used only to keep the presentation of the concepts, in these initial lectures, as clean as possible from a mathematical standpoint. In particular, people who think that MDPs are only defined when any of these sets (or all of them) are finite, and that the results about reinforcement learning hold only under this condition, are just very wrong. If someone is equipped with the right mathematical tools, many of the results extend to the more general case (but not all!).

What finiteness buys us is a simpler way of talking about *probabilities*. Where do probabilities come from? Together with the sample space Ω, we are also given a *probability measure* $\mathbb{P}$, which is, by definition, a function that assigns a number in $[0, 1]$ to the (suitable) subsets of Ω.[2] Thus, if A is such a (suitable) subset of Ω, then $\mathbb{P}(A)$ is a number in $[0, 1]$: The probability that event A happens. The mental model one should have is that Ω is the source of all randomness in the world. Think that the randomness is separated out from an experiment first and then put into Ω. For example, if you had a program that calls random number generators to calculate some value, imagine rewriting the code so that all the calls to the random number generator are made upfront and the results are recorded somewhere in the memory. Then, whenever the original code wants to get a new random number, this table of pre-recorded random numbers is consulted and the next number which has not yet been used is fed to the program. The set Ω is then the result of all possible calls to the random number generator and any $\omega \in \Omega$ is a possible outcome for these calls. Note that the rest of the calculation in the above program is deterministic. This "rest" is what is modeled by a random variable, which is a deterministic function from Ω to whatever the range of that variable is. We will actually follow this recipe later in this text.[3]

In summary, so far we have Ω, a probability measure on Ω that we call $\mathbb{P}$, and the random variables

$$\begin{aligned} S_0, S_1, S_2, \ldots &: \Omega \to \mathcal{S}\,, \\ A_0, A_1, A_2, \ldots &: \Omega \to \mathcal{A}\,, \\ R_1, R_2, R_3, \ldots &: \Omega \to \mathcal{R}\,, \end{aligned}$$

where $\mathcal{S}$ is the finite, nonempty set of states; $\mathcal{A}$ is the finite, nonempty set of actions; and $\mathcal{R}$ is the finite, nonempty set of rewards (and in particular, $\mathcal{R}$ is a set of real numbers).

2 The Markov Property

Let $t \geq 0$ denote the timestep. The random variable S_t represents what we call as the *state* at time t (to be more precise, we hope that it will be justified to call S_t a state; more on this later). The random variable A_t represents what we call as the *action* taken at time t. The random variable R_{t+1} represents what we call as the *reward* incurred after the action A_t is taken at time t.

Definition 2.1 (Markov property)**.** We say that $(S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \ldots)$, the sequence in Equation (1), has the *Markov property* if for any $t \geq 0$, $s_0, s_1, \ldots, s_{t+1} \in \mathcal{S}$, $a_0, a_1, \ldots, a_t \in \mathcal{A}$, and $r_1, r_1, \ldots, r_{t+1} \in \mathcal{R}$, it holds that

$$\mathbb{P}(R_{t+1} = r_{t+1}, S_{t+1} = s_{t+1} \,|\, S_0 = s_0, A_0 = a_0, R_1 = r_1, \ldots, R_t = r_t, S_t = s_t, A_t = a_t) \qquad (2)$$

only depends on the values of s_t, a_t, s_{t+1} and r_{t+1}. In particular, none of the other past values matter when calculating probabilities of the form shown in Equation (2).

If the process satisfies this property we can collect the probabilities that appear in Equation (2) into a 4-dimensional table, or 4-argument function, whose inputs are two states $s, s' \in \mathcal{S}$, an action $a \in \mathcal{A}$, and a reward $r' \in \mathcal{R}$. By convention, we use $p(r', s'|s, a)$ to denote this probability, but here p is really just a function from $\mathcal{R} \times \mathcal{S} \times \mathcal{S} \times \mathcal{A}$ to $[0, 1]$; that is, the mathematically pure way of writing the value that p takes at (r', s', s, a) is $p(r', s', s, a)$. We will use the non-pure notation as it is more expressive of the meaning of p, which is "the probability that we see a pair of next values for the reward and the next state given the current state and an action taken." We call P the *transition dynamics function* (and will often drop the word "function").

An equivalent way of stating the Markov property is that there exists a transition dynamics function p such that for any $t \ge 0$, $s_0, s_1, \ldots, s_{t+1} \in \mathcal{S}$, $a_0, a_1, \ldots, a_t \in \mathcal{A}$, and $r_1, r_1, \ldots, r_{t+1} \in \mathcal{R}$, it holds that

$$\begin{aligned}&\mathbb{P}(R_{t+1} = r_{t+1}, S_{t+1} = s_{t+1} \,|\, S_0 = s_0, A_0 = a_0, R_1 = r_1, \ldots, R_t = r_t, S_t = s_t, A_t = a_t)\\ &\qquad = p(r_{t+1}, s_{t+1}|s_t, a_t)\,. \end{aligned} \tag{3}$$

Note that it is somewhat sloppy to say that the sequence $(S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \ldots)$ satisfies the Markov property because these are "just" random variables and the Markov property is about the relationship between these random variables and the probability function $\mathbb{P}$. As explained in the beginning of the course, $\mathbb{P}$ (and Ω), these "global variables", are implicitly part of everything that is said in probability theory. Oh, what horror! Nevertheless, this is how people usually do things, and we follow suit, except with these occasional warnings that point out some potential sources of misunderstandings. Another slight sloppiness is that perhaps we should say that the variables $S_0, S_1, \ldots$ have the Markov property in the process. But then we are hairsplitting.

3 A second take on the Markov property

Why should we care at all about this? Recall that we are in the business of designing algorithms that will do useful things when interconnected with some environment (we shall also call the environment a *system*, short of a "controlled system", borrowing terminology from control theory). And the interface to this environment that the algorithms can use is super simple: The algorithms can send an action to the environment and get some observation and reward as a result. For example, the environment may have an interface which would look like as follows in the Python programming language.

```python
# we may have an internal state at the module level
_internal_state = None

def step(action):
    '''Returns the next observation and the associated
       reward given an action, unless action==None,
       in which case it returns just the current observation.
    '''
    global _internal_state # may use this, update this, etc.
    if action is None:
        # compute the observation, use
        # _internal_state for this purpose
        # observation = ...
        return observation
    # updating _internal_state is necessary
    # compute the observation and reward to be returned
    # (could be a function of _internal_state)
    # observation = ...
    # reward = ...
    return observation, reward
```

There is only one function that is visible from outside of this module: The `step` function. This function takes an action (of whatever type) and returns an "observation" and an associated reward (which is really just a special observation). Of course, this being a `python` module, the module may define whatever global variables it wants and the `step` function may use them. Further, the `step` function may or may not call the random number generator. If it calls the random number generator, the returned values may depend on the random numbers obtained in the call. In this case, `step` itself will return potentially different outcomes every time it is called, even if the contents of all the global variables and the action are the same. On the other hand, if random calls are not used in `step`, the results will be deterministic.

Now, if we want to write some code that "works" no matter how the `step` function is implemented (whatever goal we may have), this will be super challenging. However, if the observations returned are sufficient to *reproduce* the entire computation that `step` performs (up to the randomness), then the situation considerably simplifies!

This is because in this case what will happen in the future only depends on the last observation and the future actions; and is independent of the previous observations, rewards, and actions.

That is, the whole past can be erased: from the perspective of what will happen, how we got to the current observation does not matter. In particular, if the goal is to make future rewards "high", the best way of achieving this will be independent of the past, and will only depend on the last observation.

This is really a special case and in general we cannot hope for this property to hold. But we want to start small and see first what can be done when we are in this advantageous situation. The hope is that what we find will carry to the more complicated cases, perhaps with some extra steps. So we will consider these more complicated setups later.

Now, how does this coding-oriented explanation relate to the math we discussed before? This comes from connecting the random variables with the calls to the `step` function. In particular, we let S_0 record the observation available before we take any action (this can obtained by calling `step(None)`. We let A_0 record the first action taken (which may depend on S_0). And we let R_1, S_1 record the reward and next observation that is returned by the call to `step` with A_0, and so on. Then we have the following proposition:

Proposition 3.1. *If the observation returned from `step` has the reproducibility property in the sense that it, together with the action, is sufficient to reproduce the outcome of the call, up to the randomness of the call,*[4] *then $S_0, A_0, R_1, S_1, \dots$ will have the Markov property as defined in Definition 2.1.*

The attentive reader will notice that something is missing! What is $\mathbb{P}$ and Ω!? As explained earlier, Ω could collect all the outcomes of subsequent calls to the random number generator. For instance, let us consider the case of idealized random number generation: the random number generator returns independent random variables $U_0, U_1, \dots$ uniformly distributed on the $[0, 1]$ interval (this should be sufficient for all purposes). Then Ω and $\mathbb{P}$ are such[5] that for any $n \ge 0$, if $A_0, A_1, \dots, A_n \subset [0, 1]$ is a sequence of closed intervals, and $\ell(A) = |b - a|$ denotes the length of the interval $A = [a, b] \subset \mathbb{R}$, then

$$\mathbb{P}(U_0 \in A_0, \dots, U_n \in A_n) = \mathbb{P}(U_0 \in A_0) \times \dots \times \mathbb{P}(U_n \in A_n) = \ell(A_0) \times \dots \times \ell(A_n)\,.$$

Note that Ω should collect the outcomes of all random calls, *regardless of whether the call is used in function* `step` *or perhaps when calculating an action*.[6]

With this definition, we would now have a chance to formally prove Proposition 3.1 (at the very least, every quantity named in that proposition is well-defined!). However, we will omit this proof (left as an exercise!), but rather note that the truth of the statement *critically* depends on that the *actions were assumed to be calculated based on past information*, that is the action taken at time t should only depend on $S_0, A_0, R_1, S_1, A_1, \dots, R_t, S_t$ (and perhaps some extra randomness).

Note that if we the action was allowed to depend on *privileged* information, say, the global variable `_internal_state`, then the statement would not be true anymore. In what follows, we say that an action-selection mechanism is *causal* if the actions chosen only depend on past information. Any such action-selection mechanism will be called a *(history dependent) policy*.[7]

4 Markov States

That we called $S_0, S_1, \dots$ states from the beginning is only justified if the sequence $S_0, A_0, R_1, S_1, \dots$ possesses the Markov property. When a variable is called a state, by convention it should have the property that knowing what value it takes should suffice to predict the future. If some causal action-selection method leads to a sequence $S_0, A_0, R_1, S_1, A_1, \dots$ that does not posses the Markov property, the observations $S_0, S_1, \dots$ should not be called states. In fact, this is a frequent mistake we see in the literature and is bad practice as it devalues the concept of "state". Do not do this! Hence, in this writeup as well, it would have been better to start with just talking about S_t as observations (and even perhaps use a different symbol than S_t, which suggests a state).

The next note is that the Markov property, as defined here, is a property of the sequence $S_0, A_0, R_1, S_1, \dots$ and $(\Omega, \mathbb{P})$, but whether this property holds also depends on the (causal) action-selection method employed. It may happen that for some (causal) action-selection method, the property holds, but it does not hold always. This would not be helpful since we want to have freedom in choosing the action-selection method (as long as it is causal). This leads to the following definition:

Definition 4.1 (Observation gives Markov state)**.** If the Markov property continues to hold *regardless of the action-selection mechanism* as long as the action-selection method is causal then we say that the observation returned specifies a *Markov state*.

Note that the Markov property is the property of a sequence and the associated probability measure, but whether an observation gives a Markov state is a property of the system (environment).

5 So what is a Markov Decision Process?

When people talk about a Markov Decision Process (MDP), they mean by it the problem of choosing a causal action-selection method which is then interconnected with a system, where the observation returned has the Markov property so as to satisfy some goal related to the rewards obtained. We will stick to this definition. This goes a little bit against the "pure" interpretation that a process should be a specific stochastic process, because in this definition there is no particular fixed stochastic process. This is because the process depends on what action-selection method one employs. Nevertheless, we will use this language.

Thus, MDPs are not really a mathematical thing; it is rather an interconnected set of concepts of ideas about how actions should be chosen in a sequential setting when observations arrive in some sequence and actions can be selected in response to them.

Yet, to get something specific, we can say that an MDP is specified by (for example) the function p that made an appearance in Equation (3). Indeed, given p, *any* causal action-selection method will give rise to a stochastic process $S_0, A_0, R_1, S_1, A_1, \dots$ that satisfies the Markov property (more precisely, the S_ts satisfy the Markov property here). Alternatively, Equation (3) is satisfied.

In summary, we did not quite say what a Markov Decision Process is, but we defined what specifies a Markov Decision Process (it is p). For now, this should suffice and we will stick to this vague terminology. The point is that we work with the function p.[8] Hence, in what follows, we will use the language "let an MDP be given by p". This just means that p is a four-argument function, mapping a set $\mathcal{R} \times \mathcal{S} \times \mathcal{S} \times \mathcal{A}$ to $[0, 1]$ such that

$$\sum_{r' \in \mathcal{R}, s' \in \mathcal{S}} p(r', s' | s, a) = 1$$

holds for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, where instead of $p(r', s', s, a)$ we wrote $p(r', s' | s, a)$, as agreed earlier.

6 Checking the Markov property

There are some useful statements that usually help when it comes to checking the Markov property. For the next statements, fix $(S_0, A_0, R_1, S_1, A_1, \dots)$, Ω, and $\mathbb{P}$. The propositions are concerned with whether under $\mathbb{P}$, $(S_0, A_0, R_1, S_1, A_1, \dots)$ possesses the Markov property.

The first useful result is as follows:

Proposition 6.1. *If the Markov property holds, then for any $t \geq 0$, and $(s_0, a_0, r_1, s_1, \dots, r_{t+1}, s_{t+1}) \in \mathcal{S} \times \mathcal{A} \times \mathcal{R} \times \dots \times \mathcal{R} \times \mathcal{S}$,*

$$\begin{aligned} &\mathbb{P}(R_{t+1} = r_{t+1}, S_{t+1} = s_{t+1} \,|\, S_0 = s_0, A_0 = a_0, R_1 = r_1, \dots, R_t = r_t, S_t = s_t, A_t = a_t) \\ &\quad = \mathbb{P}(R_{t+1} = r_{t+1}, S_{t+1} = s_{t+1} \,|\, S_t = s_t, A_t = a_t) \,. \end{aligned}$$

Proof. Use Equation (3) and elementary calculations. Details are left as an exercise. Think of the worksheets from the last week. □

Let $H_0 = (S_0)$, $H_1 = (S_0, , A_0, R_1, S_1)$, $\dots$ and, in general, for $t \geq 1$, $H_t = (S_0,\, A_0, R_1, S_1,\, \dots\, A_{t-1}, R_t, S_t)$. Here, H_t is the "history" up to time t. Recall that $\mathcal{H}_0 = \mathcal{S}$ and for $t \geq 1$, $\mathcal{H}_t = \mathcal{H}_{t-1} \times \mathcal{A} \times \mathcal{R} \times \mathcal{S}$. Note that $H_t \in \mathcal{H}_t$ holds for all $t \geq 0$.

A variation of the previous proposition is this:

Proposition 6.2. *If the Markov property holds, then for any $t \geq 0$, $(s_t, a_t, r_{t+1}, s_{t+1}) \in \mathcal{S} \times \mathcal{A} \times \mathcal{R} \times \mathcal{S}$, any function $f : \mathcal{H}_{t-1} \times \mathcal{A} \times \mathcal{R} \to \mathcal{U}$ for some set $\mathcal{U}$, and $u \in \mathcal{U}$,*

$$\begin{aligned}\mathbb{P}(&R_{t+1} = r_{t+1}, S_{t+1} = s_{t+1} \,|\, f(H_{t-1}, A_{t-1}, R_t) = u\,, S_t = s_t, A_t = a_t) \\ &= \mathbb{P}(R_{t+1} = r_{t+1}, S_{t+1} = s_{t+1} \,|\, S_t = s_t, A_t = a_t)\,.\end{aligned}$$

Proof. Essentially the same as the proof of the previous proposition. Details are again left as an exercise. □

The next proposition gives a sufficient condition for the Markov property to *fail*:

Proposition 6.3. *Assume that there exists $t \geq 0$, $(s_t, a_t, r_{t+1}, s_{t+1}) \in \mathcal{S} \times \mathcal{A} \times \mathcal{R} \times \mathcal{S}$, a function $f : \mathcal{H}_{t-1} \times \mathcal{A} \times \mathcal{R} \to \mathcal{U}$ for some set $\mathcal{U}$, and $u, v \in \mathcal{U}$ such that*

$$\begin{aligned}&\mathbb{P}(R_{t+1} = r_{t+1}, S_{t+1} = s_{t+1} | f(H_{t-1}, A_{t-1}, R_t) = u, S_t = s_t, A_t = a_t) \\ \neq\, &\mathbb{P}(R_{t+1} = r_{t+1}, S_{t+1} = s_{t+1} | f(H_{t-1}, A_{t-1}, R_t) = v, S_t = s_t, A_t = a_t)\,.\end{aligned}$$

Then $(S_0, A_0, R_1, S_1, A_1, \dots)$ does not posses the Markov property.

The statement says that it suffices to find only some specific history or histories that differ except for the last value of S_t and A_t such that the probabilities of a particular next reward R_{t+1} and next state S_{t+1} differ for the Markov property to fail.

Proof. To prove this proposition one can use proof by contradiction. Assume that the Markov property holds. Then use Proposition 6.2 and derive a contradiction. The details, again, are left as an exercise. □

Notes

1. In fact, people are not sloppy here. There are powerful theorems in probability theory which assure us that unless our assumptions on probabilities and random variables are contradictory, an appropriate sample space can always be constructed. Kolmogorov's extension theorem is doing this, and there are other even more powerful theorems, like the Ionescu-Tulcea theorem.

2. So which subsets are suitable? Whichever they are, we can collect them into a set $\mathcal{F}$. Thus, $\mathcal{F}$ is a set whose elements are sets. Recalling that 2^Ω denotes the set of all subsets of Ω; we thus have $\mathcal{F} \subset 2^\Omega$. Now, whatever $\mathcal{F}$ is, we demand the following: $\emptyset, \Omega \in \mathcal{F}$; if $A \in \mathcal{F}$ then $\Omega \setminus A \in \mathcal{F}$ also holds; and if $A_1, A_2, \dots \in \mathcal{F}$ then $A_1 \cup A_2 \cup \dots \in \mathcal{F}$. We require these because $\mathbb{P}$ is a map from $\mathcal{F}$ to $[0, 1]$: for any $A \in \mathcal{F}$, $\mathbb{P}(A)$ tells us the probability that the *event* A happens. And the axioms that $\mathcal{F}$ needs to satisfy then tell us that $\mathbb{P}$ is a map from an interesting (rich) subset of Ω to $[0, 1]$. Can we choose $\mathcal{F}$ to be 2^Ω? Yes, if Ω is finite, but the answer is already no if Ω is countably infinite. Chapter 2 of the bandit book tells you more if you are curious why. So how to think of $\mathcal{F}$? $\mathcal{F}$ in general will be rich enough to contain pretty much all the sets whose definitions we will ever be able to write down. Hence, we won't be too concerned about $\mathcal{F}$ in this course. That everything is finite is also quite helpful in this regard.

3. Some texts reserve the notion of random variables to deterministic maps from Ω to the reals and call deterministic functions that map from Ω to some other set (such as the sets $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{R}$ in our context) a *random element*. We will be more flexible in that we will also use random variables (for simplicity) when talking about functions of this second type. Another complication that we sweep under the rug is that with some sets, like the reals, not all functions from Ω to the reals can be considered. This is because for some of these functions with the usual questions we can have, such as those about the probabilities of these function taking values in some sets, we could run into problems—that some of those probabilities are not even defined. Lucky for us, this problem completely vanishes if the range of the function considered is finite and the set system of the "suitable" subsets of Ω, that we assign probabilities to, includes all finite subsets of Ω (something which we will tacitly assume).

4. If we freeze the internal variables in the program and call the `step` function with some action, then, due to the randomness of this function, what we get is a distribution over rewards and observations. The reproducibility up to the randomness just means that this distribution only depends on what the last observation and the action is. Having said this, we see that the proposition is really not saying that much; it just connects the programs, which are perhaps easier to understand, with the formal probabilistic statements. (But then, are computer programs really easier to understand compared with formal probabilistic statements? We leave this question for the reader to answer!)

5. We can simply take $\Omega = [0, 1]^{\mathbb{N}}$ and $U_i(\omega) = \omega_i$ where $\omega = (\omega_0, \omega_1, \dots)$. Then there are powerful mathematical results that guarantee that $\mathbb{P}$ with the required properties can be defined for a meaningful choice of a "suitable" subset system $\mathcal{F}$ of Ω.

6. Note that here $\mathbb{P}$ is fixed regardless of the environment and the action-selection mechanism. With this choice, that is when $\mathbb{P}$ is fixed regardless of the environment and the action-selection mechanism, the sequence $S_0, A_0, R_1, S_1, A_1, \dots$ will necessarily depend on the environment and the action-selection mechanism. But the choice of Ω and $\mathbb{P}$ is far from being unique. The opposite of "fixing Ω and $\mathbb{P}$, and making $S_0, A_0, R_1, S_1, A_1, \dots$ depend on the environment and the action-selection mechanism" is to "fix Ω and $S_0, A_0, R_1, S_1, A_1, \dots$, once and for all, and then make $\mathbb{P}$ depend on the environment and the action-selection mechanism." Both approaches are equally valid and interchangeable; just that one needs to be clear about which approach one follows or the reader may not know the meaning of the various symbols, as they are not the same across these two systems!

7. Formally, a history dependent policy π is a sequence of maps from histories, of growing length, to distributions over actions: $\pi = (\pi_0, \pi_1, \dots)$, where $\pi_t : \mathcal{H}_t \to \mathcal{M}_1(\mathcal{A})$ for all $t \geq 0$. Here $\mathcal{M}_1(\mathcal{A})$ denotes the set of distributions over $\mathcal{A}$; and $\mathcal{H}_0 = \mathcal{S}$, and for $t \geq 0$, $\mathcal{H}_{t+1} = \mathcal{H}_t \times \mathcal{A} \times \mathcal{R} \times \mathcal{S}$. ($\mathcal{H}_t$ denotes the set of histories up until timestep t.) When $\mathcal{A}$, $\mathcal{R}$, and $\mathcal{S}$ are not finite, some further conditions are needed, but these are beyond the text and you can trust that these conditions are hard to violate (but it is actually a tricky discussion of why these, yet unspecified, conditions are necessary, which is mostly missing from the literature).

[8] The function p is also known as the Markov kernel, the Markov transition kernel, or just the transition function. People never stop inventing new names for the same thing.

7 Bibliography

Michael J Evans and Jeffrey S Rosenthal. *Probability and statistics: The science of uncertainty*. W H Freeman & Co, 2010. URL https://www.utstat.toronto.edu/mikevans/jeffrosenthal/book.pdf.