MITx Video | 6.041x | Spring 2014 | Video transcript | L25-3: Probability of a path

Let's do a little warm-up now before moving to the central topic of today's lecture on how to apply these transition probabilities in order to calculate some more interesting path behaviors of Markov chains. So suppose that you are given the following Markov chain, and you are asked to calculate the probability that starting in state 1, you go successively to state 2, then state 6, and then 7. So this is really what we are asked here. You start at 1, then you go to 2, 6, and 7. So how to calculate such a probability?

Well, we can use a version of the multiplicative rule we have introduced before. And so what is the specific format of that rule that I'm going to use here? You have three events like that, B, C, D. They are conditioned on A. And I will say that this is the probability of B given A, times the probability of C given A intersection B, times the probability of D given A intersection B intersection C. So this is a version of the multiplicative rule. And this is what we're going to use here, so let's do it.

This is going to be equals to the probability that X1 equals 2 given X0 equals 1, times the probability that X2 equals 6 given X0 equals 1, and X1 equals 2 times the probability that X3 equals 7 times X0 equals 1, X1 equals 2, and then X2 equals 6. So multiplication rule.

Now here, this is p12. And here, we are using the Markov property. What it says here is, what is the probability that I will be in state 6, given that I was in 1 and then 2. And we know that having the entire trajectory doesn't matter, it's just the last step. So it's essentially times, this one is p26. And for the same reason, this one is nothing else than p67. The important message here is that to find the probability of a specific trajectory like this one, you just need to multiply the transition probabilities that you find along the trajectory. So this is what we have here.

Now suppose that you want to find the probability that in four times steps, you find yourself in a specific state say, 7, given that you started in a specific initial state say, 2. This is really what we want here. You're here, and then you end up here after four steps. So how do you calculate that?

Well, one way is to use our recurrence formula for rij that we have described before. Another, for small examples like this one, when the number of times steps in the future is small-- in that case four-- one can perhaps use a brute force calculation. So what is a brute force calculation? Well, you try to enumerate all possible trajectories, and then you sum all their probabilities. So in that case, I think we have three trajectories, but I'm not so sure, so let's try to enumerate them.

So you start here, and one possibility in a one-step transition is to go to 6. Then from 6 you go to 7, then from 7 you go back to 6. And then from 6 you again go to 7, all right? So that, if we look and use the rule that we have developed before, it would be the probability of going to 6 times p67 times p76 and times p67. So this is one way of doing it. What would be another path?

Well, from 2, instead of going to 6 you could go to 1 in one transition. Then you go back to 2, then we go to 6, and then right up to 7. So essentially here, what we have is plus p21 times p12 times p26 times p67.

And what is the third way to do that? So there is a third path. You're starting from here, yes? You go here in one step. Here you do a jump on itself, and another jump, and then you go to 7. So in that case, you would have plus p26 times p66 times p66 times p67. So here we have the entire solution, all right? So this is what is called, by brute force. Now of course, if instead of 4, here, you had something like 200, the number of trajectories would grow exponentially with this number of steps. And this is not practical anymore. Using the recursion formula would have been a much better approach. And in some sense would have a much better complexity.

Essentially, a linear growth as a function of time steps in the future. More precisely, for a chain with a state space of m, at each time steps you need to update all our rij's. So for each pair of ij, so each of these would take about m squared. And so the total computational complexity would grow about n times m squared as a function of n.