Contents

1 Probabilistic Actions

Motivation

In the real world, robots do not always execute actions perfectly, for a variety of reasons. We can use discrete variables to model the state a robot is in, and actions that connect these states. To model the uncertainty with executing an action, we will have to introduce the language of probability. Below we take a Bayesian view of probability, rather than a frequentist one. Conditional probability distributions are a way to model how we can affect the state of the robot by actions. We can model this graphically, using directed edges to specify conditional probabilities on variables, and use simulation in a graphical model to explore what a sequence of actions will yield as outcome. The graphical model approach allows us to easily extend probabilistic actions to factored state representations.

1.1 Real Robots

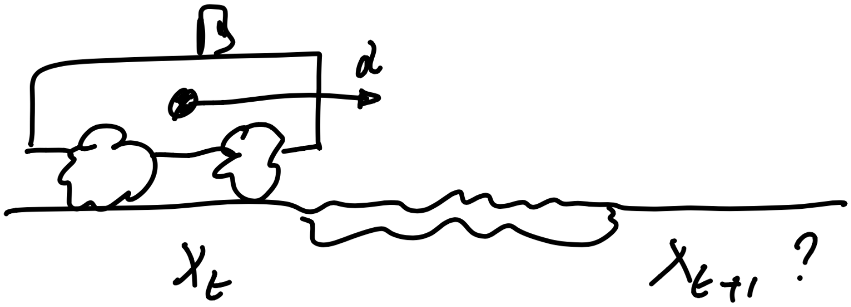


Figure 1: Mobile robot driving in mud.

In the real world, robots do not always execute actions perfectly, for a variety of reasons. For example, as shown in Figure 1, a robot may want to drive forwards in an outdoor environment but mud under its wheels might prevent it from traveling as far as we would like.

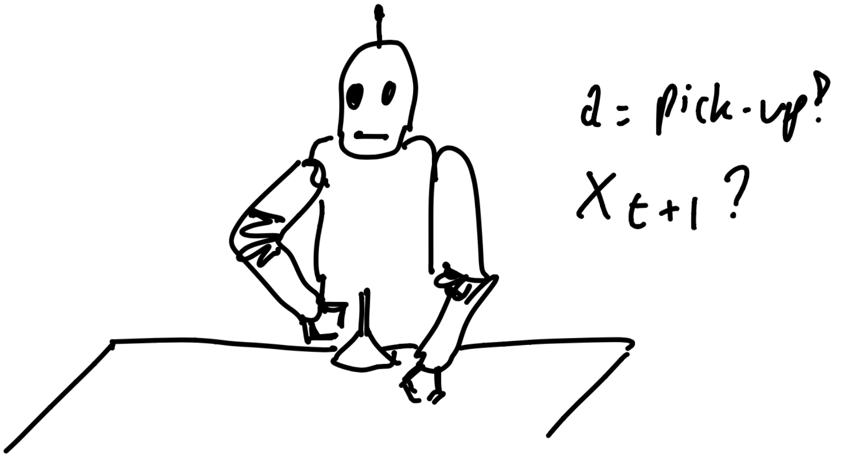


Figure 2: A humanoid attempting to pick up an object.

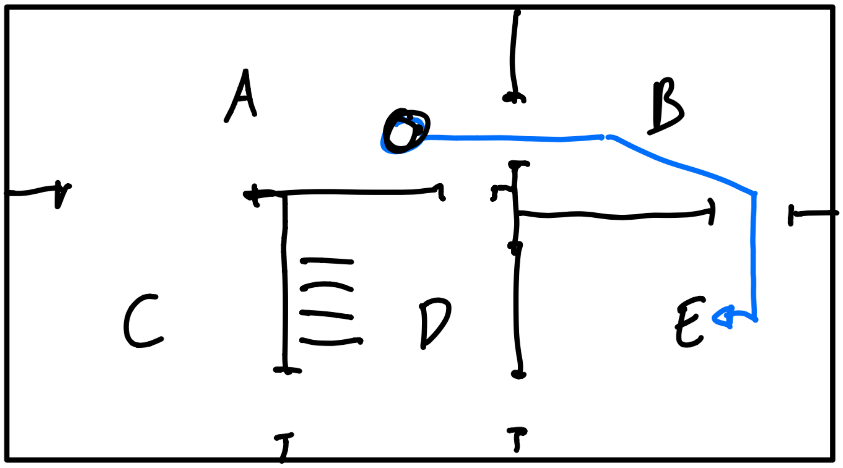


Figure 3: A robot vacuum cleaner facing an uncertain future.

Or a robot with an articulated arm (Figure 2) might attempt picking up an object, but fail. Finally, your robot vacuum cleaner might attempt to clean a particular area, schematically shown in Figure 3, only to find the door closed or its way blocked.

1.2 Atomic State via Discrete Variables

We can use discrete variables to model the state a robot is in. Even though the examples in Section 1.1 are all continuous, we can still represent states using discrete variables. For example, the robot vacuum cleaner can be inside a particular room, so we could associate a value of a discrete variable with each of the rooms in the house, as in Figure 4. We write $X \in \{A, B, C, D, E\}$.

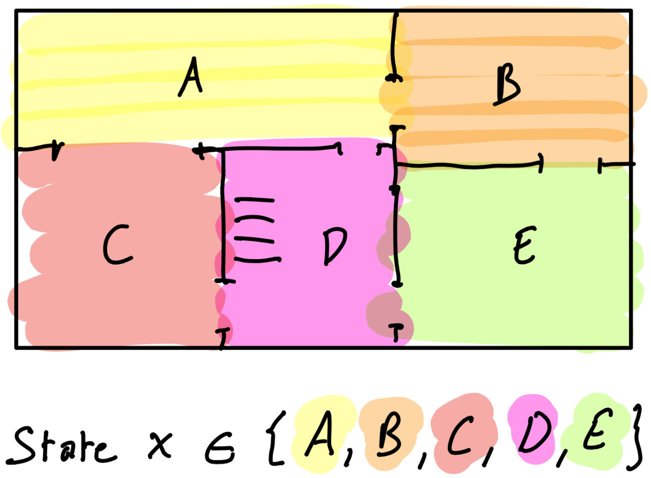


Figure 4: A discrete state representation corresponding to different rooms in the house.

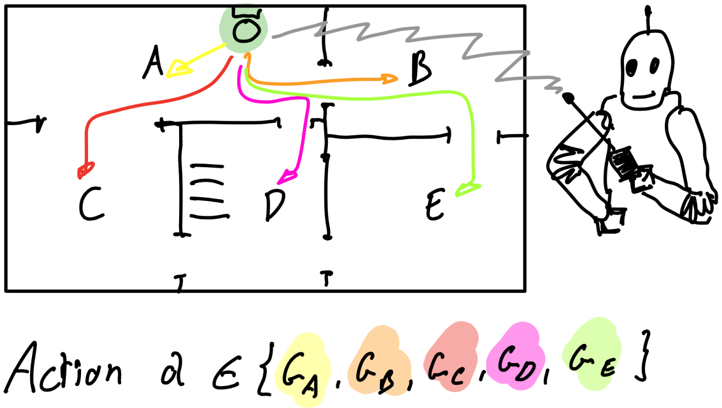


Figure 5: In addition to a set of possible states, we also have a set of possible actions.

We also need to model the actions that can cause transitions between those states. We can imagine that we have a phone interface, by which we can ask the robot to go to a particular room. Hence, if we have five rooms in the house this would correspond to five possible actions, as illustrated in Figure 5. We write $a \in \{G_A, G_B, G_C, G_D, G_E\}$. In this example we have just as many actions as there are states, but this is not necessarily the case. For example, we could have an action "return to base", which attempts to return to the base from anywhere in the house.

1.3 Probabilistic Outcomes of Actions

Even when we start from a well-known state, after giving the robot a command we cannot be sure where it will end up. In the robot vacuum example, this could depend both on the starting state and the command given to the robot. This is illustrated graphically in Figure 6. The figure illustrates the execution of an action $a = G_C$, i.e., “go to room C”, starting from the base in room A. After the robot attempts this action we are only be able to tell which room we are in with varying degree of belief in each possible value of the state.

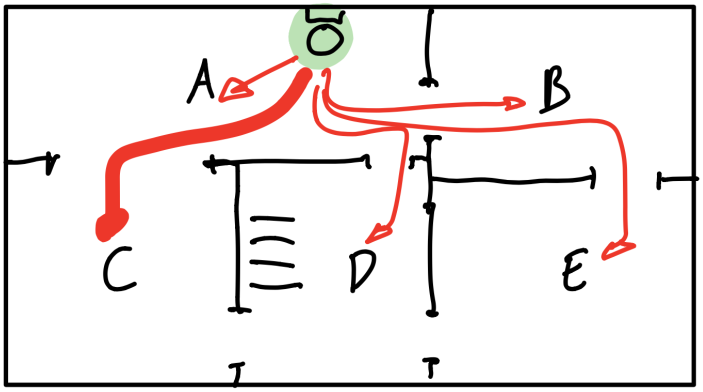


Figure 6: An action, in this case "go to room C", might succeed most of the time but fail some of the time. Above, the thickness of the arrows encodes how likely the robot ends in each room, starting from room A.

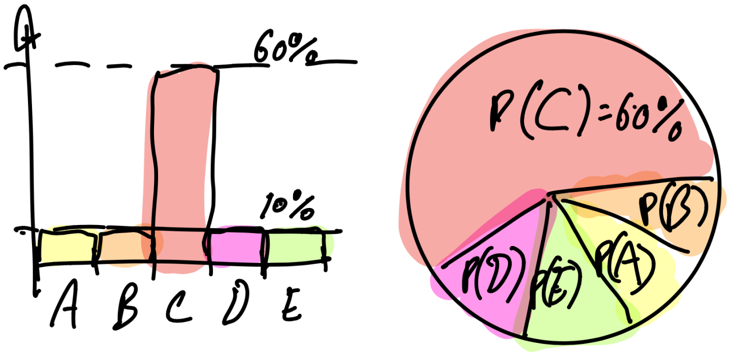


Figure 7: We can represent the probability $P(x|a = G_C)$ of being in a particular state after taking action G_C as a vector of numbers between 0 and 1, adding up to 1. We can represent this graphically as a bar chart or a pie chart.

To model the uncertainty associated with executing an action, we now introduce the language of probability. In particular, a **discrete probability distribution** assigns probability values to each value of a discrete random variable X, taking on one of a finite set of discrete values.

We can represent the parameters of a discrete probability distribution as a vector of probabilities $p_i \doteq P(X = x_i)$, called the **probability mass function** or PMF. Note that we denote discrete random variables with a capitalized symbol, e.g., X for a state, but when in a formula we talk about the *value* of a random variable, we use a lowercase x. An example is given in Table 1 and graphically illustrated in Figure 7.

Probability mass functions obey two basic axioms,

$$p_i \geq 0$$
$$\sum p_i = 1$$

i.e., the probability p_i of an outcome x_i is non-negative, and the probabilities p_i of all outcomes

x_i	$P(X = x_i)$
A	0.2
B	0.1
C	0.6
D	0.1
E	0.1

Table 1: Example of a probability mass function.

have to sum up to 1. By necessity then, all $p_i \leq 1$ as well. We frequently use percentages to express these values, in which case they have to add up to 100%.

1.4 Bayesian vs. Frequentist

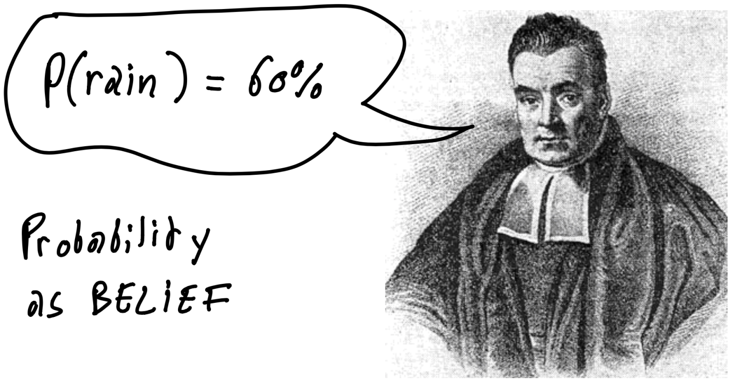


Figure 8: The Reverend Thomas Bayes gave his name to associating probabilities with the strength of beliefs rather than a frequency of events, even though this seems to have been first introduced by Laplace.

We take a Bayesian view of probability, rather than a frequentist one. This means that we see probabilities as describing our knowledge about events, rather than tallying up frequencies by which they occur. Think of the weather-person talking about the probability of rain tomorrow. Probabilities viewed this way can be used to describe knowledge about the state of the world, and how actions affect the state of an agent and the world.

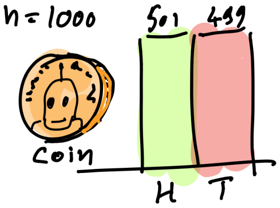


Figure 9: The caricature of the frequentist view involves counting many heads and tails.

This is to be contrasted with a frequentist view, where probabilities are used to describe the frequencies of events in a series of repeated trials. A Bayesian, instead, might qualify their knowledge

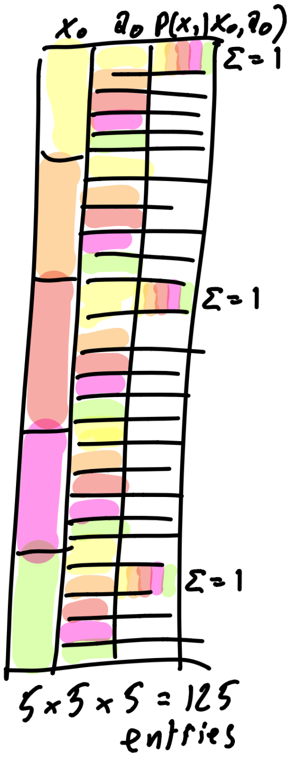


Figure 10: A conditional probability table or CPT, specifying a separate PMF on state x_1 for each of the 5 possible starting states x_0 and each of the 5 possible actions a_0.

about an event that has not even happened yet, let alone multiple times. Of course, in most cases this belief is based on experience, i.e., lots of repeated events in the past, and so it can be seen that perhaps these views are not so different after all.

1.5 Conditional Probability Distributions

Conditional probability distributions are a way to model how we can affect the state of the robot by actions. In general, conditional distributions (or *densities*, for continuous variables) are a way represent knowledge about the world in robotics. While here we focus only on actions, in later sections we also use them to model observations made by sensors.

In general, the probability mass function of a single variable X can be parameterized by one or more parameters Y, whose value we will assume as known. This corresponds to the notion of a **conditional probability distribution**, which we write as

$$P(X|Y=y).$$

The "bar" in the notion separates the variable we want to characterize from the conditions. Note that given a particular value of Y, this is just a distribution over X, with parameters given by a PMF, as before.

Applied to robots with discrete states and actions, we can model the effects of actions by the conditional probability distribution $P(X_{t+1}|X_t=x, A=a)$ on the next state X_{t+1} , given the value x of the current state X_t, and the value a of the action A.

Because Y can take on several values, we now need a **conditional probability table** or CPT to exhaustively describe our knowledge. In the vacuum robot example, we would need to create a conditional probability table that fully specifies the probability of ending up in any of the rooms, for all combinations of actions and possible starting states. For example, Figure 10 shows a CPT specifying a separate PMF on state x_1 for each of the 5 possible starting states x_0 and each of the 5 possible actions a_0.

Conditional probability tables do not *have* to be specified as giant tables. Because the state space is potentially quite large, such a **state transition model** is almost never explicitly specified, but rather exploits the semantics of the states and actions to provide a more compact representation of the associated CPT.

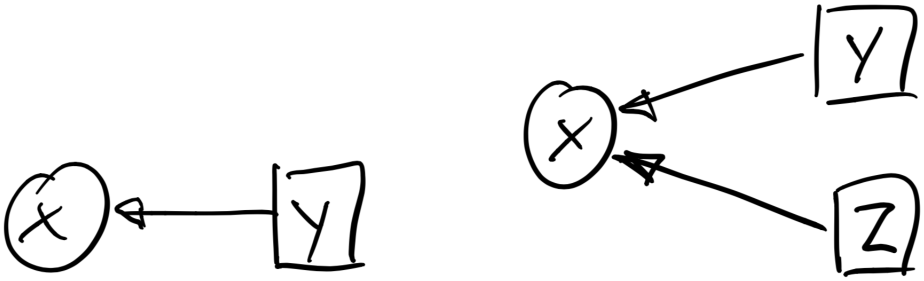
Exercises

1. Even though the CPT in Figure 10 has 125 numbers in it, how many independent degrees of freedom do we actually have when specifying this CPT?
2. Specify a parametric conditional density for the action models for the vacuum robot that is somewhat realistic, yet not completely deterministic.
3. It is possible to create models that do not reflect everyday physics. For example, how could we model the game "Portal"?

1.6 A Simple Graphical Model

Figure 11: Graphical representation of a single random variable.

Graphically, we can represent a probability mass function (PMF) over a single random variable X as a single node, with the label X, as shown in Figure 11. By convention we use a circular node to represent random variables.



(a) Graphical representation of $P(X|Y)$

(b) Graphical representation of $P(X|Y, Z)$

Figure 12: Conditional probability distributions. Above X is a random variable, but Y and Z are known parameters, indicated by the square box.

We can use directed edges to specify conditional probabilities on variables. Graphically, we represent a *known* parameter Y as a square node within a graph, and a random variable governed by a conditional probability as having an incoming edge from the parameter, as shown in Figure 12a. Note that it is possible to have multiple parameters, e.g., a conditional probability $P(X|Y, Z)$ is shown in Figure 12b.

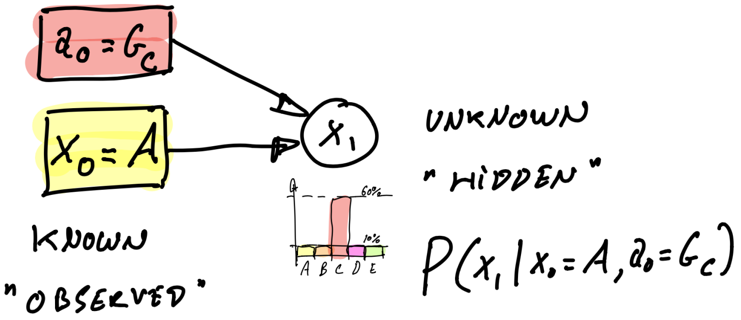


Figure 13: Graphical representation of the effect of action $a_0 = G_C$ starting from state $x_0 = A$.

As an application, we can use this simple graphical model to represent the probabilistic transitions between states, as shown in Figure 13. This models the effect of a particular action starting in a particular state.

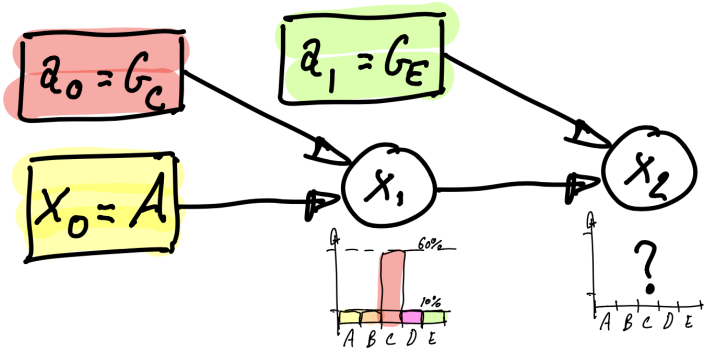


Figure 14: Graphical model of two consecutive actions $a_0 = G_C$ and $a_1 = G_E$.

A Markov chain is a graphical representation of repeated state transitions, given a known sequence of actions. Indeed, by chaining "action fragments" together as shown in Figure 14. While we know from the PMF given earlier in Table 1 where we expect to end up after one action, it is less straightforward to calculate what our belief is after *two* actions. We will defer the exact calculation to the next Chapter, but in the next Section we show how to get an approximate answer to that question via sampling.

1.7 Forward Simulation

We can use simulation in a graphical model to explore what a sequence of actions will yield as outcome. In the example above, it could very well be that if we repeatedly command the robot

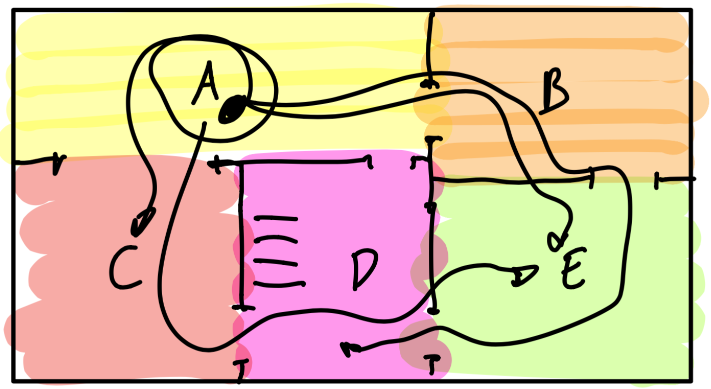


Figure 15: Multiple rollouts

to go to a particular room, we can be relatively sure that it will end up there eventually. On the other hand, if some obstacle is blocking its way, then no amount of trying might get us to the goal. And finally, it could be that by repeatedly trying, the obstacle is eventually dislodged, i.e., we have changed the state of the world. In those cases, we will have to consider the status of this obstacle as part of the state.

Given an *unconditional* distribution, we can draw a sample from it. To do so, we use an algorithm called **inverse transform sampling**. For this we need to first assign an order to the outcomes,, which we can do by adopting order associated with the (arbitrary) integer indices by which we enumerate outcomes, i.e.: $x_i < x_j$ if $i < j$. Given this order, we can the then compute the **cumulative distribution function** or CDF, which is the probability associated with the subset of outcomes with indices less than or equal to a given index i:

$$F(x_i) = P(X \leq x) = \sum_{j \leq i} P(X = x_i) = \sum_{j \leq i} p_j$$

Given this, the inverse transform algorithm is simple: generate random number $0 \leq u \leq 1$, then return x_i such that i is the smallest index such that $F(x_i) \geq u$.

x_i	$P(X = x_i)$	$F(x_i)$
A	0.1	0.1
B	0.1	0.2
C	0.6	0.8
D	0.1	0.9
E	0.1	1.0

Table 2: Cumulative distribution function.

As an example, in Table 2, we have shown the CDF for the robot state, using lexicographical order, here arranged from top to bottom. To sample a random value, drawing $u = 0.6$ would yield a sample $X = C$.

To sample from a *conditional* distribution $p(X|Y)$ we need to make sure we sample the variable Y beforehand, and then proceed simply by selecting the appropriate PMF depending on the value of Y. We can then proceed as before using the inverse transform sampling method.

Forward sampling in a Markov chain simply repeats these steps in succession, proceeding from left to right. Simulating a robot given a sequence of actions $a_1, a_2, \ldots$ is then equivalent to sampling from this Markov chain:

1. First, sample the initial state s_1 from $P(X_1)$, a prior over the state. Set $k = 1$.
2. Simulate the effect of the next action by sampling the next state x_{k+1} from

$$P(X_{k+1}|X_k = x_k, A_k = a_k).$$

3. Increase k and return to step 2.

Figure 15 sketches out 4 different "rollouts" by simulating in this way. After that, we can approximate the PMF of the final state by construction a histogram over the possible values of the state.

While simple, simulating from a forward model is a rather important technique, and underlies some of the recent successes in deep reinforcement learning, as well as the success of DeepMind in beating the world's best players of the game of Go.

Exercises

1. Implement inverse transform sampling for the CDF in Table 2, and verify your sampler produces a histogram that approximates the desired PMF.
2. Come up with a CPT for action G_E, and simulate 2 different realizations for the Markov chain in Figure 14.

1.8 Factored State Representations

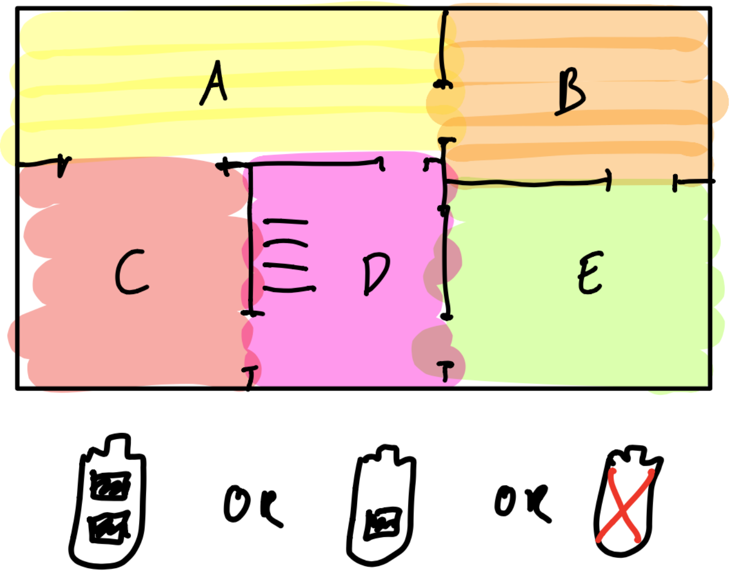


Figure 16: A factored state representation for the vacuum robot might have one variable for the room the robot is in, and another variable describing its battery status.

Factored state representations are useful if the state of the robot can be described using features that are relatively independent of each other. Continuing our example, the robot vacuum cleaner might also run out of battery power, so we could associate a different variable with its battery status, as shown in Figure 16. The state of the robot would then be specified by the combination of these two variables. Note that now the total number of possible states is combinatorial: if there are five rooms and three different battery levels, we have a total of 15 possible states for the robot.

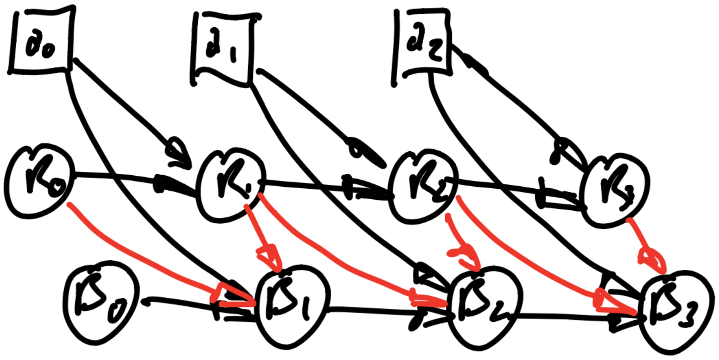


Figure 17: A Markov chain for a factored state representation. The red arrows above show that the change in batter status can depend on the room transition.

The graphical model approach allows us to easily extend probabilistic actions to factored state representations. Figure 17 shows how to do it: we just have to make sure to sample each part of the state in sequence, and allow for the possibility that each state variable depends on the earlier chosen variables. This is mathematically equivalent to having one CPT for the combined state.

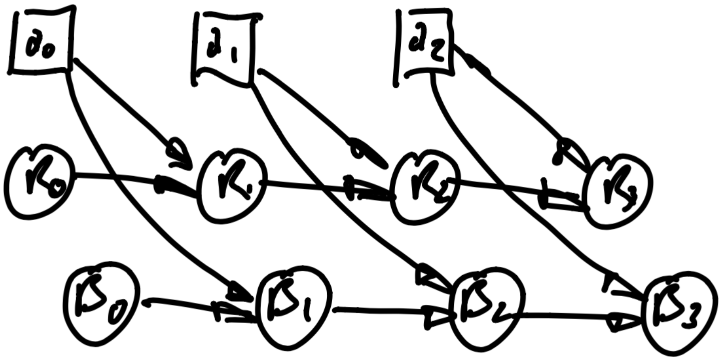


Figure 18: If the battery status transition does not depend on the room transition, but only on the action, then we have two independent Markov chains given the state sequence.

However, while in general the actions might affect the entire state, a factored state representation becomes computationally attractive when the effect of an action on each variable might be conditionally independent of what happens to the other variables. This is illustrated for our example in Figure 18, which is a sparser graph than the one in Figure 17. Indeed: the red edges are missing, indicating that the change in battery status is only dependent on the action.