

## 15. Probabilistic Reasoning over Time

**15.1** [state-augmentation-exercise] Show that any second-order Markov process can be rewritten as a first-order Markov process with an augmented set of state variables. Can this always be done *parsimoniously*, i.e., without increasing the number of parameters needed to specify the transition model?

**15.2** [markov-convergence-exercise] In this exercise, we examine what happens to the probabilities in the umbrella world in the limit of long time sequences.

1. Suppose we observe an unending sequence of days on which the umbrella appears. Show that, as the days go by, the probability of rain on the current day increases monotonically toward a fixed point. Calculate this fixed point.
2. Now consider *forecasting* further and further into the future, given just the first two umbrella observations. First, compute the probability  $P(r_{2+k}|u_1, u_2)$  for  $k = 1 \dots 20$  and plot the results. You should see that the probability converges towards a fixed point. Prove that the exact value of this fixed point is 0.5.

**15.3** [island-exercise] This exercise develops a space-efficient variant of the forward-backward algorithm described in Figure [forward-backward-algorithm](#) (page [forward-backward-algorithm](#)). We wish to compute  $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$  for  $k = 1, \dots, t$ . This will be done with a divide-and-conquer approach.

1. Suppose, for simplicity, that  $t$  is odd, and let the halfway point be  $h = (t + 1)/2$ . Show that  $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$  can be computed for  $k = 1, \dots, h$  given just the initial forward message  $\mathbf{f}_{1:0}$ , the backward message  $\mathbf{b}_{h+1:t}$ , and the evidence  $\mathbf{e}_{1:h}$ .
2. Show a similar result for the second half of the sequence.
3. Given the results of (a) and (b), a recursive divide-and-conquer algorithm can be constructed by first running forward along the sequence and then backward from the end, storing just the required messages at the middle and the ends. Then the algorithm is called on each half. Write out the algorithm in detail.
4. Compute the time and space complexity of the algorithm as a function of  $t$ , the length of the sequence. How does this change if we divide the input into more than two pieces?

**15.4** [flawed-viterbi-exercise] On page [flawed-viterbi-page](#), we outlined a flawed procedure for finding the most likely state sequence, given an observation sequence. The procedure involves finding the most likely state at each time step, using smoothing, and returning the sequence composed of these states. Show that, for some temporal probability models and observation sequences, this procedure returns an impossible state sequence (i.e., the posterior probability of the sequence is zero).

**15.5** [hmm-likelihood-exercise] Equation ([matrix-filtering-equation](#)) describes the filtering process for the matrix formulation of HMMs. Give a similar equation for the calculation of likelihoods, which was described generically in Equation ([forward-likelihood-equation](#)).

**15.6** Consider the vacuum worlds of Figure [vacuum-maze-ch4-figure](#) (perfect sensing) and Figure [vacuum-maze-hmm2-figure](#) (noisy sensing). Suppose that the robot receives an observation sequence such that, with perfect sensing, there is exactly one possible location it could be in. Is this location necessarily the most probable location under noisy sensing for sufficiently small noise probability  $\epsilon$ ? Prove your claim or find a counterexample.

**15.7** [hmm-robust-exercise] In Section [hmm-localization-section](#), the prior distribution over locations is uniform and the transition model assumes an equal probability of moving to any neighboring square. What if those assumptions are wrong? Suppose that the initial location is actually chosen uniformly from the northwest quadrant of the room and the action actually tends to move southeast[[hmm-robot-southeast-page](#)]. Keeping the HMM model fixed, explore the effect on localization and path accuracy as the southeasterly tendency increases, for different values of  $\epsilon$ .

**15.8** [roomba-viterbi-exercise] Consider a version of the vacuum robot (page [vacuum-maze-hmm2-figure](#)) that has the policy of going straight for as long as it can; only when it encounters an obstacle does it change to a new (randomly selected) heading. To model this robot, each state in the model consists of a *(location, heading)* pair. Implement this model and see how well the Viterbi algorithm can track a robot with this model. The robot's policy is more constrained than the random-walk robot; does that mean that predictions of the most likely path are more accurate?

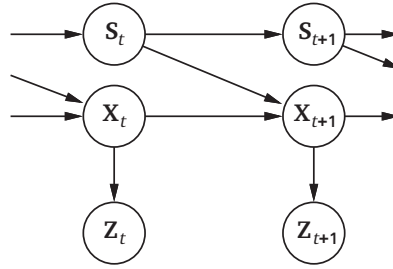
**15.9** We have described three policies for the vacuum robot: (1) a uniform random walk, (2) a bias for wandering southeast, as described in Exercise [hmm-robust-exercise](#), and (3) the policy described in Exercise [roomba-viterbi-exercise](#). Suppose an observer is given the observation sequence from a vacuum robot, but is not sure which of the three policies the robot is following. What approach should the observer use to find the most likely path, given the observations? Implement the approach and test it. How much does the localization accuracy suffer, compared to the case in which the observer knows which policy the robot is following?

**15.10** This exercise is concerned with filtering in an environment with no landmarks. Consider a vacuum robot in an empty room, represented by an  $n \times m$  rectangular grid. The robot's location is hidden; the only evidence available to the observer is a noisy location sensor that gives an approximation to the robot's location. If the robot is at location  $(x, y)$  then with probability .1 the sensor gives the correct location, with probability .05 each it reports one of the 8 locations immediately surrounding  $(x, y)$ , with probability .025 each it reports one of the 16 locations that surround those 8, and with the remaining probability of .1 it reports "no reading." The robot's policy is to pick a direction and follow it with probability .8 on each step; the robot switches to a randomly selected new heading with probability .2 (or with probability 1 if it encounters a wall). Implement this as an HMM and do filtering to track the robot. How accurately can we track the robot's path?

**15.11** This exercise is concerned with filtering in an environment with no landmarks. Consider a vacuum robot in an empty room, represented by an  $n \times m$  rectangular grid. The robot's location is hidden; the only evidence available to the observer is a noisy location sensor that gives an approximation to the robot's location. If the robot is at location  $(x, y)$  then with probability .1 the sensor

gives the correct location, with probability .05 each it reports one of the 8 locations immediately surrounding  $(x, y)$ , with probability .025 each it reports one of the 16 locations that surround those 8, and with the remaining probability of .1 it reports “no reading.” The robot’s policy is to pick a direction and follow it with probability .7 on each step; the robot switches to a randomly selected new heading with probability .3 (or with probability 1 if it encounters a wall). Implement this as an HMM and do filtering to track the robot. How accurately can we track the robot’s path?

**Figure [switching-kf-figure]** A Bayesian network representation of a switching Kalman filter. The switching variable  $S_t$  is a discrete state variable whose value determines the transition model for the continuous state variables  $\mathbf{X}_t$ . For any discrete state  $i^*$ , the transition model  $\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{X}_t, S_t = i^*)$  is a linear Gaussian model, just as in a regular Kalman filter. The transition model for the discrete state,  $\mathbf{P}(S_{t+1}|S_t)$ , can be thought of as a matrix, as in a hidden Markov model.



**15.12 [switching-kf-exercise]** Often, we wish to monitor a continuous-state system whose behavior switches unpredictably among a set of  $k$  distinct “modes.” For example, an aircraft trying to evade a missile can execute a series of distinct maneuvers that the missile may attempt to track. A Bayesian network representation of such a **switching Kalman filter** model is shown in Figure [switching-kf-figure](#).

1. Suppose that the discrete state  $S_t$  has  $k$  possible values and that the prior continuous state estimate  $\mathbf{P}(\mathbf{X}_0)$  is a multivariate Gaussian distribution. Show that the prediction  $\mathbf{P}(\mathbf{X}_1)$  is a **mixture of Gaussians**—that is, a weighted sum of Gaussians such that the weights sum to 1.
2. Show that if the current continuous state estimate  $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$  is a mixture of  $m$  Gaussians, then in the general case the updated state estimate  $\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1})$  will be a mixture of  $km$  Gaussians.
3. What aspect of the temporal process do the weights in the Gaussian mixture represent?

The results in (a) and (b) show that the representation of the posterior grows without limit even for switching Kalman filters, which are among the simplest hybrid dynamic models.

**15.13 [kalman-update-exercise]** Complete the missing step in the derivation of Equation ([kalman-one-step-equation](#)) on page [kalman-one-step-equation](#), the first update step for the one-dimensional Kalman filter.

**15.14 [kalman-variance-exercise]** Let us examine the behavior of the variance update in Equation ([kalman-univariate-equation](#)) (page [kalman-univariate-equation](#)).

1. Plot the value of  $\sigma_t^2$  as a function of  $t$ , given various values for  $\sigma_x^2$  and  $\sigma_z^2$ .
2. Show that the update has a fixed point  $\sigma^2$  such that  $\sigma_t^2 \rightarrow \sigma^2$  as  $t \rightarrow \infty$ , and calculate the value of  $\sigma^2$ .
3. Give a qualitative explanation for what happens as  $\sigma_x^2 \rightarrow 0$  and as  $\sigma_z^2 \rightarrow 0$ .

**15.15 [sleep1-exercise]** A professor wants to know if students are getting enough sleep. Each day, the professor observes whether the students sleep in class, and whether they have red eyes. The professor has the following domain theory:

- The prior probability of getting enough sleep, with no observations, is 0.7.
- The probability of getting enough sleep on night  $t$  is 0.8 given that the student got enough sleep the previous night, and 0.3 if not.
- The probability of having red eyes is 0.2 if the student got enough sleep, and 0.7 if not.
- The probability of sleeping in class is 0.1 if the student got enough sleep, and 0.3 if not.

Formulate this information as a dynamic Bayesian network that the professor could use to filter or predict from a sequence of observations. Then reformulate it as a hidden Markov model that has only a single observation variable. Give the complete probability tables for the model.

**15.16 [sleep1-exercise]** A professor wants to know if students are getting enough sleep. Each day, the professor observes whether the students sleep in class, and whether they have red eyes. The professor has the following domain theory:

- The prior probability of getting enough sleep, with no observations, is 0.6.
- The probability of getting enough sleep on night  $t$  is 0.8 given that the student got enough sleep the previous night, and 0.2 if not.
- The probability of having red eyes is 0.2 if the student got enough sleep, and 0.7 if not.
- The probability of sleeping in class is 0.1 if the student got enough sleep, and 0.3 if not.

Formulate this information as a dynamic Bayesian network that the professor could use to filter or predict from a sequence of observations. Then reformulate it as a hidden Markov model that has only a single observation variable. Give the complete probability tables for the model.

**15.17** For the DBN specified in Exercise [sleep1-exercise](#) and for the evidence values

$\mathbf{e}_1 = \text{not red eyes, not sleeping in class}$   
 $\mathbf{e}_2 = \text{red eyes, not sleeping in class}$   
 $\mathbf{e}_3 = \text{red eyes, sleeping in class}$

perform the following computations:

1. State estimation: Compute  $P(\text{EnoughSleep}_t | \mathbf{e}_{1:t})$  for each of  $t = 1, 2, 3$ .
2. Smoothing: Compute  $P(\text{EnoughSleep}_t | \mathbf{e}_{1:3})$  for each of  $t = 1, 2, 3$ .
3. Compare the filtered and smoothed probabilities for  $t = 1$  and  $t = 2$ .

**15.18** Suppose that a particular student shows up with red eyes and sleeps in class every day. Given the model described in Exercise [sleep1-exercise](#), explain why the probability that the student had enough sleep the previous night converges to a fixed point rather than continuing to go down as we gather more days of evidence. What is the fixed point? Answer this both numerically (by computation) and analytically.

**15.19** [battery-sequence-exercise] This exercise analyzes in more detail the persistent-failure model for the battery sensor in Figure [battery-persistence-figure\(a\)](#) (page [battery-persistence-figure](#)).

1. Figure [battery-persistence-figure\(b\)](#) stops at  $t = 32$ . Describe qualitatively what should happen as  $t \rightarrow \infty$  if the sensor continues to read 0.
2. Suppose that the external temperature affects the battery sensor in such a way that transient failures become more likely as temperature increases. Show how to augment the DBN structure in Figure [battery-persistence-figure\(a\)](#), and explain any required changes to the CPTs.
3. Given the new network structure, can battery readings be used by the robot to infer the current temperature?

**15.20** [dbn-elimination-exercise] Consider applying the variable elimination algorithm to the umbrella DBN unrolled for three slices, where the query is  $\mathbf{P}(R_3 | u_1, u_2, u_3)$ . Show that the space complexity of the algorithm—the size of the largest factor—is the same, regardless of whether the rain variables are eliminated in forward or backward order.