Chapter 15

Probabilistic Reasoning over Time

15.1 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?

Answer let X_t be a variable that can take the state $x_1, x_2, ..., x_k$. The first-order Markov Chain property states that :

$$P(W_t = w|W_{t-1}, W_{t-2}, W_{t-3}...) = P(W_t = w|W_{t-1})$$

while the second-order Markov Chain property is defined by :

$$P(W_t = w|W_{t-1}, W_{t-2}, W_{t-3}...) = P(W_t = w|W_{t-1}, W_{t-2})$$

We can transform the second-order Markov Chain into the first-order Markov Chain by redefining the state spaces as follow :
Let $Z_{t-1,t}$ be a variable that takes 2 consecutive states of the X_t variable, that is to say : If X_t can take x_1, x_2, x_3 as value then we define $Z_{t-1,t}$ such that $Z_{t-1,t}$ can take either $x_1x_1, x_1x_2, x_1x_3, x_2x_1, x_2x_2, x_2x_3, x_3x_1, x_3x_2, x_3x_3$.
In this new state-space we have :

$$P(Z_{t-1,t} = z_{t-1,t}|Z_{t-1,t-1}, Z_{t-2,t-1}, Z_{t-3,t-2}...) = P(Z_{t-1,t} = z_{t-1,t}|Z_{t-1,t-1})$$

15.2 In this exercise, we examine what happens to the probabilities in the umbrella world in the limit of long time sequences.

a. 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.

b. 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 $k = 1, ..., 20$ and plot the results. You should see that the probability converges towards a fixed point.

Answer

a. We want to retrieve the probability that the current day is a rainy day, that is to say R_t, knowing that we saw $u_{1:t}$. To compute this probability we can use the filtering Formula (15.5) :

$$P(R_t|u_{1:t}) = \alpha P(u_{t+1}|R_{t+1}) \sum_{R_{t-1}} P(R_t|R_{t-1})P(R_{t-1}|u_{1:t-1})$$

Furthermore we want to compute the fixed point. This condition gives us the relation :

$$P(R_t|u_{1:t}) = P(R_{t-1}|u_{1:t-1})$$

Replacing in the previous equation we get the relation :

$$P(R_t|u_{1:t}) = \alpha P(u_{t+1}|R_{t+1}) \sum_{R_{t-1}} P(R_t|R_{t-1})P(R_t|u_{1:t})$$

As there is only 2 states for the weather : there is rain or there is not... We can replace $P(R_{t-1}|u_{1:t-1})$ by p when there is rain and by $1 - p$ when there is no rain. That leads us to a system of 2 equations :

$$p = \alpha 0.9 * 0.7p + 0.3 * (1 - p)$$

$$1 - p = \alpha 0.2 * 0.3p + 0.7 * (1 - p)$$

$Solving this system, we find that p \approx 0.8933$

b. To compute all those probabilities, the easier is to find a recursive relationship between $P(R_{2+k}|U_1, U_2)$ and $P(R_{2+k-1}|U_1, U_2)$ Using Bayes rules we know that :

$$P(R_{2+k}|U_1, U_2) = \sum_{R_{2+k-1}} P(R_{2+k}|R_{2+k-1})P(R_{2+k-1}|U_1, U_2)$$

Hence we have :

$$P(R_{2+k}|U_1, U_2) = 0.7P(r_{2+k-1}|U_1, U_2) + 0.3(1 - P(r_{2+k-1}|U_1, U_2))$$

$$P(R_{2+k}|U_1, U_2) = 0.4P(R_{2+k-1}|U_1, U_2) + 0.3$$

When this relation converges we have : $P(R_{2+k}|U_1, U_2) = P(R_{2+k-1}|U_1, U_2)$, hence we have to solve :

$$P(R_{2+k}|U_1, U_2) = 0.4P(R_{2+k}|U_1, U_2) + 0.3$$

The solution is trivial : $\lim_{k \to +\infty} P(R_{2+k}|U_1, U_2) = 0.5$ Also, knowing the convergence point we can now subtract it to each terms and we get :

$$P(R_{2+k}|U_1, U_2) - 0.5 = 0.4P(R_{2+k-1}|U_1, U_2) - 0.2 = 2/5[P(R_{2+k-1}|U_1, U_2) - 0.5]$$

Rewriting $W(R_{2+k}|U_1, U_2) = P(R_{2+k}|U_1, U_2) - 0.5$ we have :

$$W(R_{2+k}|U_1, U_2) = 2/5W(R_{2+k-1}|U_1, U_2)$$

This is a geometric serie so :

$$W(R_{2+k}|U_1, U_2) = (2/5)^k W(R_2|U_1, U_2)$$

Replacing W by P we finally get :

$$P(R_{2+k}|U_1, U_2) = (2/5)^k (P(R_{2+k-1}|U_1, U_2) - 0.5) + 0.5$$

15.3 This exercice develops a space-efficient variant of the forward-backward algorithm described in Figure 15.4 (page 576). We wish to compute $P(X_k|e_{1:t})$ pour $k = 1, ..., h$. This will be done with a divide-and-conquer approach.

a. Suppose, for simplicity, that t is odd, and let the halfway point be $h = (t + 1)/2$. Show that $P(X_k|e_{1:t})$ can be computed for $k = 1, ..., h$ given just the initial forward message $f_{1:0}$, and the backward message $b_{h+1:t}$, and the evidence $e_{1:h}$.

b. Show a similar result for the second half of the sequence

c. 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.

d. 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?

Answer

a. As we want to develop a variant of the forward-backward algorithm, we already now that we can compute $P(X_k|e_{1:t})$ as [15.8] :

$$P(X_k|e_{1:t}) = \alpha f_{1:k} b_{k+1:t}$$

Also we know that [15.5] :

$$f_{1:k} = \alpha FORWARD(f_{1:k-1}, e_k)$$

Using this relation recursively, we can compute $f_{1:k}$ knowing only $f_{1:0}$ and $e_{1:k}$, Indeed : $f_{1:1} = \alpha FORWARD(f_{1:0}, e_1)$, so we can compute $f_{1:1}$ from $f_{1:0}$ and $e_{1:1}$ and then, as with now know $f_{1:1}$, we can compute $f_{1:2}$ if we know $e_{1:2}$.
Hence we can compute $f_{1:k}$ knowing only $f_{1:0}$ and $e_{1:k}$ The same argument can be applied to the backward pass and we can deduce that : $b_{k+1:t}$ can be computed knowing $b_{h+1:t}$ and $e_{k+1:h}$
Hence $P(X_k|e_{1:t})$ can be computed from $b_{h+1:t}$, $e_{1:h}$ and $f_{1:0}$

b. We can apply the same reasoning on the upper half. The result is the same replacing lower bound 0 by h and upper bound h by t, so : $P(X_k|e_{1:t})$ can be computed from $b_{t+1:t}$, $e_{h+1:t}$ and $f_{1:h}$

c. We can implement it using the merge sort algorithm and replacing the function call with our function. The base case is the same : a sequence of length 1 or 2.

d. At each recursion, the algorithm do $\Theta(t)$ operations (for example for the first level of recursion, $\Theta(h)$ operations for the first half and $\Theta(h)$ for the second half). Furthermore, there are $\Theta(log_2 t)$ so the algorithm takes $\Theta(t log_2 t)$.

15.5 Equation (15.12) 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 (15.7).

Answer Equation 15.12 also work for l message. In the book, we can see that the message calculation is identical to that for filtering: $l_{1:t+1} = FORWARD(l_1, e_{t+1})$. Hence for the message calculation we also have : $l_{1:t+1} = \alpha O_{t+1} T^T l_{1:t}$ and using [15.7] we have $L_{1:t} = P(e_{1:t}) = \sum_i l_i$

15.6 Consider the vacuum worlds of Figure 4.18 (perfect sensing) and Figure 15.7 (noisy sensing). Suppose that the robot receives an observation sequence such that, with perfect sensing, there is exactly one possible location it could he in. Is this location necessarily the most probable location under noisy sensing for sufficiently small noise probability ϵ? Prove your claim or find a counterexample.

Answer We can suppose that, under deterministic sensing we reach a unique possible location l. Hence, under deterministic sensing we have $P(X_t = l|e_{1:t}) = 1$ (the position l at step t is the only position possible after each observation we made at each time step t).

Is this location the most likely location under noisy sensing ?

To answer this question let d be the outdegree of the neighborhood graph (that is to say : the number of other possible states we can reach from the current state). Hence their is a maximum of d^t different states in which we can end up after t steps. Fixing ϵ smaller than $1/d^t$ allow this location to be the same under noise. However, ϵ depends on the length of the path t, that is to say if we fixed ϵ we could always find a path (as far as we can go) that is the only possible location under deterministic sensing but which is not under noisy sensing.

15.8 Consider a version of the vacuum robot (page 582) 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?

Answer I didn't implement it. Yet, it seems natural to think that the predictions of the most likely path is more accurate, because, instead of having at max d^t possible paths with t being the number of time step and d being the outdegree of the neighborhood graph, we only have to deal with a small number of possible headings now. Furthermore, the exact time at which the agent detects a collision with a wall helps to eliminate many states.

15.10 Consider a version of the vacuum robot (page 582) 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?

Answer I didn't implement it. Yet, it seems natural to think that the predictions of the most likely path is more accurate, because, instead of having at max d^t possible paths with t being the number of time step and d being the outdegree of the neighborhood graph, we only have to deal with a small number of possible headings now. Furthermore, the exact time at which the agent detects a collision with a wall helps to eliminate many states.

15.11 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 15.21.

a. Suppose that the discrete state Si has k possible values and that the prior continuous state estimate $P(X_0)$ is a multivariate Gaussian distribution. Show that the prediction $P(X_1)$ is a **mixture of Gaussians** — that is, a weighted sum of Gaussians such that the weights sum to 1.

b. Show that if the current continuous state estimate $P(X_t|e_{1:t})$ is a mixture of m Gaussians, then in the general case the updated state estimate $P(X_{t+1}|e_{1:t+1})$ will be a mixture of km Gaussians.

c. What aspect of the temporal process do the weights in the Gaussian mixture represent?

Answer

a. Using Bayes rule we can commute :

$$P(X_1) = \sum_{i=1}^{k} P(S_0 = i) \int_{X_0} P(X_0)P(X_1|X_0, S_0 = i)dX_0$$

Also, according to the properties of the Kalman filter [15.4.1], we know that the integral is a Gaussian. Hence the prediction distribution is a sum of k Gaussians weighted by $P(S_0)$ (As $P(S_0)$ is a probability, that ensures that $\sum_{i=1}^{k} P(S_0 = i) = 1$)

b. Applying the equation [15.18] we get :

$$P(X_{t+1}, S_{t+1}|e_{1:t+1})$$

$$= \alpha P(e_{t+1}|X_{t+1}, S_{t+1})P(X_{t+1}, S_{t+1}|e_{1:t})$$

or we know [15.17] :

$$P(X_{t+1}, S_{t+1}|e_{1:t}) = \int_{x_t} P(X_{t+1}, S_{t+1}|x_t, s_t)P(x_t, s_t|e_{1:t})dx_t$$

And so :

$$P(X_{t+1}, S_{t+1}|e_{1:t+1})$$

$$= \alpha P(e_{t+1}|X_{t+1}, S_{t+1}) \sum_{s_t=1}^{k} \int_{x_t} P(X_{t+1}, S_{t+1}|x_t, s_t) P(x_t, s_t|e_{1:t}) dx_t$$

As X_{t+1} and S_{t+1} are independents given X_t and S_t we can rewrite it :

$$P(X_{t+1}, S_{t+1}|e_{1:t+1})$$

$$= \alpha P(e_{t+1}|X_{t+1}, S_{t+1}) \sum_{s_t=1}^{k} P(S_{t+1}|s_t) P(s_t|e_{1:t}) \int_{x_t} P(X_{t+1}|x_t, s_t) P(x_t|e_{1:t}) dx_t$$

Using the hypotheses of the question and the properties of integral and sum of Gaussians, we can conclude that we have km Gaussians.