Probabilistic Reasoning over Time

Introduction

- In previous chapter we studied causal reasoning in static environments
- We will now look at changing worlds
  - uncertainty not only due to partial and noisy percepts of the world
  - but also to uncertainty about how the environment changes over time
- Agent will be able to obtain only a probabilistic assessment of the current situation
- Changing world is modeled using a random variable <u>for each aspect</u> of the world state at <u>each point in time</u> - relations among these variables describe how the state evolves (Section 3.1)
- Basic inference tasks and general structure of inference algorithms for temporal models (Section 3.2)
- Three specific kinds of models (Section 3.3 - 3.5):
  - Hidden Markov Models
  - Kalman filters
  - and dynamic Bayesian networks (which include hidden Markov models and Kalman filters as special cases)

3.1 Time and Uncertainty

- Example: treating a diabetic patient
  - we have evidence such as recent insulin doses, food intake, blood sugar measurements, and other physical signs
  - task: assess the current state of the patient, including the actual blood sugar level and insulin level
  - given this information, the doctor (or patient) makes a decision about the patient's food intake and insulin dose
  - unlike the case of earthquake-burglary-alarm, here the dynamic aspects of the problem are essential: blood sugar levels and measurements thereof can change rapidly over time
- To assess the current state from the history of evidence and to predict the outcomes of treatment actions, we must model these changes
- Same considerations arise in many other contexts, ranging from tracking the economic activity of a nation, given approximate and partial statistics, to understanding a sequence of spoken words, given noisy and ambiguous acoustic measurements.

How can dynamic situations like these be modeled?

States and observations

- Process of change can be viewed as a series of snapshots, each of which describes the state of the world at a particular time
- Each snapshot, or time slice, contains a set of random variables, some of which are observable and some of which are not
- For simplicity, we will assume that the same subset of variables is observable in each slice (although not strictly necessary)
- Notation

  $\mathbf{X}_t$ denotes the set of unobservable state variables at time t

  $\mathbf{E}_t$ denotes the set of observable evidence variables

  observation at time t is $\mathbf{E}_t = \mathbf{e}_t$ for some set of values $\mathbf{e}_t$

Another Example

- Suppose you are the security guard at some secret underground installation. You want to know whether it's raining today, but your only access to the outside world occurs each morning when you see the director coming in with, or without, an umbrella
- For each day t

  the set $\mathbf{E}_t$ contains a single evidence variable U_t (whether the umbrella appears)

  and set $\mathbf{X}_t$ contains a single state variable R_t (whether it is raining)

Conventions

- Interval between time slices also depends on the problem
- We will generally assume a fixed, finite interval, i.e. times can be labeled by integers
- We will assume that the state sequence starts at $t = 0$, e.g. $R_0, R_1, R_2, \ldots$
- We will assume that evidence starts arriving at $t = 1$ rather than $t = 0$ e.g. $U_1, U_2, \ldots$
- We will use the notation $\mathrm{a{:}b}$ to denote the sequence of integers from a to b (inclusive), and notation $\mathbf{X}_{a:b}$ to denote the corresponding set of variables from $\mathbf{X}_a$ to $\mathbf{X}_b$
  i.e. $\mathbf{U}_{1:3}$ corresponds to the variables U_1, U_2, U_3

Stationary processes and the Markov assumption

- Next step (after fixing state and evidence variables):
  specify dependencies among variables
- We could follow the procedure laid down in previous chapter placing the variables in some order and asking questions about conditional independence of predecessors, given some set of parents
- However: the set of variables is unbounded, because it includes the state and evidence variables for every time slice
- This creates two problems:

Stationary processes and the Markov assumption

- Next step (after fixing state and evidence variables):
  specify dependencies among variables
- We could follow the procedure laid down in previous chapter placing the variables in some order and asking questions about conditional independence of predecessors, given some set of parents
- However: the set of variables is unbounded, because it includes the state and evidence variables for every time slice
- This creates two problems:
  - first, we might have to specify an unbounded number of conditional probability tables, one for each variable in each slice
  - second, each one might involve an unbounded number of parents

- First problem is solved by assuming that changes in the world state are caused by a stationary process (process of change that is governed by laws that do not themselves change over time.

  don't confuse stationary with static: in a static process, the state itself does not change
- Umbrella world: conditional probability that the umbrella appears

  $$\mathbf{P}(U_t \mid Parents(U_t))$$

  is the same for all t
- Given the assumption of stationarity, we need specify conditional distributions only for the variables within a "representative" time slice.
- Second problem, handling the potentially infinite number of parents, is solved by making what is called a Markov assumption:

Current state depends on only a finite history of previous states

---

Processes satisfying this assumption were first studied in depth by the Russian statistician Andrei Markov and are called Markov processes or Markov chains.

- First-order Markov process: current state depends only on the previous state and not on any earlier states

  $$\mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{t-1}) \qquad (3.1)$$

  in a first-order Markov process, laws describing how the state evolves over time are contained entirely within the conditional distribution $\mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{t-1})$, which we call the transition model for first-order processes

- Transition model for a second-order Markov process is the conditional distribution $\mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{t-2}, \mathbf{X}_{t-1})$

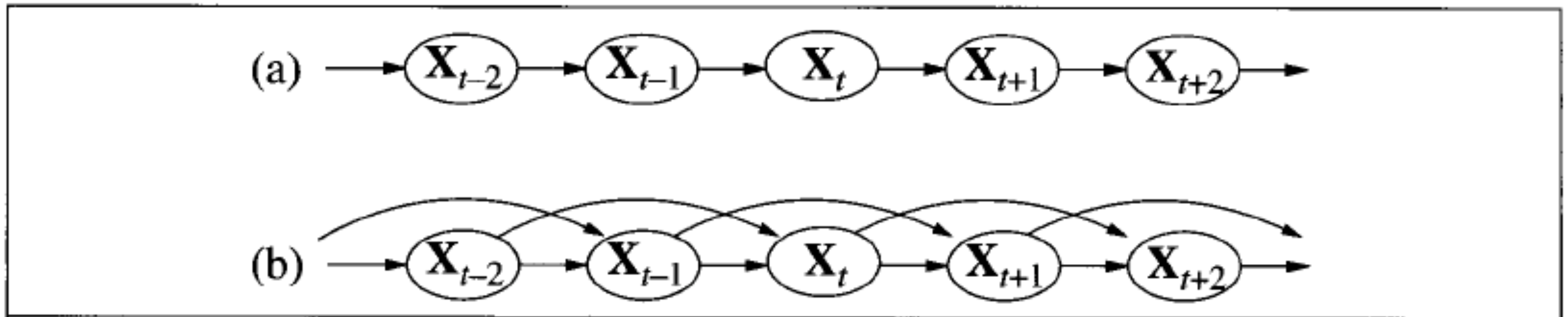


Figure 15.1 (a) Bayesian network structure corresponding to a first-order Markov process with state defined by the variables $\mathbf{X}_t$. (b) A second-order Markov process.

- In addition to restricting the parents of the state variables $\mathbf{X}_t$, we must restrict the parents of the evidence variables $\mathbf{E}_t$
- Typically, we assume that the evidence variables at time t depend only on the current state:

$$\boxed{\mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_{0:t-1}, \mathbf{E}_{0:t-1}) = \mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_t)} \qquad (3.2)$$

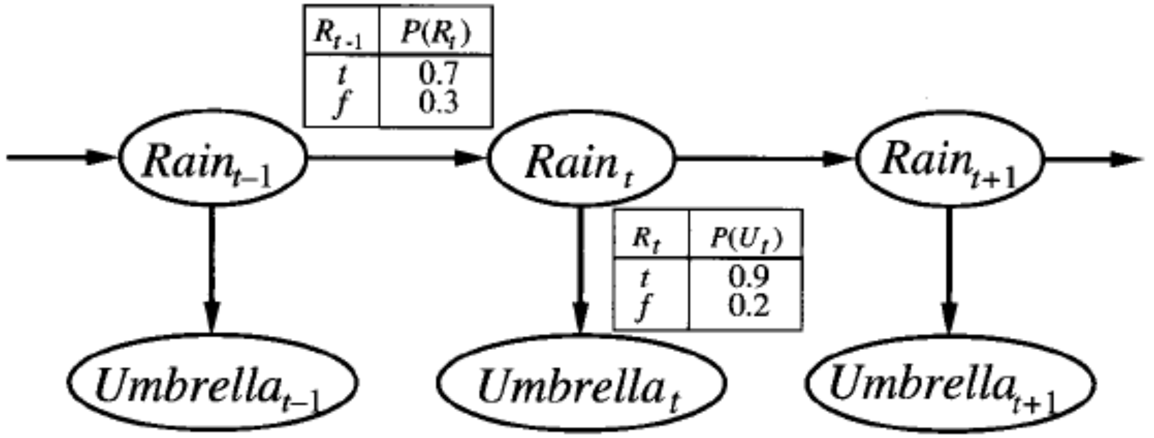
- Conditional distribution $\mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_t)$ is called the sensor model (or sometimes the observation model), because it describes how the "sensors" (evidence variables) are affected by the actual state of the world
- Direction of the dependence (the "arrow") goes from state to sensor values because the state of the world *causes* the sensors to take on particular values

  in umbrella world: the rain *causes* the umbrella to appear
- Inference process goes in the other direction!!
- In addition to transition model and sensor model, we need to specify a

  prior probability $\boxed{\mathbf{P}(\mathbf{X}_0)}$ over the states at time 0.

- These three distributions, combined with the conditional independence assertions in Equations 3.1 and 3.2, give us a specification of the complete joint distribution over all the variables; for any finite t, we have

$$\mathbf{P}(\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_t, \mathbf{E}_1, \ldots, \mathbf{E}_t) = \mathbf{P}(\mathbf{X}_0) \prod_{i=1}^{t} \mathbf{P}(\mathbf{X}_i \mid \mathbf{X}_{i-1}) \mathbf{P}(\mathbf{E}_i \mid \mathbf{X}_i)$$



R_{t-1}	$P(R_t)$
t	0.7
f	0.3

R_t	$P(U_t)$
t	0.9
f	0.2

Figure 15.2 Bayesian network structure and conditional distributions describing the umbrella world. The transition model is $P(Rain_t | Rain_{t-1})$ and the sensor model is $P(Umbrella_t | Rain_t)$.

structure in figure assumes a first-order Markov process, because probability of rain is assumed to depend only on whether it rained the previous day; whether such an assumption is reasonable depends on the domain itself

- First-order Markov assumption says that the state variables contain all the information needed to characterize the probability distribution for the next time slice
- Sometimes assumption is only approximate, as in the case of predicting rain only on the basis of whether it rained the previous day
- There are two possible fixes if the approximation proves too inaccurate:

1. Increasing the order of the Markov process model; e.g. make a second-order model by adding $Rain_{t-2}$ as parent of $Rain_t$, which might give slightly more accurate predictions

   example: in Palo Alto it very rarely rains more than two days in a row

2. Increasing the set of state variables; e.g. add $Season_t$ to allow us to incorporate historical records of rainy seasons, or add $Temperature_t$, $Humidity_t$ and $Pressure_t$ to allow us to use a physical model of rainy conditions

- Adding state variables might improve the system's predictive power but also increases the prediction requirements: we now have to predict the new variables as well

Hence we are looking for a "self-sufficient" set of variables, which really means that we have to understand the "physics" of the process being modeled

3.2 Inference in Temporal Models

Basic Inference tasks

◊ Filtering or monitoring: $\mathbf{P}(\mathbf{X}_t \mid \mathbf{e}_{1:t})$

the task of computing the posterior distribution over the current state, given all evidence to date, assuming that evidence arrives in a "continuous" stream beginning at $t = 1$

umbrella example:

compute the probability of rain today, given all the observations of the umbrella carrier made so far.

◊ Prediction: $\mathbf{P}(\mathbf{X}_{t+k} \mid \mathbf{e}_{1:t})$ for some $k > 0$

the task of computing the posterior distribution over the future state, given all evidence to date

umbrella example:

compute the probability of rain three days from now, given all the observations of the umbrella-carrier made so far.

Prediction is useful for evaluating possible courses of action.

Basic Inference tasks

◊ Smoothing or hindsight: $\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t})$ for some k such that $0 \leq k < t$

task of computing the posterior distribution over a past state, given all evidence up to the present

umbrella example:
computing the probability that it rained last Wednesday, given all the observations of the umbrella carrier made up to today.

hindsight provides a better estimate of the state than was available at the time, because it incorporates more evidence.

◊ Most likely explanation: $\text{argmax}_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t} \mid \mathbf{e}_{1:t})$

given a sequence of observations, we might wish to find the sequence of states that is most likely to have generated those observations

umbrella example:
if umbrella appears on each of the first three days and is absent on the fourth, then most likely explanation is that it rained on the first three days and did not rain on the fourth.

useful in many applications, including speech recognition (aim is to find the most likely sequence of words, given a series of sounds) and the reconstruction of bit strings transmitted over a noisy channel

Filtering and prediction

- Given the result of filtering up to time t, one can compute the result for $t+1$ from the new evidence $\mathbf{e}_{t+1}$

  $$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}) = f(\mathbf{e}_{t+1}, \mathbf{P}(\mathbf{X}_t \mid \mathbf{e}_{1:t}))$$

  process is often called recursive estimation

- calculation composed of two parts:

  first, the current state distribution is projected forward from t to $t+1$;

  then it is updated using the new evidence $\mathbf{e}_{t+1}$.

  $$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}) =$$

  $$= \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{t+1}, \mathbf{e}_{1:t}) \quad \text{(dividing up the evidence)}$$

  $$= \alpha\, \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}, \mathbf{e}_{1:t})\, \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}) \quad \text{(using Bayes' rule)}$$

  $$= \alpha\, \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})\, \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}) \quad \text{(by the Markov property of evidence)}$$

  α is a normalizing constant

  second term, $\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t})$ represents a one-step prediction of the next state

  notice $\mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})$ is obtainable directly form the sensor model

- How do we obtain the one-step prediction for the next state by conditioning on the current state $\mathbf{X}_t$?

$$\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}) =$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t, \mathbf{e}_{1:t}) P(\mathbf{x}_t \mid \mathbf{e}_{1:t})$$

$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) P(\mathbf{x}_t \mid \mathbf{e}_{1:t}) \qquad \text{(using Markov property) (3.3)}$$

first factor is simply the transition model and second is the current state distribution.

And with this we have the desired recursive formulation!!!!!!!

Remark

- filtered estimate $\mathbf{P}(\mathbf{x}_t \mid \mathbf{e}_{1:t})$ can be considered as a "message" $\mathbf{f}_{1:t}$ that is propagated forward along the sequence, modified by each transition and updated by each new observation

$$\mathbf{f}_{1:t+1} = \alpha \, \text{FORWARD}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1})$$

where FORWARD implements the update described in Equation 3.3.

Example: Filtering process for two steps in the umbrella example

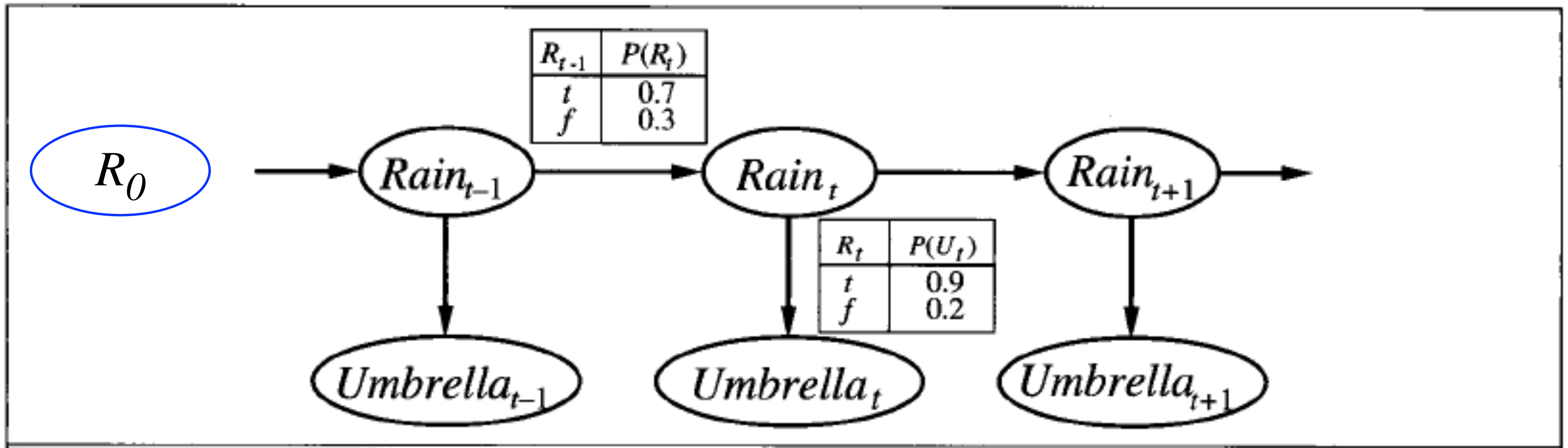


Figure 15.2 Bayesian network structure and conditional distributions describing the umbrella world. The transition model is $P(Rain_t|Rain_{t-1})$ and the sensor model is $P(Umbrella_t|Rain_t)$.

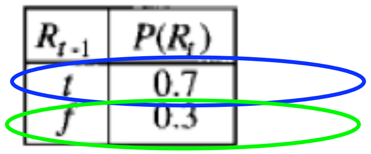
- Assume our security guard has some prior belief about whether it rained on day 0, just before the observation sequence begins: $\mathbf{P}(R_0) = \langle 0.5, 0.5 \rangle$

Example: Filtering process for two steps in the umbrella example

- Now process the two observations as follows: .
  - on day 1, the umbrella appears, so $U_1 = true$. The prediction from $t = 0$ to $t = 1$ is

$$\mathbf{P}(R_1) = \sum_{r_0} \mathbf{P}(R_1 \mid r_0)\, P(r_0) =$$

$$= \langle 0.7, 0.3\rangle \times 0.5 + \langle 0.3, 0.7\rangle \times 0.5 = \langle 0.5, 0.5\rangle$$

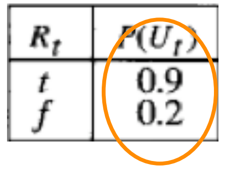


R_{t-1}	$P(R_t)$
t	0.7
f	0.3

and updating it with the evidence for $t = 1$ gives

$$\mathbf{P}(R_1 \mid u_1) = \alpha\, \mathbf{P}(u_1 \mid R_1)\, \mathbf{P}(R_1) = \alpha \langle 0.9, 0.2\rangle \langle 0.5, 0.5\rangle$$

$$= \alpha \langle 0.45, 0.1\rangle \approx \langle 0.818, 0.182\rangle$$



R_t	$P(U_t)$
t	0.9
f	0.2

  - on day 2, the umbrella appears, so $U_2 = true$. The prediction from $t = 1$ to $t = 2$ is

$$\mathbf{P}(R_2 \mid u_1) = \sum_{r_1} \mathbf{P}(R_2 \mid r_1)\, P(r_1 \mid u_1)$$

$$= \langle 0.7, 0.3\rangle \times 0.818 + \langle 0.3, 0.7\rangle \times 0.182 \approx \langle 0.627, 0.373\rangle$$

and updating it with the evidence for $t = 2$ gives

$$\mathbf{P}(R_2 \mid u_1, u_2) = \alpha\, \mathbf{P}(u_2 \mid R_2)\, \mathbf{P}(R_2 \mid u_1) = \alpha \langle 0.9, 0.1\rangle \langle 0.627, 0.373\rangle$$

$$= \alpha \langle 0.565, 0.075\rangle \approx \langle 0.883, 0.117\rangle$$

- Intuitively, the probability of rain increases from day 1 to day 2 because rain persists

Prediction

- Task of prediction can be seen simply as filtering without the addition of new evidence
- In fact, the filtering process already incorporates a one-step prediction, and it is easy to derive the following recursive computation for predicting the state at $t + k + 1$ from a prediction for $t + k$:

  $$\mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{e}_{1:t}) = \sum_{\mathbf{x}_{t+k}} \mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{x}_{t+k})\, P(\mathbf{x}_{t+k} \mid \mathbf{e}_{1:t}) \qquad (3.4)$$

  this computation involves only the transition model and not the sensor model
- Q: What happens if we try to predict further and further into the future?

Prediction

- Task of prediction can be seen simply as filtering without the addition of new evidence
- In fact, the filtering process already incorporates a one-step prediction, and it is easy to derive the following recursive computation for predicting the state at $t + k + 1$ from a prediction for $t + k$:

  $$\mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{e}_{1:t}) = \sum_{\mathbf{x}_{t+k}} \mathbf{P}(\mathbf{X}_{t+k+1} \mid \mathbf{x}_{t+k})\, P(\mathbf{x}_{t+k} \mid \mathbf{e}_{1:t}) \qquad (3.4)$$

  this computation involves only the transition model and not the sensor model

- Q: What happens if we try to predict further and further into the future?

  A: Predicted distribution for rain converges to a fixed point $\langle 0.5, 0.5\rangle$

Smoothing:

- Computing the distribution over past states given evidence up to the present

$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t})$ for $1 \leq k < t$

$$\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t}) = \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{k+1:t}, \mathbf{e}_{1:k}) \qquad \text{(dividing up the evidence)}$$

$$= \alpha\, \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k})\, \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k, \mathbf{e}_{1:k}) \qquad \text{(using Bayes' rule)}$$

$$= \alpha\, \mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:k})\, \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k) \qquad \text{(using conditional independence)}$$

$$= \alpha\, \mathbf{f}_{1:k}\, \mathbf{b}_{\mathrm{k}+1:t} \qquad (3.5)$$

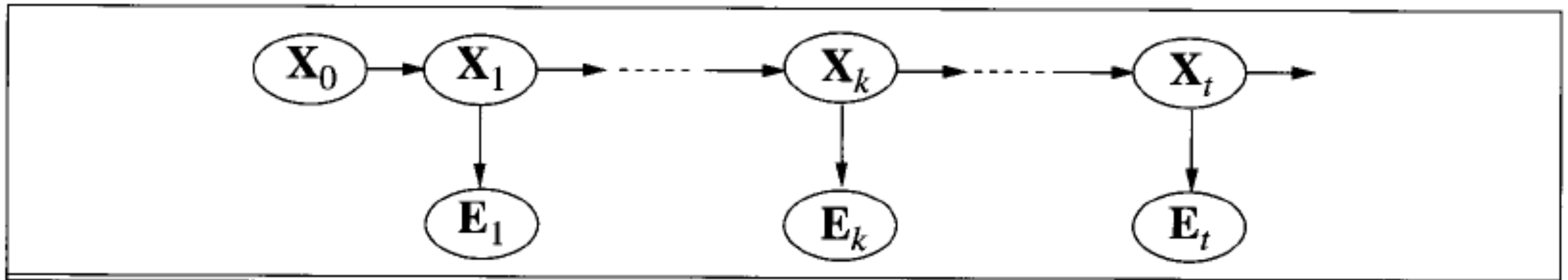


Figure 15.3 Smoothing computes $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$, the posterior distribution of the state at some past time k given a complete sequence of observations from 1 to t.

where we have defined a "backward" message $\mathbf{b}_{\mathrm{k}+1:t} = \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k)$ analogous to the forward message $\mathbf{f}_{1:k}$. The forward message $\mathbf{f}_{1:k}$ can be computed by filtering forward from 1 to k, as given by Equation 3.3.

- It turns out that the backward message $\mathbf{b}_{k+1:t}$ can be computed by a recursive process that runs backwards from t:

$$\begin{aligned}
\mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k) &= \sum_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t} \mid \mathbf{X}_k, \mathbf{x}_{k+1})\, \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k) && \text{(conditioning on } \mathbf{X}_{k+1}\text{)} \\
&= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1:t} \mid \mathbf{x}_{k+1})\, \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k) && \text{(by conditional independence)} \\
&= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}, \mathbf{e}_{k+2:t} \mid \mathbf{x}_{k+1})\, \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k) \\
&= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1} \mid \mathbf{x}_{k+1})\, P(\mathbf{e}_{k+2:t} \mid \mathbf{x}_{k+1})\, \mathbf{P}(\mathbf{x}_{k+1} \mid \mathbf{X}_k) && (3.6)
\end{aligned}$$

  where the last step follows by the conditional independence of $\mathbf{e}_{k+1}$ and $\mathbf{e}_{k+2:t}$, given $\mathbf{X}_{k+1}$.

- The first and third factor is obtained directly from the model, and the second is the "recursive call”

- Using the message notation, we have

  $$\mathbf{b}_{k+1:t} = \text{BACKWARD}(\mathbf{b}_{k+2:t}, \mathbf{e}_{k+1:t})$$

  where BACKWARD implements the update described in Equation (3.6).

Example: Umbrella scenario

- Computing the smoothed estimate for the probability of rain at $t = 1$, given the umbrella observations on days 1 and 2
- From Equation 3.5, this is given by

$$\mathbf{P}(R_1 \mid u_1, u_2) = \alpha\, \mathbf{P}(R_1 \mid u_1)\, \mathbf{P}(u_2 \mid R_1) \qquad (3.7)$$

  where the first term is known to be $\langle\, 0.818, 0.182 \,\rangle$ from the forward filtering process described earlier.

- Second term can be computed by applying the backward recursion in Equation 3.6:

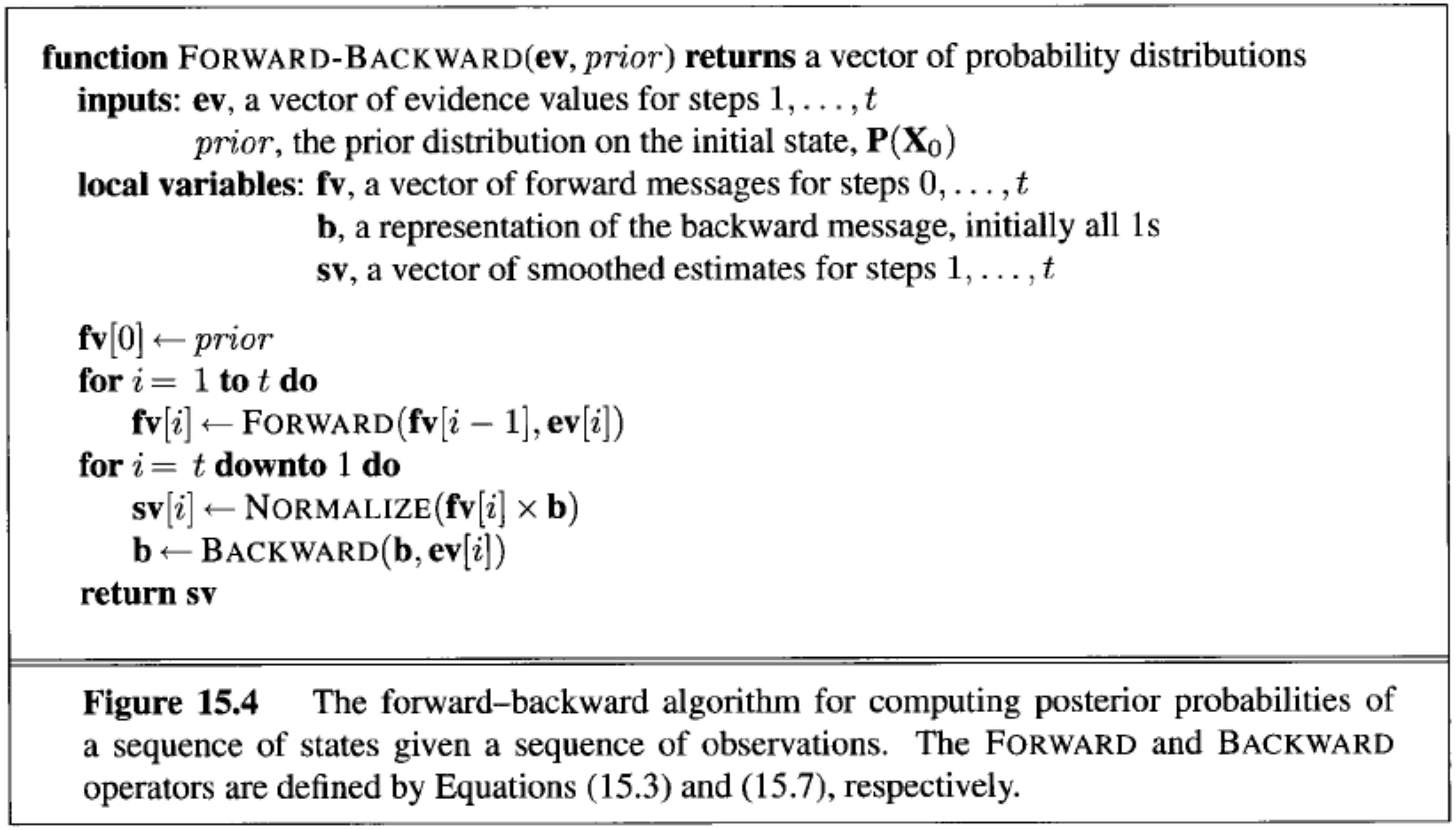
$$\mathbf{P}(u_2 \mid R_1) = \sum_{\mathbf{r}_2} P(u_2 \mid r_2)\, P(\ \mid r_2)\, \mathbf{P}(r_2 \mid R_1)$$
$$= (0.9 \times 1 \times \langle 0.7, 0.3 \rangle) + (0.2 \times 1 \times \langle 0.3, 0.7 \rangle) = \langle\, 0.69, 0.41 \,\rangle$$

- plugging this into Equation 3.7, we find that the smoothed estimate for rain on day 1 is

$$\mathbf{P}(R_1 \mid u_1, u_2) = \alpha\, \langle 0.818, 0.182 \rangle \times \langle 0.69, 0.41 \rangle \approx \langle\, 0.883, 0.117 \,\rangle$$

- Smoothed estimate is *higher* than the filtered estimate (0.818) in this case. This is because the umbrella on day 2 makes it more likely to have rained on day 2; in turn, because rain tends to persist, that makes it more likely to have rained on day 1.

Probabilistic Reasoning over Time/Inference in Temporal Models



```
function FORWARD-BACKWARD(ev, $prior$) returns a vector of probability distributions
  inputs: ev, a vector of evidence values for steps $1, \ldots, t$
          $prior$, the prior distribution on the initial state, $\mathbf{P}(\mathbf{X}_0)$
  local variables: fv, a vector of forward messages for steps $0, \ldots, t$
                   b, a representation of the backward message, initially all 1s
                   sv, a vector of smoothed estimates for steps $1, \ldots, t$

  fv$[0] \leftarrow prior$
  for $i = 1$ to $t$ do
      fv$[i] \leftarrow$ FORWARD(fv$[i-1]$, ev$[i]$)
  for $i = t$ downto 1 do
      sv$[i] \leftarrow$ NORMALIZE(fv$[i] \times$ b)
      b $\leftarrow$ BACKWARD(b, ev$[i]$)
  return sv
```

Figure 15.4 The forward–backward algorithm for computing posterior probabilities of a sequence of states given a sequence of observations. The FORWARD and BACKWARD operators are defined by Equations (15.3) and (15.7), respectively.

- Forward-backward algorithm forms the backbone of computational methods employed in many applications dealing with sequences of noisy observations, ranging from speech recognition to radar tracking of aircraft
- It has two practical drawbacks:
  1. space complexity can be too high for applications where the state space is large and the sequences are long. It uses $O(|\mathbf{f}|t)$ space where $|\mathbf{f}|$ is the size of the representation of the forward message
  2. needs to be modified to work in an online setting where smoothed estimates must be computed for earlier time slices as new observations are continuously added to the end of the sequence.

Finding the most likely sequence

- Suppose that [*true, true, false, true, true*] is the umbrella sequence for the security guard's first five days on the job
- What is the weather sequence most likely to explain this?
- Does the absence of the umbrella on day 3 mean that it wasn't raining, or did the director forget to bring it?
- If it didn't rain on day 3, perhaps (because weather tends to persist) it didn't rain on day 4 either, but the director brought the umbrella just in case.
- In all, there are 25 possible weather sequences we could pick.
- Is there a way to find the most likely one, rather than enumerating all of them?
- The easiest way to think about the problem is to view each sequence as a path through a graph whose nodes are the possible states at each time step.

Finding the most likely sequence

(a)

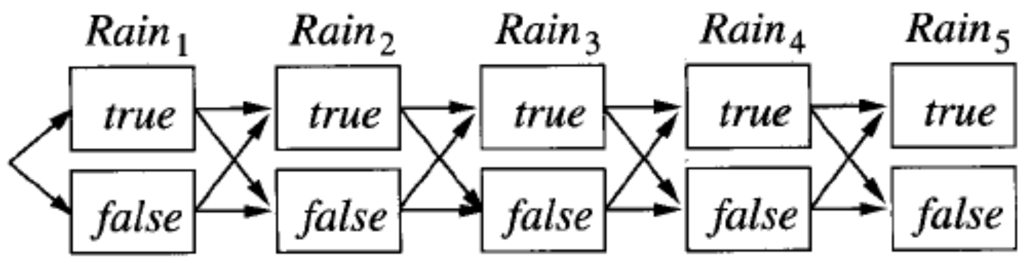


Figure 15.5 (a) Possible state sequences for $Rain_t$ can be viewed as paths through a graph of the possible states at each time step. (States are shown as square nodes to avoid confusion with nodes in a Bayesian network.)

Finding the most likely sequence

- Consider task of finding the most likely path through this graph, where the likelihood of any path is the product of the transition probabilities along the path and the probabilities of the given observations at each state
- Let us focus in particular on paths that reach the state $Rain_5 = true$
- Because of Markov property, it follows that the most likely path to the state $Rain_5 = true$ consists of
  - most likely path to some state at time 4
  - followed by a transition to $Rain_5 = true$
  - and the state at time 4 that will become part of the path to $Rain_5 = true$ is whichever maximizes the likelihood of that path.
- There is a recursive relationship between most likely paths to each state $\mathbf{x}_{t+1}$ and most likely paths to each state $\mathbf{x}_t$.

We can write this relationship as an equation connecting the probabilities of the paths:

Finding the most likely sequence

- We can write this relationship as an equation connecting the probabilities of the paths

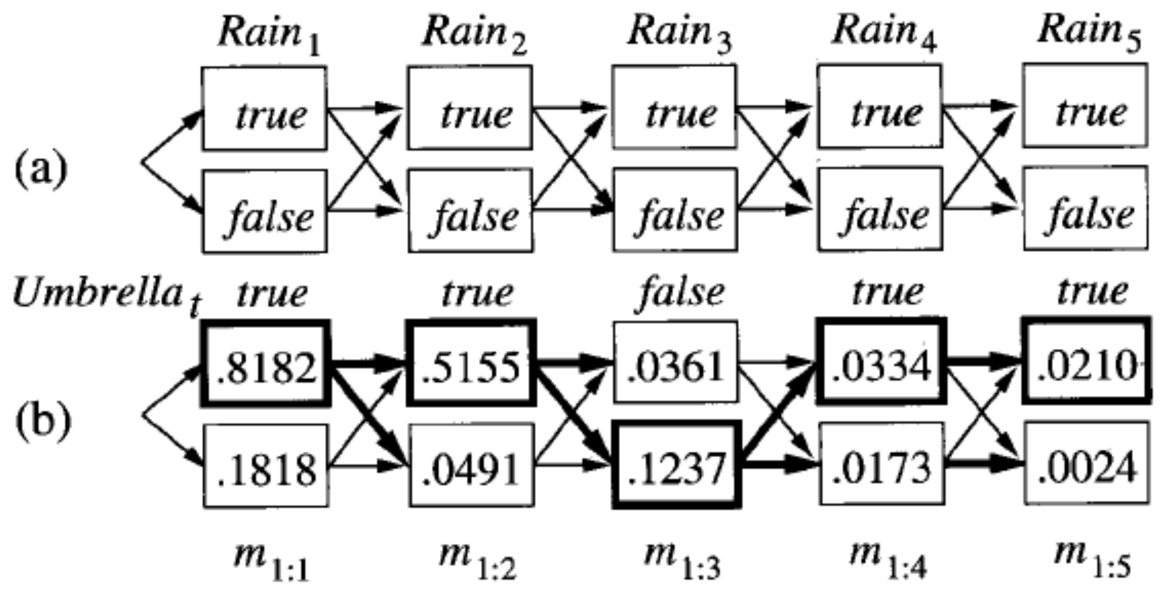
$$\max_{\mathbf{x}_1,\ldots,\mathbf{x}_t} \mathbf{P}(\mathbf{x}_1,\ldots,\mathbf{x}_t,\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$
$$= \alpha \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \left(\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x_t}) \max_{\mathbf{x}_1,\ldots,\mathbf{x}_{t-1}} \mathbf{P}(\mathbf{x}_1,\ldots,\mathbf{x}_{t-1},\mathbf{x}_t \mid \mathbf{e}_{1:t}) \right) \tag{3.8}$$

- Equation 3.8 is identical to the filtering Equation 3.3 except that
  1. The forward message $\mathbf{f}_{1:t} = \mathbf{P}(\mathbf{X}_t \mid \mathbf{e}_{1:t})$ is replaced by the message

     $$\mathbf{m}_{1:t} = \max_{\mathbf{x}_1,\ldots,\mathbf{x}_{t-1}} \mathbf{P}(\mathbf{x}_1,\ldots,\mathbf{x}_{t-1},\mathbf{X}_t \mid \mathbf{e}_{1:t})$$

     that is, the probabilities of the most likely path to each state $\mathbf{X}_t$
  2. and the summation over $\mathbf{X}_t$ in Equation 3.3 is replaced by the maximization in 3.8.

Finding the most likely sequence



$Rain_1$ $Rain_2$ $Rain_3$ $Rain_4$ $Rain_5$
(a)
true
false
$Umbrella_t$ true true false true true
(b)
.8182 .5155 .0361 .0334 .0210
.1818 .0491 .1237 .0173 .0024
$m_{1:1}$ $m_{1:2}$ $m_{1:3}$ $m_{1:4}$ $m_{1:5}$

Figure 15.5 (a) Possible state sequences for $Rain_t$ can be viewed as paths through a graph of the possible states at each time step. (States are shown as square nodes to avoid confusion with nodes in a Bayesian network.) (b) Operation of the Viterbi algorithm for the umbrella observation sequence $[true, true, false, true, true]$. For each time step t, we have shown the values of the message $\mathbf{m}_{1:t}$, which gives the probability of the best sequence reaching each state at time t. Also, for each state, the bold arrow leading into it indicates its best predecessor. Following the bold arrows back from the most likely state in $\mathbf{m}_{1:5}$ gives the most likely sequence.

Finding the most likely sequence

- The algorithm for computing the most likely sequence is similar to filtering:
  it runs forward along the sequence, computing the m message at each time step, using Equation 3.8.
- Progress of this computation is shown in Figure 15.5(b):
  at the end, it will have the probability for the most likely sequence reaching each of the final states. One can thus easily select the most likely sequence overall (the state outlined in bold)
- The algorithm we have just described is called the *Viterbi* algorithm, after its inventor.

3.5 Dynamic Bayesian Networks

- A dynamic Bayesian network, or DBN, is a Bayesian network that represents a temporal probability model
- Examples of DBNs:
  the umbrella network in Figure 15.2 and the Kalman filter network
- In general,
  - each slice of a DBN can have any number of state variables X_t and evidence variables E_t.
  - for simplicity, we will assume that the variables and their links are exactly replicated from slice to slice and the DBN represents a first-order Markov process.

HMM - Dynamic Bayesian Network (DBN)

Every hidden Markov model can be represented as a DBN
with a single state variable and a single evidence variable

Every discrete variable DBN can be represented as an HMM
We can combine all the state variables in the DBN into a single state variable whose values are all possible tuples of values of the individual state variables

If every HMM is a DBN and every DBN can be translated into an HMM, what's the difference???

HMM - DBN

- Difference:

  By decomposing the state of a complex system into its constituent variables, DBN is able to take advantage of sparseness in the temporal probability model.

- Example:

  DBN with 20 Boolean state variables, each of which has three parents in the preceding slice

  DBN transition model has 20 x 2^3 = 160 probabilities

  However: corresponding HMM has 2^{20} states and therefore 2^{40}, or roughly a trillion, probabilities in the transition matrix.

- Bad for at least three reasons:
  - first, HMM itself requires much more space
  - second, huge transition matrix makes HMM inference much more expensive
  - third, the problem of learning such a huge number of parameters makes the pure HMM model unsuitable for large problems.

The relationship between DBNs and HMMs is roughly analogous to relationship between ordinary Bayesian networks and full tabulated joint distributions.

Kalman Filter - DBN

Every Kalman filter model can be represented in a DBN with continuous variables and linear Gaussian conditional distributions

However**:** Not every DBN can be represented by a Kalman filter model**.**

Why?

Kalman Filter - DBN

Every Kalman filter model can be represented in a DBN with continuous variables and linear Gaussian conditional distributions

However: Not every DBN can be represented by a Kalman filter model.

Why?

- In a Kalman filter, the current state distribution is always a single multivariate Gaussian distribution-that is, a single "bump" in a particular location.
- DBNs can model arbitrary distributions. For many real-world applications, this flexibility is essential.
- Example: the current location of my keys.
  in my pocket, on the bedside table, on the kitchen counter, or dangling from the front door?

A single Gaussian bump that included all these places would have to allocate significant probability to the keys being in mid-air in the front hall!!

Constructing DBNs

- To construct a DBN, one must specify three kinds of information:
  - the prior distribution over the state variables $\mathbf{P(X_0)}$;
  - the transition model $\mathbf{P(X_{t+1} \mid X_t)}$;
  - and the sensor model $\mathbf{P(E_t \mid X_t)}$.

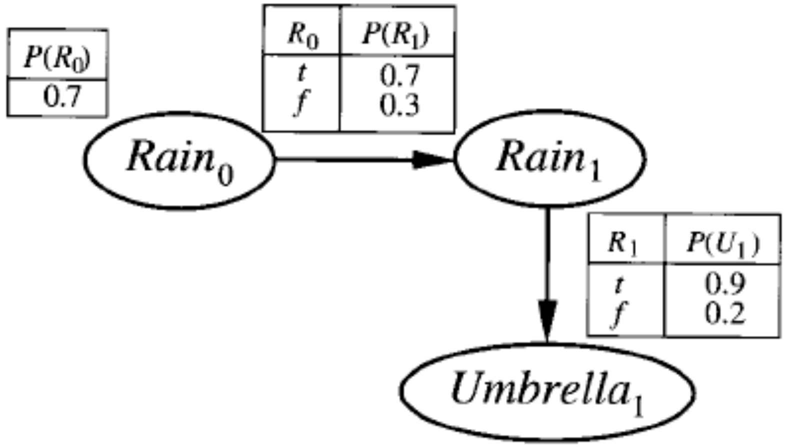
  AND

  one must also specify the topology of the connections between successive slices and between the state and evidence variables.

- Because the transition and sensor models are assumed to be stationary

  ➔ simply specify them for the first slice AND construct the complete (semi-infinite) DBN by copying the first slice.

- Example:



$P(R_0)$
0.7

R_0	$P(R_1)$
t	0.7
f	0.3

R_1	$P(U_1)$
t	0.9
f	0.2

Constructing DBNs

- More interesting example:

  Monitoring a battery-powered robot moving in the X - Y plane

  – State variables:

    $\mathrm{X}_t = (X_t, Y_t)$ for position and
    $\dot{\mathrm{X}}_t = (\dot{X}_t, \dot{Y}_t)$ for velocity.

  – We will assume some method of measuring position, e.g. GPS yielding measurements $\mathbf{Z}_t$.

  – Position at the next time step depends on the current position and velocity (as in the standard Kalman filter model) — velocity at the next step depends on the current velocity and the state of the battery

  – We add:

$Battery_t$	represents actual battery charge level, which has as parents the previous battery level and the velocity;
$Bmeter_t$	which measures the battery charge level.

Constructing DBNs

- More interesting example:

  Monitoring a battery-powered robot moving in the X - Y plane

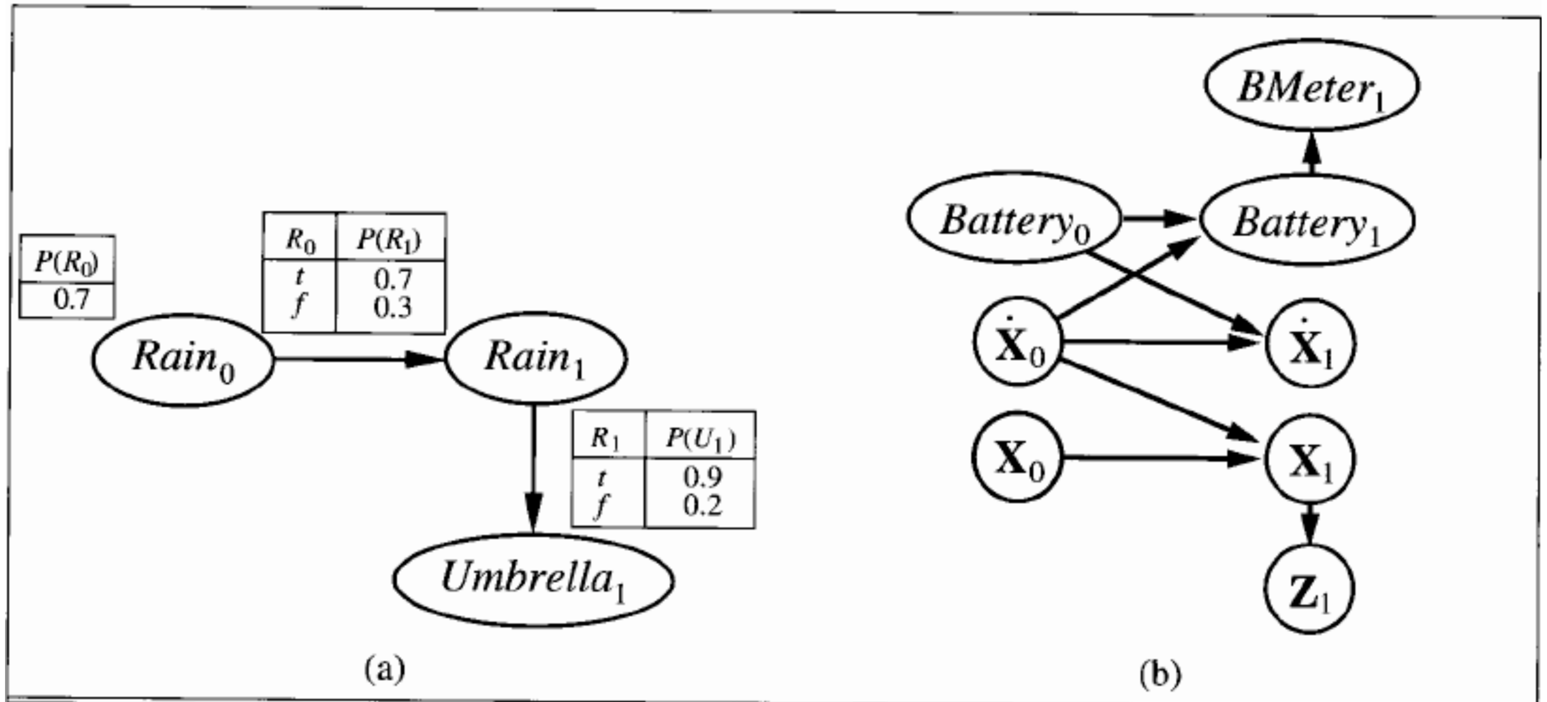


Figure 15.11 (a) Specification of the prior, transition model, and sensor model for the umbrella DBN. All subsequent slices are assumed to be copies of slice 1. (b) A simple DBN for robot motion in the X–Y plane.

Sensor - error models – transient failures – persistent failures

- Sensor model for $BMeter_t$:

  Suppose $Battery_t$ and $BMeter_t$ can take on discrete values 0 through 5

  If the meter is always accurate, then the CPT $\mathbf{P}(BMeter_t \mid Battery_t)$ should have probabilities of 1.0 "along the diagonal" and probabilities of 0.0 elsewhere.

- Remark

  For discrete variables, we can approximate a Gaussian using a distribution in which the probability of error drops off in the appropriate way

  Term Gaussian error model is used to cover both the continuous and discrete versions.

Sensor - error models – transient failures – persistent failures

- What if the sensor fails? (and real sensors DO fail!!!)

  When a sensor fails, it does not necessarily send a signal saying, "Oh, by the way, the data I'm about to send you is a load of nonsense." It simply sends the nonsense!!!

- Simplest kind of failure is called a transient failure, where the sensor occasionally decides to send some nonsense.

- Example: the battery level sensor might have a habit of sending a zero when someone bumps the robot, even if the battery is fully charged.

  5 5 5 5 ... 5 0 0 5 5 5 5

  $BM20$ $BM21$

- What will the simple Gaussian error model lead us to believe about $Battery_{21}$?

- According to Bayes: answer depends on both

  the sensor model $P(BMeter_{21} = 0 \mid Battery_{21})$ and

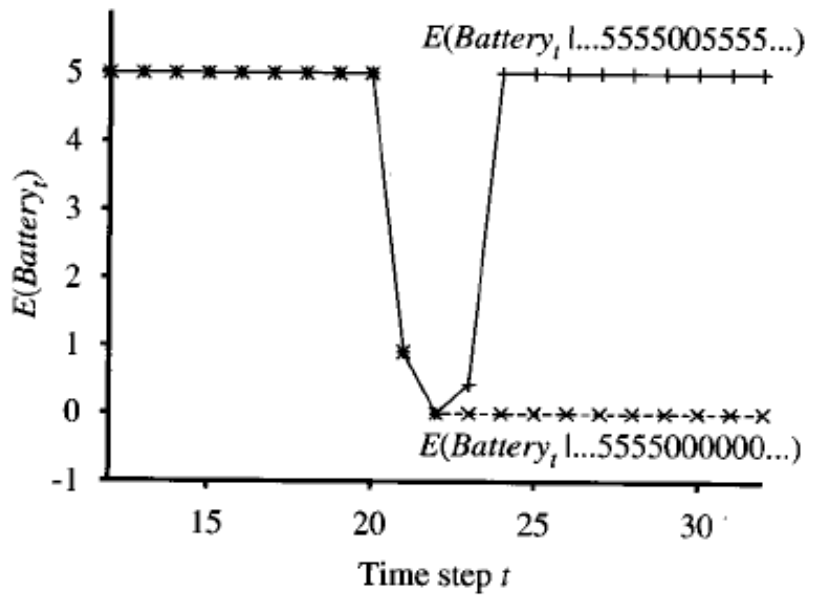
  the prediction $\mathbf{P}(Battery_{21} \mid BMeter_{1:20})$

Sensor - error models – transient failures – persistent failures

- If probability of a large sensor error is significantly less likely than the probability of a transition to $Battery_{21} = 0$, even if the latter is very unlikely, then the posterior distribution will assign a high probability to the battery's being empty
- Second reading of zero at $t = 22$ will make this conclusion almost certain
- If the transient failure then disappears and reading returns to 5 from t = 23 onwards, the estimate for the battery level will quickly return to 5.

Sensor - error models – transient failures – persistent failures

- If probability of a large sensor error is significantly less likely than the probability of a transition to $Battery_{t}$ = 0, even if the latter is very unlikely, then the posterior di
- Se
- If ds, th

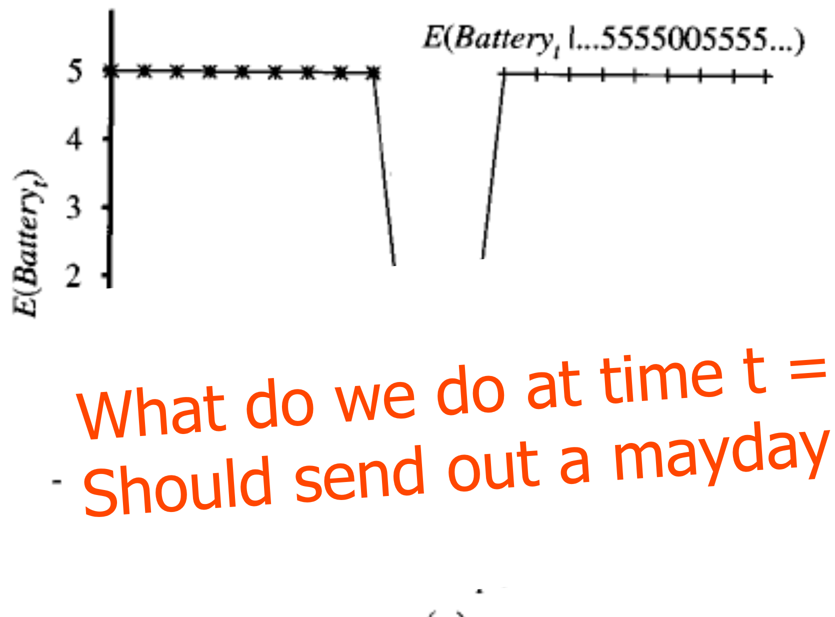


(a)

Figure 15.12 (a) Upper curve: trajectory of the expected value of $Battery_t$ for an observation sequence consisting of all 5s except for 0s at $t = 21$ and $t = 22$, using a simple Gaussian error model. Lower curve: trajectory when the observation remains at 0 from $t = 21$ onwards.

Sensor - error models – transient failures – persistent failures

- If probability of a large sensor error is significantly less likely than the probability of a transition to $Battery_t = 0$, even if the latter is very unlikely, then the posterior
  dis
- Se
- If ds,
  th



What do we do at time t = 22?
Should send out a mayday signal and shut down?

(a)

Figure 15.12 (a) Upper curve: trajectory of the expected value of $Battery_t$ for an observation sequence consisting of all 5s except for 0s at $t = 21$ and $t = 22$, using a simple Gaussian error model. Lower curve: trajectory when the observation remains at 0 from $t = 21$ onwards.

Sensor - error models – transient failures – persistent failures

- If probability of a large sensor error is significantly less likely than the probability of a transition to $Battery$ $= 0$, even if the latter is very unlikely, then the posterior dis
- Se
- If ds,

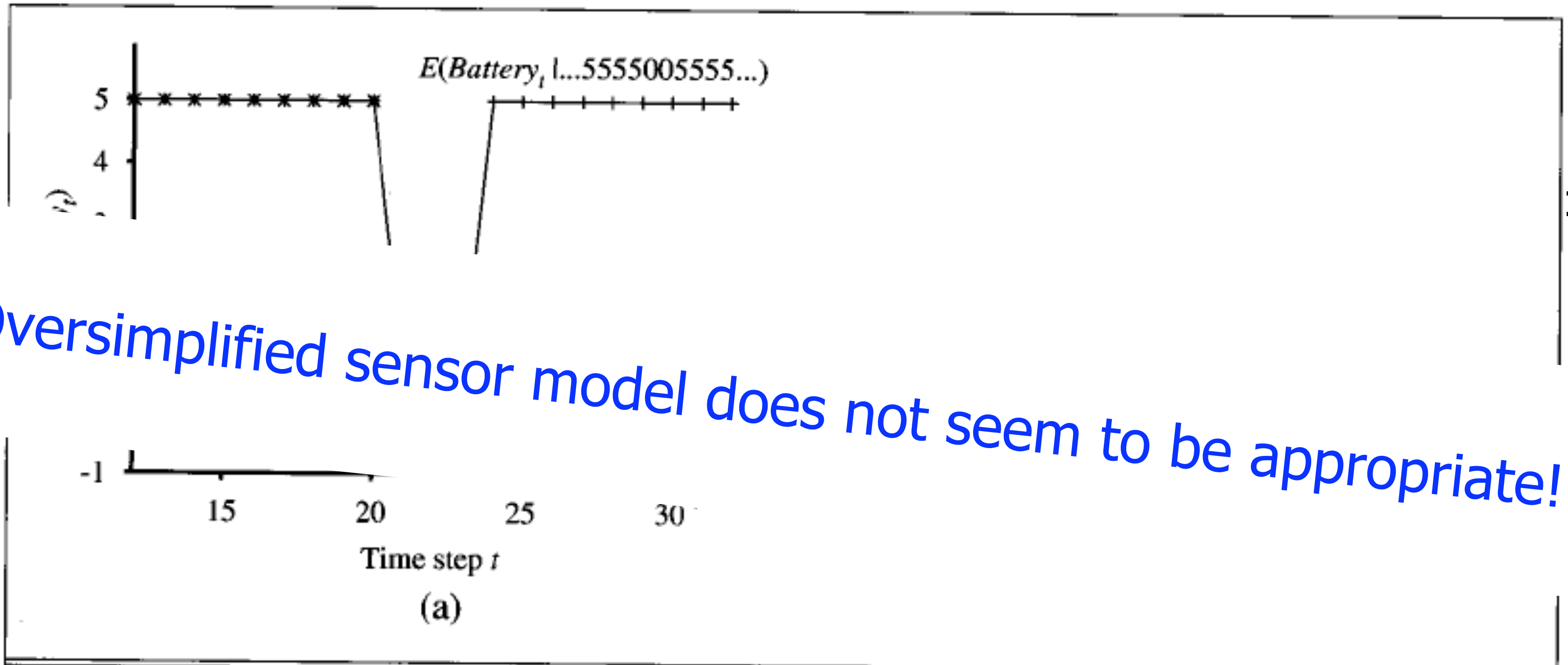


Figure 15.12 (a) Upper curve: trajectory of the expected value of $Battery_t$ for an observation sequence consisting of all 5s except for 0s at $t = 21$ and $t = 22$, using a simple Gaussian error model. Lower curve: trajectory when the observation remains at 0 from $t = 21$ onwards.

Oversimplified sensor model does not seem to be appropriate!!

Sensor - error models – transient failures – persistent failures

Q: How can this be fixed?

A: In order for the system to handle sensor failure properly, the sensor model must include the possibility of a failure.

The simplest kind of failure model for a sensor allows a certain probability that the sensor will return some completely incorrect value, regardless of the true state of the world.

Example:

If the battery meter fails by returning 0, we might say that

$$P(BMeter_t = 0 \mid Battery_t = 5) = 0.03,$$

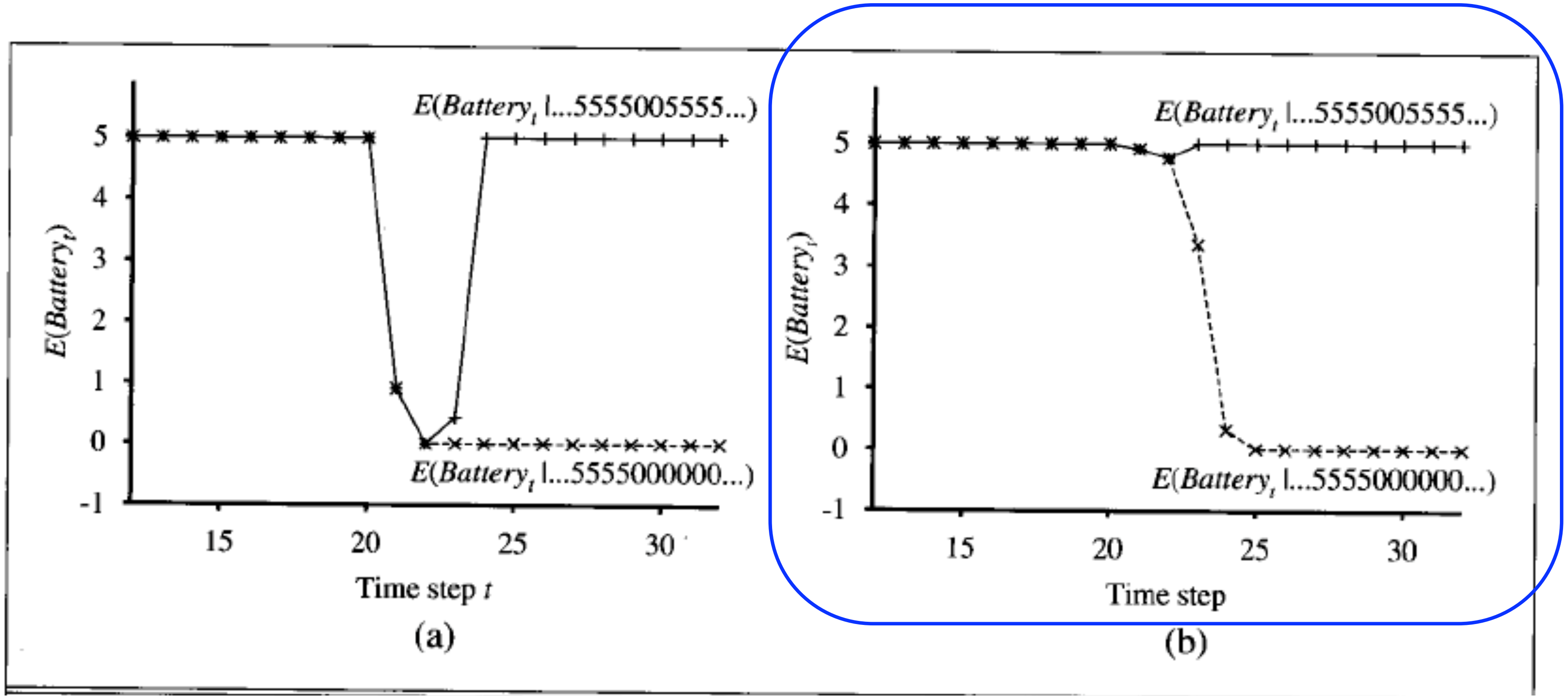
(which is presumably much larger than the probability assigned by the simple Gaussian error)

This is called transient failure model.

Q: How does this help if we are faced with a reading of 0??

Sensor - error models – transient failures – persistent failures

Q: How does this help if we are faced with a reading of 0??

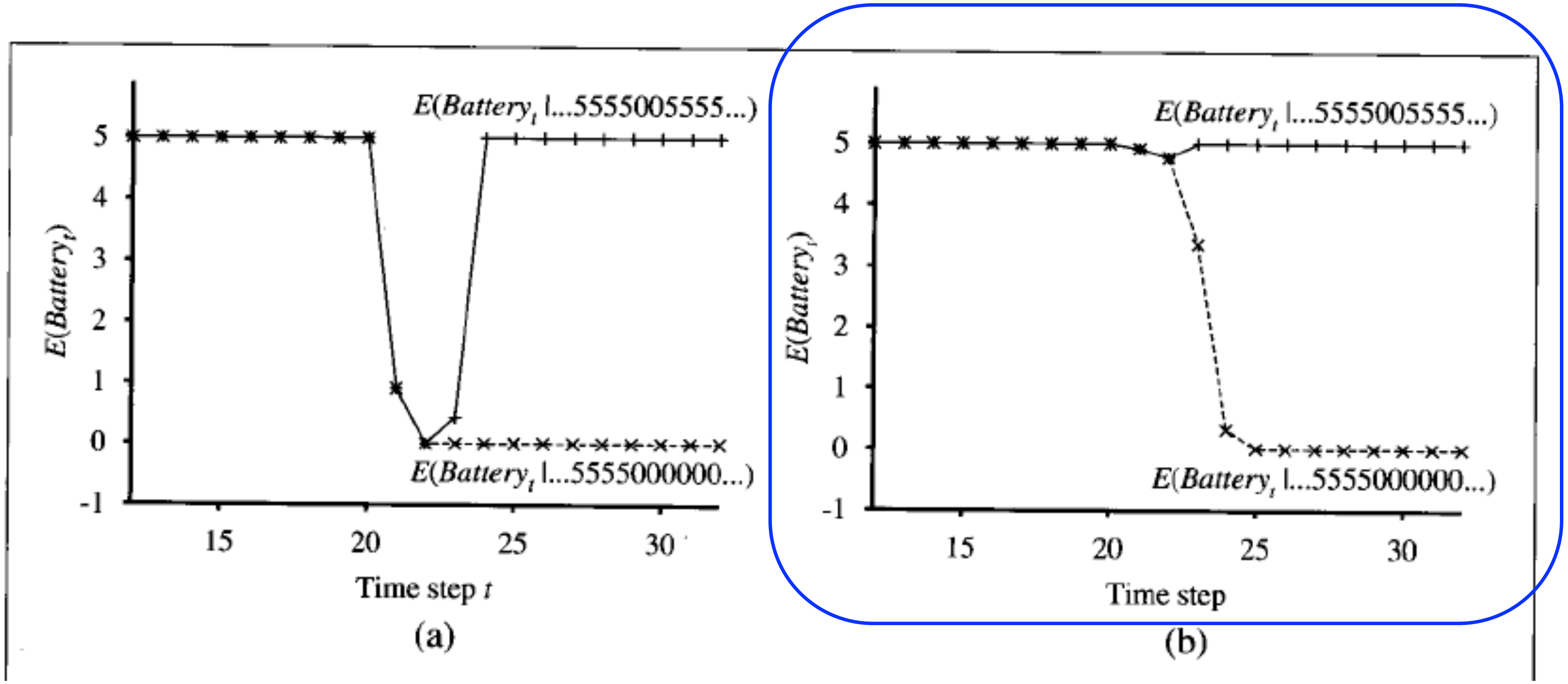


A: Provided that the predicted probability of an empty battery, according to the readings so far, is much less than 0.03, then the best explanation of the observation $BMeter_{21} = 0$ is that the sensor has temporarily failed.

The upper curve in Figure 15.12(b) shows that the transient failure model can handle transient failures without a catastrophic change in beliefs!!

Sensor - error models – transient failures – persistent failures

Q: What about persistent sensor failures?



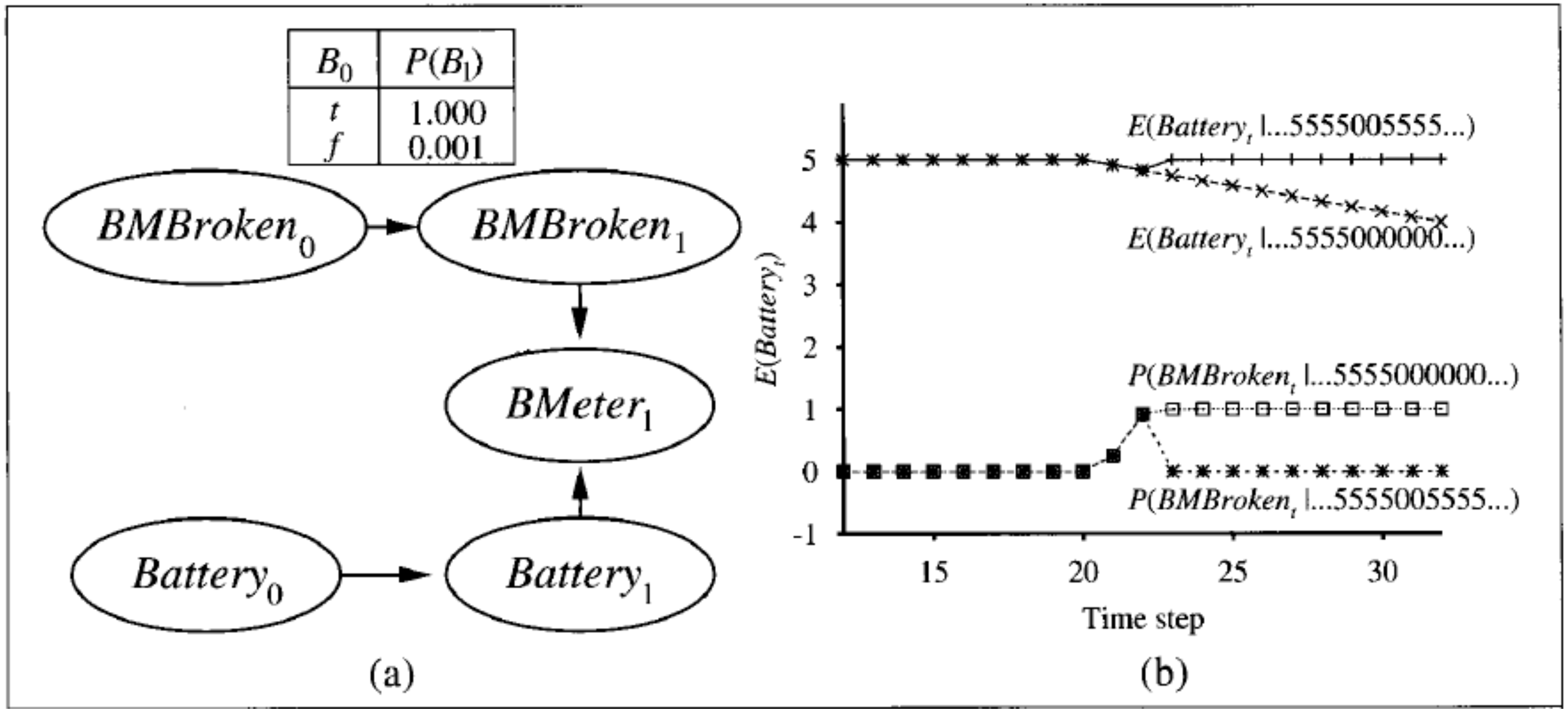
A: If the sensor returns 20 readings of 5 followed by 20 readings of 0, then the transient sensor failure model will result in the robot gradually coming to believe that its battery is empty when in fact it may be that the meter has failed.

The lower curve in Figure 15.12(b) shows the belief "trajectory" for this case. By $t = 25$ – five readings of 0 – the robot is convinced that its battery is empty.

Sensor - error models – transient failures – persistent failures

Q: What about persistent sensor failures?

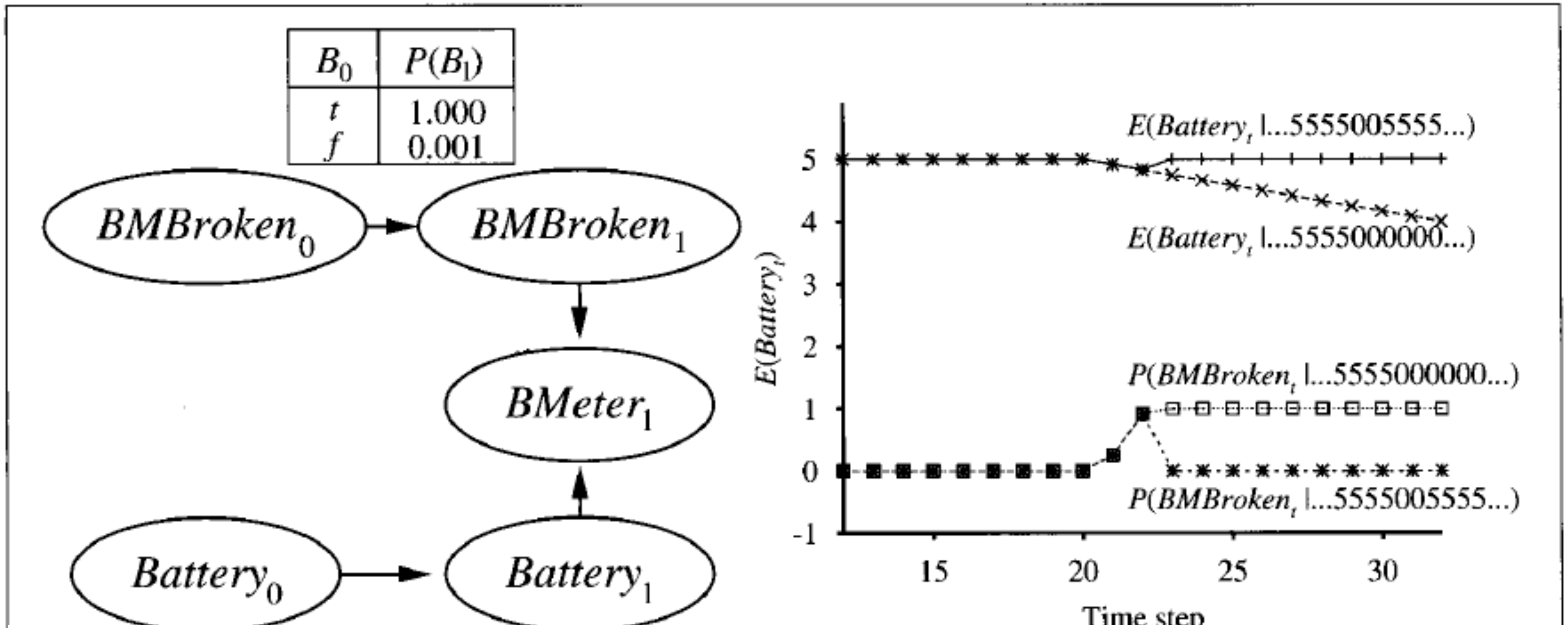
A: We need a persistent failure model that describes how the sensor behaves under normal conditions and after failure.



Persistent failure model

- Augment the hidden state of the system with an additional variable, say $BMBroken$, that describes the status of the battery meter.
- The persistence of failure must be modelled by an arc linking $BMBroken_0$ to $BMBroken_1$.
- This persistence arc has a CPT that gives a small probability of failure in any given time step, say 0.001, but specifies that the sensor stays broken once it breaks.
- When the sensor is OK, the sensor model for $BMeter$ is identical to the transient failure model; when the sensor is broken, it says $BMeter$ is always 0, regardless of the actual battery charge.

Persistent failure model



B_0	$P(B_1)$
t	1.000
f	0.001

- In case of temporary blip, the probability that the sensor is broken rises significantly after second 0 reading, but immediately drops back to zero once a 5 is observed.
- In case of persistent failure, the probability that the sensor is broken rises quickly to almost 1 and stays there.
- Once sensor is known to be broken, robot can only assume that its battery discharges at the "normal" rate, as shown by the gradually descending level of $E(Battery_t | \dots)$

3.6 Summary

- Changing state of the world is handled by using a set of random variables to represent the state at each point tin time.
- Representation can be designed to satisfy the Markov property, so that the future is dependent of the past given the present. Combined with the assumption that the process is stationary — that is, the dynamics do not change over time — this greatly simplifies the representation.
- Temporal probability model can be thought of as containing a transition model describing the evolution and a sensor model describing the observation process.
- Principal inference tasks in temporal models are filtering, prediction, smoothing, and computing the most likely explanation. Each of these can be achieved using simple, recursive algorithms whose runtime is linear in the length of the sequence.
- Three families of temporal models were studied in more depth: hidden Markov models, Kalman filters, and dynamic Bayesian networks (which include the other two as special cases.