Probabilistic Models involving Time

CMT311 Principles of Machine Learning

Probabilistic Models Involving Time

why needed?

- real-life examples often involve evolution over time **need to model temporal data**
- states that evolve over time $\rightarrow$ number of variables increases over times
  $\{x_0, x_1, x_2, \ldots, x_t, \ldots, x_T\}$
- need for models that can represent temporal data $\rightarrow$ **dynamical models**
- need for algorithms to do inference and learning in dynamical models

Finance
Price movement prediction.
Planning
Forecasting – eg how many newspaper to deliver to retailers.

Example

Weather Observation

- A security guard stationed at a secret **underground** installation wants to know whether is raining today.
- The only access to the outside world: every morning he sees the director coming in with, or without, an umbrella.

Inference in dynamical Models

Inference in dynamical Models

- **goal:** estimate state from sensor data / measurements
- **but:**
  - sensor data corrupted by noise
  - system process and measurements process are uncertain
  - state often not directly fully observable
- **result:** **uncertain** estimate
  $\Rightarrow$ uncertainty explicitly modelled as **probability distributions**

States and Observations

- View the world as a series of snapshots, or **time slices**.
- Each slice contains a set of random variables (some observable and some not)
- The same set of variables is observable in each time slice

States and Observations

- View the world as a series of snapshots, or **time slices**.
- Each slice contains a set of random variables (some observable and some not)
- The same set of variables is observable in each time slice

- **state:** $X_{0:T} = \{X_0, X_1, X_2, \ldots, X_t, \ldots, X_T\}$

  In our example for each day t we have a state variable $R_{0:T} = \{R_0, R_1, \ldots, R_t, \ldots, R_T\}$ (whether it is raining)
- **data/measurement/observations:** $E_{1:T} = \{E_1, E_2, \ldots, E_t, \ldots, E_T\}$

  In our example for each day an observation variable $U_{0:T} = \{U_0, U_1, \ldots, U_t, \ldots, U_T\}$ (whether the umbrella appears)

Dynamical Models

Given all the state/observation/input variables a dynamical model specifies the probability distribution over the latest state variables, given the previous values, that is,

$$P(X_t \mid X_{0:t-1})$$

Problem: the set $X_{0:t-1}$ is unbounded in size as t increases.

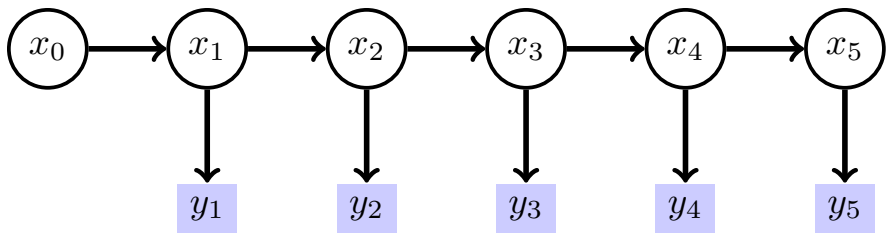
Solved by Markov property/assumption

- A stochastic process has the *Markov property* if the conditional probability distribution of future states of the process (conditional on both past and present values) depends only upon the present state, not on the sequence of events that preceded it.

$$P(X_{t+1:T} \mid X_{0:t}) = P(X_{t+1:T} \mid X_t)$$

inference in dynamical models

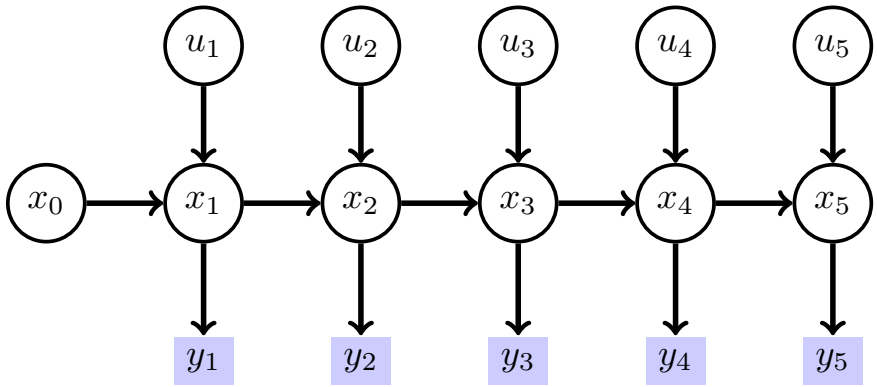
- **state:** $x_{0:T} = \{x_0, x_1, x_2, \ldots, x_t, \ldots, x_T\}$
- **data/measurement/observations:** $y_{1:T} = \{y_1, y_2, \ldots, y_t, \ldots, y_T\}$



- **prior probability distribution**: $x_0 \sim P(x_0)$
- **system model**: $x_t \sim P(x_t \mid x_{t-1})$
- **measurement model:** $y_t \sim P(y_t \mid x_t)$

Inference in dynamical Models

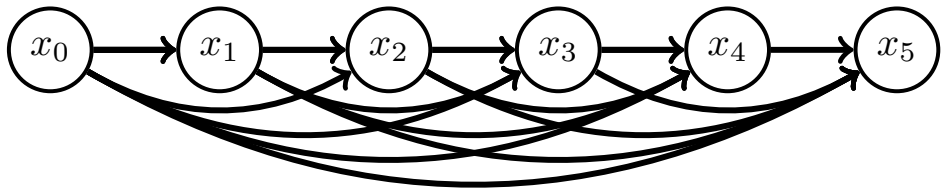
- **state:** $x_{0:T} = \{x_0, x_1, x_2, \ldots, x_t, \ldots, x_T\}$
- **data/measurement/observations:** $y_{1:T} = \{y_1, y_2, \ldots, y_t, \ldots, y_T\}$
- **inputs:** $u_{1:T} = \{u_1, u_2, \ldots, u_t, \ldots, u_T\}$



- **prior probability distribution**: $x_0 \sim P(x_0)$
- **system model**: $x_t \sim P(x_t \mid x_{t-1}, u_t)$
- **measurement model:** $y_t \sim P(y_t \mid x_t)$

Markov models

The following decomposition is always possible:
$P(x_{1:t}) = \prod_1^T P(x_t|x_{1:t-1})$



Markov chain

A Markov chain on (discrete or continuous) variables is a dynamical model in which the following conditional independence assumption holds:
$P(x_t|x_{1:t-1}) = P(x_t|x_{t-L:t-1})$

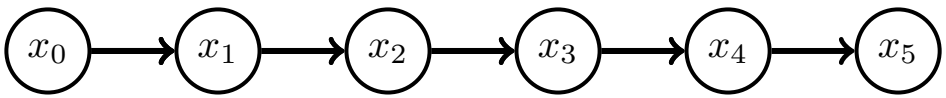
- $L \geq 1$ is the order of the Markov chain
- $x_t = \varnothing$ for $t < 1$

Markov models

A first order Markov chain

A first order Markov chain on (discrete or continuous) variables is a dynamical model in which the following conditional independence assumption holds: $P(x_t|x_{1:t-1}) = P(x_t|x_{t-1})$

- $x_t = \varnothing$ for $t < 1$

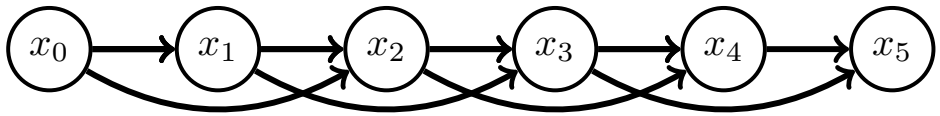


Markov models

A second order Markov chain

A second order Markov chain on (discrete or continuous) variables is a dynamical model in which the following conditional independence assumption holds: $P(x_t|x_{1:t-1}) = P(x_t|x_{t-2:t-1})$

- $x_t = \varnothing$ for $t < 1$



Stationary Markov Chain

Problem: there are infinitely many possible values of t.
Do we need to specify a different distribution for each time step?

A Markov chain is *stationary* if the state transition probabilities do not change over time

In the umbrella world: the conditional probability of rain, $P(R_t \mid R_{t|_1})$, is the same for all t, and we only have to specify one conditional probability table.

A discrete first order stationary Markov chain

- Markov chain
- discrete states
- first order
- stationary

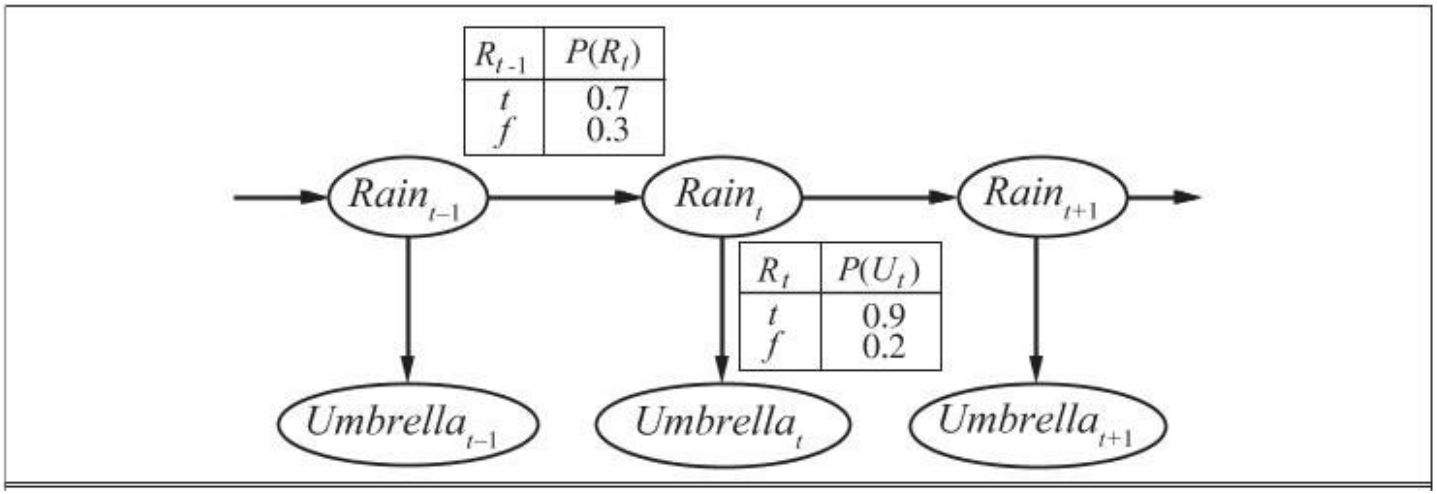
$P(x_t = s' | x_t - 1 = s) = f(s, s').$ (for some function f)

Markov models

A discrete first order stationary Markov chain can be visualised using a state transition matrix or a state-transition diagram:

Example 1:

The umbrella world



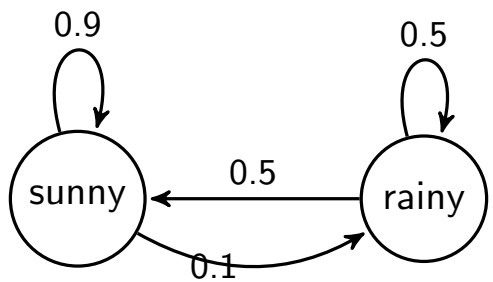
Markov models

A discrete first order stationary Markov chain can be visualised using a state transition matrix or a state-transition diagram:

Example:

- x is random variable describing weather conditions
  $dom(x) = \{rainy, sunny\}$
- $P(sunny|sunny) = 0.9$, $P(sunny|rainy) = 0.5$
- transition matrix:

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} \Rightarrow P = \begin{bmatrix} P(sunny|sunny) & P(rainy|sunny) \\ P(sunny|rainy) & P(rainy|rainy) \end{bmatrix}$$

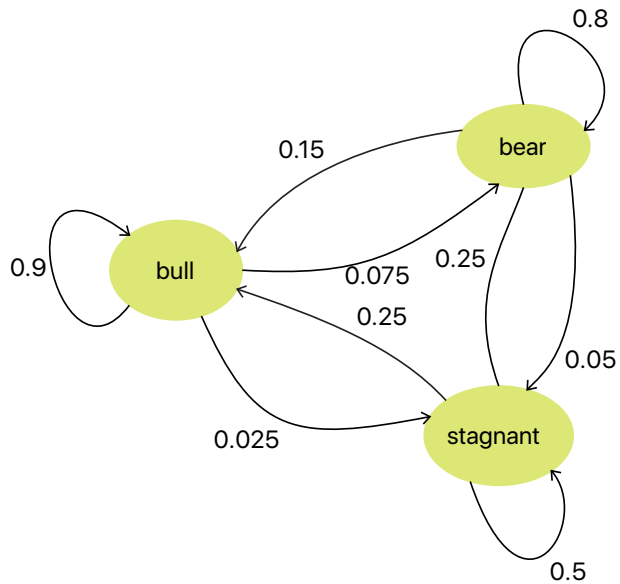


Markov models

Example 3:

x is the week behaviour of a hypothetical stock market[1],
$domx = \{bull, bear, stagnant\}$

$$P = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$



[1] A bull market is a market that is on the rise and is economically sound, while a bear market is a market that is receding

Inference in Dynamical Models

- **filtering**: inferring the present $P(X_t \mid x_0, \mathbf{e}_{1:t})$
- **prediction**: inferring the future $P(X_t \mid x_0, \mathbf{e}_{1:T})$ with $t > T$
- **smoothing**: inferring the past $P(X_t \mid x_0, \mathbf{e}_{1:T})$ with $0 \leq t \leq T$

Inference in temporal Models

Filtering

This is the task of computing the **belief state**, that is the posterior distribution over the most recent sate given all the evidence to date.

$$P(X_t \mid \mathbf{x}_0, \mathbf{e}_{1:t})$$

In our example $P(R_t \mid U_{1:t})$: computing the probability of rain today, given all the observations of the umbrella carrier made so far.

Prediction

This is the task of computing the posterior distribution over the *future state*

$$P(X_{t+k} \mid \mathbf{x_0}, \mathbf{e}_{1:t}) \text{ with } k > 0$$

In our example $P(R_{t+3}|U_{1:t})$: computing the probability of rain three days from now, given all the observations to date.

This is useful e.g. for evaluating possible courses of actions based on their expected outcomes.

Inference in temporal Models

Smoothing

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

$$P(X_k \mid \mathbf{x}_0, \mathbf{e}_{1:t}) \text{ for some } 0 \leq k < t$$

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

Smoothing provides a better estimate of the present state because it incorporates mode evidence.

Filtering and Prediction

Filtering

- A useful filtering algorithm needs to maintain the current state estimate and update it.
- Given the result of filtering up to time t, the agent needs to compute the result for $t + 1$ from the new evidence $\mathbf{e}_{t+1}$,

recursive estimation:

- the current state distribution is projected forward form t to $t + 1$
- updating using the new evidence $\mathbf{e}_{t+1}$.

Filtering and Prediction

Filtering

- A useful filtering algorithm needs to maintain the current state estimate and update it.
- Given the result of filtering up to time t, the agent needs to compute the result for $t+1$ from the new evidence $\mathbf{e}_{t+1}$,

recursive estimation:

$$
\begin{aligned}
P(X_{t+1}|\mathbf{e}_{1:t+1}) &= P(X_{t+1}|\mathbf{e}_{1:t}, \mathbf{e}_{t+1}) \quad \text{(dividing up the evidence)} \\
&= \alpha P(\mathbf{e}_{t+1}|X_{t+1}, \mathbf{e}_{1:t}) P(X_{t+1} \mid \mathbf{e}_{1:t}) \quad \text{(using Bayes' rule)} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by Markov assumption)} \\
&= \alpha P(\mathbf{e}_{t+1}|X_{t+1}) \underbrace{P(X_{t+1} \mid \mathbf{e}_{1:t})}_{\text{one step prediction}}
\end{aligned}
$$

where α is a normalising constant to make probabilities sum up to 1.

Recursive Estimation

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

Recursive Estimation

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

(using Markov assumption)

$$= \alpha P(\mathbf{e}_{t+1}|X_{t+1}) \sum_{\mathbf{x}_t} P(X_{t+1} \mid \mathbf{x}_t) \quad \underbrace{P(\mathbf{x}_t) \mid \mathbf{e}_{1:t})}_{\text{current state distribution}}$$

Recursive Estimation

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

(using Markov assumption)

$$= \alpha P(\mathbf{e}_{t+1}|X_{t+1}) \sum_{\mathbf{x}_t} P(X_{t+1} \mid \mathbf{x}_t) \underbrace{P(\mathbf{x}_t) \mid \mathbf{e}_{1:t})}_{\text{current state distribution}}$$

$P(X_t \mid \mathbf{e}_{1:t})$ can be seen as a "message" $\mathbf{f}_{1:t}$ propagated *forward* along the sequence.

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

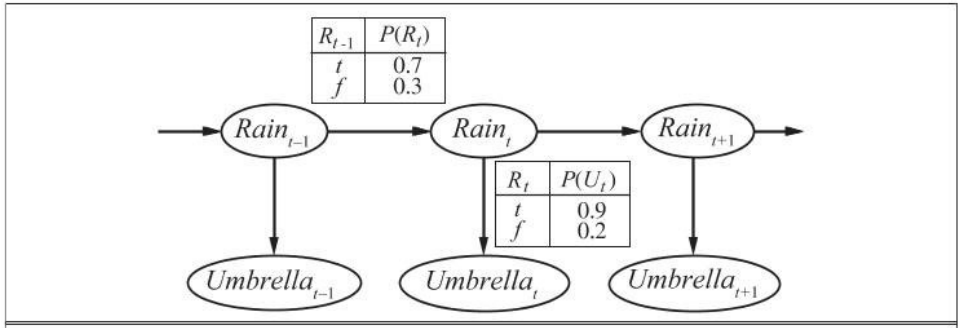
where FORWARD implements the update described the equation above. And

$$\mathbf{f}_{1:0} = P(X_0)$$

Updating

- When all the state variables are discrete, the time and space required for each update is constant;
- The constants depend on the size of the state space sn on the specific type of temporal model;
  - ⋆ Important when keeping track of of the current state distribution over an unbounded sequence of observation with limited memory,

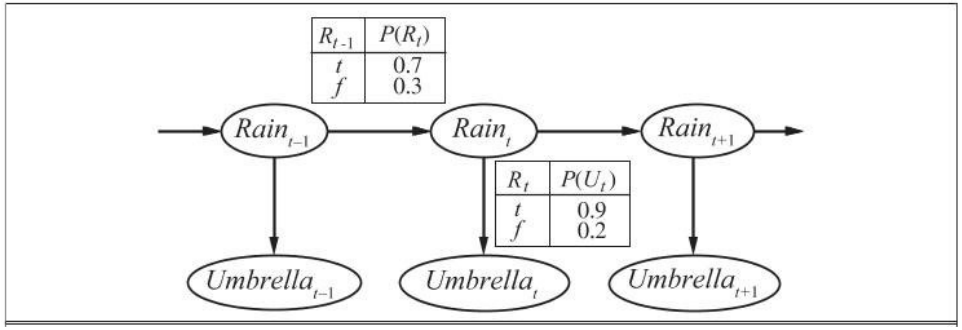
Example: Two steps in the umbrella scenario:



Compute $P(R_2 \mid u_{1:2})$:

- On day 0: no observations, only the security guard's prior beliefs; let's assume $P(R_0) = \langle 0.5, 0.5 \rangle$.

Example: Two steps in the umbrella scenario:

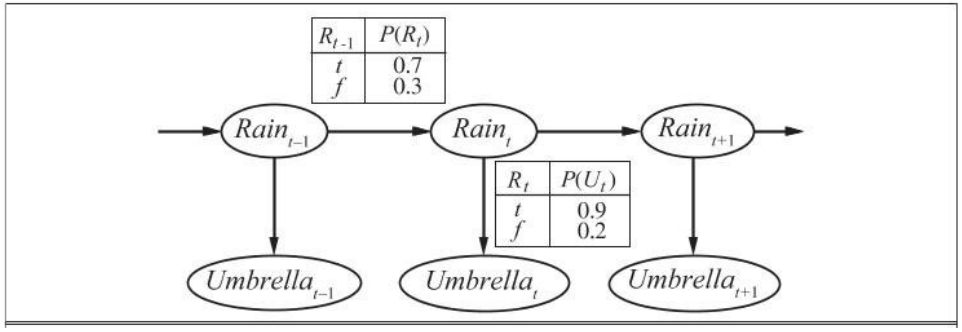


Compute $P(R_2 \mid u_{1:2})$:

- On day 1: he sees the umbrella, so $U_1 = \mathit{true}$. The prediction from t=0 to t=1 is

$$P(R1) = \sum_{r_0} P(R_1 \mid r_0) P(r_0)$$
$$= \langle 0.7, 0.3 \rangle \times 0.5 + \langle 0.3, 0.7 \rangle \times 0.5 = \langle 0.5, 0.5 \rangle$$

Example: Two steps in the umbrella scenario:



Compute $P(R_2 \mid u_{1:2})$:

- On day 1: he sees the umbrella, so $U_1 = \textit{true}$. The prediction from t=0 to t=1 is

$$P(R1) = \sum_{r_0} P(R_1 \mid r_0)P(r_0)$$
$$= \langle 0.7, 0.3\rangle \times 0.5 + \langle 0.3, 0.7\rangle \times 0.5 = \langle 0.5, 0.5\rangle$$

update step:

$$P(R_1 \mid u_1) = \alpha P(u_1 \mid R_1)P(R_1) = \alpha\langle 0.9, 0.2\rangle\langle 0.5, 0.5\rangle$$
$$= \alpha\langle 0.45, 0.1\rangle \approx \langle 0.818, 0.182\rangle.$$

Example (cont.)

- On day 2, the umbrella appears, so $U_2 = \mathit{true}$. The prediction from $t = 1$ to $t = 2$ is

$$\begin{aligned} P(R_2 \mid u_1) &= \sum_{r_1} P(R_2 \mid r_1) P(r_1 | u_1) \\ &= \langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.187 \approx \langle 0.627, 0.373 \rangle, \end{aligned}$$

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

$$\begin{aligned} P(R_2 \mid u_1, u_2) &= \alpha p(u_2 \mid R_2) p(R_2 \mid u_1) \\ &= \alpha \langle 0.9, 0.2. \rangle \langle 0.627, 0.373 \rangle \\ &= \alpha \langle 0.565, 0.075 \rangle \approx \langle 0.883, 0.117 \rangle. \end{aligned}$$

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

Forward recursion for prediction

- Prediction can be seen simply as filtering **without** the addition of new evidence.

Forward recursion for prediction

- Prediction can be seen simply as filtering **without** the addition of new evidence.
- for predicting the state at $t + k + 1$ from a prediction for $t + k$:

$$P(X_{t+k+1} \mid \mathbf{e}_{1:t}) = \sum_{\mathbf{x}_{t+k}} P(X_{t+k+1} \mid \mathbf{x}_{t+k}) P(x_{t+k} \mid \mathbf{e}_{1:t}).$$

Forward recursion for prediction

- Prediction can be seen simply as filtering **without** the addition of new evidence.
- for predicting the state at $t + k + 1$ from a prediction for $t + k$:

$$P(X_{t+k+1} \mid \mathbf{e}_{1:t}) = \sum_{\mathbf{x}_{t+k}} P(X_{t+k+1} \mid \mathbf{x}_{t+k}) P(x_{t+k} \mid \mathbf{e}_{1:t}).$$

Note: Computation involves only the transition model and not the sensor model.

Likelihood of the evidence

We can use a forward recursion to compute the likelihood of the evidence sequence $P(\mathbf{e}_{1:t})$

- Useful to compare different temporal models that might have produced the same evidence sequence

  For example, two different models for the persistence of rain.

- Use a likelihood message $\ell_{1:t}(X_t) = P(X_t, \mathbf{e}_{1:t})$.

Likelihood of the evidence

We can use a forward recursion to compute the likelihood of the evidence sequence $P(\mathbf{e}_{1:t})$

- Useful to compare different temporal models that might have produced the same evidence sequence

  For example, two different models for the persistence of rain.
- Use a likelihood message $\ell_{1:t}(X_t) = P(X_t, \mathbf{e}_{1:t})$.
- message calculation is identical to that for filtering:

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

Likelihood of the evidence

We can use a forward recursion to compute the likelihood of the evidence sequence $P(\mathbf{e}_{1:t})$

- Useful to compare different temporal models that might have produced the same evidence sequence

  For example, two different models for the persistence of rain.

- Use a likelihood message $\ell_{1:t}(X_t) = P(X_t, \mathbf{e}_{1:t})$.
- message calculation is identical to that for filtering:

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

- the actual likelihood is obtained by summing out X_t:

$$L_{1:t} = P(\mathbf{e}_{1:t}) = \sum_{\mathbf{x}_t} \ell_{1:t}(\mathbf{x}_t)$$

Smoothing

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

$$P(X_k|\mathbf{e}_{1:t}) \text{ for } 0 \leq k < t.$$

Smoothing

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

$$P(X_k|\mathbf{e}_{1:t}) \text{ for } 0 \leq k < t.$$

Recursive message-passing approach

$$\begin{aligned} P(X_k|\mathbf{e}_{1:t}) &= P(X_k|\mathbf{e}_{1:k}, \mathbf{e}_{k+1:t}) \\ &= \alpha P(X_k|\mathbf{e}_{1:k}) P(\mathbf{e}_{k+1:t}|X_k, \mathbf{e}_{1:k}) \text{(using Bayes' rule)} \\ &= \alpha P(X_k|\mathbf{e}_{1:k}) P(\mathbf{e}_{k+1:t}|X_k) \text{ (using conditional independence)} \\ &= \alpha \mathbf{f}_{1:k} \times \mathbf{b}_{k+1:t} \end{aligned}$$

where $\times$ represents point-wise multiplication of vectors. And $\mathbf{b}_{k+1:t} = P(\mathbf{e}_{k+1:t}|X_k)$ is a "backward" message.

Smoothing

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

$$P(X_k|\mathbf{e}_{1:t}) \text{ for } 0 \leq k < t.$$

Recursive message-passing approach

$$\begin{aligned} P(X_k|\mathbf{e}_{1:t}) &= P(X_k|\mathbf{e}_{1:k}, \mathbf{e}_{k+1:t}) \\ &= \alpha P(X_k|\mathbf{e}_{1:k}) P(\mathbf{e}_{k+1:t}|X_k, \mathbf{e}_{1:k}) \text{(using Bayes' rule)} \\ &= \alpha P(X_k|\mathbf{e}_{1:k}) P(\mathbf{e}_{k+1:t}|X_k) \text{ (using conditional independence)} \\ &= \alpha \mathbf{f}_{1:k} \times \mathbf{b}_{k+1:t} \end{aligned}$$

where $\times$ represents point-wise multiplication of vectors. And $\mathbf{b}_{k+1:t} = P(\mathbf{e}_{k+1:t}|X_k)$ is a "backward" message.

The forward message $\mathbf{f}_{1:k}$ can be computed by filtering forward from 1 to k, as before

Backward message Computation

Backward message $\mathbf{b}_{k+1:t}$ can be computed by a recursive process that runs backward from t:

$$P(\mathbf{e}_{k+1:t}|X_k) = \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})P(\mathbf{x}_{k+1}|X_k)$$

Of the three factors in this summation, the first and third are obtained directly from the model, and the second is the "recursive call."
Using the message notation, we have

Backward message Computation

Backward message $\mathbf{b}_{k+1:t}$ can be computed by a recursive process that runs backward from t:

$$P(\mathbf{e}_{k+1:t}|X_k) = \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})P(\mathbf{x}_{k+1}|X_k)$$

Of the three factors in this summation, the first and third are obtained directly from the model, and the second is the "recursive call."
Using the message notation, we have

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

Backward message Computation

Backward message $\mathbf{b}_{k+1:t}$ can be computed by a recursive process that runs backward from t:

$$P(\mathbf{e}_{k+1:t}|X_k) = \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})P(\mathbf{x}_{k+1}|X_k)$$

Of the three factors in this summation, the first and third are obtained directly from the model, and the second is the "recursive call."
Using the message notation, we have

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

with

$$\mathbf{b}_{t+1:t} = P(\mathbf{e}_{t+1:t}|X_t) = P(|X_t)\mathbf{1},$$

where $\mathbf{1}$ is a vector of 1s.

Backward message Computation

Backward message $\mathbf{b}_{k+1:t}$ can be computed by a recursive process that runs backward from t:

$$P(\mathbf{e}_{k+1:t}|X_k) = \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})P(\mathbf{x}_{k+1}|X_k)$$

Of the three factors in this summation, the first and third are obtained directly from the model, and the second is the "recursive call."
Using the message notation, we have

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

with

$$\mathbf{b}_{t+1:t} = P(\mathbf{e}_{t+1:t}|X_t) = P(|X_t)\mathbf{1},$$

where $\mathbf{1}$ is a vector of 1s.
(Because $\mathbf{e}_{t+1:t}$ is an empty sequence, the probability of observing it is 1.)

Exercise

Apply this algorithm to the umbrella example,