

Artificial Intelligence

14. Markov Models

Shashi Prabh

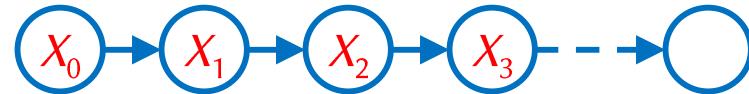
School of Engineering and Applied Science
Ahmedabad University

Uncertainty and Time

- Method: Belief state, transition model and sensor model
- Often, we want to reason about a sequence of observations where the state of the underlying system is changing
 - Speech recognition
 - Robot localization
 - User attention
 - Medical monitoring
 - Global climate
- Now we need to introduce time into our models

Discrete-Time Model

- Discrete-time model views the problem as snapshots in time, called **time slices**
- Each time slice contains a set of random variables, some observable and some not
- We will assume the same subset of variables are observable in every time slice



Discrete-Time Model

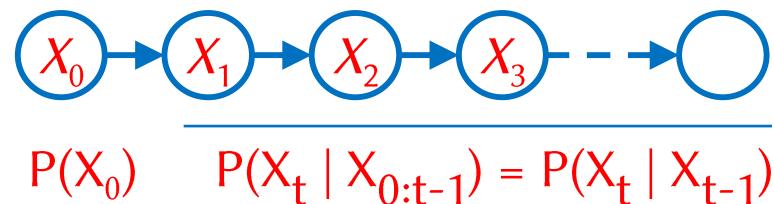
- Set of variables at time t :
 - X_t : state variables, assumed to be unobservable
 - E_t : evidence variables, set of observable variables
 - e_t : observations at time t
- Transition model $P(X_t | X_{0:t-1})$
 - Specifies the probability distribution over the latest state variables given the previous values
 - Problem: Set $X_{0:t-1}$ is unbounded in size

Notation:

$$X_{a:b} = X_a, X_{a+1}, \dots, X_b$$

Markov Models

- **Markov assumption:** current state depends on only a finite fixed number of the past states
- Processes satisfying this assumption are called **Markov chain** or **Markov processes**
- **First order Markov process** -- the present state depends only on the previous state: $P(X_t | X_{0:t-1}) = P(X_t | X_{t-1})$
 - X_{t+1} is independent of X_0, \dots, X_{t-1} given X_t



Markov Models

- Joint distribution $P(X_0, \dots, X_T) = P(X_0) \prod_t P(X_t | X_{t-1})$
- A k^{th} -order model allows dependencies on k earlier steps
 - Higher order Markov chains can be transformed to the first order chain
- Time-homogeneous or Stationary process assumption
 - Not good we had to specify a different $P(X_t | X_{t-1})$ for each t
 - We assume that the transition probabilities are the same at all times
 - Usually a justifiable assumption
 - Helps to avoid specifying potentially infinite number of probabilities

Markov Models

- Sensor Model (or, observation model)
 - Evidence variable, E_t , could depend on previous variables and observations leading to similar problem as the transition model
 - Sensor Markov assumption: a current state suffices to generate the current sensor values

$$P(E_t | X_{0:t}, E_{1:t-1}) = P(E_t | X_t)$$

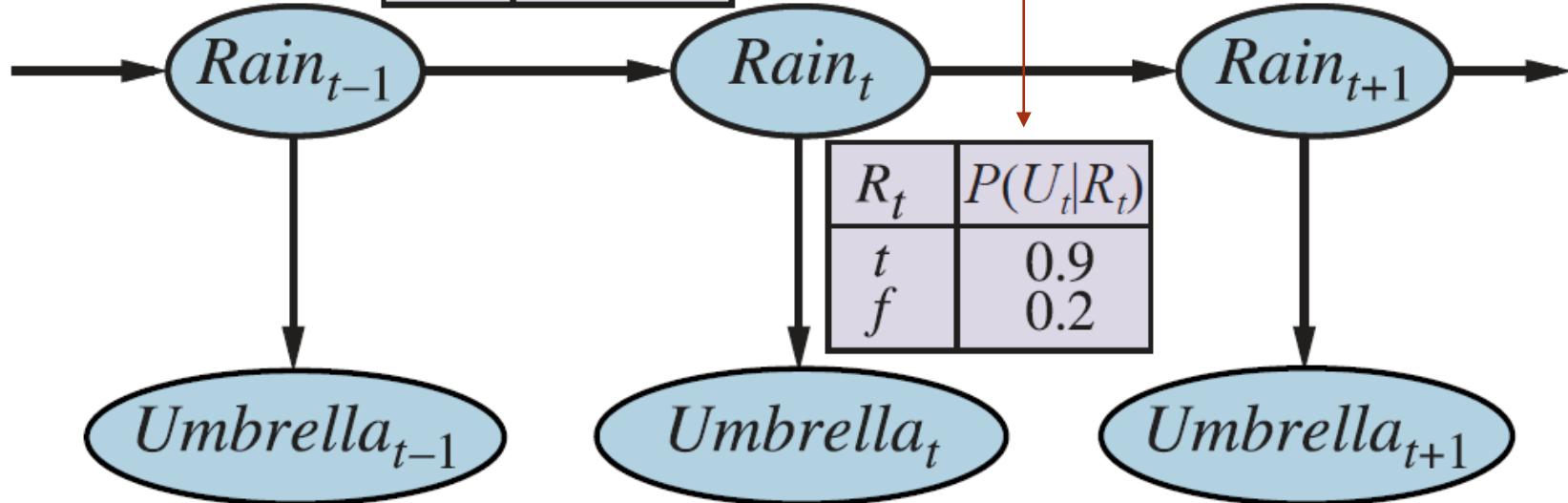
- $P(E_t | X_t)$ is our sensor model

Example

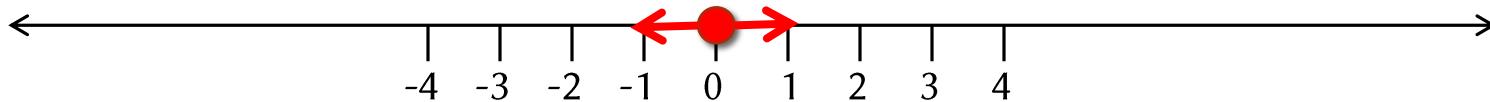
Transition Model

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

Sensor Model



Example: Random walk in one dimension

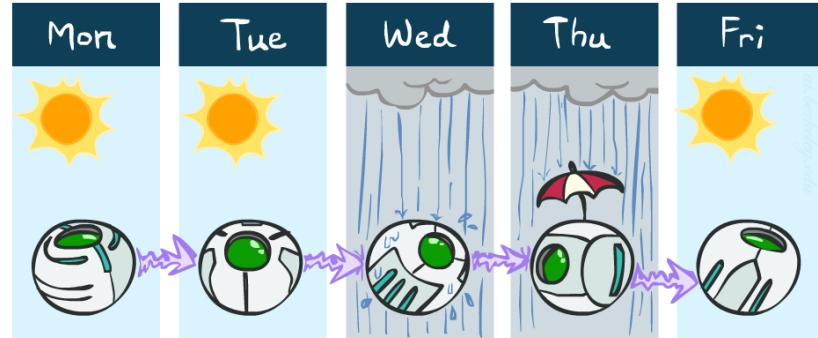


- State: location on the unbounded integer line
- Initial state: starts at 0
- Transition model: $P(X_t = k | X_{t-1} = k \pm 1) = 0.5$
- Applications: particle motion in crystals, stock prices, gambling, genetics, ...
- How far does it get as a function of t ?
 - Expected distance is $O(\sqrt{t})$
- Does it get back to 0 or can it go off for ever and not come back?
 - In 1D and 2D, returns w.p. 1; in 3D, returns w.p. 0.34053733

Example: Weather

- States {rain, sun}
- Initial distribution $P(X_0)$

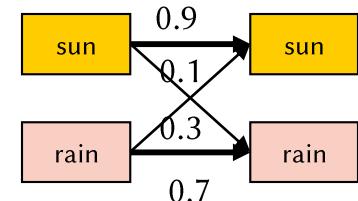
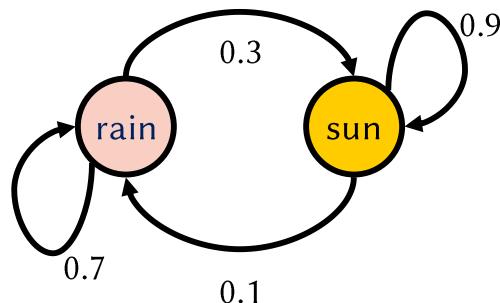
$P(X_0)$	
sun	rain
0.5	0.5



- Transition model $P(X_t | X_{t-1})$

X_{t-1}	$P(X_t X_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

More ways of representing the same CPT

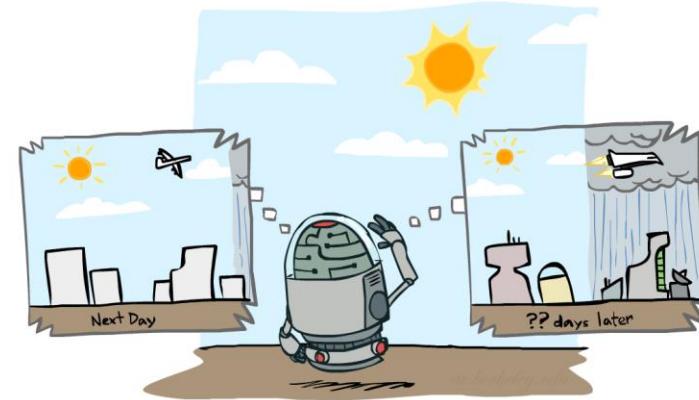


Weather prediction

- Time 0: $\langle 0.5, 0.5 \rangle$

X_{t-1}	$P(X_t X_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

- What is the weather like at time 1?



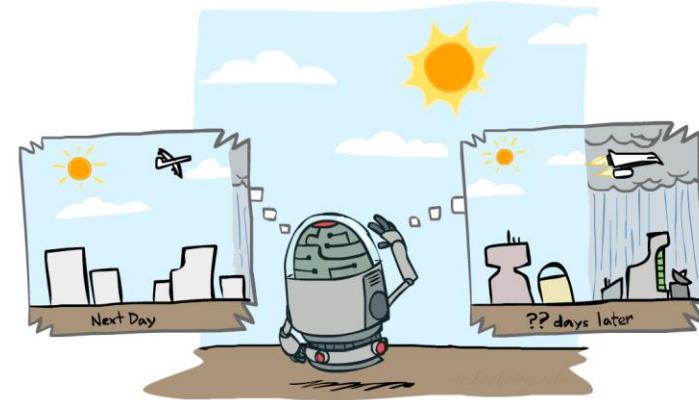
$$\begin{aligned} P(X_1) &= \sum_{x_0} P(X_1, X_0=x_0) \\ &= \sum_{x_0} P(X_0=x_0) P(X_1|X_0=x_0) \\ &= 0.5\langle 0.9, 0.1 \rangle + 0.5\langle 0.3, 0.7 \rangle = \langle 0.6, 0.4 \rangle \end{aligned}$$

Weather prediction, contd.

- Time 1: $\langle 0.6, 0.4 \rangle$

X_{t-1}	$P(X_t X_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

- What is the weather like at time 2?



$$\begin{aligned} P(X_2) &= \sum_{x_1} P(X_2, X_1=x_1) \\ &= \sum_{x_1} P(X_1=x_1) P(X_2|X_1=x_1) \\ &= 0.6\langle 0.9, 0.1 \rangle + 0.4\langle 0.3, 0.7 \rangle = \langle 0.66, 0.34 \rangle \end{aligned}$$

Weather prediction, contd.

- Time 2: $\langle 0.66, 0.34 \rangle$
- What is the weather like at time 3?

X_{t-1}	$P(X_t X_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

$$\begin{aligned} P(X_3) &= \sum_{x_2} P(X_3, X_2=x_2) \\ &= \sum_{x_2} P(X_2=x_2) P(X_3 | X_2=x_2) \\ &= 0.66\langle 0.9, 0.1 \rangle + 0.34\langle 0.3, 0.7 \rangle = \langle 0.696, 0.304 \rangle \end{aligned}$$

Homework

$P(X_0)$	
sun	rain
0	
	1

- The influence of initial distribution gets less and less over time. The distribution much later becomes independent of the initial distribution

Forward algorithm (simple form)

- What is the state at time t ?

Transition model

$$\begin{aligned} P(X_t) &= \sum_{x_{t-1}} P(X_t, X_{t-1}=x_{t-1}) \\ &= \sum_{x_{t-1}} P(X_{t-1}=x_{t-1}) P(X_t | X_{t-1}=x_{t-1}) \end{aligned}$$

- Iterate this update starting at $t=0$

- This is called a **recursive** update:

$$P_t = g(P_{t-1}) = g(g(g(g(\dots P_0))))$$

And the same thing in linear algebra

- What is the weather like at time 2?

$$P(X_2) = 0.6 \langle 0.9, 0.1 \rangle + 0.4 \langle 0.3, 0.7 \rangle = \langle 0.66, 0.34 \rangle$$

- In matrix-vector form:

$$P(X_2) = \begin{pmatrix} 0.9 & 0.3 \\ 0.1 & 0.7 \end{pmatrix} \begin{pmatrix} 0.6 \\ 0.4 \end{pmatrix} = \begin{pmatrix} 0.66 \\ 0.34 \end{pmatrix}$$

X_{t-1}	$P(X_t X_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

i.e., multiply by T^T , transpose of transition matrix

Stationary Distributions

- The limiting distribution is called the **stationary distribution** P_∞ of the chain
- It satisfies $P_\infty = P_{\infty+1} = T^T P_\infty$
- Solving for P_∞ in the example:

$$\begin{pmatrix} 0.9 & 0.3 \\ 0.1 & 0.7 \end{pmatrix} \begin{pmatrix} p \\ 1-p \end{pmatrix} = \begin{pmatrix} p \\ 1-p \end{pmatrix}$$

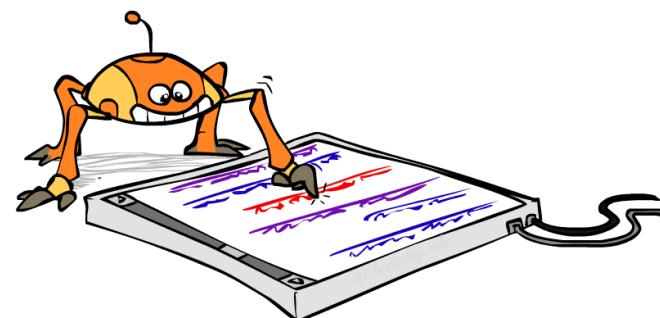
$$0.9p + 0.3(1-p) = p, \text{ or, } p = 0.75$$

- Stationary distribution is $<0.75, 0.25>$ **regardless of the starting distribution**



Example: Web browsing

- State: URL visited at step t
- Transition model:
 - With probability p , choose an outgoing link at random
 - With probability $(1-p)$, choose an arbitrary new page
- Question: What is the **stationary distribution** over pages?
 - That is, if the process runs forever, what fraction of time does it spend in any given page?
- Application: **Webpage ranking**



Quiz

- Which statement correctly captures the **Markov assumption** used in a Hidden Markov Model?
 - A. The current evidence depends only on the previous evidence.
 - B. The current state depends on all previous states.
 - C. The current state depends only on the previous state.
 - D. The evidence is conditionally independent of the current state.

Quiz

- Which statement correctly captures the **Markov assumption** used in a Hidden Markov Model?
 - A. The current evidence depends only on the previous evidence.
 - B. The current state depends on all previous states.
 - C. **The current state depends only on the previous state.**
 - D. The evidence is conditionally independent of the current state.

Quiz

1. Why can't a static Bayesian network adequately model an agent's beliefs about a changing world?
2. In a temporal model, what are the two main assumptions that simplify inference?

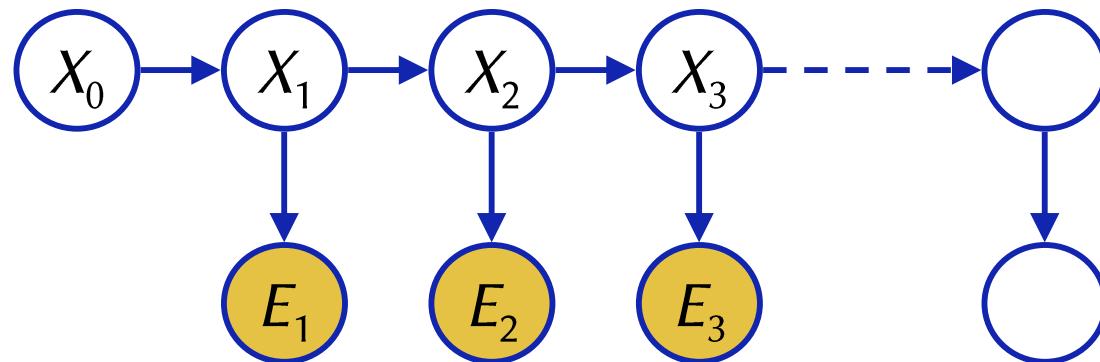
Quiz

1. Why can't a static Bayesian network adequately model an agent's beliefs about a changing world?
 - BN assumes that variables are independent of time. In dynamic environments, the current state depends on previous states, requiring temporal dependence be captured. Dynamic Bayesian Networks (DBNs) or Hidden Markov Models (HMMs) do just that.
2. In a temporal model, what are the two main assumptions that simplify inference?
 - Markov assumption: $P(X_t|X_{1:t-1})=P(X_t|X_{t-1})$
 - Stationary process assumption: transition and sensor models do not change over time, i.e., $P(X_t|X_{t-1})$ and $P(E_t|X_t)$ are constant.

Hidden Markov Models (HMM)



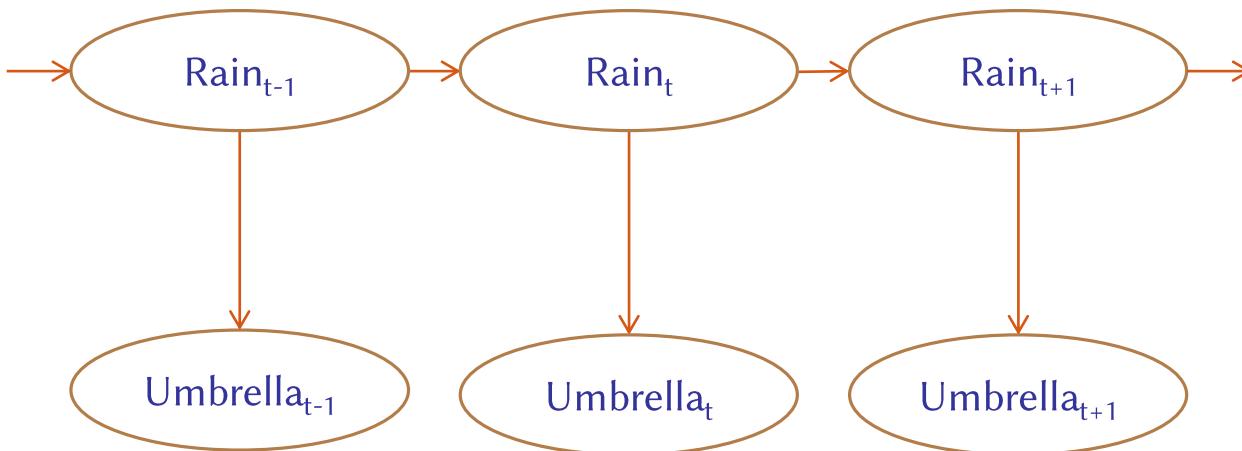
- Underlying Markov chain over belief states X
- You observe evidence E at each time step



Example: Weather HMM



- An HMM is defined by
 - Initial distribution: $P(X_0)$
 - Transition model: $P(X_t | X_{t-1})$
 - Sensor model: $P(E_t | X_t)$



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

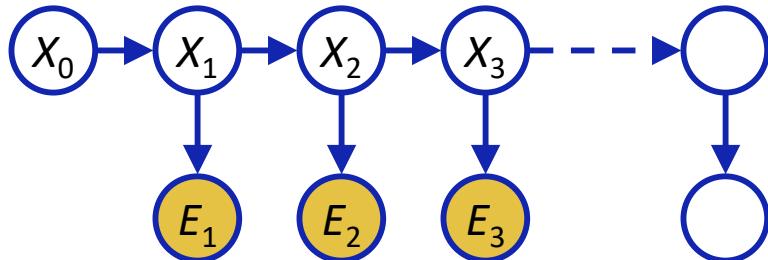
R_t	$P(U_t R_t)$	
	true	false
t	0.9	0.1
f	0.2	0.8

HMM as probability model

- Joint distribution for MM: $P(X_{0:t}) = P(X_0) \prod_{i=1:t} P(X_i | X_{i-1})$
- Joint distribution for HMM:

$$P(X_{0:t}, E_{1:t}) = P(X_0) \prod_{i=1:t} P(X_i | X_{i-1}) P(E_i | X_i)$$

- Future states are independent of the past given the present
- Current evidence is independent of everything else given the current state
- Are evidence variables independent of each other?



Useful notation:

$$X_{a:b} = X_a, X_{a+1}, \dots, X_b$$

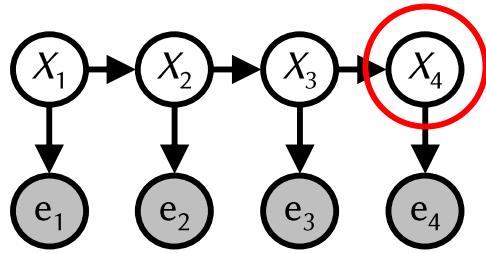
Real HMM Examples

- **Speech recognition HMMs:**
 - Observations are acoustic signals (continuous valued)
 - States are specific positions in specific words (so, tens of thousands)
- **Machine translation HMMs:**
 - Observations are words (tens of thousands)
 - States are translation options
- **Robot tracking:**
 - Observations are range readings (continuous)
 - States are positions on a map (continuous)

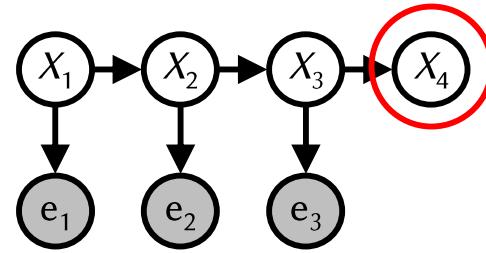
Inference tasks

- Filtering: $P(X_t | e_{1:t})$
 - Find belief state given evidences —input to the decision process of a rational agent, signal processing origin of the term
- Prediction: $P(X_{t+k} | e_{1:t})$ for $k > 0$
 - Evaluation of possible action sequences; like filtering without the evidence
- Smoothing: $P(X_k | e_{1:t})$ for $0 \leq k < t$
 - Better estimate of past states, essential for learning
- Most likely explanation: $\arg \max_{x_{1:t}} P(x_{1:t} | e_{1:t})$
 - Speech recognition, decoding with a noisy channel

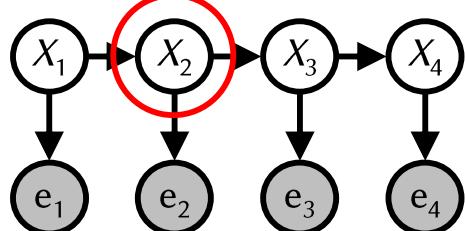
Filtering: $P(X_t|e_{1:t})$



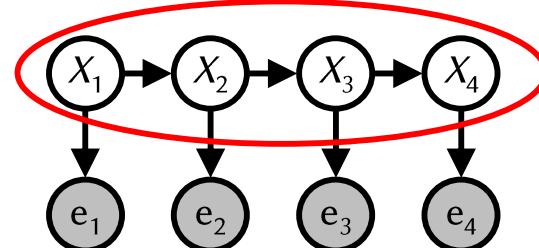
Prediction: $P(X_{t+k}|e_{1:t})$



Smoothing: $P(X_k|e_{1:t}), k < t$



Explanation: $P(X_{1:t}|e_{1:t})$



Filtering / Monitoring

- Filtering, or monitoring, or state estimation, is the task of maintaining the distribution $f_{1:t} = P(X_t|e_{1:t})$ over time
- We start with f_0 in an initial setting, usually uniform
- Filtering is a fundamental task in engineering and science
- The **Kalman filter** (continuous variables, linear dynamics, Gaussian noise) was invented in 1960 and used for trajectory estimation in the Apollo program
 - Core ideas used by Gauss for planetary observations
 - 13,80,000 references on Google Scholar [Nov 2025]

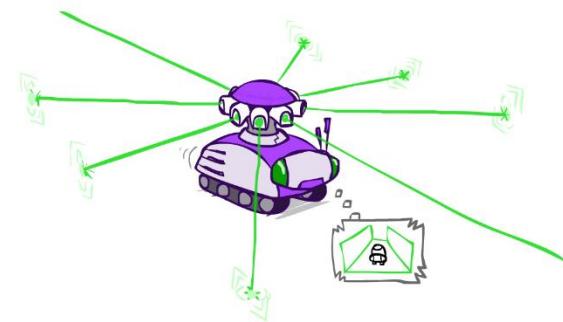
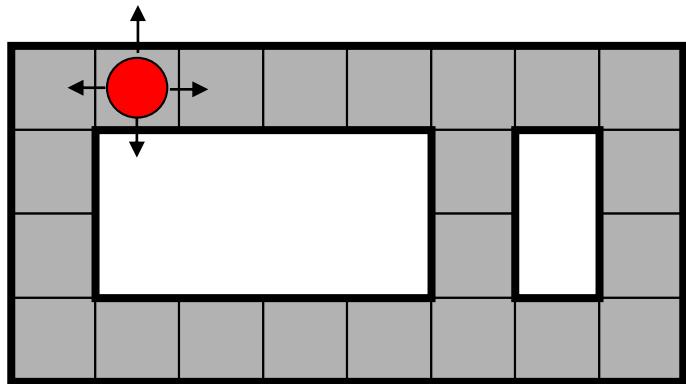
Example: Robot Localization

t=0

Sensor model: four bits for wall/no-wall in each direction, **never more than 1 mistake**

Transition model: action may fail with small probability

*Example from
Michael Pfeiffer*



Prob

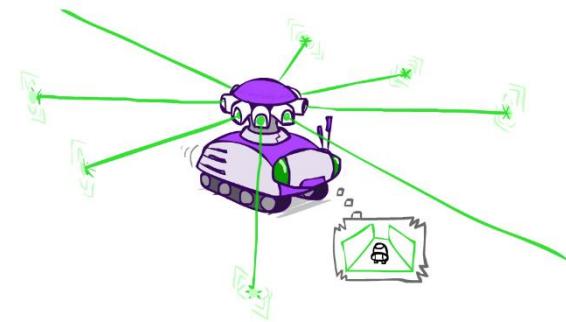
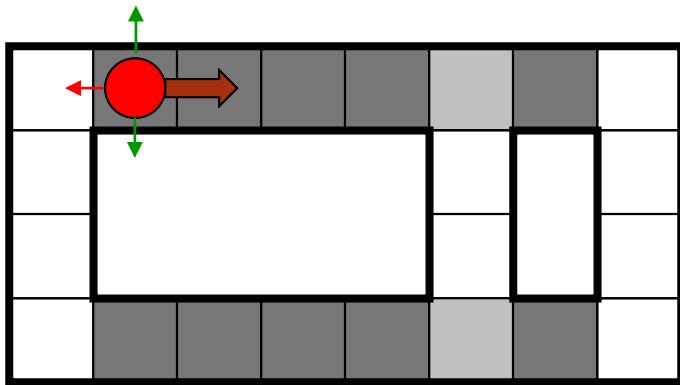
0

1

Example: Robot Localization

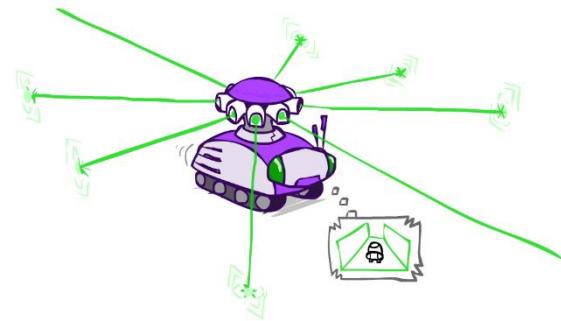
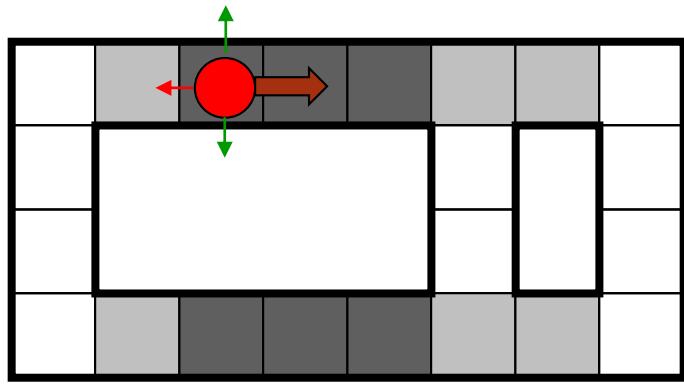
t=1

Lighter grey: was *possible* to get the reading, but *less likely* (required 1 mistake)



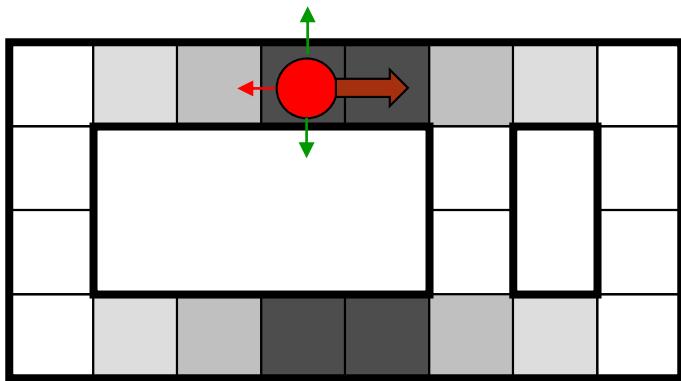
Example: Robot Localization

$t=2$

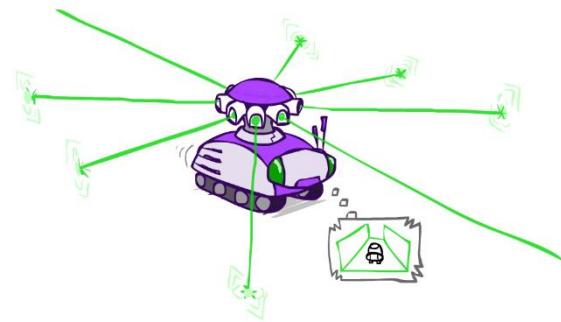


Example: Robot Localization

$t=3$

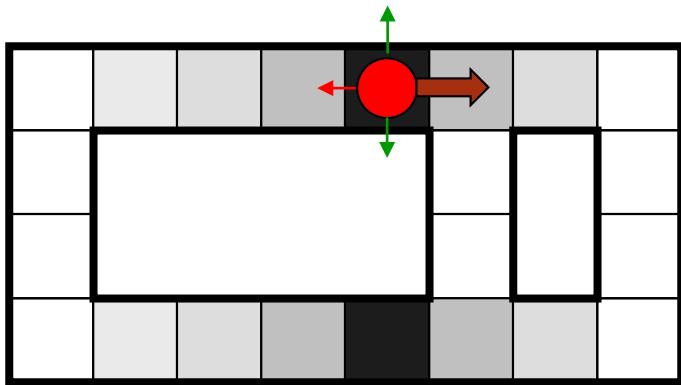


Prob

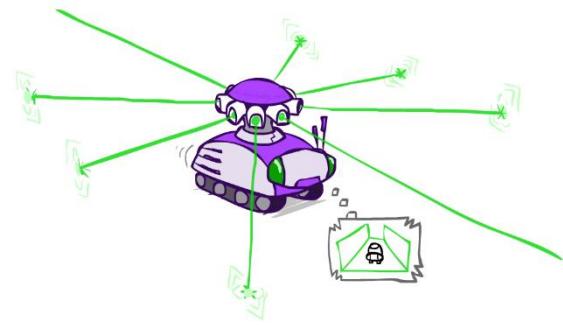


Example: Robot Localization

$t=4$

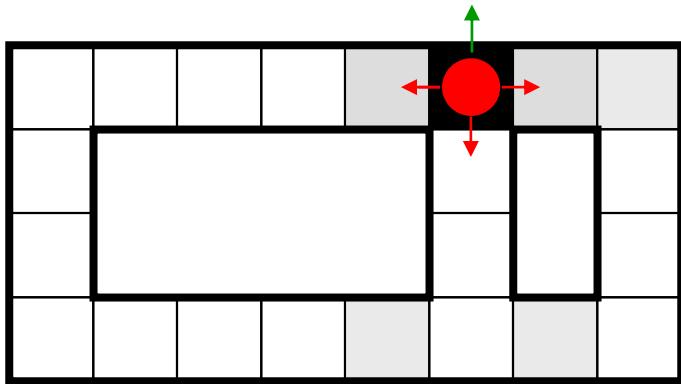


Prob

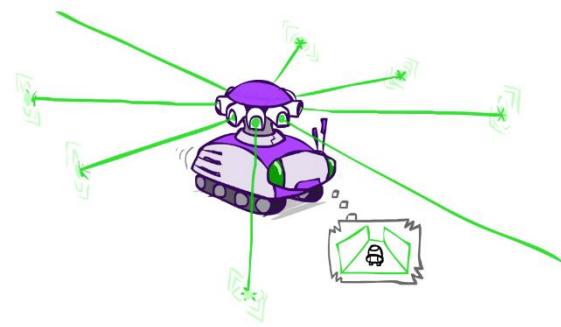


Example: Robot Localization

$t=5$



Prob



Filtering algorithm

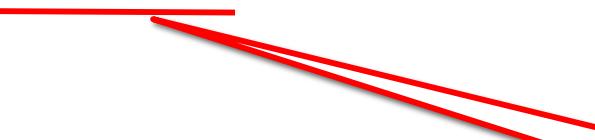
- Aim: devise a **recursive filtering** algorithm of the form
 - $P(X_{t+1}|e_{1:t+1}) = g(e_{t+1}, P(X_t|e_{1:t}))$
 - $P(X_{t+1}|e_{1:t+1}) =$

Filtering algorithm

- Aim: devise a **recursive filtering** algorithm of the form
 - $P(X_{t+1}|e_{1:t+1}) = f(e_{t+1}, P(X_t|e_{1:t}))$
 - $P(X_{t+1}|e_{1:t+1}) = P(X_{t+1}|e_{1:t}, e_{t+1})$

Filtering algorithm

- Aim: devise a **recursive filtering** algorithm of the form
 - $P(X_{t+1}|e_{1:t+1}) = f(e_{t+1}, P(X_t|e_{1:t}))$
 - $$\begin{aligned} P(X_{t+1}|e_{1:t+1}) &= P(X_{t+1}|e_{1:t}, e_{t+1}) \\ &= \alpha P(e_{t+1}|X_{t+1}, e_{1:t}) P(X_{t+1}|e_{1:t}) \\ &= \alpha \underline{P(e_{t+1}|X_{t+1})} P(X_{t+1}|e_{1:t}) \end{aligned}$$



Apply sensor Markov
conditional independence

Filtering algorithm

- Aim: devise a **recursive filtering** algorithm of the form
 - $P(X_{t+1}|e_{1:t+1}) = f(e_{t+1}, P(X_t|e_{1:t}))$
 - $$\begin{aligned} P(X_{t+1}|e_{1:t+1}) &= P(X_{t+1}|e_{1:t}, e_{t+1}) \\ &= \alpha P(e_{t+1}|X_{t+1}, e_{1:t}) P(X_{t+1}|e_{1:t}) \\ &= \alpha P(e_{t+1}|X_{t+1}) P(X_{t+1}|e_{1:t}) \\ &= \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(x_t | e_{1:t}) P(X_{t+1}|x_t, e_{1:t}) \end{aligned}$$

Condition on X_t

Filtering algorithm

- Aim: devise a **recursive filtering** algorithm of the form
 - $P(X_{t+1}|e_{1:t+1}) = f(e_{t+1}, P(X_t|e_{1:t}))$

$$\begin{aligned} \bullet P(X_{t+1}|e_{1:t+1}) &= P(X_{t+1}|e_{1:t}, e_{t+1}) \\ &= \alpha P(e_{t+1}|X_{t+1}, e_{1:t}) P(X_{t+1}|e_{1:t}) \\ &= \alpha P(e_{t+1}|X_{t+1}) P(X_{t+1}|e_{1:t}) \\ &= \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(x_t | e_{1:t}) P(X_{t+1}|x_t, e_{1:t}) \\ &= \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(X_{t+1}|x_t) P(x_t | e_{1:t}) \end{aligned}$$

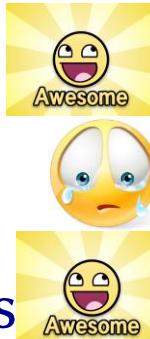
The diagram illustrates the decomposition of the recursive filtering formula. A red bracket under the term \sum_{x_t} is divided into three segments by two red arrows pointing upwards to boxes labeled "Sensor model" and "Transition model". Another red arrow points from the right side of the "Transition model" box to a red speech bubble containing the text "Apply conditional independence". Below the main equation, four red boxes are arranged horizontally: "Normalize" (under the first term), "Sensor model" (under the second term), "Transition model" (under the third term), and "Recursion" (under the fourth term).

Filtering algorithm

$$\bullet P(X_{t+1} | e_{1:t+1}) = \alpha P(e_{t+1} | X_{t+1}) \sum_{x_t} P(x_t | e_{1:t}) P(X_{t+1} | x_t)$$



- $f_{1:t+1} = \text{FORWARD}(f_{1:t}, e_{t+1})$
- Cost per time step: $O(|X|^2)$ where $|X|$ is the number of states
- Time and space costs are **constant**, independent of t
- $O(|X|^2)$ is infeasible for models with many state variables
- We get to invent really cool approximate filtering algorithms



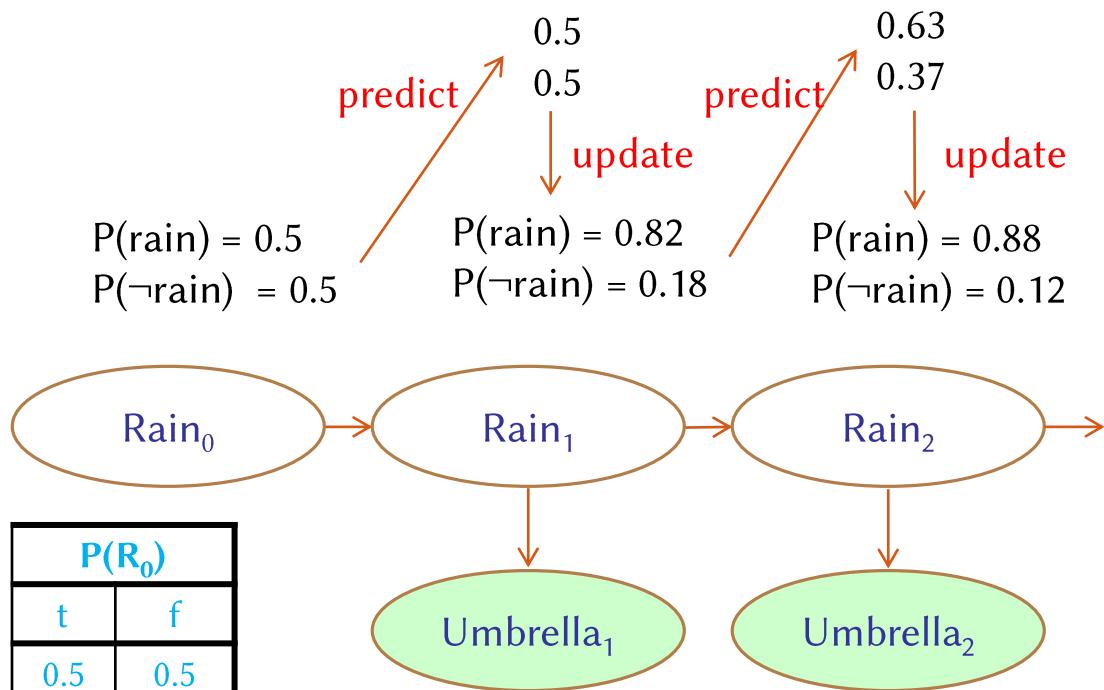
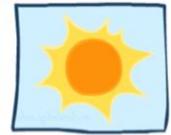
And the same thing in linear algebra

- Transition matrix T , observation matrix O_t
 - Observation matrix has state likelihoods for E_t along diagonal
 - E.g., for $U_1 = \text{true}$, $O_1 = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.9 \end{pmatrix}$
- Filtering algorithm becomes
 - $f_{1:t+1} = \alpha O_{t+1} T^T f_{1:t}$

X_{t-1}	$P(X_t X_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

W_t	$P(U_t W_t)$	
	true	false
sun	0.2	0.8
rain	0.9	0.1

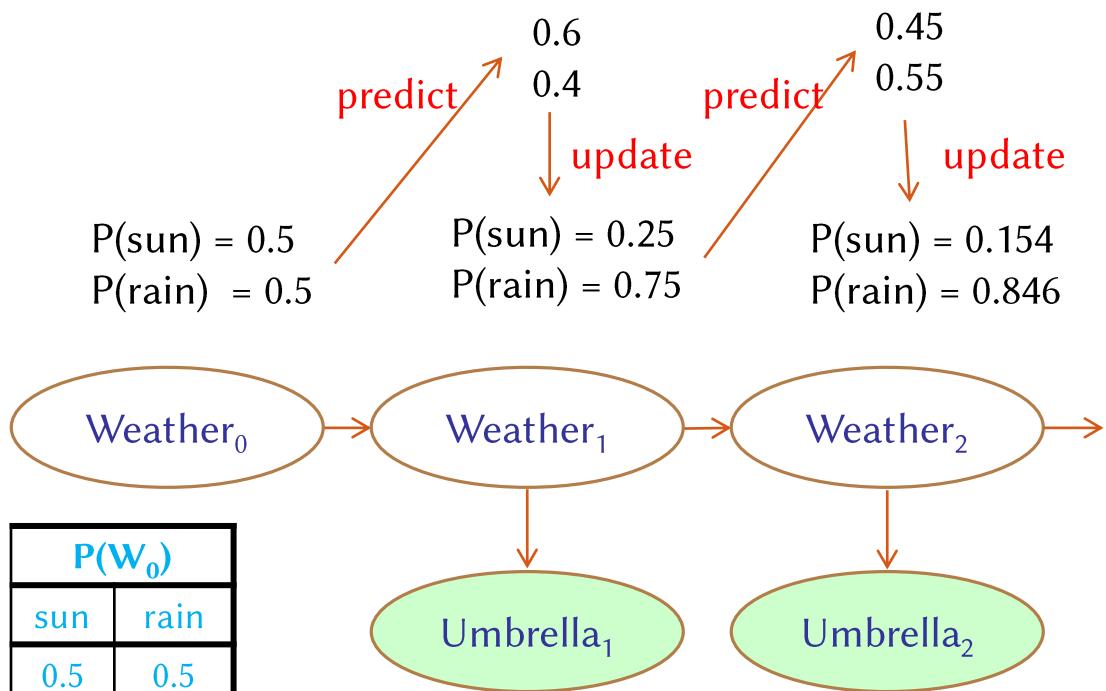
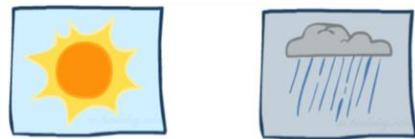
Example: Rain HMM



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

R_t	$P(U_t R_t)$	
	true	false
t	0.9	0.1
f	0.2	0.8

Example 2: Weather HMM



W_{t-1}	$P(W_t W_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

W_t	$P(U_t W_t)$	
	true	false
sun	0.2	0.8
rain	0.9	0.1

Quiz

1. Which of the following best describes the goal of filtering in temporal probabilistic reasoning?
 - A. Estimating the most likely future state of the system.
 - B. Estimating the distribution over the current state given all past evidence.
 - C. Estimating the probability of all previous states given current evidence.
2. In the recursive filtering equation, $P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(x_t | e_{1:t}) P(X_{t+1} | x_t)$, what does the summation over x_t represent?
 - A. Normalization to ensure probabilities sum to 1
 - B. Marginalization over all possible previous states
 - C. Evidence update using the sensor model

Quiz

1. Which of the following best describes the goal of filtering in temporal probabilistic reasoning?
 - A. Estimating the most likely future state of the system.
 - B. Estimating the distribution over the current state given all past evidence.
 - C. Estimating the probability of all previous states given current evidence.
2. In the recursive filtering equation, $P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(x_t | e_{1:t}) P(X_{t+1} | x_t)$, what does the summation over x_t represent?
 - A. Normalization to ensure probabilities sum to 1
 - B. Marginalization over all possible previous states
 - C. Evidence update using the sensor model

Next time

- **Reading:** Chapter 14
- **Assignments:** PS 8
- **Next:** Chapter 16. Utility theory