Introduction to Artificial Intelligence

Lecture 6: Reasoning over time

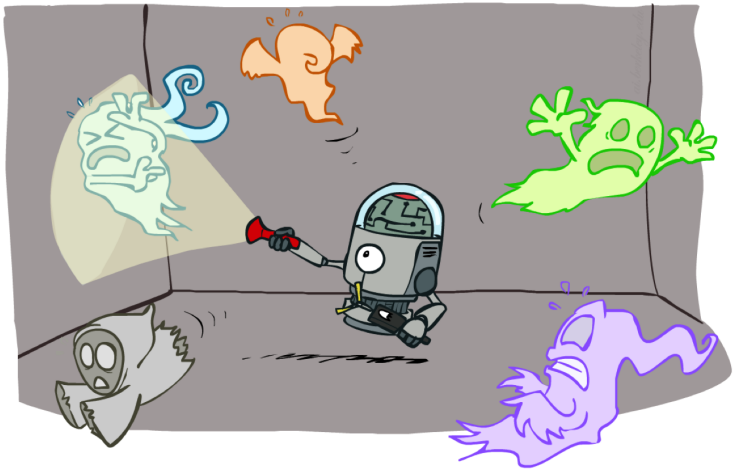
Prof. Gilles Louppe
g.louppe@uliege.be

LIÈGE université

Today

Maintain a belief state about the world, and update it as time passes and evidence is collected.

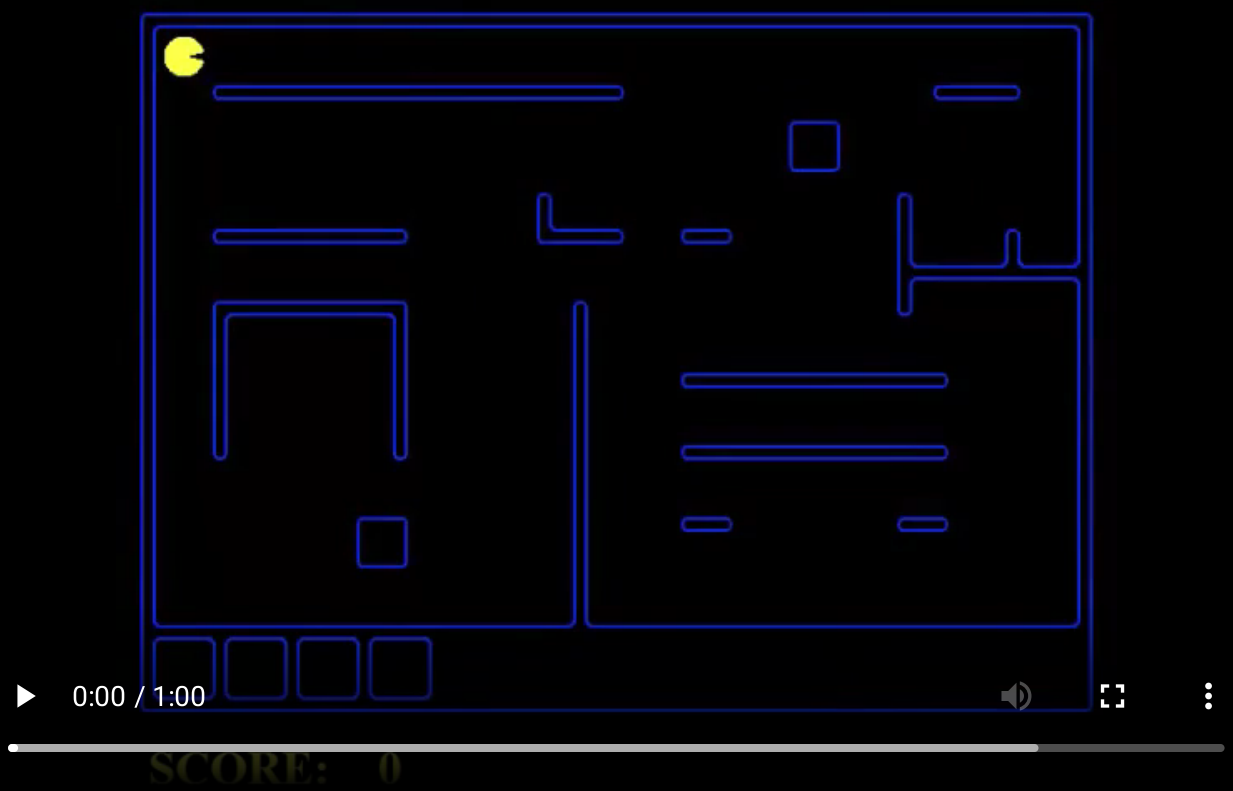
- Markov models
  - Markov processes
  - Inference tasks
  - Hidden Markov models
- Filters
  - Kalman filter
  - Particle filter



Do not overlook this lecture!

—

Credits: CS188, UC Berkeley.



Pacman revenge: How to make good use of the sonar readings?

—

Credits: CS188, UC Berkeley.

Markov models

Modelling the passage of time

We will consider the world as a discrete series of time slices, each of which contains a set of random variables:

- $\mathbf{X}_t$ denotes the set of unobservable state variables at time t.
- $\mathbf{E}_t$ denotes the set of observable evidence variables at time t.

We specify

- a prior $\mathbf{P}(\mathbf{X}_0)$ that defines our inital belief state over hidden state variables,
- a transition model $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1})$ (for $t > 0$) that defines the probability distribution over the latest state variables, given the previous (unobserved) values,
- a sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{0:t}, \mathbf{E}_{0:t-1})$ (for $t > 0$) that defines the probability distribution over the latest evidence variables, given all previous (observed and unobserved) values.

Markov processes

Markov assumption

The current state of the world depends only on its immediate previous state(s), i.e., $\mathbf{X}_t$ depends on only a bounded subset of $\mathbf{X}_{0:t-1}$.

Random processes that satisfy this assumption are called Markov processes or Markov chains.

First-order Markov processes

Markov processes such that

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

i.e., $\mathbf{X}_t$ and $\mathbf{X}_{0:t-2}$ are conditionally independent given $\mathbf{X}_{t-1}$.

Sensor Markov assumption

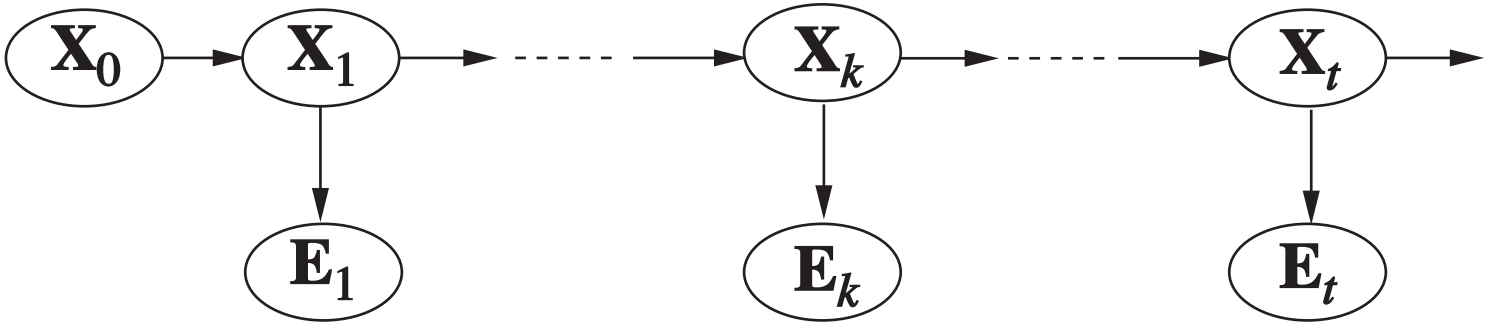
We make a (first-order) sensor Markov assumption

$$\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{0:t}, \mathbf{E}_{0:t-1}) = \mathbf{P}(\mathbf{E}_t|\mathbf{X}_t).$$

Stationarity assumption

The transition and the sensor models are the same for all t (i.e., the laws of physics do not change with time).

Joint distribution

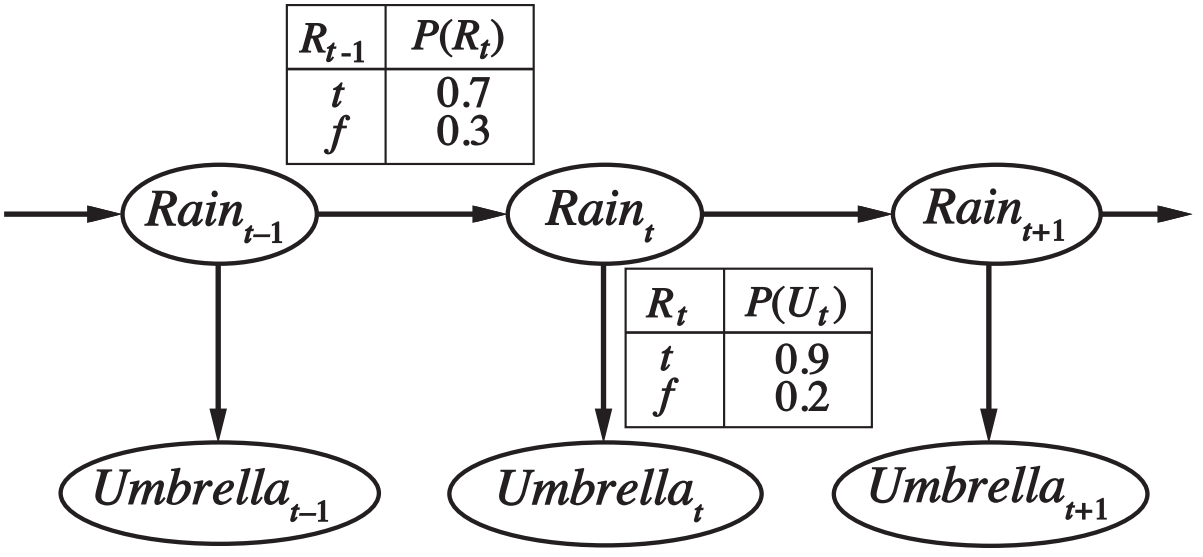


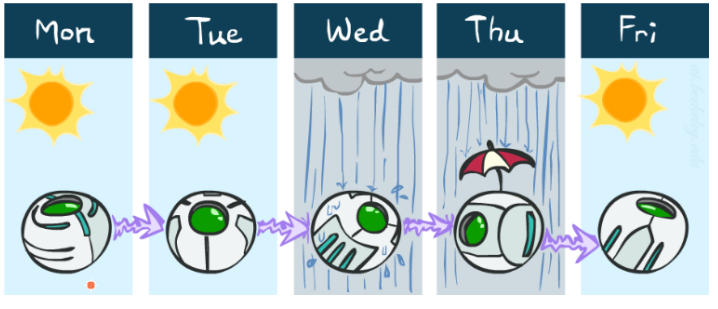
A Markov chain coupled with a sensor model can be represented as a growable Bayesian network, unrolled infinitely through time.

The joint distribution of all its variables up to t is

$$\mathbf{P}(\mathbf{X}_{0:t}, \mathbf{E}_{1:t}) = \mathbf{P}(\mathbf{X}_0) \prod_{i=1}^{t} \mathbf{P}(\mathbf{X}_i|\mathbf{X}_{i-1})\mathbf{P}(\mathbf{E}_i|\mathbf{X}_i).$$

Example: Will you take your umbrella today?





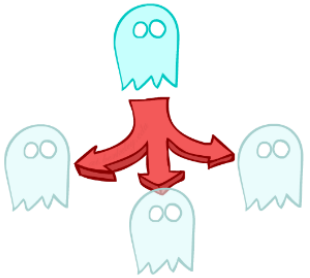
- $\mathbf{P}(\text{Umbrella}_t|\text{Rain}_t)$?
- $\mathbf{P}(\text{Rain}_t|\text{Umbrella}_{0:t-1})$?
- $\mathbf{P}(\text{Rain}_{t+2}|\text{Rain}_t)$?

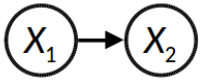
—

Inference tasks

- Prediction: $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$ for $k > 0$
  - Computing the posterior distribution over future states.
  - Used for evaluation of possible action sequences.
- Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$
  - Filtering is what a rational agent does to keep track of the current hidden state $\mathbf{X}_t$, its belief state, so that rational decisions can be made.
- Smoothing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \leq k < t$
  - Computing the posterior distribution over past states.
  - Used for building better estimates, since it incorporates more evidence.
  - Essential for learning.
- Most likely explanation: $\mathbf{arg\,max}_{\mathbf{x}_{1:t}}\ P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$
  - Decoding with a noisy channel, speech recognition, etc.

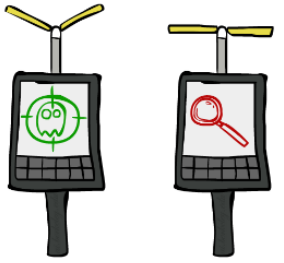
Base cases

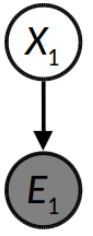




$$\mathbf{P}(\mathbf{X}_2) = \sum_{\mathbf{x}_1} \mathbf{P}(\mathbf{X}_2, \mathbf{x}_1)$$
$$= \sum_{\mathbf{x}_1} P(\mathbf{x}_1)\mathbf{P}(\mathbf{X}_2|\mathbf{x}_1)$$

(Predict) Push $\mathbf{P}(\mathbf{X}_1)$ forward through the transition model.



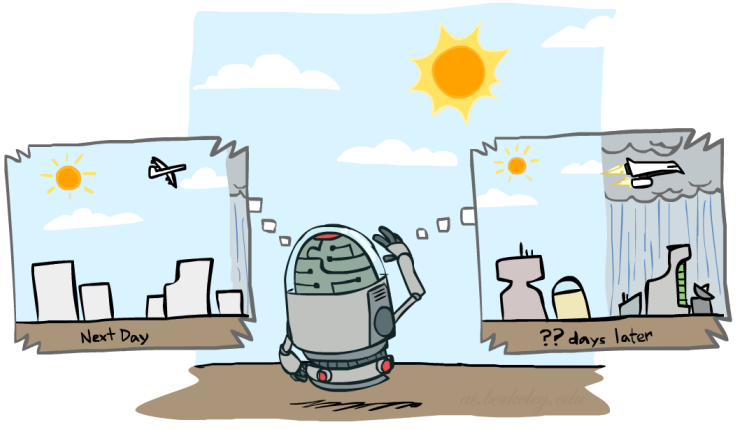


$$\mathbf{P}(\mathbf{X}_1|\mathbf{e}_1) = \frac{\mathbf{P}(\mathbf{e}_1|\mathbf{X}_1)\mathbf{P}(\mathbf{X}_1)}{P(\mathbf{e}_1)}$$
$$\propto \mathbf{P}(\mathbf{e}_1|\mathbf{X}_1)\mathbf{P}(\mathbf{X}_1)$$

(Update) Update $\mathbf{P}(\mathbf{X}_1)$ with the evidence $\mathbf{e}_1$, given the sensor model.

—

Prediction



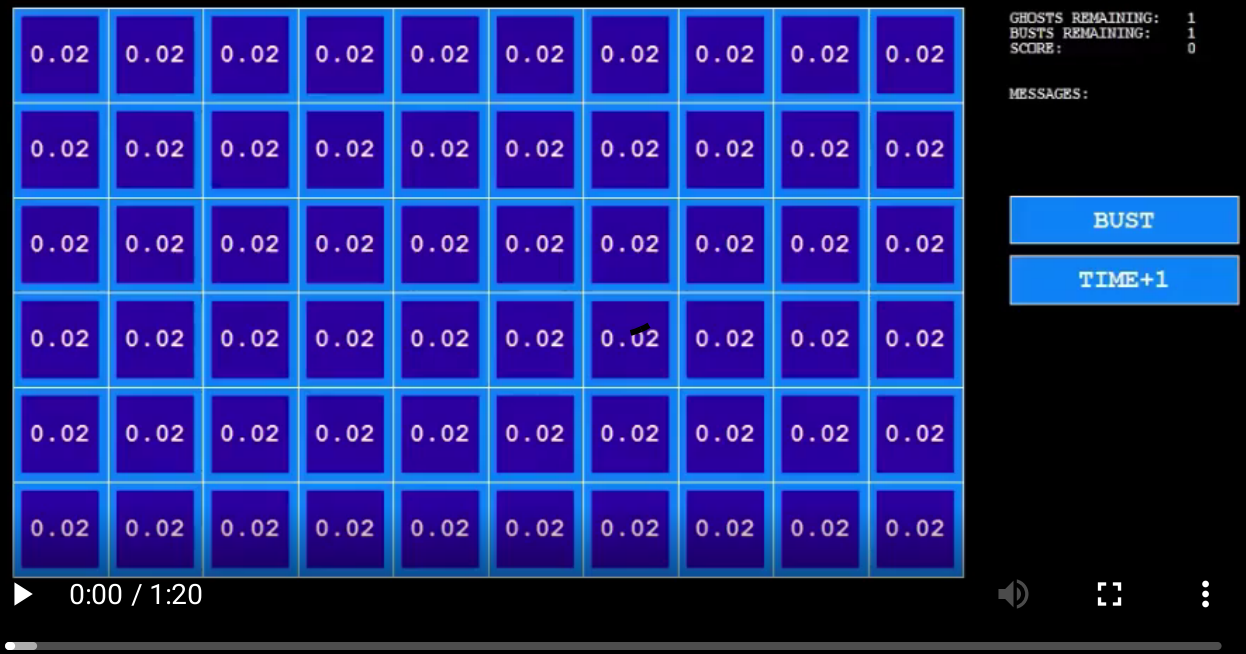
To predict the future $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$:

- Push the prior belief state $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$ through the transition model:

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

- Repeat up to $t+k$, using $\mathbf{P}(\mathbf{X}_{t+k-1}|\mathbf{e}_{1:t})$ to compute $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$.

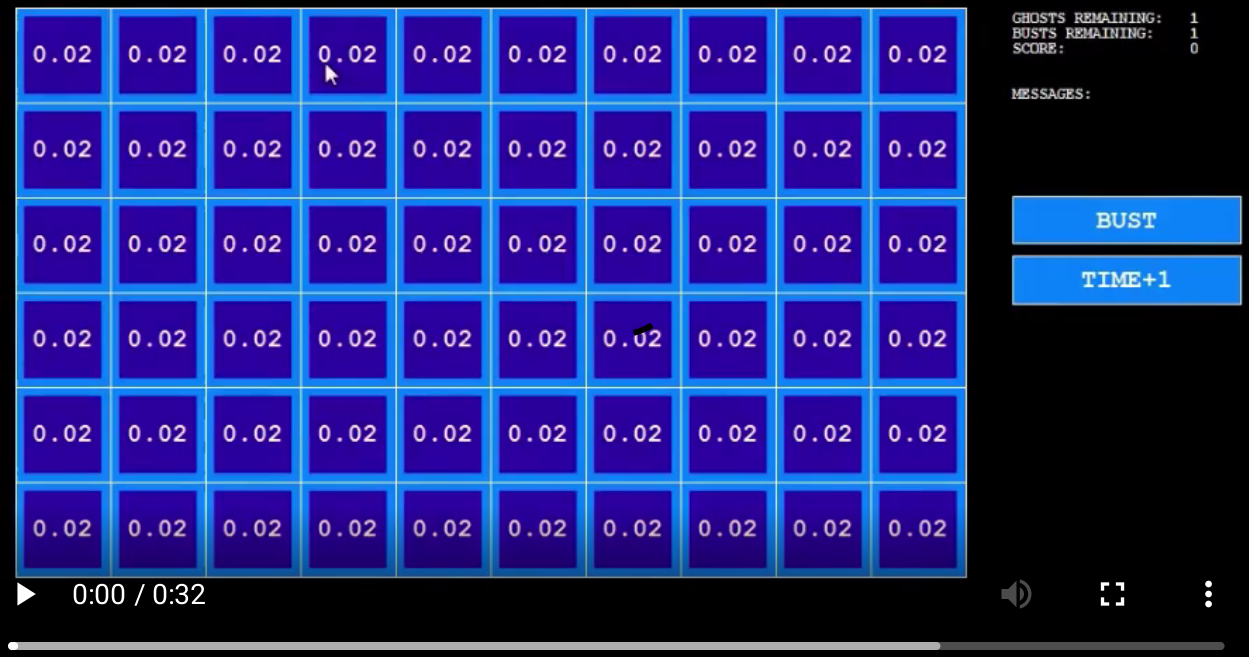
—



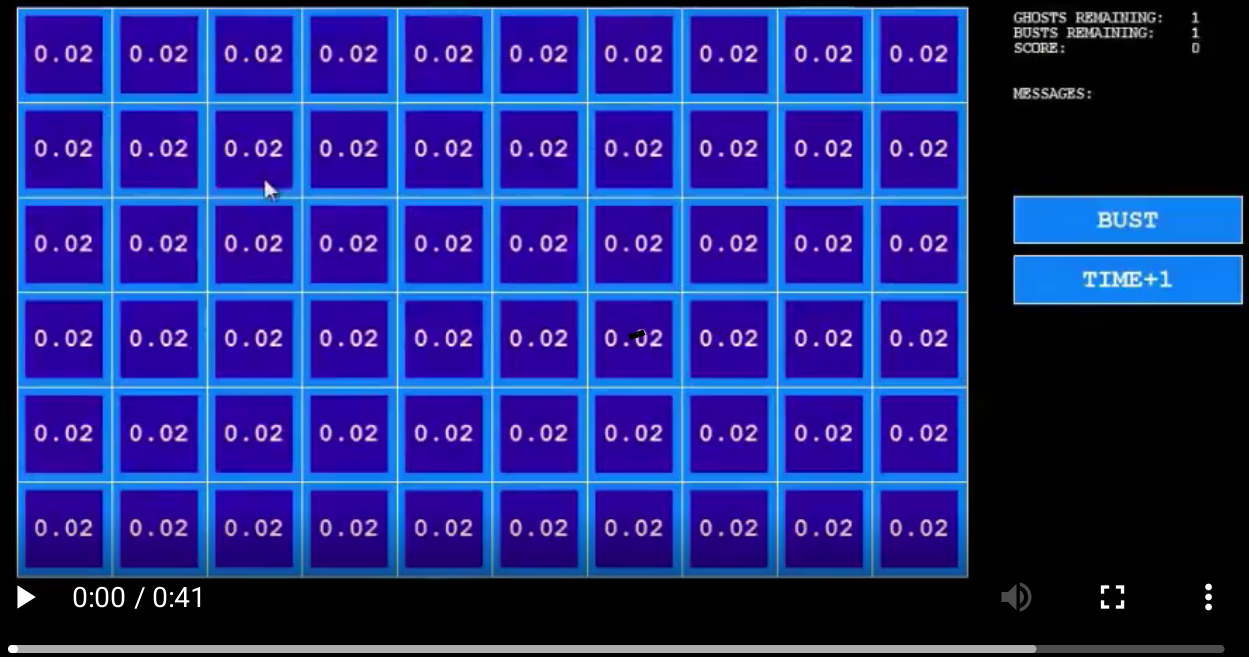
Random dynamics

—

Credits: CS188, UC Berkeley.



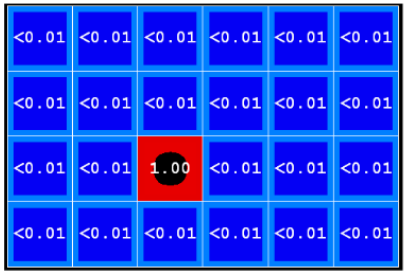
Circular dynamics



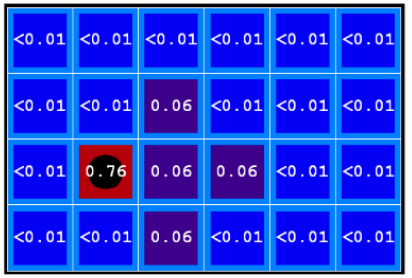
Whirlpool dynamics

—

Credits: CS188, UC Berkeley.

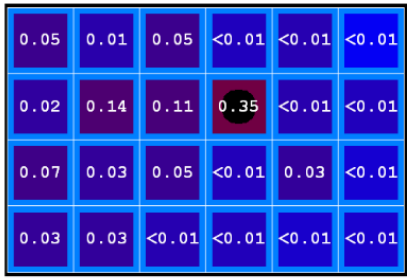


T = 1

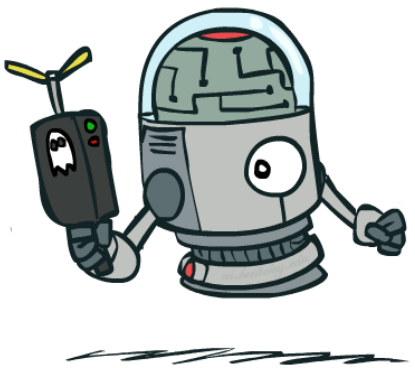


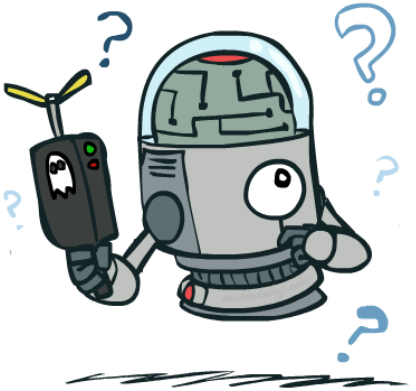
T = 2

...



T = 5





As time passes, uncertainty (usually) increases in the absence of new evidence.

—

Credits: CS188, UC Berkeley.

Stationary distributions

What if $t \to \infty$?

- For most chains, the influence of the initial distribution gets lesser and lesser over time.
- Eventually, the distribution may converge to a fixed distribution, called a stationary distribution.
- This distribution is such that

$$\mathbf{P}(\mathbf{X}_\infty) = \mathbf{P}(\mathbf{X}_{\infty+1}) = \sum_{\mathbf{x}_\infty} \mathbf{P}(\mathbf{X}_{\infty+1} | \mathbf{x}_\infty) P(\mathbf{x}_\infty).$$

$\mathbf{X}_{t-1}$	$\mathbf{X}_t$	P
sun	sun	0.9
sun	rain	0.1
rain	sun	0.3
rain	rain	0.7

Example

$$\begin{aligned} P(\mathbf{X}_\infty = \text{sun}) &= P(\mathbf{X}_{\infty+1} = \text{sun}) \\ &= P(\mathbf{X}_{\infty+1} = \text{sun} | \mathbf{X}_\infty = \text{sun}) P(\mathbf{X}_\infty = \text{sun}) \\ &\quad + P(\mathbf{X}_{\infty+1} = \text{sun} | \mathbf{X}_\infty = \text{rain}) P(\mathbf{X}_\infty = \text{rain}) \\ &= 0.9 P(\mathbf{X}_\infty = \text{sun}) + 0.3 P(\mathbf{X}_\infty = \text{rain}) \end{aligned}$$

Therefore, $P(\mathbf{X}_\infty = \text{sun}) = 3P(\mathbf{X}_\infty = \text{rain})$.

Which implies that $P(\mathbf{X}_\infty = \text{sun}) = \frac{3}{4}$ and $P(\mathbf{X}_\infty = \text{rain}) = \frac{1}{4}$.

Filtering

We want to compute a belief state $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$ and maintain it as time passes and new evidence $\mathbf{e}_{t+1}$ is collected.

This process can be implemented using the **Bayes filter** algorithm, which alternates between prediction and update steps:

- (Predict step): Project the current belief state forward from t to $t+1$ through the transition model.
- (Update step): Update this new state using the evidence $\mathbf{e}_{t+1}$.

—

Formally, the Bayes filter is defined as

$$\begin{aligned}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) &= \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}, \mathbf{e}_{t+1}) \\ &\propto \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}, \mathbf{e}_{1:t})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) \\ &\propto \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) \\ &\propto \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t, \mathbf{e}_{1:t}) P(\mathbf{x}_t|\mathbf{e}_{1:t}) \\ &\propto \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t) P(\mathbf{x}_t|\mathbf{e}_{1:t})\end{aligned}$$

where

- the normalization constant

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

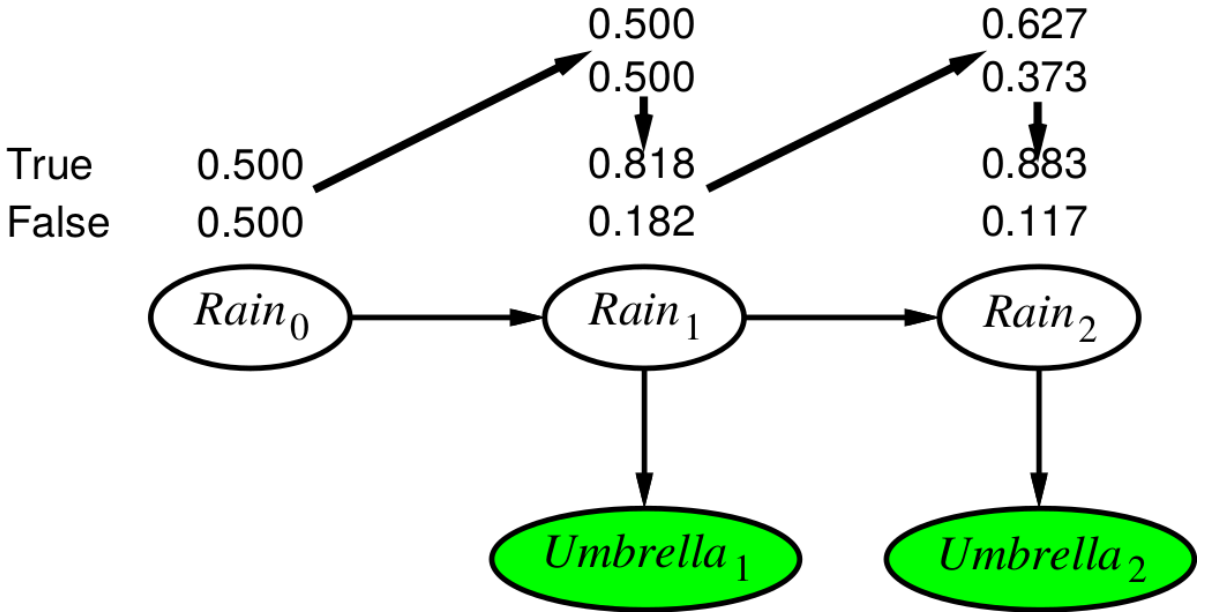
  is used to make probabilities sum to 1;

- in the last expression, the first and second terms are given by the model while the third is obtained recursively.

We can think of $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$ as a message $\mathbf{f}_{1:t}$ that is propagated forward along the sequence, modified by each transition and updated by each new observation.

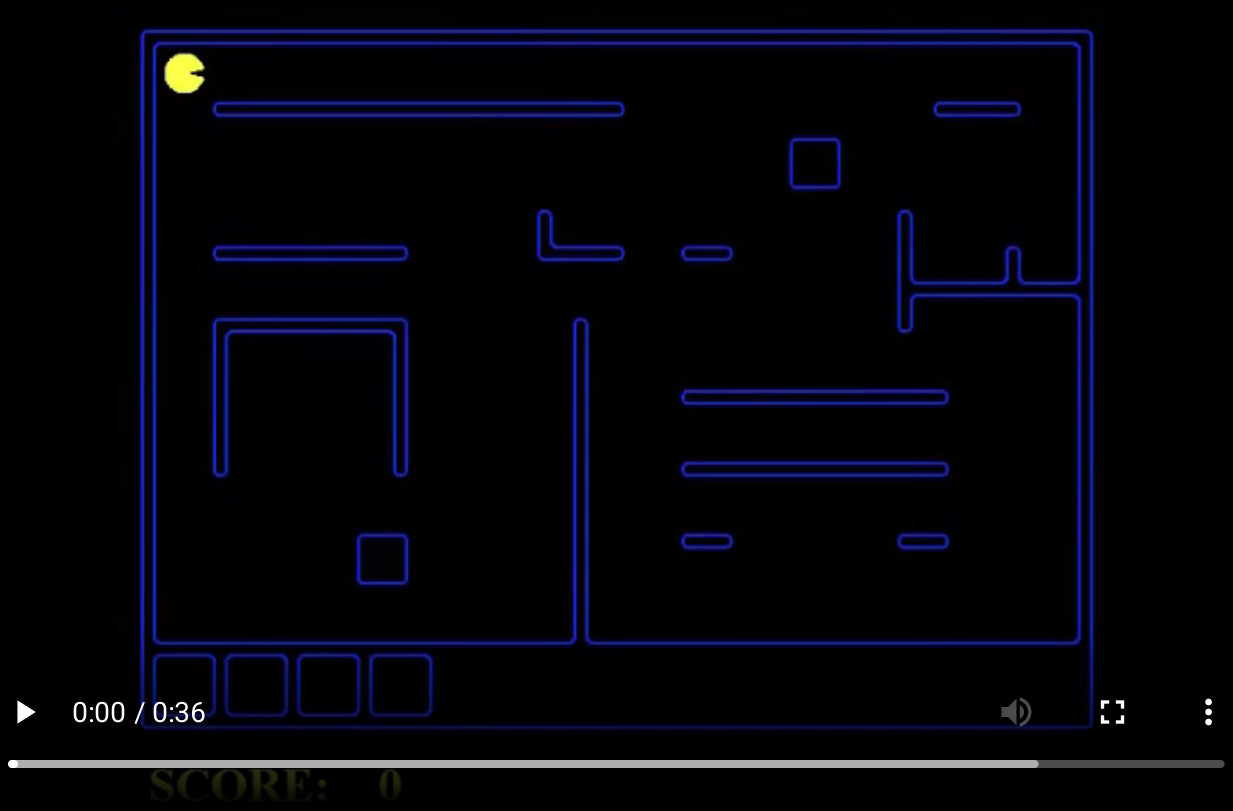
Thus, the process can be implemented as $\mathbf{f}_{1:t+1} \propto \mathbf{forward}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1})$. Its complexity is constant (in time and space) with t.

Example



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

R_t	$P(U_t)$
true	0.9
false	0.2



Ghostbusters with a Bayes filter

—

Credits: CS188, UC Berkeley.

Smoothing

We want to compute $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \leq k < t$.

Dividing the evidence $\mathbf{e}_{1:t}$ into $\mathbf{e}_{1:k}$ and $\mathbf{e}_{k+1:t}$, we have

$$\begin{aligned}\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t}) &= \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k}, \mathbf{e}_{k+1:t}) \\ &\propto \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{e}_{1:k}) \\ &\propto \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k).\end{aligned}$$

Let the backward message $\mathbf{b}_{k+1:t}$ correspond to $\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k)$. Then,

$$\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t}) = \alpha\, \mathbf{f}_{1:k} \times \mathbf{b}_{k+1:t},$$

where $\times$ is a pointwise multiplication of vectors.

This backward message can be computed using backwards recursion:

$$\begin{aligned}
\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) &= \sum_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\
&= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\
&= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k).
\end{aligned}$$

The first and last factors are given by the model. The second factor is obtained recursively. Therefore,

$$\mathbf{b}_{k+1:t} = \mathrm{backward}(\mathbf{b}_{k+2:t}, \mathbf{e}_{k+1}).$$

Forward-backward algorithm

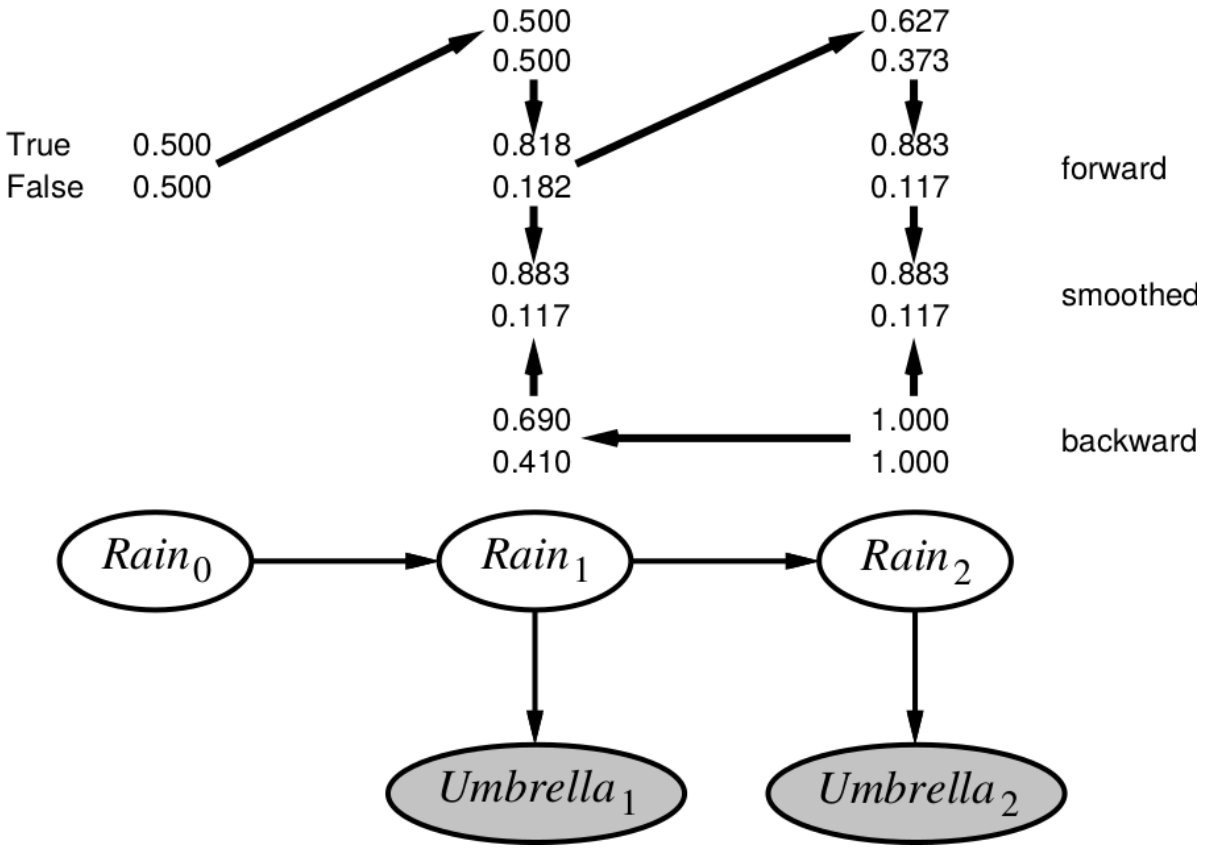
function FORWARD-BACKWARD(**ev**, $prior$) **returns** a vector of probability distributions
  inputs: **ev**, a vector of evidence values for steps $1, \ldots, t$
    $prior$, the prior distribution on the initial state, $\mathbf{P}(\mathbf{X}_0)$
  local variables: **fv**, a vector of forward messages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

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

Complexity:

- Smoothing for a particular time step k takes: $O(t)$
- Smoothing a whole sequence (because of caching): $O(t)$

Example


0.500
0.500
0.627
0.373
True 0.500
False 0.500
0.818
0.182
0.883
0.117
forward
0.883
0.117
0.883
0.117
smoothed
0.690
0.410
1.000
1.000
backward
$Rain_0$
$Rain_1$
$Rain_2$
$Umbrella_1$
$Umbrella_2$

Most likely explanation

—

Suppose that $[\text{true}, \text{true}, \text{false}, \text{true}, \text{true}]$ is the umbrella sequence.

- What is the weather sequence that is the most likely to explain this?
- Among all 2^5 sequences, is there an (efficient) way to find the most likely one?

The most likely sequence **is not** the sequence of the most likely states!

The most likely path to each $\mathbf{x}_{t+1}$, is the most likely path to some $\mathbf{x}_t$ plus one more step. Therefore,

$$\begin{aligned}&\max_{\mathbf{x}_{1:t}} \mathbf{P}(\mathbf{x}_{1:t}, \mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) \\ &\propto \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} (\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) \max_{\mathbf{x}_{1:t-1}} \mathbf{P}(\mathbf{x}_{1:t-1}, \mathbf{x}_t | \mathbf{e}_{1:t})).\end{aligned}$$

This is identical to filtering, except that

- the forward message $\mathbf{f}_{1:t} = \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$ is replaced with

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

  where $\mathbf{m}_{1:t}(i)$ gives the probability of the most likely path to state i.

- The update has its sum replaced by max.

The resulting algorithm is called the Viterbi algorithm, which computes the most likely explanation as

$$\mathbf{m}_{1:t+1} \propto \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)\mathbf{m}_{1:t}.$$

Its complexity is linear in t, the length of the sequence.

Example

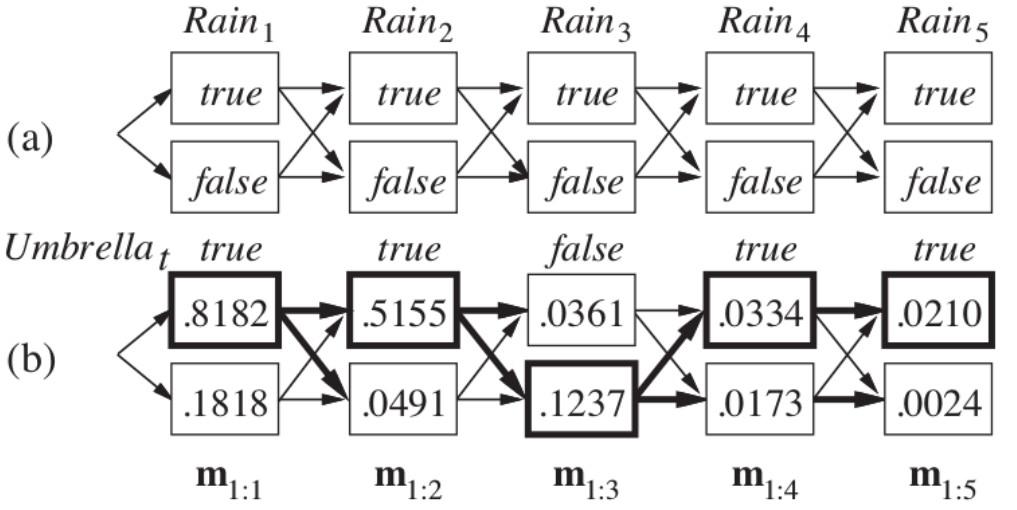


Figure 15.5 (a) Possible state sequences for $Rain_t$ can be viewed as paths through a graph of the possible states at each time step. (States are shown as rectangles to avoid confusion with nodes in a Bayes net.) (b) Operation of the Viterbi algorithm for the umbrella observation sequence $[true, true, false, true, true]$. For each 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 as measured by the product of the preceding sequence probability and the transition probability. Following the bold arrows back from the most likely state in $\mathbf{m}_{1:5}$ gives the most likely sequence.

Hidden Markov models

So far, we described Markov processes over arbitrary sets of state variables $\mathbf{X}_t$ and evidence variables $\mathbf{E}_t$.

- A hidden Markov model (HMM) is a Markov process in which the state $\mathbf{X}_t$ and the evidence $\mathbf{E}_t$ are both single discrete random variables.
  - $\mathbf{X}_t = X_t$, with domain $D_{X_t} = \{1, ..., S\}$
  - $\mathbf{E}_t = E_t$, with domain $D_{E_t} = \{1, ..., R\}$
- This restricted structure allows for a reformulation of the forward-backward algorithm in terms of matrix-vector operations.

Note on terminology

Some authors instead divide Markov models into two classes, depending on the observability of the system state:

- Observable system state: Markov chains
- Partially-observable system state: Hidden Markov models.

We follow here instead the terminology of the textbook, as defined in the previous slide.

Simplified matrix algorithms

- The prior $\mathbf{P}(X_0)$ becomes a (normalized) column vector $\mathbf{f}_0 \in \mathbb{R}_+^S$.
- The transition model $\mathbf{P}(X_t|X_{t-1})$ becomes an $S \times S$ transition matrix $\mathbf{T}$, such that

$$\mathbf{T}_{ij} = P(X_t = j|X_{t-1} = i).$$

- The sensor model $\mathbf{P}(E_t|X_t)$ is defined as an $S \times R$ sensor matrix $\mathbf{B}$, such that

$$\mathbf{B}_{ij} = P(E_t = j|X_t = i).$$

- Let the observation matrix $\mathbf{O}_t$ be a diagonal matrix whose elements corresponds to the column e_t of the sensor matrix $\mathbf{B}$.
- If we use column vectors to represent forward and backward messages, then we have

$$\mathbf{f}_{1:t+1} = \alpha \mathbf{O}_{t+1} \mathbf{T}^T \mathbf{f}_{1:t}$$

$$\mathbf{b}_{k+1:t} = \mathbf{T} \mathbf{O}_{k+1} \mathbf{b}_{k+2:t},$$

  where $\mathbf{b}_{t+1:t}$ is an all-one vector of size S.
- Therefore the forward-backward algorithm needs time $O(S^2 t)$ and space $O(St)$.

Example

Suppose that $[\text{true}, \text{true}, \text{false}, \text{true}, \text{true}]$ is the umbrella sequence.

$$\mathbf{f}_0 = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

$$\mathbf{T} = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix}$$

$$\mathbf{O}_1 = \mathbf{O}_2 = \mathbf{O}_4 = \mathbf{O}_5 = \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix}$$

$$\mathbf{O}_3 = \begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.8 \end{pmatrix}$$

See `code/lecture6-forward-backward.ipynb` for the execution.

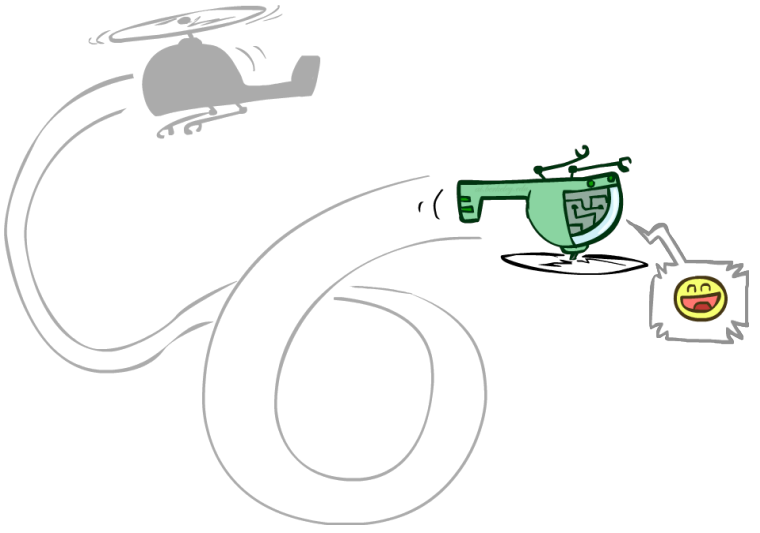
Stationary distribution

The stationary distribution $\mathbf{f}$ of a HMM is a distribution such that

$$\mathbf{f} = \mathbf{T}^T \mathbf{f}.$$

Therefore, the stationary distribution corresponds to a (normalized) eigenvector of the transposed transition matrix with an eigenvalue of 1.

Filters



Suppose we want to track the position and velocity of a robot from noisy observations collected over time.

Formally, we want to estimate continuous state variables such as

- the position $\mathbf{X}_t$ of the robot at time t,
- the velocity $\dot{\mathbf{X}}_t$ of the robot at time t.

We assume discrete time steps.

—

Credits: CS188, UC Berkeley.

Continuous variables

Let $X : \Omega \to D_X$ be a random variable.

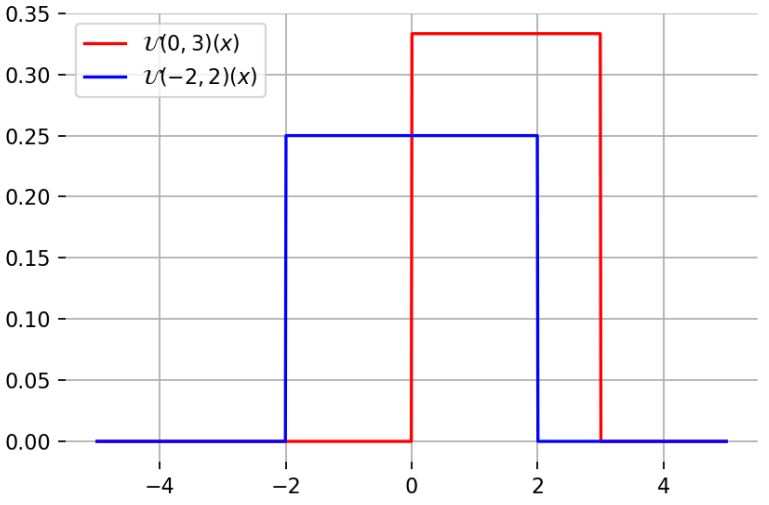
- When D_X is uncountably infinite (e.g., $D_X = \mathbb{R}$), X is called a continuous random variable.
- If X is absolutely continuous, its probability distribution is described by a density function p that assigns a probability to any interval $[a, b] \subseteq D_X$ such that

$$P(a < X \leq b) = \int_a^b p(x)dx,$$

where p is non-negative piecewise continuous and such that

$$\int_{D_X} p(x)dx = 1.$$

Uniform

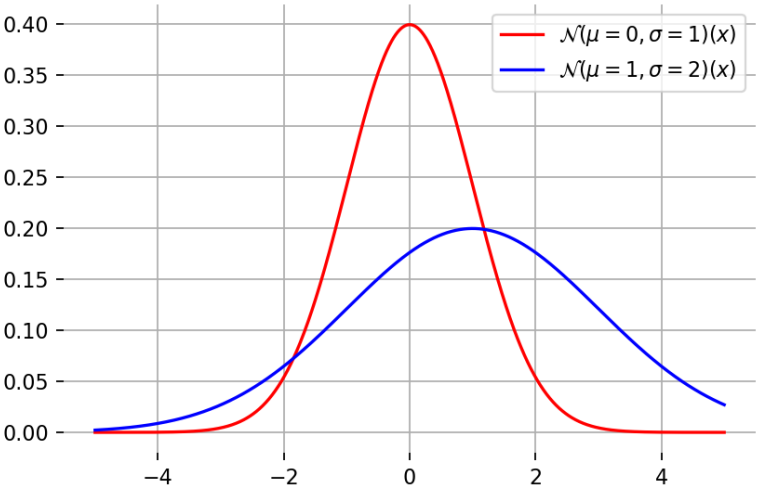


The uniform distribution $\mathcal{U}(a, b)$ is described by the density function

$$p(x) = \begin{cases} \frac{1}{b-a} & \text{if } x \in [a, b] \\ 0 & \text{otherwise} \end{cases}$$

where $a \in \mathbb{R}$ and $b \in \mathbb{R}$ are the bounds of its support.

Normal

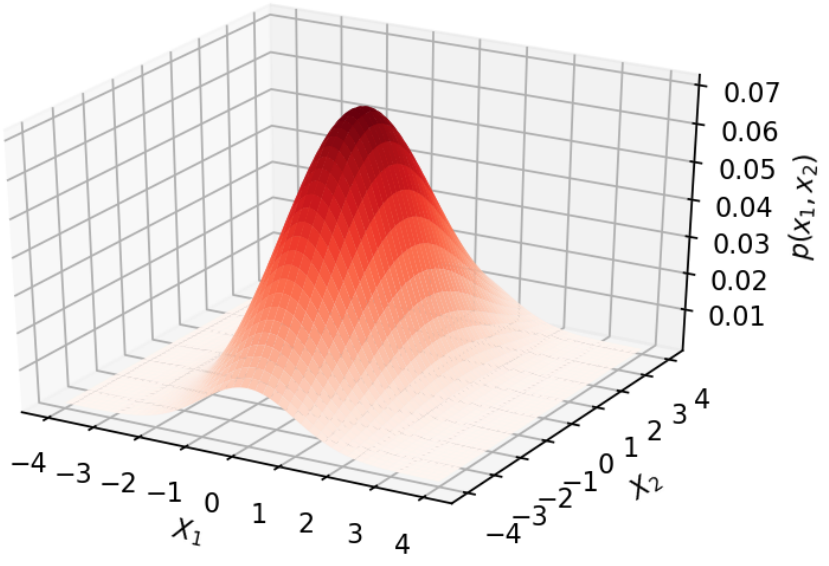


The normal (or Gaussian) distribution $\mathcal{N}(\mu, \sigma)$ is described by the density function

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

where $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$ are its mean and standard deviation parameters.

Multivariate normal



The multivariate normal distribution generalizes to n random variables. Its (joint) density function is defined as

$$p(\mathbf{x} = x_1, ..., x_n) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mathbf{m})\right)$$

where $\mathbf{m} \in \mathbb{R}^n$ and $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is positive semi-definite.

Cheat sheet for Gaussian models (Särkkä, 2013)

If $\mathbf{x}$ and $\mathbf{y}$ have the joint Gaussian distribution

$$p\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} = \mathcal{N}\left(\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} \middle| \begin{pmatrix}\mathbf{a}\\ \mathbf{b}\end{pmatrix}, \begin{pmatrix}\mathbf{A} & \mathbf{C}\\ \mathbf{C}^T & \mathbf{B}\end{pmatrix}\right),$$

then the marginal and conditional distributions of $\mathbf{x}$ and $\mathbf{y}$ are given by

$$\begin{aligned}
p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}|\mathbf{a}, \mathbf{A})\\
p(\mathbf{y}) &= \mathcal{N}(\mathbf{y}|\mathbf{b}, \mathbf{B})\\
p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}(\mathbf{x}|\mathbf{a} + \mathbf{C}\mathbf{B}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^T)\\
p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{y}|\mathbf{b} + \mathbf{C}^T\mathbf{A}^{-1}(\mathbf{x} - \mathbf{a}), \mathbf{B} - \mathbf{C}^T\mathbf{A}^{-1}\mathbf{C}).
\end{aligned}$$

If the random variables $\mathbf{x}$ and $\mathbf{y}$ have Gaussian probability distributions

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{m}, \mathbf{P})$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{H}\mathbf{x} + \mathbf{u}, \mathbf{R}),$$

then the joint distribution of $\mathbf{x}$ and $\mathbf{y}$ is Gaussian with

$$p\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} = \mathcal{N}\left(\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} \middle| \begin{pmatrix}\mathbf{m}\\ \mathbf{H}\mathbf{m} + \mathbf{u}\end{pmatrix}, \begin{pmatrix}\mathbf{P} & \mathbf{P}\mathbf{H}^T\\ \mathbf{H}\mathbf{P} & \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}\end{pmatrix}\right).$$

Continuous Bayes filter

The Bayes filter extends to continuous state and evidence variables $\mathbf{X}_t$ and $\mathbf{E}_t$.

The summations are replaced with integrals and the probability mass functions with probability densities, giving the recursive Bayesian relation

$$p(\mathbf{x}_{t+1}|\mathbf{e}_{1:t+1}) \propto p(\mathbf{e}_{t+1}|\mathbf{x}_{t+1}) \int p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{e}_{1:t}) d\mathbf{x}_t,$$

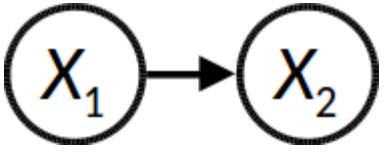
where the normalization constant is

$$Z = \int p(\mathbf{e}_{t+1}|\mathbf{x}_{t+1}) p(\mathbf{x}_{t+1}|\mathbf{e}_{1:t}) d\mathbf{x}_{t+1}.$$

Kalman filter

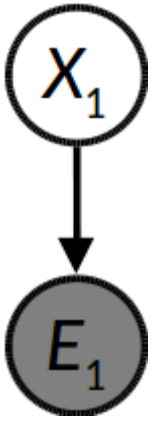
The Kalman filter is a special case of the Bayes filter, which assumes:

- Gaussian prior
- Linear Gaussian transition model
- Linear Gaussian sensor model



$$p(\mathbf{x}_{t+1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t+1}|\mathbf{A}\mathbf{x}_t + \mathbf{b}, \mathbf{\Sigma}_\mathbf{x})$$

Transition model



$$p(\mathbf{e}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{e}_t|\mathbf{C}\mathbf{x}_t + \mathbf{d}, \mathbf{\Sigma}_\mathbf{e})$$

Sensor model

Filtering Gaussian distributions

- *Prediction step:*

  If the distribution $p(\mathbf{x}_t|\mathbf{e}_{1:t})$ is Gaussian and the transition model $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$ is linear Gaussian, then the one-step predicted distribution given by

  $$p(\mathbf{x}_{t+1}|\mathbf{e}_{1:t}) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{e}_{1:t})d\mathbf{x}_t$$

  is also a Gaussian distribution.

- *Update step:*

  If the prediction $p(\mathbf{x}_{t+1}|\mathbf{e}_{1:t})$ is Gaussian and the sensor model $p(\mathbf{e}_{t+1}|\mathbf{x}_{t+1})$ is linear Gaussian, then after conditioning on new evidence, the updated distribution

  $$p(\mathbf{x}_{t+1}|\mathbf{e}_{1:t+1}) \propto p(\mathbf{e}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{e}_{1:t})$$

  is also a Gaussian distribution.

Therefore, for the Kalman filter, $p(\mathbf{x}_t|\mathbf{e}_{1:t})$ is a multivariate Gaussian distribution $\mathcal{N}(\mathbf{x}_t|\mu_t, \mathbf{\Sigma}_t)$ for all t.

- Filtering reduces to the computation of the parameters μ_t and $\mathbf{\Sigma}_t$.
- By contrast, for general (non-linear, non-Gaussian) processes, the description of the posterior grows unboundedly as $t \to \infty$.

1D example

Gaussian random walk:

- Gaussian prior:

$$p(x_0) = \mathcal{N}(x_0|\mu_0, \sigma_0^2)$$

- The transition model adds random perturbations of constant variance:

$$p(x_{t+1}|x_t) = \mathcal{N}(x_{t+1}|x_t, \sigma_x^2)$$

- The sensor model yields measurements with Gaussian noise of constant variance:

$$p(e_t|x_t) = \mathcal{N}(e_t|x_t, \sigma_e^2)$$

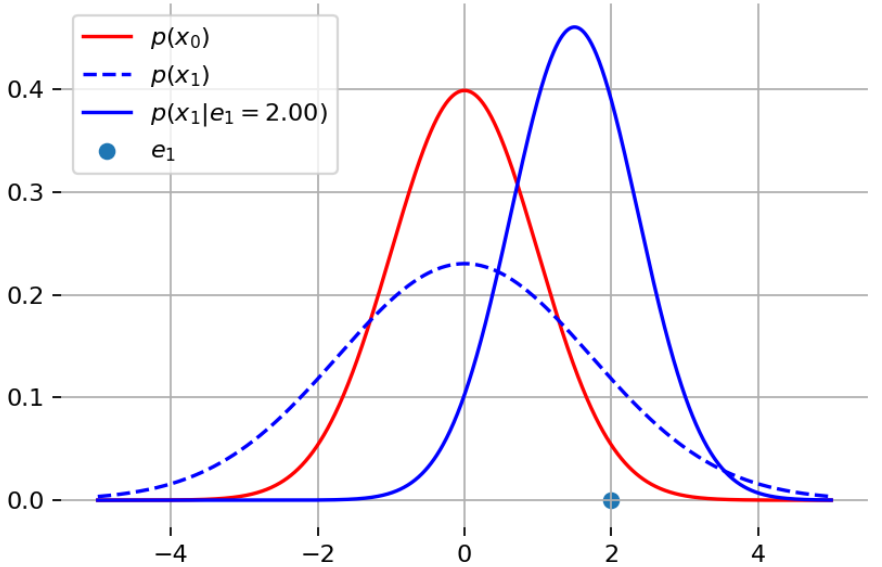
The one-step predicted distribution is given by

$$
\begin{aligned}
p(x_1) &= \int p(x_1|x_0)p(x_0)dx_0 \\
&\propto \int \exp\left(-\frac{1}{2}\frac{(x_1-x_0)^2}{\sigma_x^2}\right)\exp\left(-\frac{1}{2}\frac{(x_0-\mu_0)^2}{\sigma_0^2}\right)dx_0 \\
&\propto \int \exp\left(-\frac{1}{2}\frac{\sigma_0^2(x_1-x_0)^2+\sigma_x^2(x_0-\mu_0)^2}{\sigma_0^2\sigma_x^2}\right)dx_0 \\
&\dots \text{ (simplify by completing the square)} \\
&\propto \exp\left(-\frac{1}{2}\frac{(x_1-\mu_0)^2}{\sigma_0^2+\sigma_x^2}\right) \\
&= \mathcal{N}(x_1|\mu_0, \sigma_0^2+\sigma_x^2)
\end{aligned}
$$

Note that the same result can be obtained by using instead the Gaussian models identities.

For the update step, we need to condition on the observation at the first time step:

$$
\begin{aligned}
p(x_1|e_1) &\propto p(e_1|x_1)p(x_1) \\
&\propto \exp\left(-\frac{1}{2}\frac{(e_1-x_1)^2}{\sigma_e^2}\right)\exp\left(-\frac{1}{2}\frac{(x_1-\mu_0)^2}{\sigma_0^2+\sigma_x^2}\right) \\
&\propto \exp\left(-\frac{1}{2}\frac{\left(x_1-\frac{(\sigma_0^2+\sigma_x^2)e_1+\sigma_e^2\mu_0}{\sigma_0^2+\sigma_x^2+\sigma_e^2}\right)^2}{\frac{(\sigma_0^2+\sigma_x^2)\sigma_e^2}{\sigma_0^2+\sigma_x^2+\sigma_e^2}}\right) \\
&= \mathcal{N}\left(x_1\middle|\frac{(\sigma_0^2+\sigma_x^2)e_1+\sigma_e^2\mu_0}{\sigma_0^2+\sigma_x^2+\sigma_e^2}, \frac{(\sigma_0^2+\sigma_x^2)\sigma_e^2}{\sigma_0^2+\sigma_x^2+\sigma_e^2}\right)
\end{aligned}
$$



In summary, the update equations given a new evidence e_{t+1} are:

$$\mu_{t+1} = \frac{(\sigma_t^2 + \sigma_x^2)e_{t+1} + \sigma_e^2 \mu_t}{\sigma_t^2 + \sigma_x^2 + \sigma_e^2}$$

$$\sigma_{t+1}^2 = \frac{(\sigma_t^2 + \sigma_x^2)\sigma_e^2}{\sigma_t^2 + \sigma_x^2 + \sigma_e^2}$$

Kalman update equations

The same derivations generalize to multivariate normal distributions. Assuming the transition and sensor models

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t+1}|\mathbf{F}\mathbf{x}_t, \boldsymbol{\Sigma}_\mathbf{x})$$
$$p(\mathbf{e}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{e}_t|\mathbf{H}\mathbf{x}_t, \boldsymbol{\Sigma}_\mathbf{e}),$$

the prediction step yields

$$\begin{aligned} p(\mathbf{x}_{t+1}|\mathbf{e}_{1:t}) &= \int p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{e}_{1:t}) d\mathbf{x}_t \\ &= \mathcal{N}(\mathbf{x}_{t+1}|\mu_{t+1}^-, \boldsymbol{\Sigma}_{t+1}^-) \end{aligned}$$

where

$$\mu_{t+1}^- = \mathbf{F}\mu_t$$
$$\boldsymbol{\Sigma}_{t+1}^- = \mathbf{F}\boldsymbol{\Sigma}_t\mathbf{F}^T + \boldsymbol{\Sigma}_\mathbf{x}.$$

The update step yields the final Kalman filter equations,

$$p(\mathbf{x}_{t+1}|\mathbf{e}_{1:t+1}) = \mathcal{N}(\mathbf{x}_{t+1}|\mu_{t+1}, \mathbf{\Sigma}_{t+1})$$

where

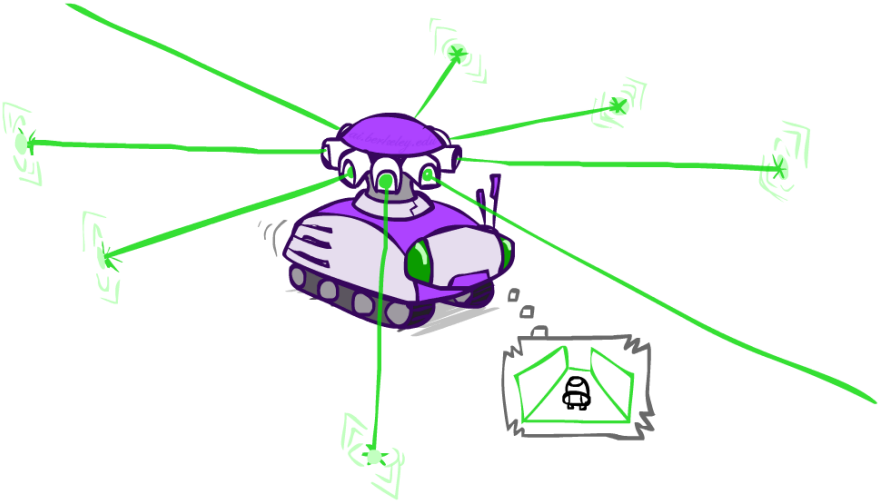
$$\begin{aligned}
\mu_{t+1} &= \mu_{t+1}^{-} + \mathbf{K}_{t+1}(\mathbf{e}_{t+1} - \mathbf{H}\mu_{t+1}^{-}) \\
\mathbf{\Sigma}_{t+1} &= (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H})\mathbf{\Sigma}_{t+1}^{-} \\
\mathbf{K}_{t+1} &= \mathbf{\Sigma}_{t+1}^{-}\mathbf{H}^{T}(\mathbf{H}\mathbf{\Sigma}_{t+1}^{-}\mathbf{H}^{T} + \mathbf{\Sigma}_{\mathbf{e}})^{-1}
\end{aligned}$$

in which $\mathbf{K}_{t+1}$ is called the **Kalman gain** and represents the relative weight given to the new observation versus the prediction.

Particle filter

When the transition and sensor models are non-linear and/or non-Gaussian, the Kalman filter is not applicable.

The **particle filter** is a sampling-based approximate inference algorithm for general continuous state-space models.



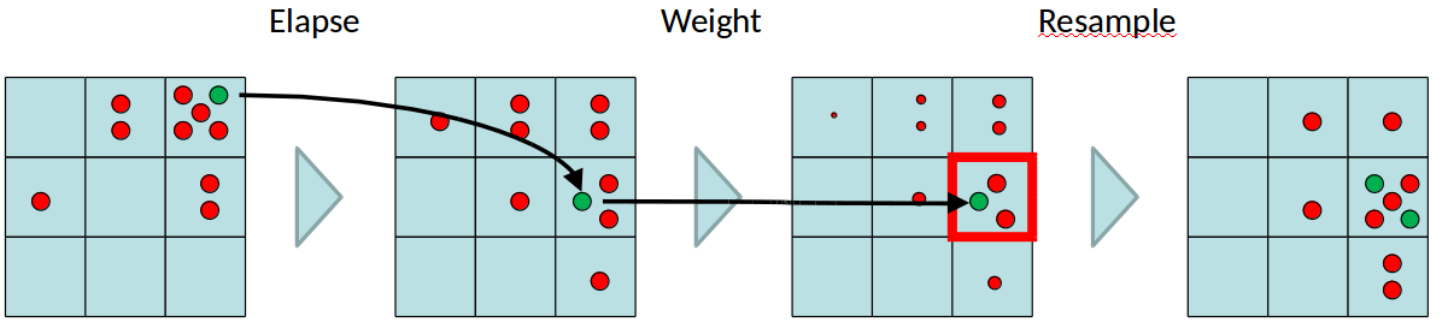
—

Core idea

- A particle filter approximates the filtering distribution $p(\mathbf{x}_t | \mathbf{e}_{1:t})$ using a set $\{\mathbf{x}_t^i\}$ of samples called **particles**.
- The particles are propagated over time using the transition model.
- The particles are weighted according to the likelihood of the evidence given the particle's state.
- Low-weight particles are discarded, and high-weight particles are duplicated (resampling).

This scales to high dimensions!

Update cycle



—

Credits: CS188, UC Berkeley.

function PARTICLE-FILTERING($\mathbf{e}, N, dbn$) **returns** a set of samples for the next time step

inputs: $\mathbf{e}$, the new incoming evidence
N, the number of samples to be maintained
dbn, a DBN with prior $\mathbf{P}(\mathbf{X}_0)$, transition model $\mathbf{P}(\mathbf{X}_1|\mathbf{X}_0)$, sensor model $\mathbf{P}(\mathbf{E}_1|\mathbf{X}_1)$

persistent: S, a vector of samples of size N, initially generated from $\mathbf{P}(\mathbf{X}_0)$

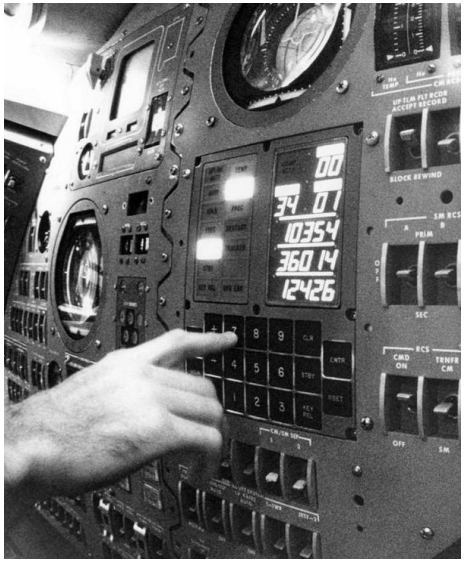
local variables: W, a vector of weights of size N

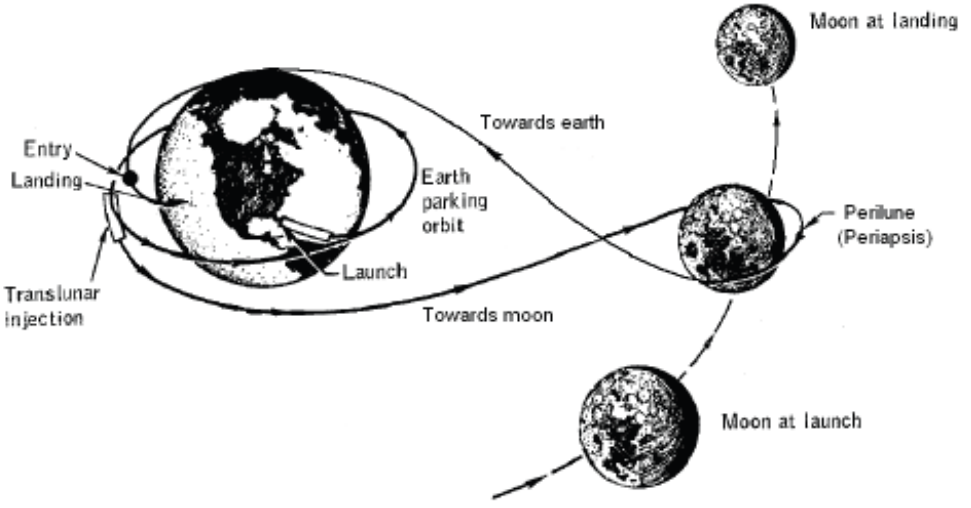
for i = 1 to N **do**
$S[i] \leftarrow$ sample from $\mathbf{P}(\mathbf{X}_1 \mid \mathbf{X}_0 = S[i])$ /* step 1 */
$W[i] \leftarrow \mathbf{P}(\mathbf{e} \mid \mathbf{X}_1 = S[i])$ /* step 2 */
$S \leftarrow$ WEIGHTED-SAMPLE-WITH-REPLACEMENT(N, S, W) /* step 3 */
return S

Applications

Apollo guidance computer

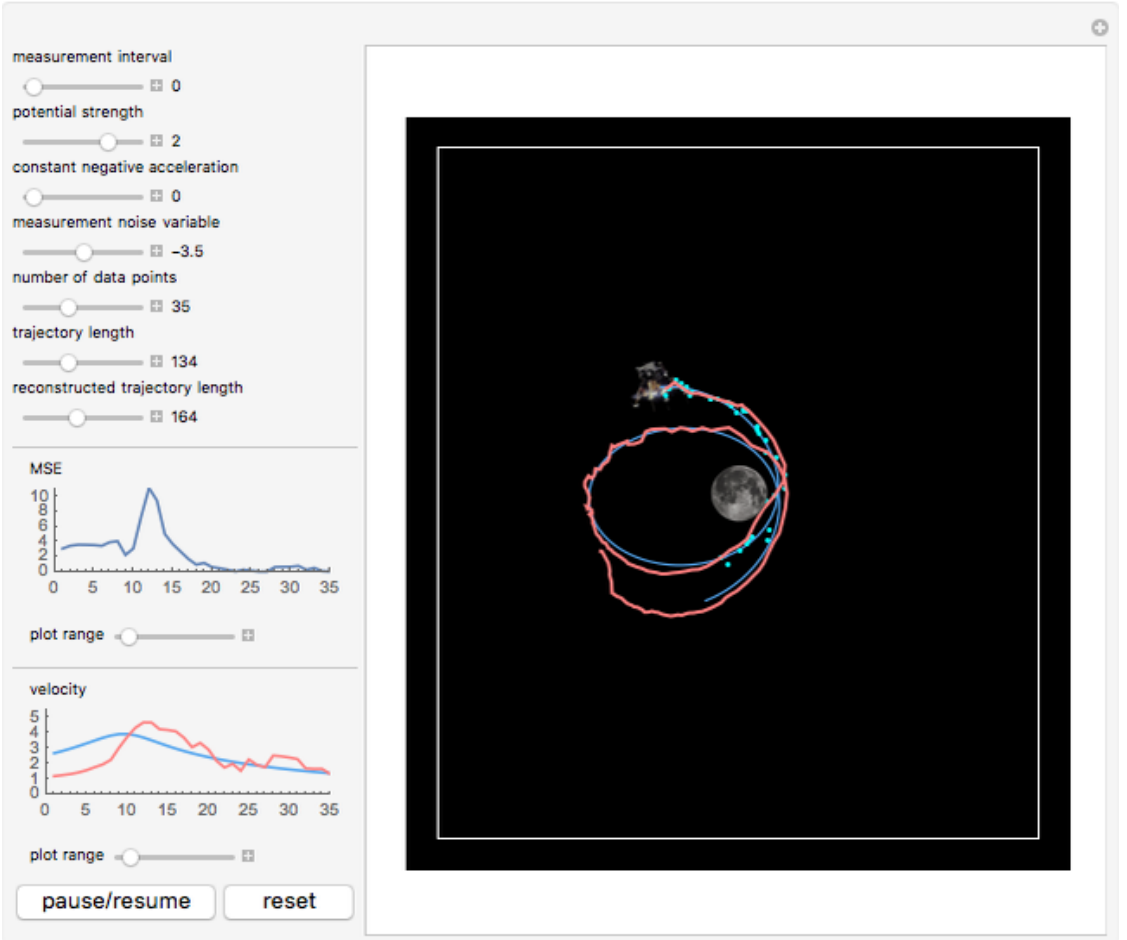
The Apollo Guidance Computer used a Kalman filter to estimate the position of the spacecraft. The Kalman filter was used to merge new data with past position measurements to produce an optimal position estimate of the spacecraft.





—

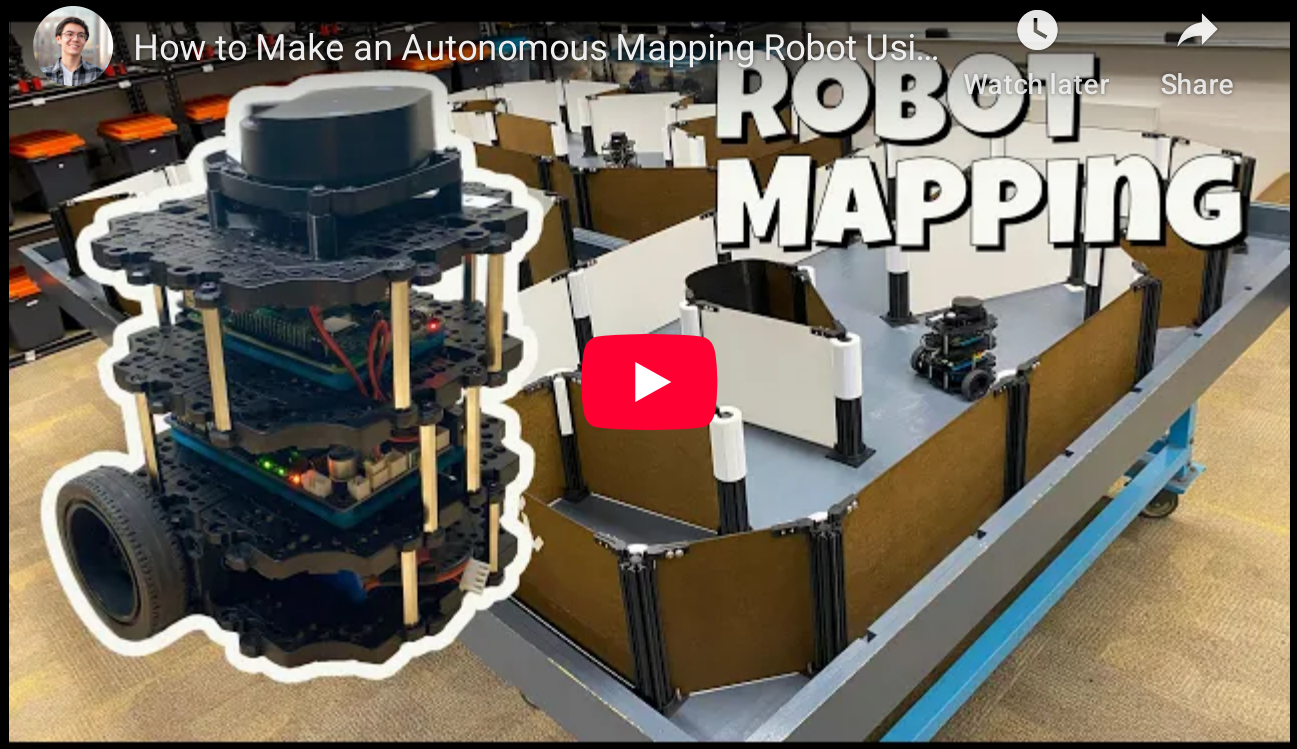
Credits: Apollo-11 source code

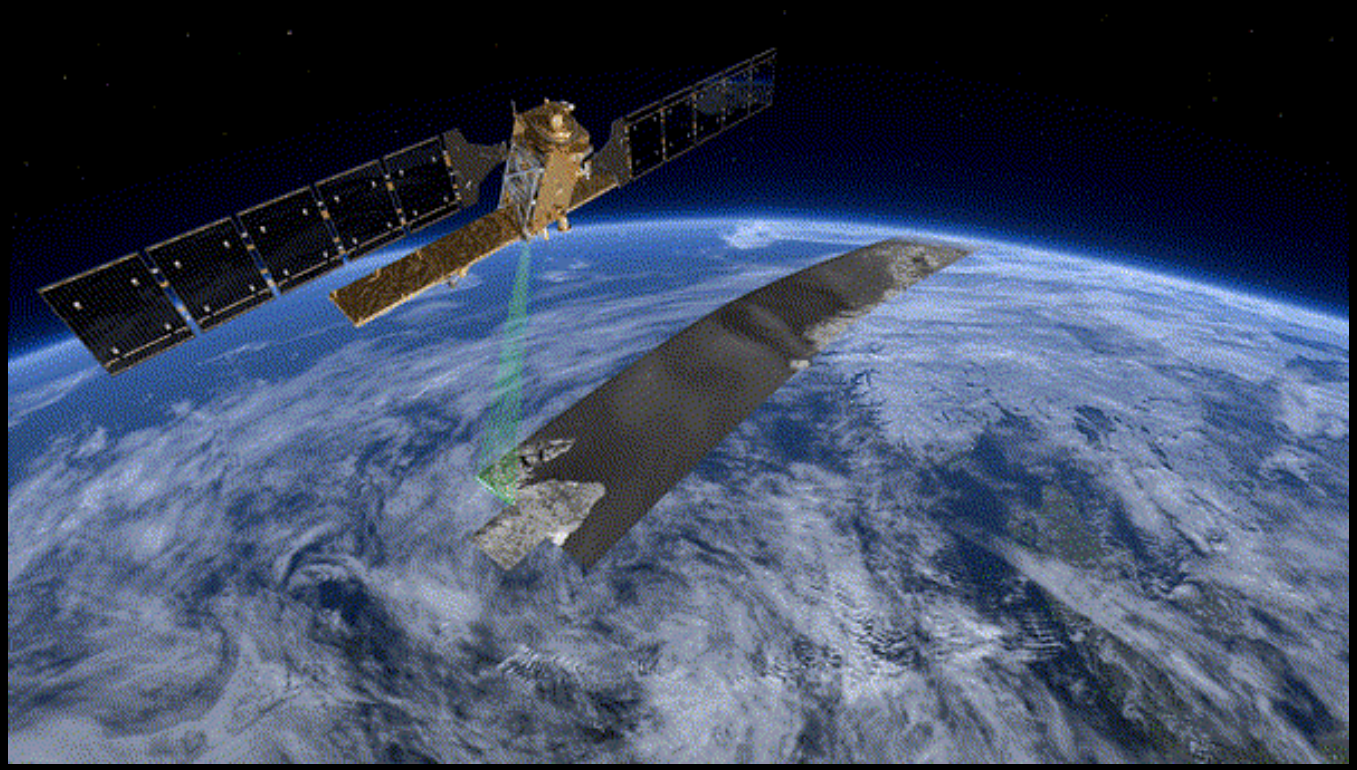


Demo: tracking an object in space using the Kalman Filter.

Robot localization

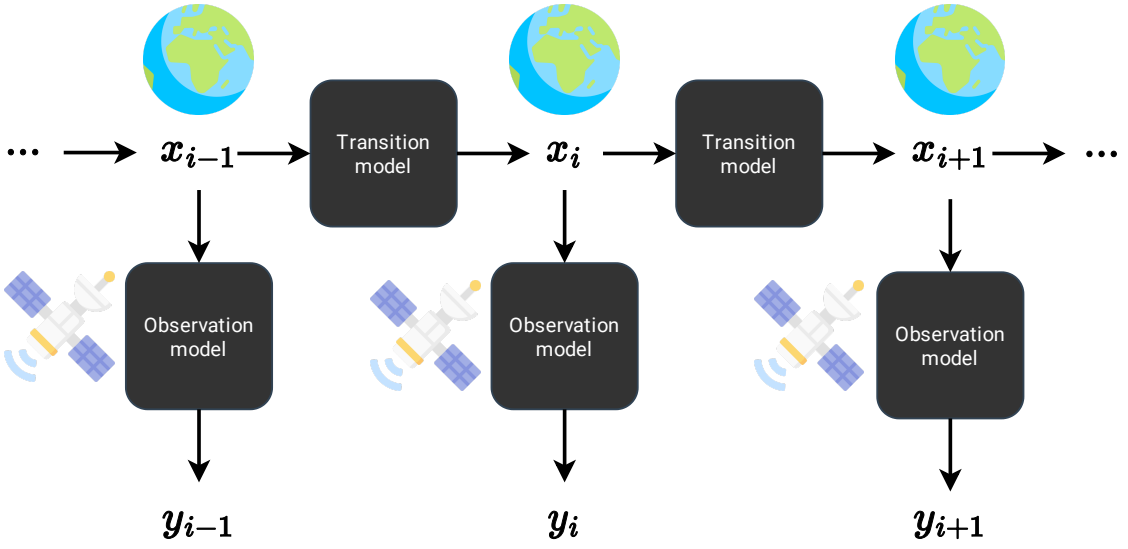
Filtering algorithms such as the Kalman filter and the particle filter are widely used for robot localization, i.e., estimating the position and orientation of a robot based on sensor data and a motion model.





Data assimilation for weather

Filtering is used to combine observations of the atmosphere with numerical models to estimate its current state and initialize weather forecasts.



Formally, the goal of **data assimilation** is to estimate plausible atmospheric trajectories $x_{1:T}$ given one or more noisy observations $y_{1:T}$ as the posterior

$$p(x_{1:T}|y_{1:T}) \propto \prod_{t=1}^{T} p(y_t|x_t)p(x_t|x_{t-1}),$$

where the transition model $p(x_t|x_{t-1})$ is given by a numerical weather prediction model and the sensor model $p(y_t|x_t)$ describes the observation process from satellites, radars, weather stations, etc.


20 YEARS OF 4DVAR
Watch later
Share
ECMWF

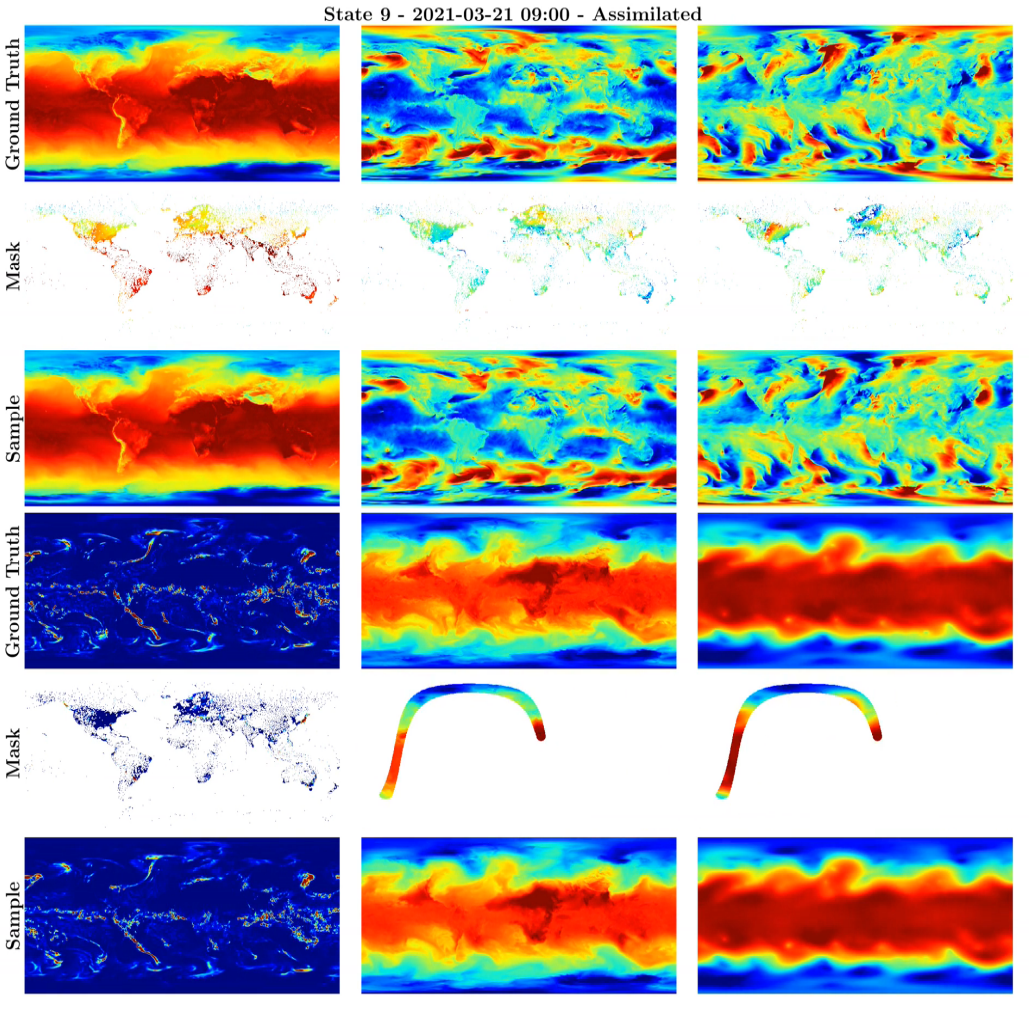
From Montefiore...

Appa (Andry et al, 2025) is a deep neural network for data assimilation. It is made of three components:

- a 500M-parameter **autoencoder** that compresses the data space x into a latent space z with a 450x compression factor;
- a 1B-parameter **latent diffusion model** that generates latent trajectories $z_{1:L}$;
- a **posterior sampling algorithm** adapted from MMPS (Rozet et al, 2024) that samples from the posterior distribution $p(z_{1:L} | y)$.

—

Credits: Andry et al, 2025 (arXiv:2504.18720).



Reanalysis of past data $p(x_{1:L} \mid y_{1:L})$.

—

Credits: Andry et al, 2025 (arXiv:2504.18720).

Summary

- Temporal models use state and sensor variables replicated over time.
  - Their purpose is to maintain a belief state as time passes and as more evidence is collected.
- The Markov and stationarity assumptions imply that we only need to specify
  - a transition model $\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{X}_t)$,
  - a sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$.
- Inference tasks include filtering, prediction, smoothing and finding the most likely sequence.
- Filter algorithms are all based on the core of idea of
  - projecting the current belief state through the transition model,
  - updating the prediction according to the new evidence.

The end.