Lecture 5: Probabilistic reasoning over time

Frederik Mallmann-Trenn

Department of Informatics
King's College London

(Version 1.3)

Today

- Introduction
- Probabilistic Reasoning I
- Probabilistic Reasoning II
- Sequential Decision Making
- Probabilistic Reasoning over Time
- Game Theory
- Argumentation I
- Argumentation II
- (A peek at) Machine Learning
- AI & Ethics

Introduction

- So far we have looked at a number of ways of using probability to handle uncertainty
  - Bayesian networks
  - Decision theory
  - Markov Decision Processes
- The formalisms are static, and so have limited ability to handle changing information.
- This week we'll look at models that can handle such dynamic situations.
  - Based on Bayesian networks

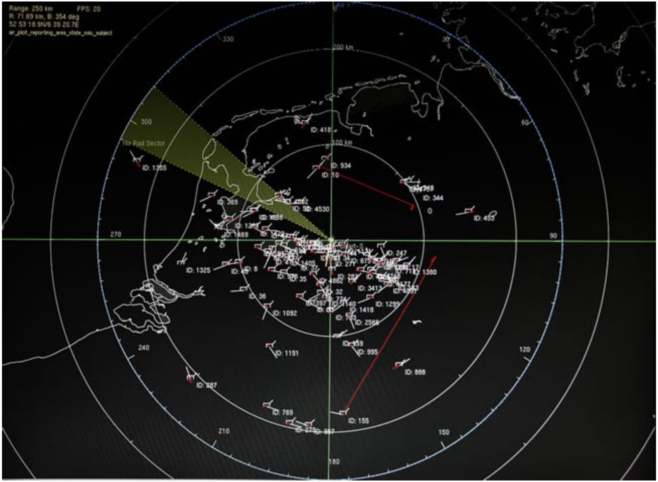
What?

(Pendleton Ward/Cartoon Network)

Robot localization

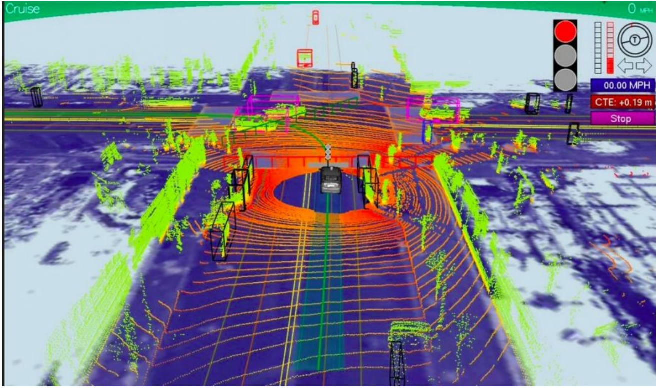
(Eric Schneider)

Tracking



(Thales)

Tracking



(Sebastian Thrun & Chris Urmson/Google)

States and observations

- The approach we'll look at considers the world to be a series of time slices.
- Each slice contains some variables:
  - The set $\mathbf{X}_t$ which we can't observe; and
  - The set $\mathbf{E}_t$ which we can observe.
- Note that $\mathbf{X}_t$ and $\mathbf{E}_t$ are **bold** to indicate that they may be **sets** of variables.
- At a given point in time we have an observation $\mathbf{E}_t = \mathbf{e}_t$.
- What would be an example?

States and observations

- Consider you live and work in some location without a window
  - Not so hard to imagine when you know KCL.

(indianapublicmedia.org/jgraham)

- You want to know whether it is raining.
- Your only information is looking at whether somebody who comes into your location each morning is carrying an umbrella.

States and observations

- Each day is one value of t.
- $\mathbf{E}_t$ contains the single variable U_t (or $Umbrella_t$).
  - Is the person carrying an umbrella?
- $\mathbf{X}_t$ contains the single variable R_t (or $Rain_t$)
  - Is it raining?

States and observations

- State sequence starts at $t = 0$, and the interval between slices in general depends on the problem.
  - Here it is one day
  - In robot localization it is pretty arbitrary
- First piece of evidence arrives at $t = 1$
- So, the umbrella world is:

$$R_0, R_1, R_2, \ldots$$

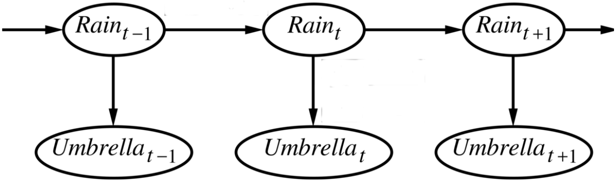
$$U_1, U_2, U_3, \ldots$$

- $a : b$ means the sequence of integers from a to b, so that $U_{2:4}$ is the sequence:

$$U_2, U_3, U_4$$

States and observations

- So the model is something like this:



Transition and sensor models

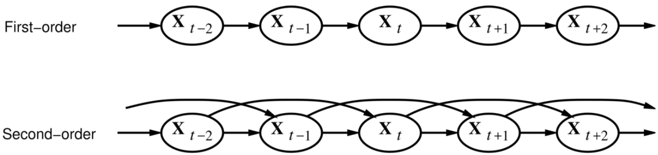
- We need to add two components to this backbone:
  - How the world evolves
    Transition model
  - What the evidence tells us
    Sensor model
- The transition model tells us:

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

what the probability is that it is raining today given the weather every previous day for as long as records have existed.

Transition and sensor models

- Luckily Professor Markov helps us out again.
- Make a Markov assumption that the value of the current state depends only on a finite fixed number of previous states.



- We commonly assume a first order Markov process, where the current state depends only on the previous state:

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

Transition and sensor models

- What would the model look like for a second order Markov process?

$$\mathbf{P}(\mathbf{X}_t | \mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t | \mathbf{X}_{t-2}, \mathbf{X}_{t-1})$$

Transition and sensor models

- Even with the Markov assumption we have a potentially infinite set of conditional probabilities.

$$\mathbf{P}(\mathbf{X}_1|\mathbf{X}_0), \mathbf{P}(\mathbf{X}_2|\mathbf{X}_1), \mathbf{P}(\mathbf{X}_3|\mathbf{X}_2) \ldots$$

- Usually circumvent this by assuming a stationary process
  - The model doesn't change
  - But the state itself can
- Thus we only have one, general $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$

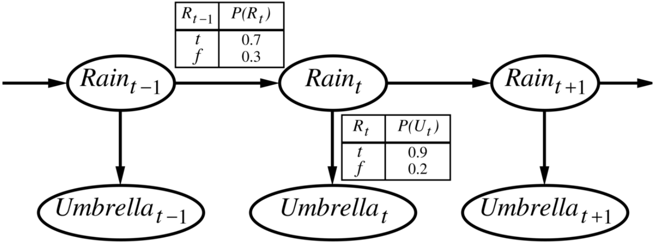
Sensor model

- The evidence variables $\mathbf{E}_t$ could depend on lots of previous variables.
- But we will assume the state is constructed in such a way that evidence only depends on the current state.
  - Again we can fix this by adding variables to the state.
- A Markov assumption for the sensor model:

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

Transition and sensor models

- Here are the transition and sensor models for the umbrella world:



- As for previous Bayesian networks, arrows run from causes to effects.

Initial state

- Also need to say how things get started:

$$\mathbf{P}(\mathbf{X}_0)$$

The prior probability over the state

And so. . .

- With this, we can then compute the complete joint probability over all the time slices:

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

- This is just the joint probability over a Bayesian network:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

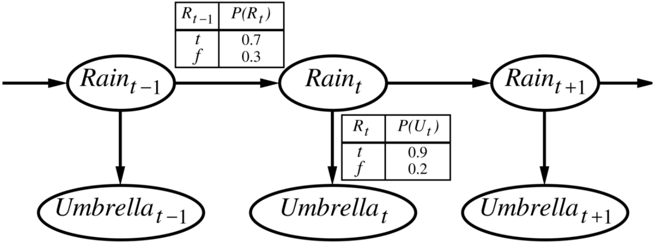
  from Lecture 3

- As we know from before, this is sufficient to compute anything we want.

Awesome!

(Pendleton Ward/Cartoon Network)

Dynamic Bayesian Network

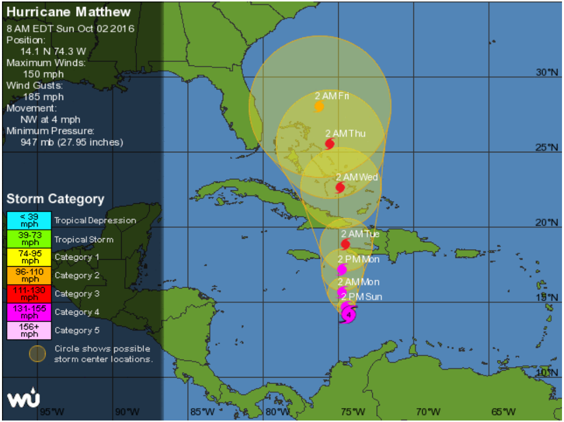


- Each slice contains some variables:
  - The set $\mathbf{X}_t$ which we can't observe; and
  - The set $\mathbf{E}_t$ which we can observe.
- Each slice is a Bayesian network.
- Whole thing is a Dynamic Bayesian Network (DBN).

Inference tasks

- What kinds of thing can we do with the model?
- Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$
  - determine belief state—input to the decision process of a rational agent.
- Prediction: $\mathbf{P}(\mathbf{X}_{t'}|\mathbf{e}_{1:t})$ for $t' > t$
  - Evaluation of possible action sequences, like filtering without the evidence.
- Smoothing: $\mathbf{P}(\mathbf{X}_{t'}|\mathbf{e}_{1:t})$ for $0 \leqslant t' < t$
  - Better estimate of past states, essential for learning.
- We'll look at how to do all these.

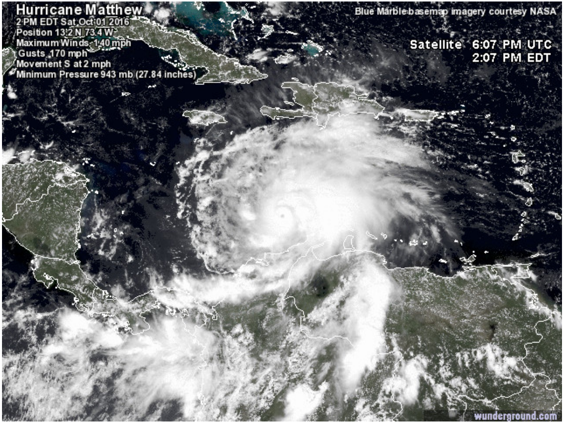
Inference tasks



(wunderground.com)

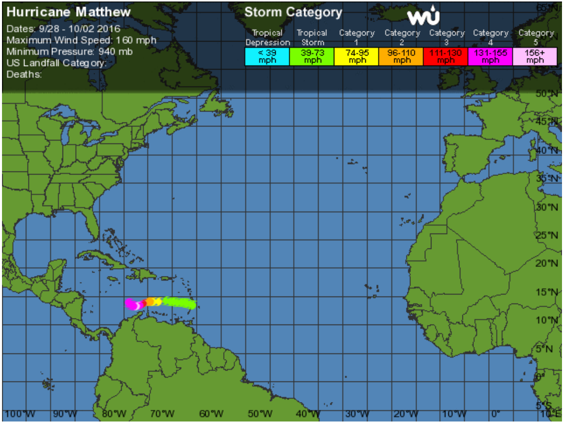
- Filtering
- Prediction

Inference tasks



(wunderground.com/NASA)

Inference tasks



(wunderground.com)

- Smoothing

Inference tasks

- Filtering
- Prediction
- Smoothing

Filtering

- A good algorithm for filtering will maintain a current state estimate and update it at each point.

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

- So we are looking for a simple function $f(\cdot)$ that we can just use to get the next estimate based on the last estimate
- Saves recomputation.
- It turns out that this is easy enough to come up with.

Filtering

- Our goal is to rewrite the l.h.s. of Equation (1)

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

- First, we divide up the evidence:

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

- Then we apply Bayes rule (see next slide), remembering the use of the normalization factor α.

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

Bayes rule?

- We are (hopefully) used to Bayes rule in its standard format:

$$\mathbf{P}(A|C) = \frac{\mathbf{P}(C|A)\mathbf{P}(A)}{\mathbf{P}(C)}$$

- The same thing holds when everything is conditioned on B:

$$\begin{aligned}\mathbf{P}(A|B,C) &= \frac{\mathbf{P}(C|A,B)\mathbf{P}(A|B)}{\mathbf{P}(C|B)} \\ &= \alpha\mathbf{P}(C|A,B)\mathbf{P}(A|B)\end{aligned}$$

- $(\alpha = 1\mathbf{P}(C|B))$
- This gives (for $A = \mathbf{X}_{t+1}$, $B = \mathbf{e}_{1:t}$, and $C = \mathbf{e}_{t+1}$,):

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

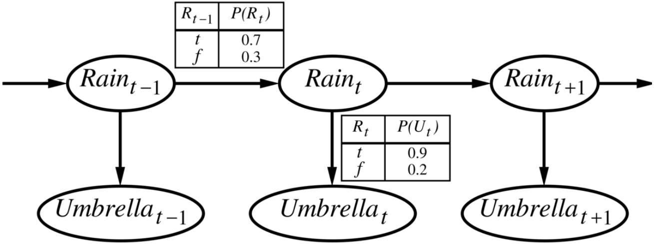
Filtering

- And after that we use the Markov assumption on the sensor model:

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

- The result of this assumption is to make that first term on the right hand side ignore all the evidence before $t+1$ — the probability of the observation at $t+1$ only depends on the value of $\mathbf{X}_{t+1}$.

Filtering


R_{t-1} | $P(R_t)$
t 0.7
f 0.3
R_t | $P(U_t)$
t 0.9
f 0.2
$Rain_{t-1}$
$Rain_t$
$Rain_{t+1}$
$Umbrella_{t-1}$
$Umbrella_t$
$Umbrella_{t+1}$

Filtering

- Let's look at that expression some more:

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

- The first term on the right updates with the new evidence and the second term on the right is a one step prediction from the evidence up to t to the state at $t + 1$.

Filtering

- Next we condition (see next slide) on the current state $\mathbf{P}(\mathbf{X}_t)$:

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

- Finally, we apply the Markov assumption again:

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\sum_{\mathbf{x}_t}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{e}_{1:t}) \quad (2)$$

and this is how we obtain the filtered probability of $\mathbf{X}_{t+1}$ given evidence $\mathbf{e}_{1:t+1}$

Conditioning?

- By law of total probability:

$$\begin{aligned}\mathbf{P}(A) &= \sum_B \mathbf{P}(A, B)\\ &= \sum_B \mathbf{P}(A|B)\mathbf{P}(B)\\ \mathbf{P}(A|C) &= \sum_B \mathbf{P}(A|B, C)\mathbf{P}(B|C)\end{aligned}$$

- So

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

where:

$$\begin{aligned}A &= \mathbf{X}_{t+1}\\ B &= \mathbf{x}_t\\ C &= \mathbf{e}_{1:t}\end{aligned}$$

Filtering

- If we define $\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1})$ to be $\mathbf{f}_{1:t+1}$, then we have a recursive update.
- $\mathbf{f}_{1:t+1}$ is a combination of:
  - $\mathbf{f}_{1:t}$, the value for the previous step.
  - $\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})$, the sensor model.
  - $\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_{t})$, the transition model.
- In other words, the probability distribution over the state variables at $t+1$ is: a function of:
  - the probability distribution over the state variables at t;
  - the sensor model;
  - and the transition model.
- Space and time constant, independent of t.
- This allows a limited agent to compute the current distribution for any length of sequence.

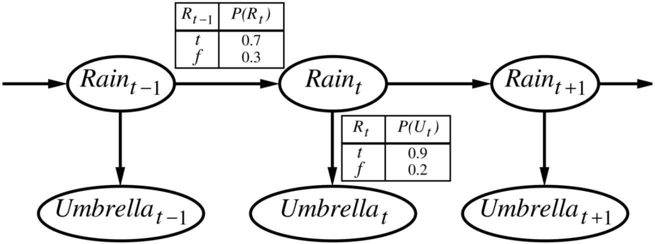
Filtering

- Since we will need it later, we'll write out $\mathbf{f}_{1:t}$:

$$\begin{aligned}\mathbf{f}_{1:t} &= \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t}) \\ &= \alpha \mathbf{P}(\mathbf{e}_t|\mathbf{X}_t) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_t|\mathbf{x}_{t-1}) P(\mathbf{x}_{t-1}|\mathbf{e}_{1:t-1})\end{aligned} \tag{3}$$

- Note that this is different to what we had on slide 34. That was for days from 1 to $t+1$. This is for days 1 to t.

Filtering the umbrella example



Filtering the umbrella example

- The prior for day 0, $P(R_0)$, is $\langle 0.5, 0.5 \rangle$.
- We can first *predict* whether it will rain on day 1 given what we already know:

$$
\begin{aligned}
\mathbf{P}(R_1) &= \sum_{r_0} \mathbf{P}(R_1 | r_0) P(r_0) \\
&= \langle 0.7, 0.3 \rangle \times 0.5 + \langle 0.3, 0.7 \rangle \times 0.5 \\
&= \langle 0.5, 0.5 \rangle
\end{aligned}
$$

- As we should expect, this just gives us the prior — that is the probability of rain when we don't have any evidence.

Filtering the umbrella example

- However, we have observed the umbrella, so that $U_1 = true$, and we can update using the sensor model:

$$\begin{aligned}\mathbf{P}(R_1|u_1) &= \alpha\mathbf{P}(u_1|R_1)\mathbf{P}(R_1)\\ &= \alpha\langle 0.9, 0.2\rangle\langle 0.5, 0.5\rangle\\ &= \alpha\langle 0.45, 0.1\rangle\\ &\approx \langle 0.818, 0.182\rangle\end{aligned}$$

- So, since umbrella is strong evidence for rain, the probability of rain is much higher once we take the observation into account.

Filtering the umbrella example

- We can then carry out the same computation for day 2, first *predict*ing whether it will rain given what we already know:

$$\begin{aligned}
\mathbf{P}(R_2|u_1) &= \sum_{r_1} \mathbf{P}(R_2|r_1)P(r_1|u_1) \\
&= \langle 0.7, 0.3\rangle \times 0.818 + \langle 0.3, 0.7\rangle \times 0.182 \\
&\approx \langle 0.627, 0.373\rangle
\end{aligned}$$

- So even without evidence of rain on the second day there is a higher probability of rain than the prior because rain tends to follow rain.
- In this model rain tends to persist.

Filtering the umbrella example

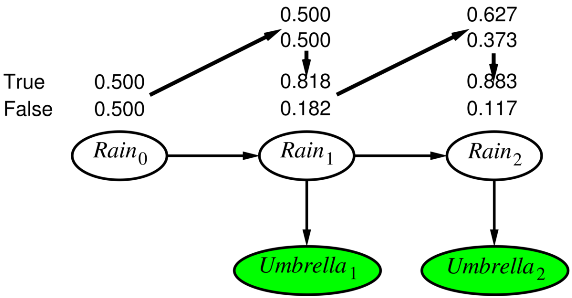
- Then we can repeat the evidence update, u_2 ($U_2 = \mathit{true}$), so:

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

- So, the probability of rain increases again, and is higher than on day 1.

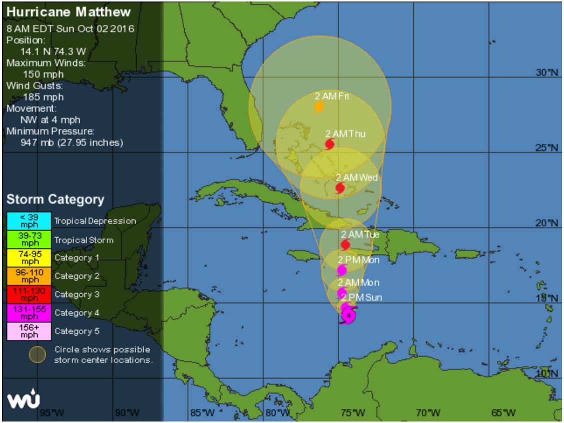
Filtering the umbrella example

- Put more succinctly:



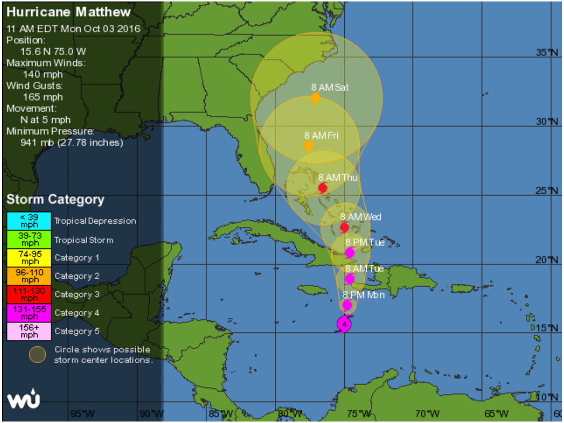
- We can think of the calculation as messages passed along the chain.

Inference tasks



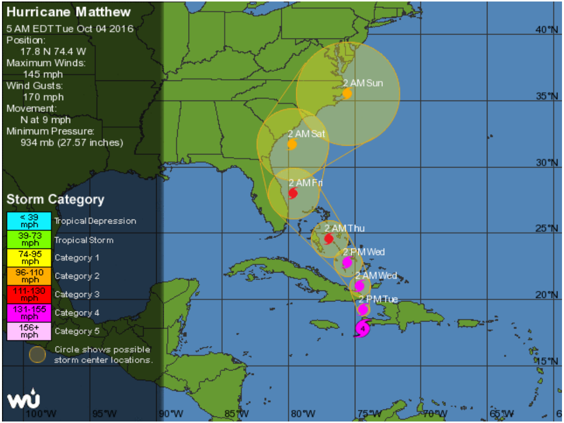
(wunderground.com)

Inference tasks



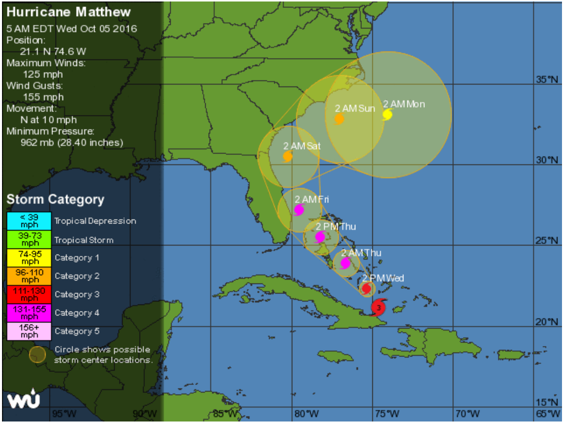
(wunderground.com)

Inference tasks



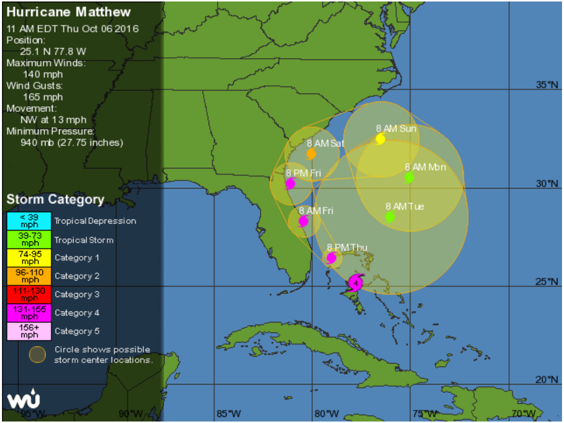
(wunderground.com)

Inference tasks



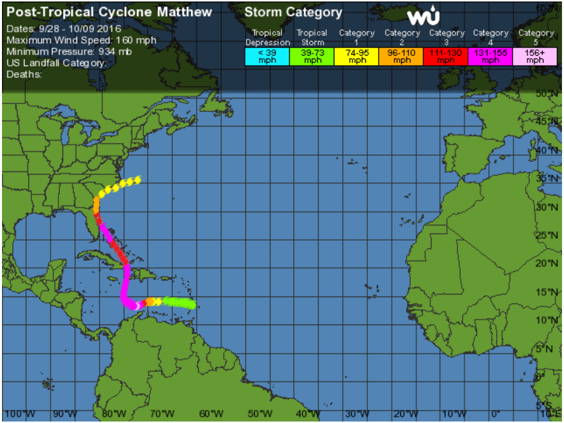
(wunderground.com)

Inference tasks



(wunderground.com)

Inference tasks



(wunderground.com)

Inference tasks

- Filtering
- Prediction
- Smoothing

Prediction

- Prediction is filtering without new evidence.

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

- Given the current state, what does the future bring?

Prediction

- As for filtering, we can use the Markov assumption to give us:

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

  which allows us to predict one step forward in time.

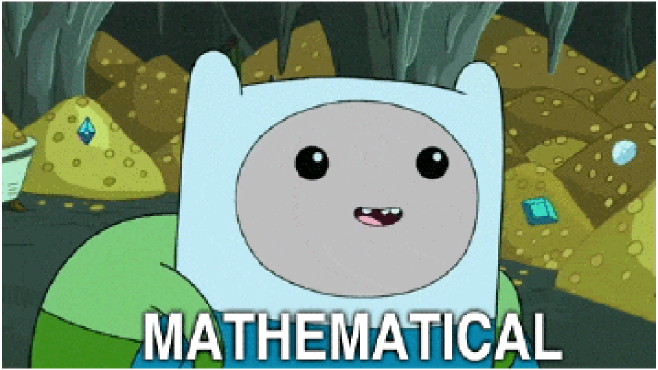
Predicting the Umbrella example

- In fact, we already did this.
- The first step in filtering was prediction.
- Given we saw an umbrella on day 1, what is the probability of rain on day 2?

$$
\begin{aligned}
\mathbf{P}(R_2|u_1) &= \sum_{r_1} \mathbf{P}(R_2|r_1)P(r_1|u_1) \\
&= \langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.182 \\
&\approx \langle 0.627, 0.373 \rangle
\end{aligned}
$$

just as on slide 40.

Sorted



(Pendleton Ward/Cartoon Network)

Predicting and Filtering

- (2) gave us a one step method for computing the filtered probability, but the worked example first computed the predicted probability and then the filtered probability.
- What's going on?

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

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

- So if we write the predicted probability of $\mathbf{X}$ at $t+1$ given evidence $\mathbf{e}_{1:t}$ as $Pred(\mathbf{X}_{t+1}|\mathbf{e}_{1:t})$, filtering is:

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

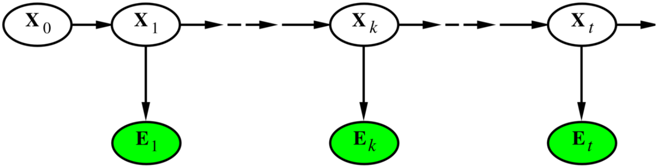
- Filtering is predicting and then updating with evidence.

Inference tasks

- Filtering
- Prediction
- Smoothing

Smoothing

- Smoothing is computing the distribution over past states given evidence up to the present.



- Want the probability over all states k, $0 \leqslant k < t$.

Smoothing

- Break the computation into two pieces, evidence from 0 to k and evidence from $k+1$ to t.
- Proceeding just as before:

$$
\begin{aligned}
\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t}) &= \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k}, \mathbf{e}_{k+1:t}) \\
&= \alpha \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{e}_{1:k}) && \text{Bayes' Rule (see over)} \\
&= \alpha \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) && \text{Cond. Independence} \\
&= \alpha \mathbf{f}_{1:k}\mathbf{b}_{k+1:t} && (4)
\end{aligned}
$$

- $\mathbf{f}$ is a “forward” message, computed just as we did for the filtering/prediction case.
- $\mathbf{b}$ is a backward message.

Bayes rule?

- Again we make use of:

$$\begin{aligned}\mathbf{P}(A|B,C) &= \frac{\mathbf{P}(A|B)\mathbf{P}(C|A,B)}{\mathbf{P}(C|B)} \\ &= \alpha\mathbf{P}(A|B)\mathbf{P}(C|A,B)\end{aligned}$$

- This gives:

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

where:

$$\begin{aligned}A &= \mathbf{X}_k \\ B &= \mathbf{e}_{1:k} \\ C &= \mathbf{e}_{k+1:t}\end{aligned}$$

Smoothing

- To compute the backwards message we condition on $\mathbf{X}_{k+1}$:

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

- Then apply the Markov assumption:

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

- Rewriting $\mathbf{e}_{k+1:t}$:

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

Smoothing

- We had:

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

- Applying conditional independence again:

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

Smoothing

- So, we have:

$$\begin{aligned}\mathbf{b}_{k+1:t} &= \mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) \\ &= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)\end{aligned} \tag{5}$$

- Rewriting $\mathbf{P}(\mathbf{e}_{k+2:t}|\mathbf{X}_k)$ as $\mathbf{b}_{k+2:t}$:

$$\mathbf{b}_{k+1:t} = \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})\mathbf{b}_{k+2:t}\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \tag{6}$$

- The first term on the right hand side is the sensor model.
  The third term is the transition model.
  The second term is the message from $k + 2$

Smoothing

- So in:

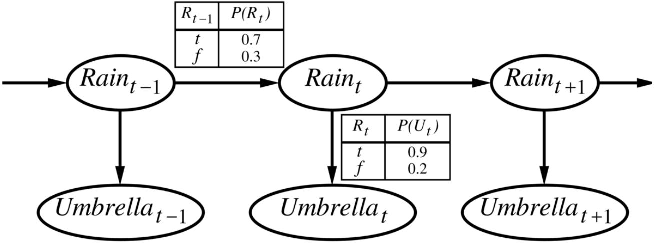
$$\mathbf{f}_{1:k}\mathbf{b}_{k+1:t}$$

  there are two recursive components.
- We have a forward component from 1 to k, and a backward component from t to k.
- The backward component is initialized with:

$$\mathbf{P}(\mathbf{e}_{t+1:t}|\mathbf{X}_t) = \mathbf{P}(\ |\mathbf{X}_t) = 1 \tag{7}$$

  since the probability of observing the set of no observations is always 1.

Smoothing the umbrella example



- Consider the umbrella world on day 1.
- Compute the smoothed probability of rain on day 1.
- Have seen umbrellas on day 1 and day 2.

Smoothing the umbrella example

- Now (4) tells us that:

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

- We want probability on day 1 given umbrellas n day 1 and 2. So $k = 1$ and $t = 2$.
- We get:

$$\mathbf{P}(\mathbf{X}_1|\mathbf{e}_{1:2}) = \alpha \mathbf{f}_{1:1}\mathbf{b}_{2:2}$$

- Or in umbrella notation, with $\mathbf{X} = R$ and $\mathbf{e} = u$:

$$\mathbf{P}(R_1|u_{1:2}) = \alpha \mathbf{f}_{1:1}\mathbf{b}_{2:2}$$

Smoothing the umbrella example

Now

$$\mathbf{P}(R_1|u_{1:2}) = \alpha \mathbf{f}_{1:1}\mathbf{b}_{2:2}$$

and (3) tells us that:

$$\mathbf{f}_{1:1} = \mathbf{P}(\mathbf{X}_1|\mathbf{e}_{1:1})$$

which in umbrella notation is:

$$\mathbf{f}_{1:1} = \mathbf{P}(R_1|u_{1:1})$$

- Similarly, (5) gives us:

$$\begin{aligned}\mathbf{b}_{2:2} &= \mathbf{P}(e_{2:2}|X_1)\\ &= \mathbf{P}(u_{2:2}|R_1)\end{aligned}$$

- Overall:

$$\mathbf{P}(R_1|u_{1:2}) = \alpha \mathbf{P}(R_1|\mathbf{u}_{1:1})\mathbf{P}(u_{2:2}|R_1)$$

Smoothing the umbrella example

- This might be easier to read without the sequence notation.
- We have:

$$\begin{aligned}\mathbf{P}(R_1|u_{1:2}) &= \alpha \mathbf{f}_{1:1}\mathbf{b}_{2:2} \\ &= \alpha \mathbf{P}(R_1|u_{1:1})\mathbf{P}(u_{2:2}|R_1)\end{aligned}$$

which is just:

$$\begin{aligned}\mathbf{P}(R_1|u_1, u_2) &= \alpha \mathbf{f}_{1:1}\mathbf{b}_{2:2} \\ &= \alpha \mathbf{P}(R_1|u_1)\mathbf{P}(u_2|R_1)\end{aligned}$$

Smoothing the umbrella example

- For the forward message, we have:

$$\begin{aligned}\mathbf{f}_{1:1} &= \mathbf{P}(R_1|u_1)\\ &= \alpha\mathbf{P}(u_1|R_1)\sum_{R_1}\mathbf{P}(R_1|r_0)P(r_0|u_0)\end{aligned}$$

Smoothing the umbrella example

- We already computed the forward message:

$$\mathbf{f}_{1:1} = \mathbf{P}(R_1|u_1)$$

since this is the filtered probability of rain on day 1 when we observed an umbrella on day 1:

$$\begin{aligned}\mathbf{f}_{1:1} &= \mathbf{P}(R_1|u_1) \\ &= \langle 0.818, 0.182 \rangle\end{aligned}$$

which we calculated on slide 40.

Smoothing the umbrella example

- For the backward message we have:

$$
\begin{aligned}
\mathbf{b}_{2:2} &= \mathbf{P}(u_2|R_1) \\
&= \sum_{r_2} P(u_2|r_2)P(u_{3:2}|r_2)\mathbf{P}(r_2|R_1)
\end{aligned}
$$

- Now, from (7) we know that:

$$
P(u_{3:2}|r_2) = P(|r_2) = 1
$$

- And so:

$$
\begin{aligned}
\mathbf{b}_{2:2} &= \sum_{r_2} P(u_2|r_2)P(u_{3:2}|r_2)\mathbf{P}(r_2|R_1) \\
&= (0.9 \times 1 \times \langle 0.7, 0.3 \rangle) \\
&\quad + (0.2 \times 1 \times \langle 0.3, 0.7 \rangle) \\
&= \langle 0.69, 0.41 \rangle
\end{aligned}
$$

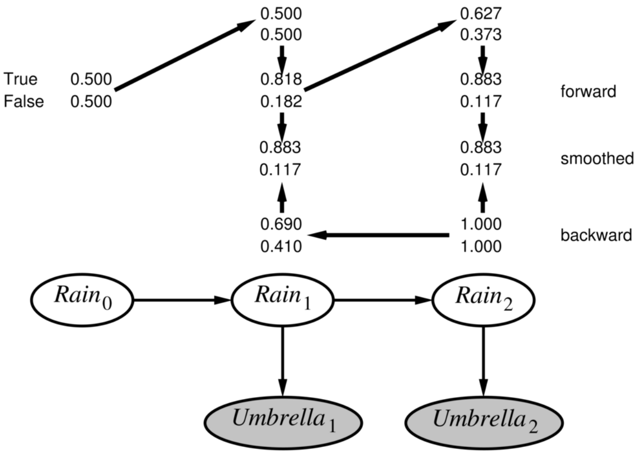
Smoothing the umbrella example

- Plugging the values for forward and backward messages back into the expression we started with gives:

$$
\begin{aligned}
\mathbf{P}(R_1 | u_1, u_2) &= \alpha \mathbf{f}_{1:1} \mathbf{b}_{2:2} \\
&= \alpha \langle 0.818, 0.182 \rangle \times \langle 0.69, 0.41 \rangle \\
&= \alpha \langle 0.56, 0.075 \rangle \\
&\approx \langle 0.882, 0.118 \rangle
\end{aligned}
$$

- Again we can think of this as message passing (next slide)

Smoothing the umbrella example



Smoothing the umbrella example

- The smoothed probability of rain on day 1 is higher than the filtered estimate because the umbrella on day 2 makes it more likely to have rained on day 1.

(indianapublicmedia.org/jgraham)

- Again these updates use constant time and space.

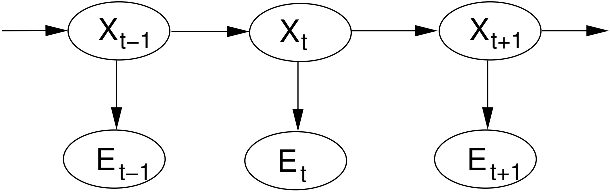
Where we are?

(Pendleton Ward/Cartoon Network)

Where we are?

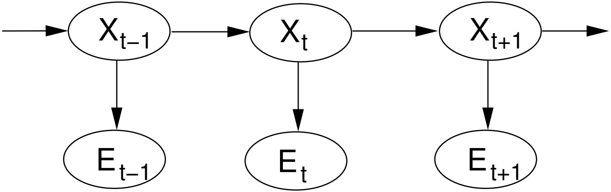
- We have a general approach to all the inference problems without thinking much about the specific details of the models.
- When we get specific, we find we can solve several common classes of problem.

Hidden Markov models



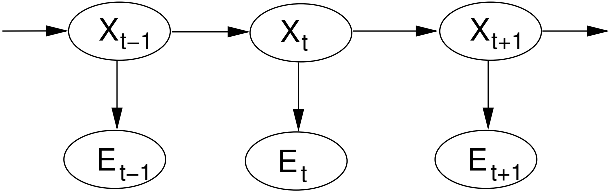
- HMMs have $\mathbf{X}_t$ as a single, discrete variable
- Usually $\mathbf{E}_t$ is also discrete
- Domain of $\mathbf{X}_t$ is $\{1, \ldots, S\}$

Hidden Markov models



- Transition matrix $\mathbf{T}_{ij} = P(X_t = j | X_{t-1} = i)$
  - e.g., $\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$
- Sensor matrix $\mathbf{O}_t$ for each time step, diagonal elements $P(e_t | X_t = i)$
  - $\mathbf{O}_t = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$ for U = true.
  - $\mathbf{O}_t = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.8 \end{pmatrix}$ for U = false.

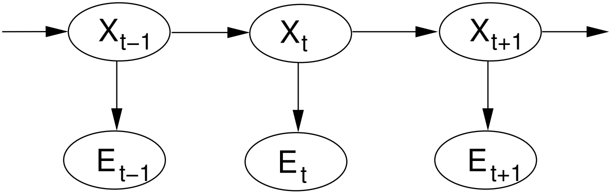
Hidden Markov models



- Forward and backward messages as column vectors:

$$
\begin{aligned}
\mathbf{f}_{1:t+1} &= \alpha \mathbf{O}_{t+1} \mathbf{T}^{\top} \mathbf{f}_{1:t} \\
\mathbf{b}_{k+1:t} &= \mathbf{T} \mathbf{O}_{k+1} \mathbf{b}_{k+2:t}
\end{aligned}
$$

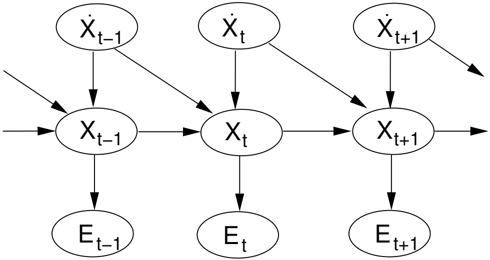
Hidden Markov models



- Forward-backward algorithm needs time $O(S^2 t)$ and space $O(St)$
- Matrix description points the way to easy implementation.

Kalman filters

- Modelling systems described by a set of continuous variables
- Gaussian prior, linear Gaussian transition model and sensor model

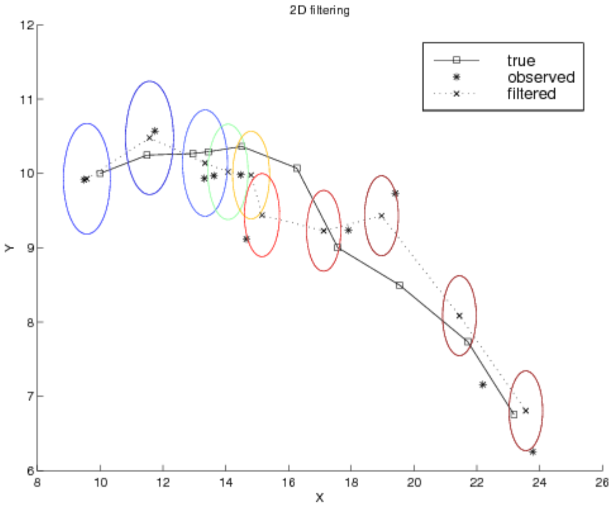


- Above is a Kalman filter for tracking in one dimension.

Kalman filters

- Robot tracking —$\mathbf{X}_t = X, Y, \dot{X}, \dot{Y}$
- Airplanes, ecosystems, economies, chemical plants, etc

Kalman filters

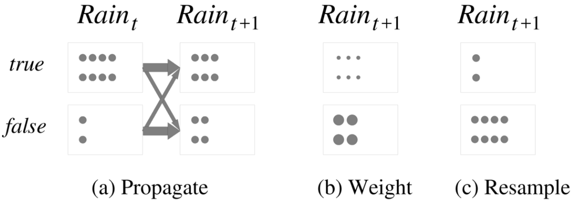


Back to DBNs

- The time slices we have looked at so far have been rather simple.
- Not a restriction on the method.
- Can have arbitrarily complex X and E.
- But our existing inference methods may have problems with computational complexity.

Particle filters

- Technique for approximate solution of a DBN.
- Basic idea: ensure that the population of samples ("particles") tracks the high-likelihood regions of the state-space
- Replicate particles proportional to likelihood for $\mathbf{e}_t$



- Time/space complexity linear in the number of particles.

Particle filters

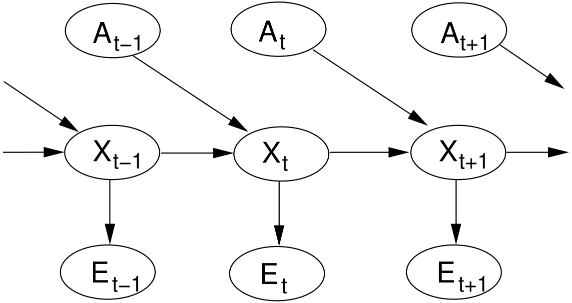
- Widely used for tracking nonlinear systems, esp. in vision

Chad Jenkins

- Also used for simultaneous localization and mapping in mobile robots
  - 10^5-dimensional state space
- Approximation error of particle filtering remains bounded over time.
- At least empirically—theoretical analysis is difficult.

Particle filters

- Slightly more complex DBN than an HMM.



- A is action.

Particle filter

- We approximate $P(x_t)$ by a set of samples:

$$P(x_t) \approx \{x_t^{(i)}, w_t^{(i)}\}_{i=1,\ldots,m}$$

- Each $x_t^{(i)}$ is a possible value of x, and each $w_t^{(i)}$ is the probability of that value (also called an importance factor).
- Initially we have a set of samples (typically uniform) that give us $P(x_0)$.
- Then we update with the following algorithm.

Particle filter

$x_{t+1} = \varnothing$

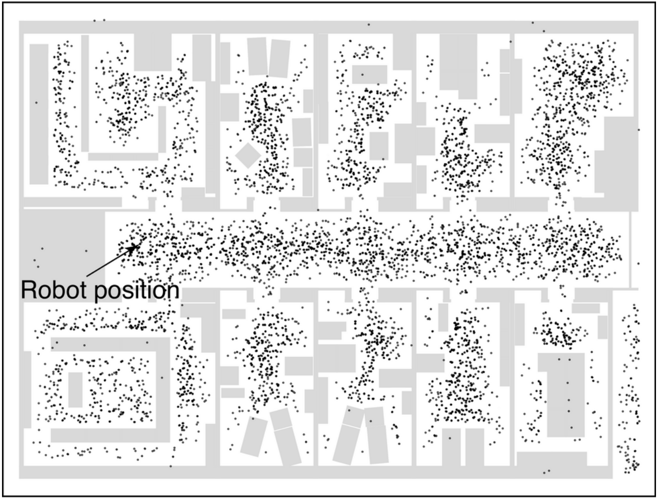
for $j = 1$ to m
// `apply the transition model`
generate a new sample $x_{t+1}^{(j)}$ from $x_t^{(j)}$, a_t and $\Pr(x_{t+1} \mid x_t, a_t)$
// `apply the sensor model`
compute the weight $w_{t+1}^{(j)} = \Pr(e_{t+1} \mid x_{t+1})$

// `pick points randomly but biased by their weight`
for $j = 1$ to m
pick a random $x_{t+1}^{(i)}$ from x_{t+1} according to $w_{t+1}^{(1)}, \ldots, w_{t+1}^{(m)}$

normalize w_{t+1} in x_{t+1}
return x_{t+1}

- And that is all it takes.

Particle filter

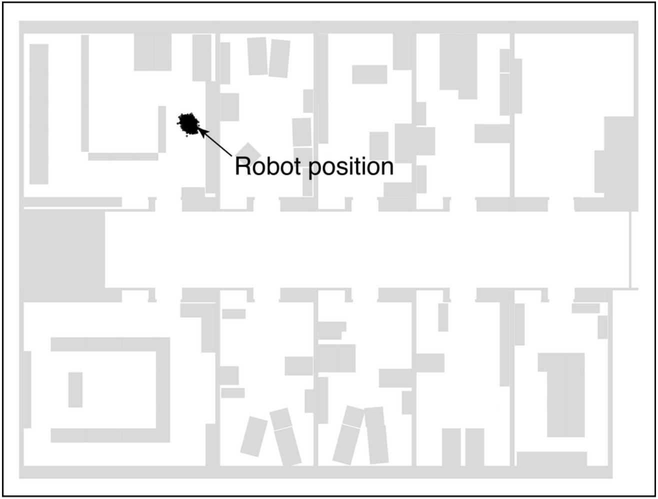


(a)

Particle filter

(b)

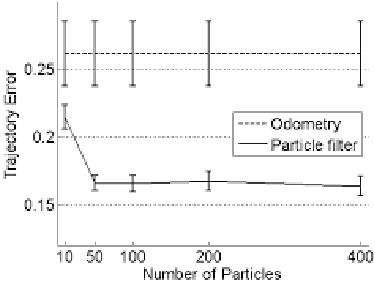
Particle filter



(c)

Particle filter

- Relation between error and number of particles:

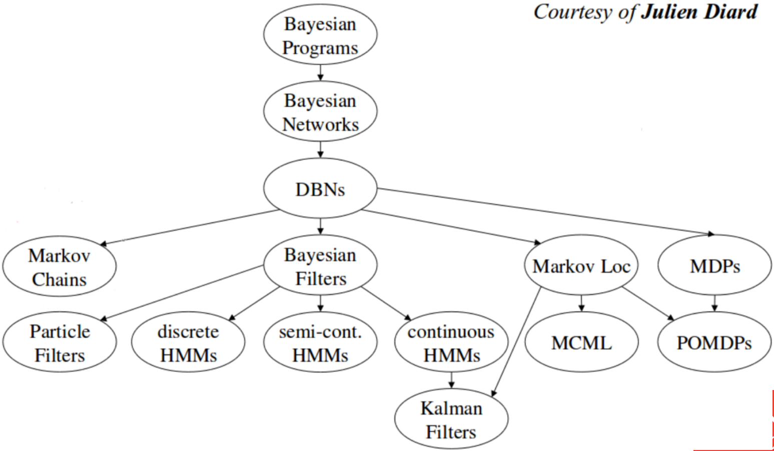


(Burguera, González and Oliver)

- Error would be reduced with more accurate sensing (laser not sonar).

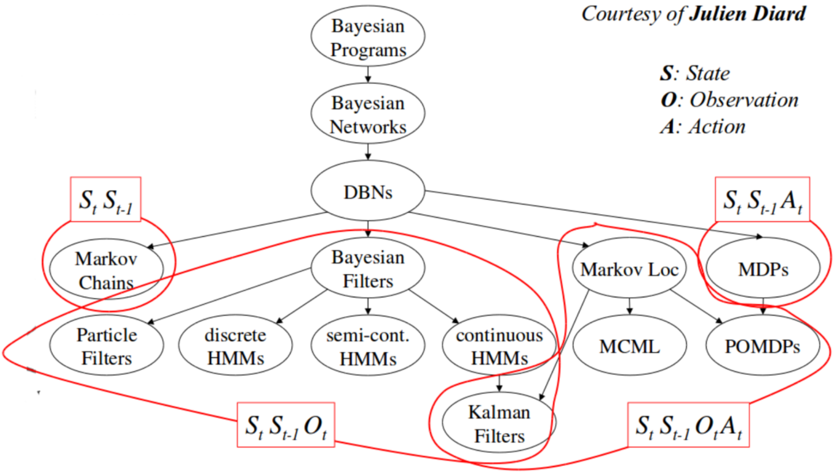
DBNs

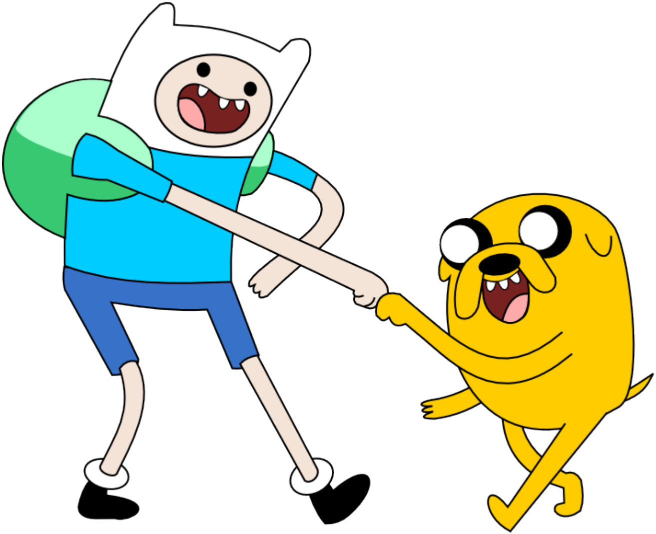
A general classification of dynamic probabilistic models



DBNs

A general classification of dynamic probabilistic models





(Pendleton Ward/Cartoon Network)

Summary

- This lecture moved from the static view of the world encorporated in Bayesian networks to something more dynamic.
- Dynamic Bayesian networks.
- We looked at the general types of inference possible in DBNs and showed how the necessary computations could be done.
- We also looked at some specific classes of problem that can be captured by DBNs.