PROBABILISTIC REASONING OVER TIME

Introduction

- The last couple of lectures looked at techniques to handle *uncertainty*
  - Bayesian networks
- The formalism is *static*, and so has limited ability to handle changing information.
- Lots of reasoning tasks involve a *dynamic world*
  - Monitoring a patient
  - Tracking an airplane
  - Identifying the location of a robot
- This week we'll look at models that can handle such dynamic situations.
  - Based on Bayesian networks

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.
- At a given point in time we have an observation $\mathbf{E}_t = \mathbf{e}_t$.
- What would be an example?

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

- 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.

- 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?

- 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:

$$U_2, U_3, U_4$$

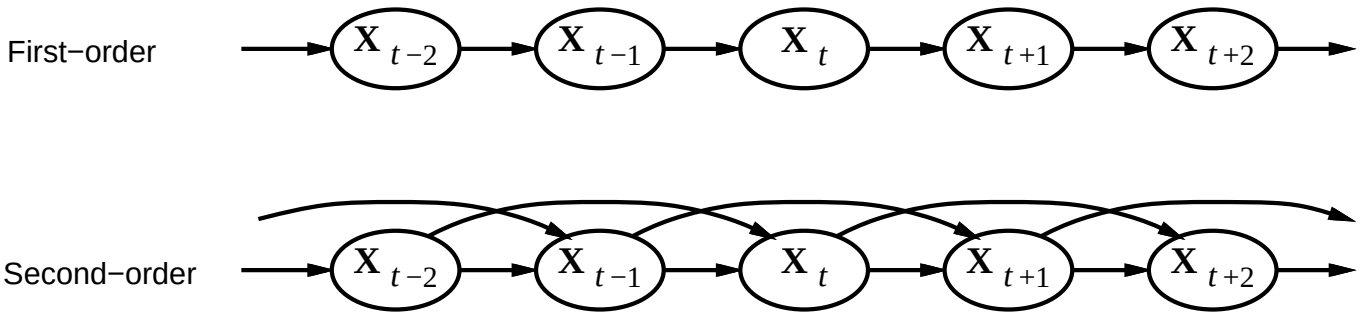
Transition and sensor models

- 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.
- What is the problem with this model?

- 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:

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

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

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

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

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

An aside

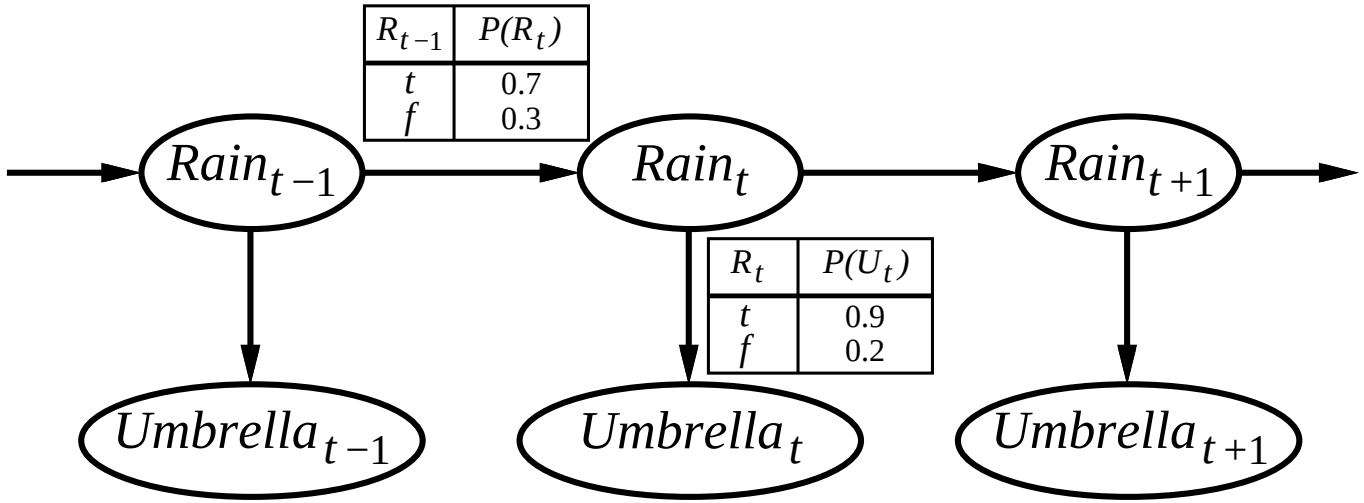
- First-order Markov assumption not exactly true in real world!
- Possible fixes:
  1. *Increase order* of Markov process
  2. *Augment state*, e.g., add $Temp_t$, $Pressure_t$
- Example: robot motion.
  - Augment position and velocity with $Battery_t$
- Example: umbrella world.
  - Augment state with $Season_t$ and/or $Pressure_t$

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.
- A Markov assumption for the sensor model:

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

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



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

- Also need to say how things get started:

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

  The prior probability over the state

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

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

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

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*: $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$ for $k > 0$
  - Evaluation of possible action sequences, like filtering without the evidence
- *Smoothing*: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \leq k < t$
  - Better estimate of past states, essential for learning
- *Most likely explanation*: $\arg\max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$
  - Speech recognition, decoding with a noisy channel

Filtering

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

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

- Saves recomputation.
- It turns out that this is easy enough to come up with.

- We rearrange the formula for:

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

- 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, remembering the use of the normalization factor α.

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

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

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

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

- 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 $t+1$.
- Next we condition on the current state $\mathbf{P}(\mathbf{X})$:

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

- Finally, we apply the Markov assumption again:

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

- We'll call the bit on the right $\mathbf{f}_{1:t}$

- $\mathbf{f}_{1:t}$ gives us the required recursive update.
  - The probability distribution over the state variables at $t + 1$ is a function of the transition model, the sensor model, and what we know about the state at time t.
- Space and time constant, independent of t.
- This allows a limited agent to compute the current distribution for any length of sequence.

Filtering the umbrella example

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

$$\begin{aligned}\mathbf{P}(\mathbf{R}_1) &= \sum_{r_0} \mathbf{P}(R_1|r_0)P(r_0) \\ &= \langle 0.7, 0.3 \rangle \times 0.5 + \langle 0.3, 0.7 \rangle \times 0.5 \\ &= \langle 0.5, 0.5 \rangle\end{aligned}$$

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

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

$$\begin{aligned}
\mathbf{P}(\mathbf{R}_1|U_1) &= \alpha\mathbf{P}(u_1|R_1)\mathbf{P}(R_1) \\
&= \alpha\langle 0.9, 0.2\rangle\langle 0.5, 0.5\rangle \\
&= \alpha\langle 0.45, 0.1\rangle \\
&\approx \langle 0.818, 0.182\rangle
\end{aligned}$$

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

- We can then carry out the same computation for Day 2, first predicting whether it will rain on day 1 given what we already know:

$$\begin{aligned}\mathbf{P}(\mathbf{R}_2|u_1) &= \sum_{r_1}\mathbf{P}(R_2|r_1)P(r_1|u_1)\\ &= \langle 0.7, 0.3\rangle \times 0.818 + \langle 0.3, 0.7\rangle \times 0.182\\ &\approx \langle 0.627, 0.373\rangle\end{aligned}$$

- 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.

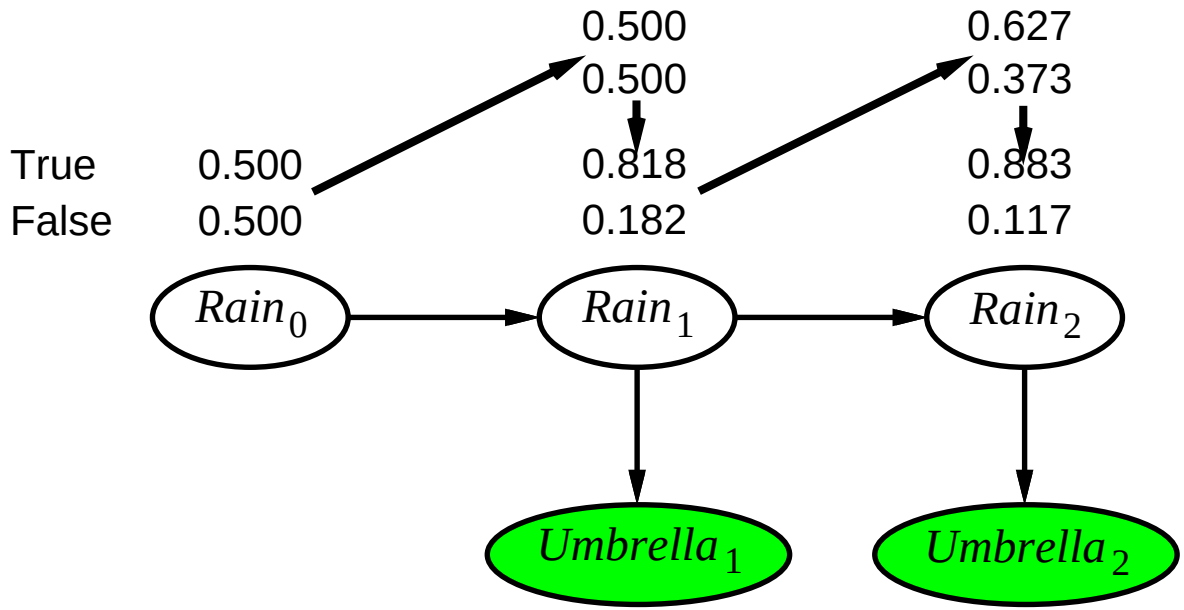
  (In this model rain tends to persist.)

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

$$\begin{aligned}
\mathbf{P}(\mathbf{R}_2|u_1, u_2) &= \alpha\mathbf{P}(u_2|R_2)\mathbf{P}(R_2|u_1) \\
&= \alpha\langle 0.9, 0.2\rangle\langle 0.627, 0.373\rangle \\
&= \alpha\langle 0.565, 0.075\rangle \\
&\approx \langle 0.883, 0.117\rangle
\end{aligned}$$

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

- Put more succinctly:



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

Prediction

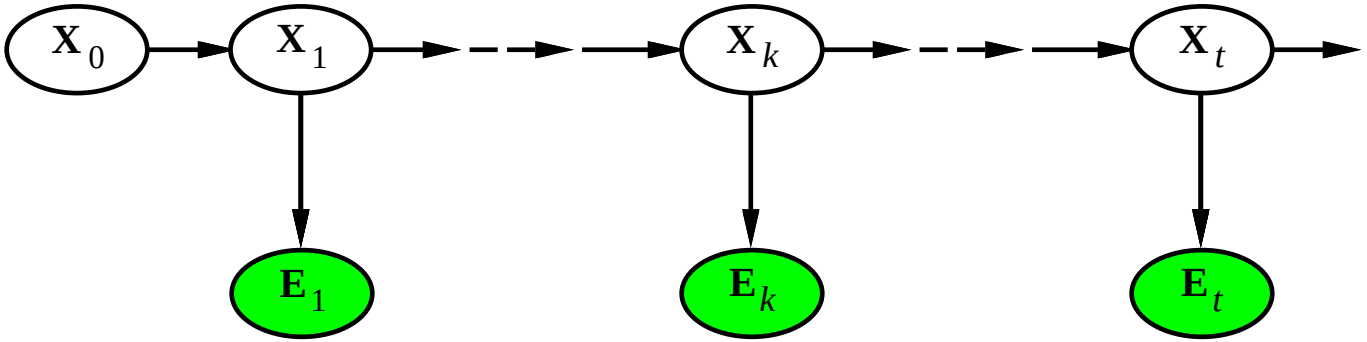
- Prediction is filtering without new evidence.

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

- Given the current state, what does the future bring?
- Just need the first part of the calculation we did above (in each step we first predicted and then updated with evidence).

Smoothing

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



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

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

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

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

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

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

- Then apply conditional independence:

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

- Condition again:

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

- The first and third terms here come from the model, the second term is the bit we compute recursively.

- 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:

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

    (the probability of observing the set of no observations is always 1).

- Consider the umbrella world on day 1. This time we update with information about umbrellas on day 1 and day 2:

$$\mathbf{P}(R_1|u_1, u_2) = \alpha\mathbf{P}(R_1|u_1)\mathbf{P}(u_2|R_1)$$

  We know that the first of these terms is $\langle 0.818, 0.182\rangle$ from before.

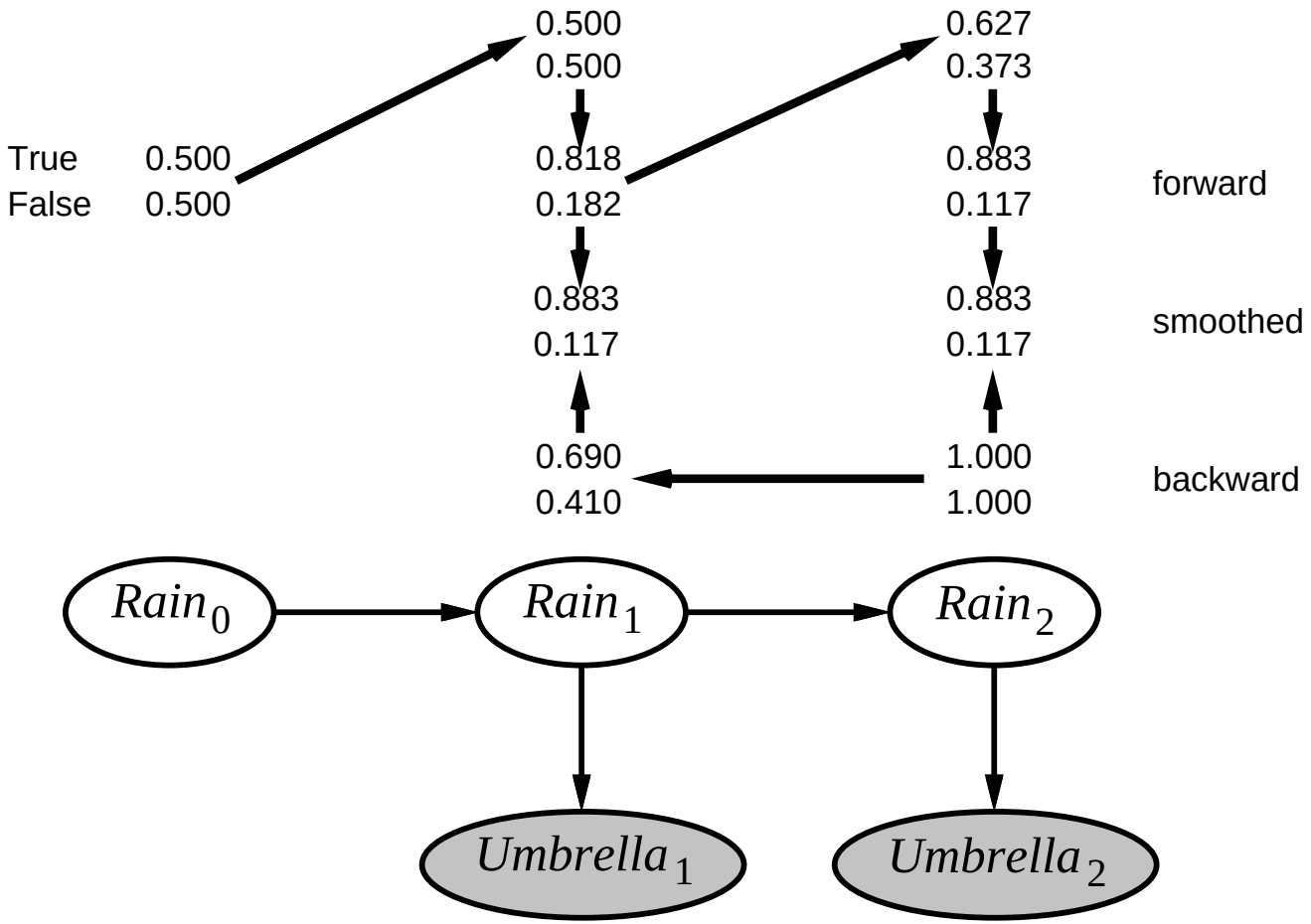
- The second term we can compute:

$$\begin{aligned}\mathbf{P}(u_2|R_1) &= \sum_{r_2} P(u_2|r_2)P(\ |r_2)\mathbf{P}(r_2|R_1)\\ &= (0.9 \times 1 \times \langle 0.70.3\rangle) + (0.2 \times 1 \times \langle 0.3, 0.7\rangle)\\ &= \langle 0.69, 0.41\rangle\end{aligned}$$

- Plugging this back into the expression we started with gives:

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

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


0.500
0.500
0.627
0.373
True
False
0.500
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
Rain0
Rain1
Rain2
Umbrella1
Umbrella2

- 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.

- Again these updates use constant time and space.

Most likely sequence

- Most likely sequence $\neq$ sequence of most likely states!!!!
- Most likely path to each $\mathbf{x}_{t+1}$ is most likely path to *some* $\mathbf{x}_t$ plus one more step

$$\max_{\mathbf{x}_1 \ldots \mathbf{x}_t} \mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{X}_{t+1} | \mathbf{e}_{1:t+1})$$

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

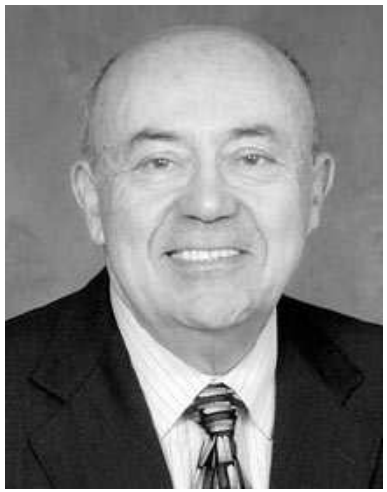
- Identical to filtering, except $\mathbf{f}_{1:t}$ replaced by:

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

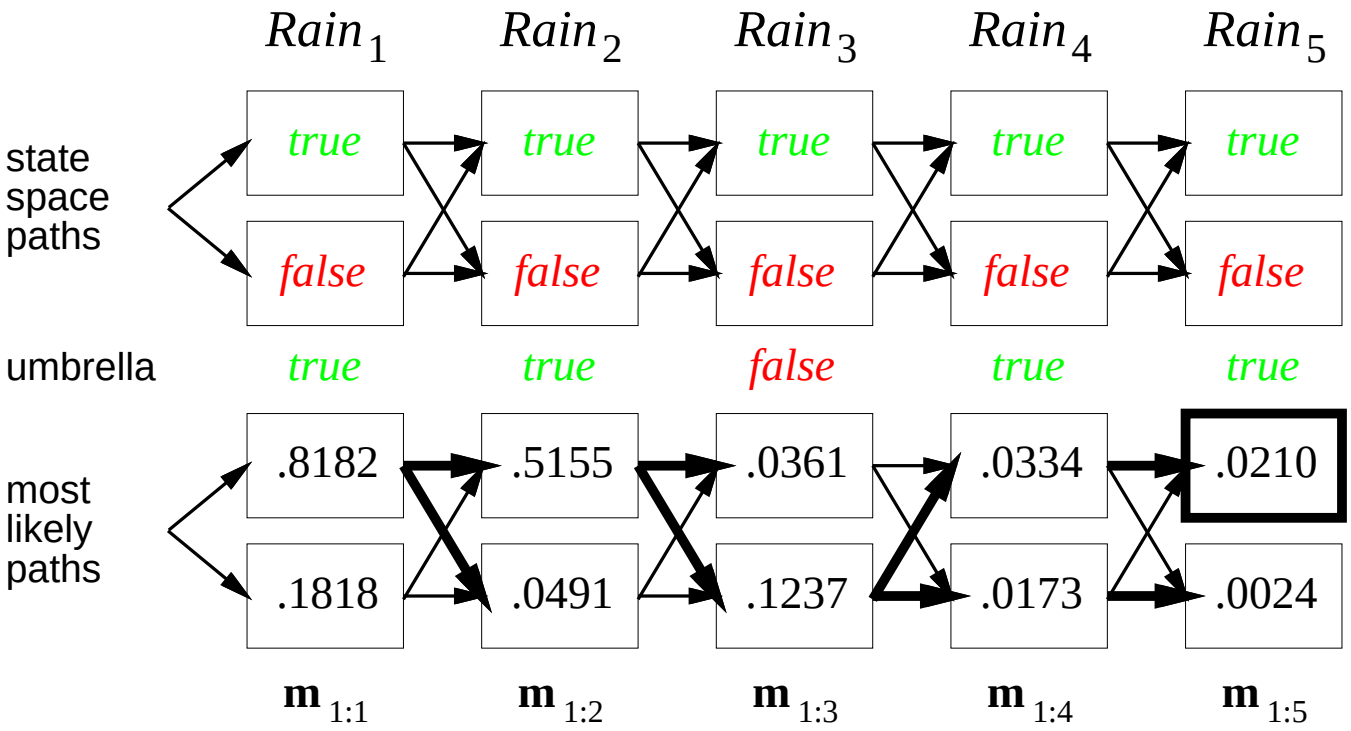
- $\mathbf{m}_{1:t}(i)$ gives the probability of the most likely path to state i.

- Update has sum replaced by max, giving the *Viterbi algorithm*:

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



Andrew Viterbi


$Rain_1$
$Rain_2$
$Rain_3$
$Rain_4$
$Rain_5$
state space paths
true
false
umbrella
true
true
false
true
true
most likely paths
.8182
.1818
.5155
.0491
.0361
.1237
.0334
.0173
.0210
.0024
$\mathbf{m}_{1:1}$
$\mathbf{m}_{1:2}$
$\mathbf{m}_{1:3}$
$\mathbf{m}_{1:4}$
$\mathbf{m}_{1:5}$

- Bold state is the most likely state at day 5.
- For each state you can tell its best predecessor (bold arrow).
- So having computed the state values, can easily read off the most likely sequence.

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 too)
- Domain of $\mathbf{X}_t$ is $\{1, \ldots, S\}$
- *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 $P(e_t | X_t = i)$
  - e.g., with $U_1 = \mathit{true}$, $\mathbf{O}_1 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$

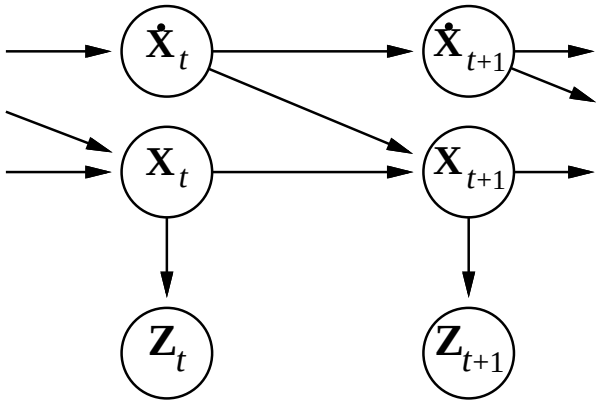
- Forward and backward messages as column vectors:

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

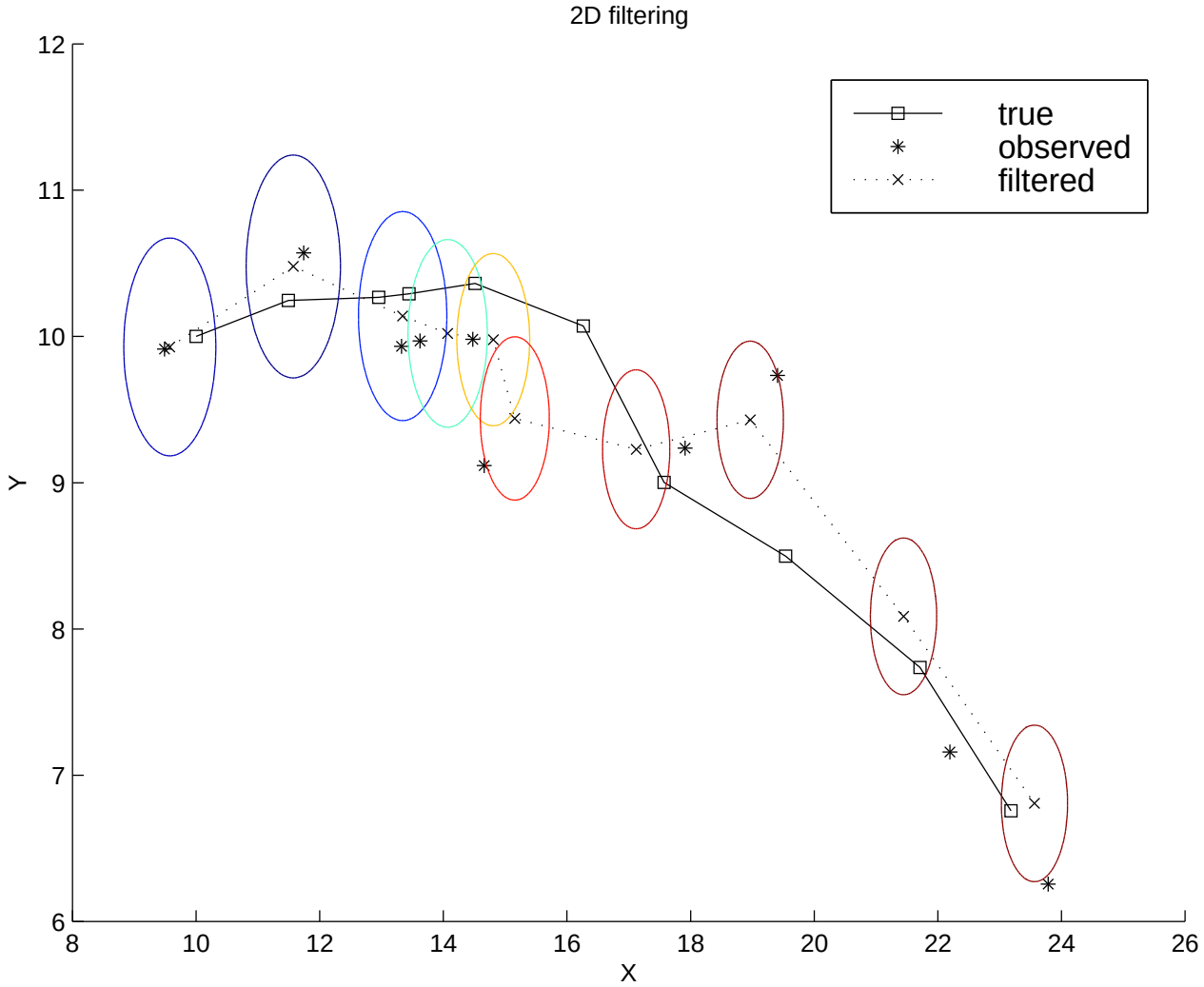
- Forward-backward algorithm needs time $O(S^2 t)$ and space $O(St)$
- Matrix description points the way to easy implementation.

Kalman filters

- Modelling systems described by a set of continuous variables
  - Robot tracking —$\mathbf{X}_t = X, Y, \dot{X}, \dot{Y}$
  - Airplanes, ecosystems, economies, chemical plants, etc



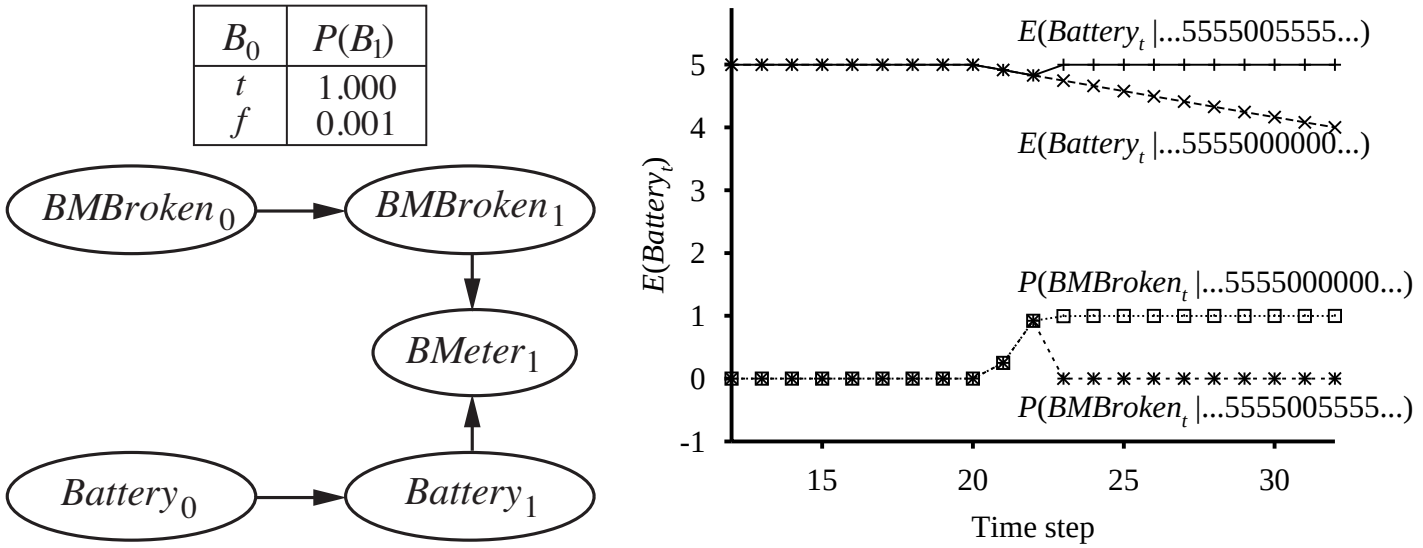
- Z is observation.
- Gaussian prior, linear Gaussian transition model and sensor model


2D filtering
true
observed
filtered
X
Y
12
11
10
9
8
7
6
8
10
12
14
16
18
20
22
24
26

- Sparse dependencies $\Rightarrow$ exponentially fewer parameters;
  - e.g., 20 state variables, three parents each
- DBN has $20 \times 2^3 = 160$ parameters, HMM has $2^{20} \times 2^{20} \approx 10^{12}$

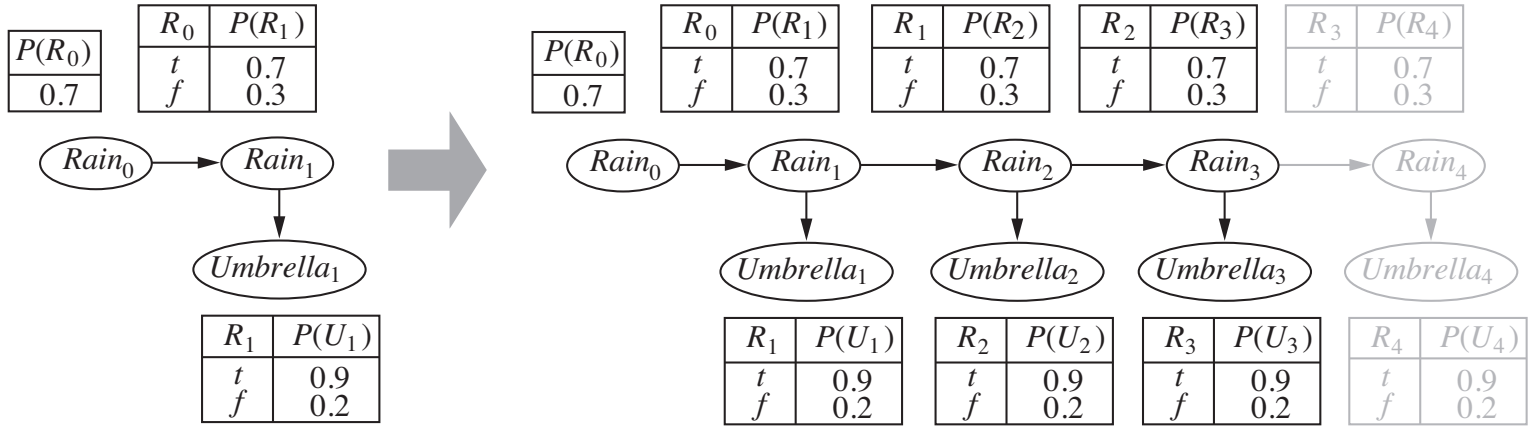
DBNs vs Kalman filters

- Every Kalman filter model is a DBN, but few DBNs are KFs;
  - Real world requires non-Gaussian posteriors
- What's the battery charge?



Inference in DBNs

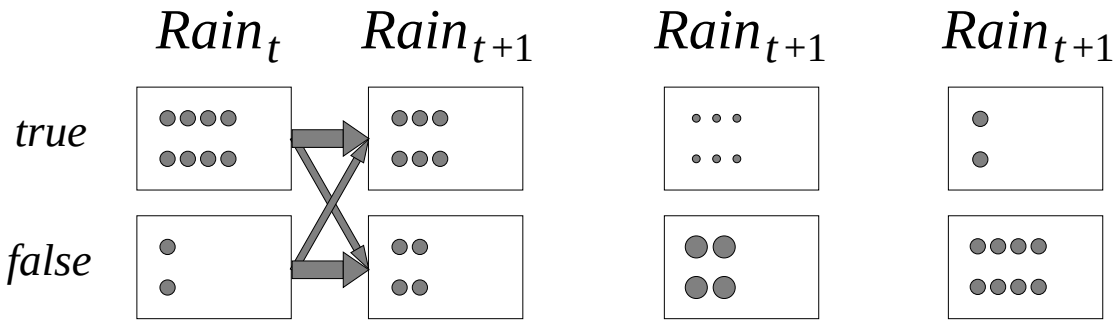
- There are exact algorithms for DBNs:



  - "Unroll" DBN to create a static Bayesian network and use standard approaches.
  - But time and space complexity is exponential.
- However, there are some good approximation algorithms.

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 $\mathbf{e}_t$

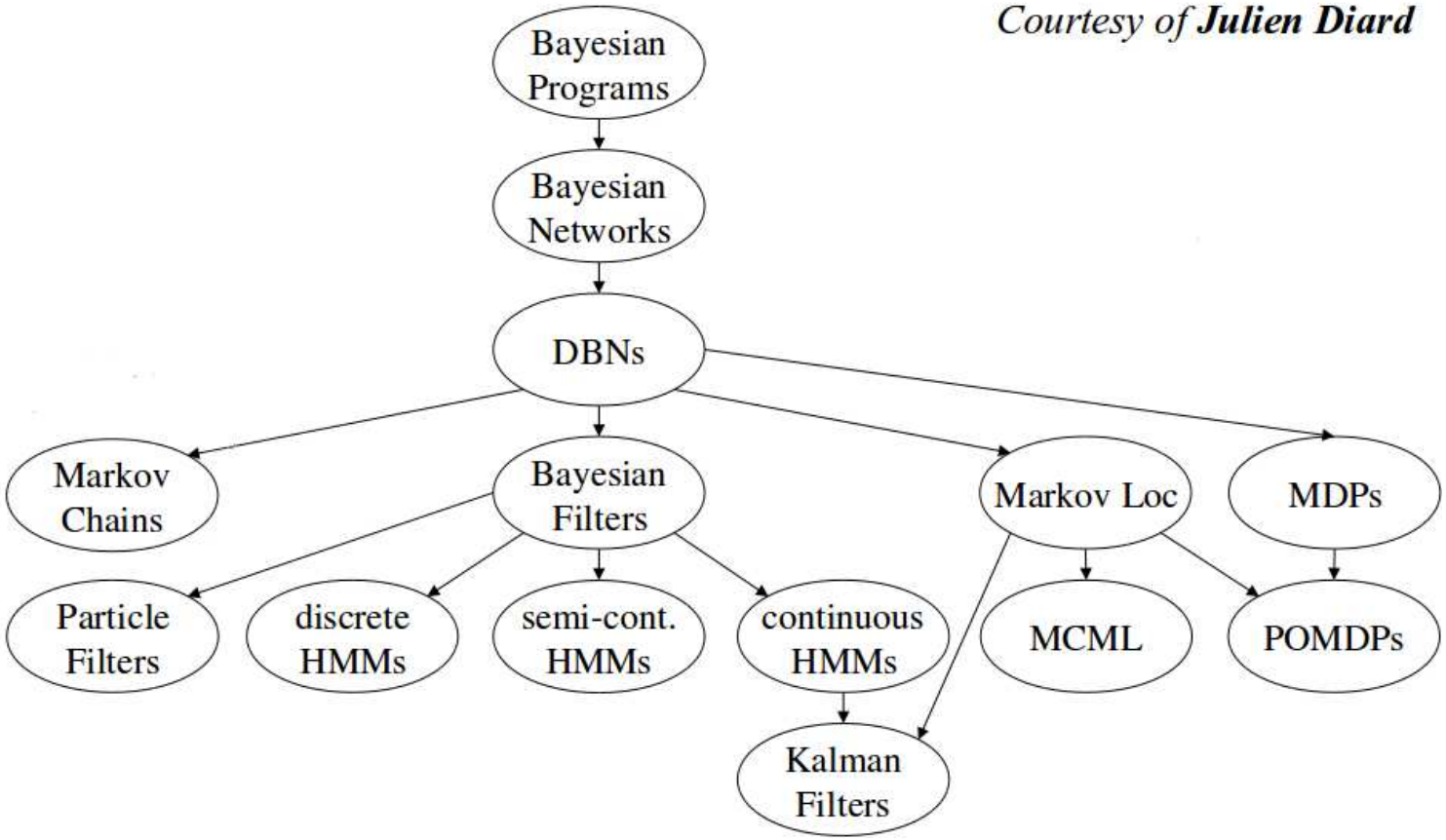


(a) Propagate (b) Weight (c) Resample

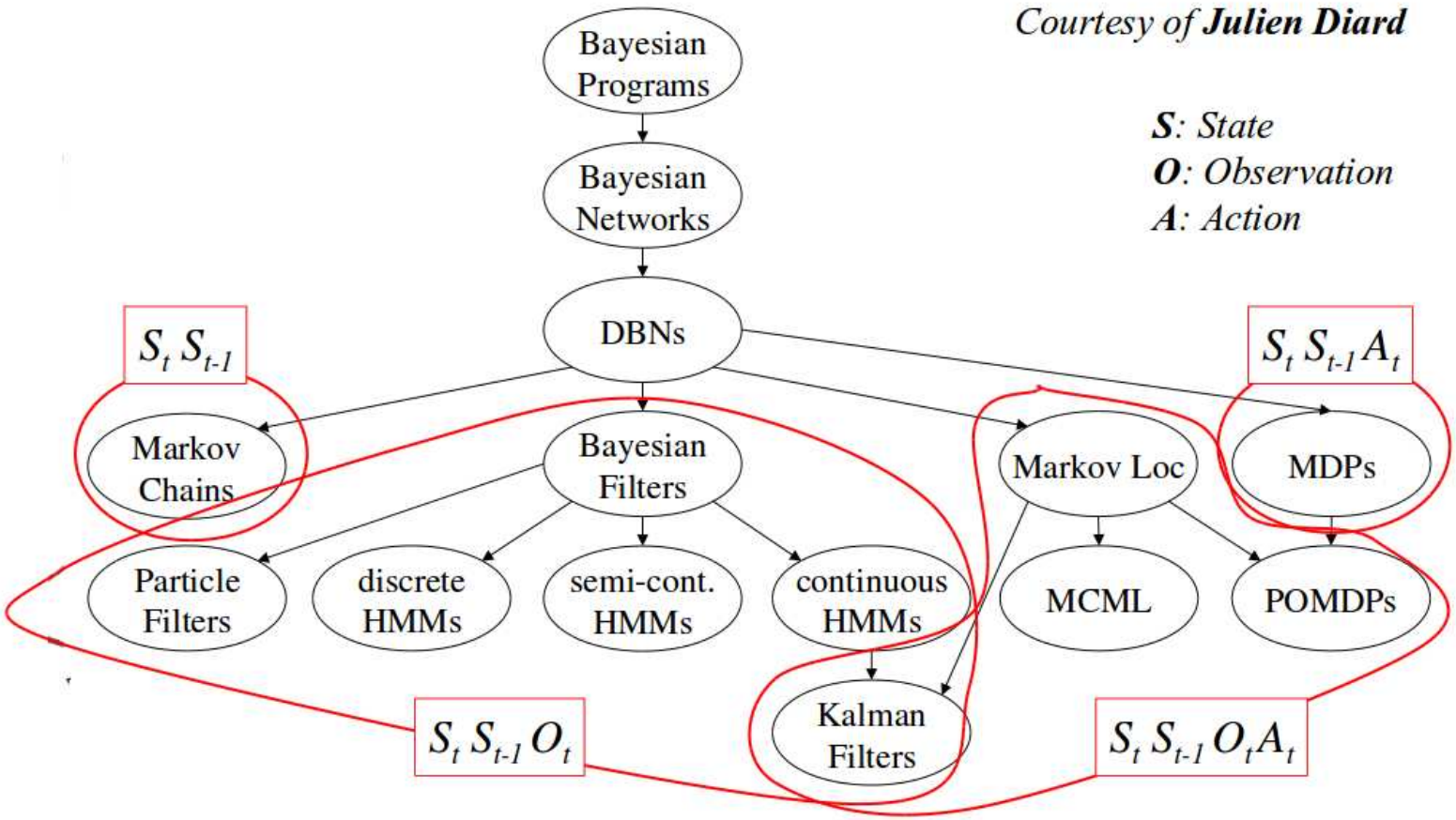
- Time/space complexity linear in the number of particles.

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

A general classification of dynamic probabilistic models



A general classification of dynamic probabilistic models



Summary

- This class 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.