PROBABILISTIC REASONING OVER TIME

Quiz information

- The first midterm quiz is on Tuesday (10/15)
- In-class (75 minutes)
- Coverage
  - AIMA Ch. 3-6
  - AIMA Ch.13-14
- Allowed one two-sided (8.5x11) cheat sheet
- Optional problems for practice
  - The solutions for optional problems on HW4 already posted on Piazza

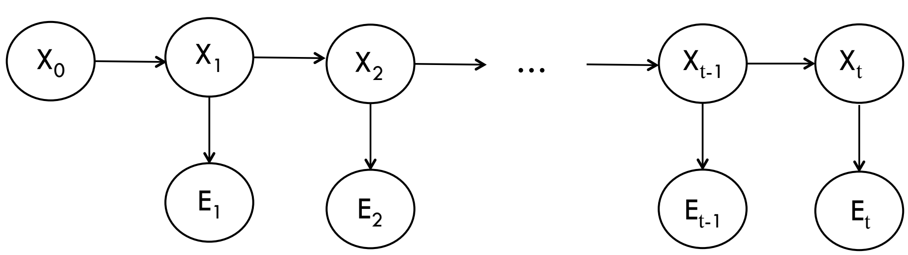
Not covered:
Chapter 2
Newton-Rhapson
Variable elimination
Gibbs sampling

Today

- Reading
  - AIMA Chapter 15.1-15.2, 15.5
- Goals
  - Types of inference
    - Filtering, prediction, smoothing, most likely explanation
    - Particle filters

Hidden Markov Model

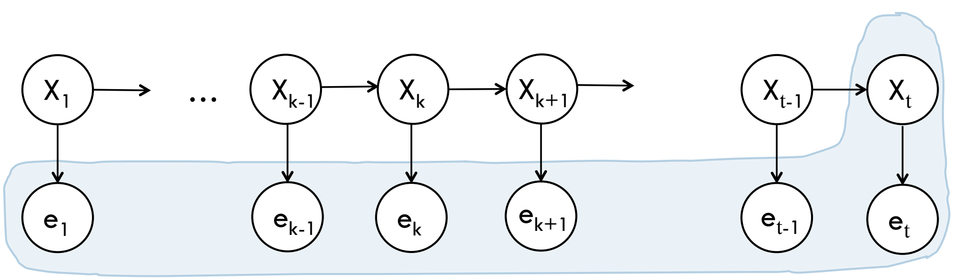
- Hidden Markov Models involve three things:
  - Transition model: $P(X_t | X_{t-1})$
  - Emission (evidence) model: $P(E_t | X_t)$
  - Prior probability: $P(X_0)$



Inference Tasks

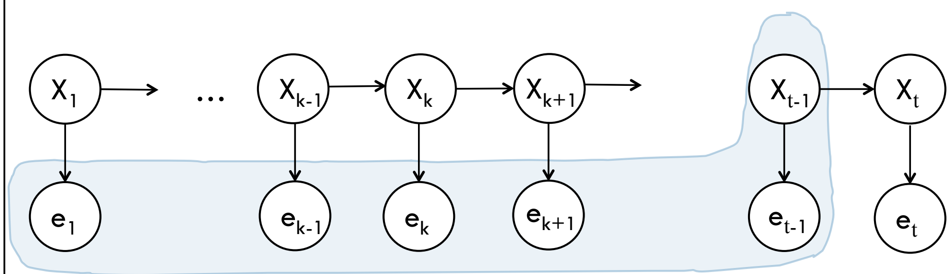
- Filtering: $P(X_t | e_{1:t})$
  - Decision making in the here and now
- Prediction: $P(X_{t+k} | e_{1:t})$
  - Trying to plan the future
- Smoothing: $P(X_k | e_{1:t})$ for $0 \leq k < t$
  - Gives a better (smoother) estimate than filtering by taking into account future evidence
- Most Likely Explanation (MLE): $\underset{x_{1:t}}{\operatorname{argmax}} P(x_{1:t} | e_{1:t})$
  - e.g., speech recognition, sketch recognition

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



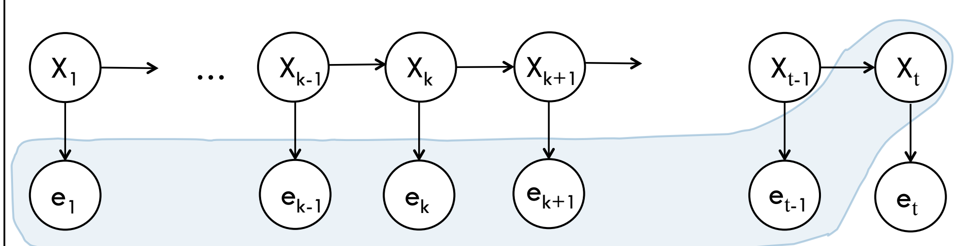
- A recursive state estimation algorithm

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



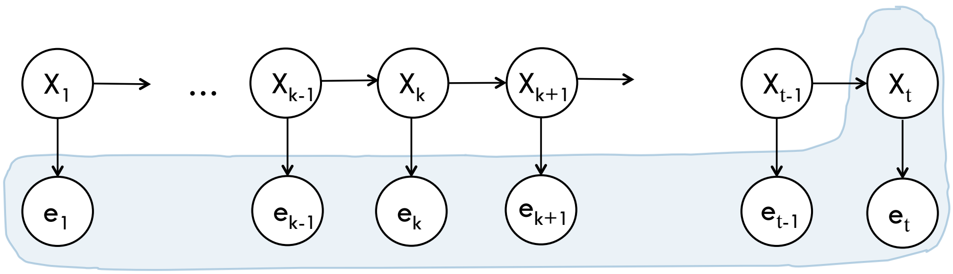
- Assume we already have $p(X_{t-1} | e_{1:t-1})$

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



- Update from state X_{t-1} to X_t

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



- Then incorporate the new evidence E_t

The Forward Algorithm

$$\begin{aligned} p(X_t|e_{1:t}) &= p(X_t|e_{1:t-1}, e_t) \\ &\propto p(e_t|X_t, e_{1:t-1})\ p(X_t|e_{1:t-1}) \\ &= \underbrace{p(e_t|X_t)}_{\text{Incorporate evidence}}\ \underbrace{p(X_t|e_{1:t-1})}_{\text{Update state}} \end{aligned}$$

The Forward Algorithm

$$\begin{aligned} p(X_t|e_{1:t}) &= p(X_t|e_{1:t-1}, e_t) \\ &\propto p(e_t|X_t, e_{1:t-1})\; p(X_t|e_{1:t-1}) \\ &= p(e_t|X_t)\; p(X_t|e_{1:t-1}) \end{aligned}$$

$$= p(e_t|X_t) \sum_{X_{t-1}} p(X_t, X_{t-1}|e_{1:t-1})$$

$$= p(e_t|X_t) \sum_{X_{t-1}} p(X_t|X_{t-1}, e_{1:t-1})\; p(X_{t-1}|e_{1:t-1})$$

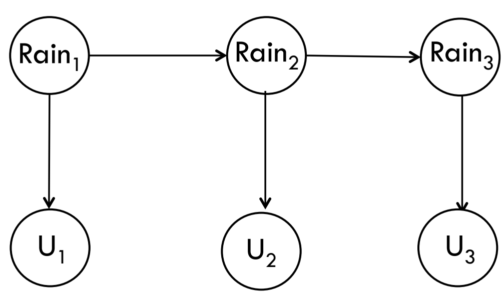
$$= \underbrace{p(e_t|X_t)}_{\text{Emission}} \sum_{X_{t-1}} \underbrace{p(X_t|X_{t-1})\; p(X_{t-1}|e_{1:t-1})}_{\text{Transmission + recursion}}$$

Filtering Example

$p(R_0) = <0.5, 0.5>$

R_{t-1}	$p(R_t \mid R_{t-1})$
T	0.7
F	0.3

R_t	$p(U_t \mid R_t)$
T	0.9
F	0.2

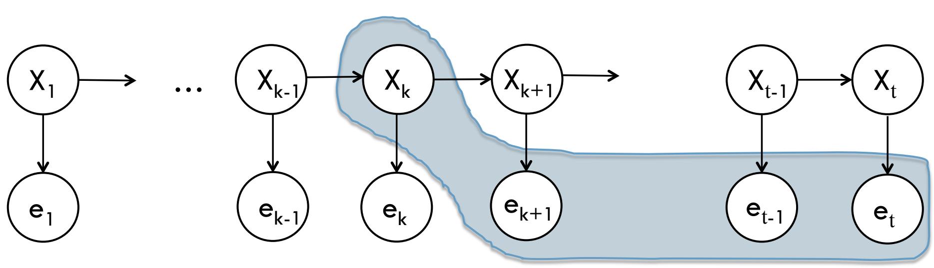


$$p(X_t|e_{1:t}) \propto p(e_t|X_t) \sum_{X_{t-1}} p(X_t|X_{t-1})\; p(X_{t-1}|e_{1:t-1})$$

Smoothing: $p(X_k \mid e_{1:t})$ for $1 \leq k < t$

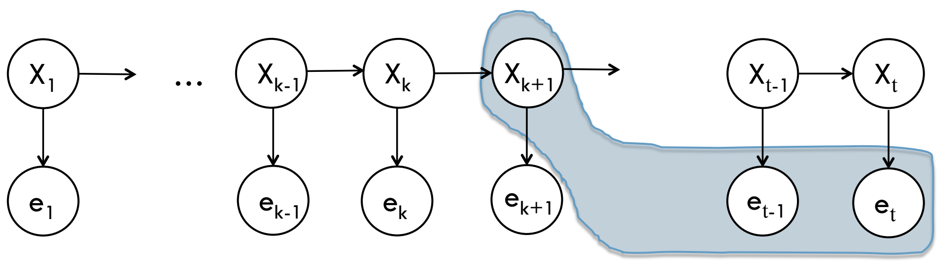
$$
\begin{aligned}
p(X_k|e_{1:t}) &= p(X_k|e_{1:k}, e_{k+1:t}) \\
&\propto p(X_k, e_{k+1:t}|e_{1:k}) \\
&= p(e_{k+1:t}|X_k, e_{1:k})\ p(X_k|e_{1:k}) \\
&= p(e_{k+1:t}|X_k)\ \underbrace{p(X_k|e_{1:k})}_{\text{Forward Algorithm}}
\end{aligned}
$$

The Backward Algorithm



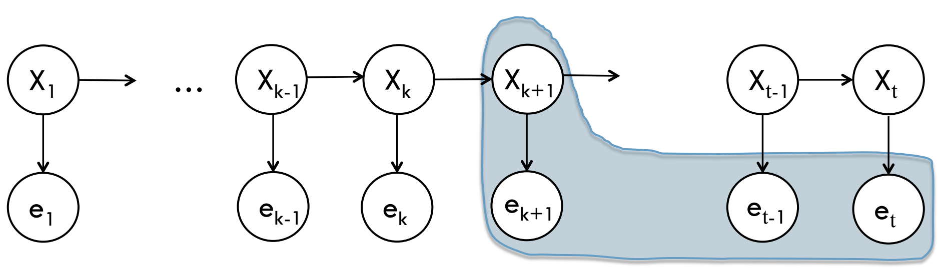
- A recursive state estimation algorithm

The Backward Algorithm



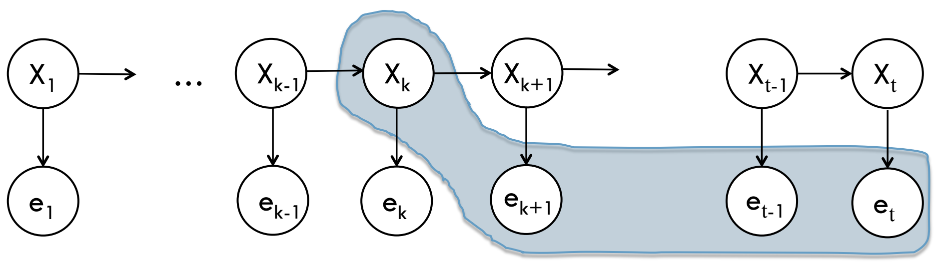
- Assume we have $p(X_{k+1} | e_{k+2:t})$

The Backward Algorithm



- Incorporate evidence via $p(e_{k+1} | X_{k+1})$

The Backward Algorithm



- Update the state via $p(X_{k+1} | X_k)$

Smoothing: $p(X_k | e_{1:t})$ for $1 \leq k < t$

The Forward Backward Algorithm

$$
\begin{aligned}
p(X_k|e_{1:t}) &= p(X_k|e_{1:k}, e_{k+1:t}) \\
&\propto p(X_k, e_{k+1:t}|e_{1:k}) \\
&= p(e_{k+1:t}|X_k, e_{1:k})\; p(X_k|e_{1:k}) \\
&= p(e_{k+1:t}|X_k)\; \underbrace{p(X_k|e_{1:k})}_{\text{Forward Algorithm}}
\end{aligned}
$$

$$
\begin{aligned}
p(e_{k+1:t}|X_k) &= \sum_{X_k+1} p(e_{k+1:t}, X_{k+1}|X_k) \\
&= \sum_{X_k+1} p(e_{k+1:t}|X_{k+1})\; p(X_{k+1}|X_k) \\
&= \sum_{X_k+1} \underbrace{p(e_{k+1}|X_{k+1})}_{\text{Emission}}\; \underbrace{p(e_{k+2:t}|X_{k+1})}_{\text{Recursion}}\; \underbrace{p(X_{k+1}|X_k)}_{\text{Transmission}}
\end{aligned}
$$

Filtering and Smoothing

- Filtering using the Forward algorithm

$$p(X_t|e_{1:t}) \propto p(e_t|X_t) \sum_{X_{t-1}} p(X_t|X_{t-1})\, p(X_{t-1}|e_{1:t-1})$$

- Smoothing uses the Forward and Backward algorithms

$$p(X_k|e_{1:t}) \propto p(e_{k+1:t}|X_k)\, p(X_k|e_{1:k}) \qquad \text{where}$$

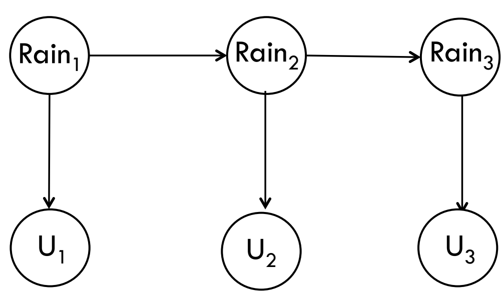
$$p(e_{k+1:t}|X_k) = \sum_{X_k+1} p(e_{k+1}|X_{k+1})\, p(e_{k+2:t}|X_{k+1})\, p(X_{k+1}|X_k)$$

Smoothing Example

$p(R_0)$ = <0.5, 0.5>

R_{t-1}	$p(R_t \mid R_{t-1})$
T	0.7
F	0.3

R_t	$p(U_t \mid R_t)$
T	0.9
F	0.2



$P(r_1 \mid u_1)$	$P(r_2 \mid u_1, u_2)$	$P(r_1 \mid u_1, u_2)$
0.818	0.883	?

Most Likely Explanation

- Find the state sequence that makes the observed evidence sequence most likely

$$\underset{X_{1:t}}{\operatorname{argmax}} P(X_{1:t} | e_{1:t})$$

- Recursive formulation:
  - The most likely state sequence for $X_{1:t}$ is the most likely state sequence for $X_{1:t-1}$ followed by the transition to X_t
  - Equivalent to Filtering algorithm except summation replaced with max
  - Called the Viterbi Algorithm

Dynamic Bayesian Networks

- Any BN that represents a temporal probability distribution using state variables and evidence variables is called a Dynamic Bayesian Network

- A Hidden Markov Model is the simplest type of DBN
  - State is represented by a single variable
  - Evidence is represented by a single variable
  - Applications
    - speech recognition
    - handwriting recognition
    - gesture recognition

Approximate Inference in Dynamic BN

- Recall approximate inference algorithms from previous lecture
  - Direct sampling, rejection sampling, likelihood weighting
  - Gibbs sampling
- Filtering in a DBN can be accomplished by applying likelihood weighting (with some modifications) to the DBN
- This is known as a Particle filter

Particle Filtering

- Likelihood weighting fixes the evidence variables and samples only the non-evidence variables
- Introduces a weight to correct for the fact that we're sampling from the prior distribution instead of the posterior distribution

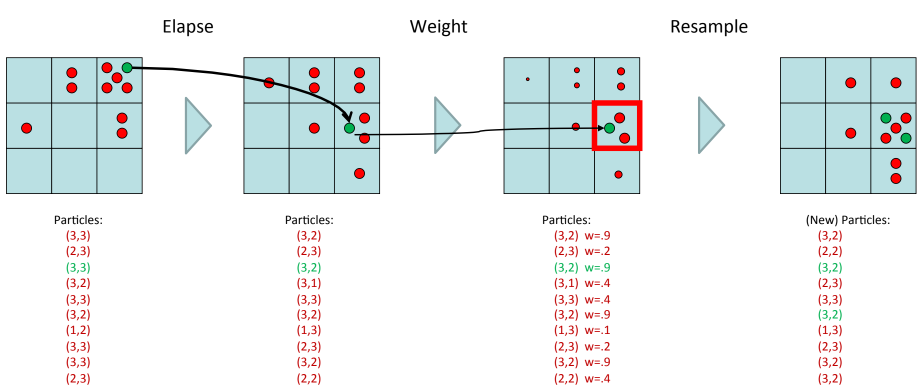
$$\text{weight} = p(e_1 \mid \text{Parents}(e_1)) * p(e_2 \mid \text{Parents}(e_2)) \ldots$$

Particle Filtering

- Initialize
  - Draw N particles (i.e. samples) for X_0 from the prior distribution $p(X_0)$
- Propagate
  - Propagate each particle forward by sampling $X_{t+1} \mid X_t$
- Weight
  - Weight each particle by $p(e_{t+1} \mid X_{t+1})$
- Resample
  - Generate N new particles by sampling proportional to the weights. The new particles are unweighted

Particle Filtering

- Particles: track samples of states rather than an explicit distribution



Compute $p(X_2 | U_2)$ using particle filter

- Step One: figure out how to sample from a discrete distribution?
  - Given a random number between [0,1] you can sample from any discrete distribution
- Step Two: Particle filtering
  - Draw N=10 particles from prior distribution
  - Propagate each particle forward by sampling $p(X_{t+1} | x_t)$
  - Weight each particle by $p(e_{t+1} | x_{t+1})$
  - Generate N=10 new particles by sampling proportional to the weights.