PROBABILISTIC REASONING OVER TIME

Today

- Reading
  - AIMA Chapter 15.1-15.2, 15.5
- Goals
  - Practice with filtering and smoothing
  - Particle Filtering

The Forward Algorithm

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

The Forward Backward Algorithm

The Forward Backward Algorithm

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

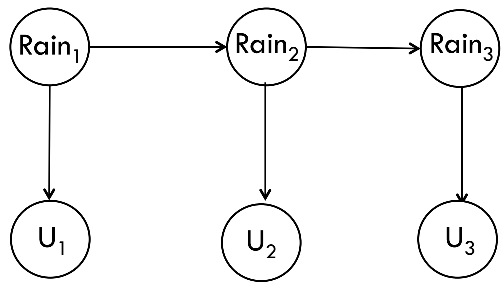
$$p(e_{k+1:t}|X_k) = \sum_{X_k+1} p(e_{k+1:t}, X_{k+1}|X_k)$$
$$= \sum_{X_k+1} p(e_{k+1:t}|X_{k+1})\ p(X_{k+1}|X_k)$$
$$= \sum_{X_k+1} \underbrace{p(e_{k+1}|X_{k+1})}_{\text{Emission}}\ \underbrace{p(e_{k+2:t}|X_{k+1})}_{\text{Recursion}}\ \underbrace{p(X_{k+1}|X_k)}_{\text{Transmission}}$$

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



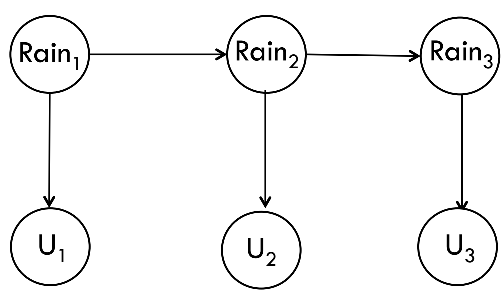
$$p(X_t|e_{1:t}) \propto p(e_t|X_t) \sum_{X_{t-1}} p(X_t|X_{t-1})\, p(X_{t-1}|e_{1:t-1})$$

Smoothing 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

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

- 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

Approximate Inference in Dynamic BN

- Recall approximate inference algorithms for general Bayesian Networks
  - Direct sampling, rejection sampling, likelihood weighting
  - Gibbs sampling

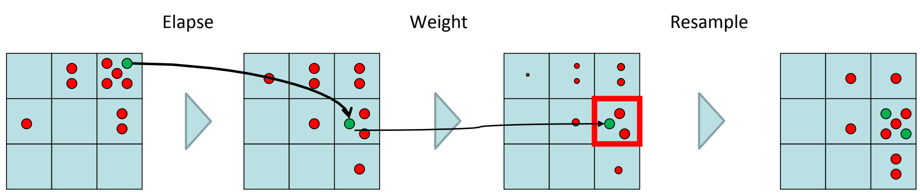
- Likelihood weighting applied to DBN (with some modifications) is known as a Particle filter

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 a value x_{t+1} from $p(X_{t+1} \mid X_t)$
- Weight
  - Weight each particle by $p(e_{t+1} | X_{t+1}=x_{t+1})$
- 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



Particles:	Particles:	Particles:	(New) Particles:
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,3)	(2,3) w=.2	(2,2)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(3,2)	(3,1)	(3,1) w=.4	(2,3)
(3,3)	(3,3)	(3,3) w=.4	(3,3)
(3,2)	(3,2)	(3,2) w=.9	(3,2)
(1,2)	(1,3)	(1,3) w=.1	(1,3)
(3,3)	(2,3)	(2,3) w=.2	(2,3)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,2)	(2,2) w=.4	(3,2)