PROBABILISTIC REASONING OVER TIME

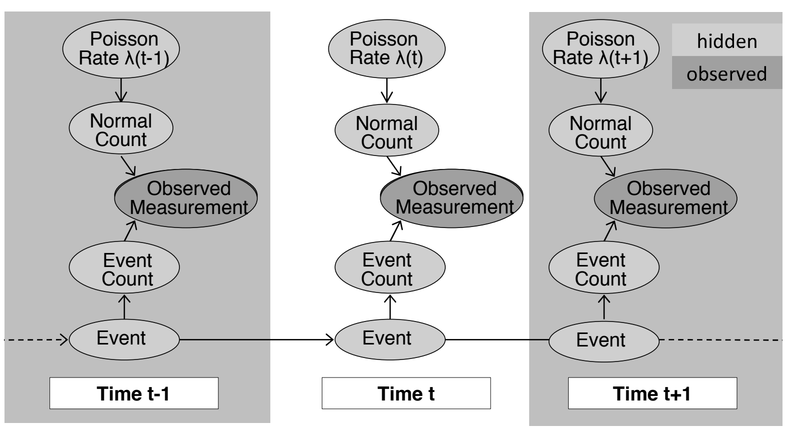
Today

- Reading
  - AIMA Chapter 15.1-15.2, 15.5
- Goals
  - Particle Filters
  - In-class inference practice

Modeling uncertainty over time

- Sometimes, we want to model a *dynamic* process: the value of the random variables change over time
  - Price of a stock
  - Patient stats, e.g. blood pressure, heart rate, blood sugar levels
  - Traffic on California highways
  - Pollution, humidity, temperature, rain fall, storms
  - Sensor tracking and detection

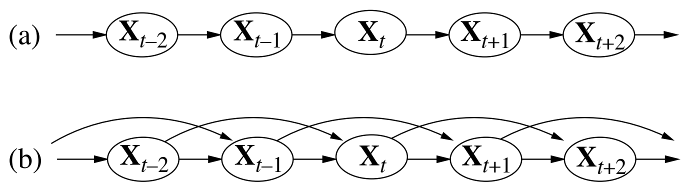
Examples of DBN



J. Hutchins et al. Probabilistic analysis of a large-scale urban traffic sensor data set.

Transition Model

- The transition model specifies the probability of X_t given the history $X_{0:t-1}$
- Markov Assumption: the state variable X_t depends on a *finite and fixed* subset of $X_{0:t-1}$
  - First order Markov Process: $P(X_t | X_{0:t-1}) = P(X_t | X_{t-1})$
  - Second order Markov Process: $P(X_t | X_{0:t-1}) = P(X_t | X_{t-1}, X_{t-2})$



Transition Model

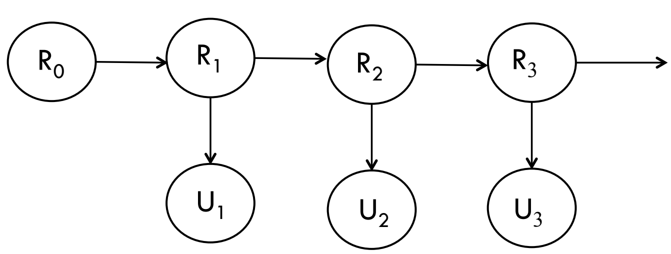
- Stationarity Assumption: the conditional distribution $P(X_t | X_{t-1})$ is the same for all t
- Only need to specify one conditional distribution for all edges

Sensor (emission) model

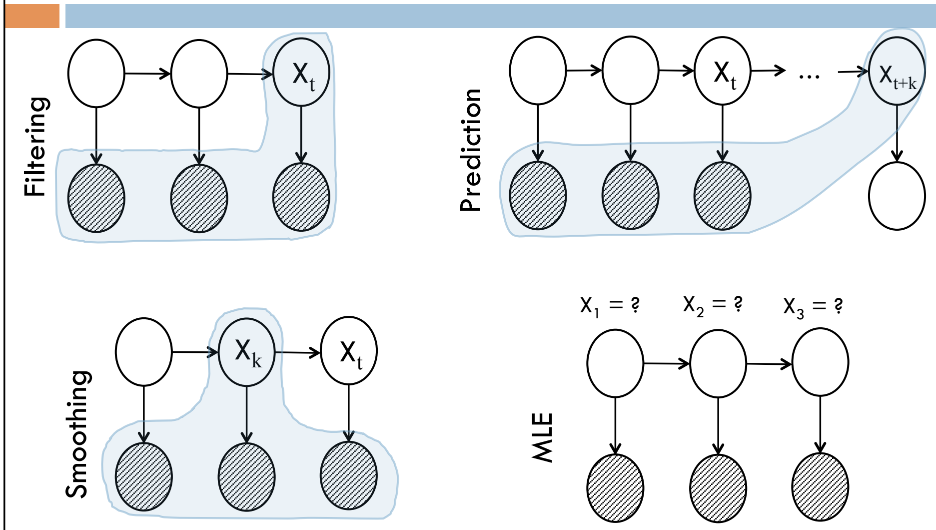
- The sensor model specifies the probability of the evidence E_t given the states $X_{0:t}$ and the evidence $E_{0:t-1}$
- Sensor Markov Assumption: the evidence E_t is independent of every other random variable given X_t
  - The state encompasses all relevant information for generating the evidence

Hidden Markov Model

- A Hidden Markov Model is the simplest type of DBN
  - State is represented by a single discrete random variable



Inference Tasks



Approximate Inference in Dynamic BN

- Recall approximate inference algorithms from previous lecture
  - Direct sampling, rejection sampling, likelihood weighting
  - Gibbs sampling

- Likelihood weighting applied to DBN (with some modifications) 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

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

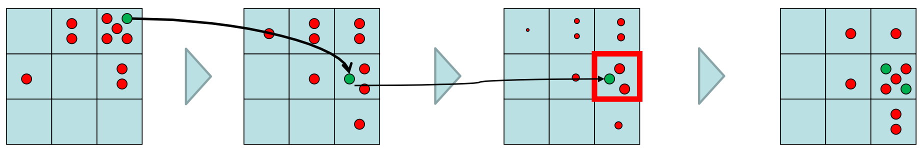
Particle Filtering

- Initialize
  - Draw S 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 $p(e_{t+1} \mid X_{t+1}=x_{t+1})$
- Resample
  - Generate S 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

Elapse | Weight | Resample



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)