

Lecture 19: April 1, 2015
cs 573: Probabilistic Reasoning
Professor Nevatia
Spring 2015

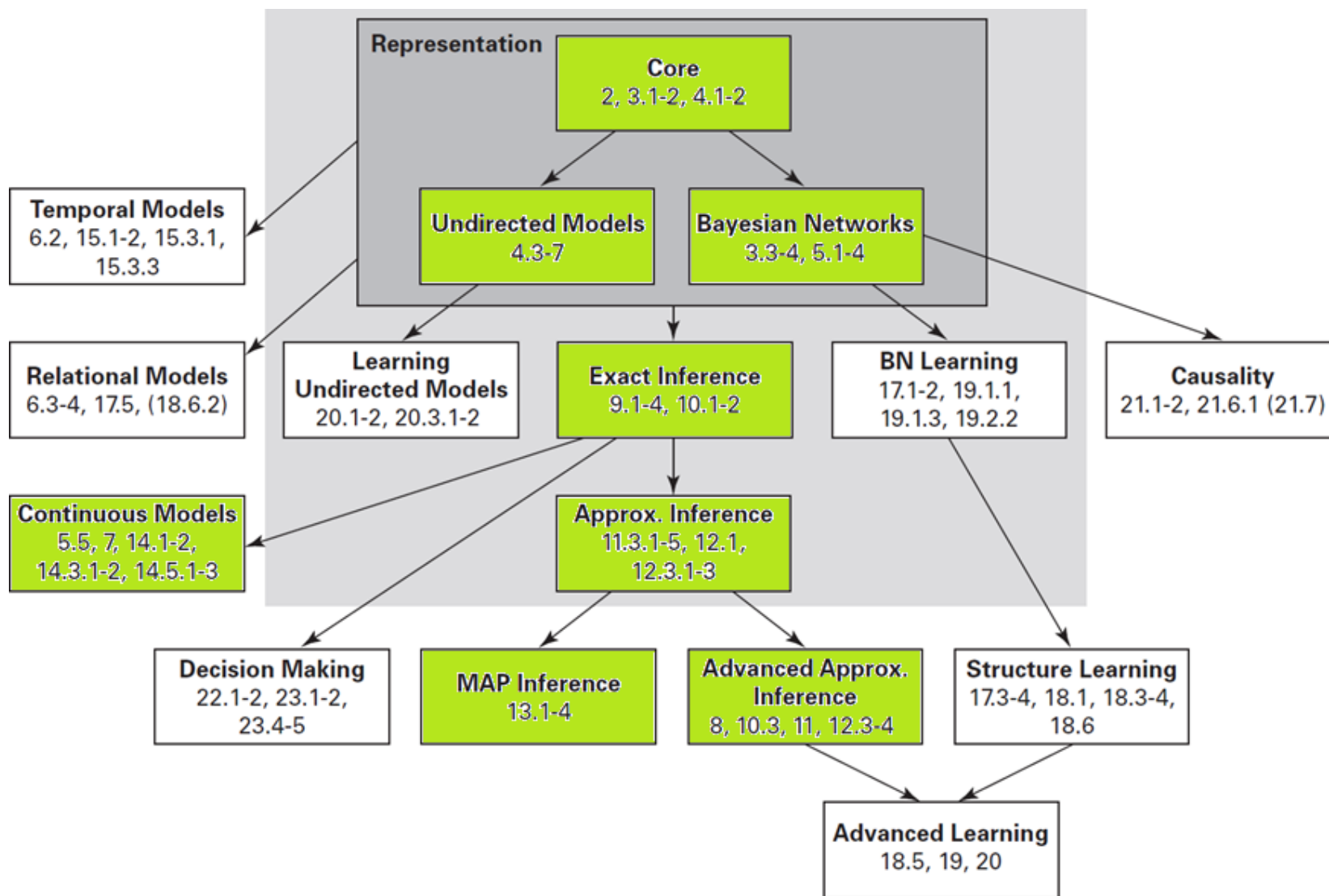
Review

- HW #6A posted, due 4/8/15
- Previous Lecture
 - MCMC
 - Metropolis-Hastings Algorithm
 - Mixing time
- Today's objective
 - Intro to temporal models

MCMC: Mixing

- How can we detect if a chain has “mixed”?
 - Distributions should be “stationary” but some variations can be expected from the distribution itself for a small number of samples
 - Distributions can also be stationary because the chain is in some high probability part of the space
 - We can initiate multiple chains, with different initialization, and test if they converge to the same distribution.
- Parallel MCMC
 - Inherently, MCMC is a sequential process.
 - We can parallelize, to some extent, by using multiple chains
 - Some parallel MCMC algorithms exist; out of scope of our course.

Book Plan



Temporal Reasoning

- Temporal models are natural in a variety of domains
 - Estimate robot's position over time (based on some observations)
 - Monitor a patient in ICU (series of measurements such as blood pressure and heart rate)
 - Infer disease from temporal data (EKG)
 - Infer activity patterns in a video stream
 - Infer words from a speech (audio) stream
 - Predict weather, earth quakes, stock market...
 - Number of variables can grow very large in temporal reasoning
 - In general, any variable at one time instance may have a direct influence on any other variable at any other time, and the influence itself could be a function of time.
 - Useful to have compact representations where possible.
- Also, we , make some simplifying assumptions for tractability.

Temporal Models (Chapter 6)

- System state at time t is an assignment to a the system variables (hidden or observed) at time t
- Notation: $X_i^{(t)}$ represents instantiation of X_i at time t
 - X_i is *not* a random variable that takes values, rather it is a *template variable*
 - Template is instantiated at different times; $X_i^{(t)}$ takes values in $\text{val}(X_i)$
- For a set of variables, \mathbf{X} , we use notation $\mathbf{X}^{(t_1:t_2)}$ ($t_1 < t_2$) to denote the set of variables $\{\mathbf{X}^{(t)} : t \in [(t_1, t_2)]\}$
 - $\mathbf{x}^{(t_1:t_2)}$ represents the assignment to this set of variables
- Trajectory: assignment to each variable $X_i^{(t)}$ for each relevant time
 - Space of trajectories is huge, need simplifying assumptions

Example

- Vehicle state: location, velocity, weather, failure (of the sensor), obs (observation of the sensor)
- One such set at each time t
- Joint probability distribution over all the sets (for some time interval) defines a probability distribution over a system trajectory (not in the same sense as the trajectory of car's position)
- Given a sequence of observations, we can answer questions about the distributions of other variables

Basic Assumptions

- Time is discretized (time slices)
- $P(\mathbf{X}^{(0:T)})$ short hand for $P(X^{(0)}, X^{(1)}, \dots, X^{(T)})$

- Using chain rule:

$$P(\mathbf{X}^{(0:T)}) = \left(\prod_{t=0, T-1} P(X^{(t+1)} | \mathbf{X}^{(0:t)}) \right) P(X^{(0)})$$

- Markov assumption if $(\{X^{(t+1)} \perp \mathbf{X}^{(0:(t-1))}\} | X^{(t)})$
 - Next state is conditionally independent of previous states given the current state
- Distribution given Markov assumption

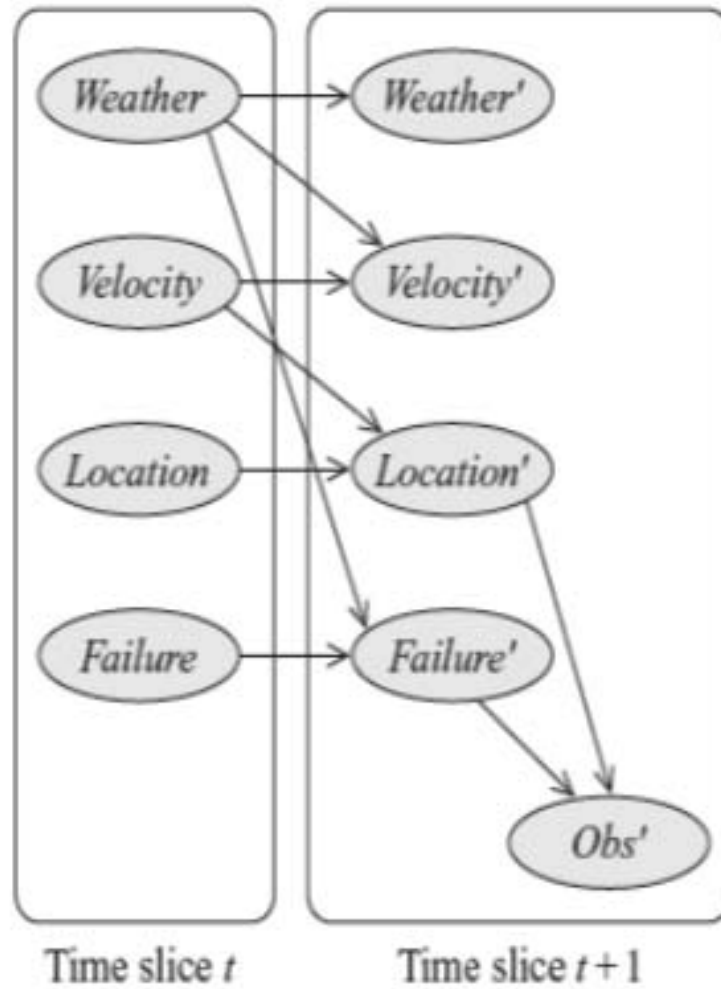
$$P(\mathbf{X}^{(0:T)}) = \left(\prod_{t=0, T-1} P(X^{(t+1)} | X^{(t)}) \right) P(X^{(0)}) \quad (\text{eq 6.1})$$

Basic Assumptions

- Vehicle example
 - If variables are just the location l and observation of location o_l , Markov property is not accurate as it does not include info about speed
 - Change in speed may depend on weather, so also include weather
 - Adding more variables in state creates better approximation to Markovian properties
- *Stationary* dynamical system: $\mathbf{P} (\mathbf{X}^{(t+1)} | \mathbf{X}^{(t)})$ same for all t
- Transition model $P(\mathbf{X}' | \mathbf{X})$

$$P(\mathcal{X}^{(t+1)} = \xi' | \mathcal{X}^{(t)} = \xi) = P(\mathcal{X}' = \xi' | \mathcal{X} = \xi).$$

2-TBN Example



(a) $\mathcal{B}_{\rightarrow}$

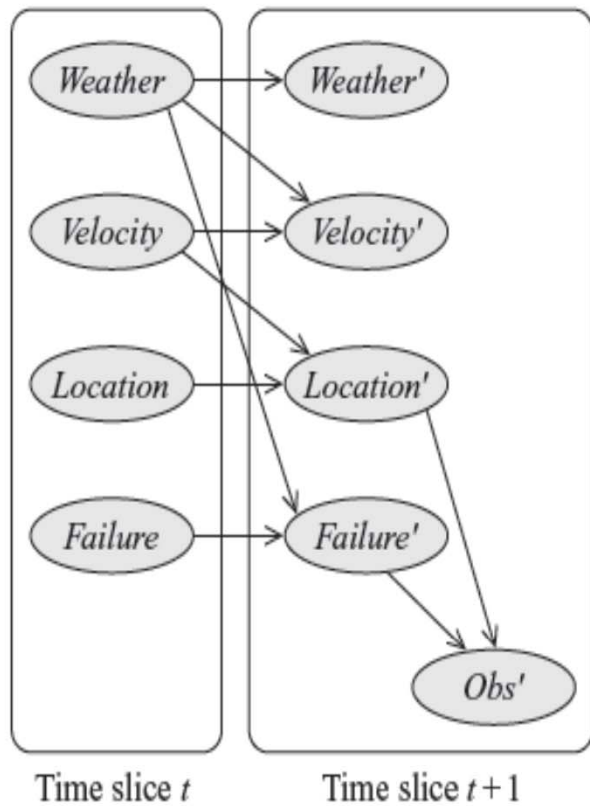
2-TBN

- 2-time slice Bayesian Network
 - Conditional Bayesian network over X' given X_I , X_I is a subset of X and is called the set of *interface* variables
- In the given example, all variables, except O , are in the interface
- Represents the following conditional distribution

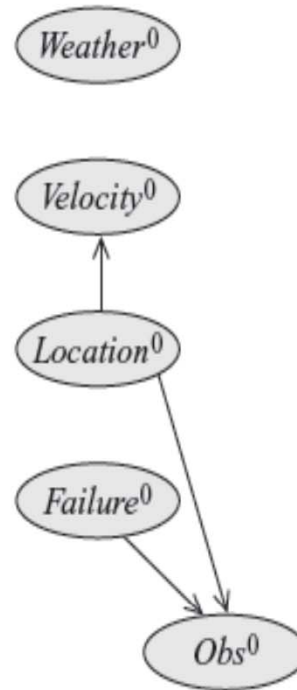
$$P(\mathcal{X}' | \mathcal{X}) = P(\mathcal{X}' | \mathcal{X}_I) = \prod_{i=1}^n P(X'_i | \text{Pa}_{X'_i}).$$

- Term inside product is called a template factor; it is instantiated multiple times.

Dynamic Bayesian Networks (DBNs)

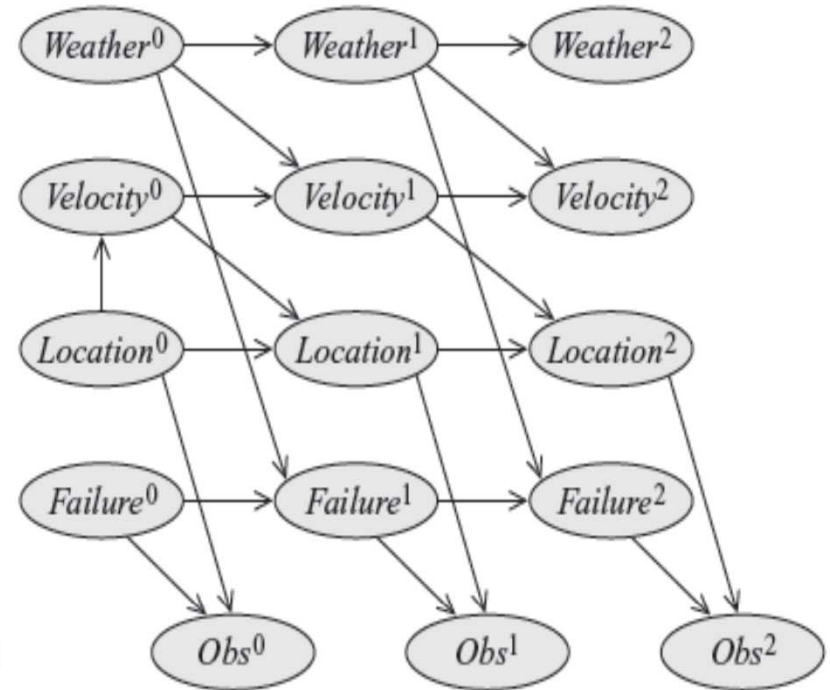


(a) $\mathcal{B}_{\rightarrow}$



Time slice 0

(b) \mathcal{B}_0



Time slice 0

Time slice 1

Time slice 2

(c) DBN unrolled over 3 steps

Dynamic Bayesian Network (DBN)

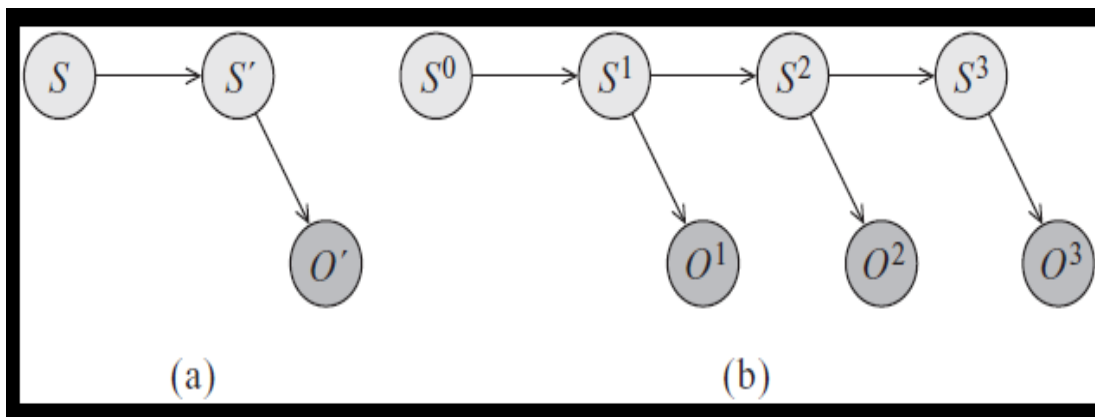
- A dynamic Bayesian network (DBN) is a pair (B_0, B_{\rightarrow})
- B_0 is a BN over $X^{(0)}$, representing initial state distribution
- B_{\rightarrow} is a 2-TBN process
- For $T \geq 0$, the distribution over $X^{(0:T)}$ is defined as an unrolled BN where for any $i = 1, \dots, n$:
 - Structure and CPDs of $X_i^{(0)}$ are same as for X_i in B_0
 - Structure and CPDs of $X_i^{(t)}$ for $t > 0$ are same as for X_i' in B_{\rightarrow}
- Generates an infinite set of BNs, one for every $T > 0$

Edges in DBN

- *Inter-time-slice* edges: edges across time slices
- *Intra-time-slice* edges: edges within a time slice
- *Persistence* edges: inter time-slice edges of the form $X \rightarrow X'$ (influence persists). A variable for which such an edge exists is called a *persistent* variable.
- Given an initial state, we can *unroll* the network over a desired interval

Hidden Markov Models (HMMs)

- A special case of a DBN
 - A single state variable and a single observation variable



State-Observation Models

- An alternative to the DBN view of a temporal model
- State evolves on its own
 - Follows a Markovian *transition* model: next state is conditionally independent of previous states given the current state
 - Transition model $P(X' | X)$
- Given the current state, observations are conditionally independent of the rest of the state sequence (past and future)
 - Observation model $P(O | X)$
- 2-TBN where O' are all leaf nodes, parents of O' only in X'
- Any 2-TBN can be transformed into such a representation (construction given in the book) but may hide some structure.
- 2 special cases
 - Linear Dynamical Systems
 - Hidden Markov Models (HMMs)

Linear Dynamical Systems

- All variables are continuous; dependencies are linear Gaussian

$$P(\mathbf{X}^{(t)} \mid \mathbf{X}^{(t-1)}) = N(\mathbf{A}\mathbf{X}^{(t-1)}; \mathbf{Q})$$

$$P(\mathbf{O}^{(t)} \mid \mathbf{X}^{(t)}) = N(\mathbf{H}\mathbf{X}^{(t)}; \mathbf{R})$$

(note the deterministic analog)

If \mathbf{X} is n -vector, \mathbf{O} is an m -vector, \mathbf{A} is $n \times n$ matrix, \mathbf{H} is $n \times m$

N is normal distribution, \mathbf{Q} and \mathbf{R} are noise matrices

Such systems commonly studied in estimation theory and control theory

A common objective is to estimate \mathbf{X} given a sequence of observations \mathbf{O}

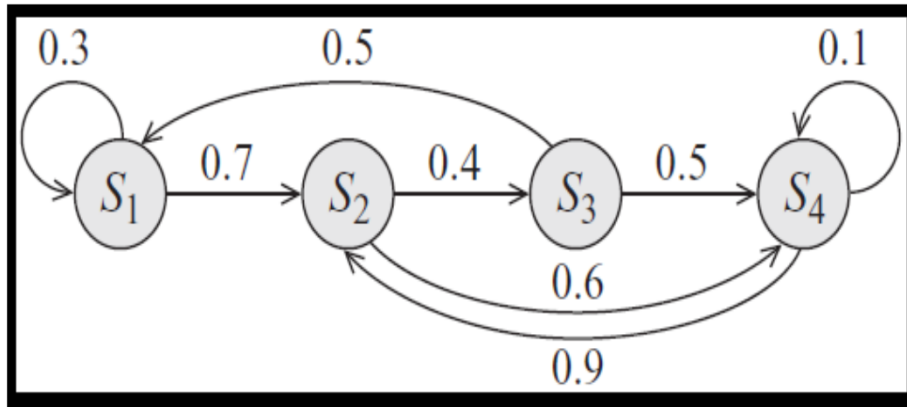
\Rightarrow Kalman-Bucy Filter provides the optimal estimate

Extensions to non-linear dynamics

Hidden Markov Models (HMMs)

- One of the most important tools for many practical problems, e.g. speech recognition
- An HMM is a DBN but the transition model $P(S_t | S_{t-1})$ is typically sparse
- State transitions can be represented as a graph
 - Note: this graph is NOT a Bayesian network, nodes do not represent random variables but *states* (or possible values of a state variable)
 - Arrows represent possible transitions and their probabilities
 - *Probabilistic finite state automation*
- Observation model is not encoded in the graph; it is associated with states separately

HMM Example



	s_1	s_2	s_3	s_4
s_1	0.3	0.7	0	0
s_2	0	0	0.4	0.6
s_3	0.5	0	0	0.5
s_4	0	0.9	0	0.1

Speech Recognition

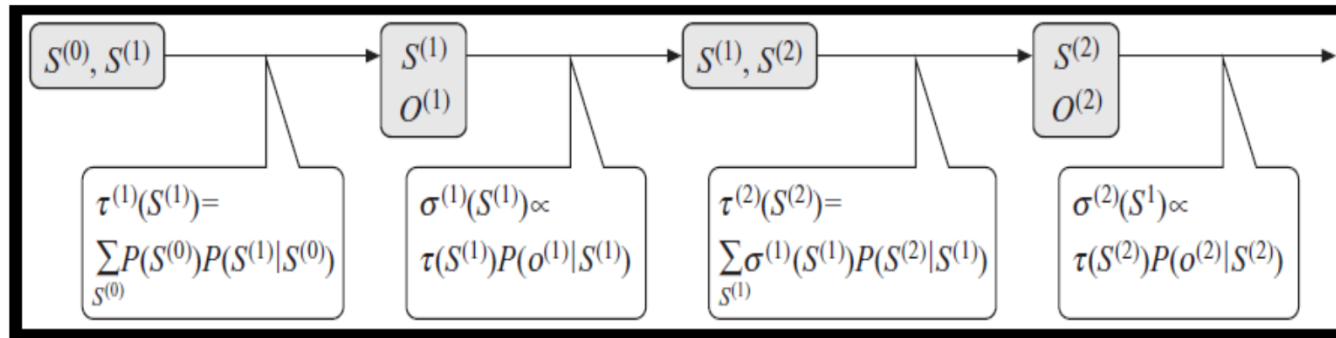
- Speech Recognition: identify sequence of words from acoustic signal.
- *Speech understanding* requires inference of *meaning*
 - Speech recognition is a necessary first step
- Prime example of utility of HMMs
- Given *signal*, find *words* that maximize $P(\text{words}|\text{signal})$
 - $P(\text{words}|\text{signal}) = P(\text{signal}|\text{words}) * P(\text{words})$
- $P(\text{words})$ is the *language* model
 - Probability of words and word sequences
- $P(\text{signal}|\text{words})$ is the acoustic model
- Slides from Russel-Norvig, AIMA book (section 23.5)
 - Details of speech will not be tested on the Exam.
 - See separate slides in Speech HMM.pdf

Inference Tasks in Temporal Models

- *Filtering*: distribution of current state, given all the observations so far: $\mathbf{P} (\mathbf{X}^{(t)} \mid \mathbf{O}^{(1:t)})$
- *Prediction*: probability distribution over \mathbf{X} at time $t' > t$, given $\mathbf{O}^{(1:t)}$.
- *Smoothing*: Compute probabilities of a state given all the evidence: $\mathbf{P} (\mathbf{X}^{(t)} \mid \mathbf{O}^{(1:T)})$
- *Most likely trajectory* (sequence of states) given all the observations: $\arg \max_{\xi^{(0:T)}} \mathbf{P} (\xi^{(0:T)} \mid \mathbf{O}^{(1:T)})$
 - Viterbi algorithm solves the last query

Exact Inference: State-Observation Models

- Can view unrolled model as any other graph: construct a clique tree for it and perform inference on it



- Filtering requires just the forward pass: normalize at each step
- Prediction is inference without evidence node (prediction can be from the start or after some time t)
- Smoothing requires a backward pass also: multiply messages and normalize
- Trajectory: solve the MAP problem
- Book derives recursive formula for forward pass but omits details of others. Instead, we look at more details from the Russell-Norvig book (slides should be self-contained)

Filtering Equations

- Notation: belief state $\sigma^{(t)}(\mathbf{X}^{(t)}) = \mathbf{P}(\mathbf{X}^{(t)} \mid \mathbf{O}^{(1:t)})$
- Goal: compute $\sigma^{(t+1)}$ from $\sigma^{(t)}$ (recursive algorithm)
- Notation: $\sigma^{(\cdot, t+1)}(\mathbf{X}^{(t+1)}) = \mathbf{P}(\mathbf{X}^{(t+1)} \mid \mathbf{O}^{(1:t)})$; \cdot says evidence up to time t only (prior belief state)
- $\sigma^{(\cdot, t+1)}(\mathbf{X}^{(t+1)}) = \mathbf{P}(\mathbf{X}^{(t+1)} \mid \mathbf{O}^{(1:t)}) =$

$$= \sum_{\mathbf{x}^t} \mathbf{P}(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(t)}, \mathbf{O}^{(1:t)}) \mathbf{P}(\mathbf{X}^{(t)} \mid \mathbf{O}^{(1:t)})$$

$$= \sum_{\mathbf{x}^t} \mathbf{P}(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(t)}) \sigma^{(t)}(\mathbf{X}^{(t)})$$
- Add the effect of $\mathbf{O}^{(t+1)}$:

$$\begin{aligned} \sigma^{(t+1)}(\mathbf{X}^{(t+1)}) &= P(\mathbf{X}^{(t+1)} \mid o^{(1:t)}, o^{(t+1)}) \\ &= \frac{P(o^{(t+1)} \mid \mathbf{X}^{(t+1)}, o^{(1:t)}) P(\mathbf{X}^{(t+1)} \mid o^{(1:t)})}{P(o^{(t+1)} \mid o^{(1:t)})} \\ &= \frac{P(o^{(t+1)} \mid \mathbf{X}^{(t+1)}) \sigma^{(\cdot, t+1)}(\mathbf{X}^{(t+1)})}{P(o^{(t+1)} \mid o^{(1:t)})}. \end{aligned}$$

- Basically, multiply by observation probability and normalize

Next Class

- Read sections 15.2 and 15.3 of the KF book