HMMs

CompSci 370

Ronald Parr
Department of Computer Science
Duke University

Overview

- Bayes nets are (mostly) atemporal
- Need a way to talk about a world that changes over time
- Necessary for planning
- Many important applications
  - Target tracking
  - Patient/factory monitoring
  - Speech recognition

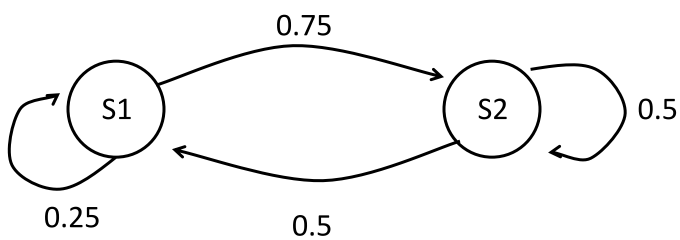
Back to Atomic Events

- We began talking about probabilities from the perspective of atomic events
- An atomic event is an assignment to every random variable in the domain
- For n binary random variables, there are 2^n possible atomic events

States

- When reasoning about time, we often call atomic events states
- States, like atomic events, form a mutually exclusive and jointly exhaustive partition of the space of possible events
- We can describe how a system behaves with a state-transition diagram

State Transition Diagram



P(S2|S1)=0.75
P(S1|S1)=0.25
P(S2|S2)=0.50
P(S1|S2)=0.50

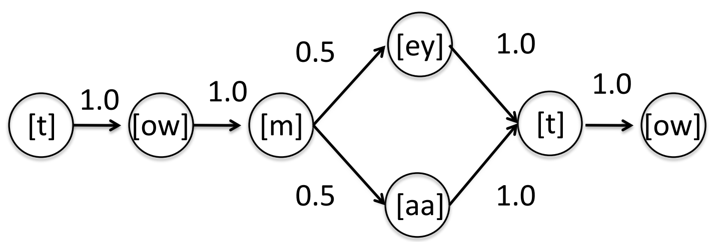
Don't confuse states with state variables!
Don't confuse states with state variables!
Don't confuse states with state variables!

Note: Time indices are implicit, really $P(S_{t+1}=S2|S_t=S1)$, etc.

Example: Speech Recognition

- Speech is broken down into atoms called phonemes, e.g., see arpanet: http://en.wikipedia.org/wiki/Arpabet
- Phonemes are pulled from the audio stream using a variety of techniques
- Words are stochastic finite automata (HMMs) with outputs that are phonemes

You say tomato, I say…



Real variations in speech between speakers can be much more subtle and complicated than this: How do we learn these?

Fun on Mac OS

- `say tomato`
- `say "[[inpt PHON]] tUXmAAtOW [[inpt TEXT]]"`
- `say "[[inpt PHON]] tUXmEYtOW [[inpt TEXT]]"`

Using HMMs for Speech Recognition

- Create one HMM for every word
- Upon hearing a word:
  - Break down word into string of phonemes
  - Compute probability that string came from each HMM
  - Go with word (HMM) that assigns highest probability to string

State Transition Diagrams

- Make a lot of assumptions
  - Transition probabilities don't change over time (*stationarity*)
  - The event space does not change over time
  - Probability distribution over next states depends only on the current state (*Markov assumption*)
  - Time moves in uniform, discrete increments

The Markov Assumption

- Let S_t be a random variable for the state at time t
- $P(S_t|S_{t-1},\ldots,S_0) = P(S_t|S_{t-1})$
- (Use subscripts for time; S0 is different from S_0)
- Markov is special kind of conditional independence
- Future is independent of past given current state

Markov Models

- A system with states that obey the Markov assumption is called a *Markov Model*
- A sequence of states resulting from such a model is called a *Markov Chain*
- The mathematical properties of Markov chains are studied heavily in mathematics, statistics, computer science, electrical engineering, etc.

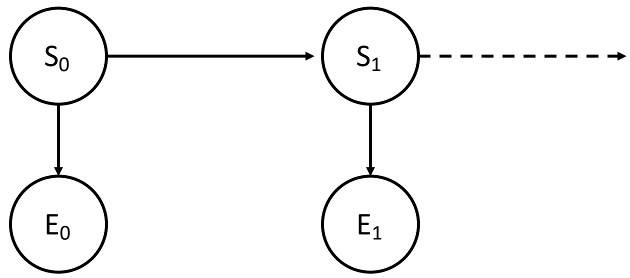
What's The Big Deal?

- A system that obeys the Markov property can be described succinctly with a transition matrix, where the i,jth entry of the matrix is P(Sj|Si)
- The Markov property ensures that we can maintain this succinct description over a potentially infinite time sequence
- Properties of the system can be analyzed in terms of properties of the transition matrix
  - Steady-state probabilities
  - Convergence rate, etc.

Observations

- Introduce E_t for the observation at time t
- Observations are like evidence
- Define the probability distribution over observations as function of current state: P(E|S)
- Assume observations are conditionally independent of other variables given current state
- Assume observation probabilities are stationary
- Note: In MDPs, we assume that every state has a unique observation associated with it, so the true state is always known

A Bayes Net View of HMMs



Note: These are random variables, not states!

Applications

- Monitoring/Filtering: $P(S_t:E_0 \ldots E_t)$
  - S is the current status of the patient/factory
  - E is the current measurement
- Prediction: $P(S_t:E_0 \ldots E_k)$, $t>k$
  - S is the current/future position of an object
  - E are our past observations
  - Project S into the future

Applications

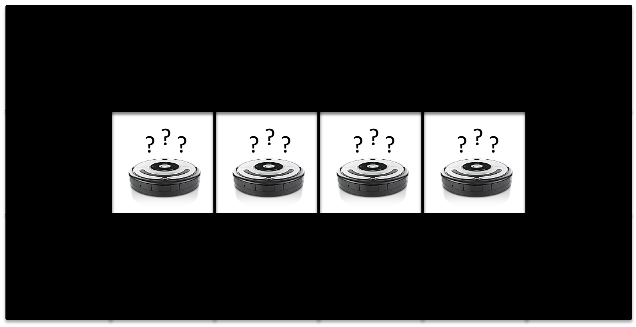
- Smoothing/hindsight: $P(S_k:E_0 \ldots E_t)$, $t>k$
  - Update view of the past based upon future
  - Diagnosis: Factory exploded at time t=20, what happened at t=5 to cause this?
- Most likely explanation
  - What is the most likely sequence of events (from start to finish) to explain observations?
  - NB: Answer is a single path, not a distribution

Example: Robot Self Tracking

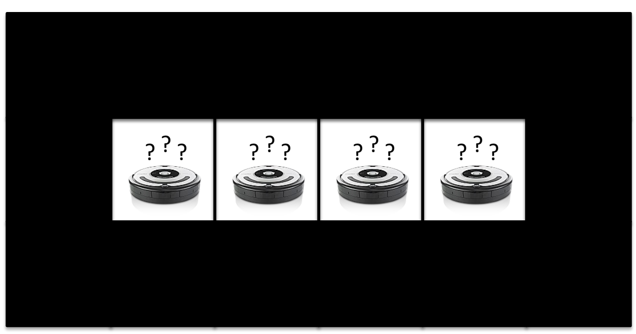
- Consider Roomba-like robot with:
  - Known map of the room
  - 4-way proximity sensors
  - Unknown initial position (kidnapped robot problem)
- We consider a discretized version of this problem
  - Map discretized into grid
  - Discrete, one-square movements

(Images from iRobot's web page)

Simple Map, Kidnapped Robot

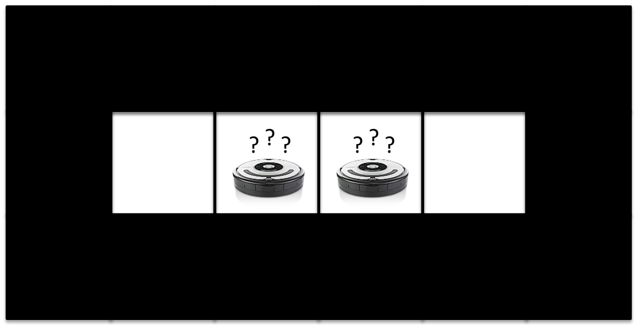


Robot Senses

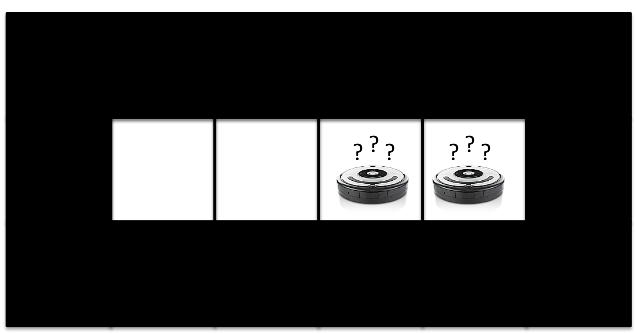


Obstacles up and down, none left and right

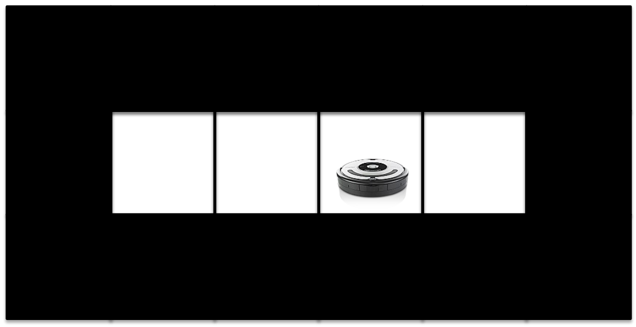
Robot Updates Distribution



Robot Moves Right, Updates



Robot Updates Probabilities



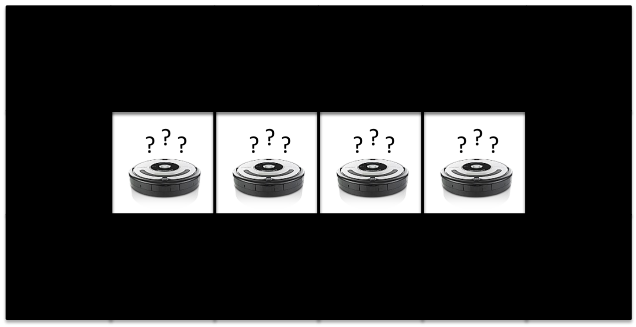
Obstacles up and down, none left and right

What Just Happened

- This was an example of robot tracking

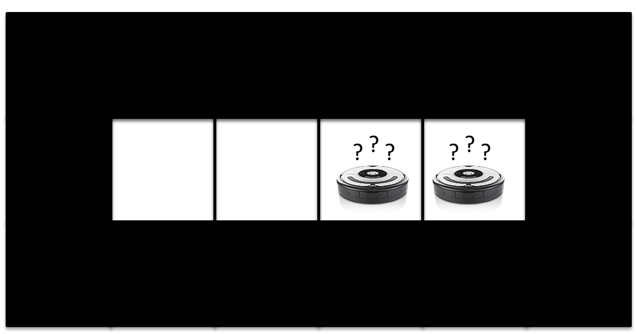
- We can also do:
  - Prediction (where would the robot be?)
  - Smoothing (where was the robot?)
  - Most likely path (what path did robot take?)

Prediction



Suppose the Robot Moves Right Twice

New Robot Position Distribution



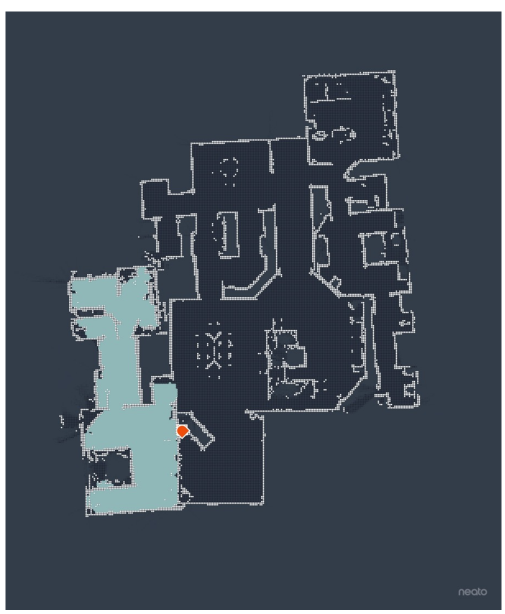
Are these probabilities uniform?

What Isn’t Realistic Here?

- Where does the map come from?
- Does the robot really have these sensors?
- Are right/left/up/down the correct sort of actions? (Even if the robot has a map, it may not know its orientation.)
- Are robot actions deterministic?
- Are sensing actions deterministic?
- Would a probabilistic sensor model conflate sensor noise and incorrect modeling?
- Can the world be modeled as a grid?

- Good news: Despite these problems, robotic mapping and localization (tracking) can actually be made to work!

…and it really is used:



The Most Likely (Viterbi) Path

- How many paths are there through the state space?
  - For n states, T time steps
  - n^T possible paths
- How do we maximize over this efficiently?
- Idea:
  - For each time time step t, store a table of size n such that $P_t(s)$ = probability of highest probability path reaching state s at time t
  - Compute P_{t+1} from P_t
  - Only need previous time step because of Markov property

Implementing the Viterbi Algorithm
(forward part)

- P_0=initial distribution
- For t=1 to T
  - P_0 = uniform or some given initial distribution
  - For NextS = 1 to n
    - P_t[NextS]=0
    - For PrevS = 1 to n
      - P_t[NextS] = max{P_t[NextS],P_{t-1}[PrevS]*P(NextS|PrevS)}
    - P_t[NextS] = P_t[NextS]*P(e_t|NextS)

What is is needed: Store argmax, reconstruct path in backward pass (compare with reconstructing the path in search)

Viterbi Path Algebraic View

From definition of Bayes net (or HMM):

$$P(S_0 \ldots S_t \mid e_0 \ldots e_t) \propto P(S_0)P(e_0 \mid S_0)\prod_{i=1}^{t} P(S_i \mid S_{i-1})P(e_i \mid S_i)$$

Suppose we want max probability sequence of states:

$$\max_{S_0 \ldots S_t} P(S_0 \ldots S_t \mid e_0 .. e_t) = \max_{S_0 \ldots S_t} P(S_0)P(e_0 \mid S_0)\prod_{i=1} P(S_i \mid S_{i-1})P(e_i \mid S_i)$$

$$= \max_{S_1 \ldots S_t} P(e_t \mid S_t)\prod_{i=1}^{t-1} P(S_{i+1} \mid S_i)P(e_i \mid S_i) \max_{S_0} P(S_1 \mid S_0)P(S_0)P(e_0 \mid S_0)$$

$$= \max_{S_2 \ldots S_t} P(e_t \mid S_t)\prod_{i=2}^{t-1} P(S_{i+1} \mid S_i)P(e_i \mid S_i) \max_{S_1} P(S_2 \mid S_1)P(e_1 \mid S_1) \max_{S_0} P(S_1 \mid S_0)P(S_0)P(e_0 \mid S_0)$$

Keep distributing max over product!

Compare with Dijkstra's algorithm, dynamic programming.

Bayes Rule Reminder

$$P(A \wedge B) = P(B \wedge A)$$

$$P(A \mid B)P(B) = P(B \mid A)P(A)$$

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Conditional Probability with Extra Evidence

- Recall: P(AB)=P(A|B)P(B)
- Add extra evidence C
  (can be a set of variables)
- P(AB|C)=P(A|BC)P(B|C)

Extending Bayes Rule

$$P(A \mid BC) = \frac{P(B \mid AC)P(A \mid C)}{P(B \mid C)}$$

How to think about this: The C is like "extra" evidence.
This forces us into one corner of the event space.
Given that we are in this corner, everything behaves the same.

Using Conditional Independence And the Markov Property

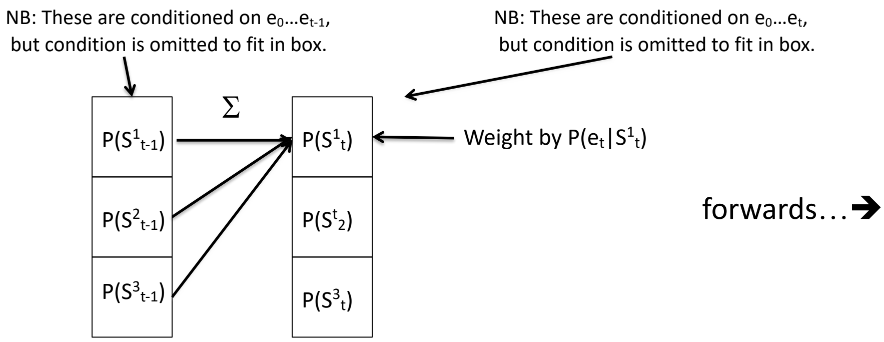
- Conditional probability w/extra evidence:
  - $P(AB|C)=P(A|BC)P(B|C)$

- $P(S_t S_{t-1} | e_{t-1} e_0) = P(S_t | S_{t-1} e_{t-1} e_0)\ P(S_{t-1} | e_{t-1} e_0)$
  $= P(S_t | S_{t-1})\ P(S_{t-1} | e_{t-1} e_0)$

Monitoring

- Given evidence up to time t, what is the probability of being in some state s at time t?
- Equivalent to: What is the **sum** of the probabilities of all paths that end in state s at time t given evidence up to time t.
- How do we compute this efficiently?
- Idea:
  - For each time time step t, store a table of size n such that $P(s_t|e_t \ldots e_0)$ = sum or probabilities of all paths reaching state s at time t
  - Compute $P(s_{t+1}|e_{t+1} \ldots e_0)$ from $P(s_t|e_t \ldots e_0)$
  - Only need previous time step because of Markov property

Implementation



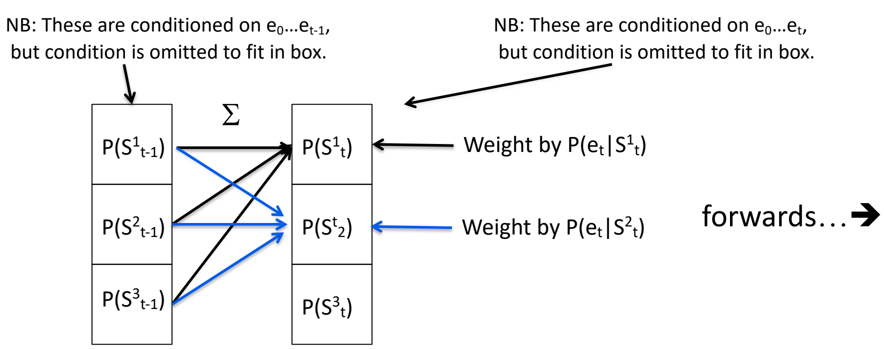
Maintain a vector of probabilities at each time step

Arcs correspond $P(s_i|s_{i-1})$ in summation of previous slide:

- Each color is a different iteration through the loop
- Add up probability of all paths that lead to each state

Initialization: Typically an initial distribution is given for time step 0 and there are no observations for time step 0.

Implementation

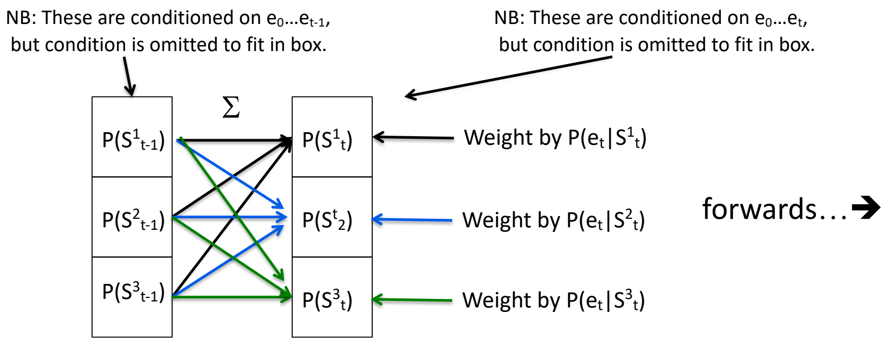


Maintain a vector of probabilities at each time step

Arcs correspond $P(s_i|s_{i-1})$ in summation of previous slide:

- Each color is a different iteration through the loop
- Add up probability of all paths that lead to each state

Implementation



Maintain a vector of probabilities at each time step

Arcs correspond $P(s_i|s_{i-1})$ in summation of previous slide:

- Each color is a different iteration through the loop
- Add up probability of all paths that lead to each state

Monitoring Derivation

We want: $P(S_t|e_t\ldots e_0)$

$$P(S_t \mid e_t..e_0) = \frac{P(e_t \mid S_t, e_{t-1}..e_0)P(S_t \mid e_{t-1}..e_0)}{P(e_t \mid e_{t-1}..e_0)}$$

$$= \alpha P(e_t \mid S_t e_{t-1}..e_0)P(S_t \mid e_{t-1}..e_0)$$

$$= \alpha P(e_t \mid S_t)P(S_t \mid e_{t-1}..e_0)$$

$$= \alpha P(e_t \mid S_t)\sum_{S_{t-1}} P(S_t \mid S_{t-1})P(S_{t-1} \mid e_{t-1}..e_0)$$

Recursive

Example

- W = employee is working
- R = employee has produced results
- supervisor observes whether employee has produced results
- Infer whether employee is working given observations

$$P(w_{t+1} \mid w_t) = 0.8$$
$$P(w_{t+1} \mid \overline{w}_t) = 0.3$$
$$P(r \mid w) = 0.6$$
$$P(r \mid \overline{w}) = 0.2$$

Problem

- Assume employee starts job in a productive (working) state
- Supervisor has observed two consecutive meetings without results
- What is probability the employee was working in the second week?

Let's Do The Math

$P(w_{t+1} | w_t) = 0.8$
$P(w_{t+1} | \overline{w}_t) = 0.3$
$P(r | w) = 0.6$
$P(r | \overline{w}) = 0.2$

$$P(W_2 | \overline{r}_2 \overline{r}_1) = \alpha_1 P(\overline{r}_2 | W_2) \sum_{W_1} P(W_2 | W_1) P(W_1 | \overline{r}_1)$$
$$P(W_1 | \overline{r}_1) = \alpha_2 P(\overline{r}_1 | W_1) \sum_{W_0} P(W_1 | W_0) P(W_0)$$
$$P(w_1 | \overline{r}_1) = \alpha_2 0.4(0.8*1.0 + 0.3*0.0) = \alpha_2 0.32$$
$$P(\overline{w}_1 | \overline{r}_1) = \alpha_2 0.8(0.2*1.0 + 0.7*0.0) = \alpha_2 0.16$$
$$P(w_1 | \overline{r}_1) = 0.67, P(\overline{w}_1 | \overline{r}_1) = 0.33$$

More Math

$P(w_{t+1} | w_t) = 0.8$
$P(w_{t+1} | \overline{w}_t) = 0.3$
$P(r | w) = 0.6$
$P(r | \overline{w}) = 0.2$
$P(w_1 | \overline{r}_1) = 0.67$
$P(\overline{w}_1 | \overline{r}_1) = 0.33$

$$P(W_2 | \bar{r}_2 \bar{r}_1) = \alpha_1 P(\bar{r}_2 | W_2) \sum_{W_1} P(W_2 | W_1) P(W_1 | \bar{r}_1)$$
$$P(w_2 | \bar{r}_2 \bar{r}_1) = \alpha_1 0.4(0.8 * 0.67 + 0.3 * 0.33) = \alpha_1 0.25$$
$$P(\overline{w}_2 | \bar{r}_2 \bar{r}_1) = \alpha_1 0.8(0.2 * 0.67 + 0.7 * 0.33) = \alpha_1 0.292$$
$$P(w_2 | \bar{r}_2 \bar{r}_1) = 0.46, P(\overline{w}_2 | \bar{r}_2 \bar{r}_1) = 0.54$$

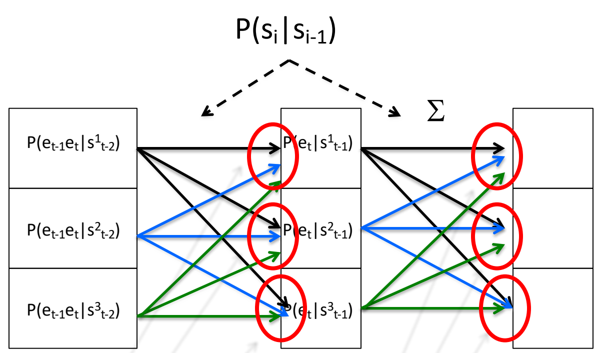
Hindsight (Smoothing)

- Given evidence up to time t, what is the probability of being in some state s at time k<t?
- Equivalent to:
  - What is the **sum** of the probabilities of all paths that end in state s at time k given evidence up to time k…
  - Weighted by all of the observations after time k.
- How do we compute probability of subsequent observations efficiently?
  - Idea:
  - For each time time step k<j<T, store a table of size n such that $P(e_t \ldots e_{j+1} | S_j)$ = probability of all evidence after time j starting from each state at time j
  - Compute from $P(e_t \ldots e_j | S_{j-1})$ from $P(e_t \ldots e_{j+1} | S_j)$ (work backwards!)
  - Only need **subsequent** time step because of Markov property

Implementation

Black, blue, green are different iterations through loop implied by summation on previous slide

← …backwards



Weight by $P(e_{t-1} | s_1)$
Weight by $P(e_{t-1} | s_2)$
Weight by $P(e_{t-1} | s_3)$

Weight by $P(e_t | s_1)$
Weight by $P(e_t | s_2)$
Weight by $P(e_t | s_3)$

There is no e_{t+1}!
What does this mean?
Can assume all ones.

Hindsight Algebra

$$P(S_k \mid e_t..e_0) = \alpha P(e_t..e_{k+1} \mid S_k, e_k..e_0)P(S_k \mid e_k..e_0)$$

$$= \alpha P(e_t..e_{k+1} \mid S_k)\boxed{P(S_k \mid e_k..e_0)}$$

Monitoring!

$$P(e_t..e_{k+1} \mid S_k) = \sum_{S_{k+1}} P(e_t..e_{k+1} \mid S_k S_{k+1})P(S_{k+1} \mid S_k)$$

$$= \sum_{S_{k+1}} P(e_t..e_{k+1} \mid S_{k+1})P(S_{k+1} \mid S_k)$$

$$= \sum_{S_{k+1}} P(e_{k+1} \mid S_{k+1})P(e_t..e_{k+2} \mid S_{k+1})P(S_{k+1} \mid S_k)$$

Recursive

Hindsight (smoothing) Summary

- Forward: Compute time k state distribution given
  - Forward distribution up to k
  - Observations up to k
  - Equivalent to monitoring up to k
- Backward: Compute conditional evidence distribution after k
  - Work backward from t to k
- Smoothed state distribution is ***proportional*** to product of forward and backward components
  (normalize to get true probabilities)

Implementation Sanity Checks

- Make sure you never double count observations: Any *path* through the HMM should multiply by each $P(e_i|s_i)$ exactly once
  (think of forward/backward as summing probabilities of paths, weighted by observations)
- Make sure you handle base cases
  - Forward message starts with initial distribution at time 0
  - Observations beyond the horizon can be ignored
    (or assume first backwards message is all ones)

Problem II

Can we revise our estimate of the probability that the employee worked at step 1?

We initially thought:

$$P(w_1 \mid \bar{r}_1) = 0.67, P(\overline{w}_1 \mid \bar{r}_1) = 0.33$$

Since the employee didn't have results at time 2, is it now less likely that they were working at time 1?

Let's Do More Math

$P(w_{t+1} \mid w_t) = 0.8$
$P(w_{t+1} \mid \overline{w}_t) = 0.3$
$P(r \mid w) = 0.6$
$P(r \mid \overline{w}) = 0.2$
$P(w_1 \mid \overline{r}_1) = 0.67$
$P(\overline{w}_1 \mid \overline{r}_1) = 0.33$

$$P(W_1 \mid \overline{r}_2\overline{r}_1) = \alpha P(W_1 \mid \overline{r}_1) P(\overline{r}_2 \mid W_1)$$
$$P(\overline{r}_2 \mid w_1) = \sum_{W_2} P(\overline{r}_2 \mid W_2) P(W_2 \mid w_1)$$

Sums probabilities of all ways of making step 2 observation given w_1

$$P(\overline{r}_2 \mid w_1) = (0.4*0.8 + 0.8*0.2) = 0.48$$
$$P(\overline{r}_2 \mid \overline{w}_1) = (0.4*0.3 + 0.8*0.7) = 0.68$$
$$P(w_1 \mid \overline{r}_2\overline{r}_1) = \alpha 0.67*0.48 = \alpha 0.3216$$
$$P(\overline{w}_1 \mid \overline{r}_2\overline{r}_1) = \alpha 0.33*0.68 = \alpha 0.2244$$
$$P(w_1 \mid \overline{r}_2\overline{r}_1) = 0.59, P(\overline{w}_1 \mid \overline{r}_2\overline{r}_1) = 0.41$$

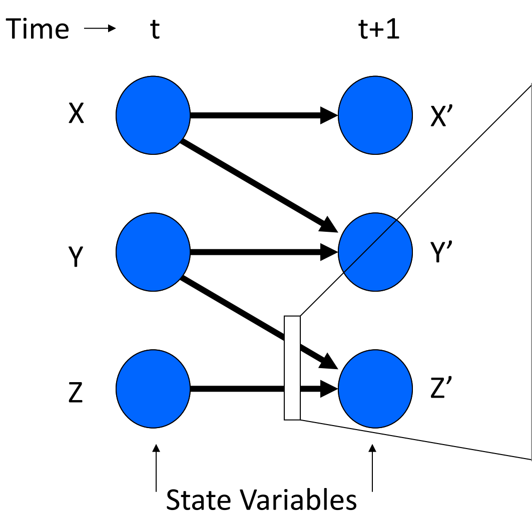
Checkpoint

- Done: Forward Monitoring and Backward Smoothing
- Monitoring is recursive from the past to the present
- Backward smoothing requires two recursive passes (forward then backward)
- Implemented as two loops (not recursively)
- Called the forward-backward algorithm
  - Independently discovered many times throughout history
  - Was classified for many years by US Govt.

What's Left?

- We have seen that filtering and smoothing can be done efficiently, so what's the catch?
- We're still working at the level of atomic events
- There are too many atomic events!
- We need a generalization of Bayes nets to let us think about the world at the level of state variables and not states

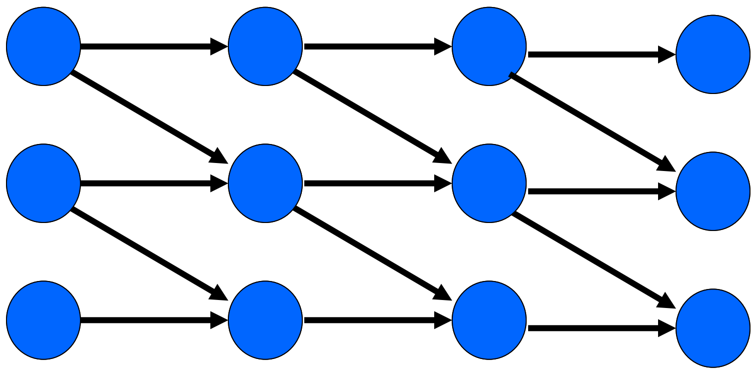
Dynamic Bayes Nets



CPT

	$P(z')$	$P(\bar{z}')$
$y_t z_t$	0.25	0.75
$y_t \bar{z}_t$	0.5	0.5
$\bar{y}_t z_t$	0.6	0.4
$\bar{y}_t \bar{z}_t$	0.3	0.7

Working With DBNs



Can we do variable elimination for DBNs?

Harsh Reality

- While BN inference in the static case was a very nice story, there are essentially no tractable, exact algorithms for DBNs
- Dealing with intractability
  - Approximate inference algorithms
    - Variational methods
    - Assumed density filtering (ADF)
  - Sampling methods
    - Sequential Importance sampling
    - Sequential Importance Sampling with Resampling (SISR, particle filter, condensation, etc.)

Continuous Variables

(outside of scope of class)

- How do we represent a probability distribution over a continuous variable?
  - Probability density function
  - Summations become integrals
- Very messy except for some special cases:
  - Distribution over variable X at time t+1 is a multivariate normal with a mean that is a linear function of the variables at the previous time step
  - This is a linear-Gaussian model

Inference in Linear Gaussian Models

- Filtering and smoothing integrals have closed form solution
- Elegant solution known as the Kalman filter
  - Used for tracking projectiles (radar)
  - State is modeled as a set of linear equations
    - S=vt
    - V=at
  - What about pilot controls?

HMM Conclusion

- Elegant algorithms for temporal reasoning over discrete atomic events, Gaussian continuous variables
  (many practical systems are approximately such)

- Exact Bayes net methods don't generalize well to state variable representation in the the temporal case: little hope for exponential savings

- Approximations required for large/complex/continuous systems