HMMs

CSCI 2951-F

Ronald Parr
Brown University

When Observations != States

- So far, assume that agent knows the state at any time
- Small exceptions:
  - Stacking frames in Atari
  - HW3 Q1 stochastic policy case

- These slides:
  - Not directly about RL/Planning
  - First step towards that: How to manage state uncertainty

Hidden Markov Models (HMMs)

- HMMs are like MDPs with features **but**
- Mapping from states to features is surjective-only not bijective
- AKA state aliasing
- State aliasing violates the Markov property
- So, what do we do?

HMM Assumptions

- Underlying Markovian state
- Known observation probabilities given state (Called O for observation, or E for evicence)
- Either of the following is a sufficient statistic:
  - Conditioning future predictions on history
  - Conditioning future predictions on distribution over underlying Markovian state

Histories vs. Distributions

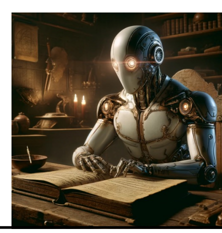
- Histories:
  - Conceptually easy
  - Potentially unbounded length

- Distributions over underlying states
  - Fixed length
  - More work to maintain

Historical Perspective on Histories

- Histories have waxed and waned in popularity
- Small histories viewed as tractable – when applicable (e.g. frame stacking in Atari)
- Long histories viewed as awkward/intractable due to variable size, exponential number of possible histories

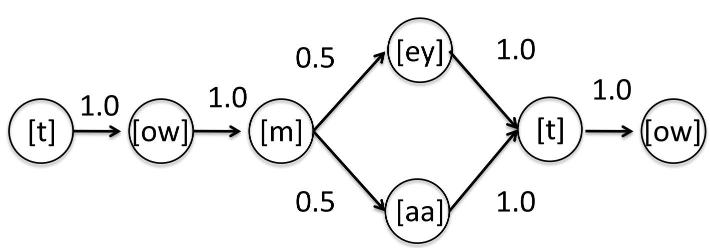
- Transformers have made history fashionable



Example: Speech Recognition

- Speech is broken down into atoms called phonemes, e.g., see arpabet: 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?

Phoneme Fun on Mac OS

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

(Sadly, appears to be broken in latest MacOS release,
but try təˈmɑːtəʊ and təˈmeɪˌtoʊ here: http://ipa-reader.xyz)

Using HMMs for (single word) 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
  - Pick word (HMM) assigning highest probability to string
- Note HMMs can be used both generatively (generate words sounds) and discriminatively (recognize/label word sounds)

Common 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

Common Applications

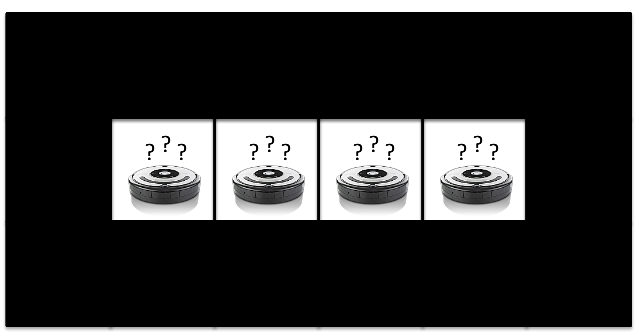
- Smoothing/hindsight: $P(S_k:E_0 \ldots E_t)$, $t>k$
  - Update view of the past based upon future
  - Diagnosis: Factory exploded at time t=20, what happened at t=5 to cause this?
- Most likely explanation
  - What is the most likely sequence of events (from start to finish) to explain observations?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

Example: Robot Self Tracking

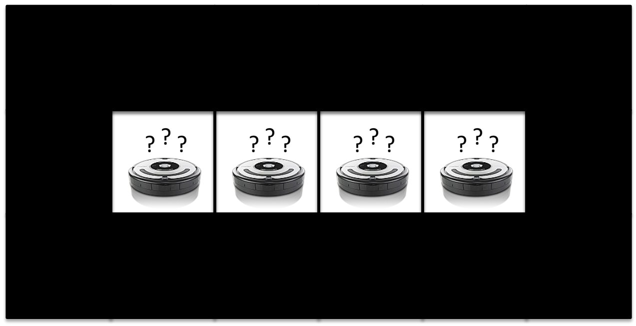
- Consider Roomba-like robot with:
  - Known map of the room
  - 4-way proximity sensors
  - Unknown initial position (kidnapped robot problem)
- We consider a discretized version of this problem
  - Map discretized into grid
  - Discrete, one-square movements

(Images from iRobot's web page)

Simple Map, Kidnapped Robot

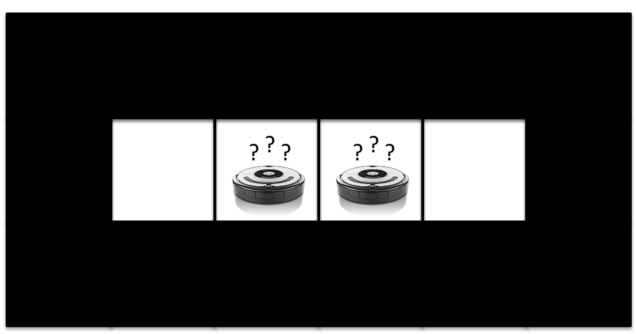


Robot Senses

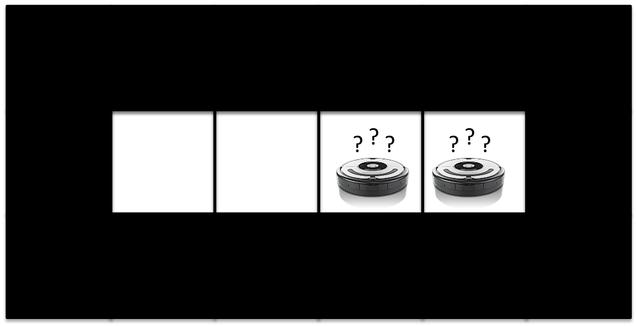


Obstacles up and down, none left and right

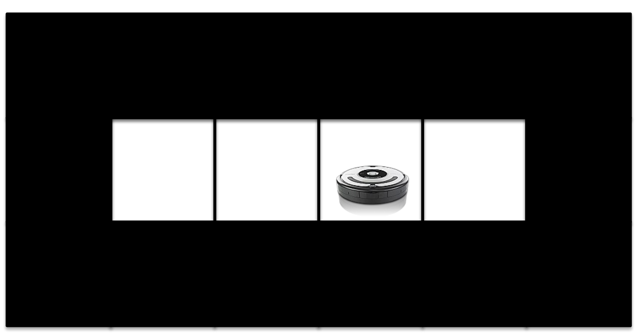
Robot Updates Distribution



Robot Moves Right, Updates



Robot Updates Probabilities



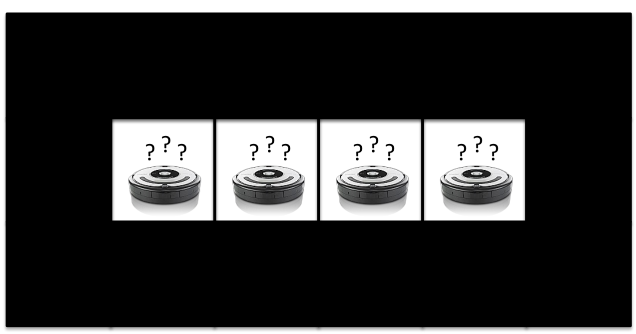
Obstacles up and down, none left and right

What Just Happened

- This was an example of robot tracking

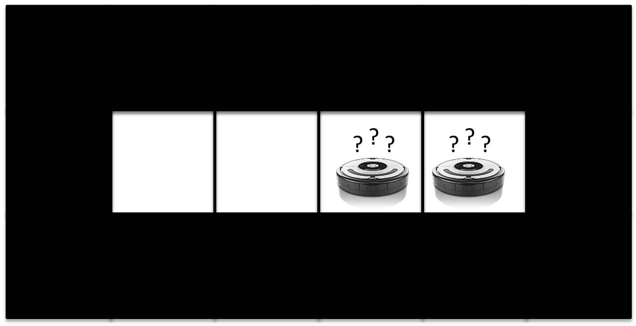
- We can also do:
  - Prediction (where would the robot be?)
  - Smoothing (where was the robot?)
  - Most likely path (what path did robot take?)

Prediction



Suppose the Robot Moves Right Twice

New Robot Position Distribution



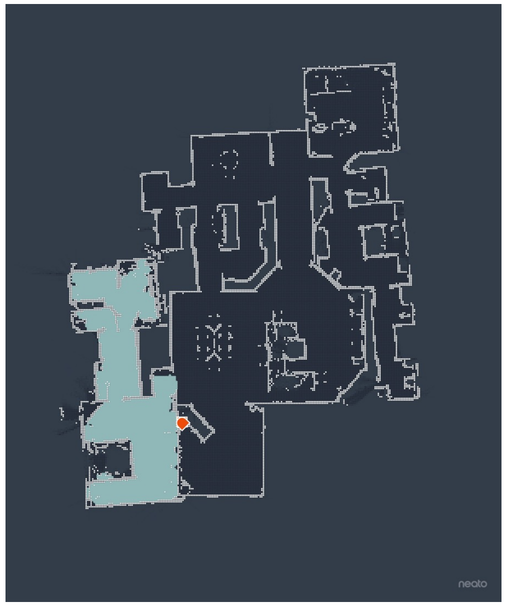
Are these probabilities uniform?

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:



HMM Basics

- What is the $P(s_0 \ldots s_t, e_0 \ldots e_t)$?
- Assume known initial distribution over s_0

$$P(s_0)P(e_0|s_0)P(s_1|s_0)P(e_1|s_1)P(s_2|s_1)P(e_2|s_2)\ldots P\left(s_t\middle|s_{\{t-1\}}\right)P(e_t|s_t)$$

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)

Bayes Rule Reminder

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

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

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

Bayes Rule With Extra Evidence

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

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

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

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

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

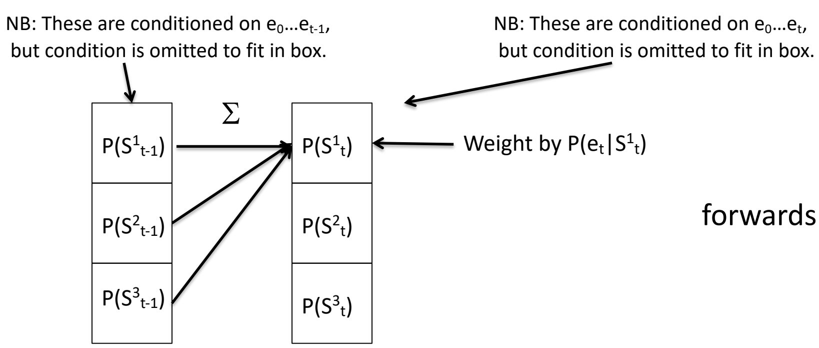
$$= \alpha P(e_t \mid S_t e_{t-1} \ldots e_0) P(S_t \mid e_{t-1} \ldots e_0)$$

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

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

Recursive

Implementation



forwards…➔

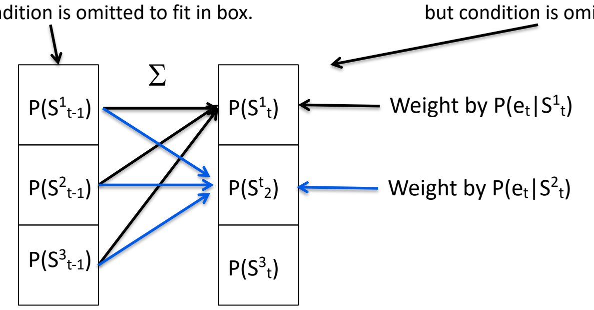
Maintain a vector of probabilities at each time step
Arcs correspond $P(s_t|s_{t-1})$ in summation of previous slide:

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

Implementation

NB: These are conditioned on $e_0 \ldots e_{t-1}$, but condition is omitted to fit in box.

NB: These are conditioned on $e_0 \ldots e_t$, but condition is omitted to fit in box.



forwards… ➔

Maintain a vector of probabilities at each time step

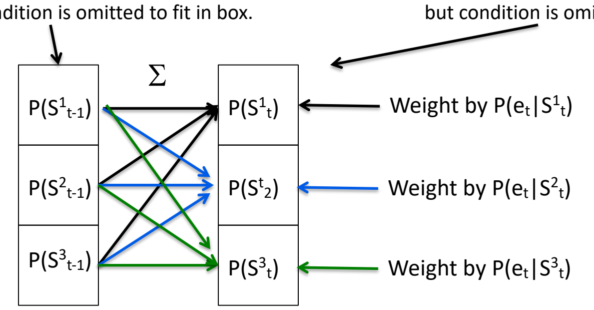
Arcs correspond $P(s_i|s_{i-1})$ in summation of previous slide:

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

Implementation

NB: These are conditioned on $e_0 \ldots e_{t-1}$, but condition is omitted to fit in box.

NB: These are conditioned on $e_0 \ldots e_t$, but condition is omitted to fit in box.



forwards… ➔

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

Example

- W = grad student is working
- R = student has produced results
- Advisor observes whether student has produced results
- Infer whether student 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 student starts semester in a productive (working) state
- Prof. has observed two consecutive meetings without results
- What is probability the student 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

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

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

Most Likely (Viterbi) Path

From definition of HMM:

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

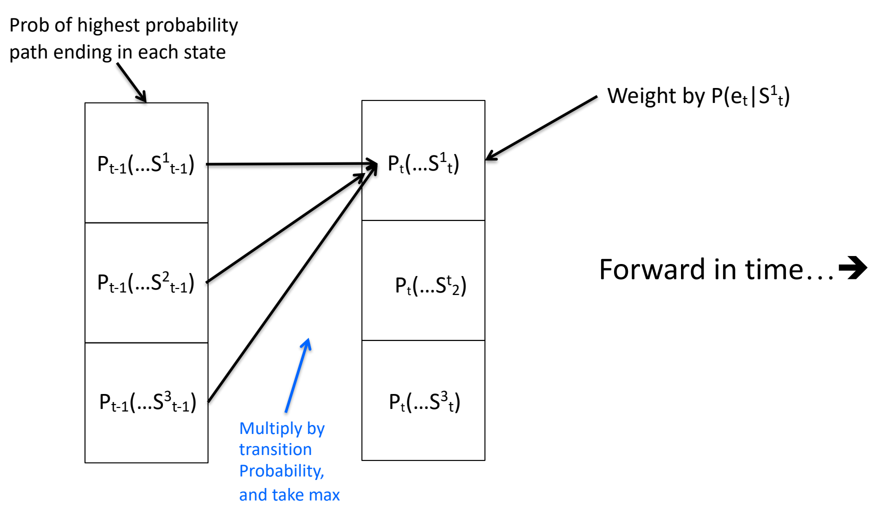
Suppose we want max probability sequence of states:

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

Keep distributing max over product!

Compare with Dijkstra's algorithm, dynamic programming.

Viterbi Visualized



Maintain a vector of probabilities at each time step
Each entry = prob of highest prob path ending in a state
Arcs correspond $P(s_i \mid s_{i-1})$ in max of previous slide

Implementing the Viterbi Algorithm
(forward part)

- P_0=initial distribution
- For t=1 to T
  - P_t = [0…0]
  - For NextS = 1 to n
    - 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)

Hindsight derivation, example included in slides but not discussed in detail

Hindsight

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

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

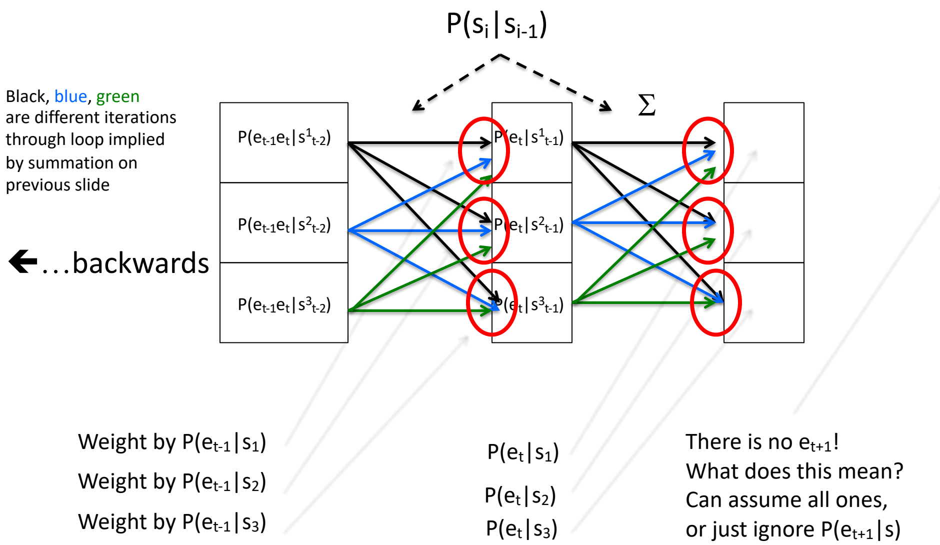
$$P(e_t..e_{k+1} \mid S_k) = \sum_{S_{k+1}} P(e_t..e_{k+1} \mid S_k S_{k+1})P(S_{k+1} \mid S_k)$$

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

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

Recursive

Implementation



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

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 student 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 he was working at time 1?

Let's Do More Math

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

$$P(W_1 | \overline{r}_2\overline{r}_1) = \alpha P(W_1 | \overline{r}_1) P(\overline{r}_2 | W_1)$$
$$P(\overline{r}_2 | w_1) = \sum_{W_2} P(\overline{r}_2 | W_2) P(W_2 | w_1)$$
$$P(\overline{r}_2 | w_1) = (0.4 * 0.8 + 0.8 * 0.2) = 0.48$$
$$P(\overline{r}_2 | \overline{w}_1) = (0.4 * 0.3 + 0.8 * 0.7) = 0.68$$
$$P(w_1 | \overline{r}_2\overline{r}_1) = \alpha 0.67 * 0.48 = \alpha 0.3216$$
$$P(\overline{w}_1 | \overline{r}_2\overline{r}_1) = \alpha 0.33 * 0.68 = \alpha 0.2244$$
$$P(w_1 | \overline{r}_2\overline{r}_1) = 0.59, P(\overline{w}_1 | \overline{r}_2\overline{r}_1) = 0.41$$

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

Understanding things in Terms of Paths

- **Viterbi algorithm:** For each s, and for each k, what is the maximum probability path that ends in s at time k?
- **Forward algorithm:** For each s, and for each k, what is the sum of the probabilities of all paths ending in s at time k?
- **Backward algorithm:** For each s, and for each k, what is the sum of the probabilities of all paths starting in s at time step k?

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.

Harsh Reality

- As with MDPs, things are great at the level of states
- Markov assumption leads to large state spaces

- Dealing with large state spaces
  - 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

(Very Brief Overview)

- 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)

- Approximations required for large/complex/continuous systems