Hidden Markov Models

CSE 415: Introduction to Artificial Intelligence
University of Washington
Winter 2019

Presented by S. Tanimoto, University of Washington, based on material by Dan Klein and Pieter Abbeel -- University of California.

You never know the true state, only a sequence of observations, and the task is to predict the true state as best as possible?

Hidden Markov Models

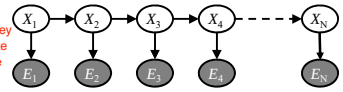
- Markov chains not so useful for most agents
  - Eventually you don't know anything anymore
  - Need observations to update your beliefs

beliefs = what states I think I am in

- Hidden Markov models (HMMs)
  - Underlying Markov chain over states S
  - You observe outputs (effects) at each time step
  - As a Bayes' net:

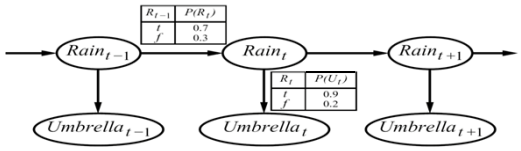
Xi describes the possible states of the system at time i and their probability distribution

Agent ONLY sees observations Ei so they never know what state they are in, only have a "best guess"



State space does not change so domain of each random variable is constant. Each random variable has an EMISSION random variable Ei, i.e. an observation about the current state.

Example

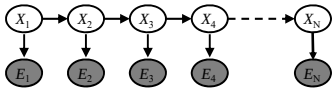


- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t|X_{t-1})$
  - Emissions: $P(E_i|X_i)$

"Observations about world/state"

state transition model

Hidden Markov Models



- Defines a joint probability distribution:

$$P(X_1, \ldots, X_n, E_1, \ldots, E_n) =$$

next state depends solely on the current state

$$P(X_{1:n}, E_{1:n}) = P(X_1)P(E_1|X_1)\prod_{t=2}^{N} P(X_t|X_{t-1})P(E_t|X_t)$$

Probability of transitioning between a specific one of the possible states at time t-1, X_t-1, to a specific one of the possible state at time t, X_t

Probability that you observe the given observation if you are actually in state Xt

HMM Computations

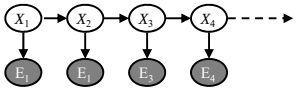
- Given
  - parameters
  - evidence $E_{1:n} = e_{1:n}$
    the values of all the observations
- Inference problems include:
  - Filtering, find $P(X_t|e_{1:t})$ for all t
  - Smoothing, find $P(X_t|e_{1:n})$ for all t
  - Most probable explanation, find $x^*_{1:n} = \text{argmax}_{x_{1:n}} P(x_{1:n}|e_{1:n})$

most likely sequence of states that occurred that generates the given observations

Real HMM Examples

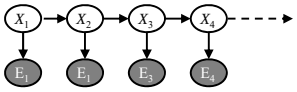
- Speech recognition HMMs:
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)

trying to predict what word was said, and your evidence is the given sound you have at a period in a signal



Real HMM Examples

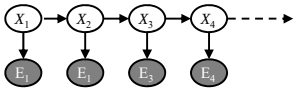
- Machine translation HMMs:
  - Observations are words (tens of thousands)
  - States are translation options



Real HMM Examples

- Robot tracking:
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)

Robot actual location is X_i while its observations from sensor readings would be E_i



Filtering / Monitoring

- Filtering, or monitoring, is the task of tracking the distribution B(X) (the belief state) over time

trying to find the most likely current state; we form an approximated probability distribution of the current state based on our evidence

- We start with B(X) in an initial setting, usually uniform
- As time passes, or we get observations, we update B(X)

- The Kalman filter (one method – Real valued values)
  - invented in the 60's as a method of trajectory estimation for the Apollo program

Inference Recap: Simple Cases

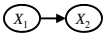
Using an observation to update the most likely state that the agent is currently in

$$P(X_1|e_1)$$

def of conditional prob

$$P(x_1|e_1) = P(x_1, e_1)/P(e_1)$$
$$\propto_{x_1} P(x_1, e_1)$$
$$= P(x_1)P(e_1|x_1)$$

using bayes rule
P(x1 I e1) is proportional to this



$$P(X_2)$$

$$P(x_2) = \sum_{x_1} P(x_1, x_2)$$
$$= \sum_{x_1} P(x_1)P(x_2|x_1)$$

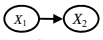
State of ending in x2 = P(being in x1) then transitioning to state x2 for any possible state x1

Convolution!

Online Belief Updates

- Every time step, we start with current P(X | evidence)
- We update for time:

Prob of transitioning to xt given the evidence



$$P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$$

sum over all possible previous states one time step ago

- We update for evidence:

$$P(x_t|e_{1:t}) \propto_{x_t} P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$

- The forward algorithm does both at once (and doesn't normalize)
- Problem: space is |X| and time is $|X|^2$ per time step

For each of the IXI possible future states, to calculate the prob we need to sum over the IXI possible states one step ago.

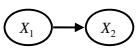
evidence is correcting possibly poor initial decisions (poor initial probability distribution)

Passage of Time

- Assume we have current belief P(X | evidence to date)

$$B(X_t) = P(X_t|e_{1:t})$$

- Then, after one time step passes:



calculating future state; same as last slide

$$P(X_{t+1}|e_{1:t}) = \sum_{x_t} P(X_{t+1}|x_t)P(x_t|e_{1:t})$$

- Or, compactly:

$$B'(X') = \sum_{x} P(X'|x)B(x)$$

- Basic idea: beliefs get "pushed" through the transitions
  - With the "B" notation, we have to be careful about what time step t the belief is about, and what evidence it includes

For initial B(S1) = B0(S1)*P(E1 I S1)

Observation

- Assume we have current belief P(X | previous evidence):
  Belief without most recent evidence.

$$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$

- Then:

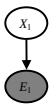
$$P(X_{t+1}|e_{1:t+1}) \propto P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t})$$

- Or:

$$B(X_{t+1}) \propto P(e|X)B'(X_{t+1})$$

belief of current state gets updated by the current evidence

- Basic idea: beliefs reweighted by likelihood of evidence
- Unlike passage of time, we have to renormalize ?



The Forward Algorithm

- We want to know: $B_t(X) = P(X_t|e_{1:t})$
- We can derive the following updates

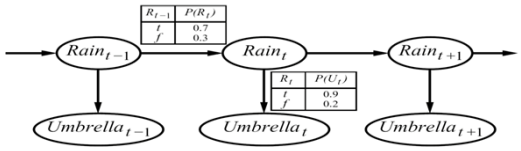
thus these values are not strictly probabilities but are still correct relative to each other

$$\begin{aligned} P(x_t|e_{1:t}) &\propto_X P(x_t, e_{1:t}) \\ &= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t}) \\ &= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) P(x_t|x_{t-1}) P(e_t|x_t) \\ &= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}, e_{1:t-1}) \end{aligned}$$

- To get $B_t(X)$ compute each entry and normalize

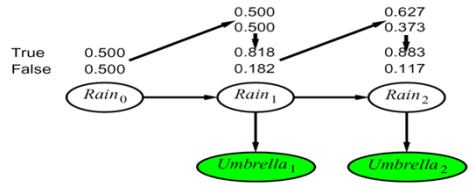
To turn into probabilities again, we have to normalize; divide by the sum of all Bt(Xt) values?

Example: Run the Filter



- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t|X_{t-1})$
  - Emissions: $P(E|X)$

Example HMM



So we never get to see the actual states of our system. All we see are the observations. However, we know the likelihood of transitioning between two given states, i.e. P(X_t I X_t-1) and the expectation model P(E I X). Our goal is to use this information to predict the current states

Summary: Filtering

- Filtering is the inference process of finding a distribution over X_T given e_1 through e_T : $P(X_T \mid e_{1:t})$
- We first compute $P(X_1 \mid e_1)$:
- For each t from 2 to T, we have $P(X_{t-1} \mid e_{1:t-1})$
- **Elapse time:** compute $P(X_t \mid e_{1:t-1})$

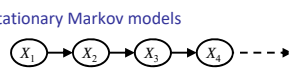
$$P(x_1|e_1) \propto P(x_1) \cdot P(e_1|x_1)$$

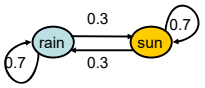
- **Ob** $P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$

$$P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$

Recap: Reasoning Over Time

- Stationary Markov models



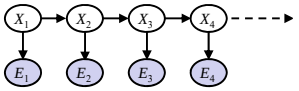


$P(X_1)$ $P(X|X_{-1})$

$P(E|X)$

X	E	P
rain	umbrella	0.9
rain	no umbrella	0.1
sun	umbrella	0.2
sun	no umbrella	0.8

- Hidden Markov models



states are hidden; only observations are seen.

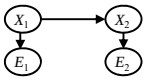
Recap: Filtering

Elapse time: compute P(X_t | $e_{1:t-1}$) try to predict a future state?

$$P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$$

Observe: compute P(X_t | $e_{1:t}$) make predictions about current state.

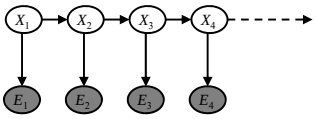
$$P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$



Belief: <P(rain), P(sun)>

$P(X_1)$	<0.5, 0.5>	*Prior on X_1*
$P(X_1 \mid E_1 = umbrella)$	<0.82, 0.18>	*Observe*
$P(X_2 \mid E_1 = umbrella)$	<0.63, 0.37>	*Elapse time*
$P(X_2 \mid E_1 = umb, E_2 = umb)$	<0.88, 0.12>	*Observe*

Best Explanation Queries



- Query: most likely seq:

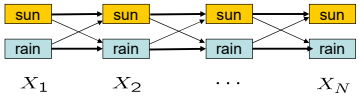
$$\arg\max_{x_{1:t}} P(x_{1:t}|e_{1:t})$$

Instead of trying to find the most likely current state, we care about the entire sequence of states that brought us to this point (and explains the evidence)

most likely sequence of states to produce the given sequence of observations.

State Path Trellis

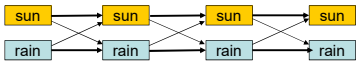
- State trellis: graph of states and transitions over time



- Each arc represents some transition $x_{t-1} \rightarrow x_t$
- Each arc has weight $P(x_t|x_{t-1})P(e_t|x_t)$
- Each path is a sequence of states
- The product of weights on a path is the seq's probability
- Can think of the Forward (and now Viterbi) algorithms as computing sums of all paths (best paths) in this graph

Find max of e1*e2*e3*…*en = max of log(e1) + log(e2) + log(e3) + .. + log(en)

Viterbi Algorithm



generate optimal sequence of states.

$$x^*_{1:T} = \arg\max_{x_{1:T}} P(x_{1:T}|e_{1:T}) = \arg\max_{x_{1:T}} P(x_{1:T}, e_{1:T})$$

$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$

$$= \max_{x_{1:t-1}} P(x_{1:t-1}, e_{1:t-1}) P(x_t|x_{t-1}) P(e_t|x_t)$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) \max_{x_{1:t-2}} P(x_{1:t-1}, e_{1:t-1})$$

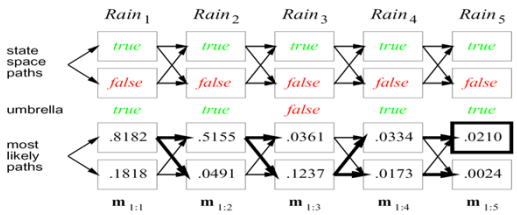
$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]$$

22

build this m array over time using dynamic programming.

Sequence of states that is most probable rather than most probable individual state (i.e. forward algorithm)..

Example



23

NOTE: these numbers are NOT probabilities.