Hidden Markov Models
Filtering

Andrea Danyluk
April 7, 2017

With thanks to CS188 slides.

Announcements

- Filtering assignment
  - Due Tuesday
- Start thinking about final projects
- Returning midterms today

Today's Lecture

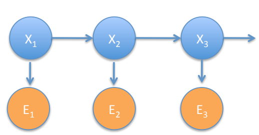
- HMMs
- Filtering

Probability Recap

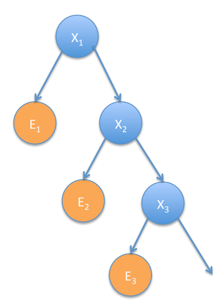
- Conditional probability $$P(x|y) = \frac{P(x,y)}{P(y)}$$
- Chain rule $$\begin{aligned} P(X_1, X_2, \ldots X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_1,X_2)\ldots \\ &= \prod_{i=1}^{n} P(X_i|X_1,\ldots,X_{i-1}) \end{aligned}$$
- X and Y are conditionally independent given Z if and only if:
  P(X|Y,Z) = P(X|Z) and *P(Y|X,Z) = P(Y|Z)*
  P(X, Y|Z) = P(X|Z)P(Y|Z)

Hidden Markov Models

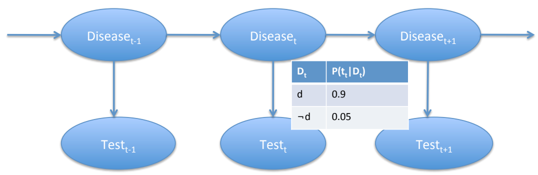
- Underlying Markov chain over states S
- You observe outputs (effects) at each time step
- A Dynamic Bayesian network



Hidden Markov Models



D_{t-1}	$P(d_t \mid D_{t-1})$
d	.99
¬d	0.2



D_t	$P(t_t \mid D_t)$
d	0.9
¬d	0.05

An HMM is defined by:

- Initial distribution: $P(X_1)$
- Transitions: $P(X_n|X_{n-1})$
- Emissions: $P(E_n|X_n)$

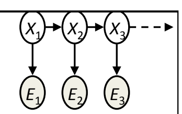
Conditional Independence

- HMMs have two important independence properties
  - *Markov* hidden process: Future depends on the past via the present
  - Current observation (emission) is independent of all else given the current state

Filtering = State Estimation

- Process of computing the belief state (posterior distribution over the most recent state), given evidence to date
- Begin with P(X) in an initial setting, usually uniform
- As time passes/get observations update belief state

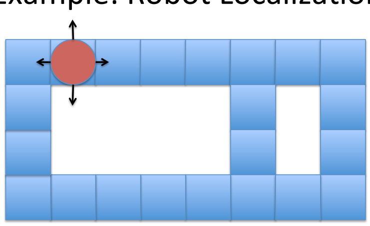
Chain Rule and HMMs



- From the chain rule, *every* joint distribution over $X_1, E_1, \ldots, X_T, E_T$ can be written as:
  $P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1)\prod_{t=2}^{T} P(X_t|X_1, E_1, \ldots, X_{t-1}, E_{t-1})P(E_t|X_1, E_1, \ldots, X_{t-1}, E_{t-1}, X_t)$
- We assume that for all *t*:
  - State independent of all past states and all past evidence given the previous state
  - Evidence is independent of all past states and all past evidence given the current state
- This gives us the following expression:

$$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})P(E_t|X_t)$$

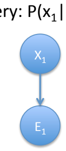
Example: Robot Localization



X_t is the location of the robot. Domain is the set of blue squares
Don't know where robot starts; assume uniform distribution over all squares
Sensor model: 4 bits (whether a wall in each direction); each sensor's error rate is ε
Neighbors(s) is a set of empty squares adjacent to s
Equally likely to move in any valid direction

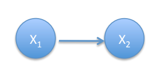
Inference: Base Cases

- Observation
  - Given: $P(X_1)$, $P(e_1|X_1)$
  - Query: $P(x_1|e_1)$ for all x_1



$P(x_1|e_1) = P(e_1,x_1)$ / $P(e_1)$
[Normalization step: do at the end.]
Focus on:
$P(e_1|x_1)\ P(x_1)$

- Passage of Time
  - Given: $P(X_1)$, $P(X_2|X_1)$
  - Query: $P(x_2)$ for all x_2



$P(x_2) = \Sigma_{\text{all } x_1} P(x_2|x_1)\ P(x_1)$

Generalizing: 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:

$X_1 \rightarrow X_2$

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

$= \Sigma_{\text{all } xt} P(X_{t+1} | x_t) P(x_t | e_{1:t})$

$= \Sigma_{\text{all } xt} P(X_{t+1} | x_t) B(x_t)$

Generalizing: Observation

- Assume we have current belief P(X | previous evidence):
- $B'(X_{t+1}) = P(X_{t+1} | e_{1:t})$
- Then, after evidence comes in:

$P(X_{t+1} | e_{1:t+1}) = P(X_{t+1}, e_{t+1} | e_{1:t}) / P(e_{1:t+1} | e_{1:t})$

$= \alpha\ P(X_{t+1}, e_{t+1} | e_{1:t})$

$= \alpha\ P(e_{t+1} | X_{t+1})\ P(X_{t+1} \mid e_{1:t})$

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

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

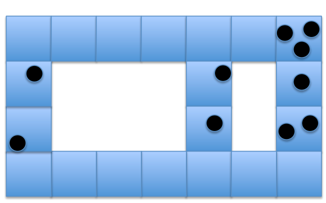
Particle Filtering

- Sometimes |X| is too big to use exact inference
- Solution: approximate inference
  - Track samples of X ; not all values
  - Aim for N << |X|
  - Samples are called particles
  - In memory, maintain a list of particles
  - Time per step is linear in the number of samples
  - Note: number of samples needed may still be large
- Robot localization
  - Remember the soccer-playing dogs?

Particle Filtering

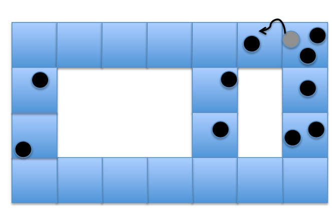
- P(x) is approximated by the number of particles with value x
- Many x will have P(x) = 0

Particle Filtering



Particles: (1,2) (1,3) (5,2) (5,3) (7,2) (7,2) (7,3) (7,4) (7,4) (7,4)

Particle Filtering: Passage of Time

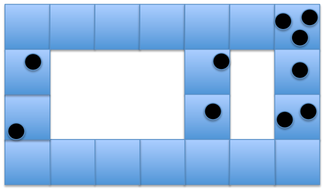


Move **each particle** by sampling its position from the transition model:

x’ = sample(P(X’|x))

This gives us a new set of N particles.

Particle Filtering: Observation



Weight **each particle** based on the evidence:

weight(x) = P(e|x)

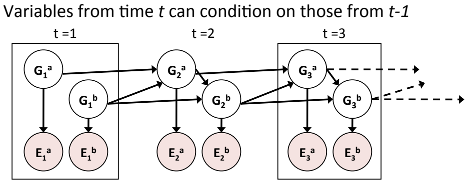
B(X) = α P(e|X)B'(X)   Recall $B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$

Particle Filtering: Resampling

- Now sample N particles from the weighted particle list
- This essentially re-normalizes the distribution

Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence
- Idea: Repeat a fixed Bayes Net structure at each time
- Variables from time *t* can condition on those from *t-1*



- Dynamic Bayes nets are a generalization of HMMs

DBN Particle Filters

- Now a single particle is a complete sample for a time step
- Initialize: Generate samples/particles for time t=1
- For example, if we're determining P(X), P(Y) and both X and Y are over domains of positions in our "map", then our particles might be
  ((1,2), (1,2)), ((1,3), (1,2)), ((5,2), (5,1)), etc.

DBN Particle Filters: Cont'd

- Passage of time: Sample a successor for each particle
  ((1,2), (1,2)) => ((1,3), (1,2))
  ((1,3), (1,2)) => ((1,3), (1,3))
  etc
- Observation: Weight each entire sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: $P(E_1^a \mid G_1^a) * P(E_1^b \mid G_1^b)$
- Resample
  - Selected samples (complete tuples) in proportion to their likelihood

Some Applications

- Robot localization
- Speech recognition
- Sequence alignment
- Computational finance
- Healthcare risk modeling