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
  \[ P(X_1, X_2, \ldots, X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots \]
  \[ = \prod_{i=1}^{n} P(X_i|X_{i-1}, \ldots, X_1) \]
• X and Y are conditionally independent given Z if and only if:
  \[ P(X|Y,Z) = P(X|Z) \text{ and } P(Y|X,Z) = P(Y|Z) \]
  \[ P(X, Y|Z) = P(X|Z)P(Y|Z) \]

Hidden Markov Models
• Underlying Markov chain over states S
• You observe outputs (effects) at each time step
• A Dynamic Bayesian network
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:
  $$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1) \prod_{t=2}^T P(X_t | X_{t-1}) P(E_t | X_t)$$
- 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:
  $$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1) \prod_{t=2}^T P(X_t | X_{t-1}) P(E_t | X_t)$$

**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 $\epsilon$
- Neighbors(s) is a set of empty squares adjacent to $s$
- Equally likely to move in any valid direction

**Inference: Base Cases**

- **Observation**
  - Given: $P(X_t), P(e_t | X_t)$
  - Query: $P(x_t | e_t)$ for all $x_t$

  $P(x_t | e_t) = P(e_t | x_t) / P(e_t)$
  [Normalization step: do at the end.]

  Focus on:
  $$P(e_t | x_t) P(x_t)$$

- **Passage of Time**
  - Given: $P(X_t), P(X_t | X_{t-1})$
  - Query: $P(x_{t+1})$ for all $x_{t+1}$

  $$P(x_{t+1}) = \sum_{x_t} P(x_{t+1} | x_t) P(x_t)$$
Generalizing: Passage of Time

- Assume we have current belief $P(X | \text{evidence to date})$
- $B(X) = P(X | e_{1:t})$
- Then, after one time step passes:

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

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

Generalizing: Observation

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

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

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

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

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: Passage of Time

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

Particle Filtering: Observation

- $B'(X_{t+1}) = P(X_{t+1} | e_{1:t})$
- Then, after evidence comes in:

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

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

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

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: Passage of Time

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

Particle Filtering: Observation

- $B'(X_{t+1}) = P(X_{t+1} | e_{1:t})$
- Then, after evidence comes in:

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

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

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

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: Passage of Time

- P(x) is approximated by the number of particles with value x
- Many x will have $P(x) = 0$
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_{t+1})

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

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

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_{t,1} | G_{t,1}) * P(E_{t,2} | G_{t,2})
- Resample
  - Selected samples (complete tuples) in proportion to their likelihood

Some Applications

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