CS 188: Artificial Intelligence

Filtering

Instructor: Anca Dragan --- University of California, Berkeley

[These slides were created by Dan Klein, Pieter Abbeel, and Anca. http://ai.berkeley.edu.]
Filtering / Monitoring

- Filtering, or monitoring, is the task of tracking the distribution $B_t(X) = P_t(X_t \mid e_1, \ldots, e_t)$ (the belief state) over time.

- We start with $B_1(X)$ in an initial setting, usually uniform.

- As time passes, or we get observations, we update $B(X)$.

- The Kalman filter was invented in the 60’s and first implemented as a method of trajectory estimation for the Apollo program.
Example: Robot Localization

Sensor model: can read in which directions there is a wall, never more than 1 mistake

Motion model: may not execute action with small prob.
Example: Robot Localization

Lighter grey: was possible to get the reading, but less likely b/c required 1 mistake
Example: Robot Localization

Prob
0 1

$t=2$
Example: Robot Localization

![Diagram of robot localization at time t=3 with probability scale from 0 to 1.]
Example: Robot Localization

t=4
Example: Robot Localization

\[ t=5 \]
Inference: Find State Given Evidence

- We are given evidence at each time and want to know

  \[ B_t(X) = P(X_t | e_{1:t}) \]

- Idea: start with \( P(X_1) \) and derive \( B_t \) in terms of \( B_{t-1} \)
  - equivalently, derive \( B_{t+1} \) in terms of \( B_t \)
Two Steps: Passage of Time + Observation

\[ B(X_t) = P(X_t|e_{1:t}) \]

\[ B'(X_{t+1}) \]

\[ B(X_{t+1}) \]
Inference: Base Cases

\[
P(X_1|e_1) = \frac{P(X_1, e_1)}{\sum_{x_1} P(x_1, e_1)}
\]

\[
P(X_1|e_1) = \frac{P(e_1|X_1)P(X_1)}{\sum_{x_1} P(e_1|x_1)P(x_1)}
\]

\[
P(X_2) = \sum_{x_1} P(x_1, X_2)
\]

\[
P(X_2) = \sum_{x_1} P(X_2|x_1)P(x_1)
\]
Passage of Time

- Assume we have current belief $P(X \mid \text{evidence to date})$
  $$B(X_t) = P(X_t \mid e_{1:t})$$

- Then, after one time step passes:
  $$P(X_{t+1} \mid e_{1:t}) = \sum_{x_t} P(X_{t+1}, x_t \mid e_{1:t})$$
  $$= \sum_{x_t} P(X_{t+1} \mid x_t, e_{1:t}) P(x_t \mid e_{1:t})$$
  $$= \sum_{x_t} P(X_{t+1} \mid x_t) P(x_t \mid e_{1:t})$$

- 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

- Or compactly:
  $$B'(X_{t+1}) = \sum_{x_t} P(X' \mid x_t) B(x_t)$$
Example: Passage of Time

- As time passes, uncertainty “accumulates”

(Transition model: ghosts usually go clockwise)
Assume we have current belief $P(X \mid \text{previous evidence})$:

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

Then, after evidence comes in:

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

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

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

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

Or, compactly:

$$B(X_{t+1}) \propto X_{t+1} \quad P(e_{t+1} | X_{t+1}) B'(X_{t+1})$$

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

- As we get observations, beliefs get reweighted, uncertainty “decreases”

Before observation

\[
\begin{array}{cccccccc}
0.05 & 0.01 & 0.05 & <0.01 & <0.01 & <0.01 \\
0.02 & 0.14 & 0.11 & 0.35 & <0.01 & <0.01 \\
0.07 & 0.03 & 0.05 & <0.01 & 0.03 & <0.01 \\
0.03 & 0.03 & <0.01 & <0.01 & <0.01 & <0.01 \\
\end{array}
\]

After observation

\[
\begin{array}{cccccccc}
<0.01 & <0.01 & <0.01 & <0.01 & 0.02 & <0.01 \\
<0.01 & <0.01 & <0.01 & 0.83 & 0.02 & <0.01 \\
<0.01 & <0.01 & 0.11 & <0.01 & <0.01 & <0.01 \\
<0.01 & <0.01 & <0.01 & <0.01 & <0.01 & <0.01 \\
\end{array}
\]

\[B(X) \propto P(e|X)B'(X)\]
Example: Weather HMM

\[ P(R_{t+1} | R_t) = 0.7 \]
\[ P(U_t | R_t) = 0.9 \]

| \( R_t \) | \( R_{t+1} \) | \( P(R_{t+1} | R_t) \) |
|---------|----------|------------------|
| +r     | +r       | 0.7              |
| +r     | -r       | 0.3              |
| -r     | +r       | 0.3              |
| -r     | -r       | 0.7              |

| \( R_t \) | \( U_t \) | \( P(U_t | R_t) \) |
|---------|----------|------------------|
| +r     | +u       | 0.9              |
| +r     | -u       | 0.1              |
| -r     | +u       | 0.2              |
| -r     | -u       | 0.8              |
Every time step, we start with current $P(X \mid \text{evidence})$

- We update for time:

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

- We update for evidence:

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

- The forward algorithm does both at once (and doesn’t normalize)
The Forward Algorithm

- We are given evidence at each time and want to know

\[ B_t(X) = P(X_t|e_{1:t}) \]

- We can derive the following updates

\[
P(x_t|e_{1:t}) \propto x_t P(x_t, e_{1:t})
\]

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

\[
= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) P(x_t|x_{t-1}) P(e_t|x_t)
\]

\[
= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}, e_{1:t-1})
\]

We can normalize as we go if we want to have \( P(x|e) \) at each time step, or just once at the end…

[Demo: Ghostbusters Exact Filtering (L15D2)]
Pacman – Sonar
Video of Demo Pacman – Sonar (with beliefs)
Particle Filtering
Particle Filtering

- Filtering: approximate solution
- Sometimes $|X|$ is too big to use exact inference
  - $|X|$ may be too big to even store $B(X)$
  - E.g. $X$ is continuous
- Solution: approximate inference
  - Track samples of $X$, not all values
  - Samples are called particles
  - Time per step is linear in the number of samples
  - But: number needed may be large
  - In memory: list of particles, not states
- This is how robot localization works in practice
- Particle is just new name for sample
Our representation of $P(X)$ is now a list of $N$ particles (samples)
- Generally, $N << |X|$
- Storing map from $X$ to counts would defeat the point

$P(x)$ approximated by number of particles with value $x$
- So, many $x$ may have $P(x) = 0!$
- More particles, more accuracy

For now, all particles have a weight of 1
Each particle is moved by sampling its next position from the transition model

$$x' = \text{sample}(P(X'|x))$$

- This is like prior sampling – samples’ frequencies reflect the transition probabilities
- Here, most samples move clockwise, but some move in another direction or stay in place

This captures the passage of time
- If enough samples, close to exact values before and after (consistent)
**Slightly trickier:**

- Don’t sample observation, fix it
- Similar to likelihood weighting, downweight samples based on the evidence

\[ w(x) = P(e|x) \]

\[ B(X) \propto P(e|X)B'(X) \]

- As before, the probabilities don’t sum to one, since all have been downweighted (in fact they now sum to (N times) an approximation of \( P(e) \))
Particle Filtering: Resample

- Rather than tracking weighted samples, we resample.

- $N$ times, we choose from our weighted sample distribution (i.e. draw with replacement).

- This is equivalent to renormalizing the distribution.

- Now the update is complete for this time step, continue with the next one.
Recap: Particle Filtering

- Particles: track samples of states rather than an explicit distribution

```
Demos: ghostbusters particle filtering (L15D3,4,5)
```
Video of Demo – Moderate Number of Particles
Video of Demo – One Particle
Video of Demo – Huge Number of Particles
In robot localization:

- We know the map, but not the robot’s position.
- Observations may be vectors of range finder readings.
- State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store $B(X)$.
- Particle filters are a main technique.
Particle Filter Localization (Sonar)

Global localization with sonar sensors

[Dieter Fox, et al.]
Particle Filter Localization (Laser)

[Dieter Fox, et al.]
Robot Mapping

- SLAM: Simultaneous Localization And Mapping
  - We do not know the map or our location
  - State consists of position AND map!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods

DP-SLAM, Ron Parr

[Demo: PARTICLES-SLAM-mapping1-new.avi]
Particle Filter SLAM – Video 1

[Sebastian Thrun, et al.]
Particle Filter SLAM – Video 2

[Dirk Haehnel, et al.]
Dynamic Bayes Nets
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.

[Demo: pacman sonar ghost DBN model (L15D6)]
Exact Inference in DBNs

- Variable elimination applies to dynamic Bayes nets
- Procedure: “unroll” the network for T time steps, then eliminate variables until $P(X_T|e_{1:T})$ is computed
- Online belief updates: Eliminate all variables from the previous time step; store factors for current time only
DBN Particle Filters

- A particle is a complete sample for a time step
- **Initialize**: Generate prior samples for the \( t=1 \) Bayes net
  - Example particle: \( \mathbf{G}_1^a = (3,3) \; \mathbf{G}_1^b = (5,3) \)
- **Elapse time**: Sample a successor for each particle
  - Example successor: \( \mathbf{G}_2^a = (2,3) \; \mathbf{G}_2^b = (6,3) \)
- **Observe**: Weight each *entire* sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: \( P(\mathbf{E}_1^a | \mathbf{G}_1^a) \times P(\mathbf{E}_1^b | \mathbf{G}_1^b) \)
- **Resample**: Select prior samples (tuples of values) in proportion to their likelihood