Probability Recap

- Conditional probability
  \[ P(x|y) = \frac{P(x, y)}{P(y)} \]

- Product rule
  \[ P(x, y) = P(x|y)P(y) \]

- Chain rule
  \[ P(X_1, X_2, \ldots, X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots = \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1}) \]

- Bayes rule
  \[ P(x|y) = \frac{P(y|x)}{P(y)}P(x) \]

- X, Y independent if and only if: \( \forall x, y : P(x, y) = P(x)P(y) \)

- X and Y are conditionally independent given Z: \( X \perp Y | Z \)

  if and only if:

  \( \forall x, y, z : P(x, y|z) = P(x|z)P(y|z) \)
Reasoning over Time or Space

- Often, we want to reason about a sequence of observations
  - Speech recognition
  - Robot localization
  - User attention
  - Medical monitoring

- Need to introduce time (or space) into our models

Markov Models Recap

- Explicit assumption for all \( t \): \( X_t \perp X_1, \ldots, X_{t-2} \mid X_{t-1} \)

- Consequence, joint distribution can be written as:

\[
P(X_1, X_2, \ldots, X_T) = P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_2) \ldots P(X_T \mid X_{T-1})
\]

\[
= P(X_1) \prod_{t=2}^{T} P(X_t \mid X_{t-1})
\]

- Additional explicit assumption:

\[
P(X_t \mid X_{t-1}) \text{ is the same for all } t
\]
Example Markov Chain: Weather

- States: $X = \{\text{rain, sun}\}$
- Initial distribution: 1.0 sun
- CPT $P(X_t \mid X_{t-1})$:

<table>
<thead>
<tr>
<th>$X_{t-1}$</th>
<th>$X_t$</th>
<th>$P(X_t \mid X_{t-1})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>sun</td>
<td>0.9</td>
</tr>
<tr>
<td>sun</td>
<td>rain</td>
<td>0.1</td>
</tr>
<tr>
<td>rain</td>
<td>sun</td>
<td>0.3</td>
</tr>
<tr>
<td>rain</td>
<td>rain</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Mini-Forward Algorithm

- Question: What’s $P(X)$ on some day $t$?

$$P(x_1) = \text{known}$$

$$P(x_t) = \sum_{x_{t-1}} P(x_{t-1}, x_t)$$

$$= \sum_{x_{t-1}} P(x_t \mid x_{t-1})P(x_{t-1})$$

Forward simulation
Example Run of Mini-Forward Algorithm

- From initial observation of sun

\[
\begin{bmatrix}
1.0 & 0.9 & 0.84 & 0.804 \\
0.0 & 0.1 & 0.16 & 0.196 \\
\end{bmatrix} \rightarrow \begin{bmatrix}
0.75 \\
0.25 \\
\end{bmatrix}
\]

\[P(X_1) \quad P(X_2) \quad P(X_3) \quad P(X_4) \quad P(X_\infty)\]

- From initial observation of rain

\[
\begin{bmatrix}
0.0 & 0.3 & 0.48 & 0.588 \\
1.0 & 0.7 & 0.52 & 0.412 \\
\end{bmatrix} \rightarrow \begin{bmatrix}
0.75 \\
0.25 \\
\end{bmatrix} \quad “?”
\]

\[P(X_1) \quad P(X_2) \quad P(X_3) \quad P(X_4) \quad P(X_\infty)\]

- From yet another initial distribution \(P(X_1)\):

\[
\begin{bmatrix}
p \\
1 - p \\
\end{bmatrix} \rightarrow \begin{bmatrix}
0.75 \\
0.25 \\
\end{bmatrix}
\]

[Demo: L13D1,2,3]

Hidden Markov Models
Hidden Markov Models

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

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

![Diagram of a Bayes’ net with hidden Markov models]

Example

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

- Defines a joint probability distribution:

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

Conditional Independence

- HMMs have two important independence properties:
  - Markov hidden process, future depends on past via the present
Conditional Independence

- HMMs have two important independence properties:
  - Markov hidden process, future depends on past via the present
  - Current observation independent of all else given current state

Quiz: does this mean that observations are independent given no evidence?
- [No, correlated by the hidden state]
HMM Computations

- Given
  - parameters
  - evidence $E_{1:n} = e_{1:n}$

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

Filtering (aka Monitoring)

- **The task of tracking the agent’s belief state, $B(x)$, over time**
  - $B(x)$ is a distribution over world states – repr agent knowledge
  - We start with $B(X)$ in an initial setting, usually uniform
  - As time passes, or we get observations, we update $B(X)$

- **Many algorithms for this:**
  - Exact probabilistic inference
  - Particle filter approximation
  - Kalman filter (one method – Real valued values)
    - invented in the 60’s for Apollo Program – real-valued state, Gaussian noise
HMM Examples

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

Example:
- Robot Localization

Example from Michael Pfeiffer

Sensor model: never more than 1 mistake
Motion model: may not execute action with small prob.
Example: Robot Localization

Example: Robot Localization

Prob

0 1

t=1

t=2
Example: Robot Localization

$t=3$

Example: Robot Localization

$t=4$
Example: Robot Localization

Other Real HMM Examples

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

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

\[ \begin{align*}
X_1 & \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \\
E_1 & \rightarrow E_2 \rightarrow E_3 \rightarrow E_4
\end{align*} \]

Inference: Base Cases

- **“Observation”**
  \[ P(X_1|e_1) \]
  \[ P(x_1|e_1) = P(x_1, e_1)/P(e_1) \]
  \[ \propto_{X_1} P(x_1, e_1) \]
  \[ = P(x_1)P(e_1|x_1) \]

- **“Passage of Time”**
  \[ P(X_2) \]
  \[ P(x_2) = \sum_{x_1} P(x_1, x_2) \]
  \[ = \sum_{x_1} P(x_1)P(x_2|x_1) \]
Passage of Time

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

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

  - Or compactly:
    
    $B'(X_{t+1}) = \sum_{x_t} P(X_{t+1} | x_t) B(x_t)$

- 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

Example: Passage of Time

- As time passes, uncertainty “accumulates”

(Transition model: ghosts usually go clockwise)

(T = 1)          (T = 2)          (T = 5)
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:
  \[
  P(X_{t+1}|e_{1:t+1}) = \frac{P(X_{t+1}, e_{t+1}|e_{1:t})}{P(e_{t+1}|e_{1:t})} / \frac{P(e_{t+1}|e_{1:t})}{P(X_{t+1}|e_{1:t})} \\
  \propto P(X_{t+1}, e_{t+1}|e_{1:t}) \\
  = P(e_{t+1}|X_{t+1}) P(X_{t+1}|e_{1:t}) \\
  = P(e_{t+1}|X_{t+1}) P(X_{t+1}|e_{1:t}) \\
  \]

- Or, compactly:
  \[ B(X_{t+1}) \propto P(e_{t+1}|X_{t+1}) B'(X_{t+1}) \]

- 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

\[
\begin{array}{cccccccccc}
0.00 & 0.10 & 0.20 & 0.30 & 0.40 & 0.50 & 0.60 & 0.70 & 0.80 & 0.90 & 1.00 \\
0.02 & 0.34 & 0.56 & 0.78 & 0.00 & 0.10 & 0.20 & 0.30 & 0.40 & 0.50 & 0.60 \\
0.07 & 0.63 & 0.84 & 0.00 & 0.10 & 0.20 & 0.30 & 0.40 & 0.50 & 0.60 & 0.70 \\
0.03 & 0.19 & 0.35 & 0.51 & 0.67 & 0.83 & 0.00 & 0.10 & 0.20 & 0.30 & 0.40 \\
\end{array}
\]

After observation

\[
\begin{array}{cccccccccc}
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
\end{array}
\]

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

B(+r) = 0.5
B(-r) = 0.5

B'(+r) = 0.5
B'(-r) = 0.5

B(+r) = 0.818
B(-r) = 0.182

B'(+r) = 0.627
B'(-r) = 0.373

B(+r) = 0.883
B(-r) = 0.117

Pacman – Sonar (P4)

[Demo: Pacman – Sonar – No Beliefs(L14D1)]
Video of Demo Pacman – Sonar (with beliefs)

Summary: Online Belief Updates

- Every time step, we start with current $P(X | \text{evidence})$
- 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:

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

- 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 use the single (time-passage+observation) updates:
  \[
P(x_t | e_{1:t}) \propto \prod_{t=1}^{T} P(x_t, e_{1:t})
  = \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t})
  = \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1})P(x_t | x_{t-1})P(e_t | x_t)
  = P(e_t | x_t) \sum_{x_{t-1}} P(x_t | x_{t-1})P(x_{t-1}, e_{1:t-1})
\]

- Complexity? \( O(|X|^2) \) time & \( O(X) \) space

---

Particle Filtering

[Diagram of 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

Representation: Particles

- Our representation of \( P(X) \) is now a list of \( N \) particles (samples)
  - Generally, \( N \ll |X| \)
  - 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
**Particle Filtering: Elapse Time**

- 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)

**Particle Filtering: Observe**

- 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
Video of Demo – Moderate Number of Particles

Video of Demo – One Particle
Video of Demo – Huge Number of Particles

Robot Localization

- 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 filtering is a main technique
Particle Filter Localization (Sonar)

![Global localization with sonar sensors](global-sonar-uw-annotated.avi)

Particle Filter Localization (Laser)

![Global localization with laser sensors](global-floor.gif)
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

[Demo: PARTICLES-SLAM-mapping1-new.avi]
Particle Filter SLAM – Video 2

[Demo: PARTICLES-SLAM-fastslam.avi]