Bayesian Methods
Recall Robot Localization

• Given
  – Sensor readings $z_1, z_2, \ldots, z_t = z_{1:t}$
  – Known control inputs $u_0, u_1, \ldots u_t = u_{0:t}$
  – Known model $P(x_{t+1} | x_t, u_t)$ with initial $P(x_1 | u_0)$
  – Known map $P(z_t | x_t)$

• Compute
  – $P(x_t | z_{1:t-1}, u_{0:t-1})$ Most likely state $x$ at time $t$ given a sequence of commands $u_{0:t-1}$ and measurements $z_{1:t-1}$

This is just a probabilistic representation of what you’ve already learned!

Let’s try to connect the dots and do a couple of examples
A Simple Example
Why Not Just Use a Kalman Filter?
Bayes Filter

- Given a sequence of measurements $z_1, \ldots, z_k$
- Given a sequence of commands $u_0, \ldots, u_{k-1}$
- Given a sensor model $P(z_k \mid x_k)$
- Given a dynamic model $P(x_k \mid x_{k-1})$
- Given a prior probability $P(x_0)$
- Find $P(x_k \mid z_{1:k}, u_{0:k-1})$
Bayesian Filtering

• Recall Bayes Theorem:

\[ P(x | z) = \frac{P(z|x)P(x)}{P(z)} \]

• Also remember conditional independence

• Think of \( x \) as the state of the robot and \( z \) as the data we know

\[
P(x_k | u_{0:k-1}, z_{1:k}) \quad \rightarrow \quad \text{Posterior Probability Distribution}
\]

\[
P(x_k | u_{0:k-1}, z_{1:k}) = \frac{P(z_k|x_k, u_{0:k-1}, z_{1:k-1})P(x_k|u_{0:k-1}, z_{1:k-1})}{P(z_k|u_{0:k-1}, z_{1:k-1})}
\]

\[= \eta_k P(z_k|x_k) \int_{x_{k-1}} P(x_k|u_{k-1}, x_{k-1})P(x_{k-1}|u_{0:k-2}, z_{1:k-1}) \]

\[\text{observation} \quad \text{state prediction} \quad \text{recursive instance}\]
Observation Model

- What is the probability that a given state $x_k$ generates the measurement $z_k$?

$$P(z_k | x_k)$$

- Same $z_k$
- Different $x_k$

This is more likely than that but how do we measure?
Observation Model

• Compute the distance of end-point of each beam to nearest obstacle
Observation Model

- How likely is the measurement $z_k$ given a state $x_k$?
- The measurement $z_k$ is what we get. So we have to stick by it.
- However, not all states will likely produce the measurement.
- We’re not interested in finding the most likely state. *We want the whole distribution*.

$$P(z_k | x_k)$$
Observation Model

- How likely is the measurement $z_k$ given a state $x_k$?
- The measurement $z_k$ is what we get. So we have to stick by it.
- However, not all states will likely produce the measurement.
- We’re not interested in finding the most likely state. *We want the whole distribution*.

$$P(z_k | x_k)$$
Observation Model

• How likely is the measurement $z_k$ given a state $x_k$?
• The measurement $z_k$ is what we get. So we have to stick by it.
• However, not all states will likely produce the measurement.
• We’re not interested in finding the most likely state. We want the whole distribution.

$$P(z_k|x_k)$$
Observation Model

• How likely is the measurement $z_k$ given a state $x_k$?
• The measurement $z_k$ is what we get. So we have to stick by it
• However, not all states will likely produce the measurement
• We’re not interested in finding the most likely state. *We want the whole distribution*
Observation Model

• How likely is the measurement $z_k$ given a state $x_k$?
• The measurement $z_k$ is what we get. So we have to stick by it
• However, not all states will likely produce the measurement
• We’re not interested in finding the most likely state. We want the whole distribution $P(z_k|x_k)$
Observation Model

• How likely is the measurement $z_k$ given a state $x_k$?
• The measurement $z_k$ is what we get. So we have to stick by it.
• However, not all states will likely produce the measurement.
• We’re not interested in finding the most likely state. We want the whole distribution.

$P(z_k|x_k)$

Doesn’t have to be Gaussian.
Observation Model
Beam Model
Observation Model

Beam Model

• Combine all the distances $d_i$ between the measured obstacle and the real obstacle (we do have the map, or part thereof)

• But other things must be considered
Range Finder
Observation Model Beam Model

• For each laser beam we can have (given a map and a state)
  – Correct range with noise
    \[
    P_{\text{hit}}(z_k | x_k, m) = \begin{cases} 
    \eta_{\text{hit}}N(z_k^*, \sigma_{\text{hit}}) & \text{if } 0 \leq z_k \leq z_{\text{max}} \\
    0 & \text{otherwise} 
    \end{cases}
    \]
  – Unexpected object
    \[
    P_{\text{short}}(z_k | x_k, m) = \begin{cases} 
    \eta_{\text{short}}\lambda e^{-\lambda z_k} & \text{if } 0 \leq z_k \leq z_k^* \\
    0 & \text{otherwise} 
    \end{cases}
    \]
  – Failures
    \[
    P_{\text{max}}(z_k | x_k, m) = I(z_k = z_{\text{max}}) \begin{cases} 
    1 & \text{if } z_k = z_{\text{max}} \\
    0 & \text{otherwise} 
    \end{cases}
    \]
  – Random measurement
    \[
    P_{\text{rand}}(z_k | x_k, m) = \begin{cases} 
    \frac{1}{z_{\text{max}}} & \text{if } 0 \leq z_k \leq z_{\text{max}} \\
    0 & \text{otherwise} 
    \end{cases}
    \]
Range Finder
Observation Model Beam Model

(a) Gaussian distribution $p_{\text{hit}}$

(b) Exponential distribution $p_{\text{short}}$

(c) Uniform distribution $p_{\text{max}}$

(d) Uniform distribution $p_{\text{rand}}$

Probabilistic Robotic
Mixing all these cases together we get

\[ P(z_k | x_k, m) = \begin{bmatrix} w_{\text{hit}} \\ w_{\text{short}} \\ w_{\text{max}} \\ w_{\text{rand}} \end{bmatrix}^T \begin{bmatrix} P_{\text{hit}}(z_k | x_k, m) \\ P_{\text{short}}(z_k | x_k, m) \\ P_{\text{max}}(z_k | x_k, m) \\ P_{\text{rand}}(z_k | x_k, m) \end{bmatrix} \]

where the weights are parameters

\[ w_{\text{hit}} + w_{\text{short}} + w_{\text{max}} + w_{\text{rand}} = 1 \]
Constants

\[ \eta_{\text{hit}} = \left( \int_{0}^{z_{\text{max}}} N(z_k^*, \sigma^2) \, dz_k \right)^{-1} \]

\[ \eta_{\text{short}} = \frac{1}{1 - e^{-\lambda_{\text{short}} z_k^*}} \]
Range Finder
Likelihood Field Model

\[ P_{\text{hit}}(z_k | x_k, m) = \varepsilon \sigma_{\text{hit}}^2 (d^2) \]

(a) Gaussian distribution \( p_{hit} \)

Probabilistic Robotic
Range Finder
Likelihood Field Model

1: Algorithm likelihood_field_range_finder_model(\(z_t, x_t, m\)):

2: \(q = 1\)

3: for all \(k\) do

4: if \(z_t^k \neq z_{\text{max}}\)

5: \(x_{z_t^k} = x + x_{k,\text{sens}} \cos \theta - y_{k,\text{sens}} \sin \theta + z_t^k \cos(\theta + \theta_{k,\text{sens}})\)

6: \(y_{z_t^k} = y + y_{k,\text{sens}} \cos \theta + x_{k,\text{sens}} \sin \theta + z_t^k \sin(\theta + \theta_{k,\text{sens}})\)

7: \(dist = \min_{x',y'} \left\{ \sqrt{(x_{z_t^k} - x')^2 + (y_{z_t^k} - y')^2} \mid (x', y') \text{ occupied in } m \right\}\)

8: \(q = q \cdot \left( z_{\text{hit}} \cdot \text{prob}(dist, \sigma_{\text{hit}}) + \frac{z_{\text{random}}}{z_{\text{max}}} \right)\)

9: return \(q\)
Range Finder
Likelihood Field Model

• Precompute the shortest distance to obstacle
  – Distance transformation
  – Brushfire algorithm
Bayes Filtering

\[ P(x_k | u_{0:k-1}, z_{1:k}) = \eta_k P(z_k | x_k) \int P(x_k | u_{k-1}, x_{k-1}) P(x_{k-1} | u_{0:k-2}, z_{1:k-1}) \]
Predict Motion (Prior Distribution)

- Suppose we have $P(x_k)$
- We have $P(x_{k+1}|x_k, u_k)$
- Put together

$$P(x_{k+1}) = \int_{x_k} P(x_{k+1}|x_k, u_k)P(x_k)dx_k$$

What is the probability distribution for $x_{k+1}$ given the command $u_k$ and all the previous states $x_k$?
System Model

Prior probability distribution $P(x_k)$

State space $X = \{1, 2, 3, 4\}$

Transition matrix: The probability $P(j|i)$ of moving from $i$ to $j$ is given by $P_{i,j}$. Each row must sum to 1.

Compute $P(x_{k+1}) = \int_{x_k} P(x_{k+1}|x_k, u_k)P(x_k)dx_k$
System Model

State space $X = \{1, 2, 3, 4\}$

<table>
<thead>
<tr>
<th>$P(x_{k+1} \mid x_k, u_k)$</th>
<th>$x_{k+1}=1$</th>
<th>$x_{k+1}=2$</th>
<th>$x_{k+1}=3$</th>
<th>$x_{k+1}=4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_k=1$</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0</td>
</tr>
<tr>
<td>$x_k=2$</td>
<td>0</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
</tr>
<tr>
<td>$x_k=3$</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0.75</td>
</tr>
<tr>
<td>$x_k=4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$$P(x_{k+1} = 1) = \sum_{x_k \in X} P(x_{k+1} = 1 \mid x_k, u_k) P(x_k)$$
System Model

State space $X = \{1, 2, 3, 4\}$

$$P(x_{k+1} = 2) = \sum_{x_k \in X} P(x_{k+1} = 2 | x_k, u_k) P(x_k)$$

| $P(x_{k+1} | x_k, u_k)$ | $X_{k+1}=1$ | $X_{k+1}=2$ | $X_{k+1}=3$ | $X_{k+1}=4$ |
|-------------------------|-------------|-------------|-------------|-------------|
| $X_k=1$                 | 0.25        | 0.5         | 0.25        | 0           |
| $X_k=2$                 | 0           | 0.25        | 0.5         | 0.25        |
| $X_k=3$                 | 0           | 0           | 0.25        | 0.75        |
| $X_k=4$                 | 0           | 0           | 0           | 1           |

Prior $P(x_k)$
System Model

State space $X = \{1, 2, 3, 4\}$

\[
P(x_{k+1} = 3) = \sum_{x_k \in X} P(x_{k+1} = 3 \mid x_k, u_k) P(x_k)
\]
System Model

State space $X = \{1, 2, 3, 4\}$

<table>
<thead>
<tr>
<th>$P(x_{k+1} \mid x_k, u_k)$</th>
<th>$x_{k+1}=1$</th>
<th>$x_{k+1}=2$</th>
<th>$x_{k+1}=3$</th>
<th>$x_{k+1}=4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_k=1$</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
<td>0</td>
</tr>
<tr>
<td>$x_k=2$</td>
<td>0</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
</tr>
<tr>
<td>$x_k=3$</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0.75</td>
</tr>
<tr>
<td>$x_k=4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$P(x_{k+1} = 4) = \sum P(x_{k+1} = 4 \mid x_k, u_k)P(x_k)$

Prior $P(x_k)$
System Model

\[
P(x_{k+1}) = \int P(x_{k+1} | x_k, u_k) P(x_k) dx_k
\]
Bayes Filter Recap

\[ P(x_k|u_{0:k-1}, z_{1:k}) \]

\[ P(x_k|u_{k-1}, x_{k-1}) \]

\[ P(x_{k-1}|u_{0:k-2}, z_{1:k-1}) \]

- Obtain \( z_k \) and apply \( P(z_k|x_k) \)
- Apply command \( u_{k-1} \)
- \( P(x_{k-1} = A) \)
- \( P(x_{k-1} = B) \)
- \( P(x_{k-1} = Z) \)
Bayes Filter Recap

\[ P(x_k | u_{0:k-1}, z_{1:k}) = \eta_k P(z_k | x_k) \int_{x_{k-1}} P(x_k | u_{k-1}, x_{k-1})P(x_{k-1} | u_{0:k-2}, z_{1:k-1}) \, dx_{k-1} \]

- Given a measurement \( z_k \), compute the probability that \( z_k \) was generated from the robot in state \( x_k \)
- Prior distribution (recursive)
  - The probability of all previous states
- State transition:
  - Given a prior distribution over all states \( x_{k-1} \) and an action \( u_{k-1} \), compute the probability of the robot transitioning to state \( x_k \)
Discrete Bayes Filter Algorithm

Algorithm Discrete_Bayes_filter( u_{0:k-1}, y_{1:k}, P(x_0) )

1. \( P(x) = P(x_0) \)
   (if you don’t know: uniform distribution)
2. \( \text{for } i=1:k \)
3. \( \text{for all states } x \in X \)
4. \[
P'(x) = \sum_{x' \in X} P(x \mid u, x') P(x')
\]
   Prediction given prior dist. and command
5. \( \text{end for} \)
6. \( \eta = 0 \)
7. \( \text{for all states } x \in X \)
8. \[
P(x) = P(z \mid x) P'(x)
\]
   Update using measurement
9. \( \eta = \eta + P(x) \)
10. \( \text{end for} \)
11. \( \text{for all states } x \in X \)
12. \[
P(x) = P(x) / \eta
\]
   Normalize to 1
13. \( \text{end for} \)
14. \( \text{end for} \)
Note About the Posterior Distribution

\[ P(x_k | u_{0:k-1}, z_{1:k}) = \eta_k P(z_k | x_k) \int_{x_{k-1}} P(x_k | u_{k-1}, x_{k-1}) P(x_{k-1} | u_{0:k-2}, z_{1:k-1}) dx_{k-1} \]

• It is a probability distribution

• What do we do with it?
  – Maximum likelihood: \( \arg \max_x P(x_k | z_k, m) \)
  – Mean Squared Error: \( E[(P(x) - P(\hat{x}))^2] \)
Kalman vs Bayes

Kalman Filter

Bayes Filter

Isard 1998
Particle Filter

• Computing $P(z_k|x_k, m)$ and $P(x_{k+1}|x_k, u_k)$ is not easy
  • In practice, it is never directly computable
  • Need to propagate an entire conditional distribution, not just one state like we did with Kalman,

• Represent probability distribution by random samples
• Estimation of non-Gaussian, nonlinear processes
• Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter, Particle filter
  • Filtering: [Rubin, 88], [Gordon et al., 93], [Kitagawa 96]
  • **Computer vision**: [Isard and Blake 96, 98]
  • Dynamic Bayesian Networks: [Kanazawa et al., 95]
Particles/Samples

• Given a distribution $P(x)$, randomly sample the distribution $N$ times
• Assign a weight $w$ to each sample according to the probability
• Ensure that $\sum_{i=1}^{N} w_i = 1$
Particle Filter
Particle Filter Algorithm

Algorithm \texttt{Particle\_filter}( u_{0:k-1}, y_{1:k}, P(x_0), \text{set of } N \text{ samples } \mathcal{M} = \{x_j, w_j\} )

1. \textbf{for } i=1:k
2. \hspace{1em} \textbf{for } j=1:N
3. \hspace{2em} compute a new state x by sampling according to \( P( x \mid u_{i-1}, x_j ) \)
4. \hspace{1em} \( x_j = x \)
5. \hspace{1em} \textbf{end}
6. \hspace{1em} \( \eta = 0 \)
7. \hspace{1em} \textbf{for } j=1:N
8. \hspace{2em} \( w_j = P( z_i \mid x_j ) \)
9. \hspace{2em} \( \eta += w_j \)
10. \hspace{1em} \textbf{end}
11. \hspace{1em} \textbf{for } j=1:N
12. \hspace{2em} \( w_j = w_j / \eta \)
13. \hspace{1em} \textbf{end}
14. \hspace{1em} \text{resample } (\mathcal{M}) \text{ according to weights } w_j
15. \hspace{1em} \textbf{end}
Sample-based Localization (sonar)
Particles/Samples

- The more particles $M$, the better is the approximation.
- Curse of dimensionality:
  \[ \sup E |x_k - \hat{x}_k| \leq \frac{C}{\sqrt{M}} \]

  Independent of time $k$

  Exponential in the dimension of the state space

  Bound on expected error

- Particle filter works well for low dimensional problems
  - Localization dimension: 3
  - SLAM dimension: $3 + 2N$ (N can be large!)
Tour of the Smithsonian National Museum of American History
Using Ceiling Maps for Localization

[Dellaert et al. 99]
Vision-Based Localization

$P(z|x)$
Measurement \( z \):

\[
P(z|x):\]

Under a Light
Next to a Light

Measurement $z$:  

$P(z|x)$:
Elsewhere

Measurement $z$: $P(z|x)$: