Foundations of State Estimation

Topics:  Bayes Filters  
Kalman Filters  
Hidden Markov Models

• Additional reading:

Issues

• Statement:
  • Given a set of observations of the world, what is the current state of the world?

• Alternatives:
  • Given a set of observations of the world, what is the state of the world at time $t$?

• Inputs:
  • Model
  • Sequence of observations, or perhaps actions and observations

• Outputs from different algorithms:
  • Most likely current state $x_i$
  • Probability distribution over possible current states $p(x_i)$
  • Most likely sequence of states over time: $x_1, x_2, x_3, x_4, \ldots x_t$
  • Sequence of probability distributions over time $p(x_1), p(x_2), p(x_3) \ldots p(x_t)$

• Choices:
  • How to compute the estimate $x_i$ or $p(x_i)$
  • How to represent $p(x_i)$
Graphical Models, Bayes’ Rule and the Markov Assumption

<table>
<thead>
<tr>
<th>Actions</th>
<th>Beliefs</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$Z_1$</td>
</tr>
<tr>
<td></td>
<td>$b_2$</td>
<td>$Z_2$</td>
</tr>
<tr>
<td>$T(x_j</td>
<td>a_i, x_i)$</td>
<td></td>
</tr>
</tbody>
</table>

States:
- $x_1$
- $x_2$

Bayes rule: $p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)}$

Markov: $p(x_t \mid x_{t-1}, a_t, a_0, z_0, a_1, z_1, \ldots, z_{t-1}) = p(x_t \mid x_{t-1}, a_t)$

The Bayes’ Filter

$p_i(x) = p(x \mid a_0, z_0, a_1, z_1, a_2, z_2, \ldots, a_t, z_t)$

According to Bayes rule:

$= p(z_t \mid x, a_0, z_0, a_1, z_1, a_2, z_2, \ldots, a_t) p(x \mid a_0, z_0, a_1, z_1, a_2, z_2, \ldots, a_t)$

According to the Markov assumption:

$= p(z_t \mid x) p(x \mid a_0, z_0, a_1, z_1, a_2, z_2, \ldots, a_t)$

Introducing an auxiliary variable:

$= p(z_t \mid x) \int p(x' \mid a_t, x') p(x' \mid a_0, z_0, a_1, z_1, \ldots, a_{t-1}, z_{t-1})$

And with recursion:

$= p(z_t \mid x) \int p(x' \mid a_t, x') p_{t-1}(x)$
Bayes Filter

The Kalman Filter

- Linear process and measurement models
- Gaussian noise
- Gaussian state estimate

Process model is
\[ x_t = Ax_{t-1} + Bu_{t-1} + q_{t-1} \]

Measurement model is
\[ z_t = Hx_{t-1} + r_t \]

Image adapted from Maybeck.
**Linear Kalman Filter**

- Only need first two moments:
  \[
  \hat{x}_t = E(x_t) \\
  C_t = E[(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T]
  \]
- Process model:
  \[
  \hat{x}_t^- = A\hat{x}_{t-1} + Bu_{t-1} \\
  C_t^- = AC_{t-1}A^T + Q
  \]
- Measurement model:
  \[
  K_t = C_t^-H^T(HC_t^-H^T + R)^{-1} \\
  \hat{x}_t = \hat{x}_t^- + K_t(z_t - H\hat{x}_t^-) \\
  C_t = (I - K_tH)C_t^-
  \]

**The Extended Kalman Filter**

- Process model is now
  \[
  x_t = f(x_{t-1}, u_{t-1}, q_{t-1})
  \]
- Measurement model is
  \[
  z_t = h(x_t, r_t)
  \]
- Linearising, we get
  \[
  \tilde{x}_t = f(x_{t-1}, u_{t-1}, 0) \\
  x_t \approx \tilde{x}_t + A(x_{t-1} - \tilde{x}_{t-1}) + Wq_{t-1} \\
  z_t \approx h(\tilde{x}_t, 0) + H(x_t - \tilde{x}_t) + Vr_t
  \]
  where A, H, W and V are the Jacobians:

\[
A = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n}
\end{bmatrix}, \quad
W = \begin{bmatrix}
\frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_n}{\partial q_1} & \cdots & \frac{\partial f_n}{\partial q_n}
\end{bmatrix}, \quad
H = \begin{bmatrix}
\frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial h_m}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_n}
\end{bmatrix}, \quad
V = \begin{bmatrix}
\frac{\partial h_1}{\partial v_1} & \cdots & \frac{\partial h_1}{\partial v_m} \\
\vdots & \ddots & \vdots \\
\frac{\partial h_m}{\partial v_1} & \cdots & \frac{\partial h_m}{\partial v_m}
\end{bmatrix}
\]
Updates for the EKF

- Still using only first two moments.
- Process model:
  \[ \hat{x}_t^- = f(\hat{x}_{t-1}, u_{t-1}, 0) \]
  \[ C_t^- = A_t C_{t-1} A_t^T + W_t Q_t W_t^T \]
- Measurement model:
  \[ K_t = C_t^- H_t^T (H_t C_t^- H_t^T + V_t R_t V_t^T)^{-1} \]
  \[ \hat{x}_t = \hat{x}_t^- + K_t (z_t - h(\hat{x}_t^-, 0)) \]
  \[ C_t = (I - K_t H_t) C_t^- \]

Robot Navigation

Leonard and Durrant-White, 1991

\[ \xi_{t-1} \]  
Initial belief

\[ \xi_t^- \]  
Action

\[ \xi_t \]  
Measurement

\[ \xi = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \]

\[ u = \begin{bmatrix} \Delta t \\ \Delta \theta \end{bmatrix} \]

\[ f(\xi, u, q) = \begin{bmatrix} x + \Delta t \cos \theta + q \Delta t \cos \theta \\ y + \Delta t \sin \theta + q \Delta t \sin \theta \\ \theta + \Delta \theta + q \Delta \theta \end{bmatrix} \]

\[ z = \begin{bmatrix} r \\ b \end{bmatrix} \]

\[ h(\xi, v) = \begin{bmatrix} \text{range} \\ \text{bearing} \end{bmatrix} = \begin{bmatrix} \sqrt{(\lambda_x - x)^2 + (\lambda_y - y)^2} + v_r \\ \tan^{-1}\left(\frac{\lambda_y - y}{\lambda_x - x}\right) - \theta + v_\theta \end{bmatrix} \]
The EKF for Robot Localization

**Prediction step**

\[
f(\xi, u, q) = \begin{bmatrix} x + \Delta t \cos \theta + q \Delta t \cos \theta \\ y + \Delta t \sin \theta + q \Delta t \sin \theta \\ \theta + \Delta \theta + q \Delta \theta \end{bmatrix}
\]

\[
A = \begin{bmatrix} 1 & 0 & -\Delta t \sin \theta \\ 0 & 1 & \Delta t \cos \theta \\ 0 & 0 & 1 \end{bmatrix}
\]

\[
W = [\Delta t \cos \theta \quad \Delta t \sin \theta \quad \Delta \theta]^T
\]

**Measurement step**

\[
h(\xi, v) = \sqrt{(\lambda_x - x)^2 + (\lambda_y - y)^2 + v_r}
\]

\[
\tan^{-1}\left(\frac{\lambda_y - y}{\lambda_x - x}\right) - \theta + v_\theta
\]

\[
H = \begin{bmatrix}
x - \lambda_x & y - \lambda_y & 0 \\
\frac{r}{\lambda_y - y} & \frac{r}{\lambda_x - x} & -1 \\
\frac{r^2}{r^2} & \frac{r^2}{r^2} & 1
\end{bmatrix}
\]

\[
V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\]

Problems with Kalman Filters

- Unimodal probability distribution
- Gaussian sensor and motion error models
  - Particularly painful for range data than can have occlusions
- Quadratic in number of state features

Data Association
Hidden Markov Models

- Discrete states, actions and observations
  - \( f(\cdot, \cdot, \cdot) \), \( h(\cdot, \cdot) \) can now be written as tables

(Somewhat) Useful for Localization in Topological Maps

Observations can be features such as corridor features, junction features, etc.
Belief Tracking

- Estimating $p_t(x)$ is now easy
- After each action $a_t$ and observation $z_t$, $\forall x \in X$, update:
  $$ p_t(x) = p(z_t | x) \sum_{x'} p(x | a_t, x') p_{t-1}(x) $$
- This algorithm is quadratic in $|X|$.
  - (Recall that Kalman Filter is quadratic in number of state features. Continuous $X$ means infinite number of states.)

The Three Basic Problems for HMMs

1) Given the history $O=a_1,z_1,a_2,z_2,...,a_T,z_T$, and a model $\lambda=(A,B,\pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of the history, given the model?

2) Given the history $O=a_1,z_1,a_2,z_2,...,a_T,z_T$ and a model $\lambda$, how do we choose a corresponding state sequence $X=x_1,x_2,...,x_T$ which is optimal in some meaningful sense (i.e., best “explains” the observations)?

3) How do we adjust the model parameters $\lambda=(A,B,\pi)$ to maximize $P(O|\lambda)$?
HMM Basic Problem 1

- Probability of history $O$ given $\lambda$ is sum over all state sequences $Q=x_1,x_2,x_3,...,x_T$:

$$P(O \mid \lambda) = \sum_{Q} P(O \mid Q, \lambda)P(Q \mid \lambda) = \sum_{q_1,q_2,...} \pi(x_1)p(z_1 \mid x_1)p(x_2 \mid x_1,a_1)p(z_2 \mid x_2)p(x_3 \mid x_2,a_2)$$

- Summing over all state sequences is $2^T \cdot |X|^T$
- Instead, build lattice of states forward in time, computing probabilities of each possible trajectory as lattice is built
- Forward algorithm is $|X|^{2T}$

HMM Basic Problem 1

1. Initialization

$$\alpha_1(i) = \pi_i p(z_1 \mid x_i)$$

2. Induction: Repeat for $t=1:T$

$$\alpha_{t+1}(i) = \left[ \sum_{j=1}^{\left| \mathcal{X} \right|} \alpha_t(j) p(x_j \mid x_i) \right] p(z_{t+1} \mid x_i)$$

3. Termination:

$$p(O \mid \lambda) = \sum_{i=1}^{\left| \mathcal{X} \right|} \alpha_t(i)$$
HMM Basic Problem 2

- Viterbi Decoding: Same principle as forward algorithm, with an extra term

1) Initialization
\[ \delta_1(i) = \pi_i p(z_1 \mid x_i) \]
\[ \psi_1(i) = 0 \]

2) Induction
\[ \delta_t(i) = \max_{1 \leq j \leq |X|} \left[ \delta_{t-1}(j) p(x_i \mid x_j, a_j) \right] p(z_t \mid x_i) \]
\[ \psi_t(i) = \arg \max_{1 \leq j \leq |X|} \left[ \delta_{t-1}(j) p(x_i \mid x_j, a_j) \right] \]

3) Termination
\[ P^* = \max_{1 \leq i \leq |X|} [\delta_T(i)] \]
\[ x_T^* = \arg \max_{1 \leq i \leq |X|} [\delta_T(i)] \]

4) Back tracking
\[ x_t^* = \psi_{t+1}(x_{t+1}^*) \]

What you should know

- Form of Bayes’ Filter
- Kalman filter representation
  - How to take a state estimation problem and represent it as a Kalman (or Extended Kalman) filter
  - What terms are needed and how to find them
- What a Hidden Markov Model is
- The Forward algorithm
- The Viterbi algorithm