Robotics

Estimation and Learning
with Dan Lee

Week 2.
Kalman Filter

2.3 Maximum-A-Posterior Estimation

Penn
Engineering

Bayesian Kalman Filtering

- Apply the MAP to Bayes’ Rule
- Solve the maximization
- Establish Kalman Filter update method

Bayesian filtering

- Bayes’ Rule

$$p(A|B) = \frac{P(\beta|\alpha)P(\alpha)}{P(\beta)}$$

- Given from Kalman model:

$$p(x_t|x_{t-1}) = \mathcal{N}(Ax_{t-1}, AP_{t-1}A^T + \Sigma_m)$$
$$p(z_t|x_t) = \mathcal{N}(Cx_t, CP_tC^T + \Sigma_o)$$

Bayesian filtering

- Apply Bayes' Rule $p(\alpha|\beta) = \dfrac{P(\beta|\alpha)P(\alpha)}{P(\beta)}$

$$p(x_t|x_{t-1}) = \mathcal{N}(Ax_{t-1}, AP_{t-1}A^T + \Sigma_m) \rightarrow \boxed{\alpha}$$

Prior

$$p(z_t|x_t) = \mathcal{N}(Cx_t, CP_tC^T + \Sigma_o) \rightarrow \boxed{\beta|\alpha}$$

Likelihood

$$\boxed{\boxed{p(x_t|z_t, x_{t-1})} = \frac{p(z_t|x_t, x_{t-1})p(x_t|x_{t-1})}{P(z_t)}}$$

Posterior

Bayesian filtering

- Posterior distribution is not Gaussian
- MAP Estimates “optimal” x_t value
- Use MAP estimates to form a new mean and variance for the state

Bayesian filtering

- Calculate the Maximum A Posteriori Estimate

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}}\, p(x_t|z_t, x_{t-1})$$

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \frac{p(z_t|x_t)p(x_t|x_{t-1})}{P(z_t)}$$

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}}\, p(z_t|x_t)p(x_t|x_{t-1})$$

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}}\, \mathcal{N}(Cx_t, CP_tC^T + \Sigma_o)\mathcal{N}(Ax_{t-1}, AP_{t-1}A^T + \Sigma_m)$$

Bayesian filtering

- Calculate the Maximum A Posteriori Estimate

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \mathcal{N}(Cx_t, CP_tC^T + \Sigma_o)\mathcal{N}(Ax_t, AP_{t-1}A^T + \Sigma_m)$$

- Simplify with these substitutions

$$\begin{gathered} P = P_t = AP_{t-1}A^T + \Sigma_m \\ R = CP_tC^T + \Sigma_o \end{gathered}$$

- Simplify the exponential form of $\mathcal{N}$ via logarithms

$$\hat{x}_t = \underset{x_t}{\operatorname{argmin}} \begin{matrix} (z_t - Cx_t)R^{-1}(z_t - Cx_t) \\ +(x_t - Ax_{t-1})P^{-1}(x_t - Ax_{t-1}) \end{matrix}$$

Bayesian filtering

- Solve optimization by setting the derivative to zero

$$\hat{x}_t = \underset{x_t}{\operatorname{argmin}} \begin{matrix} (z_t - Cx_t)R^{-1}(z_t - Cx_t) \\ +(x_t - Ax_{t-1})P^{-1}(x_t - Ax_{t-1}) \end{matrix}$$

$$0 = \frac{d}{dx_t}\begin{pmatrix} (z_t - Cx_t)R^{-1}(z_t - Cx_t) \\ +(x_t - Ax_{t-1})P^{-1}(x_t - Ax_{t-1}) \end{pmatrix}$$

Bayesian filtering

- Collect terms in the derivative

$$(C^T R^{-1} C + P^{-1}) x_t = z_t^T R^{-1} C + P^{-1} A x_{t-1}$$

$$x_t = (C^T R^{-1} C + P^{-1})^{-1} (z_t^T R^{-1} C + P^{-1} A x_{t-1})$$

- Apply the *Matrix Inversion Lemma*

$$(C^T R^{-1} C + P^{-1})^{-1} = P - P C^T (R + C P C^T)^{-1} C P$$

- Define *Kalman Gain*: $K = P C^T (R + C P C^T)^{-1}$

Bayesian filtering

- Expand the terms

$$x_t = (C^T R^{-1} C + P^{-1})^{-1} (C^T R^{-1} y_t + P^{-1} A x_{t-1})$$

$$x_t = (P - KCP)(C^T R^{-1} z_t + P^{-1} A x_{t-1})$$

$$x_t = A x_{t-1} + P C^T R^{-1} z_t - KCA x_{t-1} - KCPC^T R^{-1} y_t$$

$$x_t = A x_{t-1} - KCA x_{t-1} + (P C^T R^{-1} - KCPC^T R^{-1}) y_t$$

$$x_t = A x_{t-1} - KCA x_{t-1} + K y_t$$

$$\boxed{\hat{x}_t = A x_{t-1} + K(z_t - CA x_{t-1})}$$

- Convince yourself that $K = P C^T R^{-1} - KCPC^T R^{-1}$

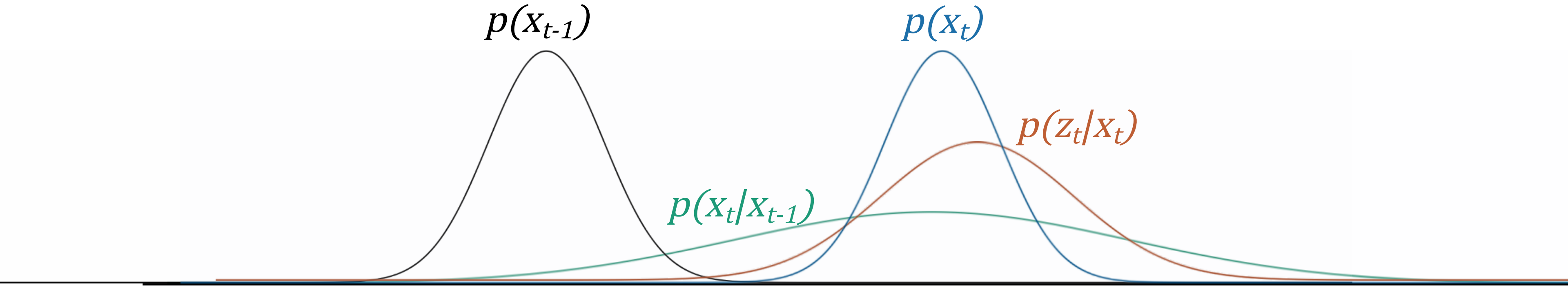
Bayesian filtering

- Must update the covariance of the state

$$\boxed{\hat{P}_t = P - KCP}$$

1D Visualization

- The position of x is moving forward
  - Uncertain motion model increases the spread
- We observe a noisy position estimate, z_t
- The corrected position has less spread than both the observation and motion adjusted state


$p(x_{t-1})$
$p(x_t)$
$p(z_t|x_t)$
$p(x_t|x_{t-1})$