CS354

Nathan Sprague

October 25, 2020

Probabilistic State Representations: Continuous

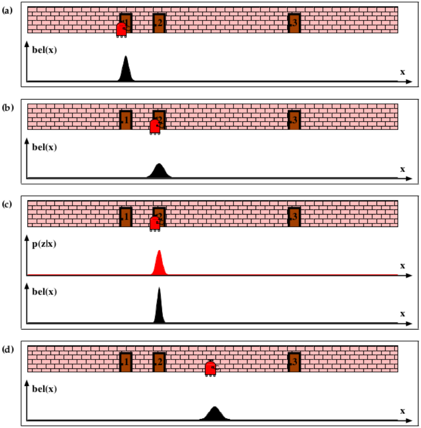

Figure 7.6 Application of the Kalman filter algorithm to mobile robot localization. All densities are represented by unimodal Gaussians.

Probabilistic Robotics. Thrun, Burgard, Fox, 2005

Combining Evidence

- Imagine two independent measurements of some unknown quantity:
  - x_1 with variance σ_1^2
  - x_2 with variance σ_2^2
- How should we combine these measurements?

Combining Evidence

- Imagine two independent measurements of some unknown quantity:
  - x_1 with variance σ_1^2
  - x_2 with variance σ_2^2
- How should we combine these measurements?
- We can take a weighted average:
  - $\hat{x} = \omega_1 x_1 + \omega_2 x_2$ (where $\omega_1 + \omega_2 = 1$)
- What should the weights be???

Combining Evidence

- Imagine two independent measurements of some unknown quantity:
  - x_1 with variance σ_1^2
  - x_2 with variance σ_2^2
- How should we combine these measurements?
- We can take a weighted average:
  - $\hat{x} = \omega_1 x_1 + \omega_2 x_2$ (where $\omega_1 + \omega_2 = 1$)
- What should the weights be???
- We want to find weights that minimize variance (uncertainty) in the estimate:
  - $\sigma^2 = E[(\hat{x} - E[\hat{x}])^2]$

Combining Evidence – Solution

(Derivation not shown...)

$$\hat{x} = \frac{\sigma_2^2 x_1 + \sigma_1^2 x_2}{\sigma_1^2 + \sigma_2^2}$$

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

Updating an Existing Estimate

Let's reinterpret x_1 to be the old state estimate and σ_1^2 to be the variance in that estimate. Now x_2 represents a new sensor reading. After some algebra...

$$\hat{x} = x_1 + \frac{\sigma_1^2(x_2 - x_1)}{\sigma_1^2 + \sigma_2^2}$$

$$\sigma^2 = \sigma_1^2 - \frac{\sigma_1^2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$$

Let $k = \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}$, these become...

$$\hat{x} = x_1 + k(x_2 - x_1)$$

$$\sigma^2 = \sigma_1^2 - k\sigma_1^2$$

1D Kalman Filter

$k = \frac{\sigma_{t-1}^2}{\sigma_{t-1}^2 + \sigma_z^2}$, these become...

$$\hat{x}_t = \hat{x}_{t-1} + k(z_{t-1} - \hat{x}_{t-1})$$

$$\sigma_t^2 = \sigma_{t-1}^2 - k\sigma_{t-1}^2$$

Vector-Valued State

- Kalman filter generalizes this to multivariate data and allows for state dynamics that are influenced by a control signal.
- We may also be combining evidence from multiple sensors
  - Sensor fusion

Linear System Models

- State can include information other than position. E.g. velocity.
- Linear model of an object moving with a fixed velocity in 2d:
  - $x_{t+1} = x_t + \dot{x}_t dt$
  - $y_{t+1} = y_t + \dot{y}_t dt$
  - $\dot{x}_{t+1} = \dot{x}_t$
  - $\dot{y}_{t+1} = \dot{y}_t$
- dt is time.
- $\dot{x}_t$ is velocity along the x axis.

Linear System Model in Matrix Form

This is equivalent to the last slide:

$$\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t$$

Kalman Filter

- Assumes:
  - Linear state dynamics
  - Linear sensor model
  - Normally distributed noise in the state dynamics
  - Normally distributed noise in the sensor model
- State Transition Model:
  - $\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_{t-1} + \mathbf{w}_{t-1}$
  - $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ (Normal distribution with mean 0 and covariance $\mathbf{Q}$)
- Sensor Model:
  - $\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t$
  - $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$

Full Example For 2d Constant Velocity

State Transition Model:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} .01 & 0 & 0 & 0 \\ 0 & .01 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Sensor Model (sensor readings based only on position):

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} .05 & 0 \\ 0 & .05 \end{bmatrix}$$

Kalman Filter in One Slide

- Predict:
  Project the state forward:
  $$\hat{\mathbf{x}}_t^- = F\hat{\mathbf{x}}_{t-1} + B\mathbf{u}_{t-1}$$
  Project the covariance of the state estimate forward:
  $$\mathbf{P}_t^- = F\mathbf{P}_{t-1}F^T + \mathbf{Q}$$
- Correct:
  Compute the Kalman gain:
  $$\mathbf{K}_t = \mathbf{P}_t^- H^T (H\mathbf{P}_t^- H^T + \mathbf{R})^{-1}$$
  Update the estimate with the measurement:
  $$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t(\mathbf{z}_t - H\hat{\mathbf{x}}_t^-)$$
  Update the estimate covariance:
  $$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t H\mathbf{P}_t^-$$

Extended Kalman Filter

- What if the state dynamics and/or sensor model are NOT linear?
- State Transition Model:
  - $\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \mathbf{w}_{t-1}$
- Sensor Model:
  - $\mathbf{z}_t = h(\mathbf{x}_t) + \mathbf{v}_t$

Jacobian

The Jacobian is the generalization of the derivative for vector-valued functions:

$$\mathbf{J} = \frac{d\mathbf{f}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

$$\mathbf{J}_{ij} = \frac{\partial f_i}{\partial x_j}$$

tex borrowed from Wikipedia

Extended Kalman Filter

- As long as f and h are differentiable, we can still use the (Extended) Kalman filter.
- Basically, we just replace the state transition and sensor update matrices with the corresponding Jacobians.