7630 – Autonomous Robotics

Introduction to Bayesian Estimation

Cédric Pradalier

Today

Outline

Bayesian Fusion

Assumptions

- A variable X is being estimated
- A set of observations $\{Z_i\}$ correlated with X are being made.

$$P(X\,Z_1\,\ldots\,Z_n) = P(X)\prod_{i=1}^{n} P(Z_i|X) \tag{1}$$

Bayesian inference:

$$P(X|Z_1\,\ldots\,Z_n) \propto P(X)\prod_{i=1}^{n} P(Z_i|X) \tag{2}$$

Bayesian Fusion: Gaussians

Assumptions

- All distributions are Gaussian: $P(X) \rightarrow \mathcal{N}(\mu_0, \Sigma_0)$, $P(Z_i|X) \rightarrow \mathcal{N}(\nu_i, \sigma_i)$.

$$P(X|Z_1 \ldots Z_n) \propto P(X) \prod_{i=1}^{n} P(Z_i|X) \tag{3}$$

- Hence $P(X|Z_1 \ldots Z_n)$ is Gaussian $\mathcal{N}(\mu_n, \Sigma_n)$.

Bayesian inference:

$$\log P(X|Z_1 \ldots Z_n) = \log P(X) + \sum_{i=1}^{n} \log P(Z_i|X) + Cst$$

Georgia Tech Lorraine

Bayesian Fusion: Gaussians

Assumptions

- $P(X) \rightarrow \mathcal{N}(\mu_0, \Sigma_0)$, $P(Z_i|X) \rightarrow \mathcal{N}(\nu_i = h(x), \sigma_i)$.

Bayesian inference:

$$\begin{aligned}
\log P(X|Z_1 \ldots Z_n) &= \log P(X) + \sum_{i=1}^{n} \log P(Z_i|X) + Cst \\
(\mu_n - x)^T \Sigma_n^{-1} (\mu_n - x) &= (\mu_0 - x)^T \Sigma_0^{-1} (\mu_0 - x) \\
&+ \sum_{i=1}^{n} (\nu_i - h(x))^T \sigma_i^{-1} (\nu_i - h(x)) + Cst
\end{aligned}$$

Bayesian Fusion Special Cases

$h(x) = x$ and $n = 1$

$$\begin{aligned}
&(\mu_1 - x)^T \Sigma_1^{-1} (\mu_1 - x) \\
&\quad = \quad (\mu_0 - x)^T \Sigma_0^{-1} (\mu_0 - x) + (\nu_1 - x)^T \sigma_1^{-1} (\nu_1 - x) + Cst
\end{aligned}$$

$$\begin{aligned}
&\mu_1^T \Sigma_1^{-1} \mu_1 + 2\mu_1^T \Sigma_1^{-1} x + x^T \Sigma_1^{-1} x \\
&\quad = \quad \mu_0^T \Sigma_0^{-1} \mu_0 + 2\mu_0^T \Sigma_0^{-1} x + x^T \Sigma_0^{-1} x \\
&\quad + \quad \nu_1^T \sigma_1^{-1} \nu_1 + 2\nu_1^T \sigma_1^{-1} x + x^T \sigma_1^{-1} x + Cste
\end{aligned}$$

Hence

$$\begin{aligned}
\Sigma_1^{-1} &= \Sigma_0^{-1} + \sigma_1^{-1} \\
\mu_1 &= \Sigma_1 \left(\Sigma_0^{-1} \mu_0 + \sigma_1^{-1} \nu_1 \right)
\end{aligned}$$

Georgia Tech Lorraine

Bayesian Fusion Special Cases

$h(x) = x$ and $n = 1$

$$
\begin{aligned}
\Sigma_1^{-1} &= \Sigma_0^{-1} + \sigma_1^{-1} \\
\mu_1 &= \Sigma_1 \left(\Sigma_0^{-1} \mu_0 + \sigma_1^{-1} \nu_1 \right)
\end{aligned}
$$

Can be rewritten as:

$$
\begin{aligned}
K &= \Sigma_0 \left(\Sigma_0 + \sigma_1 \right)^{-1} \\
\Sigma_1 &= (I - K)\Sigma_0 \\
\mu_1 &= \mu_0 + K \left(\nu_1 - \mu_0 \right)
\end{aligned}
$$

Bayesian Fusion Special Cases

$h(x) = Hx$ and $n = 1$

$$
\begin{aligned}
K &= \Sigma_0 H^T \left(H \Sigma_0 H^T + \sigma_1\right) \\
\Sigma_1 &= (I - KH)\Sigma_0 \\
\mu_1 &= \mu_0 + K\left(\nu_1 - H\mu_0\right)
\end{aligned}
$$

$h(x)$ non linear and $n = 1$

$$
\begin{aligned}
H &= \frac{\partial h}{\partial x} \longrightarrow \text{Jacobian} \\
K &= \Sigma_0 H^T \left(H \Sigma_0 H^T + \sigma_1\right)^{-1} \\
\Sigma_1 &= (I - KH)\Sigma_0 \\
\mu_1 &= \mu_0 + K\left(\nu_1 - h(\mu_0)\right)
\end{aligned}
$$

Bayesian Fusion Special Cases

Maximum A Posteriori (MAP)

- We are just interested in the maximum of $P(X|Z_1 \dots X_n)$, i.e. μ_n.

$$
\begin{aligned}
\arg\max_X P(X|Z_1\ \dots\ Z_n) &= \arg\min_X \log P(X|Z_1\ \dots\ Z_n) \\
&= \arg\min_X (\mu_0 - x)^T \Sigma_0^{-1} (\mu_0 - x) \\
&+ \sum_{i=1}^{n} (\nu_i - h(x))^T \sigma_i^{-1} (\nu_i - h(x))
\end{aligned}
$$

- Weighted least-square minimisation (or regression), non-linear if h is non-linear.

Recursive Bayesian Fusion

Assumptions

- A variable X is being estimated
- A set of observations $\{Z_i\}$ correlated with X are being made over time.
- We denote $Bel_k(X) = P(X|Z_1 \ldots Z_k)$.

Bayesian inference:

$$
\begin{aligned}
Bel_k(X) &= P(X|Z_1 \ldots Z_k) \propto P(X) \prod_{i=1}^{k} P(Z_i|X) \\
&= \left[P(X) \prod_{i=1}^{k-1} P(Z_i|X) \right] P(Z_k|X) \\
&= P(Z_k|X) \cdot Bel_{k-1}(X)
\end{aligned}
$$

Recursive Bayesian Fusion, Gaussian Assumption

Assumptions

$$Bel_k(X) = P(Z_k|X) \cdot Bel_{k-1}(X)$$

- Bel_k is Gaussian $\rightarrow \mathcal{N}(x_k, P_k)$
- $P(Z_k|X)$ is Gaussian $\rightarrow \mathcal{N}(h(X), R)$

This is Gaussian Bayesian fusion with $n = 1$:

$$\begin{aligned}
H &= \frac{\partial h}{\partial x} \longrightarrow \text{Jacobian} \\
K &= P_{k-1} H^T \left(H P_{k-1} H^T + R \right)^{-1} \\
P_k &= (I - KH) P_{k-1} \\
x_k &= x_{k-1} + K \left(z_k - h(x_{k-1}) \right)
\end{aligned}$$

Recursive Bayesian Filter

Assumptions

- A variable X_k is being estimated at time k
- A set of observations $\{Z_i\}$ correlated with X are being made over time.
- A model of changes of X_k is available as $P(X_k|X_{k-1}, U_{k-1})$.
- We denote $Bel_k(X_k) = P(X_k|Z_1 \ldots Z_k, U_1 \ldots U_k) = P(X_k|\mathcal{Z}_k, \mathcal{U}_k)$.

Recursive Bayesian Filter

Bayesian inference:

$$
\begin{aligned}
Bel_k(X_k) &= P(X_k|\mathcal{Z}_k,\mathcal{U}_k) \propto P(Z_k|X_k\mathcal{U}_k,\mathcal{Z}_{k-1})P(X_k|\mathcal{U}_k,\mathcal{Z}_{k-1}) \\
&= P(Z_k|X_k)P(X_k|\mathcal{U}_k,\mathcal{Z}_{k-1}) \\
&= P(Z_k|X_k)\int_{X_{k-1}} P(X_k|\mathcal{U}_k,\mathcal{Z}_{k-1},X_{k-1})P(X_{k-1}|\mathcal{U}_k,\mathcal{Z}_{k-1}) \\
&= P(Z_k|X_k)\int_{X_{k-1}} P(X_k|U_{k-1},X_{k-1})P(X_{k-1}|\mathcal{U}_{k-1},\mathcal{Z}_{k-1}) \\
&= P(Z_k|X_k)\int_{X_{k-1}} P(X_k|U_{k-1},X_{k-1})Bel_{k-1}(X_{k-1})
\end{aligned}
$$

Important: know how to derive that.

Recursive Bayesian Filter: (Extended) Kalman Filter

Assumptions

$$
\begin{aligned}
Bel_{k-1}(X_k) &= \int_{X_{k-1}} P(X_k|X_{k-1}) Bel_{k-1}(X_{k-1}) \\
Bel_k(X_k) &= P(Z_k|X) \cdot Bel_{k-1}(X_k)
\end{aligned}
$$

- Bel_k is Gaussian $\rightarrow \mathcal{N}(x_k, P_k)$
- $P(X_k|X_{k-1})$ is Gaussian $\rightarrow \mathcal{N}(f(X), Q)$
- $P(Z_k|X_k)$ is Gaussian $\rightarrow \mathcal{N}(h(X), R)$

(Extended) Kalman Filter: Prediction Stage

Assumptions

$$Bel_{k-1}(X_k) = \int_{X_{k-1}} P(X_k|X_{k-1}) Bel_{k-1}(X_{k-1} \rightarrow \mathcal{N}(\bar{x}_k, \bar{P}_k)$$

- $P(X_k|X_{k-1})$ is Gaussian $\rightarrow \mathcal{N}(f(X_{k-1}, U_{k-1}), Q)$

$$\begin{aligned}
A &= \frac{\partial f}{\partial x} \longrightarrow \text{Jacobian} \\
B &= \frac{\partial f}{\partial u} \longrightarrow \text{Jacobian} \\
\bar{x}_k &= f(x_{k-1}, u_{k-1}) \\
\bar{P}_k &= AP_{k-1}A^T + (BQ_uB^T) + Q
\end{aligned}$$

(Extended) Kalman Filter, Observation stage

Assumptions

$$Bel_k(X_k) = P(Z_k|X_k) \cdot Bel_{k-1}(X_k)$$

- $Bel_{k-1}(X_k)$ is Gaussian $\rightarrow \mathcal{N}(\bar{x}_k, \bar{P}_k)$
- $Bel_k(X_k)$ is Gaussian $\rightarrow \mathcal{N}(x_k, P_k)$
- $P(Z_k|X)$ is Gaussian $\rightarrow \mathcal{N}(h(X), R)$

This is Gaussian Bayesian fusion:

$$\begin{aligned}
H &= \frac{\partial h}{\partial x} \longrightarrow \text{Jacobian} \\
K &= \bar{P}_k H^T \left(H\bar{P}_k H^T + R\right)^{-1} \\
P_k &= (I - KH)\bar{P}_k \\
x_k &= \bar{x}_{k-1} + K\left(z_k - h(\bar{x}_{k-1})\right)
\end{aligned}$$

Outline

Bayesian Estimation

Examples

Homework

Objectives

Kalman Filter

- Understand the important matrices
- Design your own filter
- Use cases for robotics

System 1: Argos Float

Description

- Float buoy
- GPS measurement every second

Objective

- Continuous localisation

System 2: GPS Navigation system

Description

- GPS Navigation system in a **car**
- GPS measurement every second

Objective

- Continuous localisation

System 3: Integrated GPS Navigation system

Description

- GPS Navigation system in a **car**
- Rear wheel displacement measure (e.g. differential)

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} \frac{\Delta s_L + \Delta s_R}{2} \\ 0 \\ \frac{\Delta s_R - \Delta s_L}{e} \end{bmatrix} \tag{4}$$

- GPS measurement every second

Objective

- Continuous localisation

System 4: Integrated GPS Navigation system

Description

- GPS Navigation system in a **car**
- Speed and steering measurement

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \frac{v\tan(\beta)}{L} \end{bmatrix} \quad (5)$$

- GPS measurement every second

Objective

- Continuous localisation

System 5: Indoor Navigation system

Description

- Indoor robot (e.g. Roomba)
- Differential wheel measurement
- Known feature observation with orientation, in body frame

$$Z_i = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} \tag{6}$$

Objective

- Continuous localisation

System 6: Indoor Navigation system

Description

- Indoor robot (e.g. Roomba)
- Differential wheel measurement
- Known feature observation: position only in body frame

$$Z_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (7)$$

Objective

- Continuous localisation

System 6b: Indoor Navigation system

Description

- Indoor robot (e.g. Roomba)
- Differential wheel measurement
- Known feature observation: range and bearing in body frame

$$Z_i = \begin{bmatrix} \rho_i \\ \beta_i \end{bmatrix} \tag{8}$$

Objective

- Continuous localisation

System 7: Indoor Navigation system

Description

- Indoor robot (e.g. Roomba)
- Differential wheel measurement
- Known feature observation, bearing only in body frame

$$Z_i = \left[\begin{array}{c} \beta_i \end{array} \right] \tag{9}$$

Objective

- Continuous localisation

System 8: Indoor Navigation system

Description

- Indoor robot (e.g. Roomba)
- Differential wheel measurement
- Known feature observation, range only in body frame

$$Z_i = \left[\ \rho_i \ \right] \tag{10}$$

Objective

- Continuous localisation

System 9: Underwater system

Description

- Torpedo-shaped robot
- Lift-drag based motion model
- Known feature observation, range only (e.g. sonar pinger)

$$Z_i = \left[\; \rho_i \; \right] \tag{11}$$

Objective

- Continuous localisation

System 10: Feature mapper

Description

- Indoor robot
- Known localisation
- Observation of *n* features with known Ids.
- Observation type: range, bearing, position, pose...

Objective

- Map estimation

System 11: Feature-based SLAM

Description

- Indoor robot
- Unknown localisation
- Observation of k features with known Ids.
- Observation type: range, bearing, position, pose...

Objective

- Localisation and Map estimation

System 12: Extrinsic calibration

Description

- Indoor robot with k sensors.
- Known localisation
- Joint observation of one feature with k sensors.
- Observation type: range, bearing, position, pose...

Objective

- Sensor position with respect to reference frame (e.g. sensor 1)
- Detection of loose sensors.

System 13: auto-calibration

Description

- Indoor robot with 1 sensor.
- Differential motion model integrating wheel diameter

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} \frac{r_L \Delta\theta_L + r_R \Delta\theta_R}{2} \\ 0 \\ \frac{r_R \Delta s_R - r_L \Delta s_L}{e} \end{bmatrix} \tag{12}$$

- Observation type: range, bearing, position, pose... with respect to known map.

Objective

- Auto-estimate wheel diameter r_L and r_R and inter-wheel spacing e
- Puncture detection.

Outline

Bayesian Estimation

Examples

Homework

Mapping

Objectives

- Estimate the landmark positions knowing (perfectly) the pose of the robot observing them.
- Kalman Filter

Solution

- Write the the equation of the Kalman Filter (50%)
- Manage new landmarks in a list (50%)

Localisation

Objectives

- Localise the robot based on landmark measurements and odometry.
- Extended Kalman Filter

Solution

- Write the correct jacobian (90%)
- Fill-in the TODO section and test it (10%)

SLAM

Objectives

- Localise the robot AND build a map of landmarks based on landmark measurements and odometry.
- Extended Kalman Filter

Solution

- Write the correct jacobian (90%)
- Fill-in the TODO section and test it (10%)