NA 568 - Winter 2024

Bayes Filter and Kalman Filter

Minghan Zhu and Maani Ghaffari

January 17, 2024

UNIVERSITY OF MICHIGAN

What we learned in the last lecture

Probability recap

- Conditional probability
- Law of total probability
- Conditional independence

Bayes' rule

- Data and hypothesis.
- Prior, posterior, likelihood, and evidence.
- Causal and diagnostic reasoning.

Bayes Filters: Framework

- Given:
  - Stream of observations $z_{1:t}$ and action data $u_{1:t}$
  - Sensor/measurement model $p(z_t|x_t)$
  - Action/motion/transition model $p(x_t|x_{t-1},u_t)$
- Wanted:
  - The state X_t of dynamical system
  - The posterior of state is called belief $bel(x_t) = p(x_t|z_{1:t}, u_{1:t})$

Algorithm 1 `Bayes-filter`

Require: Belief $bel(x_{t-1}) = p(x_{t-1}|z_{1:t-1}, u_{1:t-1})$, action u_t, measurement z_t;

1: **for** all state variables **do**
2: $\overline{bel}(x_t) = \int p(x_t|x_{t-1}, u_t) bel(x_{t-1}) \mathrm{d}x_{t-1}$ // Predict using action/control input u_t
3: $bel(x_t) = \eta p(z_t|x_t)\overline{bel}(x_t)$ // Update using perceptual data z_t
4: **return** $bel(x_t)$

Bayes Filter

Algorithm 2 `Bayes-filter`

Require: Belief $bel(x_{t-1}) = p(x_{t-1}|z_{1:t-1}, u_{1:t-1})$, action u_t, measurement z_t;

for all state variables **do**

  $\overline{bel}(x_t) = \int p(x_t|x_{t-1}, u_t) bel(x_{t-1}) \mathrm{d}x_{t-1}$ // Predict using action/control input u_t

  $bel(x_t) = \eta p(z_t|x_t) \overline{bel}(x_t)$ // Update using perceptual data z_t

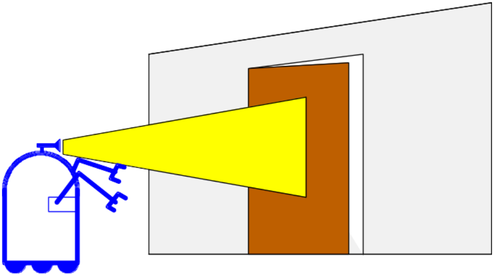
return $bel(x_t)$

Ingredients:

- Bayes' rule
- Conditional independence
- Law of total probability

Simple Example of State Estimation

- Suppose a robot obtains measurement z, e.g., using its camera;
- What is $p(\text{open}|z)$?



- $p(z|\text{open})$ is **causal**.
- $p(\text{open}|z)$ is **diagnostic**.
- Often causal knowledge is easier to obtain.
- Bayes rule allows us to use causal knowledge:

$$p(\text{open}|z) = \frac{p(z|\text{open})p(\text{open})}{p(z)}$$

Sensor model (likelihood):

- $p(z = \text{sense\_open}|\text{open}) = 0.6$
- $p(z = \text{sense\_open}|\neg\text{open}) = 0.3$

Prior knowledge (non-informative in this case):

- $p(\text{open}) = p(\neg\text{open}) = 0.5$

Sensor model (likelihood):

- $p(z = \text{sense\_open}|\text{open}) = 0.6$
- $p(z = \text{sense\_open}|\neg\text{open}) = 0.3$

Prior knowledge (non-informative in this case):

- $p(\text{open}) = p(\neg\text{open}) = 0.5$

Update/Correction:

$$p(\text{open}|z) = \frac{p(z|\text{open})p(\text{open})}{p(z|\text{open})p(\text{open}) + p(z|\neg\text{open})p(\neg\text{open})}$$

$$p(\text{open}|z = \text{sense\_open}) = \frac{0.6 \times 0.5}{0.6 \times 0.5 + 0.3 \times 0.5} = 0.6667$$

Remark

z *raises the probability that the door is open.*

- Suppose our robot obtains another observation z_2.
- How can we integrate this new information?
- More generally, how can we estimate $p(x|z_1, \ldots, z_n)$?

$$p(x|z_1, \ldots, z_n) = \frac{p(z_n|x, z_1, \ldots, z_{n-1})p(x|z_1, \ldots, z_{n-1})}{p(z_n|z_1, \ldots, z_{n-1})}$$

Assumption (Markov Assumption)

z_n *is* **independent** *of* $z_1, \ldots, z_{n-1}$ **if** *we* **know** x.

$$p(x|z_1,\ldots,z_n) = \frac{p(z_n|x, z_1,\ldots,z_{n-1})p(x|z_1,\ldots,z_{n-1})}{p(z_n|z_1,\ldots,z_{n-1})}$$

Assumption (Markov Assumption)

z_n *is* **independent** *of* $z_1,\ldots,z_{n-1}$ **if** *we* **know** x.

or equivalently we can state:

Assumption (Markov Property)

*The Markov property states that "***the future is independent of the past if the present is known.***" A stochastic process that has this property is called a* **Markov process***.*

$$p(x|z_1,\ldots,z_n) = \frac{p(z_n|x,z_1,\ldots,z_{n-1})p(x|z_1,\ldots,z_{n-1})}{p(z_n|z_1,\ldots,z_{n-1})}$$

Assumption (Markov Assumption)

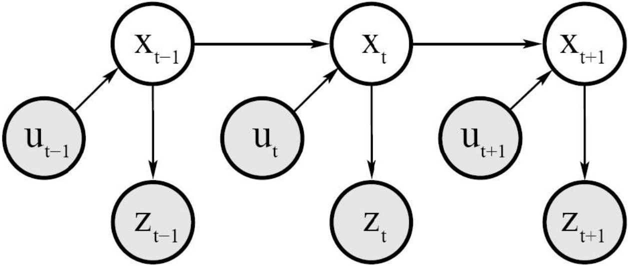
z_n *is* **independent** *of* $z_1,\ldots,z_{n-1}$ **if** *we* **know** x.

$$p(x|z_1,\ldots,z_n) = \frac{p(z_n|x)p(x|z_1,\ldots,z_{n-1})}{p(z_n|z_1,\ldots,z_{n-1})}$$

$$= \eta_n \; p(z_n|x)p(x|z_1,\ldots,z_{n-1}) = \eta_{1:n} \; \prod_{i=1}^{n} p(z_i|x)p(x)$$

where $\eta_{1:n} := \eta_1\eta_2\cdots\eta_n$.

Dynamic Bayesian Network for Controls, States, and Sensations



State Estimation

- Estimate the state x of a system given observations z and controls u
- **Goal:**

$$p(x \mid z, u)$$

Courtesy: C. Stachniss

Recursive Bayes Filter 1

$$bel(x_t) = \underline{p(x_t \mid z_{1:t}, u_{1:t})}$$

Definition of the belief

Courtesy: C. Stachniss

Recursive Bayes Filter 2

$$bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t})$$

$$= \underline{\eta \, p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) \, p(x_t \mid z_{1:t-1}, u_{1:t})}$$

Bayes' rule

Courtesy: C. Stachniss

Recursive Bayes Filter 3

$$
\begin{aligned}
bel(x_t) &= p(x_t \mid z_{1:t}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, \underline{p(z_t \mid x_t)}\, p(x_t \mid z_{1:t-1}, u_{1:t})
\end{aligned}
$$

Markov assumption

Courtesy: C. Stachniss

Recursive Bayes Filter 4

$$
\begin{aligned}
bel(x_t) &= p(x_t \mid z_{1:t}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t) \underline{\int p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t})} \\
&\qquad \underline{p(x_{t-1} \mid z_{1:t-1}, u_{1:t})}\, dx_{t-1}
\end{aligned}
$$

Law of total probability

Courtesy: C. Stachniss

Recursive Bayes Filter 5

$$
\begin{aligned}
bel(x_t) &= p(x_t \mid z_{1:t}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) \\
&\qquad\qquad p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1} \\
&= \eta\, p(z_t \mid x_t) \int \underline{p(x_t \mid x_{t-1}, u_t)}\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1}
\end{aligned}
$$

Markov assumption

Courtesy: C. Stachniss

Recursive Bayes Filter 6

$$
\begin{aligned}
bel(x_t) &= p(x_t \mid z_{1:t}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) \\
&\qquad\qquad p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1} \\
&= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1} \\
&= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:\underline{t-1}})\, dx_{t-1}
\end{aligned}
$$

Markov assumption

Courtesy: C. Stachniss

Recursive Bayes Filter 7

$$
\begin{aligned}
bel(x_t) &= p(x_t \mid z_{1:t}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) \\
&\qquad\qquad p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1} \\
&= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1} \\
&= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1})\, dx_{t-1} \\
&= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t)\, \underline{bel(x_{t-1})}\, dx_{t-1}
\end{aligned}
$$

Recursive term

Courtesy: C. Stachniss

Prediction and Correction Step

- Bayes filter can be written as a two step process
- **Prediction step**

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\ bel(x_{t-1})\, dx_{t-1}$$

- **Correction step**

$$bel(x_t) = \eta\ p(z_t \mid x_t)\ \overline{bel}(x_t)$$

Courtesy: C. Stachniss

Motion and Observation Model

- Prediction step

$$\overline{bel}(x_t) = \int \underbrace{p(x_t \mid u_t, x_{t-1})}_{\text{motion model}} \, bel(x_{t-1}) \, dx_{t-1}$$

- Correction step

$$bel(x_t) = \eta \, \underbrace{p(z_t \mid x_t)}_{\text{sensor or observation model}} \, \overline{bel}(x_t)$$

Courtesy: C. Stachniss

Bayes Filters: Framework

- Given:
  - Stream of observations $z_{1:t}$ and action data $u_{1:t}$
  - Sensor/measurement model $p(z_t|x_t)$
  - Action/motion/transition model $p(x_t|x_{t-1},u_t)$
- Wanted:
  - The state x_t of dynamical system
  - The posterior of state is called belief $bel(x_t) = p(x_t|z_{1:t}, u_{1:t})$

Algorithm 3 `Bayes-filter`

Require: Belief $bel(x_{t-1}) = p(x_{t-1}|z_{1:t-1}, u_{1:t-1})$, action u_t, measurement z_t;

1: **for** all state variables **do**
2: $\quad \overline{bel}(x_t) = \int p(x_t|x_{t-1}, u_t) bel(x_{t-1}) \mathrm{d}x_{t-1}$ // Predict using action/control input u_t
3: $\quad bel(x_t) = \eta p(z_t|x_t)\overline{bel}(x_t)$ // Update using perceptual data z_t
4: **return** $bel(x_t)$

Bayes Filters: Implementation Examples

Linear:

- Kalman Filter: unimodal linear filter
- Information Filter: unimodal linear filter

Nonlinear:

- Extended Kalman Filter: unimodal nonlinear filter with Gaussian noise assumption
- Extended Information Filter: unimodal nonlinear filter with Gaussian noise assumption
- Particle Filter: multimodal nonlinear filter

A discrete-time random process (random sequence), $\mathbf{w}_k$, is called white noise if:

$$\mathbb{E}[\mathbf{w}_k \mathbf{w}_j^{\mathsf{T}}] = \mathbf{Q}_k \delta_{kj}$$

where the Kronecker δ_{kj} is

$$\delta_{kj} = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases}$$

- The state of a dynamic system excited by white noise

  $$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{w}_k)$$

  is a discrete-time Markov process or Markov sequence.

- The state of a dynamic system is a Gaussian process iff the joint distribution of states at any finite subset of time steps are multivariate Gaussian.

- Assuming the initial condition is Gaussian, the state of a linear dynamic system excited by white Gaussian noise

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k$$

is called a Gauss-Markov process.

- Because of the whiteness of the process noise, it is Markov.
- Because of the linearity, $\mathbf{x}_k$ is Gaussian, and the joint distribution of states at different k's are Gaussian.

Kalman Filter (KF) Assumptions

- The state, $\mathbf{x}_k$, evolves according to a known linear **dynamic** equation with:
- known inputs, $\mathbf{u}_k$;
- an additive process noise, $\mathbf{w}_k$, which is a zero-mean white (uncorrelated) process with known covariance $\mathbf{Q}_k$;

$$\mathbf{x}_k^- = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{G}_k \mathbf{u}_k + \mathbf{w}_k$$

- **Measurement** model is a known linear function of the state with:
- an additive measurement noise, $\mathbf{v}_k$, which is a zero-mean white (uncorrelated) process with known covariance $\mathbf{R}_k$;

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k^- + \mathbf{v}_k$$

- Initial state is assumed to be a random variable with known mean (initial estimate) and known covariance (initial uncertainty).
- Initial state and noises are all mutually uncorrelated.
- If the initial state and the noises are Gaussian (i.e., the system is Gauss-Markov), we have some bonus.

Summary of KF Statistical Assumptions

- Initial state $\mathbf{x}_0$ (with possibly given prior information $\mathbf{z}_0$): $\mathbb{E}[\mathbf{x}_0|\mathbf{z}_0] = \hat{\mathbf{x}}_0$ and $\mathrm{Cov}[\mathbf{x}_0|\mathbf{z}_0] = \mathbf{P}_0$
- Process and measurement noise sequences are white with known covariances:
  $\mathbb{E}[\mathbf{w}_k] = \mathbf{0}$, $\mathbb{E}[\mathbf{w}_k\mathbf{w}_j^\mathsf{T}] = \mathbf{Q}_k\delta_{kj}$, and
  $\mathbb{E}[\mathbf{w}_k] = \mathbf{0}$, $\mathbb{E}[\mathbf{v}_k\mathbf{v}_j^\mathsf{T}] = \mathbf{R}_k\delta_{kj}$
- All the above are uncorrelated.

- State and measurement prediction, a.k.a., time update;
- State update, a.k.a., correction or measurement update.

KF Algorithm

Algorithm 4 `Kalman-filter`

Require: belief mean $\boldsymbol{\mu}_{k-1}$, belief covariance $\boldsymbol{\Sigma}_{k-1}$, action $\mathbf{u}_k$, measurement $\mathbf{z}_k$;

$\boldsymbol{\mu}_k^- \leftarrow \mathbf{F}_k \boldsymbol{\mu}_{k-1} + \mathbf{G}_k \mathbf{u}_k$ ▷ predicted mean

$\boldsymbol{\Sigma}_k^- \leftarrow \mathbf{F}_k \boldsymbol{\Sigma}_{k-1} \mathbf{F}_k^\mathsf{T} + \mathbf{Q}_k$ ▷ predicted covariance

$\boldsymbol{\nu}_k \leftarrow \mathbf{z}_k - \mathbf{H}_k \boldsymbol{\mu}_k^-$ ▷ innovation

$\mathbf{S}_k \leftarrow \mathbf{H}_k \boldsymbol{\Sigma}_k^- \mathbf{H}_k^\mathsf{T} + \mathbf{R}_k$ ▷ innovation covariance

$\mathbf{K}_k \leftarrow \boldsymbol{\Sigma}_k^- \mathbf{H}_k^\mathsf{T} \mathbf{S}_k^{-1}$ ▷ filter gain

$\boldsymbol{\mu}_k \leftarrow \boldsymbol{\mu}_k^- + \mathbf{K}_k \boldsymbol{\nu}_k$ ▷ corrected mean

$\boldsymbol{\Sigma}_k \leftarrow (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \boldsymbol{\Sigma}_k^-$ ▷ corrected covariance

$//\boldsymbol{\Sigma}_k \leftarrow (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \boldsymbol{\Sigma}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^\mathsf{T} + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^\mathsf{T}$ ▷ numerically stable form

return $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$

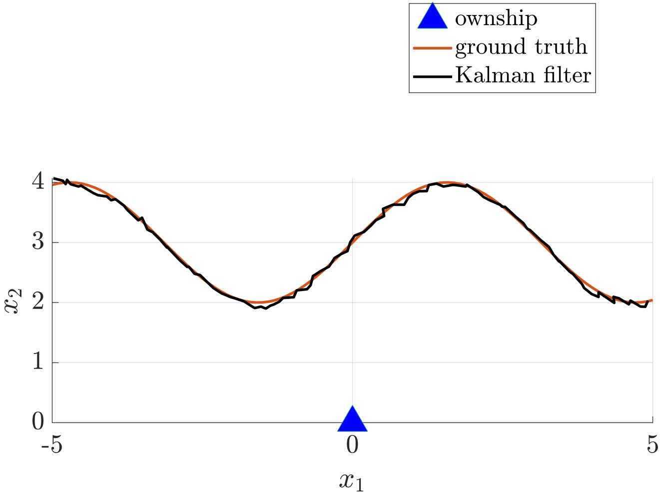
Example: KF Target Tracking

A target is moving in a 2D plane. The ownship position is known and fixed at the origin. We have access to noisy measurements that directly observe the target 2D coordinates at any time step.

$$\mathbf{F}_k = \mathbf{I}_2,\ \mathbf{G}_k = \mathbf{0}_2,\ \mathbf{H}_k = \mathbf{I}_2,\ \mathbf{Q}_k = 0.001\ \mathbf{I}_2,\ \mathbf{R}_k = 0.05^2\ \mathbf{I}_2$$

Example: KF Target Tracking

See `kf_single_target.m` for code.



Overview of Kalman Filter Algorithm

- Under the Gaussian assumption for the initial state (or initial state error) and all the noises entering into the system, the Kalman filter is the optimal MMSE state estimator.

- If these random variables are not Gaussian and one has only their first two moments, then the Kalman filter algorithm is the best linear state estimator (Linear MMSE).

Minimum Mean Square Error (MMSE) Estimation

The MMSE estimation of $\mathbf{x}$ in terms of $\mathbf{z}$ is:

$$\hat{\mathbf{x}}^{\mathrm{MMSE}} = \arg\min_{\hat{\mathbf{x}}} \mathbb{E}[(\hat{\mathbf{x}} - \mathbf{x})^2 | \mathbf{z}]$$

The solution is the conditional (posterior) mean:

$$\hat{\mathbf{x}}^{\mathrm{MMSE}} = \mathbb{E}[\mathbf{x}|\mathbf{z}] = \int \mathbf{x} p(\mathbf{x}|\mathbf{z}) \mathrm{d}\mathbf{x}$$

which can be obtained by

$$\frac{\partial \mathbb{E}[(\hat{\mathbf{x}} - \mathbf{x})^2 | \mathbf{z}]}{\partial \hat{\mathbf{x}}} = \mathbb{E}[2(\hat{\mathbf{x}} - \mathbf{x})|\mathbf{z}] = 2(\hat{\mathbf{x}} - \mathbb{E}[\mathbf{x}|\mathbf{z}]) = 0$$

- $\hat{\mathbf{x}}$: Estimate
- $\mathbf{x}$: True value

Minimum Mean Square Error (MMSE) Estimation

The mean of the posterior gives the MMSE estimation:

$$\hat{\mathbf{x}}^{\text{MMSE}} = \arg\min_{\hat{\mathbf{x}}} \mathbb{E}[(\hat{\mathbf{x}} - \mathbf{x})^2 | \mathbf{z}] = \mathbb{E}[\mathbf{x}|\mathbf{z}] = \int \mathbf{x} p(\mathbf{x}|\mathbf{z}) \mathrm{d}\mathbf{x}$$

Remark

Under the Gaussian assumption, the posterior mean estimated by the Kalman filter is exact, thus yielding the optimal MMSE estimation.

Remark

Under the Gaussian assumption, then MMSE and Maximum a Posteriori estimators coincide since the mode and mean of the Gaussian distribution are the same.

Limitations of Kalman Filter

- Nonlinear motion (process) and measurement models;
- Unknown control inputs or mode changes;
- Data association uncertainty;
- Autocorrelated or crosscorrelated noise sequences.

Readings (Bayes Filter)

- Probabilistic Robotics: Ch. 1 and 2, Understand Example 2.4.2
- State Estimation for Robotics: Ch. 2
- Lecture Notes for Mobile Robotics: Ch. 1

Readings (Kalman Filter)

- Probabilistic Robotics: Ch. 2 and Ch. 3
- State Estimation for Robotics: Ch. 3
- Lecture Notes for Mobile Robotics: Ch. 2 and 6

Next Time

- Nonlinear Kalman Filtering
- Readings:
  - Probabilistic Robotics: Ch. 3
  - State Estimation for Robotics: Ch. 4
  - Lecture Notes for Mobile Robotics: Ch. 2 and 6