Lecture 2: From Linear Regression to Kalman Filter and Beyond

Simo Särkkä

January 18, 2017
1. Batch and Recursive Estimation

2. Towards Bayesian Filtering

3. Kalman Filter and Bayesian Filtering and Smoothing

4. Summary
Consider the linear regression model

\[ y_k = \theta_1 + \theta_2 t_k + \varepsilon_k, \quad k = 1, \ldots, T, \]

with \( \varepsilon_k \sim \mathcal{N}(0, \sigma^2) \) and \( \theta = (\theta_1, \theta_2) \sim \mathcal{N}(m_0, P_0) \).

In probabilistic notation this is:

\[
p(y_k | \theta) = \mathcal{N}(y_k | H_k \theta, \sigma^2)
\]

\[
p(\theta) = \mathcal{N}(\theta | m_0, P_0),
\]

where \( H_k = (1 \ t_k) \).
The Bayesian batch solution by the Bayes’ rule:

\[
p(\theta | y_{1:T}) \propto p(\theta) \prod_{k=1}^{T} p(y_k | \theta) = N(\theta | m_0, P_0) \prod_{k=1}^{T} N(y_k | H_k \theta, \sigma^2).
\]

The posterior is Gaussian

\[
p(\theta | y_{1:T}) = N(\theta | m_T, P_T).
\]

The mean and covariance are given as

\[
\begin{align*}
m_T &= \left[ P_0^{-1} + \frac{1}{\sigma^2} H^T H \right]^{-1} \left[ \frac{1}{\sigma^2} H^T y + P_0^{-1} m_0 \right] \\
P_T &= \left[ P_0^{-1} + \frac{1}{\sigma^2} H^T H \right]^{-1},
\end{align*}
\]

where \( H_k = (1 \ t_k), H = (H_1; H_2; \ldots; H_T), y = (y_1; \ldots; y_T). \)
Assume that we have already computed the posterior distribution, which is conditioned on the measurements up to $k - 1$:

$$p(\theta \mid y_{1:k-1}) = N(\theta \mid \mathbf{m}_{k-1}, \mathbf{P}_{k-1}).$$

Assume that we get the $k$th measurement $y_k$. Using the equations from the previous slide we get

$$p(\theta \mid y_{1:k}) \propto p(y_k \mid \theta) p(\theta \mid y_{1:k-1}) \propto N(\theta \mid \mathbf{m}_k, \mathbf{P}_k).$$

The mean and covariance are given as

$$\mathbf{m}_k = \left[ \mathbf{P}_{k-1}^{-1} + \frac{1}{\sigma^2} \mathbf{H}_k^T \mathbf{H}_k \right]^{-1} \left[ \frac{1}{\sigma^2} \mathbf{H}_k^T \mathbf{y}_k + \mathbf{P}_{k-1}^{-1} \mathbf{m}_{k-1} \right]$$

$$\mathbf{P}_k = \left[ \mathbf{P}_{k-1}^{-1} + \frac{1}{\sigma^2} \mathbf{H}_k^T \mathbf{H}_k \right]^{-1}.$$
By the matrix inversion lemma (or Woodbury identity):

$$P_k = P_{k-1} - P_{k-1}H_k^T \left[ H_k P_{k-1} H_k^T + \sigma^2 \right]^{-1} H_k P_{k-1}. $$

Now the equations for the mean and covariance reduce to

$$S_k = H_k P_{k-1} H_k^T + \sigma^2 $$

$$K_k = P_{k-1} H_k^T S_k^{-1} $$

$$m_k = m_{k-1} + K_k [ y_k - H_k m_{k-1} ]$$

$$P_k = P_{k-1} - K_k S_k K_k^T. $$

Computing these for $k = 0, \ldots, T$ gives exactly the linear regression solution.

A special case of Kalman filter.
Recursive Linear Regression [3/4]
Recursive Linear Regression [3/4]
Recursive Linear Regression [3/4]
Convergence of the recursive solution to the batch solution – on the last step the solutions are exactly equal:
**General batch solution:**

- Specify the **measurement model**:

\[ p(y_{1:T} \mid \theta) = \prod_{k} p(y_{k} \mid \theta). \]

- Specify the **prior distribution** \( p(\theta) \).
- Compute **posterior distribution** by the Bayes’ rule:

\[ p(\theta \mid y_{1:T}) = \frac{1}{Z} p(\theta) \prod_{k} p(y_{k} \mid \theta). \]

- Compute point estimates, moments, predictive quantities etc. from the posterior distribution.
**General recursive solution:**

- Specify the **measurement likelihood** $p(y_k \mid \theta)$.
- Specify the **prior distribution** $p(\theta)$.
- Process measurements $y_1, \ldots, y_T$ one at a time, starting from the prior:

  $$p(\theta \mid y_1) = \frac{1}{Z_1} p(y_1 \mid \theta)p(\theta)$$

  $$p(\theta \mid y_{1:2}) = \frac{1}{Z_2} p(y_2 \mid \theta)p(\theta \mid y_1)$$

  $$p(\theta \mid y_{1:3}) = \frac{1}{Z_3} p(y_3 \mid \theta)p(\theta \mid y_{1:2})$$

  $$\vdots$$

  $$p(\theta \mid y_{1:T}) = \frac{1}{Z_T} p(y_T \mid \theta)p(\theta \mid y_{1:T-1}).$$

- The result at the last step is the **batch solution**.
The recursive solution can be considered as the online learning solution to the Bayesian learning problem.

Batch Bayesian inference is a special case of recursive Bayesian inference.

The parameter can be modeled to change between the measurement steps ⇒ basis of filtering theory.
Let assume **Gaussian random walk** between the measurements in the linear regression model:

\[
p(y_k \mid \theta_k) = \mathcal{N}(y_k \mid H_k \theta_k, \sigma^2) \\
p(\theta_k \mid \theta_{k-1}) = \mathcal{N}(\theta_k \mid \theta_{k-1}, Q) \\
p(\theta_0) = \mathcal{N}(\theta_0 \mid m_0, P_0).
\]

Again, assume that we already know

\[
p(\theta_{k-1} \mid y_1: k-1) = \mathcal{N}(\theta_{k-1} \mid m_{k-1}, P_{k-1}).
\]

The **joint distribution** of \( \theta_k \) and \( \theta_{k-1} \) is (due to Markovianity of dynamics!):

\[
p(\theta_k, \theta_{k-1} \mid y_{1:k-1}) = p(\theta_k \mid \theta_{k-1}) p(\theta_{k-1} \mid y_{1:k-1}).
\]
Integrating over $\theta_{k-1}$ gives:

$$p(\theta_k \mid y_{1:k-1}) = \int p(\theta_k \mid \theta_{k-1}) p(\theta_{k-1} \mid y_{1:k-1}) d\theta_{k-1}. $$

This equation for Markov processes is called the Chapman-Kolmogorov equation.

Because the distributions are Gaussian, the result is Gaussian

$$p(\theta_k \mid y_{1:k-1}) = \mathcal{N}(\theta_k \mid \mathbf{m}_k^{-}, \mathbf{P}_k^{-}),$$

where

$$\mathbf{m}_k^{-} = \mathbf{m}_{k-1} \quad \mathbf{P}_k^{-} = \mathbf{P}_{k-1} + \mathbf{Q}.$$
As in the pure recursive estimation, we get

\[ p(\theta_k \mid y_{1:k}) \propto p(y_k \mid \theta_k) p(\theta_k \mid y_{1:k-1}) \]
\[ \propto N(\theta_k \mid m_k, P_k). \]

After applying the matrix inversion lemma, mean and covariance can be written as

\[ S_k = H_k P_k^{-1} H_k^T + \sigma^2 \]
\[ K_k = P_k^{-1} H_k^T S_k^{-1} \]
\[ m_k = m_k^- + K_k [y_k - H_k m_k^-] \]
\[ P_k = P_k^- - K_k S_k K_k^T. \]

Again, we have derived a special case of the Kalman filter.

The batch version of this solution would be much more complicated.
In the previous slide we formulated the model as

\[ p(\theta_k \mid \theta_{k-1}) = \mathcal{N}(\theta_k \mid \theta_{k-1}, Q) \]
\[ p(y_k \mid \theta_k) = \mathcal{N}(y_k \mid H_k \theta_k, \sigma^2) \]

But in Kalman filtering and control theory the vector of parameters \( \theta_k \) is usually called “state” and denoted as \( x_k \).

More standard state space notation:

\[ p(x_k \mid x_{k-1}) = \mathcal{N}(x_k \mid x_{k-1}, Q) \]
\[ p(y_k \mid x_k) = \mathcal{N}(y_k \mid H_k x_k, \sigma^2) \]

Or equivalently

\[ x_k = x_{k-1} + q_{k-1} \]
\[ y_k = H_k x_k + r_k, \]

where \( q_{k-1} \sim \mathcal{N}(0, Q), r_k \sim \mathcal{N}(0, \sigma^2) \).
The canonical Kalman filtering model is

\[ p(x_k | x_{k-1}) = \mathcal{N}(x_k | A_{k-1} x_{k-1}, Q_{k-1}) \]

\[ p(y_k | x_k) = \mathcal{N}(y_k | H_k x_k, R_k). \]

More often, this model can be seen in the form

\[ x_k = A_{k-1} x_{k-1} + q_{k-1} \]

\[ y_k = H_k x_k + r_k. \]

The Kalman filter actually calculates the following distributions:

\[ p(x_k | y_{1:k-1}) = \mathcal{N}(x_k | m_k^- , P_k^-) \]

\[ p(x_k | y_{1:k}) = \mathcal{N}(x_k | m_k , P_k). \]
Prediction step of the Kalman filter:

\[
\begin{align*}
\mathbf{m}_k^- &= \mathbf{A}_{k-1} \mathbf{m}_{k-1} \\
\mathbf{P}_k^- &= \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}.
\end{align*}
\]

Update step of the Kalman filter:

\[
\begin{align*}
\mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \\
\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1} \\
\mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k [\mathbf{y}_k - \mathbf{H}_k \mathbf{m}_k^-] \\
\mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T.
\end{align*}
\]

These equations will be derived from the general Bayesian filtering equations in the next lecture.
• Generic non-linear state space models

\[ x_k = f(x_{k-1}, q_{k-1}) \]
\[ y_k = h(x_k, r_k). \]

• Generic Markov models

\[ x_k \sim p(x_k \mid x_{k-1}) \]
\[ y_k \sim p(y_k \mid x_k). \]

• Continuous-discrete state space models involving stochastic differential equations:

\[ \frac{dx}{dt} = f(x, t) + w(t) \]
\[ y_k \sim p(y_k \mid x(t_k)). \]
Non-linear state space model with unknown parameters:

\[ x_k = f(x_{k-1}, q_{k-1}, \theta) \]
\[ y_k = h(x_k, r_k, \theta). \]

General Markovian state space model with unknown parameters:

\[ x_k \sim p(x_k \mid x_{k-1}, \theta) \]
\[ y_k \sim p(y_k \mid x_k, \theta). \]

Parameter estimation will be considered later – for now, we will attempt to estimate the state.

Why Bayesian filtering and smoothing then?
In principle, we could just use the (batch) Bayes’ rule
\[
p(x_1, \ldots, x_T \mid y_1, \ldots, y_T) = \frac{p(y_1, \ldots, y_T \mid x_1, \ldots, x_T) p(x_1, \ldots, x_T)}{p(y_1, \ldots, y_T)},
\]

Curse of computational complexity: complexity grows more than linearly with number of measurements (typically we have \( O(T^3) \)).

Hence, we concentrate on the following:

- Filtering distributions:
  \[ p(x_k \mid y_1, \ldots, y_k), \quad k = 1, \ldots, T. \]

- Prediction distributions:
  \[ p(x_{k+n} \mid y_1, \ldots, y_k), \quad k = 1, \ldots, T, \quad n = 1, 2, \ldots, \]

- Smoothing distributions:
  \[ p(x_k \mid y_1, \ldots, y_T), \quad k = 1, \ldots, T. \]
Bayesian Filtering, Prediction and Smoothing (cont.)

Measurements
Estimate

Prediction:
Filtering:
Smoothing:

Measurements
Estimate

Simo Särkkä
Lecture 2: From Linear Regression to Kalman Filter and Beyond
Kalman filter is the classical optimal filter for linear-Gaussian models.

Extended Kalman filter (EKF) is linearization based extension of Kalman filter to non-linear models.

Unscented Kalman filter (UKF) is sigma-point transformation based extension of Kalman filter.

Gauss-Hermite and Cubature Kalman filters (GHKF/CKF) are numerical integration based extensions of Kalman filter.

Particle filter forms a Monte Carlo representation (particle set) to the distribution of the state estimate.

Grid based filters approximate the probability distributions on a finite grid.

Mixture Gaussian approximations are used, for example, in multiple model Kalman filters and Rao-Blackwellized Particle filters.
Rauch-Tung-Striebel (RTS) smoother is the closed form smoother for linear Gaussian models.

Extended, statistically linearized and unscented RTS smoothers are the approximate nonlinear smoothers corresponding to EKF, SLF and UKF.

Gaussian RTS smoothers: cubature RTS smoother, Gauss-Hermite RTS smoothers and various others.

Particle smoothing is based on approximating the smoothing solutions via Monte Carlo.

Rao-Blackwellized particle smoother is a combination of particle smoothing and RTS smoothing.
- Linear regression problem can be solved as batch problem or recursively – the latter solution is a special case of Kalman filter.
- A generic Bayesian estimation problem can also be solved as batch problem or recursively.
- If we let the linear regression parameter change between the measurements, we get a simple linear state space model – again solvable with Kalman filtering model.
- By generalizing this idea and the solution we get the Kalman filter algorithm.
- By further generalizing to non-Gaussian models results in generic probabilistic state space models.
- Bayesian filtering and smoothing methods solve Bayesian inference problems on state space models recursively.
[Linear regression with Kalman filter]