Lecture 2: From Linear Regression to Kalman Filter and Beyond

Simo Särkkä

Department of Biomedical Engineering and Computational Science
Aalto University

February 3, 2011
Contents

1. Batch and Recursive Estimation

2. Towards Bayesian Filtering

3. Kalman Filter and General Bayesian Optimal Filter

4. Summary and Demo
Consider the linear regression model

\[ y_k = a_1 + a_2 t_k + \epsilon_k, \]

with \( \epsilon_k \sim \mathcal{N}(0, \sigma^2) \) and \( a = (a_1, a_2) \sim \mathcal{N}(m_0, P_0) \).

In probabilistic notation this is:

\[
\begin{align*}
    p(y_k | a) &= \mathcal{N}(y_k | H_k a, \sigma^2) \\
    p(a) &= \mathcal{N}(a | m_0, P_0),
\end{align*}
\]

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

\[ p(a \mid y_{1:N}) \propto p(a) \prod_{k=1}^{N} p(y_k \mid a) \]

\[ = N(a \mid m_0, P_0) \prod_{k=1}^{N} N(y_k \mid H_k a, \sigma^2). \]

The posterior is Gaussian

\[ p(a \mid y_{1:N}) = N(a \mid m_N, P_N). \]

The mean and covariance are given as

\[ m_N = \left[ P_0^{-1} + \frac{1}{\sigma^2} H^T H \right]^{-1} \left[ \frac{1}{\sigma^2} H^T y + P_0^{-1} m_0 \right] \]

\[ P_N = \left[ P_0^{-1} + \frac{1}{\sigma^2} H^T H \right]^{-1}, \]

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

$$p(a \mid y_{1:k-1}) = N(a \mid m_{k-1}, P_{k-1}).$$

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

$$p(a \mid y_{1:k}) \propto p(y_k \mid a) p(a \mid y_{1:k-1}) \propto N(a \mid m_k, P_k).$$

The mean and covariance are given as

$$m_k = \left[ P_{k-1}^{-1} + \frac{1}{\sigma^2} H_k^T H_k \right]^{-1} \left[ \frac{1}{\sigma^2} H_k^T y_k + P_{k-1}^{-1} m_{k-1} \right]$$

$$P_k = \left[ P_{k-1}^{-1} + \frac{1}{\sigma^2} H_k^T H_k \right]^{-1}.$$
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 \( k = 0, \ldots, N \) gives exactly the linear regression solution – but without a matrix inversion\(^1\)!

\(^1\)Without an explicit matrix inversion.

A special case of Kalman filter.
Convergence of the recursive solution to the batch solution – on the last step the solutions are exactly equal:

![Graph showing convergence](image-url)
**General batch solution:**

- Specify the **measurement model:**

$$p(y_{1:N} | \theta) = \prod_{k} p(y_k | \theta).$$

- Specify the **prior distribution** $p(\theta)$.

- Compute **posterior distribution** by the Bayes’ rule:

$$p(\theta | y_{1:N}) = \frac{1}{Z} p(\theta) \prod_{k} p(y_k | \theta).$$

- 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 $y_1, \ldots, y_N$ one at a time, starting from the prior:

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

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

$$\vdots$$

$$p(\theta \mid y_{1:N}) = \frac{1}{Z_N} p(y_N \mid \theta)p(\theta \mid y_{1:N-1}).$$

- The posterior at the last step is the same as the batch solution.
Advantages of Recursive 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 $\Rightarrow$ basis of filtering theory.
Let assume \textbf{Gaussian random walk} between the measurements in the linear regression model:

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

\[
p(a_k | a_{k-1}) = \mathcal{N}(a_k | a_{k-1}, Q)
\]

\[
p(a_0) = \mathcal{N}(a_0 | m_0, P_0).
\]

Again, assume that we already know

\[
p(a_{k-1} | y:1:k-1) = \mathcal{N}(a_{k-1} | m_{k-1}, P_{k-1}).
\]

The \textbf{joint distribution} of \(a_k\) and \(a_{k-1}\) is (due to Markovianity of dynamics!):

\[
p(a_k, a_{k-1} | y:1:k-1) = p(a_k | a_{k-1}) p(a_{k-1} | y:1:k-1).
\]
Integrating over $a_{k-1}$ gives:

$$p(a_k | y_{1:k-1}) = \int p(a_k | a_{k-1}) p(a_{k-1} | y_{1:k-1}) da_{k-1}.$$ 

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

Because the distributions are Gaussian, the result is Gaussian

$$p(a_k | y_{1:k-1}) = N(a_k | m_k^{-}, P_k^{-}),$$

where

$$m_k^{-} = m_{k-1}$$

$$P_k^{-} = P_{k-1} + Q.$$
As in the pure recursive estimation, we get

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

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.
State Space Notation

- In the previous section we formulated the model as
  \[ p(a_k \mid a_{k-1}) = \mathcal{N}(a_k \mid a_{k-1}, Q) \]
  \[ p(y_k \mid a_k) = \mathcal{N}(y_k \mid H_k a_k, \sigma^2) \]

- But in Kalman filtering and control theory the vector of parameters \( a_k \) 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
  \[ x_k = x_{k-1} + q \]
  \[ y_k = H_k x_k + r, \]

  where \( q \sim \mathcal{N}(0, Q), \ r \sim \mathcal{N}(0, \sigma^2). \)
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:

\[
m_k^- = A_{k-1} \ m_{k-1} \\
P_k^- = A_{k-1} \ P_{k-1} \ A_{k-1}^T + Q_{k-1}.
\]

Update step of the Kalman filter:

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

These equations will be derived from the general Bayesian filtering equations in the next lecture.
Generic discrete-time state space models

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

Generic Markov models

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

Approximation methods: Extended Kalman filters (EKF), Unscented Kalman filters (UKF), sequential Monte Carlo (SMC) filters a’ka particle filters.
In continuous-discrete filtering models, dynamics are modeled in continuous time, measurements at discrete time steps.

The continuous time versions of Markov models are called as **stochastic differential equations**:

\[
\frac{dx}{dt} = f(x, t) + w(t)
\]

where \(w(t)\) is a continuous time Gaussian white noise process.

Approximation methods: Extended Kalman filters, Unscented Kalman filters, sequential Monte Carlo, particle filters.
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 a generic probabilistic state space model.
Demonstration

Batch and recursive linear regression.