Lecture 2: From Linear Regression to Kalman Filter and Beyond

Simo Särkkä

January 16, 2019

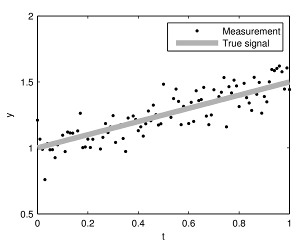
Learning Outcomes

1. Summary of the Last Lecture
2. Batch and Recursive Estimation
3. Towards Bayesian Filtering
4. Kalman Filter and Bayesian Filtering and Smoothing
5. Summary

Summary of the Last Lecture

- The purpose of is to estimate the state of a time-varying system from noisy measurements obtained from it.
- The linear theory dates back to 50's, non-linear Bayesian theory was founded in 60's.
- The efficient computational solutions can be divided into prediction, filtering and smoothing.
- Applications: tracking, navigation, telecommunications, audio processing, control systems, etc.
- The formal Bayesian estimation equations can be approximated by e.g. Gaussian approximations, Monte Carlo or Gaussian mixtures.
- Formulating physical systems as state space models is a challenging engineering topic as such.

Batch Linear Regression [1/2]



- Consider the linear regression model

$$y_k = \theta_1 + \theta_2\, t_k + \varepsilon_k, \qquad k = 1, \ldots, T,$$

  with $\varepsilon_k \sim \mathrm{N}(0, \sigma^2)$ and $\boldsymbol{\theta} = (\theta_1, \theta_2) \sim \mathrm{N}(\mathbf{m}_0, \mathbf{P}_0)$.
- In probabilistic notation this is:

$$\begin{aligned} p(y_k \mid \boldsymbol{\theta}) &= \mathrm{N}(y_k \mid \mathbf{H}_k\, \boldsymbol{\theta}, \sigma^2) \\ p(\boldsymbol{\theta}) &= \mathrm{N}(\boldsymbol{\theta} \mid \mathbf{m}_0, \mathbf{P}_0), \end{aligned}$$

  where $\mathbf{H}_k = (1 \;\; t_k)$.

Batch Linear Regression [2/2]

- The Bayesian batch solution by the Bayes' rule:

$$
\begin{aligned}
p(\boldsymbol{\theta} \mid y_{1:T}) &\propto p(\boldsymbol{\theta}) \textstyle\prod_{k=1}^{T} p(y_k \mid \boldsymbol{\theta}) \\
&= \mathrm{N}(\boldsymbol{\theta} \mid \mathbf{m}_0, \mathbf{P}_0) \textstyle\prod_{k=1}^{T} \mathrm{N}(y_k \mid \mathbf{H}_k \, \boldsymbol{\theta}, \sigma^2).
\end{aligned}
$$

- The posterior is Gaussian

$$
p(\boldsymbol{\theta} \mid y_{1:T}) = \mathrm{N}(\boldsymbol{\theta} \mid \mathbf{m}_T, \mathbf{P}_T).
$$

- The mean and covariance are given as

$$
\begin{aligned}
\mathbf{m}_T &= \left[\mathbf{P}_0^{-1} + \frac{1}{\sigma^2} \mathbf{H}^\mathsf{T} \mathbf{H}\right]^{-1} \left[\frac{1}{\sigma^2} \mathbf{H}^\mathsf{T} \mathbf{y} + \mathbf{P}_0^{-1} \mathbf{m}_0\right] \\
\mathbf{P}_T &= \left[\mathbf{P}_0^{-1} + \frac{1}{\sigma^2} \mathbf{H}^\mathsf{T} \mathbf{H}\right]^{-1},
\end{aligned}
$$

where $\mathbf{H}_k = (1 \;\; t_k)$, $\mathbf{H} = (\mathbf{H}_1; \mathbf{H}_2; \ldots; \mathbf{H}_T)$, $\mathbf{y} = (y_1; \ldots; y_T)$.

Recursive Linear Regression [1/4]

- Assume that we have already computed the posterior distribution, which is conditioned on the measurements up to $k-1$:

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

- Assume that we get the kth measurement y_k. Using the equations from the previous slide we get

$$\begin{aligned} p(\boldsymbol{\theta} \mid y_{1:k}) &\propto p(y_k \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid y_{1:k-1}) \\ &\propto \mathrm{N}(\boldsymbol{\theta} \mid \mathbf{m}_k, \mathbf{P}_k). \end{aligned}$$

- The mean and covariance are given as

$$\begin{aligned} \mathbf{m}_k &= \left[\mathbf{P}_{k-1}^{-1} + \frac{1}{\sigma^2}\mathbf{H}_k^\mathsf{T}\mathbf{H}_k\right]^{-1} \left[\frac{1}{\sigma^2}\mathbf{H}_k^\mathsf{T} y_k + \mathbf{P}_{k-1}^{-1}\mathbf{m}_{k-1}\right] \\ \mathbf{P}_k &= \left[\mathbf{P}_{k-1}^{-1} + \frac{1}{\sigma^2}\mathbf{H}_k^\mathsf{T}\mathbf{H}_k\right]^{-1}. \end{aligned}$$

Recursive Linear Regression [2/4]

- By the matrix inversion lemma (or Woodbury identity):

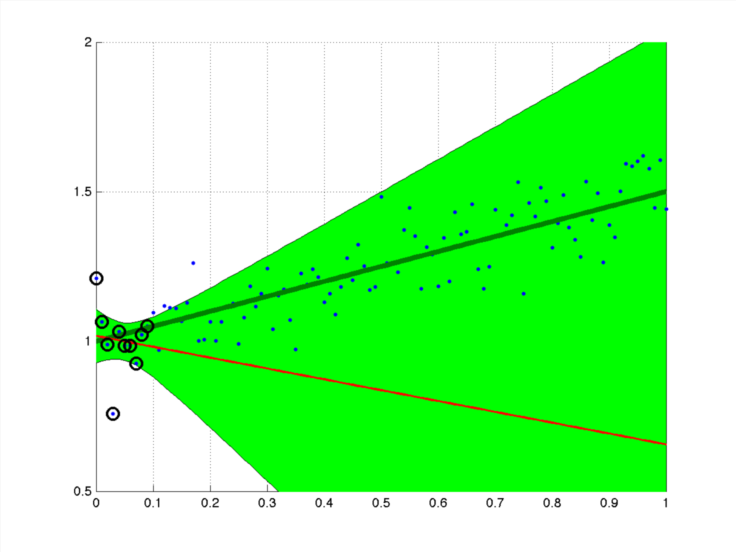
$$\mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{P}_{k-1}\mathbf{H}_k^{\mathsf{T}}\left[\mathbf{H}_k\mathbf{P}_{k-1}\mathbf{H}_k^{\mathsf{T}} + \sigma^2\right]^{-1}\mathbf{H}_k\mathbf{P}_{k-1}.$$

- Now the equations for the mean and covariance reduce to

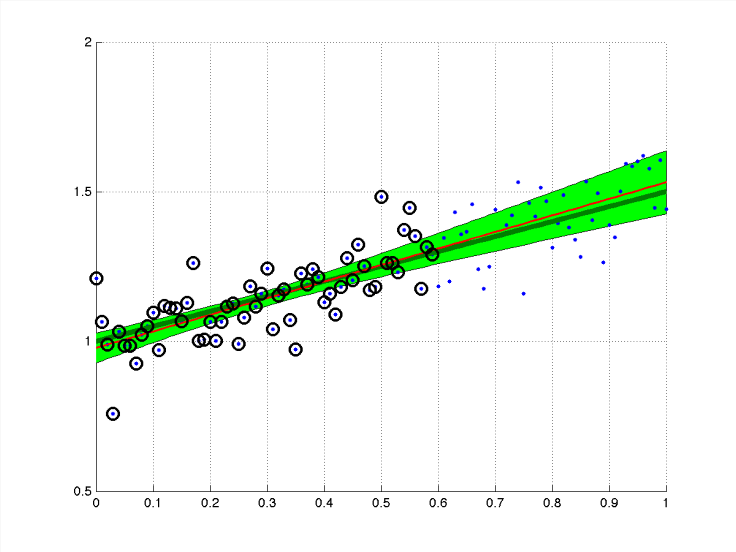
$$\begin{aligned}
S_k &= \mathbf{H}_k\mathbf{P}_{k-1}\mathbf{H}_k^{\mathsf{T}} + \sigma^2 \\
\mathbf{K}_k &= \mathbf{P}_{k-1}\mathbf{H}_k^{\mathsf{T}}S_k^{-1} \\
\mathbf{m}_k &= \mathbf{m}_{k-1} + \mathbf{K}_k[y_k - \mathbf{H}_k\mathbf{m}_{k-1}] \\
\mathbf{P}_k &= \mathbf{P}_{k-1} - \mathbf{K}_k S_k \mathbf{K}_k^{\mathsf{T}}.
\end{aligned}$$

- Computing these for $k = 0, \ldots, T$ gives exactly the linear regression solution.
- A special case of Kalman filter.

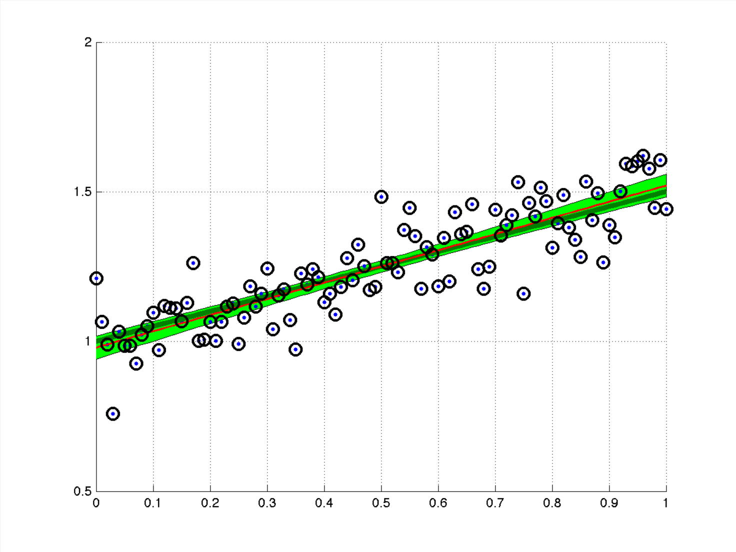
Recursive Linear Regression [3/4]



Recursive Linear Regression [3/4]

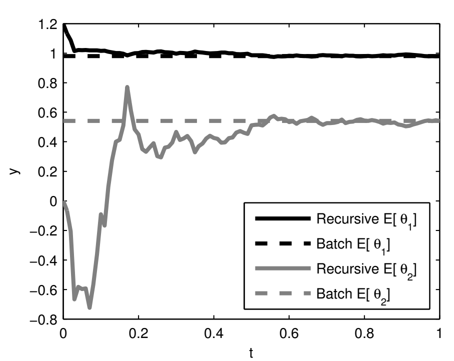


Recursive Linear Regression [3/4]



Recursive Linear Regression [4/4]

Convergence of the recursive solution to the batch solution – on the last step the solutions are exactly equal:



Batch vs. Recursive Estimation [1/2]

General batch solution:

- Specify the measurement model:

$$p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}) = \prod_k p(\mathbf{y}_k \mid \boldsymbol{\theta}).$$

- Specify the prior distribution $p(\boldsymbol{\theta})$.
- Compute posterior distribution by the Bayes' rule:

$$p(\boldsymbol{\theta} \mid \mathbf{y}_{1:T}) = \frac{1}{Z} p(\boldsymbol{\theta}) \prod_k p(\mathbf{y}_k \mid \boldsymbol{\theta}).$$

- Compute point estimates, moments, predictive quantities etc. from the posterior distribution.

Batch vs. Recursive Estimation [2/2]

General recursive solution:

- Specify the measurement likelihood $p(\mathbf{y}_k \mid \theta)$.
- Specify the prior distribution $p(\theta)$.
- Process measurements $\mathbf{y}_1, \ldots, \mathbf{y}_T$ one at a time, starting from the prior:

$$\begin{aligned}
p(\theta \mid \mathbf{y}_1) &= \frac{1}{Z_1} p(\mathbf{y}_1 \mid \theta) p(\theta) \\
p(\theta \mid \mathbf{y}_{1:2}) &= \frac{1}{Z_2} p(\mathbf{y}_2 \mid \theta) p(\theta \mid \mathbf{y}_1) \\
p(\theta \mid \mathbf{y}_{1:3}) &= \frac{1}{Z_3} p(\mathbf{y}_3 \mid \theta) p(\theta \mid \mathbf{y}_{1:2}) \\
&\vdots \\
p(\theta \mid \mathbf{y}_{1:T}) &= \frac{1}{Z_T} p(\mathbf{y}_T \mid \theta) p(\theta \mid \mathbf{y}_{1:T-1}).
\end{aligned}$$

- The result at the last step is 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.

Drift Model for Linear Regression [1/3]

- Let assume Gaussian random walk between the measurements in the linear regression model:

$$
\begin{aligned}
p(y_k \mid \boldsymbol{\theta}_k) &= \mathrm{N}(y_k \mid \mathbf{H}_k \, \boldsymbol{\theta}_k, \sigma^2) \\
p(\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_{k-1}) &= \mathrm{N}(\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_{k-1}, \mathbf{Q}) \\
p(\boldsymbol{\theta}_0) &= \mathrm{N}(\boldsymbol{\theta}_0 \mid \mathbf{m}_0, \mathbf{P}_0).
\end{aligned}
$$

- Again, assume that we already know

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

- The joint distribution of $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_{k-1}$ is (due to Markovianity of dynamics!):

$$
p(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{k-1} \mid y_{1:k-1}) = p(\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_{k-1}) \, p(\boldsymbol{\theta}_{k-1} \mid y_{1:k-1}).
$$

Drift Model for Linear Regression [2/3]

- Integrating over θ_{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

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

where

$$\begin{aligned}
\mathbf{m}_k^- &= \mathbf{m}_{k-1} \\
\mathbf{P}_k^- &= \mathbf{P}_{k-1} + \mathbf{Q}.
\end{aligned}$$

Drift Model for Linear Regression [3/3]

- As in the pure recursive estimation, we get

$$\begin{aligned} p(\boldsymbol{\theta}_k \mid y_{1:k}) &\propto p(y_k \mid \boldsymbol{\theta}_k)\, p(\boldsymbol{\theta}_k \mid y_{1:k-1}) \\ &\propto \mathrm{N}(\boldsymbol{\theta}_k \mid \mathbf{m}_k, \mathbf{P}_k). \end{aligned}$$

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

$$\begin{aligned} S_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\mathsf{T} + \sigma^2 \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^\mathsf{T} S_k^{-1} \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k [y_k - \mathbf{H}_k \mathbf{m}_k^-] \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k S_k \mathbf{K}_k^\mathsf{T}. \end{aligned}$$

- 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 slide we formulated the model as

$$
\begin{aligned}
p(\boldsymbol{\theta}_k \,|\, \boldsymbol{\theta}_{k-1}) &= \mathrm{N}(\boldsymbol{\theta}_k \,|\, \boldsymbol{\theta}_{k-1}, \mathbf{Q}) \\
p(y_k \,|\, \boldsymbol{\theta}_k) &= \mathrm{N}(y_k \,|\, \mathbf{H}_k \, \boldsymbol{\theta}_k, \sigma^2)
\end{aligned}
$$

- But in Kalman filtering and control theory the vector of parameters $\boldsymbol{\theta}_k$ is usually called "state" and denoted as $\mathbf{x}_k$.
- More standard state space notation:

$$
\begin{aligned}
p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) &= \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \mathbf{Q}) \\
p(y_k \,|\, \mathbf{x}_k) &= \mathrm{N}(y_k \,|\, \mathbf{H}_k \, \mathbf{x}_k, \sigma^2)
\end{aligned}
$$

- Or equivalently

$$
\begin{aligned}
\mathbf{x}_k &= \mathbf{x}_{k-1} + \mathbf{q}_{k-1} \\
y_k &= \mathbf{H}_k \, \mathbf{x}_k + r_k,
\end{aligned}
$$

where $\mathbf{q}_{k-1} \sim \mathrm{N}(\mathbf{0}, \mathbf{Q})$, $r_k \sim \mathrm{N}(0, \sigma^2)$.

Kalman Filter [1/2]

- The canonical Kalman filtering model is

$$
\begin{aligned}
p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) &= \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{A}_{k-1}\, \mathbf{x}_{k-1}, \mathbf{Q}_{k-1}) \\
p(\mathbf{y}_k \,|\, \mathbf{x}_k) &= \mathrm{N}(\mathbf{y}_k \,|\, \mathbf{H}_k\, \mathbf{x}_k, \mathbf{R}_k).
\end{aligned}
$$

- More often, this model can be seen in the form

$$
\begin{aligned}
\mathbf{x}_k &= \mathbf{A}_{k-1}\, \mathbf{x}_{k-1} + \mathbf{q}_{k-1} \\
\mathbf{y}_k &= \mathbf{H}_k\, \mathbf{x}_k + \mathbf{r}_k.
\end{aligned}
$$

- The Kalman filter actually calculates the following distributions:

$$
\begin{aligned}
p(\mathbf{x}_k \,|\, \mathbf{y}_{1:k-1}) &= \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k^-, \mathbf{P}_k^-) \\
p(\mathbf{x}_k \,|\, \mathbf{y}_{1:k}) &= \mathrm{N}(\mathbf{x}_k \,|\, \mathbf{m}_k, \mathbf{P}_k).
\end{aligned}
$$

Kalman Filter [2/2]

- Prediction step of the Kalman filter:

$$
\begin{aligned}
\mathbf{m}_k^- &= \mathbf{A}_{k-1}\,\mathbf{m}_{k-1} \\
\mathbf{P}_k^- &= \mathbf{A}_{k-1}\,\mathbf{P}_{k-1}\,\mathbf{A}_{k-1}^\mathsf{T} + \mathbf{Q}_{k-1}.
\end{aligned}
$$

- Update step of the Kalman filter:

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

- These equations will be derived from the general Bayesian filtering equations in the next lecture.

Probabilistic State Space Models [1/2]

- Generic non-linear state space models

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{q}_{k-1}) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{r}_k). \end{aligned}$$

- Generic Markov models

$$\begin{aligned} \mathbf{x}_k &\sim p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) \\ \mathbf{y}_k &\sim p(\mathbf{y}_k \,|\, \mathbf{x}_k). \end{aligned}$$

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

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{f}(\mathbf{x}, t) + \mathbf{w}(t) \\ \mathbf{y}_k &\sim p(\mathbf{y}_k \,|\, \mathbf{x}(t_k)). \end{aligned}$$

Probabilistic State Space Models [2/2]

- Non-linear state space model with unknown parameters:

$$
\begin{aligned}
\mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{q}_{k-1}, \boldsymbol{\theta}) \\
\mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{r}_k, \boldsymbol{\theta}).
\end{aligned}
$$

- General Markovian state space model with unknown parameters:

$$
\begin{aligned}
\mathbf{x}_k &\sim p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \boldsymbol{\theta}) \\
\mathbf{y}_k &\sim p(\mathbf{y}_k \,|\, \mathbf{x}_k, \boldsymbol{\theta}).
\end{aligned}
$$

- Parameter estimation will be considered later – for now, we will attempt to estimate the state.
- Why Bayesian filtering and smoothing then?

Bayesian Filtering, Prediction and Smoothing

- In principle, we could just use the (batch) Bayes' rule

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_T \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \frac{p(\mathbf{y}_1, \ldots, \mathbf{y}_T \mid \mathbf{x}_1, \ldots, \mathbf{x}_T)\, p(\mathbf{x}_1, \ldots, \mathbf{x}_T)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)},$$

- 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:

$$p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_k), \qquad k = 1, \ldots, T.$$

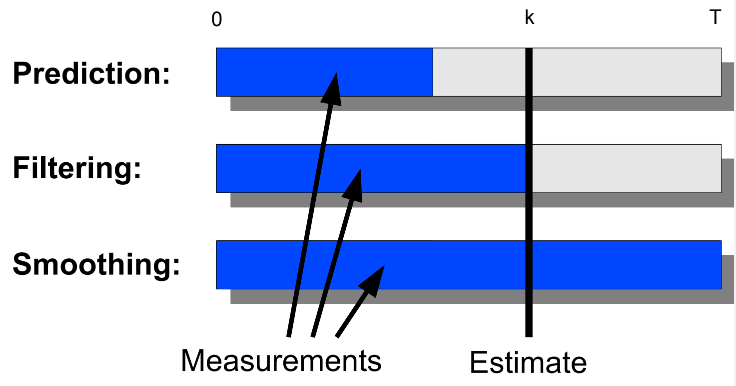
  - Prediction distributions:

$$p(\mathbf{x}_{k+n} \mid \mathbf{y}_1, \ldots, \mathbf{y}_k), \qquad k = 1, \ldots, T, \quad n = 1, 2, \ldots,$$

  - Smoothing distributions:

$$p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_T), \qquad k = 1, \ldots, T.$$

Bayesian Filtering, Prediction and Smoothing (cont.)



Filtering Algorithms

- 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.

Smoothing Algorithms

- 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.

Summary

- 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.

Demonstration

[Linear regression with Kalman filter]