Bayesian Filtering

Václav Šmídl

April 10, 2018

Least Squares Revisited

Model of linear regression with unknown parameters $\boldsymbol{x}$:

$$\boldsymbol{y} = C\boldsymbol{x} + \boldsymbol{e},$$

equals

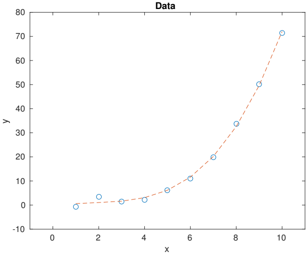
$$y_i = \boldsymbol{c}_i^T \boldsymbol{x} + e_i,$$

with sum of squares

$$\sigma = \sum_{i=1}^{n} e_i = \boldsymbol{e}^T \boldsymbol{e}.$$

In polynomial regression

$$\begin{aligned} y &= C\boldsymbol{x} \\ &= [1, x, x^2][a, b, c] \end{aligned}$$



Least Squares Revisited

Model of linear regression with unknown parameters $\boldsymbol{x}$:

$$\boldsymbol{y} = C\boldsymbol{x} + \boldsymbol{e},$$

equals

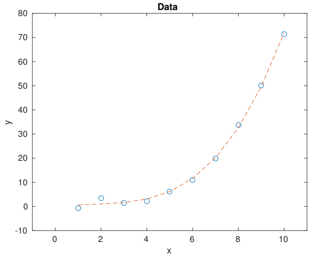
$$y_i = \boldsymbol{c}_i^T \boldsymbol{x} + e_i,$$

with sum of squares

$$\sigma = \sum_{i=1}^{n} e_i = \boldsymbol{e}^T \boldsymbol{e}.$$

In polynomial regression

$$\begin{aligned} y &= C\boldsymbol{x} \\ &= [1, x, x^2][a, b, c] \end{aligned}$$



Least Squares Revisited

Model of linear regression with unknown parameters $\boldsymbol{x}$:

$$\boldsymbol{y} = C\boldsymbol{x} + \boldsymbol{e},$$

equals

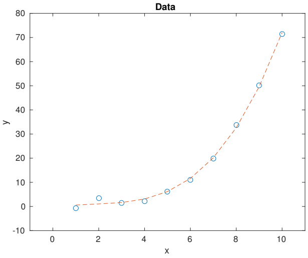
$$y_i = \boldsymbol{c}_i^T \boldsymbol{x} + e_i,$$

with sum of squares

$$\sigma = \sum_{i=1}^{n} e_i = \boldsymbol{e}^T \boldsymbol{e}.$$

In polynomial regression

$$\begin{aligned} y &= C\boldsymbol{x} \\ &= [1, x, x^2][a, b, c] \end{aligned}$$



Solution:

$$\hat{\boldsymbol{x}} = (C^T C) C^T \boldsymbol{y}.$$

Sufficient statistics

Model:

$$y_i = c_i^T \boldsymbol{x} + e_i,$$

For $i = 1 : n$

$$\hat{\boldsymbol{x}} = \left(\sum_{i=1}^{n} \boldsymbol{c}_i \boldsymbol{c}_i \right)^{-1} \sum_{i=1}^{n} \boldsymbol{c}_i y_i.$$

Sufficient statistics

Model:

$$y_i = c_i^T \boldsymbol{x} + e_i,$$

For $i = 1 : n$

$$\hat{\boldsymbol{x}} = \left(\sum_{i=1}^{n} \boldsymbol{c}_i \boldsymbol{c}_i \right)^{-1} \sum_{i=1}^{n} \boldsymbol{c}_i y_i.$$

Observing $i = n + 1$:

Sufficient statistics

Model:

$$y_i = c_i^T \boldsymbol{x} + e_i,$$

For $i = 1 : n$

$$\hat{\boldsymbol{x}} = \left(\sum_{i=1}^{n} \boldsymbol{c}_i \boldsymbol{c}_i \right)^{-1} \sum_{i=1}^{n} \boldsymbol{c}_i y_i.$$

Observing $i = n + 1$:

$$\hat{\boldsymbol{x}} = \left(\sum_{i=1}^{n} \boldsymbol{c}_i \boldsymbol{c}_i + \boldsymbol{c}_{n+1} \boldsymbol{c}_{n+1}^T \right)^{-1} \left(\sum_{i=1}^{n} \boldsymbol{c}_i y_i + \boldsymbol{c}_{n+1} y_{n+1} \right).$$

The notion of sufficient statistics:

$$\hat{\boldsymbol{x}} = V^{-1} v,$$

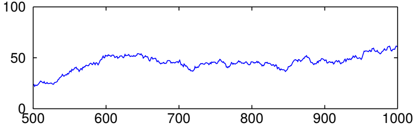
$$V = \sum_{i=1}^{n} \boldsymbol{c}_i \boldsymbol{c}_i,$$

$$v = \sum_{i=1}^{n} \boldsymbol{c}_i y_i,$$

Recursive Least Squares

Least squares :

$$V = \sum_{i=1}^{n} \boldsymbol{c}_i \boldsymbol{c}_i,$$
$$v = \sum_{i=1}^{n} \boldsymbol{c}_i y_i,$$



Recursive Least Squares

Least squares :

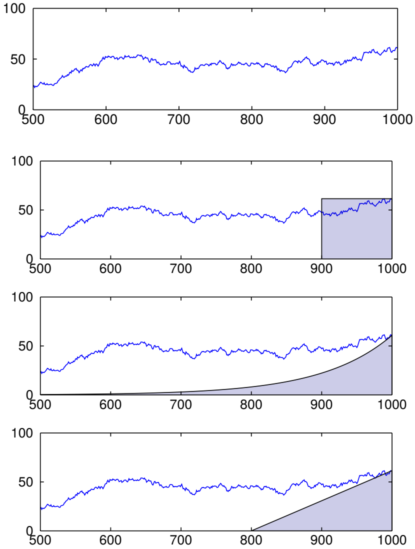
$$V = \sum_{i=1}^{n} \boldsymbol{c}_i \boldsymbol{c}_i,$$
$$v = \sum_{i=1}^{n} \boldsymbol{c}_i y_i,$$

Weighted least squares :

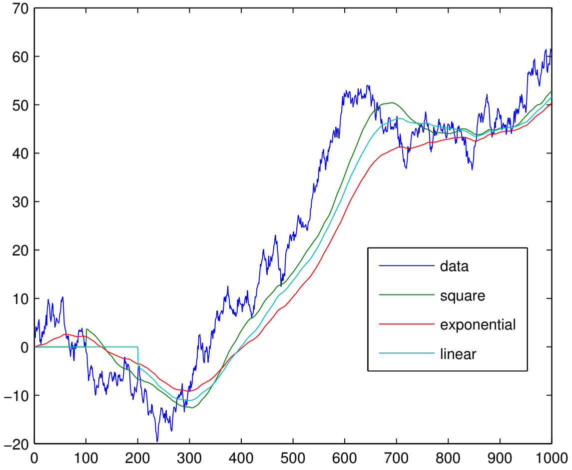
$$V = \sum_{i=1}^{n} w_i \boldsymbol{c}_i \boldsymbol{c}_i,$$
$$v = \sum_{i=1}^{n} w_i \boldsymbol{c}_i y_i,$$

Weight profile - window:

- square window
- exponential window
- linear window



Moving average - comparison



Recursive computation of moving windows of length L

Recursive least squares with scalar x:

$$\sum_{i=1}^{L} \boldsymbol{c}_i \boldsymbol{c}_i^T = \boldsymbol{c}_1 \boldsymbol{c}_1^T + \boldsymbol{c}_2 \boldsymbol{c}_2^T + \ldots + \boldsymbol{c}_L \boldsymbol{c}_L^T \qquad = \sum_{i=1}^{L-1} \boldsymbol{c}_i \boldsymbol{c}_i^T + \boldsymbol{c}_L \boldsymbol{c}_L^T$$

Recursive computation of moving windows of length L

Recursive least squares with scalar x:

$$\sum_{i=1}^{L} \boldsymbol{c}_i \boldsymbol{c}_i^T = \boldsymbol{c}_1 \boldsymbol{c}_1^T + \boldsymbol{c}_2 \boldsymbol{c}_2^T + \ldots + \boldsymbol{c}_L \boldsymbol{c}_L^T \qquad = \sum_{i=1}^{L-1} \boldsymbol{c}_i \boldsymbol{c}_i^T + \boldsymbol{c}_L \boldsymbol{c}_L^T$$

Square window:

$$\sum_{i=n-L}^{n} \boldsymbol{c}_i \boldsymbol{c}_i^T = \boldsymbol{c}_{n-L} \boldsymbol{c}_{n-L}^T + \ldots \quad + \boldsymbol{c}_n \boldsymbol{c}_n^T = \underbrace{\sum \boldsymbol{c}_i \boldsymbol{c}_i^T}_{\textit{last results}} + \boldsymbol{c}_n \boldsymbol{c}_n^T - \boldsymbol{c}_{n-L} \boldsymbol{c}_{n-L}^T,$$

Recursive computation of moving windows of length L

Recursive least squares with scalar x:

$$\sum_{i=1}^{L} \boldsymbol{c}_i \boldsymbol{c}_i^T = \boldsymbol{c}_1 \boldsymbol{c}_1^T + \boldsymbol{c}_2 \boldsymbol{c}_2^T + \ldots + \boldsymbol{c}_L \boldsymbol{c}_L^T \qquad = \sum_{i=1}^{L-1} \boldsymbol{c}_i \boldsymbol{c}_i^T + \boldsymbol{c}_L \boldsymbol{c}_L^T$$

Square window:

$$\sum_{i=n-L}^{n} \boldsymbol{c}_i \boldsymbol{c}_i^T = \boldsymbol{c}_{n-L} \boldsymbol{c}_{n-L}^T + \ldots \quad + \boldsymbol{c}_n \boldsymbol{c}_n^T = \underbrace{\sum \boldsymbol{c}_i \boldsymbol{c}_i^T}_{\textit{last results}} + \boldsymbol{c}_n \boldsymbol{c}_n^T - \boldsymbol{c}_{n-L} \boldsymbol{c}_{n-L}^T,$$

Exponential window

$$\sum_{i=1}^{n} \phi^{n-i} \boldsymbol{c}_i \boldsymbol{c}_i^T = \phi^n \boldsymbol{c}_1 \boldsymbol{c}_1^T + \ldots \quad + \boldsymbol{c}_n \boldsymbol{c}_n^T = \phi \underbrace{\sum \phi^{n-i-1} \boldsymbol{c}_i \boldsymbol{c}_i^T}_{\textit{last}} + \boldsymbol{c}_n \boldsymbol{c}_n^T,$$

How to choose window shape and length?

Recursive least squares RLS

If we already collected N measurements, incorporation the $N+1$ data record is

$$\hat{\boldsymbol{x}}_{n+1} = U_{n+1} v_{n+1},$$

$$U_{n+1} = \left(\sum_{i=1}^{n} \phi_i^{n-i} \boldsymbol{c}_i \boldsymbol{c}_i + \boldsymbol{c}_{n+1} \boldsymbol{c}_{n+1}^T \right)^{-1}$$

Recursive least squares RLS

If we already collected N measurements, incorporation the $N+1$ data record is

$$\hat{\boldsymbol{x}}_{n+1} = U_{n+1} v_{n+1},$$
$$U_{n+1} = \left(\sum_{i=1}^{n} \phi_i^{n-i} \boldsymbol{c}_i \boldsymbol{c}_i + \boldsymbol{c}_{n+1} \boldsymbol{c}_{n+1}^T \right)^{-1}$$

No need for inversion due to Matrix inversion lemma:

$$U_{n+1} = \left(\phi U_n + x_{n+1} x_{n+1}^T \right)^{-1}$$
$$= \phi^{-1} U_n - \frac{U_n x_{n+1} x_{n+1}^T U_n}{\phi^2 (1 + x_{n+1} \phi^{-1} U_n x_{n+1})}.$$

Fast and numerically stable (square root form) algorithm.

Probabilistic model of RLS

Bayesian formulation of least squares

$$
\begin{aligned}
&p(y_i|\boldsymbol{c}_i, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{c}_i \boldsymbol{x}, r), \\
&p(\boldsymbol{x}|X, Y) \propto \mathcal{N}(\boldsymbol{c}_1^T \boldsymbol{x}, r)\mathcal{N}(\boldsymbol{c}_2^T \boldsymbol{x}, r) \ldots \mathcal{N}(\boldsymbol{c}_3^T \boldsymbol{x}, r),
\end{aligned}
$$

Probabilistic model of RLS

Bayesian formulation of least squares

$$p(y_i|\boldsymbol{c}_i, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{c}_i\boldsymbol{x}, r),$$
$$p(\boldsymbol{x}|X, Y) \propto \mathcal{N}(\boldsymbol{c}_1^T\boldsymbol{x}, r)\mathcal{N}(\boldsymbol{c}_2^T\boldsymbol{x}, r)\ldots\mathcal{N}(\boldsymbol{c}_3^T\boldsymbol{x}, r),$$

Bayesian formulation (discounting) of weighted least squares

$$p(y_i|\boldsymbol{c}_i, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{c}_i^T\boldsymbol{x}, r),$$
$$p(\boldsymbol{x}|X, Y) \propto \mathcal{N}(\boldsymbol{c}_1\boldsymbol{x}, r)^{\phi}\mathcal{N}(\boldsymbol{c}_2\boldsymbol{x}, r)^{\phi^2}\ldots\mathcal{N}(\boldsymbol{c}_n\boldsymbol{x}, r)^{\phi^n}\ldots,$$

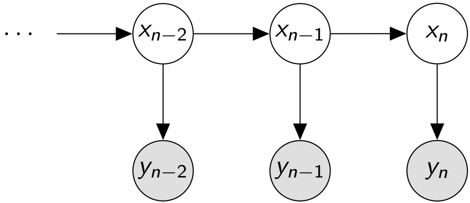
Not proper probability rule.
Solution:

- admit that $\boldsymbol{x}_n$ and $\boldsymbol{x}_{n+1}$ are different variables

State space model

Bayesian formulation, example

$$p(y_i|\boldsymbol{c}_i, \boldsymbol{x}_i) = \mathcal{N}(\boldsymbol{c}_i \boldsymbol{x}_i, r),$$
$$p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}) = \mathcal{N}(\boldsymbol{x}_{i-1}, Q),$$



Inference in state space models

Path estimation, we seek $\boldsymbol{x}_{1:n} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$

$$p(x_{1:n}|y_{1:n}) \propto p(y_n|x_n)p(x_n|x_{n-1}) \ldots p(y_1|x_1)p(x_1)$$

Filtering:

$$p(x_n|y_{1:n}) \propto \int p(x_{1:n}|y_{1:n})dx_{1:n-1}$$

Smoothing (fixed lag L):

$$p(x_{n-L}|y_{1:n}) \propto \int p(x_{1:n}|y_{1:n})dx_{1:n-L-1}dx_{n-L+1:n}$$

Prediction (h-step ahead):

$$p(x_{n+h}|y_{1:n}) \propto p(x_{n+h}|x_{n+h-1}) \ldots p(x_n|y_{1:n}).$$

Bayesian filtering

Assume that we have previous estimate $p(x_{n-1}|y_{1:n-1})$. Then:

$$p(x_n, x_{n-1}|y_{1:n}) = \frac{p(y_n|x_n)p(x_n|x_{n-1})p(x_{n-1}|y_{n-1})}{p(y_n|y_{1:n-1})}$$

$$\begin{aligned} p(x_n|y_{1:n}) &= \int p(x_n, x_{n-1}|y_{1:n})dx_{n-1} \\ &= \frac{p(y_n|x_n)\int p(x_n|x_{n-1})p(x_{n-1}|y_{n-1})dx_n}{p(y_n|y_{1:n-1})} \end{aligned}$$

Bayesian filtering:

$$\begin{aligned} p(x_n|y_{1:n-1}) &= \int p(x_n|x_{n-1})p(x_{n-1}|y_{1:n-1})dx_{n-1} && \text{prediction} \\ p(x_n|y_{1:n}) &= \frac{p(y_n|x_n)p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})}. && \text{update} \end{aligned}$$

Kalman filtering

General linear state-space model

$$
\begin{aligned}
\boldsymbol{x}_{n+1} &= A\boldsymbol{x}_n + B\boldsymbol{u}_n + v_n, & v_n &\sim \mathcal{N}(0, Q), \\
\boldsymbol{y}_n &= C\boldsymbol{x}_n + D\boldsymbol{u}_n + w_n, & w_n &\sim \mathcal{N}(0, R).
\end{aligned}
$$

where $\boldsymbol{x}_n$ is the state variable, and $\boldsymbol{y}_n$ is the observation at time n.
Filtering result:

$$
\begin{aligned}
p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}) &= \mathcal{N}(\hat{\boldsymbol{x}}_{n|n}, P_{n|n}), & p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n-1}) &= \mathcal{N}(\hat{\boldsymbol{x}}_{n|n-1}, P_{n|n-1}), \\
\hat{\boldsymbol{x}}_{n|n} &= \hat{\boldsymbol{x}}_{n|n-1} + K(\boldsymbol{y}_n - \hat{\boldsymbol{y}}_n), & \hat{\boldsymbol{x}}_{n|n-1} &= A\boldsymbol{x}_{n-1} + B\boldsymbol{u}_{n-1} \\
P_{n|n} &= (I - KC)P_{n|n-1}, & P_{n|n-1} &= AP_{n-1|n-1}A^T + Q \\
\hat{\boldsymbol{y}}_n &= C\hat{\boldsymbol{x}}_{n|n-1} + D\boldsymbol{u}_n, \\
K &= P_{n|n-1}C^T R_y^{-1}, \\
R_y &= C^T P_{n|n-1} C + R,
\end{aligned}
$$

It is a filter

Estimation without $\boldsymbol{u}_n$:

$$\begin{aligned}
\hat{x}_n &= A\hat{x}_{n-1} + K(y_n - C\hat{x}_{n-1}), \\
&= Ky_n + (A - KC)\hat{x}_{n-1} \\
&= Ky_n + (A - KC)\left(Ky_{n-1} + (A - KC)\hat{x}_{n-2}\right), \\
&= \kappa_1 y_n + \kappa_2 y_{n-1} + \kappa_3 y_{n-2} \ldots
\end{aligned}$$

It is a filter

Estimation without $\boldsymbol{u}_n$:

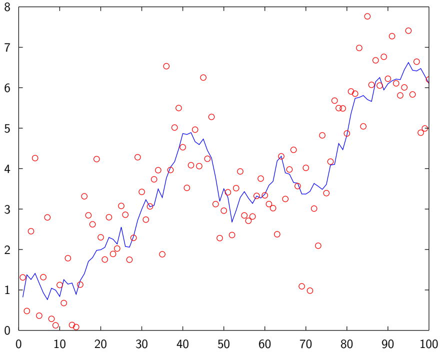
$$\begin{aligned}
\hat{x}_n &= A\hat{x}_{n-1} + K(y_n - C\hat{x}_{n-1}), \\
&= Ky_n + (A - KC)\hat{x}_{n-1} \\
&= Ky_n + (A - KC)\left(Ky_{n-1} + (A - KC)\hat{x}_{n-2}\right), \\
&= \kappa_1 y_n + \kappa_2 y_{n-1} + \kappa_3 y_{n-2} \ldots
\end{aligned}$$

- Kalman filter is a moving average filter of the measurements, with exponentially decaying weights.
- Coefficients κ depends on the physical model of the system.

Covariance matrices and their influence

Trivial example

$$x_{n+1} = x_n + v_n, \quad \mathrm{var}(v_t) = q,$$
$$y_n = x_n + w_n, \quad \mathrm{var}(w_t) = r,$$



Covariance matrices and their influence

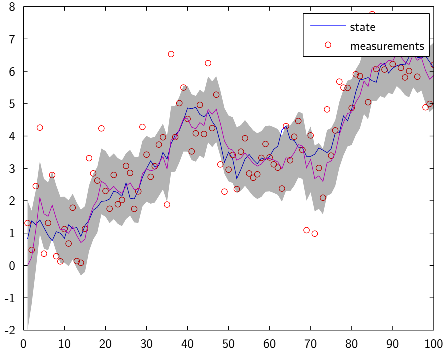
Trivial example

$$x_{n+1} = x_n + v_n, \quad \mathrm{var}(v_t) = q,$$
$$y_n = x_n + w_n, \quad \mathrm{var}(w_t) = r,$$

Kalman filter is:

$$\hat{x}_n = \hat{x}_{n-1} + k(y_n - \hat{x}_n)$$
$$k = \frac{p_n + q}{p_n + q + r},$$
$$p_n = (1-k)(p_{n-1} + q),$$

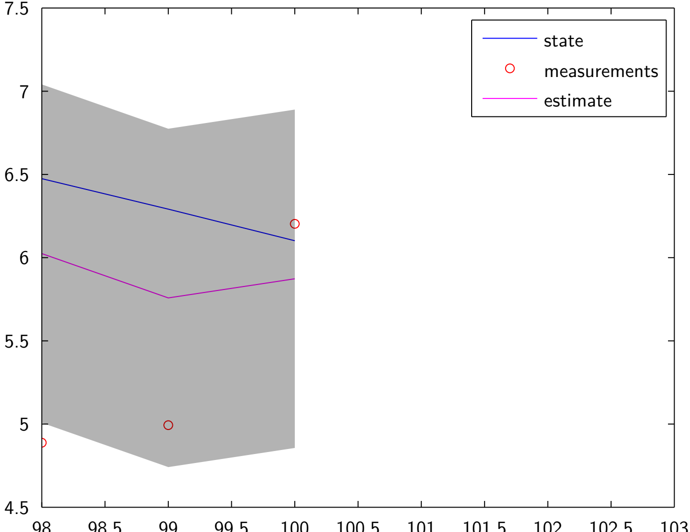
converges to steady state values. (Riccatti equation)



Visualization

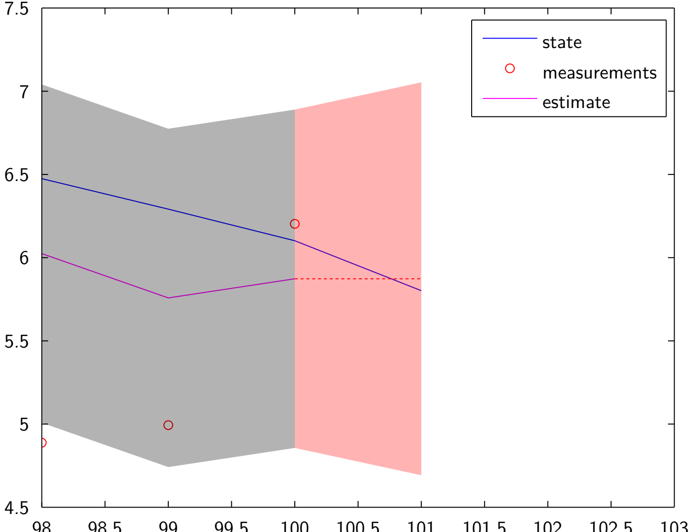
Trivial example:

$\hat{x}_n. = \hat{x}_{n-1} + k(y_n - \hat{x}_n),\ \ p_{n|n-1} = p_{n-1} + q,\ \ p_{n|n} = (1-k)p_{n|n-1}.$



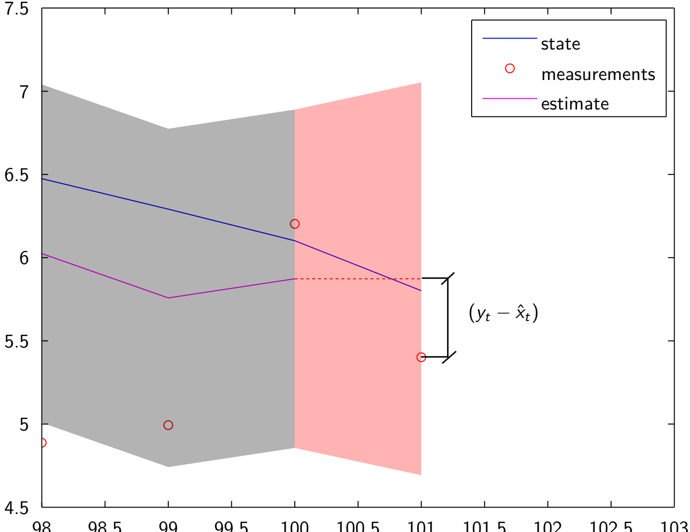
Visualization

Trivial example:
$\hat{x}_n. = \hat{x}_{n-1} + k(y_n - \hat{x}_n),\;\; p_{n|n-1} = p_{n-1} + q,\;\; p_{n|n} = (1-k)p_{n|n-1}.$



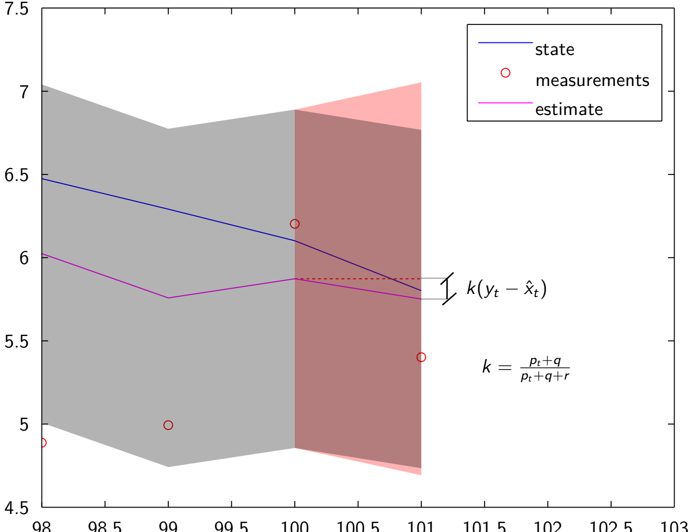
Visualization

Trivial example:
$\hat{x}_n. = \hat{x}_{n-1} + k(y_n - \hat{x}_n),\ \ p_{n|n-1} = p_{n-1} + q,\ \ p_{n|n} = (1-k)p_{n|n-1}.$



Visualization

Trivial example:
$\hat{x}_n. = \hat{x}_{n-1} + k(y_n - \hat{x}_n), \; p_{n|n-1} = p_{n-1} + q, \; \; p_{n|n} = (1-k)p_{n|n-1}.$



General state space model

Linear Gaussian models have limited power, but Bayesian filtering model can be non-linear and non-Gaussian.

Non-linear model (atmosphere):

$$\frac{dx}{dt} = f(x_t, u_t),$$

is heavily non-linear (Lorenz ode for convection).

State: weather on Earth

Observations: satellite, all weather stations

Filter: Ensemble Kalman Filter

Non-Gaussian noise (finance):

$$\begin{aligned} p(x_n|x_{n-1}) &= L(x_{n-1}, b) \\ &= \exp\left(-\frac{|x_n - x_{n-1}|}{b}\right) \end{aligned}$$

Daily returns on shares are Laplace distributed.

State: price of shares, trend in evolution

Observations: price of shares

Filter: Bayesian filter with Laplace (Finematic)

State space idea

Model of price of shares, s_n:

$$s_n = as_{n-1} + b + v_t,$$
$$y_n = s_n + w_t,$$

Prediction

$$s_{n+1} = as_n + b$$

where a and b are also unknown.

State space idea

Model of price of shares, s_n:

$$s_n = as_{n-1} + b + v_t,$$
$$y_n = s_n + w_t,$$

Prediction

$$s_{n+1} = as_n + b$$

where a and b are also unknown.
State: $x_n = [p_n, a_n, b_n]$

$$p(s_n|s_{n-1}) = L(a_{n-1}p_n + b_{n-1}, \sigma)N(a_{n-1}, \sigma_a)N(b_{n-1}, \sigma_b)$$
$$p(y_n|s_n) = N(p_n, \sigma_p),$$

How to compute? How to find $\sigma_a, \sigma_b, \sigma_p$?

General approximations

Taylor expansion: for non-linear systems with Gaussian Noise, the non-linearity is approximated by Taylor

$$x_{n+1} = f(x_n) \approx x_0 + \frac{f(x)}{dx}(x - x_0),$$

Yielding linearized system => Extended Kalman Filter

Particle Filter: posterior density is in empirical form

$$p(x_n|y_{1:n}) = \sum_{j=1}^{J} w_j \delta(x_n - x_n^{(j)}),$$

Extended Kalman Filter

Extended Kalman filter:

$$\begin{aligned} \boldsymbol{x}_n &= f(\boldsymbol{x}_n, \boldsymbol{u}_n) + v_n, \\ \boldsymbol{y}_n &= h(\boldsymbol{x}_n, \boldsymbol{u}_n) + w_n, \end{aligned}$$

Update equations:

$$\begin{aligned} p(\boldsymbol{x}_n | \boldsymbol{y}_{1:n}) &= \mathcal{N}(\hat{\boldsymbol{x}}_{n|n}, P_{n|n}), \\ \hat{\boldsymbol{x}}_{n|n} &= \hat{\boldsymbol{x}}_{n|n-1} + K(\boldsymbol{y}_n - \hat{\boldsymbol{y}}_n) \\ P_{n|n} &= (I - KC)P_{n|n-1}, \\ \hat{\boldsymbol{y}}_n &= h(\hat{\boldsymbol{x}}_{n|n-1}, \boldsymbol{u}_n), \\ K &= P_{n|n-1} C^T R_y^{-1}, \\ R_y &= C^T P_{n|n-1} C + R, \end{aligned}$$

$$\begin{aligned} p(\boldsymbol{x}_n | \boldsymbol{y}_{1:n-1}) &= \mathcal{N}(\hat{\boldsymbol{x}}_{n|n-1}, P_{n|n-1}), \\ \hat{\boldsymbol{x}}_{n|n-1} &= f(\hat{\boldsymbol{x}}_{n-1}, \boldsymbol{u}_{n-1}) \\ P_{n|n-1} &= A P_{n-1|n-1} A^T + Q \\ A &= \frac{\partial f(\boldsymbol{x}_n, \boldsymbol{u}_n)}{\partial \boldsymbol{x}_n} \Big|_{\hat{x}_{n-1|n-1}} \\ C &= \frac{\partial h(\boldsymbol{x}_n, \boldsymbol{u}_n)}{\partial \boldsymbol{x}_n} \Big|_{\hat{x}_{n|n-1}} \end{aligned}$$

Particle filter

Posterior

$$p(x_n|y_{1:n}) = \sum_{j=1}^{J} w_j \delta(x_n - x_n^{(j)}),$$

Prediction

$$\begin{aligned} p(x_n|y_{1:n-1}) &= \int p(x_n|x_{n-1}) p(x_{n-1}|y_{n-1}) dx_n \\ &= \sum_{j=1}^{J} \delta(x_{n-1} - x_{n-1}^{(j)}) p(x_n|x_{n-1}) \overset{\text{sample}}{\approx} \sum_{j=1}^{J} w_j \delta(x_{n|n-1} - x_{n|n-1}^{(j)}) \end{aligned}$$

Update

$$p(x_n|y_{1:n}) \propto p(y_n|x_n) \sum_{j=1}^{J} \delta(x_{n-1} - x_{n-1}^{(j)}) = \sum_{j=1}^{J} \overline{w}_j \delta(x_n - x_n^{(j)})$$

$$\overline{w}_j = p(y_n|x_n) w_j$$

Trick sample from weighted distribution – multinomial sampling (resampling)

Trivial example

Track sinewave in noise:

$$y_n = a_n \sin(\omega n + \phi_n) + e_n, \quad e_n \sim N(0, \sigma),$$

with known ω and unknown time-variant amplitude a_n and phase ϕ_n.

Trivial example

Track sinewave in noise:

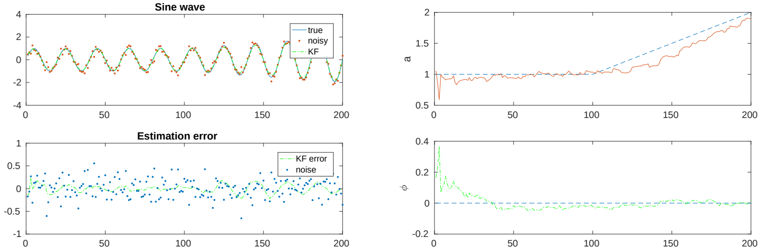
$$y_n = a_n \sin(\omega n + \phi_n) + e_n, \quad e_n \sim N(0, \sigma),$$

with known ω and unknown time-variant amplitude a_n and phase ϕ_n.
State model for slow varying parameters:

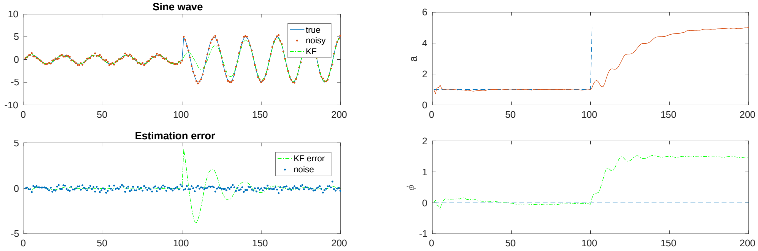
$$\begin{aligned} p(\phi_n|\phi_{n-1}) &= N(\phi_{n-1}, q_\phi), \\ p(a_n|a_{n-1}) &= N(a_{n-1}, q_a), \end{aligned}$$

State: $x_n = [a_n, \phi_n]$.
Non-linear observations => EKF.

Trivial example – slow



Trivial example – step change



Trivial example

Track sinewave in noise:

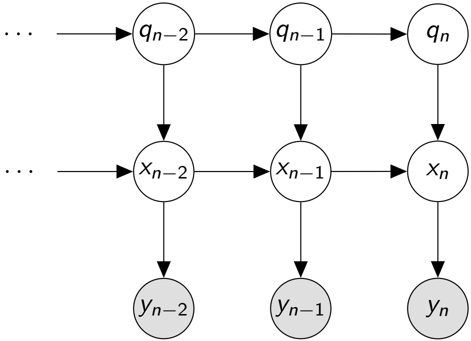
$$y_n = a_n \sin(\omega n + \phi_n) + e_n, \quad e_n \sim N(0, \sigma),$$

with known ω and unknown time-variant amplitude a_n and phase ϕ_n.
State model for fast varying parameters

$$\begin{aligned}
p(\phi_n|\phi_{n-1}) &= 0.9N(\phi_{n-1}, q_\phi) + 0.1N(\phi_{n-1}, 100q_\phi), \\
p(\phi_n|\phi_{n-1}) &= 0.9N(\phi_{n-1}, q_\phi) + 0.1U(-\pi, \pi), \\
p(\phi_n|\phi_{n-1}, q_{\phi n}) &= N(\phi_{n-1}, q_{\phi n}) \\
p(q_n|q_{n-1}) &= G(\gamma q_{n-1}, \gamma)
\end{aligned}$$

Adaptive Kalman filtering.

Adaptive Kalman filter



Assignment

Sine wave tracking		points
EKF		10
Estimation of $\boldsymbol{q}$		30
– ARD Variational Bayes		
– MAP estimate		
Particle filter		30
– mixture model		
– estimate of $\boldsymbol{q}$		

Examples of use