Inference in linear Gaussian models

Mattias Villani

Department of Statistics
Stockholm University

mattiasvillani.com @matvil @matvil mattiasvillani

Inference about the state

- **Bayesian posterior distribution**

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

- **Predictive distribution**

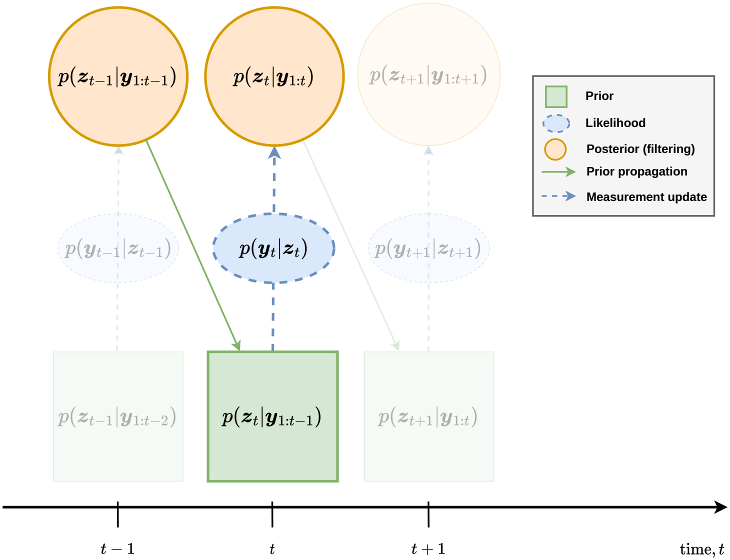
$$p(\tilde{y}|\boldsymbol{y}) = \int p(\tilde{y}|\boldsymbol{\theta}, \boldsymbol{y})p(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}$$

State-space models:

- **Filtering posterior**: $p(\boldsymbol{z}_t|\boldsymbol{y}_{1:t})$. Instantaneous posterior.
- **Smoothing posterior**: $p(\boldsymbol{z}_t|\boldsymbol{y}_{1:T})$. Retrospective posterior.
- **Joint smoothing posterior**: $p(\boldsymbol{z}_{1:T}|\boldsymbol{y}_{1:T})$.

- Prior on initial state: $p(\boldsymbol{z}_0)$. Often $\boldsymbol{z}_0 \sim N(\boldsymbol{\mu}_{0|0}, \boldsymbol{\Omega}_{0|0})$.

The Bayes filter



The Bayes filter

Bayes filter update

$$p(\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1}) \quad \Rightarrow \quad p(\mathbf{z}_t|\mathbf{y}_{1:t-1}) \quad \Rightarrow \quad p(\mathbf{z}_t|\mathbf{y}_{1:t})$$

prior propagation

$$\underbrace{p(\mathbf{z}_t|\mathbf{y}_{1:t-1})}_{\text{prior at } t} = \int p(\mathbf{z}_t|\mathbf{z}_{t-1})\underbrace{p(\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1})}_{\text{posterior at } t-1}\mathrm{d}\mathbf{z}_{t-1}$$

measurement update

$$\underbrace{p(\mathbf{z}_t|\mathbf{y}_{1:t})}_{\text{posterior at } t} \propto p(\mathbf{y}_t|\mathbf{z}_t)\underbrace{p(\mathbf{z}_t|\mathbf{y}_{1:t-1})}_{\text{prior at } t}$$

The Kalman filter

- Linear Gaussian models $\Rightarrow$ Bayes filter in closed form.
- Priors and posteriors are Gaussian. Mean/variance over time.
- **Prior at time t**

$$\boldsymbol{z}_0 \sim N(\boldsymbol{\mu}_{0|0}, \boldsymbol{\Omega}_{0|0})$$

- **Prior at time t**

$$\boldsymbol{z}_t | \boldsymbol{y}_{1:t-1} \sim N(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Omega}_{t|t-1})$$

- **Filtering posterior at time t**

$$\boldsymbol{z}_t | \boldsymbol{y}_{1:t} \sim N(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Omega}_{t|t})$$

The Kalman filter algorithm

Kalman filter update for linear Gaussian state-space models

$$p(\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1}) \quad \Rightarrow \quad p(\mathbf{z}_t|\mathbf{y}_{1:t-1}) \quad \Rightarrow \quad p(\mathbf{z}_t|\mathbf{y}_{1:t})$$

prior propagation

$$N(\boldsymbol{\mu}_{t-1,t-1}, \boldsymbol{\Omega}_{t-1,t-1}) \Rightarrow N(\boldsymbol{\mu}_{t,t-1}, \boldsymbol{\Omega}_{t,t-1})$$

$$\begin{aligned} \boldsymbol{\mu}_{t,t-1} &= \mathbf{A}\boldsymbol{\mu}_{t-1,t-1} + \mathbf{B}\mathbf{u}_t \\ \boldsymbol{\Omega}_{t,t-1} &= \mathbf{A}\boldsymbol{\Omega}_{t-1,t-1}\mathbf{A}^\top + \boldsymbol{\Sigma}_\eta \end{aligned}$$

measurement update

$$N(\boldsymbol{\mu}_{t,t-1}, \boldsymbol{\Omega}_{t,t-1}) \Rightarrow N(\boldsymbol{\mu}_{t,t}, \boldsymbol{\Omega}_{t,t})$$

$$\begin{aligned} \boldsymbol{\mu}_{t,t} &= \boldsymbol{\mu}_{t,t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{C}\boldsymbol{\mu}_{t,t-1}) \\ \boldsymbol{\Omega}_{t,t} &= (\mathbf{I} - \mathbf{K}_t\mathbf{C})\boldsymbol{\Omega}_{t,t-1} \end{aligned}$$

with Kalman gain

$$\mathbf{K}_t = \boldsymbol{\Omega}_{t,t-1}\mathbf{C}^\top \left(\mathbf{C}\boldsymbol{\Omega}_{t,t-1}\mathbf{C}^\top + \boldsymbol{\Sigma}_\varepsilon\right)^{-1}$$

Interactive: Kalman filtering local level data

Settings for the data generating model

time series length	100
true σ_ε	1.01
true σ_η	1.01

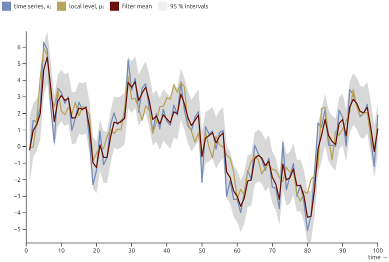
Settings for the model used for filtering

noise std σ_ε	1.01
innov std σ_η	1.01
initial mean $\mu_{0\|0}$	10
initial stdev $\sigma_{0\|0}$	100

estimate σ_ε and σ_η ☐

plot: ☑ time series, x_t ☑ local level, μ_t ☑ filter mean ☑ 95 % intervals

Simulate new data



Interactive: Kalman filtering Nile river data

noise std σ_ε 100

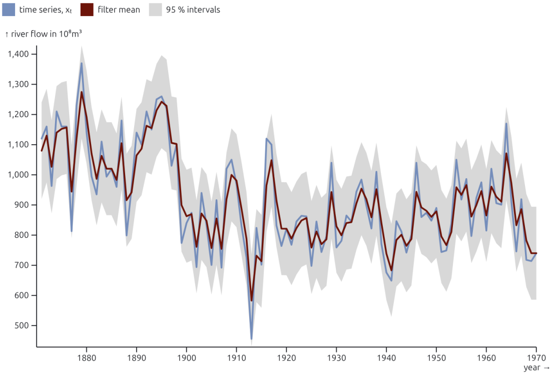
innov std σ_η 100

initial mean $\mu_{0|0}$ 1000

initial stdev $\sigma_{0|0}$ 100

estimate σ_ε and σ_η ☐

plot: ☑ time series, x_t ☑ filter mean ☑ 95 % intervals



The Kalman filter time update in Julia

```julia
function kalmanfilter_update(μ, Ω, u, y, A, B, C, Σₑ, Σₙ)

    # Prediction step - moving state forward without new measurement
    μ̄ = A*μ + B*u
    Ω̄ = A*Ω*A' + Σₙ

    # Measurement update - updating the N(μ̄, Ω̄) prior with the new data point
    K = Ω̄*C' / (C*Ω̄*C' .+ Σₑ) # Kalman Gain
    μ = μ̄ + K*(y .- C*μ̄)
    Ω = (I - K*C)*Ω̄
    return μ, Ω

end
```

The Kalman filter algorithm in Julia

```julia
function kalmanfilter(U, Y, A, B, C, Σₑ, Σₙ, μ₀, Σ₀)

    T = size(Y,1)   # Number of time steps
    n = length(μ₀)  # Dimension of the state vector

    # Storage for the mean and covariance state vector trajectory over time
    μ_traj = zeros(T, n)
    Σ_traj = zeros(n, n, T)

    # The Kalman iterations
    μ = μ₀
    Σ = Σ₀
    for t = 1:T
        μ, Σ = kalmanfilter_update(μ, Σ, U[t,:]', Y[t,:]', A, B, C, Σₑ, Σₙ)
        μ_traj[t,:] = μ
        Σ_traj[:,:,t] = Σ
    end

    return μ_traj, Σ_traj

end
```

R demo using the dlm package

- Quarto notebook: qmd | html

Sampling from the joint smoothing posterior

- **Joint smoothing posterior** decomposed backward in time:

$$
\begin{aligned}
p(\boldsymbol{z}_{1:T}|\boldsymbol{y}_{1:T}) &= p(\boldsymbol{z}_T|\boldsymbol{y}_{1:T})p(\boldsymbol{z}_{T-1}|\boldsymbol{z}_T, \boldsymbol{y}_{1:T}) \cdots p(\boldsymbol{z}_0|\boldsymbol{z}_{1:T}, \boldsymbol{y}_{1:T}) \\
&= p(\boldsymbol{z}_T|\boldsymbol{y}_{1:T}) \prod_{t=1}^{T-1} p(\boldsymbol{z}_t|\boldsymbol{z}_{t+1:T}, \boldsymbol{y}_{1:T})
\end{aligned}
$$

- The assumptions of the state-space model imply that

$$
p(\boldsymbol{z}_t|\boldsymbol{z}_{t+1:T}, \boldsymbol{y}_{1:T}) = p(\boldsymbol{z}_t|\boldsymbol{z}_{t+1}, \boldsymbol{y}_{1:t})
$$

- **Kalman smoother**: recursion to compute marginal posteriors for linear Gaussian models.
- Compute smoothed estimates backward in time

$$
\boldsymbol{\mu}_{t|T}, \boldsymbol{\Omega}_{t|T} \quad \text{for } t = T, T-1, \ldots, 1
$$

- "Initial" $\boldsymbol{\mu}_{T|T}$ and $\boldsymbol{\Omega}_{T|T}$ available from the Kalman filter.

Forward Filtering Backward Sampling (FFBS)

- Sampling from the joint smoothing posterior $p(\mathbf{z}_{1:T}|\mathbf{y}_{1:T})$.

Sampling the joint smoothing posterior in the LGSS model

Input: time series $\mathbf{y}_{1:T} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$
model parameters $\boldsymbol{\theta}$
initial state prior mean $\boldsymbol{\mu}_{0|0}$
initial state prior covariance matrix $\boldsymbol{\Omega}_{0|0}$
number of samples from the posterior m

$\mathbf{A}, \mathbf{C}, \boldsymbol{\Sigma}_\eta, \boldsymbol{\Sigma}_\varepsilon \leftarrow \textsc{setupStateSpace}(\boldsymbol{\theta})$

$\{\boldsymbol{\mu}_{t|t}, \boldsymbol{\Omega}_{t|t}, \boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Omega}_{t|t-1}\}_{t=1}^{T} \leftarrow \textsc{kalmanFilter}(\mathbf{y}_{1:T}, \mathbf{A}, \mathbf{C}, \boldsymbol{\Sigma}_\eta, \boldsymbol{\Sigma}_\varepsilon, \boldsymbol{\mu}_{0|0}, \boldsymbol{\Omega}_{0|0})$

for i in $1, \ldots, m$ **do**

  Simulate $\mathbf{z}_T^{(i)} \sim N(\boldsymbol{\mu}_{T|T}, \boldsymbol{\Omega}_{T|T})$

  for t in $T\text{-}1, T\text{-}2, \ldots, 1$ **do**

    $\boldsymbol{\mu}_{t|t+1} \leftarrow \boldsymbol{\mu}_{t|t} + \boldsymbol{\Omega}_{t|t}\mathbf{A}^\top \boldsymbol{\Omega}_{t+1|t}^{-1}(\mathbf{z}_{t+1}^{(i)} - \boldsymbol{\mu}_{t+1|t})$

    $\boldsymbol{\Omega}_{t|t+1} \leftarrow \boldsymbol{\Omega}_{t|t} - \boldsymbol{\Omega}_{t|t}\mathbf{A}^\top \boldsymbol{\Omega}_{t+1|t}^{-1}\mathbf{A}\boldsymbol{\Omega}_{t|t}$

    Simulate $\mathbf{z}_t^{(i)}|\mathbf{z}_{t+1}^{(i)} \sim N(\boldsymbol{\mu}_{t|t+1}, \boldsymbol{\Omega}_{t|t+1})$

  end

end

Inference about static parameters

- Joint posterior of the state $\boldsymbol{z}_{1:T}$ and parameters $\boldsymbol{\theta}$

$$p(\boldsymbol{\theta}, \boldsymbol{z}_{1:T} | \boldsymbol{y}_{1:T})$$

- Example: Local level model with static parameters $\boldsymbol{\theta} = (\sigma_\varepsilon^2, \sigma_\eta^2)$.

- **Gibbs sampling**:

$$\begin{aligned} \text{Sample: } & \boldsymbol{\theta} | \boldsymbol{z}_{1:T}, \boldsymbol{y}_{1:T} \\ \text{Sample: } & \boldsymbol{z}_{1:T} | \boldsymbol{\theta}, \boldsymbol{y}_{1:T} \end{aligned}$$

- **Marginal-conditional sampling**:

$$\begin{aligned} \text{Sample: } & \qquad \boldsymbol{\theta} | \boldsymbol{y}_{1:T} \\ \text{Sample: } & \boldsymbol{z}_{1:T} | \boldsymbol{\theta}, \boldsymbol{y}_{1:T} \end{aligned}$$

The Kalman filter marginalizes out the state

- Marginal posterior:

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}) \propto p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

- The **marginal likelihood** $p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta})$

$$p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1},\boldsymbol{\theta})$$

where

$$p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1},\boldsymbol{\theta}) = \int p(\boldsymbol{y}_t|\boldsymbol{z}_t,\boldsymbol{\theta})p(\boldsymbol{z}_t|\boldsymbol{y}_{1:t-1},\boldsymbol{\theta})d\boldsymbol{z}_t$$

- Linear Gaussian models: integrals are tractable and

$$p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1},\boldsymbol{\theta}) = N(\boldsymbol{y}_t|\boldsymbol{C}\boldsymbol{\mu}_{t|t-1},\ \boldsymbol{C}\boldsymbol{\Omega}_{t|t-1}\boldsymbol{C}^\top + \boldsymbol{\Sigma}_\varepsilon)$$

- **Maximum likelihood estimator**

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\arg\max}\, p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta})$$

Marginal likelihood

Marginal likelihood - linear Gaussian state-space model

Input: time series $\mathbf{y}_{1:T} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$
model parameters $\boldsymbol{\theta}$
initial state prior mean $\boldsymbol{\mu}_{0|0}$
initial state prior covariance matrix $\boldsymbol{\Omega}_{0|0}$

$\mathbf{A}, \mathbf{C}, \boldsymbol{\Sigma}_\eta, \boldsymbol{\Sigma}_\varepsilon \leftarrow \textsc{setupStateSpace}(\boldsymbol{\theta})$

$\{\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Omega}_{t|t-1}\}_{t=1}^{T} \leftarrow \textsc{kalmanFilter}(\mathbf{y}_{1:T}, \mathbf{A}, \mathbf{C}, \boldsymbol{\Sigma}_\eta, \boldsymbol{\Sigma}_\varepsilon, \boldsymbol{\mu}_{0|0}, \boldsymbol{\Omega}_{0|0})$

`lml` = 0

for t in 1:T **do**

  `lml` = `lml` + $\log N(\mathbf{y}_t | \mathbf{C}\boldsymbol{\mu}_{t|t-1}, \mathbf{C}\boldsymbol{\Omega}_{t|t-1}\mathbf{C}^\top + \boldsymbol{\Sigma}_\varepsilon)$

end

Output: log marginal likelihood $p(\mathbf{y}_{1:T}|\boldsymbol{\theta})$, `lml`.

Interactive: Kalman filtering Nile river data

noise std σ_ε 100

innov std σ_η 100

initial mean $\mu_{0|0}$ 1000

initial stdev $\sigma_{0|0}$ 100

estimate σ_ε and σ_η

plot: time series, x_t filter mean 95 % intervals