

ADVANCED MACHINE LEARNING

STATE-SPACE MODELS

LECTURE 2

Mattias Villani

**Division of Statistics and Machine Learning
Department of Computer and Information Science
Linköping University**



LECTURE OVERVIEW

- ▶ Estimating model parameters
- ▶ Bayesian inference for the LGSS model
- ▶ Live demo of some R packages

ESTIMATING MODEL PARAMETERS

- ▶ The **linear Gaussian state-space (LGSS) model**

$$\text{Measurement eq: } \mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \varepsilon_t \quad \varepsilon_t \stackrel{iid}{\sim} N(0, \Omega_\varepsilon)$$

$$\text{State eq: } \mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \nu_t \quad \nu_t \stackrel{iid}{\sim} N(0, \Omega_\nu)$$

- ▶ The elements in \mathbf{A} , \mathbf{B} , \mathbf{C} , Ω_ε and Ω_ν may be unknown.
- ▶ Example: time-varying regression with p covariates \mathbf{z}_t ($p \times 1$)

$$y_t = \mathbf{z}_t^T \boldsymbol{\beta}_t + \varepsilon_t, \quad \varepsilon_t \stackrel{iid}{\sim} N(0, \Omega_\varepsilon)$$

$$\beta_{1t} = a_1 \cdot \beta_{1,t-1} + \nu_t \quad \nu_t \stackrel{iid}{\sim} N(0, \Omega_\nu)$$

$$\vdots$$

$$\beta_{pt} = a_p \cdot \beta_{p,t-1} + \nu_t \quad \nu_t \stackrel{iid}{\sim} N(0, \Omega_\nu)$$

- ▶ Here $C = \mathbf{z}_t^T$, $\mathbf{x}_t = \boldsymbol{\beta}_t$ and $\mathbf{A} = \text{Diag}(a_1, \dots, a_p)$.
- ▶ The state space model's matrices (\mathbf{A} etc) are parametrized by $\boldsymbol{\theta} = (\theta_1, \dots, \theta_s)$. To be explicit: $A(\boldsymbol{\theta})$, $B(\boldsymbol{\theta})$, ..., $\Omega_\nu(\boldsymbol{\theta})$.

ESTIMATING MODEL PARAMETERS

- ▶ Two options: Maximum likelihood estimate (MLE) or Bayesian.
- ▶ **Likelihood function**

$$p(\mathbf{y}_1, \dots, \mathbf{y}_T | \theta) = \prod_{t=1}^T p(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \theta)$$

- ▶ How compute $p(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \theta)$? The trick: i) condition on \mathbf{x}_t , ii) exploit conditional independencies, iii) get rid of \mathbf{x}_t by integrating it out:

$$\begin{aligned} p(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \theta) &= \int p(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \mathbf{x}_t, \theta) p(\mathbf{x}_t | \mathbf{y}_{1:t-1}, \theta) d\mathbf{x}_t \\ &= \int p(\mathbf{y}_t | \mathbf{x}_t, \theta) p(\mathbf{x}_t | \mathbf{y}_{1:t-1}, \theta) d\mathbf{x}_t \end{aligned}$$

- ▶ Note:
 - ▶ $p(\mathbf{x}_t | \mathbf{y}_{1:t-1}, \theta) = \overline{\text{bel}}(\mathbf{x}_t)$ is Gaussian
 - ▶ $p(\mathbf{y}_t | \mathbf{x}_t, \theta)$ is Gaussian
 - ▶ $p(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \theta)$ is then also Gaussian [not obvious, but expected].

ESTIMATING MODEL PARAMETERS

- ▶ Remember: we are looking for the Gaussian $p(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \theta)$.
- ▶ Mean by law of iterated expectations ($E = EE$)

$$\mathbb{E}(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \theta) = \mathbf{C} \mathbb{E}(\mathbf{x}_t | \mathbf{y}_{1:t-1}, \theta) = \mathbf{C} \bar{\boldsymbol{\mu}}_t$$

- ▶ Variance by conditional variance formula ($V = EV + VE$)

$$\begin{aligned} \mathbb{V}(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \theta) &= \mathbb{E}_{\mathbf{x}_t | \mathbf{y}_{1:t-1}, \theta} [\mathbb{V}(\mathbf{y}_t | \mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta)] \\ &\quad + \mathbb{V}_{\mathbf{x}_t | \mathbf{y}_{1:t-1}, \theta} [\mathbb{E}(\mathbf{y}_t | \mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta)] \\ &= \Omega_{\varepsilon} + \mathbb{V}_{\mathbf{x}_t | \mathbf{y}_{1:t-1}, \theta}(\mathbf{C} \mathbf{x}_t) = \Omega_{\varepsilon} + \mathbf{C} \bar{\boldsymbol{\Sigma}}_t \mathbf{C}^T \end{aligned}$$

ESTIMATING MODEL PARAMETERS

- ▶ In summary, the **likelihood function** is

$$p(\mathbf{y}_1, \dots, \mathbf{y}_T | \theta) = \prod_{t=1}^T N(\mathbf{y}_t | \mathbf{C} \bar{\boldsymbol{\mu}}_t, \mathbf{C} \bar{\boldsymbol{\Sigma}}_t \mathbf{C}^T + \Omega_\varepsilon)$$

where \mathbf{C} , Ω_ε , $\bar{\boldsymbol{\mu}}_t$ and $\bar{\boldsymbol{\Sigma}}_t$ all depend on θ generally.

- ▶ The Kalman filter gives us everything we need for $p(\mathbf{y}_1, \dots, \mathbf{y}_T | \theta)$!
- ▶ **Numerical optimization** (e.g. `optim` in R) to find **MLE** $\hat{\theta}_{MLE}$.
- ▶ Approximate $\mathbb{V}(\hat{\theta}_{MLE})$ from the numerical Hessian.
- ▶ Sampling from the **posterior distribution**

$$p(\theta | \mathbf{y}_1, \dots, \mathbf{y}_T) \propto p(\mathbf{y}_1, \dots, \mathbf{y}_T | \theta) p(\theta)$$

by **Metropolis-Hastings**.

STATE SMOOTHING

- **Filtering** (real time):

$$p(\mathbf{x}_t | \mathbf{y}_{1:t})$$

- **Smoothing** (retrospective):

$$p(\mathbf{x}_t | \mathbf{y}_{1:T})$$

- Start at the end $t = T$. We already have $p(\mathbf{x}_T | \mathbf{y}_{1:T})$ from the last iteration of the Kalman filter. Work yourself backward in time to obtain $p(\mathbf{x}_{T-1} | \mathbf{y}_{1:T}), \dots, p(\mathbf{x}_1 | \mathbf{y}_{1:T})$.
- Note: the end result are the **marginal** densities at any t , $p(\mathbf{x}_t | \mathbf{y}_{1:T})$. More work to do if one also wants $p(\mathbf{x}_{t_1}, \mathbf{x}_{t_2} | \mathbf{y}_{1:T})$ for some times t_1 and t_2 .

STATE SMOOTHING

► **Algorithm Smoothing**($\mathbf{s}_{t+1}, \mathbf{S}_{t+1}, \mu_t, \Sigma_t, \bar{\mu}_{t+1}, \bar{\Sigma}_{t+1}$)

- Mean update:

$$\mathbf{s}_t = \mu_t + \Sigma_t \mathbf{A}^T \bar{\Sigma}_{t+1}^{-1} (\mathbf{s}_{t+1} - \bar{\mu}_{t+1})$$

- Covariance update:

$$\mathbf{S}_t = \Sigma_t + \Sigma_t \mathbf{A}^T \bar{\Sigma}_{t+1}^{-1} (\mathbf{S}_{t+1} - \bar{\Sigma}_{t+1}) \bar{\Sigma}_{t+1}^{-1} \mathbf{A} \Sigma_t$$

- Return $\mathbf{s}_t, \mathbf{S}_t$

BAYESIAN INFERENCE FOR THE STATE

- ▶ How to **sample** from **posterior** of the **state** $p(\mathbf{x}_1, \dots, \mathbf{x}_T | \mathbf{y}_{1:T}, \theta)$?
- ▶ Single-move Gibbs sampling (inefficient!):
 - ▶ for $t = 1, \dots, T$ do:
 - ▶ $\mathbf{x}_t | \mathbf{x}_{-t}, \mathbf{y}_{1:T} \sim \text{Normal}$
- ▶ **Forward Filtering Backward Sampling (FFBS)**:
 - ▶ Run the Kalman filter forward in time $t = 1, \dots, T$.
 - ▶ Simulate states backward in time $t = T, T - 1, \dots, 1$:

$$\mathbf{x}_t | \mathbf{x}_{t+1:T}, \mathbf{y}_{1:T}, \theta \sim N(\mathbf{h}_t, \mathbf{H}_t)$$

$$\mathbf{h}_t = \mu_t + \Sigma_t \mathbf{A}^T \bar{\Sigma}_{t+1}^{-1} (\mathbf{x}_{t+1} - \bar{\mu}_{t+1})$$

and

$$\mathbf{H}_t = \Sigma_t - \Sigma_t \mathbf{A}^T \bar{\Sigma}_{t+1}^{-1} \mathbf{A} \Sigma_t.$$

- ▶ The difference between the smoothing distributions and FFBS distributions is that in FFBS we condition on $\mathbf{x}_{t+1:T}$.
- ▶ So the FFBS gives the *joint* (smoothing) posterior for $\mathbf{x}_{1:T}$, whereas the state smoothing gives the *marginal* posterior of \mathbf{x}_t for all t .

DLM PACKAGE

► The **linear Gaussian state-space (LGSS)** model

$$\text{Measurement eq: } \mathbf{Y}_t = \mathbf{C}\mathbf{x}_t + \varepsilon_t \quad \varepsilon_t \stackrel{iid}{\sim} N(0, \Omega_\varepsilon)$$

$$\text{State eq: } \mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \nu_t \quad \nu_t \stackrel{iid}{\sim} N(0, \Omega_\nu)$$

► In the dlm package

$$\text{Measurement eq: } Y_t = F\theta_t + v_t \quad \varepsilon_t \stackrel{iid}{\sim} N(0, V)$$

$$\text{State eq: } \theta_t = G\theta_{t-1} + w_t \quad \nu_t \stackrel{iid}{\sim} N(0, W)$$

- θ_t is the state vector in dlm. Y_t are the measurements (a vector).
- The dlm notation goes back to West and Harrison's yellow DLM bible.
- The state is an **unknown**, so it is a **greek letter**.
- Measurements is a **random variable** so it is a **capital letter**.
- dlm can also handle when F, G, V, W vary of over time.

DLM PACKAGE

► DLM

$$\text{Measurement eq: } Y_t = F\theta_t + v_t \quad \varepsilon_t \stackrel{iid}{\sim} N(0, V)$$

$$\text{State eq: } \theta_t = G\theta_{t-1} + w_t \quad v_t \stackrel{iid}{\sim} N(0, W)$$

► Main functions:

- `d1m` - creates the d1m model object
- `d1mFilter` - Kalman filtering
- `d1mSmooth` - State smoothing
- `d1mLL` - computes the log-likelihood