State-Space Models

Parameter inference and software

Mattias Villani

Statistiska institutionen
Stockholms universitet

Institutionen för datavetenskap
Linköpings universitet

mattiasvillani.com @matvil mattiasvillani

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**

$$\begin{aligned} \text{Measurement eq:} \quad & \mathrm{y}_t = \mathrm{C}\mathrm{x}_t + \varepsilon_t & \varepsilon_t \overset{iid}{\sim} N(0, \Omega_\varepsilon) \\ \text{State eq:} \quad & \mathrm{x}_t = \mathrm{A}\mathrm{x}_{t-1} + \mathrm{B}\mathrm{u}_t + \nu_t & \nu_t \overset{iid}{\sim} N(0, \Omega_\nu) \end{aligned}$$

- The elements in A, **B**, C, Ω_ε and Ω_ν may be unknown.
- Example: time-varying regression with p covariates z_t $(p \times 1)$

$$\begin{aligned} y_t &= \mathrm{z}_t^T \beta_t + \varepsilon_t, & \varepsilon_t \overset{iid}{\sim} N(0, \Omega_\varepsilon) \\ \beta_{1t} &= a_1 \cdot \beta_{1,t-1} + \nu_t & \nu_t \overset{iid}{\sim} N(0, \Omega_\nu) \\ &\vdots \\ \beta_{pt} &= a_p \cdot \beta_{p,t-1} + \nu_t & \nu_t \overset{iid}{\sim} N(0, \Omega_\nu) \end{aligned}$$

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

Estimating model parameters

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

$$p\left(y_1, \dots, y_T | \theta\right) = \prod_{t=1}^{T} p\left(y_t | y_{1:t-1}, \theta\right)$$

- How compute $p\left(y_t | y_{1:t-1}, \theta\right)$? The trick: i) condition on x_t, ii) exploit conditional independencies, iii) get rid of x_t by integrating it out:

$$\begin{aligned} p\left(y_t | y_{1:t-1}, \theta\right) &= \int p\left(y_t | y_{1:t-1}, x_t, \theta\right) p\left(x_t | y_{1:t-1}, \theta\right) dx_t \\ &= \int p\left(y_t | x_t, \theta\right) p\left(x_t | y_{1:t-1}, \theta\right) dx_t \end{aligned}$$

- Note:
  - $p\left(x_t | y_{1:t-1}, \theta\right) = \overline{\mathrm{bel}}(x_t)$ is Gaussian
  - $p\left(y_t | x_t, \theta\right)$ is Gaussian
  - $p\left(y_t | y_{1:t-1}, \theta\right)$ is then also Gaussian [not obvious, but

Estimating model parameters

- Remember: we are looking for the Gaussian $p\left(\mathrm{y}_t|\mathrm{y}_{1:t-1},\theta\right)$.

- Mean by law of iterated expectations $(E = EE)$

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

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

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

Estimating model parameters

- In summary, the likelihood function is

$$p(y_1, ..., y_T|\theta) = \prod_{t=1}^{T} N\left(y_t|C\bar{\mu}_t, C\bar{\Sigma}_t C^T + \Omega_\varepsilon\right)$$

  where C, Ω_ε, $\bar{\mu}_t$ and $\bar{\Sigma}_t$ all depend on θ generally.
- The Kalman filter gives us everything we need for $p(y_1, ..., y_T|\theta)$!
- Numerical optimization (e.g. `optim` in R) to find MLE $\hat{\theta}_{MLE}$.
- Approximate $\mathbb{V}\left(\hat{\theta}_{MLE}\right)$ from the numerical Hessian.
- Sampling from the posterior distribution

$$p(\theta|y_1, ..., y_T) \propto p(y_1, ..., y_T|\theta)\, p(\theta)$$

  by Metropolis-Hastings.

State smoothing

- **Filtering** (real time):

$$p(x_t|y_{1:t})$$

- **Smoothing** (retrospective):

$$p(x_t|y_{1:T})$$

- Start at the end $t = T$. We already have $p\left(x_T|y_{1:T}\right)$ from the last iteration of the Kalman filter. Work yourself backward in time to obtain $p(x_{T-1}|y_{1:T}), \ldots, p(x_1|y_{1:T})$.

- Note: the end result are the **marginal** densities at any t, $p(x_t|y_{1:T})$. More work to do if one also wants $p(x_{t_1}, x_{t_2}|y_{1:T})$ for some times t_1 and t_2.

State smoothing

- **Algorithm** `Smoothing`$(s_{t+1}, S_{t+1}, \mu_t, \Sigma_t, \bar{\mu}_{t+1}, \bar{\Sigma}_{t+1})$

  - Mean update:

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

  - Covariance update:

$$S_t = \Sigma_t + \Sigma_t A^T \bar{\Sigma}_{t+1}^{-1} (S_{t+1} - \bar{\Sigma}_{t+1}) \bar{\Sigma}_{t+1}^{-1} A \Sigma_t$$

  - Return s_t, S_t

Bayesian inference for the state

- How to **sample** from **posterior** of the **state** $p(x_1, ..., x_T|y_{1:T}, \theta)$?
- Simulate the state trajectory **backward in time** starting with x_T:

$$p(x_{1:T}|y_{1:T}, \theta) = p(x_T|y_{1:T}, \theta)p(x_{T-1}|x_T, y_{1:T}, \theta) \cdots p(x_1|x_{T-1:2}, y_{1:T}, \theta$$

- **Forward Filtering Backward Sampling** (**FFBS**):
  - Run the Kalman filter forward in time $t = 1, ..., T$.
  - Simulate x_T from $N(\mu_T, \Sigma_T)$.
  - Simulate states backward in time $t = T-1, T-2, ..., 1$:

$$x_t|x_{t+1:T}, y_{1:T}, \theta \sim N(h_t, H_t)$$

$$h_t = \mu_t + \Sigma_t A^T \bar{\Sigma}_{t+1}^{-1} \left(x_{t+1} - \bar{\mu}_{t+1} \right)$$

  and

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

- Note: FFBS distributions conditions on $x_{t+1:T}$.
- So the FFBS gives the *joint* (smoothing) posterior for $x_{1:T}$,

dlm package in R

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

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

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

- In the `dlm` package

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

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

- 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 in R

- DLM

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

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

- Main functions:
  - `dlm` - creates the dlm model object
  - `dlmFilter` - Kalman filtering
  - `dlmSmooth` - State smoothing
  - `dlmLL` - computes the log-likelihood