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**

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

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

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

$$y_t = \mathrm{z}_t^T \beta_t + \varepsilon_t, \qquad \varepsilon_t \overset{iid}{\sim} N(0, \Omega_\varepsilon)$$

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

$$\vdots$$

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

  - 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

- Maximum likelihood estimate (MLE) or Bayesian.
- **Likelihood function**

$$p\left(\mathrm{y}_1, \ldots, \mathrm{y}_T | \theta\right) = \prod_{t=1}^{T} p\left(\mathrm{y}_t | \mathrm{y}_{1:t-1}, \theta\right)$$

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

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

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

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

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

  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 $p(\mathrm{y}_1, ..., \mathrm{y}_T|\theta)$!
- 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

$$p(\theta|\mathrm{y}_1, ..., \mathrm{y}_T) \propto p(\mathrm{y}_1, ..., \mathrm{y}_T|\theta)\, p(\theta)$$

  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 $p(x_{T-1}|y_{1:T}), ..., p(x_1|y_{1:T})$.
- 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 $p(\mathrm{x}_1, \ldots, \mathrm{x}_T | \mathrm{y}_{1:T}, \theta)$?
- Simulate state trajectory **backward in time** starting at x_T:

$$p(\mathrm{x}_{1:T}|\mathrm{y}_{1:T}, \theta) = p(\mathrm{x}_T|\mathrm{y}_{1:T}, \theta)p(\mathrm{x}_{T-1}|\mathrm{x}_T, \mathrm{y}_{1:T}, \theta) \cdots p(\mathrm{x}_1|\mathrm{x}_{T-1:2}, \mathrm{y}_{1:T}, \theta)$$

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

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

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

and

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

- Note: FFBS distributions conditions on $\mathrm{x}_{t+1:T}$.
- FFBS gives the *joint* (smoothing) posterior for $\mathrm{x}_{1:T}$, whereas the state smoothing gives the *marginal* posterior of x_t for all 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.
- The `dlm` notation goes back to West and Harrison's book.
- 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