

0.1 All without a loop (AWOL) smoothing

Traditionally in DLMs, the Kalman filter is used in order to draw from the latent states $\theta_{0:T}$ through forward filtering, backward sampling (FFBS). This requires running the Kalman filter in order to determine the marginal distribution of θ_T , then drawing $\theta_t|\theta_{t+1:T}$ for $t = T-1, T-2, \dots, 1$ (Carter and Kohn [1994], Frühwirth-Schnatter [1994]). The all without a loop (AWOL) method determines the joint distribution of $\theta_{0:T}$ and draws from it all at once, i.e. without a loop. The idea comes from Rue [2001], which introduces a Cholesky factorization algorithm for drawing from a Gaussian Markov Random Field (GMRF), and notes that the conditional distribution of $\theta_{0:T}$ given $y_{1:T}$ in a Gaussian linear statespace model is a special case. McCausland et al. [2011] improves the idea by coming up with a slightly faster method in order to draw $\theta_{0:T}$ based on essentially the same reasoning. The name comes from Kastner and Frühwirth-Schnatter [2013], who use the method along with ancillarity-sufficiency interweaving in order to speed up MCMC estimation of stochastic volatility models.

The basic idea is this. Suppose our model is

$$\begin{aligned} y_t &= F_t \theta_t + v_t \\ \theta_t &= G_t \theta_{t-1} + w_t \end{aligned}$$

with $v_t \stackrel{ind}{\sim} N(0, V_t)$ independent of $w_t \stackrel{ind}{\sim} N(0, W_t)$ for $t = 1, 2, \dots, T$ and $\theta_0 \sim N(m_0, C_0)$. Now $(y_{1:T}, \theta_{0:T})$ is joint normal conditional on $(V_{1:T}, W_{1:T})$ (in this section, everything is conditional on $V_{1:T}$ and $W_{1:T}$). So we can write $p(\theta_{0:T}|y_{1:T})$ as

$$\log p(\theta_{0:T}|y_{1:T}) = -\frac{1}{2}g(\theta_{0:T}, y_{1:T}) + K$$

where K is some constant with respect to $\theta_{0:T}$ and

$$g(\theta_{0:T}, y_{1:T}) = \theta_{0:T}' \Omega \theta_{0:T} - 2a' \theta_{0:T}.$$

However, we also have

$$\log p(\theta_{0:T}|y_{1:T}) = \log p(\theta_{0:T}, y_{1:T}) - \log p(y_{1:T}).$$

This means that

$$\begin{aligned} g(\theta_{0:T}, y_{1:T}) &= (\theta_0 - m_0)C_0^{-1}(\theta_0 - m_0) + K' \\ &\quad \sum_{t=1}^T (y_t - F_t \theta_t)' V_t^{-1} (y_t - F_t \theta_t) + \\ &\quad \sum_{t=1}^T (\theta_t - G_t \theta_{t-1})' W_t^{-1} (\theta_t - G_t \theta_{t-1}). \end{aligned}$$

where K' is another constant that doesn't depend on $\theta_{0:T}$.

So now we can identify blocks of Ω with the cross product terms of the θ_t 's and blocs of a with the single

product terms. Specifically, we Ω is a banded diagonal matrix with

$$\Omega = \begin{bmatrix} \Omega_{00} & \Omega_{01} & 0 & \ddots & 0 & 0 \\ \Omega_{10} & \Omega_{11} & \Omega_{12} & \ddots & 0 & 0 \\ 0 & \Omega_{21} & \Omega_{22} & \ddots & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \ddots & \Omega_{T-1,T-1} & \Omega_{T-1,T} \\ 0 & 0 & 0 & \ddots & \Omega_{T,T-1} & \Omega_{TT} \end{bmatrix}$$

and $a = (a'_0, a'_1, \dots, a'_T)$ where the Ω_{st} 's and a_t 's defined below:

$$\begin{aligned} \Omega_{00} &= C_0^{-1} + G_1' W_1^{-1} G_1 \\ \Omega_{tt} &= F_t' V_t^{-1} F_t + W_t^{-1} + G_{t+1}' W_{t+1}^{-1} G_{t+1} & \text{for } t = 1, 2, \dots, T-1 \\ \Omega_{TT} &= F_T' V_T^{-1} F_T + W_T^{-1} \\ \Omega_{t,t-1} &= -W_t^{-1} G_t & \text{for } t = 1, 2, \dots, T \\ \Omega_{t-1,t} &= -G_t' W_t^{-1} = \Omega_{t,t-1}' & \text{for } t = 1, 2, \dots, T \\ a_0 &= C_0^{-1} m \\ a_t &= F_t' V_t^{-1} y_t & \text{for } t = 1, 2, \dots, T. \end{aligned}$$

Together, Ω and a determine the normal distribution from which $\theta_{0:T}$ should be drawn. Rue [2001] shows how to take advantage of the sparsity of Ω in order to quickly compute its Cholesky factorization and in order to find the mean vector from a and this factorization. McCausland et al. [2011] shows that instead of computing these quantities directly, you can draw θ_T and $\theta_t | \theta_{t+1:T}$ iteratively, which ultimately reduces the number of linear algebra operations which must be performed and thus speeds up the computation despite taking advantage of essentially the same mathematical technology.

The resulting algorithm requires a couple more intermediate quantities. Let $\Sigma_0 = \Omega_{00}^{-1}$, $\Sigma_t = (\Omega_{tt} - \Omega_{t,t-1} \Sigma_{t-1} \Omega_{t-1,t})^{-1}$ for $t = 1, 2, \dots, T$, $m_0 = \Sigma_0 a_0$, and $m_t = \Sigma_t (a_t - \Omega_{t,t-1} m_{t-1})$ for $t = 1, 2, \dots, T$. Then

$$\begin{aligned} \theta_T &\sim N(m_T, \Sigma_T) \\ \theta_{t|t+1:T} &\sim N(m_t - \Sigma_t \Omega_{t,t+1} \theta_{t+1}, \Sigma_t) & \text{for } t = T-1, T-2, \dots, 0. \end{aligned}$$

McCausland et al. [2011] shows how to quickly compute the required linear algebra operations finds that this method is often faster than simply doing the Cholesky factorization. Note that this method requires that the model retains the Markov property. This suggests that if we want to use a transformation of the latent states, which should not use, e.g., some function of their differences. This would make Ω fairly complicated and force us to do the full Cholesky factorization, or to use AWOL on the θ 's and then transform them to e.g. the γ 's. A better idea would be to simply scale the states by the appropriate cholesky factor in order to retain the nice structure of Ω .

References

- Chris K Carter and Robert Kohn. On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553, 1994.
- Sylvia Frühwirth-Schnatter. Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15(2):183–202, 1994.
- Gregor Kastner and Sylvia Frühwirth-Schnatter. Ancillarity-sufficiency interweaving strategy (asis) for boosting mcmc estimation of stochastic volatility models. *Computational Statistics & Data Analysis*, 2013.
- William J McCausland, Shirley Miller, and Denis Pelletier. Simulation smoothing for state-space models: A computational efficiency analysis. *Computational Statistics & Data Analysis*, 55(1):199–212, 2011.
- Håvard Rue. Fast sampling of gaussian markov random fields. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):325–338, 2001.