Lecture 15

State-Space Models II

Reading: SS17 Chapter 6.2-6.4, Chapter 6.12; BD Chapter 9.4-9.7

MATH 8090 Time Series Analysis
Week 15

Whitney Huang
Clemson University

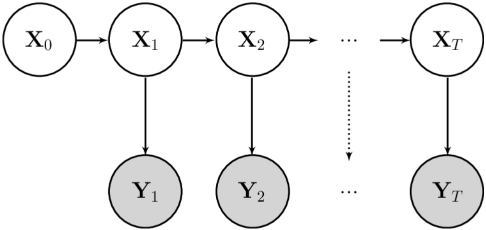
Agenda

1. **Review**

2. **Forecasting, Filtering, and Smoothing**

3. **Estimating the State-Space Model Parameters**

State-Space Model



$$\text{State: } \boldsymbol{X}_t = M_t \boldsymbol{X}_{t-1} + \boldsymbol{V}_t, \quad \boldsymbol{V}_t \overset{i.i.d.}{\sim} \mathrm{WN}(\boldsymbol{0}, Q_t), \quad t = 1, 2, \cdots$$

$$\text{Observation: } \boldsymbol{Y}_t = H_t \boldsymbol{X}_t + \boldsymbol{W}_t, \quad \boldsymbol{W}_t \overset{i.i.d.}{\sim} \mathrm{WN}(\boldsymbol{0}, R_t), \quad t = 1, 2, \cdots$$

- $\boldsymbol{X_t} \in \mathbb{R}^p$ and $\boldsymbol{Y_t} \in \mathbb{R}^q$ are the state vector and the observation vector at time t
- $\boldsymbol{M}_t$ is the $p \times p$ transition matrix, and $\boldsymbol{H}_t$ is the $q \times p$ observation matrix
- $\boldsymbol{V}_t$ and $\boldsymbol{W}_t$ are the state and observation noises

Forecasting, Filtering, and Smoothing

Goal: To estimate the underlying unobserved signal $\boldsymbol{X}_t$, given the data $\boldsymbol{Y}_{1:s} = \boldsymbol{y}_{1:s} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_s\}$:

- When $s < t$, the problem is called forecasting or prediction
- When $s = t$, the problem is called filtering
- When $s > t$, the problem is called smoothing

In addition to these estimates, we would also want to measure their precision. The solution to these problems is accomplished via the Kalman filter and Kalman smoother

The Kalman Filter: General Results

Assume the filtering distribution at time $t-1$ is

$$[\boldsymbol{X}_{t-1}|\boldsymbol{Y}_{1:t-1}] \sim \mathrm{N}\left(\boldsymbol{\mu}^a_{t-1}, \Sigma^a_{t-1}\right)$$

- **Forecast Step:** Gives the forecast distribution at time t:

  $$[\boldsymbol{X}_t|\boldsymbol{Y}_{1:t-1}] \sim \mathrm{N}\left(\boldsymbol{\mu}^f_t, \Sigma^f_t\right),$$

  where $\boldsymbol{\mu}^f_t = M_t \boldsymbol{\mu}^a_{t-1}$, and $\Sigma^f_t = M_t \Sigma^a_{t-1} M_t^T + Q_t$.
- **Update Step:** updates the forecast distribution using new data $\boldsymbol{Y}_t$

  $$[\boldsymbol{X}_t|\boldsymbol{Y}_{1:t}] \sim \mathrm{N}\left(\boldsymbol{\mu}^a_t, \Sigma^a_t\right),$$

  where $\boldsymbol{\mu}^a_t = \boldsymbol{\mu}^f_t + K_t\left(\boldsymbol{Y}_t - H_t\boldsymbol{\mu}^f_t\right)$, and $\Sigma^a_t = \left(I - K_t H_t^T\right)\Sigma^f_t$, and

  $$K_t = \Sigma^f_t H_t^T \left(H_t \Sigma^f_t H_t^T + R_t\right)^{-1}$$

  is the Kalman gain matrix

Filtering for Local Level Model: I

Let's begin with a particularly simple example of a state space model: the local level model

- Local level model:

$$Y_t = X_t + W_t, \quad \{W_t\} \sim \mathrm{N}(0, \sigma_W^2)$$
$$X_t = X_{t-1} + V_t, \quad \{V_t\} \sim \mathrm{N}(0, \sigma_V^2)$$

  and X_0 is a R.V. that

  - is uncorrelated with W_t's and V_t's
  - has $\mathrm{E}(X_0) = \mu_0$ and $\mathrm{Var}(X_0) = \sigma_0^2$

- Filtering problem is to predict unknown state X_t based on data up to time t, i.e., $\boldsymbol{Y}_{1:t} = (y_1, \cdots, y_t)^T$

Filtering for Local Level Model: II

Best linear predictor of X_t given $\boldsymbol{Y}_{1:t}$ is

$$\mu_t^a \stackrel{\text{def}}{=} \mathrm{E}(X_t|\boldsymbol{Y}_{1:t}) = \mu_t + \Sigma_{t,t}^T \Sigma_{Y,t}^{-1}(\boldsymbol{Y}_{1:t} - \boldsymbol{\mu}_{1:t}),$$

where

- $\mu_t = \mathrm{E}(X_t)$, $\boldsymbol{\mu}_{1:t}$ is a vector containing, for $j = 1, \cdots, t$,

$$\mu_j \stackrel{\text{def}}{=} \mathrm{E}(X_j) = \mathrm{E}(X_j + W_j) = \mathrm{E}(Y_j)$$

- Vector $\Sigma_{t,t}$ contains covarinces between X_t and $\boldsymbol{Y}_{1:t}$
- (i,j)th element of matrix $\Sigma_{Y,t}$ is covariance between Y_i and Y_j
- Note: $\mathrm{E}(\mu_t^a) = \mathrm{E}[\mathrm{E}(X_t|\boldsymbol{Y}_{1:t})] = \mathrm{E}(X_t) = \mu_t$
- With $\sigma_t^2 \stackrel{\text{def}}{=} \mathrm{Var}(X_t)$, MSE for predictor is

$$\mathrm{E}[(X_t - \mu_t^a)^2] = \sigma_t^2 - \Sigma_{t,t}^T \Sigma_{Y,t}^{-1} \Sigma_{t,t} \stackrel{\text{def}}{=} \Sigma_t^a$$

Forecasting for Local Level Model: I

Forecasting: estimate X_{t+1} given $\boldsymbol{Y}_{1:t}$

- Best linear predictor of X_{t+1} given $\boldsymbol{Y}_{1:t}$ is

$$\mu_{t+1}^{f} \stackrel{\text{def}}{=} \mathrm{E}(X_{t+1}|\boldsymbol{Y}_{1:t}) = \mu_{t+1} + \Sigma_{t+1,t}^{T}\Sigma_{Y,t}^{-1}(\boldsymbol{Y}_{1:t} - \boldsymbol{\mu}_{1:t}),$$

  where vector $\Sigma_{t+1,t}$ has covaraince between X_{t+1} and $\boldsymbol{Y}_{1:t}$

- Note: $\mathrm{E}(\mu_{t+1}^{f}) = \mathrm{E}[\mathrm{E}(X_{t+1}|\boldsymbol{Y}_{1:t})] = \mathrm{E}(X_{t+1}) = \mu_{t+1}$

- MSE for predictor is

$$\mathrm{E}[(X_{t+1} - \mu_{t+1}^{f})^2] = \sigma_{t+1}^2 - \Sigma_{t+1,t}^{T}\Sigma_{Y,t}^{-1}\Sigma_{t+1,t} \stackrel{\text{def}}{=} \Sigma_{t+1}^{f}$$

Forecasting for Local Level Model: II

- Let's also consider best linear predictor of Y_{t+1} given $\boldsymbol{Y}_{1:t}$:

$$Y_{t+1}^{t} \stackrel{\text{def}}{=} \mathrm{E}(Y_{t+1}|\boldsymbol{Y}_{1:t}) = \mu_{Y,t+1} + \tilde{\Sigma}_{t+1,t}^{T}\Sigma_{Y,t}^{-1}(\boldsymbol{Y}_{1:t} - \boldsymbol{\mu}_{Y,1:t}),$$

  where the vector $\tilde{\Sigma}_{t+1,t}$ has covaraínces between Y_{t+1} and $\boldsymbol{Y}_{1:t}$

- However, note that, for $j = 1, \cdots, t$

$$\mathrm{Cov}(Y_{t+1}, Y_j) = \mathrm{Cov}(X_{t+1} + W_{t+1}, Y_j) = \mathrm{Cov}(X_{t+1}, Y_j)$$

- Thus $\tilde{\Sigma}_{t+1,t} = \Sigma_{t+1,t}$, yielding

$$Y_{t+1}^{t} = \mu_{Y,t+1} + \Sigma_{t+1,t}^{T}\Sigma_{Y,t}^{-1}(\boldsymbol{y}_{1:t} - \boldsymbol{\mu}_{Y,1:t}) = \mu_{t+1}^{f}$$

  $\Rightarrow$ difference between Y_{t+1} and X_{t+1} is W_{t+1}, therefore they have the same estimator, but their MSEs differ:

$$\mathrm{E}\left[(Y_{t+1} - Y_{t+1}^{f})^2\right] = \Sigma_{t+1}^{f} + \sigma_W^2$$

Filtering for Local Level Model: III

- To implement filtering, i.e., compute μ_t^a, need to determine:

  1. $\mu_j = \mathrm{E}(X_j),\ j = 1, \cdots, t$
  2. Elements of $\Sigma_{t,t}$, i.e., covaraince between X_t and $\boldsymbol{Y}_{1:t}$
  3. Elements of $\Sigma_{Y,t}$, i.e., covariances between Y_j and Y_k, $1 \leq j \leq k \leq t$

- To compute Σ_t^a, i.e., MSE for μ_t^a, need $\sigma_t^2 = \mathrm{Var}(X_t)$ in addition to 2 and 3 above

- Since $X_t = X_{t-1} + V_t$ and $Y_t = X_t + W_t$, telescoping yields $X_j = X_0 + \sum_{l=1}^{j} V_l$ and $Y_j = X_0 + \sum_{l=1}^{j} V_l + W_j, j = 1, \cdots, t$

Filtering for Local Level Model: IV

- Using

$$X_j = X_0 + \sum_{l=1}^{j} V_l \text{ and } Y_j = X_0 + \sum_{l=1}^{j} + W_j, \quad j = 1, \cdots, t,$$

get $\mu_j = \mathrm{E}[X_j] = \mathrm{E}[X_0] = \mu_0$ and (assuming $j \le k \le t$)

$$\begin{aligned} \mathrm{Cov}(X_t, Y_j) &= \mathrm{Cov}\left(X_0 + \sum_{l=1}^{t} V_l, X_0 + \sum_{l=1}^{j} V_l + W_j \right) \\ &= \sigma_0^2 + j\sigma_V^2 \\ \mathrm{Cov}(Y_j, Y_k) &= \mathrm{Cov}\left(X_0 + \sum_{l=1}^{j} V_l + W_j, X_1 + \sum_{l=1}^{k} V_l + W_k \right) \\ &= \sigma_0^2 + j\sigma_V^2 + \delta_{jk}\sigma_W^2, \end{aligned}$$

where $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ if $j \neq k$

Filtering for Local Level Model: V

Using

$$X_t = X_0 + \sum_{l=1}^{t} V_l,$$

get

$$\sigma_t^2 = \mathbb{V}\mathrm{ar}(X_t) = \sigma_0^2 + t\sigma_V^2$$

- Now we have all the pieces needed to form μ_t^a and its MSE Σ_t^a
- **Note**: similar argument leads to pieces needed to form forecast μ_{t+1}^f and its MSE Σ_{t+1}^f

Kalman Recursions for Filtering/Forecasting: I

While straightforward conceptually, forming

$$\mu_t^a = \mu_t + \Sigma_{t,t}^T \Sigma_{Y,t}^{-1} (\boldsymbol{Y}_{1:t} - \boldsymbol{\mu}_{1:t})$$

and

$$\mu_{t+1}^f = \mu_{t+1} + \Sigma_{t+1,t}^T \Sigma_{Y,t}^{-1} (\boldsymbol{Y}_{1:t} - \boldsymbol{\mu}_{1:t})$$

via these equations requires inversion of matrix $\Sigma_{Y,t}$ whose dimension $t \times t$ becomes problematic as t gets large ☹ ⇒

> The celebrated Kalman recursions give a recipe that avoids explicit matrix inversion

- **Idea**: at time $t-1$, we have 4 quantities of interest: fitted value μ_{t-1}^a, and forecast μ_t^f and their associated MSEs Σ_{t-1}^a and Σ_t^f
- **Note**: $\mu_{t-1}^a = \mu_t^f$ for local level model (but not others)

Kalman Recursions for Filtering/Forecasting: II

- At time t, new observation Y_t becomes available
- Kalman recursion takes μ_t^f, Σ_t^f and Y_t and yields
  - fitted values μ_t^a and forecast μ_{t+1}^f
  - associated MSEs Σ_t^a and Σ_{t+1}^f
- There are six steps in the Kalman recursion:
  1. steps 1 and 2 are preparatory
  2. steps 3 and 4 yield μ_t^a and Σ_t^a (filtering)
  3. steps 5 and 6 yield μ_{t+1}^f and Σ_{t+1}^f (forecasting)

Kalman Recursions for Filtering/Forecasting: III

1. Compute innovation:

$$U_t = Y_t - Y_t^{t-1} = Y_t - \mu_t^f$$

2. Compute MSE for Y_t^{t-1}:

$$\Sigma_t^f + \sigma_W^2 \stackrel{\text{def}}{=} F_t$$

3. Compute new filtered value:

$$\mu_t^a = \mu_t^f + K_t U_t,$$

where $K_t \stackrel{\text{def}}{=} \Sigma_t^f / F_t$ is the so-called Kalman gain

4. Compute MSE for new filtered value:

$$\Sigma_t^a = \Sigma_t^f (1 - K_t)$$

Kalman Recursions for Filtering/Forecasting: IV

5. Compute new forecast:

$$\mu_{t+1}^{f} = \mu_{t}^{f} + K_t U_t = \mu_t^a$$

6. Compute MSE for new forecast:

$$\Sigma_{t+1}^{f} = \Sigma_t (1 - K_t) + \sigma_V^2 = \Sigma_t^a + \sigma_V^2$$

Recursions are carried out for $t = 0, \cdots, n$ with inputs $\mathrm{E}[X_0] = \mu_0$, $\mathrm{Var}(X_0) = \sigma_0^2$ and $Y_t' s$

Kalman Recursions for Filtering/Forecasting: V

To prove validity of steps 3 and 4, need to show that $\mu_t^f + K_t U_t$ is equal to μ_t^a, and $\Sigma_t^f(1 - K_t)$ is equal to Σ_t^a

- **Key fact**: X_t conditioned on both $U_t = Y_t - Y_t^{t-1}$ and $\boldsymbol{Y}_{1:t-1}$ is the same as X_t conditioned on $\boldsymbol{Y}_{1:t-1}$ becasue

$$\begin{aligned}\mathrm{Cov}(X_t, U_t|\boldsymbol{Y}_{1:t-1}) &= \mathrm{Cov}(X_t, Y_t - Y_t^{t-1}|\boldsymbol{Y}_{1:t-1}) \\ &= \mathrm{Cov}(X_t, X_t + W_t|\boldsymbol{Y}_{1:t-1}) = \mathrm{Var}(X_t|\boldsymbol{Y}_{1:t-1}) \\ &= \Sigma_t^f\end{aligned}$$

We have

$$\mu_t^a = \mu_t^f + \frac{\Sigma_t^f}{F_t} U_t, \text{ and } \Sigma_t^a = \Sigma_t^f - \frac{\left(\Sigma_t^f\right)^2}{F_t}$$

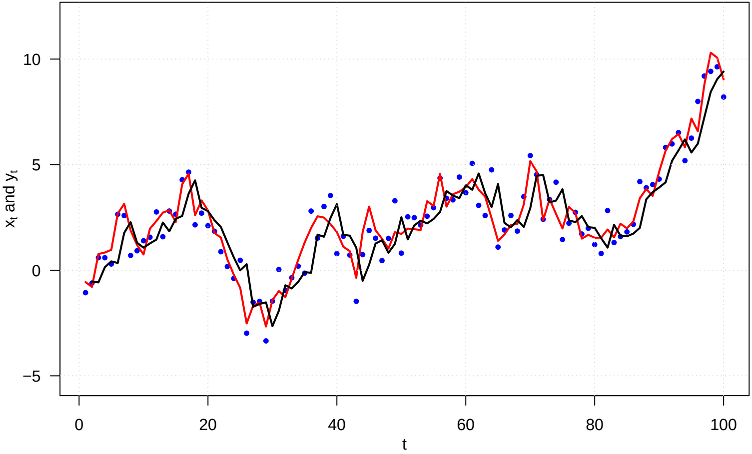
since $K_t \stackrel{\text{def}}{=} \frac{\Sigma_t^f}{F_t}$, we get required

$$\mu_t^a = \mu_t^f + K_t U_t \text{ and } \Sigma_t^a = \Sigma_t^f(1 - K_t)$$

Simulated Example: Local Level Model with SNR $= 2$

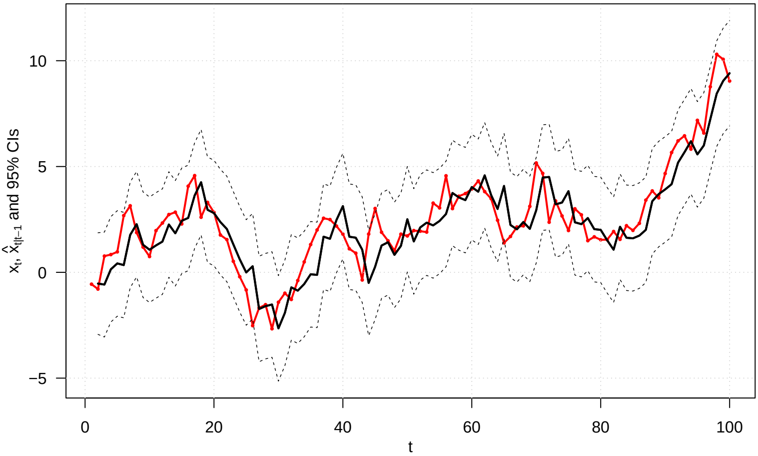
Setup: $\mu_0 = 0$, $\sigma_0^2 = 1 = \sigma_V^2$, $\sigma_W^2 = 0.5$

Time series Y_t, states X_t, and forecasts μ_t^f



Simulated Data from Local Level Model with SNR $= 2$

States X_t, forecasts μ_t^f, and 95% CIs based on Σ_t^f

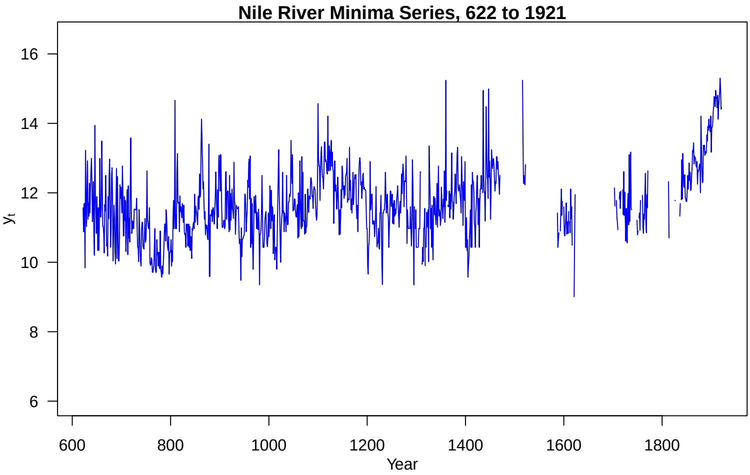


Kalman Recursions for Time Series with Missing Values: I

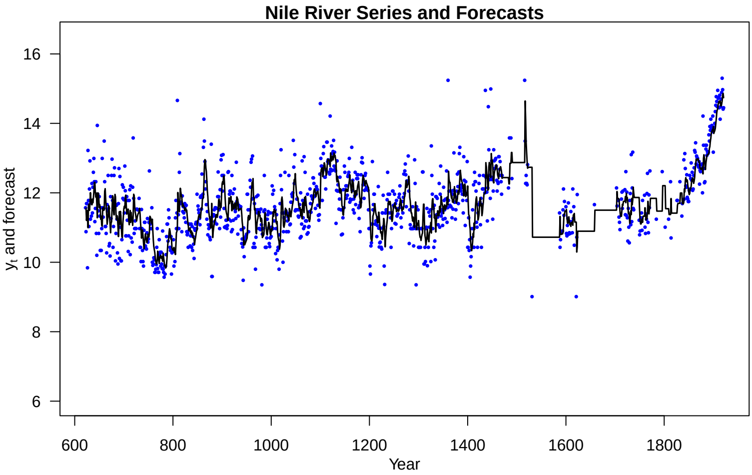
One of the strengths of state-space formulation is the capability to handle time series with missing values. Suppose $Y_1, \cdots, Y_t$ and Y_{t+3} are observed, but not Y_{t+1} and Y_{t+2}:

- use modified recursion (i.e., skip the calculation of the innovation when data is missing)
    - use $\mu^f_{t+1} \overset{\text{def}}{=} X^t_{t+1}$ and $\Sigma^f_{t+1} \overset{\text{def}}{=} \Sigma^t_{t+1}$ for X^t_{t+2} and Σ^t_{t+2}
    - use X^t_{t+2} and Σ^t_{t+2} for X^t_{t+3} and Σ^t_{t+3}
- take X^t_{t+3}, Σ^t_{t+3}, and Y_{t+3} into usual recursion to obtain $\mu^a_{t+3} = X^{t+3}_{t+3}$ and $\Sigma^a_{t+3} = \Sigma^{t+3}_{t+3}$ and $\mu^f_{t+4} = X^{t+3}_{t+4}$ and $\Sigma^f_{t+4} = \Sigma^{t+3}_{t+4}$
- need to interpret "given $t+3$" as conditioning on everything available at time $t+3$, i.e., $Y_1, \cdots, Y_t$ and Y_{t+3}

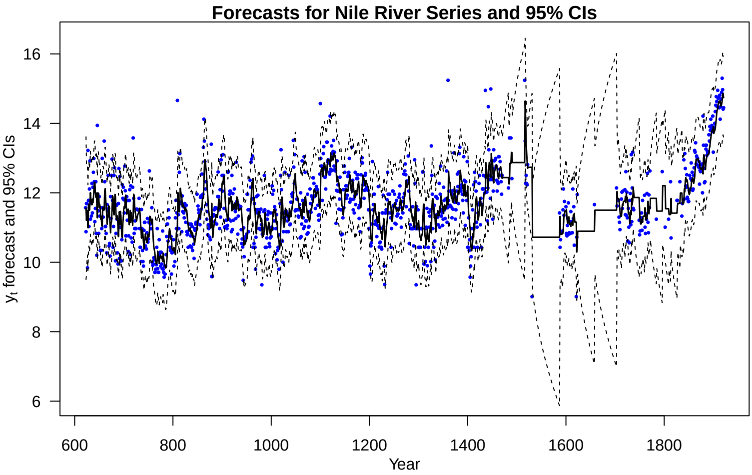
Example: Nile River Annual Minima Series



Nile River Annual Minima Series with Missing Values Imputed



Nile River Annual Minima Series Forecasts with 95 % CI



Kalman Recursions for Smoothing: I

Given time series $Y_1, \cdots, Y_n$, Kalman filter recursions give us $\mu_t^a = X_t^t$ for $t = 1, \cdots, n$

- Regression lemma says solution to smoothing problem is

$$\mu_t^s \stackrel{\text{def}}{=} \mathrm{E}[X_t | \boldsymbol{Y}_{1:n}] = \mu_t + \Sigma_{t,n}^T \Sigma_{Y,n}^{-1} (\boldsymbol{Y}_{1:n} - \boldsymbol{\mu}_{1:n})$$

- MSE for predictor, i.e., $\mathrm{E}\left[(X_t - \mu_t^s)^2\right]$, is

$$\Sigma_t - \Sigma_{t,n}^T \Sigma_{Y,n}^{-1} \Sigma_{t,n} \stackrel{\text{def}}{=} \Sigma_t^s,$$

where $\Sigma_t \stackrel{\text{def}}{=} \mathrm{Var}[X_t]$

Kalman Recursions for Smoothing: II

Using innovation U_t, innovation variance F_t, Kalman gains K_t, forecasts $\mu_t^f \stackrel{\text{def}}{=} X_t^{t-1}$ and associated MSEs $\Sigma_t^f \stackrel{\text{def}}{=} \Sigma_t^{t-1}, t = 1, \cdots, n$ computed by Kalman filter recursions, Kalman smoother recursions allow efficient computation of $\mu_t^s, t = 1, \cdots, n$

The first two steps yield desired predictor μ_t^s

1. Manipulate innovations: starting with $r_n = 0$, recursively form
$$r_{t-1} = \frac{U_t}{F_t} + (1 - K_t) r_t, \quad t = n, \cdots, 1$$
2. Combine manipulated innovations and forecasts:
$$\mu_t^s = X_t^t + \Sigma_t^{t-1} r_{t-1}, \quad t = 1, \cdots, n$$

Kalman Recursions for Smoothing: III

Next two steps yield MSE for predictor X_t^n:

3. Manipulate innovation variances: starting with $N_n = 0$, recursively form

$$N_{t-1} = \frac{1}{F_t} + (1 - K_t)^2 N_t, \quad t = n, \cdots, 1$$

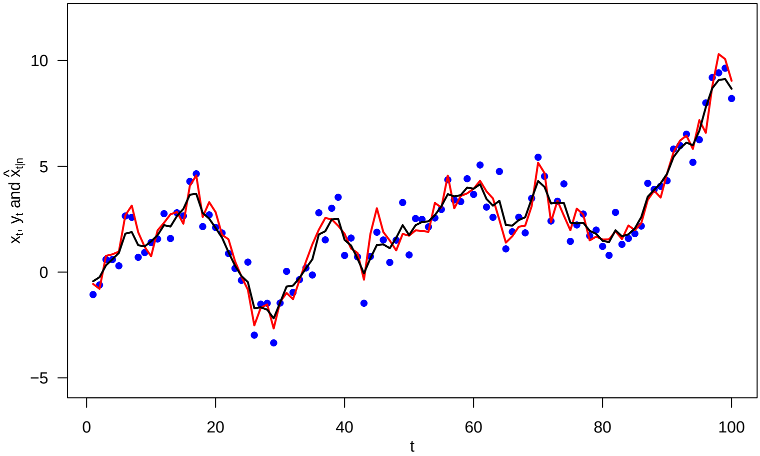
4. Combine manipulated innovation variances and forecast MSEs:

$$\Sigma_t^n = \Sigma_t^{t-1} - \left(\Sigma_t^{t-1}\right)^2 N_{t-1}, \quad t = 1, \cdots, n,$$

where Σ_t^n is the desired MSE

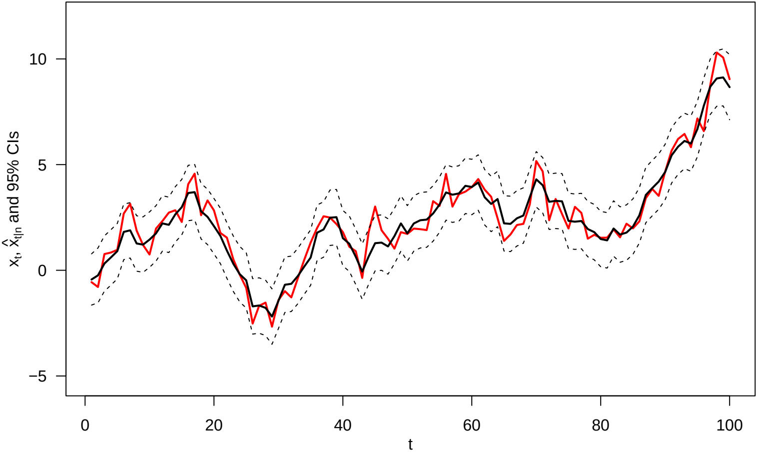
Simulated Example: Local Level Model with SNR $= 2$

Time series Y_t, states X_t, and smooths μ_t^s



Simulated Data from Local Level Model with SNR $= 2$

States X_t, smooths X_t^n, and 95% CIs based on Σ_t^s



Estimating the State-Space Model Parameters

So far, we've assumed that the parameters $\boldsymbol{\theta} = (\sigma_V^2, \sigma_W^2, \mu_0, \sigma_0^2)$ are known. In practice, we need to estimate from the data

This requires maximizing the marginal likelihood of the data $\boldsymbol{y}$, having integrated the latent time series x out. This is given by:

$$f(\boldsymbol{y}|\sigma_V^2, \sigma_W^2, \mu_0, \sigma_0^2) = \int f(\boldsymbol{y}|\boldsymbol{x}, \sigma_W^2) f(\boldsymbol{x}|\mu_0, \sigma_0^2, \sigma_V^2)$$

Maximizing over an integral can be difficult ☹

Direct Maximum Marginal Likelihood

Fortunately, our normal distribution facts tell us that the marginal distribution of $\boldsymbol{y}$ is

$$\boldsymbol{y} \sim \mathrm{N}(\mathrm{E}(\boldsymbol{x}), \mathrm{Var}(\boldsymbol{x}) + \sigma_W^2 I_n).$$

However, the direct evaluation of the marginal likelihood can be challenge due to $n \times n$ matrix inversions

Alternative, we use the innovations $U_t = Y_t - Y_t^{t-1}$ to compute the likelihood:

$$\ell(\boldsymbol{\theta}) \propto f(u_1) \prod_{t=2}^{n} f(u_t | \boldsymbol{y}_{1:t-1}).$$

We can do the following iteratively:

- Pick an initial guess $\hat{\boldsymbol{\theta}}^0$ and run the Kalman filter to get a set of innovations
- Maximizing $\boldsymbol{\theta}$ (e.g., via Newton–Raphson) with $\boldsymbol{u}$ to obtain new estimate of $\boldsymbol{\theta}$

Expectation-Maximization (EM) Maximum Marginal Likelihood

Another way to compute maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ is to use the expectation-maximization (EM) algorithm [Dempster, Laird, and Rubin, 1977]

- Initialize by choosing starting value $\boldsymbol{\theta}^0$, and compute the incomplete likelihood
- Perform the E-step to obtained X_t^n, Σ_t^n
- Perform M-step to update the estimate $\boldsymbol{\theta}$ using the complete likelihood
- Recompute the incomplete likelihood
- Repeat until convergence, i.e., $|\hat{\boldsymbol{\theta}}^N - \hat{\boldsymbol{\theta}}^{N-1}| < \epsilon$

Bayesian Estimation

Markov Chain Monte Carlo (MCMC) methods, such as the Gibbs sampler [Gelfand and Smith, 1990] or the Metropolis-Hastings algorithm [Metropolis et al., 1953; Hastings, 1970], are commonly used for Bayesian inference in state space models

Gibbs Sampler for State Space Models

1. Draw $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta}|\boldsymbol{x}_{0:n}, \boldsymbol{y}_{\mathbf{1:n}})$, where

$$p(\boldsymbol{\theta}|\boldsymbol{x}_{0:n}, \boldsymbol{y}_{1:n}) \propto \pi(\boldsymbol{\theta}) p(x_0|\boldsymbol{\theta}) \prod_{t=1}^{n} p(x_t|x_{t-1}, \boldsymbol{\theta}) p(y_t|x_t, \boldsymbol{\theta})$$

2. Draw $\boldsymbol{x}_{0:n}$ from $p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n}, \boldsymbol{\theta})$, where

$$p(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n}, \boldsymbol{\theta}) = p(x_n|\boldsymbol{y}_{1:n}, \boldsymbol{\theta}) p(x_{n-1}|x_n, \boldsymbol{y}_{1:n-1}, \boldsymbol{\theta}) \cdots p(x_0|x_1, \boldsymbol{\theta})$$

   Use forward-filtering, backward sampling (FFBS) algorithm to sequentially simulating the individual states backward