Bayesian Learning - Lab 03

Anubhav Dikshit (anudi287) and Maximilian Pfundstein (maxpf364)

2019-09-08

Contents

1 Computations with Simulated Data

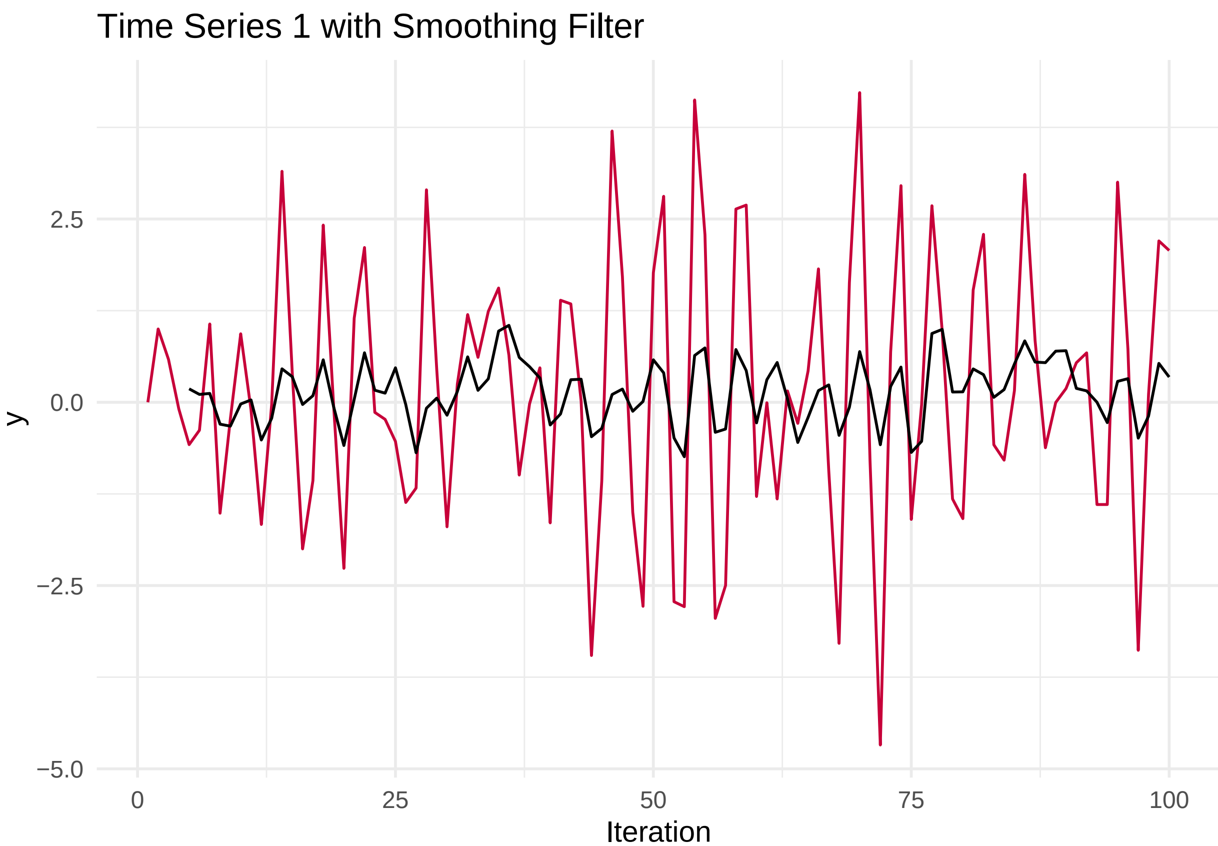
Task a): Generate two time series $x_t = -0.8x_{t-2} + w_t$, where $x_0 = x_1 = 0$ and $x_t = cos(\frac{2\pi t}{5})$ with 100 observations each.

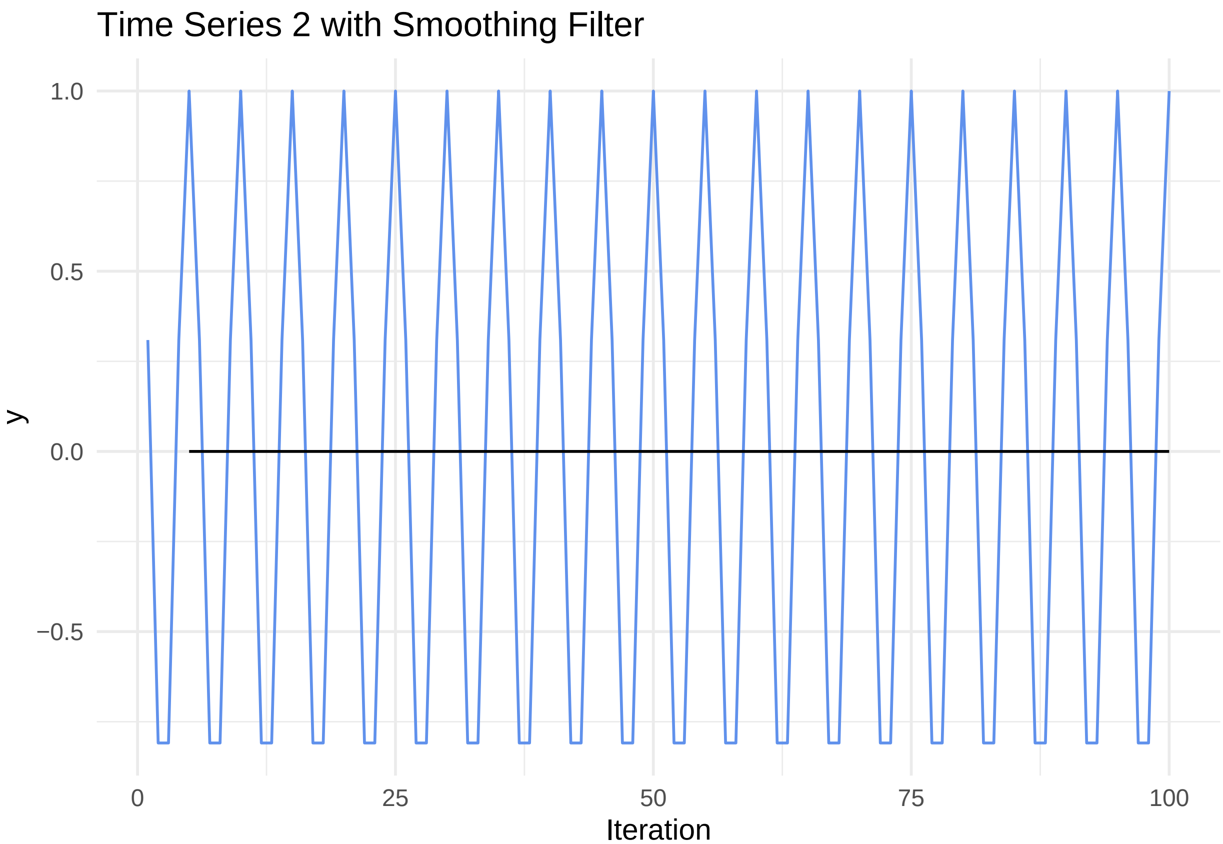
Answer: First we create two functions to sample `n` times from our time series. Then we apply a filter. As default a convoltuion is being used (moving average). With `sides = 1` only past values are considered, which makes sense for a time series.

```
################################################################################
# Exercise 1.a)
################################################################################

x0 = 0
x1 = 0

n = 100

# Series 1
generate_S1 = function(t, x0=0, x1=1) {

  series = vector(length = t)
  series[1] = x0
  series[2] = x1

  for (i in 3:t) {
    series[i] = -0.8 * series[i-2] + rnorm(n=1, mean=0, sd=1)
  }

  return(ts(series))
}

# Series 2
generate_S2 = function(t) {
  series = vector(length = t)
```

```r
  for (i in 1:t) {
    series[i] = cos(2 * pi * i / 5)
  }

  return(ts(series))
}

index = c(1:n)

series1 = generate_S1(n)
series2 = generate_S2(n)

series1_filtered = stats::filter(series1, filter = rep(0.2, 5), sides = 1)
series2_filtered = stats::filter(series2, filter = rep(0.2, 5), sides = 1)
```

As we can see the time series have been smoothed a lot, removing extremes. The first smoothed series seems to be slightly shifted. For the second time series we obtain a straight line, as the moving average of an alternating series will be 0.





Task b): Consider time series $x_t - 4_{t-1} + 2x_{t-2} + x_{t-5} = w_t + 3x_{t-2} + w_{t-4} - 4w_{t-6}$. Write an appropriate R code to investigate whether this time series is casual and invertible.

```r
################################################################################
# Exercise 1.b)
################################################################################

generate_S3 = function(t, X, W) {
  series = vector(length=t)
  white_noise = vector(length=t)

  series[1:length(X)] = X
  white_noise[1:length(W)] = W

  for (i in 7:t) {
    W[1:6] = W[2:7]
    W[7] = rnorm(1, mean=0, sd=1)
    series[i] = 4 * series[i-1] - 2 * series[i-2] - series[i-5] +
                W[7] + 3 * W[5] + W[2] - 4 * W[1]
  }

  return(ts(series))
}

series3 = generate_S3(t=n, X = rnorm(7, mean=0, sd=1), W = rnorm(7, mean=0, sd=1))
```

Causality

First we rewrite the given time series:

$$x_t = 4x_{t-1} - 2x_{t-2} - x_{t-5} + w_t + 3x_{t-2} + w_{t-4} - 4w_{t-6}$$

Applying the autoregressive operator gives us:

$$\phi(B) = 1 - 4B + 2B^2 + 0B^3 + 0B^4 + B^5$$

So Z_ϕ is given by:

$$Z_\phi = (1, -4, 2, 0, 0, 1)$$

We use the function `polyroot()` to see if any of the (complex) zero points lies withon the unit circle.

```
Z_phi = c(1, -4, 2, 0, 0, 1)

isCausal = function(Z) {
  return(all(Mod(polyroot(Z)) > 1))
}

isCausal(Z_phi)
```

```
## [1] FALSE
```

Invertibility

Using the autoregressive operator for θ, we get:

$$\theta(B) = 1 + 0B + 3B^2 + 0B^3 + B^4 + 0B^5 - 4B^6$$

So Z_θ is given by:

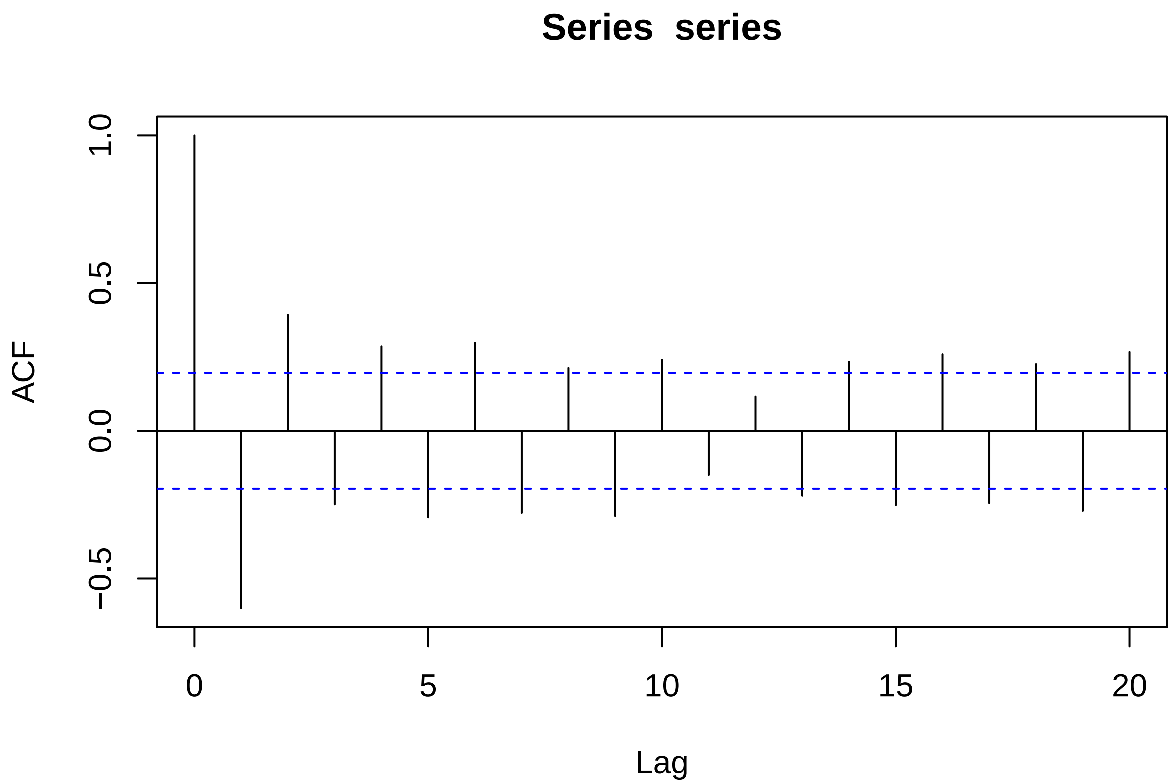
$$Z_\theta = (1, 0, 3, 0, -1, 0, -4)$$

```
Z_theta = c(1, 0, 3, 0, -1, 0, 4)

isInvertible = function(Z) {
  return(all(Mod(poly(Z)) > 1))
}

isInvertible(Z_theta)
```

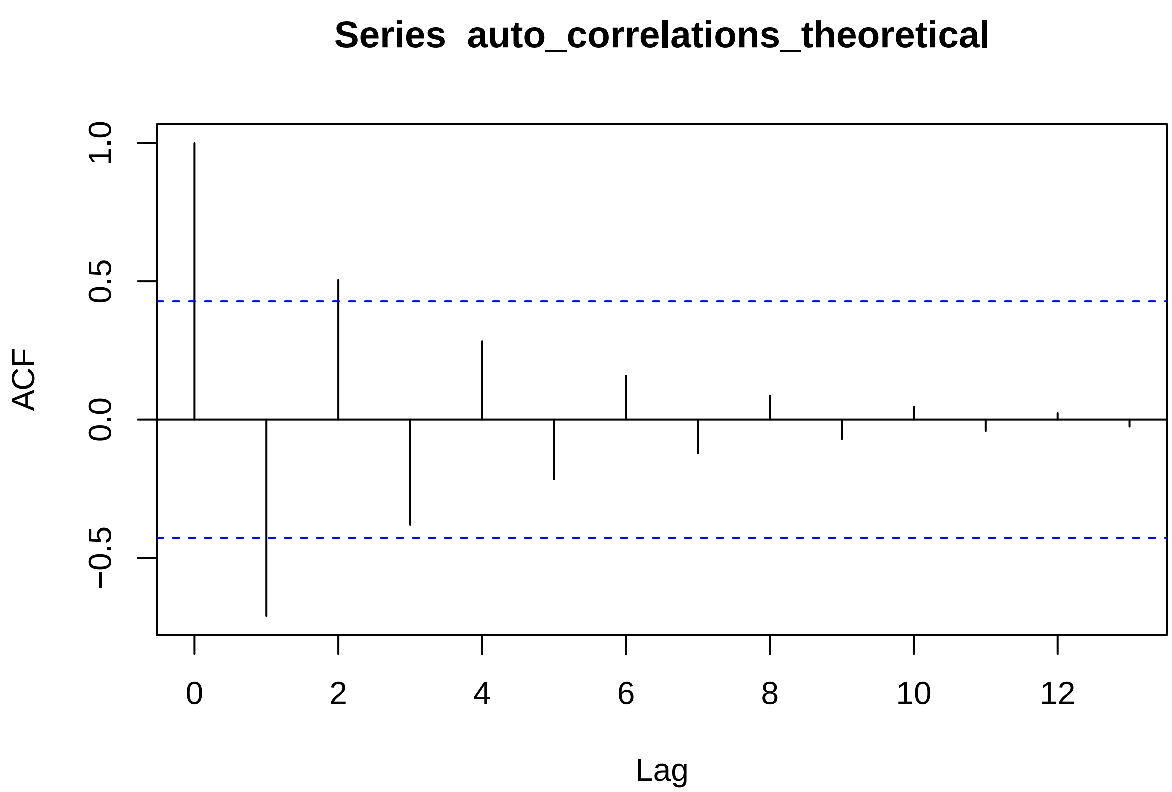
```
## [1] FALSE
```

Task: Use built-in R functions to simulate 100 observations from the process $x_t + \frac{3}{4}x_{t-1} = w_t - \frac{1}{9}w_{t-2}$, compute sample ACF and theoretical ACF, use seed 54321. Compare the ACF plots.

```
set.seed(54321)

model = list(ar = c(-3/4), ma = c(0, -1/9))
series = arima.sim(model = model, n = 100)

# Sample
auto_correlations_sample = acf(series)
```




```r
# Theoretical
auto_correlations_theoretical = ARMAacf(ar = model$ar, ma = model$ma,
                                        lag.max = 20)
acf(auto_correlations_theoretical)
```



We can see that the theoretical AC is between the blue lines after three iterations, while the sample AC exceeds the lines for a longer period of time.

2 Visualization, detrending and residual analysis of Rhine data

The data set `Rhine.csv` contains monthly concentrations of total nitrogen in the Rhine River in the period 1989-2002.

Task a): Import the data to R, convert it appropriately to *ts* object (use function `ts()`) and explore it by plotting the time series, creating scatter plots of x_t against $x_{t-1}, ..., x_{t-12}$. Analyze the time series plot and the scatter plots: Are there any trends, linear or seasonal, in the time series? When during the year is the concentration highest? Are there any special patterns in the data or scatterplots? Does the variance seem to change over time? Which variables in the scatterplots seem to have a significant relation to each other?

Task b): Eliminate the trend by fitting a linear model with respect to t to the time series. Is there a significant time trend? Look at the residual pattern and the sample ACF of the residuals and comment how this pattern might be related to seasonality of the series.

Task c): Eliminate the trend by fitting a kernel smoother with respect to t to the time series (choose a reasonable bandwidth yourself so the fit looks reasonable). Analyze the residual pattern and the sample ACF of the residuals and compare it to the ACF from step b). Conclusions? Do residuals seem to represent a stationary series?

Task d): Eliminate the trend by fitting the following so-called seasonal means model:

$$x_t = \alpha_0 + \alpha_1 t + \beta_1 I(\text{month} = 2) + ... + \beta_{12} I(\text{month} = 12) + w_t$$

where $I(x) = 1$ if x is true and 0 otherwise. Fitting of this model will require you to augment data with a categorical variable showing the current month, and then fitting a usual linear regression. Analyze the residual pattern and the ACF of residuals.

3 Analysis of oil and gas time series

Weekly time series *oil* and *gas* present in the package astsa show the oil prices in dollars per barrel and gas prices in cents per dollar.

Task a): Plot the given time series in the same graph. Do they look like stationary series? Do the processes seem to be related to each other? Motivate your answer.

Task b): Apply log-transform to the time series and plot the transformed data. In what respect did this transformation made the data easier for the analysis?

Task c): To eliminate trend, compute the first difference of the transformed data, plot the detrended series, check their ACFs and analyze the obtained plots. Denote the data obtained here as x_t (oil) and y_t (gas).

Task d): Exhibit scatterplots of x_t and y_t for up to three weeks of lead time of x_t; include a nonparametric smoother in each plot and comment the results: are there outliers? Are the relationships linear? Are there changes in the trend?

Task e): Fit the following model: $y_t = \alpha_0 + \alpha_1 I(x_t > 0) + \beta_1 x_t + \beta_2 x_{t-1} + w_t$ and check which coefficients seem to be significant. How can this be interpreted? Analyze the residual pattern and the ACF of the residuals.

4 Source Code

```
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2)
set.seed(12345)

################################################################################
# Exercise 1.a)
################################################################################

x0 = 0
x1 = 0

n = 100

# Series 1
generate_S1 = function(t, x0=0, x1=1) {

  series = vector(length = t)
  series[1] = x0
  series[2] = x1

  for (i in 3:t) {
    series[i] = -0.8 * series[i-2] + rnorm(n=1, mean=0, sd=1)
  }

  return(ts(series))
}

# Series 2
generate_S2 = function(t) {
  series = vector(length = t)

  for (i in 1:t) {
    series[i] = cos(2 * pi * i / 5)
  }

  return(ts(series))
}

index = c(1:n)

series1 = generate_S1(n)
series2 = generate_S2(n)

series1_filtered = stats::filter(series1, filter = rep(0.2, 5), sides = 1)
series2_filtered = stats::filter(series2, filter = rep(0.2, 5), sides = 1)




df = data.frame(index,
                series1 = as.numeric(series1),
```

```r
                 series2 = as.numeric(series2),
                 series1_filtered = as.numeric(series1_filtered),
                 series2_filtered = as.numeric(series2_filtered))

ggplot(df) +
  geom_line(aes(x = index, y = series1), color = "#C70039") +
  geom_line(aes(x = index, y = series1_filtered), color = "#000000") +
  labs(title = "Time Series 1 with Smoothing Filter", y = "y",
  x = "Iteration", color = "Legend") +
  theme_minimal()

ggplot(df) +
  geom_line(aes(x = index, y = series2), color = "#6091EC") +
  geom_line(aes(x = index, y = series2_filtered), color = "#000000") +
  labs(title = "Time Series 2 with Smoothing Filter", y = "y",
  x = "Iteration", color = "Legend") +
  theme_minimal()


################################################################################
# Exercise 1.b)
################################################################################

generate_S3 = function(t, X, W) {
  series = vector(length=t)
  white_noise = vector(length=t)

  series[1:length(X)] = X
  white_noise[1:length(W)] = W

  for (i in 7:t) {
    W[1:6] = W[2:7]
    W[7] = rnorm(1, mean=0, sd=1)
    series[i] = 4 * series[i-1] - 2 * series[i-2] - series[i-5] +
                W[7] + 3 * W[5] + W[2] - 4 * W[1]
  }

  return(ts(series))
}

series3 = generate_S3(t=n, X = rnorm(7, mean=0, sd=1), W = rnorm(7, mean=0, sd=1))


Z_phi = c(1, -4, 2, 0, 0, 1)

isCausal = function(Z) {
  return(all(Mod(polyroot(Z)) > 1))
}

isCausal(Z_phi)


Z_theta = c(1, 0, 3, 0, -1, 0, 4)
```

```r
isInvertible = function(Z) {
  return(all(Mod(poly(Z)) > 1))
}

isInvertible(Z_theta)


set.seed(54321)

model = list(ar = c(-3/4), ma = c(0, -1/9))
series = arima.sim(model = model, n = 100)

# Sample
auto_correlations_sample = acf(series)

# Theoretical
auto_correlations_theoretical = ARMAacf(ar = model$ar, ma = model$ma,
                                        lag.max = 20)
acf(auto_correlations_theoretical)
```