Characteristics of the ARMA models

MS-C2128 Prediction and Time Series Analysis

Fall term 2021

Week 5: Predicting using ARMA models, Kalman filter and Dynamic regression

1. Prediction
    1. Predicting using ARMA models
    2. Exponential smoothing
2. Kalman filter
3. Dynamic regression

Content

Predicting using ARMA models: Idea

- Consider a time series x_t, $t = \ldots, -2, -1, 0, 1, 2, \ldots$. We wish to predict the value x_{t+s} based on the *observed* values up to the time point t.
- The prediction $\hat{x}_{t+s|t}$ is thus some function of the values $\ldots, x_{t-2}, x_{t-1}, x_t$.
- The goal is to find a function of $\ldots, x_{t-2}, x_{t-1}, x_t$ such that the obtained value is as close as possible to the true value x_{t+s}
- An optimal prediction (in the sense of the mean square error) is obtained from the conditional expected value

$$\hat{x}_{t+s|t} := \mathrm{E}\left[x_{t+s} \mid x_t, x_{t-1}, \ldots\right]$$

Predicting using ARMA models: Idea

Assume that the process $(x_t)_{t\in T}$ has an MA representation

$$x_t = \sum_{j=0}^{\infty} \psi_j L^j \epsilon_t, \quad \psi_0 = 1, \quad \sum_{j=0}^{\infty} |\psi_j| < \infty,$$

and assume that the process $(\epsilon_t)_{t\in T}$ has been observed until the time point t (that is, we have the observations $\epsilon_t, \epsilon_{t-1}, \epsilon_{t-2}\ldots$).

Predicting using ARMA models: Idea

- Now

$$x_{t+s} = \epsilon_{t+s} + \psi_1 \epsilon_{t+s-1} + \ldots + \psi_{s-1} \epsilon_{t+1} + \psi_s \epsilon_t + \psi_{s+1} \epsilon_{t-1} + \ldots$$

and the optimal prediction at the time point t (in the sense of the mean square error) is

$$\hat{x}_{t+s|t} := \mathrm{E}\left[x_{t+s} \mid \epsilon_t, \epsilon_{t-1}, \ldots\right] = \psi_s \epsilon_t + \psi_{s+1} \epsilon_{t-1} + \psi_{s+2} \epsilon_{t-2} + \ldots$$

- The mean square error of the prediction is

$$\mathrm{E}\left[\left(x_{t+s} - \hat{x}_{t+s|t}\right)^2\right] = \left(1 + \psi_1^2 + \psi_2^2 + \ldots + \psi_{s-1}^2\right)\sigma^2.$$

Predicting using ARMA models

Instead of observing the stationary process $(\epsilon_t)_{t\in T}$, one usually observes the process $(x_t)_{t\in T}$,

$$\left(1-\phi_1 L-\ldots-\phi_p L^p\right)x_t=\left(1+\theta_1 L+\ldots+\theta_q L^q\right)\epsilon_t$$

and it is often not reasonable to use the MA(∞) representation for predicting. However, invertibility of the process guarantees that it is irrelevant whether one observes $(x_t)_{t\in T}$ or $(\epsilon_t)_{t\in T}$.

- If the process is invertible, the s-step prediction at time point t is

$$\hat{x}_{t+s|t}=\begin{cases}\phi_1\hat{x}_{t+s-1|t}+\phi_2\hat{x}_{t+s-2|t}+\ldots+\phi_p\hat{x}_{t+s-p|t} \\ \quad +\theta_s\epsilon_t+\theta_{s+1}\epsilon_{t-1}+\ldots+\theta_q\epsilon_{t+s-q} & s=1,2,\ldots,q, \\ \phi_1\hat{x}_{t+s-1|t}+\phi_2\hat{x}_{t+s-2|t}+\ldots+\phi_p\hat{x}_{t+s-p|t} & s\geq q+1,\end{cases}$$

where $\hat{x}_{\tau|t}=x_\tau$, when $\tau\leq t$ and the terms ϵ_t can be calculated recursively using the formula $\epsilon_t=x_t-\hat{x}_{t|t-1}$.

Predicting using ARMA models

Remark

- If one predicts far to the future, that is $s > q$, the prediction does not take the moving average into account as there is not enough data to calculate it.
- Above, it was assumed that we observe infinitely long history of the values x_t. In practice, one can set the unobserved (historical) values to be equal to 0. This works well assuming that there are still plenty of observations and that the coefficients $|\theta_i|$ and $|\psi_i|$ are reasonably small.

Predicting using ARMA models: Optimality

- If the series x_t is a realization from ARMA(p, q) process with known parameters $\phi_1, \phi_2, ..., \phi_p, \theta_1, \theta_2, ..., \theta_q$, then the prediction $\hat{x}_{t+s|t}$ is optimal in the sense of the mean square error

$$\text{MSE}(\hat{x}_{t+s|t}) = \text{E}\left[\left(x_{t+s} - \hat{x}_{t+s|t}\right)^2\right].$$

- In practice, the parameters of the ARMA(p, q) are not known and have to be estimated. Then the mean square error depends on estimations errors and the prediction is not, strictly speaking, optimal anymore. However, it can usually be thought to be close to optimal.

Predicting using ARMA models

- Consider $\hat{x}_{t+s|t}$ as a function of s.
  1. Under pure AR models or ARMA models, $\hat{x}_{t+s|t} \to 0$ exponentially, as $s \to \infty$.
  2. Under MA(q) model, $\hat{x}_{t+s|t}$ is equal to 0 for $s > q$.
- Thus ARMA models are suitable for short term forecasting, but not for long term forecasting!

Exponential smoothing

- Ad-hoc forecasting method, that does not rely on modeling.
- Applied widely.
  - Easy to calculate in practice.
  - Gives reasonably good predictions in practice, but is rarely optimal.

Simple exponential smoothing

- The value of x_{t+1} is predicted using a weighted sum of the previous observation $x_t, x_{t-1}, x_{t-2}, \ldots,$

$$\hat{x}_{t+1|t} = \sum_{i=0}^{\infty} w_i x_{t-i}$$

  - The weights $w_i = \alpha(1-\alpha)^i$, $0 < \alpha < 1$, decrease exponentially. (That is why the method is called exponential smoothing.)
  - The parameter α is called the smoothing factor.
- One can update the prediction using the formula

$$\hat{x}_{t+1|t} = \alpha x_t + (1-\alpha)\hat{x}_{t|t-1} = \alpha\hat{\epsilon}_t + \hat{x}_{t|t-1},$$

where $\hat{\epsilon}_t = x_t - \hat{x}_{t|t-1}$ is the prediction error at time point t.

Simple exponential smoothing

$$\hat{x}_{t+1|t} = \alpha x_t + (1 - \alpha)\hat{x}_{t|t-1}$$

- One can show that simple exponential smoothing gives optimal predictions in the case that x_t is an ARIMA(0,1,1) process:
  - Dx_t is a MA(1) process

$$Dx_t = x_t - x_{t-1} = \epsilon_t + \theta_1 \epsilon_{t-1}, \quad (\epsilon)_{t \in T} \sim WN(0, \sigma^2)$$

  - Choose $\alpha = \theta_1 + 1$

Simple exponential smoothing

$$\hat{x}_{t+1|t} = \alpha x_t + (1-\alpha)\hat{x}_{t|t-1}$$

- One can also show that simple exponential smoothing gives optimal predictions in the case that x_t is a noisy random walk, that is in the case:

$$x_t = m_t + \epsilon_t, \text{where}$$
$$m_t = m_{t-1} + \eta_t, \text{is a random walk}$$
$$(\epsilon_t)_{t\in T} \sim IID(0, \sigma_1^2), \quad (\eta_t)_{t\in T} \sim IID(0, \sigma_2^2)$$

  - Optimal α depends on the signal-noise-ratio $\frac{\text{var}(\epsilon_t)}{\text{var}(\eta_t)}$.
  - The proof relies on using the Kalman filter. (We talk about Kalman filter later.)
- Level m_t (of the random walk) can be estimated from the observations x_t:

$$m_t = \alpha x_t + (1-\alpha)m_{t-1} \quad \text{and} \quad \hat{x}_{t+1|t} = m_t.$$

Double exponential smoothing

Simple exponential smoothing does not provide good predictions when there is a trend in the data. Double exponential smoothing is an extension of the simple exponential smoothing to the cases when there is a trend.

Double exponential smoothing

In double exponential smoothing, on top of predicting, the level m and the trend β are updated:

$$\begin{aligned}\hat{x}_{t+l|t} &= m_t + l\beta_t \\ m_t &= \alpha_1 x_t + (1-\alpha_1)(m_{t-1} + \beta_{t-1}) \\ \beta_t &= \alpha_2(m_t - m_{t-1}) + (1-\alpha_2)\beta_{t-1}.\end{aligned}$$

- One can write

$$\begin{aligned}m_t &= m_{t-1} + \beta_{t-1} + \alpha_1\hat{\epsilon}_t \\ \beta_t &= \beta_{t-1} + \alpha_1\alpha_2\hat{\epsilon}_t,\end{aligned}$$

  where $\hat{\epsilon}_t = x_t - \hat{x}_{t|t-1}$.
- The parameters α are called the data smoothing factors.
- The parameters β are called the trend smoothing factors.
- For well selected α_i, this method is optimal under ARIMA(0,2,2) models.

Exponential smoothing: Comments

- Exponential smoothing is often applied using fixed smoothing parameters.
- Sometimes smoothing parameters are estimated from the data. That usually provides a better fit.
- Applying SARIMA models is recommended, if possible.

Content

Kalman filter

Kalman filter is a tool for predicting $\boldsymbol{x}_{t+1} = (x_{1(t+1)}, ..., x_{k(t+1)})$, when we only observe a noisy version, $\boldsymbol{y}_t = (y_{1t}, ..., y_{dt})$, of $\boldsymbol{x}_t$.

State space representation

Consider a situation where we wish to predict $\boldsymbol{x}_{t+1} = (x_{1(t+1)}, ..., x_{k(t+1)})$, but we only observe $\boldsymbol{y}_t = (y_{1t}, ..., y_{dt})$.
Assume that this system has the following state space representation

$$\boldsymbol{x}_{t+1} = \boldsymbol{F}\boldsymbol{x}_t + \boldsymbol{v}_{t+1}$$
$$\boldsymbol{y}_t = \boldsymbol{H}^\top \boldsymbol{x}_t + \boldsymbol{w}_t$$

where $\boldsymbol{y}_t$ and $\boldsymbol{w}_t$ are d-variate random vectors, $\boldsymbol{x}_{t+1}$ and $\boldsymbol{v}_{t+1}$ are k-variate random vectors and

$$\operatorname{cov}(\boldsymbol{v}_t, \boldsymbol{v}_s) = \begin{cases} \boldsymbol{Q}, & t = s \\ \boldsymbol{0}, & t \neq s \end{cases} \quad \text{and} \quad \operatorname{cov}(\boldsymbol{w}_t, \boldsymbol{w}_s) = \begin{cases} \boldsymbol{R}, & t = s \\ \boldsymbol{0}, & t \neq s. \end{cases}$$

Assume also that $\boldsymbol{v}_t$, $\boldsymbol{w}_t$, and $\boldsymbol{x}_t$ are mutually independent for all $t \geq 1$.

Kalman filter

Kalman filter provides a prediction for $\boldsymbol{x}_{t+1} = (x_{1(t+1)}, \ldots x_{k(t+1)})$, based on the observations $\boldsymbol{y}_t = (y_{1t}, \ldots, y_{dt})$.

- Prediction for $\boldsymbol{x}_{t+1}$ at time point t is the conditional expected value

$$\hat{\boldsymbol{x}}_{t+1|t} := \mathrm{E}[\boldsymbol{x}_{t+1} \mid \boldsymbol{Y}_t], \quad \boldsymbol{Y}_t := (\boldsymbol{y}_t^\top, \ldots, \boldsymbol{y}_1^\top).$$

Kalman filter calculates the predictions $\hat{\boldsymbol{x}}_{1|()}, \hat{\boldsymbol{x}}_{2|1}, \ldots, \hat{\boldsymbol{x}}_{T|T-1}$ recursively and every prediction is associated with the mean square error matrix

$$\boldsymbol{P}_{t+1|t} := \mathrm{E}\left[(\boldsymbol{x}_{t+1} - \hat{\boldsymbol{x}}_{t+1|t})(\boldsymbol{x}_{t+1} - \hat{\boldsymbol{x}}_{t+1|t})^\top\right]$$

Kalman filter: Algorithm

1. The initial values (has to be selected):

$$\hat{\boldsymbol{x}}_{1|()} = \mathrm{E}[\boldsymbol{x}_1]$$
$$\boldsymbol{P}_{1|()} = \mathrm{E}\left[(\boldsymbol{x}_1 - \mathrm{E}[\boldsymbol{x}_1])(\boldsymbol{x}_1 - \mathrm{E}[\boldsymbol{x}_1])^\top\right]$$

2. Recursion formulae for the prediction $\hat{\boldsymbol{x}}_{t+1|t}$ and for the matrix $\boldsymbol{P}_{t+1|t}$ are

$$\hat{\boldsymbol{x}}_{t+1|t} = \boldsymbol{F}\hat{\boldsymbol{x}}_{t|t-1} + \boldsymbol{F}\boldsymbol{P}_{t|t-1}\boldsymbol{H}\left(\boldsymbol{H}^\top \boldsymbol{P}_{t|t-1}\boldsymbol{H} + \boldsymbol{R}\right)^{-1}\left(\boldsymbol{y}_t - \boldsymbol{H}^\top \hat{\boldsymbol{x}}_{t|t-1}\right)$$
$$\boldsymbol{P}_{t+1|t} = \left(\boldsymbol{F} - \boldsymbol{K}_t\boldsymbol{H}^\top\right)\boldsymbol{F}\boldsymbol{P}_{t|t-1}\left(\boldsymbol{F} - \boldsymbol{H}\boldsymbol{K}_t^\top\right) + \boldsymbol{K}_t\boldsymbol{R}\boldsymbol{K}_t^\top + \boldsymbol{Q},$$

   where $\boldsymbol{K}_t$ is the Kalman gain

$$\boldsymbol{K}_t := \boldsymbol{F}\boldsymbol{P}_{t|t-1}\boldsymbol{H}\left(\boldsymbol{H}^\top \boldsymbol{P}_{t|t-1}\boldsymbol{H} + \boldsymbol{R}\right)^{-1}.$$

3. Prediction $\hat{\boldsymbol{y}}_{t+1|t}$ is obtained by applying the formula

$$\hat{\boldsymbol{y}}_{t+1|t} = \boldsymbol{H}^\top \hat{\boldsymbol{x}}_{t+1|t} \qquad \text{ja}$$
$$\mathrm{E}\left[(\boldsymbol{y}_{t+1} - \hat{\boldsymbol{y}}_{t+1|t})(\boldsymbol{y}_{t+1} - \hat{\boldsymbol{y}}_{t+1|t})^\top\right] = \boldsymbol{H}^\top \boldsymbol{P}_{t+1|t}\boldsymbol{H} + \boldsymbol{R}.$$

Kalman filter: Generalization

- It is possible to apply the Kalman filter approach also when one wishes to drop the linearity assumption. Then, the state space representation is given as

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}_t(\boldsymbol{z}_t, \boldsymbol{x}_t) + \boldsymbol{v}_{t+1}$$
$$\boldsymbol{y}_t = \boldsymbol{h}_t(\boldsymbol{x}_t) + \boldsymbol{w}_t,$$

  where $\boldsymbol{x}_{t+1}$, $\boldsymbol{y}_t$, $\boldsymbol{v}_{t+1}$ and $\boldsymbol{w}_t$ are as above, $\boldsymbol{z}_t$ is exogenous, independent of all the other variables, and $\boldsymbol{f}_t$ and $\boldsymbol{h}_t$ are functions that depend on time t , state $\boldsymbol{x}_t$ and on the input $\boldsymbol{z}_t$.
- In this case, it is not as straightforward to obtain predictions as in the linear case, but the predictions are usually very good.

State space representation: MA(1) process

Example

Consider the MA(1) process

$$y_t = \epsilon_t + \theta_1 \epsilon_{t-1}.$$

Define the state vector $\boldsymbol{x}_t$ and the noise $\boldsymbol{v}_{t+1}$ by setting

$$\boldsymbol{x}_t = \begin{bmatrix} \epsilon_t \\ \epsilon_{t-1} \end{bmatrix} \quad \text{and} \quad \boldsymbol{v}_{t+1} = \begin{bmatrix} \epsilon_{t+1} \\ 0 \end{bmatrix}.$$

Then

$$\boldsymbol{x}_{t+1} = \boldsymbol{F}\boldsymbol{x}_t + \boldsymbol{v}_{t+1}, \quad \boldsymbol{F} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

and

$$y_t = \boldsymbol{H}^\top \boldsymbol{x}_t, \quad \text{where} \quad \boldsymbol{H}^\top = \begin{bmatrix} 1 & \theta \end{bmatrix}.$$

Example: GPS navigation

- Assume that there are m_1 satellites that measure the pseudo-distances and their differences between an object and the satellite at time points t. Assume also that there are m_2 support stations that measure the distances between the object and the station. The vector $\boldsymbol{y}_t = (y_{1t}, ..., y_{dt})^\top$, $d = 2m_1 + m_2$, contains the measured distances.
- State vector $\boldsymbol{x}_t$ contains the location coordinates $\boldsymbol{\xi}_t$ and the speed $\boldsymbol{v}_t$ of the object.

$$\boldsymbol{x}_t = \begin{bmatrix} \boldsymbol{\xi}_t \\ \boldsymbol{v}_t \end{bmatrix}.$$

Example: GPS navigation

- Now, the corresponding state space model for the location (and the speed) of the object is

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \boldsymbol{F}\boldsymbol{x}_t + \boldsymbol{u}_t \\ \boldsymbol{y}_t &= \boldsymbol{h}(\boldsymbol{x}_t) + \boldsymbol{w}_t, \end{aligned}$$

  where $\boldsymbol{h}(\boldsymbol{x}_t)$ is a well chosen *nonlinear* function and $\boldsymbol{u}_t$ and $\boldsymbol{w}_t$ are the noise.

Content

Dynamic regression models

In statistics, time series models can be divided into two groups:

1. Autoprojective time series models are models that involve only the time series to be forecasted (e.g. ARMA models).
2. Dynamic regression models are models that may involve the time series to be forecasted and the history of another time series as well.

Static linear regression

Consider the linear time series regression model

$$y_t = \alpha + \beta x_t + \epsilon_t, \quad t \in T,$$

where the response variable y_t and the explanatory variable x_t come from time series and where

1. $\mathrm{E}[\epsilon_t \mid x_t] = 0$, $t \in T$
2. $\mathrm{var}(\epsilon_t \mid x_t) = \sigma^2$, $t \in T$
3. $\mathrm{cor}(\epsilon_t, \epsilon_s \mid x_t, x_s) = 0$, $t \neq s$.

Then the conditional expected value of the response variable y_t conditional on $x_t = x$ is

$$\mathrm{E}[y_t \mid x_t = x] = \alpha + \beta x.$$

Static linear regression

Assume that the explanatory variable x_s is the same constant x for all $s \leq t$:

$$x_s = x, \quad s \leq t$$

Then the conditional expected value of y_s is a constant for all $s \leq t$:

$$\mathrm{E}[y_s \mid x_s = x] = \alpha + \beta x = y.$$

Static linear regression

$$y_t = \alpha + \beta x_t + \epsilon_t, \quad t \in T,$$

- Assume that, at time point $t + 1$, the explanatory variable grows by one unit:
$$x_{t+1} = x + 1.$$

- Then the conditional expected value of the response variable y_{t+1} is
$$\mathrm{E}[y_{t+1} \mid x_{t+1} = x + 1] = \alpha + \beta(x + 1) = y + \beta.$$

- The conditional expected value remains the same until the value of the explanatory variable changes.

Static linear regression

$$y_t = \alpha + \beta x_t + \epsilon_t, \quad t \in T,$$

- The parameter β models the instant change in the conditional expected value of the response variable y_t as the value of x_t changes by one unit.
- The model is static:
  - The conditional expected value of the response variable does not change unless the value of the explanatory variable changes.
  - The conditional expected value of the response variable changes immediately, without any lags, when the value of the explanatory variable changes.

Dynamic regression: Idea

- In dynamic regression, the conditional expected value of the response variable may also changes slowly, or progressively, when the value of the explanatory variable changes.
- A very simple example of a dynamic regression model is the distributed lag model. The distributed lag model is a model for time series data in which a regression equation is used to predict the current values of the response variable based on both, the current and the lagged (past) values of an explanatory variable.

Distributed lag model

A simple distributed lag model is given by

$$y_t = \alpha + \beta_0 x_t + \beta_1 x_{t-1} + ... + \beta_p x_{t-p} + \epsilon_t, \quad t = p+1, p+2, ...,$$

where the residual ϵ_t is white noise on condition $x_t, ..., x_{t-p}$.

- In this model, there are $p+1$ explanatory variables. Those are the values of x_s at time points $t, t-1, ..., t-p$.
- The response variable y_t, at time point t depends on
  - The value of the variable x_s at time point t.
  - The lagged (past) values of the variable x_s.
- The conditional expected value of the response variable y_t is

$$\mathrm{E}[y_t \mid x_t, x_{t-1}, ..., x_{t-p}] = \alpha + \beta_0 x_t + \beta_1 x_{t-1} + ... + \beta_p x_{t-p}.$$

Distributed lag model

$$y_t = \alpha + \beta_0 x_t + \beta_1 x_{t-1} + ... + \beta_p x_{t-p} + \epsilon_t, \quad t = p+1, p+2, ...,$$

Remark

The distributed lag models are stationary with respect to time in the sense that the regression parameters depend only on the corresponding lags, not on the actual time points:

- The regression parameters can be seen as the derivatives of y with respect to x with different lags:

$$\beta_s = \frac{\partial y_t}{\partial x_{t-s}} = \frac{\partial y_{t+s}}{\partial x_t}.$$

- Without this condition, the parameters $\beta_0, ..., \beta_p$ would depend on time, and not only on the length s of the corresponding time interval.

Distributed lag model

$$y_t = \alpha + \beta_0 x_t + \beta_1 x_{t-1} + ... + \beta_p x_{t-p} + \epsilon_t, \quad t = p+1, p+2, ...,$$

- Assume that the explanatory variable x_s is the same constant x for all $s \in \{t-p, ..., t\}$,

$$x_s = x, \quad s \in \{t-p, ..., t\}.$$

- Then the conditional expected value of y_t

$$\mathrm{E}[y_t \mid x_t, x_{t-1}, ..., x_{t-p}] = \alpha + \beta_0 x + \beta_1 x + ... + \beta_p x = \alpha + \beta x = y,$$

where $\beta = \beta_0 + \beta_1 + ... + \beta_p$.

Distributed lag model

Assume that

$$x_s = \begin{cases} x & \text{when } s \in \{t-p, ..., t\} \\ x+1 & \text{when } s = t+1, t+2..., \end{cases}$$

Then

$$\begin{aligned}
\mathrm{E}[y_t \mid x_t, ..., x_{t-p}] &= \alpha + \beta_0 x + \beta_1 x + ... + \beta_p x = \alpha + \beta x = y \\
\mathrm{E}[y_{t+1} \mid x_{t+1}, ..., x_{t-p+1}] &= \alpha + \beta_0 (x+1) + \beta_1 x + ... + \beta_p x = y + \beta_0 \\
\mathrm{E}[y_{t+2} \mid x_{t+2}, ..., x_{t-p+2}] &= \alpha + \beta_0 (x+1) + \beta_1 (x+1) + \beta_2 x + ... + \beta_p x \\
&\vdots \\
\mathrm{E}[y_{t+p+1} \mid x_{t+p}, ..., x_t] &= \alpha + \beta_0 (x+1) + ... + \beta_{p-1} (x+1) + \beta_p x \\
\mathrm{E}[y_{t+p+1} \mid x_{t+p+1}, ..., x_{t+1}] &= \alpha + \beta_0 (x+1) + ... + \beta_p (x+1) = y + \beta
\end{aligned}$$

Distributed lag model

Assume that the value x_s is the same constant $x + 1$ and does not change after the time point $t + 1$. Then the the conditional expected value of y_s

$$\mathrm{E}[y_s \mid x_s, ..., x_{s-p}] = \alpha + \beta_0(x + 1) + ... + \beta_p(x + 1) = y + \beta$$

remains the same for $s \geq t + p + 1$.

Distributed lag model

$$y_t = \alpha + \beta_0 x_t + \beta_1 x_{t-1} + ... + \beta_p x_{t-p} + \epsilon_t, \quad t = p+1, p+2, ...,$$

The regression parameters of the model can be interpreted as follows.

(i) The parameter β_0 models the instant change in the conditional expected value of the response variable y_t as the value of x_t changes by one unit.

(ii) The sum

$$\beta = \beta_0 + \beta_1 + ... + \beta_p$$

models the long term change in the conditional expected value of the response variable y_t as the value of x_t changes by one unit.

Distributed lag model: Parameter estimation

The parameters of the distributed lag model

$$y_t = \alpha + \beta_0 x_t + \beta_1 x_{t-1} + ... + \beta_p x_{t-p} + \epsilon_t, \quad t = p+1, p+2, ...,$$

can be estimated using classic linear regression.

- Problems:
  - If the autocorrelations of the process $(x_t)_{t \in T}$ are non-zero, the explanatory variables of the regression model are multicollinear.
  - If we have n observations of the series $(x_t)_{t \in T}$, we have $n - p$ observations for estimating $p + 2$ parameters.

Distributed lag model: General form

In distributed lag models, y_t might depend on the entire history of the explanatory process, and the explanatory process may be k-variate. That is

$$y_t = \alpha + \sum_{i=0}^{\infty} \beta_{1i} x_{1(t-i)} + \sum_{i=0}^{\infty} \beta_{2i} x_{2(t-i)} + \ldots + \sum_{i=0}^{\infty} \beta_{ki} x_{k(t-i)} + \epsilon_t,$$

where $(\epsilon_t)_{y \in T} \sim WN(0, \sigma^2)$.

- If infinitely many of the regression parameters β_{ji} differ from zero (for example, $\beta_{ji} = \beta_{j0}\delta_j^i$, $|\delta_j| < 1$), then a change in the value of the explanatory variable has an effect on the conditional expected value of y_t infinitely long after that.

Simple ARMAX model

An ARMAX model is a combination of an ARMA model and a linear regression model, where the response process y_t depends on

- the history of the autoregressive part of the process
- noise
- an exogenous variable x_t.

$$y_t = \epsilon_t + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \sum_{i=0}^{b} \eta_i x_{t-i},$$

where

- $(\epsilon_t)_{t \in T} \sim WN(0, \sigma^2)$,
- $\phi_1, ..., \phi_p$ are the parameters of the AR part,
- $\theta_1, ..., \theta_q$ are the parameters of the MA part, and
- $\eta_0, ..., \eta_b$ are the parameters related to the exogenous variable x_t.

ARMAX model

As when applying ARMA models, also when applying ARMAX models, one can model seasonal changes. In ARMAX models, the number of exogenous variables is not limited to one. An ARMAX model, in general form, can be given as

$$\Phi(L)y_t = \Theta(L)\epsilon_t + \sum_{j=1}^{k} H_j(L)x_{jt},$$

where $\Phi(L), \Theta(L)$ and $H_i(L)$ are similar lag polynomials as in the case of ARMA models and the $x_{j.} = (x_{j1}, x_{j2}, ...x_{jt})$ are the observed explanatory variables.

ARIMAX model

An ARIMAX model is like an ARIMA model, but it contains one or more exogenous variables.

- The differences $D^h y_t$ (for some order h) of an ARIMAX model form an ARMAX model and the parameters for the differences can be estimated similarly as in the case of ARIMA models.

R: `arima()`, `arimax()`.

- ARIMAX models are reasonably general, but they are not suitable for long term forecasting. Moreover, ARIMAX models are linear on the explanatory variables.
- In general, we may consider models

$$f(\boldsymbol{y}_t) = g\big((\boldsymbol{y}_s)_{\{s \le t-1\}}, (\boldsymbol{x}_s)_{\{s \le t\}}\big) + \boldsymbol{\epsilon}_t,$$

  where the d-variate $\boldsymbol{y}_t = (y_{1t}, ..., y_{dt})$ depends on its own history and on the history of the k-variate time series $(\boldsymbol{x}_t)_{t \in T}$, $\boldsymbol{x}_t = (x_{1t}, ..., x_{kt})$, $t \in T$, through some nonlinear functions f and g.
- In practice, we often have to rely on these general models, but developing theory for them is very difficult.
  - One has to know (or estimate) the functions f and g and then proceed case by case.

References:

1. Brockwell, P., Davis, R. (2009): Time Series – Theory and Methods, Springer
2. Hamilton, J. (1994): Time Series Analysis, Princeton University Press
3. Ali-Löytty, S. (2004): Kalmanin suodatin ja sen laajennukset paikannuksessa, Diplomityö, TTY

Next week:

- Guest lecturer
- Summary