

Estimation of Space-Time ARMAX Model

Dongping Fang¹

¹Zurich Insurance North America, 1299 Zurich Way,
Schaumburg, IL 60196

Abstract

Space-time Autoregressive–moving-average model with exogenous inputs (STARMAX) models how a time-space target field depends on its own and its neighbors' past, and on other time-space exogenous inputs (predictor fields). Time-space data arise in many physical science studies. The interpretability of STARMAX model makes it highly desirable.

There is not a ready to use program that will fit this model. This paper extends the model by introducing the similar time-space lag structure to predictor fields. This paper also develops and implements a fast Kalman filter parameter estimation and prediction procedure that allows missing values in target process, and can be applied on big data.

Key Words: space-time, Kalman filter

1. Introduction

This is an extension of STARMA model in Pfeifer and Deutsch (1980) and space-time ARMAX model in David Stoffer (1986). The model considered here allows predictor fields and applies time and space lagged relationship on both target and predictor fields.

In fitting the model, we developed an R package that takes a very similar approach as that in R package *starma* (Cheyssou 2016).

2. Space-Time ARMAX Model

2.1 Model $STARMAX(p_{\lambda_1, \dots, \lambda_p}, q_{m_1, \dots, m_q}, \{r_j: u_{j0}, \dots, u_{jr_j}; j = 1, \dots, J\})$

Let $\mathbf{y}(t)$ be the target field and $\mathbf{x}_j(t)$, $j = 1, \dots, J$, be the j th predictor field at time t . Each of them is a vector of n elements representing n space locations.

$$\mathbf{y}(t) = \sum_{k=1}^p \sum_{l=0}^{\lambda_k} \phi_{kl} W^{(l)} \mathbf{y}(t-k) + \sum_{k=1}^q \sum_{l=0}^{m_k} \theta_{kl} W^{(l)} \boldsymbol{\varepsilon}(t-k) + \sum_{j=1}^J \sum_{k=0}^{r_j} \sum_{l=0}^{u_{jk}} \beta_{jkl} W^{(l)} \mathbf{x}_j(t-k) + \boldsymbol{\varepsilon}(t) \quad (1)$$

where

- p : the maximum AR time lag,
- λ_k : the spatial lag for AR time lag k ,
- q : the maximum MA time lag,
- m_k : the spatial lag for MA time lag k ,
- r_j : the maximum time lag for j th predictor,
- u_{jk} : the spatial lag for j th predictor time lag k ($k=0$ is allowed),

$\boldsymbol{\varepsilon}(t)$: the random error vector at time t , $\mathbf{Var}(\boldsymbol{\varepsilon}(t)) = \Sigma$,
 $W^{(l)}$: the space lag operator (more later).

2.1.1 Spatial lag operator

$W^{(l)}$ is the l -th spatial lag operator. It is a matrix of n by n where

- $w_{ij}^{(l)} \neq 0$ only if location j is a l -th order neighbor of location i
- Row sum $\sum_{j=1}^n w_{ij}^{(l)} = 1$
- 0th order: $W^{(0)} = I$
- l -th order: $W^{(l)}\mathbf{y}(t)$ is a vector with the i th element $(\sum_{j=1}^n w_{ij}^{(l)} y_j(t))$ being the weighted average of l -th order neighbors of location i
- $W^{(l)}$ can be asymmetric. Asymmetric $W^{(l)}$ is useful in directional locations such as high way traffic

2.1.2 Parameters

AR parameter matrix: a matrix with row for time lag, column for space lag (starting from 0 for space lag).

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_{10} & \cdots & \phi_{1\lambda_{ar}} \\ \vdots & \ddots & \vdots \\ \phi_{p0} & \cdots & \phi_{p\lambda_{ar}} \end{pmatrix}_{p \times (\lambda_{ar}+1)} = \begin{pmatrix} \boldsymbol{\phi}'_1 \\ \vdots \\ \boldsymbol{\phi}'_p \end{pmatrix}, \text{ where } \lambda_{ar} = \max(\lambda_1, \dots, \lambda_p)$$

MA parameter matrix: a matrix with row for time lag, column for space lag (starting from 0 for space lag).

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_{10} & \cdots & \theta_{1m_{ma}} \\ \vdots & \ddots & \vdots \\ \theta_{q0} & \cdots & \theta_{qm_{ma}} \end{pmatrix}_{q \times (m_{ma}+1)} = \begin{pmatrix} \boldsymbol{\theta}'_1 \\ \vdots \\ \boldsymbol{\theta}'_q \end{pmatrix}, \text{ where } m_{ma} = \max(m_1, \dots, m_q)$$

Predictor parameter matrix: start from 0 for both time and space lag.

$$\boldsymbol{\beta}^{(j)} = \begin{pmatrix} \beta_{00}^{(j)} & \cdots & \beta_{0u_j}^{(j)} \\ \vdots & \ddots & \vdots \\ \beta_{r_j 0}^{(j)} & \cdots & \beta_{r_j u_j}^{(j)} \end{pmatrix}_{(r_j+1) \times (u_j+1)} = \begin{pmatrix} \boldsymbol{\beta}'_0^{(j)} \\ \vdots \\ \boldsymbol{\beta}'_{r_j}^{(j)} \end{pmatrix}, \text{ where } u_j = \max(u_{j0}, \dots, u_{jr_j})$$

3. Kalman filter

Kalman Filter is described here briefly because it will be used in this paper to both fit and predict the model.

3.1 State-space form

The models of our interest can be written in the following state-space form.

$$\begin{aligned} \text{State equation:} & \quad \mathbf{z}_t = \mathbf{F}\mathbf{z}_{t-1} + \boldsymbol{\eta}_t + \boldsymbol{\Gamma}\mathbf{u}_t \\ \text{Observation equation:} & \quad \mathbf{y}_t = \mathbf{M}_t\mathbf{z}_t + \mathbf{v}_t \end{aligned}$$

where

$$\begin{aligned}
 E(\mathbf{u}_t) &= \mathbf{0}, E(\mathbf{v}_t) = \mathbf{0}, \\
 \text{var}(\mathbf{u}_t) &= \mathbf{U}, \text{var}(\mathbf{v}_t) = \mathbf{V} \\
 \text{cov}(\mathbf{u}_t, \mathbf{u}_\tau) &= \mathbf{0}, \text{cov}(\mathbf{v}_t, \mathbf{v}_\tau) = \mathbf{0}, \text{ if } t \neq \tau, \\
 \text{cov}(\mathbf{u}_t, \mathbf{v}_\tau) &= \mathbf{0} \text{ for any } t, \tau
 \end{aligned}$$

3.2 Kalman filter

Let's denote

$D_t = \{\mathbf{y}_\tau : \tau \leq t\}$: the data known upto and including time t .
 One-step ahead prediction and prediction variance of state vector:

$$\begin{aligned}
 \hat{\mathbf{z}}_{t|t-1} &= E(\mathbf{z}_t | D_{t-1}), \\
 \mathbf{P}_{t|t-1} &= E\left((\mathbf{z}_t - \hat{\mathbf{z}}_{t|t-1})(\mathbf{z}_t - \hat{\mathbf{z}}_{t|t-1})' | D_{t-1}\right)
 \end{aligned}$$

Current prediction of state vector:

$$\begin{aligned}
 \hat{\mathbf{z}}_{t|t} &= E(\mathbf{z}_t | D_t), \\
 \mathbf{P}_{t|t} &= E\left((\mathbf{z}_t - \hat{\mathbf{z}}_{t|t})(\mathbf{z}_t - \hat{\mathbf{z}}_{t|t})' | D_t\right)
 \end{aligned}$$

Predicting equation

State vector

$$\begin{aligned}
 - \hat{\mathbf{z}}_{t|t-1} &= \mathbf{F}\hat{\mathbf{z}}_{t-1|t-1} + \boldsymbol{\eta}_t \\
 - \mathbf{P}_{t|t-1} &= \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}' + \boldsymbol{\Gamma}\mathbf{U}\boldsymbol{\Gamma}'
 \end{aligned}$$

Observation vector

$$\begin{aligned}
 - \hat{\mathbf{y}}_{t|t-1} &= \mathbf{M}_t\hat{\mathbf{z}}_{t|t-1} \\
 - E(\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1})(\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1})' &= \mathbf{M}_t\mathbf{P}_{t|t-1}\mathbf{M}_t' + \mathbf{V}
 \end{aligned}$$

Updating equation

$$\begin{aligned}
 - \hat{\mathbf{z}}_{t|t} &= \hat{\mathbf{z}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}) \\
 - \mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t\mathbf{M}_t)\mathbf{P}_{t|t-1}
 \end{aligned}$$

$$\text{where } \mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{M}_t'(\mathbf{M}_t\mathbf{P}_{t|t-1}\mathbf{M}_t' + \mathbf{V}_t)^{-1}$$

3.3 Missing values in target variable

When there are missings in target field \mathbf{y}_t , we take the similar approach as that in the paper by Kohn and Ansley (1986). We also assume that there are NO missings in predictor fields.

Suppose that at time t , among the n locations only $d_t \leq n$ locations have observations. Let $\mathbf{L}_t = \{l_1, l_2, \dots, l_{d_t}\}$ denote the d_t locations at time t ; \mathbf{S}_t be the indicator matrix with d_t rows and n columns such that $\mathbf{S}_t(i, l_i) = 1$ at location with observation, and = 0 if missing. If there are no missings, $\mathbf{S}_t = \mathbf{I}$. Let $\mathbf{y}_t^* = \mathbf{S}_t\mathbf{y}_t$, $\mathbf{M}_t^* = \mathbf{S}_t\mathbf{M}$, $\mathbf{v}_t^* = \mathbf{S}_t\mathbf{v}_t$. The State-space form when there are missing becomes

$$\begin{aligned}
 \text{State equation: } \mathbf{z}_t &= \mathbf{F}\mathbf{z}_{t-1} + \boldsymbol{\eta}_t + \boldsymbol{\Gamma}\mathbf{u}_t \\
 \text{Observation equation: } \mathbf{y}_t^* &= \mathbf{M}_t^*\mathbf{z}_t + \mathbf{v}_t^*
 \end{aligned}$$

Note that only observation equation is changed. Replace $\mathbf{y}_t, \mathbf{M}_t, \mathbf{v}_t$ by $\mathbf{y}_t^*, \mathbf{M}_t^*, \mathbf{v}_t^*$ in the Kalman recursion, everything should work.

4. Parameter Estimates

Kalman filter is a recursive algorithm commonly used in time series analysis for prediction. Kalman filter can also be used to directly estimate the parameters if using the appropriate state space form of the model by setting up the parameter vector as the state vector. The algorithm we will use are based on the article by Cipra and Motyková (1987).

Model in eq (1) can be rewritten as: $\mathbf{y}_t = \mathbf{X}_t \mathbf{b} + \boldsymbol{\varepsilon}_t$ with $\mathbf{r}(\boldsymbol{\varepsilon}(t)) = \Sigma$, \mathbf{b} is the parameter vector including all parameters

$$\mathbf{b}' = (\boldsymbol{\phi}'_1, \dots, \boldsymbol{\phi}'_p, \boldsymbol{\theta}'_1, \dots, \boldsymbol{\theta}'_q, \boldsymbol{\beta}'_0^{(1)}, \dots, \boldsymbol{\beta}'_{r_1}^{(1)}, \dots, \boldsymbol{\beta}'_0^{(j)}, \dots, \boldsymbol{\beta}'_{r_j}^{(j)}),$$

and \mathbf{X}_t is a matrix with following columns corresponding to AR, MA and predictors components

$$\begin{aligned} \mathbf{X}_t = & (W^{(0)}\mathbf{y}_{t-1}, W^{(1)}\mathbf{y}_{t-1}, \dots, W^{(\lambda_{ar})}\mathbf{y}_{t-1}, \dots, W^{(0)}\mathbf{y}_{t-p}, \dots, W^{(\lambda_{ar})}\mathbf{y}_{t-p}, \\ & W^{(0)}\boldsymbol{\varepsilon}_{t-1}, \dots, W^{(m_{ma})}\boldsymbol{\varepsilon}_{t-1}, \dots, W^{(0)}\boldsymbol{\varepsilon}_{t-q}, \dots, W^{(m_{ma})}\boldsymbol{\varepsilon}_{t-q}, \\ & W^{(0)}\mathbf{x}_{1,t}, \dots, W^{(u)}\mathbf{x}_{1,t}, \dots, W^{(0)}\mathbf{x}_{1,t-r_1}, \dots, W^{(u_1)}\mathbf{x}_{1,t-r_1}, \\ & \dots \\ & W^{(0)}\mathbf{x}_{j,t}, \dots, W^{(u_j)}\mathbf{x}_{j,t}, \dots, W^{(0)}\mathbf{x}_{1,t-r_j}, \dots, W^{(u_j)}\mathbf{x}_{1,t-r_j}) \end{aligned}$$

State space representation for parameter estimation

$$\text{State equation: } \mathbf{b}_{t+1} = \mathbf{b}_t$$

$$\text{Obs equation: } \mathbf{y}_t = \mathbf{X}_t \mathbf{b}_t + \boldsymbol{\varepsilon}_t$$

where \mathbf{b}_t is the parameter estimate of \mathbf{b} at time t given all information up to and including time t

Apply the Kalman filter on above state space representation

For time $t = 1, \dots, T$

$$- \text{ Denote } \mathbf{b}_t = \hat{\mathbf{b}}_{t|t}, \mathbf{P}_t = \mathbf{P}_{t|t}$$

- Given \mathbf{b}_t & \mathbf{P}_t , when new \mathbf{y}_{t+1} & \mathbf{X}_{t+1} are available, update by

$$\mathbf{b}_{t+1} = \mathbf{b}_t + \mathbf{K}_{t+1}(\mathbf{y}_{t+1} - \mathbf{X}_{t+1}\mathbf{b}_t)$$

$$\mathbf{P}_{t+1} = \mathbf{P}_t - \mathbf{K}_{t+1}\mathbf{X}_{t+1}\mathbf{P}_t$$

$$\text{where } \mathbf{K}_{t+1} = \mathbf{P}_t \mathbf{X}'_{t+1} (\mathbf{X}_{t+1} \mathbf{P}_t \mathbf{X}'_{t+1} + \boldsymbol{\Sigma}_{t+1})^{-1}$$

Final estimate: $\mathbf{b} = \mathbf{b}_T$

5. Prediction

Here parameters are known and we want to predict into future using the STARMAX model. The state space form used to estimate the parameters is not appropriate to do prediction since we need the parameters to be fixed at each time step when we do prediction.

Model in eq (1) can be written as

$$\text{State equation: } \mathbf{z}_t = \mathbf{F}\mathbf{z}_{t-1} + \boldsymbol{\eta}_t + \boldsymbol{\Gamma}\boldsymbol{\varepsilon}_t$$

$$\text{Observation equation: } \mathbf{y}_t = \mathbf{M}\mathbf{z}_t$$

where

$$\Gamma = \begin{pmatrix} W^{(0)} \\ W^{(1)} \\ \vdots \\ W^{(\lambda_{ar})} \\ \mathbf{0} \\ \mathbf{0} \\ W^{(0)} \\ W^{(1)} \\ \vdots \\ W^{(m_{ma})} \\ \mathbf{0} \\ \mathbf{0} \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \eta_t = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ W^{(0)}\mathbf{x}_{1,t+1} \\ \vdots \\ W^{(u_1)}\mathbf{x}_{1,t+1} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ W^{(0)}\mathbf{x}_{j,t+1} \\ \vdots \\ W^{(u_j)}\mathbf{x}_{j,t+1} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{M} = (I_n, \mathbf{0}, \dots, \mathbf{0}),$$

and z, F are in very complicated form

$$\mathbf{z}_{t-1} = \begin{pmatrix} \mathbf{z}_{ar,t-1}^{(1)} \\ \mathbf{z}_{ar,t-1}^{(2)} \\ \mathbf{z}_{ar,t-1}^{(3)} \\ \mathbf{z}_{ma,t-1}^{(1)} \\ \mathbf{z}_{ma,t-1}^{(2)} \\ \mathbf{z}_{ma,t-1}^{(3)} \\ \mathbf{z}_{x_1,t-1}^{(1)} \\ \mathbf{z}_{x_1,t-1}^{(2)} \\ \mathbf{z}_{x_1,t-1}^{(3)} \\ \vdots \\ \mathbf{z}_{x_j,t-1}^{(1)} \\ \mathbf{z}_{x_j,t-1}^{(2)} \\ \mathbf{z}_{x_j,t-1}^{(3)} \end{pmatrix}$$

where

$$\mathbf{z}_{ar,t-1}^{(1)} = \begin{pmatrix} W^{(0)}\mathbf{y}_{t-1} \\ \vdots \\ W^{(\lambda_{ar})}\mathbf{y}_{t-1} \end{pmatrix}, \quad \mathbf{z}_{ar,t-1}^{(2)} = \begin{pmatrix} W^{(0)}\mathbf{y}_{t-2} \\ \vdots \\ W^{(\lambda_{ar})}\mathbf{y}_{t-2} \\ \vdots \\ W^{(0)}\mathbf{y}_{t-p+1} \\ \vdots \\ W^{(\lambda_{ar})}\mathbf{y}_{t-p+1} \end{pmatrix}, \quad \mathbf{z}_{ar,t-1}^{(3)} = \begin{pmatrix} W^{(0)}\mathbf{y}_{t-p} \\ \vdots \\ W^{(\lambda_{ar})}\mathbf{y}_{t-p} \end{pmatrix}$$

$$\mathbf{z}_{ma,t-1}^{(1)} = \begin{pmatrix} W^{(0)} \boldsymbol{\varepsilon}_{t-1} \\ \vdots \\ W^{(m_{ma})} \boldsymbol{\varepsilon}_{t-1} \end{pmatrix}, \mathbf{z}_{ma,t-1}^{(2)} = \begin{pmatrix} W^{(0)} \boldsymbol{\varepsilon}_{t-2} \\ \vdots \\ W^{(m_{ma})} \boldsymbol{\varepsilon}_{t-2} \\ W^{(0)} \boldsymbol{\varepsilon}_{t-q+1} \\ \vdots \\ W^{(m_{ma})} \boldsymbol{\varepsilon}_{t-q+1} \end{pmatrix}, \mathbf{z}_{ma,t-1}^{(3)} = \begin{pmatrix} W^{(0)} \boldsymbol{\varepsilon}_{t-q} \\ \vdots \\ W^{(m_{ma})} \boldsymbol{\varepsilon}_{t-q} \end{pmatrix},$$

$$\mathbf{z}_{x_j,t-1}^{(1)} = \begin{pmatrix} W^{(0)} \mathbf{x}_{j,t} \\ \vdots \\ W^{(u_j)} \mathbf{x}_{j,t} \end{pmatrix}, \mathbf{z}_{x_j,t-1}^{(2)} = \begin{pmatrix} W^{(0)} \mathbf{x}_{j,t-1} \\ \vdots \\ W^{(u_j)} \mathbf{x}_{j,t-1} \\ W^{(0)} \mathbf{x}_{j,t-r_j+1} \\ \vdots \\ W^{(u_j)} \mathbf{x}_{j,t-r_j+1} \end{pmatrix}, \mathbf{z}_{x_j,t-1}^{(3)} = \begin{pmatrix} W^{(0)} \mathbf{x}_{j,t-r_j} \\ \vdots \\ W^{(u_j)} \mathbf{x}_{j,t-r_j} \end{pmatrix},$$

and

$$F = \begin{pmatrix} F_{ar}^{(1)} & F_{ar}^{(2)} & F_{ar}^{(3)} & F_{ma}^{(1)} & F_{ma}^{(2)} & F_{ma}^{(3)} & F_{x_1}^{(1)} & F_{x_1}^{(2)} & F_{x_1}^{(3)} & \dots & F_{x_j}^{(1)} & F_{x_j}^{(2)} & F_{x_j}^{(3)} \\ I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{(m_{ma}+1)n \times (\lambda_{ar}+1)n} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}_{(u_1+1)n \times (m_{ma}+1)n} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & I & \mathbf{0} \end{pmatrix}$$

where

$$F_{ar}^{(1)} = \boldsymbol{\phi}'_1 \otimes \begin{pmatrix} W^{(0)} \\ W^{(1)} \\ \vdots \\ W^{(\lambda_{ar})} \end{pmatrix}, F_{ar}^{(2)} = (\boldsymbol{\phi}'_2, \dots, \boldsymbol{\phi}'_{p-1}) \otimes \begin{pmatrix} W^{(0)} \\ W^{(1)} \\ \vdots \\ W^{(\lambda_{ar})} \end{pmatrix}, F_{ar}^{(3)} = \boldsymbol{\phi}'_p \otimes \begin{pmatrix} W^{(0)} \\ W^{(1)} \\ \vdots \\ W^{(\lambda_{ar})} \end{pmatrix},$$

$$F_{ma}^{(1)} = \boldsymbol{\theta}'_1 \otimes \begin{pmatrix} W^{(0)} \\ W^{(1)} \\ \vdots \\ W^{(\lambda_{ar})} \end{pmatrix}, F_{ma}^{(2)} = (\boldsymbol{\theta}'_2, \dots, \boldsymbol{\theta}'_{q-1}) \otimes \begin{pmatrix} W^{(0)} \\ W^{(1)} \\ \vdots \\ W^{(\lambda_{ar})} \end{pmatrix}, F_{ma}^{(3)} = \boldsymbol{\theta}'_q \otimes \begin{pmatrix} W^{(0)} \\ W^{(1)} \\ \vdots \\ W^{(\lambda_{ar})} \end{pmatrix},$$

$$F_{x_j}^{(1)} = \boldsymbol{\beta}'_{j,0} \otimes \begin{pmatrix} W^{(0)} \\ W^{(1)} \\ \vdots \\ W^{(\lambda_{ar})} \end{pmatrix}, F_{x_j}^{(2)} = (\boldsymbol{\beta}'_{j,1}, \dots, \boldsymbol{\beta}'_{j,r_j-1}) \otimes \begin{pmatrix} W^{(0)} \\ W^{(1)} \\ \vdots \\ W^{(\lambda_{ar})} \end{pmatrix}, F_{x_j}^{(3)} = \boldsymbol{\beta}'_{j,r_j} \otimes \begin{pmatrix} W^{(0)} \\ W^{(1)} \\ \vdots \\ W^{(\lambda_{ar})} \end{pmatrix},$$

State-space form when there are missing

State equation: $\mathbf{z}_t = \mathbf{F} \mathbf{z}_{t-1} + \boldsymbol{\eta}_t + \boldsymbol{\Gamma} \boldsymbol{\varepsilon}_t$

Observation equation: $\mathbf{y}_t^* = \mathbf{M}_t^* \mathbf{z}_t = \mathbf{z}_t [\mathbf{L}_t]$

where \mathbf{L}_t and S_t are the same as before.

Kalman filter prediction

Predicting equation

State vector

$$\begin{aligned} \mathbf{z}_{t|t-1} &= \mathbf{F}\mathbf{z}_{t-1|t-1} + \boldsymbol{\eta}_t \\ \mathbf{P}_{t|t-1} &= \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}' + \boldsymbol{\Gamma}\boldsymbol{\Sigma}\boldsymbol{\Gamma}' \end{aligned}$$

Observation vector

$$\begin{aligned} \hat{\mathbf{y}}_{t|t-1}^* &= \mathbf{M}_t^* \mathbf{z}_{t|t-1} = \mathbf{z}_{t|t-1} [\mathbf{L}_t] \\ \hat{\boldsymbol{\varepsilon}}_{t|t-1}^* &= \mathbf{y}_t^* - \hat{\mathbf{y}}_{t|t-1}^* \\ \boldsymbol{\Sigma}_{t|t-1}^* &= \mathbf{E} \hat{\boldsymbol{\varepsilon}}_{t|t-1}^* (\hat{\boldsymbol{\varepsilon}}_{t|t-1}^*)' = \mathbf{M}_t^* \mathbf{P}_{t|t-1} \mathbf{M}_t^{*'} = \mathbf{P}_{t|t-1} [\mathbf{L}_t, \mathbf{L}_t] \\ \boldsymbol{\Sigma} &= \mathbf{P}_{t|t-1} [1:n, 1:n] \end{aligned}$$

Updating equation

$$\begin{aligned} \mathbf{z}_{t|t} &= \mathbf{z}_{t|t-1} + \mathbf{K}_t^* \hat{\boldsymbol{\varepsilon}}_{t|t-1}^* \\ \mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t^* \mathbf{M}_t^*) \mathbf{P}_{t|t-1} = \mathbf{P}_{t|t-1} - \mathbf{K}_t^* \mathbf{P}_{t|t-1} [\mathbf{L}_t, \mathbf{L}_t] \end{aligned}$$

where

$$\mathbf{K}_t^* = \mathbf{P}_{t|t-1} \mathbf{M}_t^{*'} (\boldsymbol{\Sigma}_{t|t-1}^*)^{-1} = \mathbf{P}_{t|t-1} [\mathbf{L}_t] (\boldsymbol{\Sigma}_{t|t-1}^*)^{-1}$$

6. An R package: starimax

An internal R package “starimax” is developed for Zurich Insurance Company for some projects.

starimax {starimax}

R Documentation

Space-Time ARIMAX

Description

Fit and predict the STARMAX model. Missings in target will be auto-handled. Missing in predictors are not allowed. The STARMA model is:

$$y(t) = \sum_{j=1}^p \alpha_j y(t-j) + \sum_{k=0}^q \beta_k x_{r_j}(t-k) + \sum_{l=0}^m \gamma_l \sum_{k=0}^p \alpha_k \varphi_{l,k} W_{\theta_{l,k}}(t) x_{r_j}(t-k) + \sum_{k=1}^p \rho_k \sum_{l=0}^m \gamma_l \sum_{k=0}^p \alpha_k \varphi_{l,k} W_{\theta_{l,k}}(t) y(t-k) + \sum_{k=1}^q \alpha_k \sum_{l=0}^m \gamma_l \sum_{k=0}^p \alpha_k \varphi_{l,k} W_{\theta_{l,k}}(t) \varepsilon(t-k) + \varepsilon(t)$$

Usage

starimax(datalist, wlist, paramatlist)

Arguments

datalist list of data matrices with row representing time and column representing space location. First element is for target data matrix. The 2nd, 3rd, ..., are predictor matrices. All of them should lined up with the same row and column representing the same time and location. The predictor matrices may have extra rows at end. The extra rows are used in outsample prediction.

wlist list of neighborhood weight matrices, with rowsum = 1 for non-zero rows of each matrix. Neighborhood matrix doesn't need to be symmetric so it can handle the directional neighbor relationship. The l-th element wlist[[l]] is the (l-1)-th order neighborhood weight matrix with wlist[[l]][i,j] representing the contribution of location j to location i. The wlist[[1]], the 0th order neighbor matrix, is an identity matrix representing the relationship of a location to itself. The matrices maybe sparse.

paramatlist list of 0/1 matrices with 1 indicating the non-zero parameter to be estimated. The paramatlist[[1]] and paramatlist[[2]] are for AR and MA component respectively. For example, paramatlist[[1]][t,s]=1 means t-th time lag and (s-1)-th space lag is non-zero and will be estimated. The paramatlist[[i]] of i >= 3 are for predictors, paramatlist[[3]][t,s]=1 means the (t-1)-th time lag and (s-1)-th space lag parameter is non-zero and to be estimated.

Value

para: a list of (para_est, para_sd, para_tvalue) parameter estimates, the standard deviations and t-values.

yhat: a list of (yhat, sd_yhat, resid, rmse). The in sample predictions. yhat\$yhat - prediction matrix with same shape of datalist[[1]]. The yhat\$yhat[t], t = 1 to T=nrow(datalist[[1]]), is the 1-step ahead prediction based on model built using only info upto t-1. The true out-sample prediction in insample data time point.

yhat_out: a list of (yhat_out, sd_yhat_out). The 1- & 2-step ahead outsample prediction from the last insample time point T.

7. Simulation Example

7.1 Data simulation

We will simulate some data for the 13 locations from the example road network given Figure 1 by KAMARIANAKIS & PRASTACOS (2002)

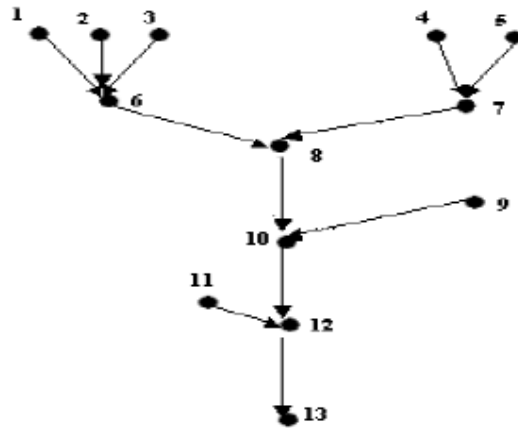


FIGURE 1. The typical road network tree structure for traffic flow. The dots represent measurement locations and the arrows the direction of flow. (KAMARIANAKIS & PRASTACOS 2002)

The $W^{(0)}, W^{(1)}, W^{(2)}$ are

```
> nbrwtlist[[1]]
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
[1,]  1  0  0  0  0  0  0  0  0  0  0  0  0
[2,]  0  1  0  0  0  0  0  0  0  0  0  0  0
[3,]  0  0  1  0  0  0  0  0  0  0  0  0  0
[4,]  0  0  0  1  0  0  0  0  0  0  0  0  0
[5,]  0  0  0  0  1  0  0  0  0  0  0  0  0
[6,]  0  0  0  0  0  1  0  0  0  0  0  0  0
[7,]  0  0  0  0  0  0  1  0  0  0  0  0  0
[8,]  0  0  0  0  0  0  0  1  0  0  0  0  0
[9,]  0  0  0  0  0  0  0  0  1  0  0  0  0
[10,] 0  0  0  0  0  0  0  0  0  1  0  0  0
[11,] 0  0  0  0  0  0  0  0  0  0  1  0  0
[12,] 0  0  0  0  0  0  0  0  0  0  0  1  0
[13,] 0  0  0  0  0  0  0  0  0  0  0  0  1

> round(nbrwtlist[[2]],digits=2)
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
[1,] 0.00 0.00 0.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[2,] 0.00 0.00 0.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[3,] 0.00 0.00 0.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[4,] 0.00 0.00 0.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[5,] 0.00 0.00 0.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[6,] 0.33 0.33 0.33 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[7,] 0.00 0.00 0.00 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[8,] 0.00 0.00 0.00 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0
[9,] 0.00 0.00 0.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[10,] 0.00 0.00 0.00 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0
[11,] 0.00 0.00 0.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[12,] 0.00 0.00 0.00 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0
[13,] 0.00 0.00 0.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0
```



```

> round(nbrwtlist[[3]],digits=2)
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
[1,] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[2,] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[3,] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[4,] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[5,] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[6,] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[7,] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[8,] 0.2 0.2 0.2 0.2 0.2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[9,] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[10,] 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0
[11,] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[12,] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0
[13,] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0

```

Model

We will simulate data using above neighborhood matrices from following model.

$$\mathbf{y}_t = 0.4W^{(0)}\mathbf{y}_{t-1} + 0.25W^{(1)}\mathbf{y}_{t-1} + 0.25W^{(0)}\mathbf{y}_{t-2} - .3W^{(1)}\boldsymbol{\varepsilon}_{t-1} + 0.6W^{(0)}\mathbf{x}_{t-1} + \boldsymbol{\varepsilon}_t$$

True parameter values

$$\boldsymbol{\Phi} = \begin{pmatrix} 0.4 & 0.25 \\ 0.25 & 0 \end{pmatrix}, \boldsymbol{\theta} = (0 \quad -0.3), \boldsymbol{\beta} = \begin{pmatrix} 0 \\ 0.6 \end{pmatrix}$$

7.2 Code for simulation and estimating parameters

```

library(MASS)
library(starimax)
# Simulate DATA, STARMA(2_(1,0), 1_(1)) with 1 predictor
n=13
epsi = mvrnorm (200, mu=rep(0,n), Sigma = diag(n))
sim <- epsi
x = mvrnorm (200, mu=rep(0.1,n), Sigma = diag(rep(0.6,n)))
for (t in 3:200) {
  sim[t,] <- (.4* nbrwtlist[[1]] + .25*nbrwtlist[[2]]) %%% sim[t-1,] +
    (.25*nbrwtlist[[1]] %%% sim[t-2,] +
    (-.3*nbrwtlist[[2]]) %%% epsi[t-1,] +
    0.6*nbrwtlist[[1]] %%% x[t-1,] + epsi[t, ]
}
y = sim[101:200,]
x=x[101:200,]

#Fit starmax model
ar_nonZeroMx <- matrix(c(1,1,1,0), 2, 2) # phi indicator matrix with 1 indicating the
non-zero parameters
ma_nonZeroMx <- matrix(c(0,1), 1, 2) # theta indicator matrix
beta_nonZeroMx = matrix(c(0,1), 2, 1) # beta indicator matrix
model_fit <- starimax(datalist=list(y,x), wlist=nbrwtlist,
  paramatlist=list(ar=ar_nonZeroMx, ma=ma_nonZeroMx, beta=beta_nonZeroMx))

```

7.3 Estimated model parameters

```

> model_fit$para$para_est      > model_fit$para$para_sd      > model_fit$para$para_tvalue
$ar                             $ar                             $ar
      [,1]      [,2]          [,1]      [,2]          [,1]      [,2]
[1,] 0.3625931 0.2790931 [1,] 0.02411159 0.05785846 [1,] 15.03812 4.823722
[2,] 0.2461651 0.0000000 [2,] 0.02411603 0.00000000 [2,] 10.20753 0.000000

$ma                             $ma                             $ma
      [,1]      [,2]          [,1]      [,2]          [,1]      [,2]
[1,]      0 -0.2776523 [1,]      0 0.0790457 [1,]      0 -3.512555

$beta                           $beta                           $beta
      [,1]          [,1]          [,1]
[1,] 0.0000000 [1,] 0.00000000 [1,] 0.0000
[2,] 0.5981624 [2,] 0.03563758 [2,] 16.7846

```

References

- Pfeifer, P.E. and Deutsch, S.J. (1980) A three-stage iterative procedure for space-time modeling. *Technometrics* 22, 35-47.
- Stoffer, D. (1986) "Estimation and identification of space-time ARMAX models in the presence of missing data." *Journal of the American Statistical Association*, Vol 81, 762 - 772
- Cipra, Tomáš, and Motyková, I. "Study on Kalman filter in time series analysis." *Commentationes Mathematicae Universitatis Carolinae* Vol. 28 (1987), No. 3, 549-563.
- Kohn, R. and Ansley, C. F. (1986) "Estimation, Prediction, and Interpolation for ARIMA Models with Missing Data". *Journal of the American Statistical Association*, Vol. 81, pp.751-761.
- Felix Cheysson (2016) Package 'starma' (<https://cran.r-project.org/web/packages/starma/starma.pdf>)