XI Congreso Galego de Estatística e Investigación de Operacións
A Coruña, 24–25–26 de outubro de 2013

MODELOS TEMPORALES LINEALES MIXTOS EN R

M.D. Esteban[1], D. Morales[1] and A. Pérez[2]

[1]Centro de Investigación Operativa, Universidad Miguel Hernández de Elche.
[2]Centro de Investigación Operativa, Universidad Miguel Hernández de Elche.
[3]Departmento Estudios Económicos y Financieros, Universidad Miguel Hernández de Elche.

RESUMO

En esta comunicación se muestra el manejo y aplicación del paquete SAERY de R, desarrollado para el cálculo de estimadores EBLUP en áreas pequeñas bajo modelos temporales lineales mixtos de área. La particularidad de los modelos lineales mixtos es que son idóneos para la estimación de efectos de factores con muchos niveles. Los modelos lineales anidados son un caso particular de estos, que permiten modelar las relaciones entre los diferentes niveles de agregación territorial y temporal. En la estimación en áreas pequeñas este punto es muy importante, pues el efecto de la componente temporal es un factor que se descompone habitualmente en varios niveles en la componente geográfica. Por este motivo, los modelos de área implementados en el paquete SAERY, cubren tanto efectos temporales autocorrelados como independientes, ofreciendo errores cuadráticos medios de los estimadores EBLUP mediante fórmulas explicitas y obtenidas a través de técnicas de remuestreo bootstrap.

Palabras e frases chave: Small area estimation, area-level models, time correlation, bootstrap, r-project.

1. INTRODUCIÓN

Let us consider the model

$$y_{dt} = \boldsymbol{x}_{dt}\boldsymbol{\beta} + u_{1,d} + u_{2,dt} + e_{dt}, \quad d = 1, \ldots, D,\ t = 1, \ldots, m_d$$

where y_{dt} is a direct estimator of the indicator of interest for area d and time instant t, $\boldsymbol{x}_{dt}$ is a vector containing the aggregated (population) values of p auxiliary variables, the random vectors $u_{1,d}$'s are independent $N(0, \sigma_1^2)$, the random vectors $(u_{d1}, \ldots, u_{dm_d})$, $d = 1, \ldots, D$, are i.i.d. componentwise independent, AR(1) or MA(1), with variance parameters σ_2^2, (σ_2^2, ϕ) and (σ_2^2, θ) respectively, the errors e_{dtj}'s are independent $N(0, \sigma_{dt}^2)$ with known σ_{dt}^2's, and the $u_{1,d}$'s, $u_{2,dt}$'s and the e_{dt}'s are independent.

The model can be written alternatively in the form

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}_1\boldsymbol{u}_1 + \boldsymbol{Z}_2\boldsymbol{u}_2 + \boldsymbol{e},$$

where $\boldsymbol{y} = \underset{1\le d\le D}{\text{col}}(\boldsymbol{y}_d)$, $\boldsymbol{y}_d = \underset{1\le t\le m_d}{\text{col}}(y_{dt})$, $\boldsymbol{u}_1 = \underset{1\le d\le D}{\text{col}}(u_{1,d})$, $\boldsymbol{u}_2 = \underset{1\le d\le D}{\text{col}}(\boldsymbol{u}_{2,d})$, $\boldsymbol{u}_{2,d} = \underset{1\le t\le m_d}{\text{col}}(u_{2,dt})$, $\boldsymbol{e} = \underset{1\le d\le D}{\text{col}}(\boldsymbol{e}_d)$, $\boldsymbol{e}_d = \underset{1\le t\le m_d}{\text{col}}(e_{dt})$, $\boldsymbol{X} = \underset{1\le d\le D}{\text{col}}(\boldsymbol{X}_d)$, $\boldsymbol{X}_d = \underset{1\le t\le m_d}{\text{col}}(\boldsymbol{x}_{dt})$, $\boldsymbol{x}_{dt} = \underset{1\le j\le p}{\text{col}'}(x_{dtj})$, $\boldsymbol{\beta} = \underset{1\le j\le p}{\text{col}}(\beta_j)$, $\boldsymbol{Z}_1 = \underset{1\le d\le D}{\text{diag}}(\boldsymbol{1}_{m_d})$, $\boldsymbol{Z}_2 = \boldsymbol{I}_{M\times M}$, $M = \sum_{d=1}^{D} m_d$.

For the AR(1) model, we have $\boldsymbol{u}_1 \sim N(\boldsymbol{0}, \boldsymbol{V}_{u_1})$, $\boldsymbol{u}_2 \sim N(\boldsymbol{0}, \boldsymbol{V}_{u_2})$,

$\boldsymbol{e} \sim N(\boldsymbol{0}, \boldsymbol{V}_e)$ are independent with $\boldsymbol{V}_{u_1} = \sigma_1^2\boldsymbol{I}_D$, $\boldsymbol{V}_{u_2} = \sigma_2^2\Omega(\phi)$,

$$\Omega(\phi) = \underset{1\le d\le D}{\text{diag}}(\Omega_d(\phi)), \quad \boldsymbol{V}_e = \underset{1\le d\le D}{\text{diag}}(\boldsymbol{V}_{ed}), \quad \boldsymbol{V}_{ed} = \underset{1\le t\le m_d}{\text{diag}}(\sigma_{dt}^2),$$

$$\Omega_d = \Omega_d(\phi) = \frac{1}{1-\phi^2}\begin{pmatrix} 1 & \phi & \dots & \phi^{m_d-2} & \phi^{m_d-1} \\ \phi & 1 & \ddots & & \phi^{m_d-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \phi^{m_d-2} & & \ddots & 1 & \phi \\ \phi^{m_d-1} & \phi^{m_d-2} & \dots & \phi & 1 \end{pmatrix}_{m_d \times m_d}.$$

The BLU estimator and predictor of $\boldsymbol{\beta}$ and $\boldsymbol{u}$ are

$$\widetilde{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{y}, \qquad \widetilde{\boldsymbol{u}} = \boldsymbol{V}_u\boldsymbol{Z}'\boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{X}\widetilde{\boldsymbol{\beta}}),$$

where $\boldsymbol{V}_u = \text{diag}(\boldsymbol{V}_{u_1}, \boldsymbol{V}_{u_2})$ and

$$\begin{aligned} \boldsymbol{V} &= \text{var}(\boldsymbol{y}) = \sigma_1^2\boldsymbol{Z}_1\boldsymbol{Z}_1' + \sigma_2^2 \underset{1\le d\le D}{\text{diag}}\,(\Omega_d(\phi)) + \boldsymbol{V}_e \\ &= \underset{1\le d\le D}{\text{diag}}\,(\sigma_1^2\mathbf{1}_{m_d}\mathbf{1}_{m_d}' + \sigma_2^2\Omega_d(\phi) + \boldsymbol{V}_{ed}) = \underset{1\le d\le D}{\text{diag}}\,(\boldsymbol{V}_d). \end{aligned}$$

2. REML estimation

The REML loglikelihood of the AR(1) model is

$$\begin{aligned} l_{REML}(\sigma_1^2, \sigma_2^2, \phi) &= -\frac{M-p}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{X}'\boldsymbol{X}| - \frac{1}{2}\log|\boldsymbol{V}| \\ &- \frac{1}{2}\log|\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X}| - \frac{1}{2}\,\boldsymbol{y}'\boldsymbol{P}\boldsymbol{y}, \end{aligned}$$

where $\boldsymbol{P} = \boldsymbol{V}^{-1} - \boldsymbol{V}^{-1}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}^{-1}$. To maximize $l_{REML}(\sigma_1^2, \sigma_2^2, \phi)$, we first define $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3) = (\sigma_1^2, \sigma_2^2, \phi)$, $\boldsymbol{V}_1 = \frac{\partial \boldsymbol{V}}{\partial \sigma_1^2} = \underset{1\le d\le D}{\text{diag}}\,(\mathbf{1}_{m_d}\mathbf{1}_{m_d}')$, $\boldsymbol{V}_2 = \frac{\partial \boldsymbol{V}}{\partial \sigma_2^2} = \underset{1\le d\le D}{\text{diag}}\,(\Omega_d(\phi))$ and $\boldsymbol{V}_3 = \frac{\partial \boldsymbol{V}}{\partial \phi} = \sigma_2^2 \underset{1\le d\le D}{\text{diag}}\,(\dot{\Omega}_d(\phi))$. By taking partial derivatives we get the scores

$$S_a = \frac{\partial l_{REML}}{\partial \theta_a} = -\frac{1}{2}\text{tr}(\boldsymbol{P}\boldsymbol{V}_a) + \frac{1}{2}\boldsymbol{y}'\boldsymbol{P}\boldsymbol{V}_a\boldsymbol{P}\boldsymbol{y}, \quad a = 1, 2, 3.$$

The Fisher information components are

$$F_{ab} = \frac{1}{2}\text{tr}(\boldsymbol{P}\boldsymbol{V}_a\boldsymbol{P}\boldsymbol{V}_b), \quad a, b = 1, 2, 3.$$

To maximize l_{REML}, Fisher-scoring updating formula is

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \boldsymbol{F}^{-1}(\boldsymbol{\theta}^k)SS(\boldsymbol{\theta}^k).$$

The EBLUP of $\mu_{dt} = \boldsymbol{x}_{dt}\beta + u_{1,d} + u_{2,dt} \approx y_{dt} = \boldsymbol{a}'\boldsymbol{y}$, $\boldsymbol{a} = \underset{1\le \ell\le D}{\text{col}}(\underset{1\le k\le m_\ell}{\text{col}}(\delta_{d\ell}\delta_{tk}))$ is $\widehat{\mu}_{dt} = \boldsymbol{x}_{dt}\widehat{\beta} + \widehat{u}_{1,d} + \widehat{u}_{2,dt}$, where $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\widehat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\widehat{\boldsymbol{V}}^{-1}\boldsymbol{y}$, $\widehat{\boldsymbol{u}} = \boldsymbol{V}_u\boldsymbol{Z}'\widehat{\boldsymbol{V}}^{-1}(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}})$. Finally, $\overline{Y}_{dt}$ is estimated by $\widehat{\overline{Y}}_{dt}^{eblup} = \widehat{\mu}_{dt}$.

3. MSE of EBLUP

For the independent component-wise model, the MSE of $\widehat{\overline{Y}}_{dt}^{eblup}$ is $MSE(\widehat{\overline{Y}}_{dt}^{eblup}) = g_1(\boldsymbol{\theta}) + g_2(\boldsymbol{\theta}) + g_3(\boldsymbol{\theta})$, $\boldsymbol{\theta} = (\sigma_1^2, \sigma_2^2)$,

$$\begin{aligned} g_1(\boldsymbol{\theta}) &= \boldsymbol{a}'\boldsymbol{Z}\boldsymbol{T}\boldsymbol{Z}'\boldsymbol{a}, \\ g_2(\boldsymbol{\theta}) &= [\boldsymbol{a}'\boldsymbol{X} - \boldsymbol{a}'\boldsymbol{Z}\boldsymbol{T}\boldsymbol{Z}'\boldsymbol{V}_e^{-1}\boldsymbol{X}]\boldsymbol{Q}[\boldsymbol{X}'\boldsymbol{a} - \boldsymbol{X}'\boldsymbol{V}_e^{-1}\boldsymbol{Z}\boldsymbol{T}\boldsymbol{Z}'\boldsymbol{a}], \\ g_3(\boldsymbol{\theta}) &\approx \text{tr}\left\{(\nabla \boldsymbol{b}')\boldsymbol{V}(\nabla \boldsymbol{b}')'E\left[(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})'\right]\right\}, \\ \boldsymbol{Q} &= (\boldsymbol{X}'\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}, \ \boldsymbol{T} = \boldsymbol{V}_u - \boldsymbol{V}_u\boldsymbol{Z}'\boldsymbol{V}^{-1}\boldsymbol{Z}\boldsymbol{V}_u, \\ \boldsymbol{b}' &= \boldsymbol{a}'\boldsymbol{Z}\boldsymbol{V}_u\boldsymbol{Z}'\boldsymbol{V}^{-1}. \end{aligned}$$

An estimator of $MSE(\widehat{\overline{Y}}_{dt}^{eblup})$ is

$$mse(\widehat{\overline{Y}}_{dt}^{eblup}) = g_1(\hat{\boldsymbol{\theta}}) + g_2(\hat{\boldsymbol{\theta}}) + 2g_3(\hat{\boldsymbol{\theta}}).$$

For the AR(1) and MA(1) models, the mean squared error is estimated by parametric bootstrap.

4. Package structure

The following functions has been developed

- `fit.saery (X, ydi, D, md, sigma2edi, model=c("INDEP","AR1","MA1"), conf.level=0.95)`

  This function calls fit.saery.AR1, fit.saery.indep, fit.saery.MA1 to fit the model with the corresponding

  - REML function REML.saery.AR1, REML.saery.indep, or REML.saery.MA1 with arguments (X, ydi, D, md, sigma2edi, sigma1.0 = 1, sigma2.0 = 1, MAXITER = 20)
  - $\widehat{\boldsymbol{\beta}}$ function BETA.U.saery.AR1, BETA.U.saery.indep, or BETA.U.saery.MA1 with arguments (X, ydi, D, md, sigma2edi, sigmau1, sigmau2, rho). The last one may not appear in the independent case, or may be theta when MA1.
  - and its inference with
    - stderr.saery.AR1 (fit), stderr.saery.indep (fit) or stderr.saery.MA1 (fit) that calculates the standard error to do confidence intervals by using a valid REML fit.
    - and pvalue (beta.fitted, fit) function that calculates p-values for the p auxiliary variables by using the $\widehat{\boldsymbol{\beta}}$'s calculated and a valid REML fit.

- `eblup.saery (X, ydi, D, md, sigma2edi, model=c("INDEP","AR1","MA1"), plot=FALSE)`

  This function calls eblup.saery.AR1, eblup.saery.indep and eblup.saery.MA1 with arguments (X, ydi, D, md, sigma2edi, plot), to calculate

  - the EBLUP of the target variable using REML and $\widehat{\boldsymbol{\beta}}$ functions
  - and the MSE of EBLUP by mse.saery.boot.AR1, mse.saery.indep or mse.saery.boot.MA1 with arguments (X, D, md, beta, sigma2edi, sigmau1, sigmau2, rho, B = 100). The rho argument may not appear in the independent case, or may be theta when MA1. Note that the last argument only appears in the AR(1) and MA(1) cases, the estimator of mse is calculated by parametric bootstrap resampling.

  The last argument of eblup.saery function is a logical value (false by dafault). When it is TRUE, eblup.saery function calls plot.saery (direct, eblup, mse.eblup, sigma2edi) and ask the user before starting a new page of graphical outputs.

  eblup.saery function returns a table with *domain, period, direct, eblup, direct variance, mse of eblup* and *residuals* columns.

5. Fitting the model

The following functions and outputs are obtained.

> fit.saery <- function(X, ydi, D, md, sigma2edi, model = c("INDEP","AR1","MA1"), conf.level = 0.95){
+ model < − toupper(model)
+ model < − match.arg(model)
+ switch(model,
+ INDEP = fit.saery.indep(X, ydi, D, md, sigma2edi, conf.level),
+ AR1 = fit.saery.AR1(X, ydi, D, md, sigma2edi, conf.level),
+ MA1 = fit.saery.MA1(X, ydi, D, md, sigma2edi, conf.level)

```
+ )
+ }
>
> output.fit < − fit.saery(X, ydi, D, md, sigma2edi, "a", 0.9)
> output.fit
$Regression
```

	Estimate	std.err	t	p.value	lower	upper
1	3.721663	0.1694916	21.95781	7.293201e-107	3.442875	4.000452

$SIGMA

	Estimate	lower	upper
sigmau1	0.007378169	0.0054066215	0.009349716
sigmau2	0.001436561	0.0009014452	0.001971677
rho	0.104141885	-0.3232740890	0.531557859

6. Calculating the EBLUP

The following functions and outputs are obtained.

```
> eblup.saery <- function(X, ydi, D, md, sigma2edi, model =
c("INDEP","AR1","MA1"), plot = FALSE){
+ model < − toupper(model)
+ model < − match.arg(model)
+ switch(model,
+ INDEP = eblup.saery.indep(X, ydi, D, md, sigma2edi, plot),
+ AR1 = eblup.saery.AR1(X, ydi, D, md, sigma2edi, plot),
+ MA1 = eblup.saery.MA1(X, ydi, D, md, sigma2edi, plot)
+ )
+ }
>
> eblup.output <- eblup.saery(X, ydi, D, md, sigma2edi, model="a", plot = FALSE)
> eblup.output
```

	Domain	Period	direct	eblup	var.direct	mse.eblup	resid
1	11	1	0.07099	0.07536	0.0003447	0.0003562	-0.0043698
2	11	2	0.07228	0.07233	0.0003959	0.0002612	-0.0000411
3	11	3	0.08266	0.07917	0.0004783	0.0004050	0.0034857
4	12	1	0.14603	0.15060	0.0004854	0.0004206	-0.0045717
5	12	2	0.08249	0.08189	0.0003663	0.0002330	0.0006042
6	12	3	0.07865	0.07952	0.0004036	0.0003220	-0.0008695
7	21	1	0.30106	0.26326	0.0015300	0.0009465	0.0377976
8	21	2	0.24181	0.24492	0.0011871	0.0008221	-0.0031025
9	21	3	0.23741	0.24648	0.0013034	0.0008112	-0.0090772
10	22	1	0.30582	0.28852	0.0013665	0.0006783	0.0172989

3. CONCLUSIÓNS

The r-package is still under construction. Algorithms and functions are based on the paper Esteban et al. (2102).

REFERENCIAS

Esteban, M.D., Morales, D., Pérez, A., Santamaría, L. (2011). Small area estimation of poverty proportions under area-level time models. Computational Statistics and Data Analysis, 56, 2840–2855.