Regional Research Institute

Working Paper Series

Impact estimates for static spatial panel data models in R

By: Gianfranco Piras, Regional Research Institute,
West Virginia University

Working Paper Number 2013-05

Website address: rri.wvu.edu

Impact estimates for static spatial panel data models in R

Gianfranco Piras
Regional Research Institute
West Virginia University

Abstract

In the present note we demonstrate how to implement the Lee and Yu (2010) procedure for fixed effects spatial panel data models available from the R (R Development Core Team 2012) package **splm** (Millo and Piras 2012). Additionally, we also show how to compute the impact estimates introduced by Kelejian, Tavlas, and Hondroyiannis (2006) and formalized in LeSage and Pace (2009). Unlike Matlab (MATLAB 2011), there was no R function specific to static panel data models for the calculation of the impact measures. After receiving numerous requests from the users of **splm**, we decided to extend the cross sectional functions available from **spdep** (Bivand 2013) to spatial panel data models.

Keywords: Spatial Panel Data Models, R, Computational Methods, Impact Measures.

1. Introduction

The present note has two objectives. On the one hand, we describe a recent addition to the R package **splm** (Millo and Piras 2012) that implements the transformation that has been set forth by Lee and Yu (2010) for fixed effects spatial panel data models. On the other hand, we show how to implement the impact measures introduced by Kelejian *et al.* (2006) and formalized in LeSage and Pace (2009) using the infrastructures available from two of the R libraries: **spdep** (Bivand 2013) and **splm**.

Lee and Yu (2010) propose an alternative estimation method based on a transformation approach. Also in the spatial panel case, the "classical" approach to eliminate the individual effects using the deviation from the time mean operator would generate residuals that are linearly dependent over the time dimension. To avoid this, they propose a different transformation based on the orthonormal eigenvector matrix of $J_T = (I_T - \frac{1}{T}\iota\iota')$. Unlike the direct approach, the orthonormal transformation yields consistent estimators with properly centered distributions.

Additionally, the structure of dependence embedded in spatial regression models is more complicated than other traditional models (LeSage and Pace 2009). A change in a single observation associated with any of the explanatory variables produces a change in the observation itself (direct impact), as well as in all other observations (indirect impact). There are methods available from **spdep** to calculate these impact measures, but **splm** did not include any additional functions specific for panel data. However, **splm** only deals with static panel models. Cross sectional methods, available from **spdep**, can be extended very easily to the spatial panel data case. For the convenience of the users, we wrote a wrapper function adapting the various panel objects in order to be able to apply the cross sectional methods to them. In the second part of the present paper we demonstrate the steps that are needed to undertake this task. We also compare the evidence obtained in the paper with available codes from Matlab.

2. Preliminaries

To illustrate our examples we use the same dataset as in Millo and Piras (2012). The examples in their paper are illustrated using the widely known Munnell (1990) data on public capital productivity in 48 US states observed over 17 years, available from the R package **Ecdat** (Croissant 2010).

Munnell's model is a Cobb-Douglas production function that relates the gross social product (gsp) of a given state in a given year to the input of public capital (pcap), private capital (pc) and labor (emp). State unemployment rate (unemp) is meant to capture business cycle effects. The model formula is defined at the end of the following script (after loading the library and sourcing the data) and includes a constant term (that will be eliminated by the fixed effects transformation).

```
R> library("splm")
R> data(Produc, package = "Ecdat")
R> fm <- log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp
```

The spatial weighting matrix based on binary contiguity for the US states is included in **splm**. We also transform the weighting matrix into a `listw` object using infrastructure from the **spdep** package, and the `listw` object into a sparse **Matrix** (Bates and Maechler 2013):

```
R> data(usaww)
R> lwstates <- mat2listw(usaww, style ="W")
R> sparse.W <- listw2dgCMatrix(lwstates)
```

3. Lee and Yu transformation approach

The model below illustrates a relatively new characteristic of **splm**, that is the implementation of the asymptotic variance covariance matrix for a fixed effects model with both lags (i.e., the spatial lag of the dependent variable as well as the spatially autocorrelated residuals). In fact, the only available feature in the previous version of the package was the numerical Hessian, and no asymptotic matrix was calculated for this case.[1] The results below are comparable to those on pg. 13 of Millo and Piras (2012). While the coefficients are exactly the same, the standard errors are slightly different. The standard errors below are obtained with an exact calculation based on the asymptotic variance covariance matrix (setting the options `LeeYu` and `Hess` to `FALSE`), while the standard errors in Millo and Piras (2012) were obtained by

[1] An asymptotic variance covariance matrix was calculated both for the fixed effects spatial lag and for the fixed effects spatial error model.

means of a numerical Hessian. As it can be observed, the difference is not substantial and no changes in the statistical significance of the coefficients is apparent. This evidence confirms the fact that the numerical Hessian implemented in R and used by previous versions of **splm** is extremely reliable.[2]

```
R> Sarar_FEML_noLY <- spml(fm, data = Produc, listw = lwstates,
+  model = "within", spatial.error= "b", effect = "individual",
+  method = "eigen", lag = T, LeeYu = F, Hess = F)
R> summary(Sarar_FEML_noLY)
```

```
Spatial panel fixed effects sarar model

Call:
spml(formula = fm, data = Produc, listw = lwstates, model = "within",
    effect = "individual", lag = T, spatial.error = "b", method = "eigen",
    LeeYu = F, Hess = F)

Residuals:
   Min. 1st Qu.  Median 3rd Qu.    Max.
-0.1340 -0.0221 -0.0032  0.0172  0.1750

Coefficients:
             Estimate Std. Error t-value  Pr(>|t|)
lambda      0.0885760  0.0263125  3.3663 0.0007618 ***
rho         0.4553116  0.0425384 10.7036 < 2.2e-16 ***
log(pcap) -0.0103497  0.0255345 -0.4053 0.6852418
log(pc)     0.1905781  0.0242829  7.8483 4.219e-15 ***
log(emp)    0.7552372  0.0290385 26.0081 < 2.2e-16 ***
unemp     -0.0030613  0.0010315 -2.9678 0.0029998 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
R> Sarar_FEML_noLY$sigma2
```

```
[1] 0.0009966284
```

To implement the Lee and Yu (2010) transformation which yields consistent estimator of the variance parameter also when T is small, the option `LeeYu` should be set to `TRUE`, with `Hess` equal to `FALSE`.[3] As noted in Lee and Yu (2010), the estimators for the other parameters are the same as those obtained by the direct approach. The variance parameter is reported after the model summary and is equal to 0.001058918. This number represents the correction suggested by `LeeYu` and is simply $\frac{T}{T-1}\hat{\sigma}^2_{nT}$ where $\hat{\sigma}^2_{nT}$ is the residuals variance. For large T,

[2] The numerical Hessian is implemented in the function `fdHess` available from **nlme** (Pinheiro, Bates, DebRoy, Sarkar, and R Core Team 2013).

[3] If `Hess` was set to `TRUE`, the estimate of the variance covariance matrix would have been based on the numerical Hessian with the degrees of freedom correction.

the ratio $\frac{T}{T-1}$ limits to one, and the transformation approach has almost no effect on the estimate of the error variance. Thus, when T is small, the transformation approach produces a degrees of freedom correction.

```
R> Sarar_FEML <- spml(fm, data = Produc, listw = lwstates,
+  model = "within", spatial.error= "b",
+  lag = T, LeeYu = T, Hess = F)
R> summary(Sarar_FEML)

Spatial panel fixed effects sarar model

Call:
spml(formula = fm, data = Produc, listw = lwstates, model = "within",
    lag = T, spatial.error = "b", LeeYu = T, Hess = F)

Residuals:
   Min. 1st Qu.  Median 3rd Qu.    Max.
-0.1340 -0.0221 -0.0032  0.0172  0.1750

Coefficients:
            Estimate Std. Error t-value  Pr(>|t|)
lambda     0.0885760  0.0271223  3.2658  0.001092 **
rho        0.4553117  0.0438475 10.3840 < 2.2e-16 ***
log(pcap) -0.0103497  0.0263203 -0.3932  0.694158
log(pc)    0.1905781  0.0250302  7.6139 2.659e-14 ***
log(emp)   0.7552372  0.0299322 25.2316 < 2.2e-16 ***
unemp     -0.0030613  0.0010633 -2.8792  0.003987 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R> Sarar_FEML$sigma2

[1] 0.001058918

R> Sarar_FEML$sigma2*(length(unique(Produc$year))-1)/length(unique(Produc$year))

[1] 0.0009966284
```

The Matlab code available from Elhorst's website,[4] produces the results reported in Table 1 for the one-way fixed effects spatial lag model.[5] In fact, Matlab does not contemplate the estimation of the full model and, therefore, to have a real comparison, we need to estimate the same model using the R function `spml` by simply setting the option `spatial.error` to `"none"`.

[4] http://www.regroningen.nl/elhorst/software.shtml

[5] The results obtained with Matlab can be reproduced using the script `matlab_script.m` associated with the paper. Data and the spatial weighted matrix are also provided in a text format that can easily be read into Matlab. All the codes are available from the author upon request.

Table 1: Matlab results that can be obtained running the script associated with the paper.

Variables	Coefficient	Asymptot t-stat	z-probability
lnpcap	−0.046677	−1.834784	0.066538
lnpc	0.186947	8.113548	0.000000
lnemp	0.624425	21.022303	0.000000
unemp	−0.004478	−5.175262	0.000000
W*dep.var.	0.275965	11.738685	0.000000

Comparing the results obtained in R to those in Table 1 we note some minor differences in terms of the estimated coefficients. These differences, however, are most likely due to the use of different optimizers between the two softwares.[6] With regard to standard errors, the two implementations both return very similar figures. It is interesting that the coefficient associated with the input of public capital in both cases is only significant at the 10% level. The similarity of these results is very encouraging.

```
R> Lag_FEML <- spml(fm, data = Produc, listw = lwstates,
+  model = "within", spatial.error= "none",
+  lag = T, LeeYu = T, Hess = F)
R> summary(Lag_FEML)
```

```
Spatial panel fixed effects lag model

Call:
spml(formula = fm, data = Produc, listw = lwstates, model = "within",
    lag = T, spatial.error = "none", LeeYu = T, Hess = F)

Residuals:
   Min. 1st Qu.  Median 3rd Qu.    Max.
-0.1510 -0.0191 -0.0025  0.0159  0.1770

Coefficients:
            Estimate Std. Error t-value  Pr(>|t|)
lambda     0.2746887  0.0235164 11.6807 < 2.2e-16 ***
log(pcap) -0.0465819  0.0254425 -1.8309   0.06712 .
log(pc)    0.1874325  0.0230441  8.1336 4.166e-16 ***
log(emp)   0.6250902  0.0297044 21.0437 < 2.2e-16 ***
unemp     -0.0044816  0.0008653 -5.1792 2.228e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

[6] See e.g., Bivand and Piras (2013) for further details.

4. Impact measures

In addition to the fitting, associated measures might be needed to assist in the interpretation of model coefficients. Explicitly, we refer to the impact of changes in right hand side variables in models including the spatial lag of the dependent variable.[7] Millo and Piras (2012) did not mention how to perform this effects estimates in the panel data case. However, Millo and Piras (2012) only deal with static panel models (i.e., to date no dynamic panels are estimated with **splm**) and cross sectional methods, available from **spdep**, can be extended very easily to the spatial panel data case. In essence, starting from the reduced form of the spatial lag model the impacts estimates are related to the expression

$$S_r(W) = (I_n - \lambda W)^{-1} I_n \beta_r \tag{1}$$

In a static panel context, the previous equation corresponds to

$$S_r(\mathbf{W}) = (I_{nT} - \lambda \mathbf{W})^{-1} I_{nT} \beta_r \tag{2}$$

where $\mathbf{W} = (I_n \otimes W)$. Essentially, the only real difference from a computational perspective is to create the matrix $\mathbf{W}$.

After receiving numerous requests from the users of **splm**, we decided to extend the cross sectional functions available from **spdep** (Bivand 2013) to spatial panel data models. For the convenience of the users, we wrote a wrapper function (`impacts.splm`) adapting the various panel objects in order to be able to apply the cross sectional methods to them. As we mentioned, the only major change is applied to the spatial weighting matrix. The panel library assumes that the spatial weighting matrix is constant over time and, therefore, the input is always an $n \times n$ matrix (generally of class `listw`). Given the dimension of the data in a panel (i.e., nT), we need to construct a block diagonal matrix (i.e., $\mathbf{W}$) whose diagonal blocks are the spatial weighting matrix itself. The users can choose to perform the calculation outside or inside the wrapper function.

If the decision is to perform the computation outside, the first step would be to perform the kronecker product of a diagonal matrix of dimension $T \times T$ and the spatial weighting matrix, as illustrated in the following example:

```
R> time <- length(unique(Produc$year))
R> s.lwstates <- kronecker(Diagonal(time), sparse.W)
```

Without going into detail, the function `trW` prepares a vector of traces of powers of the spatial weighting matrix. The default values for the number of powers and the number of samples used in the Monte Carlo simulation of the trace are, respectively, 30 and 16. The `type = "mult"` compute the power of a sparse matrix. The other option available is to perform a Monte Carlo simulation of the traces.

After preparing the numeric factor of traces, the calculation of the impacts is carried out with the function `impacts.splm`, for which a summary method is available. The function takes as arguments the object from the estimation of the spatial model, the vector of traces, and an additional argument (`R`) that, if provided, determines the number of simulations that are used to compute the distributions for the impact measures. The short summary displays the results.

[7] Given the short character of the present note, we omit most of the discussion on the impacts. An extensive discussion can be found in LeSage and Pace (2009).

```
R> set.seed(12345)
R> trMatc <- trW(s.lwstates, type="mult")
R> imp <- impacts(Lag_FEML, tr = trMatc, R = 200)
R> summary(imp, zstats=TRUE, short=T)

Impact measures (lag, trace):
                Direct     Indirect       Total
log(pcap) -0.047503680 -0.016719632 -0.06422331
log(pc)    0.191141532  0.067275126  0.25841666
log(emp)   0.637459782  0.224363523  0.86182330
unemp     -0.004570274 -0.001608576 -0.00617885
========================================================
Simulation results ( variance matrix):
========================================================
Simulated z-values:
             Direct  Indirect     Total
log(pcap) -1.709877 -1.709976 -1.717275
log(pc)    8.205119  5.848844  7.836458
log(emp)  19.575339  8.007296 16.551673
unemp     -5.261708 -4.433847 -5.162641

Simulated p-values:
          Direct     Indirect   Total
log(pcap) 0.087289   0.08727    0.085929
log(pc)   2.2204e-16 4.9500e-09 4.6629e-15
log(emp)  < 2.22e-16 1.1102e-15 < 2.22e-16
unemp     1.4272e-07 9.2566e-06 2.4349e-07
```

The alternative would be to perform all the preliminary calculations inside the wrapper. In this case, the user needs to provide the object `listw` and the number of time periods. Of course, we would obtain identical results, as from the following script (where we set the seed as in the previous example):

```
R> set.seed(12345)
R> imp2 <- impacts(Lag_FEML, listw = lwstates, time = time)
R> summary(imp2, zstats=TRUE, short=T)

Impact measures (lag, trace):
                Direct     Indirect       Total
log(pcap) -0.047503680 -0.016719632 -0.06422331
log(pc)    0.191141532  0.067275126  0.25841666
log(emp)   0.637459782  0.224363523  0.86182330
unemp     -0.004570274 -0.001608576 -0.00617885
========================================================
Simulation results ( variance matrix):
========================================================
Simulated z-values:
```

```
              Direct   Indirect     Total
log(pcap) -1.709877 -1.709976 -1.717275
log(pc)    8.205119  5.848844  7.836458
log(emp)  19.575339  8.007296 16.551673
unemp     -5.261708 -4.433847 -5.162641

Simulated p-values:
          Direct      Indirect    Total
log(pcap) 0.087289    0.08727     0.085929
log(pc)   2.2204e-16 4.9500e-09 4.6629e-15
log(emp)  < 2.22e-16 1.1102e-15 < 2.22e-16
unemp     1.4272e-07 9.2566e-06 2.4349e-07
```

Finally, we compare the previous results with those obtained by the corresponding Matlab function.[8] Results are reported in Table 2. With regard to the estimate of the impacts measure, direct, indirect, and total effects are very similar up to the third decimal digit. Given that the coefficients estimates were very close between the implementations, this is not very surprising. Some differences are noted on the inference of the various effects. However, statistical inference for the impacts estimate is based on simulation results and, therefore, it is quite normal to experience those differences.

Table 2: Matlab results of the estimation for the impacts measures.

Variables	Direct	t-stat	indirect	t-stat	total	t-stat
lnpcap	−0.0496	−1.9368	−0.0178	−1.8537	−0.0675	−1.9244
lnpc	0.1911	8.1871	0.0678	7.3996	0.2589	8.6569
lnemp	0.6370	22.2591	0.2268	9.5653	0.8637	22.5866
unemp	−0.0045	−5.1904	−0.0016	−4.6041	−0.0062	−5.1781

5. Conclusion

In this note we showed how to perform the estimation of a fixed effect spatial panel data model using the Lee and Yu (2010) transformation approach. The different transformation put forth by Lee and Yu (2010) ensures the consistency of the error variance even if T is small. Furthermore, we have shown how to implement the impacts measures using the infrastructure available from **spdep** by mean of the wrapper function `impacts.splm`. Of course, the function could be used with fixed as well as random effects panel data models estimated with both maximum likelihood and the GMM approaches.

References

[8] `direct_indirect_effects_estimates`.

Bates D, Maechler M (2013). `Matrix`*: Sparse and Dense Matrix Classes and Methods.* R package version 1.0-12, URL `http://CRAN.R-project.org/package=Matrix`.

Bivand R, Piras G (2013). "Comparing Implementations of Estimation Methods for Spatial Econometrics." RRI Working Paper 2013-01, URL `http://rri.wvu.edu/wp-content/uploads/2012/11/Piras_BivandWP2013.pdf`.

Bivand RS (2013). ***spdep****: Spatial Dependence: Weighting Schemes, Statistics and Models.* R package version 0.5-56.

Croissant Y (2010). ***Ecdat****: Data sets for econometrics.* R package version 0.1-6, URL `http://CRAN.R-project.org/package=Ecdat`.

Kelejian HH, Tavlas GS, Hondroyiannis G (2006). "A Spatial Modelling Approach to Contagion Among Emerging Economies." *Open Economies Review*, **17**(4-5), 423–441.

Lee L, Yu J (2010). "Estimation of spatial autoregressive panel data models with fixed effects." *Journal of Econometrics*, (154), 165–185.

LeSage JP, Pace KR (2009). *Introduction to Spatial Econometrics*. CRC Press, Boca Raton, FL.

Millo G, Piras G (2012). "**splm**: Spatial Panel Data Models in R." *Journal of Statistical Software*, **47**(1), 1–38.

Munnell A (1990). "Why has productivity growth declined? Productivity and Public Investment." *New England Economic Review*, pp. 3–22.

Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2013). ***nlme****: Linear and Nonlinear Mixed Effects Models.* R package version 3.1-109.

MATLAB (2011). *version 7.13 (R2011b)*. The MathWorks Inc., Natick, Massachusetts.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL `http://www.R-project.org/`.

Affiliation:

Gianfranco Piras
Regional Research Institute
West Virginia University
886 Chestnut Ridge Road
Morgantown, WV 26506-6825
Telephone: 304-293-2643
Fax: 304-293-6699
E-mail: `gpiras@mac.com`