Comparing Implementations of Estimation Methods for Spatial Econometrics

By: Roger Bivand, Norwegian School of Economics and Gianfranco Piras,
West Virginia University

Working Paper Number 2013-01

Website address: rri.wvu.edu

Comparing Implementations of Estimation Methods for Spatial Econometrics

Roger Bivand
Norwegian School of Economics

Gianfranco Piras
Regional Research Institute
West Virginia University

Abstract

Recent advances in the implementation of spatial econometrics model estimation techniques have made it desirable to compare results, which should correspond between implementations across software applications for the same data. These model estimation techniques are associated with methods for estimating impacts (emanating effects), which are also presented and compared. This review constitutes an up-to-date comparison of generalized method of moments (GMM) and maximum likelihood (ML) implementations now available. The comparison uses the cross-sectional US county data set provided by Drukker, Prucha, and Raciborski (2011c, pp. 6–7). The comparisons will be cast in the context of alternatives using the MATLAB Spatial Econometrics toolbox, Stata, Python with PySAL (GMM) and R packages including **spdep**, **sphet** and **McSpatial**.

Keywords: spatial econometrics, maximum likelihood, generalized method of moments, estimation, R, Stata, Python, MATLAB.

1. Introduction

Researchers applying spatial econometric methods to empirical economic questions now have a wide range of tools, and a growing literature supporting these tools. In the 1970s and 1980s, it was typical for researchers to use tools coded in Fortran or other general programming languages, or to seek to integrate functions into existing statistical and/or matrix language environments (Anselin 2010). The use of spatial econometrics tools was widened by the ease with which methods and examples presented in Anselin (1988b) could be reproduced using SpaceStat, written in Gauss (Aptech 2007), and shipped as a built runtime module. It was rapidly complemented by the Spatial Econometrics toolbox for MATLAB (MATLAB 2011), provided as source code together with extensive documentation (see also LeSage and Pace 2009).[1] This toolbox is under active development, and accepts contributed functions, thus broadening its appeal. In addition Griffith and Layne (1999) gave code listings for model fitting techniques using SAS (SAS Institute 2008) and SPSS (IBM Corporation 2010). A suite of commands for spatial data analysis for use with Stata (StataCorp 2007) was provided by Maurizio Pisati, and distributed using the standard contributed command system (Pisati 2001).

The thrust of SpaceStat was largely been taken over by GeoDa (Anselin, Syabri, and Kho

[1] http://www.spatial-econometrics.com/.

2006), and more recently by OpenGeoDa.[2] The Python (van Rossum 1995) spatial analysis library (PySAL; Rey and Anselin 2010)[3] has also been launched. Since the R (R Development Core Team 2012) language and environment became available in the 1990s, collaborative code development has proceeded with varying speed. Initial attempts to implement spatial econometrics techniques in R in the **spdep** (Bivand 2013) package were checked against SpaceStat, and subsequently against Maurizio Pisati's Stata code and GeoDa by comparing results for the same input data and spatial weights (Bivand and Gebhardt 2000; Bivand 2002). A broad survey of the analysis of spatial data in the R environment is given by Bivand (2006) and Bivand, Pebesma, and Gómez-Rubio (2008).

In the spirit of Rey (2009), this comparison will attempt to examine some features of the implementation of functions for fitting spatial econometrics models in Stata with those in the Spatial Econometrics toolbox for MATLAB, in R and in Python. We have chosen only to compare implementations for which the estimations can be scripted, and from which the output can be transferred back to R in binary form. A consequence of this restriction is that we have not included OpenGeoDa. Because Millo and Piras (2012) provide recent comparative results for implementations of spatial panel models in R and MATLAB, we restrict our consideration to cross-sectional models, and within that to cross-sectional models implemented in at least two application languages. This results in our putting Bayesian methods for spatial econometric models aside until they become available in other languages than MATLAB. Finally, we have chosen not to consider tests for residual autocorrelation or for model specification, [4] or other diagnostic or exploratory techniques or measures, feeling that model estimation is of more immediate importance.

Initially, we describe the framework used for our comparative study, and the data set chosen for use. Next we define the models to be compared, and then move to comparisons for generalised method of moments (GMM) estimators. The GMM presentation is a substantial extension of Piras (2010), as many theoretical results have been published since then, and have been incorporated into the **sphet** package, as well as made available in Stata and PySAL. Following the comparison of GMM implementations, we examine implementations of maximum likelihood estimators, focussing on the consequences of details in the choices of numerical methods across the alternatives. Before concluding, we compare the provision of functions for calculating emanating effects (Kelejian, Tavlas, and Hondroyiannis 2006), also known as impacts (LeSage and Fischer 2008; LeSage and Pace 2009), simultaneous spatial reaction function/reduced form (Anselin and Lozano-Gracia 2008), equilibrium effects (Ward and Gleditsch 2008), and unwittingly touched upon by Bivand (2002) in trying to create a prediction method for spatial econometrics models.

1.1. Comparative study

The comparative study was constructed around unified R scripts. The first script prepared the data from the input data set for export to MATLAB in a text file, to Stata as a `dta` file and to Python as a `dbf` file written as part of an ESRI Shapefile. The data to be exported were run through `model.frame` first to generate the intercept where needed and any dummy variables

[2] `http://geodacenter.asu.edu/ogeoda`, source code: `http://code.google.com/p/opengeoda/`.

[3] `http://github.com/pysal/pysal`.

[4] See e.g.„ Anselin, Bera, Florax, and Yoon (1996); Anselin (1988a); Anselin and Bera (1998); Florax and Folmer (1992); Cliff and Ord (1972); Florax, Folmer, and Rey (2003); Kelejian and Piras (2011); Burridge (1980)

(none were needed with the data set used here). Next, the first script read a GAL-format file of county neighbours from which to form spatial weights; a row-standardised weights object was then formed for export and use in R. Weights were exported to MATLAB in a three-column sparse matrix text file, to Stata in GWT-format and to Python in GAL-format.

This R script was then used to run R code to estimate chosen spatial econometrics models, and to write scripts for MATLAB, Stata and Python. Keeping all of the code in a single R script was intended to ensure that attention was focussed on estimating the same models across the different implementations. The scripts were run chunk-by-chunk by hand or by highlighting code for execution in the script editors of the different applications, but finally for MATLAB and Stata they were run from start to termination continuously. The scripts output binary objects containing the estimated model results; in the R case, `save` was used for the objects from a given class of models. In MATLAB, use was made of the analogous `save` function; in Stata the `file` command with `write binary` options was used; in Python `save` imported from `numpy`. With these mechanisms, and after careful investigation of the output estimation objects, we were able to ensure that we were comparing estimates of the same model components.

A second unified script was used to coordinate and document the collation of results from the four applications into tabular form for this article, typically by setting up the results from one application in a column vector, using `cbind` to combine these columns. The binary output from R was read using `load`; from Stata using the R function `readBin`; from MATLAB using `readMat` in the **R.Matlab** package; and from Python using the `npyLoad` function from the **RcppCNPy** package. The tables for presentation below were then formatted using the same rounding arguments either for the whole table or row-wise, so that no differences could be introduced by rounding results from implementations in different ways. The remaining differences, if any, come from differences in the implementations, and it is these we intend to account for as far as possible.

The analysis has been carried out on an Intel Core i7 64-bit system with 8GB RAM under Windows 7 Enterprise SP1. The software used was Stata 12.1, MATLAB R2011b with the March 2010 version of the Spatial Econometrics Toolbox,[5] R 2.15.2 (R Development Core Team 2012) with packages **spdep** 0.5-56, **sphet** 1.4-00, and **McSpatial** 1.1.1 (McMillen 2012)and their contemporary dependencies, and Python 2.7 (32-bit) with PySAL 1.4. Some of the PySAL model estimation functions may also be accessed from GeoDaSpace, but we have used PySAL directly here.

We can see from the comparison of OLS results for the selected data set shown in Table 2 that the linear algebra output of the applications used is identical, and we can assume that any differences found in comparing spatial econometrics techniques will then not be caused by divergences in linear algebra implementations. There are, however, other underlying technologies that may differ, in particular numerical optimisation. From examining source code where available, it appears that the GM methods in PySAL use the SciPy `fmin_l_bfgs_b`, function in the optimize module, based on L_BFGS_B version 2.1 from 1997,[6] a quasi-Newton function for bound-constrained optimization. Nash and Varadhan (2011) provide a helpful guide to

[5] Local modifications were made in a copy kept by agreement with its authors as a subdirectory on `https://r-forge.r-project.org/projects/spdep2/`; these changes will be mentioned in the comparison of maximum likelihood methods, as they permitted additional options and returned values to be used.

[6] `http://www.scipy.org/doc/api_docs/SciPy.optimize.lbfgsb.html`, `http://users.eecs.northwestern.edu/~nocedal/lbfgsb.html`.

available numerical optimisation technologies, and to those available in R. In **sphet**, use is made of the `nlminb` function, which is a reverse-communication trust-region quasi-Newton method from the Port library,[7] with enhanced scaling features. The same function is used by default for fitting in **spdep** when more than one parameter is to be optimised; optionally, other optimisers, including the R implementation of L_BFGS_B version 2.1. For bounded line search in **spdep**, use is made of the `optimize` function, based on Brent (1973). In the R **McSpatial** package (McMillen 2012), `optimize` is used first to conduct a line search, and then `nlm`, a Newton-type algorithm (Schnabel, Koontz, and Weiss 1985) is used to optimise all the parameters in a maximum likelihood model, possibly modifying the outcome of the line search. The GM functions in the Spatial Econometrics toolbox use an included function `minz` contributed by Michael Cliff; for bounded line search, the `MATLAB` `fminbnd` function also based on Brent (1973) is used, while when more than one parameter is to be optimised, the `MATLAB` `fminsearch` function is used — it is an implementation of the Nelder-Mead simplex algorithm. Finally, the `Stata` implementations of spatial econometric models use `optimize` mechanisms described in the help page for `maximize`, referred to in Drukker, Prucha, and Raciborski (2011b); Drukker *et al.* (2011c). The default numerical optimizer is `"nr"`, a Stata-modified Newton-Raphson algoritm, but other algorithms may be chosen (Gould, Pitblado, and Poi 2010).

1.2. Data set

We were fortunate to be able to use the simulated US Driving Under the Influence (DUI) county data set used in Drukker, Peng, Prucha, and Raciborski (2011a); Drukker *et al.* (2011c,b) for our comparison. The data used is simulated for 3109 counties (omitting Alaska, Hawaii, and US territories), and uses simulations from variables used by Powers and Wilson (2004). The counties are taken from an ESRI Shapefile downloaded from the US Census.[8] The dependent variable `dui` is defined as the alcohol-related arrest rate per 100,000 daily vehicle miles traveled (DVMT). The explanatory variables include `police` (number of sworn officers per 100,000 DVMT); `nondui` (non-alcohol-related arrests per 100,000 DVMT); `vehicles` (number of registered vehicles per 1,000 residents), and `dry` (a dummy for counties that prohibit alcohol sale within their borders, about 10% of counties). A further dummy variable `elect` takes values of 1 if a county government faces an election, 0 otherwise, and has 295 non-zero entries. Descriptive statistics for the simulated DUI data set are shown in Table 1.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
dui	15.01	19.88	20.83	20.84	21.82	26.62
police	25.28	29.73	30.72	30.70	31.67	36.78
nondui	18.01	34.41	40.19	40.98	46.74	76.50
vehicles	390.40	479.90	501.30	501.80	523.60	625.90

Table 1: Descriptive statistics for the dependent variable and the non-dummy explanatory variables, simulated DUI data set.

Table 2 shows the stylised results from estimating the relationship between the simulated dependent variable and four explanatory variables using least squares. Powers and Wilson

[7] `http://netlib.sandia.gov/port/`.

[8] `ftp://ftp2.census.gov/geo/tiger/TIGER2008/ts_2008_us_county00.zip`.

	R `lm`	Stata `reg`	MATLAB SE `ols`	Python PySAL `OLS`
(Intercept)	-5.4428237	-5.4428237	-5.4428237	-5.4428237
	(0.229431)	(0.229431)	(0.229431)	(0.229431)
police	0.5990957	0.5990957	0.5990957	0.5990957
	(0.014935)	(0.014935)	(0.014935)	(0.014935)
nondui	0.0002746	0.0002746	0.0002746	0.0002746
	(0.001088)	(0.001088)	(0.001088)	(0.001088)
vehicles	0.0156842	0.0156842	0.0156842	0.0156842
	(0.000670)	(0.000670)	(0.000670)	(0.000670)
dry	0.1060904	0.1060904	0.1060904	0.1060904
	(0.035011)	(0.035011)	(0.035011)	(0.035011)

Table 2: OLS estimation results for four implementations, simulated DUI data set (standard errors in parentheses).

(2004, p. 331) found that "there is no significant relationship between prohibition status and the DUI arrest rate when controlling for the proportionate number of sworn officers and the non-DUI arrest rate per officer" when examining data for 75 counties in Arkansas. Their best model has an adjusted R^2 of 0.397, while the simulated data has a higher value of 0.850, and the explanatory variables with the exception of `nondui` are all significant. As can be seen easily, all four implementations, R `lm`, Stata `reg`, MATLAB Spatial Econometrics Toolbox `ols`, and Python PySAL `OLS` give identical results, so confirming that all four applications are handling the same data.

Drukker *et al.* (2011c) do not specify how spatial dependence was introduced into the dependent variable and/or residuals. Using the 3109 county `SpatialPolygons` read from the ESRI Shapefile mentioned above, we recreated the Queen contiguity list of neighbours with `poly2nb` in **spdep**. The descriptive statistics for the neighbour object shown by Drukker *et al.* (2011a, p. 9) match ours exactly:

```
> library(rgdal)
> county <- readOGR("tl_2008_us_county00", "tl_2008_us_county00")
> drop <- county$STATEFP00 == "02" | county$STATEFP00 == "15" | as.character(county$STATEFP00) > "56"
> ccounty <- county[!drop,]
> library(rgeos)
> strt <- gUnarySTRtreeQuery(ccounty)
> library(spdep)
> nblist <- poly2nb(ccounty, foundInBox=strt)
> nblist

Neighbour list object:
Number of regions: 3109
Number of nonzero links: 18474
Percentage nonzero weights: 0.1911259
Average number of links: 5.942104
```

We were perplexed to find that the results from fitting the ML SARAR model (to be described below) given by Drukker *et al.* (2011c, p. 8), did not match those we obtained in R or Stata. We initially adopted row-standardised spatial weights, $\mathbf{W}$, where the county contiguities c_{ij}, taking values of 1 if contiguous, sharing at least one boundary point, and 0 otherwise, are row-standardised by dividing by row sums:

$$w_{ij} = \frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}}. \tag{1}$$

It turned out (personal communication, Rafal Raciborski) that the spatial weights used in estimation in Drukker *et al.* (2011c) were in fact minmax-normalised, where the required operation is:

$$w_{ij} = \frac{c_{ij}}{m}. \tag{2}$$

where m is the minmax value, the minimum of the pair of maxima of row and column sums:

$$m = \min(\max \sum_{i=1}^{n} c_{ij}, \max \sum_{j=1}^{n} c_{ij}). \tag{3}$$

With symmetric neighbours as seen in this case of contiguities, the maximum row and column sums are equal by definition, here 14, so the weights are all 1/14 if counties are contiguous, and 0 otherwise. Using these weights, we could reproduce the ML SARAR estimation results given by Drukker *et al.* (2011c, pp. 8), and could confirm that the underlying contiguities were the same as those used in the Stata documentation.

It is also possible that the spatial dependence in the simulated DUI data set was introduced using minmax-normalised spatial weights, as the standard deviates of Moran's I statistic are 2.374 and 2.434 respectively for the dependent variable and the least squares residuals using minmax-normalisation, and 1.623 and 1.554 respectively using row standardisation; the strength of the spatial dependency is much greater in the former case. We, however, chose to use row standardisation in this comparison, partly because row standardisation is more frequently encountered in applied work, and partly because weak spatial dependence may be more challenging for implementations that stronger dependence.

2. A general spatial model

The present discussion is almost entirely based on Kelejian and Prucha (2010), Drukker, Egger, and Prucha (2013), Arraiz, Drukker, Kelejian, and Prucha (2010) and Drukker *et al.* (2011b) that provide some important extensions to Kelejian and Prucha (1998, 1999). The model is quite general and it allows for some of the right hand side variables to be endogenous. Specifically, the point of departure will be the following Cliff-Ord spatial model:

$$\mathbf{y} = \mathbf{Y}\pi + \mathbf{X}\beta + \rho_{\text{Lag}}\mathbf{W}\mathbf{y} + \mathbf{u} \tag{4}$$

where $\mathbf{y}$ is an $n \times 1$ vector of observations on the dependent variable, $\mathbf{Y}$ is an $n \times p$ matrix of observations on p endogenous variables, $\mathbf{X}$ is a $n \times k$ matrix of observations on k exogenous variable, $\mathbf{W}$ is an $n \times n$ observed and non-stochastic spatial weighting matrix and, consequently, $\mathbf{Wy}$ is an $n \times 1$ variable that is generally referred to as the spatial lag variable; π and β are corresponding parameters; and ρ_{Lag} is the spatial autoregressive coefficient. Given the presence of $\mathbf{Y}$, the model can be viewed as a representation of a single equation of as system of equations.[9]

The error vector $\mathbf{u}$ follows a spatial autoregressive process of the form:

$$\mathbf{u} = \rho_{\mathrm{Err}}\mathbf{M}\mathbf{u} + \varepsilon \tag{5}$$

where ρ_{Err} is a scalar parameter generally referred to as the spatial autoregressive parameter, $\mathbf{M}$ is an $n \times n$ spatial weighting matrix that may or may not be the same as $\mathbf{W}$,[10] $\mathbf{Mu}$ is an $n \times 1$ vector of observation on the spatially lagged vector of residuals.

An alternative, more compact way to express the same model is:

$$\mathbf{y} = \mathbf{Z}\delta + \mathbf{u} \tag{6}$$

where $\mathbf{Z} = [\mathbf{Y}, \mathbf{X}, \mathbf{Wy}]$ is the set of all (endogenous and exogenous) explanatory variables, and $\delta = [\pi^\top, \beta^\top, \rho_{\mathrm{Lag}}]^\top$ is the corresponding vector of parameters. Finally, the assumption on which the maximum likelihood relies is that $\varepsilon \sim N(0, \sigma^2)$. In the GMM approach, the estimation theory is developed both under the assumptions that the innovations ε are homoskedastic, and heteroskedastic of unknown form.

2.1. Notation

Here we adopt the notation ρ_{Lag} for the spatial autoregressive parameter on the spatially lagged dependent variable $\mathbf{y}$, and ρ_{Err} for the spatial autoregressive parameter on the spatially lagged residuals. In Ord (1975), ρ is used for both parameters, but subsequently two schools have developed, with Anselin (1988b) and LeSage and Pace (2009) (and many others) using ρ for the spatial autoregressive parameter on the spatially lagged dependent variable $\mathbf{y}$, and λ for the spatial autoregressive parameter on the spatially lagged residuals. Kelejian and Prucha (1998, 1999) (and many others particularly in the econometrics literature) adopt the opposite notation, using λ for the spatial autoregressive parameter on the spatially lagged dependent variable $\mathbf{y}$, and ρ for the spatial autoregressive parameter on the spatially lagged residuals. Our choice is an attempt to disambiguate in this comparison, not least because the different software implementations use one or other scheme, so inviting confusion.

The names used for models also vary between software implementations, so that the general model given in Equation 4 is also known as a SARAR model, with first order autoregressive processes in the dependent variable and the residuals. The same model is termed a SAC model in LeSage and Pace (2009) and the MATLAB Spatial Econometrics Toolbox. This model and all its derivatives are simultaneous autoregressive (SAR) models in the sense used in spatial statistics (see also Ripley 1981, p. 88), in contrast to conditional autoregressive (CAR) models often used in disease mapping and other application areas. Other issues in the naming of models will be mentioned in the next session.

[9] A spatial lag of the matrix of observations on the exogenous variables $\mathbf{WX}$ may be (easily) added to the model. To avoid unnecessary complications since no particular difficulties arise in the estimation of such formulation of the model, we will not consider it in the present section. However, in the maximum likelihood implementation we will discuss the estimation of a model that includes $\mathbf{WX}$ among the set of regressors.

[10] In many empirical applications $\mathbf{W}$ and $\mathbf{M}$ are assumed to be identical. However, both R and Stata allow them to differ. From an implementation perspective, the main complication is in the definition of the instrument matrix. We will discuss this issue in the next section.

2.2. Restrictions on the general model

The general model (Equation 4) may be restricted by setting $\pi = 0$ to remove the endogenous variables. All of the models considered when comparing maximum likelihood implementations, and many GMM implementations, impose this restriction. The spatial lag model is formed as a special case with $\rho_{\text{Err}} = 0$, and the spatial error model with $\rho_{\text{Lag}} = 0$. The spatial error model with no endogenous variables is:

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{u}, \mathbf{u} = \rho_{\text{Err}}\mathbf{M}\mathbf{u} + \varepsilon \tag{7}$$

This model is known as SEM (spatial error model) in LeSage and Pace (2009), and SARE (spatial autoregressive error model) in Drukker *et al.* (2011c).

The spatial lag model with no endogenous variables is:

$$\mathbf{y} = \mathbf{X}\beta + \rho_{\text{Lag}}\mathbf{W}\mathbf{y} + \varepsilon \tag{8}$$

This model is known as SAR (spatial autoregressive model) in both LeSage and Pace (2009) and Drukker *et al.* (2011c), but may be confused with the spatial statistics abbreviation SAR — simultaneous autoregressive model; all spatial econometric SAR models are spatial statistics SAR models, but the reverse is not true.

In fitting spatial lag models and by extension any model including the spatially lagged dependent variable, it has emerged over time that, unlike the spatial error model, the spatial dependence in the parameter ρ_{Lag} feeds back, obliging analysts to base interpretation not on the fitted parameters β, but rather on correctly formulated impact measures, as discussed in references given on p. 2.

This feedback comes from the data generation process of the spatial lag model (and by extension in the general model). Rewriting:

$$\begin{aligned}
\mathbf{y} - \rho_{\text{Lag}}\mathbf{W}\mathbf{y} &= \mathbf{X}\beta + \varepsilon \\
(\mathbf{I} - \rho_{\text{Lag}}\mathbf{W})\mathbf{y} &= \mathbf{X}\beta + \varepsilon \\
\mathbf{y} &= (\mathbf{I} - \rho_{\text{Lag}}\mathbf{W})^{-1}\mathbf{X}\beta + (\mathbf{I} - \rho_{\text{Lag}}\mathbf{W})^{-1}\varepsilon
\end{aligned}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix. This means that the expected impact of a unit change in an exogenous variable r for a single observation i on the dependent variable y_i is no longer equal to β_r, unless $\rho_{\text{Lag}} = 0$. The awkward $n \times n$ $S_r(\mathbf{W}) = ((\mathbf{I} - \rho_{\text{Lag}}\mathbf{W})^{-1}\mathbf{I}\beta_r)$ matrix term is needed to calculate impact measures for the spatial lag model, and similar forms for other models including the general model, when $\rho_{\text{Lag}} \neq 0$.[11]

3. Comparing GMM implementations

The estimation of the general SARAR(1,1) model consists of various steps. Given the simultaneous presence of the endogenous variables on the right hand side of Equation 4 and the spatially autocorrelated residuals, these steps will alternate IV and GMM estimators. These

[11] When an additional endogenous variable is included, one should make additional (and generally restrictive) assumptions to be able to calculate this impact measures. Since, to the best of our knowledge, no theory is available for this case, we will exclude it from our discussion.

estimators will be based on a set of linear and quadratic moment conditions of the form:

$$\begin{aligned} E\mathbf{H}^\top\varepsilon &= 0 \qquad (9)\\ E\varepsilon^\top\mathbf{A}\varepsilon &= 0 \qquad (10)\end{aligned}$$

where $\mathbf{H}$ is an $n \times p$ non-stochastic matrix of instruments, and $\mathbf{A}$ is an $n \times n$ weighting matrix whose specification will be introduced later. At this point, it is useful to introduce the spatial Cochrane-Orcutt transformation of the model, that is:

$$\mathbf{y}^\star = \mathbf{Z}^\star\delta + \varepsilon \qquad (11)$$

where $\mathbf{y}^\star = \mathbf{y} - \rho_{\mathrm{Err}}\mathbf{My}$ and $\mathbf{Z}^\star = \mathbf{Z} - \rho_{\mathrm{Err}}\mathbf{MZ}$.

As a preview of the estimation steps, an initial IV estimator of δ leads to a set of consistent residuals. This vector of residuals constitutes the base for the derivation of the quadratic moment conditions that provide a first consistent estimate for the autoregressive parameter ρ_{Err}.[12] An estimate of δ is obtained from the transformed model after replacing the true value of ρ_{Err} with its consistent estimate obtained in the previous step. Finally, in a new GM iteration, it is possible to obtain a consistent and efficient estimate of ρ_{Err} based on generalized spatial two stage least square residuals. The asymptotic variance-covariance matrix for the coefficients is then calculated using the estimate of δ, the residuals, and the estimate of the spatial coefficient ρ_{Err}.

3.1. SARAR model

For the case of no additional endogenous variables other than the spatial lag, the "ideal" instruments should be expressed in terms of $E(\mathbf{Wy})$. This is simply because the best instruments for the right hand side variables are the conditional means and, since $\mathbf{X}$ and $\mathbf{MX}$ are non-stochastic, we can simply focus on the spatial lags $\mathbf{Wy}$ and $\mathbf{MWy}$ (see equation 11). Given the reduced form of the model

$$\mathbf{y} = (\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W})^{-1}(\mathbf{X}\beta + \mathbf{u}) \qquad (12)$$

it follows that the best instruments can be expressed in terms of the $E(\mathbf{Wy}) = \mathbf{W}(\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W})^{-1}\mathbf{X}\beta$ (Lee 2003, 2007; Kelejian, Prucha, and Yuzefovich 2004; Das, Kelejian, and Prucha 2003). Given that the roots of $\rho_{\mathrm{Lag}}\mathbf{W}$ are less than one in absolute value, Kelejian and Prucha (1999) suggested to generate an approximation to the best instruments (say $\mathbf{H}$) as the subset of the linearly independent columns of

$$\mathbf{H} = (\mathbf{X}, \mathbf{WX}, \mathbf{W^2X}, \ldots, \mathbf{W^qX}, \mathbf{MX}, \mathbf{MWX}, \ldots, \mathbf{MW^qX}) \qquad (13)$$

where q is a pre-selected finite constant and is generally set to 2 in applied studies.[13] The inclusion of instruments involving $\mathbf{M}$ in the instrument matrix $\mathbf{H}$ is only needed for the formulation of instrumental variable estimators applied to the spatially Cochrane-Orcutt transformed model. At a minimum the instruments should include the linearly independent columns of $\mathbf{X}$

[12] The estimate of ρ_{Err} could be, in turn, used to define a weighting matrix for the moment conditions in order to obtain a consistent and efficient estimator. While possible, this would imply some complications in computational term. However, the ρ_{Err} is already consistent and, therefore, can be used to define the transformed model.

[13] While R and Stata set $q = 2$ by default, Pysal leave the choice open for the users.

and $\mathbf{MX}$. In a more general setting where additional endogenous variables are present, since the system determining $\mathbf{y}$ and $\mathbf{Y}$ is not completely specified, the optimal instruments are not known. In this case it may be reasonable to use a set of instruments as above where $\mathbf{X}$ is augmented by other exogenous variables that are expected to determine $\mathbf{Y}$. Unless additional information is available, it is not recommended to take the spatial lag of these additional exogenous variables.

The starting point for the estimation of ρ_{Err} are the two following quadratic moment conditions expressed as functions of the innovation ε

$$E[\varepsilon^\top \mathbf{A}_s \varepsilon] = 0 \tag{14}$$

where the matrices $\mathbf{A}_s$ are such that $tr(\mathbf{A}_s) = 0$. Furthermore, under heteroskedasticity it is also assumed that the diagonal elements of the matrices $\mathbf{A}_s$ are zero. The reason for this is that simplifies the formulae for the variance-covariance matrix (i.e., only depends on second moments of the innovations and not on third and fourth moments). Specific suggestions for $\mathbf{A}_s$ are given below. In general, such choices will depend on whether or not the model assumes heteroskedasticity.

As we already mentioned, the actual estimation procedure consists of several steps, which we will review next. Our review of the estimation procedure will be intentionally brief, as we refer the interested reader to more specific literature (Kelejian and Prucha 2010; Drukker *et al.* 2013; Arraiz *et al.* 2010; Drukker *et al.* 2011b).

Step 1.1 - 2SLS estimator

The initial set of estimates are obtained from the untransformed model using the matrix of instruments H, yielding to:

$$\tilde{\delta} = (\tilde{\mathbf{Z}}^\top \tilde{\mathbf{Z}})^{-1} \tilde{\mathbf{Z}}^\top \mathbf{y} \tag{15}$$

where $\tilde{\mathbf{Z}} = \mathbf{PZ} = [\mathbf{X}, \widehat{\mathbf{Wy}}]$, $\widehat{\mathbf{Wy}} = \mathbf{PWy}$, $\mathbf{P} = \mathbf{H}(\mathbf{H}^\top \mathbf{H})^{-1}\mathbf{H}^\top$ and $\mathbf{H}$ was specified above. The estimate $\tilde{\delta}$ can then be used to obtain a initial estimate of the regression residuals, say $\tilde{\mathbf{u}}$. The GMM estimator of the next step is based on the assumption that this regression residuals are consistent.

Step 1.2 - Initial GMM estimator

Let $\tilde{\delta}$ be the initial estimator of δ obtained from Step 1.1, let $\tilde{\mathbf{u}} = \mathbf{y} - \mathbf{Z}\tilde{\delta}$, and $\bar{\tilde{\mathbf{u}}} = M\tilde{\mathbf{u}}$. Then, we can operationalize the sample moments corresponding to Equation 14, that is:

$$m(\tilde{\delta}, \rho_{\text{Err}}) = n^{-1} \begin{bmatrix} (\tilde{\mathbf{u}} - \rho_{\text{Err}}\bar{\tilde{\mathbf{u}}})^\top \mathbf{A}_1 (\tilde{\mathbf{u}} - \rho_{\text{Err}}\bar{\tilde{\mathbf{u}}}) \\ \vdots \\ (\tilde{\mathbf{u}} - \rho_{\text{Err}}\bar{\tilde{\mathbf{u}}})^\top \mathbf{A}_s (\tilde{\mathbf{u}} - \rho_{\text{Err}}\bar{\tilde{\mathbf{u}}}) \end{bmatrix}$$

It is convenient to rewrite $m(\tilde{\delta}, \rho_{\text{Err}})$ as an explicit function of the parameters ρ_{Err} and ρ^2_{Err} as in the following expression

$$m(\tilde{\delta}, \rho_{\text{Err}}) = \tilde{\mathbf{\Gamma}} \begin{bmatrix} \rho_{\text{Err}} \\ \rho^2_{\text{Err}} \end{bmatrix} - \tilde{\gamma}$$

where

$$\tilde{\mathbf{\Gamma}} = n^{-1}\begin{bmatrix} \tilde{\mathbf{u}}^\top(\mathbf{A}_1 + \mathbf{A}_1^\top)\bar{\tilde{\mathbf{u}}} & -\bar{\tilde{\mathbf{u}}}\mathbf{A}_1\bar{\tilde{\mathbf{u}}} \\ \vdots & \vdots \\ \tilde{\mathbf{u}}^\top(\mathbf{A}_s + \mathbf{A}_s^\top)\bar{\tilde{\mathbf{u}}} & -\bar{\tilde{\mathbf{u}}}\mathbf{A}_s\bar{\tilde{\mathbf{u}}} \end{bmatrix} \quad \text{and} \quad \tilde{\gamma} = n^{-1}\begin{bmatrix} \tilde{\mathbf{u}}^\top\mathbf{A}_1\tilde{\mathbf{u}} \\ \vdots \\ \tilde{\mathbf{u}}^\top\mathbf{A}_s\tilde{\mathbf{u}} \end{bmatrix}$$

Therefore, the initial GMM estimator for ρ_{Err} is obtained by simply minimizing the following relation:

$$\tilde{\rho}_{\mathrm{Err}} = \arg\min\left\{\left[\tilde{\mathbf{\Gamma}}\begin{pmatrix}\rho_{\mathrm{Err}} \\ \rho^2_{\mathrm{Err}}\end{pmatrix} - \tilde{\gamma}\right]^\top\left[\tilde{\mathbf{\Gamma}}\begin{pmatrix}\rho_{\mathrm{Err}} \\ \rho^2_{\mathrm{Err}}\end{pmatrix} - \tilde{\gamma}\right]\right\}.$$

The expression above can be interpreted as a nonlinear least square system of equations. The initial estimate is thus obtained as a solution of the above system.

We finally need an expression for the matrices $\mathbf{A}_s$. Drukker *et al.* (2013) suggest, for the homoskedastic case, the following expressions:[14]

$$\mathbf{A}_1 = \left\{1 + [n^{-1}tr(\mathbf{M}^\top\mathbf{M})]^2\right\}^{-1}[\mathbf{M}^\top\mathbf{M} - n^{-1}tr(\mathbf{M}^\top\mathbf{M})\mathbf{I}] \tag{16}$$

and

$$\mathbf{A}_2 = \mathbf{M} \tag{17}$$

On the other hand, when heteroskedasticity is assumed, Kelejian and Prucha (2010) recommend the following expressions for $\mathbf{A}_1$ and $\mathbf{A}_2$:[15]

$$\mathbf{A}_1 = \mathbf{M}^\top\mathbf{M} - diag(\mathbf{M}^\top\mathbf{M}) \tag{18}$$

and

$$\mathbf{A}_2 = \mathbf{M} \tag{19}$$

Step 2.1 - GS2SLS estimator

Using the estimate of ρ_{Err} obtained in Step 1.2, one can take a Cochrane-Orcutt transformation of the model as in Equation 11 and estimate it again by 2SLS. The entire procedure is known in the literature as generalized spatial two stage least square (GS2SLS):

$$\hat{\delta} = [\widehat{\mathbf{Z}}^{\star\top}\mathbf{Z}]^{-1}\widehat{\mathbf{Z}}^{\star\top}\mathbf{y} \tag{20}$$

where $\widehat{\mathbf{Z}}^\star = \mathbf{P}\mathbf{Z}^\star$, $\mathbf{Z}^\star = \mathbf{Z} - \tilde{\rho}_{\mathrm{Err}}\mathbf{W}\mathbf{Z}$, $\mathbf{y}^\star = \mathbf{y} - \tilde{\rho}_{\mathrm{Err}}\mathbf{W}\mathbf{y}$ and $\mathbf{P} = \mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top$.

Step 2.2 - Consistent and efficient GMM estimator

In the second sub-step of the second step, an efficient GMM estimator of ρ_{Err} based on the GS2SLS residuals is obtained by minimizing the following expression:

$$\hat{\rho}_{\mathrm{Err}} = \arg\min\left\{\left[\hat{\mathbf{\Gamma}}\begin{pmatrix}\rho_{\mathrm{Err}} \\ \rho^2_{\mathrm{Err}}\end{pmatrix} - \hat{\gamma}\right]^\top(\hat{\mathbf{\Psi}}^{\rho_{\mathrm{Err}}\rho_{\mathrm{Err}}})^{-1}\left[\hat{\mathbf{\Gamma}}\begin{pmatrix}\rho_{\mathrm{Err}} \\ \rho^2_{\mathrm{Err}}\end{pmatrix} - \hat{\gamma}\right]\right\}$$

[14] The scaling factor $v = \left\{1 + [n^{-1}tr(\mathbf{M}^\top\mathbf{M})]^2\right\}^{-1}$ is needed to obtain the same estimator of Kelejian and Prucha (1998, 1999)

[15] For the heteroskedastic case, Arraiz *et al.* (2010) derive a consistent and efficient GMM estimator based on 2SLS residuals. While we will not review this estimator in the present section, the R and PySal implementations allow for this additional step.

where $\hat{\boldsymbol{\Psi}}^{\rho_{\mathrm{Err}}\rho_{\mathrm{Err}}}$ is an estimator for the variance-covariance matrix of the (normalized) sample moment vector based on GS2SLS residuals. This estimator differs for the cases of homoskedastic and heteroskedastic errors.

For the homoskedastic case, the r, s (with r, s=1,2) element of $\hat{\boldsymbol{\Psi}}^{\rho_{\mathrm{Err}}\rho_{\mathrm{Err}}}$ is given by:

$$\begin{aligned}\hat{\boldsymbol{\Psi}}^{\rho_{\mathrm{Err}}\rho_{\mathrm{Err}}}_{rs} &= [\tilde{\sigma}^2]^2(2n)^{-1}tr[(\mathbf{A}_r + \mathbf{A}_{\mathbf{r}}^\top)(\mathbf{A}_s + \mathbf{A}_{\mathbf{s}}^\top)] \\ &+ \tilde{\sigma}^2 n^{-1}\tilde{\mathbf{a}}_r^\top \tilde{\mathbf{a}}_s^\top \\ &+ n^{-1}(\tilde{\mu}^{(4)} - 3[\tilde{\sigma}^2]^2) vec_D(\mathbf{A}_r)^\top vec_D(\mathbf{A}_s) \\ &+ n^{-1}\tilde{\mu}^{(3)}[\tilde{\mathbf{a}}_r^\top vec_D(\mathbf{A}_s) + \tilde{\mathbf{a}}_s^\top vec_D(\mathbf{A}_r)] \qquad (21)\end{aligned}$$

where
$\tilde{\mathbf{a}}_r = \hat{\mathbf{T}}\tilde{\alpha}_r$,
$\hat{\mathbf{T}} = \mathbf{H}\hat{\mathbf{P}}$,
$\hat{\mathbf{P}} = \hat{\mathbf{Q}}_{\mathbf{HH}}^{-1}\hat{\mathbf{Q}}_{\mathbf{HZ}}[\hat{\mathbf{Q}}_{\mathbf{HZ}}^\top \hat{\mathbf{Q}}_{\mathbf{HH}}^{-1}\hat{\mathbf{Q}}_{\mathbf{HZ}}^\top]^{-1}$,
$\hat{\mathbf{Q}}_{\mathbf{HH}}^{-1} = (n^{-1}\mathbf{H}^\top\mathbf{H})$,
$\hat{\mathbf{Q}}_{\mathbf{HZ}} = [n^{-1}\mathbf{H}^\top\mathbf{Z}]$,
$\mathbf{Z} = (\mathbf{I} - \tilde{\rho}_{\mathrm{Err}}\mathbf{M})\mathbf{Z}$,
$\tilde{\alpha}_r = -n^{-1}[\mathbf{Z}^\top(\mathbf{A}_r + \mathbf{A}_r^\top)\hat{\varepsilon}]$,
$\hat{\sigma}^2 = n^{-1}\hat{\varepsilon}^\top\hat{\varepsilon}$,
$\hat{\mu}^{(3)} = n^{-1}\sum_{i=1}^n \hat{\varepsilon}^3$, and
$\hat{\mu}^{(3)} = n^{-1}\sum_{i=1}^n \hat{\varepsilon}^4$.

For the heteroskedastic case, the r, s (with r, s=1,2) element of $\hat{\boldsymbol{\Psi}}^{\rho_{\mathrm{Err}}\rho_{\mathrm{Err}}}$ is given by:

$$\hat{\boldsymbol{\Psi}}^{\rho_{\mathrm{Err}}\rho_{\mathrm{Err}}}_{rs} = (2n)^{-1}tr[(\mathbf{A}_r + \mathbf{A}_{\mathbf{r}}^\top)\hat{\boldsymbol{\Sigma}}(\mathbf{A}_s + \mathbf{A}_{\mathbf{s}}^\top)\hat{\boldsymbol{\Sigma}}] + n^{-1}\tilde{\mathbf{a}}_r^\top\hat{\boldsymbol{\Sigma}}\tilde{\mathbf{a}}_s^\top \qquad (22)$$

where, $\hat{\boldsymbol{\Sigma}}$ is a diagonal matrix whose ith diagonal element is $\hat{\varepsilon}^2$, and $\hat{\varepsilon}$, and $\tilde{\mathbf{a}}_r$ are defined as above.[16]

Having completed the previous steps, a consistent estimator for the asymptotic VC matrix $\boldsymbol{\Omega}$ can be derived. The estimator is given by $n\widehat{\boldsymbol{\Omega}}$ where:

$$\widehat{\boldsymbol{\Omega}} = \begin{pmatrix} \widehat{\boldsymbol{\Omega}}_{\delta\delta} & \widehat{\boldsymbol{\Omega}}_{\delta\rho_{\mathrm{Err}}} \\ \widehat{\boldsymbol{\Omega}}_{\delta\rho_{\mathrm{Err}}^\top} & \widehat{\boldsymbol{\Omega}}_{\rho_{\mathrm{Err}}\rho_{\mathrm{Err}}} \end{pmatrix}$$

where
$\widehat{\boldsymbol{\Omega}}_{\delta\delta} = \hat{\mathbf{P}}^\top\widehat{\boldsymbol{\Psi}}_{\delta\delta}\hat{\mathbf{P}}$,
$\widehat{\boldsymbol{\Omega}}_{\delta\rho_{\mathrm{Err}}} = \hat{\mathbf{P}}^\top\widehat{\boldsymbol{\Psi}}_{\delta\rho_{\mathrm{Err}}}[\widehat{\boldsymbol{\Psi}}_{\rho_{\mathrm{Err}}\rho_{\mathrm{Err}}}]^{-1}\widehat{\mathbf{J}}[\widehat{\mathbf{J}}^\top[\widehat{\boldsymbol{\Psi}}_{\rho_{\mathrm{Err}}\rho_{\mathrm{Err}}}]^{-1}\widehat{\mathbf{J}}]^{-1}$,
$\widehat{\boldsymbol{\Omega}}_{\rho_{\mathrm{Err}}\rho_{\mathrm{Err}}} = [\widehat{\mathbf{J}}^\top[\widehat{\boldsymbol{\Psi}}_{\rho_{\mathrm{Err}}\rho_{\mathrm{Err}}}]^{-1}\widehat{\mathbf{J}}]^{-1}$
$\widehat{\mathbf{J}} = \widehat{\boldsymbol{\Gamma}}[1, 2\hat{\rho}_{\mathrm{Err}}]^\top$

Some of the element of the VC matrix were defined before. Once more, the estimators for $\widehat{\boldsymbol{\Psi}}_{\delta\delta}(\hat{\rho}_{\mathrm{Err}})$, and $\widehat{\boldsymbol{\Psi}}_{\delta\rho_{\mathrm{Err}}}(\hat{\rho}_{\mathrm{Err}})$ are different for the homoskedastic and the heteroskedastic cases.

[16] It is worth noticing here that the second term in equation (22) limits to zero when there are only exogenous variables in the model. As we will see later, while the implementations in R and Stata produce an estimate of this term, PySAL simply assumes it to be zero.

In the homoskedastic case:
$\widehat{\boldsymbol{\Psi}}_{\delta\delta} = \hat{\sigma^2}\widehat{\mathbf{Q}}_{\mathbf{HH}}$
$\widehat{\boldsymbol{\Psi}}_{\delta\rho_{\text{Err}}} = \hat{\sigma}^2 n^{-1}\mathbf{H}^\top[\mathbf{a}_1, \mathbf{a}_2] + \mu^{(3)}n^{-1}\mathbf{H}^\top[vec_D(\mathbf{A}_1), vec_D(\mathbf{A}_2)]$.
In the heteroskedastic case:
$\widehat{\boldsymbol{\Psi}}_{\delta\delta} = n^{-1}\mathbf{H}^\top\hat{\boldsymbol{\Sigma}}\mathbf{H}$
$\widehat{\boldsymbol{\Psi}}_{\delta\rho_{\text{Err}}} = n^{-1}\mathbf{H}^\top\hat{\boldsymbol{\Sigma}}[\mathbf{a}_1\mathbf{a}_2]$.

Homoskedasticity with and without additional endogenous variables

There are various implementations of the GMM general model (under homoskedasticity) that are presented in Table 3.[17] Some of them are based on the Kelejian and Prucha (1999) moment conditions (i.e.„ the R function `gstsls` available from **spdep**, the Spatial Econometrics Toolbox function `sac_gmm`, and PySAL `GM_Combo`). The remaining implementations are based on the Drukker *et al.* (2013) moment conditions (i.e.„ the function `spreg` available from **sphet**, the Stata function `spreg gs2sls`, and PySAL `GM_Combo_Hom`).

[17] For simplicity, in all models it is assumed that $\mathbf{W}$ and $\mathbf{M}$ are the same. As mentioned, this assumption will be relaxed later in the paper.

	R gstsls	R spreg	Stata spreg	PySAL GM_Combo	PySAL GM_Combo_Hom	SE sac_gmm
(Intercept)	-6.409919	-6.409969	-6.409968	-6.409919	-6.409969	-6.403747
	(0.418363)	(0.416359)	(0.416359)	(0.417959)	(0.416359)	(0.417963)
police	0.598107	0.598102	0.598102	0.598107	0.598102	0.598107
	(0.014918)	(0.014907)	(0.014907)	(0.014903)	(0.014907)	(0.014918)
nondui	0.000247	0.000247	0.000247	0.000247	0.000247	0.000247
	(0.001087)	(0.001086)	(0.001086)	(0.001086)	(0.001086)	(0.001087)
vehicles	0.015712	0.015713	0.015713	0.015712	0.015713	0.015712
	(0.000669)	(0.000668)	(0.000668)	(0.000668)	(0.000668)	(0.000669)
dry	0.106088	0.106098	0.106098	0.106088	0.106098	0.106088
	(0.034962)	(0.034927)	(0.034927)	(0.034929)	(0.034927)	(0.034962)
ρ_{Lag}	0.046928	0.046932	0.046932	0.046928	0.046932	0.046926
	(0.016982)	(0.016927)	(0.016927)	(0.016966)	(0.016927)	(0.016982)
ρ_{Err}	0.000957	-0.005621	-0.005621	0.000957	-0.005621	0.000957
	(0.000000)	(0.034984)	(0.034984)	(0.000000)	(0.034984)	(0.009316)

Table 3: SARAR estimation results for six implementations, DUI data set (standard errors in parentheses).

The results for all of them are reported in Table 3. Specifically, the results based on the Kelejian and Prucha (1999) moment conditions correspond to columns one, four, and six. A glance at this columns reveals that, while the estimated coefficients obtained from the function `gstsls` in R and PySAL `GM_Combo` are identical (up to the sixth digit), those obtained from the Spatial Econometrics Toolbox function `sac_gmm` are slightly different. The discrepancies with `sac_gmm` in MATLAB are related to a different sets of instruments. The SE Toolbox uses two different sets of instruments: one for estimating the "original" model, one for estimating the same model after the Cocran-Orchutt transformation. In the first step of the procedure, the matrix of instruments contains all of the linearly independent columns of $\mathbf{X}$, $\mathbf{WX}$, and $\mathbf{W}^2\mathbf{X}$ and it is the same for all three implementations. However, to estimate the transformed model, the matrix of instruments employed by MATLAB includes, along with the intercept (untransformed), the transformed exogenous variables (say $\mathbf{X}^{\star}$), and the spatial lags of them (say, $\mathbf{WX}^{\star}$ and $\mathbf{W}^2\mathbf{X}^{\star}$).[18] The R and PySAL implementations use $\mathbf{X}^{\star}$ (which may or may not include an intercept), and then spatial lags of $\mathbf{X}$. Finally, there are also differences in reported standard errors between the three implementations. These differences pertain to a degrees of freedom correction in the variance-covariance matrix operated in R and MATLAB. However, the same correction is available as an option from PySAL.[19]

It should also be noted that of the three available implementations, the SE Toolbox is the only one that produces inference for ρ_{Err}.[20]

Let us turn now to the results shown in columns two, three, and five. From table 3 it can be observed that, apart from a different decimal for the intercept calculated in Stata, all implementations otherwise match exactly. The only major differentiation among the three implementations is the possibility of setting a different matrix $\mathbf{A}_1$ in PySAL. As noted in Anselin (2013), there may be a problem with one of the sub-matrix (i.e., $\mathbf{\Omega}_{\delta\rho_{\mathrm{Err}}}$) of the variance-covariance matrix of the estimated coefficients. The standard result that the variance-covariance matrix must be block-diagonal between the model coefficients and the error parameter may be invalidated by certain choices of $\mathbf{A}_1$ (e.g., the one used by Drukker *et al.* 2013). The options of choosing alternative $\mathbf{A}_1$ in PySAL are meant to avoid this problem.

As in Drukker *et al.* (2011b), the set of explanatory variables includes `police`. Undoubtably, the size of the `police` force may be related with the arrest rates (`dui`). As a consequence, `police` is treated as an endogenous variable. Drukker *et al.* (2011b) also assume that the dummy variable `elect` (where elect is 1 if a county government faces an election, 0 otherwise) constitute a valid instrument for `police`.

Table 4 presents the results when this additional endogeneity is taken into account. Results from three different implementation are presented, namely the function `spreg` available from **sphet** under R, the function `spreg` available from Stata (setting the option to `het`), and the function `GM_Combo_Het` available from PySAL. A glance at Table 4 shows that all implementation gives the same results. As pointed out in LeSage and Pace (2009), the estimated vector of parameters in a spatial model does not have the same interpretation as in a simple

[18] This discussion is under the assumption that $\mathbf{W}$ and $\mathbf{M}$ are the same, otherwise $\mathbf{M}$ should be considered instead of $\mathbf{W}$.

[19] Unless otherwise stated, we tried to always run the default values of the various implementations.

[20] Kelejian and Prucha (1999) do not derive the joint asymptotic distribution of the parameters of the model but consider ρ_{Err} as a nuisance parameter. Instruction on how the inference on the spatial parameter is calculated in the SE Toolbox can be found on Ingmar Prucha's web page at: `http://econweb.umd.edu/~prucha/STATPROG/OLS/desols.pdf`

	R `spreg`	Stata `spivreg`	PySAL `GM_Endog_Combo_Hom`
(Intercept)	11.605968	11.605968	11.605968
	(1.666744)	(1.666744)	(1.666744)
nondui	-0.000196	-0.000196	-0.000196
	(0.002759)	(0.002759)	(0.002759)
vehicles	0.092996	0.092996	0.092996
	(0.005649)	(0.005649)	(0.005649)
dry	0.398260	0.398260	0.398260
	(0.090902)	(0.090902)	(0.090902)
police	-1.351308	-1.351308	-1.351308
	(0.141018)	(0.141018)	(0.141018)
ρ_{Lag}	0.193190	0.193190	0.193190
	(0.044310)	(0.044310)	(0.044310)
ρ_{Err}	-0.085975	-0.085975	-0.085975
	(0.030183)	(0.030183)	(0.030183)

Table 4: SARAR estimation when `police` is treated as endogenous. Results for three implementations, DUI data set (standard errors in parentheses)

linear regression model. The estimate of ρ_{Lag} is positive and significant thus indicating spatial autocorrelation in the dependent variable. Drukker *et al.* (2011b) try to find some possible explanation to this. On the one hand, the positive coefficient may be explained in terms of coordination effort among police departments in different counties. On the other hand, it might well be that an enforcement effort in one of the counties leads people that live close to the border to drink in neighboring counties. The estimate of ρ_{Err} is also significant but negative. As we will see later, those results are also in line with the evidence from the maximum likelihood estimation of the model.

One thing should be noted here. When we discussed the instrument matrix, we anticipated that when additional endogenous variables were present (as it is in this case), the optimal instruments are unknown. We also anticipated that, unless additional information was available, we did not recommended to include the spatial lag of these additional exogenous variables in the matrix of instrument. However, results reported in Table 4 do consider the spatial lags of `elect`. This is because, unlike R and PySAL, there is no option in Stata not to include them. For comparison purposes then, the default values of the corresponding arguments in R and PySAL were set to include those spatial lags.

Heteroskedasticity with and without additional endogenous variables

When the errors are assumed to be heteroskedastic of unknown form, the results are those presented in Tables 5 (when no additional endogeneity is assumed) and Table 6 (when `police` is treated as endogenous). From the observation of Table 5 it can be noticed that the implementations in R and PySAL are identical (up to the sixth decimal), and that Stata only presents very minor differences. Those differences relate to the value of the ρ_{Err} estimated coefficient (obtained trough the non-linear least square algorithm), and to the standard error of the intercept.

	R `spreg`	Stata `spreg`	PySAL `GM_Combo_Het`
(Intercept)	-6.410088	-6.410088	-6.410088
	(0.445961)	(0.445958)	(0.445961)
police	0.598088	0.598088	0.598088
	(0.018154)	(0.018154)	(0.018154)
nondui	0.000247	0.000247	0.000247
	(0.001097)	(0.001097)	(0.001097)
vehicles	0.015713	0.015713	0.015713
	(0.000784)	(0.000784)	(0.000784)
dry	0.106121	0.106121	0.106121
	(0.033807)	(0.033807)	(0.033807)
ρ_{Lag}	0.046944	0.046944	0.046944
	(0.017928)	(0.017928)	(0.017928)
ρ_{Err}	-0.000366	-0.000378	-0.000366
	(0.036803)	(0.036803)	(0.036803)

Table 5: SARAR estimation with heteroskedasticity. Results for three implementations, DUI data set (standard errors in parentheses).

When the endogeneity of the `police` variable is taken into account, the implementations in R and PySAL are again identical, and Stata presents the same minor differences (see Table 6).

W and M are different

As we anticipated, **W** and **M** need not to be the same in all applications. Results for this case are presented in this section and, since PySAL does not include this feature, are limited to the implementations of R and Stata. From an implementation perspective, the main complication in this case is the definition of the instrument matrix. While if **W** and **M** are identical the instrument need to be specified only in terms of **W**, when the two spatial weighting matrix are different the instruments should combine them as in Equation 13.

Table 7 summarizes the estimation results when `police` is treated as an endogenous variable and **W** and **M** are not the same.[21] Since the endogeneity of the `police` variable is accounted for, the default value to compute the lagged "additional" instruments (i.e., `lag.instr`) was changed in R. The results from the two implementations are almost identical except for a difference in the last decimal digit for the estimate of ρ_{Err}. Most likely, this is again due to small differences in the optimization routines adopted by the two environments. When the residuals are assumed to be heteroskedastic, similar results are observed, and, therefore, we decided not to report them in the paper.

3.2. Spatial lag model

The estimation of the spatial lag model in Equation 8 can be easily approached by two stage least squares. If the only restriction is $\rho_{\mathrm{Err}} = 0$ and there are exogenous variables other than the spatial lag in the model (i.e., if $\pi \neq 0$), additional instruments are then needed in order

[21] **M** is defined as a row standardised six nearest neighbours matrix, treating the county centroid coordinates as projected, not geographical.

	R `spreg`	Stata `spivreg`	PySAL `GM_Combo_Het`
(Intercept)	11.649298	11.649298	11.649298
	(1.873178)	(1.873179)	(1.873178)
nondui	-0.000155	-0.000155	-0.000155
	(0.002843)	(0.002843)	(0.002843)
vehicles	0.093058	0.093058	0.093058
	(0.005967)	(0.005967)	(0.005967)
dry	0.398707	0.398707	0.398707
	(0.094791)	(0.094791)	(0.094791)
police	-1.352871	-1.352871	-1.352871
	(0.149223)	(0.149223)	(0.149223)
ρ_{Lag}	0.192149	0.192149	0.192149
	(0.051833)	(0.051833)	(0.051833)
ρ_{Err}	-0.050266	-0.050263	-0.050266
	(0.039931)	(0.039931)	(0.039931)

Table 6: SARAR estimation with heteroskedasticity when `police` is treated as endogenous. Results for three implementations, DUI data set (standard errors in parentheses)

	R `spreg`	Stata `spivreg`
(Intercept)	9.210831	9.210831
	(1.454592)	(1.454592)
nondui	-0.000238	-0.000238
	(0.002480)	(0.002480)
vehicles	0.083249	0.083249
	(0.004799)	(0.004799)
dry	0.361584	0.361584
	(0.081570)	(0.081570)
police	-1.105622	-1.105622
	(0.119709)	(0.119709)
ρ_{Lag}	0.180395	0.180395
	(0.040716)	(0.040716)
ρ_{Err}	-0.011908	-0.011906
	(0.033255)	(0.033255)

Table 7: SARAR estimation when `police` is treated as endogenous and **W** and **M** are different. Results for two implementations, DUI data set (standard errors in parentheses).

to estimate the model.

When homoskedasticity is assumed, the variance covariance matrix can be estimated consistently by

$$\tilde{\sigma}^2(\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}})^{-1} \tag{23}$$

where $\tilde{\sigma}^2 = n^{-1}\sum_{i=1}^{n}\tilde{\mathbf{u}}_i^2$ and $\tilde{\mathbf{u}} = \mathbf{y} - \mathbf{Z}\tilde{\delta}$. On the other hand, when heteroskedasticity is assumed the estimation of the coefficients variance covariance matrix assumes the following sandwich form:

$$(\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}})^{-1}(\tilde{\mathbf{Z}}^\top\hat{\Sigma}\tilde{\mathbf{Z}})(\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}})^{-1} \tag{24}$$

where $\hat{\Sigma}$ is a diagonal matrix whose elements are the squared residuals.

Homoskedasticity with and without additional endogenous variables

From Table 8 we can see that there are five implementations available of the spatial lag model when considering homoskedastic innovations and no additional endogenous variable. The first two columns of the table report results obtained from R. There are multiple functions that allow the estimation of the spatial lag model available from R under the **spdep** and **sphet** packages. `stsls` was the first function made available as part of the package **spdep**. With the development of **sphet**, the wrapper function `spreg` also allows the estimation of the spatial lag model. The other three columns of Table 8 report the results obtained from Stata, PySAL, and the SE Toolbox, respectively.

	R `stsls`	R `spreg`	Stata `spreg gs2sls`	PySAL `GM_Lag`	SE `sar_gmm`
(Intercept)	-6.410152	-6.410152	-6.410152	-6.410152	-6.410152
	(0.418129)	(0.418129)	(0.417725)	(0.417725)	(0.418129)
police	0.598081	0.598081	0.598081	0.598081	0.598081
	(0.014918)	(0.014918)	(0.014904)	(0.014904)	(0.014918)
nondui	0.000247	0.000247	0.000247	0.000247	0.000247
	(0.001087)	(0.001087)	(0.001086)	(0.001086)	(0.001087)
vehicles	0.015714	0.015714	0.015714	0.015714	0.015714
	(0.000669)	(0.000669)	(0.000668)	(0.000668)	(0.000669)
dry	0.106134	0.106134	0.106134	0.106134	0.106134
	(0.034962)	(0.034962)	(0.034928)	(0.034928)	(0.034962)
ρ_{Lag}	0.046950	0.046950	0.046950	0.046950	0.046950
	(0.016977)	(0.016977)	(0.016960)	(0.016960)	(0.016977)

Table 8: Lag model estimation results for five implementations. DUI data set (standard errors in parentheses).

Given that we are considering the same matrix of instruments, the coefficient values of all implementations agree exactly. This had to be expected given that the OLS also matched. There are difference though in the reported standard errors. In the two (R and SE toolbox) functions, the error variance is calculated with a degrees of freedom correction (i.e., dividing by $n - k$), while in the other two implementations is simply divided by n.

Table 9 shows the results for the lag model when `police` is treated as endogenous. While all the coefficients are matching, difference in the standard error are produced by the same degrees of freedom correction.

	R `spreg`	Stata `spivreg`	PySAL `GM_Lag`
(Intercept)	11.507606	11.507606	11.507606
	(1.686222)	(1.684594)	(1.686222)
nondui	-0.000293	-0.000293	-0.000293
	(0.002771)	(0.002768)	(0.002771)
vehicles	0.092866	0.092866	0.092866
	(0.005663)	(0.005657)	(0.005663)
dry	0.397357	0.397357	0.397357
	(0.091419)	(0.091331)	(0.091419)
police	-1.348024	-1.348024	-1.348024
	(0.141410)	(0.141273)	(0.141410)
ρ_{Lag}	0.195595	0.195595	0.195595
	(0.045906)	(0.045862)	(0.045906)

Table 9: Lag model estimation when `police` is treated as endogenous. Results for three implementations. DUI data set (standard errors in parentheses).

Heteroskedasticity with and without additional endogenous variables

Apart from MATLAB SE Toolbox, all other implementations (including the two R functions `stsls` and `spreg`) allow the estimation of the lag model under heteroskedastic innovations. Of course, the estimated coefficients are not different from the homoskedastic case, and the only variation is in the standard errors. However, the standard errors under heteroskedasticity are equal across the four models, and, therefore, we are not reporting them in the paper.[22]

3.3. Spatial error model

In the spatial error model (i.e., $\rho_{\text{Lag}} = 0$), we can still use most of the formulas above. The first step of the estimation procedure is either OLS (when $\pi = 0$), or IV, when $\pi \neq 0$ and there are endogenous variable (other than the spatial lag) in the model. After estimating ρ_{Err} in the GMM step, we can then take the spatial Cochrane-Orcutt transformation. The resulting model can be then estimated by two stage least squares using the matrix of instruments $\mathbf{H}$, where $\mathbf{H}$ is made up of, at least, the linearly independent columns of $\mathbf{X}$, and $\mathbf{MX}$.

Homoskedasticity with and without additional endogenous variables

[22] These and other results are available from the authors upon request.

	R GMerrorsar	R spreg	Stata spreg gs2sls	PySAL GM_Error	PySAL GM_Error_Hom	SE sem_gmm
(Intercept)	-5.431921	-5.431959	-5.431959	-5.431921	-5.431960	-5.431921
	(0.229056)	(0.229067)	(0.229067)	(0.229052)	(0.229050)	(0.229056)
police	0.599854	0.599851	0.599851	0.599854	0.599851	0.599854
	(0.014888)	(0.014890)	(0.014890)	(0.014888)	(0.014887)	(0.014888)
nondui	0.000257	0.000257	0.000257	0.000257	0.000257	0.000257
	(0.001086)	(0.001086)	(0.001086)	(0.001086)	(0.001086)	(0.001086)
vehicles	0.015612	0.015612	0.015612	0.015612	0.015612	0.015612
	(0.000667)	(0.000667)	(0.000667)	(0.000667)	(0.000667)	(0.000667)
dry	0.103654	0.103663	0.103663	0.103654	0.103663	0.103654
	(0.034966)	(0.034967)	(0.034967)	(0.034966)	(0.034965)	(0.034966)
ρ_{Err}	0.050883	0.047050	0.047050	0.050883	0.051491	0.050883
	(0.080487)	(0.029543)	(0.029543)	(0.000000)	(0.028809)	(0.080487)

Table 10: Error model estimation results for six implementations. DUI data set (standard errors in parentheses).

Table 10 show the spatial error model estimation results for six different implementations. These implementations include two R functions (GMerrorsar available from **spdep**, and spreg available from **sphet**), one Stata function (spreg gs2sls), two different implementation based on PySAL (GM_Error and GM_Error_Hom), and, finally, the SE Toolbox for MATLAB (sem_gmm).

Three of them are based on the Kelejian and Prucha (1999) moment conditions (columns one, four, and six), the others are based on Drukker *et al.* (2013) moment conditions (columns two, three, and five). Let us focus first on the results obtained using Kelejian and Prucha (1999) moment conditions. All three implementations produce the same identical estimated coefficients. There are differences in terms of the standard errors. We refer to the fact that, while GMerrorsar and sem_gmm produce an estimate for the standard error of the spatial coefficient, the GM_Error function in PySAL does not. The reason is that in their original contribution Kelejian and Prucha (1999) do not derive the joint asymptotic distribution of the parameters of the model and, therefore, GM_Error do not produce any inference on ρ_{Err}.[23]

	R spreg	Stata spivreg	PySAL GM_Endog_Error_Hom
(Intercept)	15.484115	15.484115	17.326391
	(1.578958)	(1.578959)	(1.748870)
nondui	-0.000208	-0.000208	-0.000226
	(0.002755)	(0.002755)	(0.002962)
vehicles	0.092430	0.092430	0.099270
	(0.005655)	(0.005655)	(0.006272)
dry	0.395797	0.395797	0.421893
	(0.090901)	(0.090901)	(0.097922)
police	-1.337080	-1.337080	-1.508904
	(0.141153)	(0.141153)	(0.156655)
ρ_{Err}	-0.004487	-0.004483	-0.005472
	(0.025467)	(0.025467)	(0.025496)

Table 11: Error model estimation when police is treated as endogenous. Results for three implementations. DUI data set (standard errors in parentheses).

Moving now to the three implementations that are based on Drukker *et al.* (2013) moment conditions, we observe that, while Stata and spreg (available from R) present exactly the same results, some distinctions are observed in PySAL. The reason for these differences were anticipated before. In a spatial error model, the second term in Equation 22 limits to zero (when there are only exogenous variables in the model). The implementations in R and Stata produce an estimate of this term, whilePySAL does not. However, as it can be appreciated from the figures in Table 10, the differences (with this specific dataset) are very minor.

Table 11 displays the results produced by three implementations when police is instrumented. The only three softwares in which this feature is available are R, Stata, and PySAL. A glance at the table reveals that the results across implementations are very different. The differences between R and Stata are very minor and they can be attributable to differences in optimization routines. The differences with PySAL seem to be found in the different specification of the

[23] Instructions on how to produce inference are again available from Ingmar Prucha's website at http://econweb.umd.edu/~prucha/STATPROG/OLS/desols.pdf.

instrument matrix. Most likely, PySAL is missing the spatial lag of the exogenous variable (i.e., **MX**), and in only including the additional instrument (`elect`) and its spatial lag.

Heteroskedasticity with and without additional endogenous variables

The same evidence is confirmed also when heteroskedasticity is assumed and, therefore, results are not reported in the paper but are available from the authors. We will take the time for one final remark. In our theoretical review of the general spatial model we anticipated the possibility of obtaining a consistent and efficient estimator of ρ_{Err} based on 2SLS rather than GS2SLS.

Table 12 reports the results when considering this additional step (i.e., step1.c). Stata does not present this feature. Results from R and PySAL are very similar and again difference must be due to the estimation of ($\tilde{\alpha}$).

	R spreg	PySAL `GM_Endog_Error_Hom`
(Intercept)	-5.432185	-5.431451
	(0.255239)	(0.255217)
police	0.599834	0.599890
	(0.018086)	(0.018084)
nondui	0.000257	0.000256
	(0.001096)	(0.001096)
vehicles	0.015614	0.015609
	(0.000781)	(0.000780)
dry	0.103715	0.103545
	(0.033712)	(0.033710)
ρ_{Err}	0.051595	0.056201
	(0.030924)	(0.029984)

Table 12: Error estimation results when `police` is treated as endogenous. Results for two implementations (step 1.c). DUI data set (standard errors in parentheses).

HAC estimation in a spatial framework

Lastly, we are going to review a slightly different form of the model based on the assumptions that the error term follows

$$u = \mathbf{R}\varepsilon \tag{25}$$

where **R** is an $n \times n$ unknown non-stochastic matrix, and ε is a vector of innovations. The asymptotic distribution of the corresponding IV estimators involves the VC matrix:

$$\mathbf{\Psi} = n^{-1}\mathbf{H}^{\top}\mathbf{\Omega}\mathbf{H} \tag{26}$$

where $\mathbf{\Omega} = \mathbf{R}\mathbf{R}^{\top}$ denotes the VC matrix of ε. Kelejian and Prucha (2007) suggest estimating the individual r, s elements of $\mathbf{\Psi}$ as

$$\tilde{\psi}_{rs} = n^{-1}\sum_{i=1}^{n}\sum_{j=1}^{n} h_{ir}h_{js}\hat{\varepsilon}_i\hat{\varepsilon}_j K(d^*_{ij}/d) \tag{27}$$

where the subscripts refer to the elements of the matrix of instruments $\mathbf{H}$ and the vector of estimated residuals $\hat{\varepsilon}$. The Kernel function $K()$ is defined in terms of the distance measure d^*_{ij}, the distance between observations i and j. The bandwidth d is such that if $d^*_{ij} \geq d$, the associated Kernel is set to zero $(K(d^*_{ij}/d) = 0)$. In other words, the bandwidth together with the Kernel function limits the number of sample covariances.

Based on Equation 27, the asymptotic variance covariance matrix $(\tilde{\mathbf{\Phi}})$ of the S2SLS estimator of the parameters vector is given by:

$$\tilde{\mathbf{\Phi}} = n^2(\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}})^{-1}\mathbf{Z}^\top\mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-1}\tilde{\mathbf{\Psi}}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{Z}(\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}})^{-1} \tag{28}$$

Piras (2010) mentioned that the implementation was compared with a code used by Anselin and Lozano-Gracia (2008) and that the two codes gave the same results. However, at the time Piras (2010) was published, no other public implementation was available. Here we compare standard error estimates using a Triangular kernel with a variable bandwidth of the six nearest neighbours.[24]

	R spreg	PySAL GM_Lag
(Intercept)	11.850234	11.507606
	(1.874336)	(1.842620)
nondui	-0.000293	-0.000293
	(0.002827)	(0.002805)
vehicles	0.093571	0.092866
	(0.006010)	(0.005980)
dry	0.400032	0.397357
	(0.096270)	(0.095334)
police	-1.365765	-1.348024
	(0.150182)	(0.149460)
ρ_{Lag}	0.188345	0.195595
	(0.056502)	(0.054681)

Table 13: HAC estimation when `police` is treated as endogenous. Results for two implementations. DUI data set (standard errors in parentheses).

When there are no additional endogenous variables other than the spatial lag, the results from the two implementation match well and results are omitted. Some interesting differences are observed when `police` is treated as endogenous. In this case while the default in R is not to take the lags of `elect`; PySAL will include these lags in the matrix of instruments. This is reflected in the results shown in Table 13

4. Comparing maximum likelihood estimation

[24] After reading the coordinates for each location, we used the function `knearneigh` to generate the distances and, after some transformation between objects, the final kernel using the R function `read.gwt2dist`.In PySAL, the function `pysal.kernelW_from_shapefile` was used to directly read the shape file in and generate the kernel. In both cases, the centroids of the counties were used as point locations, but the distances for the nearest neighbour procedure were calculated as though the coordinates were projected, while they are in fact geographical. There are many available options for the kernel both in R and PySAL.

Having compared implementations of GMM estimators, we move to cross-section maximum likelihood (ML) estimation, recalling that ML estimation for spatial panel models was compared for MATLAB and R implementations in Millo and Piras (2012). Since Python PySAL has no ML implementations, it will not be considered. None of the ML implementations make provision for instrumenting endogenous right hand side variables, nor for accommodating heteroskedasticity — this is provided for in Bayesian implementations in the Spatial Econometrics toolbox.

In Section 1.1 (p. 4), we described the numerical optimisers used in the various applications. In many cases, the numerical optimisation functions can return numerical Hessians for use as estimators of the covariance matrix of model coefficients, which may be used instead of analytical, asymptotic covariance matrices. In other cases, the numerical Hessian may be found by examining the form of the function being optimised around the optimum, for example using finite-difference Hessian algorithms. In implementations in the MATLAB Spatial Econometrics Toolbox, use is made of `fdhess` taken from the CompEcon Toolbox by Paul Fackler, but with the relative step size hard-coded to $1.0 \cdot 10^{-8}$ in `sar`, `sdm` and `sac`, but to $1.0 \cdot 10^{-5}$ in `sem` as in the original CompEcon function. In R, use is made of `fdHess` from the **nlme** package with a default relative step size of $6.055 \cdot 10^{-6}$ used without modification. In these comparisons in R, we will usually use analytical, asymptotic covariance matrices, but numerical Hessians are used sometimes for comparison.

An extensive treatment of maximum likelihood estimation for spatial models is given by LeSage and Pace (2009, pp. 45–75). Further discussion of the general model, and issues arising in fitting two parameters when the surface of the function shows little structure may be found in Bivand (2012). Here we compare implementations in MATLAB, Stata and R, looking first at spatial lag models, from which some general conclusions are drawn about the implementations, before going on to the spatial error model and the general two-parameter spatial model.

4.1. Spatial lag model

The log-likelihood function for the spatial lag model is:

$$\begin{aligned}\ell(\beta, \rho_{\mathrm{Lag}}, \sigma^2) = -\frac{n}{2}\ln 2\pi - \frac{n}{2}\ln\sigma^2 + \ln|\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W}| \\ -\frac{1}{2\sigma^2}[((\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W})\mathbf{y} - \mathbf{X}\beta)^\top((\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W})\mathbf{y} - \mathbf{X}\beta)]\end{aligned}$$

and by extension the same framework is used for the spatial Durbin model when $[\mathbf{W}(\mathbf{WX})]$ are grouped together. Since β can be expressed as $(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W})\mathbf{y}$, all of the cross-product terms can be pre-computed as cross-products of the residuals of two ancilliary regressions: $\mathbf{y} = \mathbf{X}\beta_1 + \varepsilon_1$ and $\mathbf{Wy} = \mathbf{X}\beta_2 + \varepsilon_2$, and the sum of squares term can be calculated much faster than the log-determinant (Jacobian) term of the $n \times n$ sparse matrix $\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W}$; see LeSage and Pace (2009) for details. The legacy method for computing the log-determinant term is to use eigenvalues of $\mathbf{W}$:

$$\ln(|\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W}|) = \sum_{i=1}^{n}\ln(1 - \rho_{\mathrm{Lag}}\zeta_i) \tag{29}$$

using ρ_{Lag} to represent either parameter, and where ζ_i are the eigenvalues of $\mathbf{W}$ (Ord 1975, p. 121); other methods are reviewed in Bivand, Hauke, and Kossowski (2013).

One discrepancy that we can account for before presenting any further results is that the log-likelihood values at the optimimum differ between two implementations: 1551.08 in R **McSpatial** `sarml` and a similar value in the SE toolbox `sar` function (there is a small difference because the SE toolbox optimiser differs somewhat, and is discussed below), and -2628.58 in the other two: R **spdep** `lagsarlm` and Stata `spreg ml`.

The reason appears to be that π in the log likelihood calculation is not multiplied by 2 in the first two cases, but is in the second two. If we convert the R **McSpatial** value of by subtracting $\frac{n}{2}log(\pi)$ (line 65 in file McSpatial/R/sarml.R), and adding $\frac{n}{2}log(2\pi)$, we get -2628.58. Similarly, correcting the SE toolbox value (line 453 file spatial/sar_models/sar.m), we get a value close to that of Stata and R **spdep**. The same kind of difference appears in other reported SE toolbox log likelihood values.

	R `lagsarlm`	R `sarml`	Stata `spreg ml`	SE `sar`
(Intercept)	-6.337479	-6.337699	-6.337479	-6.349369
	(0.382022)	(0.380978)	(0.380987)	(0.088679)
police	0.598157	0.598157	0.598157	0.598145
	(0.014908)	(0.014903)	(0.014903)	(0.016146)
nondui	0.000249	0.000249	0.000249	0.000249
	(0.001086)	(0.001086)	(0.001086)	(0.001083)
vehicles	0.015711	0.015711	0.015711	0.015712
	(0.000668)	(0.000668)	(0.000668)	(0.000524)
dry	0.106131	0.106131	0.106131	0.106131
	(0.034931)	(0.034929)	(0.034929)	(0.034754)
ρ_{Lag}	0.043423	0.043430	0.043423	0.044000
	(0.014922)	(0.014782)	(0.014782)	(0.000625)

Table 14: Maximum likelihood spatial lag model estimation results for four implementations, DUI data set (standard errors in parentheses).

From Table 14, we see that the coefficient estimates of the R `lagsarlm` and Stata `spreg ml` implementations agree exactly for the chosen number of digits shown. The R `sarml` implementation differs slightly in coefficient estimates for the intercept and for ρ_{Lag}, but uses a different numerical optimiser. All these three optimise the same objective function, and reach the same optimum given the stopping value used by the optimiser. These three were also using eigenvalues to compute the log-determinant values; Stata `spreg ml` and R `sarml` only use eigenvalues, while R `lagsarlm` offers a selection of methods (Bivand *et al.* 2013), but here used eigenvalues. We will return below to differences in standard errors after explaining why the SE toolbox `sar` function yields different coefficient estimates.

The SE toolbox uses a pre-computed grid of log determinant values, choosing the nearest value of the log determinant from the grid rather than computing exactly for the current proposed value of ρ_{Lag} at each call to the log likelihood function. In order to investigate the consequences of this design choice, which has its motivation in picking from a grid of ρ_{Lag} and log-determinant values during Bayesian estimation, additions were edited into the `sar` function to return internal values so that they could be examined, to capture values from the

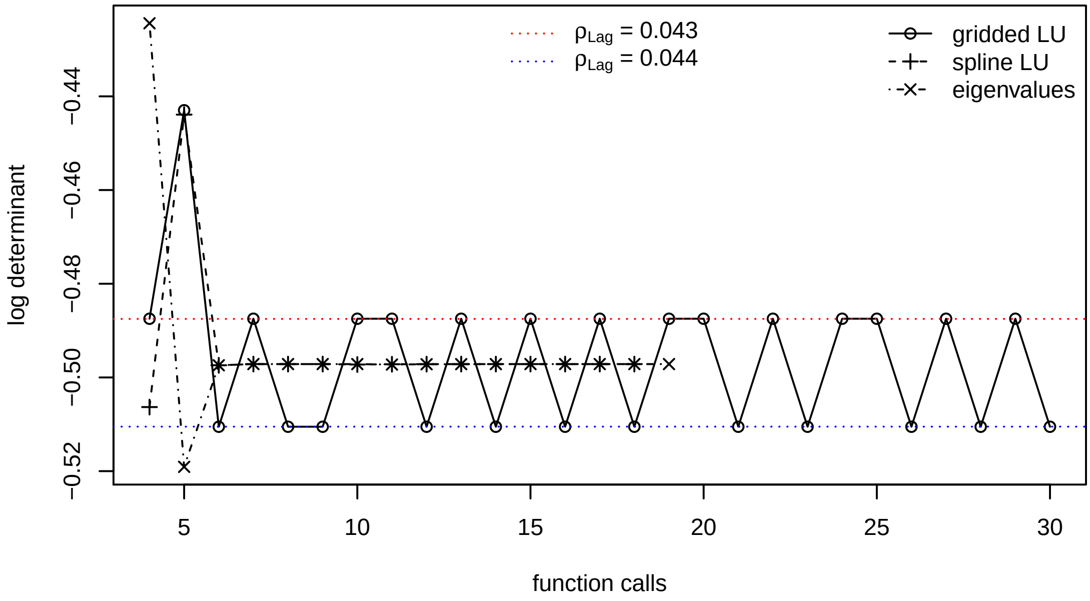

Figure 1: Spatial Econometrics Toolbox log determinant artefacts during line search; comparison of the gridded method with two alternatives.

optimiser while running, and to add three extra methods for computing the log-determinant.

The method used by default for `info.lflag = 0` in `sar_lndet` is to call `lndetfull`, which creates a coarse grid of log-determinant values for $\ln |\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W}|$ for ρ_{Lag} at 0.01 steps over a given range using the sparse matrix LU method. These log-determinant values are then spline-interpolated to a finer grid at 0.001 steps of ρ_{Lag} in `sar_lndet`.

In `info.lflag = 3`, the spline-interpolation was changed to return a fitted spline object, which could then be used in the objective function `f_sar` to interpolate using `ppval` for the current proposed value of ρ_{Lag}, so returning a much closer log-determinant value than that available from the look-up table. Finally, `info.lflag = 5` was added to provide the standard eigenvalue method, calculating the log-determinant for the current proposed value of ρ_{Lag}.

Figure 1 shows the behaviour of the optimiser for `info.lflag` taking values of 0 — the gridded LU log-determinant values (gridded LU in Table 15), 3 — values interpolated from a spline object fitted to a coarse grid of LU log-determinant values (spline LU in Table 15), and 5 — values calculated using the eigenvalues of $\mathbf{W}$ (eigen/asy in Table 15). In the `info.lflag = 0` case, behaviour is further inhibited by rounding effects in choosing values from the look up table, and by the inconsistency caused by the sum-of-squared error term in the objective function being calculated using the current proposed value of ρ_{Lag}, while the log-determinant value is for the selected grid slot. In this default scenario, the value of the log determinant oscillates between two values of ρ_{Lag} spanning the optimum, and the line search terminates after a series of jumps with tighter tolerance passed as `info.convg` as $1.0 \cdot 10^{-8}$; its default tolerance set in `sar` is $1.0 \cdot 10^{-4}$, but using a tighter tolerance did not improve performance. When permitted to compute the log-determinant for the current proposed value of ρ_{Lag} in the two alternative cases, the Brent line search function used in `sar` converges quickly.

	R lagsarlm	SE gridded LU	SE spline LU	SE eigen/asy
(Intercept)	-6.337479	-6.349369	-6.337479	-6.337479
	(0.382022)	(0.088679)	(0.373889)	(0.382022)
police	0.598157	0.598145	0.598157	0.598157
	(0.014908)	(0.016146)	(0.012629)	(0.014908)
nondui	0.000249	0.000249	0.000249	0.000249
	(0.001086)	(0.001083)	(0.001084)	(0.001086)
vehicles	0.015711	0.015712	0.015711	0.015711
	(0.000668)	(0.000524)	(0.000348)	(0.000668)
dry	0.106131	0.106131	0.106131	0.106131
	(0.034931)	(0.034754)	(0.034793)	(0.034931)
ρ_{Lag}	0.043423	0.044000	0.043423	0.043423
	(0.014922)	(0.000625)	(0.013052)	(0.014922)

Table 15: Maximum likelihood spatial lag model estimation results for alternative Spatial Econometric Toolbox log-determinant implementations, DUI data set (standard errors in parentheses).

Table 15 shows that when the MATLAB SE Toolbox `sar` function uses eigenvalues to calculate the log-determinant, and is forced to return analytical, asymptotic coefficient standard errors (right column), the output agrees exactly with that of R `lagsarlm` using eigenvalues and returning analytical, asymptotic coefficient standard errors (first column). The coefficient estimates for the interpolated LU log-determinants in column three also agree, as the line search has estimated ρ_{Lag} at the same value. Here, however, the standard errors are calculated using the numerical Hessian estimated at the optimum; `sar` by default switches between asymptotic and numerical Hessian standard errors at $n > 500$, but the criterion was changed to 4000 in the final column. The second column contains the results shown above in Table 14, final column. We have thus accounted for the differences in results of coefficient estimation between the MATLAB SE Toolbox `sar` function and the other implementations, and proceed to differences in standard errors.

	R `sarlm`	Stata NR	R `lagsarlm`
(Intercept)	0.380978	0.380987	0.382022
police	0.014903	0.014903	0.014908
nondui	0.001086	0.001086	0.001086
vehicles	0.000668	0.000668	0.000668
dry	0.034929	0.034929	0.034931
ρ_{Lag}	0.014782	0.014782	0.014922

Table 16: Maximum likelihood spatial lag model standard error estimation results for alternative implementations, DUI data set.

The standard errors reported by R `sarlm` are taken from the Hessian returned by the optimization function `nlm`. Stata `spreg ml` by default uses a modified Newton-Raphson method `nr`, reporting standard errors taken from the Hessian returned by the optimization function, rather than the analytical calculations even for small n. Table 16 compares the analytical, asymptotic standard errors in the third column with those returned using numerical Hessian values. The differences that we observe can be explained through these two different approaches, either analytical standard errors calculated using asymptotic formulae, or standard errors calculated from the numerical Hessian.

4.2. Other ML estimators

The log-likelihood function for the spatial error model is:

$$\ell(\beta, \rho_{\mathrm{Err}}, \sigma^2) = -\frac{n}{2}\ln 2\pi - \frac{n}{2}\ln\sigma^2 + \ln|\mathbf{I} - \rho_{\mathrm{Err}}\mathbf{W}|$$
$$-\frac{1}{2\sigma^2}[(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{I} - \rho_{\mathrm{Err}}\mathbf{W})^\top(\mathbf{I} - \rho_{\mathrm{Err}}\mathbf{W})(\mathbf{y} - \mathbf{X}\beta)]$$

As we can see, the problem is one of balancing the log determinant term $\ln(|\mathbf{I} - \rho_{\mathrm{Err}}\mathbf{W}|)$ against the sum of squares term. When ρ_{Err} approaches the ends of its feasible range, the log determinant term may swamp the sum of squares term.

β may be concentrated out of the sum of squared errors term, for example as:

$$\ell(\rho_{\mathrm{Err}}, \sigma^2) = -\frac{N}{2}\ln 2\pi - \frac{N}{2}\ln\sigma^2 + \ln|\mathbf{I} - \rho_{\mathrm{Err}}\mathbf{W}|$$
$$-\frac{1}{2\sigma^2}[\mathbf{y}^\top(\mathbf{I} - \rho_{\mathrm{Err}}\mathbf{W})^\top(\mathbf{I} - \mathbf{Q}_{\rho_{\mathrm{Err}}}\mathbf{Q}_{\rho_{\mathrm{Err}}}^\top)(\mathbf{I} - \rho_{\mathrm{Err}}\mathbf{W})\mathbf{y}]$$

where $\mathbf{Q}_{\rho_{\mathrm{Err}}}$ is obtained by decomposing $(\mathbf{X} - \rho_{\mathrm{Err}}\mathbf{W}\mathbf{X}) = \mathbf{Q}_{\rho_{\mathrm{Err}}}\mathbf{R}_{\rho_{\mathrm{Err}}}$.

The relationship between the log-determinant term and the sum of squares term in the log likelihood function in the spatial error model is analogous to that in the spatial lag model, but the sum of squares term involves more computation in the case of the spatial error model. In all cases, a simple line search may be used to find ρ_{Lag} or ρ_{Err}, and other coefficients may be calculated using an ancilliary regression once this has been done.

The general model is more demanding, and requires that ρ_{Lag} and ρ_{Err} be found by constrained numerical optimization in two dimensions by searching for the maximum on the surface of the log-likelihood function, which is like that of the spatial error model with additional terms in $\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W}$, here assuming that the same spatial weights are used in both processes:

$$\ell(\rho_{\mathrm{Lag}}, \rho_{\mathrm{Err}}, \sigma^2) = -\frac{N}{2}\ln 2\pi - \frac{N}{2}\ln\sigma^2 + \ln|\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W}| + \ln|\mathbf{I} - \rho_{\mathrm{Err}}\mathbf{W}|$$
$$-\frac{1}{2\sigma^2}[\mathbf{y}^\top(\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W})^\top(\mathbf{I} - \rho_{\mathrm{Err}}\mathbf{W})^\top(\mathbf{I} - \mathbf{Q}_{\rho_{\mathrm{Err}}}\mathbf{Q}_{\rho_{\mathrm{Err}}}^\top)(\mathbf{I} - \rho_{\mathrm{Err}}\mathbf{W})(\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W})\mathbf{y}]$$

The tuning of the constrained numerical optimization function, including the provision of starting values, reasonable stopping criteria, and also the choice of algorithm may all affect the results achieved. The Stata implementation uses a grid search for initial values of $(\rho_{\mathrm{Lag}}, \rho_{\mathrm{Err}})$ (Drukker *et al.* 2011c), the Spatial Econometrics toolbox uses the generalized spatial two-stage least squares estimates, with the option of a user providing initial values, and the **spdep** implementation for row-standardised spatial weights matrices uses either four candidate pairs of initial values at $0.8(L, U)$, $(0, 0)$, $0.8(U, U)$, and $0.8(U, L)$, where L anf U are two-element vectors of bounds on $(\rho_{\text{Lag}}, \rho_{\text{Err}})$, a full grid of nine points at the same settings, or user provided initial values; optimizers may be chosen by the user.

	R `errorsarlm`	Stata `spreg ml`	R `sacsarlm`	Stata `spreg ml`
(Intercept)	-5.432939	-5.432938	-6.356649	-6.356651
	(0.229072)	(0.229284)	(0.419559)	(0.421408)
police	0.599777	0.599777	0.598036	0.598036
	(0.014891)	(0.014891)	(0.014912)	(0.014923)
nondui	0.000258	0.000258	0.000250	0.000250
	(0.001087)	(0.001087)	(0.001086)	(0.001086)
vehicles	0.015619	0.015619	0.015717	0.015717
	(0.000667)	(0.000668)	(0.000669)	(0.000669)
dry	0.103890	0.103890	0.106312	0.106313
	(0.034967)	(0.034993)	(0.034930)	(0.034967)
ρ_{Lag}			0.044393	0.044393
			(0.017190)	(0.017311)
ρ_{Err}	0.045856	0.045857	-0.003813	-0.003815
	(0.029878)	(0.030069)	(0.035057)	(0.035846)

Table 17: Maximum likelihood spatial error and general spatial model estimation results for two implementations, DUI data set (standard errors in parentheses).

As we can see from Table 17, the computed coefficients agree adequately for the spatial error model implementations for R and Stata. There are minor differences in the standard errors between R and Stata, because of the use of the numerical Hessian to calculate the standard errors in Stata. The SE toolbox estimates differ somewhat because of the use of gridded log determinant values explained above for the spatial lag model case, and are not presented here.

The R **spdep** function `sacsarlm` can be used to estimate the general spatial model; results for `sacsarlm` and Stata `spreg ml` are given the two right-hand columns in Table 17. Again, the coefficient estimates are in good agreement even given the flatness of the objective function seen in Figure 2, and the closeness of ρ_{Err} to zero (optimum of -2628.6 at the values of $(\rho_{\text{Lag}}, \rho_{\text{Err}})$ given in Table 17 third column). This estimation success is assisted by the choice of starting values for numerical optimisation, without which outcomes may vary.

5. Implementing impact measures

In addition to the fitting of spatial econometric models, associated measures are needed to assist in their interpretation, in particular the impact of changes in right hand side variables in models including the spatially lagged dependent variable. In Section 1 above, we reviewed the need to calculate these emanating effects, which we will term impacts, though the cumbersome $S_r(\mathbf{W})$ matrix, where r is the index of the right hand side variable. This matrix term may be approximated using traces of powers of the spatial weights matrix as well as analytically. The average direct impacts are represented by the sum of the diagonal elements of the matrix divided by n for each exogenous variable, the average total impacts are the sum of all matrix elements divided by n for each exogenous variable, while the average indirect impacts are the

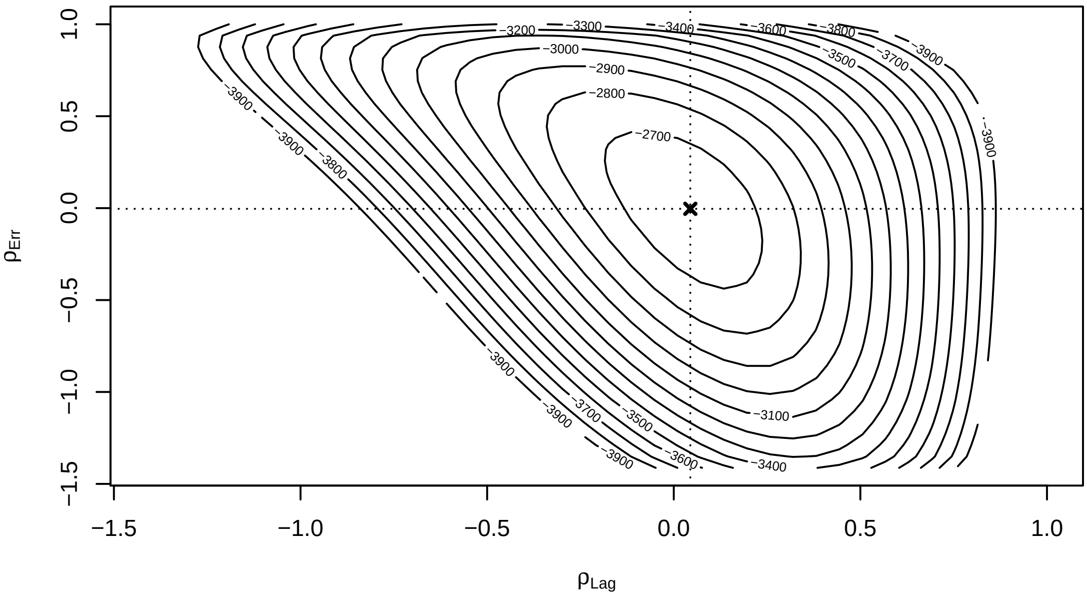

Figure 2: Contour plot values of the general spatial model log likelihood function for a 40 × 40 grid of values of $(\rho_{\text{Lag}}, \rho_{\text{Err}})$ showing values > -4000, DUI data set, calculated with R **spdep** function `sacsarlm`.

differences between these two vectors of impacts.

In Stata, the average total impacts are available by predicting from the estimated model using the original data, assigning the result to a new variable. Choosing variable r, x_r is incremented by one, and a new prediction made, once again assigning the result to a new variable. The mean of the difference between the two predictions is then the required measure (Drukker *et al.* 2011c, pp. 10–15). For the spatial lag model estimated by maximum likelihood, and the `police` variable, the value is 0.625310; one may calculate average total impacts for all models including the spatially lagged dependent variable in Stata irrespective of estimation method.

	R direct	SE direct	R total	SE total	β
police	0.598350	0.598349	0.625310	0.625310	0.598157
nondui	0.000249	0.000249	0.000261	0.000261	0.000249
vehicles	0.015717	0.015717	0.016425	0.016425	0.015711
dry	0.106165	0.106165	0.110948	0.110948	0.106131

Table 18: Direct and total impacts calculated using $\mathbf{W}$ traces for spatial lag models estimated using maximum likelihood in R, `lagsarlm` and MATLAB Spatial Econometrics toolbox (SE), `sar` with local modifications (eigenvalue log determinants); fitted β coefficient values shown for comparison, DUI data set.

In **spdep**, `impacts` methods are available for ML and GM spatial lag and general spatial model objects. The methods can use either dense matrices or truncated series of traces, so the impacts for a single model fit may be examined using dense or sparse procedures, and using different methods for computing the traces. The same methods are available for estimation functions in the **sphet** package, including the `spreg` function. Similarly, the MATLAB Spatial Econometrics toolbox model estimation functions report impacts, in their original form as the mean values of simulations to be presented below. Here the calculated impact values for the fitted values of the β coefficients are returned in addition. Table 18 shows the correspondence of the output values for the four variables in the DUI model, together with the fitted β coefficient values; in all cases, the total impacts are somewhat larger than these coefficients.

Estimated spatial models provide ways of inferring about the significance of the right hand side variables. When the spatially lagged dependent variable is present, the fitted β coefficient values and their standard errors do not provide a satisfactory basis for inference if $\rho_{\mathrm{Lag}} \neq 0$. This problem may be resolved by drawing samples from the estimated model, using a multivariate Normal distribution centred on the fitted values of $[\rho_{\mathrm{Lag}}, \beta]$, their covariance matrix, and then calculating the impacts of the sample coefficients.

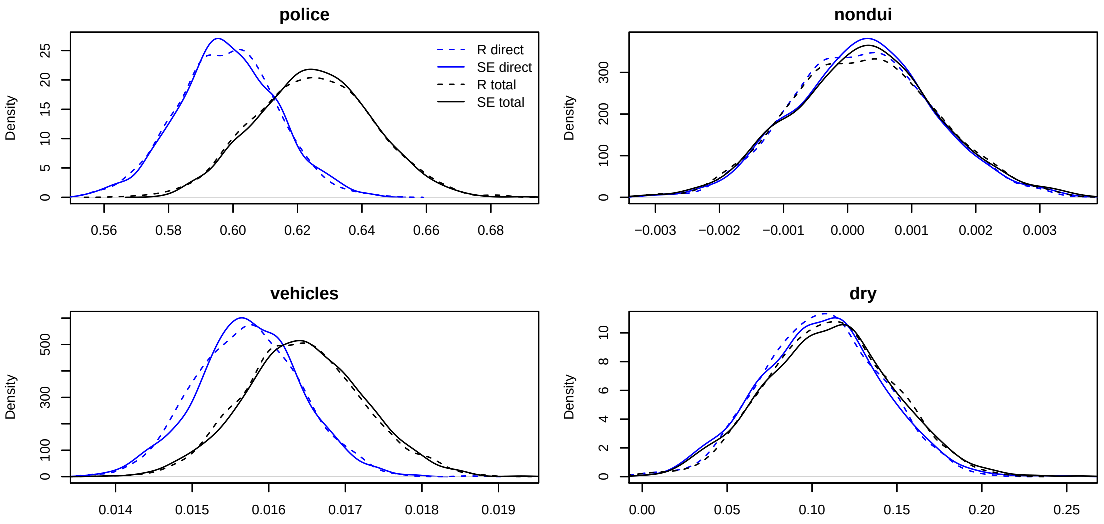

Figure 3: Monte Carlo tests for impacts from spatial lag models (direct shown in blue, total in black), implementations in Spatial Econometrics toolbox and R **spdep**, DUI data set.

As mentioned, MATLAB Spatial Econometrics toolbox model estimation functions perform by default simulation, and report the impacts as the means of the simulated coefficient impact distributions. The simulations themselves may ablo be returned, and their distribution by variable analysed to permit inference. R `impacts` methods for fitted model objects also permit the performance of simulations, returning the simulated impacts as `mcmc` objects defined in the **coda** package (they are Monte Carlo simulations, but because the Spatial Econometrics toolbox originally used MCMC simulations, the initial implementation had followed that lead, and used an `mcmc` object for output).

Figure 3 shows that the distributions of direct and total impacts in both MATLAB and R for variables `nondui` and `dry` are very similar, and that zero falls in the middle of the simulated distributions for `nondui` (the same conclusion as that seen in Table 14). For `police` and `vehicles`, the two implementations give very similar distributions, but those of direct impacts overlap the total impacts distributions only in the lower half. The simulation results permit us to infer from estimated spatial models including the spatially lagged dependent variable.

6. Concluding remarks

In conclusion, we can report that although there are some differences between results yielded when using available software implementations of spatial econometrics estimation methods on the same data set, it has been possible to establish why these differences arise. Some differences relate to differing interpretations of the underlying literature, others to choices of techniques used in implementations. Most of the methods proposed in the literature and considered here can be used in most of the applications, and in most cases will give the same or very similar results.

Fortunately, comparing functions in the MATLAB Spatial Econometrics toolbox, Python PySAL functions and the R **spdep** and **sphet** packages is eased by the fact that the code is open source, and so open to scrutiny. We have also benefited from answers to questions given by developers of these implementations, and by developers of Stata spatial econometrics functions.

What remains is to encourage researchers who use these and other software applications to take active part in discussion lists, where more experienced users can offer advice to those starting to discover the attractions of using spatial econometrics tools to tackle empirical economic questions. Once more real-world examples of the application of, for instance, impact measures, have been published, the usefulness of such advances will become more evident. Having multiple implementation in different application languages provides users with more choice, and, as we have seen, constitutes a "reality check" that gives insight into the ways that formulae can be rendered into code.

References

Anselin L (1988a). "Lagrange multiplier test diagnostics for spatial dependence and spatial heterogeneity." *Geographical Analysis*, **20**, 1–17.

Anselin L (1988b). *Spatial Econometrics: Methods and Models.* Kluwer, Dordrecht.

Anselin L (2010). "Thirty years of spatial econometrics." *Papers in Regional Science*, **89**(1), 3–25.

Anselin L (2013). "GMM Estimation of Spatial Error Autocorrelation with and without Heteroskedasticity." Manuscript.

Anselin L, Bera A (1998). "Spatial dependence in linear regression models with an introduction to spatial econometrics." In A Ullah, D Giles (eds.), "Handbook of Applied Economic Statistics," pp. 237–289. Marcel Dekker, New York.

Anselin L, Bera A, Florax R, Yoon M (1996). "Simple diagnostic tests for spatial dependence." *Regional Science and Urban Economics*, **26**, 77–104.

Anselin L, Lozano-Gracia N (2008). "Errors in variables and spatial effects in hedonic house price models of ambient air quality." *Empirical Economics*, **34**(1), 5–34.

Anselin L, Syabri I, Kho Y (2006). "GeoDa: An introduction to spatial data analysis." *Geographical Analysis*, **38**(1), 5–22. ISSN 0016-7363.

Aptech (2007). *GAUSS User Guide*. Maple Valley WA. Aptech Systems, Inc, URL http://www.aptech.com/wp-content/uploads/2012/08.

Arraiz I, Drukker DM, Kelejian HH, Prucha IR (2010). "A spatial Cliff-Ord-type model with heteroskedastic innovations: small and large sample results." *Journal of Regional Science*, **50**(2), 592–614.

Bivand R (2013). *spdep: Spatial dependence: weighting schemes, statistics and models*. R package version 0.5-56.

Bivand RS (2002). "Spatial econometrics functions in R: Classes and methods." *Journal of Geographical Systems*, **4**, 405–421.

Bivand RS (2006). "Implementing spatial data analysis software tools in R." *Geographical Analysis*, **38**(1), 23–40. ISSN 0016-7363.

Bivand RS (2012). "After 'Raising the Bar': applied maximum likelihood estimation of families of models in spatial econometrics." *Estadística Española*, **54**, 71–88.

Bivand RS, Gebhardt A (2000). "Implementing functions for spatial statistical analysis using the R language." *Journal of Geographical Systems*, **2**, 307–317.

Bivand RS, Hauke J, Kossowski T (2013). "Computing the Jacobian in Gaussian Spatial Autoregressive Models: An Illustrated Comparison of Available Methods." *Geographical Analysis*, **45**(2), 150–179.

Bivand RS, Pebesma EJ, Gómez-Rubio V (2008). *Applied Spatial Data Analysis with R*. Springer, New York.

Brent R (1973). *Algorithms for Minimization without Derivatives*. Prentice-Hall, Englewood Cliffs N.J.

Burridge P (1980). "On the Cliff-Ord test for spatial correlation." *Journal of the Royal Statistical Society B*, **42**, 107–108.

Cliff A, Ord J (1972). "Testing for Spatial Autocorrelation among Regression Residuals." *Geographical Analysis*, **4**, 267–284.

Das D, Kelejian HH, Prucha IR (2003). "Finite sample properties of estimators of spatial autoregressive models with autoregressive disturbances." *Papers in Regional Science*, **82**, 1–27.

Drukker D, Peng H, Prucha I, Raciborski R (2011a). "Creating and managing spatial-weighting matrices using the spmat command." *Technical report*, Technical report, StataCorp.

Drukker D, Prucha I, Raciborski R (2011b). "A command for estimating spatial-autoregressive models with spatial-autoregressive disturbances and additional endogenous variables." *The Stata Journal*, **1**(1), 1–13.

Drukker D, Prucha I, Raciborski R (2011c). "Maximum-likelihood and generalized spatial two-stage least-squares estimators for a spatial-autoregressive model with spatialautoregressive disturbances." *Technical report*, Technical report, StataCorp.

Drukker DM, Egger P, Prucha IR (2013). "On Two-Step Estimation of a Spatial Autoregressive Model with Autoregressive Disturbances and Endogenous Regressors." *Econometric Reviews*, **32**(5-6), 686–733.

Florax R, Folmer H (1992). "Specification and estimation of spatial linear regression models: Monte Carlo evaluation of pre-test estimators." *Regional Science and Urban Economics*, **22**, 405–432.

Florax R, Folmer H, Rey S (2003). "Specification searches in spatial econometrics: the relevance of Hendry's methodology." *Regional Science and Urban Economics*, **33**, 557–579.

Gould W, Pitblado J, Poi B (2010). *Maximum Likelihood Estimation With Stata.* StataCorp LP, College Station, TX.

Griffith DA, Layne LJ (1999). *A casebook for spatial statistical data analysis.* Oxford University Press, New York.

IBM Corporation (2010). *IBM SPSS Statistics 19.* Armonk, NY. IBM Corporation, URL http://www-01.ibm.com/software/analytics/spss/.

Kelejian H, Piras G (2011). "An extension of Kelejian's J-test for non-nested spatial models." *Regional Science and Urban Economics*, **41**, 281–292.

Kelejian H, Prucha I (1998). "Generalized spatial two-stage least squares procedure for estimating a spatial autoregressive model with autoregressive disturbances." *Journal of Real Estate Finance and Economics*, **17**(1), 99–121.

Kelejian HH, Prucha IR (1999). "A Generalized Moments Estimator for the Autoregressive Parameter in a Spatial Model." *International Economic Review*, **40**, 509–533.

Kelejian HH, Prucha IR (2007). "HAC estimation in a spatial framework." *Journal of Econometrics*, **140**(1), 131–154.

Kelejian HH, Prucha IR (2010). "Specification and estimation of spatial autoregressive models with autoregressive and heteroskedastic disturbances." *Journal of Econometrics*, **157**(1), 53–67.

Kelejian HH, Prucha IR, Yuzefovich Y (2004). "Instrumental variable estimation of a spatial autoregressive model with autoregressive disturbances: Large and small sample results." In JP LeSage, RK Pace (eds.), "Advances in Econometrics: Spatial and Spatiotemporal Econometrics," pp. 163–198. Elsevier Science Ltd., Oxford, UK.

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 (2003). "Best spatial two-stage least squares estimators for a spatial autoregressive model with autoregressive disturbances." *Econometric Reviews*, **22**, 307–335.

Lee L (2007). "GMM and 2SLS estimation of mixed regressive, spatial autoregressive models." *Journal of Econometrics*, **137**, 489–514.

LeSage J, Fischer M (2008). "Spatial Growth Regression: Model Specification, Estimation and Interpretation." *Spatial Economic Analysis*, **3**, 275–304.

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

McMillen D (2012). *McSpatial: Nonparametric spatial data analysis*. R package version 1.1.1, URL http://cran.r-project.org/src/contrib/Archive/McSpatial/McSpatial_1.1.1.tar.gz.

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

Nash JC, Varadhan R (2011). "Unifying Optimization Algorithms to Aid Software System Users: **optimx** for R." *Journal of Statistical Software*, **43**(9), 1–14.

Ord J (1975). "Estimation methods for models of spatial interaction." *Journal of the American Statistical Association*, **70**(349), 120–126.

Piras G (2010). "sphet: Spatial Models with Heteroskedastic Innovations in R." *Journal of Statistical Software*, **35**(1), 1–21.

Pisati M (2001). "sg162. Tools for spatial data analysis." *Stata Technical Bulletin*, **60**, 21–36.

Powers EL, Wilson JK (2004). "Access Denied: The Relationship between Alcohol Prohibition and Driving under the Influence." *Sociological Inquiry*, **74**(3), 318–337.

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/.

Rey S (2009). "Show me the code: spatial analysis and open source." *Journal of Geographical Systems*, **11**, 191–207.

Rey SJ, Anselin L (2010). "PySAL: A Python Library of Spatial Analytical Methods." In MM Fischer, A Getis (eds.), "Handbook of Applied Spatial Analysis," pp. 175–193. Springer, Berlin.

Ripley BD (1981). *Spatial Statistics*. Wiley, New York.

SAS Institute (2008). *SAS/IML 9.2 User's Guide*. Cary, NC. SAS Institute Inc, URL http://www.sas.com/.

Schnabel RB, Koontz JE, Weiss BE (1985). "A Modular System of Algorithms for Unconstrained Minimization." *ACM Transactions on Mathematical Software (TOMS)*, **11**(4), 419–440.

StataCorp (2007). *Stata Statistical Software: Release 10*. College Station, TX. StataCorp LP, URL http://www.stata.com/.

van Rossum G (1995). *Python Reference Manual*. CWI Report, URL http://ftp.cwi.nl/CWIreports/AA/CS-R9526.ps.Z.

Ward MD, Gleditsch KS (2008). *Spatial Regression Models*. Sage, Thousand Oaks, CA.

Affiliation:

Roger Bivand
Department of Economics
Norwegian School of Economics
Helleveien 30
5045 Bergen, Norway
E-mail: Roger.Bivand@nhh.no