Comparing Implementations of Estimation Methods for Spatial Econometrics

Roger Bivand   Gianfranco Piras

Norwegian School of Economics

Regional Research Institute at West Virginia University

22 October 2013

Overview

Outline

- Recent advances in spatial econometrics model fitting techniques have made it more desirable to be able to compare results
- Results should correspond between implementations using different applications
- A broad range of model fitting techniques are provided by the contributed R packages for spatial econometrics
- These model fitting techniques are associated with methods for estimating impacts and some tests, which will also be presented and compared

Background

- The use of spatial econometrics tools was widened by the ease with which methods and examples presented in Anselin (1988) could be reproduced using SpaceStat$^{\mathrm{TM}}$ (Anselin, 1992), written in Gauss$^{\mathrm{TM}}$ (Aptech, 2007)
- It was rapidly complemented by the Spatial Econometrics toolbox for Matlab$^{\mathrm{TM}}$(MATLAB, 2011), provided as source code together with extensive documentation (see also LeSage and Pace, 2009)
- A suite of commands for spatial data analysis for use with Stata$^{\mathrm{TM}}$ (StataCorp, 2007) was provided by Maurizio Pisati, and macros for Minitab$^{\mathrm{TM}}$ and SAS$^{\mathrm{TM}}$ were also made available
- The thrust of SpaceStat$^{\mathrm{TM}}$ has largely been taken over by GeoDa (Anselin et al., 2006), and more recently by GeoDa

Today's software

- There is now much more software available for spatial econometrics
- Stata$^{\mathrm{TM}}$ with sppack and Matlab$^{\mathrm{TM}}$ with Spatial Econometrics Toolbox are mainstream programmes; the Matlab$^{\mathrm{TM}}$ toolbox remains in the public domain, and has a community of contributors
- GeoDa and PySAL are open source, with code hosted on Google, binary versions for common platforms, and a community of users
- R with **spdep** (Bivand, 2013), **sphet**, **McSpatial** and other contributed packages is open source, and the packages are cross-platform; the packages also have a community of users and developers

Why compare?

- In the spirit of Rey (2009), this comparison will attempt to examine some features of the implementation of functions for fitting spatial econometrics models
- Firstly, it may be useful to show which kinds of functions for model fitting are available
- Next, it is comforting when one can show that fitting the same model on the same data using different implementations gives the same results
- Finally, if the results are not the same, it is helpful to be able to show why they vary, possibly because of different design choices in implementation
- Because Millo and Piras (2012) provide recent comparative results for spatial panel models, we restrict our consideration to cross-sectional models

Framework

- Initially, we describe the framework used for our comparative study, and the data set chosen for use
- Next we define the cross-sectional models to be compared
- 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
- Next come maximum likelihood estimators, focussing on the consequences of details in the choices of numerical methods across the alternatives, before examining impact measures implementations

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
- 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

Running the script

- This R script was then used to run R code to estimate chosen spatial econometrics models, and to
- It also wrote scripts for MATLAB, Stata and Python
- 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**

Collating output

- A second unified script was used to coordinate and document the collation of results from the four applications into tabular form
- 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 (Bengtsson, 2005); and from Python using the `npyLoad` function from the **RcppCNPy** package
- The tables for presentation were then formatted using the same rounding arguments either for the whole table or row-wise
- The remaining differences, if any, come from differences in the implementations, and it is these we intend to account for as far as possible

Platform

- 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, 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 Python 2.7 (32-bit) with **PySAL** 1.4
- Local modifications were made in a copy of the Spatial Econometrics Toolbox kept by agreement with its authors as a subdirectory on `https://r-forge.r-project.org/projects/spdep2/`

Numerical functions

- We can see from the comparison of OLS results for the selected data set shown below that the linear algebra output of the applications used is identical
- From examining source code, the GM methods in **PySAL** use the **SciPy** (Jones et al., 01) `fmin_l_bfgs_b`, function in the optimize module, a quasi-Newton function for bound-constrained optimization.
- In **sphet**, use is made of the `nlminb` function; the same function is used by default for fitting in **spdep** when more than one parameter is to be optimised
- For bounded line search in **spdep**, use is made of the `optimize` function, based on Brent (1973)

Numerical functions

- The GM functions in the Spatial Econometrics toolbox use an included function `minz` contributed by Michael Cliff
- The MATLAB `fminbnd` function also based on Brent (1973) is used for bounded line search
- 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
- The default numerical optimizer in Stata implementations is `"nr"`, a Stata-modified Newton-Raphson algoritm, but other algorithms may be chosen (Gould et al., 2010)

US Driving Under the Influence (DUI) county data set

- We use the simulated US Driving Under the Influence (DUI) county data set used in Drukker et al. (2011a,c,b); the data used is simulated for 3109 counties, and uses simulations from variables used by Powers and Wilson (2004)
- 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, simulated DUI data set

	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

OLS results, simulated DUI data set

	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)

Spatial weights

Drukker et al. (2011c) do not specify how spatial dependence was introduced into the dependent variable and/or residuals. 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:

```
R> library("rgeos")
R> strt <- gUnarySTRtreeQuery(ccounty)
R> library("spdep")
R> nblist <- poly2nb(ccounty, foundInBox = strt)
R> nblist

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

Spatial weights

- We used row-standardised spatial weights, **W**, where the county contiguities c_{ij}, taking values of 1 if contiguous, and 0 otherwise, are row-standardised by dividing by row sums
- It turned out that the spatial weights used in estimation in Drukker et al. (2011c) were in fact minmax-normalised
- We think that spatial dependence was introduced in the data set using minmax-normalised 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
- We chose to use row standardisation here, because row standardisation is often encountered in applied work, and weak spatial dependence may be more challenging for implementations

A general spatial model

- The present discussion is almost entirely based on Kelejian and Prucha (2010), Drukker et al. (2013), Arraiz et al. (2010) and Drukker et al. (2011b) that provide some important extensions to Kelejian and Prucha (1998, 1999)
- Specifically, the point of departure will be the following Cliff-Ord spatial model:

$$\mathbf{y} = \mathbf{Y}\pi + \mathbf{X}\beta + \rho_{\mathrm{Lag}}\mathbf{W}\mathbf{y} + \mathbf{u} \tag{1}$$

- 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

A general spatial model

- A spatial lag of the matrix of observations on the exogenous variables **WX** may be added to the model, see Elhorst (2010) and LeSage and Pace (2009)
- The error vector **u** follows a spatial autoregressive process of the form:

$$\mathbf{u} = \rho_{\mathrm{Err}}\mathbf{M}\mathbf{u} + \varepsilon \qquad (2)$$

- where ρ_{Err} is a scalar parameter generally referred to as the spatial autoregressive parameter, **M** is an $n \times n$ spatial weighting matrix that may or may not be the same as **W**
- R and Stata allow **W** and **M** to differ

A general spatial model

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

$$\mathbf{y} = \mathbf{Z}\delta + \mathbf{u} \tag{3}$$

- 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
- 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

Notation

- Here we adopt the notation ρ_{Lag} for the spatial autoregressive parameter on the spatially lagged dependent variable **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 (1988) and LeSage and Pace (2009) (and many others) using ρ for the spatial autoregressive parameter on the lagged dependent variable **y**, and λ for the spatial autoregressive parameter on the lagged residuals
- Kelejian and Prucha (1998, 1999) (and many others) adopt the opposite notation, using λ for the spatial autoregressive parameter on the lagged dependent variable **y**, and ρ for the spatial autoregressive parameter on the lagged residuals
- The names used for models also vary between software implementations

Restrictions on the general model

- The general model (Equation 1) 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 $\rho_{\mathrm{Err}} = 0$, and the spatial error model with $\rho_{\mathrm{Lag}} = 0$
- The spatial error model with no endogenous variables is:

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{u}, \mathbf{u} = \rho_{\mathrm{Err}}\mathbf{M}\mathbf{u} + \varepsilon \quad (4)$$

- The spatial lag model with no endogenous variables is:

$$\mathbf{y} = \mathbf{X}\beta + \rho_{\mathrm{Lag}}\mathbf{W}\mathbf{y} + \varepsilon \quad (5)$$

Feedback when ρ_{Lag} is included

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

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

  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 $\rho_{\mathrm{Lag}} = 0$
- The awkward $n \times n$ $S_r(\mathbf{W}) = ((\mathbf{I} - \rho_{\mathrm{Lag}} \mathbf{W})^{-1} \mathbf{I} \beta_r)$ matrix term is needed to calculate impact measures (extra care needed if $\mathbf{Y}\pi$ included)

Comparing GMM implementations

- Given the simultaneous presence of the endogenous variables on the right hand side of Equation 1 and the spatially autocorrelated residuals, IV and GMM estimators alternate
- These are based on a set of linear and quadratic moment conditions of the form:

$$E\mathbf{H}^\top\varepsilon = 0 \quad (6)$$
$$E\varepsilon^\top\mathbf{A}\varepsilon = 0 \quad (7)$$

  where $\mathbf{H}$ is an $n \times p$ non-stochastic matrix of instruments, and $\mathbf{A}$ is an $n \times n$ weighting matrix
- The spatial Cochrane-Orcutt transformation of the model is:

$$\mathbf{y}^\star = \mathbf{Z}^\star\delta + \varepsilon \quad (8)$$

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

Comparing GMM implementations

- 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}
- 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

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 $\mathbf{MWy}$
- Given the reduced form of the model

$$\mathbf{y} = (\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W})^{-1}(\mathbf{X}\beta + \mathbf{u}) \tag{9}$$

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 et al., 2004; Das et al., 2003)

SARAR model

- Given that the roots of $\rho_{\text{Lag}}\mathbf{W}$ 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

$$\mathbf{H} = (\mathbf{X}, \mathbf{WX}, \mathbf{W^2X}, \ldots, \mathbf{W^qX}, \mathbf{MX}, \mathbf{MWX}, \ldots, \mathbf{MW^qX}) \qquad (10)$$

  where q is a pre-selected finite constant and is generally set to 2 in applied studies
- 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
- 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

Moment conditions

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

$$E[\varepsilon^\top \mathbf{A}_s \varepsilon] = 0 \tag{11}$$

- 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
- Specific suggestions for $\mathbf{A}_s$ are given below. In general, such choices will depend on whether or not the model assumes heteroskedasticity

Moment conditions

- Drukker et al. (2013) suggest, for the homoskedastic case, the following expressions:

$$\mathbf{A}_1 = \left\{1 + [n^{-1} tr(\mathbf{M}^\top \mathbf{M})]^2\right\}^{-1} [\mathbf{M}^\top \mathbf{M} - n^{-1} tr(\mathbf{M}^\top \mathbf{M})\mathbf{I}] \tag{12}$$

and

$$\mathbf{A}_2 = \mathbf{M} \tag{13}$$

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

$$\mathbf{A}_1 = \mathbf{M}^\top \mathbf{M} - diag(\mathbf{M}^\top \mathbf{M}) \tag{14}$$

and

$$\mathbf{A}_2 = \mathbf{M} \tag{15}$$

Homoskedasticity without additional endogenous variables

- There are various implementations of the GMM general model (under homoskedasticity)
- For simplicity, in all models it is assumed that **W** and **M** are the same
- The R function `gstsls` available from **spdep**, the Spatial Econometrics Toolbox function `sac_gmm`, and **PySAL** `GM_Combo` are based on the Kelejian and Prucha (1999) moment conditions
- `spreg` in **sphet**, the Stata function `spreg gs2sls`, and **PySAL** `GM_Combo_Hom` are based on the Drukker et al. (2013) moment conditions

Kelejian and Prucha (1999) moment conditions

	R `gstsls`	**PySAL** `GM_Combo`	SE `sac_gmm`
(Intercept)	−6.409919	−6.409919	−6.403747
	(0.418363)	(0.417959)	(0.417963)
police	0.598107	0.598107	0.598107
	(0.014918)	(0.014903)	(0.014918)
nondui	0.000247	0.000247	0.000247
	(0.001087)	(0.001086)	(0.001087)
vehicles	0.015712	0.015712	0.015712
	(0.000669)	(0.000668)	(0.000669)
dry	0.106088	0.106088	0.106088
	(0.034962)	(0.034929)	(0.034962)
ρ_{Lag}	0.046928	0.046928	0.046926
	(0.016982)	(0.016966)	(0.016982)
ρ_{Err}	0.000957	0.000957	0.000957
			(0.009316)

Kelejian and Prucha (1999) moment conditions

- 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 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 second step, the matrix of instruments used by `sac_gmm` includes the intercept (untransformed), the transformed exogenous variables (say $\mathbf{X}^\star$), and their spatial lags
- The R and **PySAL** implementations use $\mathbf{X}^\star$ (which may or may not include an intercept), and then spatial lags of $\mathbf{X}$
- Differences in reported standard errors relate to a degrees of freedom correction in the R and MATLAB cases

Drukker et al. (2013) moment conditions

	R `spreg`	Stata `spreg`	**PySAL** `GM_Combo_Hom`
(Intercept)	−6.409969	−6.409968	−6.409969
	(0.416359)	(0.416359)	(0.416359)
police	0.598102	0.598102	0.598102
	(0.014907)	(0.014907)	(0.014907)
nondui	0.000247	0.000247	0.000247
	(0.001086)	(0.001086)	(0.001086)
vehicles	0.015713	0.015713	0.015713
	(0.000668)	(0.000668)	(0.000668)
dry	0.106098	0.106098	0.106098
	(0.034927)	(0.034927)	(0.034927)
ρ_{Lag}	0.046932	0.046932	0.046932
	(0.016927)	(0.016927)	(0.016927)
ρ_{Err}	−0.005621	−0.005621	−0.005621
	(0.034984)	(0.034984)	(0.034984)

Drukker et al. (2013) moment conditions

- Apart from a different trailing 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 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)

Homoskedasticity with additional endogenous variables

- Undoubtably, the size of the `police` force may be related with the arrest rates (`dui`), so `police` is treated as an endogenous variable
- Drukker et al. (2011b) choose the dummy variable `elect` (where elect is 1 if a county government faces an election, 0 otherwise) as a valid instrument for `police`
- Results from `spreg` available from **sphet** under R, the Stata function `spreg` setting the option to `het`, and the function `GM_Combo_Het` from **PySAL**
- All implementations give the same results

Homoskedasticity with additional endogenous variables

	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)

Heteroskedasticity with and without additional endogenous variables

- Here, the errors are assumed to be heteroskedastic of unknown form
- Such models can be estimated without additional endogenous variables, or with `police` treated as endogenous
- It can be seen that the implementations in R and **PySAL** are identical (up to the sixth decimal), and that Stata only presents very minor differences
- These 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

Heteroskedasticity without additional endogenous variables

	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)

Heteroskedasticity with additional endogenous variables

	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)

W and **M** are different

- **W** and **M** need not to be the same in all applications
- Results are limited to the implementations of R and Stata; they are very close for the homoskedastic case (shown) and the heteroskedastic case
- **M** is defined as a row standardised six nearest neighbours matrix, treating the county centroid coordinates as projected, not geographical
- Since the endogeneity of the `police` variable is accommodated, the default value to compute the lagged "additional" instruments (i.e., `lag.instr`) was changed in R

W and **M** are different, `police` endogenous

	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)

Spatial lag model

- The estimation of the spatial lag model in Equation 5 can be easily approached by two stage least squares
- There are multiple functions that allow the estimation of the spatial lag model available from R under the **spdep** (`stsls`) and **sphet** (`spreg`) packages
- Given that we are considering the same matrix of instruments, the coefficient values of all implementations agree exactly
- 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 it is simply divided by n

Spatial lag model

	R stsls	R spreg	SE sar_gmm
(Intercept)	−6.410152	−6.410152	−6.410152
	(0.418129)	(0.418129)	(0.418129)
police	0.598081	0.598081	0.598081
	(0.014918)	(0.014918)	(0.014918)
nondui	0.000247	0.000247	0.000247
	(0.001087)	(0.001087)	(0.001087)
vehicles	0.015714	0.015714	0.015714
	(0.000669)	(0.000669)	(0.000669)
dry	0.106134	0.106134	0.106134
	(0.034962)	(0.034962)	(0.034962)
ρ_{Lag}	0.046950	0.046950	0.046950
	(0.016977)	(0.016977)	(0.016977)

Spatial lag model

	Stata `spreg gs2sls`	**PySAL** `GM_Lag`
(Intercept)	−6.410152	−6.410152
	(0.417725)	(0.417725)
police	0.598081	0.598081
	(0.014904)	(0.014904)
nondui	0.000247	0.000247
	(0.001086)	(0.001086)
vehicles	0.015714	0.015714
	(0.000668)	(0.000668)
dry	0.106134	0.106134
	(0.034928)	(0.034928)
ρ_{Lag}	0.046950	0.046950
	(0.016960)	(0.016960)

Spatial lag model, `police` endogenous

	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)

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

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

$$u = \mathbf{R}\varepsilon \tag{16}$$

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:

$$\psi = n^{-1}\mathbf{H}^\top \omega \mathbf{H} \tag{17}$$

where $\omega = \mathbf{R}\mathbf{R}^\top$ denotes the VC matrix of ε

- Kelejian and Prucha (2007) suggest estimating the individual r, s elements of ψ as

$$\tilde{\psi}_{rs} = n^{-1} \sum_{i=1}^{n} \sum_{j=1}^{n} h_{ir} h_{js} \hat{\varepsilon}_i \hat{\varepsilon}_j K(d_{ij}^*/d) \tag{18}$$

where the subscripts refer to the elements of the matrix of instruments **H** and the vector of estimated residuals $\hat{\varepsilon}$

HAC estimation in a spatial framework

- 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 $(K(d^*_{ij}/d) = 0)$
- Based on Equation 18, the asymptotic variance covariance matrix $(\tilde{\phi})$ of the S2SLS estimator of the parameters vector is given by:

$$\tilde{\phi} = n^2(\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}})^{-1}\mathbf{Z}^\top\mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-1}\tilde{\psi}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{Z}(\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}})^{-1} \tag{19}$$

HAC estimation in a spatial framework

- Here we compare standard error estimates using a Triangular kernel with a variable bandwidth of the six nearest neighbours
- There are many available options for the kernel both in R and **PySAL**
- 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

HAC estimation, `police` endogenous

	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)

Spatial error model

- The first step of the estimation procedure is either OLS (when $\pi = 0$), or IV, when $\pi \neq 0$ and there are endogenous variable 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 $\mathbf{MX}$

Spatial error model

- Three are based on Kelejian and Prucha (1999) moment conditions, and three others on the Drukker et al. (2013) moment conditions
- In the first case, there are differences in terms of the standard errors; while `GMerrorsar` and `sem_gmm` produce an estimate for the standard error of the spatial coefficient, the `GM_Error` function in **PySAL** does not
- In the second, Stata and `spreg` (available from **sphet**) present exactly the same results, some distinctions are observed in **PySAL**
- In a spatial error model, a term in the underlying equations limits to zero; the implementations in R and Stata produce an estimate of this term, while **PySAL** set it to zero in the version used here

Spatial error model, Kelejian and Prucha (1999) moment conditions

	R `GMerrorsar`	**PySAL** `GM_Error`	SE `sem_gmm`
(Intercept)	−5.431921	−5.431921	−5.431921
	(0.229056)	(0.229052)	(0.229056)
police	0.599854	0.599854	0.599854
	(0.014888)	(0.014888)	(0.014888)
nondui	0.000257	0.000257	0.000257
	(0.001086)	(0.001086)	(0.001086)
vehicles	0.015612	0.015612	0.015612
	(0.000667)	(0.000667)	(0.000667)
dry	0.103654	0.103654	0.103654
	(0.034966)	(0.034966)	(0.034966)
ρ_{Err}	0.050883	0.050883	0.050883
	(0.080487)		(0.080487)

Spatial error model, Drukker et al. (2013) moment conditions

	R `spreg`	Stata `spreg gs2sls`	**PySAL** `GM_Error_Hom`
(Intercept)	−5.431959	−5.431959	−5.431960
	(0.229067)	(0.229067)	(0.229050)
police	0.599851	0.599851	0.599851
	(0.014890)	(0.014890)	(0.014887)
nondui	0.000257	0.000257	0.000257
	(0.001086)	(0.001086)	(0.001086)
vehicles	0.015612	0.015612	0.015612
	(0.000667)	(0.000667)	(0.000667)
dry	0.103663	0.103663	0.103663
	(0.034967)	(0.034967)	(0.034965)
ρ_{Err}	0.047050	0.047050	0.051491
	(0.029543)	(0.029543)	(0.028809)

Spatial error model with endogenous variables

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

Spatial error model, `police` endogenous

	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)

Comparing maximum likelihood estimation

- 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
- We described the numerical optimisers used in the various applications earlier

Numerical Hessian

- In many cases, the numerical optimisation functions can return numerical Hessians for use as estimators of the covariance matrix, 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 SE Toolbox, use is made of `fdhess`, 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`
- In R, use is made of `fdHess` from the **nlme** (Pinheiro et al., 2013) 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

Spatial lag model

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

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

- 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
- The sum of squares term can be calculated much faster than the log-determinant (Jacobian) term of the $n \times n$ sparse matrix $\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W}$
- The legacy method for computing the log-determinant term is to use eigenvalues of $\mathbf{W}$:

$$\ln(|\mathbf{I} - \rho_{\mathrm{Lag}}\mathbf{W}|) = \sum_{i=1}^{n} \ln(1 - \rho_{\mathrm{Lag}}\zeta_i) \tag{20}$$

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 et al. (2013)

Spatial lag model

	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)

Spatial lag model: log likelihood

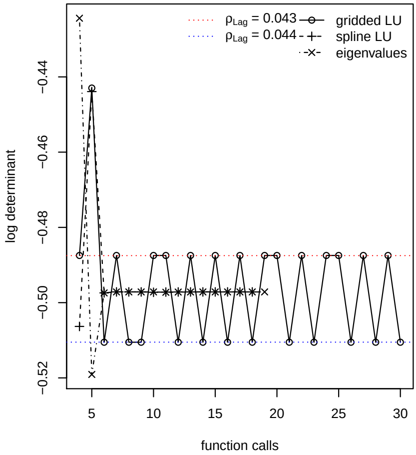
- One discrepancy that we can account for before presenting any further results is that the log-likelihood values at the optimimum differ between two implementations: 1551.08 in R **McSpatial** `sarml` and a similar value in the SE toolbox `sar` function
- R **spdep** `lagsarlm` and Stata `spreg ml` have -2628.58
- 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 $\frac{n}{2} log(\pi)$, and adding $\frac{n}{2} log(2\pi)$, we get -2628.58

Spatial lag model

- We see that the coefficient estimates of the R `lagsarlm` and Stata `spreg ml` implementations agree exactly
- 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; they were also using eigenvalues to compute the log-determinant values
- We will return below to differences in standard errors after explaining why the SE toolbox `sar` function yields different coefficient estimates

SE Toolbox log-determinant implementations

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. The figure shows the behaviour of the optimiser for `info.lflag` taking values of 0 — the gridded LU log-determinant values, and for two alternatives



SE Toolbox log-determinant implementations

	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)

Spatial lag standard errors

- 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*
- 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
- R `lagsarlm` can give quite similar results when using the numerical Hessian rather than analytical, asymptotic standard errors

Spatial lag standard errors

	R `sarml`	Stata `spreg ml`	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

Other ML estimators

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

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

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

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

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

Other ML estimators

- The general model is more demanding, and requires that ρ_{Lag} and ρ_{Err} be found by constrained numerical optimization in two dimensions
- Its log-likelihood, here assuming that the same spatial weights are used in both processes:

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

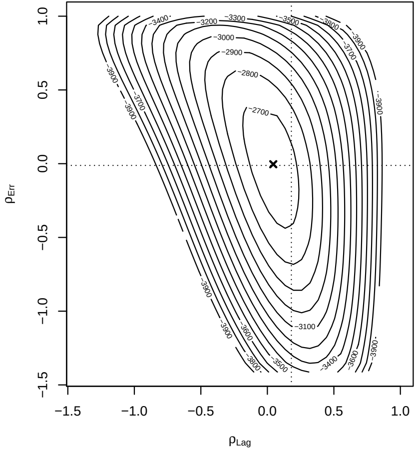
- The tuning of the constrained numerical optimization function, including the provision of starting values, reasonable stopping criteria, and also the choice of algorithm may all affect the results achieved

General model starting values

- The Stata implementation uses a grid search for initial values of $(\rho_{\rm Lag}, \rho_{\rm Err})$ (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
- 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 $(\rho_{\rm Lag}, \rho_{\rm Err})$, a full grid of nine points at the same settings, or user provided initial values

General model function surface

The surface of the objective function is flat, with a hallmark banana-shaped ridge (see also Bivand, 2012); note the closeness of ρ_{Err} to zero.



Other ML estimators

	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)

Other ML estimators

- 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

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
- The average direct impacts are represented by the sum of the diagonal elements of the $S_r(\mathbf{W})$ 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 differences between these two vectors of impacts
- Implementations in R and the SE toolbox provide impact measures and inference by Monte Carlo simulation from the fitted models

Impact measures 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

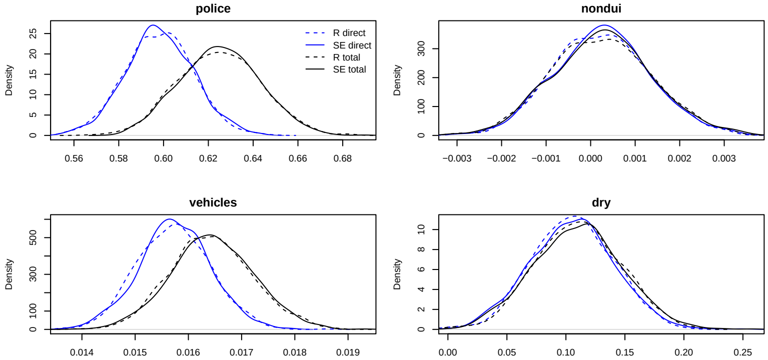
Comparing impact measures

- 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; here the calculated impact values for the fitted values of the β coefficients are returned in addition

Comparing impact measures

	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

Comparing impact measure simulations



Concluding remarks

- In conclusion, 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

Concluding remarks

- 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
- We have also benefited from answers to questions given by developers of these implementations, and by developers of Stata spatial econometrics functions
- 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. (1988). *Spatial Econometrics: Methods and Models*. Kluwer, Dordrecht.

Anselin, L. (1992). SpaceStat Tutorial. A Workbook for Using SpaceStat in the Analysis of Spatial Data.

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

Anselin, L., Syabri, I., and Kho, Y. (2006). GeoDa: An Introduction to Spatial Data Analysis. *Geographical Analysis*, 38(1):5–22.

Aptech (2007). *GAUSS User Guide*. Maple Valley WA. Aptech Systems, Inc.

Arraiz, I., Drukker, D. M., Kelejian, H. H., and Prucha, I. R. (2010). A Spatial Cliff-Ord-type Model with Heteroskedastic Innovations: Small and Large Sample Results. *Journal of Regional Science*, 50(2):592–614.

Bengtsson, H. (2005). **R.matlab** - Local and Remote Matlab Connectivity in R. Mathematical Statistics, Centre for Mathematical Sciences, Lund University, Sweden, 2005. (manuscript in progress).

Bivand, R. S. (2012). After 'Raising the Bar': Applied Maximum Likelihood Estimation of Families of Models in Spatial Econometrics. *Estadística Española*, 54:71–88.

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

Bivand, R. S., Hauke, J., and Kossowski, T. (2013). Computing the Jacobian in Gaussian spatial autoregressive models: An illustrated comparison of available methods. *Geographical Analysis*, 45(2):150–179.

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

Das, D., Kelejian, H. H., and Prucha, I. R. (2003). Finite Sample Properties of Estimators of Spatial Autoregressive Models with Autoregressive Disturbances. *Papers in Regional Science*, 82:1–27.

Drukker, D. M., Egger, P., and Prucha, I. R. (2013). On two-step estimation of a spatial autoregressive model with autoregressive disturbances and endogenous regressors. *Econometric Reviews*, 32(5-6):686–733.

Drukker, D. M., Peng, H., Prucha, I. R., and Raciborski, R. (2011a). Creating and Managing Spatial-Weighting Matrices Using the **spmat** Command. Technical report, Technical report, StataCorp.

Drukker, D. M., Prucha, I. R., and 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. M., Prucha, I. R., and Raciborski, R. (2011c). Maximum Likelihood and Generalized Spatial