The Stata Journal (2013)
13, Number 2, pp. 221–241

Maximum likelihood and generalized spatial two-stage least-squares estimators for a spatial-autoregressive model with spatial-autoregressive disturbances

David M. Drukker
StataCorp
College Station, TX
ddrukker@stata.com

Ingmar R. Prucha
Department of Economics
University of Maryland
College Park, MD
prucha@econ.umd.edu

Rafal Raciborski
StataCorp
College Station, TX
rraciborski@stata.com

Abstract. We describe the `spreg` command, which implements a maximum likelihood estimator and a generalized spatial two-stage least-squares estimator for the parameters of a linear cross-sectional spatial-autoregressive model with spatial-autoregressive disturbances.

Keywords: st0291, spreg, spatial-autoregressive models, Cliff–Ord models, maximum likelihood estimation, generalized spatial two-stage least squares, instrumental-variable estimation, generalized method of moments estimation, prediction, spatial econometrics, spatial statistics

1 Introduction

Cliff–Ord (1973, 1981) models, which build on Whittle (1954), allow for cross-unit interactions. Many models in the social sciences, biostatistics, and geographic sciences have included such interactions. Following Cliff and Ord (1973, 1981), much of the original literature was developed to handle spatial interactions. However, space is not restricted to geographic space, and many recent applications use these techniques in other situations of cross-unit interactions, such as social-interaction models and network models; see, for example, Kelejian and Prucha (2010) and Drukker, Egger, and Prucha (2013) for references. Much of the nomenclature still includes the adjective "spatial", and we continue this tradition to avoid confusion while noting the wider applicability of these models. For texts and reviews, see, for example, Anselin (1988, 2010), Arbia (2006), Cressie (1993), Haining (2003), and LeSage and Pace (2009).

The simplest Cliff–Ord model only considers spatial spillovers in the dependent variable, with spillovers modeled by including a right-hand-side variable known as a spatial lag. Each observation of the spatial-lag variable is a weighted average of the values of the dependent variable observed for the other cross-sectional units. The matrix containing the weights is known as the spatial-weighting matrix. This model is frequently referred to as a spatial-autoregressive (SAR) model. A generalized version of this model also allows for the disturbances to be generated by a SAR process. The combined SAR model

with SAR disturbances is often referred to as a SARAR model; see Anselin and Florax (1995).[1]

In modeling the outcome for each unit as dependent on a weighted average of the outcomes of other units, SARAR models determine outcomes simultaneously. This simultaneity implies that the ordinary least-squares estimator will not be consistent; see Anselin (1988) for an early discussion of this point.

In this article, we describe the `spreg` command, which implements a maximum likelihood (ML) estimator and a generalized spatial two-stage least-squares (GS2SLS) estimator for the parameters of a SARAR model with exogenous regressors. For discussions of the ML estimator, see, for example, the above cited texts and Lee (2004) for the asymptotic properties of the estimator. For a discussion of the estimation theory for the implemented GS2SLS estimator, see Arraiz et al. (2010) and Drukker, Egger, and Prucha (2013), which build on Kelejian and Prucha (1998, 1999, 2010) and the references cited therein.

Section 2 describes the SARAR model. Section 3 describes the `spreg` command. Section 4 provides some examples. Section 5 describes postestimation commands. Section 6 presents methods and formulas. The conclusion follows.

We use the notation that for any matrix $\mathbf{A}$ and vector $\mathbf{a}$, the elements are denoted as a_{ij} and a_i, respectively.

2 The SARAR model

The `spreg` command estimates the parameters of the cross-sectional model ($i = 1, \ldots, n$)

$$
\begin{aligned}
y_i &= \lambda \textstyle\sum_{j=1}^{n} w_{ij} y_j + \sum_{p=1}^{k} x_{ip}\beta_p + u_i \\
u_i &= \rho \textstyle\sum_{j=1}^{n} m_{ij} u_j + \varepsilon_i
\end{aligned}
$$

or more compactly,

$$
\mathbf{y} = \lambda \mathbf{W}\mathbf{y} + \mathbf{X}\boldsymbol{\beta} + \mathbf{u} \tag{1}
$$

$$
\mathbf{u} = \rho \mathbf{M}\mathbf{u} + \boldsymbol{\epsilon} \tag{2}
$$

1. These models are also known as Cliff–Ord models because of the impact that Cliff and Ord (1973, 1981) had on the subsequent literature. To avoid confusion, we simply refer to these models as SARAR models while still acknowledging the importance of the work of Cliff and Ord.

where

- $\mathbf{y}$ is an $n \times 1$ vector of observations on the dependent variable;
- $\mathbf{W}$ and $\mathbf{M}$ are $n \times n$ spatial-weighting matrices (with 0 diagonal elements);
- $\mathbf{Wy}$ and $\mathbf{Mu}$ are $n \times 1$ vectors typically referred to as spatial lags, and λ and ρ are the corresponding scalar parameters typically referred to as SAR parameters;
- $\mathbf{X}$ is an $n \times k$ matrix of observations on k right-hand-side exogenous variables (where some of the variables may be spatial lags of exogenous variables), and $\boldsymbol{\beta}$ is the corresponding $k \times 1$ parameter vector;
- $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of innovations.

The model in (1) and (2) is a SARAR with exogenous regressors. Spatial interactions are modeled through spatial lags. The model allows for spatial interactions in the dependent variable, the exogenous variables, and the disturbances.[2]

The spatial-weighting matrices $\mathbf{W}$ and $\mathbf{M}$ are taken to be known and nonstochastic. These matrices are part of the model definition, and in many applications, $\mathbf{W} = \mathbf{M}$. Let $\overline{\mathbf{y}} = \mathbf{Wy}$. Then

$$\overline{y}_i = \sum_{j=1}^{n} w_{ij} y_j$$

which clearly shows the dependence of y_i on neighboring outcomes via the spatial lag $\overline{y}_i$. By construction, the spatial lag $\mathbf{Wy}$ is an endogenous variable. The weights w_{ij} will typically be modeled as inversely related to some measure of proximity between the units. The SAR parameter λ measures the extent of these interactions. For further discussions of spatial-weighting matrices and the parameter space for the SAR parameter, see, for example, the literature cited in the introduction, including Kelejian and Prucha (2010); see Drukker et al. (2013) for more information about creating spatial-weighting matrices in Stata.

The innovations $\boldsymbol{\epsilon}$ are assumed to be independent and identically distributed (IID) or independent but heteroskedastically distributed, where the heteroskedasticity is of unknown form. The GS2SLS estimator produces consistent estimates in either case when the `heteroskedastic` option is specified; see Kelejian and Prucha (1998, 1999, 2010), Arraiz et al. (2010), and Drukker, Egger, and Prucha (2013) for discussions and formal results. The ML estimator produces consistent estimates in the IID case but generally not in the heteroskedastic case; see Lee (2004) for some formal results for the ML estimator, and see Arraiz et al. (2010) for evidence that the ML estimator does not generally produce consistent estimates in the heteroskedastic case.

2. An extension of the model to a limited-information-systems framework with additional endogenous right-hand-side variables is considered in Drukker, Prucha, and Raciborski (2013), which discusses the `spivreg` command.

Because the model in (1) and (2) is a first-order SAR model with first-order SAR disturbances, it is also referred to as a SARAR$(1,1)$ model, which is a special case of the more general SARAR(p,q) model. We refer to a SARAR$(1,1)$ model as a SARAR model. When $\rho = 0$, the model in equations (1) and (2) reduces to the SAR model $\mathbf{y} = \lambda\mathbf{W}\mathbf{y} + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$. When $\lambda = 0$, the model in equations (1) and (2) reduces to $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$ with $\mathbf{u} = \rho\mathbf{M}\mathbf{u} + \boldsymbol{\epsilon}$, which is sometimes referred to as the SAR error model. Setting $\rho = 0$ and $\lambda = 0$ causes the model in equations (1) and (2) to reduce to a linear regression model with exogenous variables.

`spreg` requires that the spatial-weighting matrices $\mathbf{M}$ and $\mathbf{W}$ be provided in the form of an `spmat` object as described in Drukker et al. (2013). `spreg gs2sls` supports both general and banded spatial-weighting matrices; `spreg ml` supports general matrices only.

3 The spreg command

3.1 Syntax

`spreg ml` *depvar* [*indepvars*] [*if*] [*in*], `id(`*varname*`)` [`noconstant` `level(`#`)` `dlmat(`*objname*[`, eig`]`)` `elmat(`*objname*[`, eig`]`)` `constraints(`*constraints*`)` `gridsearch(`#`)` *maximize_options*]

`spreg gs2sls` *depvar* [*indepvars*] [*if*] [*in*], `id(`*varname*`)` [`noconstant` `level(`#`)` `dlmat(`*objname*`)` `elmat(`*objname*`)` `heteroskedastic` `impower(`*q*`)` *maximize_options*]

3.2 Options for spreg ml

`id(`*varname*`)` specifies a numeric variable that contains a unique identifier for each observation. `id()` is required.

`noconstant` suppresses the constant term in the model.

`level(`#`)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`.

`dlmat(`*objname*[`, eig`]`)` specifies an `spmat` object that contains the spatial-weighting matrix $\mathbf{W}$ to be used in the SAR term. `eig` forces the calculation of the eigenvalues of $\mathbf{W}$, even if *objname* already contains them.

`elmat(`*objname*[`, eig`]`)` specifies an `spmat` object that contains the spatial-weighting matrix $\mathbf{M}$ to be used in the spatial-error term. `eig` forces the calculation of the eigenvalues of $\mathbf{M}$, even if *objname* already contains them.

`constraints(`*constraints*`)`; see [R] **estimation options**.

`gridsearch(#)` specifies the fineness of the grid used in searching for the initial values of the parameters λ and ρ in the concentrated log likelihood. The allowed range is [`.001`, `.1`]. The default is `gridsearch(.1)`.

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(#)`, [`no`]`log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(`*init_specs*`)`; see [R] **maximize**. These options are seldom used. `from()` takes precedence over `gridsearch()`.

Options for spreg gs2sls

`id(`*varname*`)` specifies a numeric variable that contains a unique identifier for each observation. `id()` is required.

`noconstant` suppresses the constant term.

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`.

`dlmat(`*objname*`)` specifies an `spmat` object that contains the spatial-weighting matrix **W** to be used in the SAR term.

`elmat(`*objname*`)` specifies an `spmat` object that contains the spatial-weighting matrix **M** to be used in the spatial-error term.

`heteroskedastic` specifies that `spreg` use an estimator that allows the errors to be heteroskedastically distributed over the observations. By default, `spreg` uses an estimator that assumes homoskedasticity.

`impower(`q`)` specifies how many powers of **W** to include in calculating the instrument matrix **H**. The default is `impower(2)`. The allowed values of q are integers in the set $2, 3, \ldots, \lfloor\sqrt{n}\rfloor$, where n is the number of observations.

maximize_options: `iterate(#)`, [`no`]`log`, `trace`, `gradient`, `showstep`, `showtolerance`, `tolerance(#)`, and `ltolerance(#)`; see [R] **maximize**. `from(`*init_specs*`)` is also allowed, but because ρ is the only parameter in this optimization problem, only initial values for ρ may be specified.

3.3 Saved results

`spreg ml` saves the following in `e()`:

Scalars

`e(N)`	number of observations	`e(p)`	significance
`e(k)`	number of parameters	`e(rank)`	rank of `e(V)`
`e(df_m)`	model degrees of freedom	`e(converged)`	`1` if converged, `0` otherwise
`e(ll)`	log likelihood	`e(iterations)`	number of ML iterations
`e(chi2)`	χ^2		

Macros

`e(cmd)`	`spreg`	`e(user)`	name of likelihood-evaluator program
`e(cmdline)`	command as typed		
`e(depvar)`	name of dependent variable	`e(estimator)`	`ml`
`e(indeps)`	names of independent variables	`e(model)`	`lr`, `sar`, `sare`, or `sarar`
`e(title)`	title in estimation output	`e(constant)`	`noconstant` or `hasconstant`
`e(chi2type)`	type of model χ^2 test	`e(idvar)`	name of ID variable
`e(vce)`	`oim`	`e(dlmat)`	name of `spmat` object used in `dlmat()`
`e(technique)`	maximization technique		
`e(crittype)`	type of optimization	`e(elmat)`	name of `spmat` object used in `elmat()`
`e(estat_cmd)`	program used to implement `estat`	`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`		

Matrices

`e(b)`	coefficient vector	`e(gradient)`	gradient vector
`e(Cns)`	constraints matrix	`e(V)`	variance–covariance matrix of the estimators
`e(ilog)`	iteration log		

Functions

`e(sample)`	marks estimation sample

`spreg gs2sls` saves the following in `e()`:

Scalars

`e(N)`	number of observations	`e(converged)`	`1` if generalized method of moments converged, `0` otherwise
`e(k)`	number of parameters		
`e(rho_2sls)`	initial estimate of ρ		
`e(iterations)`	number of generalized method of moments iterations	`e(converged_2sls)`	`1` if two-stage least-squares converged, `0` otherwise
`e(iterations_2sls)`	number of two-stage least-squares iterations		

Macros

`e(cmd)`	`spreg`	`e(idvar)`	name of ID variable
`e(cmdline)`	command as typed	`e(dlmat)`	name of `spmat` object used in `dlmat()`
`e(estimator)`	`gs2sls`		
`e(model)`	`lr`, `sar`, `sare`, or `sarar`	`e(elmat)`	name of `spmat` object used in `elmat()`
`e(het)`	`homoskedastic` or `heteroskedastic`	`e(estat_cmd)`	program used to implement `estat`
`e(depvar)`	name of dependent variable		
`e(indeps)`	names of independent variables	`e(predict)`	program used to implement `predict`
`e(title)`	title in estimation output		
`e(exogr)`	exogenous regressors	`e(properties)`	`b V`
`e(constant)`	`noconstant` or `hasconstant`		
`e(H_omitted)`	names of omitted instruments in `H` matrix		

Matrices

`e(b)`	coefficient vector	`e(delta_2sls)`	initial estimate of β and λ
`e(V)`	variance–covariance matrix of the estimators		

Functions

`e(sample)`	marks estimation sample

4 Example

In our examples, we use `spreg.dta`, which contains simulated data on the number of arrests for driving under the influence for the continental U.S. counties.[3] We use a normalized contiguity matrix taken from Drukker et al. (2013). In Stata, we type

```
. use dui
. spmat use ccounty using ccounty.spmat
```

to read the dataset into memory and to put the spatial-weighting matrix into the `spmat` object `ccounty`. This row-normalized spatial-weighting matrix was created in Drukker et al. (2013, sec. 2.4) and saved to disk in Drukker et al. (2013, sec. 11.4).

Our dependent variable, `dui`, is defined as the alcohol-related arrest rate per 100,000 daily vehicle miles traveled (DVMT). Figure 1 shows the distribution of `dui` across counties, with darker colors representing higher values of the dependent variable. Spatial patterns of `dui` are clearly visible.

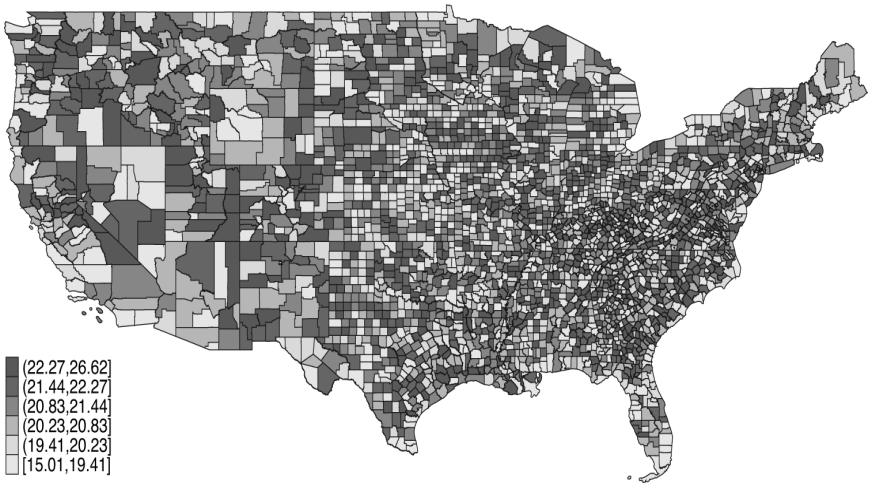

Figure 1. Hypothetical alcohol-related arrests for continental U.S. counties

3. The geographical location data came from the U.S. Census Bureau and can be found at ftp://ftp2.census.gov/geo/tiger/TIGER2008/. The variables are simulated but inspired by Powers and Wilson (2004).

Our explanatory variables include `police` (number of sworn officers per 100,000 DVMT); `nondui` (nonalcohol-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). In other words, in this illustration,
$\mathbf{X} = [$`police,nondui,vehicles,dry,intercept`$]$.

We obtain the GS2SLS parameter estimates of the SARAR model parameters by typing

```
. spreg gs2sls dui police nondui vehicles dry, id(id)
> dlmat(ccounty) elmat(ccounty) nolog
Spatial autoregressive model                    Number of obs   =     3109
(GS2SLS estimates)
```

dui	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
dui						
police	-.5591567	.0148772	-37.58	0.000	-.5883155	-.529998
nondui	-.0001128	.0005645	-0.20	0.842	-.0012193	.0009936
vehicles	.062474	.0006198	100.79	0.000	.0612592	.0636889
dry	.303046	.0183119	16.55	0.000	.2671553	.3389368
_cons	2.482489	.1473288	16.85	0.000	2.19373	2.771249
lambda						
_cons	.4672164	.0051261	91.14	0.000	.4571694	.4772633
rho						
_cons	.1932962	.0726583	2.66	0.008	.0508885	.3357038

Given the normalization of the spatial-weighting matrix, the parameter space for λ and ρ is taken to be the interval $(-1, 1)$; see Kelejian and Prucha (2010) for further discussions of the parameter space. The estimated λ is positive and significant, indicating moderate SAR dependence in `dui`. In other words, the `dui` alcohol-arrest rate for a given county is affected by the `dui` alcohol-arrest rates of the neighboring counties. This result may be because of coordination among police departments or because strong enforcement in one county leads some people to drink in neighboring counties.

The estimated ρ coefficient is positive, moderate, and significant, indicating moderate SAR dependence in the error term. In other words, an exogenous shock to one county will cause moderate changes in the alcohol-related arrest rate in the neighboring counties.

The estimated $\boldsymbol{\beta}$ vector does not have the same interpretation as in a simple linear model, because including a spatial lag of the dependent variable implies that the outcomes are determined simultaneously. We present one way to interpret the coefficients in section 5.

For comparison, we obtain the ML parameter estimates by typing

```
. spreg ml dui police nondui vehicles dry, id(id)
> dlmat(ccounty) elmat(ccounty) nolog
Spatial autoregressive model                    Number of obs   =     3109
(Maximum likelihood estimates)                  Wald chi2(4)    =  62376.4
                                                Prob > chi2     =   0.0000
```

dui	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
dui						
police	-.5593526	.014864	-37.63	0.000	-.5884854	-.5302197
nondui	-.0001214	.0005645	-0.22	0.830	-.0012279	.0009851
vehicles	.0624729	.0006195	100.84	0.000	.0612586	.0636872
dry	.3030522	.018311	16.55	0.000	.2671633	.3389412
_cons	2.490301	.1471885	16.92	0.000	2.201817	2.778785
lambda						
_cons	.4671198	.0051144	91.33	0.000	.4570957	.4771439
rho						
_cons	.1962348	.0711659	2.76	0.006	.0567522	.3357174
sigma2						
_cons	.0859662	.0021815	39.41	0.000	.0816905	.0902418

There are no apparent differences between the two sets of parameter estimates.

5 Postestimation commands

The postestimation commands supported by `spreg` include `estat`, `test`, and `predict`; see `help spreg postestimation` for the full list. Most postestimation methods have standard interpretations; for example, a Wald test is just a Wald test.

Predictions from SARAR models require some additional explanation. Kelejian and Prucha (2007) consider different information sets and define predictors as conditional means based on these information sets. They also derive the mean squared errors of these predictors, provide some efficiency rankings based on these mean squared errors, and provide Monte Carlo evidence that the additional efficiencies obtained by using more information can be practically important.

One of the predictors that Kelejian and Prucha (2007) consider is based on the information set $\{\mathbf{X}, \mathbf{W}, \mathbf{M}, \mathbf{w}_i\mathbf{y}\}$, where $\mathbf{w}_i$ denotes the ith row of $\mathbf{W}$, which will be referred to as the limited-information predictor.[4] We denote the limited-information predictor by `limited` in the syntax diagram below. Another estimator that Kelejian and Prucha (2007) consider is based on the information set $\{\mathbf{X}, \mathbf{W}, \mathbf{M}\}$, which yields the reduced-form predictor. This predictor is denoted by `rform` in the syntax diagram below.

4. Kelejian and Prucha (2007) also consider a full-information predictor. We have postponed implementing this predictor because it is computationally more demanding; we plan to implement it in future work.

Kelejian and Prucha (2007) show that their limited-information predictor can be much more efficient than the reduced-form predictor.

In addition to the limited-information predictor and the reduced-form predictor, `predict` can compute two other observation-level quantities, which are not recommended as predictors but may be used in subsequent computations. These quantities are denoted by `naive` and `xb` in the syntax diagram below.

While prediction is frequently of interest in applied statistical work, predictions can also be used to compute marginal effects.[5] A change to one observation in one exogenous variable potentially changes the predicted values for all the observations of the dependent variable because the n observations for the dependent variable form a system of simultaneous equations in a SARAR model. Below we use `predict` to calculate predictions that we in turn use to calculate marginal effects.

Various methods have been proposed to interpret the parameters of SAR models: see, for example, Anselin (2003); Abreu, De Groot, and Florax (2004); Kelejian and Prucha (2007); and LeSage and Pace (2009).

5.1 Syntax

Before using `predict`, we discuss its syntax.

`predict` [*type*] *newvar* [*if*] [*in*] [`,` `rform`|`limited`|`naive`|`xb`
`rftransform(`*matname*`)`]

5.2 Options

`rform`, the default, calculates the reduced-form predictions.

`limited` calculates the Kelejian and Prucha (2007) limited-information predictor. This predictor is more efficient than the reduced-form predictor, but we call it `limited` because it is not as efficient as the Kelejian and Prucha (2007) full-information predictor, which we plan to implement in the future.

`naive` calculates $\widehat{\lambda}\mathbf{w}_i\mathbf{y} + \mathbf{x}_i\widehat{\boldsymbol{\beta}}$ for each observation.

`xb` calculates the linear prediction $\mathbf{X}\widehat{\boldsymbol{\beta}}$.

5. We refer to the effects of both infinitesimal changes in a continuous variable and discrete changes in a discrete variable as marginal effects. While some authors refer to "partial" effects to cover the continuous and discrete cases, we avoid the term "partial" because it means something else in a simultaneous-equations framework.

`rftransform(`*matname*`)` is a seldom-used option that specifies a matrix to use in computing the reduced-form predictions. This option is only useful when computing reduced-form predictions in a loop, when the option removes the need to recompute the inverse of a large matrix. See section 5.3 for an example that uses this option, and see section 6.3 for the details. `rftransform()` may only be specified with statistic `rform`.

5.3 Example

In this section, we discuss two marginal effects that measure how changes in the exogenous variables affect the endogenous variable. These measures use the reduced-form predictor $\widehat{\mathbf{y}} = E(\mathbf{y}|\mathbf{X},\mathbf{W},\mathbf{M}) = (\mathbf{I}-\lambda\mathbf{W})^{-1}\mathbf{X}\boldsymbol{\beta}$, which we discuss in section 6.3, where it is denoted as $\widehat{\mathbf{y}}^{(1)}$. The expression for the predictor shows that a change in a single observation on an exogenous variable will typically affect the values of the endogenous variable for all n units because the SARAR model forms a system of simultaneous equations.

Without loss of generality, we explore the effects of changes in the kth exogenous variable. Letting $\mathbf{x}_k = (x_{1k},\ldots,x_{nk})'$ denote the vector of observations on the kth exogenous variable allows us to denote the dependence of $\widehat{\mathbf{y}}$ on $\mathbf{x}_k$ by using the notation

$$\widehat{\mathbf{y}}(\mathbf{x}_k) = \{\widehat{y}_1(\mathbf{x}_k),\ldots,\widehat{y}_n(\mathbf{x}_k)\}$$

The first marginal effect we consider is

$$\frac{\partial\widehat{\mathbf{y}}(\mathbf{x}_k+\delta\mathbf{i})}{\partial\delta} = \frac{\partial\widehat{\mathbf{y}}(x_{1k},\ldots,x_{i-1,k},x_{ik}+\delta,x_{i+1,k},\ldots,x_{nk})}{\partial\delta} = \frac{\partial\widehat{\mathbf{y}}(\mathbf{x}_k)}{\partial x_{ik}}$$

where $\mathbf{i} = [0,\ldots,0,1,0,\ldots,0]'$ is the ith column of the identity matrix. In terminology consistent with that of LeSage and Pace (2009, 36–37), we refer to the above effect as the total direct impact of a change in the ith unit of $\mathbf{x}_k$. LeSage and Pace (2009, 36–37) define the corresponding summary measure

$$\begin{aligned} n^{-1}\sum_{i=1}^{n}\frac{\partial\widehat{y}_i(\mathbf{x}_k+\delta\mathbf{i})}{\partial\delta} &= n^{-1}\sum_{i=1}^{n}\frac{\partial\widehat{y}_i(x_{1k},\ldots,x_{i-1,k},x_{ik}+\delta,x_{i+1,k},\ldots,x_{nk})}{\partial\delta} \\ &= n^{-1}\sum_{i=1}^{n}\frac{\partial\widehat{y}_i(\mathbf{x}_k)}{\partial x_{ik}} \end{aligned} \tag{3}$$

which they call the average total direct impact (ATDI). The ATDI is the average over $i=\{1,\ldots,n\}$ of the changes in the $\widehat{y}_i$ attributable to the changes in the corresponding x_{ik}. The ATDI can be calculated by computing $\widehat{\mathbf{y}}(\mathbf{x}_k)$, $\widehat{\mathbf{y}}(\mathbf{x}_k+\delta\mathbf{i})$, and the average of the difference of these vectors of predicted values, where δ is the magnitude by which x_{ik} is changed. The ATDI measures the average change in $\widehat{y}_i$ attributable to sequentially changing x_{ik} for a given k.

Sequentially changing x_{ik} for each $i = \{1, \ldots, n\}$ differs from simultaneously changing the x_{ik} for all n units. The second marginal effect we consider measures the effect of simultaneously changing $x_{1k}, \ldots, x_{nk}$ on a specific $\widehat{y}_i$ and is defined by

$$\frac{\partial \widehat{y}_i(\mathbf{x}_k + \delta\mathbf{e})}{\partial \delta} = \frac{\partial \widehat{y}_i(x_{1k} + \delta, \ldots, x_{ik} + \delta, \ldots, x_{nk} + \delta)}{\partial \delta} = \sum_{r=1}^{n} \frac{\partial \widehat{y}_i(\mathbf{x}_k)}{\partial x_{rk}}$$

where $\mathbf{e} = [1, \ldots, 1]'$ is a vector of 1s. LeSage and Pace (2009, 36–37) define the corresponding summary measure

$$\begin{aligned} n^{-1} \sum_{i=1}^{n} \frac{\partial \widehat{y}_i(\mathbf{x}_k + \delta\mathbf{e})}{\partial \delta} &= n^{-1} \sum_{i=1}^{n} \frac{\partial \widehat{y}_i(x_{1k} + \delta, \ldots, x_{ik} + \delta, \ldots, x_{nk} + \delta)}{\partial \delta} \\ &= n^{-1} \sum_{i=1}^{n} \sum_{r=1}^{n} \frac{\partial \widehat{y}_i(\mathbf{x}_k)}{\partial x_{rk}} \end{aligned} \tag{4}$$

which they call the average total impact (ATI). The ATI can be calculated by computing $\widehat{\mathbf{y}}(\mathbf{x}_k)$, $\widehat{\mathbf{y}}(\mathbf{x}_k + \delta\mathbf{e})$, and the average difference in these vectors of predicted values, where δ is the magnitude by which $x_{1k}, \ldots, x_{nk}$ is changed.

We now continue our example from section 4 and use the reduced-form predictor to compute the marginal effects of adding one officer per 100,000 DVMT in Elko County, Nevada. We begin by using the reduced-form predictor and the observed values of the exogenous variables to obtain predicted values for `dui`:

```
. predict y0
(option rform assumed)
```

Next we increase `police` by 1 in Elko County, Nevada, and calculate the reduced-form predictions:

```
. generate police_orig = police
. quietly replace police = police_orig + 1 if st==32 & NAME00=="Elko"
. predict y1
(option rform assumed)
```

Now we compute the difference between these two predictions:

```
. generate deltay = y1-y0
```

The output below lists the predicted difference and the level of `dui` for Elko County, Nevada:

```
. list deltay dui if (st==32 & NAME00=="Elko")
```

	deltay	dui
1891.	-.5654716	19.777429

The predicted effect of the change would be a 2.9% reduction in `dui` in Elko County, Nevada.

Below we use four commands to summarize the changes and levels in the contiguous counties:

```
. spmat getmatrix ccounty W
. generate double elko_neighbor = .
(3109 missing values generated)
. mata: st_store(.,"elko_neighbor",W[1891,.]')
. summarize deltay dui if elko_neighbor>0
```

Variable	Obs	Mean	Std. Dev.	Min	Max
deltay	9	-.0203756	.0000364	-.0204239	-.020298
dui	9	21.29122	1.6468	19.2773	23.49109

In the first command, we use `spmat getmatrix` to store a copy of the normalized-contiguity spatial-weighting matrix in Mata memory; see Drukker et al. (2013, sec. 14) for a discussion of `spmat getmatrix`. In the second and third commands, we generate and fill in a new variable for which the ith observation is 1 if it contains information on a county that is contiguous with Elko County and is 0 otherwise. In the fourth command, we summarize the predicted changes and the levels in the contiguous counties. The mean predicted reduction is less than 0.1% of the mean level of `dui` in the contiguous counties.

In the output below, we get a summary of the levels of `dui` and a detailed summary of the predicted changes for all the counties in the sample.

```
. summarize dui
```

Variable	Obs	Mean	Std. Dev.	Min	Max
dui	3109	20.84307	1.457163	15.01375	26.61978

```
. summarize deltay, detail
```

deltay

Percentiles		Smallest		
1%	-.0007572	-.5654716		
5%	0	-.0204239		
10%	0	-.0203991	Obs	3109
25%	0	-.0203991	Sum of Wgt.	3109
50%	0		Mean	-.0002495
		Largest	Std. Dev.	.0101996
75%	0	0		
90%	0	0	Variance	.000104
95%	0	0	Skewness	-54.78661
99%	0	0	Kurtosis	3035.363

Less than 1% of the sample had any socially significant difference, with no change at all predicted for at least 95% of the sample.

In some of the computations below, we will use the matrix $\mathbf{S} = (\mathbf{I}_n - \widehat{\lambda}\mathbf{W})^{-1}$, where $\widehat{\lambda}$ is the estimate of the SAR parameter and $\mathbf{W}$ is the spatial-weighting matrix. In the output below, we use the $\mathbf{W}$ stored in Mata memory in an example above to compute $\mathbf{S}$.

```
. spmat getmatrix ccounty W
. mata:
------------------------------------------------- mata (type end to exit) -----
:     b = st_matrix("e(b)")
:     lam = b[1,6]
:     S = luinv(I(rows(W))-lam*W)
:     (b[1,1]/rows(W))*sum(S)
  -.6993674779
: end
--------------------------------------------------------------------------------
```

We next compute the ATDI defined in (3). The output below shows an instructive (but slow) method to compute the ATDI. For each county in the data, we set `police` to be the original value for all the observations except the ith, which we set to `police + 1`. Then we compute the predicted value of `dui` for observation i and store this prediction in the ith observation of `y1`. (We use the `rftransform()` option to use the inverse matrix $\mathbf{S}$ computed above. Without this option, we would recompute the inverse matrix for each of the 3,109 observations, which would cause the calculation to take hours.) After computing the predicted values of `y1` for each observation, we compute the differences in the predictions and compute the sample average.

```
. drop y1 deltay
. generate y1 = .
(3109 missing values generated)
. local N = _N
. forvalues i = 1/`N´ {
  2.         quietly capture drop tmp
  3.         quietly replace police = police_orig
  4.         quietly replace police = police_orig + 1 in `i´
  5.         quietly predict tmp in `i´, rftransform(S)
  6.         quietly replace y1 = tmp in `i´
  7. }
. generate deltay = y1-y0
. summarize deltay
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      deltay |      3109   -.5633844    .0009144  -.5690784  -.5599785
. summarize dui
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         dui |      3109    20.84307    1.457163   15.01375   26.61978
```

The absolute value of the estimated ATDI is -0.56, so the estimated effect is 2.7% of the sample mean of `dui`.

As mentioned, the above method for computing the estimate of the ATDI is slow. LeSage and Pace (2009, 36–37) show that the estimate of the ATDI can also be computed as

$$\frac{\boldsymbol{\beta}_k}{n}\text{trace}(\mathbf{S})$$

where $\boldsymbol{\beta}_k$ is the kth component of $\boldsymbol{\beta}$ and $\mathbf{S} = (\mathbf{I}_n - \lambda\mathbf{W})^{-1}$, which we computed above. Below we use this formula to compute the ATDI,

```
. mata: (b[1,1]/rows(W))*trace(S)
  -.5633844076
```

and note that the result is the same as above.

Now we estimate the ATI, which simultaneously adds one more police officer per 100,000 residents to each county. In the output below, we add 1 to `police` in each observation and then calculate the differences in the predictions. We then calculate the ATI defined in (4) by computing the sample average.

```
. drop y1 deltay
. quietly replace police = police_orig + 1
. predict y1
(option rform assumed)
. generate deltay = y1-y0
. summarize deltay
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      deltay |      3109   -.6993675    .0309541  -.8945923  -.5801525
. summarize dui
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         dui |      3109    20.84307    1.457163   15.01375   26.61978
```

The absolute value of the estimated average total effect is about 3.4% of the sample mean of `dui`.

LeSage and Pace (2009, 36–37) show that the ATI is given by

$$\frac{\boldsymbol{\beta}_k}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}S_{i,j}$$

where $\boldsymbol{\beta}_k$ is the kth component of $\boldsymbol{\beta}$ and S_{ij} is the (i,j)th element of $\mathbf{S} = (\mathbf{I}_n - \lambda\mathbf{W})^{-1}$. In the output below, we use the `spmat getmatrix` command discussed in Drukker et al. (2013) and a few Mata computations to show that the above expression yields the same value for the ATI as our calculations above.

```
. spmat getmatrix ccounty W
. mata:
------------------------------------------------- mata (type end to exit) -----
:     b = st_matrix("e(b)")
:     lam = b[1,6]
:     S = luinv(I(rows(W))-lam*W)
:     (b[1,1]/rows(W))*sum(S)
  -.6993674779
: end
-------------------------------------------------------------------------------
```

In general, it is not possible to say whether the ATDI is greater than or less than the ATI. Using the expressions from LeSage and Pace (2009, 36–37), we see that

$$\text{ATI} - \text{ATDI} = \frac{\boldsymbol{\beta}_k}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} S_{i,j} - \frac{\boldsymbol{\beta}_k}{n}\sum_{i=1}^{n} S_{i,i} = \frac{\boldsymbol{\beta}_k}{n}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} S_{i,j}$$

which depends on the sum of the off-diagonal elements of $\mathbf{S}$ as well as on $\boldsymbol{\beta}_k$.

In the case at hand, one would expect the ATDI to be smaller than the ATI because the ATDI, unlike the ATI, does not incorporate the reinforcing effects of having all counties implement the change simultaneously.

6 Methods and formulas

6.1 ML estimator

Recall that the SARAR model under consideration is given by

$$\mathbf{y} = \lambda\mathbf{W}\mathbf{y} + \mathbf{X}\beta + \mathbf{u} \tag{5}$$
$$\mathbf{u} = \rho\mathbf{M}\mathbf{u} + \boldsymbol{\epsilon} \tag{6}$$

In the following, we give the log-likelihood function under the assumption that $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I})$. As usual, we refer to the maximizer of the likelihood function when the innovations are not normally distributed as the quasi-maximum likelihood (QML) estimator. Lee (2004) gives results concerning the consistency and asymptotic normality of the QML estimator when $\boldsymbol{\epsilon}$ is IID but not necessarily normally distributed. Violations of the assumption that the innovations $\boldsymbol{\epsilon}$ are IID can cause the QML estimator to produce inconsistent results. In particular, this may be the case if the innovations $\boldsymbol{\epsilon}$ are heteroskedastic, as discussed by Arraiz et al. (2010).

Likelihood function

The reduced form of the model in (5) and (6) is given by

$$\mathbf{y} = (\mathbf{I} - \lambda\mathbf{W})^{-1}\mathbf{X}\boldsymbol{\beta} + (\mathbf{I} - \lambda\mathbf{W})^{-1}(\mathbf{I} - \rho\mathbf{M})^{-1}\boldsymbol{\epsilon}$$

The unconcentrated log-likelihood function is

$$\ln L(y|\boldsymbol{\beta},\sigma^2,\lambda,\rho) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) + \ln||\mathbf{I}-\lambda\mathbf{W}|| + \ln||\mathbf{I}-\rho\mathbf{M}|| \\ -\frac{1}{2\sigma^2}\left\{(\mathbf{I}-\lambda\mathbf{W})\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\}^T(\mathbf{I}-\rho\mathbf{M})^T(\mathbf{I}-\rho\mathbf{M})\left\{(\mathbf{I}-\lambda\mathbf{W})\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\} \quad (7)$$

We can concentrate the log-likelihood function by first maximizing (7) with respect to $\boldsymbol{\beta}$ and σ^2, yielding the maximizers

$$\widehat{\boldsymbol{\beta}}(\lambda,\rho) = \left\{\mathbf{X}^T(\mathbf{I}-\rho\mathbf{M})^T(\mathbf{I}-\rho\mathbf{M})\mathbf{X}\right\}^{-1}\mathbf{X}^T(\mathbf{I}-\rho\mathbf{M})^T(\mathbf{I}-\rho\mathbf{M})(\mathbf{I}-\lambda\mathbf{W})\mathbf{y}$$
$$\widehat{\sigma}^2(\lambda,\rho) = (1/n)\left\{(\mathbf{I}-\lambda\mathbf{W})\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}}(\lambda,\rho)\right\}^T(\mathbf{I}-\rho\mathbf{M})^T(\mathbf{I}-\rho\mathbf{M}) \\ \left\{(\mathbf{I}-\lambda\mathbf{W})\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}}(\lambda,\rho)\right\}$$

Substitution of the above expressions into (7) yields the concentrated log-likelihood function

$$L_c(y|\lambda,\rho) \quad = \quad -\frac{n}{2}\left\{\ln(2\pi)+1\right\} - \frac{n}{2}\ln(\widehat{\sigma}^2(\lambda,\rho)) + \ln||\mathbf{I}-\lambda\mathbf{W}|| + \ln||\mathbf{I}-\rho\mathbf{M}||$$

The QML estimates for the autoregressive parameters $\widehat{\lambda}$ and $\widehat{\rho}$ can now be computed by maximizing the concentrated log-likelihood function. Once we have obtained the QML estimates $\widehat{\lambda}$ and $\widehat{\rho}$, we can calculate the QML estimates for β and σ^2 as $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}(\widehat{\lambda},\widehat{\rho})$ and $\widehat{\sigma}^2 = \widehat{\sigma}^2(\widehat{\lambda},\widehat{\rho})$.

Initial values

As noted in Anselin (1988, 186), poor initial starting values for ρ and λ in the concentrated likelihood may result in the optimization algorithm settling on a local, rather than the global, maximum.

To prevent this problem from happening, `spreg ml` performs a grid search to find suitable initial values for ρ and λ. To override the grid search, you may specify your own initial values in the option `from()`.

6.2 GS2SLS estimator

For discussions of the generalized method of moments and instrumental-variable estimation approach underlying the GS2SLS estimator, see Arraiz et al. (2010) and Drukker, Egger, and Prucha (2013). The articles build on Kelejian and Prucha (1998, 1999, 2010) and the references cited therein. For a detailed description of the formulas, see also Drukker, Prucha, and Raciborski (2013).

The GS2SLS estimator requires instruments. Kelejian and Prucha (1998, 1999) suggest using as instruments $\mathbf{H}$ the linearly independent columns of

$$\mathbf{X}, \mathbf{WX}, \ldots, \mathbf{W}^q\mathbf{X}, \mathbf{MX}, \mathbf{MWX}, \ldots, \mathbf{MW}^q\mathbf{X}$$

where $q = 2$ has worked well in Monte Carlo simulations over a wide range of reasonable specifications. The choice of those instruments provides a computationally convenient approximation of the ideal instruments; see Lee (2003) and Kelejian, Prucha, and Yuzefovich (2004) for further discussions and refined estimators. At a minimum, the instruments should include the linearly independent columns of $\mathbf{X}$ and $\mathbf{MX}$. When there is a constant in the model and thus $\mathbf{X}$ contains a constant term, the constant term is only included once in $\mathbf{H}$.

6.3 Spatial predictors

The `spreg` command provides for several unbiased predictors corresponding to different information sets, namely, $\{\mathbf{X}, \mathbf{W}, \mathbf{M}\}$ and $\{\mathbf{X}, \mathbf{W}, \mathbf{M}, \mathbf{w}_i\mathbf{y}\}$, where $\mathbf{w}_i$ denotes the ith row of $\mathbf{W}$; for a more detailed discussion and derivations, see Kelejian and Prucha (2007). Also in the following, $\mathbf{x}_i$ denotes the ith row of $\mathbf{X}$ and $\mathbf{u}_i$ denotes the ith element of $\mathbf{u}$.

The unbiased predictor corresponding to information set $\{\mathbf{X}, \mathbf{W}, \mathbf{M}\}$ is given by

$$\widehat{\mathbf{y}}^{(1)} = (\mathbf{I} - \lambda\mathbf{W})^{-1}\mathbf{X}\boldsymbol{\beta}$$

and is called the reduced-form predictor. If $\lambda = 0$, then $\widehat{\mathbf{y}}^{(1)} = \mathbf{X}\boldsymbol{\beta}$. This predictor can be calculated by specifying statistic `rform` to `predict` after `spreg`.

When specified, the `rftransform()` option specifies the name of a matrix in Mata memory that contains $(I - \lambda\mathbf{W})^{-1}$. The `rftransform()` option specifies a matrix that transforms the model to its reduced form. This option is useful when computing many sets of reduced-form predictions from the same $(I - \lambda\mathbf{W})^{-1}$ because it alleviates the need to recompute the inverse matrix.

Assuming that the innovations $\boldsymbol{\epsilon}$ are distributed $N(0, \sigma^2\mathbf{I})$, the unbiased predictor corresponding to information set $\{\mathbf{X}, \mathbf{W}, \mathbf{M}, \mathbf{w_i y}\}$ is given by

$$\widehat{\mathbf{y}}_i^{(2)} = \lambda\mathbf{w}_i\mathbf{y} + \mathbf{x}_i\boldsymbol{\beta} + \frac{\mathrm{cov}(\mathbf{u}_i, \mathbf{w}_i\mathbf{y})}{\mathrm{var}(\mathbf{w}_i\mathbf{y})}\{\mathbf{w}_i\mathbf{y} - E(\mathbf{w}_i\mathbf{y})\}$$

where

$$\begin{aligned}
\boldsymbol{\Sigma}^u &= (\mathbf{I} - \rho\mathbf{M})^{-1}(\mathbf{I} - \rho\mathbf{M}^T)^{-1} \\
\boldsymbol{\Sigma}^y &= (\mathbf{I} - \lambda\mathbf{W})^{-1}\boldsymbol{\Sigma}^u(\mathbf{I} - \lambda\mathbf{W}^T)^{-1} \\
E(\mathbf{w}_i\mathbf{y}) &= \mathbf{w}_i(\mathbf{I} - \lambda\mathbf{W})^{-1}\mathbf{X}\boldsymbol{\beta} \\
\mathrm{var}(\mathbf{w}_i\mathbf{y}) &= \sigma^2\mathbf{w}_i\boldsymbol{\Sigma}^y\mathbf{w}_i' \\
\mathrm{cov}(\mathbf{u}_i, \mathbf{w}_i\mathbf{y}) &= \sigma^2\sigma_i^u(\mathbf{I} - \lambda\mathbf{W}^T)^{-1}\mathbf{w}_i' \\
\sigma_i^u &\quad \text{is the } i\text{th row of } \boldsymbol{\Sigma}^u
\end{aligned}$$

We call this unbiased predictor the limited-information predictor because Kelejian and Prucha (2007) consider a more efficient predictor, the full-information predictor. The former can be calculated by specifying statistic `limited` to `predict` after `spreg`.

A further predictor considered in the literature is

$$\widehat{\mathbf{y}}_i = \lambda \mathbf{w}_i \mathbf{y} + \mathbf{x}_i \boldsymbol{\beta}$$

However, as pointed out in Kelejian and Prucha (2007), this estimator is generally biased. While this biased predictor should not be used for predictions, it has uses as an intermediate computation, and it can be calculated by specifying statistic `naive` to `predict` after `spreg`.

The above predictors are computed by replacing the parameters in the prediction formula with their estimates.

7 Conclusion

After reviewing some basic concepts related to SARAR models, we presented the `spreg ml` and `spreg gs2sls` commands, which implement ML and GS2SLS estimators for the parameters of these models. We also discussed postestimation prediction. In future work, we would like to investigate further methods and commands for parameter interpretation.

8 Acknowledgment

We gratefully acknowledge financial support from the National Institutes of Health through the SBIR grants R43 AG027622 and R44 AG027622.

9 References

Abreu, M., H. L. F. De Groot, and R. J. G. M. Florax. 2004. Space and growth: A survey of empirical evidence and methods. Working Paper TI 04-129/3, Tinbergen Institute.

Anselin, L. 1988. *Spatial Econometrics: Methods and Models*. Dordrecht: Kluwer Academic Publishers.

———. 2003. Spatial externalities, spatial multipliers, and spatial econometrics. *International Regional Science Review* 26: 153–166.

———. 2010. Thirty years of spatial econometrics. *Papers in Regional Science* 89: 3–25.

Anselin, L., and R. J. G. M. Florax. 1995. Small sample properties of tests for spatial dependence in regression models: Some further results. In *New Directions in Spatial Econometrics*, ed. L. Anselin and R. J. G. M. Florax, 21–74. Berlin: Springer.

Arbia, G. 2006. *Spatial Econometrics: Statistical Foundations and Applications to Regional Convergence*. Berlin: Springer.

Arraiz, I., D. M. Drukker, H. H. Kelejian, and I. R. Prucha. 2010. A spatial Cliff-Ord-type model with heteroskedastic innovations: Small and large sample results. *Journal of Regional Science* 50: 592–614.

Cliff, A. D., and J. K. Ord. 1973. *Spatial Autocorrelation*. London: Pion.

———. 1981. *Spatial Processes: Models and Applications*. London: Pion.

Cressie, N. A. C. 1993. *Statistics for Spatial Data*. Revised ed. New York: Wiley.

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

Drukker, D. M., H. Peng, I. R. Prucha, and R. Raciborski. 2013. Creating and managing spatial-weighting matrices with the spmat command. *Stata Journal* 13: 242–286.

Drukker, D. M., I. R. Prucha, and R. Raciborski. 2013. A command for estimating spatial-autoregressive models with spatial-autoregressive disturbances and additional endogenous variables. *Stata Journal* 13: 287–301.

Haining, R. 2003. *Spatial Data Analysis: Theory and Practice*. Cambridge: Cambridge University Press.

Kelejian, H. H., and I. R. Prucha. 1998. A generalized spatial two-stage least squares procedure for estimating a spatial autoregressive model with autoregressive disturbances. *Journal of Real Estate Finance and Economics* 17: 99–121.

———. 1999. A generalized moments estimator for the autoregressive parameter in a spatial model. *International Economic Review* 40: 509–533.

———. 2007. The relative efficiencies of various predictors in spatial econometric models containing spatial lags. *Regional Science and Urban Economics* 37: 363–374.

———. 2010. Specification and estimation of spatial autoregressive models with autoregressive and heteroskedastic disturbances. *Journal of Econometrics* 157: 53–67.

Kelejian, H. H., I. R. Prucha, and Y. Yuzefovich. 2004. Instrumental variable estimation of a spatial autoregressive model with autoregressive disturbances: Large and small sample results. In *Spatial and Spatiotemporal Econometrics*, ed. J. P. LeSage and R. K. Pace, 163–198. New York: Elsevier.

Lee, L.-F. 2003. Best spatial two-stage least squares estimators for a spatial autoregressive model with autoregressive disturbances. *Econometric Reviews* 22: 307–335.

———. 2004. Asymptotic distributions of quasi-maximum likelihood estimators for spatial autoregressive models. *Econometrica* 72: 1899–1925.

LeSage, J., and R. K. Pace. 2009. *Introduction to Spatial Econometrics*. Boca Raton: Chapman & Hall/CRC.

Powers, E. L., and J. K. Wilson. 2004. Access denied: The relationship between alcohol prohibition and driving under the influence. *Sociological Inquiry* 74: 318–337.

Whittle, P. 1954. On stationary processes in the plane. *Biometrika* 41: 434–449.

About the authors

David Drukker is the director of econometrics at StataCorp.

Ingmar Prucha is a professor of economics at the University of Maryland.

Rafal Raciborski is an econometrician at StataCorp.