Technical Aspects of Implementing GMM Estimation of the Spatial Error Model in PySAL and GeoDaSpace*

Luc Anselin† Pedro V. Amaral‡ Daniel Arribas-Bel§

December 4, 2012

1 Introduction

This paper illustrates some technical aspects of the implementation of general methods of moments (GMM) estimation of the spatial error model in GeoDaSpace and in the `spreg` module of PySAL (Rey and Anselin 2007, Anselin and Rey 2012). The principles of this approach were originally presented in Kelejian and Prucha (1998, 1999), and more recently generalized in a series of papers by Kelejian and Prucha (2010), Arraiz et al. (2010) and Drukker et al. (2012) (jointly referred to in what follows as K-P-D) . A similar but slightly different theoretical framework is outlined in Lee (2007) and Lin and Lee (2010). A detailed discussion of the methodological aspects of implementing GMM in this context is also offered in Anselin (2011), which is a companion to the current document.

GeoDaSpace is intended for the user who is somewhat familiar with the methods, but prefers a point and click environment to the command line. As

*The `spreg` PySAL module and GeoDaSpace were developed by a sub-team of PySAL developers under the direction of Luc Anselin at the GeoDa Center for Geospatial Analysis and Computation at Arizona State University. The core team consisted of Luc Anselin, Sergio Rey, David Folch, Daniel Arribas-Bel, and Pedro Amaral, with contributions by Charles Schmidt, Nicholas Malizia, Ran Wei, Jing Yao, Phil Stephens, Myunghwa Hwang, Mark McCann and Julia Koschinsky.The development of the PySAL `spreg` module and associated GeoDaSpace graphical user interface and code was supported in part by Award No. 2009-SQ-B9-K101 from the National Institute of Justice, Office of Justice Programs, U.S. Department of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect those of the Department of Justice.

†GeoDa Center for Geospatial Analysis and Computation, School of Geographical Sciences and Urban Planning, Arizona State University, Tempe, AZ luc.anselin@asu.edu

‡Department of Land Economics, Cambridge University, Cambridge, U.K., and GeoDa Center for Geospatial Analysis and Computation, Arizona State University, Tempe, AZ pvmda2@cam.ac.uk

§Department of Spatial Economics, Faculty of Economics and Business Administration, Vrije Universiteit Amsterdam darribas@feweb.vu.nl

a consequence, the number of options available in GeoDaSpace has been deliberately constrained to those that are most common. In some cases, this means that the default results produced by GeoDaSpace may differ slightly from those produced by other computing environments, specifically the spatial econometrics routines in Stata (Drukker et al. 2011) and the `sphet` module in `R` (Piras 2010). We address these differences in some detail. Most importantly, we illustrate the flexibility and comprehensiveness of the options included in the PySAL `spreg` module to implement different approaches that have been suggested in the literature (some of these options are not available through the GeoDaSpace GUI).

The description in this document pertains to GeoDaSpace alpha release version 0.7.7, dated November 29, 2012. The code in GeoDaSpace is based on the `spreg` module in PySAL which was originally released under Version 1.3 (January 2012). Functionality was extended in PySAL Version 1.4 (August 2012) and in the current development version available from the google code repository. An extensive guide to the use of the software and the various options that can be specified (for both GeoDaSpace and PySAL) is provided in Anselin and Rey (2013). We refer to that document for specific details.

The code base of the `spreg` module was completely refactored from earlier working versions and rewritten to take advantage of sparse matrix routines and other matrix algebra algorithms contained in the Python `scipy` and `numpy` modules. While these are pre-requisites for the installation of the PySAL library, the necessary routines are pre-compiled and included in the GeoDaSpace binaries, which are completely self-contained.[1] Both GeoDaSpace and PySAL continue to be under active development.

We also consider the Stata `spivreg` commands (as of Stata Version 11) and the `sphet` package in `R` (version 1.1-12, published on CRAN on 2012-04-13). With respect to the `R` package `sphet`, we consider both the current official release as well as a development release that contains considerable more functionality (specifically, an alpha release on R-Forge, revision 57, published on 2012-10-30). These two versions are referred to in what follows as `sphet1` and `sphet2`, respectively.[2]

A detailed comparison of what is implemented in each of these packages is shown in Table 1.

Given that the GMM framework is very general, several choices can be made in actual implementations that all achieve consistency as an asymptotic result,

[1] GeoDaSpace is still in alpha and bug reports and other comments are actively requested from users. Binaries of the program for Mac OSX and Windows can be downloaded from `https://geodacenter.asu.edu/software/downloads/geodaspace`. Installation instructions and the source code for PySAL (released under the open source BSD license) can be found at `http://pysal.org`.

[2] It should be noted that `sphet2` in particular is not an official release and may still undergo changes that could affect the results of our comparisons. In earlier versions of our paper, we used revision 56, published on 2012-07-22. There are significant changes between revision 57 and 56, particularly reflected in the results for models with endogenous variables, given in Tables 3 and 6. The results for `sphet2` revision 56 did not match the results for GeoDaSpace and Stata. The results obtained with revision 57 do, as reflected in the Tables.

Table 1: Comparison of Functionality: GeoDaSpace, Stata and R

Method	**G**	**Stata**	**sph1**[1]	**sph2**[2]
Standard (OLS)	●	●	●	●
heteroskedasticity (White)[3]	●	●	●	●
s.a. and heteroskedasticity (HAC)[4]	●		●	●
Standard (2SLS)	●	●	●	●
endogenous var and het. (White)[3]	●	●	●	●
endogenous var., s.a. and het. (HAC)[4]	●		●	●
Spatial lag (S2SLS)[5]	●	●	●	●
het. (White)[3]	●	●	●	●
s.a. and het. (HAC)[4]	●		●	●
endogenous var.	●	●		●
endog. var. and het. (White)[3]	●	●		●
endog. var., s.a. and het. (HAC)[4]	●			●
Spatial error (GM) (KP98/99)[6]	●	●	●	●
endogenous var.	●	●		
Spatial error and lag (GM) (KP98/99)[6]	●	●	●	●
endog. var.	●	●		
Spatial error (GMM)(KPD)[7]	●	●		●
endogenous var.	●	●		●
Spatial error and lag (GMM) (KPD)[7]	●	●		●
endog. var.	●	●		●
Spatial error with het. (GMM) (KP-Het)[8]	●	●	●	●
endog. var. and het.	●	●		●
Spatial error and lag with het. (GMM) (KP-Het)[8]	●	●	●	●
endog. var. and het.	●	●		●

[1] R packages spdep and sphet (v. 1.1-12, published on CRAN on 2012-04-13).

[2] R packages spdep and sphet (revision 57, published on R-Forge on 2012-10-30).

[3] based on White (1980)

[4] based on Kelejian and Prucha (2007)

[5] based on Anselin (1988)

[6] based on Kelejian and Prucha (1998, 1999)

[7] based on Drukker et al. (2012)

[8] based on Arraiz et al. (2010)

but yield different estimates (and/or estimated standard errors) in actual applications. To highlight potential differences that may result from such choices, we compare the results among the different packages in detail, using a common data set and model specification.

In the remainder of this paper, we first present the specification of the spatial error model and outline the main steps in the estimation methods. This is followed by a detailed discussion of six specific cases:

- spatial error model without heteroskedasticity
  - exogenous variables only
  - exogenous and endogenous variables

  - combo model with spatial lag and spatial error

- spatial error model with heteroskedasticity
  - exogenous variables only
  - exogenous and endogenous variables
  - combo model with spatial lag and spatial error

We close with some concluding remarks.

1.1 Empirical Illustration

The empirical illustrations are based on the `NAT` sample data set available from the GeoDa Center web site. These data can be downloaded from `http://geodacenter.org/downloads/data-files`. Alternatively, they are contained in the `examples` directory of the PySAL installation.

The `NAT` data set contains observations for 3085 continental U.S. counties (as polygons) on homicides and a number of potential socio-economic determinants. Details on the data and the context, as well as published empirical results can be found in Messner et al. (2000), Baller et al. (2001) and Messner and Anselin (2004).

The specific model specification included here is a regression of county homicide rates for 1990 (`HR90`) on a constant, the resource deprivation index for 1990 (`RD90`) and the unemployment rate for 1990 (`UE90`). The unemployment rate is considered as potentially endogenous and instrumented by the percent families below the poverty line in 1989 (`FP89`). The spatial weights matrix is based on queen contiguity between the counties, as contained in the `NAT_queen.gal` file in the sample data set (or the `examples` directory of the PySAL installation).

1.2 Replication

In order to facilitate the replication of our results (see, e.g., Koenker and Zeileis 2009), we include four files as digital appendices. These files contain the commands to reproduce the results presented here using PySAL (GeoDaSpace), R `sphet` and Stata `spivreg`.

The file `GMM_comparison.ipynb` is an iPython "notebook" that contains the Python commands to obtain the results in PySAL presented in the tables that follow. The assumption (and also for the other appendices) is that the data are in the current working directory. If not, the proper pathname needs to be specified.

The R code to replicate the results for `sphet` is contained in two files. The first, `NAT_R_comp_paper_edit_md.Rmd` is an R "Markdown" file. It is easiest to open in `Rstudio` with the `Sweave` option set to `knitr`. Alternatively, the second file, `NAT_comp_paper.ipynb` is an iPython notebook that invokes the `Rmagic` command to run R commands from within a Python environment.

Finally, the Stata commands to replicate the results are included in the `NAT_comp_Stata.do` "ado" file, which can be readily executed from within Stata.

2 Model and Methods

2.1 Model Specification

The most general model we consider is the so-called mixed regressive spatial autoregressive model with a spatial autoregressive error term, or the so-called SAR-SAR model.[3] In what follows, we also refer to this specification as the *combo* model.

Using the notation from Anselin (1988), the spatial lag part of the model is expressed as:

$$\mathbf{y} = \rho\mathbf{W}\mathbf{y} + \mathbf{X}\beta + \mathbf{u}. \tag{1}$$

The notation is standard, with $\mathbf{y}$ as a $n \times 1$ vector of observations on the dependent variable, $\mathbf{W}$ as a $n \times n$ spatial lag operator and $\mathbf{W}\mathbf{y}$ as the spatial lag term with spatial autoregressive parameter ρ, $\mathbf{X}$ as an $n \times k$ matrix of observations on exogenous explanatory variables with $k \times 1$ coefficient vector β, and a $n \times 1$ vector of errors $\mathbf{u}$.

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

$$\mathbf{u} = \lambda\mathbf{W}\mathbf{u} + \varepsilon \tag{2}$$

where λ is the spatial autoregressive parameter, and the innovations are potentially heteroskedastic, such that $\mathrm{E}[\varepsilon_i^2] = \sigma_i^2$. All other assumptions are standard.

In this paper, we focus on the estimation of the parameters of the error model, more specifically the coefficient λ in Equation (2). We consider this first with a classic assumption for the error variance-covariance matrix, where $\mathrm{E}[\varepsilon_i^2] = \sigma^2$. This is combined with three particular specifications for the main regression model:

- the standard regression model, containing only exogenous variables

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{u}$$

- a regression model containing both exogenous and endogenous variables

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{Y}\gamma + \mathbf{u}$$

where $\mathbf{Y}$ is a $n \times s$ matrix of observations on endogenous variables, with associated coefficient vector γ.

- a combo model containing a spatial lag term (and possibly additional endogenous variables, not considered here)

$$\mathbf{y} = \rho\mathbf{W}\mathbf{y} + \mathbf{X}\beta + \mathbf{u},$$

The second set of cases considers the same three regression specifications, but now in combination with a spatial autoregressive error term that has a heteroskedastic disturbance, $\mathrm{E}[\varepsilon_i^2] = \sigma_i^2$.

[3] Sometimes also referred to as SARAR model. We prefer the term SAR-SAR to stress that model the substantive and the error specification are spatial autoregressive – SAR

A general way to express the most encompassing model is as:

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

where, in the most comprehensive case, $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}, \mathbf{W}\mathbf{y}]$ and the $(k + s + 1) \times 1$ coefficient vector is rearranged as $\delta = [\beta'\gamma'\rho]'$ (i.e., a column vector). When endogenous variables are included (either a spatial lag, other endogenous variables, or both), a $n \times p$ matrix of instruments $\mathbf{H}$ will be needed.

2.2 Estimation Strategy

The estimation strategy outlined by K-P-D consists of two major components. One has to do with the estimation of the model coefficients using a feasible generalized least squares approach, with a consistent estimate for λ in hand. The second component deals with obtaining not only a consistent but also an efficient estimate for λ.

2.2.1 Spatially Weighted Least Squares

The rationale behind spatially weighted estimation is that a simple transformation (the so-called spatial Cochrane-Orcutt transformation) removes the spatial dependence from the error term in the regression equation:

$$\begin{aligned} (\mathbf{I} - \lambda\mathbf{W})y &= (\mathbf{I} - \lambda\mathbf{W})\mathbf{Z}\delta + (\mathbf{I} - \lambda\mathbf{W})\mathbf{u} \\ \mathbf{y}_s &= \mathbf{Z}_s\delta + \varepsilon, \end{aligned}$$

using the notation from Equation (3), and with $\mathbf{y}_s$ and $\mathbf{Z}_s$ as filtered variables, $\mathbf{y}_s = \mathbf{y} - \lambda\mathbf{W}\mathbf{y}$ and $\mathbf{Z}_s = \mathbf{Z} - \lambda\mathbf{W}\mathbf{Z}$. Finally, ε is a potentially heteroskedastic, but not spatially correlated innovation term.

Spatially weighted least squares (SWLS – Anselin 1988) or spatial Cochrane-Orcutt estimation then consists of OLS or 2SLS applied to spatially filtered variables. In the case where only exogenous variables are included in the model, this boils down to:

$$\hat{\beta}_{SWLS} = (\mathbf{X}_s'\mathbf{X}_s)^{-1}\mathbf{X}_s'\mathbf{y}_s, \tag{4}$$

with $\mathbf{X}_s = \mathbf{X} - \hat{\lambda}\mathbf{W}\mathbf{X}$, using a consistent estimate $\hat{\lambda}$ for the autoregressive parameter, and $\mathbf{y}_s$ as before, but with a consistent estimate $\hat{\lambda}$ for λ.

When both exogenous and endogenous variables are included in the model (which encompasses both non-spatial endogenous variables as well as spatially lagged dependent variables), we use the notation of Equation (3). In this instance, the estimator is referred to as Generalized Spatial Two Stage Least Squares (GS2SLS – Kelejian and Prucha 1998). It is obtained as:

$$\hat{\delta}_{GS2SLS} = [\mathbf{Z}'_s\mathbf{H}(\mathbf{H}'\mathbf{H})^{-1}\mathbf{H}'\mathbf{Z}_s]^{-1}\mathbf{Z}'_s\mathbf{H}(\mathbf{H}'\mathbf{H})^{-1}\mathbf{H}'\mathbf{y}_s, \tag{5}$$

where the notation is as before, using a consistent estimate $\hat{\lambda}$ for λ and the instrument matrix $\mathbf{H}$. Note that, importantly, the instrument matrix is $\mathbf{H}$ and

not $\mathbf{H}_s$. In other words, the instruments are not subjected to a spatial filter (i.e., a spatial Cochrane-Orcutt transformation).

The general theoretical result is that consistency is obtained as long as the estimate for the nuisance parameter $\hat{\lambda}$ is consistent. However, the efficiency of the estimates for β (and δ) can be improved by using an *optimal* estimator for λ. This is obtained through the GMM procedure. In addition, the GMM estimator for λ suggested by Kelejian and Prucha (2010) remains consistent in the presence of heteroskedasticity. The generalized moments estimator suggested in the earlier work of Kelejian and Prucha (1998, 1999) is not consistent in the presence of heteroskedasticity (and neither is the maximum likelihood estimator, see Lin and Lee 2010).

2.2.2 Consistent and Efficient Estimation of λ

The estimation of the spatial autoregressive coefficient λ is obtained from the solution of a system of moments equations, expressed as functions of the parameter and residuals. The residual vector, say $\mathbf{u}$ (in what follows, we do not use separate notation to distinguish the residuals from the error terms, since we always need residuals in practice) results from a set of initial consistent (but not efficient) estimates for the model coefficients. For example, in a model with only exogenous explanatory variables, this would be based on ordinary least squares (OLS). In the presence of endogenous explanatory variables, two stage least squares (2SLS) would be necessary.

The point of departure for K-P-D's estimation procedure are two moment conditions, expressed as functions of the innovation terms ε and their spatial lags $\varepsilon_L = \mathbf{W}\varepsilon$. They are:

$$\begin{aligned} n^{-1}\mathrm{E}[\varepsilon_L'\varepsilon_L] &= n^{-1}\mathrm{tr}[\mathbf{W}\mathrm{diag}[E(\varepsilon_i^2)]\mathbf{W}'] \quad (6)\\ n^{-1}\mathrm{E}[\varepsilon_L'\varepsilon] &= 0, \quad (7) \end{aligned}$$

where ε_L is the spatially lagged innovation vector and tr stands for the matrix trace operator. The main difference with the moment equations in Kelejian and Prucha (1998, 1999) is that the innovation vector is allowed to be heteroskedastic of general form, hence the inclusion of the term $\mathrm{diag}[E(\varepsilon_i^2)]$ in Equation 6. In the absence of heteroskedasticity, the RHS of the first condition simplifies to $\sigma^2 n^{-1} tr[\mathbf{W}\mathbf{W}']$. With σ^2 replaced by $\mathrm{E}[(n^{-1})\varepsilon'\varepsilon]$, the first moment condition then becomes:

$$\begin{aligned} n^{-1}\mathrm{E}[\varepsilon_L'\varepsilon_L] &= E[n^{-1}\varepsilon'\varepsilon](n^{-1})tr[\mathbf{W}\mathbf{W}'] \quad (8)\\ &= n^{-1}\mathrm{E}[\varepsilon'(n^{-1})tr[\mathbf{W}\mathbf{W}']\mathbf{I}\varepsilon] \quad (9) \end{aligned}$$

K-P-D introduce a number of simplifying notations that allow the moment conditions to be written in a very concise form. Specifically, they define

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{W}'\mathbf{W} - \mathrm{diag}(\mathbf{w}_{.i}'\mathbf{w}_{.i})\\ \mathbf{A}_2 &= \mathbf{W}, \end{aligned}$$

where $\mathbf{w_{.i}}$ is the i-th column of the weights matrix $\mathbf{W}$. Upon further inspection, we see that each element i of the diagonal matrix $\text{diag}(\mathbf{w}'_{.i}\mathbf{w}_{.i})$ consists of the sum of squares of the weights in the i-th column, i.e., the diagonal elements of $\mathbf{W}'\mathbf{W}$. Note that therefore the diagonal elements and thus also the trace of both matrices are zero. In Lee (2007), a slightly more general set of conditions is considered, where matrices of the form $\mathbf{A}$ could have non-zero diagonal elements, as long as the trace equals zero.

Using the new notation, the moment conditions become:

$$\begin{aligned} n^{-1}\text{E}[\varepsilon'\mathbf{A_1}\varepsilon] &= 0 \\ n^{-1}\text{E}[\varepsilon'\mathbf{A_2}\varepsilon] &= 0 \end{aligned}$$

In order to operationalize these equations, the (unobservable) innovation terms ε are replaced by their counterpart expressed as a function of regression residuals. Since $\mathbf{u} = \lambda\mathbf{W}\mathbf{u} + \varepsilon$, it follows that $\varepsilon = \mathbf{u} - \lambda\mathbf{W}\mathbf{u} = \mathbf{u}_s$, the spatially filtered residuals. The operational form of the moment conditions is then:

$$\begin{aligned} n^{-1}\text{E}[\mathbf{u}'_s\mathbf{A_1}\mathbf{u}_s] &= 0 \\ n^{-1}\text{E}[\mathbf{u}'_s\mathbf{A_2}\mathbf{u}_s] &= 0 \end{aligned}$$

The initial consistent estimate for λ is obtained by solving these moment conditions.

The estimates for λ obtained from the nonlinear least squares are consistent, but not efficient. Optimal estimates are found from a weighted nonlinear least squares procedure, or, $\text{argmin}_\lambda \mathbf{m}'\mathbf{\Psi}^{-1}\mathbf{m}$, where $\mathbf{\Psi}$ is a weighting matrix. The optimal weights correspond to the inverse variance of the moment conditions.

K-P-D show the general expression for the elements of the 2×2 matrix $\mathbf{\Psi}$ to be of the form:

$$\psi_{q,r} = (2n)^{-1}\text{tr}[(\mathbf{A}_q + \mathbf{A}'_q)\mathbf{\Sigma}(\mathbf{A}_r + \mathbf{A}'_r)\mathbf{\Sigma}] + n^{-1}\mathbf{a}'_q\mathbf{\Sigma}\mathbf{a}_r,$$

for $q, r = 1, 2$ and with $\mathbf{\Sigma}$ as a diagonal matrix with as elements $(u_i - \lambda u_{L_i})^2 = u^2_{s_i}$, i.e., the squares of the spatially filtered residuals. The second term in this expression is quite complex, and we refer for further technical details to Kelejian and Prucha (2010), Arraiz et al. (2010), Drukker et al. (2012), as well as Anselin (2011). However, it is important to note that this second term becomes zero when there are only exogenous explanatory variables in the model (i.e., when OLS is applicable). The term derives from the expected value of a cross product of expressions in the $\mathbf{Z}$ matrix and the error term $\mathbf{u}$. Hence, when no endogenous variables are included in $\mathbf{Z}$, the expected value of this cross product amounts to $\text{E}[\mathbf{u}] = 0$.

3 The Spatial Error Model with Homoskedasticity

The estimation of the spatial error model without heteroskedasticity uses Equation 8 for $\mathbf{A}_1$ instead of the general expression in Equation 6.

Table 2: Spatial error model with exogenous variables and homoskedasticity

Variable	GeoDaSpace	PySAL[1]	sphet2	Stata	PySAL[2]
CONSTANT	6.6762	6.6586	6.6762	6.9884	6.9884
	0.3498)	(0.3609)	(0.3498)	(0.3605)	(0.3605)
RD90	3.9450	3.9417	3.9450	3.9945	3.9945
	(0.1553)	(0.1598)	(0.1553)	(0.1612)	(0.1612)
UE90	-0.0770	-0.0745	-0.0770	-0.1240	-0.1240
	(0.0471)	(0.0479)	(0.0471)	(0.0490)	(0.0490)
lambda	0.4150	0.4656	0.4149	0.4124	0.4124
	(0.0192)	(0.0187)	(0.0194)	(0.0194)	(0.0194)

[1]PySAL with option A1='het'

[2]PySAL using the code to match Stata estimates

3.1 Exogenous Variables Only

When all the explanatory variables in the model are exogenous, the estimation boils down to SWLS, or OLS with spatially filtered variables. As pointed out in Anselin (2011), the use of the expression for matrix $\mathbf{A}_1$ as suggested in Drukker et al. (2012) yields a variance-covariance matrix for the coefficients that is not block-diagonal between the estimates $\hat{\beta}$ and $\hat{\lambda}$, which thus violates a fundamental result for FGLS (see, e.g., Breusch 1980). The block-diagonality is only obtained for matrices $\mathbf{A}_{1,2}$ for which the diagonal consists of zeros, instead of only having their trace equal zero. Therefore, Anselin (2011) suggests the use of $\mathbf{A}_1 = \mathbf{W}'\mathbf{W} - \text{diag}(\mathbf{W}'\mathbf{W})$, which is an alternative GMM estimator. As a result, the value for $\hat{\lambda}$ using this approach will differ from programs that do not make this adjustment.

To illustrate this point, we carry out the estimation of our example regression model in GeoDaSpace, `sphet2` (the K-P-D method is not included in `sphet1`) and Stata, using the default settings in each. We also carry out two estimations in PySAL with custom settings for the options. These results illustrate respectively the estimates obtained with the alternative form for $\mathbf{A}_1$ and a replication of the Stata results.

One reason for the different results in Stata is that it appears that estimation for the coefficient estimates is implemented as 2SLS, even when no endogenous variables are present in the model. This does not yield estimates equal to the results of Equation 4. We can mimic the results in Stata by labeling the explanatory variables `RD90` and `UE90` as endogenous and using them as instruments as well. Consequently, the constant term becomes the only truly exogenous variable in this setup. Without spatially filtered variables, this would simply yield the OLS results. More specifically, consider the standard result for 2SLS:

$$\hat{\delta}_{2SLS} = [\mathbf{Z}'\mathbf{H}(\mathbf{H}'\mathbf{H})^{-1}\mathbf{H}'\mathbf{Z}]^{-1}\mathbf{Z}'\mathbf{H}(\mathbf{H}'\mathbf{H})^{-1}\mathbf{H}'\mathbf{y}.$$

Substituting $\mathbf{X}$ for both $\mathbf{Z}$ and $\mathbf{H}$ yields:

$$\begin{aligned} \hat{\delta}_{2SLS} &= [\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}]^{-1}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}, \\ &= [\mathbf{X}'\mathbf{X}]^{-1}\mathbf{X}'\mathbf{y}, \end{aligned}$$

the OLS estimate. However, this result does not apply to the spatially weighted least squares (or spatial Cochrane-Orcutt). Substituting $\mathbf{X}$ for $\mathbf{H}$ and the spatially filtered $\mathbf{X}_s$ for $\mathbf{Z}_s$ in Equation 5 yields:

$$\hat{\delta}_{GS2SLS} = [\mathbf{X'}_s\mathbf{X}(\mathbf{X'X})^{-1}\mathbf{X'X}_s]^{-1}\mathbf{X'}_s\mathbf{X}(\mathbf{X'X})^{-1}\mathbf{X'y}_s,$$

which does not reduce to SWLS, Equation 4. The main reason for this is that the instruments are not spatially filtered, as pointed out before. Only if the instruments would also be spatially filtered would the expression reduce to SWLS, however this is not the case for GS2SLS. Hence, it would seem that the estimate provided by Stata in this case is not the proper SWLS estimate.

The results are reported in Table 2. The default for GeoDaSpace essentially matches the results for `sphet2`. However, these results differ somewhat from what Stata yields. This is entirely due to the particular way in which spatially weighted least squares are implemented in Stata for the OLS case. When the option for `A1` is set to `het`, the value for $\hat{\lambda}$ is slightly higher due to the use of a different set of moment equations, as mentioned above. When PySAL is tricked to use the 2SLS routine for estimation, the results are identical to those from Stata.

The Python code to obtain the PySAL results in Table 2 is given in Listing 1 (for the sake of readability, we have omitted the `>>>` Python prompt).

3.2 Exogenous and Endogenous Variables

When endogenous variables are included in addition to the exogenous variables, estimation is based on GS2SLS, as in Equation 5. To illustrate this, we take `UE90` as the endogenous variable with `FP89` as the instrument in our example homicide rate regression. The results for four estimation procedures are listed in Table 3. First is GeoDaSpace with the default for homoskedasiticy. This completely matches the results for `sphet2` in the third column and Stata in column four of the Table (`sphet1` does not include this functionality). Column two shows the estimates obtained when the `A1` option is set to `'het'`. As we have seen in the previous case, the estimate for λ is slightly different, which results in slight differences for the β as well. The code for the two options in PySAL is given in Listing 2.

3.3 Combo Model with Spatial Lag and Spatial Error

As the third case with homoskedasticity, we consider the combo model that includes both a spatially lagged dependent variable and a spatial error term. Several options are available for this case in GeoDaSpace and PySAL. In addition to the selection of `A1` (in PySAL), the order of the spatial lag operator need to be determined to construct the spatially lagged explanatory variables ($\mathbf{WX}$) as instruments for the spatial lag term. The default in GeoDaSpace (and PySAL) is to use only the first order lag, whereas both `sphet2` (`sphet1` does not include this estimator) and Stata use the second order spatial lag to construct

Listing 1: PySAL code for error model – homoskedasticity – exogenous only

```
# preliminaries - import the needed modules
import numpy as np
import pysal as ps
# create the variables from the nat.dbf data set
db = ps.open('nat.dbf','r')
hr90 = db.by_col("HR90")
y = array(hr90)
y.shape = (len(hr90),1)
x_names = ['RD90','UE90']
x = np.array([db.by_col(var) for var in x_names]).T
# create the spatial weights as queen contiguity
w = ps.queen_from_shapefile("NAT.SHP")
# row-standardize the weights
w.transform = 'r'
# PySAL with GeoDaSpace default settings and results
reg1 = ps.spreg.GM_Error_Hom(y,x,w,name_y='HR90', \
           name_x = x_names, name_w = 'nat_queen.gal', \
           name_ds='NAT')
print reg1.summary
# PySAL with A1 = 'het'
reg1a = ps.spreg.GM_Error_Hom(y,x,w,A1='het', \
             name_y='HR90',name_x = x_names, \
             name_w = 'nat_queen.gal',name_ds='NAT')
print reg1a.summary
# PySAL base class to mimic Stata estimates
# Note: base class does not have a summary method,
# so that results need to be printed explicitly
ones = np.ones(y.shape)
reg1c = ps.spreg.error_sp_hom.BaseGM_Endog_Error_Hom(y, \
             ones,yend=x, q=x, w=w, A1='hom_sc')
print reg1c.betas
print map(np.sqrt, reg1c.vm.diagonal())
```

the instruments. With the lag order set to two in GeoDaSpace (an option under `Instruments` in the `GeoDaSpace Preferences` panel in the GeoDaSpace GUI), its results are identical to those for `sphet2` and Stata, as illustrated in Table 4.

The results in columns two and three of the Table illustrate the effect of the various options on the estimates. In column two, only the first order spatial lags are used to construct the instruments, which is the default in GeoDaSpace. The estimates of both λ and ρ are affected, but only slightly. The effect of taking `A1` as `'het'` shown in column three seems to be more pronounced, especially of the estimate for λ. The code to implement the various options in PySAL is given in Listing 3.

Table 3: Spatial error model with endogenous variables and homoskedasticity

Variable	**GeoDaSpace**	**PySAL**[1]	**sphet2**	**Stata**
CONSTANT	21.0606	21.0288	21.0606	21.0606
	(1.5385)	(1.5362)	(1.5385)	(1.5385)
RD90	8.2420	8.2376	8.2420	8.2420
	(0.4888)	(0.4881)	(0.4888)	(0.4888)
UE90[2]	-2.2438	-2.2392	-2.2438	-2.2438
	(0.2290)	(0.2286)	(0.2290)	(0.2290)
lambda	0.4944	0.4934	0.4944	0.4944
	(0.0217)	(0.0216)	(0.0217)	(0.0217)

[1]PySAL with option A1='het'

[2]UE90 instrumented by FP89

Listing 2: PySAL code for error model – homoskedasticity – endogenous

```
# y, x and w created as before, db as open data set
# extract exogenous, endogenous from x
xex = x[:,0]
xex.shape = (len(hr90),1)
yend = x[:,1]
yend.shape = (len(hr90),1)
# create the instrument from the nat.dbf data set
fp89 = db.by_col("FP89")
q = array(fp89)
q.shape = (len(fp89),1)
# PySAL with GeoDaSpace default settings and results
reg2 = ps.spreg.GM_Endog_Error_Hom(y,xex,yend,\
           q,w,name_y="HR90",name_x=["RD90"],\
           name_yend=["UE90"],name_q=["FP89"],\
           name_w="natqueen.gal",name_ds="nat.shp")
print reg2.summary
# PySAL with A1 = 'het'
reg2a = ps.spreg.GM_Endog_Error_Hom(y,xex,yend,\
             q,w,A1='het',name_y="HR90",name_x=["RD90"],\
             name_yend=["UE90"],name_q=["FP89"],\
             name_w="natqueen.gal",name_ds="nat.shp")
print reg2a.summary
```

Table 4: Combo model with homoskedasticity

Variable	GeoDaSpace[1]	PySAL[2]	PySAL[3]	sphet2	Stata
CONSTANT	6.9362	6.9530	6.9406	6.9362	6.9362
	(0.5120)	(0.5161)	(0.5327)	(0.5120)	(0.5120)
RD90	4.0061	4.0089	4.0074	4.0061	4.0061
	(0.1764)	(0.1762)	(0.1758)	(0.1764)	(0.1764)
UE90	-0.0978	-0.0854	-0.0957	-0.0978	-0.0978
	(0.0481)	(0.0483)	(0.0490)	(0.0481)	(0.0481)
W_HR90	-0.0190	-0.0356	-0.0220	-0.0190	-0.0190
	(0.0513)	(0.0519)	(0.0543)	(0.0513)	(0.0513)
lambda	0.4364	0.4521	0.5098	0.4364	0.4364
	(0.0421)	(0.0415)	(0.0376)	(0.0421)	(0.0421)

[1]GeoDaSpace with lags = 2

[2]PySAL with w_lags = 1 (GeoDaSpace default)

[3]PySAL with option A1='het' and w_lags = 2

Listing 3: PySAL code for combo model – homoskedasticity

```
# y, x and w created as for homoskedastic - exogenous case
# PySAL with w_lags = 2 and A1 = hom_sc (default)
reg3 = ps.spreg.GM_Combo_Hom(y,x,w=w,w_lags=2,\
           name_y='HR90',name_x = x_names, \
           name_w = 'nat_queen.gal',name_ds='NAT')
print reg3.summary
# PySAL with GeoDaSpace default settings (w_lags = 1)
rreg3a = ps.spreg.GM_Combo_Hom(y,x,w=w,name_y='HR90',\
              name_x = x_names, name_w = 'nat_queen.gal',\
              name_ds='NAT')
print reg3a.summary
# PySAL with A1 = 'het' and w_lags = 2
reg3b = ps.spreg.GM_Combo_Hom(y,x,w=w,w_lags=2,\
             A1='het',name_y='HR90',name_x = x_names, \
             name_w = 'nat_queen.gal',name_ds='NAT')
print reg3b.summary
```

4 The Spatial Error Model with Heteroskedasticity

4.1 Exogenous Variables Only

When all the explanatory variables are exogenous, we encounter a similar situation for the Stata output as in the homoskedastic case. Again, Stata uses a 2SLS estimation routine for the OLS results, which yields different estimates from GeoDaSpace and `sphet`.

The results are given in Table 5. The estimates for GeoDaSpace and `sphet2` are virtually identical, with a slight difference in the value for λ and most of the standard errors identical (with slight differences for some). However, the results for the older `sphet1` differ from those for `sphet2`. We can almost replicate the results for `sphet1` by setting the option `step1c = True` in PySAL, as shown in the second column. This refers to a slight difference in the estimation steps between Arraiz et al. (2010) and Drukker et al. (2012). In the former, the initial consistent estimation from the unweighted optimization of the moment equations is followed by a second efficient estimation before moving to the spatially weighted least squares. In the subsequent paper by Drukker et al. (2012), this additional step is skipped. The default in GeoDaSpace (and PySAL) is to follow the latter approach, but the former can be invoked with the `step1c` option.

The estimated regression coefficients are identical between this call to PySAL and `sphet1`, but there is a slight difference in the estimate for λ and in the coefficient standard errors.

As before, the only way to replicate the results from Stata is to use the 2SLS estimation procedure as shown before. The estimates are given in the last column of the Table.

The code to carry out these procedures in PySAL is given in Listing 4.

Table 5: Spatial error model with exogenous variables and heteroskedasticity

Variable	**G-Space**	**PySAL**[1]	**sphet1**	**sphet2**	**Stata**	**PySAL**[2]
CONSTANT	6.6586	6.5782	6.5782	6.6586	6.9777	6.9777
	(0.4749)	(0.4749)	(0.4594)	(0.4745)	(0.4622)	(0.4622)
RD90	3.9417	3.9275	3.9275	3.9417	3.9911	3.9911
	(0.2602)	(0.2604)	(0.2316)	(0.2599)	(0.2326)	(0.2325)
UE90	-0.0745	-0.0630	-0.0630	-0.0745	-0.1225	-0.1225
	(0.0611)	(0.0611)	(0.0589)	(0.0611)	(0.0592)	(0.0592)
lambda	0.4753	0.4763	0.4756	0.4740	0.4721	0.4721
	(0.0235)	(0.0235)	(0.0237)	(0.0237)	(0.0236)	(0.0236)

[1] PySAL with option step1c = True

[2] PySAL using the code to match Stata estimates

Listing 4: PySAL code for error model – heteroskedasticity – exogenous only

```
# all arrays and weights objects as before
# PySAL with GeoDaSpace default settings and results
reg4 = ps.spreg.GM_Error_Het(y,x,w,name_y="HR90",\
           name_x=x_names,name_w="natqueen.gal",\
           name_ds="nat.shp")
print reg4.summary
# PySAL with step1c = True
reg4a = ps.spreg.GM_Error_Het(y,x,w,step1c=True,\
             name_y="HR90",name_x=x_names,\
             name_w="natqueen.gal",name_ds="nat.shp")
print reg4a.summary
# PySAL to mimic Stata output
reg4b = ps.spreg.error_sp_het.BaseGM_Endog_Error_Het(y,ones,\
             yend=x, q=x, w=w)
print reg4b.betas
print map(np.sqrt, reg4b.vm.diagonal())
```

4.2 Exogenous and Endogenous Variables

In the presence of endogenous variables, the heteroskedastic case proceeds in the same fashion as the homoskedastic one, using GS2SLS.

We use the same example as before, with `UE90` as the endogenous variable and `FP89` as the instrument. The results for four estimation procedures are listed in Table 6.

The first columnn shows the results for GeoDaSpace (and PySAL) with all the default options. This completely matches the results for `sphet2` (`sphet1` does not include this functionality) and Stata in columns three and four of the Table.

Column two shows the estimates obtained when the `step1c` option is set to `True`. The corresponding estimate for λ is slightly different, which results in slight differences for the β as well.

The code for the two options in PySAL is given in Listing 5.

4.3 Combo Model with Spatial Lag and Spatial Error

The final specification we consider pertains to the combo model with heteroskedasticity, listed in Table 7. As in the homoskedastic case, an important option is the order of the spatial lag for the exogenous variables that is used to construct the instruments. The default for GeoDaSpace (and PySAL) is first order only, whereas the default in `sphet2` and Stata is second order. A second potential difference in the estimates is whether the additional step (`step1c`) from Arraiz et al. (2010) in included. This is controlled by setting the `step1c = True` option in PySAL (the default is `False`).

Table 6: Spatial error model with endogenous variables and heteroskedasticity

Variable	GeoDaSpace	PySAL[1]	sphet2	Stata
CONSTANT	21.0288	21.2384	21.0288	21.0288
	(2.5629)	(2.5165)	(2.5629)	(2.5629)
RD90	8.2376	8.2662	8.2376	8.2376
	(0.7817)	(0.7637)	(0.7817)	(0.7817)
UE90[2]	-2.2392	-2.2695	-2.2392	-2.2392
	(0.3902)	(0.3830)	(0.3902)	(0.3902)
lambda	0.4667	0.4298	0.4667	0.4667
	(0.0298)	(0.0322)	(0.0298)	(0.0298)

[1]PySAL with option step1c = True

[2]UE90 instrumented by FP89

Listing 5: PySAL code for error model – heteroskedasticity – endogenous

```
# all arrays and weights object as before
# PySAL with GeoDaSpace default settings and results
reg5 = ps.spreg.GM_Endog_Error_Het(y,xex,yend,q,w,\
           name_y="HR90",name_x=["RD90"],\
           name_yend=["UE90"],name_q=["FP89"],\
           name_w="natqueen.gal",name_ds="nat.shp")
print reg5.summary
# PySAL with step1c = True
reg5a = ps.spreg.GM_Endog_Error_Het(y,xex,yend,q,w,\
             step1c=True,name_y="HR90",name_x=["RD90"],\
             name_yend=["UE90"],name_q=["FP89"],\
             name_w="natqueen.gal",name_ds="nat.shp")
print reg5a.summary
```

As shown in the first and last two columns of Table 7, the results for GeoDaSpace/PySAL with the lag option set to order two completely match the estimates obtained by `sphet2` and Stata. When only the first order lag is used (the default in GeoDaSpace/PySAL), the estimate for λ is somewhat higher, with associated minor changes in the β estimates. Using the additional estimation step (with second order lags) yields results in PySAL (column 3) that approach those from `sphet1` (column 4), with the highest value for λ of all the options.

The code for the various settings in PySAL is given in Listing 6.

Table 7: Combo model with heteroskedasticity

Variable	G-Space[1]	PySAL[2]	PySAL[3]	sphet1	sphet2	Stata
CONSTANT	6.9406	6.9452	7.0209	7.0196	6.9406	6.9406
	(0.8600)	(0.8722)	(0.8836)	(0.8251)	(0.8600)	(0.8600)
RD90	4.0074	4.0063	4.0054	4.0057	4.0074	4.0074
	(0.3261)	(0.3242)	(0.3198)	(0.3212)	(0.3261)	(0.3261)
UE90	-0.0957	-0.0830	-0.0640	-0.0643	-0.0957	-0.0957
	(0.0664)	(0.0671)	(0.0677)	(0.0640)	(0.0664)	(0.0664)
W_HR90	-0.0220	-0.0370	-0.0709	-0.0702	-0.0220	-0.0220
	(0.0876)	(0.0905)	(0.0918)	(0.0839)	(0.0876)	(0.0876)
lambda	0.5584	0.5961	0.6406	0.6399	0.5584	0.5584
	(0.0507)	(0.0500)	(0.0480)	(0.0460)	(0.0507)	(0.0507)

[1] GeoDaSpace with lags = 2

[2] PySAL with w_lags = 1 (GeoDaSpace default)

[3] PySAL with option step1c = True and w_lags = 2

Listing 6: PySAL code for combo model – heteroskedasticity

```
# all arrays and weights object as before
# PySAL with GeoDaSpace settings for w_lags = 2
reg6 = ps.spreg.GM_Combo_Het(y,x,w=w,w_lags=2,\
           name_y='HR90',name_x = x_names, \
           name_w = 'nat_queen.gal',name_ds='NAT')
print reg6.summary
# PySAL with w_lags  = 1 (default)
reg6a = ps.spreg.GM_Combo_Het(y,x,w=w,\
             name_y='HR90',name_x = x_names, \
             name_w = 'nat_queen.gal',name_ds='NAT')
print reg6a.summary
# PySAL with w_lags = 2 and step1c = True
reg6c = ps.spreg.GM_Combo_Het(y,x,w=w,w_lags=2,\
             step1c=True,name_y='HR90',name_x = x_names, \
             name_w = 'nat_queen.gal',name_ds='NAT')
print reg6c.summary
```

5 Concluding Remarks

The GMM estimators proposed by K-P-D constitute an important contribution to theoretical and applied econometrics. Heteroskedasticity is the rule rather than the exception in empirical cross-sectional work. As a result, estimates for the error spatial autoregressive coefficient that are robust to heteroskedasticity are extremely useful.

The proposed estimators are asymptotic in nature. Moreover, the suggested framework is very general, such that several alternative formulations for the moment equations may yield asymptotically equivalent results. In practice,

these alternatives will tend to lead to numerical differences in finite samples. In addition, in computational practice, the optimization of the moment equation conditions may result in minor differences due to the particular optimizer that is applied.

In this paper, we have illustrated how choices made in the implementation of the estimation routines may yield (slightly) different coefficient estimates. The differences are rather subtle and tend to occur at the third decimal level (in some instances at the second decimal). We have shown how the comprehensive implementation in the PySAL `spreg` routines provides the flexibility to select options that allow a range of estimation approaches. By doing so, we have illustrated how PySAL can replicate both the results obtained in R and Stata as well as assess the sensitivity of those results to the choices made in the software implementation. We confirm the power of open source software where the "documentation is in the code," which allows researchers to know exactly which option has been selected, rather than the all too common black box approach used in proprietary commercial software (see also Yalta and Yalta 2010, for a similar argument). We also provide full replicability of our results in the form of scripts to carry out the estimation in all the three software environments covered.

From a methodological perspective, it would seem that further investigation of the relative performance of the various asymptotically equivalent approaches is warranted.

References

Anselin, L. (1988). *Spatial Econometrics: methods and models*. Kluwer Academic Publishers, Dordrecht.

Anselin, L. (2011). GMM estimation of spatial error autocorrelation with and without heteroskedasticity. Technical report, GeoDa Center for Geospatial Analysis and Computation – Arizona State University. Available at `https://geodacenter.asu.edu/software/downloads/geodaspace`.

Anselin, L. and Rey, S. J. (2012). Spatial econometrics in an age of cybergiscience. *International Journal of Geographical Information Science*. in press.

Anselin, L. and Rey, S. J. (2013). *Using GeoDaSpace and PySAL for Modern Spatial Econometrics*. GeoDa Center for GeoSpatial Analysis and Computation, Arizona State University, Tempe, AZ. forthcoming.

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*, 50:592–614.

Baller, R., Anselin, L., Messner, S., Deane, G., and Hawkins, D. (2001). Structural covariates of U.S. county homicide rates: Incorporating spatial effects. *Criminology*, 39(3):561–590.

Breusch, T. (1980). Useful invariance results for generalized regression models. *Journal of Econometrics*, 13:327–340.

Drukker, D. M., Egger, P., and Prucha, I. (2012). On two-step estimation of a spatial autoregressive model with autoregressive disturbances and endogenous regressors. *Econometric Reviews*. forthcoming.

Drukker, D. M., Prucha, I. R., and Raciborski, R. (2011). A command for estimating spatial-autoregressive models with spatial-autoregressive disturbances and additional endogenous variables. Technical report, Stata Corp, College Station, TX.

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

Kelejian, H. H. and Prucha, I. R. (1999). A generalized moments estimator for the autoregressive parameter in a spatial model. *International Economic Review*, 40(2):509–33.

Kelejian, H. H. and Prucha, I. R. (2007). HAC estimation in a spatial framework. *Journal of Econometrics*, 140(1):131–154.

Kelejian, H. H. and Prucha, I. R. (2010). Specification and estimation of spatial autoregressive models with autoregressive and heteroskedastic disturbances. *Journal of Econometrics*, 157:53–67.

Koenker, R. and Zeileis, A. (2009). On reproducible econometric research. *Journal of Applied Econometrics*, 24:833–847.

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

Lin, X. and Lee, L.-F. (2010). GMM estimation of spatial autoregressive models with unknown heteroskedasticity. *Journal of Econometrics*, 157:34–52.

Messner, S., Anselin, L., Hawkins, D., Deane, G., Tolnay, S., and Baller, R. (2000). *An Atlas of the Spatial Patterning of County-Level Homicide, 1960–1990*. National Consortium on Violence Research, Carnegie-Mellon University, Pittsburgh, PA (CD-ROM).

Messner, S. F. and Anselin, L. (2004). Spatial analyses of homicide with areal data. In Goodchild, M. and Janelle, D., editors, *Spatially Integrated Social Science*, pages 127–144. Oxford University Press, New York, NY.

Piras, G. (2010). sphet: Spatial models with heteroskedastic innovations in R. *Journal of Statistical Software*, 35:1–21.

Rey, S. and Anselin, L. (2007). PySAL, a Python library of spatial analytical methods. *The Review of Regional Studies*, 37(1):5–27.

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4):817–838.

Yalta, A. T. and Yalta, A. Y. (2010). Should economists use open source software for doing research. *Computational Economics*, 35:371–394.