Spatial Econometrics

Updated October 21, 2010 • Josh Mills

Summary: This handout provides background on spatial econometrics, a relatively new extension of econometrics concerned with quantitatively modeling spatial dependence. The basic types of spatial econometric models are covered, as well as software packages that can estimate spatial econometric models.

Contents

Introduction ... 1
The Ingredients for Spatial Econometric Models ... 2
The Weights Matrix ... 3
Spatial Econometric Models ... 6
Estimation and Testing for Spatial Dependence ... 8
Interpretation of Spatial Econometric Models ... 11
Spatial Panel Models: In Brief ... 12
Spatial Count Models ... 13
Software for Spatial Econometric Analysis ... 14
An Application of Spatial Econometrics: Census Data in Tippecanoe County ... 15
References ... 35
Additional References: Applications ... 37
Additional References: Spatial Count Data Modeling ... 38

Introduction

Statistical models have traditionally focused on establishing links between some dependent variable with a number of factors, or explanatory variables. These models, however, do not take into account the concept of space. For example, a model predicting crash frequencies at signalized intersections may include observations from a certain geographical area, such as West Lafayette or Tippecanoe County, but may not account for the locations of these crashes. A series of signalized intersections along US-52, a main arterial in the Lafayette area, have high crash frequencies, but a typical statistical model, such as a negative binomial model, would not account for the fact that, for example, the intersection with Greenbush Road is close to the intersection with Union Street. These spatial relationships may be important. Since the Greenbush Road intersection is downstream from the Union Street intersection, anything that occurs at the Union Street intersection, such as a high number of crashes, may have some effect on the Greenbush Road intersection - in other words, there could be spillover effects.

Spatial relationships can also play an important role in economics. The opening of the Subaru plant in Lafayette, for example, resulted in changes to the local economy, changes to traffic patterns, and other changes in nearby areas. Workers at the Subaru plant commute from different locations in west central Indiana, and some of these workers may have moved to Lafayette from farther away areas, such as Rensselaer or Fowler, to have a shorter commute. Thus, the opening of the Subaru plant affected not just Tippecanoe County but also adjacent counties, such as Clinton, Benton, Warren, White, Montgomery, and Carroll counties - in other words, there were spillover effects.

Spatial econometric models can account for these spillover effects. These types of models use a quantitative representation of spatial relationships for a certain area, known in the literature as a spatial weights matrix, for modeling purposes. Different types of spatial econometric models exist. Some models account for spatial spillovers, whereas other models account for variance over space.

Spatial econometric models have been used to analyze determinant of economic growth in eastern China (Shanzi 2010), how spatial patterns in rainfall and growth affect participation in civil conflicts (Jensen and Gleditsch 2009), how urban public policies impact housing values (Baumont 2009), how insect derivatives mitigate whitefly damage in cotton (Richards et al. 2008), whether creative people are attracted to places that allow creative activity to flourish (Wojan et al. 2007), the impacts of transportation infrastructure on property values (Cohen and Paul 2007), the impact of transit station location on land values in South Korea (Kim 2007), factors that affect the growth rate of elephant populations in Africa (Frank and Maurseth 2006), spatial patterns of election results in Portugal (Caleiro and Guerreiro 2005), the spatial distribution of Internet adoption in European Regions (Billon et al. 2009), what factors influence rural-urban land conversion (Huang et al. 2009; Zhou and Kockelman 2008), and how urban residential development impacts water quality (Atasoy et al. 2006).

The Ingredients for Spatial Econometric Models

Prior to understanding spatial econometric models, it is important to determine whether one has the right type of data for such models. In general, these are the requirements for estimating spatial econometric models:

1. **A study area.** What is the geographical scope of your study? Examples include: ZIP Codes in three adjacent states, pavement quality in different INDOT districts, crash frequencies in Tippecanoe County.

2. **A set of point-based or area-based data.** Examples of point-based data include crashes, home sale prices for a neighborhood or community, and traffic intersections. Examples of area-based data include economic data at the state, county, or ZIP Code levels, polling data (such as election results for the 2008 general election), and traffic analysis zones.

3. **A GIS-based file (such as an ESRI Shapefile)** that graphically represents the locations of the data. Area-based data can generally be obtained online, usually from the Census Bureau, whereas point-based data, in transportation, are usually obtained from internal, non-public databases. Shapefiles can be generated through most geographical software packages including ArcGIS and TransCAD.

4. **Motivation for using spatial econometric models.** This pertains to the main goals of your research. Is there some theory or example that shows evidence that, for example, home prices are affected by the prices of neighboring properties? Employment changes in one county affect adjacent counties? A full or nearly full parking lot will affect where a driver chooses to park? A literature review must be performed to determine that there is a reason to use spatial econometric models - otherwise, using these models would be akin to using a hammer to clean a window - the tool to address the research problem must be appropriate for the research problem.

The Weights Matrix

Suppose that for a model, the study area consists of the states of Ohio, Indiana, Illinois, and Kentucky. A spatial econometric model would then consist of four observations. A weights matrix is used to represent which of these spatial units (in this case, states) are neighbors of each other. There are multiple ways to specify a spatial weights matrix - distance, contiguity, economic distance, and others (see LeSage and Pace 2009, for example). **Distance** and **contiguity** are the most common types of weights matrices used.

With a **contiguity-based weights matrix**, a state is a neighbor of another state if they share a common border. For this study area, Ohio would be a neighbor of Indiana and Kentucky but not Illinois. The two most commonly used contiguity-based criteria are rook contiguity and queen contiguity. These are discussed in detail in Bivand et al (2008).

Distance-based weights matrices determine neighbors based on some distance threshold. For area-based data, such as states, the geometric centroid of each polygon is used to calculate this distance. For example, assume the centroid-to-centroid distance from Indiana to Kentucky was 150 miles and the centroid-to-centroid distance from Indiana to Illinois was 100 miles. If a distance threshold of 125 miles were established, then any state that has a centroid-to-centroid distance of 125 miles from Indiana is considered a neighbor of Indiana. Illinois would be considered a neighbor of Indiana, whereas Kentucky would not be considered a neighbor.

K-nearest-neighbors also uses the distance threshold concept but chooses a distance threshold such that each observation has exactly "K" neighbors. With a weights matrix based on the 2-nearest neighbors criterion, the two closest states, based on centroid-to-centroid distance, would be considered neighbors of the state of interest. For example, if the centroid-to-centroid distance from Indiana to Ohio were 115 miles, then Illinois and Ohio would be considered neighbors of Indiana, but not Kentucky.

Weights matrices can be represented in multiple ways. The two most common types are the **binary** type, in which if a state is a neighbor of another state, its entry has a value of 1 and 0 otherwise. The other type **standardizes each row** of the weights matrix such that each of its elements sum to one. For a given sample size N, weights matrices have dimension N X N. For this example, the weights matrix will have dimension 4 X 4. Table 1 shows a binary contiguity matrix for the study area, whereas Table 2 shows a row-standardized contiguity matrix. Tables 1 and 2 represent grid-based data, which is often used in remote sensing and image processing.

The type of weights matrix used can potentially have a significant effect on the estimated model. Figure 1 shows a map of the Census tracts in Tippecanoe county with a queen-based contiguity matrix, a distance threshold matrix, and a 2-nearest neighbors matrix. Note how the distance threshold matrix (center) is considerably more dense than the other two matrices.

Table 1: Binary style contiguity-based weights matrix.

State	Ohio	Indiana	Illinois	Kentucky
Ohio	0	1	0	1
Indiana	1	0	1	1
Illinois	0	1	0	1
Kentucky	1	1	1	0

Table 2: Row-standardized style contiguity-based weights matrix.

State	Ohio	Indiana	Illinois	Kentucky
Ohio	0	0.5	0	0.5
Indiana	0.333	0	0.333	0.333
Illinois	0	0.5	0	0.5
Kentucky	0.333	0.333	0.333	0

Note that the diagonals in both weights matrices have values of zero. As will be seen in the following section, spatial econometric models assume that each spatial unit not does consider itself to be its own neighbor. Violating this assumption can result in a considerably more complex model that cannot easily be interpreted.

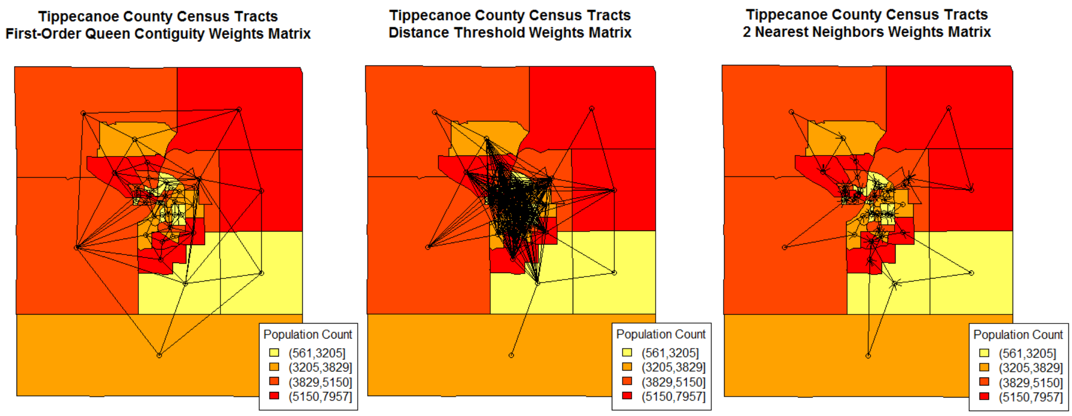
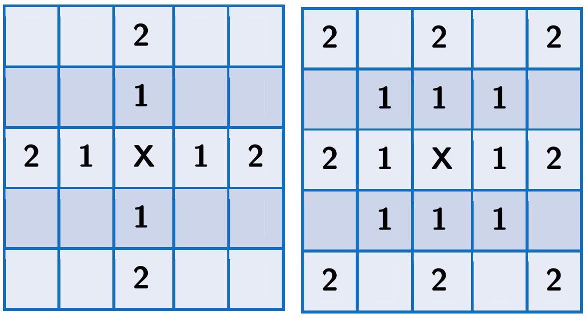


Figure 1: Weights matrices for Tippecanoe County Census tracts.

Spatial spillovers do not just affect an area's neighbors. They also affect the neighbors of the neighbors, rippling out until the spillover effects reach the limits of the study area. First-order neighbors are the closest neighbors of a spatial unit based on the criterion used, such as what is represented in Tables 1 and 2. Second-order neighbors are the neighbors of those first-order neighbors. Third-order neighbors are the neighbors of the second-order neighbors (see LeSage and Pace 2009 for additional details).



Consider the simple grid-based examples shown in Figure 2. The spatial unit of interest is represented by the X. First-order neighbors are represented by a 1, and second-order neighbors are represented by a 2.

Numerical manipulation of large weights matrices can be computationally intensive using full matrix methods. A 3200 x 3200 weights matrix, for example, has 10,240,000 elements. However, a majority of these elements in a typical weights matrix have a value of zero. In the study area, it is unlikely that a typical weights matrix would consider a ZIP Code in southeastern Ohio to be a neighbor of a ZIP Code in northwestern Illinois. Sparse matrix methods can be used to "compress" large weights matrices and account for the large number of elements with a value of zero. LeSage and Pace (2009) discuss in detail the advantages of using sparse matrix methods over full matrix methods.

Figure 2: First-order and second-order neighbors for rook contiguity (left) and queen contiguity (right).

To illustrate, consider the sparsity patterns of two different weights matrices for a study area (consisting of 3200 ZIP Codes), five nearest neighbors (5 NN) and seventy-five nearest neighbors (75 NN), shown in Figure 3. The layout of these sparsity patterns follows the same format as the weights matrices shown in Tables 1 and 2. White regions indicate zero values, while dark regions indicate nonzero values. Note that the 75 NN weights matrix (on the right) is considerably more dense than the 5 NN weights matrix. For both weights matrices, nonzero values comprise less than one-quarter of the total number of elements, illustrating why sparse matrix methods are necessary.

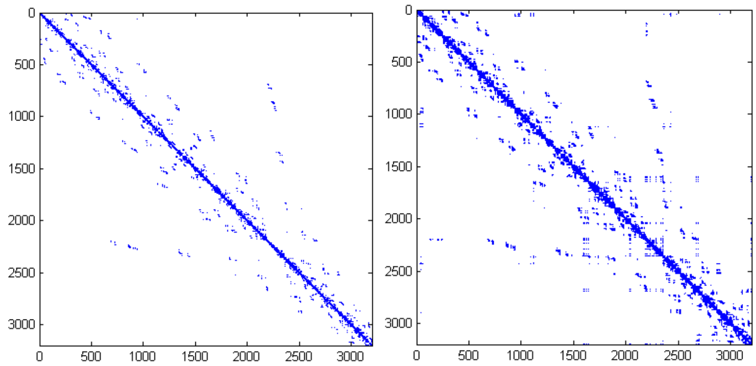


Figure 3: Sparsity patterns for the five nearest neighbors weights matrix (left) and the seventy-five nearest neighbors weights matrix (right).

Spatial Econometric Models

The concept of second-order, third-order, and higher-order effects can best be illustrated through the most common spatial econometric model, the **spatial lag model**. This model takes the form (LeSage and Pace 2009):

$$y = \rho W y + X\beta + \varepsilon \qquad (1)$$

where W represents the spatial weights matrix, ρ represents the spatial lag (or spatial autoregressive) parameter, and ε represents the vector of normally-distributed residuals. Eq. (1) can be rewritten as:

$$y = (I - \rho W)^{-1}(X\beta + \varepsilon) \qquad (2)$$

where I is the identity matrix. The term $(I - \rho W)^{-1}$ is the **spatial multiplier**. Written out as an infinite series, we have (LeSage and Pace 2009):

$$(I - \rho W)^{-1} = I + \rho W + \rho^2 W^2 + \rho^3 W^3 + \ldots \qquad (3)$$

This expansion of the spatial multiplier illustrates the impact on the second-order, third-order, and higher-order neighbors. Note that, similar to the AR(1) coefficient in time series analysis, the dependent variable is on both sides of Eq. (1). Interpretation of these models, however, is considerably more complex than AR(1) models. Time moves in one of two directions. Spatial models are multidirectional - space, by its nature, moves in multiple directions. The magnitude and impact of the spatial spillovers depend on the specification of the weights matrix - this is why the choice of the weights matrix is important. The weights matrices in Figure 2 yielded different spatial econometric models.

In time series, independent variables are sometimes lagged by one or two time periods to see if previous values of those variables have significant effects. Similarly, independent variables can be spatially lagged. However, since space is multidirectional, the value of these spatially lagged variables is dependent on the weights matrix. A state-level spatially lagged employment variable for the 2 nearest-neighbors criterion, for example, would represent the average of the employment variable for the state of interest, averaged (or distributed) out among the two nearest neighbors. Thus, if this spatially lagged variable was significant, then employment would have some effect not only on the state of interest but also on the two nearest neighbors of that state. A variant of the spatial lag model that spatially lags all independent variables is known as the **Spatial Durbin model** (SDM, LeSage and Pace 2009);

$$y = \rho W y + X\beta + WX\lambda + \varepsilon \qquad (4)$$

where λ is the vector of coefficients for spatially lagged independent variables WX. Use of this model as opposed to the spatial lag model in Eq. (1) can potentially remove omitted variable bias, discussed in detail in LeSage and Pace (2009).

In the spatial lag model, spatial dependence was assumed to be present in the dependent variable. In the **spatial error model**, spatial dependence is assumed to be present in the disturbance term (but NOT the dependent variable), such that (LeSage and Pace 2009);

$$\begin{aligned} y &= X\beta + \varepsilon \\ \varepsilon &= \lambda W \varepsilon + u \\ \varepsilon &= (I - \lambda W)^{-1} u \end{aligned} \qquad (5)$$

where ε again represents the disturbance, no longer distributed iid normal (identically and independently) and u is now the iid normal set of disturbances. λ represents the spatial error parameter. $(I - \lambda W)^{-1}$ is another form of the spatial multiplier and has a similar pattern of decay of effects with distance. An similar model is the **spatial moving average** (SMA) model (Fingleton 2008):

$$\begin{aligned} y &= X\beta + \varepsilon \\ \varepsilon &= \lambda W u + u \\ \varepsilon &= (I - \lambda W) u \end{aligned} \qquad (6)$$

As can be observed by comparing Eq. (5) and Eq. (6), the spatial multiplier is not present in the SMA model. The SMA model is used to model localized effects. By its specification, spatial effects will affect only the first-order neighbors as defined by the weights matrix.

Spatial dependence can be modeled simultaneously, both in the dependent variable and in the disturbances. There are two such models: the **spatial autocorrelation model** (SAC), which combines Eq. (1) and Eq. (5), and the **spatial autoregressive moving average model** (SARMA), which combines Eq. (1) and Eq. (6).

SAC:
$$\begin{aligned} y &= \rho W y + X\beta + \varepsilon \\ \varepsilon &= \lambda W \varepsilon + u \end{aligned} \quad (7)$$

SARMA:
$$\begin{aligned} y &= \rho W y + X\beta + \varepsilon \\ \varepsilon &= \lambda W u + u \end{aligned} \quad (8)$$

Rewriting Eq. (7) yields (LeSage and Pace 2009):

$$y = (I - \rho W)^{-1} X\beta + (I - \rho W)^{-1} (I - \lambda W)^{-1} \varepsilon \quad (9)$$

Note that here, two different spatial multipliers affect the disturbance term. In theory, two different weights matrices can be used for the spatial lag and spatial error components.

Geographically weighted regression (GWR) can be used to model spatial heterogeneity. Routines for GWR are available within R. The reader is directed to Fotheringham et al. (2002) for the technical details.

Estimation and Testing for Spatial Dependence

Moran's I statistic is the most general test statistic used to test for spatial dependence. Two inputs are required: (1) A variable of interest or residuals from a model estimated by OLS, and (2) A weights matrix. The test statistic takes the form (Bivand et al. 2008):

$$I = \frac{n}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}} \cdot \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (10)$$

where w_{ij} represents elements of the spatial weights matrix for sample size n, and $\bar{y}$ is the mean of the variable of interest. A p-value of this test statistic of 0.05 or lower indicates that spatial autocorrelation is present in some form. This test is not specific, however, about the nature of the spatial autocorrelation, whether it takes the form of spatial lag or spatial error.

The Moran test is implemented in a number of different software packages including ArcGIS, GeoDa, and OpenGeoDa. The Getis-Ord test (Bivand et al. 2008) is another general test for spatial autocorrelation. Variants of these tests exist that test for localized spatial correlation, and Local Indicators of Spatial Autocorrelation (LISA) statistics can also identify localized clusters of spatial dependence. See Anselin (2005) for a detailed discussion, with examples demonstrated in GeoDa.

A set of **Lagrange Multiplier (LM) tests** for spatial autocorrelation specifically test for whether a spatial error, spatial lag, or SAC/SARMA model would be most appropriate. Lagrange Multiplier tests follow the following general form (Anselin 2006):

$$LM = S'\mathcal{I}^{-1}S \qquad (11)$$

where S is the score function and $\mathcal{I}$ is the information matrix. Prior to defining these, it is necessary to define the log-likelihood equations for the spatial lag and spatial error models. Estimation is typically done through maximum likelihood (MLE), but more robust methods that account for heteroskedasticity use either two-stage least squares/instrumental variables (Kelejian and Prucha 1998; Bivand 2010) or generalized method of moments (Kelejian and Prucha 2007; Piras 2010). Generalized method of moments (GMM) estimators are particularly useful for large sample sizes. Sample size is typically not an issue with cross-sectional spatial models but can become an issue with spatial panel models (Mills 2010). With large samples, MLE estimators require large memory allocation due to how estimation is performed. These memory allocations can be as large as 27 GB, which is well beyond the typical capacity of today's computers.

The log-likelihood function for the spatial lag model is (Anselin 2006):

$$L(\beta,\Sigma_\theta,\rho;y) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_\theta| + \ln|I-\rho W| - \frac{1}{2}\frac{(y-\rho Wy - X\beta)^2}{\Sigma_\theta} \qquad (12)$$

where θ is the vector of parameters to be estimated, Σ_θ is the variance-covariance matrix of the residuals, and $|I-\rho W|$ is the Jacobian. It is this Jacobian term that can dramatically increase computation time for large samples (Anselin 2006).

The log-likelihood function for the spatial error model does not contain the Jacobian term, thus simplifying calculations (Anselin 2006):

$$L(\beta,\Sigma_\theta;y) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_\theta| + -\frac{1}{2}\frac{(y-X\beta)^2}{\Sigma_\theta} \qquad (13)$$

With these log-likelihood functions in mind, we can now define the score function and the information matrix. The Score function is simply the first derivative of the log-likelihood function:

$$S = \frac{\partial L(\theta)}{\partial \theta} \qquad (14)$$

The information matrix is the expectation of the negative of the second derivative of the log-likelihood function, or the Hessian:

$$\mathcal{I} = -E\left[\frac{\partial^2 L(\theta)}{\partial \theta\, \partial \theta'}\right] \qquad (15)$$

These two elements are used as the building blocks of LM tests, which are used in multiple areas of statistics. Knowledge of the inner workings of these types of functions, and maximum likelihood estimation theory in general, eases the understanding of more complex models such as spatial econometric models. The LM tests for spatial lag and spatial error have the null hypothesis that the spatial lag and spatial error parameters, respectively, are qual to zero. Both tests converge to a chi-squared distribution with one degree of freedom. The LM test for spatial lag is (Anselin 1988):

$$LM_\rho = \left[\frac{\left(WX\hat{\beta}'\right)'M\left(WX\hat{\beta}'\right)}{\hat{\sigma}^2} + tr\left(W'W + WW\right)\right] \xrightarrow{d} \chi^2_{[1]} \qquad (16)$$

where $\hat{\beta}$ is the vector of parameters estimated by OLS, $\hat{\sigma}^2$ is the estimated OLS variance-covariance matrix, tr represents the trace of a matrix (the sum of its diagonals), and M is some fundamental idempotent matrix with

$$M = I - X'(X'X)^{-1} X' \qquad (17)$$

The LM test for spatial lag is:

$$LM_\lambda = \frac{1}{tr(W'W + WW)} \cdot \left[n \cdot \frac{e'We}{e'e}\right]^2 \xrightarrow{d} \chi^2_{[1]} \qquad (18)$$

Robust variants of these tests also exist. These LM tests account for the possible presence of spatial lag for a spatial error model and vice versa. Another test also determines whether a SAC/SARMA model should be used. (Bivand et al. 2008). For these tests, the most appropriate model to use is determined by the p-value: The model whose test has the lowest p-value should be chosen for modeling. For example, if the spatial lag test yields a p-value of 0.03 and the spatial error test yields a p-value of 0.12, then the spatial lag model would be most appropriate.

An alternative specification of the spatial lag model eliminates the use for a log-determinant. Rewriting Eq. (1), we have:

$$\begin{aligned} Sy &= X\beta + \varepsilon \\ S &= (I - \rho W) \end{aligned} \qquad (19)$$

LeSage and Pace (2007) refer to this as the spatial autoregressive (SAR) specification. As observed by the log-likelihood function in Eq. (12), the presence of the log-Jacobian term can considerably complicate calculations. The **matrix exponential spatial specification (MESS)** eliminates the need for this term, such that

$$S(\alpha) = e^{\alpha W} \qquad (20)$$

where α is some real scalar parameter for the infinite series (LeSage and Pace 2007). The resulting log-likelihood is (LeSage and Pace 2007):

$$\begin{aligned} L(\beta,\sigma,\alpha;y) &= -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln(\sigma^2) + \ln\left|S(\alpha)\right| - \frac{1}{2}\frac{\left(S(\alpha)y - X\beta\right)^2}{\sigma^2} \\ L(\alpha;y) &= -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln(\sigma^2) - \frac{n}{2}\ln\left(y'S(\alpha)'MS(\alpha)y\right) \end{aligned} \qquad (21)$$

Note that the second line represents a **concentrated log-likelihood function.** It is typical for the derivation of the log-likelihood functions of complex statistical models to involve simplification of the initial log-likelihood function. This is typically done through a combination of substitution and using the convenient properties of the calculations being used, which in this case is the exponential function. The shortcuts and assumptions used to concentrate the log-likelihood in Eq. (21) are discussed in detail in LeSage and Pace (2007). The spatial lag parameter can be inferred from:

$$\rho = 1 - e^{\alpha} \qquad (22)$$

Interpretation of Spatial Econometric Models

For spatial lag models, SDM models, or SAC/SARMA models, interpretations cannot be fully drawn from examining the estimated coefficients. Marginal effects, a commonly used measure in econometrics and statistics, are used to identify what changes take place with a *ceteris paribus* one-unit change in some independent variable. These marginal effects are particularly useful with spatial models. LeSage and Pace (2009) identified three types of marginal effects for these models:

(1) **Total effects**, or the total effect on the dependent variable as a result of the one-unit change in some independent variable,

(2) **Direct effects**, the effect on the dependent variable for the spatial unit of interest.

(3) **Indirect effects**, the *net* effect on the dependent variables of the neighbors (usually the first-order neighbors) of the spatial unit of interest.

Consider this simple example: For an employment model in which the independent variable of interest is income, the direct effects have a value of 0.15, the indirect effects have a value of 0.08, and the total effects, therefore, have a value of 0.23. A two nearest-neighbors weights matrix is used, which means that each spatial unit has exactly two neighbors. Assume the spatial unit of interest is Tippecanoe County and the two nearest neighboring counties are White County and Clinton County. According to these marginal effects, a one-unit increase in income will increase employment in Tippecanoe County by 0.15 percent and by an *average* of 0.04 percent in White and Clinton Counties. This example illustrates how the choice of the weights matrix could have a substantial impact on model interpretation.

Spatial Panel Models: In Brief

The spatial models described in the previous section are cross-sectional models. That is, that is, they are valid only for data taken at one point in time. Using these models with longitudinal, or panel data, which consist of observations for several cross-sectional units observed over a period of time, is akin to estimating a pooled model, in which cross-sectional and time-specific variance are not accounted for. While this can be done, it comes with the same risks as estimating a panel data model using ordinary least squares (OLS). **Spatial panel models**, which account for cross-sectional and time-specific variance as well as spatial and temporal correlation, have been developed by Elhorst (2003, 2010), Baltagi et al. (2007), Kapoor et al. (2007), and others. The one-way random-effects model, which captures within-group (cross-sectional) variance in the intercept term, can be estimated with the following log-likelihood function (Elhorst 2010):

$$LL = -\frac{NT}{2}\log\left(2\pi\sigma^2\right) + T\log\left|I_N - \rho W\right| - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(y_{it}^* - \rho\left[\sum_{j=1}^{N} w_{ij}y_{jt}\right] - x_{it}^*\beta\right)^2 \quad (23)$$

where

$$y_{it}^* = y_{it} - (1-\theta)\frac{1}{T}\sum_{i=1}^{t} y_{it} \text{ and } x_{it}^* = x_{it} - (1-\theta)\frac{1}{T}\sum_{i=1}^{t} x_{it} \quad (24)$$

where θ represents the share of the cross-sectional variance of the overall variance with $0 \le \theta \le 1$ and

$$\theta = \sqrt{\frac{\sigma^2}{\left(T\sigma_\mu^2 + \sigma^2\right)}} \quad (25)$$

where σ_{μ}^{2} represents the between-group variance and σ^{2} represents the residual, or within-group variance. The corresponding asymptotic variance-covariance matrix is (Elhorst 2010):

$$Asy.Var\left(\beta,\theta,\sigma^{2}\right)=\begin{bmatrix} \frac{X^{*}{}'X^{*}}{\sigma^{2}} & 0 & 0 \\ 0 & N\left(1+\frac{1}{\theta^{2}}\right) & -\frac{N}{\sigma^{2}} \\ 0 & -\frac{N}{\sigma^{2}} & \frac{NT}{2\sigma^{4}} \end{bmatrix}^{-1} \tag{26}$$

The estimation techniques and, particularly, the computing time for spatial panel models can be considerably more complex than the cross-sectional models. Evaluation of Eq. (23) all require estimation of the log-Jacobian using maximum likelihood methods. For example, the computational requirements of a 32,000 by 32,000 weights matrix (3200 ZIP Codes for ten years) represent a mountain summit that simply cannot be reached without the aid of supercomputer clusters. Spatial lag panel models with random region effects were estimated for the study area using MATLAB-based routines written by J. Paul Elhorst (2010) and James LeSage which use a Monte-Carlo approximation of the spatial weights matrix to reduce computation time (LeSage and Pace 2009) by a factor of over 3000, compared to using full maximum likelihood-methods. Even with these computationally efficient methods, however, estimation of models that simultaneously capture spatial correlation, random region effects, and serial correlation remains difficult without a considerably larger sample size. The reader is referred to the aforementioned studies for the technical details.

Spatial Count Models

The models described in previous sections pertain to normally distributed, continuous data, as spatial econometric models have their roots in the field of economics, where such data are common. Data in other disciplines, such as crash data in transportation, is often discrete and follows Poisson or negative binomial distributions. Estimation of such models cannot be done easily using maximum likelihood methods. Often, Bayesian methods using Markov Chain Monte Carlo (MCMC) routines must be used for model estimation. Poisson-lognormal, Poisson-gamma, and Poisson-lognormal with conditionally autoregressive (CAR) were used in Wang et al.'s study of how traffic congestion impacts road accidents in England (2009). Bayesian hierarchical models were used by Quddus (2008) to identify regional contributing factors to traffic crashes. Wakefield (2007) used Bayesian methods to model spatial dependence in male lip cancer incidence data. Griffith (2006) explored the differences between using Winsorization and spatial filtering with spatially correlated Poisson models. Gschlößl and Czado (2008) predicted the number of insurance claims data from a German car insurance company using a combination of Poisson, negative binomial, and zero-inflated models that incorporated spatial effects using a Gaussian CAR prior. Spatial CAR models were used to model bird point count surveys by Webster et al. (2008), and zero-inflated spatial CAR Poisson models were used by Agarwal et al. (2002) to model the pixel-based spatial distribution of the terrestrial isopod, *Hemilepistus reaumuri*. Other estimation methods, such as quasi-likelihood (QL), have been explored by Lin (2010). The reader is referred to those studies, as well as textbooks by Haining (2003) and Lawson (2009) for additional details.

Software for Spatial Econometric Analysis

Several software packages can perform spatial econometric-related tests and estimate spatial econometric models. **GeoDa and OpenGeoDa** (Anselin 2010), available for free at the GeoDa Center, are used for exploratory data analysis and for preliminary tests for spatial correlation. They require the use of shapefiles, which can be created and exported by any number of GIS-based software packages. Anselin (2005) provides details on using the program. PySAL, a library for spatial analysis in Python, was also recently released by the center.

LeSage (2010) has published a **MATLAB**-based econometrics toolbox, which can estimate many spatial econometric models, both using maximum likelihood and Bayesian methods. It also contains code for estimating spatial panel models, contributed primarily by Elhorst (2010). Documentation for the toolbox includes a special section for the spatial econometrics commands (LeSage 1999) as well as learning materials related to spatial econometrics (LeSage 1998).

The majority of the spatial econometric models in the literature can be estimated using packages in the open-source R statistical package (Bivand 2010; Bivand et al. 2008; Millo and Piras 2010; Piras 2010). Most (but not all) of the functionality offered by GeoDa and LeSage's spatial econometrics toolbox. Packages for geographically weighted regression, spatial panel, and spatial count data models are also available in R, and spatial logit and probit models are available in MATLAB, as well as Bayesian methods for cross-sectional spatial econometric models for both continuous and discrete distributions.The tools available in R will be demonstrated in the following section.

An Application of Spatial Econometrics: Census Data in Tippecanoe County

This example will demonstrate several add-on packages for R (see Purdue ITE 2010 for information on how to install and use add-on packages in R). These packages include:

(1) spdep (Spatial econometric models) (Bivand 2010)
(2) sphet (Spatial econometric models that account for heteroskedasticity) (Piras 2010)
(3) UScensus2000 (Census 2000 demographic data and shapefiles) (Almquist 2010)
(4) sp (Classes/methods for spatial data) (Pebesma and Bivand 2005; Bivand et al. 2008)
(5) maptools (Tools for reading/manipulating spatial data) (Lewin-Koh and Bivand 2010)

A number of add-on packages exist for R that can download and extract US Census 2000 Summary File 1 shapefiles at the county, Census Designated Place (CDP), Metropolitan Statistical Area (MSA), Census block, and Census block group levels. In this example, SF1 demographic data for Tippecanoe County, IN data will be downloaded and displayed. While R is strong with displaying graphics, using the multitude of possible options can be difficult to learn. Helper functions for displaying attractive maps have been written by the package's authors to facilitate the process.

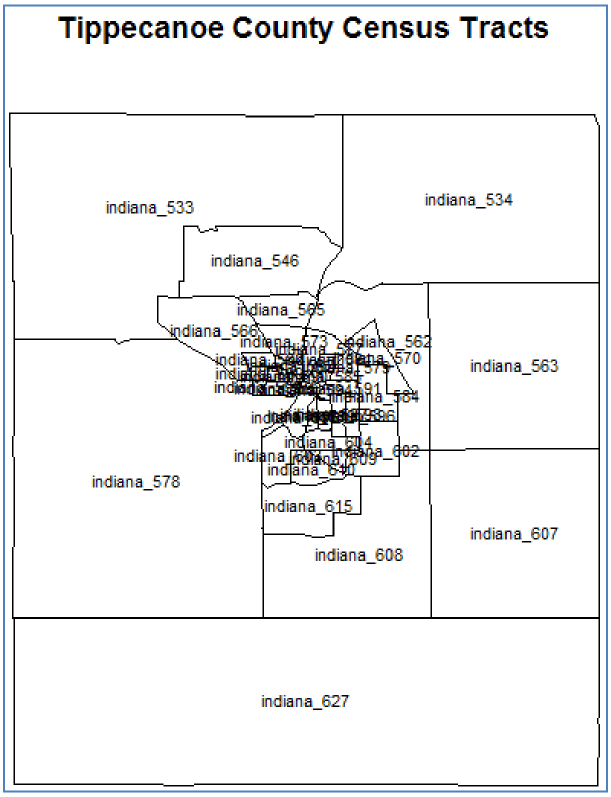


Figure 4: Census Tract map of Tippecanoe County, with default labels.

County-level data can be obtained using the COUNTY command. The sample code below will extract Census tract-level data for Tippecanoe County, Indiana and then display a map of the county (Figure 4 shows the Census tracts for Tippecanoe County, Indiana, created using the syntax below).

Syntax
```
tippecanoe<-county(name="Tippecanoe", state="indiana",level=c("tract"))
plot(tippecanoe)
text(coordinates(tippecanoe), labels=row.names(tippecanoe),cex=0.7)
title("Tippecanoe County Census Tracts")
```

The CEX command controls the font size. The TEXT and TITLE commands are used to add labels and a title to the map. Trial-and-error is often needed to properly display labels.

The extracted data can be exported to an ESRI shapefile using the WRITEPOLYSHAPE command from the MAPTOOLS package.

Syntax
```
writePolyShape(tippecanoe,"F:/lots_of_data/tippecanoe.shp")
```

The attribute table can be obtained using the ATTR command.

Syntax
```
tip_data<-attr(tippecanoe,"data")
```

A primary variable of interest is population. The SPPLOT command can be used to display the distribution of population in the county. Note that the default color scheme tends to repeat colors and is counterintuitive. The COL.REGIONS option allows different color schemes to be used. The BPY.COLORS set, from the SP package, is best suited for black-and-white printing. The maps below in Figure 5 show population data using the default color scheme (left) and the black-and-white-friendly color scheme (right). The number of colors to use (20 in this example) may need to be changed to avoid repeated colors.

Syntax
```
spplot(tippecanoe,"pop2000",col.regions=bpy.colors(20))
spplot(tippecanoe,"pop2000",col.regions=heat.colors(20))
spplot(tippecanoe,"pop2000",col.regions=cm.colors(20))
spplot(tippecanoe,"pop2000",col.regions=topo.colors(20))
spplot(tippecanoe,"pop2000",col.regions=terrain.colors(20))
spplot(tippecanoe,"pop2000",col.regions=rainbow(20))
spplot(tippecanoe,"pop2000")
```

Output

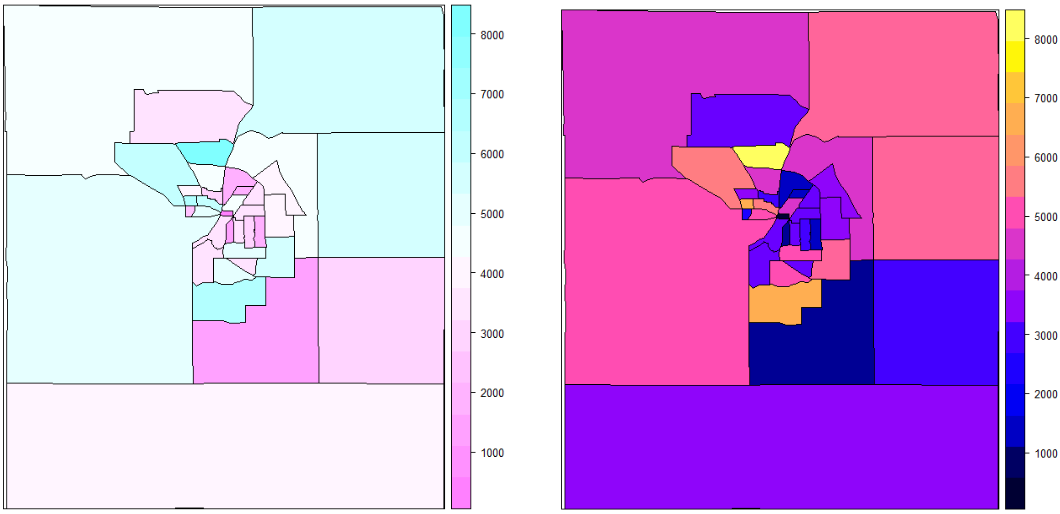


Figure 5: Displayed graphics using the default coloring scheme (left) and the bpy.colors black-and-white-friendly color scheme (right).

Alternatively, maps can be created using the PLOT command (which uses a special method for SpatialPolygons, the class created using these commands). The syntax below shows a helper function written by the USCENSUS2000 package authors to simplify the process of displaying colors. This code is then used to display a population map of Tippecanoe County (see Figure 6) with a legend and using the helper function as an argument for COL.

Syntax

```
#  Helper functions and sample plotting code from
#  Zack W. Almquist (2010). UScensus2000: US Census 2000 Suite of
#  R Packages. R package version 0.07.
#  http://CRAN.R-project.org/package=UScensus2000
############################################
## Helper function for handling coloring of the map
############################################
color.map<- function(x,dem,y=NULL){
       l.poly<-length(x@polygons)
       dem.num<- cut(dem ,breaks=unique(ceiling(quantile(dem))),dig.lab = 6)
       dem.num[which(is.na(dem.num)==TRUE)]<-levels(dem.num)[1]
       l.uc<-length(table(dem.num))
if(is.null(y)){
       ##commented out, but creates different color schemes
       ## using runif, may take a couple times to get a good color scheme.
       ##col.heat<-rgb( runif(l.uc,0,1), runif(l.uc,0,1) , runif(l.uc,0,1) )
       col.heat<-heat.colors(16)[c(14,8,4,1)] ##fixed set of four colors
}else{
       col.heat<-y
       }
dem.col<-cbind(col.heat,names(table(dem.num)))
colors.dem<-vector(length=l.poly)
for(i in 1:l.uc){
       colors.dem[which(dem.num==dem.col[i,2])]<-dem.col[i,1]
       }
```

```
out<-list(colors=colors.dem,dem.cut=dem.col[,2],table.colors=dem.col[,1])
return(out)
}
############################################
## End Helper function for handling coloring of the map
############################################

colors.use<-color.map(tippecanoe,tippecanoe$pop2000)
plot(tippecanoe,col=colors.use$colors)

title(main="Census Tracts of\n Tippecanoe County IN, 2000",
sub="Quantiles (equal frequency)")
legend("bottomright",legend=colors.use$dem.cut,fill=colors.use$table.colors,
bty="o",title="Population Count",bg="white")
text(coordinates(tippecanoe),labels=row.names(tippecanoe),cex=0.7)

plot(tippecanoe,col=bpy.colors(5))
```

Output

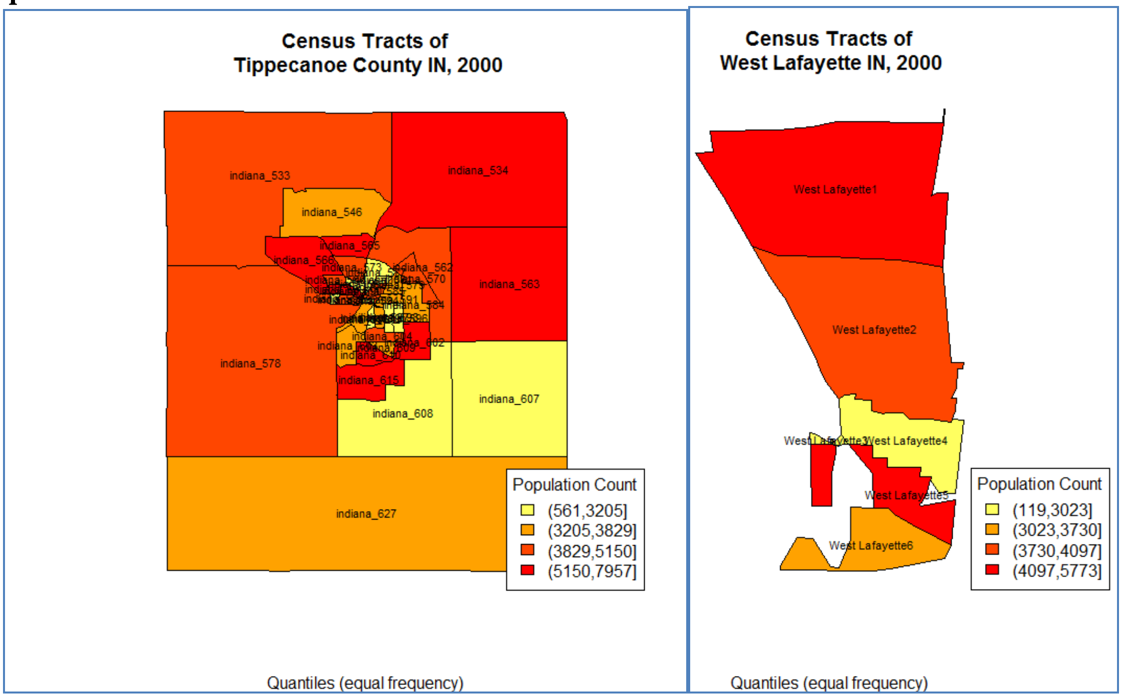


Figure 6: Color-theme map for Tippecanoe County (left) and West Lafayette, Indiana (right).

The POLY.CLIPPER command can be used to extract community-level, or more formally, Census Designated Place (CDP) demographic data. In this example, demographic data for Census Tracts located in the city of West Lafayette will be extracted and then plotted using the PLOT command (see Figure 5). The BB.EPSILON parameter can be changed if the "clipped" data do not match the actual data.

Syntax

```
WestLafayette<-poly.clipper(name="West Lafayette",state="Indiana",
level=c("tract"),bb.epsilon=0.01)
```

```
colors.use2<-color.map(WestLafayette,WestLafayette$pop2000)

plot(WestLafayette,col=colors.use2$colors)

title(main="Census Tracts of\n West Lafayette IN, 2000",
sub="Quantiles (equal frequency)")
legend("bottomright",legend=colors.use2$dem.cut,fill=colors.use2$table.colors,
bty="o",title="Population Count",bg="white")
text(coordinates(WestLafayette),labels=row.names(WestLafayette),cex=0.7)

# Compare to
http://www2.census.gov/plmap/pl_trt/st18_Indiana/c18157_Tippecanoe/CT18157_A01.pdf
```

Additional SF1 demographic data can be downloaded using the USCENSUS2000ADD package for each state. This requires knowledge of what each variable in the master SF1 file represents (http://www.census.gov/prod/cen2000/doc/sf1.pdf). For example, Census tract-level population for different races, such as White alone (P003003), Black or African American alone (P003004), American Indian and Alaska Native alone (P003005), Asian alone (P003006), Native Hawaiian and Other Pacific Islander alone (P003007), or some other race alone (P003008) for the state of Indiana can be downloaded using the following syntax.

Syntax

```
add_me <- c("P003003","P003004","P003005","P003006","P003007","P003008")
IN_new <- demographics.add(dem=add_me,state="indiana",level=c("tract"))
```

One concern with working with socioeconomic data is the presence of spatial autocorrelation. For example, the opening of the Subaru plant in Lafayette affected nearby surrounding areas, and over time, the spatial distribution of employment as well as traffic patterns began to change (the plant can be seen as a large traffic generator). Using the Tippecanoe County data set from the US Census section, changes in one Census tract may have effects on adjacent Census tracts.

A numerical representation of a Census tract's "neighbors" can take the form of a spatial weights matrix. The spatial weights matrix can changed based on certain criteria. For example, a Census tract may be a neighbor if its border is contiguous to the Census tract of interest. Alternatively, a Census tract may be a neighbor if it is located within 10 miles of the Census tract of interest. The specification of the weights matrix may yield different findings, so great care should be taken when specifying a weights matrix.

In this example, weights matrices will be defined using three criteria: queen contiguity, a distance threshold of 0.3 degrees (on a lat-long scale), and two nearest neighbors. This is accomplished using the SPDEP package and the POLY2NB, DNEARNEIGH, and KNEARNEIGH commands, respectively. The NB2LISTW command converts the resulting "neighbor lists" to formal weights matrices. The SUMMARY command can be used to display information on the percentage of nonzero entries in the weights matrix and the most and least connected regions.

Syntax

```
tipxy <- coordinates(tippecanoe)

tip.queen <- poly2nb(tippecanoe,queen=TRUE)
tip.queenw <- nb2listw(tip.queen,style="W")
summary(tip.queenw)

tip.d <- dnearneigh(tipxy,3,10, longlat=TRUE)
tip.dw <- nb2listw(tip.d,style="W")
summary(tip.dw)

tip.nn2 <- knn2nb(knearneigh(tipxy, k=2, longlat=TRUE))
tip.nn2w <- nb2listw(tip.nn2,style="W")
summary(tip.nn2w)
```

Output

```
> summary(tip.queenw)
Characteristics of weights list object:
Neighbour list object:
Number of regions: 37
Number of nonzero links: 216
Percentage nonzero weights: 15.77794
Average number of links: 5.837838
Link number distribution:

 3  4  5  6  7  8  9 10 11
 4  7  9  6  2  3  4  1  1
4 least connected regions:
indiana_581 indiana_599 indiana_609 indiana_627 with 3 links
1 most connected region:
indiana_562 with 11 links

Weights style: W
Weights constants summary:
   n   nn S0       S1       S2
W 37 1369 37 13.36039 155.3556
>
> tip.d <- dnearneigh(tipxy,3,10, longlat=TRUE)
> tip.dw <- nb2listw(tip.d,style="W")
> summary(tip.dw)
Characteristics of weights list object:
Neighbour list object:
Number of regions: 37
Number of nonzero links: 686
Percentage nonzero weights: 50.10957
Average number of links: 18.54054
Link number distribution:

 1  2  3  7  8 15 16 17 18 19 20 21 23 24 25 26 27 28
 1  1  2  1  1  2  1  1  2  6  3  4  2  2  3  1  2  2
1 least connected region:
37 with 1 link
2 most connected regions:
6 30 with 28 links

Weights style: W
Weights constants summary:
   n   nn S0       S1       S2
W 37 1369 37 6.004295 155.7006
>
> tip.nn2 <- knn2nb(knearneigh(tipxy, k=2, longlat=TRUE))
> tip.nn2w <- nb2listw(tip.nn2,style="W")
```

```
> summary(tip.nn2w)
Characteristics of weights list object:
Neighbour list object:
Number of regions: 37
Number of nonzero links: 74
Percentage nonzero weights: 5.405405
Average number of links: 2
Non-symmetric neighbours list
Link number distribution:

 2
37
37 least connected regions:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 with 2 links
37 most connected regions:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 with 2 links

Weights style: W
Weights constants summary:
   n   nn S0   S1  S2
W 37 1369 37 26.5 164
```

Plots of Tippecanoe County with each of the three types of weights matrices are shown below.

Syntax

```
plot(tippecanoe,col=colors.use$colors)
plot(tip.queenw,tipxy,add=T,arrows=T)
title("Tippecanoe County Census Tracts\nFirst-Order Queen Contiguity Weights Matrix")
legend("bottomright",legend=colors.use$dem.cut,fill=colors.use$table.colors,
bty="o",title="Population Count",bg="white")

plot(tippecanoe,col=colors.use$colors)
plot(tip.dw,tipxy,add=T,arrows=T)
title("Tippecanoe County Census Tracts\nDistance Threshold Weights Matrix")
legend("bottomright",legend=colors.use$dem.cut,fill=colors.use$table.colors,
bty="o",title="Population Count",bg="white")

plot(tippecanoe,col=colors.use$colors)
plot(tip.nn2w,tipxy,add=T,arrows=T)
title("Tippecanoe County Census Tracts\n2 Nearest Neighbors Weights Matrix")
legend("bottomright",legend=colors.use$dem.cut,fill=colors.use$table.colors,
bty="o",title="Population Count",bg="white")
```

Output

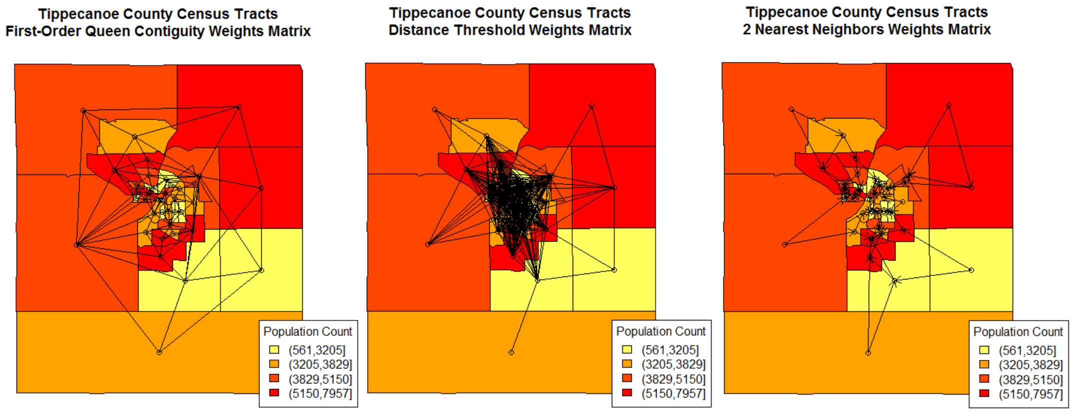


Figure 1 (Repeated): Weights matrices for Tippecanoe County Census tracts

Moran's I statistic can be used to test for the presence of spatial correlation. The test is general; it will only indicate whether spatial correlation is present for a given variable and weights matrix but not the type and nature of the spatial correlation. A local variant of this test can be used to identify localized effects. The global Moran test can be called using the MORAN.TEST command, and the local Moran test can be called using the LOCALMORAN command. With these commands and other commands in the SPDEP package, two required inputs are (1) a variable with associated data frame and (2) a weights matrix. The following example will conduct global and local Moran tests for all three weights matrices used for median age.

Syntax

```
moran.test(tip_data$med.age, listw=tip.queenw, randomisation=TRUE,
alternative="two.sided")
localmoran(tip_data$med.age,listw=tip.queenw)

moran.test(tip_data$med.age, listw=tip.dw, randomisation=TRUE,alternative="two.sided")
localmoran(tip_data$med.age,listw=tip.dw)

moran.test(tip_data$med.age, listw=tip.nn2w, randomisation=TRUE,
alternative="two.sided")
localmoran(tip_data$med.age,listw=tip.nn2w)
```

Output

```
> moran.test(tip_data$med.age, listw=tip.queenw,
randomisation=TRUE,alternative="two.sided")

        Moran's I test under randomisation

data:  tip_data$med.age
weights: tip.queenw

Moran I statistic standard deviate = 4.5862, p-value = 4.514e-06
alternative hypothesis: two.sided
sample estimates:
Moran I statistic       Expectation          Variance
       0.38626142       -0.02777778        0.00815043

> localmoran(tip_data$med.age,listw=tip.queenw)
```

```
               Ii        E.Ii     Var.Ii        Z.Ii    Pr(z > 0)
 [1,]  0.206493444 -0.02777778 0.21853933  0.50113431 3.081383e-01
 [2,]  0.707883360 -0.02777778 0.21853933  1.57366763 5.778216e-02
 [3,]  0.120428328 -0.02777778 0.16966535  0.35980692 3.594958e-01
 [4,]  0.626638276 -0.02777778 0.06303122  2.60661022 4.572169e-03
 [5,]  0.724343218 -0.02777778 0.21853933  1.60887724 5.382159e-02
 [6,]  0.265597465 -0.02777778 0.16966535  0.71224085 2.381578e-01
 [7,]  0.076769291 -0.02777778 0.09635439  0.33680313 3.681327e-01
 [8,]  0.803733603 -0.02777778 0.21853933  1.77870282 3.764425e-02
 [9,] -0.385436885 -0.02777778 0.13708270 -0.96600186 8.329784e-01
[10,]  0.029992909 -0.02777778 0.08277828  0.20079335 4.204301e-01
[11,] -0.491006634 -0.02777778 0.07191740 -1.72734317 9.579470e-01
[12,]  0.163722209 -0.02777778 0.13708270  0.51722252 3.025004e-01
[13,]  2.198026913 -0.02777778 0.16966535  5.40369056 3.264177e-08
[14,]  0.004998568 -0.02777778 0.29999596  0.05984155 4.761409e-01
[15,]  1.226686403 -0.02777778 0.16966535  3.04552159 1.161385e-03
[16,] -0.263560906 -0.02777778 0.13708270 -0.63682690 7.378812e-01
[17,]  0.166091741 -0.02777778 0.08277828  0.67383154 2.502092e-01
[18,]  2.111370907 -0.02777778 0.13708270  5.77762894 3.788033e-09
[19,]  2.113120264 -0.02777778 0.09635439  6.89700026 2.655604e-12
[20,]  0.087337229 -0.02777778 0.13708270  0.31091424 3.779329e-01
[21,]  0.548315149 -0.02777778 0.21853933  1.23233204 1.089125e-01
[22,]  1.241708947 -0.02777778 0.13708270  3.42875803 3.031749e-04
[23,]  0.196717580 -0.02777778 0.16966535  0.54501792 2.928706e-01
[24,]  0.094752566 -0.02777778 0.08277828  0.42587824 3.350983e-01
[25,]  0.043032952 -0.02777778 0.16966535  0.17191053 4.317539e-01
[26,]  0.001177819 -0.02777778 0.11380938  0.08583087 4.658004e-01
[27,]  0.333825161 -0.02777778 0.21853933  0.77351216 2.196097e-01
[28,]  0.002812258 -0.02777778 0.29999596  0.05584988 4.777307e-01
[29,] -0.233174902 -0.02777778 0.09635439 -0.66169616 7.459170e-01
[30,] -0.167675461 -0.02777778 0.16966535 -0.33963617 6.329347e-01
[31,] -0.013353467 -0.02777778 0.08277828  0.05013452 4.800076e-01
[32,]  0.570743355 -0.02777778 0.21853933  1.28030867 1.002183e-01
[33,]  0.550501459 -0.02777778 0.11380938  1.71414902 4.325068e-02
[34,] -0.037338030 -0.02777778 0.29999596 -0.01745467 5.069631e-01
[35,]  0.112424210 -0.02777778 0.16966535  0.34037495 3.667871e-01
[36,] -0.001067516 -0.02777778 0.16966535  0.06484576 4.741484e-01
[37,]  0.555040832 -0.02777778 0.29999596  1.06408349 1.436454e-01
attr(,"call")
localmoran(x = tip_data$med.age, listw = tip.queenw)
attr(,"class")
[1] "localmoran" "matrix"
>
> moran.test(tip_data$med.age,listw=tip.dw,randomisation=TRUE,alternative="two.sided")

        Moran's I test under randomisation

data:  tip_data$med.age
weights: tip.dw

Moran I statistic standard deviate = -1.1376, p-value = 0.2553
alternative hypothesis: two.sided
sample estimates:
Moran I statistic       Expectation          Variance
     -0.087374415      -0.027777778       0.002744703

> localmoran(tip_data$med.age,listw=tip.dw)
                Ii        E.Ii      Var.Ii        Z.Ii  Pr(z > 0)
 [1,]  0.0559321813 -0.02777778 0.299995961  0.15283381 0.43926467
 [2,]  1.0606898845 -0.02777778 0.462909222  1.59980708 0.05482069
 [3,] -0.0484733887 -0.02777778 0.020716086 -0.14378858 0.55716629
 [4,] -0.2750766049 -0.02777778 0.010372386 -2.42819026 0.99241281
 [5,]  0.4515709284 -0.02777778 0.096354385  1.54424362 0.06126466
```

```
 [6,] -0.1489741719 -0.02777778 0.009079424 -1.27192220 0.89829962
 [7,]  0.0551397508 -0.02777778 0.011764807  0.76445854 0.22229702
 [8,] -0.6473311393 -0.02777778 0.013268622 -5.37855735 0.99999996
 [9,]  0.1442183466 -0.02777778 0.023043418  1.13303993 0.12859874
[10,] -0.0519429869 -0.02777778 0.020716086 -0.16789459 0.56666690
[11,] -1.3745392667 -0.02777778 0.113809377 -3.99210233 0.99996725
[12,] -0.4257045166 -0.02777778 0.025615732 -2.48627826 0.99354565
[13,] -0.4185939909 -0.02777778 0.014897755 -3.20193242 0.99931745
[14,] -0.0219231542 -0.02777778 0.023043418  0.03856786 0.48461746
[15,] -0.4351306628 -0.02777778 0.028473860 -2.41405758 0.99211202
[16,]  0.1257053491 -0.02777778 0.014897755  1.25747751 0.10429039
[17,]  0.2266002394 -0.02777778 0.039334744  1.28260053 0.09981603
[18,] -0.4762385687 -0.02777778 0.016668551 -3.47356598 0.99974320
[19,] -0.5102774424 -0.02777778 0.031668238 -2.71134679 0.99664947
[20,] -0.1450328165 -0.02777778 0.025615732 -0.73261891 0.76810455
[21,] -0.1841051700 -0.02777778 0.016668551 -1.21083832 0.88702131
[22,] -0.4858953297 -0.02777778 0.025615732 -2.86235530 0.99789747
[23,] -0.0234759593 -0.02777778 0.039334744  0.02169022 0.49134753
[24,] -0.0391713805 -0.02777778 0.023043418 -0.07505638 0.52991507
[25,] -0.2036506956 -0.02777778 0.025615732 -1.09886813 0.86408721
[26,] -0.2789170040 -0.02777778 0.025615732 -1.56913808 0.94169212
[27,] -0.2060616718 -0.02777778 0.028473860 -1.05654729 0.85464088
[28,] -0.0009528373 -0.02777778 0.025615732  0.16760439 0.43344726
[29,]  0.1018165577 -0.02777778 0.010372386  1.27246743 0.10160354
[30,]  0.0690149708 -0.02777778 0.009079424  1.01581278 0.15485927
[31,] -0.1788089668 -0.02777778 0.020716086 -1.04933170 0.85298726
[32,]  0.2546825894 -0.02777778 0.299995961  0.51570319 0.30303086
[33,]  0.0863459778 -0.02777778 0.035261912  0.60774742 0.27167750
[34,] -0.0509226207 -0.02777778 0.020716086 -0.16080531 0.56387663
[35,]  0.0845357872 -0.02777778 0.013268622  0.97503296 0.16477195
[36,]  0.0230185801 -0.02777778 0.013268622  0.44098078 0.32961346
[37,]  0.6590758408 -0.02777778 0.951649005  0.70408613 0.24068957
attr(,"call")
localmoran(x = tip_data$med.age, listw = tip.dw)
attr(,"class")
[1] "localmoran" "matrix"
>
>
moran.test(tip_data$med.age,listw=tip.nn2w,randomisation=TRUE,alternative="two.sided")

        Moran's I test under randomisation

data:  tip_data$med.age
weights: tip.nn2w

Moran I statistic standard deviate = 4.5123, p-value = 6.414e-06
alternative hypothesis: two.sided
sample estimates:
Moran I statistic       Expectation          Variance
       0.57121056       -0.02777778        0.01762179

> localmoran(tip_data$med.age,listw=tip.nn2w)
                Ii         E.Ii    Var.Ii        Z.Ii    Pr(z > 0)
 [1,] -0.028940190 -0.02777778 0.4629092 -0.00170849 5.006816e-01
 [2,]  1.060689884 -0.02777778 0.4629092  1.59980708 5.482069e-02
 [3,]  0.024712415 -0.02777778 0.4629092  0.07714899 4.692525e-01
 [4,]  1.378186574 -0.02777778 0.4629092  2.06645710 1.939267e-02
 [5,]  1.023406813 -0.02777778 0.4629092  1.54500920 6.117202e-02
 [6,]  0.397557026 -0.02777778 0.4629092  0.62514823 2.659369e-01
 [7,]  0.792357379 -0.02777778 0.4629092  1.20541756 1.140211e-01
 [8,]  1.697665766 -0.02777778 0.4629092  2.53602095 5.606001e-03
 [9,] -1.966497277 -0.02777778 0.4629092 -2.84948951 9.978105e-01
[10,] -0.126939687 -0.02777778 0.4629092 -0.14574611 5.579391e-01
```

```
[11,] -0.959960881 -0.02777778 0.4629092 -1.37010329 9.146727e-01
[12,]  0.161245626 -0.02777778 0.4629092  0.27782266 3.905742e-01
[13,]  3.097872304 -0.02777778 0.4629092  4.59401533 2.173987e-06
[14,]  0.044523534 -0.02777778 0.4629092  0.10626696 4.576853e-01
[15,]  3.114686510 -0.02777778 0.4629092  4.61872850 1.930493e-06
[16,] -0.195539797 -0.02777778 0.4629092 -0.24657312 5.973807e-01
[17,]  0.079110774 -0.02777778 0.4629092  0.15710256 4.375820e-01
[18,]  3.122996341 -0.02777778 0.4629092  4.63094212 1.820028e-06
[19,]  3.280327291 -0.02777778 0.4629092  4.86218387 5.804885e-07
[20,] -0.018726602 -0.02777778 0.4629092  0.01330323 4.946929e-01
[21,]  1.358565368 -0.02777778 0.4629092  2.03761826 2.079406e-02
[22,]  2.290767222 -0.02777778 0.4629092  3.40774911 3.275055e-04
[23,]  0.115314586 -0.02777778 0.4629092  0.21031417 4.167112e-01
[24,] -0.132312822 -0.02777778 0.4629092 -0.15364343 5.610546e-01
[25,]  0.488951272 -0.02777778 0.4629092  0.75947759 2.237835e-01
[26,]  0.257526645 -0.02777778 0.4629092  0.41933449 3.374858e-01
[27,]  0.426145810 -0.02777778 0.4629092  0.66716743 2.523326e-01
[28,]  0.002270312 -0.02777778 0.4629092  0.04416406 4.823868e-01
[29,] -0.296100797 -0.02777778 0.4629092 -0.39437558 6.533481e-01
[30,]  0.405519456 -0.02777778 0.4629092  0.63685124 2.621109e-01
[31,]  0.342111832 -0.02777778 0.4629092  0.54365604 2.933391e-01
[32,]  0.149665594 -0.02777778 0.4629092  0.26080257 3.971224e-01
[33,] -0.372380785 -0.02777778 0.4629092 -0.50649032 6.937438e-01
[34,]  0.005934234 -0.02777778 0.4629092  0.04954921 4.802408e-01
[35,] -0.204757503 -0.02777778 0.4629092 -0.26012111 6.026148e-01
[36,]  0.018681534 -0.02777778 0.4629092  0.06828493 4.727794e-01
[37,]  0.300155060 -0.02777778 0.4629092  0.48198885 3.149069e-01
attr(,"call")
localmoran(x = tip_data$med.age, listw = tip.nn2w)
attr(,"class")
[1] "localmoran" "matrix"
```

The tests indicate that spatial correlation is statistically significant under the queen contiguity and two nearest-neighbors weights matrices as noted by the p-values. Localized effects are particularly noticeable using queen contiguity. Suppose a model is to be estimated to predict median age at the census tract level and that this model will be a function of (1) The number of households with married couples without children, (2) The number of vacant households, and (3) The number of residents between the ages of 18-21. SPDEP includes a suite of statistical tests to be used on OLS residuals to test for different types of spatial correlation. These types are:

(1) **Spatial lag:** Spatial correlation is present in the dependent variable. The dependent variable is thus a function of its spatial lag - for example, median age in one Census tract could be affected by median age in the neighbors of that Census tract. This is a multidimensional analogue of temporal autocorrelation - in autoregressive AR(1) correlation for example, the dependent variable is a function of the value of the dependent variable from a previous time period.

(2) **Spatial error:** Spatial correlation is present within the disturbance term of the model. The closest analogue in time-series analysis is the first-order moving average MA(1).

Spatial ARAR models simultaneously model spatial lag and spatial error. These series of tests can be called using the LM.LMTESTS command. In most cases, the "all" option should be used with the TEST statement. These tests require an OLS model estimated using the

LM command as an input. In this example, the aforementioned model will be estimated using OLS. The suite of tests will then be performed on the residuals from the OLS model using the two nearest-neighbors weights matrix.

Syntax

```
t1 <- med.age ~ 1 + marhh.no.c + hh.vacant + age.18.21

tt1 <- lm(t1, data=tip_data)
summary(tt1)

lm.LMtests(tt1, tip.nn2w, test="all")
```

Output

```
> t1 <- med.age ~ 1 + marhh.no.c + hh.vacant + age.18.21
>
> tt1 <- lm(t1, data=tip_data)
> summary(tt1)

Call:
lm(formula = t1, data = tip_data)

Residuals:
    Min      1Q  Median      3Q     Max
-6.5697 -2.2793  0.6591  2.2119  5.8904

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 30.3273875  1.3892142  21.831  < 2e-16 ***
marhh.no.c   0.0127016  0.0028821   4.407 0.000105 ***
hh.vacant   -0.0249069  0.0093076  -2.676 0.011510 *
age.18.21   -0.0021898  0.0004684  -4.675 4.79e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.147 on 33 degrees of freedom
Multiple R-squared: 0.6973,     Adjusted R-squared: 0.6698
F-statistic: 25.34 on 3 and 33 DF,  p-value: 1.082e-08

>
> lm.LMtests(tt1, tip.nn2w, test="all")

        Lagrange multiplier diagnostics for spatial dependence

data:
model: lm(formula = t1, data = tip_data)
weights: tip.nn2w

LMerr = 0.369, df = 1, p-value = 0.5436


        Lagrange multiplier diagnostics for spatial dependence

data:
model: lm(formula = t1, data = tip_data)
weights: tip.nn2w

LMlag = 4.7202, df = 1, p-value = 0.02981


        Lagrange multiplier diagnostics for spatial dependence
```

```
data:
model: lm(formula = t1, data = tip_data)
weights: tip.nn2w

RLMerr = 0.6127, df = 1, p-value = 0.4338


        Lagrange multiplier diagnostics for spatial dependence

data:
model: lm(formula = t1, data = tip_data)
weights: tip.nn2w

RLMlag = 4.964, df = 1, p-value = 0.02588


        Lagrange multiplier diagnostics for spatial dependence

data:
model: lm(formula = t1, data = tip_data)
weights: tip.nn2w

SARMA = 5.333, df = 2, p-value = 0.0695
```

The p-values of the tests indicate that a spatial lag model (LMlag) or a spatial ARAR/ARMA model (SARMA) may be appropriate The robust spatial error and spatial lag tests (RLMerr and RLMlag, respectively) test for spatial error in the possible presence of spatial lag and vice versa. The spatial lag model can be estimated using the LAGSARLM command. A variant of this model, the spatial Durbin model, spatially lags all independent variables can be specified by TYPE="MIXED." In addition, if heteroskedasticity is a concern, the spatial lag model can be estimated using robust spatial two-stage least squares with the STSLS command. The example below will estimate a spatial lag model, a spatial Durbin model, and a 2SLS spatial lag model without and with the robust estimator.

Syntax

```
ttt1 <- lagsarlm(t1, data=tip_data, tip.nn2w, type="lag")
summary(ttt1)

ttt2 <- lagsarlm(t1, data=tip_data, tip.nn2w, type="mixed")
summary(ttt2)

ttt3 <- stsls(t1, data=tip_data, tip.nn2w)
summary(ttt3)

ttt4 <- stsls(t1, data=tip_data, tip.nn2w, robust=TRUE)
summary(ttt4)
```

Output

```
> ttt1 <- lagsarlm(t1, data=tip_data, tip.nn2w, type="lag")
> summary(ttt1)

Call:lagsarlm(formula = t1, data = tip_data, listw = tip.nn2w, type = "lag")

Residuals:
      Min        1Q    Median        3Q       Max
-5.793641 -1.424237 -0.031659  1.918979  5.404923
```

```
Type: lag
Coefficients: (asymptotic standard errors)
               Estimate  Std. Error z value  Pr(>|z|)
(Intercept) 23.57229130  3.40946154  6.9138 4.719e-12
marhh.no.c   0.01168842  0.00261014  4.4781 7.532e-06
hh.vacant   -0.02230890  0.00834930 -2.6719  0.007541
age.18.21   -0.00181405  0.00043304 -4.1891 2.800e-05

Rho: 0.21633, LR test value: 4.4825, p-value: 0.034244
Asymptotic standard error: 0.10358
    z-value: 2.0885, p-value: 0.036756
Wald statistic: 4.3617, p-value: 0.036756

Log likelihood: -90.557 for lag model
ML residual variance (sigma squared): 7.7394, (sigma: 2.782)
Number of observations: 37
Number of parameters estimated: 6
AIC: 193.11, (AIC for lm: 195.6)
LM test for residual autocorrelation
test value: 0.28192, p-value: 0.59545

>
> ttt2 <- lagsarlm(t1, data=tip_data, tip.nn2w, type="mixed")
> summary(ttt2)

Call:lagsarlm(formula = t1, data = tip_data, listw = tip.nn2w, type = "mixed")

Residuals:
     Min       1Q   Median       3Q      Max
-5.62446 -1.13030  0.17484  1.38252  5.60934

Type: mixed
Coefficients: (asymptotic standard errors)
                  Estimate  Std. Error z value  Pr(>|z|)
(Intercept)    29.17142603  6.28470222  4.6417 3.456e-06
marhh.no.c      0.01367933  0.00300702  4.5491 5.387e-06
hh.vacant      -0.02318035  0.00871656 -2.6593 0.0078292
age.18.21      -0.00163576  0.00045141 -3.6236 0.0002905
lag.marhh.no.c -0.00270135  0.00477833 -0.5653 0.5718469
lag.hh.vacant   0.01055218  0.01587041  0.6649 0.5061166
lag.age.18.21  -0.00108535  0.00067811 -1.6005 0.1094787

Rho: 0.040898, LR test value: 0.061938, p-value: 0.80346
Asymptotic standard error: 0.18387
    z-value: 0.22243, p-value: 0.82398
Wald statistic: 0.049473, p-value: 0.82398

Log likelihood: -88.59219 for mixed model
ML residual variance (sigma squared): 7.0324, (sigma: 2.6519)
Number of observations: 37
Number of parameters estimated: 9
AIC: 195.18, (AIC for lm: 193.25)
LM test for residual autocorrelation
test value: 0.18403, p-value: 0.66793

>
> ttt3 <- stsls(t1, data=tip_data, tip.nn2w)
> summary(ttt3)

Call:stsls(formula = t1, data = tip_data, listw = tip.nn2w)

Residuals:
```

```
      Min        1Q    Median        3Q       Max
-5.888618 -1.294960 -0.069785  1.971233  5.543087

Coefficients:
               Estimate  Std. Error t value  Pr(>|t|)
Rho          0.27735684  0.12185732  2.2761 0.0228413
(Intercept) 21.66674002  4.02844332  5.3784 7.513e-08
marhh.no.c   0.01140261  0.00280304  4.0679 4.743e-05
hh.vacant   -0.02157604  0.00898253 -2.4020 0.0163057
age.18.21   -0.00170807  0.00049367 -3.4599 0.0005403

Residual variance (sigma squared): 8.977, (sigma: 2.9962)

>
> ttt4 <- stsls(t1, data=tip_data, tip.nn2w, robust=TRUE)
> summary(ttt4)

Call:stsls(formula = t1, data = tip_data, listw = tip.nn2w, robust = TRUE)

Residuals:
      Min        1Q    Median        3Q       Max
-5.888618 -1.294960 -0.069785  1.971233  5.543087

Coefficients:
               Estimate Robust std. Error z value  Pr(>|z|)
Rho          0.27735684        0.13701485  2.0243   0.04294
(Intercept) 21.66674002        4.59981491  4.7104 2.473e-06
marhh.no.c   0.01140261        0.00263770  4.3229 1.540e-05
hh.vacant   -0.02157604        0.00537992 -4.0105 6.060e-05
age.18.21   -0.00170807        0.00042314 -4.0367 5.422e-05

Residual variance (sigma squared): 8.977, (sigma: 2.9962)
```

The spatial lag parameter is positive and statistically significant in all models except the spatial Durbin model. None of the spatially lagged independent variables in the spatial Durbin model are significant. Note that the p-values of the variables significantly changed with the robust spatial two-stage least squares model. Analysis of the marginal effects can show how neighboring Census tracts are impacted by changes in independent variables using the IMPACTS command. Standard errors can be calculated using Monte Carlo methods if the R=(number of iterations) and ZSTATS=TRUE options are specified. The example below will calculate marginal effects for the first spatial lag model.

Syntax

```
ttt1.i<- impacts(ttt1, listw=tip.nn2w, R=1000, zstats=TRUE, short=TRUE)
summary(ttt1.i)
```

Output

```
> ttt1.i<- impacts(ttt1, listw=tip.nn2w, R=1000, zstats=TRUE, short=TRUE)
> summary(ttt1.i)
Impact measures (lag, exact):
                Direct     Indirect        Total
marhh.no.c  0.011817716  0.003097291  0.014915007
hh.vacant  -0.022555682 -0.005911592 -0.028467273
age.18.21  -0.001834119 -0.000480702 -0.002314821
========================================================
Simulation results (asymptotic variance matrix):
Direct:
```

```
Iterations = 1:1000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 1000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                Mean        SD  Naive SE Time-series SE
marhh.no.c  0.011861 0.0027084 8.565e-05      7.858e-05
hh.vacant  -0.022851 0.0082045 2.594e-04      2.424e-04
age.18.21  -0.001839 0.0004122 1.303e-05      1.408e-05

2. Quantiles for each variable:

                2.5%       25%       50%       75%     97.5%
marhh.no.c  0.006770  0.010022  0.011786  0.013750  0.017072
hh.vacant  -0.039178 -0.028256 -0.022808 -0.017174 -0.007361
age.18.21  -0.002608 -0.002123 -0.001839 -0.001552 -0.001026

========================================================
Indirect:

Iterations = 1:1000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 1000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                 Mean        SD  Naive SE Time-series SE
marhh.no.c  0.0031086 0.0007098 2.245e-05      2.060e-05
hh.vacant  -0.0059890 0.0021503 6.800e-05      6.353e-05
age.18.21  -0.0004821 0.0001080 3.416e-06      3.689e-06

2. Quantiles for each variable:

                 2.5%        25%        50%        75%      97.5%
marhh.no.c  0.0017744  0.0026268  0.0030891  0.0036038  0.0044743
hh.vacant  -0.0102681 -0.0074056 -0.0059777 -0.0045010 -0.0019292
age.18.21  -0.0006834 -0.0005564 -0.0004821 -0.0004066 -0.0002688

========================================================
Total:

Iterations = 1:1000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 1000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                Mean        SD  Naive SE Time-series SE
marhh.no.c  0.014970 0.0034183 1.081e-04      9.918e-05
hh.vacant  -0.028840 0.0103548 3.274e-04      3.059e-04
age.18.21  -0.002321 0.0005202 1.645e-05      1.777e-05

2. Quantiles for each variable:

                2.5%       25%       50%       75%     97.5%
marhh.no.c  0.008545  0.012649  0.014875  0.017354  0.021546
```

```
hh.vacant  -0.049446 -0.035662 -0.028785 -0.021675 -0.009290
age.18.21  -0.003291 -0.002679 -0.002322 -0.001958 -0.001295
```

The most important output is displayed first (the remaining output consists of the standard errors for each type of impact). The "Direct" column represents the local impacts on a Census tract and the "Indirect" column represents the impacts on neighboring Census tracts. In this example, a one-unit change in the number of vacant household decreases the median age by -0.023 in the Census tract of interest and by a total of -0.0059 in the two nearest neighbors of that Census tract (or an average of -0.00295 per neighbor).

Spatial error models can be estimated using the ERRORSARLM and the SPAUTOLM commands. The FAMILY option can be changed to estimate spatial moving average models (SMA) to capture localized impacts or conditionally autoregressive models (CAR).

Syntax

```
ttt5 <- errorsarlm(t1, data=tip_data, tip.nn2w)
ttt6 <- spautolm(t1, data=tip_data, tip.nn2w, family="SAR")
ttt7 <- spautolm(t1, data=tip_data, tip.nn2w, family="SMA")
ttt8 <- spautolm(t1, data=tip_data, nb2listw(make.sym.nb(tip.nn2), style="B"),
family="CAR")
```

Spatial error models can also be estimated by the Generalized Method of Moments (GMM) using the GMERRORSAR command. The GMARGIMAGE command can be used to visualize the argmin process used to find the spatial error parameter (lambda), and the HAUSMAN.TEST can be used to conduct a spatial Hausman test to test whether the coefficients estimated by OLS are significantly different from the coefficients estimated from the spatial error model.

Syntax

```
ttt100 <- GMerrorsar(t1, data=tip_data, listw=tip.nn2w,returnHcov=TRUE)
summary(ttt100)

# Map the argmin function used to find spatial error parameter lambda
ttt100z<-GMargminImage(ttt100)
levs <- quantile(ttt100z$z, seq(0, 1, 1/12))
image(ttt100z, breaks=levs, xlab="lambda", ylab="s2")
points(ttt100$lambda, ttt100$s2, pch=3, lwd=2)
contour(ttt100z, levels=signif(levs, 4), add=TRUE)

# Test for significant differences between OLS, SEM

Hausman.test(ttt100)
```

Output

```
> summary(ttt100)

Call:
GMerrorsar(formula = t1, data = tip_data, listw = tip.nn2w, returnHcov = TRUE)

Residuals:
     Min       1Q   Median       3Q      Max
-6.42972 -2.95045  0.52438  2.14809  5.93364
```

```
Type: GM SAR estimator
Coefficients: (GM standard errors)
              Estimate  Std. Error z value  Pr(>|z|)
(Intercept) 30.06981098 1.36436829 22.0394 < 2.2e-16
marhh.no.c   0.01302796 0.00286872  4.5414 5.588e-06
hh.vacant   -0.02559460 0.00861433 -2.9712  0.002967
age.18.21   -0.00190071 0.00045651 -4.1635 3.133e-05

Lambda: 0.15942
Number of observations: 37
Number of parameters estimated: 6

> Hausman.test(ttt100)

        Spatial Hausman test (approximate)

data:  med.age ~ 1 + marhh.no.c + hh.vacant + age.18.21
Hausman test = 7.4996, df = 4, p-value = 0.1117
```

The p-value indicates that the estimates are not significantly different, which implies that the spatial error model is unbiased (the null hypothesis assumes both the OLS and spatial error estimators are unbiased). Shown below is the argmin image.

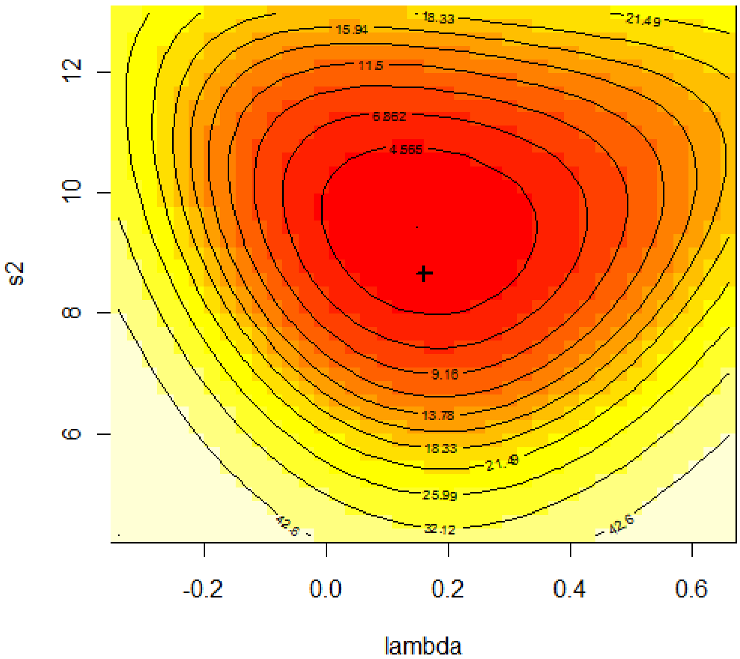


Figure 7: Argmin image from spatial error model estimated using generalized moments.

Spatial ARAR models can be estimated using the SACSARLM command. This command can use two different weights matrices, one for the spatial lag term, the other for the spatial error term. In this example, a spatial ARAR model will be estimated using two nearest-neighbors for the spatial lag term and the distance threshold matrix for the spatial error term.

Syntax

```
ttt9 <- sacsarlm(t1, data=tip_data, listw=tip.nn2w,listw2=tip.dw)
summary(ttt9)
```

Output

```
> ttt9 <- sacsarlm(t1, data=tip_data, listw=tip.nn2w,listw2=tip.dw)
> summary(ttt9)

Call:sacsarlm(formula = t1, data = tip_data, listw = tip.nn2w, listw2 = tip.dw)

Residuals:
      Min        1Q    Median        3Q       Max
-5.294877 -1.471924  0.051992  1.694360  4.872865

Type: sac
Coefficients: (asymptotic standard errors)
               Estimate  Std. Error z value  Pr(>|z|)
(Intercept) 23.99263479  3.17055278  7.5673 3.819e-14
marhh.no.c   0.01235672  0.00246689  5.0090 5.470e-07
hh.vacant   -0.02572765  0.00793758 -3.2412   0.00119
age.18.21   -0.00180864  0.00041904 -4.3161 1.588e-05

Rho: 0.20227
Asymptotic standard error: 0.095572
    z-value: 2.1164, p-value: 0.034308
Lambda: -0.68138
Asymptotic standard error: 0.42412
    z-value: -1.6066, p-value: 0.10815

LR test value: 7.7691, p-value: 0.020558

Log likelihood: -88.91372 for sac model
ML residual variance (sigma squared): 6.9276, (sigma: 2.632)
Number of observations: 37
Number of parameters estimated: 7
AIC: 191.83, (AIC for lm: 195.6)
```

Interestingly, the spatial error term is now (marginally) significant using a different weights matrix. Note that the magnitude of the spatial lag parameter did not significantly change when compared to the spatial lag model. A robust variant that corrects for heteroskedasticity can be estimated using the GSTSLS command from the SPHET package.

Syntax

```
# SAC with robust correction, sphet
# Note how the p-value of the lag parameter drops to 0.06 vs. 0.03
# If you get a negative lag or rho parameter, sample size
# may be too small

ttt90 <- gstsls(t1,data=tip_data,listw=tip.nn2w,listw2=tip.dw,robust=TRUE)
summary(ttt90)
```

Output

```
> summary(ttt90)

Call:gstsls(formula = t1, data = tip_data, listw = tip.nn2w, listw2 = tip.dw,
    robust = TRUE)

Residuals:
       Min         1Q     Median         3Q        Max
-5.3332468 -1.2382329 -0.0045799  1.6154866  4.9044945

Type: GM SARAR estimator
Coefficients: (GM standard errors)
              Estimate Std. Error z value  Pr(>|z|)
Wyt          0.2309161  0.1254193  1.8412    0.0656
(Intercept) 23.1157659  4.0966625  5.6426 1.675e-08
marhh.no.c   0.0121537  0.0022757  5.3407 9.259e-08
hh.vacant   -0.0252029  0.0051741 -4.8710 1.111e-06
age.18.21   -0.0017689  0.0004084 -4.3312 1.483e-05

Lambda: -0.67857
Number of observations: 37
Number of parameters estimated: 7
```

The spatial lag models estimated previously assumed that the distance decay effect of spatial correlation (i.e. effects of neighbors increase with distance - one would not expect changes in North Vernon, in southeastern Indiana, to have an appreciable effect on West Lafayette) has a geometric functional form. The matrix exponential spatial lag model (MESS), on the other hand, assumes this decay function is exponential, as opposed to geometric. The spatial lag MESS model can be estimated In SPDEP using the LAGMESS command, which has a syntax similar to LAGSARLM and STSLS.

Syntax

```
ttt91 <- lagmess(t1, data=tip_data, tip.nn2w)
summary(ttt91)
```

Output

```
Matrix exponential spatial lag model:

Call:
lagmess(formula = t1, data = tip_data, listw = tip.nn2w)

Coefficients:
               Estimate  Std. Error t value  Pr(>|t|)
(Intercept) 23.32748316  1.30002938 17.9438 < 2.2e-16
marhh.no.c   0.01152244  0.00269711  4.2721 0.0001544
hh.vacant   -0.02203113  0.00871004 -2.5294 0.0163848
age.18.21   -0.00181454  0.00043834 -4.1396 0.0002260

Residual standard error: 2.9446 on 33 degrees of freedom
Multiple R-squared: 0.66387,    Adjusted R-squared: 0.63331
F-statistic: 21.725 on 3 and 33 DF,  p-value: 5.9628e-08

Alpha: -0.25515, standard error: 0.11982
    z-value: -2.1294, p-value: 0.033219
LR test value: 4.91, p-value: 0.026702
Implied rho: 0.225199
```

Note that the spatial lag MESS parameter, which is implicit with the matrix exponential spatial lag model, is not significantly different when compared to the spatial lag parameter in the other models.

References

Almquist, Z.W. (2010). UScensus2000: US Census 2000 Suite of R Packages. R package version 0.07. http://CRAN.R-project.org/package=UScensus2000

Anselin, L. (1988). Lagrange Multiplier Test Diagnostics for Spatial Dependence and Spatial Heterogeneity, *Geographical Analysis*, 20, 1-17.

Anselin, L. (2005). Exploring Spatial Data with GeoDa™: A Workbook. http://geodacenter.asu.edu/system/files/geodaworkbook.pdf

Anselin, L. (2006). Spatial Econometrics. In Mills, T.C., and K. Patterson (eds.), *Palgrave Handbook of Econometrics, Volume 1: Econometric Theory*. Palgrave/Macmillan.

Anselin, L. (2010). GeoDa Center for Geospatial Analysis and Computation. http://geodacenter.asu.edu/

Baltagi, B.H., Song, S.H., Jung, B.C., and W. Koh (2007).Testing for Serial Correlation, Spatial Autocorrelation and Random Effects Using Panel Data, *Journal of Econometrics*, 140 (2007), 5-51.

Bivand, R., et al. (2010). spdep: Spatial dependence: weighting schemes, statistics and models. R package version 0.5-4. http://CRAN.R-project.org/package=spdep

Bivand, R.S. Pebesma, E.J., and V. Gómez-Rubio (2008). *Applied Spatial Data Analysis with R*. Springer Science + Business Media, LLC, New York, NY.

Elhorst, J.P. (2003). 'Specification and Estimation of Spatial Panel Data Models,' International *Regional Science Review*, (2003), 26, 244.

Elhorst, J.P. (2010). Spatial Panel Data Models. In Fischer MM, Getis A (Eds.) *Handbook of Applied Spatial Analysis*, pp. 377-407. Springer: Berlin Heidelberg New York. http://www.regroningen.nl/elhorst/software.shtml

Fingleton, B. (2008). A Generalized Method of Moments Estimator for a Spatial Model with Moving Average Errors, with Application to Real Estate Prices. In Arbia. G., and B.H. Baltagi (eds.), *Spatial Econometrics: Methods and Applications*. Physica-Verlag/Springer.

Fotheringham, A.S., Brunsdon, C., and M. Charlton (2002). Geographically Weighted Regression: The Analysis of Spatially Varying Relationships. John Wiley & Sons, Hoboken NJ.

Kapoor, M., Kelejian, H.H. and Prucha, I.R. (2007) Panel Data Model with Spatially Correlated Error Components, *Journal of Econometrics*,140, 97–130.

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.

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

Lewin-Koh, N.J., and R. Bivand (2010). maptools: Tools for reading and handling spatial objects. R package version 0.7-34. http://CRAN.R-project.org/package=maptools

LeSage, J.P. (1998). Spatial Econometrics. http://www.spatial-econometrics.com/html/wbook.pdf

LeSage, J.P. (1999). The Theory and Practice of Spatial Econometrics. http://www.spatial-econometrics.com/html/sbook.pdf

LeSage, J.P. (2010). Econometrics Toolbox: by James P. LeSage. http://www.spatial-econometrics.com/

LeSage, J.P., and R.K. Pace (2007). A Matrix Exponential Spatial Specification, *Journal of Econometrics*, 140, 190-214.

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

Millo, G. and G. Piras (2010). splm: Econometric Models for Spatial Panel Data. R package version 0.1-21/r82. http://R-Forge.R-project.org/projects/splm/

Mills, J. (2010). *Spatial Panel Econometric Analysis of the Economic Impacts of Bypasses: A Regional Approach*. IVth World Conference of the Spatial Econometrics Association, June 2010, Chicago IL. http://www.agecon.purdue.edu/sea_2010/Sessions/SessionIndex.htm

Pebesma, E.J., and R.S. Bivand (2005). Classes and methods for spatial data in R. R News 5 (2), http://cran.r-project.org/doc/Rnews/

Piras, G. (2010). sphet: Spatial Models with heteroskedastic innovations. R package version 0.1-22. http://CRAN.R-project.org/package=sphet

Purdue Institute of Transportation Engineers (2010). An Introduction to R. http://0pbg2.tk

Additional References: Applications

Atasoy, M., Palmquist, R.B., and D.J. Phaneuf (2006). Estimating the Effects of Urban Residential Development on Water Quality Using Microdata, *Journal of Environmental Management*, Vol. 79, Issue 4, 399.

Baumont, C. (2009). Spatial Effects of Urban Public Policies on Housing Values, *Papers in Regional Science*, Vol. 88, Issue 2, 301-326.

Billion, M., Ezcurra, R., and F. Lera-López (2009). Spatial Effects in Website Adoption by Firms in European Regions, *Growth and Change*, Vol. 40, Issue 1, 54.

Caleiro, A., and G. Guerreiro (2005). Understanding the Election Results in Portugal, *Portuguese Economic Journal*, Vol. 4, Issue 3, 207-228.

Cohen, J.P., and C.M. Paul (2007). The Impacts of Transportation Infrastructure on Property Values: A Higher-Order Spatial Econometrics Approach, *Journal of Regional Science*, Vol. 47, Issue 3, 457-478.

Frank, B., and P.B. Maurseth (2006). The Spatial Econometrics of Elephant Population Change: A Note, *Ecological Economics*, Vol. 60, Issue 1, 320-323.

Huang, B., Zhang, L., and B. Wu (2009). Spatiotemporal Analysis of Rural-Urban Land Conversion, *International Journal of Geographical Information Science*, 23, 3 (2009), 379-398.

Jensen, P.S., and K.S. Gleditsch (2009). Rain, Growth, and Civil War: The Importance of Location, *Defense and Peace Economics*, Vol. 20, Issue 5, 359-372.

Kim, J. (2007). Discriminant Impact of Transit Station Location on Office Rent and Land Value in Seoul: An Application of Spatial Econometrics, *Journal of Transport Economics & Policy*, Vol. 41, Issue 2, 219-245.

Richards, T.J. Eaves, J., Manfredo, M., Naranjo, S.E., Chu, C.-C., and T.J. Henneberry (2008). Spatial-Temporal Model of Insect Growth, Diffusion, and Derivative Pricing, *American Journal of Agricultural Economics,* Vol. 90, Issue 4, 962-978.

Shanzi, K. (2010). Determinants of Economic Growth and Spread-backwash Effects in Western and Eastern China, *Asian Economic Journal*, Vol. 24, Issue 2, 179-202.

Wojan, T.R., Lambert, D.M., and D.A. Mcgranahan (2007). Emoting with Their Feet: Bohemian Attraction to Creative Milieu, *Journal of Economic Geography*, Vol. 7, Issue 6, 711-736.

Zhou, B., and K.M. Kockelman (2008). Neighborhood Impacts on Land Use Change: a Multinomial Logit Model of Spatial Relationships, *Annals of Regional Science* (2008), 42, 321-340.

Additional References: Spatial Count Data Modeling

Agarwal, D.K., Gelfand, A.E., and S. Citron-Pousty (2002). Zero-Inflated Models with Application to Spatial Count Data, *Environmental and Ecological Statistics*, 9 (2002), 341-355.

Griffith, D.A. (2006). Assessing Spatial Dependence in Count Data: Winsorized and Spatial Filter Specification Alternative to the Auto-Poisson Model, *Geographical Analysis*, 38 (2006), 160-179.

Gschlößl, S., and C. Czado (2008). Modelling Count Data with Overdispersion and Spatial Effects, *Statistical Papers*, 49 (2008), 531-552.

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

Lawson, A.B. (2009). Bayesian Disease Mapping: Hierarchical Modeling in Spatial Epidemiology, CRC Press, Boca Raton FL.

Lin, P. (2010). A Working Estimating Equation for Spatial Count Data, *Journal of Statistical Planning and Inference*, 140 (2010), 2470-2477.

Quddus, M.A. (2008). Modelling Area-Wide Count Outcomes with Spatial Correlation and Heterogeneity: An Analysis of London Crash Data, *Accident Analysis and Prevention*, 40 (2008), 1486-1497.

Wakefield, J. (2007). Disease Mapping and Spatial Regression with Count Data, *Biostatistics*, 8 (2), 158-183.

Wang, C., Quddus, M.A., and S.G. Ison (2009). Impact of Traffic Congestion on Road Accidents: A Spatial Analysis of the M25 Motorway in England, *Accident Analysis and Prevention*, 41, 4, 798-808.

Webster, R.A., Pollock, K.H., and T.R. Simons (2008). Bayesian Spatial Modeling of Data from Avian Point Count Surveys, *Journal of Agricultural, Biological, and Environmental Statistics*, 13, 2, 121-139.