sphet: Spatial Models with Heteroskedastic Innovations in R

Gianfranco Piras
Cornell University

Abstract

This introduction to the R package **sphet** is a (slightly) modified version of **?**, published in the *Journal of Statistical Software*.

sphet is a package for estimating and testing spatial models with heteroskedastic innovations. We implement recent generalized moments estimators and semiparametric methods for the estimation of the coefficients variance-covariance matrix. This paper is a general description of **sphet** and all functionalities are illustrated by application to the popular Boston housing dataset. The package in its current version is limited to the estimators based on **???**. The estimation functions implemented in **sphet** are able to deal with virtually any sample size.

Keywords: spatial models, R, computational methods, semiparametric methods, kernel functions, heteroskedasticity.

1. Introduction

sphet is a package for estimating and testing a variety of spatial models with heteroskedastic innovations. The estimation procedures are based on generalized moments (GM).

An increasing number of datasets contain information on the location of the observations along with the values of the variables of interest. Taking into account the spatial nature of the data can improve efficiency or, in some cases, even be essential for consistency. This increasing interest in spatial data is corroborated by the large number of packages in the R language (**?**) for analyzing multiple typologies of spatial data under different methodological perspectives. Among these alternatives, **spdep** is one of several packages that deals with spatial dependence (**?????**). **spdep** includes functions to create and manipulate spatial objects (i.e., creating spatial weights matrices). It also contains a collection of tests for spatial autocorrelation and functions for estimating spatial models. For what concerns estimation features, general method of moments (GMM), instrumental variables (IV) and maximum likelihood (ML) are supported.

sphet complements but does not overlap with the econometric features already available in **spdep**. Specifically, **spdep** focuses on spatial lag and error models, whereas our departure point is the following model:

$$\begin{aligned} y &= X\beta + \lambda W y + u \\ u &= \rho W u + \varepsilon \end{aligned} \tag{1}$$

? do not assume any specific structure for the disturbance process. Their focus is to develop an estimation theory (for the regression parameters) that is robust against possible misspecification of the disturbances. Nonetheless, the general assumptions made on the disturbance process cover the spatial autoregressive process as a special case.

Although Model (1) is well established in the econometrics literature, regional scientists and geographers generally follow a different approach. They start from an OLS regression and try to determine (via Lagrange multipliers (LM) tests on estimated residuals) whether the true data generating process is a spatial error, or a spatial lag model. This is unfortunate because the spatial patterns implied by Model (1) are richer than those implied by either the spatial error or the spatial lag model (**?**). We believe that providing this alternative implementation is a useful contribution to both scientific communities. Related to this, **?** give results concerning the joint asymptotic distribution of IV and GMM estimators for Model (1). Their results permit testing the (joint) hypothesis of no spatial spillovers originated from the endogenous variables or from the disturbances. As a consequence, if one of the corresponding coefficients turns out not to be statistically different from zero, one could still go back to the estimation of a reduced model.

In **sphet**, we only concentrate on GM and IV methods, leaving aside the ML approach. Reasons for this will be discussed throughout the paper. In general terms, GM requires weaker assumptions than ML. Additionally, there are still various unsolved problems related to the ML approach. Numerical difficulties related to the computation of the Jacobian term (**??**) may potentially limit the application of ML to large datasets. However, various solutions have been proposed in the literature that have attenuated the problem (see e.g., **??????**, among others). **?** derives the conditions that ensure consistency and asymptotic normality of ML estimators for the general spatial model considered but some of the assumptions are stronger than those required by GM. On the other hand, there is reasonably general theory for the GM approach in the cross sectional case (**???**).

One final point relates to the possible presence of heteroskedasticity in the innovations of the model. Since spatial units may differ in important characteristics (e.g., size) homoskedasticity is a strong assumption that may not hold in many applied spatial problems. **?** and **?** are two typical examples of empirical applications that require the use of spatial heteroskedasticity and autocorrelation consistent (HAC) estimators. The estimation theory developed by **?** for the quasi-maximum likelihood estimator under the assumption of homoskedastic innovations does not carry over to the case of heteroskedastic innovations. To support this, **?** provide simulation evidence that when the innovations are heteroskedastic, ML produces inconsistent estimates. We implement various GM and IV procedures to obtain spatial HAC estimators of the variance-covariance matrix of the model coefficients. In its current version, the package is limited to methodologies implemented in **??** and **?**. The `gstslshet` code was tested against the original Stata (**?**) code used to produce the simulation results in **?**. The function `stslshac` give results like those implemented in Python (**?**) and used in **?**. Unfortunately, none of the cited implementations are available to the general public. The estimation functions implemented in **sphet** are able to handle virtually any sample size.

To overcome part of the technical difficulties that arise in the implementation, we make extensive use of classes of objects defined in **spdep** as for example spatial weights matrix objects (of class `listw`). Furthermore, we also make substantial use of code from the **Matrix** package (**?**).

The remainder of the present paper is a general description of **sphet** and all functionalities are illustrated by application to the popular Boston housing dataset (**?**).

The Boston data contain information on property values and characteristics in the area of Boston, Massachusetts and has been widely used for illustrating spatial models. Specifically, there is a total of 506 units of observation for each of which a variety of attributes are available, such as: (corrected) median values of owner-occupied homes (`CMEDV`); per-capita crime rate (`CRIM`); nitric oxides concentration (`NOX`); average number of rooms (`RM`); proportions of residential land zoned for lots over 25,000 sq. ft (`ZN`); proportions of non-retail business acres per town (`INDUS`); Charles River dummy variable (`CHAS`); proportions of units built prior to 1940 (`AGE`); weighted distances to five Boston employment centers (`DIS`); index of accessibility to highways (`RAD`); property-tax rate (`TAX`); pupil-teacher ratios (`PTRATIO`); proportion of blacks (`B`); and % of the lower status of the population (`LSTAT`).

The dataset with Pace's tract coordinates is available to the R community as part of **spdep**.

The `library("sphet")` command loads **sphet**; the `data("boston", package = "spdep")` command loads the Boston data from **spdep**.

```
R> library(sphet)
R> library(spdep)
R> data("boston", package = "spData")
```

The spatial weights matrix is a sphere of influence neighbors list also available from **spdep** once the Boston data are loaded:

```
R> listw <- nb2listw(boston.soi)
```

2. Tools

sphet supports a series of capabilities to generate `distance` objects that are required by the semiparametric estimation of the variance-covariance matrix discussed in Section 3. These functionalities can be accessed through the functions `distance` and `read.gwt2dist`.

The function `distance` reads points coordinates and generates a `matrix`. The object created is similar to the content of a '`.GWT`' file. A '`.GWT`' file format is a well known structure among spatial statisticians and econometricians. These (memory efficient) types of weights matrices can be calculated using **GeoDa** (**?**) and other softwares. In fact, one advantage of R over other packages is the availability of these models that are well established among users. The generated object is made up of three columns. The third column lists the distances, while the first two columns contain the id of the two points for which the distance is calculated.

Currently, five distance measures are implemented, namely: Euclidean ($d_{ij} = \sqrt{\sum (x_i - y_i)^2}$), Chebyshev ($d_{ij} = \max(|x_i - y_i|)$), Bray-Curtis ($d_{ij} = \sum |x_i - y_i| / \sum |x_i + y_i|$), Canberra ($d_{ij} = \sum |x_i - y_i| / \sum |x_i| + |y_i|$) and the Great Circle distance. For details on how to calculate this last distance measure one should see the function `rdist.earth` in the package **fields** (**?**).

The following instructions demonstrate the usage of the function `distance`. We first generate a set of XY- coordinates corresponding to one hundred points. The first coordinate is a random sample from the uniform distribution on the interval $(0, 70)$, whereas the second

coordinate is generated on the interval $(-30, 20)$. The object `coord1` is a matrix whose first column is intended to contain the identification for the points.

```
R> set.seed(1234)
R> X <- runif(100, 0, 70)
R> Y <- runif(100, -30, 20)
R> coord1 <- cbind(seq(1, 100), X, Y)
```

The (optional) id variable has the principal scope of providing the ordering of the observations. When specified, it could be the first column of the argument `coord`. Alternatively, it could be specified separately as `region.id`. When `region.id` is not `NULL` and `coord` has three columns (i.e., an id variable has been specified twice) the function performs some checks to make sure that the two variables point to the same ordering. On the other hand, if an id variable is not specified at all, it is assumed to be a sequence from one to the number of observations (i.e., the number of coordinates).

```
R> thm1 <- distance(coord = coord1, region.id = NULL,
+                   output = FALSE, type = "inverse", measure = "euclidean")
R> print(head(thm1))
```

```
     from to   distance
[1,]    1  2 0.02253759
[2,]    1  3 0.02718421
[3,]    1  4 0.02727669
[4,]    1  5 0.01903487
[5,]    1  6 0.02524238
[6,]    1  7 0.10601540
```

The `measure` argument specifies the distance measure and should be one of `"euclidean"`, `"gcircle"`, `"chebyshev"`, `"braycur"`, and `"canberra"`. The `type` argument is used to define the distance criteria and should be one of `"inverse"`, `"NN"` or `"distance"`. Both `"inverse"` and `"distance"` can be specified along with a `cutoff`. The `cutoff` takes up three values: `1`, `2`, and `3` indicating the lower, median and upper quantile of the distance distribution. Specifically, when `cutoff` is set to `1`, only observations within a distance less than the first quantile are neighbors to each other. All other interactions are considered negligible. `"NN"` (nearest neighbors) should be specified along with `nn`, the argument to define the number of nearest neighbors, as it is illustrated in the following example.

```
R> thm2 <- distance(coord1, region.id = NULL,
+                   output = FALSE, type = "NN", nn = 6)
R> print(head(thm2))
```

```
     from to  distance
[1,]    1  7  9.432592
[2,]    1  8  9.567595
[3,]    1 19  6.744797
```

```
[4,]    1 55 10.333073
[5,]    1 65  9.115394
[6,]    1 93  4.854875
```

When `output` is TRUE, the function writes the data to a file. The output file can have any format. In particular, it could be a '`.GWT`' file. When `firstline` is `TRUE`, an header line is added to the '`.GWT`' file. The first element is simply a place holder, the second is the number of observations. The name of the shape file and of the id variable can be specified by the options `shape.name` and `region.id.name` respectively. If an output file is produced, the name of the file can be set by `file.name`.

```
R> thm3 <- distance(coord1, region.id = NULL, output = TRUE,
+                   type = "distance", cutoff = 1, measure = "gcircle",
+                   shape.name = "shapefile", region.id.name = "id1",
+                   firstline = TRUE, file.name = "dist_100.GWT")
R> class(thm3)

[1] "matrix"          "distance.matrix"
```

The value is a `matrix` of three columns. The third column lists the distances, while the first two columns contain the id of the two points for which the distance is calculated.

To create an object of class `distance`, one should use the function `read.gwt2dist`, as in the following example:

```
R> id1 <- seq(1, nrow(boston.utm))
R> tmp <- distance(boston.utm, region.id = id1, output = TRUE,
+                  type = "NN", nn = 10, shape.name = "shapefile",
+                  region.id.name = "id1", firstline = TRUE,
+                  file.name = "boston_nn_10.GWT")
R> coldist <- read.gwt2dist(file = "boston_nn_10.GWT", region.id = id1, skip = 1)
```

The function `read.gwt2dist` reads a '`.GWT`' file (e.g., generated using the function `distance`). In this example we are using the matrix of tract point coordinates projected to UTM zone 19 `boston.utm` available from **spdep** to generate a '`.GWT`' file of the 10 nearest neighbors. The file '`boston_nn_10.GWT`" is then inputted to the function `read.gwt2dist` along with the `region.id`.

It is worth noticing that the function `read.gwt2dist` could also read other extensions (such as '`.txt`'). It is important, however, that the input file exhibits the general format described above. When the file has a '`.GWT`' extension, the number of observations is generally retrieved from the first line. Alternatively, it is fixed to the length of the (`unique`) `region.id` variable. The argument `skip` determines the number of lines to disregard before reading the data. The value is an object of class `distance`. We generate a new class of objects to be able to perform some of the checks necessary to make sure that the distance measure specified in the function `stslshac` is appropriate.

```
R> class(coldist)
```

```
[1] "sphet"    "distance" "nb"       "GWT"
```

A `summary` method is available to print some of the basic features of the `distance` object. In particular, the total number of observations and some general descriptive statistics for both distances and neighbors are displayed. We believe that this information is of guidance while choosing the type of bandwidth to employ in the spatial HAC estimation discussed in Section 3.

```
R> summary(coldist)

 Number of observations:
 n: 506

 Distance summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5441  0.9588  1.5843  2.0848  2.6389 11.6388

 Neighbors summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     10      10      10      10      10      10
```

3. Estimation functions

Spatial models in **sphet** are fitted by the functions `stslshac` and `gstslshet`. Below, we first review some of the theory and then demonstrate the use of these functions.

3.1. Spatial two stages least squares with HAC standard errors

Consider the following spatial model:

$$y = X\beta + \lambda Wy + \varepsilon \tag{2}$$

or also, more compactly

$$y = Z\gamma + \varepsilon \tag{3}$$

with $Z = [X, Wy]$ and $\gamma = [\beta^\top, \lambda]^\top$. The presence of the spatially lagged dependent variable Wy introduces a form of endogeneity. Under typical specifications, Wy will be correlated with the disturbances ε, which motivates an instrumental variable approach. The spatial two stage least squares (S2SLS) regression is a straightforward extension of the "classical" two stage least squares procedure. The selection of instruments as an approximation to ideal instruments has been extensively discussed in the literature (see e.g., **????**, among others) and is based on the reduced form of the model. In empirical problems, the matrix of instruments can be defined in the following way:

$$H = (X, WX, \ldots, W^q X) \tag{4}$$

where, typically, $q \leq 2$. The matrix of instruments implemented in **sphet** is $H = (X, WX, W^2X)$.

The S2SLS estimator for the parameter vector γ can be obtained as:

$$\hat{\gamma}_{S2SLS} = [\hat{Z}^\top Z]^{-1}\hat{Z}^\top y \tag{5}$$

where $\hat{Z} = PZ = [X, \widehat{Wy}]$, $\widehat{Wy} = PWy$ and $P = H(H^\top H)^{-1}H^\top$. Statistical inference is generally based on the asymptotic variance covariance matrix:

$$Var(\hat{\gamma}_{S2SLS}) = \hat{\sigma}^2(\hat{Z}^\top Z)^{-1} \tag{6}$$

with $\hat{\sigma}^2 = e^\top e/n$ and $e = y - Z\hat{\gamma}_{S2SLS}$.

? propose a HAC consistent estimation of the variance covariance matrix of Model (2). The spatial HAC estimator is robust against possible misspecification of the disturbances and allows for (unknown) forms of heteroskedasticity and correlation across spatial units. The disturbance vector is assumed to be generated by the following very general process:

$$\varepsilon = R\xi \tag{7}$$

where ξ is a vector of innovations and R is an $n \times n$ non stochastic matrix whose elements are not known. Additionally, R is non-singular and the row and column sums of R and R^{-1} are bounded uniformly in absolute value by some constant (for technical details see **??????**). This specification of the error term covers SARMA(p,q) processes as special cases. Even if we assume such a general specification for the disturbance process we still have to be concerned about possible misspecifications (e.g., due to an incorrect specification of the weights matrices). The asymptotic distribution of corresponding IV estimators involves the variance covariance matrix

$$\Psi = n^{-1}H^\top \Sigma H \tag{8}$$

where $\Sigma = RR^\top$ denotes the variance covariance matrix of ξ.
?, propose to estimate the r, s elements of Ψ by:

$$\hat{\psi}_{rs} = n^{-1}\sum_{i=1}^{n}\sum_{j=1}^{n} h_{ir}h_{js}\hat{\varepsilon}_i\hat{\varepsilon}_j K(d^*_{ij}/d) \tag{9}$$

where the subscripts refer to the elements of the matrix of instruments H and the vector of estimated residuals $\hat{\varepsilon}$. **?** also contains a generalization to several distance measures. In that case, the expression for the spatial HAC estimator of the true variance covariance matrix assumes a slightly different form. However, we only implement the estimator based on the case of a single measure. $K()$ is a Kernel function used to form the weights for the different covariance elements. The Kernel function is a real, continuous and symmetric function that determines the pairs of observations included in the cross products in (9). The Kernel function is defined in terms of a distance measure. More specifically, d^*_{ij} represent the distance between observations i and j and d is the bandwidth. Note that **?** allows for the case where the researcher measures these distances with error. More in detail, the distance measure employed by the researcher is given by

$$d^*_{ij} = d_{ij} + \upsilon_{ij}$$

where υ_{ij} denotes the measurement error. The only assumption made on the random measurement error is that it is independent on the innovations of the model ξ. The bandwidth is such that if $d^*_{ij} \geq d$, the associated Kernel is set to zero ($K(d^*_{ij}/d) = 0$). In other words,

the bandwidth plays the same role as in the time series literature; Together with the Kernel function it limits the number of sample covariances. Furthermore, the bandwidth can be assumed either fixed or variable. A fixed bandwidth corresponds to a constant distance for all spatial units. On the other hand, a variable bandwidth varies for each observation (i.e., the distance corresponding to the n-nearest neighbors).

Based on the spatial HAC estimator of Ψ given in (9), the asymptotic variance covariance matrix ($\hat{\Phi}$) of the S2SLS estimator of the parameters vector is given by:

$$\hat{\Phi} = n^2(\hat{Z}^\top \hat{Z})^{-1} Z^\top H (H^\top H)^{-1} \hat{\Psi} (H^\top H)^{-1} H^\top Z (\hat{Z}^\top \hat{Z})^{-1} \tag{10}$$

Therefore, small sample inference can be based on the approximation $\hat{\gamma} \sim N(\gamma, n^{-1}\hat{\Phi})$.

Demonstration

The function that deals with the spatial HAC estimator is `stslshac`. Crucial arguments are `listw`, `distance`, `type` and `bandwidth`. `stslshac` requires the specification of two different lists: one for the spatial weights matrix W and one to define the distance measure d. As in **spdep**, `listw` is the argument that handles the spatial weights matrix W in the form of a list. The object `listw` can be generated for example by the function `nb2listw` available in **spdep**. On the other hand, the argument `distance` that specifies the distance measure, is an object of class `distance` created for example by `read.gwt2dist`. Note that the two objects, although belonging to a different `class`, may be generated according to the same definition.

```
R> res <- stslshac(log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2)
+                  + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT),
+                  data = boston.c, listw,
+                  distance = coldist, type = "Triangular", HAC = TRUE)
R> summary(res)

======================================================
======================================================
                   Spatial Lag Model
                  HAC standard errors
======================================================
======================================================

Call:
stslshac(formula = log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) +
    I(RM^2) + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B +
    log(LSTAT), data = boston.c, listw = listw, HAC = TRUE, distance = coldist,
    type = "Triangular")

Residuals:
      Min.    1st Qu.     Median    3rd Qu.       Max.
-0.5356002 -0.0758562 -0.0045074  0.0719613  0.7128012

Coefficients:
```

```
               Estimate SHAC St.Er. t-value  Pr(>|t|)
Wy            0.45924669  0.05282792  8.6933 < 2.2e-16 ***
(Intercept)   2.40246917  0.28952447  8.2980 < 2.2e-16 ***
CRIM         -0.00735568  0.00157665 -4.6654 3.081e-06 ***
ZN            0.00036435  0.00034007  1.0714 0.2839913
INDUS         0.00119920  0.00161139  0.7442 0.4567549
CHAS1         0.01192878  0.03432896  0.3475 0.7282275
I(NOX^2)     -0.28873634  0.11796316 -2.4477 0.0143778 *
I(RM^2)       0.00669906  0.00206524  3.2437 0.0011798 **
AGE          -0.00025810  0.00047774 -0.5403 0.5890173
log(DIS)     -0.16042849  0.03681622 -4.3575 1.315e-05 ***
log(RAD)      0.07170438  0.01606094  4.4645 8.025e-06 ***
TAX          -0.00036857  0.00009780 -3.7685 0.0001642 ***
PTRATIO      -0.01295698  0.00394072 -3.2880 0.0010091 **
B             0.00028845  0.00013032  2.2134 0.0268689 *
log(LSTAT)   -0.23984212  0.03454865 -6.9422 3.862e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The argument `type` deals with the specification of the kernel function. Currently, six different kernels are available:

1. `"Epanechnikov"`: $K(z) = 1 - z^2$,
2. `"Triangular"`: $K(z) = 1 - z$,
3. `"Bisquare"`: $K(z) = (1 - z^2)^2$,
4. `"Parzen"`: $K(z) = 1 - 6z^2 + 6|z|^3$ if $z \leq 0.5$ and $K(z) = 2(1 - |z|)^3$ if $0.5 < z \leq 1$,
5. `"TH"` (Tukey - Hanning): $K(z) = \frac{1+\cos(\pi z)}{2}$,
6. `"QS"` (Quadratic Spectral): $K(z) = \frac{25}{12\pi^2 z^2}(\frac{\sin(6\pi z)/5)}{6\pi z/5} - \cos(6\pi z)/5))$.

If the kernel `type` is not one of the six implemented, the function will terminate with an error message. It is good practice to test the robustness of model specification to different Kernel functions. Note that if the argument `HAC` is set to `FALSE` (default is `TRUE`), the "classical" two stage least square estimator of the variance covariance matrix is provided.

```
R> res <- stslshac(log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2)
+                  + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT),
+                  data = boston.c, listw, distance = coldist, HAC = FALSE)
R> summary(res)

================================================
================================================
                  Spatial Lag Model
================================================
```

```
==================================================

Call:
stslshac(formula = log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) +
    I(RM^2) + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B +
    log(LSTAT), data = boston.c, listw = listw, HAC = FALSE,
    distance = coldist)

Residuals:
      Min.    1st Qu.     Median    3rd Qu.       Max.
-0.5356002 -0.0758562 -0.0045074  0.0719613  0.7128012

Coefficients:
               Estimate  Std. Error  t-value  Pr(>|t|)
Wy           4.5925e-01  3.8485e-02  11.9330 < 2.2e-16 ***
(Intercept)  2.4025e+00  2.1710e-01  11.0661 < 2.2e-16 ***
CRIM        -7.3557e-03  1.0345e-03  -7.1100 1.160e-12 ***
ZN           3.6435e-04  3.9311e-04   0.9268 0.3540112
INDUS        1.1992e-03  1.8365e-03   0.6530 0.5137794
CHAS1        1.1929e-02  2.6632e-02   0.4479 0.6542202
I(NOX^2)    -2.8874e-01  9.2546e-02  -3.1199 0.0018091 **
I(RM^2)      6.6991e-03  1.0192e-03   6.5728 4.938e-11 ***
AGE         -2.5810e-04  4.0940e-04  -0.6304 0.5284073
log(DIS)    -1.6043e-01  2.6107e-02  -6.1451 7.993e-10 ***
log(RAD)     7.1704e-02  1.4926e-02   4.8038 1.557e-06 ***
TAX         -3.6857e-04  9.5315e-05  -3.8668 0.0001103 ***
PTRATIO     -1.2957e-02  4.1334e-03  -3.1347 0.0017203 **
B            2.8845e-04  8.0266e-05   3.5937 0.0003261 ***
log(LSTAT)  -2.3984e-01  2.2470e-02 -10.6740 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that this last result corresponds to the one obtained by `stsls` in **spatialreg** (with `robust = FALSE`). Both functions have a logical argument (`W2X`) that if set to `TRUE` (the default) uses the matrix of instruments $H = (X, WX, W^2X)$ in the spatial two stages least squares. Since (**?**) show the advantages of including W^2X in the matrix of instruments, we strongly recommend to leave the argument `W2X` at its default value. In this example, no substantial differences are observed in terms of significance of the coefficients when using the robust estimator. It would be good practice to always estimate HAC standard errors at least to compare them with traditional results. If this leads to different significance levels, one should always present robust results.

```
R> res <- spatialreg::stsls(log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2)
+               + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT),
+               data = boston.c, listw = listw)
R> summary(res)
```

```
Call:spatialreg::stsls(formula = log(CMEDV) ~ CRIM + ZN + INDUS +
    CHAS + I(NOX^2) + I(RM^2) + AGE + log(DIS) + log(RAD) + TAX +
    PTRATIO + B + log(LSTAT), data = boston.c, listw = listw)

Residuals:
       Min         1Q     Median         3Q        Max
-0.5356002 -0.0758562 -0.0045074  0.0719613  0.7128012

Coefficients:
               Estimate  Std. Error  t value  Pr(>|t|)
Rho           4.5925e-01  3.8485e-02  11.9330 < 2.2e-16
(Intercept)   2.4025e+00  2.1710e-01  11.0661 < 2.2e-16
CRIM         -7.3557e-03  1.0345e-03  -7.1100 1.160e-12
ZN            3.6435e-04  3.9311e-04   0.9268 0.3540112
INDUS         1.1992e-03  1.8365e-03   0.6530 0.5137794
CHAS1         1.1929e-02  2.6632e-02   0.4479 0.6542202
I(NOX^2)     -2.8874e-01  9.2546e-02  -3.1199 0.0018091
I(RM^2)       6.6991e-03  1.0192e-03   6.5728 4.938e-11
AGE          -2.5810e-04  4.0940e-04  -0.6304 0.5284073
log(DIS)     -1.6043e-01  2.6107e-02  -6.1451 7.993e-10
log(RAD)      7.1704e-02  1.4926e-02   4.8038 1.557e-06
TAX          -3.6857e-04  9.5315e-05  -3.8668 0.0001103
PTRATIO      -1.2957e-02  4.1334e-03  -3.1347 0.0017203
B             2.8845e-04  8.0266e-05   3.5937 0.0003261
log(LSTAT)   -2.3984e-01  2.2470e-02 -10.6740 < 2.2e-16

Residual variance (sigma squared): 0.020054, (sigma: 0.14161)
```

`stsls` allows an heteroskedasticity correction to the coefficients' variance covariance matrix by setting the argument `robust` to `TRUE`. The additional argument `legacy` chooses between two different implementations of the robustness correction. When it is set to `FALSE` (the default used in our examples), a White consistent estimator of the variance-covariance matrix is provided. On the other hand, if `legacy` equals `TRUE` a GLS estimator is performed that yields different coefficient estimates. Results are displayed in the following example.

```
R> res <- spatialreg::stsls(log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2)
+               + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT),
+               data = boston.c, listw = listw, robust = TRUE)
R> summary(res)

Call:spatialreg::stsls(formula = log(CMEDV) ~ CRIM + ZN + INDUS +
    CHAS + I(NOX^2) + I(RM^2) + AGE + log(DIS) + log(RAD) + TAX +
    PTRATIO + B + log(LSTAT), data = boston.c, listw = listw,
    robust = TRUE)

Residuals:
       Min         1Q     Median         3Q        Max
```

```
-0.5356002 -0.0758562 -0.0045074  0.0719613  0.7128012

Coefficients:
               Estimate HC0 std. Error z value  Pr(>|z|)
Rho           4.5925e-01     4.4828e-02 10.2446 < 2.2e-16
(Intercept)   2.4025e+00     2.6000e-01  9.2401 < 2.2e-16
CRIM         -7.3557e-03     1.4999e-03 -4.9042 9.380e-07
ZN            3.6435e-04     3.2956e-04  1.1056 0.2689200
INDUS         1.1992e-03     1.5598e-03  0.7688 0.4420040
CHAS1         1.1929e-02     3.2084e-02  0.3718 0.7100470
I(NOX^2)     -2.8874e-01     1.0235e-01 -2.8211 0.0047852
I(RM^2)       6.6991e-03     1.7285e-03  3.8757 0.0001063
AGE          -2.5810e-04     4.3159e-04 -0.5980 0.5498210
log(DIS)     -1.6043e-01     3.0484e-02 -5.2627 1.420e-07
log(RAD)      7.1704e-02     1.5858e-02  4.5216 6.137e-06
TAX          -3.6857e-04     9.8735e-05 -3.7329 0.0001893
PTRATIO      -1.2957e-02     3.7330e-03 -3.4709 0.0005187
B             2.8845e-04     1.0412e-04  2.7703 0.0056004
log(LSTAT)   -2.3984e-01     3.1408e-02 -7.6365 2.243e-14

Residual variance (sigma squared): 0.020054, (sigma: 0.14161)
```

The heteroskedasticity correction leads to results similar to those obtained with the spatial HAC estimator implemented in **sphet** (on the Boston housing data). The spatial HAC seems to be more conservative in that it changes the significance level of some of the variables (i.e., `NOX2`, `RM2` and `B`).

The argument `bandwidth` by default sets the bandwidth for each observation to the maximum distance for that observation. Alternatively, a fixed bandwidth can be used as in the next example that fixes as bandwidth the maximum distance (overall).

```
R> fix <- max(unlist(attributes(coldist)$GeoDa$dist))
R> res <- stslshac(log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2)
+                  + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT),
+                  data = boston.c, listw, distance = coldist, type = "Parzen", bandwidth
R> summary(res)

====================================================
====================================================
                   Spatial Lag Model
                  HAC standard errors
====================================================
====================================================

Call:
stslshac(formula = log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) +
    I(RM^2) + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B +
```

```
    log(LSTAT), data = boston.c, listw = listw, distance = coldist,
    type = "Parzen", bandwidth = fix)

Residuals:
      Min.    1st Qu.     Median    3rd Qu.       Max.
-0.5356002 -0.0758562 -0.0045074  0.0719613  0.7128012

Coefficients:
               Estimate SHAC St.Er. t-value  Pr(>|t|)
Wy           0.45924669  0.05697902  8.0599 7.634e-16 ***
(Intercept)  2.40246917  0.31795278  7.5561 4.155e-14 ***
CRIM        -0.00735568  0.00188529 -3.9016 9.555e-05 ***
ZN           0.00036435  0.00038618  0.9435 0.3454415
INDUS        0.00119920  0.00168144  0.7132 0.4757243
CHAS1        0.01192877  0.03516686  0.3392 0.7344553
I(NOX^2)    -0.28873634  0.13602087 -2.1227 0.0337760 *
I(RM^2)      0.00669906  0.00269441  2.4863 0.0129088 *
AGE         -0.00025810  0.00055829 -0.4623 0.6438569
log(DIS)    -0.16042849  0.04435452 -3.6170 0.0002981 ***
log(RAD)     0.07170438  0.01742553  4.1149 3.873e-05 ***
TAX         -0.00036857  0.00010993 -3.3526 0.0008006 ***
PTRATIO     -0.01295698  0.00450533 -2.8759 0.0040285 **
B            0.00028845  0.00016362  1.7629 0.0779146 .
log(LSTAT)  -0.23984212  0.03955045 -6.0642 1.326e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

`summary` and `print` methods are available. The `summary` method provides a short description of the model followed by a printout of the most recent call, a summary of the residuals, and the table of the results. The first row of the table of estimated coefficients is always the spatial lag of the dependent variable Wy. Note that the name of the column of the standard errors clearly make reference to the use of the spatial HAC estimator. On the other hand, the `print` method simply prints basic information.

One final point deserves to be mentioned. As we anticipated in the introduction, we are not giving much attention to the ML approach because of various shortcomings. It is technically possible, however, to make heteroskedasticity corrections to standard errors in a ML context using functions in the **lmtest** (**?**) and **sandwich** (**??**) packages.[1] An example on how to perform the correction is given in the help file of the function `bptest.sarlm` available from **spdep**. We run this example on the Boston data set and we did not observe big shifts in the estimated coefficients or substantial differences in their significance.

3.2. General spatial two stage least squares

[1] As a referee correctly pointed out, to fully explore the ML approach one would need to implement `vcov` methods for `lagsarlm`. The theory has not yet been developed and therefore we leave this for future developments.

In Section 3.1 we assumed a very general form for the disturbance process. As an alternative, one could assume that the disturbance process is known to follow a first order spatial autoregressive process

$$\varepsilon = \rho W \varepsilon + \xi \tag{11}$$

where the innovations $\xi_1, \ldots, \xi_n$ are assumed independent with zero mean and non-constant variance σ_i^2. **?** define a GM estimator for ρ and give results for both consistency and asymptotic normality. Note that **?** had only proven consistency of the GM estimator under the assumption of homoskedastic innovations. In other words, ρ was seen as a nuisance parameter (e.g., the distribution of the regression's parameters does not depend on the estimator for ρ). **?** also give results for the joint asymptotic distribution of the GM estimator for ρ and the IV estimators for γ. The suggested estimation procedure consists of two steps alternating GM and IV estimators. Each of the two steps includes sub-steps. In the first step, γ is estimated by S2SLS with the matrix of instruments defined in Equation 4. The estimated residuals from the first step are employed to obtain a sample analogue of the moment conditions (for greater detail on the specification of the moments conditions see **??**). An initial GM estimator for ρ is defined in terms of these moment conditions and S2SLS residuals. The initial estimator can be viewed as an unweighted nonlinear least square estimator. Although fully consistent, it is not efficient because of a lack of weighting. This is why in the third sub-step of the first step, an efficient GM estimator for ρ is derived based on S2SLS residuals and moments conditions appropriately weighted by an estimator of the variance covariance matrix of the limiting distribution of the normalized sample moments. Following **?**, the consistent and efficient estimator for ρ is used to take a spatial Cochrane-Orcutt transformation of the model. In the second step of the estimation procedure the transformed model is estimated by S2SLS: this is the generalized spatial two-stage least square (GS2SLS) procedure presented in **?**. Specifically, the GS2SLS estimator for the parameter vector γ is defined as:

$$\tilde{\gamma}_{GS2SLS} = [\hat{Z_*^\top} Z_*]^{-1} \hat{Z_*^\top} y_* \tag{12}$$

where $y_* = y - \hat{\rho} W y$, $Z_* = Z - \hat{\rho} W Z$, $\hat{Z}_* = P Z_*$, and $P = H(H^\top H)^{-1} H^\top$. In the second and final sub-step of the second step, new sample moments are obtained by replacing S2SLS residuals by GS2SLS residuals obtained from Equation 12. The efficient GM estimator for ρ based on GS2SLS residuals is obtained from

$$\tilde{\rho} = \underset{\rho}{\operatorname{argmin}} [m(\rho, \hat{\gamma})^\top \hat{\Upsilon}^{-1} m(\rho, \hat{\gamma})] \tag{13}$$

where the weighting matrix $\hat{\Upsilon}^{-1}$ is an estimator of the variance covariance matrix of the limiting distribution of the sample moments. Under the assumptions made in **?**, $\tilde{\gamma}$ and $\tilde{\rho}$ are both consistent and asymptotically (jointly) normal. Therefore, small sample inference can be based on the following estimator for the variance covariance matrix:

$$\tilde{\Omega} = n^{-1} \begin{bmatrix} \tilde{P^\top} & 0 \\ 0 & (\tilde{J^\top \Upsilon^{-1} J})^{-1} \tilde{J^\top \Upsilon^{-1}} \end{bmatrix} \tilde{\Upsilon}_o \begin{bmatrix} \tilde{P} & 0 \\ 0 & \tilde{\Upsilon^{-1}} \tilde{J} (\tilde{J^\top \Upsilon^{-1} J})^{-1} \end{bmatrix}$$

where

$$\tilde{\Upsilon}_o = \begin{bmatrix} \tilde{\Upsilon}_{\gamma\gamma} & \tilde{\Upsilon}_{\gamma\rho} \\ \tilde{\Upsilon}_{\gamma\rho}^\top & \tilde{\Upsilon} \end{bmatrix}$$

and $\tilde{\Upsilon}_{\gamma\gamma} = n^{-1}H^\top\tilde{\Sigma}H$, $\tilde{\Upsilon}_{\gamma\rho} = n^{-1}H^\top\tilde{\Sigma}\tilde{a}$. Finally, P, J, Σ and a are all expressions (based on the instruments and the transformed variables) needed to estimate the asymptotic variance covariance matrix of the moment conditions. The tilde explicitly denotes that all quantities are based on estimated residuals from GS2SLS procedure.

As a final point, a joint test of significance on the spatial coefficients can be derived. Let B be a $1 \times (K+2)$ matrix and $\tilde{\theta} = [\tilde{\gamma}, \tilde{\rho}^\top]^\top$ the $(K+2) \times 1$ vector of estimated parameters (including the two spatial parameters). A restriction (e.g., that both spatial parameters are zero) can then be formulated in terms of

$$B\theta = 0$$

A Wald test can then be based on (**?**):

$$Wald = [B\tilde{\theta}]^\top[B\tilde{\Omega}B^\top]^{-1}[B\tilde{\theta}] \tag{14}$$

and under H_0 will have a chi-squared distribution with one degree of freedom (i.e., the number of rows in B).

A complication appears from a computational perspective. The estimation of the variance covariance matrix of the limiting distribution of the normalized sample moments based on two stages least squares residuals involves the inversion of an $n \times n$ matrix. This is the main reason for transforming the object of class `listw` into a sparse `Matrix` and use code from the **Matrix** package to calculate the inverse. However, for very large problems the inversion could still be computationally intensive. In these cases the inverse can be calculated using the approximation

$$(I - \rho W)^{-1} = I + \rho W + \rho^2 W^2 + ... + \rho^n W^n. \tag{15}$$

where the last element of the sum depends on each specific W and ρ and is determined through a condition. [2] For particular spatial weights matrices the results obtained using the approximation could be very close to the actual inverse of the variance covariance matrix. Furthermore, the inverse only influences the expression of the estimated variance covariance matrix of the limiting distribution of the normalized sample moments based on 2SLS residuals. In other words, small differences in the weighting matrix may imply even smaller differences in the estimated value of the spatial parameter resulting from the optimization procedure. As an example, on the Boston data the value of ρ resulting from the correct inverse is 0.1778462. If using the the approximation the value of ρ turns out to be 0.1779276. We would suspect that with larger datasets (for sparse W) the difference should be even smaller.

A second issue is related to the initial values of the optimization search for the parameter ρ.[3] The default is to start from 0.2. As an alternative the user can either specify a different value or take as initial value the estimated coefficient of a regression of the S2SLS residuals on their spatial lag (i.e., fit a spatial autoregressive model on the residuals). The initial value in successive steps is the optimal parameter in previous steps.

[2] Roughly speaking, the function will keep adding terms until the absolute value of the `sum` of all elements of the matrix $\rho^i W^i$ is greater than a fixed ϵ.

[3] After checking different alternatives, we decided to use the function `nlminb` in the optimization of the objective function since it appears to reduce computational time.

Demonstration

The function that allows estimating the model described in this Section is `gstslshet`. It is also possible to estimate a restricted version of the model for which the parameter λ is set to zero by changing the logical argument `sarar`. Such a model is generally referred to in the literature as a spatial error model (**?**). The syntax of the function is straightforward in its basic arguments. The model to be estimated is described by a `formula`, an optional `data.frame` can be specified, and the spatial weights matrix is an object of class `listw`. The argument `initial.value` manages the starting point of the optimization process in the search for the optimal ρ. The default value for `initial.value` is `0.2`. Any other numeric value (within the search interval) is acceptable. Alternatively, if `initial.value` is set to `"SAR"` the optimization will start from the estimated coefficient of a regression of the 2SLS residuals over their spatial lag.

```
R> res <- gstslshet(log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2)
+                   + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT),
+                   data = boston.c, listw = listw, initial.value = 0.2)
R> summary(res)

Call:
gstslshet(formula = log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) +
    I(RM^2) + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B +
    log(LSTAT), data = boston.c, listw = listw, initial.value = 0.2)

Residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.56939 -0.07316 -0.00168  0.00053  0.07150  0.74031

Coefficients:
              Estimate  Std. Error t-value  Pr(>|t|)
(Intercept)  2.51316605  0.26749367  9.3952 < 2.2e-16 ***
CRIM        -0.00662744  0.00144522 -4.5858 4.523e-06 ***
ZN           0.00038299  0.00036563  1.0475 0.2948733
INDUS        0.00159352  0.00179772  0.8864 0.3753967
CHAS1       -0.00447974  0.03689065 -0.1214 0.9033480
I(NOX^2)    -0.27295899  0.11561412 -2.3609 0.0182283 *
I(RM^2)      0.00744059  0.00199637  3.7270 0.0001937 ***
AGE         -0.00045400  0.00045572 -0.9962 0.3191333
log(DIS)    -0.16517174  0.03484858 -4.7397 2.140e-06 ***
log(RAD)     0.07453521  0.01752830  4.2523 2.116e-05 ***
TAX         -0.00041956  0.00010763 -3.8981 9.697e-05 ***
PTRATIO     -0.01412661  0.00410143 -3.4443 0.0005725 ***
B            0.00035970  0.00011182  3.2168 0.0012964 **
log(LSTAT)  -0.24593826  0.03213364 -7.6536 1.954e-14 ***
lambda       0.42407826  0.04463747  9.5005 < 2.2e-16 ***
rho          0.29587455  0.08614291  3.4347 0.0005932 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The argument `inverse` (default: `TRUE`) should be altered only when strictly necessary depending on the number of cross sectional observations. In this case, the inverse will be calculated by Equation 15. The precision of the approximation can be managed through the argument `eps`. The next example illustrates the use of the approximated inverse in the context of a model where λ is assumed to be zero (`sarar = FALSE`).

```
R> res <- gstslshet(log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2)
+                   + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT),
+                   data = boston.c, listw = listw, initial.value = 0.2,
+                   inverse = FALSE, eps = 1e-18, sarar = FALSE )
R> summary(res)

Call:
gstslshet(formula = log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) +
    I(RM^2) + AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B +
    log(LSTAT), data = boston.c, listw = listw, initial.value = 0.2,
    eps = 1e-18, inverse = FALSE, sarar = FALSE)

Residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.83800 -0.09179 -0.00405  0.00096  0.09557  0.89644

Coefficients:
               Estimate  Std. Error t-value  Pr(>|t|)
(Intercept)  4.03649663  0.24703623 16.3397 < 2.2e-16 ***
CRIM        -0.00660146  0.00136354 -4.8414 1.289e-06 ***
ZN           0.00027056  0.00041940  0.6451  0.518864
INDUS        0.00039648  0.00244150  0.1624  0.870997
CHAS1       -0.00905744  0.04181711 -0.2166  0.828523
I(NOX^2)    -0.35168188  0.16150148 -2.1776  0.029438 *
I(RM^2)      0.00778390  0.00249859  3.1153  0.001838 **
AGE         -0.00078626  0.00052412 -1.5002  0.133573
log(DIS)    -0.13775233  0.05362777 -2.5687  0.010209 *
log(RAD)     0.07034884  0.02122046  3.3151  0.000916 ***
TAX         -0.00049033  0.00012096 -4.0538 5.040e-05 ***
PTRATIO     -0.02181338  0.00466238 -4.6786 2.888e-06 ***
B            0.00056242  0.00012377  4.5441 5.516e-06 ***
log(LSTAT)  -0.29352100  0.03656123 -8.0282 9.891e-16 ***
rho          0.67496196  0.04584224 14.7236 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

After a brief description of the model, the `summary` method prints the most recent call along with a summary of the residuals and the table of the estimated coefficients. The last rows of the table contain the spatial coefficients: the spatial autoregressive (λ) and the spatial autocorrelation (ρ) coefficients for the full model, and the spatial autocorrelation coefficient when the option `sarar` is `FALSE`. For the general case of the unrestricted model, after the table

of estimated coefficients, the `summary` method reports the results of a Wald test that both spatial coefficients are zero. The `p-val` can be used to test the significance of the chi-squared statistics. As before, the `print` method only displays basic information.

4. Conclusions and future development

The **sphet** package currently contains most of the newly developed robust methodologies in spatial econometrics (**??**).

A possible addition could be the implementation of the methodology proposed in **?** and **?**. Also several alternative HAC estimators could be added such as, for example, those presented in **???**

Other planned additions include **?** and **?** that relate to a panel data model in which the number of time periods T limits to infinity.

Acknowledgments

Earlier developments of the present library were presented and discussed during various editions of the Spatial Econometrics Advanced Institute. The valuable input from all participants is gratefully acknowledged. The research was partly supported by the Nuclei of the Millennium Science Initiative Program "Regional Sciences and Public Policy" (MIDEPLAN – Chile). Usual disclaimers apply.

Affiliation:

Gianfranco Piras
Department of City and Regional Planning
Cornell University
and
IDEAR, Universidad Catolica del Norte
323 West Sibley Hall
Ithaca, New York 14853 USA
E-mail: gpiras@mac.com