NHH

INSTITUTT FOR SAMFUNNSØKONOMI

DEPARTMENT OF ECONOMICS

SAM 26 2010

ISSN: 0804-6824

OCTOBER 2010

Discussion paper

Comparing estimation methods for spatial econometrics techniques using R

BY

ROGER BIVAND

Norges Handelshøyskole

NORWEGIAN SCHOOL OF ECONOMICS AND BUSINESS ADMINISTRATION

Comparing estimation methods for spatial econometrics techniques using R

Roger Bivand
Norwegian School of Economics and Business Administration*

Abstract

Recent advances in spatial econometrics model fitting techniques have made it more desirable to be able to compare results and timings. Results should correspond between implementations using different applications, while timings are more readily compared within a single application. A broad range of model fitting techniques are provided by the contributed R packages for spatial econometrics. These model fitting techniques are associated with methods for estimating impacts and some tests, which will also be presented and compared. This review constitutes an up-to-date demonstration of techniques now available in R, and mentions some that will shortly become more generally available.

1 Background

Researchers applying spatial econometrics to empirical economic questions now have a wide range of tools, and a growing literature supporting these tools. During the 1990s, it was typical for researchers to use tools coded in Fortran or other general programming languages, or to seek to integrate functions into existing statistical and/or matrix language environments. The use of spatial econometrics tools was widened by the ease with which methods and examples presented in Anselin (1988) could be reproduced using SpaceStat™, written in Gauss™, and shipped as a built runtime module. It was rapidly complemented by the Spatial Econometrics toolbox for Matlab™, provided as source code together with extensive documentation.[1] This toolbox is under active development, and accepts contributed functions,

*Department of Economics, Norwegian School of Economics and Business Administration, Helleveien 30, N-5045 Bergen, Norway; E-mail: Roger.Bivand@nhh.no

[1] http://www.spatial-econometrics.com/.

thus broadening its appeal. In addition Griffith and Layne (1999) gave code listings for model fitting techniques using SAS™ and SPSS™. A suite of commands for spatial data analysis for use with Stata™ was provided by Maurizio Pisati, and distributed using the standard contributed command system (Pisati, 2001).

The thrust of SpaceStat™ has largely been taken over by GeoDa (Anselin et al., 2006), and more recently by OpenGeoDa.[2] The same team has just launched the Python spatial analysis library.[3] Since the R language and environment became available in the later 1990s, collaborative code development has proceeded with varying speed. Initial attempts to implement spatial econometrics techniques were checked against SpaceStat™, and subsequently against Maurizio Pisati's Stata™ code and GeoDa by comparing results for the same input data and spatial weights (Bivand and Gebhardt, 2000; Bivand, 2002).

More recently, comparisons on the same hardware under Linux have been made using OpenGeoDa under Wine,[4] and using Octave[5] instead of Matlab™ with the Spatial Econometrics toolbox. The source code of the R **spdep** package is available from the Comprehensive R Archive Network (CRAN), and the current development status is accessible at R-Forge;[6] binary packages are also available at CRAN.

In the spirit of Rey (2009), this comparison will attempt to examine some features of the implementation of functions for fitting spatial econometrics models in **spdep** with those in the Spatial Econometrics toolbox (release 7, GNU Octave 3.0.5) and in OpenGeoDa (release 0.9.8.14, Wine 1.0.1). In addition, associated measures will also be compared. Within the Spatial Econometrics toolbox and **spdep**, it is possible to choose between technical details in implementation, and the consequences of such choices will also be considered.

The analysis has been carried out on an Intel Core-2 Duo 64-bit system with 4GB RAM running R 2.11.1 (R Development Core Team, 2010), **Matrix** 0.999375-43, and **spdep** 0.5-21, under Red Hat Enterprise Linux 5; a threaded GotoBLAS 1.26 library optimised for the hardware was used, with gfortran 4.1.2 for Fortran compilation. Two data sets distributed with **spdep** are used; both originated from the Spatial Econometrics toolbox, and are provided here with pre-build lists of spatial neighbours. A broad survey of the analysis of spatial data in the R environment is given by Bivand (2006); Bivand et al. (2008).

[2] `http://geodacenter.asu.edu/ogeoda`, source code not yet exposed at: `http://code.google.com/p/opengeoda/`.

[3] `http://code.google.com/p/pysal/`.

[4] Wine emulates the Microsoft™ Windows™ operating environment

[5] `http://www.gnu.org/software/octave`.

[6] `https://r-forge.r-project.org/projects/spdep/`

1.1 US 1980 election turnout data set

The US county data set with 3107 observations includes a 1980 Presidential election turnout variable with a single county (Hinsdale County, CO) with a value over unity — most likely from cross-border voting in this remote rural area. We define a formula relating this variable to income ($1000) per inhabitant over age 19, the number with college degrees as a proportion of all over age 19, and homeownership as a proportion of all over age 19. The right hand side variables are taken as logarithms, as in the file `data/elect.txt` in the Spatial Econometrics toolbox.

```
> library(spdep)
> data(elect80)
> eform <- formula(pc_turnout ~ log(pc_income) + log(pc_college) +
+     log(pc_homeownership))
```

A shapefile is written for OpenGeoDa after adding the logarithms of the right hand side variables to the `SpatialPointsDataFrame` object `elect80`:

```
> elect80$l_pc_income <- log(elect80$pc_income)
> elect80$l_pc_college <- log(elect80$pc_college)
> elect80$l_pc_homeownership <- log(elect80$pc_homeownership)
> writeSpatialShape(elect80, "elect80")
```

Similarly, a model matrix is generated using the formula defined above and the `elect80` object, and augmented with the dependent variable for export to be used in Octave:

```
> mm0 <- model.matrix(eform, data = elect80)
> mm <- cbind(elect80$pc_turnout, mm0)
> write.table(mm, file = "Elect80.txt", row.names = FALSE,
+     col.names = FALSE)
```

The data set provided in **spdep** includes a number of `nb` objects listing the neighbours of the counties in the data set using different definitions. Here we will use a Queen contiguity scheme constructed using a shapefile from the USGS National Atlas site, file: `co1980p020.tar.gz`. This object contains four counties with no neighbours:

```
> e80_queen

Neighbour list object:
Number of regions: 3107
Number of nonzero links: 18126
Percentage nonzero weights: 0.1877671
Average number of links: 5.833923
4 regions with no links:
1183 1189 1832 2945
```

Because of this, an option is set to permit computations under the assumption that the lagged value of a variable for a county with no neighbours may be set to zero (Bivand and Portnov, 2004). We write a GAL file to be read into OpenGeoDa, and a triplet-type sparse matrix text file to read into Octave:

```
> set.ZeroPolicyOption(TRUE)
> write.nb.gal(e80_queen, file = "e80_queen.gal")
> elw <- nb2listw(e80_queen)
> esn <- listw2sn(elw)
> write.sn2dat(esn, file = "E80_queen_W.txt")
```

In order to be confident that all three applications, **spdep**, the Spatial Econometrics toolbox and OpenGeoDa, are working on the same data and row-standardised spatial weights, we fit a linear model using the usual R function `lm`, and test its residuals for spatial autocorrelation using functions provided in **spdep**:

```
> eout_lm <- lm(eform, data = elect80)
> eout_lm_I <- lm.morantest(eout_lm, elw)
> eout_lm_LM <- lm.LMtests(eout_lm, elw, test = "all")
```

Table 1: Comparison of US 1980 election turnout OLS results.

	R/spdep	SE toolbox	OpenGeoDa
(Intercept)	1.5021	1.5021	1.5021
log(pc_income)	-0.2029	-0.2029	-0.2029
log(pc_college)	0.3297	0.3297	0.3297
log(pc_homeownership)	0.2504	0.2504	0.2504
Moran's I	0.457	0.457	0.457
LMerr	1789.2	1789.2	1789.2
LMlag	1375.9		1375.9
RLMerr	461.2		461.2
RLMlag	47.9		47.9
SARMA	1837.1		1837.1

As can be seen from Table 1, all three applications provide identical results for the fitted coefficients, and for the residual spatial autocorrelation statistics provided, using the Classic choice in OpenGeoDa, and the `ols`, `moran` and `lmerror` commands in the Spatial Econometrics toolbox.

1.2 Lucas County, OH, housing data set

The Lucas County, Ohio, housing data set has 25,357 observations of single family homes sold 1993–1998, and is fully described in the file `data/house.txt` in the Spatial Econometrics toolbox. It is used here to supplement conclusions drawn for

the 1980 US election turnout data set, which is of a size that permits dense matrix methods, since only sparse or approximate methods are feasible for larger N.

```
> data(house)
> hform <- formula(log(price) ~ age + I(age^2) + I(age^3) +
+     log(lotsize) + rooms + log(TLA) + beds + syear)
> mm0 <- model.matrix(hform, data = house)
> mm <- cbind(log(house$price), mm0)
> write.table(mm, file = "House.txt", row.names = FALSE,
+     col.names = FALSE)
> mmdf <- as.data.frame(mm)
> coordinates(mmdf) <- coordinates(house)
> writeSpatialShape(mmdf, "mm_house")
```

Once again, we write out the model matrix and dependent variable for reading into Octave. The dependent variable is the logarithm of the sale price, and the right hand side variables are powers of the scaled age of the house, the logarithm of the lotsize in square feet, the number of rooms, the logarithm of the total living area in square feet, the number of bedrooms, and year of sale dummy variables represented as a factor variable in R. Because of the relative complexity of the model matrix, a new SpatialPointsDataFrame is constructed from it for output to OpenGeoDa.

The list of neighbours provided with the data set in **spdep** is a sphere of influence graph constructed from a triangulation of the point coordinates of the houses after projection to the Ohio North NAD83 (HARN) Lambert Conformal Conical specification (EPSG:2834). It is relatively sparse, with less than three neighbours per observation on average:

```
> LO_nb

Neighbour list object:
Number of regions: 25357
Number of nonzero links: 74874
Percentage nonzero weights: 0.01164489
Average number of links: 2.952794
```

Again, the neighbours are output in row-standardised form to be read into Octave and OpenGeoDa:

```
> write.nb.gal(LO_nb, "LO_nb.gal")
> hlw <- nb2listw(LO_nb)
> hsn <- listw2sn(hlw)
> write.sn2dat(hsn, file = "House_W.txt")
```

The ouput (Table 2) from fitting linear models in each of the three applications shows that the data and the spatial weights used are the same in all three cases. Having established this, we can be confident that any differences observed below stem from differences in implementation across and within the applications, rather than from differences in data.

```
> hout_lm <- lm(hform, data = house)
> hout_lm_I <- lm.morantest(hout_lm, hlw)
> hout_lm_LM <- lm.LMtests(hout_lm, hlw, test = "all")
```

Table 2: Comparison of Lucas county (OH) house price OLS results.

	R/spdep	SE toolbox	OpenGeoDa
(Intercept)	2.900533	2.900533	2.900533
age	1.938229	1.938229	1.938229
I(age∧2)	-3.981144	-3.981144	-3.981144
I(age∧3)	1.183394	1.183394	1.183394
log(lotsize)	0.176678	0.176678	0.176678
rooms	0.009485	0.009485	0.009485
log(TLA)	0.900787	0.900787	0.900787
beds	-0.016600	-0.016600	-0.016600
syear1994	0.044930	0.044930	0.044930
syear1995	0.087001	0.087001	0.087001
syear1996	0.109115	0.109115	0.109115
syear1997	0.145471	0.145471	0.145471
syear1998	0.201824	0.201824	0.201824
Moran's I	0.4897		0.4897
LMerr	7511.4	7511.4	7511.4
LMlag	10400.1		10400.1
RLMerr	123.7		123.7
RLMlag	3012.4		3012.4
SARMA	10523.8		10523.8

2 Comparing estimation methods

The spatial lag model (Cliff and Ord, 1973; Ord, 1975; Bivand, 1984; Anselin, 1988; LeSage and Pace, 2009) is the most frequently encountered specification in spatial econometrics:

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

where $\mathbf{y}$ is an $(N \times 1)$ vector of observations on a dependent variable taken at each of N locations, $\mathbf{X}$ is an $(N \times k)$ matrix of exogenous variables, β is an $(k \times 1)$ vector of parameters, ε is an $(N \times 1)$ vector of independent and identically distributed disturbances and ρ is a scalar spatial lag parameter.

In the spatial Durbin model, the spatially lagged exogenous variables are added to the model:

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

where γ is an $((k-1)\times 1)$ vector of parameters where $\mathbf{W}$ is row-standardised, and a $(k\times 1)$ vector otherwise. It is clear that these two models are estimated in the same way.

The spatial error model may be written as (Cliff and Ord, 1973; Ord, 1975; Ripley, 1981; Anselin, 1988; LeSage and Pace, 2009):

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{u}, \qquad \mathbf{u} = \lambda\mathbf{W}\mathbf{u} + \varepsilon,$$

where λ is a scalar spatial error parameter, and $\mathbf{u}$ is a spatially autocorrelated disturbance vector with constant variance and covariance terms specified by a fixed spatial weights matrix and a single coefficient λ:

$$\mathbf{u} \sim N(0, \sigma^2(\mathbf{I} - \lambda\mathbf{W})^{-1}(\mathbf{I} - \lambda\mathbf{W}')^{-1})$$

When the Common Factor condition is met: $\beta = -\rho\gamma$, the spatial Durbin and spatial error models are equivalent. We will not be considering the general model with both a spatial lag and a spatial error term here.

These models may be estimated using a number of approaches, among which maximum likelihood estimation has a strong position, and also forms the basis for Bayesian estimation (Bayesian estimation is not discussed here). In addition, spatial two stage least squares and generalized method of moments approaches are prefered by some analysts (Kelejian and Prucha, 1998, 1999); these may be extended to provide a heteroskedasticity and autocorrelation consistent (HAC) estimator (Kelejian and Prucha, 2007; Piras, 2010). Finally, matrix exponential methods may be used to fit spatial regression models (LeSage and Pace, 2007).

2.1 Maximum likelihood estimation

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

$$\ell(\beta, \rho, \sigma^2) = -\frac{N}{2}\ln 2\pi - \frac{N}{2}\ln\sigma^2 + \ln|\mathbf{I} - \rho\mathbf{W}|$$
$$-\frac{1}{2\sigma^2}\left[((\mathbf{I} - \rho\mathbf{W})\mathbf{y} - \mathbf{X}\beta)'((\mathbf{I} - \rho\mathbf{W})\mathbf{y} - \mathbf{X}\beta)\right]$$

and by extension the same framework is used for the spatial Durbin model when $[\mathbf{W}(\mathbf{W}\mathbf{X})]$ are grouped together. Since β can be expressed as $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{I} - \rho\mathbf{W})\mathbf{y}$, all of the cross-product terms can be pre-computed as cross-products of the residuals of two ancilliary regressions: $\mathbf{y} = \mathbf{X}\beta_1$ and $\mathbf{W}\mathbf{y} = \mathbf{X}\beta_2$, and the sum of squares term can be calculated much faster than the log determinant (Jacobian) term of the $N \times N$ sparse matrix $\mathbf{I} - \rho\mathbf{W}$; see LeSage and Pace (2009) for details.

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

$$\ell(\beta, \lambda, \sigma^2) = -\frac{N}{2}\ln 2\pi - \frac{N}{2}\ln\sigma^2 + \ln|\mathbf{I} - \lambda\mathbf{W}|$$
$$-\frac{1}{2\sigma^2}\left[(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{I} - \lambda\mathbf{W})'(\mathbf{I} - \lambda\mathbf{W})(\mathbf{y} - \mathbf{X}\beta)\right]$$

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

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

where $\mathbf{Q}_\lambda$ is obtained by decomposing $(\mathbf{X} - \lambda\mathbf{W}\mathbf{X}) = \mathbf{Q}_\lambda\mathbf{R}_\lambda$.

The relationship between the log determinant term and the sum of squares term in the log likelihood function in the spatial error model is analogous to that in the spatial lag model, but the sum of squares term involves more computation in the case of the spatial error model. In all cases, a simple line search may be used to find ρ or λ, and other coefficients may be calculated using an ancilliary regression once this has been done.

Detailed reviews of methods for computing the Jacobian may be found in LeSage and Pace (2009); Smirnov and Anselin (2009); Bivand (2010), and interested readers are referred to these. The comparisons within **spdep** made here use methods for computing the Jacobian presented in full in Bivand (2010), and include the dense matrix eigenvalue method `eigen` (Ord, 1975, p. 121), the updating Cholesky decomposition method `Matrix` using functions in the R **Matrix** package for sparse matrix operations, the Monte Carlo method `MC` using the R **Matrix** package introduced by Barry and Pace (1999), and the Chebyshev method again using the R **Matrix** package (Pace and LeSage, 2004).

When sparse matrix methods or approximations are used, motivated by the size of N, no standard errors for the coefficients in spatial lag and spatial Durbin models will be available, nor will the standard error of λ be available in the spatial error case. This may be addressed by computing a numerical Hessian for an augmented function fitting both ρ or λ and β starting at the line search maximum likelihood optimum. If there are variables that are nearly collinear, or if variables are poorly scaled, then inverting the numerical Hessian (in R computed using `fdHess` in **nlme** or using `optim`) will lead to problems for those variables, with standard errors being set to `NA`.

We will fit the basic spatial lag model using two different methods to calculate the Jacobian:

```
> elag_ML_eigen <- lagsarlm(eform, data = elect80, listw = elw,
+     method = "eigen")

> elag_ML_Matrix <- lagsarlm(eform, data = elect80, listw = elw,
+     method = "Matrix")
```

Table 3: Comparison of US 1980 election turnout ML spatial lag results; for **spdep** and OpenGeoDa, z-values in parentheses, for SE toolbox, t-values in parentheses.

	R/spdep eigen	R/spdep Matrix	SE toolbox	OpenGeoDa
rho	0.5547	0.5547	0.54800	0.5544
	(34.715)	(38.09)	(30.44)	(34.680)
(Intercept)	0.7805	0.7805	0.78918	0.7809
	(25.352)	(26.514)	(25.87)	(25.363)
log(pc_income)	-0.0895	-0.0895	-0.09087	-0.0896
	(-9.698)	(-9.677)	(-7.34)	(-9.705)
log(pc_college)	0.1568	0.1568	0.15887	0.1569
	(17.952)	(18.057)	(31.73)	(17.962)
log(pc_homeownership)	0.2142	0.2142	0.21465	0.2142
	(25.680)	(26.370)	(37.81)	(25.681)
Log likelihood	3943.8	3943.8	5013.4	3943.7
Sigma squared	0.004333	0.004333	0.0043	0.004334
seconds	29.90	0.254		

As we would expect, because the eigenvalue and updating sparse Cholesky methods are both exact within machine precision, they find the same value for ρ — they are after all using the same line search function `optimize`, with the same termination criterion. The z-values differ somewhat, because those for the eigenvalue method use dense matrix techniques to find the coefficient standard errors, while the `Matrix` method approximates using a numerical Hessian. The big difference is in the timings, with the calculation of the eigenvalues and operations on large dense matrices, even using a threaded, optimised linear algebra BLAS library, taking almost half a minute, as compared with just half a second for the `Matrix` method.

The coefficients and z-values returned by OpenGeoDa agree closely with those from **spdep** — z-values possibly because of use of an effective algorithm (Smirnov, 2005, no documentation of OpenGeoDa algorithms is available at this time), while those from the Spatial Econometrics toolbox differ somewhat. The main reason for the difference is that the Jacobian values are computed using a Monte Carlo method on a grid, leading to an approximate result for ρ, rather than the search for ρ being continued. This application also returns concentrated log likelihood values, rather than the full values, which has no influence on the results. When $N > 500$, the variance-covariance matrix of coefficients is computed using a finite

difference Hessian implementation. The gridded MC ρ value leads to the optimum being marginally offset, and this feeds through into the computation of the variance-covariance matrix, although inferences would not be affected.

Timings are not given for the non-R applications, because it is not known to what extent the use of Octave rather than Matlab™, or of Wine rather than the application's native platform, might bias counts. We will return to the spatial Durbin model below in connection with the calculation of impact measures. We repeat the comparison for the spatial error specification:

```
> eerr_ML_eigen <- errorsarlm(eform, data = elect80, listw = elw,
+     method = "eigen")

> eerr_ML_Matrix <- errorsarlm(eform, data = elect80, listw = elw,
+     method = "Matrix")
```

Table 4: Comparison of US 1980 election turnout ML spatial error results; for **spdep** and OpenGeoDa, z-values in parentheses, for SE toolbox, t-values in parentheses.

	R/spdep eigen	R/spdep Matrix	SE toolbox	OpenGeoDa
lambda	0.7159	0.7159	0.708	0.7152
	(45.422)	(48.16)	(143.715)	(45.3062)
(Intercept)	1.2029	1.2029	1.207	1.2033
	(36.836)	(36.836)	(128.254)	(36.8561)
log(pc_income)	-0.1086	-0.1086	-0.110	-0.1087
	(-9.029)	(-9.029)	(-28.598)	(-9.0409)
log(pc_college)	0.1794	0.1794	0.182	0.1796
	(14.671)	(14.671)	(21.296)	(14.6954)
log(pc_homeownership)	0.2564	0.2564	0.256	0.2564
	(30.191)	(30.191)	(31.511)	(30.1862)
Log likelihood	4056.8	4056.8	5121.7	4056.5
Sigma squared	0.003812	0.003812	0.0038	0.003813
seconds	30.16	0.563		

Again we see from Table 4 that the eigenvalue and sparse matrix methods in **spdep** give the same coefficient estimates, and z-values that are the same apart from that for λ. Since the variance-covariance matrix is block-diagonal, the imprecision in the estimates for the variance of λ and σ^2 do not affect those for β. The OpenGeoDa coefficient and standard error estimates are very close to those of **spdep**, possibly for the reasons noted above. The Spatial Econometrics toolbox estimates again differ because of the use of a gridded MC Jacobian but are close to the other implementations (t-values are reproduced, but are not from the latest updated release). The timings for the **spdep** functions include the computation of the variance-covariance matrix under the alternative required for the Hausman test

described in Section 3.2 below. Dense matrices are used for the eigenvalue method, while sparse powering is used otherwise (LeSage and Pace, 2009, 110–113).

Moving on to the larger data set, we use the updating sparse Cholesky method and the Monte Carlo method for computing the Jacobian, the latter to compare more fairly with the Spatial Econometrics toolbox estimates:

```
> hlag_ML_Matrix <- lagsarlm(hform, data = house, listw = hlw,
+     method = "Matrix")
> set.seed(100831)
> hlag_ML_MC <- lagsarlm(hform, data = house, listw = hlw,
+     method = "MC")
```

Encouragingly, all three applications reported in Table 5, and both methods for computing the Jacobian in **spdep**, yield very similar estimates for fitting the spatial lag model to the Lucas county housing price data set ($N = 25357$). The Monte Carlo approximation used with continuous line search in the second column is very close to the estimates from the use of the updating sparse Cholesky method, but takes only half the time — although neither 5 seconds nor 2.5 seconds can be considered excessive for fitting a model with large N and numerous right hand side variables.

Completing this discussion, we examine three **spdep** Jacobian methods together with results from the two other applications:

```
> herr_ML_Matrix <- errorsarlm(hform, data = house, listw = hlw,
+     method = "Matrix", control = list(compiled_sse = TRUE))
> set.seed(100831)
> herr_ML_MC <- errorsarlm(hform, data = house, listw = hlw,
+     method = "MC", control = list(compiled_sse = TRUE))
> herr_ML_Chebyshev <- errorsarlm(hform, data = house,
+     listw = hlw, method = "Chebyshev", control = list(compiled_sse = TRUE))
```

Table 6 again shows a reassuring level of agreement between the spatial error estimates from the applications and different **spdep** implementations. The **spdep** timings here use compiled code for computing the sum of squares term in the log likelihood function in both the line search and the computation of the Hessian; it reduces run time somewhat for larger N. Almost half of the time taken to fit the spatial error model using the Chebyshev method is in fact spent on preparing the variance-covariance matrix for the Hausman test described in Section 3.2 below — this may be dropped, but the default is to provide it to encourage use of the test.

2.2 The analytical-numerical mixed Hessian

With some data sets, models, and variable scaling — fortunately not those used in these examples, one meets difficulties in inverting the numerical Hessian returned from finite difference computation. This unfortunate problem may be worked around

Table 5: Comparison of Lucas county (OH) house price spatial lag results; for **spdep** and OpenGeoDa, z-values in parentheses, for SE toolbox, t-values in parentheses.

	R/spdep Matrix	R/spdep MC	SE toolbox	OpenGeoDa
rho	0.522814	0.521996	0.51700	0.522754
	(139.8)	(139.0)	(221.3439)	(132.4212)
(Intercept)	0.258328	0.262460	0.28771	0.258630
	(4.037)	(4.0643)	(12.2487)	(3.9653)
age	1.308469	1.309454	1.31547	1.308541
	(23.303)	(23.2933)	(23.5627)	(23.1779)
I(age∧2)	-2.321326	-2.323922	-2.33978	-2.321516
	(-22.693)	(-22.7298)	(-23.0196)	(-22.5843)
I(age∧3)	0.654895	0.655721	0.66077	0.654955
	(11.876)	(11.9007)	(11.9974)	(11.8353)
log(lotsize)	0.072975	0.073138	0.07413	0.072987
	(23.874)	(23.9084)	(25.7704)	(23.5512)
rooms	-0.002534	-0.002515	-0.00240	-0.002533
	(-1.137)	(-0.9998)	(-0.9948)	(-0.8326)
log(TLA)	0.577833	0.578338	0.58142	0.577870
	(58.240)	(57.8472)	(103.6739)	(56.7161)
beds	0.015621	0.015571	0.01526	0.015618
	(4.066)	(3.8116)	(3.4030)	(3.4570)
syear1994	0.044475	0.044476	0.04448	0.044475
	(5.999)	(5.9876)	(6.0118)	(6.0197)
syear1995	0.086074	0.086075	0.08608	0.086074
	(11.923)	(11.8588)	(11.9127)	(11.9290)
syear1996	0.105937	0.105942	0.10597	0.105938
	(15.134)	(15.0689)	(15.1431)	(15.1591)
syear1997	0.147347	0.147344	0.14733	0.147347
	(21.226)	(21.1113)	(21.2249)	(21.2555)
syear1998	0.200722	0.200723	0.20073	0.200722
	(28.135)	(28.0305)	(28.1265)	(28.1600)
Log likelihood	-7670.4	-7693	961.1	-7670.8
Sigma squared	0.09479	0.09483	0.0951	0.09479
seconds	2.042	1.553		

Table 6: Comparison of Lucas county (OH) house price spatial error results; for **spdep** and OpenGeoDa, z-values in parentheses, for SE toolbox, t-values in parentheses.

	Matrix	MC	Chebyshev	SE toolbox	OpenGeoDa
lambda	0.619403	0.617843	0.619980	0.603000	0.619266
	(131.3)	(130.8)	(131.3)	(262.021)	(147.677)
(Intercept)	4.676450	4.669978	4.678849	4.609076	4.675881
	(59.812)	(59.715)	(59.847)	(67.513)	(59.803)
age	1.079835	1.082513	1.078846	1.108437	1.080070
	(13.140)	(13.174)	(13.127)	(18.374)	(13.143)
I(age∧2)	-2.574235	-2.580610	-2.571877	-2.641354	-2.574795
	(-18.751)	(-18.796)	(-18.735)	(-25.100)	(-18.755)
I(age∧3)	0.952080	0.954363	0.951234	0.975919	0.952280
	(13.840)	(13.868)	(13.829)	(16.927)	(13.842)
log(lotsize)	0.193845	0.194110	0.193746	0.196492	0.193868
	(40.845)	(40.921)	(40.817)	(116.472)	(40.851)
rooms	0.004376	0.004398	0.004369	0.004597	0.004378
	(1.441)	(1.447)	(1.439)	(1.598)	(1.441)
log(TLA)	0.625435	0.626045	0.625209	0.631914	0.625488
	(57.765)	(57.782)	(57.759)	(59.505)	(57.766)
beds	0.017266	0.017208	0.017288	0.016644	0.017261
	(3.839)	(3.822)	(3.845)	(3.677)	(3.837)
syear1994	0.040547	0.040547	0.040546	0.040560	0.040547
	(5.733)	(5.727)	(5.734)	(5.714)	(5.732)
syear1995	0.083232	0.083227	0.083235	0.083180	0.083232
	(11.984)	(11.972)	(11.988)	(11.941)	(11.983)
syear1996	0.103309	0.103310	0.103308	0.103323	0.103309
	(15.411)	(15.397)	(15.416)	(15.425)	(15.410)
syear1997	0.147440	0.147418	0.147448	0.147213	0.147438
	(22.071)	(22.048)	(22.080)	(22.030)	(22.069)
syear1998	0.195470	0.195460	0.195473	0.195375	0.195469
	(28.694)	(28.666)	(28.705)	(28.617)	(28.692)
Log likelihood	-9180.5	-9209.2	-9178.4	-610.68	-9181.2
Sigma squared	0.1004	0.1005	0.1004	0.1016	0.1004
seconds	3.971	3.453	2.939		

by replacing most of the matrix with analytical values, termed the analytical-numerical mixed Hessian by LeSage and Pace (2009, pp. 54–60). The awkward trace term for the interaction between λ and σ^2 — $tr(\mathbf{W}(\mathbf{I} - \tilde{\lambda}\mathbf{W})^{-1})$ — may be approximated by a series of traces of the powered weights matrix, either computed using sparse matrix or Monte Carlo techniques. Because sparse matrices become more and more dense as the power rises, it is also possible to use the technique due to Smirnov and Anselin (2009) to accumulate in vectors in an N-loop, which can be split among cores in a cluster. Since the 3107 US counties are small enough to allow us to try these approaches, we can check their equivalence.

```
> eW <- as(as_dgRMatrix_listw(elw), "CsparseMatrix")
> set.seed(100831)
> etr_MC <- trW(eW, m = 24, type = "MC")
> etr_mult <- trW(eW, m = 24, type = "mult")

> eWs <- spdep:::listw2U_Matrix(spdep:::similar.listw_Matrix(elw))
> etr_mom <- trW(eWs, m = 24, type = "moments")
> library(snow)
> cl <- makeSOCKcluster(2)
> set.ClusterOption(cl)
> etr_mom1 <- trW(eWs, m = 24, type = "moments")
> stopCluster(cl)
> set.ClusterOption(NULL)

> all.equal(etr_mom, etr_mom1, check.attributes = FALSE)

[1] TRUE

> all.equal(etr_mult, etr_mom1, check.attributes = FALSE)

[1] TRUE
```

As we see, the Smirnov and Anselin (2009) algorithm provides traces of powers that are equal within machine precision to the `mult` approach, which simply takes traces of successive powers of the sparse weights matrix. This leads to the matrix becoming dense after a small number of powers, and so is only feasible up to moderate N. Beyond this, the `MC` and `moments`[7] approaches remain. The relative timings for this N are: `MC` 0.269s, `mult` 4.719s, and `moments` without parallelization 36.55s, with parallelization on two cores 21.75s.

If we fit the spatial lag models again, using two types of traces of the powered weights matrix, we can compare the z-values for the coefficients of these with those of the exact eigenvalue-based estimates, and from the unadjusted finite difference Hessian:

[7] A `moments` method for computing the Jacobian is under consideration for ML model estimation functions in **spdep**.

```
> elag_ML_Matrix_trmult <- lagsarlm(eform, data = elect80,
+     listw = elw, method = "Matrix", tr = etr_mult)
> elag_ML_Matrix_trMC <- lagsarlm(eform, data = elect80,
+     listw = elw, method = "Matrix", tr = etr_MC)
```

Table 7: Comparison of US 1980 election turnout ML spatial lag results; z-values in parentheses.

	Eigen	Matrix	Matrix tr(mult)	Matrix tr(MC)
rho	0.5547	0.5547	0.5547	0.5547
	(34.715)	(38.09)	(37.64)	(37.63)
(Intercept)	0.7805	0.7805	0.7805	0.7805
	(25.352)	(26.514)	(26.388)	(26.386)
log(pc_income)	-0.0895	-0.0895	-0.0895	-0.0895
	(-9.698)	(-9.677)	(-9.667)	(-9.667)
log(pc_college)	0.1568	0.1568	0.1568	0.1568
	(17.952)	(18.057)	(18.002)	(18.001)
log(pc_homeownership)	0.2142	0.2142	0.2142	0.2142
	(25.680)	(26.370)	(26.367)	(26.367)
Log likelihood	3943.8	3943.8	3943.8	3943.8
Sigma squared	0.004333	0.004333	0.004333	0.004333
seconds	29.90	0.254	0.252	0.258

Table 7 shows that there is very little difference between the exact truncated series of traces of the powered weights matrix, and the Monte Carlo truncated series in the z-values output. The adjusted numerical Hessian z-values usually lie between the exact values and the unadjusted numerical Hessian values, although closer to the latter than the former. This suggests that this approach to augmenting the numerical Hessian should be used in practice when possible, and not only when difficulties are encountered in inverting the unadjusted numerical Hessian.

```
> hW <- as(as_dgRMatrix_listw(hlw), "CsparseMatrix")
> set.seed(100831)
> htr_MC <- trW(hW, m = 24, type = "MC")

> hlag_ML_Matrix_trmom <- lagsarlm(hform, data = house,
+     listw = hlw, method = "Matrix", tr = htr_mom)
> hlag_ML_Matrix_trMC <- lagsarlm(hform, data = house,
+     listw = hlw, method = "Matrix", tr = htr_MC)
```

A similar exercise may be undertaken for the larger data set, with time taken for `MC` 2.307s, and `moments` without parallelization 513.5s, with parallelization on two cores 346.9s. The timings indicate that claims in Smirnov and Anselin (2009) may have been somewhat optimistic with regard to the efficiency of the algorithm,

although careful coding in a compiled language might speed up computation. The z-value for ρ for the unadjusted numerical Hessian is 139.8, for the adjusted numerical Hessian with `moments` traces 136.0, and with `MC` traces 136.0. As can be seen, the difference between the results for the two methods for computing the truncated series of traces of the powered weights matrix is minimal; the difference between the standard errors of ρ using these techniques is -1.264e-07.

2.3 Spatial 2SLS and GMM techniques

In addition to maximum likelihood, spatial two stage least squares and generalized method of moments approaches have been proposed by Kelejian and Prucha (1998, 1999). The `stsls` and `GMerrorsar` functions were contributed to **spdep** by Luc Anselin, and have been revised in minor ways by this author and Gianfranco Piras. In particular, `stsls` now uses $[\mathbf{X}, (\mathbf{WX}), (\mathbf{WWX})]$ as instruments, and `GMerrorsar` can use a number of different functions for numerical optimization.

```
> hlag_stsls <- stsls(hform, data = house, listw = hlw)
> herr_GM <- GMerrorsar(hform, data = house, listw = hlw,
+     returnHcov = TRUE)
```

The `stsls` function have been extended to provide a heteroskedasticity and autocorrelation consistent (HAC) estimator (Kelejian and Prucha, 2007; Piras, 2010) in the **sphet** package. An additional, auxiliary spatial weights object is used to account for heteroskedasticity not otherwise accommodated in the standard specification.

```
> library(sphet)
> hk10 <- knn2nb(knearneigh(coordinates(house), k = 10))
> hdists <- nbdists(hk10, coordinates(house))
> hlwd <- nb2listw(hk10, glist = hdists, style = "B")
> class(hlwd) <- c("sphet", "distance", "nb", "GWT")
> hlag_stslshac <- stslshac(hform, data = house, listw = hlw,
+     distance = hlwd, type = "Triangular")
```

From Table 8 we can see that the coefficient values of all three GM lag estimators agree exactly. The spatial coefficient values are similar to the ML estimate. The z-values and t-values of the ordinary GM lag estimators also agree exactly, and lie between the ML z-values and the z-values of the GM HAC lag estimator; the z-values of the GM HAC lag estimator are sometimes the largest, sometimes the smallest in absolute value. In this case, ρ is similar between the ML and GM estimators, but it is worth noting that this model accounts for around 80% of the variation in the dependent variable. The difference between ML and GM ρ in the smaller data set, in which a little less than 50% of the variation in the dependent variable is accounted for, is much greater, ML lag $\rho = 0.555$, GM lag $\rho = 0.268$.

Table 8: Comparison of Lucas county (OH) house price spatial lag results; for **spdep** z-values in parentheses, for SE toolbox, t-values in parentheses.

	Matrix lag	GM lag	GM HAC lag	SE toolbox GM lag
rho	0.522814	0.527795	0.527795	0.527795
	(139.8)	(82.8774)	(60.6645)	(82.8774)
(Intercept)	0.258328	0.233157	0.233157	0.233157
	(4.037)	(3.3119)	(2.6957)	(3.3119)
age	1.308469	1.302469	1.302469	1.302469
	(23.303)	(22.9968)	(11.9207)	(22.9968)
I(age^2)	-2.321326	-2.305514	-2.305514	-2.305514
	(-22.693)	(-22.2457)	(-11.2922)	(-22.2457)
I(age^3)	0.654895	0.649860	0.649860	0.649860
	(11.876)	(11.7378)	(5.6272)	(11.7378)
log(lotsize)	0.072975	0.071987	0.071987	0.071987
	(23.874)	(22.3376)	(16.4858)	(22.3376)
rooms	-0.002534	-0.002649	-0.002649	-0.002649
	(-1.137)	(-0.8711)	(-0.7592)	(-0.8711)
log(TLA)	0.577833	0.574756	0.574756	0.574756
	(58.240)	(54.0755)	(45.8492)	(54.0755)
beds	0.015621	0.015928	0.015928	0.015928
	(4.066)	(3.5177)	(3.1816)	(3.5177)
syear1994	0.044475	0.044471	0.044471	0.044471
	(5.999)	(6.0257)	(6.8135)	(6.0257)
syear1995	0.086074	0.086065	0.086065	0.086065
	(11.923)	(11.9414)	(13.3389)	(11.9414)
syear1996	0.105937	0.105907	0.105907	0.105907
	(15.134)	(15.1728)	(15.5060)	(15.1728)
syear1997	0.147347	0.147365	0.147365	0.147365
	(21.226)	(21.2865)	(21.9719)	(21.2865)
syear1998	0.200722	0.200711	0.200711	0.200711
	(28.135)	(28.1986)	(30.3074)	(28.1986)
Sigma squared	0.09479	0.09459	0.09459	0.0945
seconds	2.042	0.233	8.928	

Table 9: Comparison of Lucas county (OH) house price spatial error results; for **spdep** z-values in parentheses, for SE toolbox, t-values in parentheses.

	Matrix error	GM error	SE toolbox GM error
lambda	0.619403	0.445980	0.445981
	(131.3)		(56.820)
(Intercept)	4.676450	4.039891	4.039891
	(59.812)	(50.318)	(48.699)
age	1.079835	1.402878	1.402878
	(13.140)	(17.396)	(16.836)
I(age∧2)	-2.574235	-3.245746	-3.245746
	(-18.751)	(-23.494)	(-22.738)
I(age∧3)	0.952080	1.166465	1.166465
	(13.840)	(16.430)	(15.901)
log(lotsize)	0.193845	0.207524	0.207524
	(40.845)	(46.117)	(44.633)
rooms	0.004376	0.006591	0.006591
	(1.441)	(1.971)	(1.908)
log(TLA)	0.625435	0.699655	0.699655
	(57.765)	(60.076)	(58.142)
beds	0.017266	0.009620	0.009620
	(3.839)	(1.942)	(1.879)
syear1994	0.040547	0.041296	0.041296
	(5.733)	(5.256)	(5.087)
syear1995	0.083232	0.083276	0.083276
	(11.984)	(10.808)	(10.460)
syear1996	0.103309	0.103967	0.103967
	(15.411)	(13.969)	(13.519)
syear1997	0.147440	0.145675	0.145675
	(22.071)	(19.662)	(19.030)
syear1998	0.195470	0.195246	0.195246
	(28.694)	(25.797)	(24.967)
Sigma squared	0.1004	0.1157	0.1235
seconds	3.971	1.216	

Table 9 shows again that the two GM error estimators agree in coefficient values. They do not, however, agree in the z-values/t-values, nor do they agree with the ML coefficients or z-values. The difference in λ is noticable, and feeds through into the estimates of the other coefficients.

2.4 Matrix exponential techniques

Matrix exponential methods may be used to fit spatial regression models (LeSage and Pace, 2007). Code for fitting the spatial lag model was contributed to **spdep** by Eric Blankmeyer, and as Table 10 indicates, the two independent implementations give close results for the coefficients and σ^2:

```
> hlag_mess <- lagmess(hform, data = house, listw = hlw)
```

Because numerical optimization is used to find the the optimum of a log likelihood function, results may differ even when the same parameter q is used, as is the case here. The standard error of α is in any case calculated using a numerical Hessian procedure; in the `lagmess` function, the remaining standard errors come from an ancilliary linear regression. For more details, see LeSage and Pace (2009, pp. 236–278).

3 Comparing associated measures

In addition to the fitting of spatial econometric models, associated measures are needed to assist in their interpretation. Here we will discuss two such measures, one permitting the impact of changes in right hand side variables in spatial lag and spatial Durbin models to be interpreted, the other to test whether the coefficients of spatial error models and linear models are significantly different from one another, expressed as a Hausman test. Our concern here is to provide analysts with the functions and methods needed to apply these recent additions to spatial econometrics, and to compare reference implementations.

3.1 Implementing impact measures

In fitting spatial lag and spatial Durbin models, it has emerged over time that, unlike the spatial error model, the spatial dependence in the parameter ρ feeds back, obliging analysts to base interpretation not on the fitted parameters β, and γ where appropriate, but rather on correctly formulated impact measures (LeSage and Pace, 2009).

This feedback comes from the fact that the elements of the variance-covariance matrix of the coefficients for the maximum likelihood spatial error model linking λ

Table 10: Comparison of Lucas county (OH) house price matrix exponential spatial lag results; for **spdep** z-values in parentheses, for SE toolbox, t-values in parentheses.

	R/spdep lagmess	SE toolbox lag MESS
alpha	-0.554305	-0.554302
	(-107.7)	(-288.0654)
(Intercept)	0.646518	0.646529
	(9.4422)	(16.7893)
age	1.463764	1.463766
	(23.8532)	(23.9120)
I(age∧2)	-2.670442	-2.670448
	(-24.0239)	(-24.0140)
I(age∧3)	0.760874	0.760876
	(12.6529)	(12.6628)
log(lotsize)	0.090868	0.090869
	(28.0265)	(28.7039)
rooms	-0.001293	-0.001293
	(-0.3894)	(-0.4674)
log(TLA)	0.651705	0.651706
	(60.3551)	(431.5221)
beds	0.012442	0.012442
	(2.5221)	(2.5370)
syear1994	0.047726	0.047726
	(5.9135)	(5.9147)
syear1995	0.091701	0.091701
	(11.6347)	(11.6359)
syear1996	0.113129	0.113129
	(14.8210)	(14.8234)
syear1997	0.156383	0.156383
	(20.6566)	(20.6580)
syear1998	0.212281	0.212281
	(27.2724)	(27.2715)
Log likelihood	-8343	-100933
Sigma squared	0.1131	0.1131
seconds	5.158	

and β are zero, $\partial^2\ell/(\partial\beta\partial\rho) = \mathbf{0}$, while in the spatial lag model (and by extension in the spatial Durbin model): $\partial^2\ell/(\partial\beta\partial\rho) \neq \mathbf{0}$. In the spatial error model, for right hand side variable r, $\partial y_i/\partial x_{ir} = \beta_r$ and $\partial y_i/\partial x_{jr} = 0$ for $i \neq j$; in the spatial lag model, $\partial y_i/\partial x_{jr} = ((\mathbf{I} - \rho\mathbf{W})^{-1}\mathbf{I}\beta_r)_{ij}$, where $\mathbf{I}$ is the $N \times N$ identity matrix, and $(\mathbf{I} - \rho\mathbf{W})^{-1}$ is known to be dense (LeSage and Pace, 2009, p. 33–42).

The variance-covariance matrix of the coefficients and the series of traces of the powered weights matrix are the key ingredients needed to compute impact measures for spatial lag and spatial Durbin models; both of these are based on the representation of weights matrices as sparse matrices. We can also compute the measures analytically for smaller data sets; here we will contrast the 1980 US election and Lucas (OH) data sets, where the former is small enough to permit all the output values to be compared.

An estimate of the coefficient variance-covariance matrix is needed for Monte Carlo simulation of the impact measures, although the measures themselves may be computed without an estimate of this matrix. LeSage and Pace (2009, pp. 33–42, 114–115) and LeSage and Fischer (2008) provide the background and implementation details for impact measures.

The awkward $S_r(\mathbf{W}) = ((\mathbf{I} - \rho\mathbf{W})^{-1}\mathbf{I}\beta_r)$ matrix term needed to calculate impact measures for the lag model, and $S_r(\mathbf{W}) = ((\mathbf{I} - \rho\mathbf{W})^{-1}(\mathbf{I}\beta_r - \mathbf{W}\gamma_r))$ for the spatial Durbin model, may be approximated using traces of powers of the spatial weights matrix as well as analytically. The average direct impacts are represented by the sum of the diagonal elements of the matrix divided by N for each exogenous variable, the average total impacts are the sum of all matrix elements divided by N for each exogenous variable, while the average indirect impacts are the differences between these two impact vectors.

We have seen above in Section 2.2 how to compute the required truncated series of traces of powered spatial weights matrices. In **spdep**, `impacts` methods are available for ML spatial lag and spatial Durbin fitted model objects, and for GM spatial lag objects, since variance-covariance matrices can be calculated using techniques already discussed. The methods can use either dense matrices or truncated series of traces, so the impacts for a single model fit may be examined using dense or sparse procedures, and using different ways of computing the traces:

```
> set.seed(100831)
> eimp_lag_ML_eigen_lw <- impacts(elag_ML_eigen, listw = elw,
+     R = 1999)

> set.seed(100831)
> eimp_lag_ML_Matrix_trmult2 <- impacts(elag_ML_Matrix_trmult,
+     tr = etr_mult, R = 1999)
> set.seed(100831)
> eimp_lag_ML_Matrix_trMC2 <- impacts(elag_ML_Matrix_trMC,
+     tr = etr_MC, R = 1999)
```

We are interested in establishing whether the extra time needed to compute dense matrix exact coefficient and variance-covariance matrix estimates affect the Monte Carlo test results. Computing the dense matrix Monte Carlo measures of impacts dispersion is very time-consuming; for $R = 1999$ draws, it took here 84193s, that is almost 24 hours. The timings for the updating sparse Cholesky were with exact power trace series used both to adjust the numerical Hessian and for the Monte Carlo calculations (from powering a sparse matrix): 0.728s, and 0.749s with the Monte Carlo power trace series used in both steps. For comparison we add the results taken from the estimation of impact measures using the Spatial Econometrics toolbox function.

```
> elag_stsls <- stsls(eform, data = elect80, listw = elw)
> set.seed(100831)
> eimp_lag_stsls_trMC <- impacts(elag_stsls, tr = etr_MC,
+     R = 1999)
```

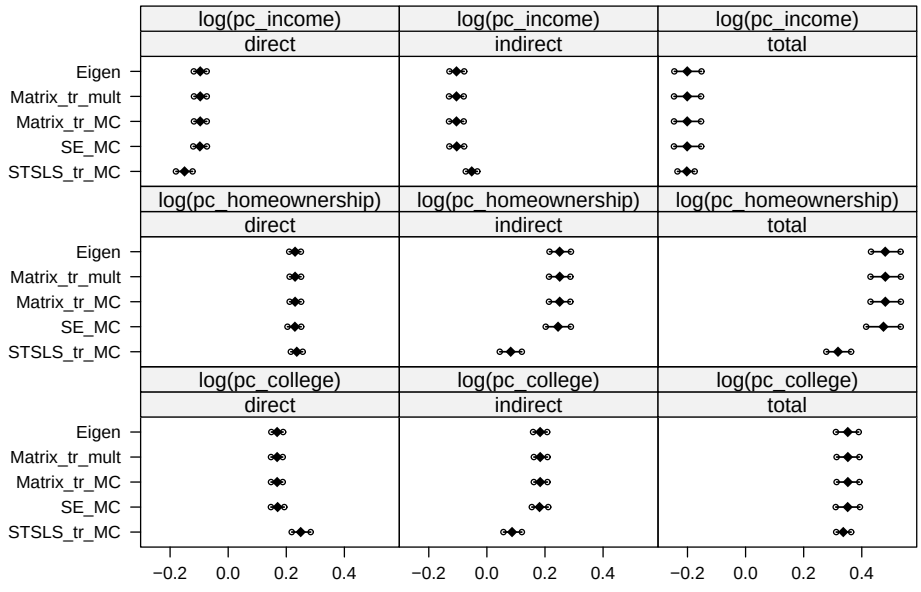

Figure 1: Comparison of the dispersion of impact measures for the three right hand side variables in the US election data set, impacts marked by diamond; 98% highest posterior density range for Monte Carlo impact measure simulations.

In addition we include impacts calculated for the GM lag `stsls` fit, which in the smaller data set estimates ρ as 0.2679, rather than 0.5547 in the ML case —

naturally this difference feeds through into the estimates of β. The GM lag variance-covariance matrix estimates are also different, which affects the Monte Carlo draws, because the multivariate Normal draws are taken using the fitted coefficients and their variance-covariance matrix.

Figure 1 summarizes the results of the impact measures and their dispersion for the three right hand side variables in the US election data set. The impact measures are marked by diamonds, and the dispersion is indicated by horizontal lines spanning the 1%–99% highest posterior density range for each compared type of impact and variable. The methods used are: Eigen — spdep dense matrix for model fit and simulation; Matrix_tr_mult — spdep updating sparse Cholesky with exact power trace series for Hessian adjustment and simulation; Matrix_tr_MC — spdep updating sparse Cholesky with Monte Carlo power trace series for Hessian adjustment and simulation; STSLS_tr_MC — spdep `stsls` with Monte Carlo power trace series for simulation; and SE_MC — Spatial Econometrics toolbox `sar` function results. The SE_MC method used $R = 1000$ draws, the remaining methods $R = 1999$ draws.

The four sets of results for the maximum likelihood estimators, three in **spdep** and one in the Spatial Econometrics toolbox, are very close to each other, both in point estimate and dispersion of impact measures. We have already seen that there are small differences between the Spatial Econometrics toolbox, and the **spdep** model fitting function in the line search for ρ and in the computation of the numerical Hessian, leading to the slight differences seen here. The only large differences are between the GM lag estimator and the ML estimators. We can safely conclude that ML impact measures are not impaired by using Monte Carlo approximations to the power trace series, so that an effective choice is to fit using a maximum likelihood estimator and a Monte Carlo approximation to the power trace series to adjust the numerical Hessian-based variance-covariance matrix, and to evaluate the impact measures using the same Monte Carlo approximation to the power trace series.

Let us turn now to the spatial Durbin model of the Lucas county housing data, for which impact measures are required for satisfactory interpretation:

```
> hSD_ML_Matrix_trMC <- lagsarlm(hform, data = house, listw = hlw,
+     type = "mixed", method = "Matrix", tr = htr_MC)
> set.seed(100831)
> himp_SD_ML_Matrix_trMC2 <- impacts(hSD_ML_Matrix_trMC,
+     tr = htr_MC, R = 1999)
```

Once the Monte Carlo approximation to the power trace series has been computed, here taking 2.307s, fitting the model for over 25,000 observations and 12 right hand side variables (6.056s), the impact measures can be output at little extra cost, taking just 1.315s. The main challenge is to present the voluminous output for the direct, indirect and total impacts and their measures of dispersion, as we see

Table 11: Comparison of Lucas county (OH) house price spatial Durbin impacts; for **spdep** z-values in parentheses, for SE toolbox, t-values in parentheses.

	Matrix MC direct	SE direct	Matrix MC indirect	SE indirect	Matrix MC total	SE total
age	1.116635	1.116326	0.903016	0.901730	2.019652	2.018056
	(13.9621)	(13.7515)	(6.82024)	(7.01399)	(13.6664)	(13.8478)
I(age^2)	-2.090591	-2.091193	-1.391627	-1.388742	-3.482218	-3.479935
	(-15.4446)	(-15.4222)	(-5.74951)	(-5.95153)	(-12.3207)	(-12.6045)
I(age^3)	0.545785	0.545164	-0.057494	-0.057812	0.488291	0.487352
	(7.8822)	(7.9677)	(-0.44421)	(-0.45240)	(3.0459)	(3.1315)
log(lotsize)	0.111786	0.111652	0.015387	0.015401	0.127173	0.127053
	(23.2954)	(23.1832)	(2.14509)	(2.19660)	(17.3521)	(18.0693)
rooms	0.001723	0.001760	0.002335	0.002142	0.004058	0.003902
	(0.4794)	(0.5017)	(0.25726)	(0.24617)	(0.3570)	(0.3496)
log(TLA)	0.695545	0.696486	0.486106	0.485408	1.181651	1.181894
	(60.6295)	(62.1220)	(19.43009)	(19.24185)	(37.6649)	(37.1994)
beds	0.002605	0.002526	-0.067847	-0.067422	-0.065242	-0.064896
	(0.5002)	(0.4699)	(-5.26156)	(-5.21854)	(-3.9857)	(-3.8895)
syear1994	0.039575	0.039852	-0.004850	-0.004455	0.034726	0.035397
	(4.3891)	(4.6116)	(-0.20853)	(-0.20750)	(1.2101)	(1.2744)
syear1995	0.084237	0.084400	0.001933	0.001660	0.086170	0.086060
	(9.7066)	(9.8766)	(0.05934)	(0.08153)	(3.0738)	(3.2571)
syear1996	0.103682	0.103901	-0.002139	-0.002002	0.101543	0.101899
	(12.4602)	(12.8060)	(-0.13309)	(-0.10082)	(3.8154)	(3.9965)
syear1997	0.143098	0.143091	-0.025365	-0.025592	0.117733	0.117499
	(17.1701)	(18.0463)	(-1.25323)	(-1.28779)	(4.4897)	(4.5974)
syear1998	0.203306	0.203529	0.020313	0.020286	0.223619	0.223814
	(23.9975)	(24.3157)	(0.94477)	(0.99041)	(8.1506)	(8.4462)

from Table 11. Here dispersion is expressed as a z-value based on the standard deviation of the simulations in both applications (termed t-values in the Spatial Econometrics toolbox function). This is an unfortunate abbreviation of the more appropriate quantile measure, but is perhaps unavoidable when there are many variables. As can be seen, the two applications give very similar results for the impact measures as calculated using the Lucas county data set and the spatial Durbin representation.

3.2 Implementing a Hausman test

Pace and LeSage (2008) introduce a spatial Hausman test intended to check whether the regression coefficients of a spatial error model differ significantly from those of the underlying linear model assuming $\lambda = 0$. If they are not seen as the same, the model is misspecified (see also LeSage and Pace, 2009, pp. 61–63). The spatial Hausman test is constructed as:

$$T = (\beta_o - \beta_s)'(\Omega_o - \Omega_s)^{-1}(\beta_o - \beta_s),$$

where β_o are the linear model coefficients, β_s are the spatial error model coefficients, and $\Omega_s = \tilde{\sigma}^2(\mathbf{X}(\mathbf{I} - \tilde{\lambda}\mathbf{W})'(\mathbf{I} - \tilde{\lambda}\mathbf{W})\mathbf{X})^{-1}$ is the estimated variance-covariance matrix of the spatial error model coefficients. The Ω_o term is more complicated, being not the estimated variance-covariance matrix of the linear model coefficients, but a variance-covariance matrix adjusted to suit the assumed null of the spatial error process, using the estimated value of λ:

$$\Omega_o = \tilde{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{I} - \tilde{\lambda}\mathbf{W})^{-1}(\mathbf{I} - \tilde{\lambda}\mathbf{W}')^{-1}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

If we write $\mathbf{A} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$, we can represent half of Ω_o as:

$$(\mathbf{I} - \tilde{\lambda}\mathbf{W}')^{-1}\mathbf{A} = \sum_{j=0}^{\infty}(\tilde{\lambda}^j\mathbf{W}'^j)\mathbf{A} = \mathbf{A} + (\tilde{\lambda}\mathbf{W}')\mathbf{A} + \ldots$$

Since $\mathbf{A}$ is an $N \times k$ matrix, with $k \ll N$, we can approximate each half of Ω_o by the sum of a truncated power series, not requiring the inversion of $N \times N$ matrix $(\mathbf{I} - \tilde{\lambda}\mathbf{W}')$. We truncate the series at the point at which the mean of the next additional term does not exceed a very small tolerance value. This is implemented in the `powerWeights` function, and is used when spatial error models are fitted using sparse matrix techniques. By default the `errorsarlm` function returns a component with the matrix part of Ω_o, which is used in the test. This also means that the spatial Hausman test may be performed on large data sets, such as the Lucas, Ohio house price data set with $N = 25357$.

As the Hausman test is not yet available in the Spatial Econometrics toolbox, we compare here using a script kindly provided by James LeSage. Because the script uses dense matrix techniques, we compare using the smaller data set:

```
> Hausman.test(eerr_ML_Matrix)

        Spatial Hausman test (approximate)

data:  pc_turnout ~ log(pc_income) + log(pc_college) +
        log(pc_homeownership)
Hausman test = 146.3798, df = 4, p-value < 2.2e-16
```

When running the provided script with values of the fitted coefficients and σ^2 from the function output from the Spatial Econometrics toolbox functions, the test statistic is 150.22 (see also Table 4 in Section 2.1 above); if the script is given the fitted coefficients and σ^2 from `errorsarlm`, the result is 146.38, that is identical with the implementation of the Hausman test in **spdep**. It thus appears that the implementation based on the sum of a truncated power series performs adequately, and that we should be able to depend on test results for other data sets. Applying the Hausman test to the Lucas county spatial error model, we find:

```
> Hausman.test(herr_ML_Matrix)

        Spatial Hausman test (approximate)

data:  log(price) ~ age + I(age^2) + I(age^3) + log(lotsize) + rooms
        +
    log(TLA) + beds + syear
Hausman test = 3115.981, df = 13, p-value < 2.2e-16
```

There is no doubt that the estimated coefficients of the right hand side variables of the linear and spatial error models differ. If we continue to explore the relative fit of the spatial models using a likelihood ratio test, or by comparing AIC values, we see that the spatial Durbin model differs clearly from the spatial error model, and fits the house price data better:

```
> LR.sarlm(hSD_ML_Matrix_trMC, herr_ML_Matrix)

        Likelihood ratio for spatial linear models

data:
Likelihood ratio = 3745.901, df = 12, p-value < 2.2e-16
sample estimates:
Log likelihood of hSD_ML_Matrix_trMC
                          -7307.507
    Log likelihood of herr_ML_Matrix
                          -9180.458

> AIC(herr_ML_Matrix)

[1] 18390.92

> AIC(hSD_ML_Matrix_trMC)
```

```
[1] 14669.01
```

In conclusion, we see that the Hausman test may also be applied to the estimated GM spatial error model:

```
> Hausman.test(herr_GM)

        Spatial Hausman test (approximate)

data:  log(price) ~ age + I(age^2) + I(age^3) + log(lotsize) + rooms
        +
    log(TLA) + beds + syear
Hausman test = 3974.049, df = 13, p-value < 2.2e-16
```

4 Extensions

Fortunately, comparing functions in the R **spdep** package with functions in the Spatial Econometrics toolbox is eased by the fact that the code is open source, and so open to scrutiny. When OpenGeoDa achieves the same status, it will become more obvious where its strengths lie, and it will be possible for others to contribute implementations of additional functionality. Since there is as yet no option to fit spatial Durbin models directly, or to calculate impact measures, comparisons of these techniques have been restricted to the Spatial Econometrics toolbox and **spdep**.

The publication of the **sphet** package and the accompanying article by Piras (2010) signals an interesting extension to a new range of specifications. In addition, the **splm** package for spatial panel models is under active development by Gianfranco Piras and Giovanni Millo on R-forge, and may already be downloaded for use.[8] Collaborative development using platforms of this kind is very beneficial, for a description see Theussl et al. (2010). Within **spdep** itself, provision is being made through modularization to permit users to choose between different ways of calculating the Jacobian (Bivand, 2010). It is also intended to provide a function to fit a general spatial regression model using different fitting techniques, which is needed to contrast with possibly more appropriate modelling strategies, such as the spatial Durbin model.

What remains is to encourage researchers who use these and other software applications to take active part in discussion lists, where more experienced users can offer advice to those starting to discover the attractions of using spatial econometrics tools to tackle empirical economic questions. Once more real-world examples of the application of, for instance, impact measures, have been published, the usefulness of such advances will become more evident. Having multiple implementation in different application languages provides users with more choice, and, as we have

[8] `https://r-forge.r-project.org/projects/splm/`, R Packages menu

seen, constitutes a “reality check” that gives insight into the ways that formulae can be rendered into code.

References

Anselin, L. (1988). *Spatial Econometrics: Methods and Models*. Kluwer, Dordrecht.

Anselin, L., Syabri, I., and Kho, Y. (2006). GeoDa: An introduction to spatial data analysis. *Geographical Analysis*, 38(1):5–22.

Barry, R. and Pace, R. (1999). Monte Carlo estimates of the log determinant of large sparse matrices. *Linear Algebra and its Applications*, 289(1-3):41–54.

Bivand, R. S. (1984). Regression modeling with spatial dependence: an application of some class selection and estimation methods. *Geographical Analysis*, 16:25–37.

Bivand, R. S. (2002). Spatial econometrics functions in R: Classes and methods. *Journal of Geographical Systems*, 4:405–421.

Bivand, R. S. (2006). Implementing spatial data analysis software tools in R. *Geographical Analysis*, 38(1):23–40.

Bivand, R. S. (2010). Computing the Jacobian in spatial models: an applied survey. Discussion paper 2010:20, Department of Economics, Norwegian School of Economics and Business Administration.

Bivand, R. S. and Gebhardt, A. (2000). Implementing functions for spatial statistical analysis using the R language. *Journal of Geographical Systems*, 2:307–317.

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

Bivand, R. S. and Portnov, B. A. (2004). Exploring spatial data analysis techniques using R: the case of observations with no neighbours. In Anselin, L., Florax, R. J. G. M., and Rey, S. J., editors, *Advances in Spatial Econometrics: Methodology, Tools, Applications*, pages 121–142. Springer, Berlin.

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

Griffith, D. A. and Layne, L. J. (1999). *A casebook for spatial statistical data analysis*. Oxford University Press, New York.

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

Kelejian, H. H. and Prucha, I. R. (1999). A generalized moments estimator for the autoregressive parameter in a spatial model. *International Economic Review*, 40:509–533.

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

LeSage, J. and Fischer, M. (2008). Spatial growth regression: Model specification, estimation and interpretation. *Spatial Economic Analysis*, 3:275–304.

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

LeSage, J. P. and Pace, R. K. (2007). A matrix exponential spatial specification. *Journal of Econometrics*, 140(1):190–214.

Ord, J. (1975). Estimation methods for models of spatial interaction. *Journal of the American Statistical Association*, 70(349):120–126.

Pace, R. and LeSage, J. (2004). Chebyshev approximation of log-determinants of spatial weight matrices. *Computational Statistics & Data Analysis*, 45(2):179–196.

Pace, R. and LeSage, J. (2008). A spatial hausman test. *Economics Letters*, 101:282–284.

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

Pisati, M. (2001). sg162. Tools for spatial data analysis. *Stata Technical Bulletin*, 60:21–36.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rey, S. (2009). Show me the code: spatial analysis and open source. *Journal of Geographical Systems*, 11:191–207.

Ripley, B. D. (1981). *Spatial Statistics*. Wiley, New York.

Smirnov, O. (2005). Computation of the information matrix for models with spatial interaction on a lattice. *Journal of Computational and Graphical Statistics*, 14(4):910–927.

Smirnov, O. and Anselin, L. (2009). An O(N) parallel method of computing the Log-Jacobian of the variable transformation for models with spatial interaction on a lattice. *Computational Statistics & Data Analysis*, 53(8):2980 – 2988.

Theussl, S., Ligges, U., and Hornik, K. (2010). Prospects and challenges in R package development. *Computational Statistics*.

NHH

Norges Handelshøyskole

Norwegian School of Economics
and Business Administration

NHH
Helleveien 30
NO-5045 Bergen
Norway

Tlf/Tel: +47 55 95 90 00
Faks/Fax: +47 55 95 91 00
nhh.postmottak@nhh.no
www.nhh.no