Journal of Geographical Systems manuscript No.
(will be inserted by the editor)

Implementing functions for spatial statistical analysis using the R language

Roger Bivand[1], **Albrecht Gebhardt**[2]

[1] Department of Geography, Norwegian School of Economics and Business Administration, Breiviksveien 40, N-5045 Bergen, Norway
[2] Department of Applied Statistics, Institute of Mathematics and Statistics, University of Klagenfurt, Villacher Str. 161, A-9020 Klagenfurt, Austria

Received: date / Revised version: date

Abstract R is a language similar to S for statistical data analysis, based on modern programming concepts and released under the GNU General Public License. It permits the integration of program scripts with compiled dynamically loaded libraries of functions when computing speed is important. Following a broad outline of existing collections of functions for spatial statistics written for S , we show how they may be ported to R , and compare their characteristics. We further demonstrate how existing work may be extended to topics not yet covered, and how libraries of functions may be constructed.

Functions for three types of spatial statistics are covered: spatially continuous data, point pattern data, and area data. We present packaged R functions for spatial statistical analysis, and their application to standard data sets. Both the development of R , and of these functions, is on-going, but have now reached a critical mass making R an attractive platform for teaching and applying spatial statistics.

1 Introduction

While spatial statistics ought arguably to be at least as frequently taught and used as time series methods, it has been the case over many years that lack of access to such methods in statistics or GIS software has hindered diffusion. Where software has become available, as indeed with GIS, cost per seat has been a further consideration at least in the academic community. Observing the research practices of statisticians, it has been possible over recent years to see a tendency for new methods to be published both in written form, and as collections of scripts written in the S language, and subsequently archived at Statlib (`http://lib.stat.cmu.edu/S/`). With the advent of R , a statistical programming language similar to S made available under the GNU General Public License version 2 (main archive: `http://www.ci.tuwien.ac.at/R/`), many packages have been ported from S

to R , including some for spatial statistics. Since R can be installed without difficulty on all Unix systems, and also exists in binary form for MS Windows9* and NT 4, writing or porting spatial statistics software for a wider range of potential users seems to be becoming easier.

Our concern here is to review packages already ported, to describe work in progress where no S scripts could be identified as appropriate, and to point to areas requiring further effort[1] . We begin by setting the scene with regard to spatial statistics, going on to introduce the R language.

Since observations of spatial data are unlikely to be independent, it is perhaps surprising that not more use has been made of this source of information. With an adequate choice of explanatory variables, this spatial dependence may be readily drawn into a model, and cease to be a nuisance. However, spatial dependence is not necessarily just a nuisance, but may help us to capture important facets of the realities of spatial processes. The literature on spatial statistics is substantial (see Cliff and Ord, 1973, 1981, Ripley, 1981, Upton and Fingleton, 1985, Griffith, 1988, Anselin, 1988, Haining, 1990, and more recently Cressie, 1993, and Bailey and Gatrell, 1995, among many others). We will here give a brief introduction to some of the key issues. Three recent surveys, including available software, are Levine (1996), Gatrell and Bailey (1996), and Bivand (1998).

After having set the context, we will review the component areas of spatial statistics, dealing in turn with point pattern analysis, geostatistics, and lattice (area) data analysis, showing what has been done in R , and giving examples largely taken from the literature. Where possible, results from R scripts have been checked against results from using other software. First, however, we will present the background of the S and R languages, and the opportunities and challenges offered by open-source software[2] .

2 S , R and open-source software

The history of S is relatively long, and as with so many other innovations in software, stems from researchers at Bell Laboratories; for a detailed account see Becker (1994). The two major sources on the language are Becker, Chambers, and Wilks (1988) and Chambers and Hastie (1992); Venables and Ripley provide a very useful introduction to applied statistics using S (1997). S presents the data analyst with a rich toolbox of components, permitting both the routine processing of statistical tasks, and the programming of new functions not initially included in the language. It employs vectors as basic building blocks, both permitting the convenient use of linear algebra operations, and the application of standard or user-defined functions to data. It also incorporates data structures, and an object-based task dispatch approach based on methods and classes. Many of these elements have been added with time, and do change. S is now only available commercially as S-PLUS [3] , a fact which has concerned US users, who face a Federal requirement that they ought not to develop software in a language not available from multiple independent sources. S-PLUS is an excellent system, but is not always ideal for teaching purposes because of its per seat cost, even where student and academic

pricing may be available. In 1996, MathSoft introduced a spatial statistics module for S-PLUS (Kaluzny et al., 1996), and links to GIS software are also available.

The S-PLUS spatial statistics module includes a fairly wide range of techniques for spatial data analysis, covering many of the key methods presented in Haining (1991) and Cressie (1993); Ripley acted as a consultant on the design and development of the module. Since Venables and Ripley had included software for some geostatistical and point pattern analyses in their book (1997; first edition 1994), it is not surprising that some of their work is reflected in the module. In addition, two techniques parallelling those used by Pace and Barry (1997) are included: using quadtree data structures for finding spatial neighbours, and using sparse matrix methods for fitting spatial linear regression models. The module does not cover a number of frequently used techniques in point pattern and irregular lattice data analysis, and it is indeed curious that Anselin's work is not cited in Kaluzny et al. (1996).

The background to R is interesting, in that it is based on two computer programming traditions: Bell Laboratories through S , and MIT through Scheme. As Itaka (1998) relates, meeting the first edition of the classic *Structure and Interpretation of Computer Programs* (1985, second edition: Abelson, Sussman and Sussman, 1996) opened up "a wonderful view of programming". He met Scheme at the same time as an early version of S , and quickly realized that some features of modern programming language design, such as lexical scoping, can lead to superior practical solutions. These advantages have, among others, brought Tierney, the author of Lisp-Stat (1990), into the R core development team. Differences in the underlying computing models between S and R are many and important, and sometimes concern the active user. They are outlined both in R system documentation, the R-FAQ available at the archive site, and in the R complement[4] to Venables and Ripley (1997). Venables and Ripley are also actively porting their own work to R , partly as a service to their readers, and partly because the differences in the underlying computing models tease out potential coding infelicities. Many of these issues are actively debated on R discussion lists — for details, see the R archives. Because of the closeness between S-PLUS and R , the code described below has either been ported to R from S-PLUS with no or minor modifications, while code written in the R environment can be moved back to S-PLUS with the same facility.

The key differences lie in scoping and memory management. Scoping is concerned with the environments in which symbol/value pairs are searched during the evaluation of a function, and poses a problem for porting from S to R , when use has been made of the S model, rather than that derived from Scheme (Itaka and Gentleman, 1996, p. 301–310). Memory management differs in that S generates large numbers of data files in the working directory during processing, allocating and reallocating memory dynamically. R , being based on Scheme, starts by occupying a user-specified amount of memory, but subsequently works within this, not committing any intermediate results to disk. Within the memory heap, a simple but efficient garbage collector makes the best possible use of the memory at the program's disposal. This may hinder the analysis of very large data sets, since all data have to be held in memory, but in practice this problem has been alleviated by falling memory prices. One reason cited by some authors for also using R , is

that access to the same compilers as those used by MathSoft to permit the dynamic loading of compiled C and Fortran object code is not a problem with R : you just need the same (open-source) compiler that you used to install R in the first place. This has affected users of S-PLUS on MS Windows platforms in particular.

Itaka (1998) places the future of R clearly within open-source software. Indeed, the rapid development of R as a computing environment for data analysis and graphics bears out many of the points made by Raymond (1997) in his perceptive paper on the dynamics of user/developer interaction. Over several years, it has begun to be clear that, even when well-regarded commercial software products are available, communities of users and developers are often able to gain a momentum based on very rapid debugging by large numbers of interested participants — bazaar-style development. Indeed, commercial organizations can benefit from the activity ensuing from this kind of brainstorming: over half the web sites on the Internet use open-source servers, and most of the rest reply on Perl, an open-source language, to deliver active content. Open source is not limited to Unix or Unix-based operating systems, since open-source compilers and associated tools have been ported to proprietary desk-top systems like MS Windows95, MS NT 4, and others. These in turn have permitted software, like R , to be ported to these platforms, with little or no version slack.

3 Point pattern analysis

Point pattern analysis is concerned with the location of events, and with answering questions about the distribution of those locations, specifically whether they are clustered, randomly or regularly distributed. Point pattern analysis is very sensitive to the definition of the study area, since a regularly distributed pattern can be made to seem clustered by including large margins within the study area. Measures are also subject to boundary corrections, and most often study area boundaries have to be defined as convex polygons over the study area, or in the simplest form as rectangles bounding the points under analysis. It is of course always important to plot the events to detect outliers visually, together with the boundaries being applied (Bailey and Gatrell, 1995, Cressie, 1993).

The immediate place to begin in point pattern analysis is the spatial package accompanying Venables and Ripley (1997), ported to R , and officially released at the R archive site. Loading the package, we find that the functions available cover the input of point process objects, setting the rectangular window used for edge-correction, and calculation of Ripley's K function and simulated envelopes about it under various alternative point processes.

Running the example given in Venables and Ripley (1997, p. 482–3, Figure 16.9) for one plot of $L(t)$ showing the envelope of 100 binomial simulations, we get the result shown in Figure 1; distance units are in metres, and that the estimated function differs significantly from straightness. This package is limited both in the use of only rectangular regions, and in only providing the K function. It may be supplemented by the Splancs package by Rowlingson and Diggle (1993)[5] , which required very few modifications under porting to R . It consists of a number of

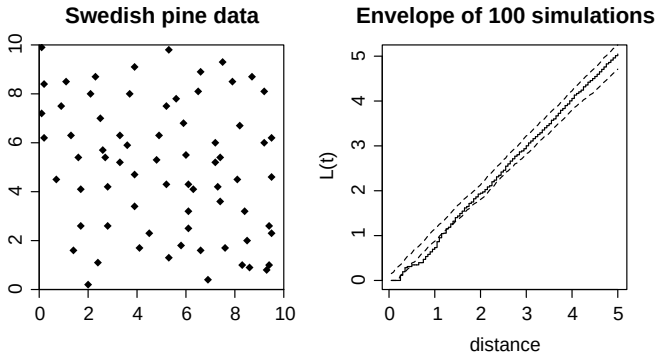

Fig. 1 Swedish pine data set example using the `spatial` package.

functions, differing from the `spatial` package, for the input of point patterns, kernel density estimation, and spatial point pattern analysis including case-control studies.

In order to try out Splancs, the example used above was re-run using `sbox()` to set the rectangular study region; had `bbox()` been used, it would have emulated the `spatial` package more exactly. As Bailey and Gatrell (1995) point out, in point pattern analysis much depends on the interpretation of graphical output produced under varying assumptions about the bounding region and the spatial scale and nature of the process. Consequently, the ease with which R (and of course S) permit results to be plotted is a major advantage.

Turning to Bailey and Gatrell (1995, INFOMAP data sets) as a source of examples for the validation of selected Splancs functions under R , we replicate Bailey and Gatrell's Figures 4.1 and 4.7 (1995, pages 122 and 130) for the locations of 'thefts from property' offences in Oklahoma City, including information on the ethnic background of the offenders. We have here used a region bounded by `sbox()` of the joint data set in the absence of indications that another boundary was used in the original. Using random labelling, we are testing whether the 'black' crimes are just a random subset of the overall pattern of all crimes. As the plot of the K_{12} function `k12hat()` against the random labelling envelope `Kenv.label()` shows, the visual impression that offences committed by 'blacks' are more spatially clustered is confirmed.

In addition, Splancs includes functions for raised incidence and space-time clustering. It would be of advantage if both Splancs and the `spatial` package defined point objects in the same way, or at least if conversion functions were written, permitting data to be moved between them without uncertainty about the bounding region being used. Both initially date from a period when object-oriented mechanisms were only beginning to enter S , and as Venables and Ripley comment, the `S-PLUS` spatial statistics module provides more comprehensive and polished facilities than their package, including $\hat{F}$, $\hat{G}$, K and L functions, envelopes about K and L functions, and kernel density estimation, using convex bounding polygons. It will be recalled that Splancs functions permit concave bounding polygons, making its approach the most general.

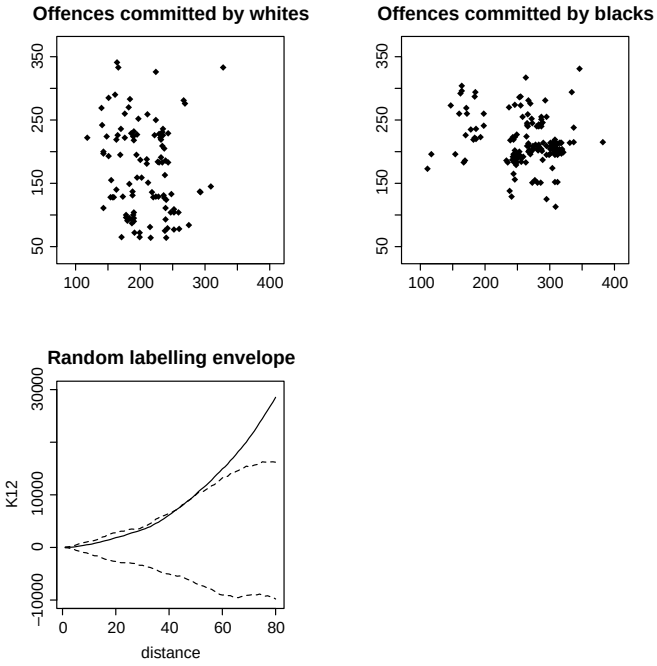

Fig. 2 Oklahoma City crime data set presented and analysed using Splancs functions.

4 Geostatistics

Today there are some geostatistical packages for `S-PLUS` , either available for free from the `S` and `S-PLUS` archive at StatLib (`http://lib.stat.cmu.edu/S/`) like `spatial` or `funfits`and also commercial packages like S+SpatialStats. Now, with the availability of `R` , the question arises of which of these packages can be used with this free free statistical environment. Of course it is not possible to use commercial `S-PLUS` modules like S+SpatialStats with `R` . But this is more than compensated by the existing free geostatistical packages, which all can be or are being ported to `R` . Moreover, the open source concept of `R` gives better chances to modify these packages to meet particular needs.

But even with the core functionality of `R` , some spatial analysis can be done. Linear models can be used for trend removal, although care has to be shown in predicting from fitted trend surfaces, and local smoothing techniques (libraries `locfit` and `modreg`) can be used to generate locally weighted regression surfaces or to estimate densities. An important geostatistical package comes with Venables and Ripley (1997). Chapter 15 of their book describes the use of this package in detail. The first port of the `spatial` package was made available in late 1997. Meanwhile, they took notice of the development of `R` , and now maintain the package for `R` along with the `S-PLUS` version. The advantage of this implementation of kriging prediction is that critical computations are carried out in dynamically loaded C functions.

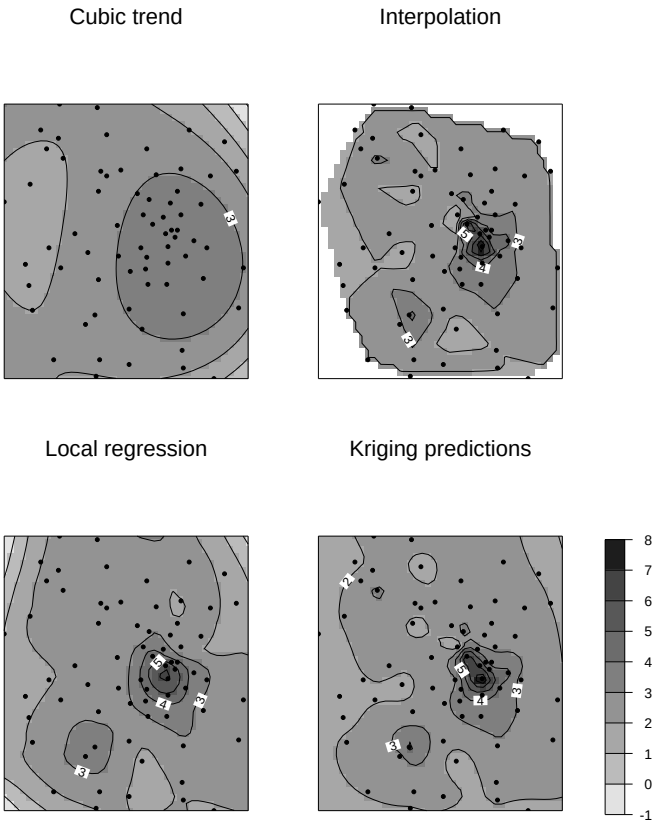

Fig. 3 Four representations of the surface of log PCB scores, Pontypool data set.

Figure 3 shows the results of using four surface modelling functions. The data modelled are taken from Bailey and Gatrell (1995), and report on the scores on a number of PCB indices for 70 locations around a possible emission site near Pontypool; logarithms of the PCB scores are used here. Firstly a cubic trend surface is fitted using the `surf.ls()` function from the `spatial` package, the predicted values being obtained using the `trmat()` function on the trend surface model object. Next, the `interp()` function from the `akima` package, an interpolating algorithm based on FORTRAN code by H. Akima (1996), containing code for linear and bicubic spline interpolation, is used. Thirdly, the `loess()` local regression function from `modreg` package, using `predict.loess()` to generate the predictions. Finally, after estimation of the empirical variogram from the cubic trend surface model, and the fitting of a theoretical model by eye, kriging predictions were made using `surf.gls()` and `prmat()` from the `spatial` package. All of the methods used show the clustering of higher values around the possible emission site, with the global trend surface model being least satisfactory. Both kriging and local regression can also provide standard errors of prediction.

The spatial package does not provide functions for fitting theoretical variograms; these may be found in the sgeostat package (ported to R from S+GeoStat, Majure, 1995), which contains many variogram exploration tools, for instance the spacebox() function for examining a variogram cloud using boxplots. There are even implementations of robust variogram estimation as described in Cressie (1993). The sgeostat package is written throughout in interpreted code, making it more transparent at the cost of execution speed, especially for kriging prediction.

5 Lattice data analysis and spatial econometrics

In this section, we will use the term lattice data for data associated with an arbitrary division of the area being studied into an irregular lattice; these are the kinds of data most often associated with spatial econometrics techniques. Where possible, results have been cross-checked with other software.

In the S-PLUS spatial statistics module, neighbourhood relationships are represented by a data frame, in which the first two columns identify the pairs of zones that are neighbours, and the weight assigned to the relationship. A fourth column is used to record which type of relationship is referred to by column 3: where several different definitions are used, columns three and four are repeated, permitting subsetting for example to those neighbours with centroids less than 50 km apart which were not contiguous in the sense of belonging to zones with shared boundaries. This format provides for convenient construction of distance-based matrices, but no automatic tally of the numbers and identifications of neighbours of a given zone or site.

Work below has followed earlier structured formats, returning an R list object containing a vector card with the numbers of neighbours of each zone, and two vectors of pointers controlling access to the neighbour table map such that the sequence ip1[i]:ip2[i] contains card[i] elements. At present, weights are either standardized or non-standardized, but general forms may readily be implemented by adding a vector of length(map), containing the weight values. All the list components are integer vectors, taking 3*length(card) + sum(card) positions of integer memory, a matter of less importance now than formerly. Unfortunately for both formats, access to vector and matrix elements is costly in R and S , so that the preparation and testing of weights matrices using interpreted code only is not fast, and could be speeded up by using dynamically loaded C or Fortran functions or subroutines.

Using the dist() function in the mva package, it is easy to construct naive distance banding and incremental weights matrices for moderate numbers of zones. When the number is larger, say over 300, other techniques should be used to avoid the unnecessary computation of distances between points far from each other. This is done by Pace and Barry (1997), and in the S-PLUS module, by using quadtrees to partition the locations. Pace and Barry also suggest the use of an alternative neighbourhood criterion stipulating the *m* nearest neighbours to *i* as being included, for fixed *m*. This avoids overlarge numbers of neighbours in denser parts of the map, while preserving neighbourhood influence in less dense parts. Finally, the imple-

mentation of higher order lags based on contiguities is required, using methods based on Anselin and Smirnov (1996).

The Moran and Geary coefficients may be tested using analytical expectations and variances (Cliff and Ord, 1973) based largely on the neighbourhood structure assumed in the spatial weighting matrix, and are asymptotically normally distributed. New measures have been introduced among others by Brett and Pinkse (1997) for spatial independence based on characteristic functions.

The data set used for the example given here was included in the materials of the ESDA with LISA conference held in Leicester in 1996[6] , see Brunsdon (1998) and Dykes (1998). The data are for the five midwest states of Illinois, Indiana, Michigan, Ohio, and Wisconsin, and stems from the 1990 census. The boundary data stems from the same source, but the units used are not well documented; there are 437 counties included in the data set.

The implementation of these measures is not a major difficulty, but once again we find that the accessing of the individual elements in the neighbourhood list vectors imposes a time cost. Since in future versions we want to permit subsetting, the weight matrix sums needed cannot be computed once for all, but need to be refreshed at each run. Even for this relatively large case, using built-in full matrix methods is only slightly slower than interpreted look-up for the spatial lagging of a variable. There seem to be good reasons for moving at least a kernel of neighbourhood list functions from interpreted to compiled code, although, for smaller numbers of zones, response time is acceptable.

While global measures permit us to test for spatial patterning over the whole study area, it may be the case that there is significant autocorrelation in only a smaller section, which is swamped in the context of the whole. Both distance statistics (Getis and Ord, 1992, 1996, Ord and Getis, 1995), and the local indicators of spatial association derived by Anselin (1995b, see also Getis and Ord, 1996), resemble passing a moving window across the data, and examining dependence within the chosen region for the site on which the window is centred. The specifications for the window can vary, using perhaps contiguity or distance at some spatial lag from the considered zone or point. In addition, Anselin (1996) has suggested that a plot of x_i against its spatial lag $\sum_j w_{ij}x_j$, termed a Moran scatterplot, particularly used with dynamic linked visualization, may assist in revealing local patterning.

The local indicators of spatial association presented by Getis and Ord (1996) have been implemented in interpreted code, and the G_i and G_i^* statistics checked against Chen and Getis (1998) for accuracy. Once again, it is clear that accessing the distance matrix used, here for neighbours within 0.45 distance units of each other, in interpreted code imposes time penalties, although not more than for the global measures. The function `lisa.g()` returns an $n \times 2$ matrix with values of the G_i and G_i^* statistics, ready for further analysis and mapping.

In the example below, we examine the value of these local indicators for a variable displaying moderate, but still significant, global spatial autocorrelation, the percentage of the population over 25 with a higher or professional degree. A summary of G_i^* is presented, together with the values of the statistics for counties

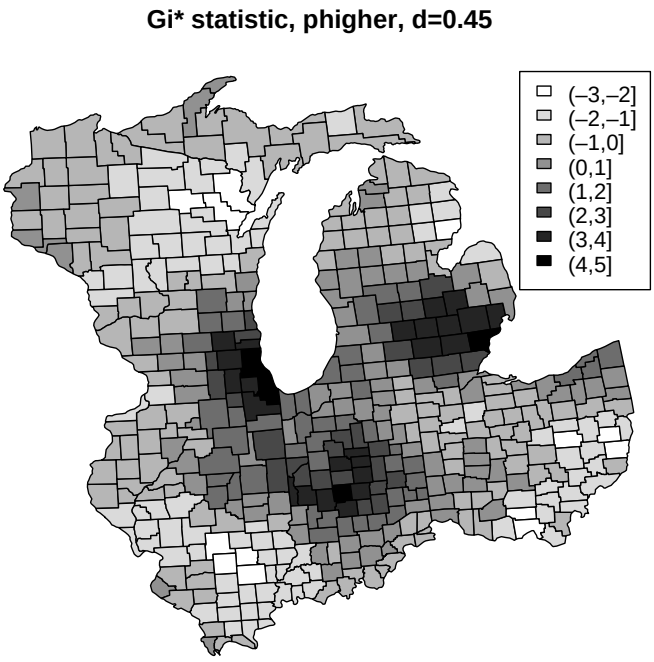

Fig. 4 G_i^* statistic for percentage over 25 years with a higher/professional degree, 5 US midwestern states, 1990 census.

with a difference between G_i and G_i^* statistics of more than 1; Washtenaw County in Michigan, contains Ann Arbor. The values of G_i^* are shown in Figure 4.

```
> mw <- read.table("midwest.tab",T, row.names=1)
> names(mw)
[1] "id.no"     "poly.no"   "pdens"     "pnonwhite" "pschool"   "pcollege"
[7] "phigher"   "ppoverty"  "met"
> mw.d45 <- getw("mw-d0.45.gal")
Read 7955 items
> testw(mw.d45)
> global(mw.d45, mw$phigher)

Global spatial dependence measures (randomization assumption):
(Weights row-standardized)

         Statistic Expectation Variance Std. deviate p-value
Morans I 0.08757   -0.00229    0.00032   5.0445      4.547e-07
Gearys C 0.91966    1.00000    0.00050  -3.6006      0.0003175

> mw.g <- lisa.g(mw.d45, mw$phigher)
> summary(mw.g$Gstari)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.4680 -0.9784 -0.1537  0.1177  0.9346  4.3550
> rownames(mw.g) <- rownames(mw)
> mw.g[abs(mw.g$Gi-mw.g$Gstari) > 1,]
                      Gi      Gstari
CHAMPAIGN (IL) -0.1720360  1.11684010
JACKSON (IL)   -2.1725006 -1.15061936
MONROE (IN)     0.6055050  1.60587581
WASHTENAW (MI)  2.3217825  3.70831371
DANE (WI)      -1.0906615 -0.04876026
```

Estimation methods for models using lattice data and taking spatial dependence into account are as mature as global statistics for spatial autocorrelation (Ord, 1975, Hepple, 1976); the form of model most commonly used is known as the simultaneous autoregression (SAR). Ten years have now passed since Anselin and Griffith (1988) surveyed the regional science and economic geography literature to see how far these methods were being applied to data sets for which they should have been suited. The low penetration they reported seemed related to the lack of access to these tools in standard statistical packages, addressed subsequently by Anselin and Hudak (1992), Bivand (1992), Griffith (1993), and others. The most substantial effect has been achieved by Anselin's "SpaceStat" program, permitting the estimation of most of the specification tests and models described in the literature (1995a).

A problem solved in Anselin et al. (1996) is that of tests for spatial lag and spatial error specifications being mutually contaminated by each other, that is the original LM test for non-zero ρ also responds to non-zero λ and vice-versa. The new tests take into account the possible non-zero value of the nuisance parameter, and appear to discriminate well between the two alternative forms.

We will test the residuals from a least squares model of crime, related to income and housing value for 49 neighbourhoods in Columbus, Ohio, from the data set provided with SpaceStat, and extensively used in the literature. This model is also used in Anselin et al. (1996) to illustrate the modified tests which have been made robust to the influence of the nuisance parameter. The following example shows the regression results in brief, followed by a table of specification tests for spatial dependence:

```
> col.lm <- lm(CRIME ~ INCOME + HOUSING, data=columbus)
> col.lm

Coefficients:
(Intercept)       INCOME      HOUSING
    68.6189      -1.5973      -0.2739

> lm.sptests(col.lm, colw)

Diagnostics for spatial dependence:
       statistic df prob.
LMerr  5.7230    1  0.01674
LMlag  9.3634    1  0.00221
RLMerr 0.0795    1  0.77797
RLMlag 3.7199    1  0.05377
SARMA  9.4429    2  0.00890
```

In the test results, the values obtained correspond to Table 2 in Anselin et al. (1996, p. 87), with LMerr as their LM_ψ, LMlag as LM_ϕ, RLMerr as LM^*_ψ, RLMlag as LM^*_ϕ, and SARMA as $LM_{\phi\psi}$. The results also agree with SpaceStat output. Moran scatterplots corresponding to the components of the LM tests are shown in Figure 5.

Ord (1975) gives the Maximum Likelihood methods for estimating the spatial lag and spatial error SAR models; no satisfactory alternatives have been found subsequently, chiefly because of the important role of the Jacobian expressing the spatial transformation of either the dependent variable in the spatial lag model, or the disturbance in the spatial error model. To complete the model, the variance-

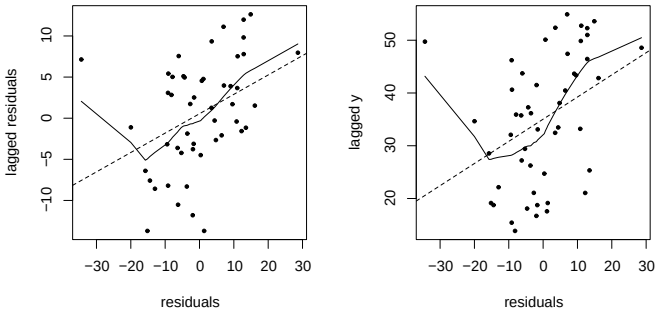

Fig. 5 Moran scatterplots for the residual and dependent variable, Columbus data set.

covariance matrix of the parameters needs to be estimated. In many cases it is approximated numerically following non-linear optimization of the likelihood function, but SpaceStat derives its estimates of the asymptotic standard errors analytically (Anselin 1995a, Anselin and Hudak, 1992). For larger N, this can take considerable time, requiring the inversion of an $N \times N$ matrix. As Pace and Barry (1997) have conclusively demonstrated, a feasible solution to modelling situations with large N is to exploit the sparse nature of the spatial weighting matrix, both saving memory and making computation practical in reasonable time without supercomputer resources. They also provide a profile likelihood solution to the calculation of coefficient estimate standard errors, avoiding the computation of the information matrix.

Following Pace and Barry (1997) and the `S-PLUS` `slm()` function, use is made of both eigenvalue computation and sparse matrix methods[7] to implement maximum likelihood models for three types: lag, mixed, and error models. The mixed model is a lag model but also includes the lagged **X** variables on the right hand side. The `smlr()` function has been implemented using the same interface, print, and summary object-oriented approach as the regular `lm()` function, which is used to construct the model frame for estimation. As yet neither weighting nor subsetting are supported, but at least subsetting is not difficult to add to the function as it stands. The log likelihood function is maximized with reference to the spatial parameter estimate using `optimize()`, and the remaining parameter estimates are reached using generalized least squares. The standard errors of the coefficients are, following Pace and Barry (1997), computed using the restricted least squares estimator, rather than computing the information matrix directly.

```
> col.lag <- smlr(formula = CRIME ~ INCOME + HOUSING,
+         data = columbus, colw, type="lag", method="sparse")
> summary(col.lag)

Coefficients: (restricted least squares standard errors)
            Estimate  Std. Error z value Pr(>|z|)
(Intercept) 45.079468  4.046868  11.1393 < 2.2e-16 ***
INCOME      -1.031616  0.285543  -3.6128 0.0003029 ***
HOUSING     -0.265927  0.088192  -3.0153 0.0025670 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Rho: 0.43102 LR test value: 9.9733 p-value: 0.0015883
```

```
Log likelihood: -182.3904 for lag model
Residual variance (sigma squared): 95.495
Number of observations: 49
Number of parameters estimated: 4
AIC: 372.78 SC: 380.35
```

The above example shows the calculation and results of computing the spatial lag model for the Columbus crime response variable. Compared with Anselin (1995a, tables 27.2 and 27.3), the results are identical with the exception of the standard errors of the regression coefficients, and consequently the significant tests on the β coefficient estimates. The Likelihood Ratio test on ρ is correct, as are the measures of fit. The reason for this discrepancy is that the restricted least squares estimates ignore the fact that $\frac{\partial^2 \ell}{\partial \beta \partial \rho} \neq \mathbf{0}$. In the error case, however, $\frac{\partial^2 \ell}{\partial \beta \partial \lambda} = \mathbf{0}$, so that the results of the spatial error model shown below agree with Anselin (1995a, tables 29.2 and 19.3) also with regard to the standard error results:

```
> col.error <- smlr(formula = CRIME ~ INCOME + HOUSING,
+          data = columbus, colw, type="error", method="sparse")
> summary(col.error)

Coefficients: (restricted least squares standard errors)
            Estimate  Std. Error z value Pr(>|z|)
(Intercept) 59.893276  5.366141  11.1613 < 2.2e-16 ***
INCOME      -0.941312  0.330568  -2.8476 0.0044056 **
HOUSING     -0.302252  0.090476  -3.3407 0.0008357 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Lambda: 0.56179 LR test value: 7.9934 p-value: 0.0046949

Log likelihood: -183.3804 for error model
Residual variance (sigma squared): 95.574
Number of observations: 49
Number of parameters estimated: 4
AIC: 372.76 SC: 378.44
```

Finally, we check the test for the Common Factor hypothesis against Anselin's (1995a, table 29.3) LR test result, finding, as he does, that the hypothesis cannot readily be rejected, and that thus the spatial error specification can be accepted, even though the value of the log likelihood function is higher for the mixed model; the information criteria AIC and SC also favour the error model here.

```
> LR.smlr(col.mixed, col.error)

                  mixed    error
LL             -181.393 -183.380
AIC             374.787  372.761
SC              386.138  378.436
Sigma squared    91.791   95.574

LR test value: 3.9739 p-value: 0.13711
```

As global measures of spatial association have been supplemented by local indicators, Fotheringham, Charlton, and Brunsdon (1996, 1997) and Brunsdon, Fotheringham, and Charlton (1996) have been developing weighting schemes to allow possible differences in local parameter estimates for regression models to be revealed. Moving from the global to local settings, one would perhaps expect the local parameter estimates to vary, but within the bounds of their global standard

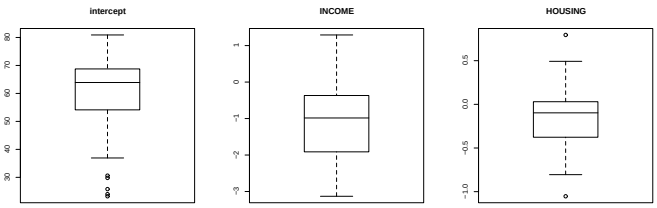

Fig. 6 Boxplots for geographically weighted regression coefficient estimates, Columbus data set.

error based confidence intervals, that is with divergences of more than ± 2 less than five times in a hundred. The weighting scheme used so far is distance based, weighting zone *i* with unity, and with weights declining with increasing distance from *i*. There are similarities with kernel regression techniques, although these use weighting in attribute space, rather than across the observations. Currently, cross-validation is used to select an appropriate global bandwidth parameter, which then determines the form of the distance decay function used to define the weights for each observation. There are clearly substantial difficulties involved in making statistical inferences from results of this kind of procedure, although it has proved very useful in showing up missing variables.

Having access to the Fortran source code for geographically weighted regression[8] , it has been possible not only to re-create its functioning in `R` , but also to check results from the same data set, providing assurance that the two solutions are equivalent. Using the `dist()` and `optimize()` functions mentioned above, and the `lm.wfit()` function for fitting weighted least squares models, it proved uncomplicated to provide the basic framework for geographically weighted regression. It remained to replicate the weighting function, which in the original code is a Gaussian distance decay based weighting scheme $e^{-d_{ik}^2/h^2}$, where d_{ik} is the distance between *i* and *k*, and *h* is the bandwidth. In `R` the function was implemented by computing the squared distance matrix once for all, and stepping across its columns in successive weighing passes.

Even working in interpreted code, the fact that the functions used are compiled, and accessed without much other computation made the `R` solution comparable to the compiled Fortran program in speed, both for cross-validation and for estimation, following the calibration of the bandwidth. One reason for the relative speed of the `R` version is that the distance matrix is only computed once, rather than being calculated element for element at each pass through the *n* weighting iterations during fitting, and *n* times the number of optimization function calls during cross-validation. Using a simple interpreted function also permits the testing of alternative functional forms. Further progress will depend on the introduction of quadtree or similar techniques to hinder the use of weights so close to zero as to make little difference to the fitted parameters for each pass.

Using the Columbus data set, cross-validation gives a bandwidth of 3.217 distance units. Figure 6 shows that, while the income and housing variables have significant negative coefficient estimates in all the global models reported above, sign changes do occur in both cases, more particularly for housing.

6 Concluding remarks

We have demonstrated that R can be extended to cater for spatial statistical data analysis through the porting of existing code written for S-PLUS and §. Much of the code is already available on CRAN, the R archive site, and more will follow. Compared with S-PLUS , especially its widespread Windows version, it is very easy to build your own C or FORTRAN code and load your functions into R . This is the reason for success in porting many packages to R . For users within the steadily growing Unix/Linux communities, compiling and linking user-written or ported C or FORTRAN functions can be done on the fly with simple commands like "R COMPILE yourlib", but recently also straightforward ports with R on Windows have succeeded. Both use the GNU C compiler, which is bundled with every Linux distribution. The same task becomes more complicated when using S-PLUS and Windows, requiring specific commercial C and FORTRAN compilers, and finding the correct settings for integration with S-PLUS has not been easy. Finally, free availability, and ease of use, even for development tasks, are very important advantages of R , which surely will show feedback in the future in terms of new packages, a growing team of users and developers and rapid bug fixes.

We hope that we have shown that R provides a viable platform for work both in developing spatial statistics, and for their application[9] . Since R is also available for Windows platforms, we are confident that it will be possible to increase access even to relatively fresh research contributions within an environment supporting statistical analysis and graphical visualization. We trust that the examples given above, although not part of problem-oriented research, suggest that realistic data sizes are feasible. Some parts of the work reported here are already distributed through the R archives, and we hope that the remainder will achieve this status in due time, after more adequate testing.

Notes

1. It had been our intention to show how these packages could use an interface between R and GIS software, but because of delays in the release of GRASS 5.0, no specific results will be presented.
2. Open source is a registered certification mark of software in the public interest; see http://www.opensource.org.
3. S-PLUS is owned and copyrighted with all rights reserved by MathSoft, Inc.
4. available online from the book's website
5. Splancs code is available from http://www.maths.lancs.ac.uk/~rowlings/Splancs/.
6. http://midas.ac.uk/argus/esdalisa/midwest/Data.html.
7. Use is made of the "sparse" software available from Netlib (http://www.netlib.org), by Kundert and Sangiovanni-Vincentelli (1988).
8. http://www.ncl.ac.uk/~ngeog/GWR/.
9. Both of the authors use R for teaching, including use in a fairly complete course based on Bailey and Gatrell (1995).

References

1. Abelson, H., Sussman, G. and Sussman, J. 1996 *Structure and interpretation of computer programs.* (Cambridge MA: MIT Press).
2. Akima, H. 1996 Algorithm 761: scattered-data surface fitting that has the accuracy of a cubic polynomial. *ACM Transactions on Mathematical Software*, 22, 362–371.
3. Anselin, L. 1988 *Spatial econometrics: methods and models.* (Dordrecht: Kluwer).
4. Anselin, L. 1995a SpaceStat, a software program for the analysis of spatial data, version 1.80. Regional Research Institute, West Virginia University, Morgantown, WV.
5. Anselin, L. 1995b Local indicators of spatial association - LISA. *Geographical Analysis*, 27, 93–115.
6. Anselin, L. 1996 The Moran scatterplot as an exploratory spatial data analysis tool to assess local instability in spatial association. In M. M. Fischer, H. J. Scholten and D. Unwin (eds) *Spatial analytical perspectives on GIS*, (London: Taylor & Francis), 111–125.
7. Anselin, L., Bera, A. K., Florax, R. and Yoon, M. J. 1996 Simple diagnostic tests for spatial dependence. *Regional Science and Urban Economics*, 26, 77–104.
8. Anselin, L. and Griffith, D. A. 1988 Do spatial effects really matter in regression analysis? *Papers of the Regional Science Association*, 65, 11–34.
9. Anselin, L. and Hudak, S. 1992 Spatial econometrics in practice: a review of software options. *Regional Science and Urban Economics*, 22, 509–536.
10. Anselin, L. and Smirnov, O. 1996 Efficient algorithms for constructing proper higher order spatial lag operators. *Journal of Regional Science*, 36, 67–89.
11. Bailey, T. and Gatrell, A. (1995) *Interactive spatial data analysis.* Harlow: Longman.
12. Becker, R. 1994 A brief history of S. In P. Dirschedl and R. Osterman (eds) *Computational statistics*, (Heidelberg: Physica Verlag), 81–110.
13. Becker, R., Chambers, J. and Wilks, A. 1988 *The new S language.* (New York: Chapman and Hall).
14. Bivand, R. S. 1992 SYSTAT-compatible software for modelling spatial dependence among observations. *Computers and Geosciences*, 18, 951–963.
15. Bivand, R. 1998 Software and software design issues in the exploration of local dependence. *The Statistician*, 47, 499–508.
16. Brett, C. and Pinkse, J. 1997 Those taxes are all over the map!: a test for spatial independence of municipal tax rates in British Columbia. *International Regional Science Review*, 20, 131–151.
17. Brunsdon, C. 1998 Exploratory spatial data analysis and local indicators of spatial association with XLISP-STAT. *The Statistician*, 47, 471–484.
18. Brunsdon, C., Fotheringham, A. S. and Charlton, M. 1996 Geographically weighted regression: a method for exploring spatial nonstationarity. *Geographical Analysis*, 28, 281–298.
19. Chambers, J. and Hastie, T. (eds) 1992 *Statistical models in S.* (New York: Chapman and Hall).
20. Chen, D. and Getis, A. 1998 Point pattern analysis (PPA). Software package and documentation, Department of Geography, San Diego State University.
21. Cliff, A. D. and Ord, J. K. 1973 *Spatial autocorrelation.* (London: Pion).
22. Cliff, A. D. and Ord, J. K. 1981 *Spatial processes - models and applications.* (London: Pion).
23. Cressie, N. A. C. 1993 *Statistics for spatial data.* (New York: Wiley).
24. Dykes, J. 1998 Cartographic visualization: exploratory spatial data analysis and local indicators of spatial association using Tcl/Tk and cdv. *The Statistician*, 47, 485–497.

25. Fotheringham, A. S., Charlton, M. and Brunsdon, C. 1996 The geography of parameter space: an investigation of spatial non-stationarity. *International Journal of Geographical Information Systems*, 10, 605–627.
26. Fotheringham, A. S., Charlton, M. and Brunsdon, C. 1997 Two techniques for exploring non-stationarity in geographical data. *Geographical Systems*, 4, 59–82.
27. Gatrell, A. and Bailey, T. 1996 Interactive spatial data analysis in medical geography. *Social Science & Medicine*, 42, 843–855.
28. Getis, A. and Ord, J. K. 1992 The analysis of spatial association by use of distance statistics. *Geographical Analysis*, 24, 189-206 (cf. also *Geographical Analysis*, 25, 276).
29. Getis, A., Ord, J. K. 1996 Local spatial statistics: an overview. In P. Longley and M. Batty (eds) *Spatial analysis: modelling in a GIS environment* (Cambridge: Geoinformation International), 261–277.
30. Griffith, D. A. 1988 *Advanced spatial statistics.* (Dordrecht: Kluwer).
31. Griffith, D. A. 1993 *Spatial regression analysis on the PC: spatial statistics using SAS.* (Washington DC: Association of American Geographers).
32. Haining, R. P. 1990 *Spatial data analysis in the social and environmental sciences.* (Cambridge: Cambridge University Press).
33. Hepple, L. W. 1976 A Maximum Likelihood model for econometric estimation with spatial series. In I. Masser (ed) *Theory and practice in regional science*, (London: Pion), 90–104.
34. Ihaka R, 1998 R: past and future history. Unpublished paper, `http://www.ci.tuwien.ac.at/R/doc/interface98-paper/paper.html`.
35. Ihaka, R. and Gentleman, R. 1996 R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5, 299-314.
36. Kaluzny, S. P., Vega, S. C., Cardoso, T. P. and Shelly, A. A. 1996 *S+SPATIALSTATS users manual version 1.0.* Seattle: MathSoft Inc.
37. Kundert, K. and Sangiovanni-Vincentelli, A. 1988 Sparse user's guide, version 1.3a. Department of Electrical Engineering and Computer Science, University of California, Berkeley.
38. Levine, N. 1996 Spatial statistics and GIS. *Journal of the American Planning Association*, **62**, 381–391.
39. Majure, J. 1995 An Object-oriented Framework for Geostatistical Modeling in S+. Iowa State University, Geographic Information Systems, Support and Research Facility. Available at `http://www.gis.iastate.edu/SGeoStat/homepage.html`.
40. Ord, J. K. 1975 Estimation methods for models of spatial interaction. *Journal of the American Statistical Association*, 70, 120–126.
41. Ord, J. K. and Getis, A. 1995 Local spatial autocorrelation statistics: distributional issues and an application. *Geographical Analysis*, 27, 286–306.
42. Pace, R. K. and Barry, R. 1997 Quick computation of spatial autoregressive estimators. *Geographical Analysis*, 29, 232–247.
43. Raymond, E. 1997 The cathedral and the bazaar. `http://www.ccil.org/~esr/writings/cathedral-paper.html`.
44. Ripley, B. 1981 *Spatial statistics.* (New York: Wiley).
45. Rowlingson, B. and Diggle, P. 1993 Splancs: spatial point pattern analysis code in S-Plus. *Computers and Geosciences*, 19, 627–655.
46. Tierney, L. 1990 *LISP-STAT: an object-oriented environment for statistical computing and dynamic graphics.* (New York: Wiley).
47. Upton, G. J. G. and Fingleton, B. 1985 *Spatial data analysis by example: point pattern and quantitative data.* (London: Wiley).
48. Venables, W. N. and Ripley, B. D. 1997 *Modern applied statistics with S-PLUS.* (New York: Springer); `http://www.stats.ox.ac.uk/pub/MASS2`.