

Computing the Jacobian in Gaussian spatial models: an illustrated comparison of available methods

Abstract

When fitting spatial regression models by maximum likelihood using spatial weights matrices to represent spatial processes, computing the Jacobian, $\ln(|\mathbf{I} - \lambda\mathbf{W}|)$, remains a central problem. In principle, and for smaller data sets, the use of the eigenvalues of the spatial weights matrix provides a very rapid and satisfactory resolution. Analytical eigenvalues are available for large regular grids. For larger problems not on regular grids, including those induced in spatial panel and dyadic (network) problems, solving the eigenproblem is not feasible, and a number of alternatives have been proposed. This paper surveys selected alternatives, and comment on their relative usefulness, covering sparse Cholesky and sparse LU factorizations, and approximations such as Monte Carlo, Chebyshev, and using lower order moments with interpolation. The results are presented in terms of componentwise differences between sets of Jacobians for selected data sets. In conclusion, recommendations are made for a number of analytical settings.

1 Introduction

Spatial regression models are fitted in a wide range of disciplines, from political and regional science to epidemiology and ecology. Testing for spatial autocorrelation, and the study of methods for specifying and fitting spatial regression models have been central topics in spatial analysis and quantitative geography for many decades, but have not lost their relevance and research interest. Perhaps the wider availability of software for fitting spatial regression models signals the need for a review of topics such as that of the computation of the Jacobian, to establish how far we have come, and to indicate areas for fresh research. We have chosen to restrict our attention to Gaussian spatial

1 models with a continuous response variable, although conclusions reached here
2 may also be extended to the discrete response case.

3 Where maximum likelihood methods are chosen for fitting spatial regres-
4 sion models, problems can arise when data sets become large because it is
5 necessary to compute the determinant of an $n \times n$ matrix when optimizing the
6 log-likelihood function, where n is the number of observations. As Bayesian
7 methods for spatial regression may also require the handling of the same ma-
8 trix, they may face the same technical issues of memory management and
9 algorithm choice. We have chosen here to term the problem we are considering
10 the “Jacobian”, although the expression of interest is $\ln|\mathbf{I} - \lambda\mathbf{W}|$, where $|\cdot|$
11 here denotes the determinant of matrix \cdot , \mathbf{I} is the identity matrix, λ is a spatial
12 coefficient, and \mathbf{W} is an $n \times n$ matrix of fixed spatial weights, so the problem
13 perhaps ought to be termed finding the logarithm of the determinant of the
14 Jacobian. In order to optimize the log-likelihood function with respect to λ ,
15 successive new values of this calculation are required.

16 The often sparse matrix of spatial weights \mathbf{W} represents a graph of rela-
17 tionships between observations. For small numbers of observations, there are
18 no difficulties in treating the spatial weights matrix as dense, and computing
19 the log-determinant using the eigenvalues of \mathbf{W} . As the determinant of $\mathbf{I} - \lambda\mathbf{W}$
20 takes values between 1 when $\lambda = 0$, and 0 when λ is at one of its boundaries
21 and the matrix is singular, the logarithm of the determinant is bounded by
22 0 and $-\infty$, often being a negative number of large absolute value. Numerical
23 stability may be affected by the stage in algorithms at which logarithms are
24 taken. When available memory and time is insufficient, and solving the eigen-
25 problem is not an option, alternatives may be chosen. These include sparse
26 matrix methods such as LU and Cholesky factorization, analytical eigenvalues
27 for regular grids, and approximations of a number of kinds. In this compar-
28 ison, we examine eigenvalue methods (Ord, 1975; Griffith and Sone, 1995),
29 sparse LU and Cholesky factorizations (Pace and Barry, 1997d), and trace-
30 based (Smirnov and Anselin, 2009), Monte Carlo (Barry and Pace, 1999), and
31 Chebyshev approximations (Pace and LeSage, 2004). We do not include char-
32 acteristic polynomial methods (Smirnov and Anselin, 2001), or associated sin-
33 gular value decomposition or eigendecomposition methods.

34 Although it may seem that the computation of the Jacobian is an unimport-
35 tant technical detail in comparison with the substantive concerns of analysts,
36 we feel that this review may provide helpful insight for practical research using
37 spatial regression with spatial weights matrices representing spatial processes.
38 Users of GeoDa and OpenGeoDa (Anselin et al., 2006) often wonder why in-
39 trinsically asymmetric spatial weights, such as those generated by k -nearest
40 neighbor criteria, do not work for larger data sets — the reason being that

1 the Jacobian in larger data sets is computed using a characteristic poly-
2 nomial method that assumes intrinsic symmetry (Smirnov and Anselin, 2001).
3 Users of spatial regression functions in the R package **spdep** ask why they ex-
4 ceed memory limits with larger data sets — they have been using the default
5 eigenvalue method of computing the Jacobian, but could have selected an al-
6 ternative method. Users of maximum likelihood spatial regression functions
7 in StataTM are at present restricted to data sets of a size for which the eigen-
8 value approach is appropriate. In the MatlabTM Spatial Econometrics toolbox,
9 the user of maximum likelihood and Bayesian spatial regression functions is
10 offered the choice of the sparse LU method or Monte Carlo approximations, in
11 addition to an interpolation method. Consequently, we wish to offer guidance
12 on the alternatives available, some results on their relative accuracy, and notes
13 in conclusion on their relative run times.

14 Walde et al. (2008) report on the outcome of a broader comparison of dif-
15 ferent methods and approximations for computing the Jacobian as part of a
16 “contest” between maximum likelihood and generalized method of moments
17 model fitting methods. We show below that their conclusions with regard to
18 calculation of the Jacobian based on eigenvalues and on Cholesky factoriza-
19 tion are incorrect. This points up our position that implementation is of the
20 essence, and that cross-checking implementations against theoretical develop-
21 ments is of value. In addition, the accuracy and stability of numerical methods
22 require care both in theory and in implementation, and are reported on in de-
23 tail below, in addition to execution timings where relevant. The underlying
24 concerns are that methods for computing the Jacobian should be robust, ac-
25 curate, clearly implementable using well-tried code, and that inferences made
26 by users should not be impacted by choice of numerical methods.

27 We continue by defining spatial regression models to be treated here, the
28 data sets to be used for this comparison, and how we, following Higham (2002),
29 understand accuracy in fixed precision computing. We move to a consideration
30 of eigenvalue-based approaches to the Jacobian, including the consequences of
31 asymmetry in the weights matrix, and the presence of no-neighbor observa-
32 tions and unconnected subgraphs. Next, we present and analyze sparse ma-
33 trix LU and Cholesky factorizations, and show their performance compared
34 to eigenvalue methods, also for a large data set with a million observations,
35 constituting a “useful impartial benchmark” suggested by an anonymous ref-
36 eree. Finally, we examine other approximate methods, including Monte Carlo,
37 Chebyshev, and trace-based methods; all of these involve some truncation,
38 leading to impaired accuracy in some ranges of values of the spatial coefficient.
39 The results reported here have been calculated using R (R Core Team, 2012),
40 and many of the computations have been independently checked using the

1 Spatial Econometrics toolbox¹ for MatlabTM, provided as source code together
 2 with extensive documentation, and customized MatlabTM code.

3 1.1 Spatial regression models

4 Assuming that the variance of the disturbance term is constant, we start from
 5 the standard linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \boldsymbol{\sigma}^2 \mathbf{I}) \quad (1)$$

6 where \mathbf{y} is an $(n \times 1)$ vector of observations on a dependent variable taken at
 7 each of n locations, \mathbf{X} is an $(n \times k)$ matrix of exogenous variables, $\boldsymbol{\beta}$ is an $(k \times 1)$
 8 vector of parameters, $\boldsymbol{\varepsilon}$ is an $(n \times 1)$ normally distributed vector of disturbances
 9 with zero mean, fixed variance $\boldsymbol{\sigma}^2$ and identity matrix \mathbf{I} . There are a number of
 10 alternative forms of spatial regression models using spatial weights matrices to
 11 represent spatial processes; here we consider the simultaneous autoregressive
 12 form, because the computation of the Jacobian presents similar challenges
 13 for the conditional autoregressive, autoregressive response and spatial moving
 14 average representations. This model may be written as (Cliff and Ord, 1973;
 15 Ord, 1975; Ripley, 1981):

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}, \quad \mathbf{u} = \lambda \mathbf{W}\mathbf{u} + \boldsymbol{\varepsilon}, \quad (2)$$

16 where λ is a scalar spatial error parameter, and \mathbf{u} is a spatially autocorrelated
 17 disturbance vector with zero mean and variance and covariance terms specified
 18 by a fixed spatial weights matrix \mathbf{W} and a coefficient λ . The disturbance vector
 19 \mathbf{u} is defined as:

$$\mathbf{u} \sim N(\mathbf{0}, \boldsymbol{\sigma}^2 (\mathbf{I} - \lambda \mathbf{W})^{-1} (\mathbf{I} - \lambda \mathbf{W}')^{-1}), \quad (3)$$

20 where \mathbf{W}' denotes the transpose matrix of \mathbf{W} .

21 It is usual in the literature to define the neighbor relationship in terms
 22 of sets $N_{(i)}$ of neighbors of zone or site i . These are coded in the form of a
 23 weights matrix \mathbf{W} , with a zero diagonal, and the off-diagonal non-zero elements
 24 often scaled to sum to unity in each row (termed row-standardized or row-
 25 normalized, and also row-stochastic, weights matrices), with typical elements:

$$w_{ij} = \frac{c_{ij}}{\sum_{j=1}^n c_{ij}} \quad (4)$$

¹<http://www.spatial-econometrics.com/>.

1 where $c_{ij} = 1$ if i is a neighbor of j and $c_{ij} = 0$ otherwise. This implies no use of
2 other information than that of neighborhood set membership. Set membership
3 may be defined on the basis of shared boundaries, of centroids lying within
4 distance bands, or other a priori grounds. In general, the number of neighbors
5 for each observation is small compared to n , so that \mathbf{W} is usually sparse. It
6 may be reasonable, based on knowledge of the underlying spatial interaction
7 processes, to specify c_{ij} in other ways, for example trade or migration flows, or
8 in other ways that introduce asymmetry. Indeed, the spatial weights defined
9 here by row-standardization are asymmetric, but if $c_{ij} = c_{ji}$ for all i, j pairs,
10 the matrix \mathbf{W} is similar to a symmetric matrix.

11 Kelejian and Prucha (2010, pp. 55–56) discuss the parameter space of the
12 autoregressive parameter λ , in particular the impacts of different choices of
13 normalization schemes. They suggest that normalization by the largest eigen-
14 value of the non-normalized weights, or by the minimum of the maxima of
15 row and column sums of the non-normalized weights, may be preferable to
16 row-standardization, and that some practices reflected in the literature may
17 not hold in general. In addition, Tiefelsdorf et al. (1999) introduce a variance-
18 stabilizing coding scheme, moderating some of the unfortunate side effects
19 induced by the use of row-standardization, especially with regard to observa-
20 tions often at the edges of study areas with few neighbors. For the purposes
21 of this paper, we choose in the main to stay with row-standardization, feeling
22 that these results deserve to be studied fully in separate work.

23 Ord (1975) gives a maximum likelihood method for estimating the spatial
24 error simultaneous autoregressive model. Unlike the time series case, the loga-
25 rithm of the determinant of the $(n \times n)$ matrix $(\mathbf{I} - \lambda\mathbf{W})$ does not tend to zero
26 with increasing sample size; it constrains the parameter values to their feasible
27 range between the inverses of the smallest and largest eigenvalues of \mathbf{W} , ζ_{\min}
28 and ζ_{\max} , i.e.:

$$\lambda \in (1/\zeta_{\min}, 1/\zeta_{\max}), \quad (5)$$

29 where ζ_i , $i = 1, \dots, n$ are the eigenvalues of \mathbf{W} .

30 Let us observe that in the case of positive autocorrelation $\ln(|\mathbf{I} - \lambda\mathbf{W}|) \rightarrow$
31 $-\infty$ for $\lambda \rightarrow 1/\zeta_{\max}$, where $\ln|\mathbf{I} - \lambda\mathbf{W}|$ denotes the determinant of the matrix
32 $\mathbf{I} - \lambda\mathbf{W}$. The log-likelihood function for the spatial error model is of the form:

$$\begin{aligned} \ell(\boldsymbol{\beta}, \lambda, \sigma^2) = & -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) + \ln(|\mathbf{I} - \lambda\mathbf{W}|) \\ & - \frac{1}{2\sigma^2} [(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{I} - \lambda\mathbf{W})'(\mathbf{I} - \lambda\mathbf{W})(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})]. \end{aligned} \quad (6)$$

1 As we can see, the problem is one of balancing the log-determinant term
2 against the sum of squares term. When λ approaches the ends of its feasible
3 range, the log-determinant term may swamp the sum of squares term (Bivand,
4 1984). In this sense, the log-determinant term needs to be estimated in order
5 to fit the spatial coefficient, but its value is not typically reported, being simply
6 one of the terms making up the value of the log-likelihood at the optimum.
7 The Jacobian problem addressed in this review is that of computing the $\ln(|\mathbf{I} -$
8 $\lambda\mathbf{W}|)$ term for values of λ proposed by the line search function. In addition,
9 with moderate to large n , the calculation of the analytical variance-covariance
10 matrix of the model coefficients is impeded by the need to handle dense $n \times n$
11 matrices. The variance-covariance matrix may be approximated by a numerical
12 Hessian, the computation of which also involves the values of the Jacobian.
13 Typically, the number of calls to the Jacobian function involved in computing
14 the numerical Hessian is larger than in the line search to find $\hat{\lambda}$ at the optimum
15 (LeSage and Pace, 2009, pp. 54–60).

16 1.2 Accuracy

17 Following Higham (2002, pp. 3–5), we report summaries of the component-
18 wise absolute difference $|x_i - \hat{x}_i|$ between different Jacobian values for a range
19 of values i of the spatial coefficient; $|\cdot|$ is here the absolute value of scalar \cdot .
20 In some cases we also report summaries of componentwise relative differences:
21 $|x_i - \hat{x}_i|/|x_i|$. When the spatial coefficient is zero, the Jacobian is also zero. It
22 should be borne in mind that all finite-precision computation necessarily in-
23 volves round-off error, although most operations are programmed in ways that
24 reduce this to the minimum possible. The unit round-off for double precision
25 in IEEE arithmetic is $1.11e - 16$ (Higham, 2002, p. 41), providing a baseline
26 for deterioration in accuracy.

27 It is also worth noting that the Jacobian values are part of the value of the
28 log-likelihood function utilized in numerical optimization, where the tolerance
29 of the line search or optimizer is typically larger than the differences between
30 Jacobian computation methods. In R, the default tolerance of the line search
31 function `optimize`² is $1.2207e - 04$. This means that if the difference between
32 Jacobian values is smaller than the tolerance being used, then the chosen value
33 of the spatial coefficient is likely only to differ by an amount that does not affect
34 the optimizer. Naturally, a different value of the spatial coefficient affects both
35 the sum of squared errors term and the Jacobian, in opposite directions, so
36 that a maximum componentwise absolute difference of less than the optimizer

²This function, like the MatlabTM function `fminbnd` used in the Spatial Econometrics toolbox, is an implementation of Brent (1973).

1 tolerance should give effectively equivalent results.

2 1.3 Data sets

3 In this study, we use five irregular data sets of neighbor relationships and
 4 four regular sets (spatial weights). The irregular sets are: queen contiguities
 5 between 2,478 Polish gminy (NUTS 5 local government districts); queen con-
 6 tiguities between 3,489 US counties (including Alaska and Hawaii); queen con-
 7 tiguities between cells on a 1° grid for world land areas omitting Antarctica
 8 with 15,260 observations; queen contiguities between 32,698 US 2000 Cen-
 9 sus Zip Code Tabulation Areas (ZCTA, omitting Alaska and Hawaii); and
 10 queen contiguities between 64,878 US census tracts in 2000 (omitting Alaska
 11 and Hawaii). These five sets of spatial weights are row-standardized. The
 12 regular sets are two variants each of 50×50 ($n = 2,500$) and $1,000 \times 1,000$
 13 ($n = 1,000,000$) grids, the variants being rook and queen contiguities (binary
 14 spatial weights only). By definition, rook and queen contiguities produce sym-
 15 metric weights, because if observation i is a neighbor of j , then j must also be
 16 a neighbor of i . In addition, use is made of a row-standardized $1,000 \times 1,000$
 17 rook contiguity data set to compare with interpolated approximations to the
 18 Jacobian.

Table 1: Summary graph characteristics for the five irregular data sets.

	Gminy	US Counties	World grid	US ZCTA	US census tracts
Subgraphs	1	219	49	243	30
Non-empty subgraphs	1	12	42	14	13
Cyclical graphs	0	9	16	10	4
Singleton subgraphs	0	207	7	229	17
Max. neighbors	18	14	8	75	30
Largest subgraph	2478	3153	8998	32419	64692
N	2478	3489	15258	32698	64878

19 Table 1 shows that only the Polish gminy data set has a single connected
 20 graph; the other irregular data sets have multiple disjoint connected subgraphs,
 21 including singleton subgraphs, that is observations with no neighbors. The
 22 regular data sets all have single subgraphs, and with the Polish gminy data set
 23 constitute simple undirected graphs. For the remaining four irregular data sets,
 24 the subgraphs may be represented as blocks of spatial weights lying along the
 25 diagonal of the weights matrix without influence on each other. By definition,
 26 all the two-node subgraphs are cyclical, and some other subgraphs meet the

- 1 same condition, that is that, for every location, no pair of its neighbours are
- 2 connected (Smirnov and Anselin, 2009, pp. 2984–2985).

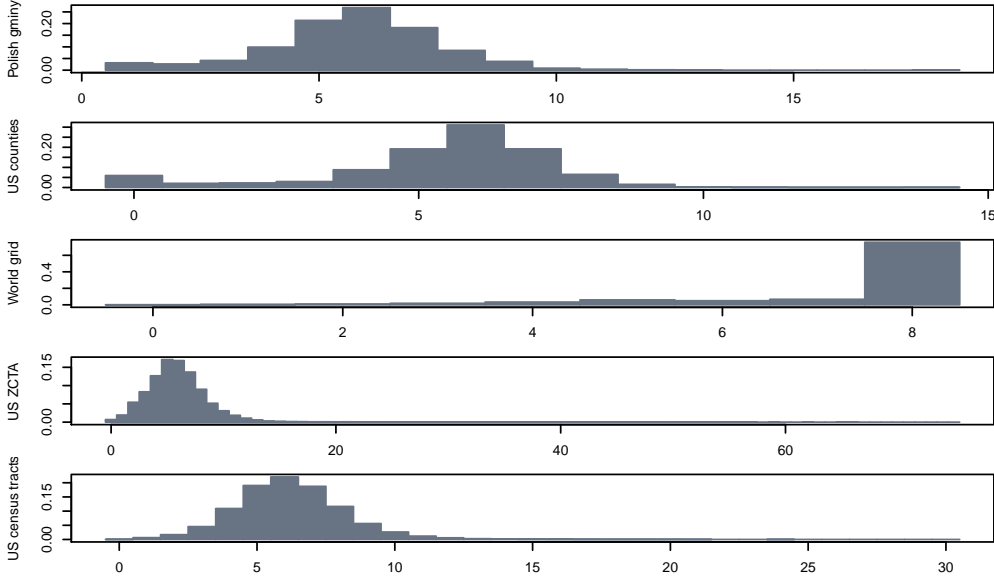


Figure 1: Distributions of numbers of neighbors for the five irregular data sets.

3 Figure 1 shows the distributions of contiguous neighbors for the five irreg-
 4 ular data sets. It is worth observing that the examples represent distributions
 5 with different levels of skewness, with the world grid data set being right-
 6 skewed, having many spatial entities with eight neighbors; the other data sets
 7 are left-skewed. The bottom row of Table 1 lists the maximum number of
 8 neighbors by data set. The very large numbers of contiguous neighbors for
 9 some observations in the two latter data sets occur because of the by the way
 10 these observations are structured, for example central entities with radiating
 11 neighbors, such as Central Park in New York (the entity in the US census tract
 12 data set with 30 neighbors). In the ZCTA data set, all the entities with over
 13 26 neighbors are three-digit codes with either XX or HH suffixes, indicating
 14 parks, forest lands etc., or water bodies. The selection of irregular data sets
 15 is similar to those chosen by Smirnov and Anselin (2001).

16 In addition, we have generated intrinsically asymmetric neighbors, taking
 17 the 6 nearest neighbors using Great Circle distances from geographical co-
 18 ordinates for each of the five irregular data sets, using polygon centroids to
 19 approximate point support observations. The spatial weights in these schemes
 20 are row-standardized. We use these to explore the consequences of weights
 21 asymmetry in computing the Jacobian.

2 Eigenvalue methods

The first published versions of the eigenvalue method for finding the Jacobian (Ord, 1975, p. 121) present it in product form:

$$\ln(|\mathbf{I} - \lambda\mathbf{W}|) = \ln\left(\prod_{i=1}^n (1 - \lambda\zeta_i)\right) \quad (7)$$

instead of the equivalent summation form:

$$\ln(|\mathbf{I} - \lambda\mathbf{W}|) = \sum_{i=1}^n \ln(1 - \lambda\zeta_i) \quad (8)$$

where ζ_i are the eigenvalues of \mathbf{W} . In the product form, it may become difficult to compute, because the value of the determinant may underflow (become indistinguishable from zero before taking the logarithm) if care is not shown.

2.1 Computing eigenvalues

One specific problem addressed by Ord (1975, p. 125) is that of the eigenvalues of the asymmetric row-standardized matrix \mathbf{W} with underlying symmetric neighbor relations $c_{ij} = c_{ji}$. If we calculate the row sums of weights by: $\mathbf{C}\mathbf{1} = \mathbf{w} = [w_j]$, where $\mathbf{1}$ is a vector of ones, we can get: $\mathbf{W} = \mathbf{C}\mathbf{D}$, where $\mathbf{D} = \text{diag}(1/w_j)$. By similarity, the eigenvalues of \mathbf{W} are equal to those of: $\mathbf{D}^{\frac{1}{2}}\mathbf{C}\mathbf{D}^{\frac{1}{2}}$ (see also Griffith and Layne, 1999, pp. 128–130). Of course, if the underlying neighbor relations are not symmetric, the eigenvalues of \mathbf{W} are not necessarily real; the consequences of using such asymmetric weights matrices are explored below. The handling of intrinsically asymmetric weights matrices is also discussed by LeSage and Pace (2009, pp. 88–89).

In addition to choices with regard to the underlying neighbor relations used to structure covariance between observations, by no means all applications use row standardization of spatial weights matrices. Row standardization has the convenient consequence that the largest eigenvalue of \mathbf{W} is known to be equal to one by design. However, the value of the smallest eigenvalue is unknown, but in line search for the spatial coefficient λ , the relevant interval is often taken as $[-1, 1)$, as positive spatial autocorrelation is assumed to be more common, so that inaccuracy on the lower bound is probably less important.

Row standardization upweights neighbor relations for observations with few neighbors, and downweights relations for those with many neighbors. Tiefelsdorf et al. (1999) propose a variance-stabilising scheme instead of row standardization, which for underlying symmetric neighbor relations also yields an asymmetric spatial weights matrix that is similar to symmetric.

Table 2: Lower and upper interval bounds on λ : $1/\zeta_{\min}, 1/\zeta_{\max}$, 2,478 Polish gminy queen contiguity data set, 50×50 rook contiguity regular grid, and 50×50 queen contiguity regular grid, for binary weights (B), binary weights scaled to sum to n (C), variance-stabilizing weights (S) — real part only, row standardized weights (W) — real part only, and variance-stabilizing weights (S (sym)) and row standardized weights (W (sym)) transformed to symmetry by similarity.

	B	C	S	W	S (sym)	W (sym)
Gminy lower	-0.2586	-1.4835	-1.8462	-1.2199	-1.8462	-1.2199
Gminy upper	0.1522	0.8733	0.9393	1.0000	0.9393	1.0000
50x50 rook lower	-0.2505	-0.9819	-0.9910	-1.0000	-0.9910	-1.0000
50x50 rook upper	0.2505	0.9819	0.9910	1.0000	0.9910	1.0000
50x50 queen lower	-0.2510	-1.9478	-1.9733	-1.9034	-1.9733	-1.9034
50x50 queen upper	0.1254	0.9730	0.9858	1.0000	0.9858	1.0000

1 Many disciplines using spatial regression methods simply use unstandard-
2 ized neighbor relations matrices which may or may not be symmetric. Table
3 2 shows the lower and upper bounds for λ for the same set of symmetric con-
4 iguous neighbors for 2,478 Polish gminy, and the 50×50 regular grid with
5 rook and queen contiguities, under different weights representations. The un-
6 derlying eigenvalues have been calculated using the R `eigen` function, using
7 the standard LAPACK functions and with symmetry of the input matrix de-
8 termined by the internal code. As can be seen, the intervals vary greatly,
9 depending on choices of specification, and that the lower bound on the spatial
10 coefficient λ in the row-standardized case is less than -1 for the Gminy and
11 50×50 queen contiguity data sets.

12 When the weights matrix may be re-ordered into subgraph blocks on the
13 diagonal, and a final set of all-zero rows and columns, corresponding to no-
14 neighbor observations, the numbers of eigenvalues equal to zero and one change
15 from the single subgraph case.³ It should be noted that in many practical
16 examples, the appearance of subgraph blocks is not expected, resulting from
17 boundary corruption affecting the determination of contiguities; in such cases,
18 the boundary topologies should be cleaned before proceeding. In the case of
19 the row-standardized Polish gminy data set, the largest eigenvalue is 1, and
20 there are two eigenvalues with an absolute value less than $1e - 14$. For the
21 row-standardized US counties data set, with 219 subgraphs of which 207 are
22 singleton, there are 12 eigenvalues with values of 1,⁴ that is, one eigenvalue of

³We are grateful to the Editor for drawing our attention to this issue.

⁴In fact greater than $1 - (1e - 14)$.

1 1 per non-singleton subgraph. If we compute the eigenvalues of the subsetted
2 blocks separately and add in 207 zeros, the resulting vector is equal to the
3 eigenvalues of the whole data set to a tolerance of $1e-14$. There are in total
4 225 zero eigenvalues, of which 207 are in the singleton subgraphs, none in the
5 8 subgraphs with 2 nodes, 1 in a 7-node subgraph, 5 in a 30-node subgraph,
6 none in a 76-node subgraph, and the remaining 12 in the largest 3,153-node
7 subgraph. Of the 12 non-empty subgraphs, corresponding to the 12 eigenvalues
8 with values of 1, 9 (of which 8 are two-node subgraphs) are cyclical, and
9 correspond to 9 eigenvalues with values of -1 (Smirnov and Anselin, 2009, pp.
10 2984–2985).

11 Where subgraphs appear in which there are larger numbers of spatial enti-
12 ties, computation of eigenvalues may be split between blocks as noted above;
13 this may make it possible to compute the eigenvalues of spatial weights ma-
14 trices that would otherwise be intractable. This has been done here with the
15 world raster grid data set, in which the largest subgraph has 8998 entities,
16 which, although large, is not too large for the eigenvalues to be computed
17 on a standard 64-bit computer. A discussion of subgraph blocks is also pro-
18 vided by Smirnov and Anselin (2009, pp. 2984–2985). Some practical conse-
19 quences of using spatial weights with no-neighbor observations are discussed
20 by Bivand and Portnov (2004), but detailed analysis of the subgraph problem
21 for eigenvalues of spatial weights matrices does not yet appear to have been
22 undertaken.

23 One point that needs to be taken forward from this discussion is that al-
24 though, for simultaneous autoregressive models, neither \mathbf{W} nor $(\mathbf{I} - \lambda\mathbf{W})$ are
25 required to be symmetric positive definite matrices, such an assumption may
26 make computing the Jacobian more feasible. There are obvious limits on n ,
27 because in general dense matrices have to be used to find the eigenvalues of \mathbf{W} ,
28 which impact both the use of eigenvalues in computing the Jacobian and in
29 setting the search interval for λ . Because the line search interval for λ can be
30 manipulated, so far some attention has been given to finding the extreme eigen-
31 values of sparse \mathbf{W} for large n by Griffith (2004a) and Griffith and Luhanga
32 (2011). In addition, it may be noted that the Spatial Econometrics toolbox
33 uses the MatlabTM `eigs` function, based on ARPACK code for finding a few
34 eigenvalues of large sparse matrices, here the minimum and maximum values.

35 A consequence of this discussion is that implementation is of the essence,
36 something that we feel is demonstrated by Walde et al. (2008)⁵. They un-
37 dertook 3,000 Monte Carlo runs pitting different fitting methods against each
38 other. In fact, all the fitting methods except one are maximum likelihood with

⁵We are grateful to Janette Walde for her willingness to clarify questions arising during our study, and for sharing code excerpts with us.

1 differing methods for computing the Jacobian. The simulation scenario is for
2 a regular 4,900 observation grid, a simultaneous autoregressive process with a
3 λ coefficient value of 0.5, an intercept of one and a uniform random x variable
4 within zero and one and a coefficient of one; the remaining error is assumed to
5 be normal with zero mean and a standard deviation of one. The final fitting
6 method is generalized method of moments, which we do not address here.

7 By comparing methods for computing the Jacobian inside model fitting
8 simulation runs, it is not possible to see how well or poorly the actual Ja-
9 cobian values are computed for various values of λ , but rather how well the
10 optimization technique performs in providing the Jacobian method in the func-
11 tion returning the log-likelihood with the proposed λ values. This raises doubt
12 as to whether the optimization technique gives all methods of computing the
13 Jacobian a fair chance.

14 With regard to the Ord eigenvalue method for computing the Jacobian,
15 Walde et al. (2008, p. 158) conclude that it fails dramatically for the $n = 4,900$
16 grid used in their simulation, when in principle for a fixed \mathbf{W} , the eigenval-
17 ues are also fixed, and consequently any variation in their Monte Carlo runs
18 cannot be coming from this source. Their numerical results suggest that the
19 line search for λ often halted at its lower bound, naturally leading to poor per-
20 formance. Curiously, they do find that the characteristic polynomial method
21 of Smirnov and Anselin (2001) performs excellently, but in their implementa-
22 tion, this method computes the characteristic coefficients from the eigenvalues
23 of a small number of dense block matrices constructed by limited divide-and
24 conquer. Consequently, the difference between the outcomes for the eigenvalue
25 and characteristic polynomial methods which in effect use the same eigenvalues
26 to find the Jacobian is paradoxical.

27 **2.2 Complexity of eigenvalues**

28 The analyst may find that underlying symmetric spatial neighbor relationships
29 are sufficient, such as polygon contiguities or some graph neighbor schemes,
30 such as triangulation, which are also planar. Distance schemes are usually
31 symmetric by design, but may not be planar graphs. However, should the
32 analyst wish to specify the spatial relationships in an intrinsically asymmet-
33 ric way, using k-nearest neighbors, some graph measure, or some distance
34 measure (one-way street) relationships, model fitting methods should accom-
35 modate these choices. In addition, intrinsically asymmetric weights may be
36 constructed by weighting symmetric links differently by direction, as in the
37 use of trade flows as a weighting variable.

38 So far, relatively little attention has been given to the consequences of try-
39 ing to fit spatial regression models with asymmetric spatial weights. LeSage and Pace

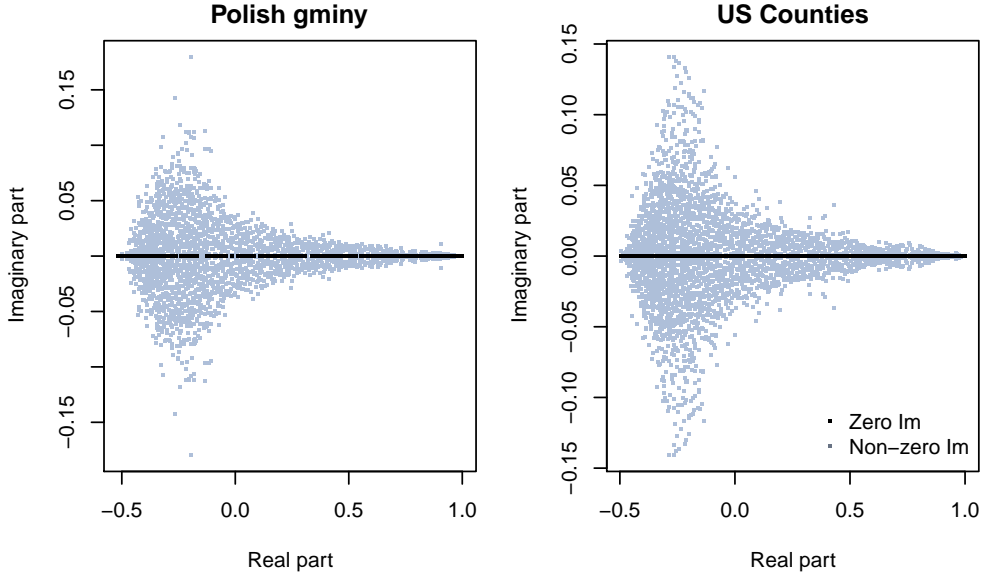


Figure 2: Plots of the real and imaginary parts of the eigenvalues of the row-standardized 6 nearest neighbor weights for the Polish gminy and US counties data sets.

1 (2009, pp. 89) analysed the determinant domain of matrix:

$$\mathbf{Z} = \mathbf{I} - \lambda \mathbf{W} \quad (9)$$

2 in the case in which matrix \mathbf{W} is not similar to a symmetric matrix. Using
 3 similar arguments, we would like to analyze the value of the log-determinant of
 4 matrix \mathbf{Z} . Such a matrix may have complex eigenvalues, but an advantage in
 5 this case is that the complex eigenvalues come in conjugate pairs (see Bernstein,
 6 2009, p. 263). So, the number of complex eigenvalues is even. Let us denote
 7 it by $2k$. Further, let us use the following notation (using j as the index to
 8 avoid confusion with the imaginary unit): $\omega_j = a_j + ib_j$ and $\omega_j^* = a_j - ib_j$ for
 9 values of complex eigenvalues in pairs (i is an imaginary unit) and ζ_j for real
 10 eigenvalues of \mathbf{W} . Then:

$$\ln(|\mathbf{Z}|) = L_1 + L_2, \quad (10)$$

11 where:

$$L_1 = \sum_{j=1}^k \ln[(1 - \lambda \omega_j)(1 - \lambda \omega_j^*)] = \sum_{j=1}^k \ln[(1 - \lambda a_j)^2 + (\lambda b_j)^2], \quad (11)$$

1 and:

$$L_2 = \sum_{j=k+1}^{n-k} \ln(1 - \lambda \zeta_j). \quad (12)$$

2 $\exp(L_1)$ is a real positive quantity, which together with (5) implies that $|\mathbf{Z}|$
3 is also a real quantity independently of the number of complex eigenvalues.
4 But the number of complex eigenvalues of the \mathbf{W} and their scale as well as the
5 magnitude of λ affect the scale of values of $|\mathbf{Z}|$. From the localization theorem
6 of Gerschgorin (see Varga, 2009, Theorem 1.11), we know that the absolute
7 values of imaginary parts of eigenvalues of \mathbf{Z} are contained in the interval $[0, \lambda]$.
8 Let us observe that in the case of complex eigenvalues having small imaginary
9 parts, omitting them in calculation does not have a big influence on the value
10 of $\ln|\mathbf{Z}|$. It is worth noting that for row-standardized version of \mathbf{W} (that is, \mathbf{W}
11 is row-stochastic) the spectral radius of \mathbf{W} is equal to 1, which implies that:

$$|\omega_j| \leq 1, \quad j = 1, \dots, k, \quad (13)$$

12 and

$$|\zeta_j| \leq 1, \quad j = k+1, \dots, n-k. \quad (14)$$

13 and $\max_j \zeta_j = 1$ (see Bernstein, 2009, p. 298 – Perron Frobenius Theorem).

14 Figure 2 shows plots of the real and imaginary parts of the eigenvalues
15 of the row-standardized weights for both the Polish and the US data sets;
16 real eigenvalues (with zero imaginary part) are plotted as black points, while
17 complex eigenvalues are gray. It is easy to see how the conjugate pairs of
18 complex eigenvalues spread out above and below the real eigenvalues. It is
19 clearly important to ensure that the Jacobian is computed using logarithms
20 of complex numbers in Equation (8), otherwise terms in b_j in Equation (11)
21 would be omitted; the `clog` function was only added to the ISO C standard in
22 1999, also for double-precision numbers, so earlier implementations depended
23 on platform-specific functions. Fortran 77 had a complex logarithm function
24 for single precision complex numbers.

25 **2.3 Analytical eigenvalues**

26 In this discussion of the use of eigenvalues in computing the Jacobian, we have
27 not considered the accuracy of the method. We can approach this question for
28 a small but important subset of planar tessellations, regular rectangular grids.
29 Analytical eigenvalues for binary spatial weights matrices of rook contiguities

1 are given by Ord (1975, p. 126), and discussed further by Griffith and Sone
 2 (1995, p. 169). The eigenvalues are:

$$\zeta_{pq} = 2 \cos\left(\frac{p\pi}{P+1}\right) + 2 \cos\left(\frac{q\pi}{Q+1}\right) \quad (15)$$

3 for $p = 1; \dots, P$ and $q = 1, \dots, Q$ for a regular grid with P and Q rows and
 4 columns. A development for queen contiguities on the same grid is given by
 5 Griffith and Sone (1995, p. 170), see also Gasim (1988):

$$\zeta_{pq} = 2 \cos\left(\frac{p\pi}{P+1}\right) + 2 \cos\left(\frac{q\pi}{Q+1}\right) + 4 \cos\left(\frac{p\pi}{P+1}\right) \times \cos\left(\frac{q\pi}{Q+1}\right) \quad (16)$$

Table 3: Summaries of componentwise absolute differences between Jacobians calculated using analytical and computed eigenvalues for the 50×50 regular grid.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Rook	0.00e+00	8.88e-15	4.26e-14	1.09e-13	1.42e-13	8.53e-13
Queen	0.00e+00	0.00e+00	3.55e-15	1.78e-14	1.42e-14	2.27e-13

6 The analytical eigenvalues involve trigonometrical functions, but certainly
 7 fewer operations than the computed eigenvalues, here output by LAPACK rou-
 8 tine DSYEVR with an absolute tolerance of zero. Table 3 shows summaries
 9 of componentwise absolute differences between Jacobians calculated using an-
 10 alytical and computed eigenvalues for the 50×50 regular grid, for rook and
 11 queen contiguities. In the rook case, the spatial coefficient λ takes the values
 12 in the sequence from -0.24 to 0.24 in steps of 0.01 , 49 values in all, while for
 13 the queen contiguities, the coefficient takes values from -0.24 to 0.12 in steps
 14 of 0.01 , 37 values in all, reflecting the different feasible ranges of λ for these
 15 two schemes. Computing the eigenvalues took over two orders of magnitude
 16 longer than using the analytical methods for this grid size; given the eigenval-
 17 ues, the calculation of the Jacobians takes the same time for both methods.
 18 The maximum componentwise absolute differences are very small, and give us
 19 a baseline for subsequent methods for this regular grid, and for the much larger
 20 $1,000 \times 1,000$ regular grid to be used below. In this large n case, calculating
 21 the analytical eigenvalues took under $0.2s$, and each Jacobian less than $0.1s$.

3 Sparse Matrix methods

When spatial regression models began to be taken up in applied research, hardware constraints on computing eigenvalues for moderate n prompted work on alternative methods for computing the Jacobian. In a series of contributions, Pace and Barry (1997b,c,d) show that sparse matrix methods can be used to find the log-determinant directly.⁶ The methods of choice are the Cholesky factorization of a sparse, symmetric, positive-definite matrix, and the LU factorization if symmetry requirements on the matrix need to be relaxed. Walde et al. (2008) find, by implication, that the Jacobian values from Cholesky factorization and the LU factorization for the same $(\mathbf{I} - \lambda\mathbf{W})$ matrix differ.⁷ This finding is incompatible with theoretical and practical results from numerical analysis. Naturally, for the same symmetric, positive-definite matrix, one would expect the log-determinants based on the Cholesky factorization and the LU factorization to be identical within machine precision (Higham, 2002, pp. 196).

3.1 Sparse Cholesky and LU methods

Rue (2001) utilizes Cholesky factorization extensively in computations on Gaussian Markov random fields, and reports no reservations concerning the choice of the method. The accuracy and stability of Gaussian elimination has been the subject of detailed study since the advent of numerical computation. As LU factorization is a form of Gaussian elimination, its possible weaknesses are well-known, and it is regarded as a highly reliable way of solving linear systems when the matrix to be factorized is nonsingular (Higham, 2002, pp. 160–166). In a recent popular review discussing the use of Cholesky factorization, Higham (2009) states that: “Rounding error analysis shows that Cholesky factorization has excellent numerical stability properties.” If the matrix to be factorized is symmetric positive definite, the Cholesky factorization can be obtained from the LU factorization, or computed directly at half the cost of the LU factorization (Higham, 2002, pp. 196–201). While it appears that we may be able to rely on the theoretical results of the analysis of numerical algorithms with regard to the accuracy and stability of Cholesky and LU methods, it is prudent to compare the Jacobians computed in this way with those calculated using eigenvalues.

The log determinant of symmetric positive definite $\mathbf{I} - \lambda\mathbf{W}$ for Cholesky factorization may be expressed as:

⁶The S-PLUS SpatialStats module also uses sparse matrix methods (Kaluzny et al., 1998).

⁷Correspondence with Janette Walde, who made code extracts available, indicates that the Cholesky Jacobian was erroneously divided by 2, explaining the discrepancy.

$$\ln(|\mathbf{I} - \lambda\mathbf{W}|) = 2 \sum_{i=1}^n \ln l_{ii} \quad (17)$$

1 where l_{ii} are the elements of the diagonal of \mathbf{L} , given by solving $(\mathbf{I} - \lambda\mathbf{W}) = \mathbf{L}\mathbf{L}'$
2 (Walde et al., 2008, p. 154). Note that the spatial weights matrix used here
3 in the case of row-standardized weights is the matrix $\mathbf{D}^{\frac{1}{2}}\mathbf{C}\mathbf{D}^{\frac{1}{2}}$ defined above in
4 Section 2.1. In the **Matrix** and **spam** implementations, the log determinant is
5 extracted directly, avoiding the need to take logarithms following factorization.
6 In the LU case for a nonsingular matrix, the log determinant is:

$$\ln(|\mathbf{I} - \lambda\mathbf{W}|) = \sum_{i=1}^n \ln |u_{ii}| \quad (18)$$

7 where u_{ii} are the elements of the diagonal of \mathbf{U} from $(\mathbf{I} - \lambda\mathbf{W}) = \mathbf{L}\mathbf{U}$, where \mathbf{U}
8 is an upper triangular matrix.

Table 4: Summaries of componentwise absolute differences between Jacobians calculated using analytical eigenvalues and sparse Cholesky and sparse LU factorization for the $1,000 \times 1,000$ rook weights.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Matrix Cholesky	0.00e+00	9.77e-09	5.01e-08	9.53e-08	1.10e-07	5.14e-07
spam Cholesky	0.00e+00	2.91e-11	7.64e-11	8.67e-11	1.09e-10	3.78e-10
Matlab Cholesky	0.00e+00	9.78e-09	5.01e-08	9.54e-08	1.10e-07	5.14e-07
Matrix LU	0.00e+00	8.64e-12	3.00e-11	4.56e-11	6.18e-11	1.96e-10
Matlab LU	0.00e+00	6.04e-09	2.27e-08	1.03e-07	1.16e-07	5.58e-07

9 Sparse Cholesky and LU factorization implementations are provided in
10 MatlabTM and in the **Matrix** package in R (Bates and Maechler, 2012), with an
11 additional sparse Cholesky factorization implementation in the **spam** package
12 in R as described by Furrer and Sain (2010) using methods due to Ng and Peyton
13 (1993). It appears that MatlabTM and **Matrix** both use implementations due to
14 Davis (2006), at least for Cholesky factorisation. Following up the helpful sug-
15 gession of an anonymous referee to carry out a “useful impartial benchmark”,
16 we compare Cholesky and LU factorization Jacobians with those calculated us-
17 ing analytical eigenvalues for large $1,000 \times 1,000$ regular grids. Table 4 shows
18 summaries of the componentwise absolute differences between Jacobians cal-
19 culated using analytical eigenvalues (Equation 15) and using five sparse matrix
20 factorizations. The five are **Matrix** and MatlabTM Cholesky methods — which
21 are effectively identical, as one would expect from their common code base —
22 and **spam** Cholesky, in addition to **Matrix** and MatlabTM LU methods. The

1 maximum componentwise absolute differences are very small, and are all sev-
 2 eral orders of magnitude less than the default threshold for line search in opti-
 3 mizing the log-likelihood function. Table 5 repeats the benchmark for the large
 4 queen contiguity grid, reaching the same conclusion that the sparse Cholesky
 5 and LU factorizations available in MatlabTM and R packages typically reach the
 6 same function optimum as Jacobians calculated using eigenvalues (Equation
 7 16).

Table 5: Summaries of componentwise absolute differences between Jacobians calculated using analytical eigenvalues and sparse Cholesky and sparse LU factorization for the $1,000 \times 1,000$ queen weights.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Matrix Cholesky	0.00e+00	1.37e-08	3.06e-08	1.09e-07	1.64e-07	5.27e-07
spam Cholesky	0.00e+00	4.46e-11	8.46e-11	1.10e-10	1.27e-10	6.69e-10
Matlab Cholesky	0.00e+00	1.37e-08	3.07e-08	1.09e-07	1.64e-07	5.27e-07
Matrix LU	0.00e+00	2.18e-11	3.71e-11	5.99e-11	7.28e-11	3.20e-10
Matlab LU	0.00e+00	9.52e-09	2.67e-08	5.61e-08	7.78e-08	2.45e-07

Table 6: Maximum componentwise absolute differences between eigenvalue-based Jacobians and five sparse Cholesky versions (R **Matrix** simplicial and supernodal decompositions and decomposition chosen by a CHOLMOD-internal heuristic, R **spam** using pivoting schemes MMD and RCM) and a sparse LU factorization (R **Matrix**), for five symmetric spatial weights matrices.

	50×50 rook	50×50 queen	Polish gminy	US Counties	World grid
Matrix simplicial	1.65e-12	3.01e-12	1.42e-12	2.39e-12	4.84e-11
Matrix supernodal	1.76e-12	2.96e-12	1.53e-12	2.05e-12	4.71e-11
Matrix CHOLMOD	1.65e-12	3.01e-12	1.42e-12	2.39e-12	4.84e-11
spam MMD	9.09e-13	3.69e-13	9.95e-14	2.27e-13	2.05e-12
spam RCM	9.09e-13	2.34e-13	1.14e-13	2.27e-13	2.27e-12
Matrix LU	1.02e-12	3.41e-13	5.68e-14	1.14e-13	7.08e-13

8 For completeness, we also examined the componentwise absolute differences
 9 between sets of Jacobian values for λ in the range $[-0.9, 0.99]$, in steps of 0.01,
 10 in total 190 values for row-standardized spatial weights matrices for the Pol-
 11 ish gminy, the US counties and the world raster grid data sets. To this we
 12 added the binary rook and queen contiguity 50×50 regular grids. The imple-
 13 mentations of sparse Cholesky factorization in the **Matrix** and **spam** packages
 14 are independent of each other, with the former using approximate minimal

1 degree (AMD) ordering, and the latter multiple minimum degree (MMD, de-
 2 fault) or reverse Cuthill-McKee (RCM) pivoting; choice of pivoting methods
 3 is discussed in Pace and Barry (1997a) and LeSage and Pace (2009, p. 87).
 4 The implementation in the **Matrix** package provides simplicial or supernodal
 5 decomposition, which can be specified directly. In addition, a heuristic is pro-
 6 vided in the CHOLMOD code used by **Matrix**, which chooses the preferred
 7 decomposition method automatically (here termed CHOLMOD).

8 Table 6 confirms that the maximum componentwise absolute differences
 9 between the Jacobians computed using the various Cholesky and LU factor-
 10 izations, and those calculated using computed eigenvalues are extremely small,
 11 despite the differences in graph characteristics between the data sets.

Table 7: Summaries of componentwise absolute differences between Jacobians
 calculated using computed eigenvalues and LU factorization for the two smaller
 asymmetric 6 nearest neighbor weights sets.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Polish gminy	0.00e+00	5.51e-14	1.56e-13	3.28e-13	3.41e-13	6.20e-12
US Counties	0.00e+00	5.78e-14	1.62e-13	3.13e-13	4.99e-13	2.96e-12

12 The same conclusion may be drawn from the summaries of componentwise
 13 absolute differences, shown in Table 7, between Jacobians based on LU factor-
 14 izations for the six nearest neighbor spatial weights for the Polish gminy and
 15 the US counties data sets, and those based on computed eigenvalues.

16 In the implementation of the sparse LU method in the Spatial Economet-
 17 rics toolbox for MatlabTM, use is made of a pre-computed column approximate
 18 minimum degree permutation. The assumption is that the fill-reducing per-
 19 mutation will be invariant in λ , and that in successive computations of the
 20 Jacobian, time can be saved by permuting $(\mathbf{I} - \lambda\mathbf{W})$ using the saved permuta-
 21 tion before sparse LU factorization. The savings in MatlabTM are of the order
 22 of 20%, with somewhat smaller savings seen in R with both row and column
 23 permutation.

24 3.2 Updating Cholesky factorizations

25 A promising innovation for reducing the computational burden of computing
 26 the Jacobian in spatial regression models when the spatial weights are symmet-
 27 ric or similar to symmetric was introduced in the **Matrix** package in March 2008
 28 (Bates and Maechler, 2012), based on Davis and Hager (1999), Davis (2005)
 29 and described in Davis (2006, pp. 63–66). A comparable facility was intro-
 30 duced into the **spam** package in June 2008 (Furrer and Sain, 2010). Because

1 the pattern of sparseness in the matrix for which the log-determinant is to
 2 be found does not change, it is possible to carry out the Cholesky factoriza-
 3 tion once, and then update the values respecting the fill-reducing permutation
 4 found when the factorization was first undertaken. This incurs a moderate
 5 set-up cost, but speeds up the finding of each Jacobian value for successive λ
 6 proposed by the optimizer.

7 In Equation (17) above, the Jacobian was shown to be taken as the sum
 8 of logarithms of the diagonal of the Cholesky factorization. If we replace the
 9 \mathbf{LL}' factorization with \mathbf{LDL}' , where \mathbf{D} is a diagonal matrix, we see that we are
 10 really only interested in this part of the factorization, and is indeed analogous
 11 to the eigenvalue procedure, where only the diagonal matrix of eigenvalues is
 12 used.⁸ If we note that the Jacobian values are required for many values of λ
 13 but fixed \mathbf{W} , we can re-express the log determinant, changing the sign of λ for
 14 convenience, as:

$$\ln(|\mathbf{I} + \lambda\mathbf{W}|) = n \ln(\lambda) + 2 \ln \left(\left| \mathbf{W} + \frac{1}{\lambda} \mathbf{I} \right| \right) \quad (19)$$

15 The updating procedure uses the Cholesky factorizations of \mathbf{W} and $-\mathbf{W}$
 16 computed once only, and depending on the value of λ , we either return zero
 17 for λ within machine precision of zero, or switch on the sign of λ . The `update`
 18 method for objects returned by the `Cholesky` method of the `Matrix` package
 19 utilizes the pre-computed factorization of $-\mathbf{W}$, and takes as additional argu-
 20 ments the original spatial weights matrix $-\mathbf{W}$ and $1/\lambda$ for positive λ , with
 21 sign switches for negative λ . The determinant method used returns the loga-
 22 rithm of the modulus of the determinant. A non-exported function in `Matrix`,
 23 `ldetL2up`, vectorizes this process for a vector of values of λ .

24 In the `spam` case, the Cholesky factorization is updated by starting with an
 25 initial $(\mathbf{I} - \lambda\mathbf{W}) = \mathbf{LL}'$ and a seed value for λ . The update is computed using
 26 \mathbf{L} and the current sparse matrix $(\mathbf{I} - \lambda\mathbf{W})$ for the value of λ proposed by the
 27 optimizer. Furrer and Sain (2010, p. 10) describe the approach used, which
 28 uses methods due to Ng and Peyton (1993).

29 Table 8 shows that updating the Cholesky factorization using functions in
 30 `Matrix` and `spam` yields maximum componentwise absolute differences to the
 31 eigenvalue-based Jacobians that are very small. Those for `spam` using multi-
 32 ple minimum degree ordering are the same as those where updating was not
 33 used, reported in Table 6, while those for `Matrix` are slightly less accurate
 34 than their non-updating equivalents. Naturally, the use of Cholesky and up-
 35 dated Cholesky factorizations is limited to symmetric and similar to symmetric

⁸This discussion is based with permission on email exchanges with Douglas Bates and Martin Mächler.

Table 8: Maximum componentwise absolute differences between eigenvalue-based Jacobians and two updating sparse Cholesky versions (R **Matrix** factorization chosen by a CHOLMOD-internal heuristic, and R **spam** using the MMD pivoting scheme MMD), for five symmetric spatial weights matrices.

	50 × 50 rook	50 × 50 queen	Polish gminy	US Counties	World grid
Matrix CHOLMOD	2.01e-10	1.07e-10	1.21e-10	2.92e-10	3.37e-09
spam MMD	9.09e-13	3.69e-13	9.95e-14	2.27e-13	2.05e-12

1 spatial weights matrices, that is, those with real eigenvalues.

2 4 Approximations

3 In addition to the methods for computing the Jacobian presented above, many
 4 approximations have been proposed over the last twenty years, with trace-
 5 based methods going back to Martin (1993). Walde et al. (2008) try out a
 6 number of approximations to the Jacobian, of which the Monte Carlo and
 7 Chebyshev approximations are presented here, using sparse matrix operations
 8 from the **Matrix** package throughout. They also implement the characteristic
 9 polynomial approach due to Smirnov and Anselin (2001) and used in GeoDa
 10 and in OpenGeoDa.⁹

11 4.1 Interpolation

12 Interestingly, Barry and Pace (1999) preface their Monte Carlo approximation
 13 method, covered below, by quoting extensively from Griffith and Sone (1995),
 14 who motivate the need for approximating the Jacobian in larger data sets. In
 15 a sequence of papers, Griffith and co-authors show that approximations may
 16 be available for some spatial configurations of observations. Griffith and Sone
 17 (1995) begin by describing analytical ways of calculating the eigenvalues of
 18 a regular square surface partitioning, following Ord (1975), with extensions.
 19 Further developments are based on approximating equations requiring knowl-
 20 edge of the largest and smallest eigenvalues of the weights matrix. In Griffith
 21 (2000), work is continued for regular square and hexagonal tessellations, and
 22 extended in Griffith (2004a) to the approximation of the principal eigenvalues

⁹<http://geodacenter.asu.edu/ogeoda>. The code for the OpenGeoDa implementa-
 tion was released in late 2011, and shows that a re-implementation in R, or even a linking
 of this implementation, with its custom sparse matrix code, to R would be challenging;
<http://code.google.com/p/opengeoda/source/browse/trunk/Regression/polym.h>.

1 of the spatial weights matrix, themselves needed to approximate the Jaco-
 2 bian. This is advanced by Griffith and Luhanga (2011) with further results
 3 on the approximation of the principal eigenvalues for adjacency matrices of
 4 connected, undirected planar graphs in the context of inertia, the numbers of
 5 positive, negative, and zero eigenvalues. Finally, Griffith (2004b) provides ap-
 6 proximations to terms required in the characteristic polynomial approach due
 7 to Smirnov and Anselin (2001).

8 Here we only examine the approximation to the Jacobian given for a row-
 9 standardized rook contiguity spatial weights for a $P \times Q$ regular grid by Griffith
 10 (2004b, pp. 857–859), who provides the following relationship calibrated using
 11 nonlinear regression:

$$\begin{aligned}\hat{q}_2 &= 0.11735 + 0.10091 \left(\frac{1}{P^{5/4}} + \frac{1}{Q^{5/4}} \right) + \frac{0.42844}{PQ} \\ \hat{q}_4 &= 0.07421 + 0.05730 \left(\frac{1}{P^{2/3}} + \frac{1}{Q^{2/3}} \right) + \frac{0.66001}{PQ} \\ \hat{q}_{20} &= 0.05521 + 0.52467 \left(\frac{1}{P^{7/4}} + \frac{1}{Q^{7/4}} \right) + \frac{2.48015}{PQ} \\ \ln(|\mathbf{I} - \lambda \mathbf{W}|) &= -\ln(1 + \hat{q}_2(\lambda^2) + \hat{q}_4(\lambda^4) + \hat{q}_{20}(\lambda^{20}))PQ\end{aligned}\quad (20)$$

Table 9: Summaries of componentwise differences between interpolated and Cholesky factorization Jacobians for a row-standardized rook 1000×1000 data set.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Absolute	0.00e+00	1.46e+02	3.36e+02	4.26e+02	4.73e+02	3.20e+03
Relative	2.17e-04	5.50e-03	1.32e-02	2.45e-02	4.66e-02	6.20e-02

12 The Jacobian is interpolated by inserting the value of λ into Equation (20).
 13 Table 9 shows summaries of componentwise absolute and relative differences
 14 between interpolated and Cholesky factorization Jacobians for a large regular
 15 grid. The results confirm the claim in Griffith (2004b) that this approxima-
 16 tion performs surprisingly well, with a small maximum componentwise relative
 17 difference.

18 4.2 Chebyshev approximations

19 A second approximation is proposed by Pace and LeSage (2004), who elaborate
 20 a Chebyshev factorization, where:

$$\ln(|\mathbf{I} - \lambda \mathbf{W}|) \approx \sum_{j=1}^{q+1} c_j(\lambda) \text{tr}(\mathbf{T}_{j-1}(\mathbf{W})) - \frac{n}{2} c_1(\lambda) \quad (21)$$

1 where $\mathbf{T}_0(\mathbf{W}) = \mathbf{I}$, $\mathbf{T}_1(\mathbf{W}) = \mathbf{W}$, $\mathbf{T}_2(\mathbf{W}) = 2\mathbf{W}^2 - \mathbf{I}$, $\mathbf{T}_{k+1}(\mathbf{W}) = 2\mathbf{W}\mathbf{T}_k(\mathbf{W}) -$
2 $\mathbf{T}_{k-1}(\mathbf{W})$, and q represents the highest power of the approximating polyno-
3 mial. The matrix traces can be set up without knowledge of the λ values
4 entering into the Jacobian, and may be constructed more efficiently as shown
5 by Pace and LeSage (2004, p. 188); the maximum value of q is taken here as
6 5 (see also LeSage and Pace, 2009, pp. 105–108).

7 The $q + 1$ coefficients $c_j(\lambda)$ are given by:

$$c_j(\lambda) = \left(\frac{2}{q+1} \right) \sum_{k=1}^{q+1} \ln \left[1 - \lambda \cos \left(\frac{\pi(k-0.5)}{q+1} \right) \right] \cos \left(\frac{\pi(j-1)(k-0.5)}{q+1} \right). \quad (22)$$

8 No matrix operations are involved in calculating the approximations to the
9 Jacobian for successive values of λ , yielding very fast look-up times.

Table 10: Summaries of componentwise absolute differences between Jacobians calculated using sparse Cholesky factorization and Chebyshev $q = 5$ Jacobians for five symmetric data sets.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Polish gminy	0.00e+00	1.87e-04	1.33e-02	2.66e-01	1.26e-01	4.93e+00
US Counties	0.00e+00	4.57e-05	7.76e-03	4.03e-01	2.29e-01	4.30e+00
World grid	0.00e+00	2.46e-03	1.82e-01	5.02e+00	3.05e+00	7.37e+01
US ZCTA	0.00e+00	7.92e-04	2.34e-02	3.34e+00	1.28e+00	4.27e+01
US census tracts	0.00e+00	9.56e-04	8.51e-02	8.51e+00	3.62e+00	1.28e+02

10 Table 10 shows summaries of componentwise absolute differences between
11 Jacobians calculated using sparse Cholesky factorization and Chebyshev $q = 5$
12 Jacobians. It suggests that performance using $q = 5$ is adequate with mod-
13 erately sized data sets, but perhaps that the results for maximum absolute
14 differences are unsatisfactory for larger n . Figure 3 compares the behaviour
15 of the Chebyshev approximation with $q = 5$ and that of Monte Carlo approxi-
16 mations with sparse Cholesky Jacobians at the upper tail, with large values of
17 λ . The right-hand panel shows how absolute differences between Chebyshev
18 and Cholesky Jacobians for the US census tracts data set grow from moderate
19 values of λ , and grow well before differences mount for the mean of the 100
20 simulations of Monte Carlo approximations. The differences top out at about
21 128 for $\lambda = 0.99$, but are clearly a problem. We return to these results again
22 in the following section with reference to the Monte Carlo approximation.

4.3 Monte Carlo approximations

Barry and Pace (1999) propose the use of a Monte Carlo approximation with two tuning parameters, p and m (see also LeSage and Pace, 2009, pp. 96–105). The outcome is minus the mean of p random variates V_i , calculated from an $n \times p$ matrix of drawings from the Normal distribution with zero mean and unit variance, and m products of this matrix and the spatial weights matrix \mathbf{W} . The set-up function prepares a list of these expansion products, storing trace estimates of powers of \mathbf{W} in m vectors of length p ; the first two traces may be replaced by their analytical values, as in the MatlabTM Spatial Econometrics toolbox and as described by LeSage and Pace (2009, p. 99). Zhang and Leithead (2007) suggest that the p candidate draws could be subject to selection to eliminate inappropriately generated seeds, but this has not been found necessary here. The original description due to Barry and Pace (1999) is followed, using $p = 16$ and $m = 30$ as in Walde et al. (2008).

The implementation here re-uses the same vectors of random numbers for each λ value by calculating a list of expansion products, but pre-calculates m matrix operations on $n \times p$ matrices to save time. The method is as follows:

$$V_i = -n \sum_{j=1}^m \frac{\mathbf{x}'_i \mathbf{W}^j \mathbf{x}_i}{\mathbf{x}'_i \mathbf{x}_i} \frac{\lambda^k}{j} \quad (23)$$

for $i = 1, \dots, p$, \mathbf{W} with real eigenvalues in $[-1, 1]$, and $\mathbf{x}_i \sim N(0, \mathbf{I})$.

Table 11: Summaries of componentwise absolute differences between Jacobians calculated using sparse Cholesky factorization and mean $p = 16, m = 30$ Monte Carlo Jacobians for 100 simulations for five symmetric data sets.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Polish gminy	0.00e+00	2.77e-03	2.44e-02	2.55e-01	7.87e-02	1.30e+01
US Counties	0.00e+00	1.42e-02	5.78e-02	3.96e-01	1.42e-01	2.70e+01
World grid	0.00e+00	4.92e-03	1.64e-02	1.51e+00	6.12e-02	1.16e+02
US ZCTA	0.00e+00	3.35e-02	1.04e-01	1.44e+00	2.83e-01	1.00e+02
US census tracts	0.00e+00	4.52e-03	4.48e-02	4.19e+00	1.73e-01	3.12e+02

To follow up the extensive presentation of the Monte Carlo method in LeSage and Pace (2009), we undertook a simulation study of 100 runs per data set. Consequently, the results reported here are not representative of typical numbers of random variates p , although the number of series terms remains fixed at $m = 50$. The results here are means and standard deviations from the 100 simulations for each value of λ for $p = 16$ and $p = 32$, for $m = 30$ traces. Figure 3 shows the problems faced when using Monte Carlo-based

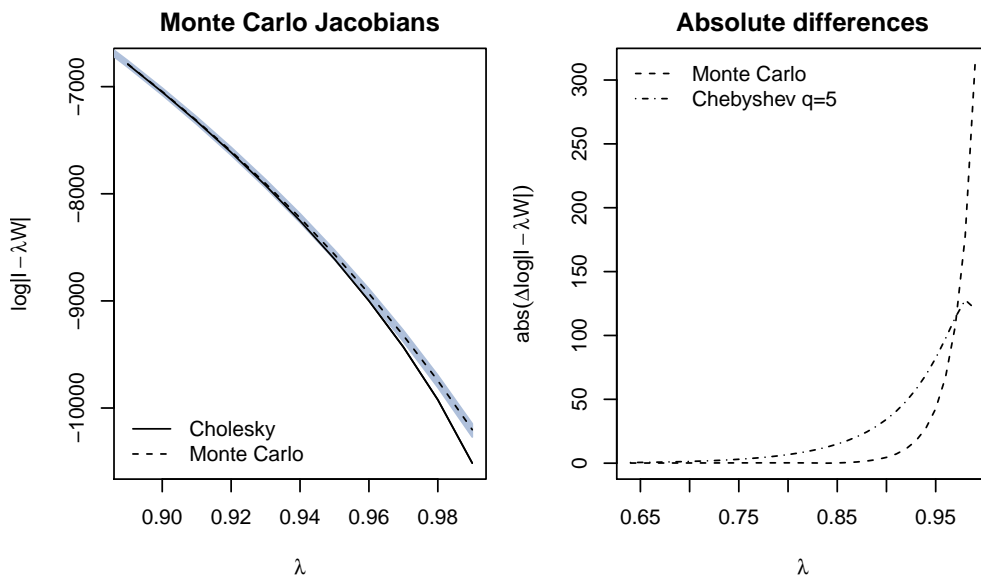


Figure 3: US census tracts data set — Left panel: Cholesky and mean $p = 16$, $m = 30$ Monte Carlo Jacobians for large values of λ , showing the envelope for 100 MC simulations; right panel: componentwise absolute differences between Cholesky and mean $p = 16$, $m = 30$ Monte Carlo Jacobians and Cholesky and Chebyshev Jacobians for large values of λ .

1 Jacobians at the upper tail for the US census tracts data set. The Monte
 2 Carlo Jacobians take values that are higher than the Cholesky Jacobians, and
 3 the 100 simulation envelope, shown as a gray polygon, does not include the
 4 Cholesky Jacobians for $\lambda \geq 0.94$ for this data set. Unfortunately, doubling the
 5 number of random variates to $p = 32$ has minimal impact on the problem. In
 6 the right-hand panel, we can see that as λ approaches its upper bound, the
 7 absolute difference increases very rapidly.

8 Table 11 shows that this problem is pervasive, and in all of the five data
 9 sets, the largest absolute difference was found for $\lambda = 0.99$. Because λ was
 10 only examined for values from -0.90 to 0.99 , the lower end of its range is not
 11 exposed to the same challenges. However, the problem only affects the extreme
 12 edge of the feasible range of λ . In Barry and Pace (1999, p. 42), use is made
 13 of the assumption that the weights matrix is symmetric or similar to symmet-
 14 ric, with real eigenvalues. With regard to the Monte Carlo approximation,
 15 LeSage and Pace (2009, p. 97) relax this requirement.

Table 12: Summaries of componentwise absolute differences between Jacobians
 calculated using sparse LU factorization and mean $p = 16, m = 30$ Monte Carlo
 Jacobians for 100 simulations for five asymmetric data sets.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Polish gminy	0.00e+00	3.28e-03	2.78e-02	2.93e-01	9.57e-02	1.54e+01
US Counties	0.00e+00	1.14e-02	4.43e-02	4.12e-01	9.30e-02	2.91e+01
World grid	0.00e+00	4.78e-03	2.06e-02	8.52e-01	5.52e-02	6.57e+01
US ZCTA	0.00e+00	3.70e-02	1.48e-01	3.69e+00	3.27e-01	2.78e+02
US census tracts	0.00e+00	7.68e-03	7.16e-02	1.05e+01	2.31e-01	8.48e+02

16 Turning then to our asymmetric examples, we can examine the absolute
 17 differences between the Monte Carlo approximations and LU Jacobians. Table
 18 12 shows summaries of the componentwise absolute differences, which appear
 19 similar in character to their symmetric matrix counterparts.

20 For completeness, we may examine the assumption made in Barry and Pace
 21 (1999, p. 42) that “after a suitable rescaling” of the matrix \mathbf{W} its eigenvalues
 22 are in $[-1, 1]$, and λ is in $(-1, 1)$. The term “suitable rescaling” is not discussed
 23 by Barry and Pace, but may commonly refer to row standardization; it does not
 24 appear to indicate different rescaling in positive and negative ranges. Table 13
 25 shows that while for the large $1,000 \times 1,000$ regular grid with rook neighbors
 26 and binary weights, so with no rescaling, the Monte Carlo Jacobians are an
 27 acceptable match for the analytical eigenvalue Jacobians, this does not hold for
 28 the regular grid with binary queen neighbors. The Monte Carlo Jacobians move
 29 sharply towards negative infinity after the analytical Jacobian for negative λ

Table 13: Summaries of componentwise absolute differences between analytical Jacobians and Monte Carlo Jacobians for 1000×1000 data sets, with binary weights.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Rook, p=16, m=30	0.00e+00	8.68e-01	6.47e+00	6.34e+01	3.25e+01	1.14e+03
Rook, p=30, m=50	0.00e+00	2.87e-01	6.48e-01	1.32e+01	4.84e+00	2.50e+02
Queen, p=16, m=30	0.00e+00	1.26e+00	1.64e+01	1.73e+09	2.89e+04	4.68e+10
Queen, p=30, m=50	0.00e+00	3.07e-01	1.67e+00	2.41e+14	4.05e+05	7.88e+15

1 moves below that calculated near the top of its range. In both of these cases
2 the assumptions of the method are not met, but in the binary rook case,
3 the minimum and maximum eigenvalues are ± 3.9998 , in the binary queen
4 case they are $-3.999961, 7.999941$, with corresponding bounds for λ of ± 0.25
5 and $-0.250, 0.125$. If the binary queen weights are rescaled to sum to n , this
6 problem is still present, suggesting that the effective bounds on λ for the Monte
7 Carlo method are $(-1/\max(\lambda), 1/\max(\lambda))$.¹⁰

8 A further discussion is given by Pace and LeSage (2009) with reference to
9 the use of a sampling approach to computing the log-determinant with log-
10 pivots; they repeat here that both symmetric and asymmetric spatial weights
11 may be used. In addition, a “computational method for calculating lower
12 moments of the actual distribution of eigenvalues of the spatial weights matrix
13 and applying those for the efficient calculation of the log-determinant” has
14 recently been proposed by Smirnov and Anselin (2009, p. 2980), and it is to
15 this approach that we now turn.

16 4.4 Computing the lower order moments of eigenvalues

17 The development of the Smirnov and Anselin (2009) procedure starts from the
18 assumption that spatial weights matrix \mathbf{W} is either symmetric or obtained by
19 a similarity transformation from a row-standardized matrix with intrinsically
20 symmetric neighbor relations. The interest is in using the lower order moments
21 of the set of (real) eigenvalues of \mathbf{W} :

$$\Omega^j = \sum_{i=1}^n \zeta_i^j \quad (24)$$

¹⁰The existence of this problem was known when the approach was first proposed; if estimates are needed for larger negative values of λ , they may be constructed by extrapolation (Pace, personal communication).

1 where Ω^j is the j -th non-central moment, and ζ_i is the i -th eigenvalue of \mathbf{W} ¹¹.
 2 Further development shows that:

$$\ln(|\mathbf{I} - \lambda \mathbf{W}|) = - \sum_{j=1}^{\infty} \frac{1}{j} \lambda^j \text{tr} \mathbf{W}^j = \sum_{j=1}^{\infty} \frac{1}{j} \lambda^j \Omega^j \quad (25)$$

3 in which we see the importance of the series of traces of powers of \mathbf{W} (low
 4 order moments of the eigenvalues of \mathbf{W}). As Smirnov and Anselin (2009, p.
 5 2982–2983) point out, because the $\frac{1}{j} \lambda^j$ terms decrease quite steeply for small
 6 j , the problem may be expressed in two parts (see also Martin, 1993):

$$\ln(|\mathbf{I} - \lambda \mathbf{W}|) = \sum_{j=1}^m \frac{1}{j} \lambda^j \Omega^j - R_m(\lambda) \quad (26)$$

7 where

$$R_m(\lambda) = \lim_{n \rightarrow \infty} \sum_{j=m+1}^n \frac{1}{j} \lambda^j \Omega^j \quad (27)$$

8 is a correction term. Li et al. (2012) also use the two first moments of the
 9 eigenvalues (traces) of \mathbf{W} in a refinement of the approximate profile-likelihood
 10 estimator of λ .

11 4.4.1 Exact moments of eigenvalues

12 There are two interesting innovations in Smirnov and Anselin (2009) which
 13 we consider separately. The first is that the exact low order moments of the
 14 eigenvalues of \mathbf{W} may be computed using sparse matrix techniques, with:

$$\Omega^j = \text{tr} \mathbf{W}^j = \sum_{i=1}^n \boldsymbol{\eta}_i' \mathbf{W}^j \boldsymbol{\eta}_i \quad (28)$$

15 where $\boldsymbol{\eta}_i$ is a vector of canonical base, that is a vector of length n of zeros, except
 16 for the i -th element equal to 1. As is shown in Smirnov and Anselin (2009, pp.
 17 2983–2984), the storage requirements for this operation are not large, and
 18 it may be parallelized. Table 14 shows that Equation (28) is successful in
 19 computing the exact low order moments of the eigenvalues of \mathbf{W} .

20 The second innovation is concerned with the approximation of $R_m(\lambda)$, in
 21 which the successive moments are interpolated from the four highest exact
 22 moments $m-3, \dots, m$. This innovation is justified by an analysis of the asymp-
 23 totic properties of the moments of the eigenvalues (Smirnov and Anselin, 2009,

¹¹A theoretical analysis of lower order moments of such matrices is given by Griffith (2003, 42–46), with developments for chosen tessellations.

Table 14: Summaries of componentwise absolute differences between low order moments computed using Equation (24) and Equation (28), $m = 30$.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Gminy	0.00e+00	2.84e-14	6.75e-14	8.44e-14	1.15e-13	3.13e-13
Counties	9.88e-15	9.41e-14	2.42e-13	3.41e-13	5.06e-13	1.70e-12
World grid	5.27e-14	5.68e-13	1.72e-12	2.87e-12	3.52e-12	1.20e-11

1 pp. 2984–2985). The use of the factor of unit positive eigenvalues (the num-
2 ber of non-empty subgraphs) and the factor of unit negative eigenvalues (the
3 number of cyclical subgraphs) permits the unit roots of \mathbf{W} to be set apart,
4 so that the remainder of the moments converges to zero, and shows that in-
5 terpolation of higher-order moments will provide sufficiently accurate results
6 (see equation (7) Smirnov and Anselin, 2009, p. 2985). The implementation
7 reported here uses only the simplified form shown in Equation 26 (see equa-
8 tion (5) Smirnov and Anselin, 2009, p. 2983), neither computing nor using
9 the dominant eigenvalues of \mathbf{W} . We indicate our usage below by terming the
10 implementation simplified.

Table 15: Summaries of componentwise absolute differences between Jacobians calculated using eigenvalues, and the simplified Smirnov and Anselin (2009) procedure without interpolation (A), with interpolation and truncation (B), and with interpolation without truncation (C), row-standardized Polish gminy and US Counties data sets converted to symmetric matrices by similarity, 190 values for λ between -0.9 and 0.99.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Gminy (A)	0.00e+00	3.55e-15	8.14e-11	1.59e-01	2.52e-05	1.19e+01
Gminy (B)	0.00e+00	3.78e-15	7.71e-11	1.44e-01	2.63e-05	1.09e+01
Gminy (C)	0.00e+00	3.78e-15	7.71e-11	1.44e-01	2.63e-05	1.09e+01
Counties (A)	0.00e+00	4.17e-14	9.54e-11	3.52e-01	2.45e-05	2.76e+01
Counties (B)	0.00e+00	4.17e-14	5.84e-11	2.94e-01	3.52e-05	2.38e+01
Counties (C)	0.00e+00	4.17e-14	5.84e-11	2.94e-01	3.52e-05	2.38e+01

11 Table 15 shows summaries of componentwise absolute differences between
12 Smirnov and Anselin (2009) and eigenvalue Jacobians. The simplified Smirnov-
13 Anselin approximations, with $m = 30$ traces computed using Equation (28),
14 and log determinants with Equation (26), perform very well in terms of com-
15 ponentwise absolute differences for most of the considered range of λ for all
16 considered variants of $R_m(\lambda)$, the interpolation term given in equation 27,

1 where we interpolate successive values of $\Omega^j, j = m + 1, \dots, n$ and $j \geq 0$ as
 2 (Smirnov and Anselin, 2009, p. 2985):

$$\Omega^{m+2j} = \Omega^m \times \left(\frac{\Omega^m}{\Omega^{m-2}} \right)^j, \quad (29)$$

3 and

$$\Omega^{m+2j-1} = \Omega^{m-1} \times \left(\frac{\Omega^{m-1}}{\Omega^{m-3}} \right)^j. \quad (30)$$

4 The three variants of $R_m(\lambda)$ shown in Table 15 are: (A) no interpolation
 5 used, $R_m(\lambda) = 0$; (B) interpolation truncated when the increment in the sum-
 6 mation in Equation (27) falls below a small threshold; and (C) interpolation
 7 with the summation running to n . Using the interpolation reduces the compo-
 8 nentwise absolute differences shown in the table, but in these two cases, there
 9 is no advantage in allowing the summation of the interpolation term $R_m(\lambda)$
 10 to run to n . However, had the weights been of rook neighbors on a regular
 11 grid, truncating the summation would have led to error, as all odd moments
 12 are zero (Smirnov and Anselin, 2009, p. 2985), and truncation would have
 13 occurred prematurely.

Table 16: Jacobian values for large λ computed using eigenvalues and traces ($m = 30$ low order moments with interpolation) for row-standardized Polish gminy and US Counties data sets converted to symmetric matrices by similarity.

λ	Gminy (eigen)	Gminy (trace)	Counties (eigen)	Counties (trace)
0.90	-279.39	-279.22	-400.43	-400.12
0.91	-290.06	-289.80	-416.37	-415.90
0.92	-301.35	-300.95	-433.35	-432.63
0.93	-313.36	-312.75	-451.52	-450.41
0.94	-326.20	-325.26	-471.14	-469.40
0.95	-340.05	-338.59	-492.52	-489.78
0.96	-355.14	-352.84	-516.17	-511.77
0.97	-371.85	-368.15	-542.90	-535.66
0.98	-390.84	-384.69	-574.30	-561.80
0.99	-413.60	-402.67	-614.49	-590.70

14 Table 16 shows that the worrying maximum absolute differences between
 15 eigenvalue Jacobians and simplified Smirnov-Anselin approximation Jacobians
 16 with $m = 30$ traces seen in Table 15 appear near the upper bound of λ . The

1 same effect is found near the lower bound, but is not reported here. From about
2 $\lambda = 0.9$, the eigenvalue Jacobians begin to take successively lower values than
3 the simplified Smirnov-Anselin Jacobians as λ approaches its upper bound,
4 indicating that the interpolation term is not offering sufficient correction for
5 these values of λ . The simplified Smirnov-Anselin approximation Jacobians
6 used here are computed using the exact methods for low order moments and
7 interpolation described by Smirnov and Anselin (2009, pp. 2983–2985), and
8 presented here as Equations (26)–(30).

9 4.4.2 Approximation of low order traces

10 It is of interest to explore the relationship of the exact method for computing
11 the low order moments of the eigenvalues of \mathbf{W} introduced by Smirnov and Anselin
12 (2009) with approximations already present in the literature, and implemented
13 in software. Barry and Pace (1999), as we have seen above in connection
14 with the Monte Carlo method, wrote the following approximation of log-
15 determinant:

$$\ln(|\mathbf{I} - \lambda \mathbf{W}|) \approx \bar{V}, \quad (31)$$

16 where

$$\bar{V} = -\frac{n}{p} \sum_{j=1}^m \sum_{i=1}^p \frac{\mathbf{x}_i' \mathbf{W}^j \mathbf{x}_i}{\mathbf{x}_i' \mathbf{x}_i} \cdot \frac{\lambda^j}{j}, \quad i = 1, \dots, p, \quad \mathbf{x}_i \sim N(0, \mathbf{I}). \quad (32)$$

17 In their case, the \mathbf{x}_i are p draws of normal random variates of length n . In
18 the Smirnov and Anselin (2009) method, we have:

$$\ln(|\mathbf{I} - \lambda \mathbf{W}|) \approx - \sum_{j=1}^m \frac{1}{j} \lambda^j \Omega^j, \quad (33)$$

19 where $\Omega^j = \sum_{i=1}^n \boldsymbol{\eta}_i' \mathbf{W}^j \boldsymbol{\eta}_i$, $\boldsymbol{\eta}_i$ are vectors of canonical base in \mathbb{R}^n .

$$\ln(|\mathbf{I} - \lambda \mathbf{W}|) \approx - \sum_{j=1}^m \frac{\lambda^j}{j} \sum_{i=1}^n \boldsymbol{\eta}_i' \mathbf{W}^j \boldsymbol{\eta}_i = - \sum_{j=1}^m \sum_{i=1}^n \boldsymbol{\eta}_i' \mathbf{W}^j \boldsymbol{\eta}_i \frac{\lambda^j}{j}. \quad (34)$$

20 In (32) we can put $\mathbf{x}_i = \boldsymbol{\eta}_i$. Vectors of canonical base satisfy the condition
21 $\boldsymbol{\eta}_i' \boldsymbol{\eta}_i = 1$. If additionally we assume $p = n$, we obtain equation (34). From this,
22 we can see that while both procedures truncate the power series, the approaches
23 they use to compute the traces differ, despite their apparent similarity. In the
24 light of this result, it is of interest to see how the use of the Monte Carlo
25 method for calculating the initial m traces of the power series of \mathbf{W} compares
26 with the Smirnov-Anselin algorithm 1 approach.

Table 17: Summaries of componentwise relative differences for the first 30 traces between Smirnov-Anselin algorithm 1 and means of 100 simulated Monte Carlo traces, for $p=16$ and $p=32$, for five symmetric data sets.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Polish gminy $p=16$	4.02e-16	2.80e-03	4.64e-03	3.78e-03	5.04e-03	5.13e-03
Polish gminy $p=32$	4.02e-16	2.46e-03	2.71e-03	2.57e-03	2.87e-03	3.22e-03
US Counties $p=16$	3.07e-15	1.16e-03	1.86e-03	3.98e-03	6.90e-03	9.56e-03
US Counties $p=32$	3.07e-15	1.20e-03	1.42e-03	2.16e-03	3.62e-03	4.10e-03
World grid $p=16$	5.40e-15	3.94e-04	5.95e-04	6.14e-04	8.23e-04	1.36e-03
World grid $p=32$	5.40e-15	2.92e-04	4.85e-04	4.82e-04	6.77e-04	8.37e-04
US ZCTA $p=16$	1.24e-14	1.13e-03	1.33e-03	1.47e-03	2.05e-03	2.27e-03
US ZCTA $p=32$	1.24e-14	6.21e-04	7.29e-04	7.13e-04	8.76e-04	9.83e-04
US census tracts $p=16$	5.15e-14	8.32e-05	1.46e-04	1.67e-04	2.37e-04	3.87e-04
US census tracts $p=32$	5.14e-14	1.48e-04	1.94e-04	1.76e-04	2.18e-04	2.86e-04

1 Table 17 reports the componentwise relative differences for the first 30
2 traces between Smirnov-Anselin algorithm 1 and Monte Carlo traces. The
3 Monte Carlo traces reported here are means over 100 simulations for two dif-
4 ferent values of p : 16 and 32. We report relative differences (see Section 1.2)
5 in this case, as we are interested in the relative impact of Monte Carlo ap-
6 proximations of the traces on the computation of the log determinant. As we
7 see, the differences introduced by Monte Carlo approximation are tangible al-
8 though not large; in the case of rook neighbor regular grids, the Monte Carlo
9 trace estimates do not respect the condition that all odd moments are zero.

10 Figure 4 may be read in conjunction with Figure 3 and Table 16, here
11 comparing the values of the log determinant for the US census tracts data
12 set for high values of λ . In the left panel, neither the interpolated simpli-
13 fied Smirnov-Anselin approximation nor the mean of 100 simulations of Monte
14 Carlo approximations keep up with the fall of the Jacobian towards minus in-
15 finity at the upper bound of λ . Of course, the Monte Carlo approximations are
16 not interpolated, and are further from the Cholesky Jacobians than the inter-
17 polated simplified Smirnov-Anselin approximations. The right panel of Figure
18 4 shows the absolute differences between the Cholesky Jacobians and the ap-
19 proximations. The mean of 100 simulated Monte Carlo approximations and
20 the non-interpolated simplified Smirnov-Anselin approximations perform simi-
21 larly, with improvements in performance yielded by interpolation when $R_m(\lambda)$
22 is used. When λ is close to its bounds, it is unavoidable that approxima-
23 tion methods are subject to some loss of accuracy, although their performance

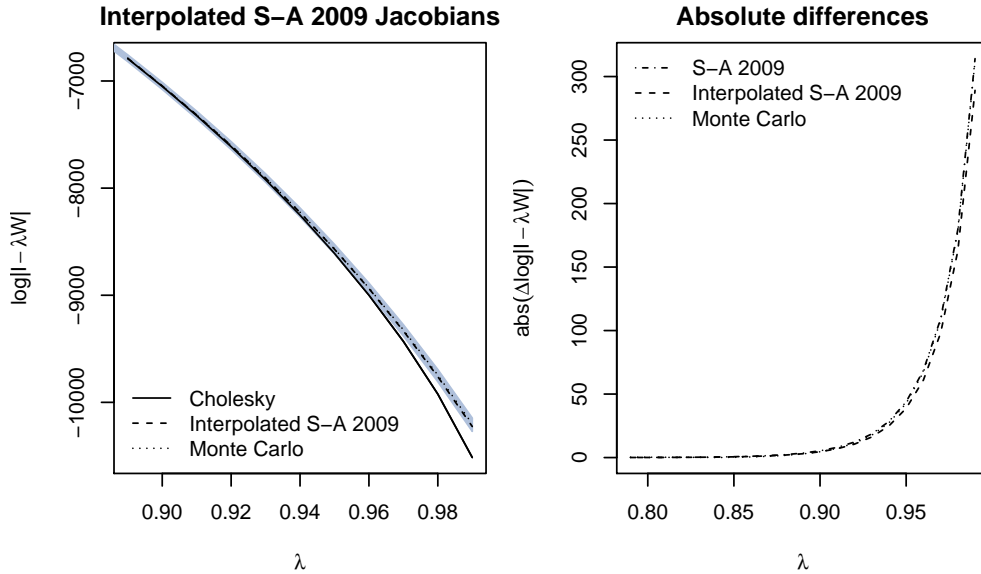


Figure 4: US census tracts data set — Left panel: Cholesky, interpolated $m = 30$ simplified Smirnov-Anselin, and mean $p = 16$, $m = 30$ Monte Carlo Jacobians for large values of λ , showing the envelope for 100 MC simulations; right panel: componentwise absolute differences between Cholesky and $m = 30$ simplified Smirnov-Anselin, interpolated $m = 30$ simplified Smirnov-Anselin and mean $p = 16$, $m = 30$ Monte Carlo Jacobians for large values of λ .

1 across the broad range of moderate values of λ is encouraging.

2 5 Conclusions

3 We have reviewed implementation details of sparse matrix and approximate
 4 approaches to computing the Jacobian log determinant term needed in fitting
 5 Gaussian spatial regression models using maximum likelihood and Bayesian
 6 methods. Many of the implementation details are not obvious to users, but
 7 do affect their ability to get work done. Our conclusions are that the use of
 8 eigenvalues and sparse matrix factorizations are equivalent — the Jacobian
 9 values for given λ are the same within acceptable absolute differences, and
 10 the differences do not affect the inferences drawn on the fitted models. When
 11 data sets are of a size for which the calculation of eigenvalues is feasible, their
 12 use remains the best established and most thoroughly studied (Griffith, 2000,
 13 2003). For queen or rook criterion regular grids (complete rectangular regions),
 14 analytical eigenvalues may be computed very fast for grids of arbitrary size,

1 and their use is effectively unlimited by memory or run time (Griffith and Sone,
2 1995).

3 In modeling situations with larger data sets for which the dense matrix
4 techniques needed to calculate the eigenvalues of the spatial weights matrix
5 are impractical because of memory and/or time limitations, and in which
6 analytical eigenvalues are not available, Cholesky factorization may be used
7 with intrinsically symmetric spatial weights, and LU factorization with intrinsi-
8 cally asymmetric spatial weights. Where the weights are symmetric, Cholesky
9 factorization is typically about twice as fast as LU factorization. Updating
10 Cholesky factorizations seem to provide speed benefits increasing in n and
11 in the sparseness of the weights, varying from three times faster than non-
12 updating factorizations for the smaller irregular data sets to almost ten times
13 faster for the US census tracts data set.

14 Turning to approximations, we find that our simplification of Smirnov and Anselin
15 (2009) provides a useful addition to Monte Carlo (Barry and Pace, 1999) and
16 Chebyshev Pace and LeSage (2004) approximations, but that all of these ap-
17 proaches suffer from degrading accuracy close to the upper bound of λ . The
18 introduction of an interpolated term by Smirnov and Anselin (2009) mitigates
19 the degradation somewhat, but not completely. This suggests that more terms
20 in the power series should be taken as λ approaches its bound, increasing m ,
21 the number of traces used. The timings for Monte Carlo approximations are
22 moderate, but as n — the number of observations, m — the number of trace
23 terms, and p — the number of random variates used in Monte Carlo simula-
24 tion increase, set-up run times increase. In the case of the memory-conserving
25 Smirnov and Anselin (2009) method (Equation 28), careful parallel program-
26 ming is required to achieve reasonable times for calculating m traces for large
27 n , with Monte Carlo approximation typically being considerably faster.

28 The approach used by the MatlabTM Spatial Econometrics toolbox pre-
29 computes a grid of log determinant values along the line between the minimum
30 and maximum feasible values of λ . When the value of the log-likelihood func-
31 tion is computed, the value of the Jacobian is retrieved from this grid based
32 on the value of λ proposed by the numerical optimizer or the Bayesian proce-
33 dure in use (LeSage and Pace, 2009). This differs from the implementations in
34 GeoDa, OpenGeoDa, StataTM, and the R package **spdep**, in which only set-up
35 terms, such as eigenvalues, are pre-computed. The run times are then a com-
36 bination of set-up time, which is typically long for eigenvalue calculation, and
37 the time taken to find each log determinant, which is short when eigenvalues
38 are used but long for Cholesky or LU factorization.

39 Summarizing, if the data set is small, the eigenvalue method may be used
40 with both symmetric and asymmetric spatial weights matrices. For larger,

1 sparse, symmetric spatial weights matrices, the analyst can choose between
2 the Cholesky and updating Cholesky method, the characteristic polynomial
3 approximation, the lower order moments approximation, the Chebyshev ap-
4 proximation, or the Monte Carlo approximation. If the data take the form
5 of a regular grid, and binary spatial weights with the rook or queen neigh-
6 bor criterion are appropriate, analytical eigenvalues may be used. Finally, for
7 larger, sparse, asymmetric spatial weights matrices, the choice stands between
8 the LU method and the Monte Carlo approximation. Care should be taken
9 when using the Monte Carlo approximation if the assumptions with regard to
10 the feasible range of the spatial coefficient underlying the method described
11 by Barry and Pace (1999) and LeSage and Pace (2009) are not satisfied.

12 **References**

- 13 Anselin, L., Syabri, I., and Kho, Y. (2006). GeoDa: An introduction to spatial
14 data analysis. *Geographical Analysis*, 38(1):5–22.
- 15 Barry, R. and Pace, R. (1999). Monte Carlo estimates of the log determinant of
16 large sparse matrices. *Linear Algebra and its Applications*, 289(1-3):41–54.
- 17 Bates, D. and Maechler, M. (2012). *Matrix: Sparse and Dense Matrix Classes*
18 *and Methods*. R package version 1.0-6.
- 19 Bernstein, D. (2009). *Matrix Mathematics: Theory, Facts, and Formulas*.
20 Princeton University Press, Princeton.
- 21 Bivand, R. S. (1984). Regression modeling with spatial dependence: an appli-
22 cation of some class selection and estimation methods. *Geographical Analy-*
23 *sis*, 16:25–37.
- 24 Bivand, R. S. and Portnov, B. A. (2004). Exploring spatial data analysis tech-
25 niques using R: the case of observations with no neighbours. In Anselin, L.,
26 Florax, R. J. G. M., and Rey, S. J., editors, *Advances in Spatial Economet-*
27 *rics: Methodology, Tools, Applications*, pages 121–142. Springer, Berlin.
- 28 Brent, R. (1973). *Algorithms for Minimization without Derivatives*. Prentice-
29 Hall, Englewood Cliffs N.J.
- 30 Cliff, A. D. and Ord, J. K. (1973). *Spatial Autocorrelation*. Pion, London.
- 31 Davis, T. and Hager, W. (1999). Modifying a sparse cholesky factorization.
32 *SIAM Journal on Matrix Analysis and Applications*, 20(3):606–627.

- 1 Davis, T. A. (2005). Algorithm 849: A concise sparse cholesky factorization
2 package. *ACM Trans. Math. Softw.*, 31(4).
- 3 Davis, T. A. (2006). *Direct Methods for Sparse Linear Systems*. Society for
4 Industrial and Applied Mathematics, Philadelphia.
- 5 Furrer, R. and Sain, S. R. (2010). spam: A sparse matrix R package with
6 emphasis on MCMC methods for Gaussian Markov random fields. *Journal*
7 *of Statistical Software*, 36(10):1–25.
- 8 Gasim, A. A. (1988). First-order autoregressive models: a method for obtaining
9 eigenvalues for weighting matrices. *Journal of Statistical Planning and*
10 *Inference*, 18:391–398.
- 11 Griffith, D. and Luhanga, U. (2011). Approximating the inertia of the adja-
12 cency matrix of a connected planar graph that is the dual of a geographic
13 surface partitioning. *Geographical Analysis*, 43:383–402.
- 14 Griffith, D. A. (2000). Eigenfunction properties and approximations of selected
15 incidence matrices employed in spatial analyses. *Linear Algebra and its*
16 *Applications*, 321(1-3):95–112.
- 17 Griffith, D. A. (2003). *Spatial Autocorrelation and Spatial Filtering*. Springer,
18 New York.
- 19 Griffith, D. A. (2004a). Extreme eigenfunctions of adjacency matrices for pla-
20 nar graphs employed in spatial analyses. *Linear Algebra and its Applications*,
21 388:201–219.
- 22 Griffith, D. A. (2004b). Faster maximum likelihood estimation of very large
23 spatial autoregressive models: An extension of the Smirnov-Anselin result.
24 *Journal of Statistical Computation and Simulation*, 74(12):855–866.
- 25 Griffith, D. A. and Layne, L. J. (1999). *A Casebook for Spatial Statistical Data*
26 *Analysis*. Oxford University Press, New York.
- 27 Griffith, D. A. and Sone, A. (1995). Trade-offs associated with normalizing con-
28 stant computational simplifications for estimating spatial statistical models.
29 *Journal of Statistical Computation and Simulation*, 51:165–183.
- 30 Higham, N. J. (2002). *Accuracy and stability of numerical algorithms, 2nd ed.*
31 Society for Industrial and Applied Mathematics, Philadelphia.
- 32 Higham, N. J. (2009). Cholesky factorization. *Wiley Interdisciplinary Reviews:*
33 *Computational Statistics*, 1(2):251–254.

- 1 Kaluzny, S. P., Vega, S. C., Cardoso, T. P., and Shelly, A. A. (1998).
2 *S+SpatialStats, User Manual for Windows and UNIX*. Springer-Verlag,
3 Berlin.
- 4 Kelejian, H. H. and Prucha, I. R. (2010). Specification and estimation of spatial
5 autoregressive models with autoregressive and heteroskedastic disturbances.
6 *Journal of Econometrics*, 157:53–67.
- 7 LeSage, J. and Pace, R. (2009). *Introduction to Spatial Econometrics*. CRC
8 Press, Boca Raton, FL.
- 9 Li, H., Calder, C., and Cressie, N. (2012). One-step estimation of spatial
10 dependence parameters: Properties and extensions of the APLE statistic.
11 *Journal of Multivariate Analysis*, 105:68–84.
- 12 Martin, R. (1993). Approximations to the determinant term in gaussian max-
13 imum likelihood estimation of some spatial models. *Communications in*
14 *Statistics — Theory and Methods*, 22:189–205.
- 15 Ng, E. G. and Peyton, B. W. (1993). Block sparse cholesky algorithms on
16 advanced uniprocessor computers. *SIAM Journal on Scientific Computing*,
17 14:1034–1056.
- 18 Ord, J. (1975). Estimation methods for models of spatial interaction. *Journal*
19 *of the American Statistical Association*, 70(349):120–126.
- 20 Pace, R. and Barry, R. (1997a). Fast CARs. *Journal of Statistical Computation*
21 *and Simulation*, 59(2):123–145.
- 22 Pace, R. and Barry, R. (1997b). Fast spatial estimation. *Applied Economics*
23 *Letters*, 4(5):337–341.
- 24 Pace, R. and Barry, R. (1997c). Quick computation of spatial autoregressive
25 estimators. *Geographical Analysis*, 29(3):232–247.
- 26 Pace, R. and Barry, R. (1997d). Sparse spatial autoregressions. *Statistics &*
27 *Probability Letters*, 33(3):291–297.
- 28 Pace, R. and LeSage, J. (2004). Chebyshev approximation of log-determinants
29 of spatial weight matrices. *Computational Statistics & Data Analysis*,
30 45(2):179–196.
- 31 Pace, R. and LeSage, J. (2009). A sampling approach to estimate the log
32 determinant used in spatial likelihood problems. *Journal of Geographical*
33 *Systems*, 11:209–225.

- 1 R Core Team (2012). *R: A Language and Environment for Statistical Com-*
2 *puting*. R Foundation for Statistical Computing, Vienna, Austria. ISBN
3 3-900051-07-0.
- 4 Ripley, B. D. (1981). *Spatial Statistics*. Wiley, New York.
- 5 Rue, H. (2001). Fast sampling of Gaussian Markov random fields. *Journal of*
6 *the Royal Statistical Society, Series B*, 63(2):325–338.
- 7 Smirnov, O. and Anselin, L. (2001). Fast maximum likelihood estimation of
8 very large spatial autoregressive models: a characteristic polynomial ap-
9 proach. *Computational Statistics & Data Analysis*, 35(3):301–319.
- 10 Smirnov, O. and Anselin, L. (2009). An O(N) parallel method of computing
11 the Log-Jacobian of the variable transformation for models with spatial in-
12 teraction on a lattice. *Computational Statistics & Data Analysis*, 53(8):2980
13 – 2988.
- 14 Tiefelsdorf, M., Griffith, D. A., and Boots, B. (1999). A variance-stabilizing
15 coding scheme for spatial link matrices. *Environment and Planning A*,
16 31:165–180.
- 17 Varga, R. (2009). *Matrix Iterative Analysis*. Springer, Berlin.
- 18 Walde, J., Larch, M., and Tappeiner, G. (2008). Performance contest between
19 MLE and GMM for huge spatial autoregressive models. *Journal of Statistical*
20 *Computation and Simulation*, 78(2):151–166.
- 21 Zhang, Y. and Leithead, W. (2007). Approximate implementation of the loga-
22 rithm of the matrix determinant in Gaussian process regression. *Journal of*
23 *Statistical Computation and Simulation*, 77(4):329–348.