

SSUS WORKING PAPER SERIES

Fast and unbiased log-determinants for large spatial autoregressive models

Elijah Knaap[1]

[1]University of California, Irvine, Department of Urban Planning and Public Policy

Spatial Science and Urban Systems · UC Irvine 2026-11-04

ABSTRACT Spatial autoregressive models are among the most important tools in applied spatial analysis. A major computational bottleneck in these models is computing the Jacobian term, the cost of which has limited traditional sample sizes considerably, and thus is the subject of a long line of research examining methods that compute this term quickly and accurately. This work is particularly important for Bayesian estimation, where the Jacobian must be computed thousands of times during a sampling run. In this paper, we introduce two new methods for computing the Jacobian term that extend the state of the art, allow extremely fast (and functionally exact) estimation, and thus provide a route to highly scalable spatial autoregressive modeling. Our first approach converts the Jacobian term into an analytic function that can be interpolated with extremely high accuracy, at little cost, via the AAA algorithm, and is appropriate for most spatial weights matrices, directed or undirected. Our second method is a matrix-free approach leveraging stochastic Chebyshev traces, and is useful when the spatial weights matrix is dense or has complex structure such as stars or hub-and-spoke patterns. We show that existing methods are less efficient than our techniques, and that the most scalable techniques in current use introduce bias as the autoregressive parameter approaches the boundary of its stability region; ours do not. Using our methods, a spatial autoregressive model can be fit to the entire set of U.S. census tracts ($n = 85{,}230$) in a Bayesian framework with 2,000 draws in about a second of Jacobian computation per chain, where the next-best unbiased estimator takes several minutes.

KEYWORDS spatial econometrics; spatial autoregressive model; log-determinant; rational approximation; Bayesian estimation

1 Introduction

Spatial autoregressive models are among the most important tools in applied spatial analysis. They are the standard means of estimating spillovers, diffusion, and neighborhood effects across regional science, economic geography, and spatial econometrics. The spatial autoregressive (SAR) model (Anselin, 1988; Ord, 1975) specifies

$$y = \rho W y + X\beta + \varepsilon, \tag{1}$$

where W is an $n \times n$ spatial weights matrix, ρ the autoregressive parameter, and X a matrix of covariates. Its reduced form, $y = (I_n - \rho W)^{-1}(X\beta + \varepsilon)$, places the Jacobian

$$\mathcal{J}(\rho) \;=\; \log \lvert I_n - \rho W \rvert \tag{2}$$

in the likelihood. The same term appears in the spatial error and spatial Durbin models and their extensions to limited and count outcomes. Computing it is the principal computational bottleneck of these models. A log-determinant of an $n \times n$ matrix costs $O(n^3)$ in general, and the estimator needs a new one at every value of ρ it considers, so the term has long limited the sample sizes spatial models can reach.

The cost has accordingly been the subject of a long line of research. Every contribution to it answers the same two demands: the Jacobian must be computed quickly, and it must be computed accurately. Ord (1975) diagonalized W once, which is exact but cubic. Later work exploited the sparsity of W, factoring $I_n - \rho W$ by sparse Cholesky or LU (Bivand et al., 2013; Pace & Barry, 1997) and precomputing its values on a grid of ρ for interpolation during estimation (LeSage & Pace, 2009). A parallel branch gave up exactness for speed, approximating the Jacobian through the moments of the spectrum of W (Smirnov & Anselin, 2009), a Chebyshev expansion in W (Pace & LeSage, 2004), or Monte Carlo estimates of its traces (Barry & Pace, 1999). Bivand et al. (2013) compare the resulting methods across the major software ecosystems and find the sparse methods fast and accurate at moderate sizes, and the approximations accurate over most of the range of ρ but in error near its upper bound.

The problem is most acute in Bayesian estimation. A maximum-likelihood optimizer visits a dozen or so values of ρ; a Markov chain Monte Carlo sampler visits one or more at every draw, so a slice-sampling chain of 2,000 draws after warmup evaluates the Jacobian about 20,700 times. At that count neither demand can be relaxed. An exact method that refactors at every draw spends minutes to hours per chain on a large problem, and an approximate method that is fast enough must also be accurate in the particular sense that matters to a posterior, which is not the sense in which approximations have been judged. On the census tracts of the United States, $n = 85{,}230$, refactoring $I_n - \rho W$ at every draw spends 8.9 min per chain with the best current implementation. The most recent Bayesian spatial package (Kuschnig, 2022) requires a dense eigendecomposition that is out of reach at that size.

In this paper we introduce two new methods for computing the Jacobian that extend the state of the art in both branches, and a result on how approximation error reaches an estimate that explains where the existing methods fail. Together the two methods allow extremely fast and functionally exact estimation, exact in the sense that the error they leave in the Jacobian does not move the estimate of ρ, and thus provide a route to highly scalable spatial autoregressive modeling.

Our first method converts the Jacobian into an analytic function to be reconstructed, rather than a quantity to be recomputed at every ρ. $\mathcal{J}$ is analytic in ρ on the interior of the stability region, its only singularities lying at or beyond $\rho = \pm 1$, so its values at a couple of dozen well-placed nodes determine it everywhere. We compute those values by sparse factorization, reusing one symbolic analysis across all of them, and interpolate them with the AAA rational approximation algorithm (Nakatsukasa et al., 2018), whose poles fit the logarithmic singularities of the Jacobian. Its error moves the estimate of ρ by less than 10^{-5}, it costs 24 factorizations however many values of ρ the estimator visits, and each evaluation thereafter takes a few microseconds. The method applies to most spatial weights matrices, directed or undirected: symmetrizable weights factor by Cholesky, and for directed weights such as k-nearest-neighbor graphs and flows we extend symbolic reuse to sparse LU, which current practice refactors from scratch.

Our second method is matrix-free and serves the weights matrices on which factorization grows costly or fails: dense matrices, and sparse ones with complex structure such as stars, hubs, and the cliques that co-located units form in address-level data. It expands the Jacobian in Chebyshev polynomials of W, as Pace & LeSage (2004) do, but where they compute a few traces exactly and truncate, it takes the lowest traces exactly, estimates the higher ones by random probes, and fixes those probes once per fit, so that it needs only matrix-vector products to set up and scalar arithmetic to evaluate.

We also show that existing methods are less efficient, and that the most scalable methods in current use introduce bias as ρ approaches the boundary of its stability region. The Barry–Pace estimator, the standard for large and dense problems, biases the posterior mean of ρ upward by a fifth to a third of a standard deviation at $\rho = 0.9$ on the census tracts, on every one of sixteen draws of its probes, and by more than a full standard deviation at $\rho = 0.95$; Smirnov–Anselin and grid-and-spline interpolation are unbiased in the interior and biased near the boundary. Our first method leaves the estimate unmoved, and the mean bias of our second is indistinguishable from zero, below the Monte Carlo error of the chain it serves.

Finally, we show that an approximate Jacobian should be judged by the slope of its error over the range of ρ an estimate occupies, rather than by the size of that error, and that this slope explains the bias. Approximation error enters a posterior as an exponential tilt: the approximate posterior is the exact one multiplied by the exponential of the error. A constant error therefore cancels in the normalization, exactly, so only the variation of the error across the posterior's support can move an estimate. To first order the shift in the posterior mean is the slope of the error times the posterior variance. For an error of any shape the shift is bounded by the range of the error over

the support, and the same slope, divided by the curvature of the likelihood, moves a maximum-likelihood estimate. The literature has judged these methods by the absolute error of the log-determinant, a measure that bounds the shift only loosely and does not predict it. Bivand et al. (2013) observed the Barry–Pace error we measure: on the U.S. census tracts its Jacobians lie above the exact values, its envelope excludes them for $\rho \geq 0.94$, and doubling the probes "has a minimal impact." Judged by absolute error, they concluded that "the problem only affects the extreme edge of the feasible range." Judged by its tilt, the same error rises steeply exactly where the posterior has mass, so it moves the estimate materially from $\rho = 0.85$ or 0.9 onward, well short of the edge at 0.99. Conversely, our matrix-free estimator carries an absolute error orders of magnitude larger than an exact method's, yet its error is nearly level across any realistic posterior. The result is not specific to spatial models; it governs any estimator whose log-density carries an approximated term.

Using our methods, a Bayesian spatial autoregressive model of all 85,230 census tracts of the United States spends 1.1 s per chain of 2,000 draws on the Jacobian. The fastest alternative that leaves the estimate unbiased over the full range of ρ is Pace–LeSage, which spends 4.5 min on contiguity weights and, run outside its specification, about 48.3 s on directed weights.

We proceed as follows. Section 2 sets out the methods in current use, and Section 3 derives what an error in the Jacobian does to an estimate, the criterion by which we judge every method that follows. Section 4 sets out the property of the Jacobian on which the first method rests, Section 5 and Section 6 develop the two methods, and Section 7 compares seven methods on synthetic weights, on the census tracts, and on dense weights built from address-level data. Section 8 concludes with the choice between the two methods.

2 Existing approaches

Every method in this literature divides its work into a precomputation that does not depend on ρ and an evaluation that does,

$$\mathcal{P} \;=\; \mathcal{P}(W), \qquad \widehat{\mathcal{J}}(\rho) \;=\; \mathcal{E}(\mathcal{P}, \rho), \tag{3}$$

and the methods differ in what $\mathcal{P}$ holds and in the variable it is computed in. They fall into the two branches Bivand et al. (2013) distinguish.

2.1 Exact computation

Eigenvalues. Ord (1975) diagonalizes W once, at $O(n^3)$ cost, and evaluates $\mathcal{J}(\rho) = \sum_i \log(1 - \rho\lambda_i)$ in $O(n)$ at every ρ thereafter. It remains the accuracy benchmark, but the cubic precomputation confines it to a few thousand observations. The most recent Bayesian spatial package, `bsreg` (Kuschnig, 2022), computes either a dense eigendecomposition or a grid of dense determinants, both $O(n^3)$.

Sparse factorization at every ρ. A sparse Cholesky or LU factorization of $I_n - \rho W$ gives $\mathcal{J}$ exactly from the diagonal of the factor, at a cost set by the nonzeros of the factor rather than by n^3. When W is row-standardized from a symmetric graph it is similar to a symmetric matrix (Ord, 1975), so the determinant is that of a symmetric positive definite matrix and Cholesky applies. The R ecosystem exploits this through CHOLMOD and `spam` (Chen et al., 2008; Furrer & Sain, 2010; Ng & Peyton, 1993), computing the symbolic analysis of the sparsity pattern once and repeating only the numeric factorization at each ρ (Bivand et al., 2013). PySAL's `spreg` (Rey & Anselin, 2007) calls a fresh SuperLU factorization, symbolic analysis included, at every ρ its optimizer visits when its sparse option is chosen, and otherwise forms W as a dense matrix. In both cases $\mathcal{P}$ holds at most a symbolic analysis, leaving the numerical work to the evaluation, so the number of factorizations equals the number of evaluations.

A grid in ρ. The Spatial Econometrics Toolbox (LeSage, 2021) precomputes $\mathcal{J}$ by sparse LU on a grid of roughly one hundred equispaced values of ρ and interpolates between them by spline during estimation (LeSage & Pace, 2009; Pace & Barry, 1997). The same grid serves its maximum-likelihood and Bayesian routines, which query it at each proposed ρ. The grid decouples the number of factorizations from the number of evaluations, but it fixes both the count and the interpolant by convention, independent of the function being approximated.

2.2 Approximation

Moments of the spectrum. Smirnov & Anselin (2001) computed the characteristic polynomial of W and evaluated $\mathcal{J}$ from its coefficients. Smirnov & Anselin (2009) replaced the polynomial with the m lowest moments of the spectrum, the traces $\operatorname{tr}(W^j)$, substituted them into the power series of $\mathcal{J}$, and interpolated the truncated remainder geometrically from the behavior of the highest moments. The method precomputes once and evaluates in microseconds, so it shares the cost structure the first method below adopts. Bivand et al. (2013) find its accuracy very good over most of the range of ρ, with absolute errors of 11 to 28 log-units near the bounds, where "the interpolation term is not offering sufficient correction."

Chebyshev expansion in the matrix argument. Pace & LeSage (2004) expand $\log(1 - \rho x)$ in Chebyshev polynomials and apply the expansion to the symmetrized weights,

$$\mathcal{J}(\rho) \;\approx\; \sum_{j=0}^{q} c_j(\rho)\, \operatorname{tr}\big(T_j(W)\big)\,, \tag{4}$$

where $c_j(\rho)$ are scalar coefficients and T_j is the Chebyshev polynomial of degree j. The traces are exact and cheap for $j \le 2$; higher orders require either sparse matrix powers, whose fill grows with the order, or Monte Carlo probes. Pace & LeSage (2004) specify the expansion for weights that are symmetric or similar to a symmetric matrix.

Monte Carlo traces. Barry & Pace (1999) estimate $\operatorname{tr}(W^j)$ by random probe vectors (Hutchinson, 1989) and substitute the estimates into the Taylor series $\log(1 - \rho x) = -\sum_j \rho^j x^j / j$. The estimator needs only matrix-vector products, so it scales to any W

that can be stored. It is the standard-bearer of the approximation branch for large and dense problems. Truncated at order p, the series converges at the rate $|\rho|^p$, which slows as ρ approaches one.

A parallel literature in numerical linear algebra estimates the log-determinant of a single large sparse matrix, by stochastic Chebyshev expansions (Han et al., 2017) or stochastic Lanczos quadrature (Ubaru et al., 2017). The second method below adapts the first of them to the one-parameter family $\{I_n - \rho W\}$ that spatial estimation poses, and to the question of what its error does to an estimate of ρ.

Interpolating the log-determinant across parameter values also has a precedent in Gaussian-process estimation. Dong et al. (2017) estimate it at a few values of the kernel hyperparameters by stochastic Lanczos quadrature and fit a cubic radial basis function surrogate between them, which the optimizer then queries in place of the log-determinant. The first method below shares that structure and differs from it in three respects: it interpolates exact values from sparse factorizations, where the surrogate interpolates stochastic estimates; it works in one scalar parameter whose singularities are known in advance; and it sets the number of nodes before any factorization runs, by the bias the interpolation leaves in the estimate.

2.3 What the two branches leave open

The exact branch spends its factorizations on evaluations or on a conventional grid, and extends symbolic reuse only to symmetrizable weights. The Toolbox's grid routines are documented for symmetric W and do not reuse a symbolic factorization even there; each precomputes a column ordering once and calls a fresh LU at every grid point. For asymmetric W its only path is the Monte Carlo estimator.[1] Directed weights, such as k-nearest-neighbor graphs, adaptive kernels, travel-time matrices, and origin-destination flows (LeSage & Pace, 2008), therefore meet the most expensive options in either branch. The approximation branch reaches large and dense problems, but its methods have been judged by absolute error, which Section 3 shows is the wrong measure of what the error does to an estimate.

3 Only the variation of the error reaches the estimate

The methods of Section 2 make errors of very different sizes, yet an estimate of ρ responds to the variation of an error near the estimate, not to its size. That variation sets the node count of our first method and allows our second a large absolute error. Write the error as $e(\rho) = \widehat{\mathcal{J}}(\rho) - \mathcal{J}(\rho)$. The approximate posterior is an exponential tilt of the exact one,

$$\hat{p}(\rho \mid y) \;\propto\; p(\rho \mid y)\, e^{\,e(\rho)}. \tag{5}$$

[1] The Toolbox routines `lndetfull` and `lndetint` compute the ordering with `colamd` once and factor with `lu` at every grid point; the asymmetric path is `fdet_mc3`, the estimator of Barry & Pace (1999).

If e is constant over the support of the posterior, the factor $e^{e(\rho)}$ cancels in the normalization and $\hat{p} = p$ exactly. Put differently, an approximation that is uniformly wrong by a hundred log-units gives the right answer. Only the variation of e across the region where the posterior has mass can move anything. We call that variation the tilt of the error.

Expanding Equation 5 to first order gives the shift the tilt induces in the posterior mean,

$$\hat{\mathbb{E}}[\rho] - \mathbb{E}[\rho] \;\approx\; \operatorname{Cov}\big(\rho, e(\rho)\big) \;\approx\; e'(\bar{\rho})\,\sigma_\rho^2, \tag{6}$$

so the shift in posterior standard deviations is $e'(\bar{\rho})\,\sigma_\rho$, the slope of the error at the posterior mean times the posterior's width. The expansion estimates the shift, but a bound on it needs no expansion. If e varies over a range R across the support of the posterior, the weight $e^{e(\rho)}$ lies in an interval whose width is $e^R - 1$ times its lower end, so the Cauchy–Schwarz inequality gives

$$\frac{\big|\hat{\mathbb{E}}[\rho] - \mathbb{E}[\rho]\big|}{\sigma_\rho} \;\le\; \frac{e^R - 1}{2} \;\approx\; \frac{R}{2} \tag{7}$$

for an error of any shape, monotone or oscillating.

The same mechanism governs a maximum-likelihood estimate. If the log-likelihood ℓ carries the error e, its maximizer moves to first order by $e'(\rho)/(-\ell''(\rho))$, the slope of the error at the estimate over the curvature of the likelihood: the tilt argument applied at a single point rather than across a posterior's support. The criterion therefore holds whether the estimator is a sampler or an optimizer.

Two consequences follow for the comparison of methods. First, the maximum absolute error over the interval, the measure by which the literature has compared these methods (Bivand et al., 2013), does not predict the effect on an estimate and bounds that effect only through Equation 7, a bound that becomes vacuous once the error exceeds a log-unit or so: a large error that is level across the posterior moves nothing, and a small one that is steep there can move the estimate by more than its standard error. Second, the shift in standard deviations scales with σ_ρ for a given slope, while the slope of a truncation error grows toward the singularity at $\rho = 1$. An error that is harmless in the interior can therefore bias the estimate near the boundary.

4 The Jacobian is a function of one variable

Writing $\lambda_1, \dots, \lambda_n$ for the eigenvalues of W,

$$\mathcal{J}(\rho) \;=\; \sum_{i=1}^{n} \log\left(1 - \rho\lambda_i\right), \tag{8}$$

a function holomorphic on the complex plane except for branch cuts emanating from the points $\rho = 1/\lambda_i$. For row-standardized W the spectral radius is one, so every

such point lies on or outside the unit circle; the Perron eigenvalue places one at $\rho = 1$, and a bipartite component, such as a rook lattice, places another at $\rho = -1$. On any closed subinterval of $(-1, 1)$ the function is therefore analytic, so classical approximation theory determines it to high accuracy from its values at a few well-placed nodes (Trefethen, 2013).

A factorization is needed to evaluate $\mathcal{J}$ at a point, but not to hold $\mathcal{J}$ as a function once its values at enough nodes are known. If a couple of dozen factorizations determine the function to within the accuracy estimation requires, the factorization count for an entire estimation run becomes a constant set by the function, whether the estimator visits a dozen values of ρ or a hundred thousand.

In the terms of Equation 3, the cost of such a method splits into a fixed setup, the number of factorizations times the cost of each, and a per-ρ evaluation of the interpolant that involves no matrix at all. Refactorization at every ρ sits at the opposite extreme, with no setup and a full factorization at every evaluation. Grid and spline precomputes in ρ as well, but its count is a convention; the method below precomputes in ρ and lets the function set the count. The moment and series methods, by contrast, precompute in the spectrum of W, so the accuracy they deliver depends on the speed with which a spectral series decays for the W at hand. Their setup, sparse powers of W, grows with the walk structure of the graph.

5 Method 1: factorize and interpolate

The first method reconstructs $\mathcal{J}$ from exact values at a set of nodes. It involves three choices: the factorization that computes each value, the interpolant that recombines the values, and the number of nodes.

5.1 Factorization with symbolic reuse

Each node costs one sparse factorization of $I_n - \rho W$, whose factors give the log-determinant from their diagonals. The sparsity pattern of $I_n - \rho W$ depends on W alone, so one symbolic analysis, the fill-reducing ordering and the structure of the factors, serves every node: n_c nodes cost one symbolic analysis and n_c numeric factorizations. When W is row-standardized from a symmetric graph, we factor the similar symmetric matrix by Cholesky, as the R ecosystem does (Bivand et al., 2013; Ord, 1975; Pace & LeSage, 2004).[2] Cholesky computes one triangular factor where LU computes two, so it takes about half the work wherever both apply (Golub & Van Loan, 2013).

Directed weights admit no such similarity. Current practice refactors them from scratch, yet the machinery that separates symbolic from numeric factorization exists for LU as well. KLU (Davis & Palamadai Natarajan, 2010), developed for circuit

[2]The similarity needs the degrees of the original graph. We recover them from W alone, up to one scale per connected component, by propagating the ratios $D_i/D_j = W_{ji}/W_{ij}$ in a breadth-first traversal of $O(\text{nnz})$ work; the per-component scale does not affect the determinant.

simulation, reuses both the symbolic analysis and the pivot sequence of a first factorization, so each later factorization of the same pattern is numeric work only, exactly as CHOLMOD's is. We know of no use of it in spatial econometrics. With it we extend symbolic reuse to directed W: k-nearest-neighbor graphs, adaptive kernels, and flow matrices factor once symbolically and then at the cost of a numeric refactorization per node.

5.2 Rational interpolation in ρ

The singularities of Equation 8 are logarithmic branch points at $\rho = 1/\lambda_i$, the nearest at $\rho = 1$ and at or beyond $\rho = -1$, a hundredth or more beyond the ends of the working interval $[-0.99, 0.99]$. Polynomial interpolation converges geometrically there, but at a rate that collapses as the singularity approaches the interval, and degrades to algebraic convergence once it reaches it. Rational functions, with poles placed along the branch cut, retain root-exponential convergence in the number of terms even then, so they degrade only mildly as the interval approaches ± 1 (Trefethen, 2013). The AAA algorithm (Nakatsukasa et al., 2018) places those poles without being told where the singularities lie. It represents the approximant in barycentric form (Berrut & Trefethen, 2004),

$$r(\rho) = \sum_{j=1}^{s} \frac{w_j f_j}{\rho - \rho_j} \Big/ \sum_{j=1}^{s} \frac{w_j}{\rho - \rho_j}\,, \tag{9}$$

where the ρ_j are support points, $f_j = \mathcal{J}(\rho_j)$ exact values from the factorizer, and w_j barycentric weights. At each iteration the algorithm adds as a support point the sample of largest residual and recomputes the weights as the singular vector of a Loewner matrix built from the remaining samples. We factor at n_c Chebyshev-spaced nodes and let AAA select its support points and screen its residuals on those same nodes, so the selection costs nothing beyond the factorizations already spent. Evaluation at any ρ is then $O(s)$ scalar arithmetic, with $s \le n_c/2$, and involves no matrix. Because support points are selected on the nodes themselves, we check the interpolant between nodes by the bias it induces (Section 5.3). We remove any spurious pole AAA places inside the interval, since it would produce a spike between nodes.

For directed W the eigenvalues are complex. They come in conjugate pairs, so $\log|I_n - \rho W|$ remains real-valued. Every $1/\lambda_i$ still lies on or outside the unit circle, so the function remains analytic on the open interval. The ellipse that governs convergence on $[-0.99, 0.99]$ touches the unit circle only at ± 1, so the node rule below applies to directed weights unchanged. The bias it leaves there is as small as on undirected weights (Table 2).

A cubic spline on an equispaced grid, the interpolant of current practice, has no such adaptation. It is a fixed, local, low-order fit whose error swings in sign and magnitude with the position of a point between two knots. Nothing in its construction relates the number of knots to the accuracy it delivers, so practice defaults to a large conventional

grid. The two interpolants therefore differ less in their accuracy at a given count than in whether the count can be set in advance from the interval, which for AAA it can.

5.3 The number of nodes

We set the number of factorizations by the bias the interpolation error induces in the estimate of ρ. By Section 3, an error e in $\mathcal{J}$ moves the maximizer of the likelihood to first order by $e'(\rho)/(-\ell''(\rho))$. The error is n times a per-observation error whose size at a fixed number of nodes barely changes with n, and the curvature $-\ell''$ is n times the information one observation carries about ρ. The two factors of n cancel, so the bias in $\hat{\rho}$ depends only on the interval, the structure of W, the number of nodes, and the per-observation information: in units of ρ it is the same at every sample size. We require it to stay below 10^{-4}, an order of magnitude finer than the three decimals to which estimates of ρ are commonly reported. In units of the estimate's standard deviation the same bias grows as $\sqrt{n}$, since the standard deviation shrinks, so we report it in those units as well.

To measure it, we simulate from Equation 1 and maximize the concentrated likelihood twice on the same data, once with exact factorizations and once with the interpolant, across 350 datasets: seven symmetrizable weights matrices (rook lattices at $n = 2{,}500$, 10,000 and 40,000, and queen and distance-band weights on California block groups and on the census tracts of the United States), five values of ρ from -0.95 to 0.98, two noise levels, and five seeds. Table 1 reports the largest bias at each node count on the full interval $[-0.99, 0.99]$.

Table 1: Largest absolute bias AAA induces in the maximum-likelihood estimate of ρ on $[-0.99, 0.99]$, against exact factorization on the same data, in units of ρ and of its sampling standard deviation σ_ρ. Maximum over seven symmetrizable matrices, five values of ρ from -0.95 to 0.98, two noise levels, and five seeds.

Nodes	Largest bias, in ρ	Largest bias, in σ_ρ
8	0.088	36
10	0.024	9.4
12	1.9×10^{-3}	5.1
14	5.1×10^{-4}	0.888
16	3.0×10^{-4}	0.298
18	6.9×10^{-5}	0.070
20	3.3×10^{-5}	0.101
24	8.4×10^{-6}	0.036
32	4.5×10^{-7}	1.2×10^{-3}
48	6.2×10^{-8}	1.9×10^{-4}
96	8.2×10^{-8}	1.9×10^{-4}

AAA meets the tolerance from 18 nodes onward. We fix the count by a rule calibrated on this sweep, $n_c = \lceil 3.4/\ln \chi \rceil$, clamped to $[20, 96]$, where $\chi = |t| + \sqrt{t^2 - 1}$ is the

Bernstein parameter of the interval, t being the image of the nearer singularity at ± 1 under the map of the interval onto $[-1, 1]$ (Trefethen, 2013). The rule depends on the interval alone, so it fixes the count before any factorization runs. Its constant, 3.4, is chosen so that the full interval receives 24 nodes, a margin above the 18 the tolerance requires, since the error of any interpolant concentrates near ± 1. At 24 nodes the largest bias is 8.4×10^{-6}, under a tenth of the tolerance. It holds steady across fragmentation: on the distance-band tracts, with 39,906 connected components, it is 8.1×10^{-6}. In standard deviations the largest bias is 0.036, on those tracts at $\rho = 0.98$, where the sampling standard deviation is smallest; growing as $\sqrt{n}$, it would reach 0.05 standard deviations near $n = 160{,}000$ on those weights, where a larger count restores the margin at the cost of a few more factorizations. An entire estimation run therefore spends 24 factorizations, where grid and spline spends a hundred and refactorization spends one per evaluation.

Every node remains a sparse factorization, so the method reaches exactly as far as the factorization does. When the fill of the factors exhausts memory or time, no interpolant helps, so the second method takes over.

6 Method 2: matrix-free stochastic Chebyshev

Factorization grows costly on weights that are dense or that carry dense blocks: stars and hubs, co-located units joined into cliques by a distance band, kernels with wide bandwidths. Such weights arise routinely in address-level microdata. On housing-unit weights for one metropolitan rental market, where a median of 25 units and as many as 1,068 share a street address, the co-located units form cliques under distance-based weights, the largest forcing a dense front of 1,068 rows in any sparse factorization. The work of a factorization then grows with the cube of the largest clique and its memory with the square, so at some size the factorization no longer completes at all. For such weights matrix-free techniques become the faster route or the only one; the standard among them is Barry–Pace.

The second method keeps the matrix-free structure of Barry & Pace (1999) and replaces what it estimates. We rescale W so that its spectrum lies in $[-1, 1]$,

$$\tilde{W} = \frac{2W - (\lambda_{\max} + \lambda_{\min}) I_n}{\lambda_{\max} - \lambda_{\min}}, \qquad \mathcal{J}(\rho) \approx \sum_{j=0}^{p} c_j(\rho)\, \mu_j, \quad \mu_j = \operatorname{tr}\big(T_j(\tilde{W})\big), \tag{10}$$

where the $c_j(\rho)$ are the Chebyshev coefficients of $\log|a(\rho) - b(\rho)x|$ on $[-1, 1]$, with $a = 1 - \rho(\lambda_{\max} + \lambda_{\min})/2$ and $b = \rho(\lambda_{\max} - \lambda_{\min})/2$. The bracket $[\lambda_{\min}, \lambda_{\max}]$ must enclose the spectrum, since T_j grows without bound outside $[-1, 1]$; for row-standardized W the Perron eigenvalue and Gershgorin's theorem give the enclosing bracket $[-1, 1]$, so $\tilde{W} = W$. The moments μ_j are independent of ρ, since the entire dependence on ρ sits in the scalar coefficients. We estimate them once, with random probes, through the three-term recurrence $v_{j+1} = 2\tilde{W} v_j - v_{j-1}$, at a cost of p batched sparse matrix-vector products, identical in structure and cost to Barry–Pace (Han et al., 2017). Each

evaluation then recovers the coefficients by a discrete cosine transform and combines them with the stored moments, with no matrix work.

The construction differs from Barry–Pace in three respects.

The truncation rate. On the bracket $[-1, 1]$ the Chebyshev coefficients decay as $\chi(\rho)^{-j}$ with $\chi(\rho) = 1/|\rho| + \sqrt{1/\rho^2 - 1}$, from 3.7^{-p} at $|\rho| = 0.5$ to 1.15^{-p} at $|\rho| = 0.99$. The Taylor series of Barry–Pace also converges geometrically, but at the strictly slower rate $|\rho|^p$, so at a fixed order its truncation error grows steeply as ρ approaches one. Section 3 shows that the steepness, not the size, reaches an estimate. The advantage holds for negative ρ as well, since $\chi(\rho)^{-1} < |\rho|$ for either sign.

Exact low-order moments. The traces of low powers of W are computable exactly in sparse work, $\operatorname{tr}(W^4)$ from one sparse product and $\operatorname{tr}(W^6)$ from two, using $\operatorname{tr}(AB) = \langle A, B^\top \rangle$. Writing d for the depth taken exactly,

$$\widehat{\mathcal{J}}(\rho) \;=\; \sum_{j=0}^{d} c_j(\rho)\, \mu_j \;+\; \sum_{j=d+1}^{p} c_j(\rho)\, \hat{\mu}_j, \tag{11}$$

where the first sum is exact and the second estimated by probes. Pace & LeSage (2004) take the lowest traces exactly and truncate; the estimator here takes the same traces exactly and estimates the tail by probes to high order. Removing the lowest moments from the probe estimate removes the largest coefficients from its variance and leaves the probes to cover only the tail, a deflation of the kind Gambhir et al. (2017) use to reduce the variance of trace estimators. We take $d = 4$, which needs one sparse product W^2, and reduce the depth above a mean degree of 16, where the fill of W^2 would dominate the setup.

Probes fixed once per fit. If an estimator draws fresh probes at every call, the Jacobian it returns is a random function of ρ, so a sampler built on it targets no fixed posterior. The pseudo-marginal construction (Andrieu & Roberts, 2009) requires an unbiased estimate of the likelihood. Probes estimate $\mathcal{J}$, however, while the likelihood contains $\exp(\mathcal{J})$, whose plug-in estimate Jensen's inequality biases; debiasing it is possible in principle (Lyne et al., 2015) but costly and unstable. We therefore draw the probes once per fit, from a recorded seed, so that within a fit $\widehat{\mathcal{J}}$ is a deterministic function of ρ, the sampler targets one well-defined posterior, and the fit is reproducible. The residual probe error then acts on that posterior as a fixed perturbation, whose effect Section 3 quantifies. Fixing probes across a family of matrices is established practice in Gaussian-process estimation (Aune et al., 2014; Dong et al., 2017; Gardner et al., 2018).

Beyond the one sparse product for the exact moments, the setup is linear in the nonzeros of W and involves no fill. The truncation theory above presumes a real spectrum. On directed W the eigenvalues are complex and the series converges only for those inside the Bernstein ellipse, so the method runs unchanged but without that guarantee. Its behavior there rests on the measurements of Section 7.

7 Experiments

The tilt argument of Section 3 predicts where each method's bias should appear. The Barry–Pace truncation converges at the rate $|\rho|^p$, so its error rises steeply toward $\rho = 1$. The dropped terms are positive for $\rho > 0$, so the error is also one-signed: near the boundary, where the truncation dominates the probe noise, it should bias $\hat{\rho}$ upward on every draw of its probes. The remainder that Smirnov–Anselin interpolate is accurate in the interior and fails near the bounds, so its bias should appear only there. A spline's error depends on where the posterior falls between two knots, so its tilt should be small in the interior and uncontrolled near ± 1, where the logarithmic singularity defeats a local cubic. The Chebyshev truncation of stochastic Chebyshev decays at the faster Bernstein rate, so its error should be nearly level across any realistic posterior. Its probe error, zero-mean over the draw of probes, should leave a bias small and centered near zero. An interpolant whose node count holds the bias of Equation 6 below tolerance on the whole interval should leave the estimate unmoved everywhere.

7.1 Design

We compare our two methods against five in current use, each run as its literature specifies it or as current software practices it, with symbolic reuse credited only where a method specifies it. Factorize-and-interpolate factors symmetrizable W by Cholesky and directed W by KLU, at the 24 nodes its rule sets (Section 5.3). Stochastic Chebyshev draws 50 probes, fixed once per fit, with exact moments to depth $d = 4$ and the order set by the bound its tail coefficients place on the truncation (Section 6).

From the approximation branch we take the three approximations Bivand et al. (2013) compare. We run Barry–Pace (Barry & Pace, 1999) at Taylor order 30, the order Bivand et al. (2013) used, with 100 probes, so the two probe-based methods differ in order as well as in basis. We run Smirnov–Anselin (Smirnov & Anselin, 2009) in the simplified form Bivand et al. (2013) benchmarked, with $m = 30$ exact moments and the geometric remainder. We run Pace–LeSage (Pace & LeSage, 2004) with exact traces through order $q = 30$, computed by sparse recurrence to match the order of Smirnov–Anselin. Bivand et al. (2013) benchmarked it at $q \leq 5$, so $q = 30$ is the method's best case. Pace & LeSage (2004) specify the expansion for symmetrizable weights; because its trace recurrence needs only products with W, we also run it on the directed weights, outside that specification.

From the exact branch we take grid-and-spline interpolation (LeSage & Pace, 2009; Pace & Barry, 1997) and refactorization at every ρ. For the first we factor 100 equispaced knots by a fresh sparse LU each and fit a cubic spline between them. The Toolbox likewise refactors at every grid point, reusing only a precomputed column ordering; a fresh factorization is the closest match to it available outside MATLAB. For refactorization we follow the R ecosystem on symmetrizable W, repeating a CHOLMOD numeric factorization at each ρ with the symbolic analysis reused (Bivand et al., 2013),

and factor directed W by a fresh sparse LU, since no incumbent specifies reuse there. On the tracts we also time the fresh SuperLU factorization that `spreg` calls at every ρ.

We use four weights matrices. Two are synthetic, at $n = 10{,}000$: rook contiguity on a regular polygon grid, and directed k-nearest neighbors with $k = 6$ on uniform random points. Two are empirical, on all 85,230 census tracts of the United States in 2020: queen contiguity, with 530,892 nonzeros, and directed k-nearest neighbors with $k = 5$. We build all of them with `libpysal` (Rey & Anselin, 2007) and row-standardize them. Ground truth is a fresh sparse LU factorization at every ρ.

We report three things for each method: (1) its maximum absolute error on a 33-point grid over $[-0.99, 0.99]$, the literature's evaluator; (2) its tilt, the least-squares slope of the signed error over a window of $\pm 2h$ about $\bar{\rho}$, multiplied by a reference width $h = 0.01$, which by Equation 6 is the shift in standard deviations the error would induce in a posterior of that width; and (3) the bias it induces in the posterior mean of ρ directly. For the last, we simulate one dataset from Equation 1 at each true value of ρ from 0.3 to 0.95, with two covariates and unit error variance, integrate β and σ_ε^2 out of the posterior under flat priors, and compute the marginal posterior of ρ on a grid of step 0.0005 spanning ± 0.04 about the true value, once with the exact Jacobian and once tilted by each method's error curve (Equation 5). The bias therefore belongs to the Jacobian alone, free of sampler noise. For the two probe-based methods the error curve depends on the probes, so we run each over 8 independent probe draws in the sweeps across ρ, positive and negative, and over 16 in the focused comparisons of Section 7.3. We judge a bias material when it approaches the Monte Carlo standard error of the chain the estimate would come from, 0.02 to 0.05 posterior standard deviations for a 2,000-draw chain in these designs. Timings use a stream of distinct values of ρ on one machine (Apple M3 Max), in double precision.

7.2 The bias each method induces in ρ

Table 2 and Figure 1 report the bias each method induces in the posterior mean of ρ, in posterior standard deviations, as the true value of ρ moves toward the boundary. For the two probe-based methods each cell is the mean over 8 probe draws.

Table 2: Bias in the posterior mean of ρ, in posterior standard deviations, by true ρ (columns). Marginal posterior of ρ tilted by each method's error curve; for stochastic Chebyshev and Barry–Pace, the mean over 8 probe draws. On the directed k-NN weights Pace–LeSage runs outside its specification.

Weights	Method	0.3	0.5	0.7	0.85	0.9	0.95
rook, $n =$ 10,000	Factorize and interpolate (AAA)	0.000	0.000	0.000	0.000	0.000	0.000

Weights	Method	0.3	0.5	0.7	0.85	0.9	0.95
rook, $n = 10{,}000$	Pace–LeSage ($q = 30$)	0.000	0.000	0.000	0.000	0.000	0.000
rook, $n = 10{,}000$	Stochastic Chebyshev (50 probes)	0.000	0.000	-0.002	-0.006	-0.010	-0.019
rook, $n = 10{,}000$	Grid and spline (100 knots)	0.000	0.000	0.000	0.000	-0.001	+0.048
rook, $n = 10{,}000$	Smirnov–Anselin ($m = 30$)	0.000	0.000	0.000	0.000	+0.005	+0.065
rook, $n = 10{,}000$	Barry–Pace	+0.001	+0.004	+0.007	+0.022	+0.095	+0.547
k-NN, $n = 10{,}000$	Factorize and interpolate (AAA)	0.000	0.000	0.000	0.000	0.000	0.000
k-NN, $n = 10{,}000$	Pace–LeSage ($q = 30$), directed	0.000	0.000	0.000	0.000	0.000	0.000
k-NN, $n = 10{,}000$	Stochastic Chebyshev (50 probes)	0.000	0.000	-0.001	-0.002	-0.003	-0.005
k-NN, $n = 10{,}000$	Grid and spline (100 knots)	0.000	0.000	0.000	0.000	-0.003	+0.051

Weights	Method	0.3	0.5	0.7	0.85	0.9	0.95
k-NN, n = 10,000	Smirnov–Anselin ($m = 30$)	0.000	0.000	0.000	+0.001	+0.005	+0.065
k-NN, n = 10,000	Barry–Pace	-0.001	-0.002	-0.004	+0.019	+0.135	+0.790
queen tracts, n = 85,230	Factorize and interpolate (AAA)	0.000	0.000	0.000	0.000	0.000	0.000
queen tracts, n = 85,230	Pace–LeSage ($q = 30$)	0.000	0.000	0.000	0.000	0.000	0.000
queen tracts, n = 85,230	Stochastic Chebyshev (50 probes)	0.000	0.000	-0.001	-0.002	-0.003	-0.004
queen tracts, n = 85,230	Grid and spline (100 knots)	0.000	0.000	0.000	-0.001	-0.003	+0.126
queen tracts, n = 85,230	Smirnov–Anselin ($m = 30$)	0.000	0.000	0.000	+0.001	+0.013	+0.176
queen tracts, n = 85,230	Barry–Pace	-0.001	-0.004	-0.009	+0.021	+0.217	+1.428

Factorize-and-interpolate leaves the estimate unmoved in every cell, on symmetrizable and directed weights alike, and so does Pace–LeSage at its best-case order where it applies. The remaining methods differ in where their bias appears, as Section 3 predicts. Barry–Pace is negligible in the interior, lifts off at $\rho = 0.85$, and is material from $\rho = 0.9$ on all three matrices, where it reaches +0.095, +0.135, and +0.217 standard deviations. At $\rho = 0.95$ it reaches +1.428 on the tracts, displacing the estimate by more

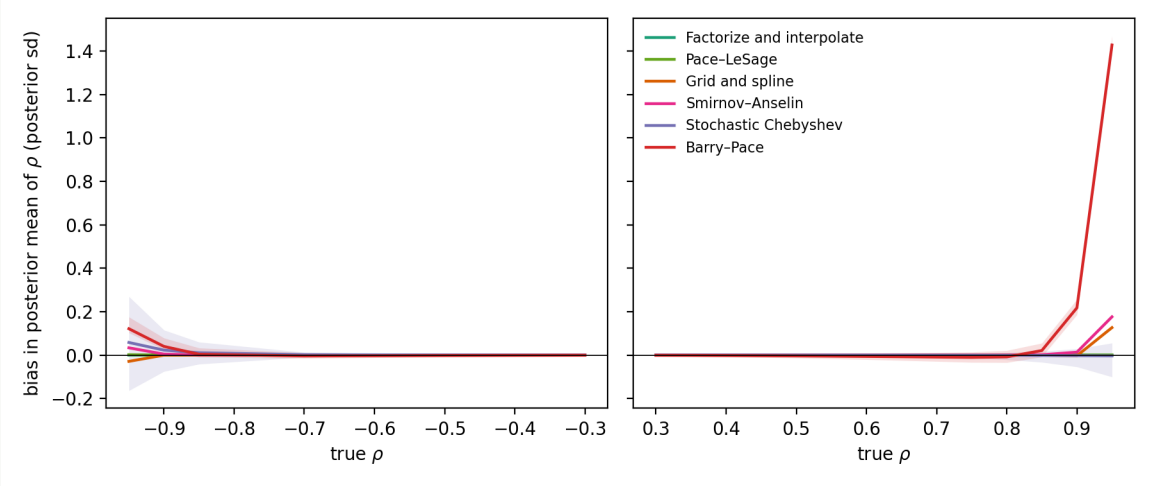


Figure 1: Bias in the posterior mean of ρ by true ρ on the queen-contiguity census tracts (n = 85,230), negative dependence at left and positive at right. Stochastic Chebyshev and Barry–Pace are drawn as the mean over 8 probe draws, with a band spanning the draws. Factorize-and-interpolate and Pace–LeSage lie on zero throughout; Barry–Pace lifts off near either boundary, and Smirnov–Anselin and grid and spline follow, more modestly, at ± 0.95.

than its own posterior standard deviation. Smirnov–Anselin and grid and spline hold to the interior and lift off at $\rho = 0.95$, to +0.176 and +0.126 standard deviations on the tracts.

Stochastic Chebyshev holds its mean bias to 0.019 standard deviations or less in every cell, the largest at $\rho = 0.95$ on the rook lattice, where Barry–Pace's is +0.547. The mean lies below the Monte Carlo error of the chain any such estimate would come from, so it is undetectable in practice; the method's two settings, its probes and its order, reduce it further if an analysis requires. A single fit draws one set of probes, so the bias a given fit carries is one draw from a distribution around that mean; on the tracts at $\rho = 0.9$ it ranges from -0.054 to +0.029 across 8 draws. The number of probes controls that spread (Section 7.3).

Run on the directed k-NN weights, outside the symmetrizable weights for which it is specified, Pace–LeSage at its best-case order is as exact as on symmetrizable weights. The trace recurrence needs only products with W. At $q = 30$ its truncation is invisible here. Its obstacle on directed weights is therefore cost, which Section 7.6 takes up, and not accuracy.

7.3 Barry–Pace's bias is systematic; stochastic Chebyshev's is below Monte Carlo noise

Barry–Pace's error has two components that behave differently across probe draws: the Taylor truncation, present in every draw and positive for $\rho > 0$, and the probe noise, zero-mean around it. Across 16 independent probe draws at $\rho = 0.9$, the bias of Barry–Pace ranges from +0.048 to +0.151 standard deviations on the rook lattice and from +0.186 to +0.312 on the tracts. Every draw is biased upward. More probes do not

remove the bias, because the truncation carries it. Bivand et al. (2013) observed the source of this bias on the same tracts: the Monte Carlo Jacobians lie above the exact values near the upper bound, and doubling the probes "has a minimal impact on the problem." The bias follows from that error, once its tilt rather than its size is taken as the measure.

The bias belongs to the Taylor basis, not to the order at which Barry–Pace is truncated. Table 3 runs Barry–Pace at the order stochastic Chebyshev selects on the same weights, 42 on the synthetic matrices and 48 on the tracts, with 100 probes and with the 50 that match stochastic Chebyshev's matrix-vector products, over 16 probe draws each. The higher order roughly halves the bias, but at $\rho = 0.95$ it remains +0.232, +0.351, and +0.419 standard deviations, against -0.009, -0.004, and -0.005 for stochastic Chebyshev at the same order and budget. A Taylor series in ρ converges at $|\rho|^p$, so near the boundary no practical order closes the gap to the Chebyshev rate.

Table 3: Mean bias in the posterior mean of ρ, in posterior standard deviations, over 16 probe draws, by true ρ; the last column gives the range across draws at 0.95. p is the order stochastic Chebyshev selects on each matrix.

Weights	Method	0.85	0.9	0.95	Range at 0.95
rook	Barry–Pace, order 30, 100 probes	+0.018	+0.091	+0.544	+0.50 to +0.61
rook	Barry–Pace, order p, 100 probes	+0.007	+0.024	+0.232	+0.18 to +0.30
rook	Barry–Pace, order p, 50 probes	-0.001	+0.016	+0.224	+0.07 to +0.35
rook	Stochastic Chebyshev, order p, 50 probes	-0.003	-0.005	-0.009	-0.05 to +0.03
k-NN	Barry–Pace, order 30, 100 probes	+0.024	+0.140	+0.793	+0.74 to +0.87
k-NN	Barry–Pace, order p, 100 probes	+0.003	+0.033	+0.351	+0.29 to +0.43
k-NN	Barry–Pace, order p, 50 probes	+0.007	+0.035	+0.351	+0.27 to +0.48

Weights	Method	0.85	0.9	0.95	Range at 0.95
k-NN	Stochastic Chebyshev, order p, 50 probes	-0.001	-0.002	-0.004	-0.04 to +0.04
queen tracts	Barry–Pace, order 30, 100 probes	+0.045	+0.246	+1.460	+1.40 to +1.53
queen tracts	Barry–Pace, order p, 100 probes	+0.009	+0.033	+0.419	+0.35 to +0.49
queen tracts	Barry–Pace, order p, 50 probes	+0.012	+0.039	+0.429	+0.33 to +0.51
queen tracts	Stochastic Chebyshev, order p, 50 probes	-0.002	-0.003	-0.005	-0.10 to +0.06

Stochastic Chebyshev's probe count, by contrast, governs a spread around a mean near zero. In a sweep over 12 to 200 probes at $\rho \in \{0.5, 0.8, 0.9, 0.95\}$, 16 draws each, the mean bias stays within 0.01 standard deviations of zero from 25 probes on, while the spread narrows as the number of probes grows. At $\rho = 0.5$ and 0.8, even 12 probes keep every fit within about 0.03 standard deviations. Near the boundary the spread grows: at $\rho = 0.95$ on the tracts, the largest bias across 16 draws is 0.101 standard deviations at 50 probes and 0.034 at 100. A probe costs setup time only, since each evaluation sums a Chebyshev series whose length does not depend on the number of probes. We therefore take 50 probes as the default and 100 when the posterior sits at or beyond $\rho = 0.9$, which holds every fit in this sweep within 0.034 standard deviations; strong negative dependence and weights with hubs call for more (Section 7.4, Section 7.7). The comparisons use 50 throughout.

Fixed probes make these distributions checkable. Left unseeded, as in its current implementations, Barry–Pace returns a different Jacobian on every run. In full Markov chain Monte Carlo fits of a design with a wide posterior (rook lattice, $n = 2{,}500$, $\rho = 0.9$), holding the data and the sampler's seed fixed, five draws of Barry–Pace probes shifted the posterior mean away from that of the exact Jacobian by 0.34 to 0.57 standard deviations, so that two analyses of the same data report visibly different posteriors. The shift is larger there than on the rook lattice of Table 2 because the posterior is wider: Equation 6 scales the shift with σ_ρ.

7.4 Negative spatial dependence

Negative dependence is less common in applied work than positive dependence, but it arises, for instance where neighboring units compete (Elhorst & Zigova, 2014). The lower boundary of the stability region poses the same problem as the upper. Table 4 repeats the sweep of Table 2 for ρ from -0.3 to -0.95.

Table 4: Bias in the posterior mean of ρ, in posterior standard deviations, at negative true ρ (columns). Stochastic Chebyshev and Barry–Pace, the mean over 8 probe draws. The rook lattice is bipartite, so its Jacobian has a singularity at -1; the other matrices do not.

Weights	Method	-0.3	-0.5	-0.7	-0.85	-0.9	-0.95
rook, $n = 10{,}000$	Factorize and interpolate (AAA)	0.000	0.000	0.000	0.000	0.000	0.000
rook, $n = 10{,}000$	Pace–LeSage ($q = 30$)	0.000	0.000	0.000	0.000	0.000	0.000
rook, $n = 10{,}000$	Stochastic Chebyshev (50 probes)	0.000	0.000	-0.001	-0.004	-0.005	-0.009
rook, $n = 10{,}000$	Grid and spline (100 knots)	0.000	0.000	0.000	0.000	0.000	-0.060
rook, $n = 10{,}000$	Smirnov–Anselin ($m = 30$)	0.000	0.000	0.000	-0.001	-0.006	-0.072
rook, $n = 10{,}000$	Barry–Pace	+0.002	+0.005	+0.011	0.000	-0.081	-0.564
k-NN, $n = 10{,}000$	Factorize and interpolate (AAA)	0.000	0.000	0.000	0.000	0.000	0.000

Weights	Method	-0.3	-0.5	-0.7	-0.85	-0.9	-0.95
k-NN, n = 10,000	Pace–LeSage (q = 30), directed	0.000	0.000	0.000	0.000	0.000	-0.004
k-NN, n = 10,000	Stochastic Chebyshev (50 probes)	0.000	0.000	-0.002	-0.010	-0.021	-0.061
k-NN, n = 10,000	Grid and spline (100 knots)	0.000	0.000	0.000	0.000	0.000	0.000
k-NN, n = 10,000	Smirnov–Anselin (m = 30)	0.000	0.000	0.000	0.000	+0.003	+0.042
k-NN, n = 10,000	Barry–Pace	-0.001	-0.002	-0.004	+0.005	+0.046	+0.227
queen tracts	Factorize and interpolate (AAA)	0.000	0.000	0.000	0.000	0.000	0.000
queen tracts	Pace–LeSage (q = 30)	0.000	0.000	0.000	0.000	0.000	+0.002
queen tracts	Stochastic Chebyshev (50 probes)	0.000	0.000	+0.002	+0.011	+0.023	+0.058
queen tracts	Grid and spline (100 knots)	0.000	0.000	0.000	0.000	0.000	-0.029

Weights	Method	-0.3	-0.5	-0.7	-0.85	-0.9	-0.95
queen tracts	Smirnov–Anselin ($m = 30$)	0.000	0.000	0.000	+0.001	+0.004	+0.033
queen tracts	Barry–Pace	-0.001	-0.002	-0.004	+0.004	+0.040	+0.121

The pattern of Table 2 holds with one difference of sign. Factorize-and-interpolate leaves the estimate unmoved in every cell, and Pace–LeSage nearly so. Barry–Pace approaches or exceeds the materiality standard at $\rho = -0.9$ and exceeds it on every matrix at −0.95. Its bias is systematic: on the rook lattice every one of its 8 draws at $\rho = -0.95$ lies between -0.636 to -0.514 standard deviations. There the sign is reversed because the bipartite lattice's Jacobian is symmetric in ρ, so the slope of the error, and with it the bias, changes sign with ρ. Smirnov–Anselin and grid and spline again hold to the interior and lift off at the boundary.

Stochastic Chebyshev's cells at $\rho = -0.95$ on the non-bipartite matrices, -0.061 and +0.058, are means over 8 draws of a wide spread. On these weights the posterior of ρ near −1 is several times wider than near +1, so each draw shifts the estimate further: Equation 6 scales the shift with that width. Repeated over more draws on the k-NN weights, the mean is indistinguishable from zero at every probe count, 0.001 ± 0.0367 at 50 probes and 0.0071 ± 0.0094 at 800, while the spread of a single fit falls with the square root of the number of probes, from 0.2076 standard deviations at 50 probes to 0.0377 at 800 and 0.0168 at 3,200. Strong negative dependence on such weights therefore calls for several hundred probes rather than 50. The additional probes cost setup time only.

7.5 Absolute error does not predict the bias

Table 5 sets the literature's evaluator beside the tilt on the two synthetic matrices.

Table 5: Maximum absolute error on a 33-point grid over [−0.99, 0.99] and tilt at $\bar{\rho} = 0.9$ (slope of the signed error over $\bar{\rho} \pm 2h$, times $h = 0.01$), $n = 10{,}000$. Stochastic Chebyshev and Barry–Pace at one probe draw.

Method	Max. error, rook	Max. error, k-NN	Tilt at 0.9, rook	Tilt at 0.9, k-NN
Factorize and interpolate (AAA)	0.054	0.048	-6.5×10^{-5}	-6.3×10^{-6}
Pace–LeSage ($q = 30$)	0.035	—	-5.7×10^{-7}	—

Method	Max. error, rook	Max. error, k-NN	Tilt at 0.9, rook	Tilt at 0.9, k-NN
Stochastic Chebyshev (50 probes)	2.9	0.52	0.030	-2.3×10^{-3}
Grid and spline (100 knots)	0.012	0.031	4.3×10^{-3}	0.011
Smirnov–Anselin ($m = 30$)	13	25	0.023	0.035
Barry–Pace	60	125	0.65	0.91

The two measures rank the methods differently. Grid and spline has the smallest maximum error in the table, yet a tilt at $\rho = 0.9$ about 70 times factorize-and-interpolate's on the rook lattice and 1,700 times on the k-NN weights, and a bias at $\rho = 0.95$ that factorize-and-interpolate does not have. Stochastic Chebyshev and Smirnov–Anselin have tilts of the same order on the rook lattice, but on the k-NN weights stochastic Chebyshev's is fifteen times the smaller, though its maximum error is fifty times smaller as well. The maximum errors sit near the ends of the interval, between the knots for grid and spline and at ± 0.99 for the others, while the tilt measures the error where the posterior has mass. One probe draw cannot show the sign of the tilt across draws: that of Barry–Pace is positive on every draw at $\rho = 0.9$, while that of stochastic Chebyshev changes sign from one draw to the next (Section 7.3).

Table 6 tests the first-order account of Section 3 against the measured shifts at $\rho = 0.9$. Each predicted shift is the slope of the error over the posterior, times the posterior's own standard deviation (Equation 6), and each bound is Equation 7 with the range of the error over the posterior's support.

Table 6: Predicted and measured shift in the posterior mean of ρ at $\rho = 0.9$, in posterior standard deviations, and the range bound of Equation 7. One probe draw for stochastic Chebyshev and Barry–Pace.

Weights	Method	Predicted shift	Measured shift	Bound $(e^R - 1)/2$
rook, $n = 10{,}000$	Factorize and interpolate (AAA)	-0.0000	-0.0000	1.2×10^{-4}
rook, $n = 10{,}000$	Pace–LeSage ($q = 30$)	-0.0000	-0.0000	1.5×10^{-6}

Weights	Method	Predicted shift	Measured shift	Bound $(e^R - 1)/2$
rook, $n = 10{,}000$	Stochastic Chebyshev (50 probes)	+0.0073	+0.0072	0.0626
rook, $n = 10{,}000$	Grid and spline (100 knots)	+0.0009	-0.0010	0.0169
rook, $n = 10{,}000$	Smirnov–Anselin ($m = 30$)	+0.0055	+0.0048	0.0495
rook, $n = 10{,}000$	Barry–Pace	+0.1585	+0.1513	6.17
queen tracts	Factorize and interpolate (AAA)	+0.0000	+0.0000	1.5×10^{-4}
queen tracts	Pace–LeSage ($q = 30$)	+0.0000	+0.0000	5.0×10^{-5}
queen tracts	Stochastic Chebyshev (50 probes)	-0.0147	-0.0144	0.477
queen tracts	Grid and spline (100 knots)	+0.0027	-0.0033	0.155
queen tracts	Smirnov–Anselin ($m = 30$)	+0.0160	+0.0133	0.587
queen tracts	Barry–Pace	+0.2184	+0.1909	1.6×10^{4}

Wherever the error is monotone across the posterior, the slope predicts the shift to within a fifth of its value or better, from Barry–Pace's fifth of a standard deviation down to Smirnov–Anselin's hundredths, including each stochastic Chebyshev draw with its sign. The error of grid and spline oscillates between knots, so its slope misreads the sign of a shift of a thousandth of a standard deviation, but the bound, which holds for an error of any shape, still contains the shift. The bound holds in every cell but is loose exactly where the error is large, as for Barry–Pace on the tracts, where the error's range over the support exceeds ten log-units.

Bivand et al. (2013) attribute the Smirnov–Anselin boundary error to an interpolation term that no longer corrects the remainder. Their absolute-error table registers that failure as a boundary artifact; the tilt registers it as the bias in Table 2. The failure is also predictable: modeling the omitted moments as geometric with a polynomial cor-

rection, the signature of a two-dimensional spectrum, gives the tilt of Smirnov–Anselin in closed form from its own highest moments. The derived value of ρ at which that tilt reaches 1.8×10^{-3} is 0.852, 0.844, and 0.809 on the rook, k-NN, and queen-tract weights, against measured crossings of 0.86, 0.85, and 0.81. The bias of Smirnov–Anselin is therefore a property of its remainder, computable in advance.

7.6 Cost at scale

Table 7 reports the cost of each method on the two tract matrices, per sampling chain of 2,000 retained draws after 1,000 warmup iterations. A slice update of ρ takes about 6.9 evaluations per iteration in slice-sampling fits of these models, warmup included, so a chain evaluates the Jacobian about 20,700 times.

Table 7: Jacobian cost on the census tracts ($n = 85{,}230$), queen contiguity and directed k-NN. Setup and evaluation columns from the queen matrix; the k-NN column from separate k-NN timings. Per chain is setup plus $6.9 \times 3{,}000$ evaluations. Pace–LeSage is not specified for directed weights, and its cost there, run outside that specification, is given in the text; the generic-LU row times the fresh SuperLU factorization PySAL's `spreg` calls at every ρ.

Method	Setup (s)	Evaluation (µs)	Queen, per chain	k-NN, per chain
Factorize and interpolate (AAA)	0.98	4.7	1.1 s	1.0 s
Stochastic Chebyshev (50 probes)	1.28	167.3	4.7 s	4.5 s
Barry–Pace	1.74	4.3	1.8 s	1.5 s
Smirnov–Anselin ($m = 30$)	360.67	16.8	6.0 min	53.6 s
Pace–LeSage ($q = 30$)	269.66	11.5	4.5 min	—
Grid and spline (100 knots)	17.86	2.7	17.9 s	14.7 s
Refactorization per ρ	0.14	25,895.7	8.9 min	48.6 min
Refactorization per ρ, generic LU	0.00	167,756.9	57.9 min	—

Factorize-and-interpolate spends 1.1 s per chain on the queen tracts and 1.0 s on the directed tracts, nearly all of it the 24 factorizations of the setup; each evaluation

takes under five microseconds at any n. Refactorization at every ρ spends 8.9 min per chain on the queen tracts with CHOLMOD and symbolic reuse, 499 times as long, and 57.9 min with the fresh factorization `spreg` performs. On the directed tracts, where current practice has no symbolic reuse, it spends 48.6 min. The gap widens with the length of the chain, because refactorization pays a factorization at every evaluation and the interpolant pays only for its setup: at 50,000 draws the queen tracts cost 2.6 s against 2.5 h.

Pace–LeSage, the only incumbent approximation that leaves the estimate unbiased over the full range of ρ, spends 4.5 min per chain on the queen tracts, nearly all of it computing thirty exact traces by sparse matrix powers. Smirnov–Anselin, whose moments come from the same sparse powers, spends 6.0 min and is biased at the boundary. The fill of W^j counts the walks of length j in the graph and does not depend on the ordering of its rows, so no ordering reduces it; the spectral methods have the accuracy of exact moments and a setup governed by the walk structure of the graph, while factorize-and-interpolate spends 24 factorizations at any scale. Grid and spline spends 17.9 s on its hundred knots and remains biased at the boundary.

The walk structure also explains why Pace–LeSage costs less on the directed tracts, where it runs outside its specification, than on the queen tracts: about 48.3 s, some 32 times factorize-and-interpolate. Its setup is a sequence of sparse products, which gain nothing from symmetry, so a directed matrix costs it no more than a symmetric one. By the thirtieth term the queen tracts carry 9,792 nonzeros per row against 1,729 for the directed k-NN tracts, whose five nearest neighbors, not all mutual, reach fewer tracts in thirty steps than queen contiguity does.

The matrix-free estimators are cheap at this size: Barry–Pace spends 1.8 s per chain and stochastic Chebyshev 4.7 s. Stochastic Chebyshev is the slower of the two by a few seconds, since each of its evaluations sums a longer series. Unlike Barry–Pace, it holds its bias below the materiality standard (Table 2).

None of these figures is available from PySAL's maximum-likelihood estimator at this size. Its default path forms W as a dense matrix, 58 GB here. Its sparse path factors $I_n - \rho W$ afresh at every ρ and completes the likelihood, but its variance computation then forms the explicit inverse of $I_n - \rho W$, which is structurally dense, so it exhausts memory in the same way. Neither step needs a dense matrix: the curve of the first method supplies the likelihood and, through its derivatives, the variance as well (Section 9). `bsreg` (Kuschnig, 2022) requires a dense eigendecomposition or dense determinants, both $O(n^3)$.

A single maximum-likelihood fit reverses the comparison between precomputing and refactoring, because a bounded optimizer visits only about a dozen values of ρ and a dozen refactorizations finish before 24 factorizations do. On the queen tracts, at a dozen evaluations per fit, the precomputation becomes the faster route once W is reused for 3 fits, as in a specification search across several models, a profile-likelihood interval, or a bootstrap.

7.7 Dense weights

The second method exists for weights whose dense blocks make factorization costly. We simulate three such matrices. Two place units at street addresses with heavy-tailed counts, joining co-located units into cliques: one adds a distance band between nearby addresses and is symmetrizable, and one adds each unit's four nearest units at other addresses and is directed. The third joins 25 hubs to 2,000 nodes each of a rook lattice. At $n = 8{,}000$ to 10,000 we compute the exact Jacobian from dense eigenvalues and measure bias as in Table 2; at $n = 32{,}000$, with about 20 million nonzeros and as many as 2,300 units at one address, we measure cost only.

Table 8: Bias in the posterior mean of ρ, in posterior standard deviations, on the simulated dense weights ($n = 8{,}000$ for the address weights, 10,025 for the hub lattice). Stochastic Chebyshev and Barry–Pace, the mean over 8 probe draws; the last column is the largest bias of any single draw across the row.

Weights	Method	0.5	0.7	0.85	0.9	0.95	Largest single draw
address cliques, distance band	Factorize and interpolate (AAA)	0.000	0.000	0.000	0.000	0.000	0.000
address cliques, distance band	Stochastic Chebyshev (50 probes)	-0.001	-0.003	-0.007	-0.010	-0.015	0.068
address cliques, distance band	Barry–Pace	-0.002	-0.004	+0.020	+0.136	+0.740	0.777
address cliques, directed k-NN	Factorize and interpolate (AAA)	0.000	0.000	0.000	0.000	0.000	0.000
address cliques, directed k-NN	Stochastic Chebyshev (50 probes)	+0.001	+0.001	0.000	-0.002	-0.005	0.044

Weights	Method	0.5	0.7	0.85	0.9	0.95	Largest single draw
address cliques, directed k-NN	Barry–Pace	0.000	0.000	+0.016	+0.096	+0.514	0.550
lattice with hubs	Factorize and interpolate (AAA)	0.000	0.000	0.000	0.000	0.000	0.000
lattice with hubs	Stochastic Chebyshev (50 probes)	0.000	-0.001	-0.002	-0.002	+0.002	0.154
lattice with hubs	Barry–Pace	+0.001	+0.004	+0.010	+0.017	+0.047	0.166

The bias follows Table 2. Factorize-and-interpolate leaves the estimate unmoved on all three matrices, and stochastic Chebyshev holds its mean within 0.015 standard deviations, while Barry–Pace reaches +0.740 and +0.514 on the address weights at $\rho = 0.95$. The hub lattice is the exception in two respects: Barry–Pace's bias there stays small, and the hubs widen the spread of stochastic Chebyshev's single draws, to 0.15 standard deviations at 50 probes, so weights with hubs, like strong negative dependence, call for more probes.

Table 9: Jacobian cost on the simulated dense weights. Per chain is setup plus 6.9 × 3,000 evaluations.

Weights	n	Nonzeros	Max. degree	nnz(W^2)/nnz(W)	Method	Setup (s)	Evaluation (µs)	Per chain
address cliques, band	8,000	1,518,470	622	1.2	Factorize and interpolate (AAA)	2.13	5.1	2.2 s

Weights	n	Nonzeros	Max. degree	nnz(W^2)/nnz(W)	Method	Setup (s)	Evaluation (µs)	Per chain
address cliques, band	8,000	1,518,470	622	1.2	Stochastic Chebyshev (50 probes)	0.50	154.3	3.7 s
address cliques, band	8,000	1,518,470	622	1.2	Barry–Pace	0.70	4.8	0.8 s
address cliques, k-NN	8,000	1,087,380	549	1.2	Factorize and interpolate (AAA)	0.67	5.2	0.8 s
address cliques, k-NN	8,000	1,087,380	549	1.2	Stochastic Chebyshev (50 probes)	0.37	153.8	3.6 s
address cliques, k-NN	8,000	1,087,380	549	1.2	Barry–Pace	0.50	4.8	0.6 s
lattice with hubs	10,025	139,600	2000	460.0	Factorize and interpolate (AAA)	0.35	5.1	0.5 s
lattice with hubs	10,025	139,600	2000	460.0	Stochastic Chebyshev (50 probes)	3.24	158.3	6.5 s

Weights	n	Nonzeros	Max. degree	nnz(W^2)/nnz(W)	Method	Setup (s)	Evaluation (µs)	Per chain
lattice with hubs	10,025	139,600	2000	460.0	Barry–Pace	0.12	4.6	0.2 s
address cliques, band	32,000	20,837,302	2288	1.1	Factorize and interpolate (AAA)	57.90	5.0	58.0 s
address cliques, band	32,000	20,837,302	2288	1.1	Stochastic Chebyshev (50 probes)	7.55	168.8	11.0 s
address cliques, band	32,000	20,837,302	2288	1.1	Barry–Pace	9.41	4.6	9.5 s
address cliques, k-NN	32,000	18,516,746	2265	1.1	Factorize and interpolate (AAA)	19.09	5.0	19.2 s
address cliques, k-NN	32,000	18,516,746	2265	1.1	Stochastic Chebyshev (50 probes)	6.68	170.8	10.2 s
address cliques, k-NN	32,000	18,516,746	2265	1.1	Barry–Pace	8.32	4.8	8.4 s

None of these matrices defeats factorization at the sizes we test. At $n = 8{,}000$ factorize-and-interpolate remains the cheaper exact route. The cost moves with the size of the dense blocks, more than with n. At $n = 32{,}000$ factorize-and-interpolate spends 58.0 s per chain on the symmetrizable address weights and 19.2 s on the

directed ones, nearly all of it the 24 factorizations of the setup, while stochastic Chebyshev spends 11.0 s and 10.2 s (Table 9), a setup of a few dozen batched sparse products whose cost is linear in the nonzeros. Barry–Pace costs about the same as stochastic Chebyshev here and carries the bias of Table 8. On the hub lattice the exact low-order moments dominate stochastic Chebyshev's setup: the hubs give W^2 460.0 times the nonzeros of W at a mean degree the guard of Section 6 does not flag. The guard should therefore key on the estimated fill of W^2, which the hubs reveal and the mean degree hides.

8 Discussion

We have introduced two methods for the log-determinant of a spatial autoregressive model, one for each branch of the literature, and a criterion for judging any approximation to it. The first extends the sparse-factorization practice the field already trusts. One symbolic analysis serves the whole family $\{I_n - \rho W\}$, now for directed weights as well as symmetrizable ones. An AAA rational interpolant reconstructs the Jacobian from 24 exact values so accurately that the bias it leaves in $\hat{\rho}$ is below 10^{-5} on every symmetrizable matrix tested, and invisible in the posterior on the directed ones. A Bayesian model of every census tract in the United States spends about a second per chain on the Jacobian under this method, where the fastest alternative that leaves the estimate unbiased over the full range of ρ spends minutes on contiguity weights and most of a minute on directed ones. The second method is built for dense weights, where it is several times faster than factorization at the scale of a rental market. It avoids the boundary bias that makes the incumbent matrix-free estimator unsafe for strong spatial dependence.

The choice between the two methods turns on the cost of one sparse factorization of $I_n - \rho W$. When that cost is small, as on every contiguity, distance-band, and nearest-neighbor matrix of the census tracts, we factor and interpolate, by Cholesky when W is symmetrizable and by LU when it is directed. When dense blocks make it large, as on the address weights of Section 7.7, we estimate the Jacobian matrix-free by stochastic Chebyshev. A factorization is built anyway, so timing the first one decides the route before the rest are spent. Among the methods we compared, these are the only ones that leave the estimate unbiased over the full range of ρ at a cost of seconds on the tracts; the spectral expansion of Pace & LeSage (2004) does as well but at a setup of minutes, and the moment method of Smirnov & Anselin (2009) and the Monte Carlo estimator of Barry & Pace (1999) carry the boundary bias of Table 2.

Because approximation error enters a posterior as an exponential tilt, an estimate responds not to the size of an approximation's error but to the variation of that error near where the estimate falls. The two can differ by orders of magnitude in either direction. By that criterion the least accurate estimator in absolute terms can be the right choice, and a method whose absolute errors look modest can displace an estimate by more than its standard error. Evaluating Jacobian approximations by absolute error,

as the literature has done, therefore neither certifies the fast methods nor rules out the inaccurate ones, so an evaluation should report the tilt of each method's error, or its bias in the estimate, across the range of ρ it will serve.

The evidence has three limits. The dense weights we test are simulated, and all of them still factor at the sizes we test, so the second method's advantage there is speed; the size at which factorization fails outright depends on the largest dense block more than on n. The spread of a single stochastic Chebyshev fit grows where the posterior is wide or the weights carry hubs, so strong negative dependence and graphs with hubs call for several hundred probes. Finally, the bias the first method leaves is constant in units of ρ but grows as $\sqrt{n}$ in standard deviations, so at several hundred thousand observations its node count should rise by a few factorizations to keep the same margin.

The first method also carries beyond the log-determinant. Its curve's first two derivatives are the resolvent traces that the maximum-likelihood information matrix, the gradient of a Hamiltonian sampler, and the average direct impact all require, so the same 24 factorizations deliver maximum-likelihood estimates with standard errors on the census tracts in seconds (Section 9), where current software cannot form the dense inverse it uses for them.

The methods apply to any model whose Jacobian is $\log|I_n - \rho W|$: the spatial lag, error, and Durbin models, their extensions to counts and binary outcomes, and separable spatial flow models (LeSage & Pace, 2008). The first applies more widely still, wherever a log-density carries an expensive log-determinant that is analytic in one scalar parameter, such as the dependence parameter of a conditional autoregressive model. There the node count must come from that model's own singularities, since the singularity at ± 1 belongs to row-standardized weights. Wherever the condition holds, the number of factorizations stops tracking the estimator and starts tracking the function.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used Claude Opus 5 (Anthropic) for programming assistance and to copyedit prose and format LaTeX. After using these tools the author reviewed and edited the content; any errors or omissions are the author's alone.

Data and code availability

Both methods are implemented in the `neighbayes` package, available on PyPI. The replication scripts that produce every number in this paper, and the scripts that build the weights matrices from 2020 Census geography with `libpysal`, are archived at [DOI supplied on acceptance].

References

Andrieu, C., & Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, *37*(2), 697–725. https://doi.org/10.1214/07-AOS574

Anselin, L. (1988). *Spatial econometrics: Methods and models*. Kluwer Academic Publishers.

Aune, E., Simpson, D. P., & Eidsvik, J. (2014). Parameter estimation in high dimensional Gaussian distributions. *Statistics and Computing*, *24*(2), 247–263. https://doi.org/10.1007/s11222-012-9368-y

Barry, R. P., & Pace, R. K. (1999). Monte carlo estimates of the log determinant of large sparse matrices. *Linear Algebra and Its Applications*, *289*(1–3), 41–54. https://doi.org/10.1016/S0024-3795(97)10009-X

Berrut, J.-P., & Trefethen, L. N. (2004). Barycentric lagrange interpolation. *SIAM Review*, *46*(3), 501–517. https://doi.org/10.1137/S0036144502417715

Bivand, R. S., Hauke, J., & Kossowski, T. (2013). Computing the jacobian in gaussian spatial autoregressive models: An illustrated comparison of available methods. *Geographical Analysis*, *45*(2), 150–179. https://doi.org/10.1111/gean.12008

Chen, Y., Davis, T. A., Hager, W. W., & Rajamanickam, S. (2008). Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software*, *35*(3), 1–14. https://doi.org/10.1145/1391989.1391995

Davis, T. A., & Palamadai Natarajan, E. (2010). Algorithm 907: KLU, a direct sparse solver for circuit simulation problems. *ACM Transactions on Mathematical Software*, *37*(3), 1–17. https://doi.org/10.1145/1824801.1824814

Dong, K., Eriksson, D., Nickisch, H., Bindel, D., & Wilson, A. G. (2017, November 9). *Scalable Log Determinants for Gaussian Process Kernel Learning*. https://doi.org/10.48550/arXiv.1711.03481

Elhorst, J. P., & Zigova, K. (2014). Competition in research activity among economic departments: Evidence by negative spatial autocorrelation. *Geographical Analysis*, *46*(2), 104–125. https://doi.org/10.1111/gean.12031

Furrer, R., & Sain, S. R. (2010). spam: A sparse matrix R package with emphasis on MCMC methods for Gaussian Markov random fields. *Journal of Statistical Software*, *36*(10), 1–25.

Gambhir, A. S., Stathopoulos, A., & Orginos, K. (2017). Deflation as a Method of Variance Reduction for Estimating the Trace of a Matrix Inverse. *SIAM Journal on Scientific Computing*, *39*(2), A532–A558. https://doi.org/10.1137/16M1066361

Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., & Wilson, A. G. (2018). GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18*, 7587–7597. https://dl.acm.org/doi/10.5555/3327757.3327857

Golub, G. H., & Van Loan, C. F. (2013). *Matrix computations* (4th ed.). Johns Hopkins University Press.

Han, I., Malioutov, D., Avron, H., & Shin, J. (2017). Approximating spectral sums of large-scale matrices using stochastic chebyshev approximations. *SIAM Journal on Scientific Computing*, *39*(4), A1558–A1585. https://doi.org/10.1137/16M1078148

Hutchinson, M. F. (1989). A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics — Simulation and Computation*, *18*(3), 1059–1076.

Kuschnig, N. (2022). Bayesian spatial econometrics: A software architecture. *Journal of Spatial Econometrics*, *3*(1), 6. https://doi.org/10.1007/s43071-022-00023-w

LeSage, J. P. (2021). *Spatial econometrics toolbox for MATLAB*. https://www.spatial-econometrics.com/. https://www.spatial-econometrics.com/

LeSage, J. P., & Pace, R. K. (2008). Spatial econometric modeling of origin-destination flows. *Journal of Regional Science*, *48*(5), 941–967. https://doi.org/10.1111/j.1467-9787.2008.00573.x

LeSage, J. P., & Pace, R. K. (2009). *Introduction to spatial econometrics*. Chapman & Hall/CRC.

Lyne, A.-M., Girolami, M., Atchadé, Y., Strathmann, H., & Simpson, D. (2015). On Russian Roulette Estimates for Bayesian Inference with Doubly-Intractable Likelihoods. *Statistical Science*, *30*(4), 443–467. https://doi.org/10.1214/15-STS523

Nakatsukasa, Y., Sète, O., & Trefethen, L. N. (2018). The AAA algorithm for rational approximation. *SIAM Journal on Scientific Computing*, *40*(3), A1494–A1522. https://doi.org/10.1137/16M1106122

Ng, E. G., & Peyton, B. W. (1993). Block Sparse Cholesky Algorithms on Advanced Uniprocessor Computers. *SIAM Journal on Scientific Computing*, *14*(5), 1034–1056. https://doi.org/10.1137/0914063

Ord, J. K. (1975). Estimation methods for models of spatial interaction. *Journal of the American Statistical Association*, *70*(349), 120–126. https://doi.org/10.1080/01621459.1975.10480272

Pace, R. K., & Barry, R. (1997). Quick computation of spatial autoregressive estimators. *Geographical Analysis*, *29*(3), 232–247. https://doi.org/10.1111/j.1538-4632.1997.tb00959.x

Pace, R. K., & LeSage, J. P. (2004). Chebyshev approximation of log-determinants of spatial weight matrices. *Computational Statistics & Data Analysis*, *45*(2), 179–196. https://doi.org/10.1016/S0167-9473(02)00321-3

Rey, S. J., & Anselin, L. (2007). PySAL: A Python library of spatial analytical methods. *The Review of Regional Studies*, *37*(1), 5–27. https://doi.org/10.52324/001c.8285

Smirnov, O., & Anselin, L. (2001). Fast maximum likelihood estimation of very large spatial autoregressive models: A characteristic polynomial approach. *Computa-*

tional Statistics & Data Analysis, *35*(3), 301–319. https://doi.org/10.1016/S0167-9473(00)00018-9

Smirnov, O., & Anselin, L. (2009). An $O(N)$ parallel method of computing the log-jacobian of the variable transformation for models with spatial interaction on a lattice. *Computational Statistics & Data Analysis*, *53*(8), 2980–2988. https://doi.org/10.1016/j.csda.2008.10.010

Trefethen, L. N. (2013). *Approximation theory and approximation practice*. SIAM.

Ubaru, S., Chen, J., & Saad, Y. (2017). Fast estimation of $\mathrm{tr}(f(A))$ via stochastic Lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, *38*(4), 1075–1099. https://doi.org/10.1137/16M1104974

9 Maximum-likelihood inference from the same curve

The first method delivers more than the log-determinant. Maximum-likelihood estimation of Equation 1 maximizes the concentrated likelihood, which carries $\mathcal{J}(\rho)$. Its standard errors come from the information matrix (Anselin, 1988), which carries three traces of the resolvent. Writing $W_A = W(I_n - \hat{\rho} W)^{-1}$, they are

$$\mathrm{tr}_1 = \mathrm{tr}(W_A), \qquad \mathrm{tr}_2 = \mathrm{tr}(W_A^2), \qquad \mathrm{tr}_3 = \mathrm{tr}(W_A^{\top} W_A). \tag{12}$$

Current software computes them by forming $(I_n - \hat{\rho} W)^{-1}$ explicitly. The inverse of a sparse matrix is dense, 58 GB on the census tracts. That step, not the log-determinant, exhausts the memory of PySAL's maximum-likelihood estimator at that size (Section 7.6).

Two of the three traces are derivatives of the Jacobian. By Jacobi's formula,

$$\mathcal{J}'(\rho) = -\mathrm{tr}(W_A), \qquad \mathcal{J}''(\rho) = -\mathrm{tr}(W_A^2), \tag{13}$$

The AAA curve of Equation 9 has both in closed form. With $r = N/D$, $N = \sum_j w_j f_j/(\rho - \rho_j)$ and $D = \sum_j w_j/(\rho - \rho_j)$,

$$r' = \frac{N' - rD'}{D}, \qquad r'' = \frac{N'' - 2r'D' - rD''}{D}, \tag{14}$$

where each derivative of N and D is a sum over the same support points with the denominator raised one power. Evaluating them costs microseconds, like the curve itself. The third trace is not a derivative of $\mathcal{J}$, but it is the squared Frobenius norm of W_A, so a Hutchinson estimate $\mathrm{tr}_3 \approx K^{-1} \sum_k \| W(I_n - \hat{\rho} W)^{-1} z_k \|^2$ over K random sign vectors needs one sparse factorization at $\hat{\rho}$ and K solves with it. The same factorization supplies the reduced-form prediction $(I_n - \hat{\rho} W)^{-1} X\hat{\beta}$ the information matrix also needs. No dense matrix is formed at any step.

Table 10 fits the spatial lag model by maximum likelihood this way, with the concentrated likelihood and information matrix of PySAL's `spreg` (Rey & Anselin, 2007),

the curve's 24 factorizations, and $K = 50$. At $n = 10{,}000$ the reference is the exact estimator: $\hat{\rho}$ from the exact likelihood and the traces from the dense inverse. On the census tracts, where the dense inverse cannot be formed, tr_1 and tr_2 are checked against finite differences of the exact log-determinant and tr_3 against 1,000 probes.

Table 10: Maximum-likelihood estimation of the spatial lag model from the AAA curve. Relative errors of the traces of Equation 12 and of the standard errors of $(\hat{\beta}, \hat{\rho})$ against the reference described in the text; fit time covers the curve's setup, the optimization, and the information matrix.

Weights	ρ	$\hat{\rho}$	tr_1, tr_2 rel. error	tr_3 rel. error	Largest SE rel. difference	Fit (s)
rook, $n =$ 10,000	0.5	0.4947	7.1×10^{-8}	2.4×10^{-3}	1.8×10^{-4}	0.5
rook, $n =$ 10,000	0.9	0.8994	1.1×10^{-5}	2.5×10^{-3}	2.0×10^{-4}	0.2
k-NN, $n =$ 10,000	0.5	0.4945	2.4×10^{-11}	3.8×10^{-3}	2.9×10^{-4}	0.2
k-NN, $n =$ 10,000	0.9	0.8961	1.1×10^{-6}	6.0×10^{-3}	4.8×10^{-4}	0.2
queen tracts	0.5	0.4961	1.1×10^{-6}	1.0×10^{-3}	7.6×10^{-5}	1.4
queen tracts	0.9	0.8988	2.3×10^{-5}	1.9×10^{-3}	1.4×10^{-4}	1.4
k-NN tracts	0.5	0.4971	1.3×10^{-6}	2.9×10^{-3}	2.2×10^{-4}	1.5
k-NN tracts	0.9	0.8987	2.6×10^{-5}	5.8×10^{-3}	4.4×10^{-4}	1.5

At $n = 10{,}000$ the estimate from the curve agrees with the exact estimate to every one of the six decimals recorded, the two derivative traces are accurate to 2.6×10^{-5} or better, and the 50-probe estimate of tr_3 to 6.0×10^{-3}. The standard errors differ from the reference by at most 4.8×10^{-4}, a relative difference far below the precision to which standard errors are reported. On the census tracts the whole fit, estimate and standard errors, takes 1.5 seconds or less.

The same derivative serves Bayesian estimation. $\mathcal{J}'(\rho)$ is the gradient that Hamiltonian samplers need. The average direct impact of Equation 1, $n^{-1}\mathrm{tr}\big((I_n - \rho W)^{-1}\big) = 1 - \rho\,\mathcal{J}'(\rho)/n$, follows from it without the dense inverse that impact calculations otherwise form. Every quantity of spatial autoregressive inference that depends on ρ through the Jacobian or its first two derivatives therefore comes off the one curve.