BUILDING SPATIAL CONDITIONAL AUTOREGRESSIVE MODELS IN THE STAN PROGRAMMING LANGUAGE

OSF PREPRINT

✉ **Connor Donegan***

Geospatial Information Sciences
The University of Texas at Dallas
Richardson, Texas 75080
`Connor.Donegan@UTDallas.edu`

and

Department of Population and Data Sciences
University of Texas Southwestern Medical Center
Dallas, Texas 75390
`Connor.Donegan@UTSouthwestern.edu`

June 20, 2022

ABSTRACT

Modeling data collected by areal units, such as counties or census tracts, is a core component of population health research and of growing interest to the social sciences. Bayesian inference has both practical and philosophical advantages over classical statistical techniques, and advances in Markov chain Monte Carlo (MCMC) are expanding the range of research questions to which Bayesian inference may be applied. This code snippet introduces code for fitting spatial conditional autoregressive (CAR) models with the Stan modeling language. Stan is an expressive programming language that uses a dynamic Hamiltonian Monte Carlo (HMC) algorithm to draw samples from user-specified probability models. This paper discusses various CAR model specifications and introduces computationally efficient implementations for Stan users. The `geostan` R package provides, among other things, convenience functions to ease the process of building custom spatial models in Stan. The paper demonstrates use of the code by modeling United States county, age- and sex-specific mortality data, including censored observations. The demonstration highlights the importance of thoughtful analysis of model residuals.

Keywords spatial statistics · Bayesian inference · mortality · population health

1 Introduction: Bayesian Spatial Analysis

This code snippet details efficient functions for fitting conditional autoregressive (CAR) models to spatial data (1) using the Stan modeling language (2, 3). CAR models are well-suited for modeling disease incidence and mortality rates given data collected over areal units such as census tracts, counties, or states. The CAR model can be used as a likelihood for areal data, as a probability model for unobservables (abstractions) like disease risk, or as a probability model for imperfectly observed values such as survey-based covariates (4–6). Stan uses a dynamic Hamiltonian Monte Carlo (HMC) algorithm (7–9) to draw samples from user-specified probability distributions. After the user specifies the desired probability model using Stan’s expressive syntax, the model is compiled into C++ for efficient execution.

The BUGS (10), JAGS (11), GeoBUGS (12), and NIMBLE (13) projects provide similar resources to researchers who are interested in building hierarchical Bayesian models (HBMs) without having to write original computer code for their own Markov chain Monte Carlo (MCMC) samplers. Given a user-specified model, these programs compile

*I am grateful for helpful feedback from Yongwan Chun and Daniel Griffith on material covered in this paper. I am solely responsible for any oversights or errors.

MCMC routines and return MCMC samples to the user. MCMC remains the only available method to obtain samples from the *joint* posterior probability distributions of even moderately complex models. Joint probabilities are required to analyze quantities of interest derived from more than one, possibly correlated model parameters or fitted values (such as measures of (spatial) inequality, variance, autocorrelation, change over time, and prior and posterior predictive distributions).

Stan represents an important advance for MCMC methods. While it is not the best solution to every Bayesian inference problem (14), a well-written Stan model can often produce considerable gains in both efficiency and reliability relative to other algorithms (15). Still, Stan remains underutilized for spatial modeling. Many models available in GeoBUGS are not yet available to Stan users, although intrinsic conditional autoregressive (16), Gaussian process (3), and eigenvector spatial filtering (17) models have been developed in Stan. While one can use Stan’s built-in multivariate normal density functions to fit CAR models, a major advantage of the CAR model is that computations can be simplified by exploiting the sparsity of the precision matrix. Currently, there is one implementation of the CAR model in Stan (18) but it does not benefit from sparse matrix computations and is compatible with only one CAR model specification (the WCAR model, as described in Section 2.2).

This paper provides a brief introduction to CAR model specifications before detailing two new, highly efficient Stan implementations of the CAR model. The paper demonstrates use of the code by modeling age-specific female US county mortality rates for the years 2015-2019. The analysis leverages a set of convenience functions, available through the `geostan` R package (`geostan`), for building custom spatial models in Stan. The `geostan` package also provides convenient interface to pre-compiled Stan models, including the CAR models introduced here, although the package itself is beyond the purview of this paper.

2 CAR model specifications

2.1 The CAR model

The auto-Gaussian model belongs to a wider class of *auto-models* for spatial data that employ the Markov property of conditional independence over a regular or irregular lattice (or graph structure), i.e., Markov Random Fields (MRFs) (1, 19). The material in this section largely follows the presentation in Cressie and Wikle (19, pp. 167-203). Define the lattice, $D \equiv \{s_1, s_2, \dots, s_n\}$, as a collection of discrete sites and let $Y(s_i)$ be our observation of variable Y at the i^{th} site. Then the neighborhood, $N(s_i)$, of the i^{th} site consists of the collection of sites that are considered “neighbors” of s_i (i.e., their index values). Often, neighbors are defined as areal units that share a border, although $N(s_i)$ may alternatively be defined to include all sites within some threshold distance from s_i . The resulting connectivity or distance structure can be stored in an $n \times n$ non-negative adjacency matrix, C , where $c_{i,i} = 0$ and $c_{i,j}$ is non-zero only if $j \in N(s_i)$.

The Markov property states that the probability distribution for $Y(s_i)$ given knowledge of all other sites, $\mathbf{Y}(-s_i)$, may be reduced to its probability distribution given only its neighboring values, $\mathbf{Y}(N(s_i))$:

$$p(Y(s_i) | \mathbf{Y}(-s_i)) = p(Y(s_i) | \mathbf{Y}(N(s_i))). \quad (1)$$

The CAR model is a multivariate Gaussian probability model. Conditional on knowledge of neighboring values, the CAR model assigns the following probability distribution to $Y(s_i)$:

$$p(Y(s_i) | \mathbf{Y}(N(s_i))) = \text{Gau}(Y(s_i) | \mu(s_i) + \sum_{j \in N(s_i)} \rho \cdot c_{i,j} (Y(s_j) - \mu(s_j)), \tau_i^2). \quad (2)$$

The weighted summation of neighboring values (in deviation form) constitutes a *spatial trend*, and $\mu(s_i)$ may equal the global mean (intercept) or it may incorporate covariates. Given specific properties of Σ detailed below, the conditional probabilities for $Y(s_i)$ specified by Equation (2) define the following joint probability distribution for $\mathbf{Y}(s)$:

$$\mathbf{Y} \sim \text{Gau}(\boldsymbol{\mu}, [\mathbf{I} - \rho C]^{-1} \mathbf{M}). \quad (3)$$

Whereas geostatistical (Gaussian process) models encode spatial information directly into the covariance matrix, Σ , the lattice structure of the CAR model, represented by the sparse connectivity matrix C , is encoded directly into the precision matrix, Σ^{-1} :

$$\Sigma^{-1} = \mathbf{M}^{-1} [\mathbf{I} - \rho C]. \quad (4)$$

The parameter ρ expresses the type (positive or negative) and degree of spatial autocorrelation (SA), $\mathbf{I}$ is the identity matrix, and $\mathbf{M}$ is a diagonal matrix containing the conditional variances, $m_{i,i} = \tau_i^2$, for $Y(s_i) | \mathbf{Y}(N(s_i))$. Typically, $\mathbf{M} = \tau^2 \mathbf{\Delta}$, where $\mathbf{\Delta} = \text{diag}(\Delta_1, \dots, \Delta_n)$ is a diagonal matrix with fixed (‘known’) positive values. If $\rho = 0$ and $\mathbf{M} = \tau^2 \mathbf{I}$, then the CAR model reduces to a Gaussian model with independent observations.

To ensure that Σ is symmetric and positive definite, the following conditions are required (19, 20):

1. $\mathbf{CM}$ is symmetric, i.e., $\Delta_j c_{i,j} = \Delta_i c_{j,i}$,
2. $m_{i,i} > 0$ for all $i \in 1, \dots, n$,
3. $\lambda_{min}^{-1} < \rho < \lambda_{max}^{-1}$ where λ_i are the ordered eigenvalues of the matrix $\mathbf{C}$.

The constraint on ρ ensures that $(\mathbf{I} - \rho\mathbf{C})$ is positive definite; since all conditional variances $m_{i,i}$ must be positive, this ensures that Σ also is positive definite (20, p. 250). These requirements allow for a variety of specifications of $\mathbf{C}$ and $\mathbf{M}$.

2.2 Adjacency-based specifications

Following Cressie and Wikle (19), let the binary neighborhood (i.e., adjacency) matrix be denoted as $\mathbf{A}$, and its elements as $a_{i,j}$, such that $a_{i,j} = a_{j,i} = 1$ if s_j and s_i are neighbors ($s_j \in N(s_i)$), and, otherwise, $a_{i,j} = a_{j,i} = 0$. All diagonal elements are also zero: $a_{i,i} = 0$. Then the number of sites in $N(s_i)$ is $|N(s_i)| = \sum_{j=1}^n a_{i,j}$.

Table 1 presents various specifications of $\mathbf{CM}$. The weighted (WCAR) specification sets $\mathbf{C}$ equal to the row-standardized version of $\mathbf{A}$ ($c_{i,j} = \frac{a_{i,j}}{|N(s_i)|}$), and the conditional variances are equal to $m_{i,i} = \frac{\tau^2}{|N(s_i)|}$. This is perhaps the most commonly employed specification. The autocorrelated CAR (ACAR) (19, 21, 22) uses the same conditional variances but a different specification of $\mathbf{C}$: $c_{i,j} = \frac{a_{i,j}}{|N(s_j)|^{1/2}|N(s_i)|^{-1/2}}$. The motivation for this specification is to address what the authors describe as an unattractive feature of the WCAR specification, specifically that the partial correlations between sites $Y(s_i)$ and $Y(s_j)$ are not equal to ρ , and thus ρ cannot be interpreted as a spatial analog to the unitless partial correlation coefficient with range $[-1, 1]$. Whereas the WCAR model has squared partial correlation of $\rho^2 |N(s_i)|^{-1} |N(s_j)|^{-1}$, the ACAR model has partial correlation equal to ρ (Cressie and Wikle 2011, p. 188; cf. Cressie 2015, p. 557). However, λ_{max}^{-1} is not equal to unity with the ACAR specification (it is often closer to zero than to one), though standardized value, $\frac{\rho}{1/\lambda_{max}}$ (or $\frac{\rho}{1/\lambda_{min}}$), may be calculated for the sake of interpretation. Other aspects of the CAR model render its autocorrelation parameter materially distinct from the more familiar correlation coefficient. In particular, it often approaches its maximum value even for moderate levels of spatial autocorrelation, but, at such high levels, minuscule increases in ρ correspond to large changes in the degree of spatial autocorrelation (as measured by Moran's I or other means) (see 20, p. 252). So long as this property of the model is not misinterpreted, there is no reason why it should detract from the model's appeal.

2.3 Distance-based specifications

In the inverse-distance based (DCAR) model, connectivity ($c_{i,j} = c(d_{i,j})$) is a decreasing function of distance. “Distance” is often measured in terms of geodesic distance, although some argue that distance may be measured in attribute space (e.g., by demographic characteristics) (20). The differences between the DCAR_W (20, p. 253) and DCAR_A (23, p. 557) specifications are important; for one, the DCAR_W model row-standardizes the inverse distances (this, plus the specification of $\mathbf{M}$, provides an analogy to the WCAR specification). This *relativizes* the influence of distance, such that the amount of decay in connectivity induced by a fixed distance is non-constant across observations. Thus, the degree of connectivity in the DCAR_W model is *not* a function of distance alone, and one can readily see that this may conflict with researcher intentions. By contrast, the DCAR_A specification scales the inverse distances by a *single* value, $C(k)$, the maximum inverse distance among all pairs of neighboring sites, so that connectivity is a function of distance. Scaling by $C(k)$ aids model comparison for different values of k (23, p. 557). Cliff and Ord (24, p. 144) provide an overview of inverse-distance weighting schemes and they include a second parameter, γ , to the weighting scheme, such that $c(d_{i,j}) = (d_{i,j} + \gamma)^{-k}$.

3 Efficient Calculation of the Multivariate Gaussian p.d.f.

To implement the CAR model in Stan, we require an efficient calculation of the log of the multivariate Gaussian probability density function (pdf) with sparse precision matrix, Σ^{-1} . The pdf is,

$$p(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2}} \frac{1}{\sqrt{|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})\right) \quad (5)$$

where $|\boldsymbol{\Sigma}|$ is the determinant of the covariance matrix. The log-pdf is:

$$-\frac{n}{2} \log(2\pi) + \frac{1}{2} \log(|\boldsymbol{\Sigma}^{-1}|) - \frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu}). \quad (6)$$

Table 1: Weighted (WCAR), autocorrelated (ACAR), and inverse-distance (DCAR) CAR specifications (19–23).

	Δ_i	$c_{i,j}$	$\Delta_j c_{i,j} = \Delta_i c_{j,i}$
WCAR	$ N(s_i) ^{-1}$	$a_{i,j} N(s_i) ^{-1}$	$ N(s_i) ^{-1} N(s_j) ^{-1}$
ACAR	$ N(s_i) ^{-1}$	$a_{i,j} N(s_j) ^{1/2} N(s_i) ^{-1/2}$	$ N(s_j) ^{-1/2} N(s_i) ^{-1/2}$
DCAR _W	$\left[\sum_{j \in N(s_i)} d_{i,j}^{-k} \right]^{-1}$	$\frac{d_{i,j}^{-k}}{\sum_{j \in N(s_i)} d_{i,j}^{-k}}$	$\frac{d_{i,j}^{-k}}{\sum_{j \in N(s_i)} d_{i,j}^{-k} \sum_{i \in N(s_j)} d_{j,i}^{-k}}$
DCAR _A	$ N(s_i) ^{-1}$	$\frac{d_{i,j}^{-k}}{C(k)} \left[\frac{ N(s_j) }{ N(s_i) } \right]^{1/2}$	$\frac{d_{i,j}^{-k}}{C(k)} N(s_i) ^{-1/2} N(s_j) ^{-1/2}$

Note: $a_{i,j}$ are elements of the binary neighborhood matrix, $d_{i,j}$ is a distance metric, $k \in \{1, 2\}$, and $C(k)$ is the maximum $d_{i,j}^{-k}$ for all $j \in N(s_i), i = 1, \dots, n$

Two software implementations will be presented: the first is applicable to any valid CAR model specification, while the second is valid only for WCAR (including DCAR_W) specifications. The input data required for each function can be produced easily using the `prep_car_data` function provided by the `geostan` R package (`geostan`) (see Section 5).

3.1 General implementation

If $\mathbf{z} = \mathbf{y} - \boldsymbol{\mu}$, then the quadratic term to calculate is (23, p. 408):

$$\mathbf{z}' \mathbf{M}^{-1} (\mathbf{I} - \rho \mathbf{C}) \mathbf{z}. \quad (7)$$

The challenge is to efficiently calculate $(\mathbf{I} - \rho \mathbf{C}) \mathbf{z}$. Since even $(\mathbf{I} - \rho \mathbf{C})$ is sparse, we take advantage of sparse matrix operations and compressed row storage (CRS). Using CRS, non-zero matrix elements are stored in a vector. We keep a static CRS representation of $(\mathbf{I} - \mathbf{C})$, together with the index positions of $-\mathbf{C}$ so that those elements can be operated on independently. For each MCMC iteration, the program simply multiplies the non-zero elements of $-\mathbf{C}$ by ρ . Then $(\mathbf{I} - \rho \mathbf{C}) \mathbf{z}$ is calculated as a sparse matrix-vector product. The calculation of the quadratic term is completed by finding the dot product of the resulting vector and $\mathbf{z}' \mathbf{M}^{-1}$.

Given the determinant of the precision matrix,

$$|\boldsymbol{\Sigma}^{-1}| = |\tau^{-2} \mathbf{I}| |\boldsymbol{\Delta}^{-1}| |(\mathbf{I} - \rho \mathbf{C})|, \quad (8)$$

we require the following log-determinant:

$$\log(|\boldsymbol{\Sigma}^{-1}|) = \log(\tau^{-2n}) + \log(|\boldsymbol{\Delta}^{-1}|) + \log(|(\mathbf{I} - \rho \mathbf{C})|). \quad (9)$$

The term $\log(|\boldsymbol{\Delta}^{-1}|)$ is a constant that only needs to be calculated once. The determinant of $(\mathbf{I} - \rho \mathbf{C})$ may be obtained by (23, p. 471):

$$|(\mathbf{I} - \rho \mathbf{C})| = \prod_{k=1}^n (1 - \rho \lambda_k), \quad (10)$$

where λ_k are the eigenvalues of $\mathbf{C}$. The log determinant of the precision matrix is thus equal to:

$$\log(|\boldsymbol{\Sigma}^{-1}|) = -2n * \log(\tau) + \log(|\boldsymbol{\Delta}^{-1}|) + \sum_{k=1}^n \log(1 - \rho \lambda_k) \quad (11)$$

Listing 1 contains the `car_normal_lpdf` Stan function, implementing these calculation.

3.2 WCAR implementation

The WCAR model admits of an alternative calculation of the quadratic term which offers greater computational efficiency. $\mathbf{A}$ remains a symmetric connectivity matrix, and let $\mathbf{D} = \boldsymbol{\Delta}^{-1}$ be a diagonal matrix containing the row sums of $\mathbf{A}$ on the diagonal. Then, the quadratic term in Equation 6 is equivalent to the following (18, 20):

$$\mathbf{z}' \boldsymbol{\Sigma}^{-1} \mathbf{z} = \frac{1}{\tau^2} \mathbf{z}' (\mathbf{D} - \rho \mathbf{A}) \mathbf{z} \quad (12a)$$

$$= \frac{1}{\tau^2} \mathbf{z}' \mathbf{D} \mathbf{z} - \rho \mathbf{z}' \mathbf{A} \mathbf{z}. \quad (12b)$$

```

/**
 * Log probability density of the conditional autoregressive (CAR) model
 *
 * @param y Process to model
 * @param mu Mean vector
 * @param tau Scale parameter
 * @param rho Spatial dependence parameter
 * @param ImC Sparse representation of (I - C): non-zero values only
 * @param ImC_v Column indices for values in ImC
 * @param ImC_u Row starting indices for values in ImC
 * @param Cidx Indices for the off-diagonal elements in ImC
 * @param Delta_inv Diagonal elements from the inverse of Delta, where M =
 *   Delta * tau^2 is a diagonal matrix containing the conditional
 *   variances.
 * @param log_det_Delta_inv Log determinant of Delta inverse
 * @param lambda Eigenvalues of C
 * @param n Length of y
 *
 * @return Log probability density
 */
real car_normal_lpdf(vector y, vector mu,
                    real tau, real rho,
                    vector ImC, int[] ImC_v, int[] ImC_u, int[] Cidx,
                    vector Delta_inv, real log_det_Delta_inv, vector
                    lambda,
                    int n) {
  vector[n] z = y - mu;
  vector[num_elements(ImC)] ImrhoC = ImC;           // (I - rho C)
  vector[n] zMinv = (1 / tau^2) * z .* Delta_inv;    // z' * M^{-1}
  vector[n] ImrhoCz;                                 // (I - rho * C) * z
  vector[n] ldet_ImrhoC;
  ImrhoC[Cidx] = rho * ImC[Cidx];
  ImrhoCz = csr_matrix_times_vector(n, n, ImrhoC, ImC_v, ImC_u, z);
  for (i in 1:n) ldet_ImrhoC[i] = loglm(rho * lambda[i]);
  return 0.5 * (
    -n * log( 2 * pi() )
    - 2 * n * log(tau)
    + log_det_Delta_inv
    + sum(ldet_ImrhoC)
    - dot_product(zMinv, ImrhoCz)
  );
}

```

Listing 1: A Stan function to efficiently calculate log-probability for any valid CAR model specification.

```

/**
 * Log probability density of the conditional autoregressive (CAR) model:
 *   WCAR specifications only.
 *
 * @param y Process to model
 * @param mu Mean vector
 * @param tau Scale parameter
 * @param rho Spatial dependence parameter
 * @param A_w Sparse representation of the symmetric connectivity matrix,
 *   A
 * @param A_v Column indices for values in A_w
 * @param A_u Row starting indices for values in A_u
 * @param D_inv The row sums of A; i.e., the diagonal elements from the
 *   inverse of Delta, where M = Delta * tau^2 is a diagonal matrix
 *   containing the conditional variances.
 * @param log_det_D_inv Log determinant of Delta inverse.
 * @param lambda Eigenvalues of C (or of symmetric matrix Delta^{-1/2}*C*
 *   Delta^{1/2}); for the WCAR specification, C is row-standardized.
 * @param n Length of y
 *
 * @return Log probability density
 */
real wcar_normal_lpdf(vector y, vector mu,
                      real tau, real rho,
                      vector A_w, int[] A_v, int[] A_u,
                      vector Delta_inv, real log_det_Delta_inv,
                      vector lambda,
                      int n) {
  vector[n] z = y - mu;
  real ztDz;           // z transpose * D * z
  real ztAz;           // z transpose * A * z
  vector[n] ldet_ImrhoC;
  ztDz = (z .* Delta_inv)' * z;
  ztAz = z' * csr_matrix_times_vector(n, n, A_w, A_v, A_u, z);
  for (i in 1:n) ldet_ImrhoC[i] = log1m(rho * lambda[i]);
  return 0.5 * (
    -n * log( 2 * pi() )
    -2 * n * log(tau)
    + log_det_Delta_inv
    + sum(ldet_ImrhoC)
    - (1 / tau^2) * (ztDz - rho * ztAz));
}

```

Listing 2: A Stan function to efficiently calculate log-probability density of the WCAR model (not valid for other specifications).

Equation 12b is easily calculated, with efficiency obtained by using sparse matrix multiplication for Az , and the remainder of terms in the log density (Equation 6) remaining the same as previously stated. This calculation also accommodates the $DCAR_W$ specification; in that case, the non-zero elements of A contain inverse distances (see Table 1). Listing 2 contains the efficient `wcar_normal_lpdf` function to implement the WCAR model in Stan.

4 Sampling efficiency

A highly efficient implementation of the CAR model for MCMC analysis is useful not just for taking on projects of large scale, but also for implementing models of greater inferential complexity. MCMC sampling efficiency varies with numerous factors including the number of observations, model specification, and model parameterization. The purpose of this section is to examine the performance of the new Stan functions for a commonly employed base model

specification and spatial arrangement, and to compare with an established alternative (NIMBLE). The results showcase how Stan’s HMC algorithm is able to generate samples containing considerably less serial autocorrelation than other MCMC algorithms, which can enable users to complete analyses using far fewer samples (see Section 5).

Following common disease mapping analyses, the following specification was used: case counts, $Y(s_i)$, are assigned a Poisson likelihood conditional on population-at-risk, $P(s_i)$, times the level of risk, $e^{\phi(s_i)}$; the WCAR model (Table 1) is assigned to the vector of log-risks:

$$\begin{aligned} Y(s_i) &\sim \text{Poisson}(P(s_i) \cdot e^{\phi(s_i)}) \\ \phi(\mathbf{s}) &\sim \text{Gaussian}(\alpha \cdot \mathbf{1}, \Sigma) \end{aligned} \quad (13)$$

The model was applied to US county, age-specific (55-64) all-cause female mortality data, aggregated over the years 2014-2018 (25) (previously analyzed in Donegan, Chun, and Griffith (6)). Mortality data is available for $N = 2,875$ counties (excluding Alaska, which effectively lacks county equivalents) (Figure 1). No covariates were employed; instead, a constant-mean ($\mu = \alpha \cdot \mathbf{1}$) WCAR model was assigned to the log-rates, with diffuse priors. The same model was specified using three different software implementations: the `car_normal_lpdf` and `wcar_normal_lpdf` functions in Stan, plus NIMBLE’s (version 0.11.1) built-in CAR model, `dcar_proper`. An attempt was made to use OpenBUGS also, but the software failed for reasons that appear to be related to the size of the dataset.

For each software implementation, four MCMC chains were initiated serially (not in parallel), with 15,000 samples drawn from each chain. The first 7,500 samples were discarded as warmup. Convergence was evaluated using the split R-hat statistic (26); all results have $\hat{R} \in 1 \pm 0.003$. *Sampling efficiency*, η , was measured as effective sample size (ESS) per second of sampling: $\eta = \text{ESS}/\text{sec}$ (26); the time required to compile each model into C++ was excluded from the sampling time. The *relative efficiency*, ω , of each software implementation was calculated using NIMBLE as the reference level; for example,

$$\omega(\text{car_normal_lpdf}) = \eta(\text{car_normal_lpdf})/\eta(\text{dcar_proper}). \quad (14)$$

All MCMC diagnostics were calculated using RStan’s `monitor` function (27).

NIMBLE completed sampling in 125 minutes and yielded $\eta = 0.75$ ESS/second for the intercept, α (Table 2a). Stan models using the `car_normal_lpdf` and `wcar_normal_lpdf` completed sampling in 47 and 30 minutes, respectively. The added sampling efficiency of the Stan models is considerably greater when measured by ESS/second than total sampling time. For the intercept, α , sampling with `car_normal_lpdf` and `wcar_normal_lpdf`, respectively, were 9.6 and 17 times more efficient than NIMBLE. All three implementations produced posterior distributions with effectively identical summary values (means and tail-area quantiles) (Table 2b).

Table 2: Comparison of three CAR model software implementations given a log-linear Poisson model of county mortality rates ($N=2,875$).

(a) Sampling efficiency

	Minutes [†]	ESS per second*		
		α	ρ	τ
Nimble: <code>dcar_proper</code>	125	0.75 (1)	0.53 (1)	0.34 (1)
Stan: <code>car_normal_lpdf</code>	47	7.18 (9.57)	7.35 (13.87)	6.82 (20.06)
Stan: <code>wcar_normal_lpdf</code>	30	12.86 (17.15)	11.63 (21.94)	12.85 (37.79)

[†] 4 MCMC chains, 15,000 samples each.

* Relative efficiency in parentheses.

(b) Posterior summary: mean [2.5%, 97.5%]

	α	ρ	τ
Nimble: <code>dcar_proper</code>	-4.87 [-4.99,-4.74]	1 [0.99, 1]	0.39 [0.38, 0.4]
Stan: <code>car_normal_lpdf</code>	-4.87 [-4.99,-4.75]	1 [0.99, 1]	0.39 [0.38, 0.4]
Stan: <code>wcar_normal_lpdf</code>	-4.87 [-4.99,-4.75]	1 [0.99, 1]	0.39 [0.37, 0.4]

5 US county mortality rates

5.1 Structured risk and spatial smoothing

The following analysis of mortality data is concerned with structured, population-level patterns of risk (28), which is not directly observable. For a fixed risk level, crude mortality rates are highly variable when the population at risk is small. Whereas a conventional ‘random effects’ model will smooth outlying rates towards a global mean, the CAR model smooths towards a local area mean (Equation 2). This spatial smoothing expresses a degree of prior scepticism towards local outliers, which is balanced against the evidence summarized by the Poisson likelihood (Equation 13) such that the amount of smoothing imposed is a decreasing function of population size. The ability to infer a spatial trend in the level of risk, along with the characteristic magnitude of variability around that trend (τ), can also improve inferences about the level of risk in places for which no observations exist or, as in this demonstration analysis, in places with censored observations.

This analysis will be completed using the RStan interface to Stan (27), accessed from the R environment for statistical computing (29). Building on the disease mapping specification from Equation 13, the analysis will include a model for censored mortality counts and will make use of specialized functions provided by the `geostan` package for building custom spatial CAR models in Stan (`geostan`). The analysis concludes by pointing to an important limitation of spatial smoothers such as the CAR model, which renders thoughtful examination of residuals a necessity for spatial analysis.

Figure 1: Crude female county mortality per 100,000, ages 55-64, years 2014-2016.

5.2 Model specification

The model specification follows Equation 13, again using the constant-mean WCAR specification. Here we will consider appropriate prior distributions for the CAR parameters α , τ , and ρ . The mean log-mortality rate, α , must be negative, and cursory familiarity with the subject matter is sufficient to realize that, say, $e^{-10} = 4.5$ deaths per 100,000 and $e^{-1} = 36,788$ deaths per 100,000 are implausible extremes. Given the log-link function, τ is expected to be less than unity. We center our fairly vague prior distribution for α on $e^{-5} = 674$ deaths per 100,000 (about equal to the US female, age 55-64, mortality rate for 2009-2013), and assign a half-standard normal (positively constrained) prior to τ :

$$\alpha \sim \text{Gau}(-5, 2^2) \quad (15a)$$

$$\rho \sim \text{beta}(2, 1) \quad (15b)$$

$$\tau \sim \text{Gau}(0, 1^2)(0, \infty) \quad (15c)$$

The support for ρ is $(\lambda_{min}^{-1}, \lambda_{max}^{-1}) = (-1, 1)$, although the beta prior on ρ indicates certainty that the predominant spatial patterns will be non-negative. These are weakly informative prior distributions (see 30, p. 55-57).

The mortality data include 237 counties for which the deaths have been censored by CDC. For confidentiality, CDC suppresses “all sub-national data representing zero to nine deaths or births” (31). Bayesian inference provides straightforward methods to model censored data (3, 32, 33). In this case, each censored datum indicates a death count

between zero and nine; hence, for a given population at risk and mortality rate, the likelihood statement for a censored observation is (see 3, pp. 80-81):

$$p(Y(s_i) < 10 | \phi(s_i), P(s_i)) = \sum_{\tilde{y}=0}^9 \text{Pois}(\tilde{y} | \phi(s_i) \cdot P(s_i)). \quad (16)$$

The log probability of the $n_1 = 237$ censored observations, conditional on a given $P(s_i)$ and $\phi(s_i)$, is the sum of each of their respective log probabilities, where each is specified by Equation 16. Thus, the log-likelihood of all $N = n_1 + n_2$ observations can be calculated as the sum of the log-likelihoods of n_1 suppressed death counts (Equation 16) and n_2 observed death counts (Equation 13). Because the target of our inference is ϕ , there is no need to obtain posterior probability distributions for the unobserved counts themselves.

Listing 3 contains the Stan code for the above model. The first section of the code, the functions block, directs Stan to read the functions stored in the file `CAR-functions.stan`, including `wcar_normal_lpdf`. The data block specifies the structure of the input data (type and dimensions); notice that the censored observations are identified by their index value (`censored_idx`). Unknown parameters are declared in the parameters block and the probability model itself is specified in the model block. Equation 16 is implemented using the log-cumulative distribution function for the Poisson distribution (`poisson_lcdf`). Each `target +` statement increments the total log probability by the specified amount; this is used by Stan's HCM algorithm.

Notice that the CAR model is specified following Equation 13, which has the CAR model centered on the intercept ($\mu = \alpha \cdot \mathbf{1} \rightarrow \text{mu_phi} = \text{rep_vector}(\alpha, n)$). This is mathematically identical, but computationally distinct, from the following zero-mean (centered) CAR parameterization:

$$\begin{aligned} Y(s_i) &\sim \text{Poisson}(P(s_i) \cdot e^{\alpha + \phi(s_i)}) \\ \phi(s) &\sim \text{Gaussian}(\mathbf{0}, \Sigma) \end{aligned} \quad (17)$$

However, it is important in practice to know that the CAR models show far superior performance with the parameterization found in Equation 13 and Listing 3.

5.3 Demonstration analysis

We start the analysis by loading required R packages and reading the county mortality data and spatial connectivity matrix (a sparse matrix) into R.

```
> pkgs <- c("rstan", "geostan", "tidyverse", "tigris")
> lapply(pkgs, require, character.only = TRUE)
> mort <- read_csv("female-55-64-county-mortality-2014-2018.txt")
> sdf <- tigris::counties(year = 2018, cb = TRUE, res = "20m")
> sdf <- merge(sdf, mort, by = "GEOID")
> A <- read_rds("connectivity.rds")
```

The `geostan::prep_car_data` function requires the binary connectivity matrix, `A`, as its input, and returns the list of data required by the CAR model. We specify the WCAR option using the `style` argument:

```
> car.dl <- prep_car_data(A, style = "WCAR")
Range of permissible rho values: -1 1
```

The `car.dl` object is a list with all of the data that is specific to the CAR model; we will now add to that list our mortality data with log-population at risk as well as the index positions of the censored and uncensored observations:

```
> disease.dl <- list(n_censored = sum(is.na(df$deaths.female)),
+                   n_observed = sum(!is.na(df$deaths.female)),
+                   censored_idx = which(is.na(df$deaths.female)),
+                   observed_idx = which(!is.na(df$deaths.female)),
+                   censor_point = 9,
+                   y = as.numeric(na.omit(df$deaths.female)),
+                   at_risk = df$pop.at.risk.female
+                   )
> dl <- c(disease.dl, wcar.dl)
```

```

functions {
#include CAR-functions.stan
}

data {
  // mortality data
  int n_censored;
  int n_observed;
  int censored_idx[n_censored];
  int observed_idx[n_observed];
  int n;
  int censor_point;
  int y[n_observed];
  vector[n] at_risk;
  // CAR data
  vector[n] Delta_inv;
  real log_det_Delta_inv;
  vector[n] lambda;
  int nAx_w;
  vector[nAx_w] Ax_w;
  int Ax_v[nAx_w];
  int Ax_u[n + 1];
}

parameters {
  real alpha;
  vector[n] phi;
  real<lower=0> tau;
  real<lower=0, upper=1/max(lambda)> rho;
}

model {
  vector[n_observed] mu_y = at_risk[observed_idx] .* exp( phi[observed_idx] );
  vector[n_censored] mu_censored = at_risk[censored_idx] .* exp( phi[
    censored_idx] );
  vector[n] mu_phi = rep_vector(alpha, n);
  target += poisson_lpmf(y | mu_y);
  target += poisson_lcdf(censor_point | mu_censored);
  target += wcar_normal_lpdf(phi | mu_phi, tau, rho,
    Ax_w, Ax_v, Ax_u,
    Delta_inv,
    log_det_Delta_inv,
    lambda, n);
  target += normal_lpdf(alpha | -5, 2);
  target += beta_lpdf(rho | 2, 1);
  target += std_normal_lpdf(tau);
}

```

Listing 3: Stan code for the US county female mortality model (mortality-wcar.stan).

It is often unnecessary to provide Stan with initial values for its MCMC algorithm. The model for censored observations, however, typically requires the additional guidance that comes from setting initial values for the MCMC algorithm. The following R code chunk creates a list of functions that Stan will use to set the initial values of the MCMC chains. The functions point the algorithm in the direction of the prior distributions:

```
> init_f <- function () list(alpha = rnorm(1, -5, 1),
+                             rho = rbeta(n=1, 1.5, 1),
+                             tau = rgamma(n=1, 2, 2),
+                             phi = rnorm(n=dl$n, -5, 1)
+                             )
```

The Stan model from Listing 3 is stored in the file `mortality-wcar-normal.stan`; provided with this file, `rstan::stan_model` compiles the model into C++. The `rstan::sampling` function draws samples from the posterior distribution of model parameters using Stan's HMC algorithm. We will initiate four independent MCMC chains (`chains = 4`) in parallel (`cores = 4`), draw 600 warmup draws from each, and then draw 600 samples from each for the analysis (`iter = 1200`). This will provide 2,200 post-warmup samples from the joint posterior distribution. We also pass in our list of data and `init_f`; setting `refresh = 0` quiets some of the default printing behavior:

```
> car_model <- stan_model('mortality-wcar.stan')
> S <- sampling(car_model,
+              data = dl,
+              iter = 1200,
+              chains = 4,
+              cores = 4,
+              refresh = 0,
+              init = init_f
+              )
```

This required ≈ 1.6 minutes to finish sampling. Note that Stan will automatically warn of any problematic HMC diagnostics, including 'divergent transitions' indicating a failure of the algorithm to adequately explore the joint posterior distribution (9, 34, 35). In this case, there were no warnings. The following code gathers a variety of MCMC diagnostics, including the split $\hat{R}$ statistic (to evaluate if the MCMC chains converged to a single distribution, as indicated by $\hat{R}$ near to 1; values greater than about 1.05 may be reason for concern) and ESS, which accounts for autocorrelation in the chains (30). Again, the results are reassuring:

```
> m <- monitor(as.array(S), print = FALSE)
> range(m$Bulk_ESS)
[1] 483 3666
> range(m$Rhat)
[1] 0.9976705 1.0268612
```

The lowest ESS (483) is sufficient for this analysis, although it is not a measure of ESS for any single model parameter; instead, this happens to be the ESS for the total log-probability of the model (`lp_`). A number of parameters have ESS larger than the nominal sample size, a (desirable) product of negative autocorrelation in the chains.

Figure 2 shows histograms of the posterior samples for the three CAR model parameters (α , ρ , and τ), obtained by:

```
> plot(S, pars = c('alpha', 'rho', 'tau'), plotfun = 'hist')
```

The mean county mortality rate is around $e^{-4.89} = 752$ deaths per 100,000, the posterior of ρ is centered on 0.98 and τ on 0.51. Figures 3a and 3b are maps of the means and standard deviations of the probability distributions for the county mortality rates. These posterior summary values were obtained using the following code:

```
> phi <- exp(as.matrix(S, pars = "phi"))
> phi.mu <- apply(phi, 2, mean)
> phi.sd <- apply(phi, 2, sd)
```

5.4 Autocorrelation mixtures and residual analysis

The CAR model provided a principled method for obtaining estimates of mortality rates for the counties with suppressed data, and smoothed over some of the 'noise' in low-population areas. However, there are important limitations to

Figure 2: Histograms of samples from the posterior distributions of parameters (α , ρ , τ) from the CAR county mortality model.

Figure 3: Posterior summaries of county female mortality per 100,000, ages 55-64, years 2014-2016.

spatial smoothing. Many real datasets contain a mixture of positive and negative SA in which the positive component outweighs the negative (36, 37). Positive SA patterns indicate a degree of similarity across nearby areas, where negative SA is characterized by low values surrounded by high values and/or *vice versa*. Native American reservations, often surrounded by predominantly White rural areas, provide an important case of negative SA which requires attention in most real modeling exercises with U.S. county data. Table 3 shows that the counties whose crude values were given the largest ‘adjustment’ by the model are rural, majority non-White counties surrounded by majority White counties. For example, Buffalo County, South Dakota—primarily the Crow Creek Reservation—has a crude female mortality of 2,141 per 100,000 but a model estimate of just 926, a dramatic reduction toward the area average. These estimates are highly suspect, and in the case of this mortality data, would be best addressed in a serious study by adding pertinent covariates and, if possible, examining previous years of mortality data for any areas in question.

6 Conclusion

This paper has introduced code for building spatial CAR models using the Stan modeling language. The functions provided here are highly efficient implementations of the CAR model, which is crucial for supporting more computationally intensive but important tasks, such as modeling measurement error in spatial covariates (6) or extending into the spatio-temporal domain (19). The code provided for the demonstration analysis can be extended to build custom spatial models. The need for thoughtful spatial analysis of residuals has been emphasized, with the case of mortality on Native American reservations highlighting the seriousness of this task. The *geostan* package provides, among other things, helpful functions for converting spatial connectivity matrices into the required format for various CAR model

Table 3: Five largest residuals by absolute value. Lag-white is the average percent-White of neighboring counties.

County	Demographics					Mortality per 100,000		
	At risk	White	Lag-White	Native	Black	Crude	Modeled	Residual
Buffalo, SD	467	0.14	0.79	0.81	0.00	2,141	926	1,214
Dewey, SD	1456	0.20	0.89	0.78	0.00	1,579	879	699
Sioux, ND	962	0.13	0.76	0.81	0.00	1,559	895	663
Mahnomen, MN	1680	0.45	0.93	0.38	0.01	1,428	854	573
East Carroll, LA	2187	0.29	0.66	0.00	0.70	2,057	1,513	544

specifications. While not the focus of this paper, *geostan* also provides a more accessible interface to pre-compiled versions of these CAR models which may be suitable for many routine analyses.

Computational details

The results in this paper were obtained using R 4.0.4 with Rstan 2.21.3, *geostan* 0.2.1, *posterior* 1.1.0, *coda* 0.19-4, *tidyverse* 1.3.1, *tigris* 1.5, and *nimble* 0.11.1, on a Dell xps 13 laptop with the Ubuntu 20.04 operating system and Intel Core i7-8565U CPU 1.80GHz $\times$ 8 processor. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/>.

References

- (1) Julian Besag. “Spatial interaction and the statistical analysis of lattice systems”. In: *Journal of the Royal Statistical Society B* 36.2 (1974), pp. 192–225.
- (2) Bob Carpenter et al. “Stan: A probabilistic programming language”. In: *Journal of statistical software* 76.1 (2017), pp. 1–32.
- (3) Stan Development Team. *Stan User’s Guide*. Version 2.27. 2019.
- (4) Hong Xia and Bradley P Carlin. “Spatio-temporal models with errors in covariates: mapping Ohio lung cancer mortality”. In: *Statistics in Medicine* 17.18 (1998), pp. 2025–2043.
- (5) Emily L. Kang, Desheng Liu, and Noel Cressie. “Statistical analysis of small-area data based on independence, spatial, non-hierarchical, and hierarchical models”. In: *Computational Statistics & Data Analysis* 53 (8 2009), pp. 3016–3032.
- (6) Connor Donegan, Yongwan Chun, and Daniel A. Griffith. “Modeling community health with areal data: Bayesian inference with survey standard errors and spatial structure”. In: *Int. J. Env. Res. Public Health* 18.13 (2021), p. 6856. DOI: 10.3390/ijerph18136856.
- (7) David MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- (8) Radford M Neal. “MCMC using Hamiltonian dynamics”. In: *Handbook of Markov chain Monte Carlo*. Ed. by Steve Brooks et al. CRC Press, 2011.
- (9) Michael Betancourt. “A conceptual introduction to Hamiltonian Monte Carlo”. In: *arXiv preprint https://arxiv.org/abs/1701.02434v2* (2017).
- (10) David Lunn et al. “The BUGS project: Evolution, critique and future directions”. In: *Statistics in medicine* 28.25 (2009), pp. 3049–3067.
- (11) Sarah Depaoli, James P Clifton, and Patrice R Cobb. “Just another Gibbs sampler (JAGS) flexible software for MCMC implementation”. In: *Journal of Educational and Behavioral Statistics* 41.6 (2016), pp. 628–649.
- (12) Andrew Thomas et al. *GeoBUGS User Manual*. 2004. URL: <https://www.mrc-bsu.cam.ac.uk/wp-content/uploads/geobugs12manual.pdf>.
- (13) P. de Valpine et al. “Programming with models: writing statistical algorithms for general model structures with NIMBLE”. In: *Journal of Computational and Graphical Statistics* 26 (2017), pp. 403–417. DOI: 10.1080/10618600.2016.1172487.
- (14) Levi John Wolf, Luc Anselin, and Daniel Arribas-Bel. “Stochastic efficiency of Bayesian Markov chain Monte Carlo in spatial econometric models: An empirical comparison of exact sampling methods”. In: *Geographical Analysis* 50.1 (2018), pp. 97–119.

- (15) Michael Betancourt and Mark Girolami. “Hamiltonian Monte Carlo for hierarchical models”. In: *Current Trends in Bayesian Methodology with Applications*. Ed. by Satyanshu K. Upadhyay et al. CRC Press, 2015.
- (16) Mitzi Morris et al. “Bayesian hierarchical spatial models: Implementing the Besag York Mollié model in stan”. In: *Spatial and Spatio-temporal Epidemiology* 31 (2019), p. 100301.
- (17) Connor Donegan, Yongwan Chun, and Amy E Hughes. “Bayesian estimation of spatial filters with Moran’s eigenvectors and hierarchical shrinkage priors”. In: *Spatial Statistics* (2020), p. 100450.
- (18) Max Joseph. “Exact sparse CAR models in Stan”. In: *Stan Case Studies* 3 (2016). URL: <https://mc-stan.org/users/documentation/case-studies/mbjoseph-CARStan.html>.
- (19) Noel Cressie and Christopher K. Wikle. *Statistics for Spatio-Temporal Data*. John Wiley & Sons, 2011.
- (20) Robert P. Haining and Guangquan Li. *Modelling Spatial and Spatio-Temporal Data: A Bayesian Approach*. CRC Press, 2020.
- (21) Noel Cressie, Olivier Perrin, and Christine Thomas-Agnan. “Likelihood-based estimation for Gaussian MRFs”. In: *Statistical Methodology* 2.1 (2005), pp. 1–16.
- (22) Stephan R Sain and Noel Cressie. “A spatial model for multivariate lattice data”. In: *Journal of Econometrics* 140.1 (2007), pp. 226–259.
- (23) Noel Cressie. *Statistics for Spatial Data*. Revised edition [1993]. John Wiley & Sons, 2015.
- (24) AD Cliff and JK Ord. *Spatial Processes: Models and Applications*. Pion, 1981.
- (25) Centers for Disease Control and Prevention, National Center for Health Statistics. *Underlying Cause of Death 1999-2018 on CDC Wonder Online Database*. Accessed at <http://wonder.cdc.gov/ucd-iiid10.html> October 19 2020. 2020.
- (26) Aki Vehtari et al. “Rank-normalization, folding, and localization: An improved R-hat for assessing convergence of MCMC”. In: *arXiv preprint <https://arxiv.org/abs/1903.08008>* (2019).
- (27) Stan Development Team. *RStan: the R interface to Stan*. R package version 2.21.2. 2020. URL: <http://mc-stan.org/>.
- (28) Nancy Krieger. *Epidemiology and the People’s Health*. Oxford University Press, 2011.
- (29) R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2021. URL: <https://www.R-project.org/>.
- (30) Andrew Gelman et al. *Bayesian Data Analysis*. CRC Press, 2014.
- (31) Centers for Disease Control and Prevention. *Compressed Mortality File: Years 1968-1978 with ICD-8 Codes, 1979-1998 with ICD-9 Codes and 1999-2016 with ICD-10 Codes*. This page last reviewed: Tuesday, May 19, 2020. Accessed August 18, 2021. 2020. URL: <https://wonder.cdc.gov/wonder/help/cmef.html#>.
- (32) Harrison Quick. “Estimating County-Level Mortality Rates Using Highly Censored Data from CDC WONDER”. In: *Preventing Chronic Disease* 16 (2019), p. 180441. DOI: 10.5888/pcd16.180441.
- (33) Pavel Chernyavskiy, Mark P. Little, and Philip S. Rosenberg. “Spatially varying age-period-cohort analysis with application to US mortality, 2002-2016”. In: *Biostatistics* 21.4 (2020), pp. 845–859. DOI: 10.1093/biostatistics/kxz009.
- (34) Michael Betancourt. “Robust Statistical Workflow with RStan”. In: *Stan Case Studies* 4 (2017). https://mc-stan.org/users/documentation/case-studies/rstan_workflow.html.
- (35) Michael Betancourt. “Identifying Biased Inference with Divergences”. In: *Stan Case Studies* 4 (2017). https://mc-stan.org/users/documentation/case-studies/divergences_and_bias.html.
- (36) Daniel A Griffith. “Negative spatial autocorrelation: One of the most neglected concepts in spatial statistics”. In: *Stats* 2.3 (2019), pp. 388–415.
- (37) Daniel A Griffith et al. “Geospatial socio-economic/demographic data: The existence of spatial autocorrelation mixtures in georeferenced data—Part I”. In: *Transactions in GIS* (2021).