

Applying an Intrinsic Conditional Autoregressive Reference Prior for Areal Data

Erica M. Porter

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Statistics

Christopher T. Franck, Chair
Marco A.R. Ferreira
Leanna L. House

April 5, 2019
Blacksburg, Virginia

Keywords: Bayesian Analysis, Spatial Statistics
Copyright 2019, Erica M. Porter

Applying an Intrinsic Conditional Autoregressive Reference Prior for Areal Data

Erica M. Porter

ABSTRACT

Bayesian hierarchical models are useful for modeling spatial data because they have flexibility to accommodate complicated dependencies that are common to spatial data. In particular, intrinsic conditional autoregressive (ICAR) models are commonly assigned as priors for spatial random effects in hierarchical models for areal data corresponding to spatial partitions of a region. However, selection of prior distributions for these spatial parameters presents a challenge to researchers. We present and describe **ref.ICAR**, an R package that implements an objective Bayes intrinsic conditional autoregressive prior on a vector of spatial random effects. This model provides an objective Bayesian approach for modeling spatially correlated areal data. **ref.ICAR** enables analysis of spatial areal data for a specified region, given user-provided data and information about the structure of the study region. The **ref.ICAR** package performs Markov Chain Monte Carlo (MCMC) sampling and outputs posterior medians, intervals, and trace plots for fixed effect and spatial parameters. Finally, the functions provide regional summaries, including medians and credible intervals for fitted values by subregion.

Applying an Intrinsic Conditional Autoregressive Reference Prior for Areal Data

Erica M. Porter

GENERAL AUDIENCE ABSTRACT

Spatial data is increasingly relevant in a wide variety of research areas. Economists, medical researchers, ecologists, and policymakers all make critical decisions about populations using data that naturally display spatial dependence. One such data type is areal data; data collected at county, habitat, or tract levels are often spatially related. Most convenient software platforms provide analyses for independent data, as the introduction of spatial dependence increases the complexity of corresponding models and computation. Use of analyses with an independent data assumption can lead researchers and policymakers to make incorrect, simplistic decisions. Bayesian hierarchical models can be used to effectively model areal data because they have flexibility to accommodate complicated dependencies that are common to spatial data. However, use of hierarchical models increases the number of model parameters and requires specification of prior distributions. We present and describe **ref.ICAR**, an R package available to researchers that automatically implements an objective Bayesian analysis that is appropriate for areal data.

Grant Information

The development of the R package presented in this thesis was supported by the Virginia Tech College of Science Dean's Discovery Fund.

Acknowledgements

I am extremely grateful to my advisor, Dr. Chris Franck, for believing in me and involving me in research. You have provided me with invaluable support, enthusiasm, knowledge, and opportunity. I have enjoyed and appreciated every research challenge we encountered. You are an unfailingly brilliant and kind professor, and I am so fortunate to have your example.

I would also like to thank Dr. Ferreira for his guidance as graduate director and for greatly improving my understanding of spatial and Bayesian statistics. Your expertise and advice have been critical to my early development as a researcher and statistician.

Furthermore, I thank my friends and members of my research group at Virginia Tech, Steve Walsh, Tom Metzger, and Mohamed El Khouly. You have all taught me about presenting, proofreading, and publishing research, and I hope to apply all of the lessons which you have shared. I would also like to thank Matt Keefe, for providing wonderful notes and positive comments as I began my research in a subject that was new to me.

To Sierra Merkes, Shane Bookhultz, and all of my friends in the Virginia Tech Department of Statistics, thank you for listening, working problems with me, and providing new perspectives on research and life. You have made this department my home, and I am lucky to work alongside each of you every day.

Finally, thank you to my twin brother, Joseph, for your support and constant unrealistic optimism this year. I likely would not have found research if it hadn't been for you. More importantly, you remind me to laugh and to appreciate the small accomplishments. I am so proud and grateful to you.

Contents

1	Introduction	1
1.1	Background	1
1.2	Package Overview	3
1.3	Thesis Outline	3
2	Package Documentation	5
3	Package Vignette	16
4	Conclusion	28
	Appendices	31
A	ref.ICAR Code	31

Chapter 1

Introduction

1.1 Background

A number of fields require analysis of spatial data to answer relevant research questions. Similarities in proximate data points are of interest in ecology, econometrics, sociology, epidemiology, and other areas. The interest is usually both in identifying the underlying spatial pattern and in accounting for spatial correlation to properly identify relationships between variables. Failing to account for spatial structure in the data can produce highly inaccurate analyses. However, the introduction of spatial parameters and distance structures significantly increases the complexity of models to analyze areal data, often making such analyses inaccessible to researchers. Therefore, development of theory and software for the analysis of spatial data is a research need that must be addressed.

One type of spatial data that is of interest when studying a defined region is areal data. Areal data consists of aggregated values corresponding to subregions that spatially partition a finite region and is of particular interest in fields including ecology (Hoef et al., 2018), sociology (Goodchild and Janelle, 2004), and epidemiology (Lee, 2011). For example, within sociology and public policy, areal data can be used to describe distribution of living conditions by

counties (Keefe et al., 2019) or census tracts (Hogan and Tchernis, 2004). In some cases, particularly in health applications, individual-level data is anonymous, but aggregated data for sub-populations is available to researchers, making areal data models an essential tool for analysis (Goodchild and Janelle, 2004). Ecology studies often analyze satellite images and data collected on a grid (Zhu et al., 2010) or irregular subregions formed by natural and managed habitats (Hoef et al., 2018). In each of these cases, the response of interest is measured over a finite number of fixed subregions such as counties, census tracts, districts, and other functional areas that are defined by public policy, scientific regions, or natural processes.

Bayesian hierarchical models, including conditional autoregressive (CAR) models (Besag, 1974), are commonly used to model areal data due to their ability to accommodate both fixed effects and spatial dependence parameters. Conditional autoregressive models use the full conditional distribution for each subregion in the data region, which depends on the responses recorded for neighboring subregions. In the Bayesian hierarchical model approach, an intrinsic conditional autoregressive (ICAR) model (Besag et al., 1991) is often used to model the spatial random effects in the hierarchical model.

In order to define the conditional distributions for the ICAR model, researchers must select prior distributions for the parameters of the ICAR model. Selection of priors can be difficult when there is a lack of information available about the response to specify effective prior distributions and hyperparameters. To address this problem for researchers analyzing areal data, Keefe et al. (2019) introduced an objective prior for the parameters of an intrinsic conditional autoregressive model for the spatial random effects in a Gaussian hierarchical model. This is the first in the literature to act as an automatic prior for the ICAR model, conveniently removing the need for researchers to select priors and hyperparameters. Keefe et al. (2019) derive an explicit expression for a reference prior and demonstrate, through

simulations and a case study, that the reference prior for the ICAR model has good frequentist properties and performs well under various strengths of spatial dependence. Prior to the work of Keefe et al. (2019), objective Bayesian analysis was not available for areal data via conditional autoregressive models.

1.2 Package Overview

This thesis presents **ref.ICAR**, an R package that implements the objective Bayes ICAR model and reference prior for a hierarchical model with a vector of spatial random effects for areal data (Keefe et al., 2019). **ref.ICAR** makes available functions that construct a neighborhood matrix, implement the ICAR reference prior, perform Markov Chain Monte Carlo (MCMC) sampling for the model parameters, provide parameter estimates and intervals, provide regional medians and intervals for fitted values by subregion, and create trace plots for all model parameters. The functions in **ref.ICAR** can be used to analyze an areal data set and automatically perform the entire analysis given a user-provided shapefile, or a specified neighborhood matrix.

1.3 Thesis Outline

This thesis contains the documentation for the R package **ref.ICAR**, published on the Comprehensive R Archive Network (CRAN) (Porter et al., 2018). Chapter 2 consists of the **ref.ICAR** package documentation. This documentation lists and describes the purpose, usage, arguments, and outputs for each function contained in **ref.ICAR**. The documentation also provides short examples for using each of the functions. Chapter 3 is a vignette that gives a detailed example for using the **ref.ICAR** package. Chapter 4 concludes by summarizing

the package's uses and relevance to further spatial statistics research. Finally, the code for the functions in **ref.ICAR** is presented in the Appendix.

Chapter 2

Package Documentation

Package ‘ref.ICAR’

November 19, 2018

Title Objective Bayes Intrinsic Conditional Autoregressive Model for Areal Data

Version 1.0

Author Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Maintainer Erica M. Porter <ericamp@vt.edu>

Depends R (>= 3.1.0)

Description Implements an objective Bayes intrinsic conditional autoregressive prior. This model provides an objective Bayesian approach for modeling spatially correlated areal data using an intrinsic conditional autoregressive prior on a vector of spatial random effects.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports rgdal, spdep, mvtnorm, coda, MCMCglmm, Rdpack, graphics

RdMacros Rdpack

Suggests maptools, maps, MASS, sp, knitr, rmarkdown, RColorBrewer, captioner, rcrossref

VignetteBuilder knitr

BuildManual yes

RoxygenNote 6.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-19 18:00:06 UTC

R topics documented:

```
ref.ICAR-package . . . . .
ref.analysis . . . . .
ref.MCMC . . . . .
```

```

ref.plot . . . . .
ref.summary . . . . .
reg.summary . . . . .
shape.H . . . . .

```

ref.ICAR-package	<i>Objective Bayes Intrinsic Conditional Autoregressive Model for Areal Data</i>
------------------	--

Description

Implements the objective prior for intrinsic conditional autoregressive (ICAR) random effects proposed by Keefe et al. (2018). This model provides an objective Bayesian approach for modeling spatially correlated areal data using an ICAR prior on a vector of spatial random effects.

Details

```

Package: ref.ICAR
Type: Package
Version: 1.0
Date: 2018-10-10
License: GPL (>= 2)

```

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira
 Maintainer: Erica M. Porter <ericamp@vt.edu>

References

Keefe MJ, Ferreira MAR, Franck CT (2018). "Objective Bayesian Analysis for Gaussian Hierarchical Models with Intrinsic Conditional Autoregressive Priors." *Bayesian Analysis*. doi: 10.1214/18BA1107, <https://doi.org/10.1214/18-BA1107>.

Keefe MJ, Ferreira MAR, Franck CT (2018). "On the formal specification of sum-zero constrained intrinsic conditional autoregressive models." *Spatial Statistics*, **24**, 54–65. doi: 10.1016/j.spasta.2018.03.007, <https://doi.org/10.1016/j.spasta.2018.03.007>.

Examples

```

## Refer to the vignette attached to the package,
## and to selected help files.

```

ref.analysis	<i>MCMC Analysis and Summaries for Reference Prior on an Intrinsic Autoregressive Model for Areal Data</i>
--------------	--

Description

Performs analysis on a geographical areal data set using the objective prior for intrinsic conditional autoregressive (ICAR) random effects (Keefe et al. 2018). It takes a shapefile, data, and region names to construct a neighborhood matrix and perform Markov chain Monte Carlo sampling on the unstructured and spatial random effects. Finally, the function obtains regional estimates and performs posterior inference on the model parameters.

Usage

```
ref.analysis(shape.file, X, y, x.reg.names, y.reg.names,
  shp.reg.names = NULL, iters = 10000, burnin = 5000,
  verbose = TRUE, tauc.start = 1, beta.start = 1, sigma2.start = 1,
  step.tauc = 0.5, step.sigma2 = 0.5)
```

Arguments

shape.file	A shapefile corresponding to the regions for analysis.
X	A matrix of covariates, which should include a column of 1's for models with a non-zero intercept
y	A vector of responses.
x.reg.names	A vector specifying the order of region names contained in X.
y.reg.names	A vector specifying the order of region names contained in y.
shp.reg.names	A vector specifying the order of region names contained in the shapefile, if there is not a NAME column in the file.
iters	Number of MCMC iterations to perform. Defaults to 10,000.
burnin	Number of MCMC iterations to discard as burn-in. Defaults to 5,000.
verbose	If FALSE, MCMC progress is not printed.
tauc.start	Starting MCMC value for the spatial dependence parameter.
beta.start	Starting MCMC value for the fixed effect regression coefficients.
sigma2.start	Starting MCMC value for the variance of the unstructured random effects.
step.tauc	Step size for the spatial dependence parameter.
step.sigma2	Step size for the variance of the unstructured random effects.

Value

A list containing H, MCMC chains, parameter summaries, fitted regional values, and regional summaries.

H	The neighborhood matrix.
MCMC	Matrix of MCMC chains for all model parameters.
beta.median	Posterior medians of the fixed effect regression coefficients.
beta.hpd	Highest Posterior Density intervals for the fixed effect regression coefficients.
tauc.median	Posterior median of the spatial dependence parameter.
tauc.hpd	Highest Posterior Density interval for the spatial dependence parameter.
sigma2.median	Posterior median of the unstructured random effects variance.
sigma2.hpd	Highest Posterior Density interval for the unstructured random effects variance.
tauc.accept	Final acceptance rate for the spatial dependence parameter.
sigma2.accept	Final acceptance rate for the unstructured random effects variance.
fit.dist	Matrix of fitted posterior values for each region in the data.
reg.medians	Vector of posterior medians for fitted response by region.
reg.hpd	Data frame of Highest Posterior Density intervals by region.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Examples

```
## Refer to the vignette attached to the package.
```

ref.MCMC

*MCMC for Reference Prior on an Intrinsic Conditional Autoregressive
Random Effects Model for Areal Data*

Description

Implements the Metropolis-within-Gibbs sampling algorithm proposed by Keefe et al. (2018), to perform posterior inference for the intrinsic conditional autoregressive model with spatial random effects.

Usage

```
ref.MCMC(y, X, H, iters = 10000, burnin = 5000, verbose = TRUE,  
  tauc.start = 1, beta.start = 1, sigma2.start = 1,  
  step.tauc = 0.5, step.sigma2 = 0.5)
```


Arguments

<code>y</code>	Vector of responses.
<code>X</code>	Matrix of covariates. This should include a column of 1's for models with a non-zero intercept.
<code>H</code>	The neighborhood matrix.
<code>iters</code>	Number of MCMC iterations to perform. Defaults to 10,000.
<code>burnin</code>	Number of MCMC iterations to discard as burn-in. Defaults to 5,000.
<code>verbose</code>	If FALSE, MCMC progress is not printed.
<code>tauc.start</code>	Starting value for the spatial dependence parameter.
<code>beta.start</code>	Starting value for the vector of fixed effect regression coefficients.
<code>sigma2.start</code>	Starting value for the variance of the unstructured random effects.
<code>step.tauc</code>	Step size for the spatial dependence parameter
<code>step.sigma2</code>	Step size for the variance of the unstructured random effects.

Value

A list containing MCMC chains and parameter summaries.

<code>MCMCchain</code>	Matrix of MCMC chains.
<code>tauc.MCMC</code>	MCMC chains for the spatial dependence parameter.
<code>sigma2.MCMC</code>	MCMC chains for the variance of the unstructured random effects.
<code>phi.MCMC</code>	MCMC chains for the spatial random effects.
<code>beta.MCMC</code>	MCMC chains for the fixed effect regression coefficients.
<code>accept.sigma2</code>	Final acceptance number for variance of the unstructured random effects.
<code>accept.tauc</code>	Final acceptance number for spatial dependence parameter.
<code>accept.phi</code>	Final acceptance number for spatial random effects.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

References

- Keefe MJ, Ferreira MAR, Franck CT (2018). "Objective Bayesian Analysis for Gaussian Hierarchical Models with Intrinsic Conditional Autoregressive Priors." *Bayesian Analysis*. doi: 10.1214/18BA1107, <https://doi.org/10.1214/18-BA1107>.
- Keefe MJ, Ferreira MAR, Franck CT (2018). "On the formal specification of sum-zero constrained intrinsic conditional autoregressive models." *Spatial Statistics*, **24**, 54–65. doi: 10.1016/j.spasta.2018.03.007, <https://doi.org/10.1016/j.spasta.2018.03.007>.

Examples

```
#### Fit the model for simulated areal data on a grid ####

### Load extra libraries
library(sp)
library(methods)
library(spdep)
library(mvtnorm)

### Generate areal data on a grid
rows=5; cols=5
tauc=1
sigma2=2; beta=c(1,5)

### Create grid
grid <- GridTopology(c(1,1), c(1,1), c(cols,rows))
polys <- as(grid, "SpatialPolygons")
spgrid <- SpatialPolygonsDataFrame(polys,data=data.frame(row.names=row.names(polys)))

### Create neighborhood matrix
grid.nb <- poly2nb(spgrid,queen=FALSE)
W <- nb2mat(grid.nb, style="B")

### Put spatially correlated data in grid
p <- length(beta)
num.reg <- (rows*cols)
if(p>1){x1<-rmvnorm(n=num.reg,mean=rep(0,p-1),sigma=diag(p-1))} else{x1<-NULL}
X <- cbind(rep(1,num.reg),x1)
Dmat <- diag(apply(W,1,sum))
H <- Dmat - W
row.names(H) <- NULL

### Obtain true response vector
theta_true <- rnorm(num.reg,mean=0,sd=sqrt(sigma2))
Q <- eigen(H,symmetric=TRUE)$vectors
eigH <- eigen(H,symmetric=TRUE)$values
D <- diag(eigH)
Qmat <- Q[,1:(num.reg-1)]
phimat <- diag(1/sqrt(eigH[1:(num.reg-1)]))
z <- t(rmvnorm(1,mean=rep(0,num.reg-1),sigma=diag(num.reg-1)))
phi_true <- sqrt((1/tauc)*sigma2)*(Qmat%*%phimat%*%z)
Y <- X%*%beta + theta_true + phi_true

### Fit the model
set.seed(5432)
model <- ref.MCMC(y=Y,X=X,H=H,itters=15000,burnin=5000,verbose=TRUE,tauc.start=.1,beta.start=-1,
sigma2.start=.1,step.tauc=0.5,
step.sigma2=0.5)

#### Small example for checking
model <- ref.MCMC(y=Y,X=X,H=H,itters=1000,burnin=50,verbose=TRUE,tauc.start=.1,beta.start=-1,
```

```
sigma2.start=.1,step.tauc=0.5,  
step.sigma2=0.5)
```

`ref.plot`*Trace Plots for Parameters in ICAR Model*

Description

This function creates trace plots for the parameters in the ICAR reference prior model (Keefe et al. 2018).

Usage

```
ref.plot(MCMCchain, X, burnin, num.reg)
```

Arguments

<code>MCMCchain</code>	Matrix of MCMC chains for the model parameters.
<code>X</code>	Matrix of covariates.
<code>burnin</code>	Number of MCMC iterations from <code>MCMCchain</code> discarded as burn-in.
<code>num.reg</code>	Number of regions in the areal data set.

Value

Trace plots for the fixed effect regression coefficients, the precision parameter, and the unstructured random effects variance.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Examples

```
## Refer to the vignette attached to the package.
```

ref.summary

*Parameter Summaries for MCMC Analysis***Description**

Takes a matrix of MCMC chains, iterations, and acceptance values to return posterior summaries of the parameters, including posterior medians, intervals, and acceptance rates.

Usage

```
ref.summary(MCMCchain, tauc.MCMC, sigma2.MCMC, beta.MCMC, phi.MCMC,
  accept.phi, accept.sigma2, accept.tauc, iters = 10000, burnin = 5000)
```

Arguments

MCMCchain	Matrix of MCMC chains for the ICAR model parameters.
tauc.MCMC	MCMC chains for the spatial dependence parameter.
sigma2.MCMC	MCMC chains for the variance of the unstructured random effects.
beta.MCMC	MCMC chains for the fixed effect regression coefficients.
phi.MCMC	MCMC chains for the spatial random effects.
accept.phi	Final acceptance number for spatial random effects.
accept.sigma2	Final acceptance number for variance of the unstructured random effects.
accept.tauc	Final acceptance number for the spatial dependence parameter.
iters	Number of MCMC iterations in MCMCchain.
burnin	Number of MCMC iterations discarded as burn-in for MCMCchain.

Value

Parameter summaries

beta.median	Posterior medians of the fixed effect regression coefficients.
beta.hpd	Highest Posterior Density intervals for the fixed effect regression coefficients.
tauc.median	Posterior median of the spatial dependence parameter.
tauc.hpd	Highest Posterior Density interval for the spatial dependence parameter.
sigma2.median	Posterior median of the unstructured random effects variance.
sigma2.hpd	Highest Posterior Density interval for the unstructured random effects variance.
tauc.accept	Final acceptance rate for the spatial dependence parameter.
sigma2.accept	Final acceptance rate for the unstructured random effects variance.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Examples

```
## Refer to the vignette attached to the package.
```

reg.summary	<i>Regional Summaries for Areal Data Modeled by ICAR Reference Prior Model</i>
-------------	--

Description

This function takes data and sampled MCMC chains for an areal data set and gives fitted posterior values and summaries by region using the model by (Keefe et al. 2018).

Usage

```
reg.summary(MCMCchain, X, Y, burnin)
```

Arguments

MCMCchain	Matrix of MCMC chains, using the sampling from (Keefe et al. 2018).
X	Matrix of covariates.
Y	Vector of responses.
burnin	Number of MCMC iterations discarded as burn-in in MCMCchain.

Value

A list of the fitted distributions by region, and medians and credible intervals by region.

fit.dist	Matrix of fitted posterior values for each region in the data.
reg.medians	Vector of posterior medians for fitted response by region.
reg.cred	Data frame of credbile intervals by region.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Examples

```
## Refer to the vignette attached to the package.
```

`shape.H`*Creating a Neighborhood Matrix for Areal Data from a Shapefile*

Description

Takes a path to a shape file and creates a neighborhood matrix. This neighborhood matrix can be used with the objective ICAR model (Keefe et al. 2018).

Usage

```
shape.H(shape.file)
```

Arguments

`shape.file` File path to a shapefile.

Value

A list containing a neighborhood matrix and the `SpatialPolygonsDataFrame` object corresponding to the shape file.

`H` A neighborhood matrix.

`map` `SpatialPolygonsDataFrame` object from the provided shapefile.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Examples

```
#### Load extra libraries
library(sp)
library(rgdal)

### Read in a shapefile of the contiguous U.S. from package data
system.path <- system.file("extdata", "us.shape48.shp", package = "ref.ICAR", mustWork = TRUE)
shp.layer <- gsub('.shp', '', basename(system.path))
shp.path <- dirname(system.path)
us.shape48 <- readOGR(dsn = path.expand(shp.path), layer = shp.layer, verbose = FALSE)

shp.data <- shape.H(system.path)
names(shp.data)
```

Chapter 3

Package Vignette

Applying an ICAR reference prior

1 Introduction

The **ref.ICAR** package performs objective Bayesian analysis using a reference prior proposed by Keefe et al. (2018a). This model provides an approach for modeling spatially correlated areal data, using an intrinsic conditional autoregressive (ICAR) component with a reference prior on a vector of spatial random effects.

2 Functions

ref.ICAR can be used to analyze areal data corresponding to a contiguous region, provided a shapefile or neighborhood matrix and data. The functions implemented by **ref.ICAR** are summarized below.

- `shape.H()` Takes a file path to a shapefile and forms the appropriate neighborhood matrix for analysis with the ICAR reference prior model.
- `ref.MCMC()` Implements the posterior sampling algorithm proposed by Keefe et al. (2018a). Generates MCMC chains for parameter and regional inferences.
- `ref.summary()` Provides posterior inferences for the model parameters, τ_c , β , and σ^2 . Includes posterior medians, highest posterior density (HPD) intervals, and acceptance rates for each parameter.
- `reg.summary()` Provides fitted posterior values and summaries for each subregion in the areal data set. Includes posterior medians and HPD intervals by region.
- `ref.plot()` Outputs trace plots for the model parameters, τ_c , β , and σ^2 .
- `ref.analysis()` Performs analysis by sequentially implementing each of the functions above. This function produces plots and a list containing MCMC chains, parameter estimates, regional estimates, and sampling acceptance rates.

3 Model

The model implemented by **ref.ICAR** is summarized below.

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\theta} + \boldsymbol{\phi}$$

where

- \mathbf{Y} is an $n \times 1$ vector for the response variable, where n corresponds to the number of regions in the shapefile. The current version of the package does not allow for missing data.
- \mathbf{X} is a matrix of covariates. This can include a vector $\mathbf{1}_n$ for an intercept, and additional columns corresponding to quantitative predictors.
- $\boldsymbol{\beta}$ is the $p \times 1$ vector of fixed effect regression coefficients, where p corresponds to the number of columns in \mathbf{X} .
- $\boldsymbol{\theta}$ is an $n \times 1$ vector of independent and normally distributed unstructured random effects defined with mean 0 and variance σ^2 .

- ϕ is an $n \times 1$ vector of spatial random effects that is assigned an intrinsic CAR prior with the sum-zero constraint $\sum_{i=1}^n \phi_i = 0$ (Keefe et al. 2018b).

The model assumes a signal-to-noise ratio parameterization for the variance components of the random components of the model, so σ^2 and τ_c are used as below.

$$\phi \sim \left(\mathbf{0}, \frac{\sigma^2}{\tau_c} \Sigma_\phi \right)$$

The parameter τ_c controls the strength of spatial dependence, and given the neighborhood structure, Σ_ϕ is a fixed matrix. Specifically, Σ_ϕ is the Moore-Penrose inverse of H , where the neighborhood matrix H is an $n \times n$ symmetric matrix constructed as follows.

$$(H)_{ij} = \begin{cases} h_i & \text{if } i = j \\ -g_{ij} & \text{if } i \in N_j \\ 0 & \text{otherwise,} \end{cases}$$

where $g_{ij} = 1$ if subregions i and j are neighbors, $g_{ij} = 0$ if subregions i and j are not neighbors, and h_i is the number of neighbors of subregion i . Therefore, the neighborhood matrix H is an $n \times n$ symmetric matrix where the diagonal elements correspond to the number of neighbors for each subregion in the data, and each off-diagonal element equals -1 if the corresponding subregions are neighbors.

Provided a path to a shapefile, the `shape.H()` function in **ref.ICAR** constructs H as specified above, and checks for symmetry and contiguous regions (i.e. no islands) prior to analysis. The functions `shape.H()` and `ref.analysis()` requires a file path to a shapefile. If a user wants to analyze areal data without a corresponding shapefile (e.g. neuroimaging), they will need to construct H as above and use this H in `ref.MCMC()`. `ref.plot()`, `ref.summary()`, and `reg.summary()` can then be used with the MCMC chains obtained from `ref.MCMC()`. Additionally, if a user performs analysis without `ref.analysis()`, the regions corresponding to data values in X and y must match the region order in H ; otherwise inferences will be matched to incorrect regions.

4 Example

Consider an example of areal data over the contiguous United States. Figure 1 represents the average SAT scores reported in 1999 for each of the contiguous United States and Washington D.C. This example will explore these data and use the **ref.ICAR** package to fit a model to the response, Verbal SAT scores, considering spatial dependence and a single covariate, percent of eligible students that took the SAT in each state in 1999. This data was analyzed in *Hierarchical Modeling and Analysis for Spatial Data* (Banerjee et al. 2014). The data are available online at <https://www.counterpointstat.com/hierarchical-modeling-and-analysis-for-spatial-data.html>. We make it available in the **ref.ICAR** package with permission from the authors. The shapefile is found from <http://www.arcgis.com/home/item.html?id=f7f805eb65eb4ab787a0a3e1116ca7e5>.

These data and the accompanying shapefile are attached to the **ref.ICAR** package. The files can be loaded into R as shown below. The `readOGR()` function from package **rgdal** is used to read the shapefile.

```
library(rgdal)

system.path <- system.file("extdata", "us.shape48.shp", package = "ref.ICAR",
  mustWork = TRUE)
shp.layer <- gsub(".shp", "", basename(system.path))
```



```
shp.path <- dirname(system.path)

us.shape48 <- readOGR(dsn = path.expand(shp.path), layer = shp.layer,
  verbose = FALSE)
```

The SAT data can be loaded into R from **ref.ICAR** using `read.table()`.

```
library(utils)

data.path <- system.file("extdata", "states-sats48.txt", package = "ref.ICAR",
  mustWork = TRUE)

sats48 <- read.table(data.path, header = T)
us.shape48$verbal <- sats48$VERBAL
us.shape48$percent <- sats48$PERCENT
```

Now that the shapefile and data are loaded, the observed data can be plotted as a choropleth map (Figure 1). This map illustrates the spatial dependence to be analyzed by the model. The Midwestern states and Utah exhibit the highest average SAT scores, and overall, neighboring states have similar average scores.

```
library(RColorBrewer)
library(sp)

verbal.brk <- quantile(sats48$VERBAL, c(0, 0.2, 0.4, 0.6, 0.8,
  1)) + c(-0.1, 0, 0, 0, 0, 0.1)
verbal.color <- brewer.pal(5, "Greys")
splot(us.shape48, "verbal", at = verbal.brk, col.regions = verbal.color,
  par.settings = list(panel.background = list(col = "white")),
  main = "Observed Verbal SAT Scores")
```

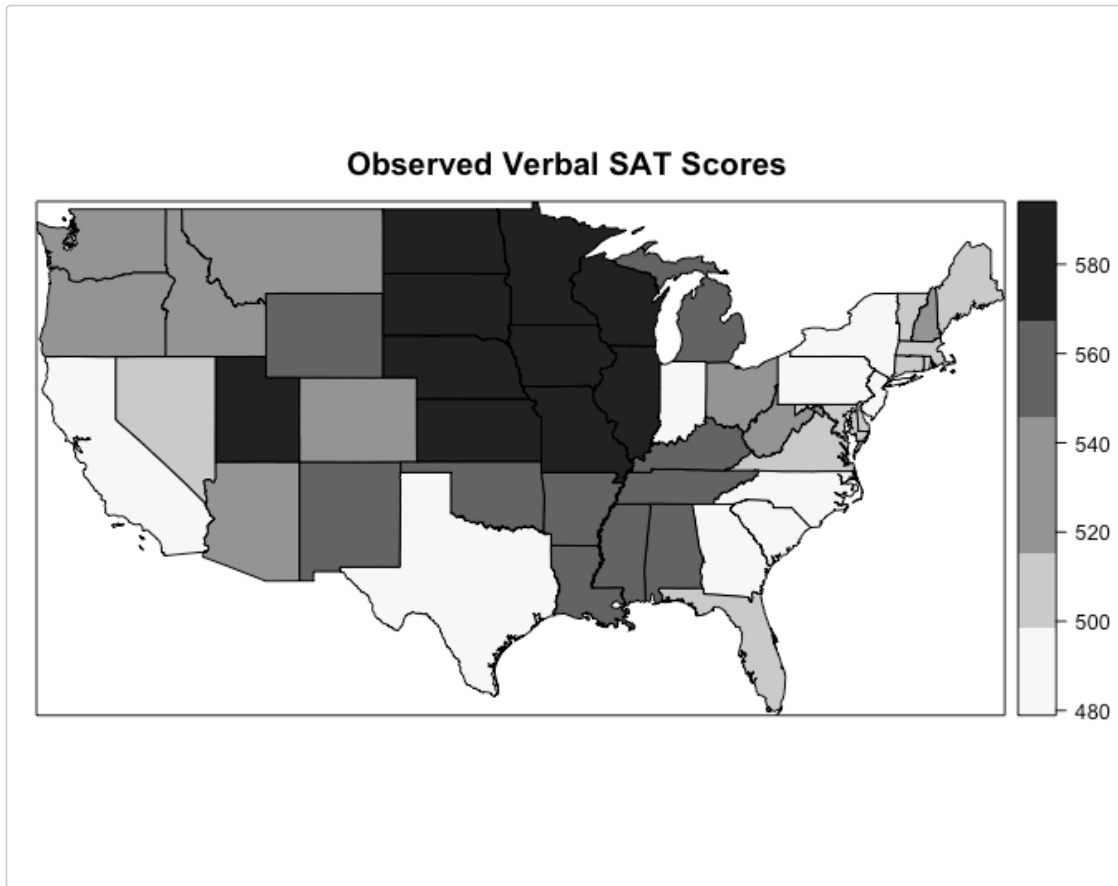


Figure 1: Observed Verbal SAT Scores

Similarly, the covariate, percent of eligible students taking the SAT, can be plotted over the contiguous United States. These data exhibit a seemingly inverse relationship to the SAT scores; lower percentages of students take the SAT in the Midwest.

```
percent.brk <- quantile(sats48$PERCENT, c(0, 0.2, 0.4, 0.6, 0.8,
1)) + c(-0.1, 0, 0, 0, 0, 0.1)
percent.color <- brewer.pal(5, "Greys")
spplot(us.shape48, "percent", at = percent.brk, col.regions = percent.color,
par.settings = list(panel.background = list(col = "white")),
main = "Percent Taking SAT")
```

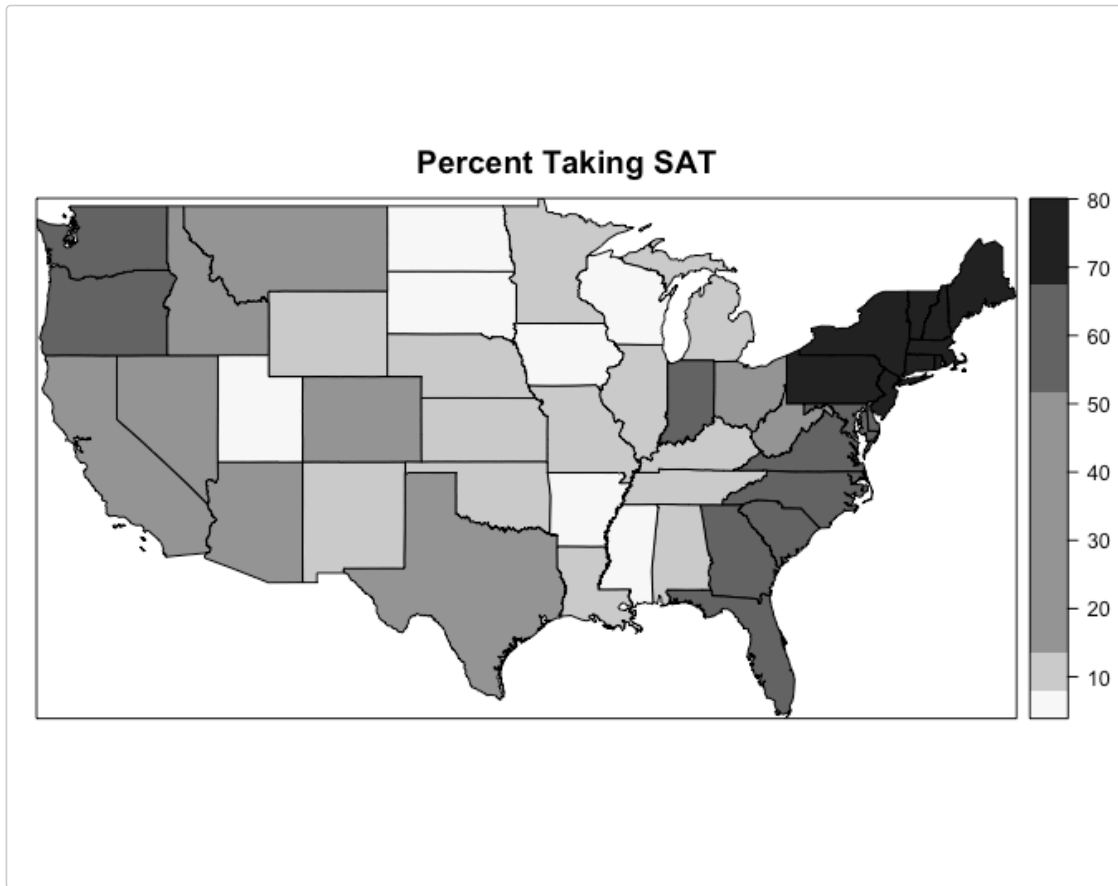


Figure 2: Percent of eligible students taking the SAT

Employing the functions in **ref.ICAR**, the `shape.H()` function first takes the path to the shape file (obtained above), and returns a list of two objects. This list contains the neighborhood matrix, H and a `SpatialPolygonsDataFrame` object corresponding to the shapefile, to be used by the remaining functions.

```
library(ref.ICAR)

shp.data <- shape.H(system.path)
H <- shp.data$H

class(shp.data$map)

## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"

length(shp.data$map)
```

```
## [1] 49
```

The response and covariates, Y and X must be defined before fitting the model. The response, Y , is Verbal SAT scores. X has two columns corresponding to an intercept and the predictor, percent of eligible students taking the SAT in 1999.

```
Y <- sats48$VERBAL
x <- sats48$PERCENT
X <- cbind(1, x)
```

Then sampling can be performed using `ref.MCMC()`. The default starting values are used below, with MCMC iterations and burn-in larger than the default.

```
set.seed(3456)

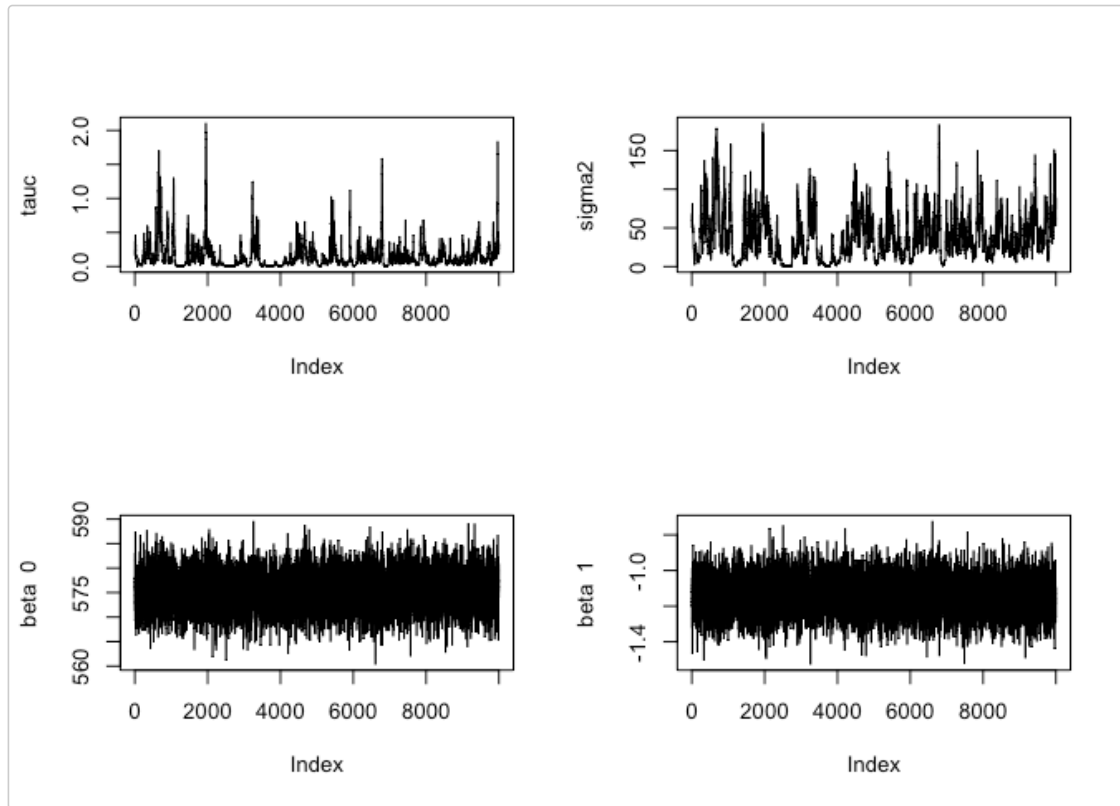
ref.SAT <- ref.MCMC(y = Y, X = X, H = H, iters = 15000, burnin = 5000,
  verbose = FALSE)

names(ref.SAT)

## [1] "MCMCchain"      "tauc.MCMC"      "sigma2.MCMC"    "beta.MCMC"
## [5] "phi.MCMC"       "accept.phi"     "accept.sigma2"  "accept.tauc"
```

The object `ref.SAT` contains MCMC chains for each of the parameters in the model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\theta} + \boldsymbol{\phi}$, using a signal-to-noise ratio parameterization. From these, the function `ref.plot()` creates trace plots for each parameter to visually confirm convergence.

```
par(mfrow = c(2, 2))
ref.plot(ref.SAT$MCMCchain, X, burnin = 5000, num.reg = length(Y))
```



The remaining components for the analysis are the functions for parameter and regional inferences. The function `ref.summary()` provides posterior medians and intervals for the model parameters β , τ_c , and σ^2 . The function `ref.summary()` provides medians and Highest Posterior Density intervals for the fitted y values for each subregion in the data.

```
summary.params <- ref.summary(MCMCchain = ref.SAT$MCMCchain,
  tauc.MCMC = ref.SAT$tauc.MCMC, sigma2.MCMC = ref.SAT$sigma2.MCMC,
  beta.MCMC = ref.SAT$beta.MCMC, phi.MCMC = ref.SAT$phi.MCMC,
  accept.phi = ref.SAT$accept.phi, accept.sigma2 = ref.SAT$accept.sigma2,
  accept.tauc = ref.SAT$accept.tauc, iters = 15000, burnin = 5000)
```

```
names(summary.params)
```

```
## [1] "beta.median" "beta.hpd" "tauc.median" "tauc.hpd"
## [5] "sigma2.median" "sigma2.hpd" "tauc.accept" "sigma2.accept"
```

```
summary.params
```

```
## $beta.median
## [1] 575.461382 -1.142777
```

```
##
## $beta.hpd
##           lower           upper
## var1 568.028129 582.5163300
## var2 -1.334464 -0.9487258
##
## $tauc.median
## [1] 0.08581395
##
## $tauc.hpd
##           lower           upper
## 0.003402439 0.5212723440
##
## $sigma2.median
## [1] 31.2014
##
## $sigma2.hpd
##           lower           upper
## 0.1604447 96.0084162
##
## $tauc.accept
## [1] 0.3436667
##
## $sigma2.accept
## [1] 0.3436667
```

The posterior medians for β_0 and β_1 are 575.496 and -1.145, respectively. Additionally, the HPD interval for β_1 does not include 0, which indicates that as the percent of eligible students taking the SAT increases, average Verbal SAT score tends to decrease. The τ_c median is 0.08, with HPD interval between 0.0014 and 0.5237.

```
summary.region <- reg.summary(ref.SAT$MCMCchain, X, Y, burnin = 5000)

us.shape48$verbalfits <- summary.region$reg.medians
spplot(us.shape48, "verbalfits", at = verbal.brk, col.regions = verbal.color,
       main = "Posterior Medians for Verbal SAT", par.settings = list(panel.background = list(col =
       "white")))
```

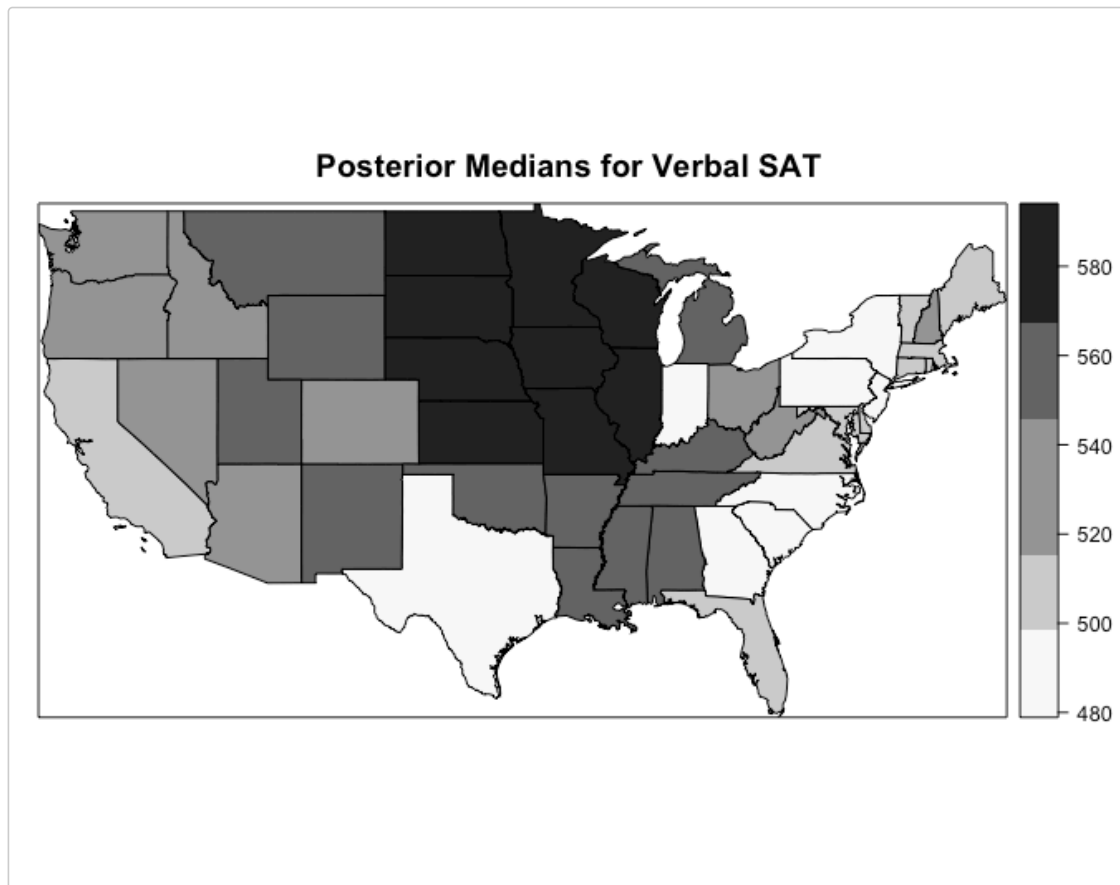


Figure 3: Posterior Medians for Verbal SAT

Finally, the function `ref.analysis()` in **ref.ICAR** performs the entire reference analysis, including:

- Obtaining the neighborhood matrix, H
- Running the MCMC chains
- Producing parameter trace plots
- Producing parameter summaries
- Producing regional summaries

`ref.analysis()` requires the following user inputs: X , y , a path to a shapefile, a vector of region names corresponding to the values in X , and a vector of region names corresponding to the values in response y . The region names in each of X and y must match and are required because `ref.analysis()` reorders the data according to the region order in the shapefile. This ensures that the data values match to the correct entries in the neighborhood matrix H ; otherwise analysis might map predicted values to incorrect regions. If the provided shapefile does not have a specified NAME column, the user will be asked to also provide a vector of names corresponding to the shapefile. This vector is called `shp.reg.names` in the documentation and function arguments; the default value is NULL.

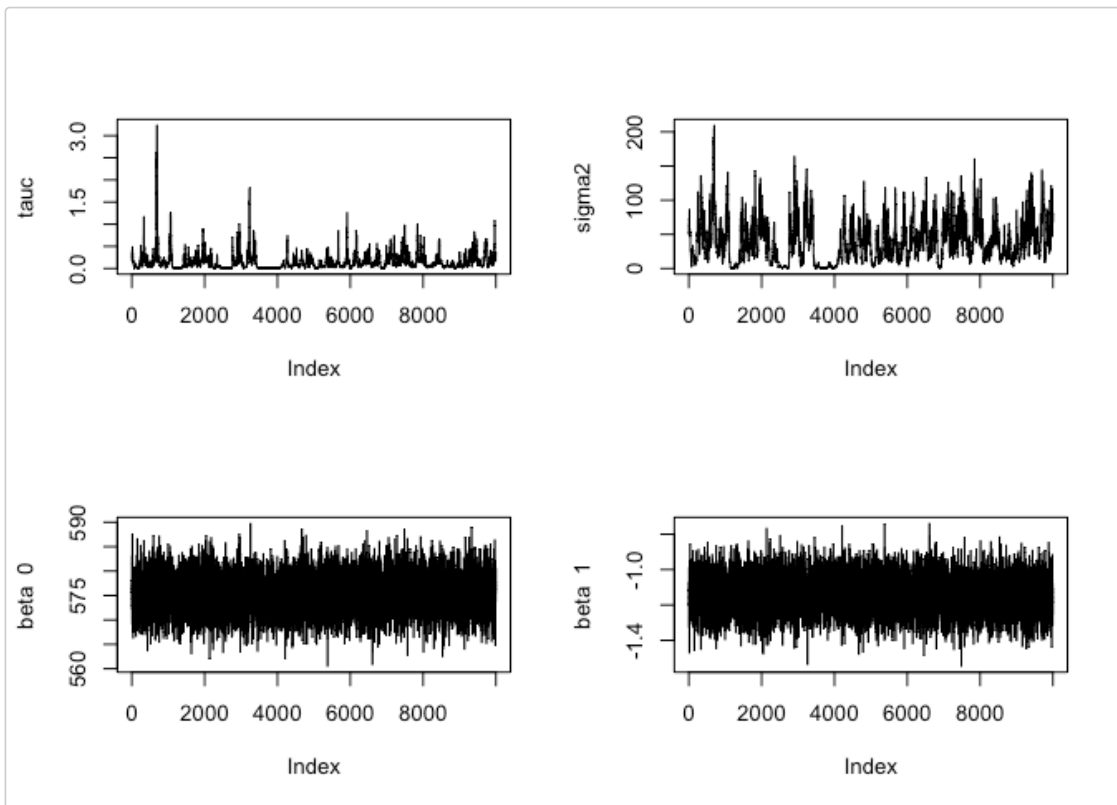
```
### The SAT scores and percent of students are already arranged
### by state alphabetically
```

```

x.reg.names <- us.shape48$NAME
y.reg.names <- us.shape48$NAME

set.seed(3456)
par(mfrow = c(2, 2))
sat.analysis <- ref.analysis(system.path, X, Y, x.reg.names,
  y.reg.names, shp.reg.names = NULL, iters = 15000, burnin = 5000,
  verbose = FALSE, tauc.start = 0.1, beta.start = -1, sigma2.start = 0.1,
  step.tauc = 0.5, step.sigma2 = 0.5)

```



```
names(sat.analysis)
```

```

## [1] "H"          "MCMC"        "beta.median" "beta.hpd"
## [5] "tauc.median" "tauc.hpd"    "sigma2.median" "sigma2.hpd"
## [9] "tauc.accept" "sigma2.accept" "fit.dist"     "reg.medians"
## [13] "reg.hpd"

```

References

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014), *Hierarchical modeling and analysis for spatial data, second edition*, Chapman; Hall/CRC. <https://doi.org/10.1201/b17115>.

Keefe, M. J., Ferreira, M. A. R., and Franck, C. T. (2018a), “Objective Bayesian Analysis for Gaussian Hierarchical Models with Intrinsic Conditional Autoregressive Priors,” *Bayesian Analysis*, International Society for Bayesian Analysis. <https://doi.org/10.1214/18-BA1107>.

Keefe, M. J., Ferreira, M. A. R., and Franck, C. T. (2018b), “On the formal specification of sum-zero constrained intrinsic conditional autoregressive models,” *Spatial Statistics*, Elsevier {BV}, 24, 54–65. <https://doi.org/10.1016/j.spasta.2018.03.007>.

Chapter 4

Conclusion

This thesis details **ref.ICAR**, an R package that implements objective Bayesian analysis for spatially correlated areal data using the ICAR reference prior by Keefe et al. (2019). The ICAR reference prior provides a default prior specification for a vector of spatial random effects in a Bayesian hierarchical model for areal data. **ref.ICAR**, available on CRAN Porter et al. (2018), makes available functions for sampling, estimation, intervals, plots, and fitted values using the ICAR reference prior, without the need to specify subjective priors or hyperparameters. **ref.ICAR** contains functions for both the entire analysis and for individual elements of the analysis, so that researchers can select the amount of output generated. Chapter 2 contained documentation for all functions in **ref.ICAR**, with relevant examples as well as function argument and parameter descriptions. Chapter 3 presented the package vignette with a long-form example Carlin et al. (2014) and further explanations for the use of each function in the package. **ref.ICAR** provides computational accessibility for the analysis of spatially correlated areal data through open source software and readily-available documentation detailing relevant theory and application.

Bibliography

- Besag, J. (1974, jan). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)* 36(2), 192–225.
- Besag, J., J. York, and A. Mollié (1991, mar). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics* 43(1), 1–20.
- Carlin, B. P., A. E. Gelfand, and S. Banerjee (2014). *Hierarchical modeling and analysis for spatial data*. Chapman and Hall/CRC.
- Goodchild, M. and D. Janelle (2004). *Spatially Integrated Social Science*. Spatial Information Systems. Oxford University Press.
- Hoef, J. M. V., E. E. Peterson, M. B. Hooten, E. M. Hanks, and M.-J. Fortin (2018, jan). Spatial autoregressive models for statistical inference from ecological data. *Ecological Monographs* 88(1), 36–59.
- Hogan, J. W. and R. Tchernis (2004, jun). Bayesian factor analysis for spatially correlated data, with application to summarizing area-level material deprivation from census data. *Journal of the American Statistical Association* 99(466), 314–324.
- Keefe, M. J., M. A. R. Ferreira, and C. T. Franck (2019, 03). Objective bayesian analysis

- for gaussian hierarchical models with intrinsic conditional autoregressive priors. *Bayesian Anal.* 14(1), 181–209.
- Lee, D. (2011). A comparison of conditional autoregressive models used in bayesian disease mapping. *Spatial and Spatio-temporal Epidemiology* 2(2), 79 – 89.
- Porter, E. M., M. J. Keefe, C. T. Franck, and M. A. R. Ferreira (2018). *ref.ICAR: Objective Bayes Intrinsic Conditional Autoregressive Model for Areal Data*. R package version 1.0.
- Zhu, J., H.-C. Huang, and P. E. Reyes (2010, jun). On selection of spatial linear models for lattice data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72(3), 389–402.

Appendix A

ref.ICAR Code

```
## Erica M. Porter
## March 2019
## R code for package ref.ICAR functions

##### Function to create a neighborhood matrix from a user-provided shapefile #####
shape.H <- function(shape.file) {

  shp.layer <- gsub('.shp','',basename(shape.file))
  #shp.path <- gsub(basename(shape.file),'',shape.file)
  shp.path <- dirname(shape.file)
  map <- readOGR(dsn = path.expand(shp.path), layer = shp.layer, verbose = FALSE)

  map.nb <- poly2nb(map)
  W <- nb2mat(map.nb, style = "B")
  Dmat <- diag(apply(W,1,sum))
  lambda <- 0

  H <- Dmat - W
  row.names(H) <- NULL
  return(list(H=H,map=map))
}

##### Function to perform eigenvalue decomposition and construct covariance matrices for MCMC #####
ref.input <- function(H,X,y,lambda=0) {
  if (is.matrix(X)==FALSE & is.vector(X)==TRUE) {X <- as.matrix(X, ncol=1)}

  check.mat(H)

  num.reg <- length(y)
  row.names(H) <- NULL
  Q <- eigen(H,symmetric=TRUE)$vectors
  Qmat <- Q[,1:(num.reg-1)]
  eigH <- eigen(H,symmetric=TRUE)$values
  phimat <- diag(1/sqrt(lambda + eigH[1:(num.reg-1)]))
  D <- diag(eigH)
  Sig_phi <- matrix(0,num.reg,num.reg) #initialize
  for(i in 1:(num.reg-1)){
    total <- (1/(lambda + eigH[i]))*Q[,i]%*%t(Q[,i])
    Sig_phi <- Sig_phi + total
  }
  s1 <- diag(num.reg) - X%*%solve(t(X)%*%X)%*%t(X)
  Q.star <- eigen(s1)$vectors[,1:(num.reg-ncol(X))]
  M <- t(Q.star)%*%Sig_phi%*%Q.star
  U <- eigen(M)$vectors
  L <- Q.star%*%U
  xi <- eigen(t(Q.star)%*%Sig_phi%*%Q.star)$values

  return(list(D=D,Q=Q,xi=xi,Qmat=Qmat,phimat=phimat,Sig_phi=Sig_phi,X=X))
}

##### Function to perform sampling from the model with ICAR prior #####
ref.MCMC <- function(y,X,H,itters=10000,burnin=5000,verbose=TRUE,
  tauc.start=1,beta.start=1,sigma2.start=1,step.tauc=0.5,
  step.sigma2=0.5){

  if (is.matrix(X)==FALSE & is.vector(X)==TRUE) {X <- as.matrix(X, ncol=1)}

  num.reg<-length(y)

  #Eigenvalue decomposition for sampling
  if (is.matrix(X)==FALSE & is.vector(X)==TRUE) {X <- as.matrix(X, ncol=1)}

  check.mat(H)
  row.names(H) <- NULL
  Q <- eigen(H,symmetric=TRUE)$vectors
  Qmat <- Q[,1:(num.reg-1)]
  eigH <- eigen(H,symmetric=TRUE)$values
  phimat <- diag(1/sqrt(eigH[1:(num.reg-1)]))
  D <- diag(eigH)
  Sig_phi <- matrix(0,num.reg,num.reg) #initialize
```

```

for(i in 1:(num.reg-1)){
  total <- (1/(eigH[i]))*Q[,i]**t(Q[,i])
  Sig_phi <- Sig_phi + total
}
s1 <- diag(num.reg) - X**solve(t(X)**X)**t(X)
Q.star <- eigen(s1)$vectors[,1:(num.reg-ncol(X))]
M <- t(Q.star)**Sig_phi**Q.star
U <- eigen(M)$vectors
L <- Q.star**U
xi <- eigen(t(Q.star)**Sig_phi**Q.star)$values

#Initialize Metropolis Algorithm
tauc.MCMC<-matrix(0,nrow=iters,ncol=1)
sigma2.MCMC<-matrix(0,nrow=iters,ncol=1)
beta.MCMC<-matrix(0,nrow=iters,ncol=ncol(X))
phi.MCMC<-matrix(0,nrow=iters,ncol=num.reg)
tauc.MCMC[1,]<-tauc.start
sigma2.MCMC[1,]<-sigma2.start
phi.MCMC[1,]<-scale((1:num.reg)/100,center=TRUE,scale=FALSE)
beta.MCMC[1,]<-beta.start

#initialize acceptance rates
accept.phi<-1
accept.tauc<-1
accept.sigma2<-1
accept.beta<-1

for(i in 2:iters){

  curr.sigma2<-sigma2.MCMC[i-1,]
  curr.tauc<-tauc.MCMC[i-1,]
  curr.phi<-phi.MCMC[i-1,]
  curr.beta<-beta.MCMC[i-1,]

  #Metropolis step for sigma2 and tauc jointly
  logprop.sigma2<-rnorm(1,mean=log(curr.sigma2),sd=step.sigma2) #Propose log(sigma2) using normal
distribution
  prop.sigma2<-exp(logprop.sigma2)
  logprop.tauc<-rnorm(1,mean=log(curr.tauc),sd=step.tauc) #Propose log(tauc) using normal
distribution
  prop.tauc<-exp(logprop.tauc)

  N0<-(-1/2)*(num.reg*log(prop.sigma2) + sum(log((1+(1/(prop.tauc*(diag(D)[1:(num.reg-1)])))))) -
  ((1/(2*prop.sigma2))*t(y-X**curr.beta)**(Q**diag(c((1+(1/(prop.tauc*(diag(D)[1:(num.reg-
1)]))))))^(-1,1))**t(Q))**t(y-X**curr.beta)) -
  log(prop.sigma2) + ((0.5)*log((sum((xi/(prop.tauc+xi))^2)) - ((1/(num.reg - ncol(X)))*
(sum((xi/(prop.tauc+xi))^2)))) + log(prop.tauc) +log(prop.sigma2) - log(prop.tauc)
  D0<-(-1/2)*(num.reg*log(curr.sigma2) + sum(log((1+(1/(curr.tauc*(diag(D)[1:(num.reg-1)])))))) -
  ((1/(2*curr.sigma2))*t(y-X**curr.beta)**(Q**diag(c((1+(1/(curr.tauc*(diag(D)[1:(num.reg-
1)]))))))^(-1,1))**t(Q))**t(y-X**curr.beta)) -
  log(curr.sigma2) + ((0.5)*log((sum((xi/(curr.tauc+xi))^2)) - ((1/(num.reg - ncol(X)))*
(sum((xi/(curr.tauc+xi))^2)))) + log(curr.tauc) + log(curr.sigma2) - log(curr.tauc)

  if(log(runif(1,0,1))<(N0-D0)){
    sigma2.MCMC[i,]<-prop.sigma2
    accept.sigma2<-accept.sigma2 + 1
    tauc.MCMC[i,]<-prop.tauc
    accept.tauc<-accept.tauc + 1
  }
  else{
    sigma2.MCMC[i,]<-curr.sigma2
    tauc.MCMC[i,]<-curr.tauc
  }
  curr.sigma2<-sigma2.MCMC[i,]
  curr.tauc<-tauc.MCMC[i,]

  #Gibbs step for betas
  Ainv<-(Q**diag(c((1+(1/(curr.tauc*(diag(D)[1:(num.reg-1)]))))))^(-1,1))**t(Q))
  mu<-solve(t(X)**Ainv**X)**t(X)**Ainv**y
  V<-curr.sigma2*solve(t(X)**Ainv**X)
  beta.MCMC[i,]<-t(rmnorm(1,mean=mu,sigma=V))
  curr.beta<-beta.MCMC[i,]

```

```

    accept.beta<-accept.beta + 1

    #Sample Phis from full conditional
    mu1<-X%*%curr.beta
    mu2<-rep(0,num.reg)
    Sig_11<-curr.sigma2*(diag(num.reg) + ((1/curr.tauc)*Sig_phi))
    Sig_12<-(curr.sigma2/curr.tauc)*Sig_phi
    Sig_21<-Sig_12
    Sig_22<-Sig_12
    phi.fc.mean<-mu2 + Sig_21%*%M.P.Inverse(Sig_11)%*(y - mu1)
    phi.fc.var <- Sig_22 - Sig_21%*%M.P.Inverse(Sig_11)%*Sig_12
    #make variance matrix symmetric to account for machine precision
    phi.fc.var.sym <- (phi.fc.var + t(phi.fc.var))/2

    prop.phi<-t(rmvnorm(1,mean=phi.fc.mean,sigma=phi.fc.var.sym,method="svd"))

    phi.MCMC[i,]<-prop.phi
    accept.phi<-accept.phi+1

    if(verbose==TRUE) {
      if((100*(i/iters))%5==0){print(paste(100*(i/iters),'% complete',' at ',date(),sep=''))}
    }
  }

MCMCchain <- cbind(tauc.MCMC,sigma2.MCMC,beta.MCMC,phi.MCMC)

return(list(MCMCchain=MCMCchain,tauc.MCMC=tauc.MCMC,sigma2.MCMC=sigma2.MCMC,beta.MCMC=beta.MCMC,
phi.MCMC=phi.MCMC,accept.phi=accept.phi,accept.sigma2=accept.sigma2,accept.tauc=accept.tauc))
}

##### Function to provide posterior estimates and intervals for model parameters #####
ref.summary <-
function(MCMCchain,tauc.MCMC,sigma2.MCMC,beta.MCMC,phi.MCMC,accept.phi,accept.sigma2,accept.tauc,iters=
{
  beta.median <- apply(cbind(beta.MCMC[(burnin+1):iters,]),2,median)
  beta.obj <- mcmc(beta.MCMC,start=(burnin+1),end=iters)
  beta.hpd <- HPDinterval(beta.obj,prob=0.95)[,]
  tauc.median <- median(tauc.MCMC[(burnin+1):iters])
  tauc.obj <- mcmc(tauc.MCMC,start=(burnin+1),end=iters)
  tauc.hpd <- HPDinterval(tauc.obj,prob=0.95)[,]
  tauc.accept <- accept.tauc/iters
  sigma2.median <- median(sigma2.MCMC[(burnin+1):iters])
  sigma2.obj <- mcmc(sigma2.MCMC,start=(burnin+1),end=iters)
  sigma2.hpd <- HPDinterval(sigma2.obj,prob=0.95)[,]
  sigma2.accept <- accept.sigma2/iters

  return(list(beta.median=beta.median,beta.hpd=beta.hpd,tauc.median=tauc.median,tauc.hpd=tauc.hpd,
    sigma2.median=sigma2.median,sigma2.hpd=sigma2.hpd,
    tauc.accept=tauc.accept,sigma2.accept=sigma2.accept))
}

##### Function to create trace plots for the model parameters #####
ref.plot <- function(MCMCchain, X, burnin, num.reg) {
  plot(MCMCchain[(burnin+1):nrow(MCMCchain),1], type='l', ylab="tauc")
  plot(MCMCchain[(burnin+1):nrow(MCMCchain),2], type='l', ylab="sigma2")
  #check for an intercept to correctly label plots
  if (mean(as.matrix(X[,1]))==1) {
    for (i in 3:(dim(MCMCchain)[2]-num.reg)) {
      plot(MCMCchain[(burnin+1):nrow(MCMCchain),i], type='l', ylab=paste("beta ",i-3))
    }
  } else {for (i in 3:(dim(MCMCchain)[2]-num.reg)) {
    plot(MCMCchain[(burnin+1):nrow(MCMCchain),i], type='l', ylab=paste("beta ",i-2))
  }
}

##### Function to produce inferences and intervals for each region in the data #####
reg.summary <- function(MCMCchain,X,Y,burnin) {
  num.reg <- length(Y)

  # specify fixed effect regression coefficients
  betavec <- MCMCchain[((burnin+1):nrow(MCMCchain)),3:(3+ncol(X)-1)]

```



```

# specify model values for spatial effects
phi.composite <- MCMCchain[((burnin+1):nrow(MCMCchain)), (3+ncol(X)):ncol(MCMCchain)]

# fitted posteriors by region
fit.dist <- matrix(0,nrow=nrow(MCMCchain)-burnin, ncol=num.reg)
for(k in 1:(nrow(MCMCchain)-burnin)) {
  fit.dist[k,] <- X[k,]*betavec[k,] + as.matrix(phi.composite[k,])
}

# regional medians and intervals
reg.medians <- apply(fit.dist,2,median)

reg.cred <- data.frame(t(apply(fit.dist, 2, FUN = quantile, prob = c(0.025, 0.975))))

reg.obj <- mcmc(fit.dist,start=(burnin+1),end=nrow(fit.dist))
reg.hpd <- HPDinterval(reg.obj,prob=0.95)[,]

return(list(fit.dist=fit.dist,reg.medians=reg.medians,reg.hpd=reg.hpd))
}

##### Function to perform entire analysis (sampling, inferences, and plots) #####
ref.analysis <- function(shape.file,X,y,x.reg.names,y.reg.names,shp.reg.names=NULL,itters=10000,
burnin=5000,verbose=TRUE,tauc.start=1,beta.start=1,sigma2.start=1,step.tauc=0.5,step.sigma2=0.5) {

  ## Check for missing data
  if(mean(is.na(y))!=0) {stop("The current version of the package does not support missing data.")}
  num.reg <- nrow(as.matrix(y))

  ## Get neighborhood matrix and SpatialPolygonsDataFrame (region names/order)
  dat <- shape.H(shape.file)
  H <- dat$H
  map <- dat$map

  ## Attach name columns to data
  x.data <- data.frame(as.vector(x.reg.names),X)
  y.data <- data.frame(as.vector(y.reg.names),y)
  colnames(x.data)[1] <- c("NAME")
  colnames(y.data)[1] <- c("NAME")

  ## Extract the column of region names from the shapefile
  ## If no NAME column in the shapefile, the user will need to input a vector for shp.reg.names
  if(is.null(shp.reg.names)==FALSE) {
    shape.names <- shp.reg.names
  } else if ("NAME" %in% colnames(map@data)){
    shape.names <- map$NAME} else {
    warning("Please enter a vector of names corresponding to your shapefile")}

  ## Re-order the data to be in the same region order as the shapefile
  x.data <- x.data[order(factor(x.data$NAME,levels=shape.names)),]
  y.data <- y.data[order(factor(y.data$NAME,levels=shape.names)),]

  ## Remove region name columns for sampling
  Y <- as.vector(y.data[,-1])
  X <- as.matrix(x.data[,-1])

  ## Perform sampling using ICAR reference prior (Keefe et al., 2018)
  MCMC <- ref.MCMC(y=Y,X=X,H=H,itters,burnin,verbose,
    tauc.start,beta.start,sigma2.start,step.tauc,
    step.sigma2)

  ## Parameter summaries
  ref.params <- ref.summary(MCMCchain=MCMC$MCMCchain,tauc.MCMC=MCMC$tauc.MCMC,
sigma2.MCMC=MCMC$sigma2.MCMC,beta.MCMC=MCMC$beta.MCMC,phi.MCMC=MCMC$phi.MCMC,
    accept.phi=MCMC$accept.phi,accept.sigma2=MCMC$accept.sigma2,
    accept.tauc=MCMC$accept.tauc,itters,burnin)

  ## Extract parameter summaries to avoid list returns for user
  beta.median <- ref.params$beta.median
  beta.hpd <- ref.params$beta.hpd

```

```
tauc.median <- ref.params$tauc.median
tauc.hpd <- ref.params$tauc.hpd

sigma2.median <- ref.params$sigma2.median
sigma2.hpd <- ref.params$sigma2.hpd

sigma2.accept <- ref.params$sigma2.accept
tauc.accept <- ref.params$tauc.accept

## Parameter trace plots
ref.plot(MCMC$MCMCchain,X,burnin,num.reg)

## Regional medians and intervals
regions <- reg.summary(MCMC$MCMCchain,X,Y,burnin)

fit.dist <- regions$fit.dist
reg.medians <- regions$reg.medians
reg.hpd <- regions$reg.hpd

return(list(H=H,MCMC=MCMC$MCMCchain,beta.median=beta.median,
           beta.hpd=beta.hpd,tauc.median=tauc.median,tauc.hpd=tauc.hpd,
           sigma2.median=sigma2.median,sigma2.hpd=sigma2.hpd,
           tauc.accept=tauc.accept,sigma2.accept=sigma2.accept,
           fit.dist=fit.dist,reg.medians=reg.medians,reg.hpd=reg.hpd))
}
```