CARBayes

- This package allows for the simple command-based fitting of spatial models via MCMC
- It is set up to be like `lm` function in R so that a formula can be specified and then fitted. Hence the are two steps in the modeling process.
- Uses MALA with Metropolis-Hastings (like STAN)
  - This allows for efficient sampling of a wide range of models
- Single chains only
- Posterior median values reported rather than mean values
- Leroux model extensively used

Package Description (V5.2)

Implements a class of univariate and multivariate spatial generalised linear mixed models for areal unit data, with inference in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation.
The response variable can be binomial, Gaussian, multinomial, Poisson or zero-inflated Poisson (ZIP), and spatial autocorrelation is modelled by a set of random effects that are assigned a conditional autoregressive (CAR) prior distribution.
A number of different models are available for univariate spatial data, including models with no random effects as well as random effects modelled by different types of CAR prior, including the BYM model (Besag et al. (1991) <doi:10.1007/BF00116466>),
the Leroux model (Leroux et al. (2000) <doi:10.1007/978-1-4612-1284-3_4>) and the localised model (Lee et al. (2015) <doi:10.1002/env.2348>).
Additionally, a multivariate CAR (MCAR) model for multivariate spatial data is available, as is a two-level hierarchical model for modelling data relating to individuals within areas.

CARBayes models

- A range of models are available on CARBayes
- The spatial structure is input via a binary weight matrix: W
  - This can be derived from polygon objects

```
W.nb <- poly2nb(SCmap)
W.mat <- nb2mat(W.nb, style="B")
```

- Convergence of the single chain is checked via Geweke diagnostic
- The summary produces a DIC, WAIC, LMPL

Model formulation

- Observed count data with a variety of likelihoods, and the structure of the model is of the form:

$$y_i \sim f(\mu_i, \kappa) \quad i = 1, \ldots, m$$

$$g(\mu_i) = x_i^T \beta + O_i + u_i$$

O_i is an offset

u_i is a random effect (UH or CH or Leroux)

$$\beta \sim N(\mu_\beta, \Sigma_\beta)$$

κ is only used for Gaussian models

Poisson count model example

- Log link is assumed
- offset with log of expected count

$$log(\mu_i) = x_i^T \beta + O_i + u_i$$

$$O_i = log(Exp_i)$$

where Exp is the expected count

UH model

```
form<-Y1998~1+offset(log(Exp98))
modelUH<-
S.CARleroux(form,family="poisson",data=S
Cresp, W=W.mat,rho=0, burnin=10000,
n.sample=11000)
modelUH$summary
print(modelUH)
modelUH$samples
modelUH$modelfit
```

Some output commands

- The model object (`model`) can be interrogated

```
ls(model)
 "accept""fitted.values""formula"
"localised.structure" "model"
"modelfit" "residuals"
"samples" "summary.results" "X"
```

`accept` is the acceptance probabilities for the sampled parameters

Convergence

- `model$summary.results`:
  - Gives the Geweke diagnostic which is on the range of (-1.96 , 1.96). Converged parameters should fall between these limits.
- This is also reported in `print(model)`

Other output

- The sample output can be obtained by
  `model$samples`
- Goodness of fit measures:

`model$modelfit`

  - DIC, pD, WAIC, pW, LMPL, loglikelihood provided
- Residuals and fitted values:

`model$residuals`

  - Raw and pearson residuals for univariate models

`model$fitted.values`

  - gives fitted values as vector for univariate models

Model variants

- No spatial effect:

```
S.glm()
```

- Bym model

```
S.CARbym(formula,family,data,W,burnin,n.sample)
```

- Leroux model:

```
S.CARleroux(......rho=NULL..)
```

With the spatial dependence parameter `rho=0` then a UH model is fitted, otherwise rho is estimated.

Model Variants

- `S.CARlocalised`
  - Allows neighboring areas to have different values visa piecewise constant intercepts
- `S.CARdissimilarity`
  - Allows locally defined CAR components to detect boundaries.

Respiratory cancers example

CARBayes_ALL_models_V2_Scresp_data.txt

```
form<-Y1998~1+offset(log(Exp98))
modelUH<-
S.CARleroux(form,family="poisson",data=S
Cresp, W=W.mat,rho=0, burnin=10000,
n.sample=11000)
modelUH$summary
print(modelUH)
modelUH$samples
modelUH$modelfit
```

Output

modelUH$modelfit					
	DIC	p.d	WAIC	p.w	LMPL
	314.2667	15.45435	315.1744	12.58543	-161.238
loglikelihood					
	-141.679				

modelUH$summary								
Median	2.50%	97.50%	n.sample	%	accept	n.effective	Geweke.diag	
(Intercept) 0.0037	-0.0352	0.0416	1000		39.8	225.5	-1.1	
tau2 0.0095	0.0036	0.021	1000		100	63.8	-2.1	
rho 0.0000	0	0	NA		NA	NA	NA	

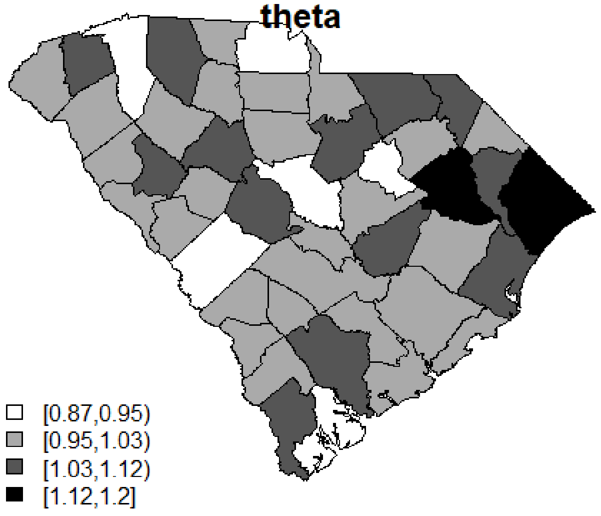
Interpretation

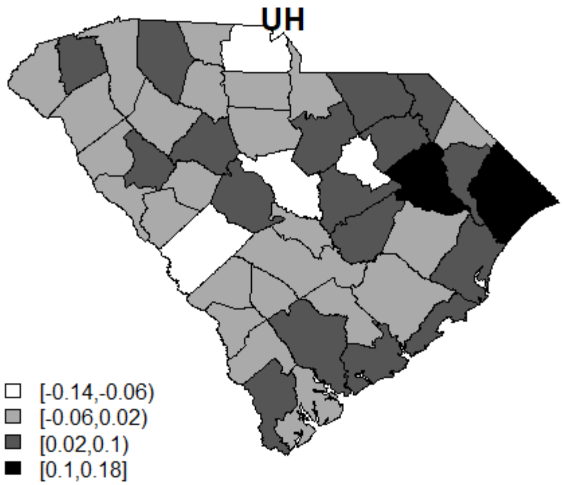
- Intercept and Tau2 are reasonably converged although a slightly longer run or thin may be preferred.
- DIC very close to WAIC in this case. Both relative measures and so they can only really be examined when compared to a different model flitted to the same data.
- LMPL is also relative and the larger the value the better
- The code also included a computation of the MSPE (136.87)

Plotting theta and UH effect

```
sample.theta<-matrix(0,nrow=L,ncol=m)
for (i in 1:L){
for(j in 1:m){
sample.theta[i,j]<-
sample.fitted[i,j]/SCresp$Exp98[j]
}}
theta.est<-colMeans(sample.theta[,])
library(fillmap)
fillmap(SCmap,"theta",theta.est,n.col=4,leg.loc
="bottomleft",leg.cex=1.0)
UHeffect<-log(theta.est)-0.0005;x11()
fillmap(SCmap,"UH",UHeffect,n.col=4,leg.loc="bo
ttomleft",leg.cex=1.0)
```

Plots


theta
[0.87,0.95)
[0.95,1.03)
[1.03,1.12)
[1.12,1.2]


UH
[-0.14,-0.06)
[-0.06,0.02)
[0.02,0.1)
[0.1,0.18]

ICAR model

```
form<-Y1998~1+offset(log(Exp98))
modelCH<-
S.CARleroux(form,family="poisson",data=S
Cresp, W=W.mat,rho=1, burnin=10000,
n.sample=11000)
modelCH$summary
modelCH$modelfit
```

- DIC = 315.01 pD = 11.12, WAIC = 318.72, pW = 11.798

Convolution model

```
form<-Y1998~1+offset(log(Exp98))
modelCONV<-
S.CARbym(form,family="poisson",data=SCresp, W=W.mat, burnin=10000,
n.sample=11000)
modelCONV$summary
modelCONV$modelfit
```

- Results:
  - DIC = 314.06, WAIC = 313.32

Leroux model

```
form<-Y1998~1+offset(log(Exp98))
modelLER<-
S.CARleroux(form,family="poisson",data=S
Cresp, W=W.mat, burnin=50000,
n.sample=55000)
modelLER$summary
modelLER$modelfit
```

- DIC = 316.15, WAIC = 318.95

Summary

- CARBayes provides a range of easy-to-use command- based model specifications
- A basic range models (UH, ICAR, Convolution, Leroux) and specialist pre-programmed models
- Easy to add continuous covariates or factors via the formula
- Additional facilities to fit other binomial and Gaussian models as well as some multivariate models
- Fast sampling via MALA
- CARBayesST allows the fitting of space-time models

- But:
  - Multiple chains and mean calculations not provided
  - Disadvantage: focus on Leroux model, GP model not available
  - Not as flexible model specification as available in BUGS or nimble