Chapter 4 - Spatial processes R packages and software
Lecture notes

Odd Kolbjørnsen and Geir Storvik

February 13, 2017

Last time

- General set up for "Kriging type problems"
  - Introductory example
  - Kriging case
  - General case
- Change of support
- Spatial moving average models
  - Construction
  - Correlation function
- Non gaussian observations
  - Monte Carlo
  - Laplace approximation
  - INLA

Today

- To day computations
  - Software
  - Computer
    - R
    - INLA
- Next timel lattice models

Software

- There is allways an existing software that "does your job"
- The challenge is to figure out how this works
- Even if the software does not do excatly what you want maybe it is good enough
- Reasons **not** to make your own computer code
  - existing code is tested (less bugs)
  - existing code is optimized (speed)
  - often related to publications
  - easier to get others to "accept it"
- Reasons to make your own computer code
  - understand the methodology better
  - improve/develop exsiting methodology
  - combining with other techniques
  - compete with existing computer code
  - because you have too much spare time

Software options

- Wiki List of spatial analysis software
  - `.../wiki/List_of_spatial_analysis_software/`
  - LuciadLightspeed,Open GeoDa, CrimeStatc, SaTScan, SAGA, IDRISI, Biodiverse, ERDAS IMAGINE, TerraLens ++
- ArcGIS: Geoprocessing, visualization (GIS = Geographic information system)
- GeoBUGS: Bayesian analysis (WinBugs)
- GeoDa: Explantory Spatial Anaysis +
- STARS: Space-time
- SAS/STAT: spatial analysis (limited)
- SPLUS: SpatialStats
- Matlab: Spatial-statistics toolbox (fast but limited)
- GEOEAS, SGeMS, GSLIB ,COHIBA, CRAVA ++
- **R**

Spatial analysis in R

- CRAN (The Comprehensive R Archive Network)

  URL: https://CRAN.R-project.org/view=Spatial

  - Classes for spatial data: sp, spacetime
  - Handling spatial data: geosphere
  - Reading and writing spatial data: rgdal
  - Visualisation
  - Point pattern analysis
  - Geostatistics: geoR, gstat, spatial
  - Disease mapping and areal data analysis: INLA
  - Spatial regression: nlme
  - Ecological analysis
- install.packages(...)
- update.packages(...)
- library(...)

Hierarcical and hierarcical Bayesian approach

Hierarchical model

	Variable	Densities	Notation in book
Data model:	$\mathbf{Z}$	$p(\mathbf{Z}\|\mathbf{Y},\theta)$	$[\mathbf{Z}\|\mathbf{Y},\theta]$
Process model:	$\mathbf{Y}$	$p(\mathbf{Y}\|\theta)$	$[\mathbf{Y}\|\theta]$
Parameter:	$\boldsymbol{\theta}$		

Simultaneous model: $p(\mathbf{y},\mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{z}|\mathbf{y},\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta})$
Marginal model: $L(\boldsymbol{\theta}) = p(\mathbf{z}|\boldsymbol{\theta}) = \int_{\mathbf{y}} p(\mathbf{z},\mathbf{y}|\boldsymbol{\theta})d\mathbf{y}$
Inference: $\widehat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$

Bayesian approach: Include model on $\boldsymbol{\theta}$

	Variable	Densities	Notation in book
Data model:	$\mathbf{Z}$	$p(\mathbf{Z}\|\mathbf{Y},\theta)$	$[\mathbf{Z}\|\mathbf{Y},\theta]$
Process model:	$\mathbf{Y}$	$p(\mathbf{Y}\|\theta)$	$[\mathbf{Y}\|\theta]$
Parameter model:	$\boldsymbol{\theta}$	$p(\boldsymbol{\theta})$	$[\boldsymbol{\theta}]$

Simultaneous model: $p(\mathbf{y},\mathbf{z},\boldsymbol{\theta})$
Marginal model: $p(\mathbf{z}) = \int_{\boldsymbol{\theta}}\int_{\mathbf{y}} p(\mathbf{z},\mathbf{y}|\boldsymbol{\theta})d\mathbf{y}d\boldsymbol{\theta}$
Inference: $\widehat{\boldsymbol{\theta}} = \int_{\boldsymbol{\theta}} \boldsymbol{\theta} p(\boldsymbol{\theta}|\mathbf{z})d\boldsymbol{\theta}$

INLA

INLA = Integrated nested Laplace approximation (software)
C-code, R-interface
Can be installed by

```
source("http://www.math.ntnu.no/inla/givemeINLA.R")
```

Both empirical Bayes and Bayes

R-INLA: An R-package for INLA

Håvard Rue

Department of Mathematical Sciences
NTNU, Norway

February 2, 2011

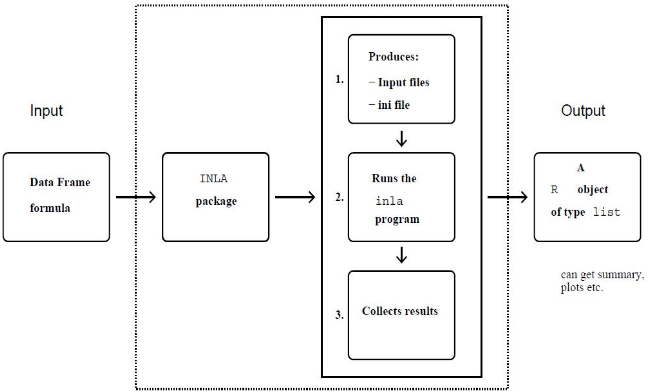
Full presentation on the course web page

Implementing INLA

All procedures required to perform INLA need to be carefully implemented to achieve a good speed; easy to implement a slow version of INLA.

- The `GMRFLib`-library
- The `inla`-program
- The `INLA` package for `R`
  - `R`-interface to the `inla`-program. (That's why its not on CRAN.)
  - Convert "formula"-statements into ".ini"-file definitions
  - Similar interface as other `R` packages

The INLA *package for* R



Main functions of the INLA *package*

- `f()` Define your model as a formula
- `inla()` Run the analysis
- `summary()`
- `plot()`
- `inla.hyperpar()`
- `inla.cpo()`

Documentation is available at `www.r-inla.org` and has help-pages in R

Model specification the INLA *package II*

The model is specified in R through a `formula`, similar to glm/gam++:

```
> formula = y ~ x1 + x2 + f(x3, ...)
```

The `f()` function is used to specify various "random"-effects in the model.
Some models

- `iid, iid1d, ii2d, iid3d`: random effects
- `rw1, rw2, ar1`: smooth effect of covariates or time effect
- `seasonal`: seasonal effect
- `besag`: spatial effect (CAR model)
- `generic`: user defined precision matrix

Example - simulated data

Assume

$$\begin{aligned} Y(\mathbf{s}) =& \beta_0 + \beta_1 x(\mathbf{s}) + \sigma_\delta \delta(\mathbf{s}), && \{\delta(\mathbf{s})\} \text{ matern correlation} \\ Z(\mathbf{s}_i) =& Y(\mathbf{s}_i) + \sigma_\varepsilon \varepsilon(\mathbf{s}_i), && \{\varepsilon(\mathbf{s}_i)\} \text{ independent} \end{aligned}$$

Simulations:

- Simulation on 20×30 grid
- $\{x(\mathbf{s})\}$ sinus curve in horizontal direction
- Parametervalues $(\beta_0, \beta_1) = (2, 0.3), \sigma_\delta = 1, \sigma_\varepsilon = 0.3$ and $(\theta_1, \theta_2) = (2, 3)$

Latent models in INLA

The R-INLA project

- The home of the R-INLA project
- Barrier Models
- Books
- Contact us, stay updated, get help or report an error
- Discussion forum
- Download
- Examples and tutorials
  - Case studies and code from papers
  - Tutorials
  - Volume I
  - Volume II
- FAQ
- Getting started
- Help
- Internal use
- Models
  - Latent models
  - Likelihoods
  - Priors
  - Tools to manipulate models and likelihoods
- News
- Papers
- Related projects
- Related R-packages
  - R-package: AnimalINLA
  - R-package: excursions
  - R-Package: INLABRU
  - R-package: ShrinkBayes
- Scientific meetings
- We're hiring!
- Sitemap

Models >

Latent models

Here is a list of all ***models for the latent Gaussian field*** which are implemented in the R inla package. For

Model	Name	Description
Independent random variables	`iid`	indep.pdf
Linear	`linear`	linear.pdf
Constrained linear	`clinear`	clinear.pdf
Random walk of order 1	`rw1`	rw1.pdf
Random walk of order 2	`rw2`	rw2.pdf
Continuous random walk of order 2	`crw2`	crw2.pdf
Model for seasonal variation	`seasonal`	seasonal.pdf
Model for spatial effect	`besag`	besag.pdf
Model for spatial effect	`besagproper`	besagproper.pdf
Model for weighted spatial effects	`besag2`	besag2.pdf
Model for spatial effect + random effect	`bym`	bym.pdf
Autoregressive model of order 1	`ar1`	ar1.pdf
Autoregressive model of order p	`ar`	ar.pdf
The Ornstein-Uhlenbeck process	`ou`	ou.pdf
User defined structure matrix, type 0	`generic0`	generic0.pdf
User defined structure matrix, type1	`generic1`	generic1.pdf
User defined structure matrix, type2	`generic2`	generic2.pdf
Model for correlated effects with Wishart prior (dimension 1, 2, 3, 4 and 5).	`iid1d, iid2d, iid3d, iid4d, iid5d`	iid123d.pdf
Classical random effect model	`z`	z.pdf
Random walk of 2nd order on a lattice	`rw2d`	rw2d.pdf
Gaussian field with Matern covariance function	`matern2d`	matern2d.pdf
Classical measurement error model	`mec`	mec.pdf

Latent model = INLA language for: "a Gaussian thing we use in the model"

The question is not whether you use a Gaussian model or not but how you do it!

Matern model - setup

C:\Forelesning\STK4150\R\inla_matern_linmodel.R - R Editor

```
#Necessary libraries
library(INLA)
library(geoR)
set.seed(231171)

#Size of grid (note you might want to reduce this if the computation is slow
nrow=20
ncol=30
n = nrow*ncol

#Indices in both directions
i.m = matrix(rep(1:ncol,each=nrow),nrow=nrow)
j.m = matrix(rep(1:nrow,ncol),ncol=ncol)
#Covariate
x.m = sin(j.m+0*i.m)
image(x.m,zlim=c(-2,2),col=rainbow(256))

#Converting matrices to vectors, inla requires input to be vectors
x = inla.matrix2vector(x.m)
i = inla.matrix2vector(i.m)
j = inla.matrix2vector(j.m)

#Parameter values
beta = c(2,0.3);sigma.y=1
#Making covariance matrix through distance matrix
d = as.matrix(dist(cbind(i,j),diag=TRUE,upper=TRUE))
C = sigma.y^2*matern(d,2,3)
par(mfrow=c(1,1))
image(beta[1]+beta[2]*x.m,zlim=c(-1,5),col=rainbow(256))
image(C,zlim=c(-2,2),col=rainbow(256),ylim=rev(c(0 ,1)))
```

Code on webpage

Matern model - setup

C:\Forelesning\STK4150\R\inla_matern_linmodel.R - R Editor

```r
#Necessary libraries
library(INLA)
library(geoR)
set.seed(231171)

#Size of grid (note you might want to reduce this if the computation is slow
nrow=20
ncol=30
n = nrow*ncol

#Indices in both directions
i.m = matrix(rep(1:ncol,each=nrow),nrow=nrow)
j.m = matrix(rep(1:nrow,ncol),ncol=ncol)
#Covariate
x.m = sin(j.m+0*i.m)
image(x.m,zlim=c(-2,2),col=rainbow(256))

#Converting matrices to vectors, inla requires input to be vectors
x = inla.matrix2vector(x.m)
i = inla.matrix2vector(i.m)
j = inla.matrix2vector(j.m)

#Parameter values
beta = c(2,0.3);sigma.y=1
#Making covariance matrix through distance matrix
d = as.matrix(dist(cbind(i,j),diag=TRUE,upper=TRUE))
C = sigma.y^2*matern(d,2,3)
par(mfrow=c(1,1))
image(beta[1]+beta[2]*x.m,zlim=c(-1,5),col=rainbow(256))
image(C,zlim=c(-2,2),col=rainbow(256),ylim=rev(c(0 ,1)))
```

Packages

Set seed to have reproducible calculations

dimensions

Spatial parameters

Covariate

Format to fit INLA

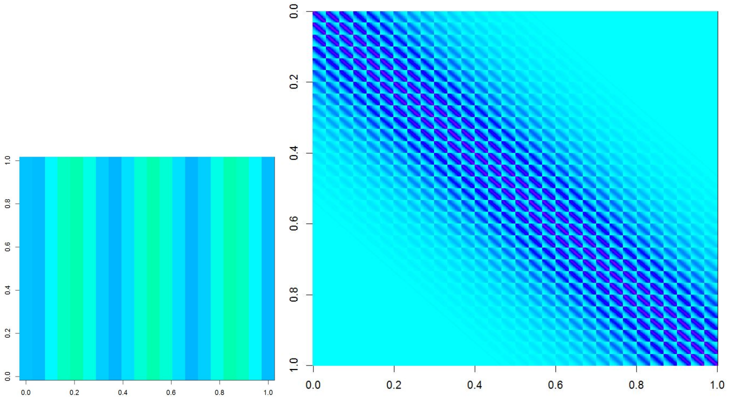
parameters

Distance matrix: 600 x 600

geoR

Plot mean and covariance

Process model mean and covariance



Mean: $20 \times 30 = 600$, Covariance: $600 \times 600 = 360000$

Matern model - parametrization

C:\Forelesning\STK4150\R\inla_matern_linmodel.R - R Editor

```
#Simulating data using Cholesky decomposition
C.chol = t(chol(C))
y = beta[1]+x*beta[2] + C.chol%*%matrix(rnorm(n),ncol=1)
#Additive Gaussian observation noise
z = y+0.3*rnorm(n)

y.m = inla.vector2matrix(y,nrow,ncol)
z.m = inla.vector2matrix(z,nrow,ncol)

#Plotting covariate and latent process and observations
par(mfrow=c(2,3))
image(x.m,zlim=c(-1,5),col=rainbow(256))
image(y.m,zlim=c(-1,5),col=rainbow(256))
image(z.m,zlim=c(-1,5),col=rainbow(256))

#Making dataframe for inla call, using delta for spatially correlated
#random effect
GridIndex = 1:n
data=data.frame(z=z,x=x,GridIndex=GridIndex)

#Defining model through a formula statement
#Note that the matern2d model assumes GridIndex is defined on a grid with
#dimensions nrow and ncol

formula= z ~ 1 + x +  f(GridIndex, model="matern2d", nu=3, nrow=nrow, ncol=ncol)

#inla call using empirical Bayes approach
result.eb=inla(formula, family="gaussian", data=data,
  control.predictor = list(compute = TRUE),
  control.inla=list(int.strategy="eb"))
summary(result.eb)

#inla call using Bayes approach
result.b=inla(formula, family="gaussian", data=data, control.predictor = list(compute = TRUE))
summary(result.b)
```

Matern model - parametrization

C:\Forelesning\STK4150\R\inla_matern_linmodel.R - R Editor

```
#Simulating data using Cholesky decomposition
C.chol = t(chol(C))
y = beta[1]+x*beta[2] + C.chol%*%matrix(rnorm(n),ncol=1)
#Additive Gaussian observation noise
z = y+0.3*rnorm(n)

y.m = inla.vector2matrix(y,nrow,ncol)
z.m = inla.vector2matrix(z,nrow,ncol)

#Plotting covariate and latent process and observations
par(mfrow=c(2,3))
image(x.m,zlim=c(-1,5),col=rainbow(256))
image(y.m,zlim=c(-1,5),col=rainbow(256))
image(z.m,zlim=c(-1,5),col=rainbow(256))

#Making dataframe for inla call, using delta for spatially correlated
#random effect
GridIndex = 1:n
data=data.frame(z=z,x=x,GridIndex=GridIndex)

#Defining model through a formula statement
#Note that the matern2d model assumes GridIndex is defined on a grid with
#dimensions nrow and ncol

formula= z ~ 1 + x +  f(GridIndex, model="matern2d", nu=3, nrow=nrow, ncol=ncol)

#inla call using empirical Bayes approach
result.eb=inla(formula, family="gaussian", data=data,
  control.predictor = list(compute = TRUE),
  control.inla=list(int.strategy="eb"))
summary(result.eb)

#inla call using Bayes approach
result.b=inla(formula, family="gaussian", data=data, control.predictor = list(compute = TRUE))
summary(result.b)
```

- Synthetic case and data
- Format to fit INLA
- Plot
- Numbering all grid nodes linearly
- Format
- Empirical Bayes
- Bayesian approach

Output INLA

Empirical Bayes vs Bayes

```
> (result.eb)

Call:
c("inla(formula = formula, family = \"gaussian\", data = data, control.predictor = list(compute = TRUE), ",  "    control.inla = list(int.strategy = \"

Time used:
 Pre-processing    Running inla Post-processing           Total
      0.3276081       3.2136819       0.3432090       3.8844991

Integration Strategy: Empirical Bayes

Model contains 3 hyperparameters
The model contains 2 fixed effect (including a possible intercept)

Likelihood model: gaussian

The model has 1 random effects:
1.'GridIndex' is a Matern2D model

> (result.b)

Call:
"inla(formula = formula, family = \"gaussian\", data = data, control.predictor = list(compute = TRUE))"

Time used:
 Pre-processing    Running inla Post-processing           Total
      0.3276088       3.8844841       0.2652051       4.4772980

Integration Strategy: Model contains 3 hyperparameters
The model contains 2 fixed effect (including a possible intercept)

Likelihood model: gaussian

The model has 1 random effects:
1.'GridIndex' is a Matern2D model
```

Output INLA

Empirical Bayes

```
> summary(result.eb)

Call:
c("inla(formula = formula, family = \"gaussian\", data = data, control.predictor = list(compute = TRUE), ",  "

Time used:
 Pre-processing    Running inla Post-processing           Total
         0.3432          8.7831          0.3276          9.4539

Fixed effects:
              mean     sd 0.025quant 0.5quant 0.975quant   mode kld
(Intercept) 1.7991 0.0826     1.6369   1.7990     1.9611 1.7991   0
x           0.3385 0.0645     0.2119   0.3385     0.4650 0.3385   0

Random effects:
Name      Model
 GridIndex   Matern2D model

Model hyperparameters:
                                          mean     sd 0.025quant 0.5quant 0.975quant   mode
Precision for the Gaussian observations 13.213 1.3969    10.6733   13.144     16.160 13.009
Precision for GridIndex                  1.398 0.2839     0.8915    1.385      1.999  1.368
Range for GridIndex                      4.901 0.3261     4.3208    4.878      5.600  4.822

Expected number of effective parameters(std dev): 259.71(0.00)
Number of equivalent replicates : 2.31

Marginal log-Likelihood:  -385.97
Posterior marginals for linear predictor and fitted values computed
```

Matern model - parametrization

	Model	Range par	Smoothness par
Book	$\propto \{\lvert\lvert\mathbf{h}\rvert\rvert/\theta_1\}^{\theta_2} K_{\theta_2}(\lvert\lvert\mathbf{h}\rvert\rvert/\theta_1)$	θ_1	θ_2
geoR	$\propto \{\lvert\lvert\mathbf{h}\rvert\rvert/\phi\}^{\kappa} K_{\kappa}(\lvert\lvert\mathbf{h}\rvert\rvert/\phi)$	ϕ	κ
INLA	$\propto \{\lvert\lvert\mathbf{h}\rvert\rvert * \kappa\}^{\nu} K_{\nu}(\lvert\lvert\mathbf{h}\rvert\rvert * \kappa)$	$1/\kappa$	ν

Note: INLA reports an estimate of another "range", defined as

$$r = \frac{\sqrt{8}}{\kappa}$$

corresponding to a distance where the covariance function is approximately zero. An estimate of the range parameter can then be obtained by

$$\frac{1}{\kappa} = \frac{r}{\sqrt{8}}$$

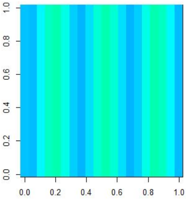
Note: INLA works with *precisions* $\tau_y = \sigma_y^{-2}, \tau_z = \sigma_z^{-2}$.

Result - INLA - empirical Bayes

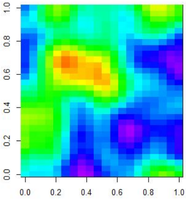
Parameter	True value	Estimate Run 1
β_1	2.0	1.7991
β_2	0.3	0.3385
σ_z	0.3	$\frac{1}{\sqrt{13.213}} = 0.275$
σ_y	1.0	$\frac{1}{\sqrt{1.398}} = 0.846$
ϕ_1	2.0	$\frac{4.901}{\sqrt{8}} = 1.7328$

Spatial prediction Run 1

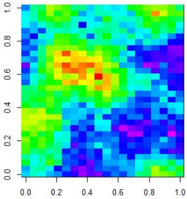
Prior mean

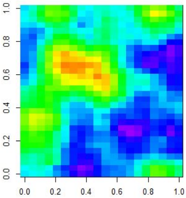


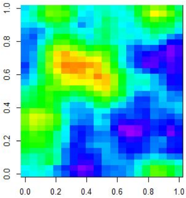
"True" intensity



Gaussian data







Empirical Bayes and Bayesian prediction

Result - INLA - empirical Bayes

Parameter	True value	Estimate Run 1	Estimate Run 2
β_1	2.0	1.799	1.9719
β_2	0.3	0.3385	0.4741
σ_z	0.3	$\frac{1}{\sqrt{13.213}} = 0.275$	$\frac{1}{\sqrt{13.6917}} = 0.2756$
σ_y	1.0	$\frac{1}{\sqrt{1.398}} = 0.846$	$\frac{1}{\sqrt{0.0014}} = 26.72$*
ϕ_1	2.0	$\frac{4.901}{\sqrt{8}} = 1.7328$	$\frac{17.44}{\sqrt{8}}$ =6.16*

* Commonly seen variance vs range tradeoff.

Large variance and long range vs

"on scale" variance and "on scale" range

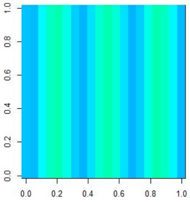
Allways compare your estimates with data variance.

Here: $\widehat{\text{Var}(Z)} = 1.03$, theortically ($\text{Var}(Z) = 1.09 = 1.0 + 0.3^2$)

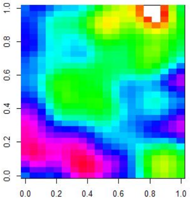
This is an argument for a Bayesian approach with an informative prior

Spatial prediction Run 2

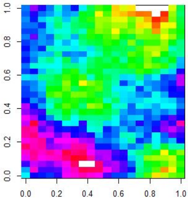
Prior mean

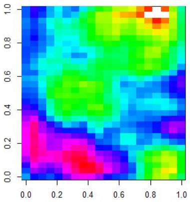


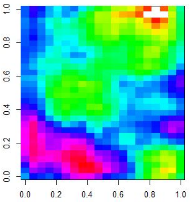
"True" intensity



Gaussian data







Empirical Bayes and Bayesian prediction

Example - simulated data

Assume

$$
\begin{aligned}
Y(\mathbf{s}) =& \beta_0 + \beta_1 x(\mathbf{s}) + \sigma_\delta \delta(\mathbf{s}), \qquad \{\delta(\mathbf{s})\} \text{ matern correlation} \\
Z(\mathbf{s}_i) \sim& \text{Poisson}(\exp(Y(\mathbf{s}_i))
\end{aligned}
$$

Simulations:

- Simulation on 20×30 grid
- $\{x(\mathbf{s})\}$ sinus curve in horizontal direction
- Parametervalues $(\beta_0, \beta_1) = (1, 0.3), \sigma_\delta = 1, \sigma_\varepsilon = 0.3$ and $(\theta_1, \theta_2) = (2, 3)$

Spatial prediction, Code Poisson

Empirical Bayes (note constant term changed from 2 to 1, hence y-1)

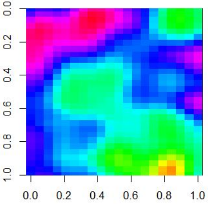
```
## poisson data
intensity=y-1;  # changed just to reduce the intensity slightly
z = rpois(n,exp(intensity))
z.m = inla.vector2matrix(z,nrow,ncol)

data=data.frame(z=z.m,x=x,GridIndex=GridIndex)
formula= z ~ 1 + x +  f(GridIndex, model="matern2d", nu=3, nrow=nrow, ncol=ncol
result.eb.pois=inla(formula, family="poisson", data=data,  control.inla=list(int.strategy="eb"))


pred.y.eb.pois = result.eb.pois$summary.fixed[1,1]+result.eb.pois$summary.fixed[2,1]*x+
            result.eb.pois$summary.random$GridIndex$mean
pred.y.m.eb.pois = inla.vector2matrix(pred.y.eb.pois,nrow,ncol)
par(mfrow=c(2,2))
image(intensity,zlim=c(-2,5),col=rainbow(256),ylim=rev(c(0 ,1)))
image(z.m,zlim=c(-1,15),col=rainbow(256),ylim=rev(c(0 ,1)))
image(pred.y.m.eb.pois,zlim=c(-2,5),col=rainbow(256),ylim=rev(c(0 ,1)))
```

Parameter prediction, Poisson

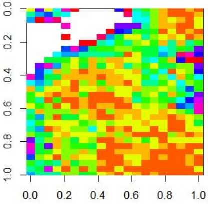
```
> summary(result.eb.pois)

Call:
"inla(formula = formula, family = \"poisson\", data = data, control.inla = list(int.strategy = \"eb\"))"

Time used:
 Pre-processing    Running inla Post-processing           Total
         0.3432          5.0857          0.1716          5.6005

Fixed effects:
              mean     sd 0.025quant 0.5quant 0.975quant   mode kld
(Intercept) 1.5271 0.1110     1.3086   1.5272     1.7444 1.5276   0
x           0.3266 0.0884     0.1534   0.3265     0.5002 0.3264   0

Random effects:
Name      Model
 GridIndex   Matern2D model

Model hyperparameters:
                          mean     sd 0.025quant 0.5quant 0.975quant  mode
Precision for GridIndex 0.7305 0.1903     0.4133   0.7128      1.155 0.681
Range for GridIndex     5.4164 0.4146     4.6727   5.3900      6.300 5.328

Expected number of effective parameters(std dev): 171.86(0.00)
Number of equivalent replicates : 3.491

Marginal log-Likelihood:  -1325.95
```

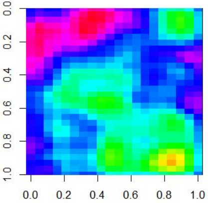
Spatial prediction, Poisson

"True" intensity



Poisson data (counts)





Estimated intensity

Inla engine: GMRFLib

Principle in GRMF = Gaussian Markov Random field is to invert the relation in "smoothing of independent Gaussian variables"

$$\mathbf{Y} = \mathbf{L}\varepsilon, \ \varepsilon \sim N(\mathbf{0}, \mathbf{I}) \Rightarrow \boldsymbol{Y} \sim N(\mathbf{0}, \boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T)$$

Inla engine: GMRFLib

Principle in GRMF = Gaussian Markov Random field is to invert the relation in "smoothing of independent Gaussian variables"

$$\mathbf{Y} = \mathbf{L}\varepsilon,\ \varepsilon \sim N(\mathbf{0}, \mathbf{I}) \Rightarrow \boldsymbol{Y} \sim N(\mathbf{0}, \boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T)$$

- Select the smooting kernel to be the cholesky factor: $\mathbf{Y} = \mathbf{L}\varepsilon$
- Then flip the equation and use the pression matrix $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$:

$$\mathbf{L}^{-1}\mathbf{Y} = \varepsilon,\ \varepsilon \sim N(\mathbf{0}, \mathbf{I}) \Rightarrow \boldsymbol{Y} \sim N(\mathbf{0}, \boldsymbol{Q} = \mathbf{L}^{-1}\mathbf{L}^{-T})$$

The formulations are equivalent but $\mathbf{L}$ and $\mathbf{L}^{-1}$ differs greately. This has major impact on computations.

Choleskey versus inverse choleskey in AR(1)

$$Y_1 = \sigma_1 \varepsilon_1, \quad Y_i = \alpha Y_{i-1} + \sigma \varepsilon_i$$

"Inverse formulation":

$$\varepsilon_1 = \frac{1}{\sigma_1} Y_1, \;\; \varepsilon_i = \frac{1}{\sigma} Y_i - \frac{\alpha}{\sigma} Y_{i-1}, \; i > 1$$

"Direct formulation":

$$Y_i = \alpha^{i-1} \sigma_1 \varepsilon_1 + \sigma \sum_{j=2}^{i} \alpha^{i-j} \varepsilon_j$$

The "Inverse formulation" can model long range dependency with a sparse matrix (few non-zero elements).

Complexity inverse formulation, – solving $\mathbf{L}^{-1}\mathbf{Y} = \boldsymbol{\varepsilon}$: $\sim n$

Complexity direct formulation, – multiply $\mathbf{Y} = \mathbf{L}\boldsymbol{\varepsilon}$: $\sim n^2$