KSBound: Kernel Stick-Breaking Prior Distribution for Spatial Boundary Detection

KSBound_Example

[1] Simulate data from the proposed model:

- Setting the reproducibility seed and initializing packages for data simulation:

```
set.seed(3354)

library(KSBound)
library(mnormt)  #Multivariate normal distribution
library(fields)
```

```
## Loading required package: spam
```

```
## Loading required package: dotCall64
```

```
## Loading required package: grid
```

```
## Spam version 2.5-1 (2019-12-12) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```

```
## Loading required package: maps
```

```
## See https://github.com/NCAR/Fields for
##  an extensive vignette, other supplements and source code
```

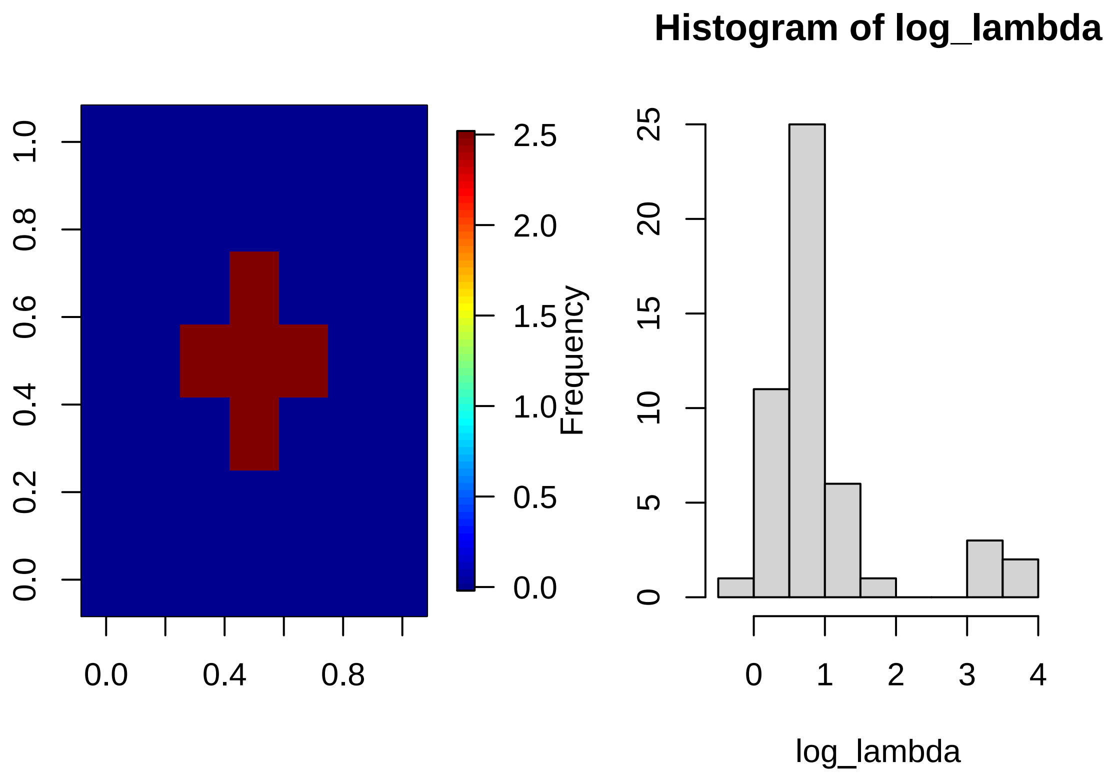
- Creating a Grid:

```
n.space<-7
x.easting<-1:n.space
x.northing<-1:n.space
Grid<-expand.grid(x.easting,
                  x.northing)
K<-nrow(Grid)

distance<-as.matrix(dist(Grid))
W<-array(0, c(K,K))
W[distance == 1]<-1  #Rook definition
diag(W)<-0
```

- Simulating Data:

```
offset<-rep(0,
            times = nrow(W))

beta_true<-c(0.75,
             -0.35)
```

```r
#Design Matrix
x<-matrix(1,
          nrow = nrow(W),
          ncol = 2)
x[,2]<-scale(rnorm(n = nrow(W)))

#Simulating the Data
phi<-rep(0,
         times = nrow(W))

center<-25
phi[W[center,] == 1]<-phi[W[center,] == 1] +
                      2.50
phi[center]<-phi[center] +
             2.50
neighbor_set<-c(1:nrow(W))[W[center,] == 1]

par(mfrow = c(1,2))
image.plot(matrix(c(phi),
                  nrow = n.space,
                  ncol = n.space,
                  byrow=TRUE))

log_lambda<-x%*%beta_true +
            phi

hist(log_lambda)
```




```r
y<-rpois(n = nrow(W),
         lambda = exp(log_lambda))
```

[2] Fit KSBound to estimate spatial boundaries:

```r
results<-KSBound(mcmc_samples = 25000,
                 m_max = 2000,
                 spatial_neighbors = W,
                 y = y,
                 offset = offset,
                 x = x,
                 mhvar_beta = rep(0.10,
                                  times = ncol(x)),
                 mhvar_theta = rep(0.01,
                                   times = 2000),
                 alpha_a_prior = 10.00,
                 alpha_b_prior = 10.00,
                 keep_all_ind = 0.00)
```

```
## Progress: 10%
## beta Acceptance (min): 24%
## beta Acceptance (max): 25%
## theta Acceptance (min): 1%
## theta Acceptance (max): 61%
## ****************************
## Progress: 20%
## beta Acceptance (min): 24%
```

```
## beta Acceptance (max): 24%
## theta Acceptance (min): 1%
## theta Acceptance (max): 59%
## ****************************
## Progress: 30%
## beta Acceptance (min): 24%
## beta Acceptance (max): 25%
## theta Acceptance (min): 0%
## theta Acceptance (max): 61%
## ****************************
## Progress: 40%
## beta Acceptance (min): 24%
## beta Acceptance (max): 24%
## theta Acceptance (min): 1%
## theta Acceptance (max): 59%
## ****************************
## Progress: 50%
## beta Acceptance (min): 24%
## beta Acceptance (max): 24%
## theta Acceptance (min): 1%
## theta Acceptance (max): 59%
## ****************************
## Progress: 60%
## beta Acceptance (min): 24%
## beta Acceptance (max): 24%
## theta Acceptance (min): 2%
## theta Acceptance (max): 58%
## ****************************
## Progress: 70%
## beta Acceptance (min): 24%
## beta Acceptance (max): 25%
## theta Acceptance (min): 1%
## theta Acceptance (max): 58%
## ****************************
## Progress: 80%
## beta Acceptance (min): 24%
## beta Acceptance (max): 25%
## theta Acceptance (min): 1%
## theta Acceptance (max): 59%
## ****************************
## Progress: 90%
## beta Acceptance (min): 24%
## beta Acceptance (max): 25%
## theta Acceptance (min): 1%
## theta Acceptance (max): 58%
## ****************************
## Progress: 100%
## beta Acceptance (min): 24%
## beta Acceptance (max): 25%
## theta Acceptance (min): 1%
## theta Acceptance (max): 58%
## ****************************
```

[3] Analyzing Output:

```r
par(mfrow=c(2,2))
plot(results$beta[1, 5001:25000],
     type="l",
     ylab="beta",
     xlab="Sample")
abline(h=beta_true[1],
       col="red",
       lwd=2)   #True value
plot(results$beta[2, 5001:25000],
     type="l",
     ylab="beta",
     xlab="Sample")
abline(h=beta_true[2],
       col="red",
       lwd=2)   #True value
plot(rowMeans(results$theta_g[,5001:25000]),
     pch=16,
     ylab="eta",
     xlab="Time",
     ylim = c(-1,3))
points(phi,
       pch=16,
       col="red")   #True values
```