DVCP: Directionally-Varying Change Points Model

DVCP_Example

[1] Install Packages:

```
library(devtools)
```

```
## Loading required package: usethis
```

```
install_github("warrenjl/DVCP")
```

```
## Skipping install of 'DVCP' from a github remote, the SHA1 (fff1c099) has not changed since last insta
##   Use `force = TRUE` to force installation
```

```
library(DVCP)
library(mnormt)
```

[2] Simulate Data for Analysis:

```
set.seed(46219)
n<-2000

locs<-matrix(0.00,
             nrow = n,
             ncol = 2)
locs[,1]<-runif(n = n,
                min = -1.00,
                max = 1.00)
locs[,2]<-runif(n = n,
                min = -1.00,
                max = 1.00)

p<-c(0.00,
     0.00)

p_dists<-rep(0.00,
             times = n)
for(j in 1:n){
   p_dists[j]<-sqrt((locs[j,1] - p[1])^2 +
                    (locs[j,2] - p[2])^2)
   }
max_p_dist<-max(p_dists)

angles<-rep(0.00,
            times = n)

angles[(locs[,1] > 0.00) & (locs[,2] > 0.00)]<-
asin(locs[((locs[,1] > 0.00) & (locs[,2] > 0.00)), 2]/
p_dists[(locs[,1] > 0.00) & (locs[,2] > 0.00)])*180.00/pi

angles[(locs[,1] < 0.00) & (locs[,2] > 0.00)]<-
(90.00 - asin(locs[((locs[,1] < 0.00) & (locs[,2] > 0.00)), 2]/
p_dists[(locs[,1] < 0.00) & (locs[,2] > 0.00)])*180.00/pi) + 90.00

angles[(locs[,1] < 0.00) & (locs[,2] < 0.00)]<-
-asin(locs[((locs[,1] < 0.00) & (locs[,2] < 0.00)), 2]/
```

```
p_dists[(locs[,1] < 0.00) & (locs[,2] < 0.00)])*180/pi + 180.00

angles[(locs[,1] > 0.00) & (locs[,2] < 0.00)]<-
(90.00 + asin(locs[((locs[,1] > 0.00) & (locs[,2] < 0.00)), 2]/
p_dists[(locs[,1] > 0.00) & (locs[,2] < 0.00)])*180.00/pi) + 270.00

angle_key<-c(1:n)

angle_dists<-matrix(0.00,
                    nrow = n,
                    ncol = n)
for(j in 1:n){
   for(k in 1:n){
      angle_dists[j,k]<-min(abs(angles[j] - angles[k]),
                            360.00 - abs(angles[j] - angles[k]))
      }
   }
angle_dists<-angle_dists/100.00  #Scaling

phi_true<-0.10
corr_mat<-exp(-phi_true*angle_dists)
sigma2_true<-1.00
eta_true<-rmnorm(n = 1,
                 mean = rep(0.00,
                            times = n),
                 varcov = (sigma2_true*corr_mat))
eta_true<-eta_true -
          mean(eta_true)
lambda_true<-0.50
cps_true<-lambda_true*exp(eta_true)
indicators_true<-as.numeric(p_dists <= cps_true)

theta_true<-1.00
h<-exp(-p_dists)

x<-matrix(1.00,
          nrow = n,
          ncol = 1)
beta0_true<-2.00

sigma2_epsilon_true<-0.10

y<-rnorm(n = n,
         mean = (beta0_true + theta_true*h*indicators_true),
         sd = sqrt(sigma2_epsilon_true))
```

[3] Fit DVCP:

```
results<-DVCP(mcmc_samples = 22000,
              burnin = 2000,
              thin = 20,
              adapt = 2000,
              likelihood_indicator = 1,  #Gaussian Response
              h_model = 2,  #Exponential h function
```

```
              approx_angles = c(seq(0,
                                    359.00,
                                    length.out = 25)),  #Approximation Number
              y = y,
              x = x,
              distance_to_p = p_dists/max(p_dists),  #Scaling
              unique_angles = angles,
              angle_key = angle_key,
              metrop_var_lambda = (0.20^2),
              metrop_var_eta = rep((0.25^2), times = 25),
              metrop_var_phi_eta = (1.05^2),
              adapt_lambda = (0.05^2),
              adapt_eta = (0.20^2),
              adapt_phi_eta = (0.50^2))
```

```
## Exponential
## Progress: 10%
## lambda Acceptance: 35%
## eta Acceptance (min): 12%
## eta Acceptance (max): 24%
## phi_eta Acceptance: 22%
## *************************
## Exponential
## Progress: 20%
## lambda Acceptance: 26%
## eta Acceptance (min): 17%
## eta Acceptance (max): 34%
## phi_eta Acceptance: 24%
## *************************
## Exponential
## Progress: 30%
## lambda Acceptance: 23%
## eta Acceptance (min): 18%
## eta Acceptance (max): 41%
## phi_eta Acceptance: 25%
## *************************
## Exponential
## Progress: 40%
## lambda Acceptance: 22%
## eta Acceptance (min): 19%
## eta Acceptance (max): 42%
## phi_eta Acceptance: 26%
## *************************
## Exponential
## Progress: 50%
## lambda Acceptance: 22%
## eta Acceptance (min): 20%
## eta Acceptance (max): 43%
## phi_eta Acceptance: 27%
## *************************
## Exponential
## Progress: 60%
## lambda Acceptance: 21%
## eta Acceptance (min): 20%
```

```
## eta Acceptance (max): 44%
## phi_eta Acceptance: 27%
## ************************
## Exponential
## Progress: 70%
## lambda Acceptance: 21%
## eta Acceptance (min): 21%
## eta Acceptance (max): 44%
## phi_eta Acceptance: 27%
## ************************
## Exponential
## Progress: 80%
## lambda Acceptance: 21%
## eta Acceptance (min): 21%
## eta Acceptance (max): 44%
## phi_eta Acceptance: 27%
## ************************
## Exponential
## Progress: 90%
## lambda Acceptance: 20%
## eta Acceptance (min): 21%
## eta Acceptance (max): 45%
## phi_eta Acceptance: 27%
## ************************
## Exponential
## Progress: 100%
## lambda Acceptance: 20%
## eta Acceptance (min): 21%
## eta Acceptance (max): 45%
## phi_eta Acceptance: 27%
## ************************
```

[4] Plotting Results:

```
plot(locs)
points(matrix(p,
              nrow = 1,
              ncol = 2),
       pch = "*",
       col = "red",
       cex = 3.00)
true_cp_locs<-matrix(0.00,
                     nrow = n,
                     ncol = 2)

true_cp_locs[((locs[,1] > 0.00) & (locs[,2] > 0.00)), 2]<-
sin(angles[(locs[,1] > 0.00) & (locs[,2] > 0.00)]*(pi/180.00))*
(lambda_true*exp(eta_true[(locs[,1] > 0.00) & (locs[,2] > 0.00)]))

true_cp_locs[((locs[,1] > 0.00) & (locs[,2] > 0.00)), 1]<-
cos(angles[(locs[,1] > 0.00) & (locs[,2] > 0.00)]*(pi/180.00))*
(lambda_true*exp(eta_true[(locs[,1] > 0.00) & (locs[,2] > 0.00)]))

true_cp_locs[((locs[,1] < 0.00) & (locs[,2] > 0.00)), 2]<-
sin((180.00 - angles[(locs[,1] < 0.00) & (locs[,2] > 0.00)])*
```

```
(pi/180.00))*(lambda_true*exp(eta_true[(locs[,1] < 0.00) & (locs[,2] > 0.00)]))

true_cp_locs[((locs[,1] < 0.00) & (locs[,2] > 0.00)), 1]<-
-cos((180.00 - angles[(locs[,1] < 0.00) & (locs[,2] > 0.00)])*
(pi/180.00))*(lambda_true*exp(eta_true[(locs[,1] < 0.00) & (locs[,2] > 0.00)]))

true_cp_locs[((locs[,1] < 0.00) & (locs[,2] < 0.00)), 2]<-
sin((180.00 - angles[(locs[,1] < 0.00) & (locs[,2] < 0.00)])*
(pi/180.00))*(lambda_true*exp(eta_true[(locs[,1] < 0.00) & (locs[,2] < 0.00)]))

true_cp_locs[((locs[,1] < 0.00) & (locs[,2] < 0.00)), 1]<-
-cos((180.00 - angles[(locs[,1] < 0.00) & (locs[,2] < 0.00)])*
(pi/180.00))*(lambda_true*exp(eta_true[(locs[,1] < 0.00) & (locs[,2] < 0.00)]))

true_cp_locs[((locs[,1] > 0.00) & (locs[,2] < 0.00)), 2]<-
-sin((360.00 - angles[(locs[,1] > 0.00) & (locs[,2] < 0.00)])*
(pi/180.00))*(lambda_true*exp(eta_true[(locs[,1] > 0.00) & (locs[,2] < 0.00)]))

true_cp_locs[((locs[,1] > 0.00) & (locs[,2] < 0.00)), 1]<-
cos((360.00 - angles[(locs[,1] > 0.00) & (locs[,2] < 0.00)])*
(pi/180.00))*(lambda_true*exp(eta_true[(locs[,1] > 0.00) & (locs[,2] < 0.00)]))

true_cp_locs<-true_cp_locs[order(angles),]
lines(true_cp_locs,
      lwd = 4.00,
      col = "green")

cp_ests<-rep(0.00,
             times = n)
for(j in 1:n){
   cp_ests[j]<-max_p_dist*median(results$lambda_keep*exp(results$eta_keep[j,]))
   }

est_cp_locs<-matrix(0.00,
                    nrow = n,
                    ncol = 2)

est_cp_locs[((locs[,1] > 0.00) & (locs[,2] > 0.00)), 2]<-
sin(angles[(locs[,1] > 0.00) & (locs[,2] > 0.00)]*
(pi/180.00))*(cp_ests[(locs[,1] > 0.00) & (locs[,2] > 0.00)])

est_cp_locs[((locs[,1] > 0.00) & (locs[,2] > 0.00)), 1]<-
cos(angles[(locs[,1] > 0.00) & (locs[,2] > 0.00)]*
(pi/180.00))*(cp_ests[(locs[,1] > 0.00) & (locs[,2] > 0.00)])

est_cp_locs[((locs[,1] < 0.00) & (locs[,2] > 0.00)), 2]<-
sin((180.00 - angles[(locs[,1] < 0.00) & (locs[,2] > 0.00)])*
(pi/180.00))*(cp_ests[(locs[,1] < 0.00) & (locs[,2] > 0.00)])

est_cp_locs[((locs[,1] < 0.00) & (locs[,2] > 0.00)), 1]<-
-cos((180.00 - angles[(locs[,1] < 0.00) & (locs[,2] > 0.00)])*
(pi/180.00))*(cp_ests[(locs[,1] < 0.00) & (locs[,2] > 0.00)])
```

```
est_cp_locs[((locs[,1] < 0.00) & (locs[,2] < 0.00)), 2]<-
sin((180.00 - angles[(locs[,1] < 0.00) & (locs[,2] < 0.00)])*
(pi/180.00))*(cp_ests[(locs[,1] < 0.00) & (locs[,2] < 0.00)])

est_cp_locs[((locs[,1] < 0.00) & (locs[,2] < 0.00)), 1]<-
-cos((180.00 - angles[(locs[,1] < 0.00) & (locs[,2] < 0.00)])*
(pi/180.00))*(cp_ests[(locs[,1] < 0.00) & (locs[,2] < 0.00)])

est_cp_locs[((locs[,1] > 0.00) & (locs[,2] < 0.00)), 2]<-
-sin((360.00 - angles[(locs[,1] > 0.00) & (locs[,2] < 0.00)])*
(pi/180.00))*(cp_ests[(locs[,1] > 0.00) & (locs[,2] < 0.00)])

est_cp_locs[((locs[,1] > 0.00) & (locs[,2] < 0.00)), 1]<-
cos((360.00 - angles[(locs[,1] > 0.00) & (locs[,2] < 0.00)])*
(pi/180.00))*(cp_ests[(locs[,1] > 0.00) & (locs[,2] < 0.00)])

est_cp_locs<-est_cp_locs[order(angles),]
lines(est_cp_locs,
      lwd = 4.00,
      col = "blue")
```