CASNET package RQA walk through for didactic purposes

James Twose

November 15th, 2018

```
#set working directory
setwd("/home/james/Data_Science")

#load casnet package
#install package from repository if this is first time of use
#devtools::install_github(https://github.com/FredHasselman/casnet)
library(casnet)

#read in eeg time series data
eeg <- read.csv("A001SB1_2.csv", sep = ",")
#subset x1 node and 50 instances to keep the speed up
eegx1 <- ts(eeg$x1[1:50])
length(eegx1)
```

```
## [1] 50
```

```
#casnet package output
#estimate the embedding lag (based on first minimum computed from average mutual information)
emlag <- est_emLag(eegx1)$optimal.lag[1]
emlag #output
```

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

```
est_emLag(eegx1)
```

```
##   selection.method optimal.lag       ami
## 1    first.minimum         1.0 0.9029512
## 2   global.minimum         3.0 0.1326533
## 3      maximum.lag        12.5 0.1487273
```

```
#estimate the embedding dimension (based on based on False Nearest Neighbours
#the embedding dimension from casnet is the n-1 emdim at
#which the emlag has an attractor tolerance of 0
#this will be used to create a binary matrix for plotting
#thresholded RQAs)
emdim <- est_emDim(y = eegx1, delay = emlag)$EmbeddingDim
emdim #output
```

```
## [1] 6
```

```
est_emDim(y = eegx1, delay = emlag)
```

```
##   EmbeddingLag EmbeddingDim
## 1            1            6
```

```
#creates a recurrence matrix based on the previously
#estimated embedding dimension and embedding lag
eegx1rm <- rp(y1 = eegx1, emDim = emdim, emLag = emlag)
#note that the size of the matrix is 44 columns by
```

```r
#44 rows - this is due to the lag (1) and dimension (6)
#that we estimated 50 - (1 x 6) = 44, and that each column
#is lagged by one, and that there is a diagonal of zeros -
#the line of inflection
head(eegx1rm[, 1:6]) #the first 6 x 6 of the matrix to get an idea
```

```
##           [,1]      [,2]     [,3]     [,4]      [,5]      [,6]
## [1,]  0.000000 13.664525 21.16899 15.05633  9.386156 17.985310
## [2,] 13.664525  0.000000 14.70284 22.23537 19.331360  9.837603
## [3,] 21.168991 14.702841  0.00000 19.74192 29.266733 21.416012
## [4,] 15.056325 22.235367 19.74192  0.00000 20.429316 29.512808
## [5,]  9.386156 19.331360 29.26673 20.42932  0.000000 22.394466
## [6,] 17.985310  9.837603 21.41601 29.51281 22.394466  0.000000
```

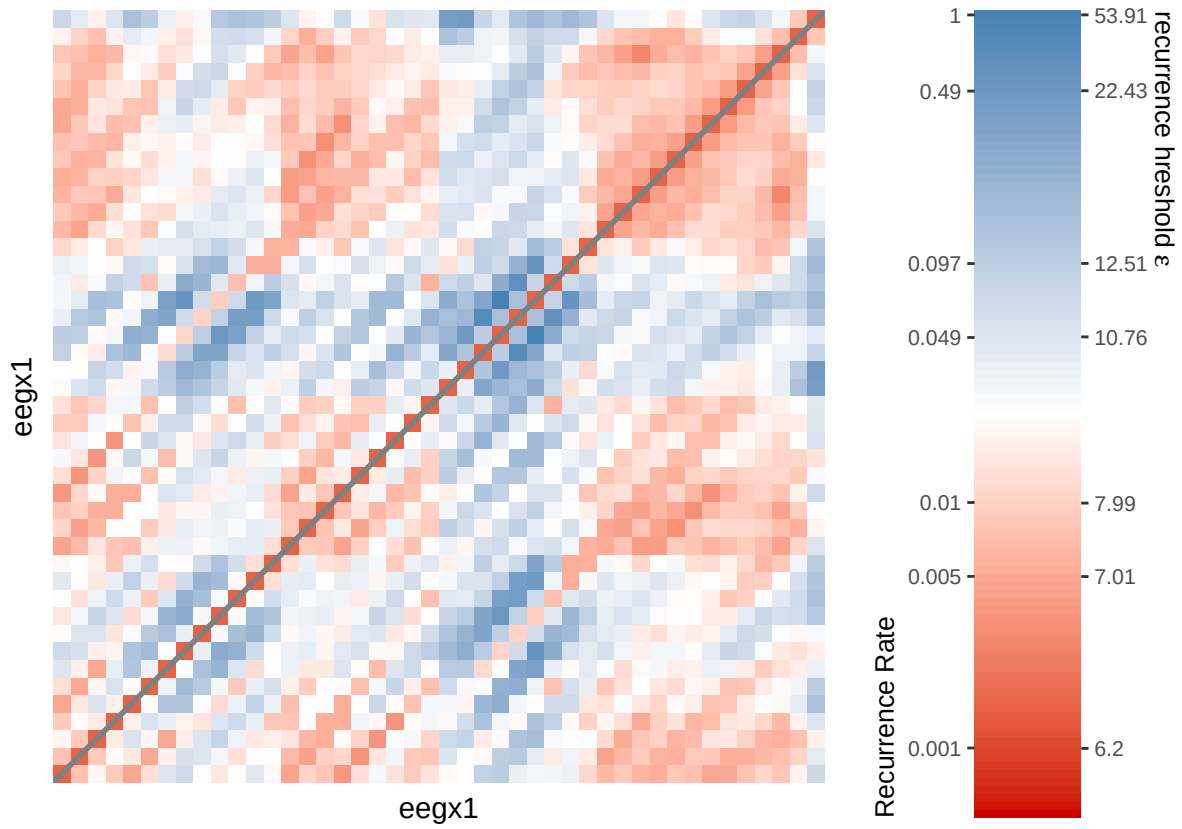
```r
#estimate the embedding radius - function created by Fred Hasselman,
#finds a radius of which a target (RQA) measure has a specific percentage
#(default is RR (Recurrence Rate) = 0.05) this is later used to plot a thresholded
#recurrence plot
(emrad <- crqa_radius(eegx1rm, targetMeasure = "RR", targetValue = 0.05, tol = 0.5, maxIter = 10)$Radius
```

```
##
## Auto-recurrence: Setting diagonal to (1 + max. distance) for analyses

## lower and upper are both 0 (no band, just diagonal)
##  using: diag(mat) <- 54.9145...

##
## Searching for a radius that will yield 0.05 for RR

##
## Converged! Found an appropriate radius...

## [1] 10.76312
```

```r
crqa_radius(eegx1rm, targetValue = 0.05, tol = 0.5, maxIter = 10)
```

```
##
## Auto-recurrence: Setting diagonal to (1 + max. distance) for analyses

## lower and upper are both 0 (no band, just diagonal)
##  using: diag(mat) <- 54.9145...

##
## Searching for a radius that will yield 0.05 for RR

##
## Converged! Found an appropriate radius...

##   iter    Measure   Radius targetValue tollo tolhi startRadius rp.size
## 1    1 0.04855372 10.76312        0.05 0.025 0.075    10.76312    1892
##   AUTO Converged
## 1 TRUE      TRUE
```

```r
#plot the unthresholded recurrence plot
# - this is basically a heat map of the recurrence matrix
rp_plot(eegx1rm)
```




```
#now place the previously calculated embedding radius on the unthresholded matrix
#to create a thresholded matrix (this is a binary matrix, where if the value is
#bigger than the embedding radius (10.76ish in this case), it gets assigned a ".", if
#not it is assigned a "1")
eegx1rmth <- rp(y1 = eegx1, emDim = emdim, emLag = emlag, emRad = emrad)
eegx1rmth[1:6, 1:6]  #the first 6 x 6 of the matrix to get an idea
```

```
## 6 x 6 sparse Matrix of class "dgCMatrix"
##
## [1,] 1 . . . 1 .
## [2,] . 1 . . . 1
## [3,] . . 1 . . .
## [4,] . . . 1 . .
## [5,] 1 . . . 1 .
## [6,] . 1 . . . 1
```

```
head(eegx1rm[, 1:6]); eegx1rmth[1:6, 1:6] #for comparison
```

```
##           [,1]      [,2]     [,3]     [,4]      [,5]      [,6]
## [1,]  0.000000 13.664525 21.16899 15.05633  9.386156 17.985310
## [2,] 13.664525  0.000000 14.70284 22.23537 19.331360  9.837603
## [3,] 21.168991 14.702841  0.00000 19.74192 29.266733 21.416012
## [4,] 15.056325 22.235367 19.74192  0.00000 20.429316 29.512808
## [5,]  9.386156 19.331360 29.26673 20.42932  0.000000 22.394466
## [6,] 17.985310  9.837603 21.41601 29.51281 22.394466  0.000000
```

```
## 6 x 6 sparse Matrix of class "dgCMatrix"
##
```

```
## [1,] 1 . . . 1 .
## [2,] . 1 . . . 1
## [3,] . . 1 . . .
## [4,] . . . 1 . .
## [5,] 1 . . . 1 .
## [6,] . 1 . . . 1
```

```
#plot the thresholded recurrence plot
# - this is basically a plot of 1s and .s based on the recurrence matrix
rp_plot(eegx1rmth)
```