Birla Institute of Technology and Science

Foundations of Data Science

(CS F320)

I Sem 2021-22

Lab Sheet #10: Dynamic time warping

Dynamic time warping (DTW) is the name of a class of algorithms for comparing series of values with each other. The rationale behind DTW is, given two time series, to stretch or compress them locally in order to make one resemble the other as much as possible.

The dtw package allows R users to compute time series alignments mixing freely a variety of continuity constraints, restriction windows, endpoints, local distance definitions, and so on. The package also provides functions for visualizing alignments and constraints using several classic diagram types

Installing 'dtw' package

To install the package run the 'install.packages()' command and load it with 'library()' command.

```
install.packages("dtw")
library(dtw)
```

Creating Sequence Data

```
n=100
idx<-seq(0,6.28,len=n);
query<-sin(idx)+runif(n)/10;
reference<-cos(idx)
plot(query, type = "l")
lines(reference, col = "blue")
```

dtw() function takes two vector arguments representing the input time series, performs the minimization, and returns an object of class dtw encapsulating all the alignment information

```
alignment<-dtw(query,reference,keep=TRUE);
alignment$costMatrix

dm<-matrix(10,4,4)+diag(rep(1,4))
al<-dtw(dm,k=T,step=symmetric1)
a2<-dtw(dm,k=T,step=symmetric2)

a1$costMatrix
a2$costMatrix
```

Display the recursion formula for the properly symmetric continuity constraint

```
symmetric1
```

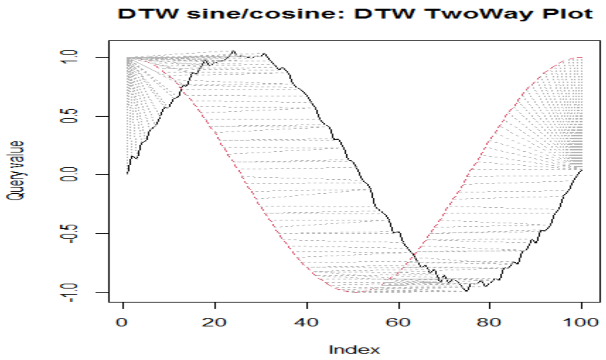
```
Step pattern recursion:
g[i,j] = min(
    g[i-1,j-1] +     d[i  ,j  ] ,
    g[i  ,j-1] +     d[i  ,j  ] ,
    g[i-1,j  ] +     d[i  ,j  ] ,
 )

 Normalization hint: NA
```

The so-called symmetric recursion printed above allows an unlimited number of elements of the query to be matched to a single element of the reference, and vice-versa; in other words, there is no limit in the amount of time expansion or compression allowed at any point

Plot style of simple alignment and Two Way Plot

```
plot(alignment, type="alignment", main="DTW sine/cosine: simple alignment plot")
plot(alignment, type="twoway",main="DTW sine/cosine: Two Way Plot")
```



Align two synthetic bivariate time series (a query with 10 time points, and a reference of length 5), assuming the Manhattan local distance for element pairs

```
query <- cbind(1:10,1)
ref <- cbind(11:15,2)
dtw(query,ref,dist.method="Manhattan")$distance
```

Find the best path through a 6 × 6 local distance matrix. We enter the given matrix as follows:

```
lm <- matrix(nrow = 6, ncol = 6, byrow = TRUE, c(
  + 1, 1, 2, 2, 3, 3,
  + 1, 1, 1, 2, 2, 2,
  + 3, 1, 2, 2, 3, 3,
  + 3, 1, 2, 1, 1, 2,
  + 3, 2, 1, 2, 1, 2,
  + 3, 3, 3, 2, 1, 2))
```

Find the best global path going through the grid, we invoke dtw with one matrix argument.

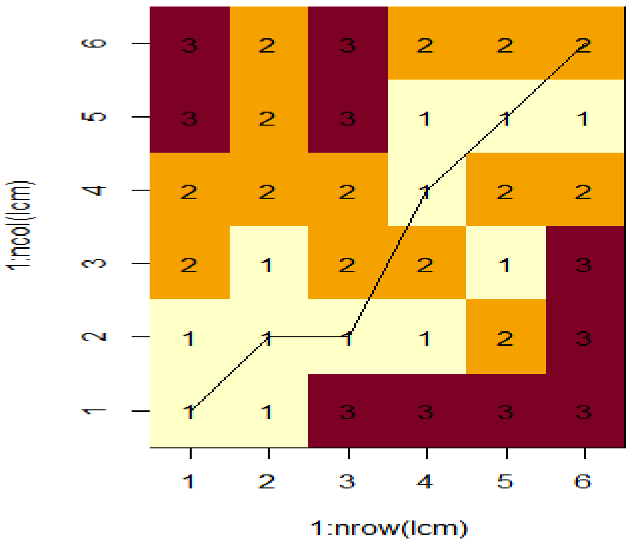
```
alignment <- dtw(lm,step=asymmetric,keep=TRUE)
#From the result, we extract the cost matrix  and the normalized distance.
alignment$costMatrix
alignment$normalizedDistance
```

The optimal alignment between the whole test and any prefix of the reference; in other words, we lift the constraint on the reference end-point. This is achieved with the setting open.end = TRUE.

```
alignmentOE <- dtw(lm,step=asymmetric,keep=TRUE,open.end=TRUE)
alignmentOE$jmin
alignmentOE$normalizedDistance

#Retriving Cost Matrices

lcm <- alignment$localCostMatrix
image(x=1:nrow(lcm),y=1:ncol(lcm),lcm)
text(row(lcm),col(lcm),label=lcm)
lines(alignment$index1,alignment$index2)
```



It shows how to display them along with the alignment path superimposed. No better warping path exists that passes by (1, 1) through (6, 6) under the constraints of the asymmetric pattern

Retriving cumulative Cost Matrices

```
ccm <- alignment$costMatrix
image(x=1:nrow(ccm),y=1:ncol(ccm),ccm)
text(row(ccm),col(ccm),label=ccm)
lines(alignment$index1,alignment$index2)
```

Working on Dataset

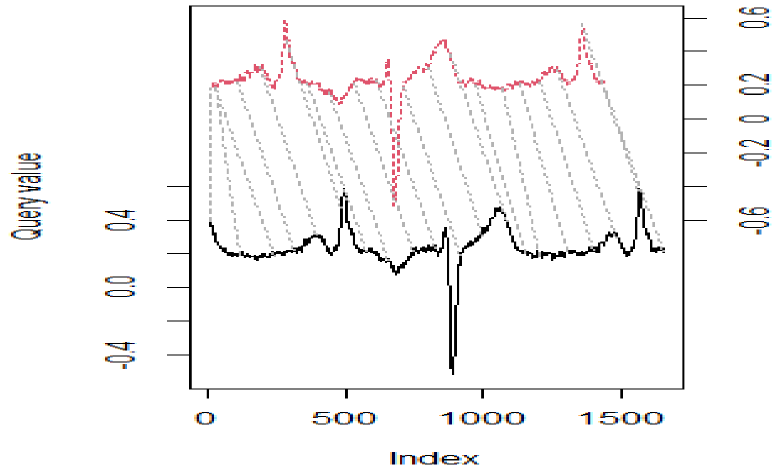
The aami3a time series included in dtw package contains a reference electrocardiogram from the PhysioBank dataset .

```
data("aami3a")
```

Extract two non-overlapping windows from it as reference and test data, and compute their optimal alignment with the default dtw settings

```
ref <- window(aami3a,start=0,end=2)
test <- window(aami3a,start=2.7,end=5)
plot(dtw(test,ref,k=TRUE),type="two",off=1,match.lty=2,match.indices=20)
```

The two extracted segments are shown below:



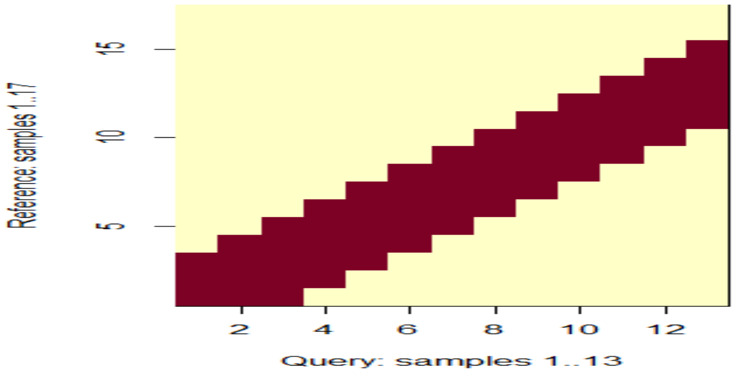
```
alignment <- dtw(test,ref)
alignment$distance
```

Compute the alignment, assuming the well-known asymmetric pattern

```
alignment <- dtw(test,ref,step.pattern=asymmetric)
alignment$distance
```

.The allowed region for the warping curve under the "sakoechiba" global constraint can be identified using dtwWindow

```
dtwWindow.plot(sakoeChibaWindow, window.size=2,reference=17, query=13)
```



Exercise

1. Design a three-way plot of the asymmetric alignment between a noisy sine and a cosine in [0, 2π], with visual guide lines drawn every π/4.