

Clustering a Similarity Matrix

Parismita Das

5 March 2017

Creating Similarity Matrix

We are given with a symmetrical Dissimilarity matrix, and as all elements are positive in the given matrix, the kernel SOM and relational SOM are equivalent. Hence using the equation:

$$s(i, j) = -\frac{1}{2} \left(\delta^2(x_i, x_j) - \frac{1}{n} \sum_{k=1}^n \delta^2(x_i, x_k) - \frac{1}{n} \sum_{k=1}^n \delta^2(x_k, x_j) + \frac{1}{n^2} \sum_{k,k'=1}^n \delta^2(x_k, x_{k'}) \right)$$

The algorithm to calculate the Similarity Matrix

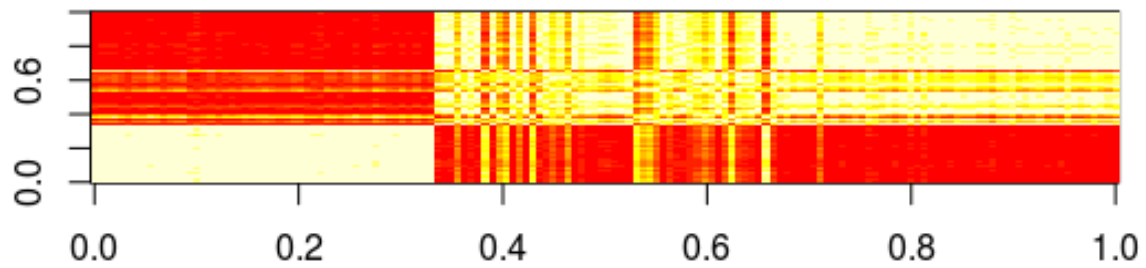
```
# algo to calculate similarity matrix s from the equation given where
#K is the dissimilarity matrix delta(xi,xj)
#Dkk is sum[k,k':1 to n](delta(xk,xk')^2), Dij is delta(xi,xj)^2,
#Dik and Dkj is sum[k:1 to n](delta(xi,xk)^2) and sum[k:1 to n](delta(xk,xj)^2) resp
n <- 150
K <- K^2
Dkk <- sum(K)
Dij <- K
Dik <- apply(K, 1, sum)
Dkj <- apply(K, 2, sum)
Dikkj <- 1/n*outer(Dik,Dkj,FUN = '+')
Dk <- matrix(1/n^2*Dkk,nrow=150,ncol=150,byrow=TRUE)

# the similarity matrix
s <- -0.5*(Dij - Dikkj + Dk)
```

The adjClustBand_heap function silently assumes that the diagonal of the matrix is 1. Hence we scale the similarity matrix such that the diagonal of the matrix is 1.

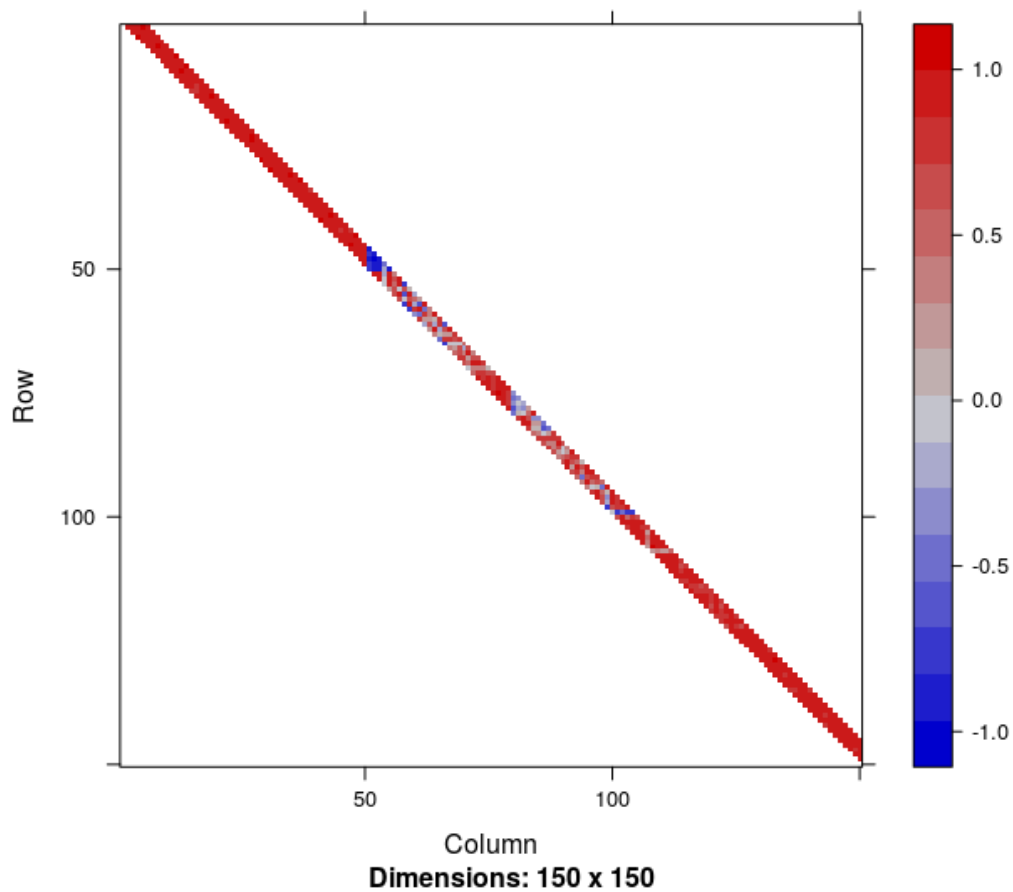
```
# scaling of the similarity matrix and making diagonal column as 1,
# as required by adjClustBand_heap
s_scaled <- sweep(s,1,sqrt(diag(s)), '/')
s_scaled <- sweep(s_scaled,2,sqrt(diag(s)), '/')
```

We get the Similarity Matrix Image as:



Now Extracting the diagonal band of width 5 that is to be used for clustering, using the function HeapHop

```
# extracting diagonal band
low <- 1
high <- h
delta <- col(s) - row(s)
s[delta < low | delta > high] <- 0
```



Creating Dendrograms and cluster plot, clustering via adjClustBand

```
fit <- adjclust:::adjClustBand_heap(x, p, h, blMin=1)
plot(fit)
```

