Feature Space Diagram in R

John Erickson

2022-12-07

Introduction

This is a first effort to implement "feature space diagrams" in R, inspired by https://towardsdatascience.com /escape-the-correlation-matrix-into-feature-space-4d71c51f25e5

```
library(lessR)
library(Hmisc)
library(corrplot)
library(gplots)
library(igraph)
library(tidyverse)
```

Workflow

Our workflow is generally as described in the original post:

- Generate the correlation matrix
- Take the absolute value of correlation matrix and subtract each value from 1. The result is a distance matrix.
- Use PCA to reduce our NxN matrix to Nx2.
- Plot each feature's location using the two principal components.
- Use Feature Agglomeration to generate feature clusters.
- Color each feature by its cluster.
- Draw lines to represent relationships of at least r = 0.7 (or user's choosing)

Data Load

```
# TODO: User uploads data
# Load data
boston <- read.csv("boston.csv", header = TRUE, fileEncoding="latin1")
WDBC <- read.csv("WDBC.csv", header = TRUE, fileEncoding="latin1")

# Ensure we have a matrix
#mydata <- as.matrix(boston)
mydata <- as.matrix(WDBC[,3:32])
```

Initial Correlation Matrix

```
# Simple correlation matrix calculation
# TODO: user chooses method
mydata.cor <- cor(mydata, method = c("pearson", "kendall", "spearman"), use = "complete.obs")

# # Optional
```

```
# corrplot(mydata.cor)
#
# palette <- colorRampPalette(c("green", "white", "red")) (20)
# heatmap.2(x = mydata.cor, col = palette, symm = TRUE)

# Reorder correlation matrix based on: https://rdrr.io/cran/lessR/man/corReorder.html
mydata.cor.ro <- corReorder(mydata.cor)
```


dst
hclust (*, "complete")
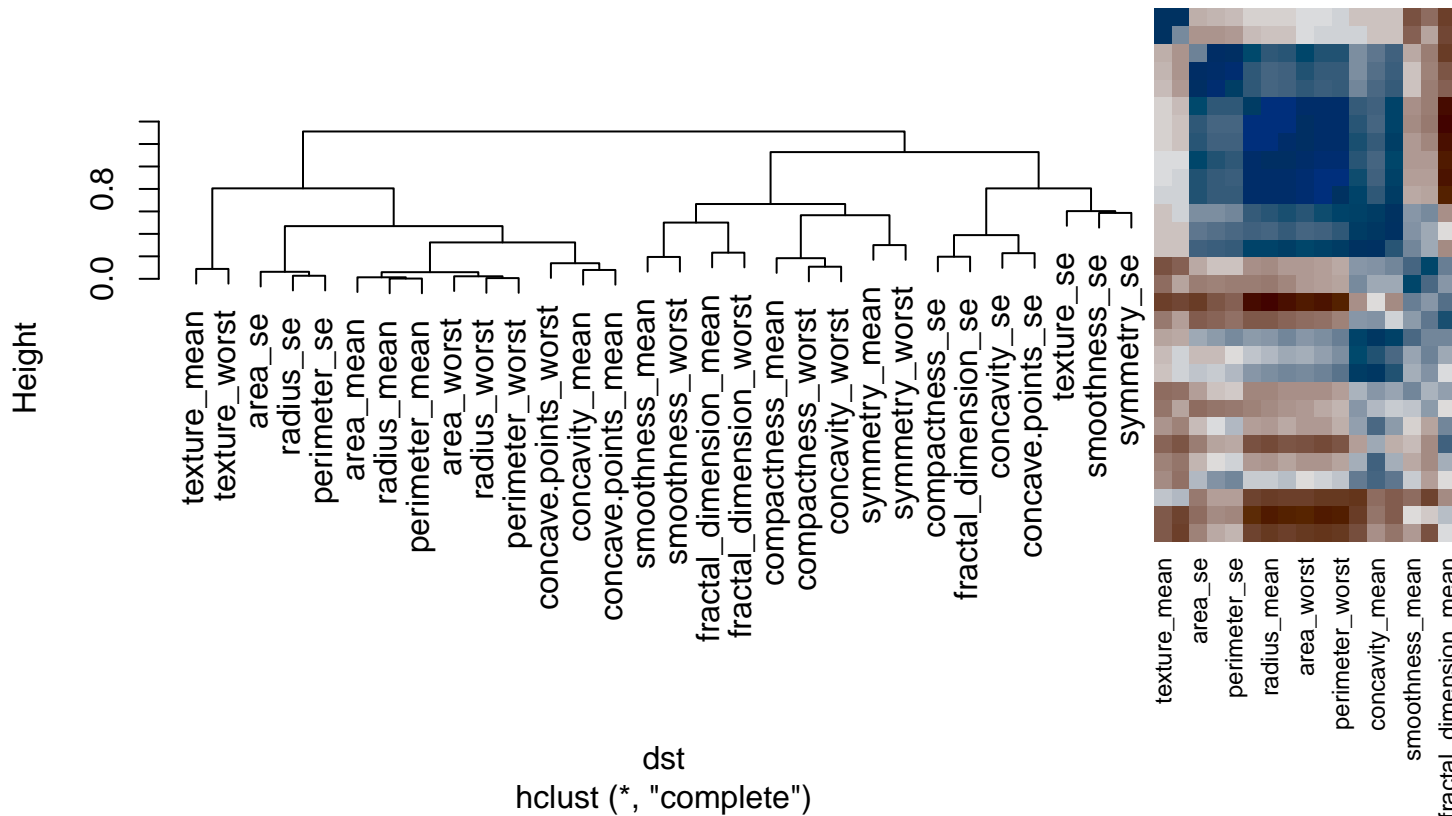
Distance Matrix Calculation

```
# Distance matrix (Absolute value et.al.)
mydata.cor.ro.1 <- abs(mydata.cor.ro) - 1
```

PCA on Distance Matrix

```
# PCA
mydata.cor.ro.1.pca <- prcomp(mydata.cor.ro.1)

# Plottable version:
mydata.cor.ro.1.pca.plot <- as.data.frame(mydata.cor.ro.1.pca$x[,1:2])
```

Finding the Clusters

```
# Pull out our clusters
mydata.cor.ro.1.pca.plot$cluster <- as.factor(kmeans(mydata.cor.ro.1.pca.plot$x[,1:2], centers=6)$cluster)
mydata.cor.ro.1.pca.plot$name <- rownames(mydata.cor.ro.1)
```

Finding the Graph!

This is the hard part...

```r
# Determine connectivity!
# filtering
selector <- ((abs(mydata.cor.ro.1) <= 0.35 ) * 1)

for (i in 1:length(mydata.cor.ro.1.pca.plot$cluster)) {selector[,i] <- selector[,i] * as.numeric(mydata
for (i in 1:length(mydata.cor.ro.1.pca.plot$cluster)) {selector[i,i] <- 0 }

# Switching to networks
# Create igraph structure
network <- graph_from_adjacency_matrix(selector, weighted = TRUE)

# Gets us our edges!
mydata.edges <- get.data.frame(network)

# replace `from` with X1 and Y1, and `to` with X2 and Y2
# These will be our line segments!
mydata.segments <- mydata.edges %>%
  left_join(mydata.cor.ro.1.pca.plot, by=c("from"="name")) %>%
  select(-cluster) %>%
  mutate(X1=PC1, Y1=PC2) %>%
  select(-PC1, -PC2)

mydata.segments <- mydata.segments %>%
  left_join(mydata.cor.ro.1.pca.plot, by=c("to"="name")) %>%
  select(-cluster) %>%
  mutate(X2=PC1, Y2=PC2) %>%
  select(-PC1, -PC2)

mydata.cor.ro.1.pca.plot$cluster <- as.factor(mydata.cor.ro.1.pca.plot$cluster)

# This really isn't "weight"...
mydata.segments$weight <- as.factor(mydata.segments$weight)
```

Building the Plot

```r
# Adding line segments to PCA plot
ggplot(mydata.cor.ro.1.pca.plot, aes(x=PC1, y=PC2)) +
  geom_point(aes(color=cluster, size=4), show.legend = FALSE) +
  geom_text(aes(label=name),hjust=1, vjust=0) +
  geom_segment(aes(x=X1,y=Y1,xend=X2,yend=Y2,color=weight), data=mydata.segments) +
#  labs(title="Feature Space Diagram for the Boston Housing data set")
  labs(title="Feature Space Diagram for the Wisconsin Breast Cancer data set")
```

Feature Space Diagram for the Wisconsin Breast Cancer data set
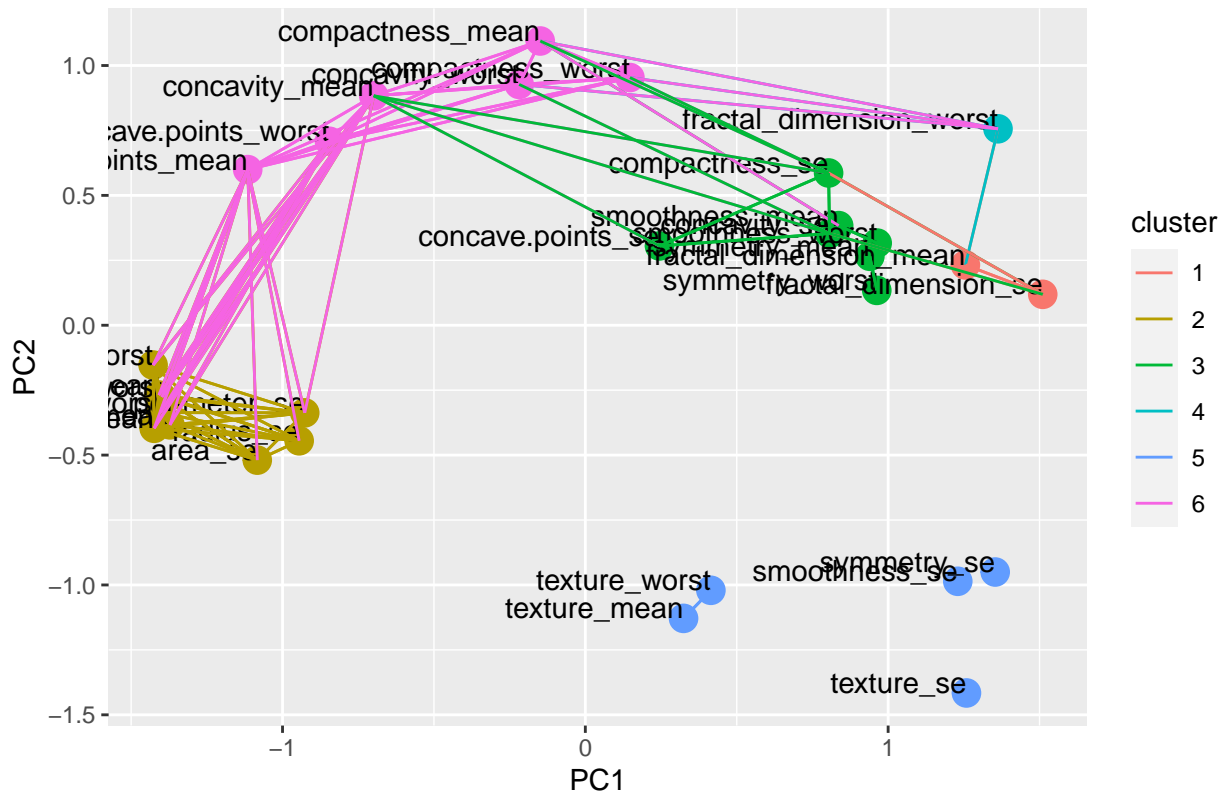
```
# ggplot() +
#   geom_segment(aes(x=X1,y=Y1,xend=X2,yend=Y2,color=weight), data=mydata.segments.test)
```