Feature Space Diagram in R

John Erickson

2022-12-09

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)

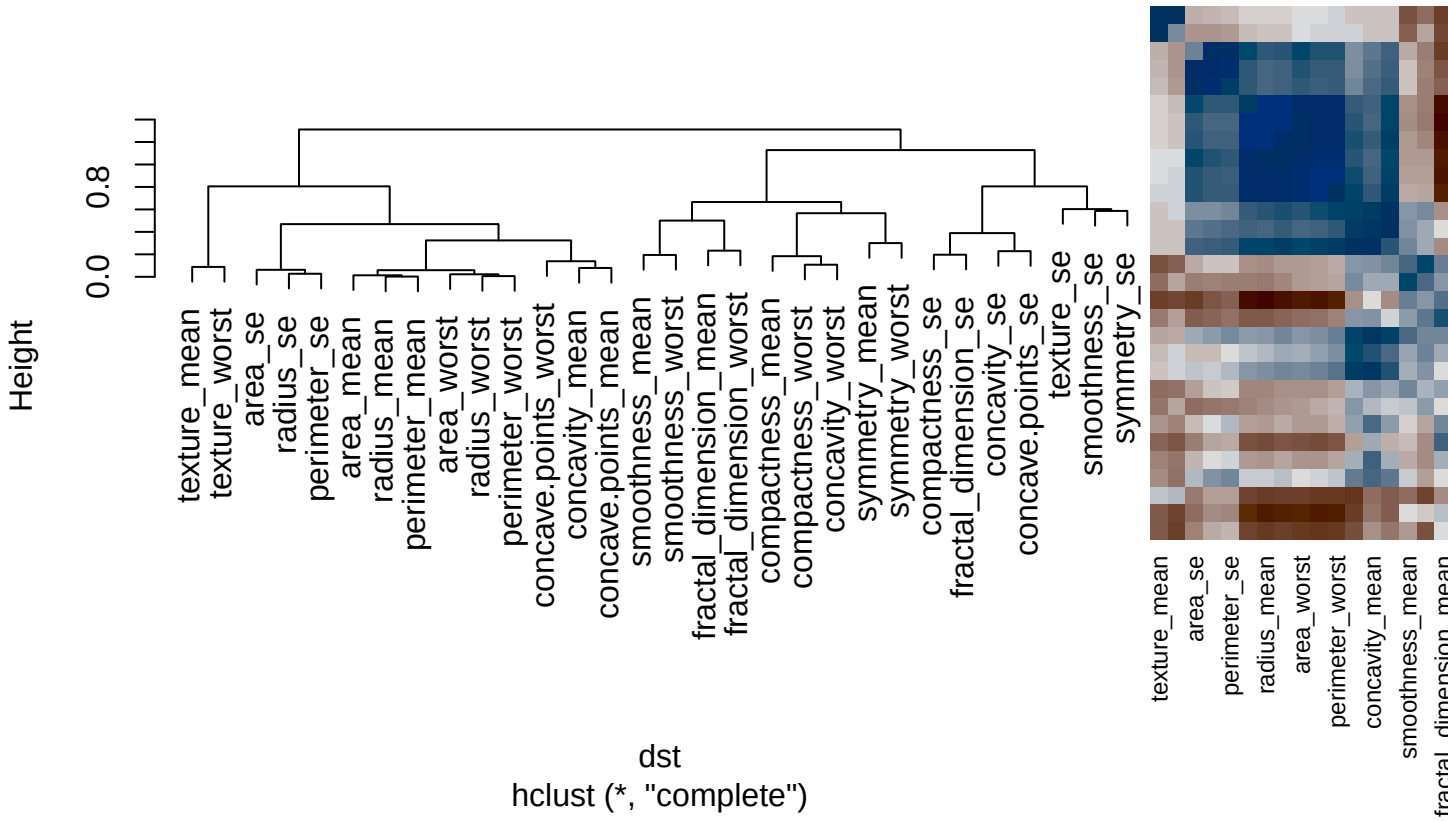
Data Load

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

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

Initial Correlation Matrix

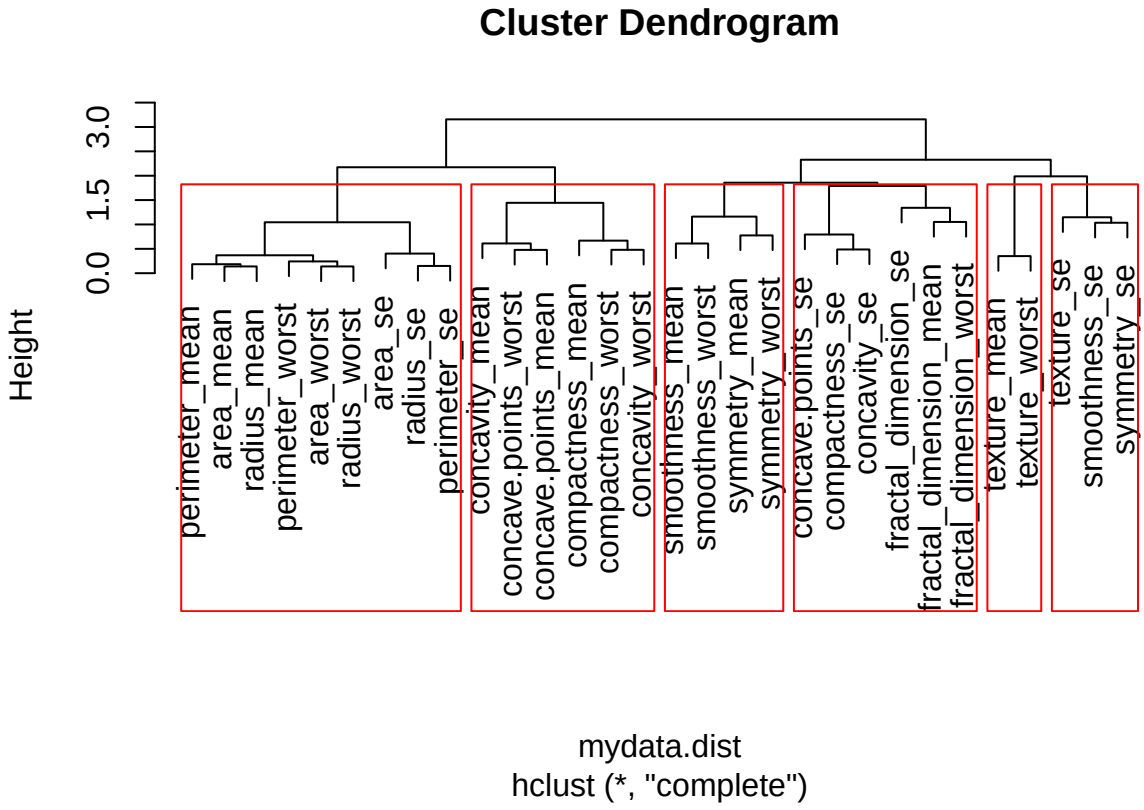
```
# Simple correlation matrix calculation
# TODO: user chooses method
mydata.cor <- cor(mydata, method = c("pearson", "kendall", "spearman"), use = "complete.obs")

# # 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)
```



Distance Matrix and Hiearchical/Agglomerative Clustering

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

# Calculate distances using stats::dist
mydata.dist <- dist(mydata.cor.ro.1)

# Calculate clusters using stats::hclust
mydata.hclust <- hclust(mydata.dist)
plot(mydata.hclust) # Plot it, just because

mydata.hclust.groups <- cutree(mydata.hclust, k=6)
rect.hclust(mydata.hclust, k=6, border="red") # Plot, highlighted by rectangles
```



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

Finding the Graph!

This is the hard part...

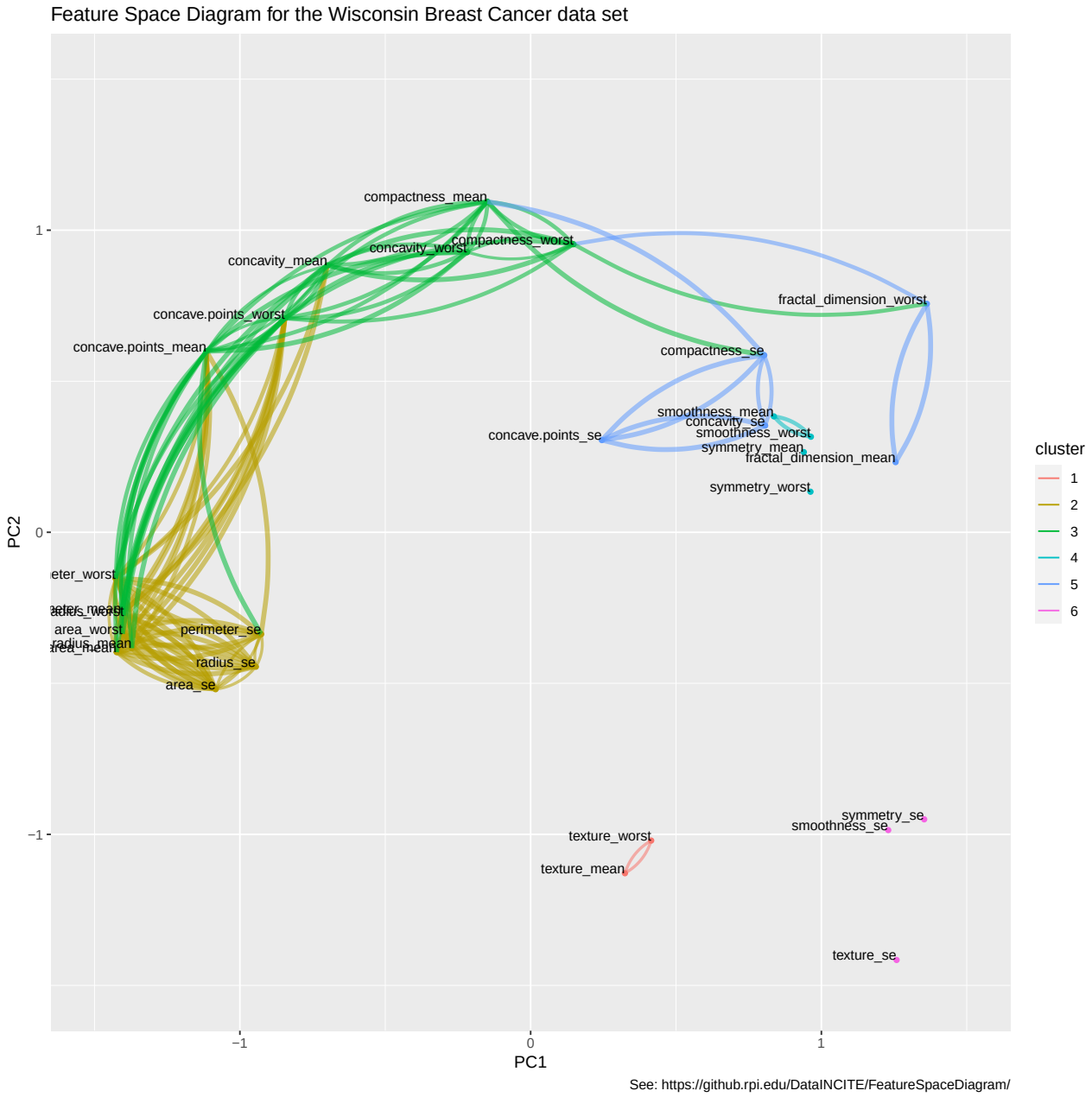
```
# Determine connectivity!
# filtering
selector <- ((abs(mydata.cor.ro.1) <= 0.3 ) * 1)

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 }

# Re-scale the line weights
```

```r
range <- c(1,10)
domain <- c(min(abs(mydata.cor.ro.1)[abs(mydata.cor.ro.1)>0]),max(abs(mydata.cor.ro.1)))
line.lm <- lm(range~domain)

adjust <- function(x){
  # stupid lm trick to scale
    line.lm$coefficients[[2]] * x + line.lm$coefficients[[1]]
  }

# Repeat selector matrix, this time for strength of relationships
thickness <- ((abs(mydata.cor.ro.1) <= 0.3 ) * 1)
for (i in 1:length(mydata.cor.ro.1.pca.plot$cluster)) {thickness[i,i] <- 0 } # remove diagonal
for (i in 1:length(mydata.cor.ro.1.pca.plot$cluster)) {
  for (j in 1:length(thickness[,i])) {
    if (thickness[j,i] != 0 ) {
      thickness[j,i] <- adjust(thickness[j,i] * abs(mydata.cor.ro.1[j,i]))
    } else { thickness[j,i] <- 0}
  }
}

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

# Gets us our edges!
mydata.edges <- get.data.frame(network) %>% # this is where "weight" is introduced
  rename(Cluster = weight)

mydata.edges.t <- get.data.frame(network.t) %>% # this is where "weight" is introduced
  rename(thickness = weight) %>%
  mutate(thickness = as.integer(thickness))


# 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.segments <- mydata.segments %>%
  rename(cluster = Cluster)

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

```
# NEW: cluster used for coloring
mydata.segments$cluster <- as.factor(mydata.segments$cluster)
mydata.segments$thickness <- as.integer(mydata.edges.t$thickness)
```

Building the Plot

```
# Adding line segments to PCA plot
p <- ggplot(mydata.cor.ro.1.pca.plot, aes(x=PC1, y=PC2)) +
  geom_point(aes(color=cluster, size=2), show.legend = FALSE) +
  xlim(-1.5,1.5) +
  ylim(-1.5,1.5) +
  geom_curve(aes(x=X1,y=Y1,xend=X2,yend=Y2,color=cluster,size=thickness, alpha=0.4), curvature=0.2, data
  guides(alpha = FALSE) +
  scale_size(range = c(0.1, 1.5), guide = guide_none()) +
#  labs(title="Feature Space Diagram for the Boston Housing data set")
  labs(title="Feature Space Diagram for the Wisconsin Breast Cancer data set") +
  labs(caption = "See: https://github.rpi.edu/DataINCITE/FeatureSpaceDiagram/") +
  geom_text(aes(label=name),hjust=1, vjust=0, size=3)


p
```




```
ggsave("WDBC_fsd.png", p)
```

```
## Saving 10 x 10 in image
```