

k-means with Imputation

ClustImpute package

PMS

16 May, 2023

Preliminary

Loading & Cleaning Data

```
set.seed(2023)
library(cluster)
library(ClustImpute)
library(ggplot2)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(clusterCrit)
load('../.../.../local_data/codes/create_master/master_pms_df.Rdata')
```

```
#removing rows with NA for all indices, as well as for population = 0
master$PMS_DBPOP = as.numeric(as.character(master$PMS_DBPOP))
```

```
## Warning: NAs introduced by coercion
```

```
master = master[master$PMS_DBPOP != 0,]
master = master[!is.na(master$PMS_DBPOP),]
idx = c("PMS_prox_idx_emp", "PMS_prox_idx_pharma", "PMS_prox_idx_childcare", "PMS_prox_idx_health", "PMS_prox_idx_education")
master = master[(rowSums(is.na(master[,idx])) < 10),]
nrow(master)
```

```
## [1] 341425
```

Assumptions of the Algorithm

[This algorithm](#) “draws the missing values iteratively based on the current cluster assignment so that correlations are considered on this level”. Also, “penalizing weights are imposed on imputed values and successively decreased (to zero) as the missing data imputation gets better”. The idea is that the missing value is imputed by those other observations that are more similar to it (ie. in the same cluster).

Algorithm steps:

1. It replaces all NAs by random imputation, i.e., for each variable with missings, it draws from the marginal distribution of this variable not taking into account any correlations with other variables
2. Weights < 1 are used to adjust the scale of an observation that was generated in step 1. The weights are calculated by a (linear) weight function that starts near zero and converges to 1 at `n_end`.

3. A k-means clustering is performed with a number of `c_steps` steps starting with a random initialization.
4. The values from step 2 are replaced by new draws conditionally on the assigned cluster from step 3.
5. Steps 2-4 are repeated `nr_iter` times in total. The k-means clustering in step 3 uses the previous cluster centroids for initialization.
6. After the last draws a final k-means clustering is performed.

All Metrics Together

Implementation

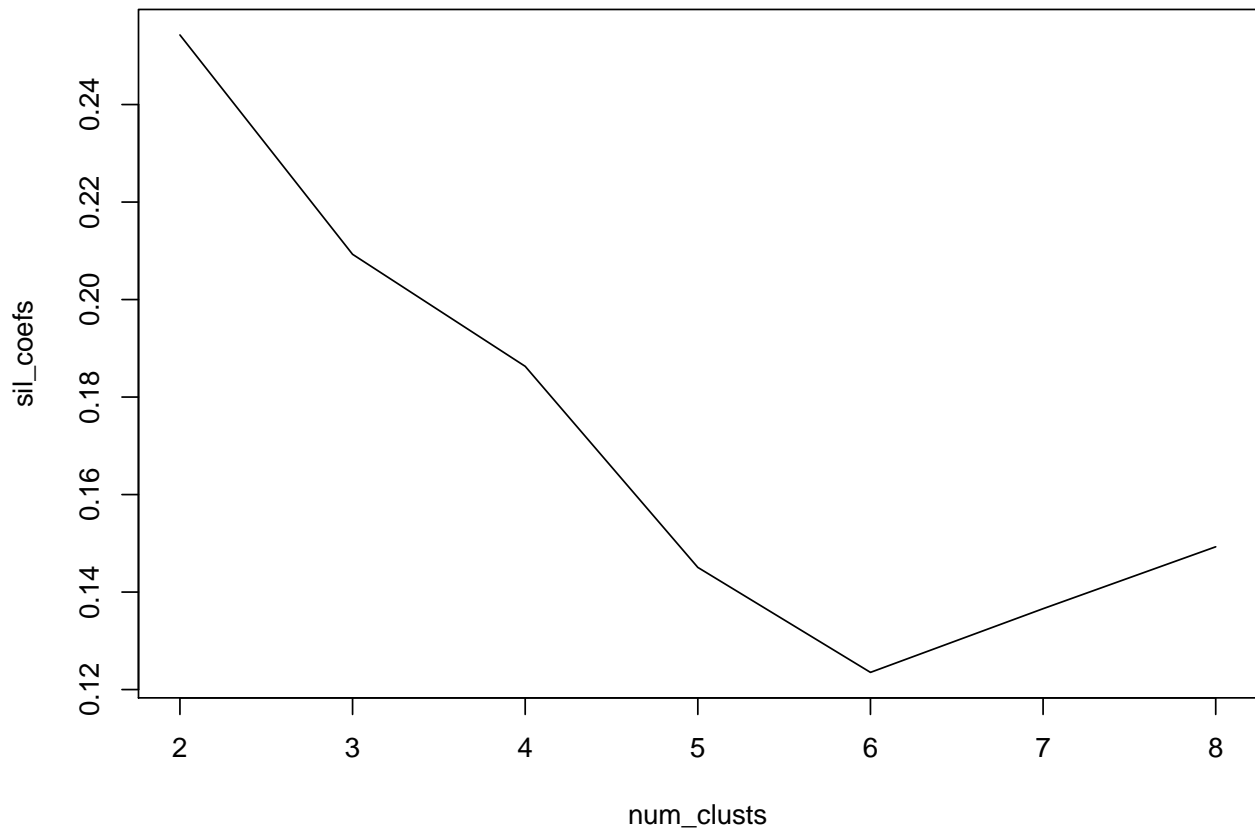
(with 5% subsampling)

```
#cluster data
subsample = nrow(master)/20 #subsampling
subsam = master[sample(nrow(master), subsample), idx]
sum(is.na(subsam))

## [1] 78804

#algorithm
sil_coefs = c()
counter = 1
num_clusts = 2:8
for (i in num_clusts){
  nr_iter = 10 # iterations of procedure
  n_end = 10 # step until convergence of weight function to 1
  #nr_cluster = 3 # number of clusters
  c_steps = 50 # number of cluster steps per iteration
  res = ClustImpute(subsam,nr_cluster=i, nr_iter=nr_iter, c_steps=c_steps, n_end=n_end)
  sil_coefs[counter] = intCriteria(as.matrix(res$complete_data),res$clusters, 'Silhouette')$silhouette
  counter = counter + 1
}

#plot silhouette coefficients
plot(sil_coefs~num_clusts, type = 'l')
```

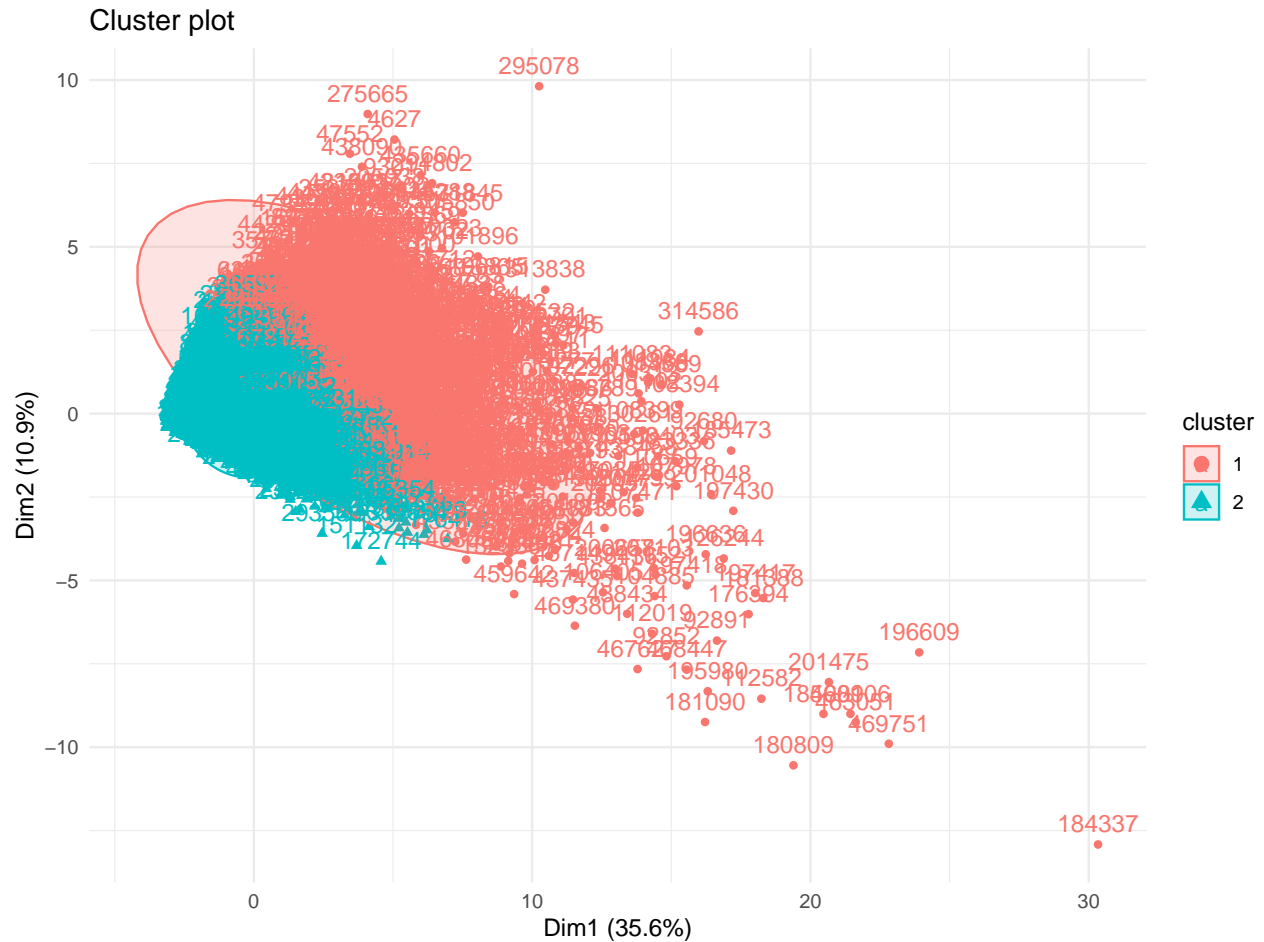


```

#re-run algorithm with highest sil
res = ClustImpute(subsam,nr_cluster=num_clusts[which(sil_coefs == max(sil_coefs))], nr_iter=nr_iter, c_

#plot
# ggplot(res$complete_data,aes(prox_idx_emp,prox_idx_pharma,color=factor(res$clusters))) + geom_point()
pass = list(data = res$complete_data, cluster = res$clusters)
fviz_cluster(pass, ellipse.type = "norm") + theme_minimal()

```



Cut-off Values

```

#####
# OLD METHOD
#####

cutoffs = list()
for (k in idx){
  clus_medians = c()
  counter = 1
  for (i in unique(res$clusters)){
    clus_medians[counter] = median(res$complete_data[res$clusters == i,k])
    counter = counter + 1
  }
}

```

```

cutoff = c()
clus_medians = sort(clus_medians)
for (j in 1:(length(clus_medians)-1)){
  cutoff[j] = (clus_medians[j] + clus_medians[j+1])/2
}
cutoffs[[k]] = cutoff
print(k)
print(round(cutoff, 5))
}

```

```

## [1] "PMS_prox_idx_emp"
## [1] 0.04265
## [1] "PMS_prox_idx_pharma"
## [1] 0.04085
## [1] "PMS_prox_idx_childcare"
## [1] 0.09785
## [1] "PMS_prox_idx_health"
## [1] 0.0135
## [1] "PMS_prox_idx_grocery"
## [1] 0.0637
## [1] "PMS_prox_idx_educpri"
## [1] 0.15668
## [1] "PMS_prox_idx_educsec"
## [1] 0.1164
## [1] "PMS_prox_idx_lib"
## [1] 0.0944
## [1] "PMS_prox_idx_parks"
## [1] 0.0672
## [1] "PMS_prox_idx_transit"
## [1] 0.0209

```

```
#plot em
```

```

library(ggplot2)
library(tidyverse)

```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
```

```

## v tibble 3.1.8      v dplyr 1.1.0
## v tidyr 1.3.0       v stringr 1.5.0
## v readr 2.1.2       v forcats 0.5.2
## v purrr 1.0.1

```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```

## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

```

```
library(stringr)
```

```
labs = str_sub(idx, 14) #labels
```

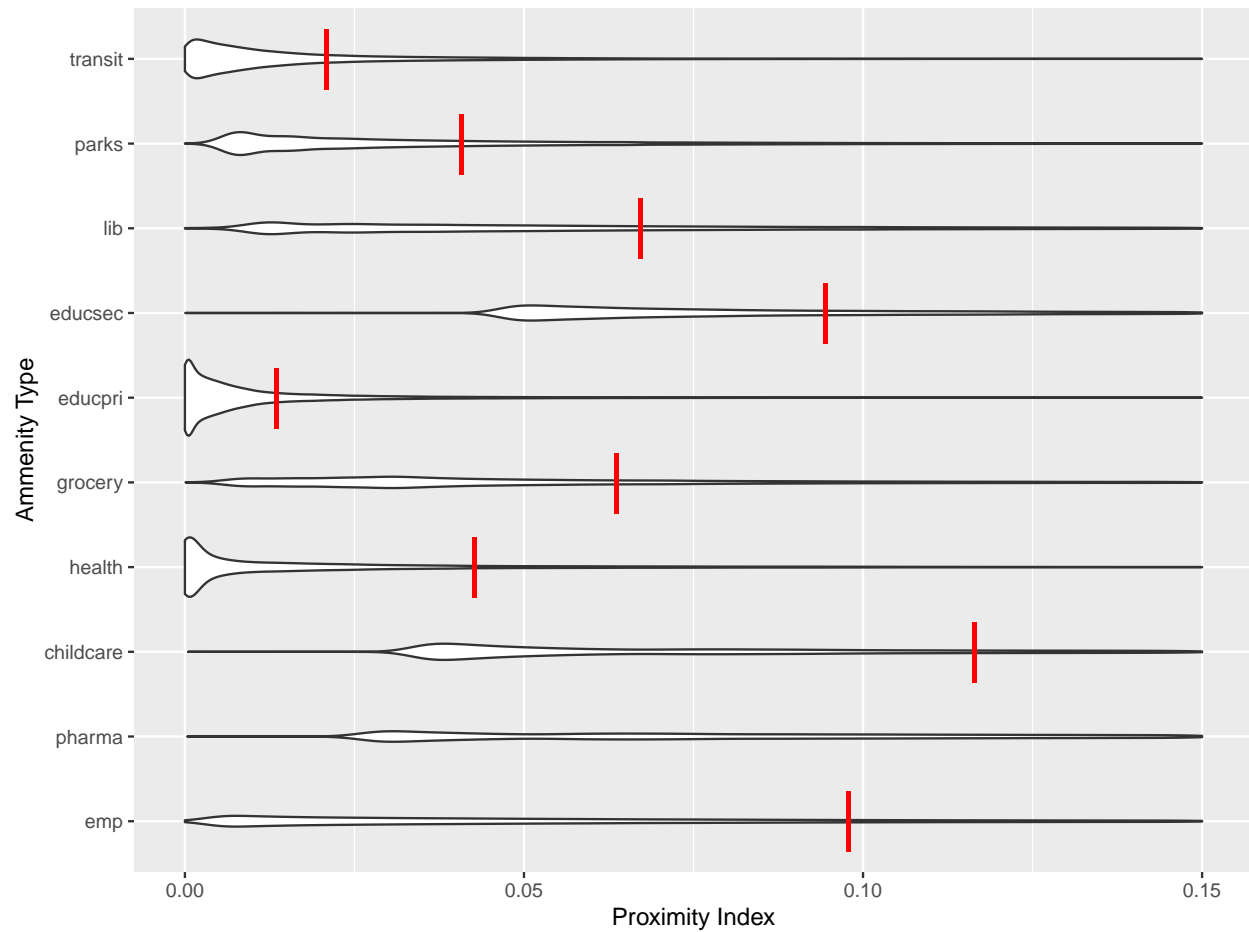
```
hline = pivot_longer(as.data.frame(cutoffs), all_of(idx)) #cutoff lines
```

```
df_long = pivot_longer(master[,idx], all_of(idx))
```

```
ggplot(df_long, aes(x=value, y=name)) + geom_violin() + scale_y_discrete(labels=labs) + scale_x_continuous
```

```
## Warning: Removed 1738820 rows containing non-finite values (`stat_ydensity()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



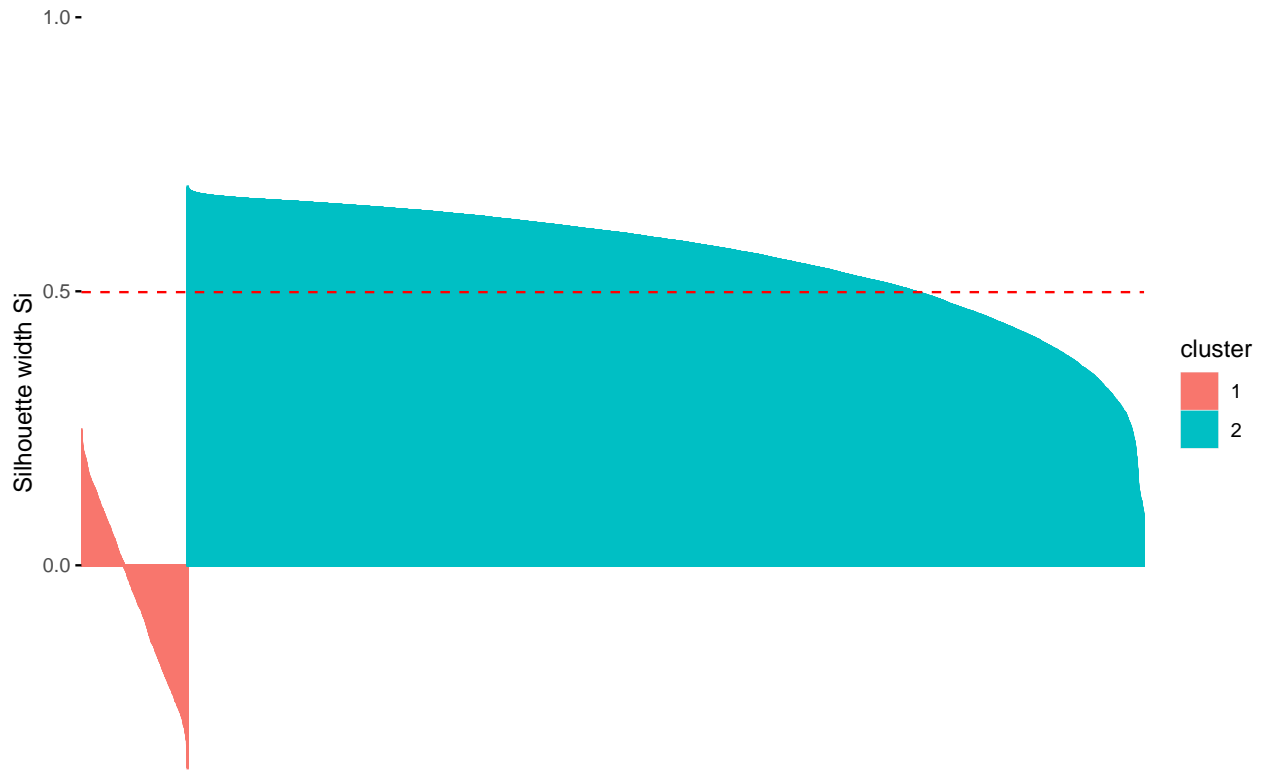
Silhouette Plot

```
# plt = cluster::silhouette(res$clusters, dist(res$complete_data))
# plot(plt, col = 1:4)
# abline(v=mean(plt[,3]), col="red", lty=2)
```

```
sil = silhouette(res$clusters, dist(res$complete_data))
fviz_silhouette(sil)
```

```
##   cluster  size ave.sil.width
## 1      1  1708      -0.05
## 2      2 15363       0.56
```

Clusters silhouette plot
Average silhouette width: 0.5



Cluster Profiles

```
for (k in sort(unique(res$clusters))){
  temp = master[res$clusters == k,]
  print(paste('Cluster #', k))
  print(paste('Num of DBs in cluster: ', as.character(nrow(temp))))
  print('CSD Type:')
  print(table(temp$CSDTYPE)) #replace with grouped type later
  cat('\n DB Population: \n')
  print(summary(temp$PMS_DBPOP))
  cat('\n Index of Remoteness: \n')
  print(summary(temp$IOR_Index_of_remoteness))
  cat('\n Provinces: \n')
  print(table(temp$PROVINCE))
  cat('\n Amenity dense: \n')
  print(table(temp$PMS_amenity_dense))
  cat('\n\n\n ')
}
```

```
## [1] "Cluster # 1"
## [1] "Num of DBs in cluster:  34160"
## [1] "CSD Type:"
##
##      C   CG   COM   CT   CU   CV   CY   DM   HAM   ID   IGD   IM   IRI   LGD   LOT   M
## 1825    4    22   111    12  819 9294   599   16    1    1    3  244    1   170   41
```

```

## MD ME MU NL NO NV P PE RCR RDA RGM RM RV S-É SA SC
## 1447 1905 1435 2 71 6 403 238 35 707 481 1063 29 5 10 348
## SET SM SNO SV T TC TP TV V VL VN
## 1 159 64 19 3855 12 1747 199 6180 571 5
##
## DB Population:
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 5.00 23.00 53.00 92.62 109.00 997.00
##
## Index of Remoteness:
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
## 0.00000 0.09688 0.14941 0.19267 0.28219 0.97797 229
##
## Provinces:
##
## Alberta BritishColumbia NewBrunswick
## 1058 1926 335
## NorthwestTerritories NovaScotia Ontario
## 21 1177 6627
## Quebec Saskatchewan
## 2212 211
##
## Amenity dense:
##
## 0 1 2 F
## 30559 3192 409 0
##
##
## [1] "Cluster # 2"
## [1] "Num of DBs in cluster: 307265"
## [1] "CSD Type:"
##
## C CG CN COM CT CU CV CY DM HAM ID IGD IM
## 16193 18 1 203 875 54 7674 82727 5572 106 14 18 38
## IRI LGD LOT M MD ME MU NH NL NO NV P PE
## 2505 30 1416 426 13878 16696 12955 1 27 636 92 3655 2511
## RCR RDA RGM RM RV S-É SA SC SÉ SET SG SM SNO
## 332 6355 4242 9317 169 20 75 3022 10 5 1 1379 577
## SV T TC TK TP TV V VL VN
## 170 34652 108 1 15573 1696 55860 5343 37
##
## DB Population:
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 5.00 23.00 53.00 93.11 109.00 999.00
##
## Index of Remoteness:
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
## 0.0000 0.0969 0.1499 0.1930 0.2823 0.9804 2120
##
## Provinces:
##
## Alberta BritishColumbia NewBrunswick
## 9481 16994 2918

```



```

## NorthwestTerritories      NovaScotia      Ontario
##              146              10454      59906
##              Quebec      Saskatchewan
##              20112              1835
##
## Amenity dense:
##
##      0      1      2      F
## 274728 28722 3815      0
##
##
##
##

```

Conclusion

text

Linked with Index of Remoteness

Implementation

#

Cut-off Values

#

Silhouette Plot

#

Cluster Profiles

#

Conclusion

text