Practical 414

Stefano De Sabbata

2020-12-07

Unsupervised machine learning

Stefano De Sabbata

Introduction

The field of **machine learning** sits at the intersection of computer science and statistics, and it is a core component of data science. According to Mitchell (1997), *"the field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience."*

Machine learning approaches are divided into two main types.

- **Supervised**:
  - training of a *"predictive"* model from data;
  - one (or more) attribute of the dataset is used to "predict" another attribute.
- **Unsupervised**:
  - discovery of *descriptive* patterns in data;
  - commonly used in data mining.

Clustering is a classic unsupervised machine learning task, which aims to *"automatically divides the data into* ***clusters*** *, or groups of similar items"*(Lantz, 2019). In computer science, a wide range of approaches has been developed to tackle clustering. Among those approaches, the most commons are centroid-based approaches (such as k-means) and hierarchical approaches. Other approaches include density based clustering methods (such as DBSCAN) and midex approaches (such as bagged clustering), which combine different aspects of centroid-based and hierarchical approaches.

K-means

The k-mean approach clusters n observations (x) in k clusters (c) by minimising the within-cluster sum of squares (WCSS) through an iterative process. That is, the algorithm calculates the distance between each observation (i.e., each case, object, row in the table) and the centroid of its cluster. The square values of those distances are summed up for each cluster, and then for the whole dataset. The aim of the algorithm is minimase that value.

$$WCSS = \sum_{c=1}^{k} \sum_{x \in c} (x - \overline{x}_c)^2$$

To minimise WCSS, while trying to identify `k` clusters, k-mean first randomly select `k` observations as initial centroids. Then, k-means repeats the two steps below. Every time k-means repeats those two steps the new centroids will be closer to the two actual center. The process continues until centroids don't change anymore (within a certain margin of error) or until it has reached a maximum number of iterations set by the analyst.

- **assignment step**: observations assigned to closest centroids
- **update step**: calculate means for each cluster, as new centroid

Geodemographic Classification

In GIScience, clustering approaches are commonly used to create *geodemographic classifications*. For instance, Gale *et al.*, 2016 created the 2011 Output Area Classification (2011 OAC) starting from an initial set of 167 prospective variablesfrom the UK Census 2011.

In the process of creating the classification, 86 variables were removed from the initial set, including highly correlated variables that don't bring additional informatio to the classification process. Furthermore, 41 variable were retained as they were, whereas 40 were combined, to create final set of 60 variables. The k-mean approach was then applied to cluster the census Output Areas (OAs) into 8 supergroups, 26 groups and 76 subgroups.

The paper provides a detail report of the process. In particular, it is interesting to see how the authors applied a process of variable selection involving repeated clustering while excluding one variable, to see how the within sum of square measure (WCSS) would be affected. Variable that produced significantly higher WCSS when excluded were considered for exclusion from the final analysis, in order to increase the homogeneity of the clusters.

Once the clustering is completed, the final step in geodemographic classification is the interpretation of the resulting cluster, which is commonly done by observing theaverage values of the variables for each cluster.

Examples

Clustering two age groups

```
leicester_2011OAC_ages <- readr::read_csv("2011_OAC_Raw_uVariables_Leicester.csv")
```

```
# u086  Dwellings, Household Spaces and Accomodation Type   Count   Household_Spaces    Housing Type
# u087  Dwellings, Household Spaces and Accomodation Type   Count   Household_Spaces    Housing Type
# u088  Dwellings, Household Spaces and Accomodation Type   Count   Household_Spaces    Housing Type
# u089  Dwellings, Household Spaces and Accomodation Type   Count   Household_Spaces    Housing Type

leicester_dwellings <-
  leicester_2011OAC %>%
  dplyr::select(
    OA11CD,
    u086:u089
  ) %>%
  # scale across
  dplyr::mutate(
    dplyr::across(
      u086:u089,
      scale
    )
```

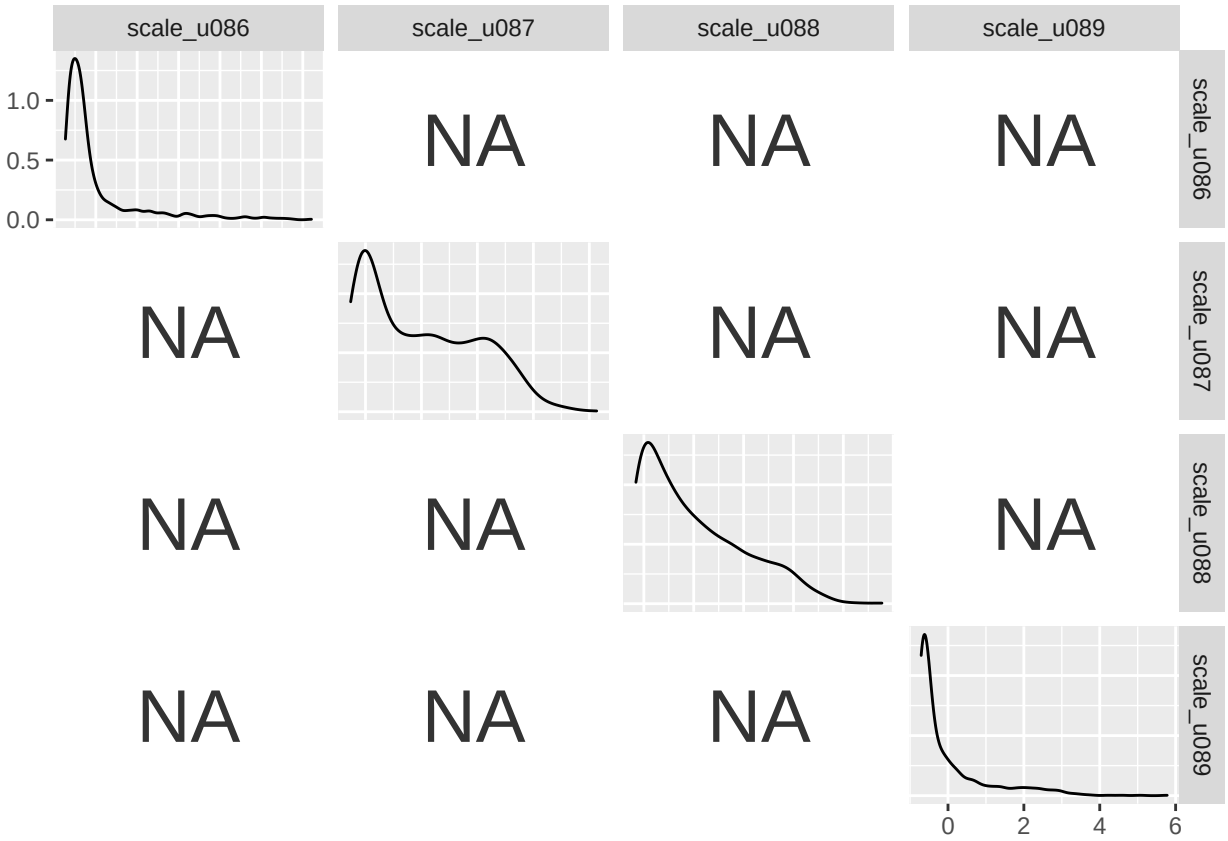
```
  ) %>%
  # rename columns
  dplyr::rename_with(
    function(x){ paste0("scale_", x) },
    u086:u089
  )
```

```
# install.packages("GGally")
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.0.2
```

```
leicester_dwellings %>%
  dplyr::select(scale_u086:scale_u089) %>%
  GGally::ggpairs()
```

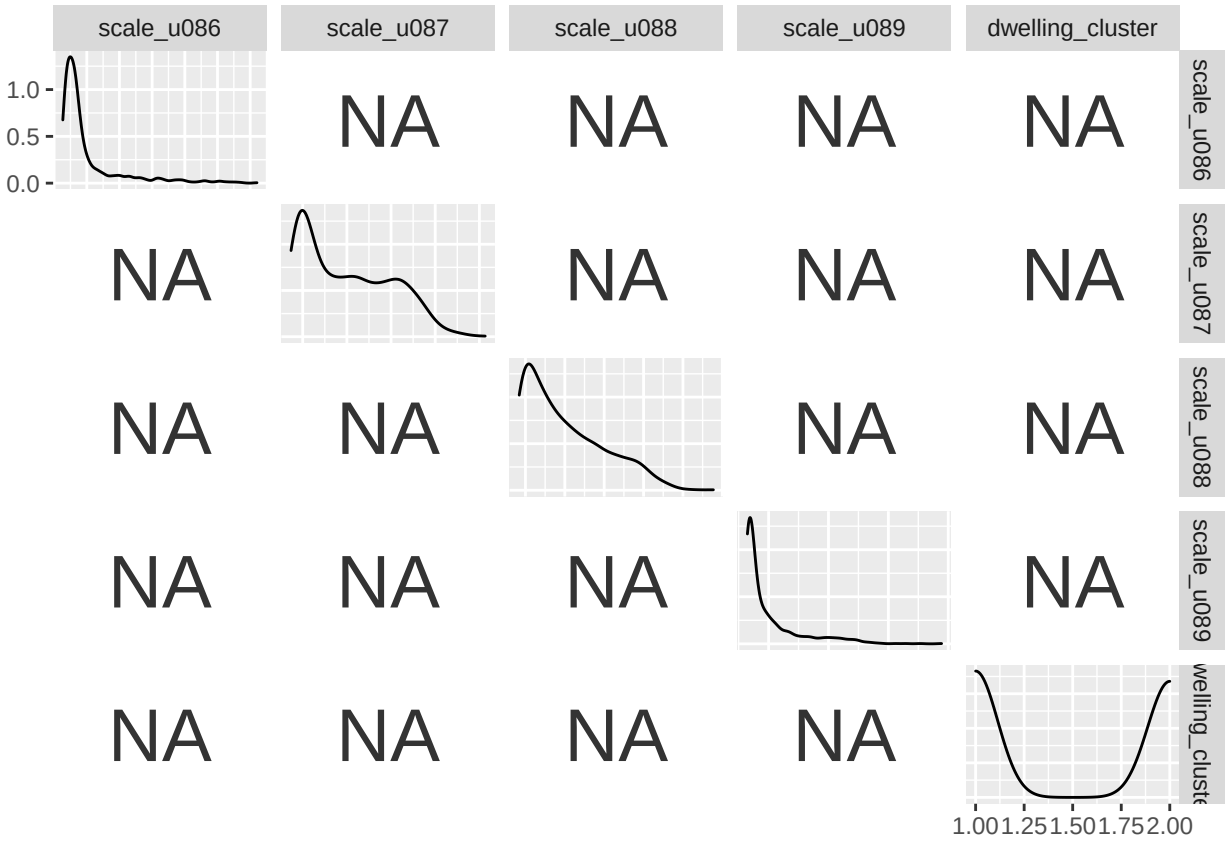
```
## plot: [1,1] [======>-------------------------------------------------------------------------------------
## plot: [1,2] [============>-------------------------------------------------------------------------------
## plot: [1,3] [====================>-------------------------------------------------------------------------
## plot: [1,4] [==========================>-------------------------------------------------------------------
## plot: [2,1] [================================>-------------------------------------------------------------
## plot: [2,2] [========================================>-------------------------------------------------------
## plot: [2,3] [================================================>-------------------------------------------------
## plot: [2,4] [============================================================>-------------------------------
## plot: [3,1] [========================================================================>-----------------------
## plot: [3,2] [==============================================================================>---------------
## plot: [3,3] [======================================================================================>---------
## plot: [3,4] [==============================================================================================>---
## plot: [4,1] [=================================================================================================
## plot: [4,2] [=================================================================================================
## plot: [4,3] [=================================================================================================
## plot: [4,4] [=================================================================================================
```




```
dwelling_kmeans <- leicester_dwellings %>%
  dplyr::select(scale_u086:scale_u089) %>%
  stats::kmeans(centers=2, iter.max=50)

leicester_dwellings <-
  leicester_dwellings %>%
  tibble::add_column(
    dwelling_cluster = dwelling_kmeans %$% cluster
  )
```

```
leicester_dwellings %>%
  dplyr::select(scale_u086:scale_u089, dwelling_cluster) %>%
  GGally::ggpairs()
```


scale_u086
scale_u087
scale_u088
scale_u089
dwelling_cluster
1.0
0.5
0.0
NA
1.00
1.25
1.50
1.75
2.00