Sushrut Shendre Follow

Apr 29, 2020 · 7 min read ★ · Listen

Clustering datasets having both numerical and categorical variables

With the advent of machine learning in the modern era, businesses have seen a transformation in the way they make decisions and drive profits. One of the most important attributes of a successful firm is how well they can understand their customers and how well they can cater to their personalized needs. It has become increasingly important for firms to understand that the ‘one size fits all’ model doesn’t work anymore.

This brings us to the topic of clustering. Clustering is nothing but segmentation

categorical variables is often the case in real-life problems. This article discusses a clustering approach using Gower distance, the PAM (Partitioning Around Medoids) method, and silhouette width and explains each of the steps with an implementation in R.

We are using the '**germancredit**' data taken from UCI's Machine Learning Repository: (https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/). To import the data in R, we use the '**read.table**' command as follows:

```
set.seed(2020)
german_credit = read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/german.data")
```

We observe that the column names of this data set are not self-explanatory, and therefore we rename the columns with the help of the documentation.

```
colnames(german_credit) = c("chk_acct", "duration", "credit_his",
"purpose", "amount", "saving_acct", "present_emp", "installment_rate",
```



Further, to make things easier, we select only a few variables that are relatively more important. Please note that I have selected these particular columns only for illustration purposes, and readers are free to experiment with a different set of columns.

```
library(dplyr)
german_credit_clean = german_credit %>% select(duration, amount,
installment_rate, present_resid, age, n_credits, n_people, chk_acct,
credit_his, sex, property, housing, present_emp)
```

We see that this data set contains both numerical (continuous) as well as categorical variables.

Continuous Variables:

· duration: duration in months
· amount: the credit amount of the loan
· installment_rate: Installment rate in percentage of disposable income
· present_resid: time since living in the present residence
· age: Age in years

· credit_his: Credit history
· property: Property
· housing: Housing
· present_emp: Present employment since

Categorical variables:

· chk_account: status of an existing checking account
· sex: Personal status and sex
· credit_his: Credit history
· property: Property
· housing: Housing
· present_emp: Present employment since

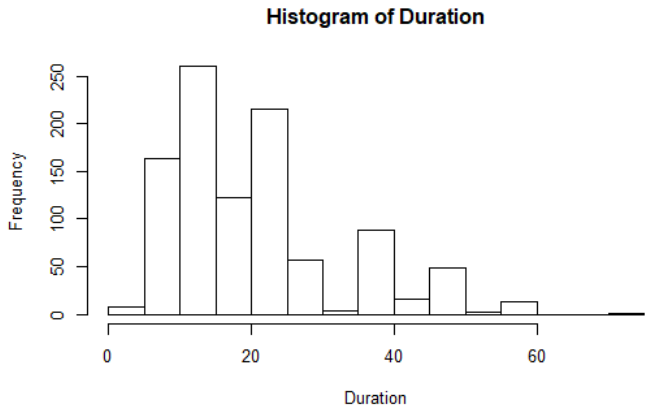
Now that we are done with understanding and preprocessing the data, we turn our heads towards clustering. The basic premise of clustering is the similarity/dissimilarity between the data points. As we know, the K-means algorithm iterates over and over until it attains a state wherein all points of a cluster are similar to each other, and points belonging to different clusters are dissimilar to each other. This similarity/dissimilarity is defined by the distance

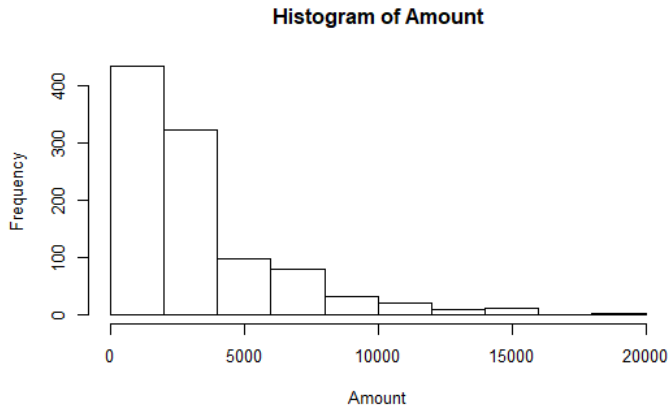
A popular choice for the metric to measure this distance is the Euclidean distance. Some researchers also advocate the use of Manhattan distance for particular types of problems. However, both of these distance metrics are applicable only for continuous data. In our data which contains mixed data types, Euclidean and Manhattan distances are not applicable and therefore, algorithms such as K-means and hierarchical clustering would fail to work.

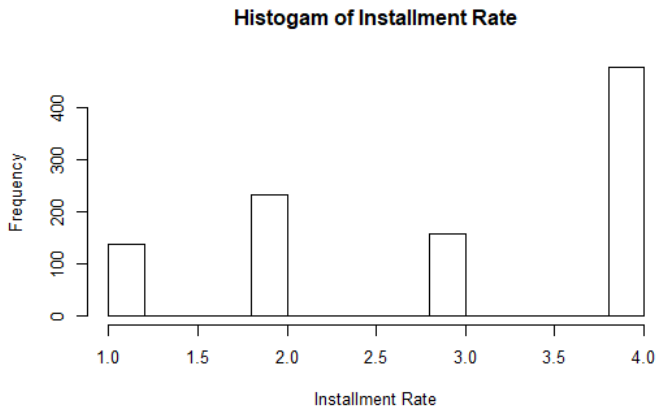
Therefore, we use the Gower distance which is a metric that can be used to calculate the distance between two entities whose attributes are a mix of categorical and quantitative values. This distance is scaled in a numerical range of 0 (identical) and 1 (maximally dissimilar). The details about the mathematics of Gower distance are quite complicated and left out for another article.

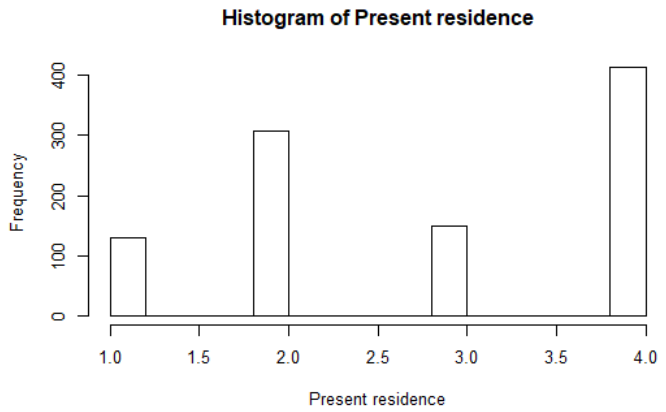
In R, the Gower distance can be calculated using the '**daisy**' package. Before we use the '**daisy**' function, we observe the distribution of the numerical variables to understand if any of them need transformations. We find that the variable *amount* needs a log transformation due to the positive skew in its distribution.

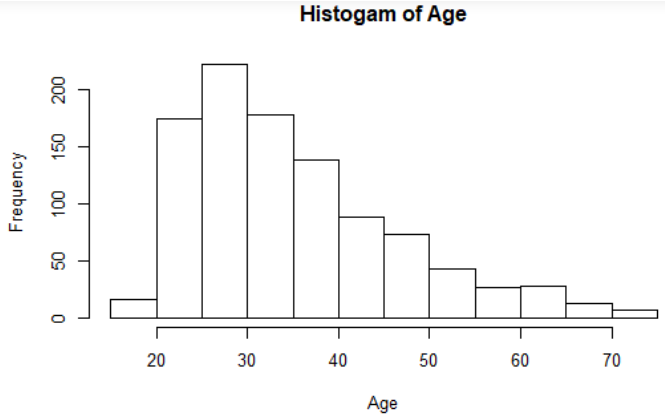


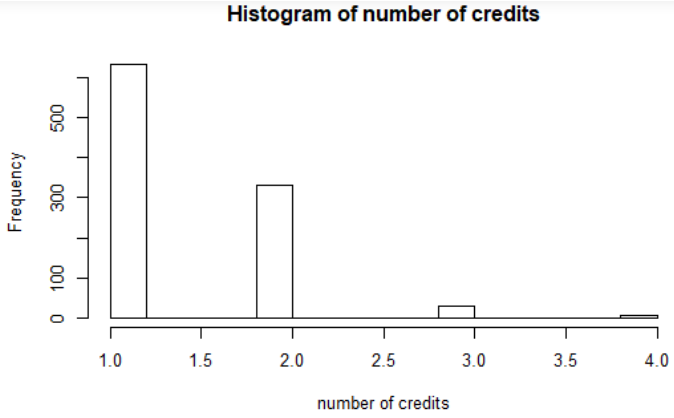


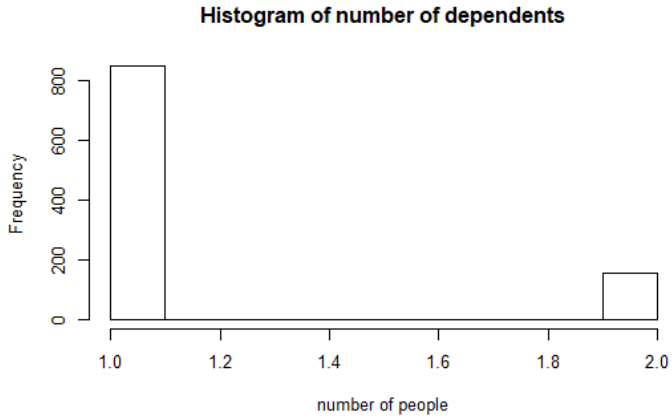












The following code shows the implementation of the Gower distance calculation. Note that we have specified '2' as the parameter in the *type* argument. This ensures that the second variable, *amount* undergoes a log transformation before the Gower calculations are performed.

```
metric = "gower" ,
type = list(logratio = 2))
```

Next, we check the summary of the **'gower_df'** data frame. The type 'I' and 'N' tell us whether the respective columns are interval-scaled (numerical), or nominal.

```
> summary(gower_df)
499500 dissimilarities, summarized :
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.001062 0.361480 0.441360 0.439330 0.519650 0.855590
Metric :  mixed ;  Types = I, I, I, I, I, I, I, N, N, N, N, N, N
Number of objects : 1000
```

K-medoids and Partitioning Around Medoids

Now that we have a distance matrix in place, we need to choose a clustering algorithm to infer similarities/dissimilarities from these distances. Just like K-means and hierarchical algorithms go hand-in-hand with Euclidean distance, the Partitioning Around Medoids (PAM) algorithm goes along with the Gower distance. PAM is an iterative clustering procedure just like the K-means, but with some slight differences. Instead of centroids in K-means clustering, PAM iterates over and over until the **medoids** don't change their positions. The medoid of a

Silhouette Width to select the optimal number of clusters

The silhouette width is one of the very popular choices when it comes to selecting the optimal number of clusters. It measures the similarity of each point to its cluster, and compares that to the similarity of the point with the closest neighboring cluster. This metric ranges between -1 to 1, where a higher value implies better similarity of the points to their clusters. Therefore, a higher value of the Silhouette Width is desirable. We calculate this metric for a range of cluster numbers and find where it is maximized. The following code shows the implementation in R:

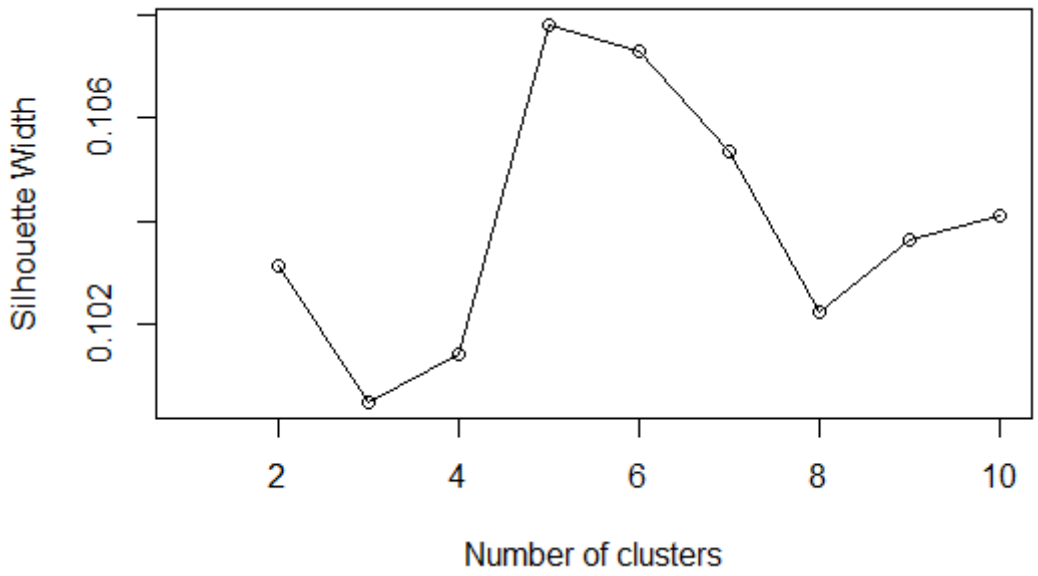
```
silhouette <- c()
silhouette = c(silhouette, NA)

for(i in 2:10){
  pam_clusters = pam(as.matrix(gower_df),
                 diss = TRUE,
                 k = i)
  silhouette = c(silhouette ,pam_clusters$silinfo$avg.width)
}

plot(1:10, silhouette,
     xlab = "Clusters",
```



The silhouette width is plotted versus the number of clusters, as shown:



As can be seen, the value is maximized at 5. Hence, we can conclude that clustering the data into 5 clusters gives us the best segmentation possible. Having said that, we construct a PAM model with 5 clusters, and try to interpret the behavior of these clusters with the help of the medoids.

duration	amount	installment_rate	present_resid	age	n_credits	n_people	chk_acct	credit_his	sex	property	housing	present_emp
15	1829	4	4	46	2	1	A14	A34	A93	A123	A152	A75
15	2631	3	2	25	1	1	A12	A32	A92	A121	A152	A73
36	8335	3	4	47	1	1	A11	A32	A93	A124	A153	A75
15	4657	3	2	30	1	1	A14	A32	A93	A123	A152	A73
21	2606	4	4	28	1	1	A11	A32	A92	A122	A151	A72

The figure above shows the medoids table, where each row represents a cluster. Using this table, we can infer that customers belonging to Cluster 1 have the following characteristics: the duration is 15 months, the credit amount is 1829$, the installment rate is 4, they have been living in the present residence since 4 months, their average age is 46 years, they have 2 loans pending, they have 1 dependent, they have no checking account (A14), their credit history is critical (A34), they are male singles (A93), they have a car as their property (A123), live in their own housing (A152), and have been employed for more or equal to 7 years (A75). Please note that not all customers will be exactly like this; and that the medoids are only a representation of the median values.

Similar interpretations can be made for the other clusters. To dig deeper into the characteristics of each cluster, we find the summary stats. The R code for the same is as shown below, along with the statistics for the first cluster.

```
pam_summary <- german_credit_clean %>%
```

```
   duration        amount      installment_rate present_resid       age          n_credits
Min.   : 4.00   Min.   :  338   Min.   :1.000    Min.   :1.000   Min.   :20.00   Min.   :1.000
1st Qu.:12.00   1st Qu.: 1257   1st Qu.:3.000    1st Qu.:3.000   1st Qu.:33.00   1st Qu.:1.000
Median :18.00   Median : 1948   Median :4.000    Median :4.000   Median :39.00   Median :2.000
Mean   :19.11   Mean   : 2776   Mean   :3.366    Mean   :3.297   Mean   :40.56   Mean   :1.801
3rd Qu.:24.00   3rd Qu.: 3522   3rd Qu.:4.000    3rd Qu.:4.000   3rd Qu.:47.00   3rd Qu.:2.000
Max.   :60.00   Max.   :15653   Max.   :4.000    Max.   :4.000   Max.   :68.00   Max.   :4.000
   n_people     chk_acct  credit_his  sex      property   housing    present_emp    cluster
Min.   :1.000   A11: 37   A30:  7    A91: 11   A121: 61   A151: 22   A71: 16     Min.   :1
1st Qu.:1.000   A12: 58   A31: 10    A92: 31   A122: 57   A152:216   A72: 22     1st Qu.:1
Median :1.000   A13: 12   A32: 29    A93:191   A123:107   A153:  8   A73: 21     Median :1
Mean   :1.215   A14:139   A33: 23    A94: 13   A124: 21              A74: 50     Mean   :1
3rd Qu.:1.000             A34:177                                    A75:137     3rd Qu.:1
Max.   :2.000                                                                    Max.   :1
```

Visualization

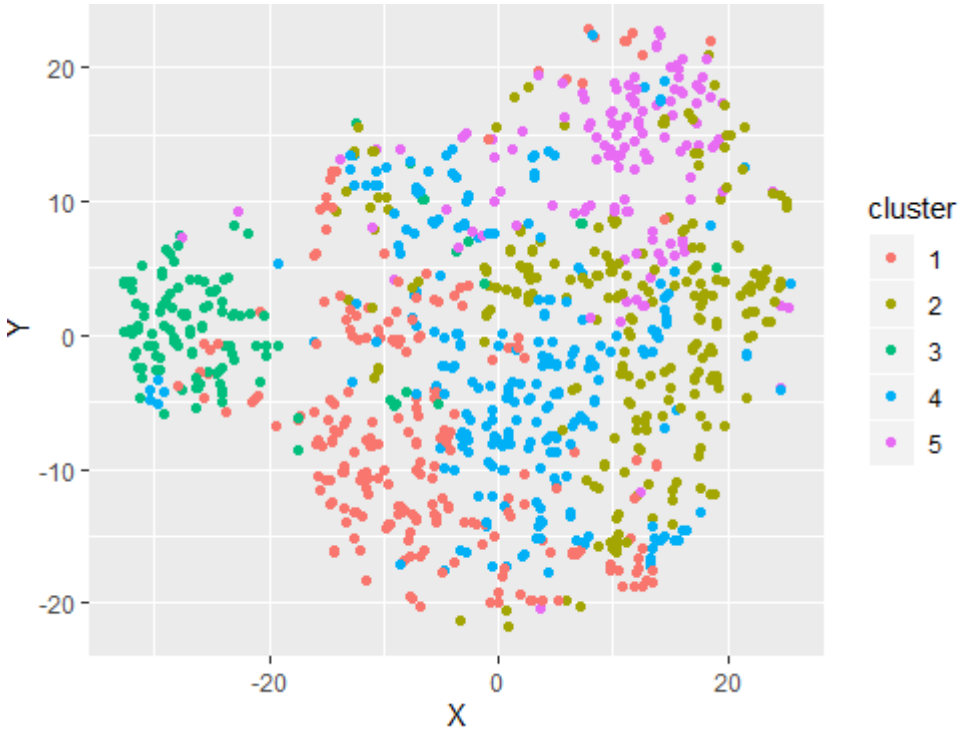
Finally, let us wrap this article up with some fancy visualizations. For this, we use the t-SNE or the t-Distributed Stochastic Neighbor Embedding technique. This technique provides a great way to visualize a multi-dimensional data set such as the one we are working on.

```
library(Rtsne)
library(ggplot2)

tsne_object <- Rtsne(gower_df, is_distance = TRUE)
```

```
  setNames(c("X", "Y")) %>%
  mutate(cluster = factor(pam_german$clustering))

ggplot(aes(x = X, y = Y), data = tsne_df) +
  geom_point(aes(color = cluster))
```



As we can see, t-SNE has helped us visualize multi-dimensional data into a

This brings us to the end of this article. Summarizing, we discussed the concepts of Gower distance, PAM clustering, and Silhouette width to cluster data having both numerical and categorical features. Hope this helps all the data people around here who come across such data sets.

Happy Clustering!