Optimal K for clustering

Rufeng Ma
August 10, 2020, (3 min read)

Introduction

Clustering is an important part of the Natural Language Processing (NLP). As the name suggests, clustering helps us to group similar data together by calculating the distance between points. There are two types of traditional clustering are predominantly used, they are

- K-means clustering
- Hierarchical clustering

In our COVID-Twitter project, we use the K-means method to determine top-level clusters and sub-clusters. K-means look for a fixed number of clusters in a dataset by identifying ‘K’ numbers of centroids. THen allocates every data point to the nearest cluster. The ‘means’ refers to averaging of the data, Figure 1.

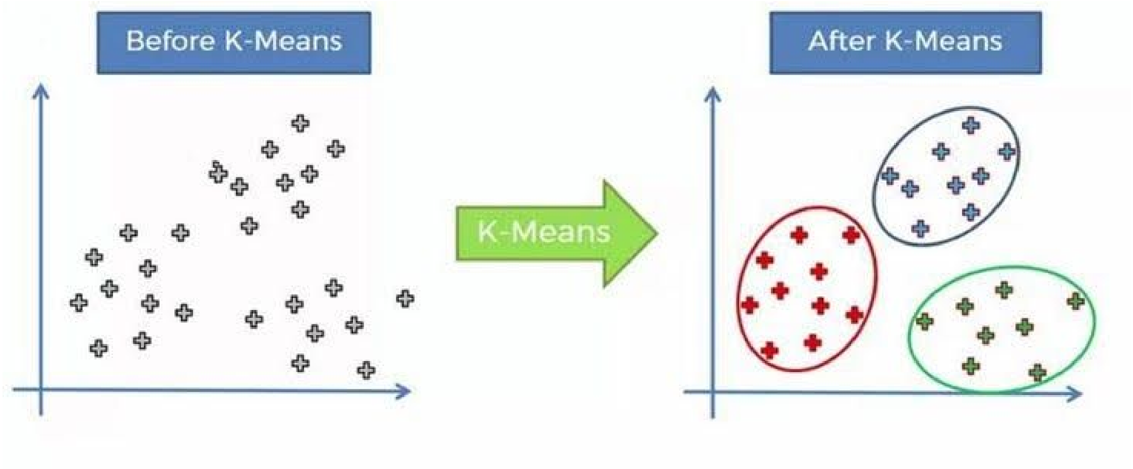


Figure 1. Example with 3 centroids, K=3 [1]

However, here is an obvious question:

How do we determine the feasible number of clusters?

Answering this question is critical. Correctly choice of K is often ambiguous, if we increase the K, the error in the resulting clustering will be lower. The extreme case is considering each data point as a cluster, then the error will be zero. But if the K is too small, the clustering could not give us too much useful information. In our practice, we summarized the theme of the big cluster. We found the summarization of those big clusters is too general, so they are not too valuable for further study. In summary, we must consider and balance the following aspects, when we are choosing the optimal K:

- Low error (prefer **big K**)
- Data compression, or computation efficiency (prefer **small K**)
- Meaningful for sentiment study (prefer **proper K**, not too general, and not too specific)

Elbow method and Silhouette coefficient

In the previous version clustering for the COVID-Twitter project, we used the elbow method. This is the most common method to determine K. The objective function is the relation between the average intra-cluster distance and the average inner-cluster distance. This method is rapid and accurate. But it needs manually choosing the K after the elbow plot is done, Figure 2.

More importantly, if we would like to use an interactive R notebook to generate the clusters with frequently updated twitter data, we would like to have an automatic method to choose the best K. Then the new method

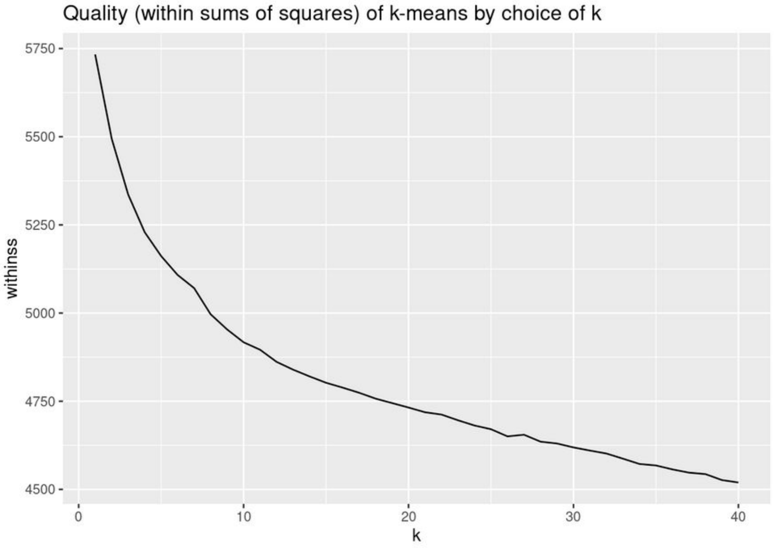


Figure 2. K-means withiness plot for choosing the best K manually.

Then we are considering changing the elbow method to the silhouette coefficient method. This method is also calculating the goodness of a clustering technique. The function is:

$$Silhouette\ Score\ =\ (b - a)/max(a, b)$$

Here ***a*** is the average intra-cluster distance, and ***b*** is the average inter-cluster distance, like Figure 3. For choosing the next K, we just need to choose the maximum one in the score list for the following clustering process.

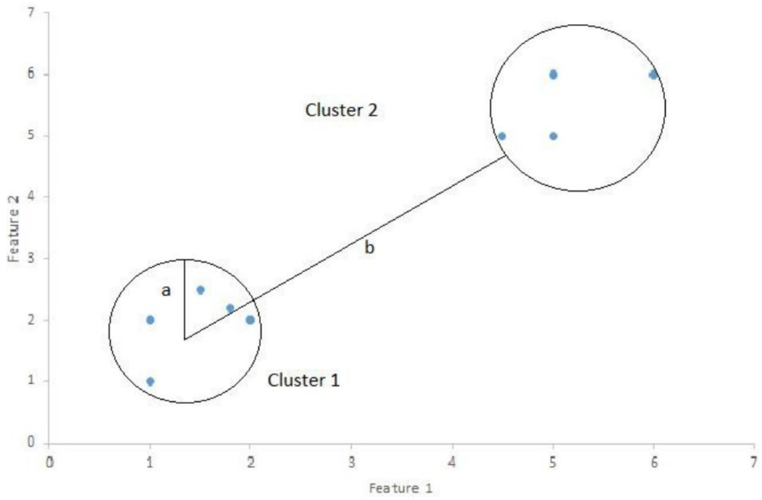
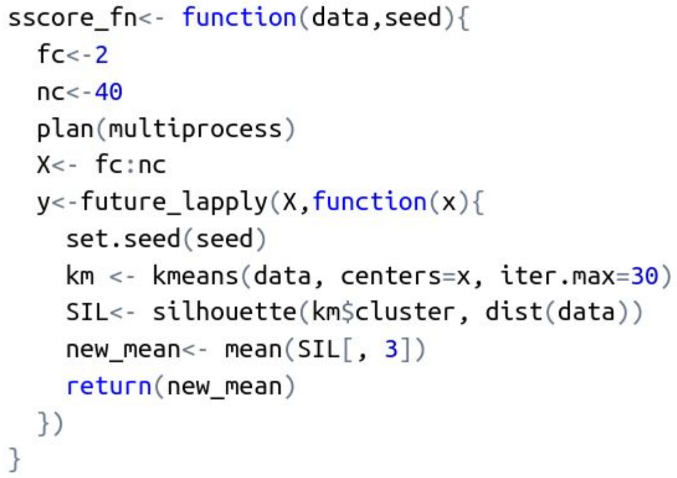


Figure 3, a sketch to show the intra-cluster distance and inter-cluster distance in the Silhouette score.

Implementation:

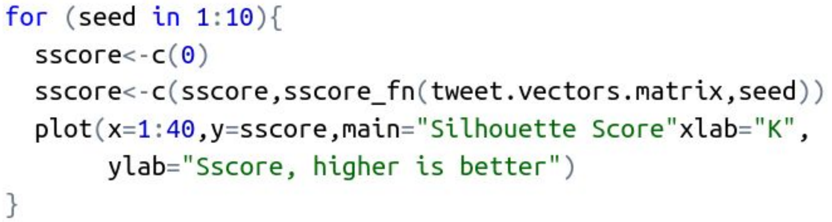
<u>Function:</u>



```
sscore_fn<- function(data,seed){
  fc<-2
  nc<-40
  plan(multiprocess)
  X<- fc:nc
  y<-future_lapply(X,function(x){
    set.seed(seed)
    km <- kmeans(data, centers=x, iter.max=30)
    SIL<- silhouette(km$cluster, dist(data))
    new_mean<- mean(SIL[, 3])
    return(new_mean)
  })
}
```

This is a parallel computing process on one node. This function called kmeans function. Then using the kmean$cluster values to get the silhouette score. Those two functions ‘kmeans’ and ‘silhouette’ are all embedded functions.

<u>Call and plot:</u>



```
for (seed in 1:10){
  sscore<-c(0)
  sscore<-c(sscore,sscore_fn(tweet.vectors.matrix,seed))
  plot(x=1:40,y=sscore,main="Silhouette Score"xlab="K",
       ylab="Sscore, higher is better")
}
```

We set seed 10 times to have a statistical plot, Figure 4. We plotted all the average silhouette scores with the standard deviations after run 10 experiments. The datasets are tweets from Jan-01-2020 to Aug-01-2020. The **tweet.vectors.matrix** contains 10,000 data points that were randomly chosen from the whole dataset.

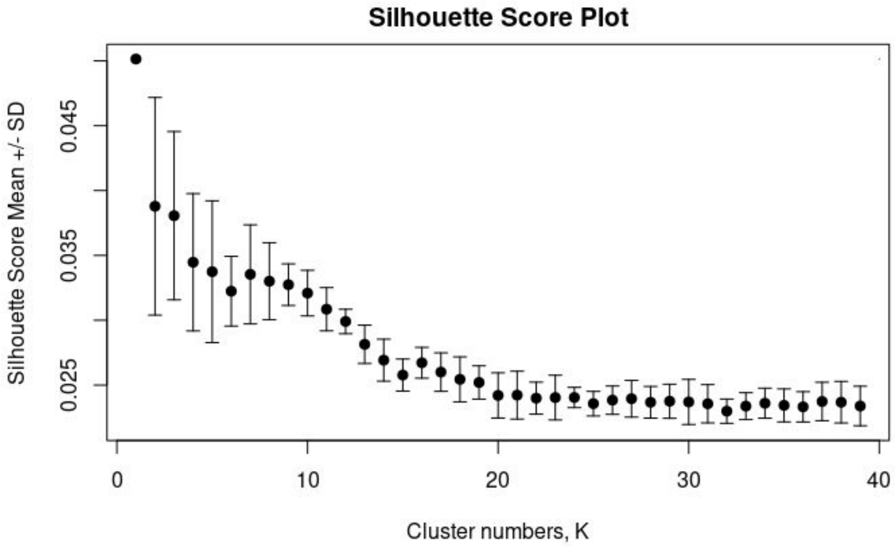


Figure 4. The average silhouette scores for all Ks (K from 2 to 40), 10 times experiment with 10 random seeds.

Result:

From the silhouette score plot, we have multiple findings:

- When the K is larger than about 10. error bars are shorter. This means each small clusters are overlapping or not too compact.
- When the K is within 2 and 10, the error bar is large. The silhouette scores have a fluctuation. It worth going to visualize and analyze the clusters when K=2:10 to see what is happening.
- The weirdest point is when K=2. The standard deviation equals to zero. That means all silhouette scores for K=2 are exactly the same.

References:

[1]https://medium.com/@rohithramesh1991/unsupervised-text-clustering-using-natural-language-processing-nlp-1a8bc18b048d

[2]https://towardsdatascience.com/clustering-metrics-better-than-the-elbow-method-6926e1f723a6