TO DETERMINE THE OPTIMAL NUMBER OF CLUSTERS

[1]Dr.P.Rathiga, [2]P.Selvi
[1]Head and Associate Professor, [2]Research Scholar,
Department of Computer Science,
[1]Navarasam Arts and Science College for Women,
Arachalur,Erode(Dt) -638101,Tamil Nadu, India.

ABSTRACT— In this paper we propose an approach to determining the number of clusters in a data set, a quantity often labelled *k* as in the *k*-means algorithm, is a frequent problem in data clustering, and is a distinct issue from the process of actually solving the clustering problem. For a certain class of clustering algorithms in particular *k*-means, *k*-medoids and expectation–maximization algorithm, there is a parameter commonly referred to as *k* that specifies the number of clusters to detect. Other algorithms such as DBSCAN and OPTICS algorithm do not require the specification of this parameter; hierarchical clustering avoids the problem altogether. The correct choice of *k* is often ambiguous, with interpretations depending on the shape and scale of the distribution of points in a data set and the desired clustering resolution of the user. In addition, increasing *k* without penalty will always reduce the amount of error in the resulting clustering, to the extreme case of zero error if each data point is considered its own cluster (i.e., when *k* equals the number of data points, *n*). Intuitively then, the optimal choice of k will strike a balance between maximum compression of the data using a single cluster, and maximum accuracy by assigning each data point to its own cluster. If an appropriate value of *k* is not apparent from prior knowledge of the properties of the data set, it must be chosen somehow. There are several categories of methods for making this decision.

Keywords — data clustering , k-means algorithm , k-medoids and expectation–maximization algorithm , optics algorithm , hierarchical clustering.

CLXVII.INTRODUCTION

Determining the optimal number of clusters in a data set is a fundamental issue in partitioning clustering, such as k-means clustering, which requires the user to specify the number of clusters k to be generated. Unfortunately, there is no definitive answer to this question. The optimal number of clusters is somehow subjective and depends on the method used for measuring similarities and A simple and popular solution consists of inspecting the dendrogram produced using hierarchical clustering to see if it suggests a particular number of clusters. Unfortunately, this approach is also subjective. In this chapter, describe different methods for determining the optimal number of clusters for k-means, k-medoids (PAM) and hierarchical clustering. These methods include direct methods and statistical testing methods:

Direct methods: consists of optimizing a criterion, such as the within cluster sums of squares or the average silhouette. The corresponding methods are named elbow and silhouette methods, respectively.

Statistical testing methods: consists of comparing evidence against null hypothesis. An example is the gapstatistic

.In addition to elbow, silhouette and gap statistic methods, there are more than thirty other indices and methods that have been published for identifying the optimal number of clusters. We'll provide R codes for computing all these 30 indices in order to decide the best number of clusters using the "majority rule".
For each of these methods:

- To describe the basic idea and the algorithm
- To provide easy-o-use R codes with many examples for determining the optimal number of clusters and visualizing the output.

CLXVIII .CLUSTERING METHODS

Elbow method:

The basic idea behind partitioning methods, such as k-means clustering, is to define clusters such that the total intra-cluster variation [or total within-cluster sum of square (WSS)] is minimized. The total WSS measures the compactness of the clustering and we want it to be as small as possible. The Elbow method looks at the total WSS as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't improve much better the total WSS.

The optimal number of clusters can be defined as follow:

1. Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
2. For each k, calculate the total within-cluster sum of square .
3. Plot the curve of WSS according to the number of clusters k.
4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

X-means clustering:

In statistics and data mining, X-means clustering is a variation of k-means clustering that refines cluster assignments by repeatedly attempting subdivision, and keeping the best resulting splits, until a criterion such as the Akaike information criterion (AIC) or Bayesian information criterion (BIC) is reached.

Information criterion approach:

Another set of methods for determining the number of clusters are information criteria, such as the Akaike information criterion (AIC), Bayesian information criterion (BIC), or the Deviance information criterion (DIC) — if it is possible to make a likelihood function for the clustering model. For example: The *k*-means model is "almost" a Gaussian mixture model and one can construct a likelihood for the Gaussian mixture model and thus also determine information criterion values.

An information–theoretic approach:

Rate distortion theory has been applied to choosing *k* called the "jump" method, which determines the number of clusters that maximizes efficiency while minimizing error by information-theoretic standards.[6] The strategy of the algorithm is to generate a distortion curve for the input data by running a standard clustering algorithm such as k-means for all values of *k* between 1 and *n*, and computing the distortion (described below) of the resulting clustering. The distortion curve is then transformed by a negative power chosen based on the dimensionality of the data. Jumps in the resulting values then signify reasonable choices for *k*, with the largest jump representing the best choice.
The distortion of a clustering of some input data is formally defined as follows: Let the data set be modeled as a *p*-dimensional random variable, *X*, consisting of a mixture distribution of *G* components with common covariance, Γ. If we let $c_1 \ldots\ldots c_k$ be a set of *K* cluster centers, with c_x the closest center to a given sample of *X*, then the minimum average distortion per dimension when fitting the *K* centers to the data is:

$d_{k=1}$ $\kappa^{c1\ldots ck}$

This is also the average Mahalanobis distance per dimension between *X* and the set of cluster centers *C*. Because the minimization over all possible sets of cluster centers is prohibitively complex, the distortion is computed in practice by generating a set of cluster centers using a standard clustering algorithm and computing the distortion using the result. The pseudo-code for the jump method with an input set of *p*-dimensional data points *X* is:

JumpMethod(X):

Let Y = (p/2)

Init a list D, of size n+1

Let D[0] = 0

For k = 1 ... n:

Cluster X with k clusters (e.g., with k-means)

Let d = Distortion of the resulting clustering

D[k] = d^(-Y)

Define J(i) = D[i] - D[i-1]

Return the k between 1 and n that maximizes J(k)

Gap statistic method:

The *gap statistic* has been published by R. Tibshirani, G. Walther, and T. Hastie (Standford University, 2001). The approach can be applied to any clustering method. The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data. The estimate of the optimal clusters will be value that maximize the gap statistic (i.e, that yields the largest gap statistic). This means that the clustering structure is far away from the random uniform distribution of points.
The algorithm works as follow:

1. Cluster the observed data, varying the number of clusters from k = 1, …, *kmax*, and compute the corresponding total within intra-cluster variation *Wk*.
2. Generate B reference data sets with a random uniform distribution. Cluster each of these reference data sets with varying number of clusters k = 1, …, *kmax*, and compute the corresponding total within intra-cluster variation *Wkb*.
3. Compute the estimated gap statistic as the deviation of the observed *Wk* value from its expected value *Wkb* under the null hypothesis: Gap(k)=1B∑b=1Blog(W∗kb)−log(Wk)Gap(k)=1B∑b=1Blog(Wkb∗)−log(Wk). Compute also the standard deviation of the statistics.
4. Choose the number of clusters as the smallest value of k such that the gap statistic is within one standard deviation of the gap at k+1: $Gap(k) \geq Gap(k+1) - s_{k+1}$.

Computing the number of clusters using R

In this section, we'll describe two functions for determining the optimal number of clusters:

1. *fviz_nbclust*() function [in *factoextra* R package]: It can be used to compute the three different methods [elbow, silhouette and gap statistic] for any partitioning clustering methods [K-means, K-medoids (PAM), CLARA, HCUT]. Note that the *hcut*() function is available only in factoextra package. It computes hierarchical clustering and cut the tree in k pre-specified clusters.
2. *NbClust*() function [in *NbClust* R package] (Charrad et al. 2014): It provides 30 indices for determining the relevant number of clusters and proposes to users the best clustering scheme from the different results obtained by varying all combinations of number of clusters, distance measures, and clustering methods. It can simultaneously computes all the indices and determine the number of clusters in a single function call.

Required R packages:

We'll use the following R packages:

factoextra to determine the optimal number clusters for a given clustering methods and for data visualization. NbClust for computing about 30 methods at once, in order to find the optimal number of clusters.

To install the packages, type this:

```
pkgs <- c("factoextra",  "NbClust")

install. Packages(pkgs)
```

Load the packages as follow:

```
library(factoextra)

library(NbClust)
```

Data preparation:

We'll use the USArrests data as a demo data set. We start by standardizing the data to make variables comparable.

```
# Standardize the data

df <- scale(USArrests)

head(df)

##            Murder Assault UrbanPop     Rape

## Alabama    1.2426   0.783   -0.521 -0.00342
```

Alaska 0.5079 1.107 -1.212 2.48420

Arizona 0.0716 1.479 0.999 1.04288

Arkansas 0.2323 0.231 -1.074 -0.18492

California 0.2783 1.263 1.759 2.06782

Colorado 0.0257 0.399 0.861 1.86497

fviz_nbclust() function: Elbow, Silhouhette and Gap statistic methods:
The simplified format is as follow:

fviz_nbclust(x, FUNcluster, method = c("silhouette", "wss", "gap_stat"))

x: numeric matrix or data frame

FUNcluster: a partitioning function. Allowed values include kmeans, pam, clara and hcut (for hierarchical clustering).

method: the method to be used for determining the optimal number of clusters.

The R code below determine the optimal number of clusters for k-means clustering:

```
# Elbow method
fviz_nbclust(df, kmeans, method = "wss") +
    geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
# Silhouette method
fviz_nbclust(df, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")
# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
# Use verbose = FALSE to hide computing progression.
set.seed(123)
fviz_nbclust(df, kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+
  labs(subtitle = "Gap statistic method")
## Clustering k = 1,2,..., K.max (= 10): .. done
## Bootstrapping, b = 1,2,..., B (= 50)  [one "." per sample]:
## ..... 50
```

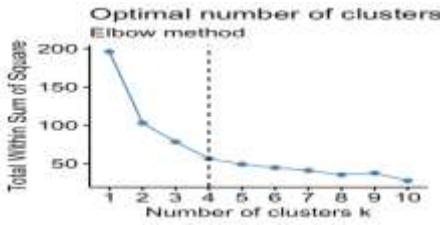

Fig 2.1elbow method

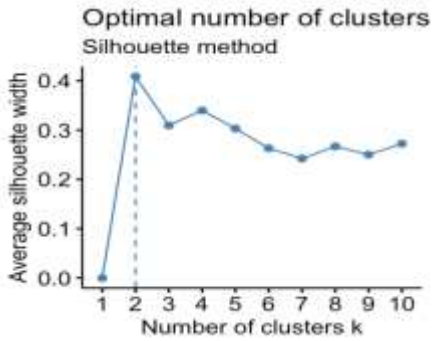

Fig 2.2 Silhouette method

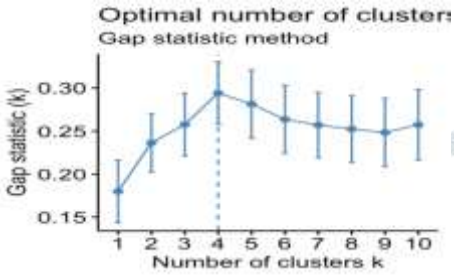

Fig 2.3 gap statistic method

According to these observations, it's possible to define k = 4 as the optimal number of clusters in the data.

NbClust() function: 30 indices for choosing the best number of clusters:
The simplified format of the function *NbClust*() is:

```
NbClust(data = NULL, diss = NULL, distance = "euclidean",
      min.nc = 2, max.nc = 15, method = NULL)
```

data: matrix
diss: dissimilarity matrix to be used. By default, diss=NULL, but if it is replaced by a dissimilarity matrix, distance should be "NULL"
distance: the distance measure to be used to compute the dissimilarity matrix. Possible values include "euclidean", "manhattan" or "NULL".
min.nc, max.nc: minimal and maximal number of clusters, respectively
method: The cluster analysis method to be used including "ward.D", "ward.D2", "single", "complete", "average", "kmeans" and more.
To compute *NbClust*() for kmeans, use method = "kmeans".
To compute *NbClust*() for hierarchical clustering, method should be one of c("ward.D", "ward.D2", "single", "complete", "average").
The R code below computes *NbClust*() for k-means:

```
## Among all indices:
## ===================
## * 2 proposed  0 as the best number of clusters
## * 10 proposed  2 as the best number of clusters
## * 2 proposed  3 as the best number of clusters
## * 8 proposed  4 as the best number of clusters
## * 1 proposed  5 as the best number of clusters
## * 1 proposed  8 as the best number of clusters
## * 2 proposed  10 as the best number of clusters
##
## Conclusion
## =========================
## * According to the majority rule, the best number of clusters is  2 .
```

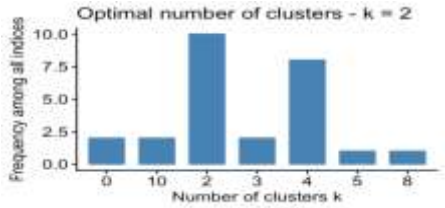

Fig 2.4 majority rule

- 2 proposed 0 as the best number of clusters
- 10 indices proposed 2 as the best number of clusters.
- 2 proposed 3 as the best number of clusters.
- 8 proposed 4 as the best number of clusters.

According to the majority rule, the best number of clusters is 2.

CLXXII.CONCLUSION

In this article, we described different methods for choosing the optimal number of clusters in a data set. These methods include the elbow, the silhouette and the gap statistic methods. We demonstrated how to compute these methods using the R function *fviz_nbclust*() [in *factoextra* R package]. Additionally, we described the package *NbClust*(), which can be used to compute simultaneously many other indices and methods for determining the number of clusters. After choosing the number of clusters k, the next step is to perform partitioning clustering as described at: k-means clustering.

REFERENCES:

[1] David J. Ketchen Jr; Christopher L. Shook (1996). "The application of cluster analysis in Strategic Management Research: An analysis and critique". Strategic Management Journal. **17** (6): 441–458. doi:10.1002/(SICI)1097-0266(199606)17:6<441::AID-SMJ819>3.0.CO;2-G.[dead link]

[2] Cyril Goutte, Peter Toft, Egill Rostrup, Finn Årup Nielsen, Lars Kai Hansen (March 1999). "On Clustering fMRI Time Series". NeuroImage. **9** (3): 298–310. doi:10.1006/nimg.1998.0391. PMID 10075900.

[3] ^ Robert L. Thorndike (December 1953). "Who Belongs in the Family?". Psychometrika. **18** (4): 267–276. doi:10.1007/BF02289263.

[4] ^ D. Pelleg; AW Moore. X-means: Extending K-means with Efficient Estimation of the Number of Clusters (PDF). Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000). Retrieved 2016-08-16.

[5] ^ Cyril Goutte, Lars Kai Hansen, Matthew G. Liptrot & Egill Rostrup (2001). "Feature-Space Clustering for fMRI Meta-Analysis". Human Brain Mapping. **13** (3): 165–183. doi:10.1002/hbm.1031. PMC 6871985. PMID 11376501. Archived from the originalon 2012-12-17.

[6] ^ Catherine A. Sugar; Gareth M. James (2003). "Finding the number of clusters in a data set: An information-theoretic approach". Journal of the American Statistical Association. **98**(January): 750–763. doi:10.1198/016214503000000666