Cluster Analysis

Xiaogang Su, Ph.D.
Department of Mathematical Sciences
University of Texas at El Paso (UTEP)
xsu@utep.edu

February 26, 2018

Contents

1 Background

Suppose the data set is huge, it is very likely that the data are heterogeneous. This means that the data might fall in several distinct groups, with members within each subgroup being similar to each other but different from members of other groups. Since it is possible that there are different models or patterns in each group, it is very difficult to spot any single pattern or model to represent the whole data set. Creating clusters of similar cases reduces the model complexity within clusters that allows data mining techniques more likely to succeed in each cluster. Even if the data do not have natural groups, partition data into homogeneous groups can be very useful. For example, it is well known that the customer preference for their products depends on geographic and demographic factors. Thus, we can use geographic and demographic factors to group customers into several segments and to develop marketing strategy for each segment. Although customers do not form these marketing segments naturally, it is much easier to develop efficient marketing strategy for each segment separately than to one single marketing strategy to target all customers.

Clustering is one important unsupervised data-mining tool. Unlike supervised data mining tools, cluster analysis has no assumptions that are made concerning the number of clusters or cluster structure. The basic objective in clustering is to discover natural groupings of the cases or variables based on some similarity or distance (dissimilarity) measures. Although there is no target variable to be predicted, clustering technique can be used in many ways. First, it can be used in missing value imputation. For example, one can use cluster mean to impute missing value for a numerical variable instead of using overall mean because overall mean does not consider the between-variable relationship. Secondly, one can use clustering to detect outliers because outliers typically belong to clusters with only one case. Thirdly, one can use clustering to discover the characteristics of clusters if he suspects that there are meaningful groupings that may represent groups of cases. After finding these meaningful groups, one can then develop different ways to deal with each group such as target marketing that will be discussed later. Fourthly, we can use cluster analysis to partition a complex data structure into several subsets in order to give supervised data-mining techniques such as decision tree or neural network a better chance of finding a good predictive model.

Since the benefits of cluster analysis are easy to see, there are huge amount of research effort in the past several decades on cluster analysis and there are many "automatic clustering" techniques available. However, different clustering techniques lead to different types of clusters and it is very difficult to tell whether a cluster analysis exercise has been successful because cluster analysis is an unsupervised data mining exercise. To use cluster techniques effectively, we need to at least understand several important aspects of choosing clustering techniques.

1.1 Choose Similarity/Distance Measure

In order to decide whether a set of cases can be split into subgroups with members are similar to other members within the same group than members from other groups, we need to define what we

mean "similar". Central to all of the goals of cluster analysis is the notion of the degree of similarity (or dissimilarity) between the individual objects being clustered. This can only come from subject matter considerations. Suppose we want to group a deck of ordinary playing cards into clusters. There are many ways to do so such as

- Two clusters: One cluster has all the face card and another cluster has all other cards.
- Four clusters: Each suit of thirteen cards forms one cluster.
- Two clusters: Red suits are in one cluster and black suits are in another cluster.
- Thirteen clusters: Each cluster has all cards that have the same face value.

Obviously, the clusters obtained are very different with different "similarity measure". Thus, we need to know how to choose a similarity measure before selecting clustering technique.

The distance measure between cases is pivotal in many clustering methods. In its strict mathematical definition, a *distance* or *metric* on space $\mathcal{X}$ is a function $d: \mathcal{X} \times \mathcal{X} \to [0, +\infty)$ that satisfies the following properties:

(i) $d(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}', \mathbf{x})$
(ii) $d(\mathbf{x}, \mathbf{x}') > 0$ for $\mathbf{x} \neq \mathbf{x}'$.
(iii) $d(\mathbf{x}, \mathbf{x}') = 0$ if $\mathbf{x} = \mathbf{x}'$.
(iv) $d(\mathbf{x}_1, \mathbf{x}_2) \leq d(\mathbf{x}_1, \mathbf{x}_3) + d(\mathbf{x}_2, \mathbf{x}_3)$.

However, as we will see, many distances in statistics may not satisfy these properties, including the Mahalanobis distance.

The most commonly used distance measure is the Minkowski metric. Given two cases with p variables: $\mathbf{x} = (x_1, \ldots, x_p)^T \in \mathbb{R}^p$ and $\mathbf{x}' \in \mathbf{R}^p$, the Minkowski metric ($\mathcal{L}_r$ norm) is defined as

$$d(\mathbf{x}, \mathbf{x}') = \left[\sum_{j=1}^{p} \left| x_j - x'_j \right|^r \right]^{1/r}. \tag{1}$$

For $r = 2$, $d(\mathbf{x}, \mathbf{x}')$ becomes the Euclidean distance. For $r = 1$, $d(\mathbf{x}, \mathbf{x}')$ becomes to the mean absolute deviation between the two cases. For $r = \infty$, $d(\mathbf{x}, \mathbf{x}')$ becomes to the maximum absolute deviation between the two cases.

Let $\mathbf{X}_{n \times p} = (x_{ij})$ denote the $n \times p$ data matrix. Let $\mathbf{x}_i \in \mathbf{R}^p$ denote the ith row vector of $\mathbf{X}$ for $i = 1, \ldots, n$. We denote distance between ith and i'-th observations as

$$d_{ii'} := d(\mathbf{x}_i, \mathbf{x}_{i'})$$

for $i, i' = 1, \ldots, n$.

Since this measure of distance assume some degree of commensurability between the different variables. Thus, it would be effective if each variable measured using the same units and each variable is equally important. But, it is very unlikely that all variables in a data mining exercise were measured with the same unit. One way to deal with this incommensurability is to standardize the data by divided each of the variables by its sample standard deviation. In addition, if we have the idea of the relative importance of these variables, then we can weight them (after standardization) to yield the weighted standardize distance measure. Alternatively, one may normalize their range

$$x'_{ij} := \frac{x_{ij} - \min_i(x_{ij})}{\max_i(x_{ij}) - \min_i(x_{ij})}$$

so that all variables range from 0 to 1 after transformation.

Another commonly used distance measure is the statistical distance that takes the variance-covariance structure into consideration, i.e., the squared Mahalanobis distance

$$d^2(i,i') = (\mathbf{x}_i - \mathbf{x}_{i'})^T \mathbf{S}^{-1} (\mathbf{x}_i - \mathbf{x}_{i'}).$$

Mahalanobis distance is invariant to measurement scale or any nonsingular linear transformation. However, its calculation involves the inverse of the sample variance-covariance matrix $\mathbf{S}$, which could add to computing burden with large-scale data. Besides, without prior knowledge of the distinct groups and the number of groups, the variance-covariance structure cannot be computed.

Sometimes the available data is represented directly in terms of either distance (dissimilarity, difference) or proximity (alikeness, resemblance, or affinity) between pairs of objects. These can be either similarities or dissimilarities (difference or lack of affinity). For example, in social science experiments, participants are asked to judge by how much certain objects differ from one another. Dissimilarities can then be computed by averaging over the collection of such judgments. Nevertheless, subjectively judged dissimilarities are seldom distances in the strict sense, since the triangle inequality $d(i,j) \leq d(i,k) + d(j,k)$ again does not hold.

The Pearson's (sample) correlation coefficient, when computed between two row vectors $\{\mathbf{x}, \mathbf{x}'\}$ of $\mathbf{X}$,

$$r(\mathbf{x}, \mathbf{x}') = \frac{\sum_{j=1}^{p}(x_j - \bar{x})(x'_j - \bar{x}')}{\sqrt{\sum_{j=1}^{p}(x_j - \bar{x})^2 \cdot \sum_{j=1}^{p}(x'_j - \bar{x}')^2}}, \quad \text{with } \bar{x} = \frac{1}{p}\sum_{j=1}^{p} x_j \text{ and } \bar{x}' = \frac{1}{p}\sum_{j=1}^{p} x'_j$$

is also a common similarity (vs. distance) measure, especially when clustering *variables*. However, it can be easily shown that, if the observations have been standardized, then

$$\| \mathbf{x} - \mathbf{x}' \|_2^2 \;=\; \sum_{j=1}^{p}(x_j - x'_j)^2 \;\propto\; 1 - r(\mathbf{x}, \mathbf{x}'),$$

where $\propto$ means 'proportional to' up to a multiplicative constant. Hence clustering based on correlation (similarity) is equivalent to that based on squared Euclidean distance (dissimilarity).

For these reasons, Euclidean distance is often preferred for clustering purpose. There are many other definitions available for distance. It is important to emphasize that different proximity measures can dramatically affect the formations of the resulting clusters. What distance definition is best suitable in the project depends on the context and the types of variables (nominal, ordinal, interval, or ratio), among other factors.

1.2 Problem Statement

The raw data matrix $\mathbf{X}$ is $n \times p$

$$\mathbf{X} \;=\; \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

and the proximity or dissimilarity matrix $\mathbf{D}$ is $n \times n$

$$\mathbf{D} = \begin{bmatrix} 0 & d_{12} & \cdots & d_{1n} \\ d_{21} & 0 & \cdots & d_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ d_{n1} & d_{n2} & \cdots & 0 \end{bmatrix} = (d_{ii'})$$

is a symmetric matrix with diagonal elements 0. Assume that $\mathbf{X}$ has been normalized so that the distance is invariant to measurement scale. If the Euclidean distance matrix is used for $\mathbf{D}$, the following operation transforms $\mathbf{X}$ to $\mathbf{D}$:

$$\mathbf{D} = \mathbf{J}\text{diag}(\mathbf{B}) - 2\mathbf{B} + \text{diag}(\mathbf{B})\mathbf{J},$$

where $\mathbf{J}$ denotes the $n \times n$ matrix with all entries equal to 1 and $\mathbf{B} = \mathbf{X}\mathbf{X}^T$ is the kernel matrix.

The task of cluster analysis is to partition data set into K disjoint groups (called clusters) of cases such that the cases within each cluster are as homogeneous as possible, that is, given a sample data $\mathbf{X}$ consisting of n cases $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, our task is to find K disjoint clusters $\mathcal{C}_K = \{c_1, c_2, \ldots, c_K\}$ such that each case $\mathbf{x}_i$ is assigned to one and only one cluster c_k. It is natural to consider $\mathcal{C}_K$ as the permuted set of the indexes so that $\bigcup_k c_k = \{1, 2, \ldots, n\}$ and $c_k \cap c_{k'} = \emptyset$. Thus the notational use of both $i \in c_k$ and $\mathbf{x}_i \in c_k$ is allowed.

Many clustering methods rely on the distance/proximity matrix $\mathbf{D}$ only, often referred to distance-based clustering methods; while other procedures entail the raw data $\mathbf{X}$ or both, e.g., model-based clustering. For distance-based clustering, it is convenient to relate the proximity matrix to graph theory. Given a distance matrix $\mathbf{D}$, first transfer it into a adjacency or proximity or similarity matrix so that each element now represents how similar (instead of distant) two observations are from each other. For simplicity, the same notation $\mathbf{D}$ is used for similarity measure as well. Then a connected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E}$ can be drawn from $\mathbf{D}$, where $\mathbf{V} = \{1, \ldots, n\}$ is the set of vertices, each vertex corresponding to an observation, and $\mathbf{E}$ is the set of edges. Some color scheme such as gray scale may be used to denote the magnitude of similarity; edges with little similarity may be erased after applying a threshold. The graph corresponding to the clustered data $\mathcal{C}_K$ would reduce to a similarity matrix $\mathbf{D}_K$ with a block diagonal structure. Thus the problem statement for distance-based clustering can be described in general as the following optimization problem:

$$\min_{K, \mathcal{C}_K} \|\mathbf{D} - \mathbf{P}\mathbf{D}_K\mathbf{P}\|_F, \tag{2}$$

where $\mathbf{P}$ denotes a permutation matrix. A permutation matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$ is an orthogonal (i.e., $\mathbf{P}^T = \mathbf{P}^{-1}$) 0-1 binary matrix with exactly one entry of 1 in each row and each column and 0s elsewhere. Problem (2) can be equivalently formulated as $\min_{K, \mathcal{C}_K} \|\mathbf{P}\mathbf{D}\mathbf{P} - \mathbf{D}_K\|_F$. Pre-multiplying $\mathbf{P}\mathbf{D}$ permutes the rows of $\mathbf{D}$ while post-multiplying $\mathbf{D}\mathbf{P}$ permutes its columns. The same operations on both rows and columns are needed to keep $\mathbf{D}$ or $\mathbf{D}_K$ symmetric afterwards. Note that both K and the scheme $\mathcal{C}_K$ are decision variables in solving (2), which is an NP-hard problem. One has to seek approximate solutions. However, this general formulation allows for refinement of the results obtained from a clustering procedure.

When dealing with huge data or dynamic data (e.g., data streams), it is important to make the clustering algorithms incremental and the cluster summaries need to be readily available. For this purpose, many algorithms record three measures for each cluster c, which include

- n_c — The sample size in cluster c;

- $\sum_{i \in c} x_{ij}$ — the sum of observed values on variable X_j in cluster c;
- $\sum_{i \in c} x_{ij}^2$ — the sum of squared values for variable X_j in cluster c.

With the above three quantities $\left\{ n_c, \sum_{i \in c} x_{ij}, \sum_{i \in c} x_{ij}^2 \right\}$, one can compute the mean and s.d. in each cluster and the within/between-cluster variations. They can also be easily updated when designing an on-line clustering algorithm.

1.3 Determine the Right Number of Clusters

One important question to ask in cluster analysis is "what is the right number of clusters?". In the k-mean cluster algorithm, the original choice number of K determines the number of clusters that will be found. If this number does not match to the natural structure of the data, the technique will not obtain good results. Unless there is good prior knowledge on how many clusters exist in the data, it is very difficult to choice the number of K before applying k-mean cluster technique. Typically, the miner needs to try several different K's before finding the "right" number of K and obtain the so-called scree plot. In general, the best set of clusters is the one that does best job of keeping the distance between members in the same cluster small and the distance between members of adjacent clusters large. Many methods for automatic determination of K have been suggested. We shall introduce a few typical ones. However, if the purpose of clustering is to detect unexpected or outlying patterns, the right number of clusters might be relatively large so that clusters with a small number of observations, which are likely to be outliers, could be extracted.

1.4 Post Hoc Analysis

Clustering is a powerful unsupervised knowledge discovery technique, however, it has weakness and limitations. For example, if one does not know what he is looking for, one may not recognize it when one finds it. Although the clustering technique can help to find clusters, it is up to the user to interpret them. This is crucial in many applications such as customer segmentation so that different marketing or intervention strategies can be designed and tailored. Two common approaches can help the user to understand clusters:

- Use graphical tools or summary statistics to exam the within cluster distribution for each variable;
- Building a supervised model with the cluster label as the target variable and using it to derive rules explaining how to assign new records to the correct cluster.

There are also so many clustering related indices that have been proposed to compare or measure the performance of clustering results from different perspectives, as explained by Desgraupes (2013) in detail. To examplify, two indices for comparing two clustering results, the Jaccard index and the Rand index, are discussed here. They are similar to the stability measure in Section 3.4. Given two clustering results $\mathcal{C}$ and $\mathcal{C}'$. Let N_{11} denote the number of pairs that are clustered together by both $\mathcal{C}$ and $\mathcal{C}'$; N_{00} is the number of pairs that are not clustered together by both $\mathcal{C}$ and $\mathcal{C}'$; N_{10} is the number of pairs that are clustered together by $\mathcal{C}$ but not $\mathcal{C}'$; and N_{01} is the number of pairs that are clustered together by $\mathcal{C}'$ but not $\mathcal{C}$. Note that $\sum_{i=0}^{1} \sum_{i'=0}^{1} N_{ii'} = \binom{n}{2}$.

- The Jaccard index is given by $JI = N_{11}/(N_{11} + N_{10} + N_{01})$, i.e., proportion of pairs that cluster together in both methods relative to those pairs that cluster together in at least one method.

- The Rand index is given by $RI = (N_{11} + N_{00})/(N_{11} + N_{10} + N_{01} + N_{00})$.

RI ranges from 0 (no pair classified in the same way under both clusterings) to 1 (identical clusterings). The value of RI depends on both, the number of clusters and the number of elements. The Rand Index is highly dependent upon the number of clusters. In the (unrealistic) case of independant clusterings, the Rand Index converges to 1 as the number of clusters increases which is undesirable for a similarity measure. The Jaccard index JI disregards the pairs of elements that are in different clusters for both clusterings.

2 Clustering Methods

The best way to find clusters is to examine all possible clusters. However, it takes too long to examine all clusters even with the fastest and largest computer. Because of this problem, a wide variety of clustering algorithms have emerged that find "reasonable" clusters without having to look at all possible clusters.

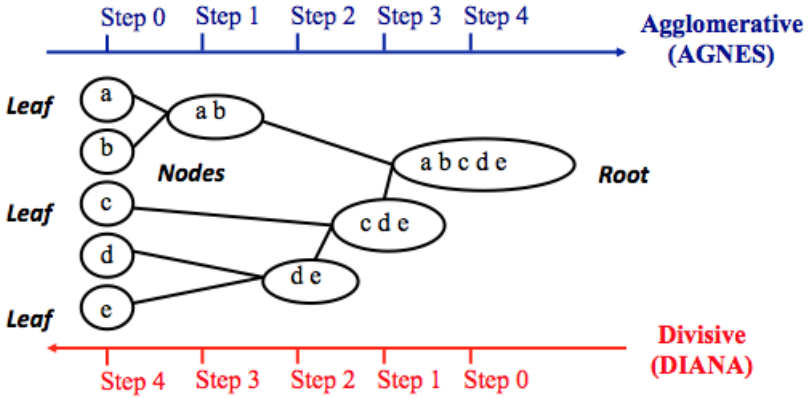

Figure 1: Illustration of Hierarchical Clustering: Agglomerative vs. Divisive.

2.1 Hierarchical Clustering

Hierarchical clustering techniques proceed by either a series of successive merges (bottom-up) or a series of successive divisions (top-down). Agglomerative hierarchical clustering methods start with the individual cases. Thus, there are as many clusters as cases. Most "similar" cases are first merged to form a reduced number of clusters. This is repeated until just one cluster with all cases.

An agglomerative algorithm for clustering can be described as follows: A tree dendrogram can

Algorithm 1 Agglomerative Hierarchical Clustering

Let $\mathcal{C} = \{\mathbf{x}_i : i = 1, \ldots, n\}$, i.e., each observation forming a cluster;
while there is more than one cluster left **do**
  Let c and c' be the two clusters that have the minimum distance;
  $c := c \cup c'$;
  Remove c';
end while

be drawn from the hierarchical clustering algorithm. See Figure 1 for the graphical illustration.

However, there is one important issue that has not yet been solved. That is, how do we define the distance between two clusters $d(c, c')$. There are five linkage methods designed for this purpose.

(1) In "*Single Linkage*" method, the distance between two clusters is defined as

$$d(c, c') \ := \ \min \left\{ d_{ii'} | i \in c \text{ and } i' \in c' \right\}$$

The clusters formed by single linkage method will not be affected by the distance measures used if these distance measures have the same relative ordering. It also has the property that if two pairs of clusters are equidistance it does not matter which one is merged first. The overall result will be the same. Single linkage method is the only clustering method that can find non-ellipsoidal clusters. The tendency of single linkage method to pick up long string like cluster is knows as chaining. This tendency and sensitivity to outlier cases and perturbation of the data combined make it less useful in customer segmentation application.

(2) In "*Complete Linkage*" method, the distance between two clusters is defined as

$$d(c, c') \ := \ \max \left\{ d_{ii'} | i \in c \text{ and } i' \in c' \right\}$$

The clusters formed by complete linkage method will not be affected by the distance measures used if these distance measures have the same relative ordering. Complete linkage method tends to find clusters to be equal size in terms of the volume of space occupied, making it is particularly suitable in customer segmentation application.

(3) In "*Average Linkage*" method, the distance between two clusters is defined as

$$d(c, c') \ := \ \frac{\sum_{i \in c} \sum_{i' \in c'} d_{ii'}}{n_c \cdot n_{c'}}.$$

The clusters formed by average linkage method will be affected by the distance measures used even if these distance measures have the same relative ordering. This makes average linkage less attractive in data mining application.

(4) The *centroid* linkage clustering computes the dissimilarity between the centroid for cluster 1 (a mean vector of length p variables) and the centroid for cluster 2. With the centroid method,

$$d(c, c') \ := \ d(\bar{\mathbf{x}}_c, \bar{\mathbf{x}}_{c'}),$$

where $\{\bar{\mathbf{x}}_c, \bar{\mathbf{x}}_{c'}\}$ denotes the center or mean vector of all observations in cluster c and c', respectively.

(5) The *Ward's minimum variance* method minimizes the total within-cluster variance from the perspective of analysis of variance (ANOVA). At each step the pair of clusters with minimum between-cluster distance are merged. To motivate Ward's method, note that the sum of squares for variation increases every time we merge two clusters. The increase amount in SS is given by

$$\begin{aligned} \delta \ &= \ \sum_{i \in (c \cup c')} (\mathbf{x}_i - \bar{\mathbf{x}}_{c \cup c'})^2 - \left\{ \sum_{i \in c} (\mathbf{x}_i - \bar{\mathbf{x}}_c)^2 + \sum_{i \in c'} (\mathbf{x}_i - \bar{\mathbf{x}}_{c'})^2 \right\} \\ &= \ \frac{n_c \, c_{c'}}{n_c + n_{c'}} (\bar{\mathbf{x}}_c - \bar{\mathbf{x}}_{c'})^2 . \end{aligned}$$

With agglomerative clustering, the sum of squares starts out at zero (because every point is in its own cluster) and then grows as we merge clusters. Ward's method keeps this growth as *small* as possible. Therefore, the Ward's method is very similar to the centroid method, however, it encourages merging smaller clusters first when the distance between centers is the same for, say, two pairs of clusters.

In hierarchical methods, one essentially needs to update some entries of the distance matrix $\mathbf{D}$ after cluster merging or dividing at each step. As illustrated in the following toy example, the dimension of $\mathbf{D}$ shrinks in agglomerative clustering or expands in divisive clustering. The corresponding entries of $\mathbf{D}$ can be recursively updated based on previous entries, which turns out to be a linear combination as formulated in the Lance-Williams algorithm. Specifically, if clusters c_i and c_j are merged together into $c_i \cup c_j$, then the distance between c_k and the new cluster can be computed as

$$d_{(ij)k} = \alpha_{ij}\, d_{ij} + \alpha_{ik}\, d_{ik} + \alpha_{jk}\, d_{jk},$$

with coefficients $\{\alpha_{ij}, \alpha_{ik}, \alpha_{jk}\}$ varying for different methods of defining the cluster distance.

Most hierarchical clustering methods only need the distance matrix. This means that it does not need to store all variable values for each case. Suppose we can compute the "distance" between variables, these methods can be applied in *variable clustering* as well. We will briefly address this issue in Section 4. These methods mentioned here have drawbacks. First, a case will stay in the same cluster once it is assigned to this cluster. This means that reallocation is not allowed in the clustering process even if one case has wrongly assigned to a cluster. Secondly, these methods are sensitive to outliers and "noise". Thus, we need to try several different cluster methods and, within each method, to try several distance measures. If the outcomes from all methods are consistent with one another, perhaps a set of good clusters has been found. Also, we can add small errors to each case before applying clustering method to see how stable the clusters are.

Example 1. (Single Linkage, Complete Linkage, and Average Linkage)
The distance between pairs of five cases are given below:

$$\begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & 2 & 8 & 0 \end{bmatrix}.$$

Cluster the five cases using each procedure and draw the dendograms and compare the results:

(a) Single linkage hierarchical procedure.
(b) Complete linkage hierarchical procedure.
(c) Average linkage hierarchical procedure.

2.2 K-Means Clustering

The K-means clustering (Lloyd, 1982) is another popular method. It aims to solve the following optiomization problem

$$\underset{\mathcal{C}, \boldsymbol{\mu}_k}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{i \in c_k} (\mathbf{x}_i - \boldsymbol{\mu}_k)^2.$$

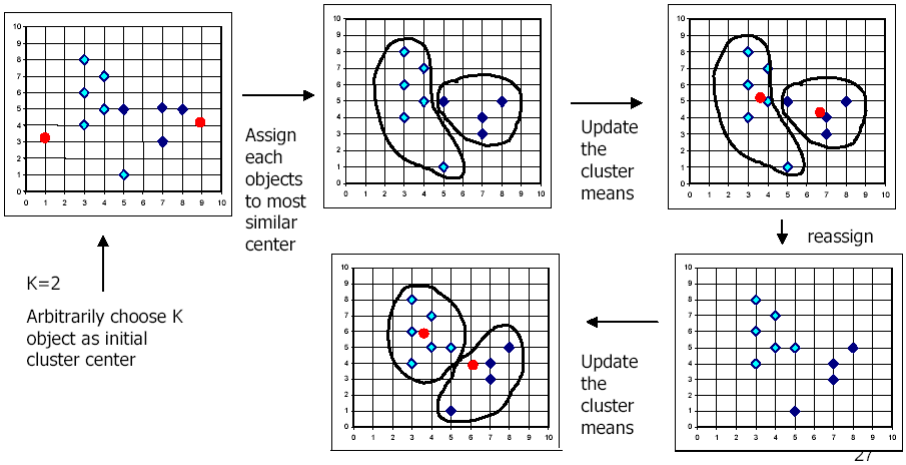

Figure 2: Illustration of K-Means Clustering.

The problem is nonconvex and NP-hard in terms of computational complexity. However, efficient heuristic algorithms that converge to a local optimum exist. One procedure is given in Algorithm 2.

A graphical illustration is provided in Figure 2. The time complexity of K-means algorithm is $O(KIn)$, where I is the number of iterations. Since K, the number of clusters, is fixed in partition based clustering methods, the selection of K is very important. If the number of natural clusters is different from K, the results supplied by the partition based clustering algorithm will not be satisfactory. One way to avoid this problem is to try several different numbers of K and run the algorithm several times. Then use a method (see Section 3) to select the "right" number of clusters.

Algorithm 2 K-means Clustering

Choose K;
Let $\{\bar{\mathbf{x}}_k : k = 1, 2, \ldots, K\}$ be K randomly selected points in the learning data $\mathcal{L}$;
for $i = 1$ **to** I **do**
  # Form clusters.
  for $k = 1$ **to** K **do**
    $c_k = \{\mathbf{x} \in \mathcal{L} \mid d(\bar{\mathbf{x}}_k, \mathbf{x}) \leq d(\bar{\mathbf{x}}_{k'}, \mathbf{x}),\ \text{for } k' \neq k = 1, \ldots, K\}$.
  end for
  # Compute the new cluster centers;
  for $k = 1$ **to** K **do**
    $\bar{\mathbf{x}}_k = \sum_{i \in c_k} \mathbf{x}_i / n_k$;
  end for
end for

There are many variants of K-means clustering. As an example, the K-medoids clustering provides a more robust version to outlying observations, yet with dramatically increased computation cost. They only differ in the way of defining the cluster center. Instead of using the mean vector as the center as in K-means, K-medoids clustering uses the observation in the cluster that minimizes

total distance to other points in that cluster, namely,

$$\mathbf{m}_k := \underset{\mathbf{x}_i \in c_k \cap \mathcal{D}}{\operatorname{argmin}} \sum_{i':\mathbf{x}_{i'} \in c_k} d(\mathbf{x}_i, \mathbf{x}_{i'}).$$

Yet another alternative is to used cluster medians to achieve robustness. One noteworthy issue is you may get a different clustering result (a local optimum) for each run owing to the nonconvex nature. In addition, the K-means clusterings for different K values are not necessarily nested.

2.3 Spectral Clustering

Spectral cluster analysis is intuitively motivated from graph theory. Any distance or adjacency matrix can be related to a weighted undirected graph. The aim of spectral clustering is to cut this weighted (undirected) graph into a number of disjoint pieces (clusters) such that the intra-cluster weights (similarities) are high and the inter-cluster weights are low. In the following, I try to explain the essential ideas behind this seemingly mysterious algorithm, by following von Luxburg (2007).

Spectral analysis has several variants, some based on unnormalized graph Laplacian and other based on the normalized graph Laplacian. I will cover only the unnormalized version. Spectral analysis usually works with an adjacency or affinity matrix. There are many ways to transfer a distance matrix into an adjacency matrix, such as applying the radial basis function or Gaussain similarity function

$$d_{ii'} := \exp\left(-\frac{\| \mathbf{x}_i - \mathbf{x}_{i'} \|^2}{2\sigma^2}\right),$$

with variance or scale parameter σ^2.

Let $\mathbf{D}$ denote the adjacency or affinity matrix and $\mathbf{W} = \operatorname{diag}(\mathbf{DJ})$ with diagonal elements equal to the row/column sum of $\mathbf{D}$. Define the unnormalized graph Laplacian matrix as $\mathbf{L} = \mathbf{W} - \mathbf{D}$. Some properties of the Laplacian $\mathbf{L}$ are listed in the following proposition. Note that the diagonal elements $d_{ii'}$ of $\mathbf{D}$ (which are actually hard to define) do not show up in $\mathbf{L}$.

Proposition 1. *(Properties of Graph Laplacian* $\mathbf{L}$*)*

a. For every vector $\mathbf{v} \in \mathbb{R}^n$*, we have*

$$\mathbf{v}^T \mathbf{L} \mathbf{v} = \frac{1}{2} \sum_{i,i'=1}^{n} d_{ii'}(v_i - v_i')^2.$$

b. Matrix $\mathbf{L}$ *is symmetric and positive semi-definite, with eigenvalues* $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ *(sorted in an ascending order here!) The eigenvector associated with eigenvalue 0 is* $\mathbf{j} = (1, \ldots, 1)^T$.

c. Suppose that a graph $\mathcal{G}$ *consists of* K *connected components* $\mathcal{G}_1, \ldots \mathcal{G}_K$*, while there is no connection between these components. Then the multiplicity of the eigenvalue 0 of the Laplacian* $\mathbf{L}$ *(associated with* $\mathcal{G}$*) equals* K*. The eigenspace of eigenvalue* 0 *is spanned by the indicator vectors* $\{\mathbf{1}_{\mathcal{G}_1}, \ldots, \mathbf{1}_{\mathcal{G}_K}\}$ *of those components, where the vector* $\mathbf{1}_A \in \mathbb{R}^n$ *has entries equal to 1 if* $i \in A$ *and 0 otherwise.*

Do properties (b) and (c) seem conflicting to you? They should not. In (b), the graph Laplacian $\mathbf{L}$ is based on a completely connected graph $\mathcal{G}$ (seen with an observed data set for example); in (c), we

assume that graph $\mathcal{G}$ consists of K connected components $\mathcal{G}_1, \ldots \mathcal{G}_K$. In fact, property (c) is proved with (b), where each subgraph $\mathcal{G}_k$ satisfies (b). Property (c) is crucial by saying that the clustering structures in data would be revealed by the eigenvectors of the graph Laplacian $\mathbf{L}$ associated with its eigenvalue 0, yet up to certain permutation. This motivates the following algorithm for executing spectral clustering, as given in Algorithm 3.

Algorithm 3 Spectral Clustering

- Transform the distance matrix $\mathbf{D}$ into a similarity matrix;
- Compute the graph Laplacian $\mathbf{L} = \text{diag}DW - \mathbf{D}$;
- Apply eigen or spectral decomposition to $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$;
- Plot the sorted (in the ascending order) eigenvalues in $\mathbf{\Lambda}$ and determine the best K;
- Obtain matrix $\mathbf{U}_K$ by taking the first K columns of $\mathbf{U}$;
- Apply a K-means algorithm to cluster the rows of $\mathbf{U}_K$, which gives the cluster memberships for each observation i, for $i = 1, \ldots, n$.

An alternative method for spectral clustering is similar to the top-down divisive hierarchical clustering, where data are bisected recursively. The essential idea is that of Fiedler (1989) Theory of spectral graph partitioning. Given a connected graph $\mathcal{G}$, it is desired to divide $\mathcal{G}$ into *two* connected components or subgraphs, $\mathcal{G}_1$ and $\mathcal{G}_2$, in such a way to minimize the number of edges which have one end vertex in each of the two subgraphs, while the two subgraphs have a nearly equal number of vertices (as close to equal as is possible). By definition, a *connected* graph is a graph in which any two vertices are connected to each other by *paths* (i.e., one or more edges).

Recall that the eigenvalues of L satisfies $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Fiedler's theory considers the second eigenvector $\mathbf{u}_2$ associated with the second smallest eigenvalue λ_2, which is known as the Fiedler vector. Rewrite it by specifying all its elements: $\mathbf{u}_2 = (u_{12}, u_{22}, \ldots, u_{n2})^T$. To form an optimal partition, simply create two subgraphs $\mathcal{G}_1$ and $\mathcal{G}_2$ such that the ith observation goes to $\mathcal{G}_1$ if $u_{i2} > 0$ and to $\mathcal{G}_2$ if $u_{i2} < 0$ otherwise. Fiedler showed that this partition method attempts to minimize the number of edges between $\mathcal{G}_1$ and $\mathcal{G}_2$. Next, either of $\mathcal{G}_1$ and $\mathcal{G}_2$ is partitioned in a similar fashion. Repeating the procedure leads to a hierarchical clustering of the data.

There are options available in spectral cluster analysis. First, the similarity matrix $\mathbf{D}$ to start can be modified by applying a threshold value ϵ (ϵ-neighborhood graph) so that similarity less than ϵ are eliminated as 0. Alternatively, a (mutually) k-nearest neighbor graph can be used, in which only the largest k elements in each row of $\mathbf{D}$ are used. Secondly, the algorithm can be executed based on the normalized graph Laplacian. Two commonly used ones are defined as

$$\begin{aligned} \mathbf{L}_1 : &= \mathbf{W}^{-1/2}\mathbf{L}\mathbf{W}^{-1/2} = \mathbf{I} - \mathbf{W}^{-1/2}\mathbf{D}\mathbf{W}^{-1/2}; \quad \text{(symmetric)} \\ \mathbf{L}_2 : &= \mathbf{W}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{W}^{-1}\mathbf{D} \quad \text{(random walk)}. \end{aligned}$$

The spectral clustering can be justified from three different theoretical settings: (a), graph cut via relaxation; (b) random walks; and (c) matrix perturbation theory. See the tutorial article of von Luxburg (2007) for more details.

According to the properties of the graph Laplacian, one convenient choice of K is to choose the number such that all eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \lambda_K$ are very small, but λ_{K+1} becomes relatively large. Figure 3 is an illustration taken from von Luxburg (2007), which shows the performance of eigengap using three data sets with different signal/noise strength. One advantage of spectral clustering is that we can identify an 'optimal' number of clusters from eigengap without actually seeing clustering results.

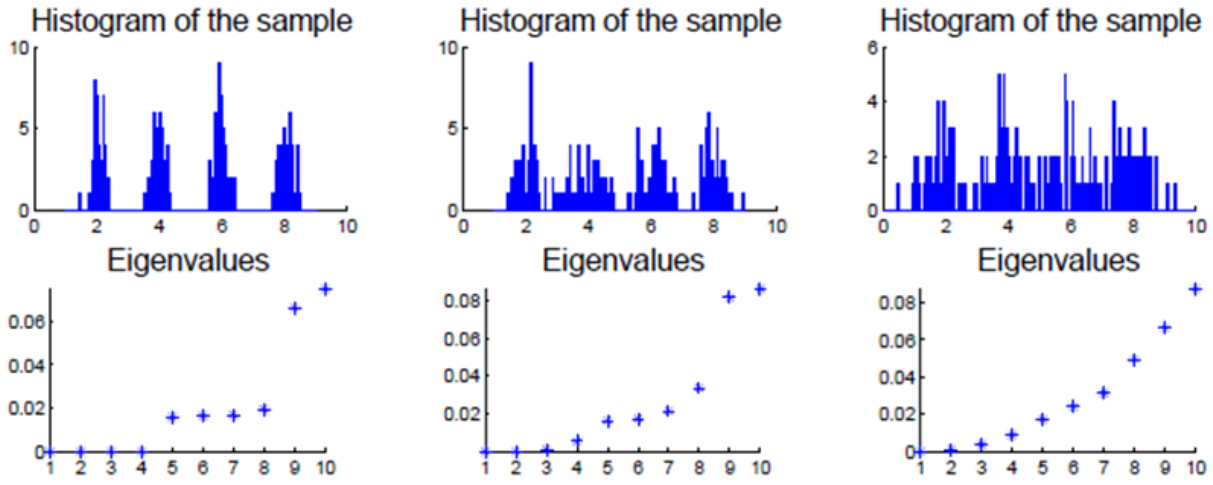

Figure 3: Determine the number of clusters using the eigenvalues, taken from von Luxburg (2007).

2.4 Other Clustering Methods

There are many other clustering methods proposed in the literature. Model-based clustering methods are gaining popularity in some fields. In this approach, finite mixture models, e.g., Gaussian mixtures, are assumed for the underlying distributions. Cluster analysis involves recovering these distributions from the observed data. Let $f_k(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ denote the multivariate normal density for the k-th cluster c_k and π_k be the mixing probability. Then the likelihood is

$$\begin{aligned} L(\boldsymbol{\pi}; \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K; \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K) &= \prod_{k=1}^{K} \prod_{i \in c_k} \pi_k \, f_k(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &\propto \prod_{k=1}^{K} \prod_{i \in c_k} \pi_k \, |\boldsymbol{\Sigma}_k|^{-1/2} \exp\left\{ -\frac{(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k)}{2} \right\}, \end{aligned}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^T$ with $\sum_k \pi_K = 1$. Typically, EM algorithm and the Bayesian methods are common approaches for model estimation. To determine the optimal number of clusters, a model selection criterion such as BIC can be used naturally. Alternative, this method can be naturally combined with the ℓ_1 regularization. In particular, the fused LASSO approach can aid in automatic selection of K.

Another popular group of methods are density-based. This direction is exemplified by mean/medoid-shift, mode-seeking, and bump hunting. We briefly introduce the most popular mean-shift algorithm (Cheng, 1995). The idea involves estimation of the joint density of data, which is done nonparametrically via KDE (kernel density estimation):

$$p(\mathbf{x}) = \frac{1}{nh^p} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h} \right), \tag{3}$$

where the kernel function $K(\mathbf{t})$ is typically the Gaussian density $K(\mathbf{t}) = (2\pi)^{-p/2} \exp(-\|\mathbf{t}\|^2/2)$; the standard deviation parameter $h > 0$ is the bandwidth and the term h^p shows up naturally in

multivariate Gaussian density. Each observation will be clustered towards its nearest mode (i.e., a local maximum) of the density. To seek the mode, taking the derivative of (3) gives the gradient

$$\bigtriangledown p \;=\; \underbrace{\frac{2}{nh^{p+2}(2\pi)^{p/2}} \cdot \left[\sum_{i=1}^{n} \dot{K}\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)\right]}_{\text{Term A}} \cdot \underbrace{\left[\frac{\sum_{i=1}^{n} \mathbf{x}_i \dot{K}\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)}{\sum_{i=1}^{n} \dot{K}\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)} - \mathbf{x}\right]}_{\text{Term B}},$$

where $\dot{K}(\cdot)$ is the derivative of function $K(\dot{})$. Term A is proportional to the density estimate at $\mathbf{x}$; Term B is the gradient direction pointing toward the steepest increase of $p(\mathbf{x})$ – the ascending direction from $\mathbf{x}$ to its nearest model. Setting $\bigtriangledown \mathbf{p} = \mathbf{0}$ amounts to setting Term B to 0, suggesting the following iterative formula for updating:

$$\mathbf{x}_{(t+1)} \;:=\; \frac{\sum_{i=1}^{n} \mathbf{x}_i \dot{K}\left(\frac{\mathbf{x}_{(t)}-\mathbf{x}_i}{h}\right)}{\sum_{i=1}^{n} \dot{K}\left(\frac{\mathbf{x}_{(t)}-\mathbf{x}_i}{h}\right)}$$

at the t-th step. While other kernel choices can be used as well, the Gaussian kernel yields a simple form for the updating formula since $K(\cdot) \propto \dot{K}(\cdot)$ are proportional with extra constants cancelled out in the ratio. We have

$$\mathbf{x}_{(t+1)} \;:=\; \frac{\sum_{i=1}^{n} \mathbf{x}_i \exp\left(\frac{\|\mathbf{x}_{(t)}-\mathbf{x}_i\|^2}{2h^2}\right)}{\sum_{i=1}^{n} \exp\left(\frac{\|\mathbf{x}_{(t)}-\mathbf{x}_i\|^2}{2h^2}\right)}.$$

For clustering purpose, we need to find the nearest mode for each $\mathbf{x}_i$. Thus the above iteration starts at each $\mathbf{x}_i$. In high dimensions, the density estimation could only be done poorly. However, it turns out that observations can be accurately clustered even with a poorly estimated density.

Clustering can be used to group variables or columns. One popular distance measure for numerical variables is Pearson correlation coefficient. For categorical variables, we typical use the association measure as the distance between them.

Example 2. Suppose the correlation matrix is

$$\begin{bmatrix}
X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 & X_8 \\
1 & & & & & & & \\
.643 & 1 & & & & & & \\
-.103 & -.348 & 1 & & & & & \\
-.82 & -.086 & .100 & 1 & & & & \\
-.259 & -.260 & .435 & .034 & 1 & & & \\
-.152 & -.010 & .028 & -.288 & .176 & 1 & & \\
.045 & .211 & .115 & -.164 & -.019 & -.374 & 1 & \\
-.013 & -.328 & .005 & .486 & -.007 & -.561 & -.185 & 1
\end{bmatrix}$$

Use Single linkage and complete linkage to find the clusters.

3 Determining the Number of Clusters

No matter what clustering method is used, one must select the number K of clusters. The choice of K depends on the goal. Sometimes K is defined as part of the problem, i.e., market segmentation. However, more often K is estimated from the data. There have been many proposals for selecting K and there will perhaps be more. In particular, the R package **NbCluster** implements about thirty indices for determining the best number of clusters for hierarchical and k-means clustering.

Generally speaking, the issues really depends on the specific data sets. In scenarios where the true number of clusters manifests itself well, many methods agree each other well; on the other hand, in scenarios where there is no clear cut, these methods would disagree and eventually one has to resort to subjective judgment or the majority vote from all the indices. In the following, a few methods for selecting the number of clusters that are representative are presented. Recent developments in this direction are based on ℓ_1 regularization.

3.1 Plot of Within-Cluster Scatter

Given the results from a cluster analysis, a natural criterion for measuring its performance is the within-cluster point scatter $W(\mathcal{C})$, as given below

$$\begin{aligned} W(\mathcal{C}) &= \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k} \| \mathbf{x}_i - \mathbf{x}_{i'} \|^2 \\ &= \sum_{k=1}^{K} n_k \sum_{\mathcal{C}(i)=k} \| \mathbf{x}_i - \bar{\mathbf{x}}_{(k)} \|^2 \end{aligned}$$

where $\bar{\mathbf{x}}_{(k)}$ denotes the mean vector of the k-th cluster and n_k is the number of observations in the k-th cluster and the notation $\mathcal{C}(i) = k$ is equivalent to $i \in c_k$, meaning that the i-th observation has been clustered into the k-th cluster according to this cluster analysis $\mathcal{C}$. It is worth noting that $W(\mathcal{C})$ is slightly different from sum of squares within clusters (SSW) in MANOVA owing to the factor n_k.

The most common approach is to try a number of K values in $\{1, 2, \ldots, K_{\max}\}$. For each K, obtain the corresponding within-cluster error $W(\mathcal{C})$ values, denoted as $\{W_1, W_2, \ldots, W_{K_{\max}}\}$. Then plot them (or logarithm of W_K) versus each K. This graph is also referred to the *scree plot* (a term borrowed from principal components analysis PCA). It is worth noting that W_K always decreases with increasing K, as long as the clustering results $\mathcal{C}_K$ are nested into the $K-1$-th clustering results $\mathcal{C}_{K-1}$, as guaranteed in hierarchical clustering. In k-means clustering, this may not be the case – the clustering results for different K's are not nested. Nevertheless, $W(\mathcal{C})$ is still expected to have a general decreasing pattern.

A preferable choice of K is where a "kink" occurs, here meaning that there exhibits a sudden drop with $W_{K-1} - W_K$, but no dramatic decrease with $W_K - W_{K+1}$. See the figure on the right. The scree plot suggests the best choice of K is perhaps 3.

3.2 The Cubic Clustering Criterion (CCC) from SAS

The details of the cubic clustering criterion (CCC) are given in an SAS technical report (SAS Institute Inc., 1983). The main idea is outlined below.

3.2.1 R^2 in Cluster Analysis

First denote the total variation in data $SST(\mathcal{C}) = \sum_{i=1}^{n} \parallel \mathbf{x}_i - \bar{\mathbf{x}} \parallel^2$, which remains invariant to the choice of clustering $\mathcal{C}$. Further assume that the data matrix $\mathbf{X}$ has been standardized so that $\bar{\mathbf{x}} = \mathbf{0}$. Thus

$$SST(\mathcal{C}) = \sum_{i=1}^{n} \parallel \mathbf{x}_i \parallel^2 = \text{trace}\left(\mathbf{X}^T\mathbf{X}\right).$$

Introduce a cluster indicator matrix $\mathbf{Z} = (z_{ik}) \in \mathbb{R}^{n\times K}$ such that $z_{ik} = 1$ if the ith observation belongs to the k-th cluster and 0 otherwise. Then it can be verified that the within-cluster variation can be written as

$$SSW(\mathcal{C}) = \text{trace}\left\{\mathbf{X}^T\mathbf{X} - \mathbf{X}^T\mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{X}\right\} = \text{tr}\left\{\mathbf{X}^T\mathcal{P}_{\mathbb{C}(\mathbf{Z})^\perp}\mathbf{X}\right\},$$

noting that $\mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T = \mathcal{P}_{\mathbb{C}(\mathbf{Z})}$ is the projection matrix on the column space of $\mathbf{Z}$, $\mathbb{C}(\mathbf{Z})$ and $\mathbb{C}(\mathbf{Z})^\perp$ denotes the subspace that is perpendicular to $\mathbb{C}(\mathbf{Z})$.

The R^2 can be defined accordingly

$$R^2 = 1 - \frac{SSW(C)}{SST(C)} = 1 - \frac{\text{tr}\left(\mathbf{X}^T\mathcal{P}_{\mathbb{C}(\mathbf{Z})^\perp}\mathbf{X}\right)}{\text{tr}\left(\mathbf{X}^T\mathbf{X}\right)}$$

and carries the usual interpretation of the proportion of total variation in $\mathbf{X}$ that can be accounted for by the clusters. Note that a smaller $SSW(C)$ corresponds to a larger R^2, again because $SST(C)$ is invariant to clustering.

To compute R^2, one simply regresses $\text{vec}(\mathbf{X})$ on $\mathbf{Z}\otimes\mathbf{I}_p$ and obtain its coefficient of determination. The operator $\text{vec}(\mathbf{X})$ stacks the columns of $\mathbf{X}$ to form a $np \times 1$ vector. The $\otimes$ sign denotes the Kronecker product; thus $\mathbf{Z}\otimes\mathbf{I}_p = \text{diag}[\mathbf{Z}, \ldots, \mathbf{Z}]$ is of dimension $np \times Kp$.

3.2.2 The Null Hypothesis for Cluster Analysis

The main idea of CCC is to compare the observed R^2 with its expected value under the null hypothesis of no cluster structure. Now it comes to the question of what constitutes a proper null setting for cluster analysis. A natural choice would be that the data have been sampled from a uniform distribution on a p-dimensional hypercube. This null setting is also used in the gap statistic of Tibshirani et al. (2001). There are a couple of alternative null settings. One is that all permutations of the distance values in the distance matrix are equally likely. Under this null setting, a permutation test or a rank test seemingly could be used to select K. Another is to assume that data are randomly drawn from a multivariate normal distribution. However, neither is found satisfactory in practical studies.

The CCC advocates to compute the expected R^2 via approximations based on the uniformly distributed hypercube, while the alternative is that the data have been sampled from a mixture of spherical multivariate normal distribution with equal variances and equal sampling probabilities. A large positive value of the CCC implies that the obtained R^2 is greater than what would be expected if sampling from a uniform distribution and therefore indicates the possible presence of the clustering structure.

Some basic reasoning is described below. Let r_j be the edge length of the hypercube along the j-th dimension. One choice for r_j would be the range of X_j. But CCC uses the square root of the

j-th eigenvalue of the sample variance-covariance matrix $\widehat{\mathbf{\Sigma}} = \mathbf{X}^T\mathbf{X}/(n-1)$ given data matrix $\mathbf{X}$, so that under the null hypothesis, the length of hyperbox in the jth dimension is proportional to the standard deviation of the jth principal component of the data, denoted as PC_j. Assume further that r_j's are arranged in the descending order.

The volume of the hypercube is $V = \prod_{j=1}^{p} r_j$. If the hyper-box is divided into K hyper-cubes with equal length l, then we have

$$l = (V/K)^{1/p}.$$

Let $u_j = r_j/l$ be the number of hypercubes along PC_j. Note that the total-sample variation for PC_j is proportional to r_j^2, while the within-cluster variation for PC_j is proportional to l^2. Thus $R^2 = 1 - \sum_{j=1}^{p} l^2 / \sum_{j=1}^{p} r_j^2 = 1 - p/\sum_{j=1}^{p} u_j^2$.

In order to have a better approximation and deal with the scenario when $K \leq p$, it is further assumed that the clusters are hyperboxes with edge length l in the first p' dimensions, and edge length r_j in the remaining dimensions. Then a heuristic small-sample approximation for the expected values of R^2 is derived and the CCC measure is obtained after some further variance stabilizing transformation.

The specific steps for computing the recommended CCC are list below.

(i) Choose p' to be largest integer less than K such that $u_{p'} = r_{p'}/l$ is not less than l. Compute $V' = \prod_{j=1}^{p'} r_j$ and $l' = (V'/K)^{1/p'}$;

(ii) Compute an approximation to the population

$$R^2 \approx 1 - \frac{p' + \sum_{j=p'+1}^{p} u_j^2}{\sum_{j=1}^{p} u_j^2}.$$

(iii) Obtain the approximation for its expected value

$$E(R^2) \approx 1 - \frac{\sum_{j=1}^{p'} \frac{1}{n+u_j} + \sum_{j=p'}^{p} \frac{u_j^2}{n+u_j}}{\sum_{j=1}^{p} u_j^2} \cdot \frac{(n-q)^2}{n} \cdot \frac{n+4}{n}.$$

(iv) Finally, CCC is computed as

$$\text{CCC} = \ln\left\{\frac{1 - E(R^2)}{1 - R^2}\right\} \cdot \frac{\sqrt{np'/2}}{\{0.001 + E(R^2)\}^{1.2}}.$$

To use in practice, one seeks the maximum CCC value among those greater than 2 or 3.

3.3 The GAP Statistic

The GAP statistic of Tibshirani et al. (2001) is based on similar ideas of CCC but resorts to simulation in computing the expected value. One benefit of using simulation is easy availability of precision measures (e.g., standard errors). GAP uses $\log(W_K)$, by comparing its empirical curve with different K to the reference curve obtained from the uniformly distributed data over a hyperbox, the null hypothesis used in the derivation of CCC. The null reference curve is obtained via simulation, i.e., by averaging $\log(W_K)$ values from B simulated samples. The GAP statistic tries to estimate the optimal number of clusters to be the place where the gap between the two curves is largest.

Two choices for the reference distribution or data $\mathbf{X}'$ were suggested in Tibshirani, Walther, and Hasite (2001):

(a). Generate $\mathbf{X}'$ by simulating each reference variable X'_j uniformaly over the range of the observed X_j in $\mathbf{X}$;

(b). Generate data based on principal components of the data. Recall the SVD of matrix $\mathbf{X} = \mathbf{UDV}^T$. The principal components of data are given by $\mathbf{Z} = \mathbf{XV} = \mathbf{UD}$. Draw features $\mathbf{Z}'$ uniformly over the range of columns in $\mathbf{Z}$ (column by column). Then transform back to form $\mathbf{X}' = \mathbf{Z}'\mathbf{V}^T$.

Points on the GAP curve (e.g., Figure 4) are basically the difference of the expected $\log(W_K)$ curve minus the observed curve. Denote them by G_K. Let s_K denote the sample standard deviation of the $\log(W_K)$ values from simulation; then $s'_K = s_K\sqrt{1+1/B}$ is the standard error of G_K, where B is the total number of simulated datasets from the null distribution. This is because, under the null hypothesis, s_K is an estimate of standard deviation of $\log(W_K)$ and the averaged $\log(W_K)$ over B independently simulated datasets has standard error s_K/B. A heuristic 1-SE rule is suggested for choosing the optimal number of clusters $K^\star$:

$$K^\star = \arg\min_K \left\{K \,|\, G_K \geq G_{K+1} - s'_{K+1}\right\}.$$

That is, $K^\star$, is the smallest K that produces a gap within one standard error of the gap at $K+1$. An alternative way of applying the 1-SE rule is to choose the smallest K with G_K falling above $G^\star - s^\star$, where $G' = \max_K G_K$ and $s^\star$ is its associated SE.

The following illustration in Figure 4 is taken from Hastie, Tibshirani, and Friedman (2009, Figure 14.11).

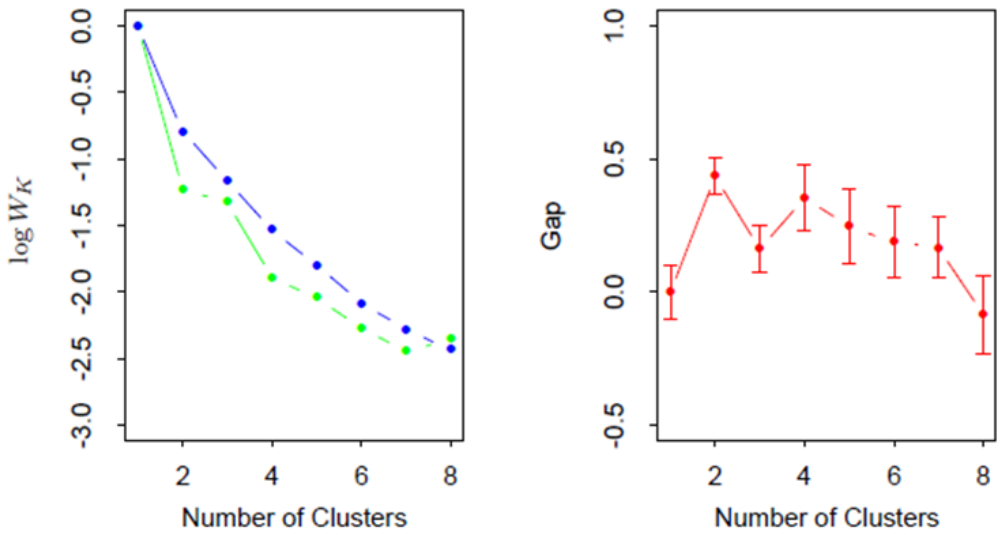

Figure 4: (Left Panel): observed (green) and expected (blue) values of $\log W_K$ for some simulated data. Both curves have been translated to equal zero at one cluster. (Right Panel): Gap curve, equal to the difference between the observed and expected values of $\log W_K$. The GAP estimate $K^\star$ is the smallest K producing a gap within one standard deviation of the gap at $K+1$; here $K^\star = 2$.

3.4 Stability

Stability has long been used for selecting the optimal number of clusters. Stability compares the group memberships provided by two clustering algorithms or same algorithms yet with different options (e.g., K – the number of clusters). Given any pair of observations (i, i') and two clustering results $(\mathcal{C}, \mathcal{C}')$, let $\mathcal{C}(i)$ denote the cluster assigned to observation i by the clustering algorithm $\mathcal{C}$, similar definitions applied to $\mathcal{C}(i')$, $\mathcal{C}'(i)$, and $\mathcal{C}'(i')$. To measures the disagreement in grouping between $\mathcal{C}$ and $\mathcal{C}'$, one nature way is to consider whether $\mathcal{C}$ groups observations (i, i') into the same or different clusters if $\mathcal{C}'$ has grouped them into the same/different clusters. More formally, we consider the quantity

$$I\{\mathcal{C}(i) = \mathcal{C}(i')\} + I\{\mathcal{C}'(i) = \mathcal{C}'(i')\},$$

which equals 1 if $\mathcal{C}$ and $\mathcal{C}'$ do not agree with each other and equals 0 or 2 if they agree. Thus, it is convenient to use

$$\alpha(i, i') = I\left[I\{\mathcal{C}(i) = \mathcal{C}(i')\} + I\{\mathcal{C}'(i) = \mathcal{C}'(i')\} \ = \ 1\right].$$

The stability of one clustering algorithm $\mathcal{C}$ may be defined as the average $\alpha(i, i')$ when $\mathcal{C}$ is compared to its own variations when applied to a number of perturbed samples. Several methods for executing the estimation of stability are available. Below is an algorithm based on permutation and cross-validation in Wang (2010).

Suppose we are using one type of clustering algorithm and want to determine K, the best number of clusters. Two ways for determining the final $K^\star$ are suggest: the first is to choose $K^\star$ to be the

Algorithm 4 Permutation-Based Stability for Selecting the Number of Clusters

for $b = 1$ **to** B **do**
  Permute original sample $\mathcal{L}$ to obtain $\mathcal{L}^{(b)}$;
  Partition $\mathcal{L}^{(b)}$ into three parts $\left\{\mathcal{L}_1^{(b)}, \mathcal{L}_2^{(b)}, \mathcal{L}_3^{(b)}\right\}$;
  for $K = 1$ **to** $K_{\max}$ **do**
    Train clustering $\mathcal{C}$ using $\mathcal{L}_1^{(b)}$ with K clusters;
    Train clustering $\mathcal{C}'$ using $\mathcal{L}_2^{(b)}$ with K clusters;
    Applying both $\mathcal{C}$ and $\mathcal{C}'$ to $\mathcal{L}_3^{(b)}$, compute the stability measure as $\tau_k^{(b)} = \sum_{i<i' \in \mathcal{L}_3^{(b)}} \alpha(i, i')$.
  end for
  Let $K^{(b)}$ be the k that corresponds to the smallest $\tau_k^{(b)}$;
end for
Compute $\tau_k = \sum_{b=1}^{B} \tau_k^{(b)}/B$;

mode of $\{K^{(1)}, K^{(2)}, \ldots, K^{(B)}\}$; the second is to choose $K^\star$ with the smallest τ_k.

Stability works well empirically. However, asymptotically stability is fully determined by the behavior of the objective function on which the clustering algorithm is formulated (Ben-David, von Luxburg, and Pal, 2006). Any clustering algorithm is asymptotically stable as long as it can be formulated as an optimization problem with a certain objective function that has a unique global minimizer; otherwise it is unstable. It has been shown that the instability measure of K-means clustering with various Ks converges to zero as sample size diverges. Therefore, they conclude stability is not a well-suited tool to determine the number of clusters – it is determined by the symmetries of the data which may be unrelated to clustering parameters. Despite these negative results about stability, Wang (2010) argued that although the instability measures may converge to

zero, the rates of convergence can behave differently when different numbers of clusters are specified. Therefore, stability is still used as a selection criterion for K.

4 Notes and Further Reading

Partly because of its unsupervised nature, there are many open or controversial issues in cluster analysis. Concerning its definition, what really constitutes clusters is subjective to opinions. From statistical point of view, it is quite natural to consider clusters as points that gather around modes of the probability density function. On the other hand, sometimes all that is available is the distance matrix. Thus the perspective offered from connected subgraphs is also intuitive for distance-based clustering method. We have seen that single linkage method works well in picking up chain-like peculiar shapes. Nevertheless, Hartigan (1981) showed that single linkage is not consistent in sense that it fails to pick up high-density clusters. The key reason is that the single linkage method amounts to a mode-seeking clustering method which, however, sets $k = 1$ in a KNN (K nearest neighbors) for density estimation. Hartigan (1981) is among the first few theoretical work for cluster analysis and sets up a general framework for many follow-up researches.

There is a huge literature in cluster analysis, making it rather difficult for one to keep up with. Nearly every method has numerous variants. For example, it has been found that the K-means method is rather sensitive to its initial starting or seeding point. To improve, Arthur and Vassilvitskii (2007) suggested the so-called 'K-means++' method, where the seeding points are made as distant from each other as possible, and demonstrated with simulation that K-Means++ achieves faster convergence to a lower within-cluster scatter than the original K-Means method.

Many a methods are proposed for selecting the optimal number of clusters. We have discussed approaches based on formal statistical significance testing or confidence estimation of a performance measure by referring to some assumed null settings and those proposal based on stability. Nevertheless, at the end of the day, many times it may be still very difficult to make a decision. In R package **NbCluster**, the majority voting from these methods is summarized as an ultimate resolution. It is always advisable to resort to a field expert so that the number of clusters is determined on the basis of the cluster interpretation.

References

Arthur, D. and Vassilvitskii, S. (2007). K-means++: The Advantages of Careful Seeding. SODA 07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027-1035.

Ben-David, S., von Luxburg, U., and Pal, D. (2006). A sober look at clustering stability. COLT'06 Proceedings of the 19th annual conference on Learning Theory, pp.5-19

Berry, M. J. A. and Gordon, L. S. (2000) Chapter 5 of *Mastering Data Mining*, John Wiley & Sons, Inc.: New York, New York.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(8): 790–799.

Desgraupes, B. (2013). Clustering Indices. URL https://cran.r-project.org/web/packages/clusterCrit/vignettes/clusterCrit.pdf.

Fiedler, M. (1989). Laplacian of graphs and algebraic connectivity, *Combinatorics and Graph Theory*, **25**: 57–70.

Hartigan, J. A. (1981). Consistency of Single Linkage for High-Density Clusters. *Journal of the American Statistical Association*, **76**: 388–394.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*, 2nd Edition. Springer.

Jognson, R. A. and Wichern, D. W. (1982). Chapter 11 of *Applied Multivariate Statistical Analysis*, Prentice-Hall, Inc.: Englewood Cliffs, New Jersey.

Lloyd, S. P. (1982). Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, **28**: 129-137.

Meila, M. (2007). Comparing clusterings – An information based distance. *Journal of Multivariate Analysis*, **98**: 873–895.

Rud, O. P. (2001). *Data Mining Cook Book*, John Wiley & Sons, Inc.: New York, N.Y.

SAS Institute Inc. (1983). SAS Technical Report A-108, *Cubic Clustering Criterion*. Cary, NC: SAS Institute Inc. 56pp. URL http://support.sas.com/documentation/onlinedoc/v82/techreport_a108.pdf

Tibshirani, R., Walther, G., and Hasite, T. (2001). Estimating the number of clusters in data set via the gap statistic. *Journal of the Royal Statistical Society, Series B*, **63**: 411-423.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, **17**: 395–416.

Wang, J. H. (2010). Consistent selection of the number of clusters via cross validation. *Biometrika*, **97**(4): 893-904.

Wong, M. A. and Lane, T. (1983), A K-th Nearest Neighbor Clustering Procedure. *Journal of the Royal Statistical Society, Series B*, **45**: 362–368.

APPENDIX

A R Implementation for Cluster Analysis

The CRAN Task View maintains a website for all R packages designed for cluster analysis:
http://cran.cnr.berkeley.edu/web/views/Cluster.html.
In particular, some quick notes are taken below.

- Functions `hclust()` from package **stats** and `agnes()` from the **cluster** package are the primary functions for agglomerative hierarchical clustering, `functiondiana()` can be used for divisive hierarchical clustering. Faster alternatives to `hclust()` are provided by the packages **fastcluster** and **flashClust**.
- Function `dendrogram()` from **stats** and associated methods can be used for improved visualization for cluster dendrograms.
- Function `kmeans()` from package **stats** provides several algorithms for computing partitions with respect to Euclidean distance.
- Function `pam()` from package **cluster** implements partitioning around medoids and can work with arbitrary distances. Function `clara()` is a wrapper to `pam()` for larger data sets. Silhouette plots and spanning ellipses can be used for visualization.
- Functionality to compare the similarity between two cluster solutions is provided by `cluster.stats()` in package **fpc**.
- Package **kernlab** provides a weighted kernel version of the k-means algorithm by `kkmeans` and spectral clustering by `specc`.
- To compute the gap statistic, either function `gap` in **SAGx**, `nps.plot()` in **popgen**, or `clusGap` in **cluster**.
- Package **NbCluster** determines the best number of clusters for hierarchical and k-means clustering. Thirty indices (including the CCC criterion) are provided.

B Comparison of Different Clustering Methods

One common practice in clustering is to apply different clustering methods to toy datasets that are designed to have various shapes and see if they are able to successfully uncover the cluster structures. In the following illustration, the datasets are generated by using functions from R package **mlbench**. Several methods are applied: hierarchical clustering with single, complete, average linkages, Ward's method, K-means, and spectral clustering.

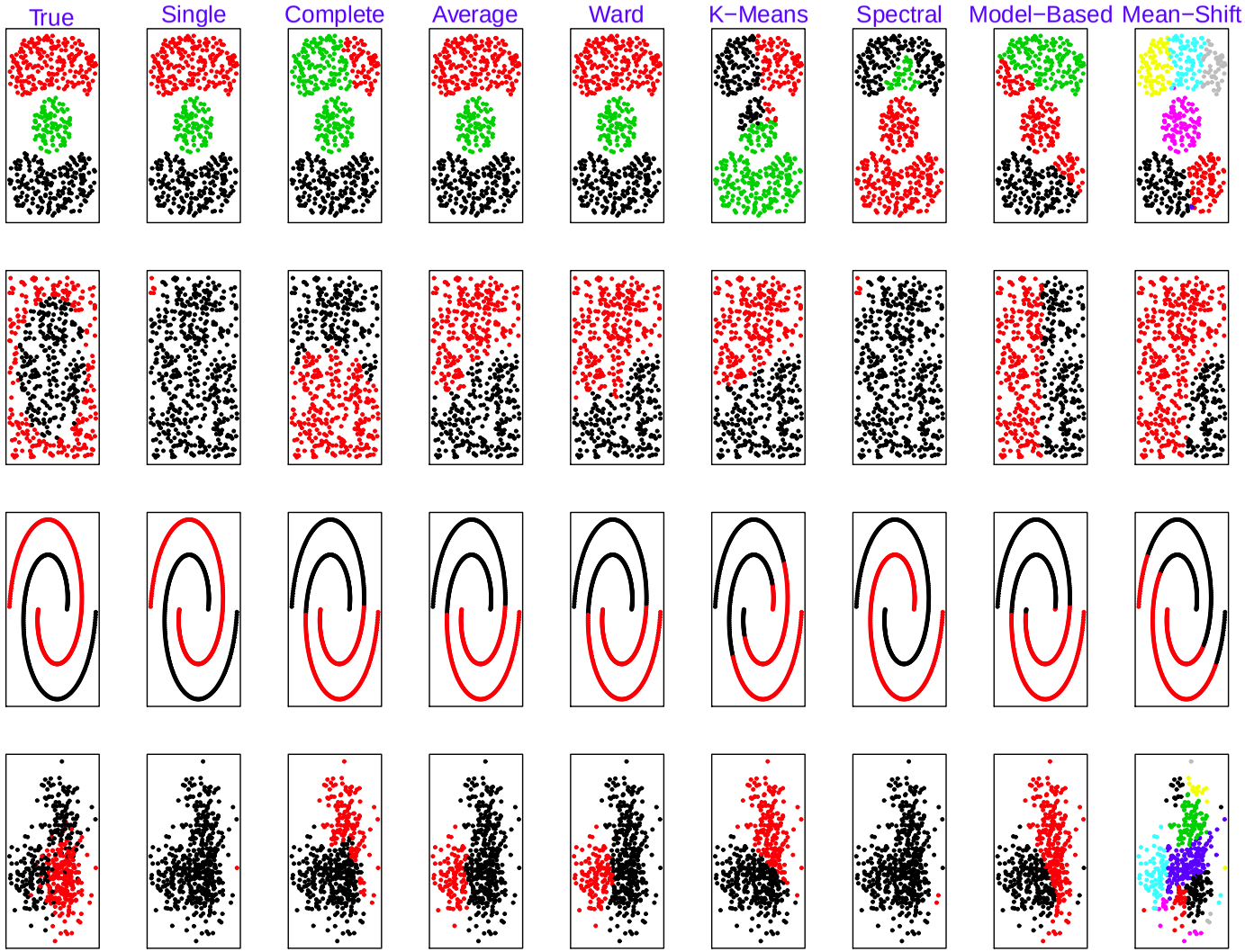

Figure 5: Comparing Different Clustering Methods.