Chapter 5: Cluster Analysis[a]

The goal of cluster analysis is to group the observations $\{x_i^0\}_{i=1}^n$ into different clusters, so that observations belonging to the same cluster are more similar than those belonging to different clusters.

The first step of cluster analysis is therefore to measure the similarity, or alternatively the distance, between the observations. This step is crucial since different measures of similarities often lead to a different clustering of the data.

This chapter therefore naturally starts by introducing various measures of similarity and of distance between observations.

We will then introduce three popular clustering approaches, namely agglomerative clustering, K-means clustering and spectral clustering.

[a]The main references for this chapter are [3, Section 14.3] and [9].

Distance and similarity measures

- A measure of distance d_{il} between observations x_i^0 and x_l^0 is such that
  - $d_{il} \geq 0$ (non-negativity);
  - $d_{il} = 0$ if and only if $x_i^0 = x_l^0$ (identity of indiscernible);
  - $d_{il} = d_{li}$ (symmetry).

  Remark: If $d_{il} \leq d_{is} + d_{sl}$ (triangle inequality) then $d_{..}$ is said to be a metric (i.e. a distance function in the mathematical sense).

- A measure of similarity s_{il} between observations x_i^0 and x_l^0 is such that
  - $s_{il} \in [0, 1]$,
  - $s_{il} = 1$ if and only if $x_i^0 = x_l^0$,
  - $s_{il} = s_{li}$,
  - s_{il} is close to zero if x_i^0 and x_l^0 are "very different".

A similarity s_{il} can be transformed into a bounded distance d_{il} using e.g. $d_{il} = 1 - s_{il}$ or to an unbounded distance using e.g. $d_{il} = -\log(s_{il})$.

Conversely, a distance d_{il} can be transformed into a similarity s_{il} using e.g. $s_{il} = \exp(-d_{il}^2/c^2)$ for some $c \in (0, \infty)$ or $s_{il} = 1 - (d_{il}/c)^\lambda$ for some $\lambda \geq 1$ and c such that $\max\{d_{il}\}_{i,l=1}^n \leq c < \infty$.

Quantitative variables and categorial variables

When all the variables are quantitative a possible measure of distance between observations x_i^0 and x_l^0 is the Minkowski distance:

$$d_{il} = \Big(\sum_{j=1}^{p} |x_{ij}^0 - x_{il}^0|^\lambda \Big)^{1/\lambda}, \quad \lambda \in \mathbb{N} \cup \{\infty\}$$

with the convention $d_{il} = \max\{|x_{ij}^0 - x_{lj}^0|\}_{j=1}^p$ (supremum distance) when $\lambda = \infty$.

When x_i^0 and x_l^0 are vectors in $\{0,1\}^p$ (i.e. the observations are categorial variables) we usually first compute a similarity measure s_{il} between these two observations, which is then used to obtain a measure of distance d_{il} (as discussed above).

For categorial data, the similarity measure s_{il} is typically based on the association table between x_i^0 and x_l^0.

Letting

- a be the number of attributes present in both x_i^0 and x_l^0,
- b be the number of attributes present in x_i^0 but not in x_l^0,
- c be the number of attributes present in x_l^0 but not in x_i^0,
- d be the number of attributes absent in both x_i^0 and x_l^0,

the association table between observations x_i^0 and x_l^0 is given by

		Obs. x_l^0	
		1	0
Obs. x_i^0	1	a	b
	0	c	d

Standard similarity measures for categorical variables

Two popular measures of similarity that can be computed from an association table are

- The simple matching coefficient, defined by

$$s_{il} = \frac{a+d}{a+b+c+d}$$

and which is the proportion of attributes that are the same in both x_i^0 and x_l^0 (either both present or both absent).

- The Jaccard's coefficient, defined by

$$s_{il} = \frac{a}{a+b+c}$$

and which is the proportion of attributes present in both x_i^0 and x_l^0 out of all the attributes that are present in at least one of these two observations.

Computing similarities between categorial data: Example

Observations (matrix $\boldsymbol{X}^0$)

obs.\attributes	1	2	3	4	5	6	7
L=lion	1	1	0	1	0	1	0
G=giraffe	1	1	0	0	1	1	0
H=human	0	0	0	1	0	0	0
S=sheep	1	0	1	0	0	1	1

Attributes

1) has a tail,
2) is a wild animal,
3) is a farm animal,
4) eats other animals,
5) has long neck,
6) walks on four legs,
7) provides clothing material w/o being killed.

The association table e.g. between Lion and Giraffe and between the Lion and Human are

L \ G	1	0
1	3	1
0	1	2

L \ H	1	0
1	1	3
0	0	3

and the simple matching and Jaccard's coefficients are

Simple matching s_{il}

	L	G	H	S
L	1	5/7	4/7	3/7
G		1	2/7	3/7
H			1	2/7
S				1

Jaccard's s_{il}

	L	G	H	S
L	1	3/5	1/4	2/6
G		1	0/5	2/6
H			1	0/5
S				1

Table 5.1: Similarity measures between some animals.

General measures of similarity/distance

In practice variables of different types are often gathered (e.g. some components of x_i^0 may be categorical and others may be continuous) and, in order to measure the distance or the similarity between the observations, it is necessary to find a way to combine them in a meaningful way.

One approach to combine different measures of similarity consists in using Gower's general (combined) coefficient of similarity, defined by

$$s_{il} = \frac{\sum_{j=1}^{p} w_j s_{j,il}}{\sum_{j=1}^{p} w_j}$$

where $w_j \in [0, 1]$ is a weight indicating the importance of the jth variable and where $s_{j,il}$ is the similarity between x_{ij}^0 and x_{lj}^0.

Gower's coefficient is usually not applied directly on the distances $d_{j,il}$ between x_{ij}^0 and x_{lj}^0 because, the measures of distance being typically unbounded, a single $d_{j,il}$ can have a massive impact on the value of Gower's coefficient.

Therefore, if we work with distances instead of similarities, we should first transform the former into similarities (see above) and then combine the resulting similarities using Gower's coefficient.

Remark: The distances should be on an appropriate scale before being converted into similarities.

Agglomerative clustering

The general principle of agglomerative clustering, which belongs to the class of hierarchical clustering methods, is as follows:

Step 1: Start with n clusters containing only one individual each; define the distances d_{il} between each pair of observations x_i^0 and x_l^0.

Step 2: Merge the two closest clusters into one new cluster.

Step 3: Compute the distances between the remaining clusters and the new cluster obtained in Step 2.

Step 4: Repeat Steps 2–3 until all individuals are in the same cluster.

Step 5 Use a dendrogram to choose the final set of clusters (see below).

Remark: The various agglomerative methods differ primarily with respect to how the distance between two clusters (not between individual observations) are defined in Step 3.

In what follows we let $d_{t,(r,s)}$ be the distance between an old cluster t and the new cluster (r, s) obtained in Step 2 by merging clusters r and s.

The Lance-Williams recursive formula for computing $d_{t,(r,s)}$

A general recursive formula (due to Lance-Williams) for computing $d_{t,(r,s)}$ is

$$d_{t,(r,s)} = \alpha_r d_{rt} + \alpha_s d_{st} + \beta d_{rs} + \gamma |d_{rt} - d_{st}| \tag{5.1}$$

for some real numbers α_r, α_s, β and γ.

The following table gives the values of $\alpha_r, \alpha_s, \beta, \gamma$ for some common methods (with n_r and n_s the size of the clusters r and s):

	α_r	α_s	β	γ
single linkage	1/2	1/2	0	-1/2
complete linkage	1/2	1/2	0	1/2
group average	$\frac{n_r}{n_r+n_s}$	$\frac{n_s}{n_r+n_s}$	0	0
centroid	$\frac{n_r}{n_r+n_s}$	$\frac{n_s}{n_r+n_s}$	$-\frac{n_r n_s}{(n_r+n_s)^2}$	0
median	1/2	1/2	-1/4	0

The following proposition gives a simpler expression for $d_{t,(r,s)}$ under single and complete linkage.

Proposition 5.1 *Single linkage is equivalent to* $d_{t,(r,s)} = \min\{d_{tr}, d_{ts}\}$ *while complete linkage is equivalent to* $d_{t,(r,s)} = \max\{d_{tr}, d_{ts}\}$.

Proof. Plugging in (5.1) the parameters from the table gives, for single linkage,

$$d_{t,(r,s)} = \frac{1}{2}d_{rt} + \frac{1}{2}d_{st} - \frac{1}{2}|d_{rt} - d_{st}| = \begin{cases} d_{rt} & \text{when } d_{rt} \le d_{st}, \\ d_{st} & \text{when } d_{rt} > d_{st}. \end{cases} = \min\{d_{rt}, d_{st}\}$$

and, for complete linkage,

$$d_{t,(r,s)} = \frac{1}{2}d_{rt} + \frac{1}{2}d_{st} + \frac{1}{2}|d_{rt} - d_{st}| = \begin{cases} d_{rt} & \text{when } d_{rt} > d_{st}, \\ d_{st} & \text{when } d_{rt} \le d_{st}. \end{cases} = \max\{d_{rt}, d_{st}\}.$$

□

Continuation of the last example

We let $d_{il} = 1 - s_{il}$ be the distance between observations x_i^0 and x_l^0 induced by their similarity measures.

Then, if s_{il} is the single matching similarity (see Table 5.1), we obtain the following distances:

Simple matching d_{il}

	L	G	H	S
L	0	0.29	0.43	0.57
G		0	0.71	0.57
H			0	0.71

and the resulting clustering algorithm yields:

1. We merge L and G to (L,G).
2. We use single linkage to compute the distances between the remaining individuals and the new cluster (L,G), which yields:

Single linkage $d_{t,(r,s)}$

	(L,G)	H	S
(L,G)	0	0.43	0.57
H		0	0.71

3. We merge (L,G) with H.
4. We use single linkage to compute the distance between S and (L,G,H), which is 0.57.
5. We merge (L,G,H) with S to obtain the final cluster (L,G,H,S).

Dendrogram

The agglomerative clustering process is often visualized using a dendrogram.

The dendrogram is a plot where the x-axis shows the observations to be clustered (the order is arbitrary) and where the y-axis shows the distance at which a merge takes place.

For the previous example, the corresponding dendrogram is as follows:

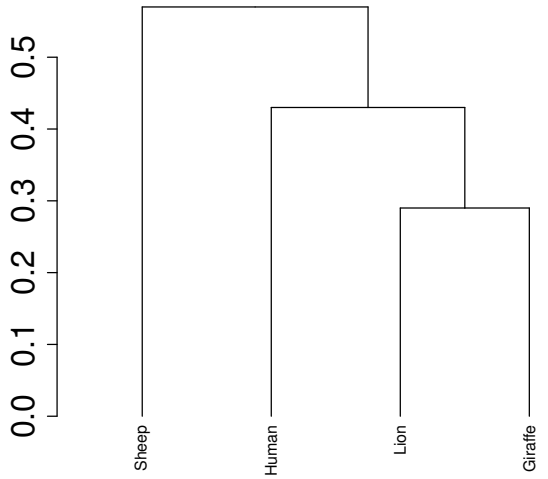

Figure 5.1: Dendrogram for the animals example.

Remark: The dendrogram is particularly convenient to interpret when complete linkage is used. Indeed, in this case the y-axis gives the maximum distance between two observations that belong to the same cluster. Conversely, given a chosen maximum distance that two observations belonging in the same cluster can have, the dendrogram allows to directly identify the corresponding clustering of the data.

Agglomerative clustering: The mtcars dataset

For the mtcars dataset, already considered in Chapter 1, all the variables are quantitative and therefore the distances between the observations are computed using the Euclidean distance. Since the different variables are not all expressed in the same unit they are standardized before computing the distances.

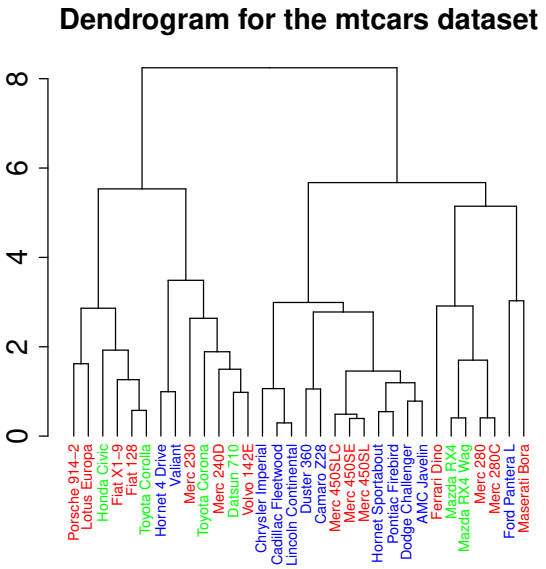

Figure 5.2: Dendrogram for the mtcars dataset (complete linkage).

From Figure 5.2 we see that if, for instance, we the let 4 be the maximum distance between two observations belonging in the same cluster then five clusters should be used. Remark that the clustering of the data given in Figure 5.2 is "coherent" with the 2 dimensional reduction of the mtcars dataset shown in Figure 1.7.

Remark: One way to choose the number of cluster is to identify an obvious stretch where to "cut through" the dendrogram. Based on this idea, we see from Figure 5.2 that taking two or five clusters seems appropriate for the mtcars dataset.

K-means clustering

The objective of K-means clustering (which belongs to the class of partitional clustering methods) is to find a partition of the sample into K sets $C'_1, \dots, C'_K$ such that the sum of the squared within-cluster distances to the cluster means are minimised.

Formally, letting $\mathcal{S}_K$ be the set containing all the partitions of $\{1, \dots, n\}$ into K non-empty sets, the goal of K-means clustering is to compute

$$\{C_k\}_{k=1}^K \in \underset{\{C'_k\}_{k=1}^K \in \mathcal{S}_K}{\operatorname{argmin}} \sum_{k=1}^K \sum_{i \in C'_k} \|x_i^0 - c'_k\|^2, \tag{5.2}$$

where $c'_k = |C'_k|^{-1} \sum_{i \in C'_k} x_i^0$ is the mean of the kth cluster.

The cost for solving exactly (5.2) is of order $\mathcal{O}(n^{pK+1})$ [5] and, consequently, computing $\{C_k\}_{k=1}^K$ in an intractable task.

In practice, K-means clustering therefore relies on the use of a heuristic algorithm to solve (5.2), that converges quickly to a local minimum of the mapping

$$\mathcal{S}_K \ni \{C'_k\}_{k=1}^K \mapsto \sum_{k=1}^K \sum_{i \in C'_k} \|x_i^0 - c'_k\|^2.$$

Remark: K-means clustering measures the distance between an observation and a cluster mean using the Euclidean norm (or any other norm on $\mathbb{R}^p$) and as such is mainly intended to situations where each component of x_i^0 is a quantitative variable.

Lloyds's K-means clustering algorithm

Input: Integer $K \geq 2$ and real number $\epsilon > 0$.

1: Sample $(i_1, \ldots, i_K)$ without replacement from the set $\{1, \ldots, n\}$.

2: Let $c'_k = x^0_{i_k}$ and $C'_k = \{i_k\}$ for all $k = 1, \ldots, K$.

3: Let $\rho_0 = 0$ and $\rho_1 = \sum_{k=1}^{K} \sum_{i \in C'_k} \|x^0_i - c'_k\|^2$.

while $|\rho_0 - \rho_1| \leq \epsilon$ **do**

4: Let $k_i \in \operatorname{argmin}_{k \in \{1,\ldots,K\}} \|x^0_i - c'_k\|^2$ for $i = 1, \ldots, n$,

5: Let $C'_k = \bigcup_{i:k_i=k} \{k_i\}$ and $c'_k = \frac{1}{|C'_k|} \sum_{i \in C'_k} x^0_i$ for $k = 1, \ldots, K$.

6: Let $\rho_0 = \rho_1$ and $\rho_1 = \sum_{k=1}^{K} \sum_{i \in C'_k} \|x^0_i - c'_k\|^2$.

end while

Return: Partition $\{C'_k\}_{k=1}^{K}$ of $\{1, \ldots, n\}$.

Remark: The output of this algorithm depends on the centres chosen at the beginning. It is therefore common to run the algorithm several times and assess to what extend its output depends on that choice.

Remark: K-means clustering is based on minimising the (sum of) distances $\|x^0_i - c'_k\|$, and therefore the distances between the data points must be 'meaningful'. In particular, all the variables should be either measured in the same unit or standardized.

Choosing the number K of clusters

It should be clear that the squared distances of the observations to their clusters means can only get smaller if we increase the number of clusters.

In particular, if we let $K = n$ in the above algorithm then the value of ρ_1 calculated on its Line 3 is zero.

Therefore, it is not possible to try different values for K and to choose this parameter based on the last value of ρ_1 computed by a K-means algorithm.

A common approach for choosing K is the "knee finding" (or "elbow finding") approach. Denoting by $C_1^{(K)}, \ldots, C_K^{(K)}$ the clusters returned by a K-means algorithm, this approach amounts to plotting the function

$$K' \mapsto \sum_{k=1}^{K'} \sum_{i \in C_k^{(K')}} \|x_i^0 - c_k\|^2 \tag{5.3}$$

and to choose a K for which the slope of this function changes abruptly.

K-means clustering: The mtcars Dataset

The left plot of Figure 5.3 shows the function defined in (5.3) for the mtcars dataset[a]. From this plot we see that the function has two "knees", namely one at $K = 2$ and one at $K = 5$ (which is coherent with what we observe in Figure 5.2, obtained with the agglomerative clustering method).

We choose $K = 5$ and the resulting clustering is represented in the right plot of Figure 5.3, where the observations are mapped into a two-dimensional space using the first two principal components.

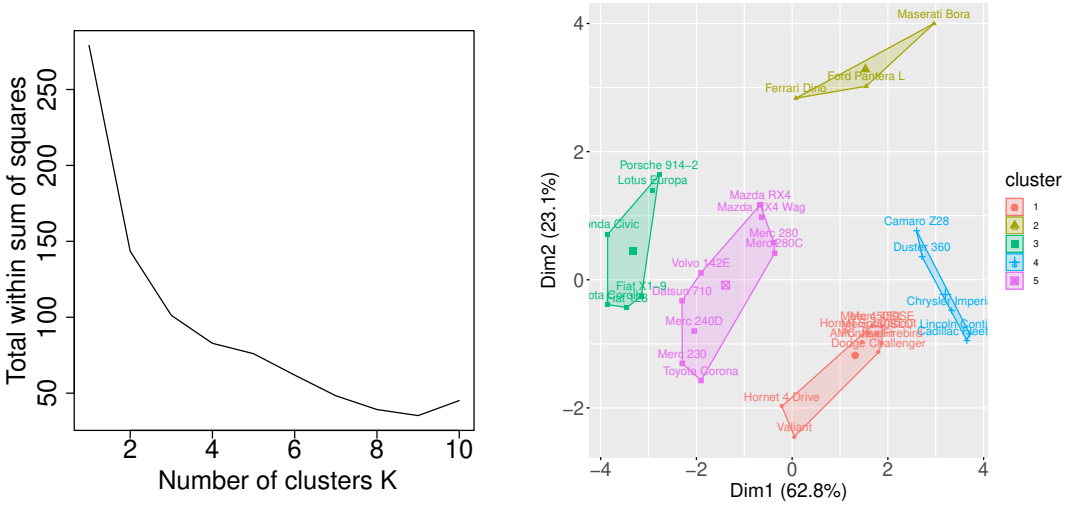

Figure 5.3: K-means clustering of the mtcars dataset.

We remark that if we want to split the data into five clusters then the clustering of the observations obtained with K-means clustering is very similar to that obtained with the agglomerative clustering method (see Figure 5.2).

[a]The variables in this dataset being not all expressed in the same unit they are standardized before computing the distances.

Spectral Clustering

In K-means clustering the cluster C'_k an observations x_i^0 belongs to is such that

$$k = \underset{j\in\{1,\dots,K\}}{\operatorname{argmin}} \|x_i^0 - c_k\|.$$

Hence, K-means clustering leads to spherical clusters, and thus assumes that two different groups of observations are linearly separable. However, for some datasets this assumption is clearly not reasonable (see below for an example).

By contrast, spectral clustering has the advantage to not impose the clusters to be convex.

Spectral clustering uses as data the matrix $[s_{il}]$ of pairwise similarities (recall that distances can be transformed into similarities) and represents the observations using a similarity graph $G = (V, E)$, which

- is an undirected graph (i.e. the edges have no direction),
- has vertices $V = \{x_i^0\}_{i=1}^n$ (i.e. each vertex represents an observation) and edges $E = \{(i, l)\}_{i,l=1}^n$,
- is such that each edge $(i, l) \in E$ has a weight $w_{il} = w_{li} \geq 0$, where $w_{il} \neq 0$ only if $s_{il} > 0$.

Then, the key idea of spectral clustering is to replace the clustering problem by a graph-partitioning problem, where the goal is to partition G such that

- the edges connecting vertices belonging to two different groups have low weights,
- the edges connecting vertices belonging to the same group have large weights.

Defining the similarity graph G

Given the similarities $[s_{il}]$, defining G amounts to choosing the matrix $\boldsymbol{W} = [w_{il}]$ of weights.

Remark: The matrix $\boldsymbol{W}$ is called the adjacency matrix of G.

One approach is to consider a fully connected graph; that is to let $\boldsymbol{W} = [s_{il}]$.

Another approach consists in letting $\boldsymbol{W}$ be such that

$$w_{il} = \begin{cases} s_{il}, & (i,l) \in \mathcal{N}_K \\ 0, & (i,l) \notin \mathcal{N}_K \end{cases}, \quad \forall i, l \in \{1, \ldots, n\}$$

where $\mathcal{N}_K$ is such that $(i,l) \in \mathcal{N}_K$ if and only if $s_{il} > 0$ and $s_{il} < s_{ii'}$ for at most $K-1$ values of $i' \in \{1, \ldots, n\} \setminus \{l\}$. In this case, G is called the mutual K-nearest-neighbour graph.

For every $i = 1, \ldots, n$ let $g_{ii} = \sum_{l=1}^{n} w_{il}$ be the degree of the vertex x_i^0 and let

$$\boldsymbol{L} = \boldsymbol{G} - \boldsymbol{W}, \quad \boldsymbol{G} = \mathrm{diag}(g_{11}, \ldots, g_{nn})$$

be the graph Laplacian.

Formalizing the graph partitioning problem

Let M be the number of clusters we want to compute and $\mathcal{S}_M$ be the set containing all the partitions of $\{1, \ldots, n\}$ into M non-empty sets.

Then, a natural way to formalize the graph partitioning problem is to compute $\{A_m^*\}_{m=1}^M \in \mathcal{S}_M$ such that

$$\{A_m^*\}_{m=1}^M \in \underset{\{A_m\}_{m=1}^M \in \mathcal{S}_M}{\operatorname{argmin}} \frac{1}{2} \sum_{m=1}^M W(A_m) \tag{5.4}$$

where $W(A) = \sum_{i \in A,\ l \notin A} w_{il}$ is the sum of the edges that connect a vertex x_i^0 with $i \in A$ and a vertex x_l^0 with $l \notin A$.

The partition $\{A_m^*\}_{m=1}^M$ defined in (5.4) however tends to be unbalanced[a]. To avoid this problem, the graph partitioning problem considered in spectral clustering amounts to computing

$$\{\tilde{A}_m^*\}_{m=1}^M \in \underset{\{A_m\}_{m=1}^M \in \mathcal{S}_M}{\operatorname{argmin}} \operatorname{RatioCut}(\{A_m\}_{m=1}^M) \tag{5.5}$$

where

$$\operatorname{RatioCut}(\{A_m\}_{m=1}^M) = \frac{1}{2} \sum_{m=1}^M \frac{W(A_m)}{|A_m|}, \quad \{A_m\}_{m=1}^M \in \mathcal{S}_M.$$

Solving (5.5) is computationally expensive, and the spectral clustering algorithm provides an approximate solution to this optimization problem.

[a] That is, $\{A_m^*\}_{m=1}^M$ is often such that $|A_m^*|$ is large for some m and small for others.

Preliminary: A useful lemma

Lemma 5.1 *We have* $f^\top \boldsymbol{L} f = \frac{1}{2}\sum_{i=1}^n \sum_{l=1}^n w_{il}(f_i - f_l)^2$ *for all* $f \in \mathbb{R}^n$.

Proof: Let $f \in \mathbb{R}^n$, then

$$\begin{aligned} f^\top \boldsymbol{L} f = f^\top \boldsymbol{G} f - f^\top \boldsymbol{W} f &= \sum_{i=1}^n g_{ii} f_i^2 - \sum_{i=1}^n \sum_{l=1}^n w_{il} f_i f_l \\ &= \frac{1}{2}\Big(\sum_{i=1}^n g_{ii} f_i^2 - 2\sum_{i=1}^n \sum_{l=1}^n w_{il} f_i f_l + \sum_{l=1}^n g_{ll} f_l^2 \Big) \\ &= \frac{1}{2}\sum_{i=1}^n \sum_{l=1}^n w_{il}(f_i - f_l)^2 \end{aligned}$$

and the proof is complete. □

A first important consequence of Lemma 5.1 is that the matrix $\boldsymbol{L}$ is positive semi-definite (but not positive definite). Below we denote by $\lambda_1 \geq \cdots \geq \lambda_n = 0$ its eigenvalues.

A second important consequence of Lemma 5.1 is to show that $(1, \ldots, 1) \in \mathbb{R}^n$ is an eigenvector of $\boldsymbol{L}$ associated to the eigenvalue $\lambda_n = 0$.

To see this let let $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ and $\boldsymbol{B} \in O(n)$ be such that $\boldsymbol{L} = \boldsymbol{B}\boldsymbol{\Lambda}\boldsymbol{B}^\top$. Then, $\boldsymbol{L}^{1/2} = \boldsymbol{B}\boldsymbol{\Lambda}^{1/2}\boldsymbol{B}^\top$, showing that $\boldsymbol{L}$ and $\boldsymbol{L}^{1/2}$ have the same eigenvectors, and that 0 is also an eigenvalue of $\boldsymbol{L}^{1/2}$. Let f be a corresponding eigenvector of $\boldsymbol{L}^{1/2}$. Then, $0 = \|\boldsymbol{L}^{1/2} f\| = f^\top \boldsymbol{L} f$ and, by Lemma 5.1, $f^\top \boldsymbol{L} f = 0$ if and only if all the components of f are identical. Hence, $(1, \ldots, 1) \in \mathbb{R}^n$ is an eigenvector of $\boldsymbol{L}^{1/2}$ associated to the eigenvalue $\lambda = 0$, and this $(1, \ldots, 1) \in \mathbb{R}^n$ is an eigenvector of $\boldsymbol{L}$ associated to $\lambda_n = 0$.

Rewriting (5.5) for $M = 2$

We first rewrite (5.5) in term of the graph Laplacian $\boldsymbol{L}$.

For any $A \subsetneq \{1, \ldots, n\}$ let $f_A = (f_{A,1}, \ldots, f_{A,n})^\top$ be defined by

$$f_{A,i} := \begin{cases} \sqrt{(n - |A|)/|A|}, & i \in A \\ -\sqrt{|A|/(n - |A|)}, & i \notin A. \end{cases}, \quad i = 1, \ldots, n. \tag{5.6}$$

Then, we have the following result:

Lemma 5.2 *Let $M = 2$, $A \subsetneq \{1, \ldots, n\}$ and f_A be as defined in (5.6). Then, $f_A^\top \boldsymbol{L} f_A = n \times \mathrm{RatioCut}(\{A, \bar{A}\})$.*

Proof: Using Lemma 5.1 for the first equality, we have

$$\begin{aligned} f_A^\top \boldsymbol{L} f_A = \frac{1}{2} \sum_{i,l=1}^{n} w_{il}(f_{A,i} - f_{A,l})^2 &= \frac{1}{2} \sum_{i \in A,\, l \notin A} w_{il} \Big(\sqrt{(n - |A|)/|A|} + \sqrt{|A|/(n - |A|)} \Big)^2 \\ &= \frac{1}{2} \sum_{i \in A,\, l \notin A} w_{il} \Big(\frac{n - |A|}{|A|} + \frac{|A|}{n - |A|} + 2 \Big) \\ &= \frac{1}{2} W(A) \Big(\frac{(n - |A|) + |A|}{|A|} + \frac{|A| + (n - |A|)}{n - |A|} \Big) \\ &= \frac{1}{2} W(A) \Big(\frac{n}{|A|} + \frac{n}{n - |A|} \Big) \\ &= \frac{n}{2} \Big(\frac{W(A)}{|A|} + \frac{W(A^c)}{|A|^c} \Big) \\ &= n \times \mathrm{RatioCut}(\{A, \bar{A}\}). \end{aligned}$$

□

It is trivial to see that for all $A \subsetneq \{1, \ldots, n\}$ we have both $f_A^\top 1 = 0$ and $\|f_A\|^2 = n$, and thus, by Lemma 5.2, solving (5.5) is equivalent to solving

$$\min_{A \subsetneq \{1, \ldots, n\}} f_A^\top \boldsymbol{L} f_A \text{ subject to } f_A^\top 1 = 0, \quad \|f_A\| = \sqrt{n}. \tag{5.7}$$

Finding an approximate solution to (5.5) for $M = 2$

The optimization problem (5.7) is hard to solve because it is a discrete optimization problem.

The first approximation we make is to discard the discreteness, that is, instead of solving (5.7) we aim at computing

$$f^* \in \operatorname*{argmin}_{f \in \mathcal{B}} f^\top \boldsymbol{L} f, \quad \mathcal{B} = \big\{\tilde{f} \in \mathbb{R}^n : \ \tilde{f}^\top 1 = 0, \ \|\tilde{f}\| = \sqrt{n}\big\}. \tag{5.8}$$

The solution f^* to (5.8) is given in the next lemma.

Lemma 5.3 *Let $M = 2$ and v be the eigenvector of length $\sqrt{n}$ corresponding to the second smallest eigenvalue of $\boldsymbol{L}$. Then, $f^* = v$ is solution to* (5.8).

Proof: As shown above, $f^{(1)} := 1$ is the eigenvector of length $\sqrt{n}$ associated to the smallest eigenvalue $\lambda_n = 0$ of $\boldsymbol{L}$. Hence, (5.8) can be rewritten as

$$\min_{f \in \mathbb{R}^n} f^\top \boldsymbol{L} f \text{ subject to } f^\top f^{(1)} = 0, \quad \|f\| = \sqrt{n}$$

and the result follows from similar computations as in the proof of Theorem 1.1.□

To obtain an approximate solution to (5.7) we need to "transform" the vector $f^* \in \mathbb{R}^n$ into an element of $\mathcal{F}_M := \{f_A, \, A \subseteq \{1, \ldots, n\}\}$.

For $M = 2$ every element $f \in \mathcal{F}_M$ can take only two values, and transforming f^* into an element of $\mathcal{F}_M$ can therefore be achieved using K-means clustering with $K = 2$ and with "observations" $f_1^*, \ldots, f_n^*$.

Letting C_1' and C_2' be the resulting two clusters computed by K-means clustering, the sets $A_1 = \{i : \ f_i^* \in C_1'\}$ and $A_2 = \{i : \ f_i^* \in C_2'\}$ provide an approximate solution to the initial problem (5.5).

A first spectral clustering algorithm

The procedure we just described for finding an approximate solution to (5.5), which can be extended to any number $M \geq 2$ of clusters (see [9]), is summarized in the following algorithm.

Input: Number of clusters $M \geq 2$.

(i) Compute $\boldsymbol{Z} \in \mathbb{R}^{n \times M}$, the matrix having as columns the M eigenvectors corresponding to the M smallest eigenvalues of $\boldsymbol{L}$.

(ii) Cluster the points $\{z_i\}_{i=1}^n$ using K-means clustering with $K = M$, to obtain clusters $C'_1, \ldots, C'_M$,

(iii) Let $A_m = \{i : z_i \in C'_m\}$ for $m = 1, \ldots, M$.

Return: Partition $\{A_m\}_{m=1}^M$ of $\{1, \ldots, n\}$.

Remark: Recall that the smallest eigenvalue of $\boldsymbol{L}$ is 0 with associated eigenvector $(1, \ldots, 1)$. Hence, when performing K-means clustering on $z_1, \ldots, z_n$ on Line 2 of the algorithm, the first component of z_i is one for all $i \in \{1, \ldots, n\}$, and thus has no impact on the definition of the clusters $\{C'_m\}_{m=1}^M$.

Another formalization of the graph partitioning problem

A popular alternative to the formulation (5.5) of the graph partitioning problem is to compute $\{A_m^*\}_{m=1}^M \in \mathcal{S}_M$ such that

$$\{\tilde{A}_m^*\}_{m=1}^M \in \underset{\{A_m\}_{m=1}^M \in \mathcal{S}_M}{\operatorname{argmin}} \ \text{NormalizedCut}(\{A_m\}_{m=1}^M) \tag{5.9}$$

where

$$\text{NormalizedCut}(\{A_m\}_{m=1}^M) = \frac{1}{2}\sum_{m=1}^M \frac{W(A_m)}{\sum_{i,l\in A_m} w_{il}}, \quad \{A_m\}_{m=1}^M \in \mathcal{S}_M.$$

Proceeding as for the optimization problem (5.5), it can be shown that an approximate solution to (5.9) is obtained by replacing, in the above spectral algorithm, the Laplacian $\boldsymbol{L}$ by the symmetric normalized Laplacian $\boldsymbol{L}^* := \boldsymbol{G}^{-1/2}\boldsymbol{L}\boldsymbol{G}^{-1/2}$ (see [9]).

The resulting spectral algorithm is however often expressed using the matrix $\tilde{\boldsymbol{L}} := \boldsymbol{G}^{-1/2}\boldsymbol{W}\boldsymbol{G}^{-1/2}$, and not $\boldsymbol{L}^*$. To understand why we can do that remark that

$$\boldsymbol{L}^* = \boldsymbol{G}^{-1/2}(\boldsymbol{G}-\boldsymbol{W})\boldsymbol{G}^{-1/2} = \boldsymbol{I}_n - \boldsymbol{G}^{-1/2}\boldsymbol{W}\boldsymbol{G}^{1/2} = \boldsymbol{I}_n - \tilde{\boldsymbol{L}}$$

so that if $\tilde{\boldsymbol{L}} = \boldsymbol{C}\tilde{\boldsymbol{\Lambda}}\boldsymbol{C}^\top$ for some $\boldsymbol{C} \in O(n)$ and diagonal matrix $\tilde{\boldsymbol{\Lambda}}$ then

$$\boldsymbol{L}^* = \boldsymbol{I}_n - \boldsymbol{C}\tilde{\boldsymbol{\Lambda}}\boldsymbol{C}^\top = \boldsymbol{C}(\boldsymbol{I}_n - \tilde{\boldsymbol{\Lambda}})\boldsymbol{C}^\top. \tag{5.10}$$

In words, the diagonal matrix $\boldsymbol{I}_n - \tilde{\boldsymbol{\Lambda}}$ contains the n eigenvalues of $\boldsymbol{L}^*$ and the matrix $\boldsymbol{C}$ contains the corresponding eigenvectors, can be computed from $\tilde{\boldsymbol{L}}$.

Remark: Since $\boldsymbol{L}^*$ is positive-semi definite and has at least one zero eigenvalue then, by (5.10), the largest eigenvalue of $\tilde{\boldsymbol{L}}$ is one.

Another spectral clustering algorithm

Using the above computations, it follows that replacing $\boldsymbol{L}$ by $\boldsymbol{L}^*$ in the spectral algorithm introduced earlier is equivalent to the following algorithm.

Input: Number of clusters $M \geq 2$.

(i) Compute $\boldsymbol{Z} \in \mathbb{R}^{n \times M}$, the matrix having as columns the M eigenvectors corresponding to the M largest eigenvalues of $\tilde{\boldsymbol{L}}$.

(ii) Cluster the points $\{z_i\}_{i=1}^n$ using K-means clustering with $K = M$, to obtain clusters $C'_1, \ldots, C'_M$,

(iii) Let $A_m = \{i : z_i \in C'_m\}$ for $m = 1, \ldots, M$.

Return: Partition $\{A_m\}_{m=1}^M$ of $\{1, \ldots, n\}$.

Spectral clustering: Practicalities

- In practice we often use a fully connected graph, in which case $\boldsymbol{W} = [s_{il}]$.

- When $d_{il} = \|x_i^0 - x_l^0\|$ and $s_{il} = \exp(-d_{il}^2/c)$ the matrix $[s_{il}]$ is the Gram matrix associated to the Gaussian kernel (see Chapter 4)

$$k(x, x') = \exp(-\|x - x'\|^2/c^2).$$

  Note that any non negative and bounded kernel k can be used to construct a matrix of similarities $[s_{il}]$, and that the output of the clustering process depends on the choice of k.

- Let $1 = \tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_n \geq 0$ denotes the M eigenvalues of $\tilde{\boldsymbol{L}}$. Then, the theory suggests that if the observations are such that, in some sense, there exist M completely disconnected clusters then $\lambda_1 = \cdots = \lambda_M = 1$ while $\lambda_{M+1} < 1$ [7].

  This result suggests to choose M such that $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_M$ is large but $\tilde{\lambda}_{M+1}$ is small. This approach for selecting M in practice is known as the eigengap heuristic.

K-means clustering vs spectral clustering: Toy example

We consider $n = 200$ bivariate observations $\{x_i^0\}_{i=1}^n$, where half of them are sampled from a Gaussian distribution and half of them are sampled from a banana shape distribution. The objective of this example is to use K-means and spectral clustering to recover the two groups of data points. For this example we let $d_{il} = \|x_i^0 - x_l^0\|$ and spectral clustering is implemented with $s_{il} = \exp(-d_{il}^2/c^2)$, where $c = 0.2$, and with $\boldsymbol{W} = [s_{il}]$.

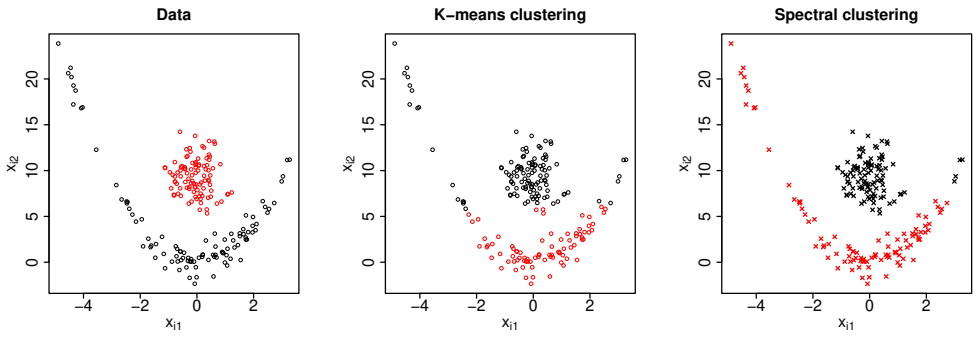

Figure 5.4: Comparison of K-means and spectral clustering on a toy example.

As we can observe from the left plot of Figure 5.4, the two true clusters are not spherical and, as expected, K-means clustering performs poorly on this data set; see the middle plot of Figure 5.4. On the other hand, we observe in the last plot of this figure that spectral clustering performs well on this example.

Remark: For this example the performance of spectral clustering is very sensitive to the choice of the parameter $c > 0$ used in the definition of s_{il}.

References

[1] Bishop, C. M. (2006). Pattern recognition. *Machine learning*, 128(9).

[2] Comon, P. (1994). Independent component analysis, a new concept? *Signal processing*, 36(3):287–314.

[3] Friedman, J., Hastie, T., Tibshirani, R., et al. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

[4] Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430.

[5] Inaba, M., Katoh, N., and Imai, H. (1994). Applications of weighted voronoi diagrams and randomization to variance-based k-clustering. In *Proceedings of the tenth annual symposium on Computational geometry*, pages 332–339.

[6] Mardia, K., Kent, J., and Bibby, J. (1979). *Multivariate analysis*. Probability and mathematical statistics. Academic Press Inc.

[7] Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14.

[8] Paulsen, V. I. and Raghupathi, M. (2016). *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Cambridge Studies in Advanced Mathematics. Cambridge University Press.

[9] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.