Clustering

Peng Chen, Jiaqi Xia

School of Management
University of Science and Technology of China

January 20, 2022

Table of Contents

1. Introduction
   - Hierarchical Clustering Methods
   - Partition Clustering Methods
   - Self-Organizing Map
   - Convex Clustering
2. Spectral Clustering
   - Introduction
   - Derivation
   - Summary
   - Practical Details
3. Dirichlet Process
   - Motivation
   - Definition
   - Clustering

Outline

1. Introduction
   - Hierarchical Clustering Methods
   - Partition Clustering Methods
   - Self-Organizing Map
   - Convex Clustering
2. Spectral Clustering
   - Introduction
   - Derivation
   - Summary
   - Practical Details
3. Dirichlet Process
   - Motivation
   - Definition
   - Clustering

Unsupervised Learning

- In supervised learning, we have training samples $(x_1, y_1), \cdots, (x_N, y_N)$ and our interest is to infer the function of x to predict y.
- In unsupervised learning, we have only N p-dimensional observations $(x_1, x_2, \cdots, x_N)$, which is our main interest.
- The dimension p is sometimes much higher, and the properties of interest are often more complicated.

Example: Human Tumor Microarray Data

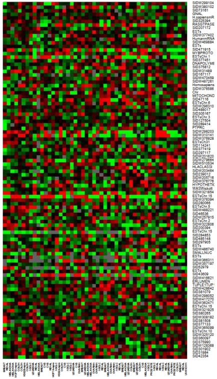


Figure: Human Tumor Microarray Data: a 6830×64 matrix of real numbers, each representing an expression measurement for a gene (row) and sample (column).

Example: Human Tumor Microarray Data

- Here we cluster the samples, each of which is a vector of length 6830, corresponding to expression values for the 6830 genes.

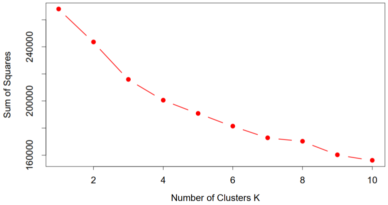


Figure: The SSE of K-means algorithm with different choice of the number of clusters K.

Cluster	Breast	CNS	Colon	K562	Leukemia	MCF7
1	3	5	0	0	0	0
2	2	0	0	2	6	2
3	2	0	7	0	0	0
Cluster	Melanoma	NSCLC	Ovarian	Prostate	Renal	Unknown
1	1	7	6	2	9	1
2	7	2	0	0	0	0
3	0	0	0	0	0	0

Figure: Number of cancer cases of each type, in each of the three clusters from K-means clustering.

Clustering

- Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).
- How to group objects, i.e. how to measure the similarities between objects?
- How to choose the number of clusters K?

Measures of Similarities

- Most algorithms presume a matrix $\mathbf{D}$ of dissimilarities with nonnegative entries and zero diagonal elements: $d_{ii} = 0, i = 1, \cdots, N$.
- If the original data were collected as similarities, a suitable monotone-decreasing function can be used to convert them to dissimilarities.
- Most algorithms assume symmetric dissimilarity matrices, so if the original matrix $\mathbf{D}$ is not symmetric it may be replaced by $(\mathbf{D} + \mathbf{D}^\top)/2$.

Dissimilarities Based on Attributes

- In order to construct a dissimilarity matrix, we must first construct pairwise dissimilarities between the objects.
- We define a dissimilarity $d_k(x_{ik}, x_{jk})$ between values of the kth attribute, and then define

$$d(x_i, x_j) = \sum_{k=1}^{p} d_k(x_{ik}, x_{j_k})$$

as the dissimilarity between objects i and j.

Dissimilarities Based on Attributes

- **Quantitative variables** are represented by continuous real-valued numbers. Error between them can be a monotone-increasing function of their absolute difference

$$d(x_i, x_j) = l(|x_i - x_j|).$$

- **Ordinal variables** are often represented as ordered contiguous integers. Error are generally defined by replacing their M original values with

$$\frac{i - 1/2}{M}, \quad i = 1, \cdots, M.$$

- **Categorical variables** are unordered categorical variables, the degree-of-difference between pairs of values must be delineated explicitly.

Object Dissimilarity

- Then we can combine the p-individual attribute dissimilarities into a single overall measure of dissimilarity $d(x_i, x_j)$ between two objects (x_i, x_j).
- This is nearly always done by means of a weighted average (convex combination)

$$d(x_i, x_j) = \sum_{k=1}^{p} w_k \cdot d_k(x_{ik}, x_{jk}), \quad \sum_{k=1}^{p} w_k = 1.$$

- The choice of w_k should be based on subject matter considerations.

Measures of Similarities

- The measure of the similarity between instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is mainly decided by the distance $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$.
- **Minkowski Measures** $(L_r$ distance): $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{k=1}^{p} |x_{ik} - x_{jk}|^r\right)^{\frac{1}{r}}$.
- + The Euclidean (L_2) distance is mostly applied to find similarity between two objects, which are expressed numerically.
- + It is highly sensitive to noise and usually not applied to data with hundreds of attributes also features with high values tend to dominate others.

Measures of Similarities

- **Cosine Measure**: $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle}{||\boldsymbol{x}_i|| \cdot ||\boldsymbol{x}_j||}$.
- + It is popular in in text mining and information retrieval.
- + It is invariant to rotation but not to linear transformations.
- **Pearson Correlation Measure**: $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^\top (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_j)}{||\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i|| \cdot ||\boldsymbol{x}_j - \bar{\boldsymbol{x}}_j||}$.
- + It is a measure of linear correlation between two variables.

Measures of Similarities

- **Extended Jaccard Measure**: $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\boldsymbol{x}_i^\top \boldsymbol{x}_j}{||\boldsymbol{x}_i||^2 + ||\boldsymbol{x}_j||^2 - \boldsymbol{x}_i^\top \boldsymbol{x}_j}$.
  - The binary Jaccard coefficient measures the degree of overlap between two sets and is computed as the ratio of the number of shared attributes.
  - It is suitable sufficiently to be employed in the documents or word similarity measurement.
- **Dice Coefficient Measure**: $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{2\boldsymbol{x}_i^\top \boldsymbol{x}_j}{||\boldsymbol{x}_i||^2 + ||\boldsymbol{x}_j||^2}$.
  - It is F1 score when applied to binary data.
- **Kernel trick**.

Choose K

- For data segmentation, the number of clusters K is the part of the problem. For example, a company employ K sales people, and then partition a customer database into K segments, one for each sales person.
- Often, cluster analysis is used to provide a descriptive statistic for ascertaining the extent to which the observations comprising the data base fall into natural distinct groupings. Here the number of such groups K^* is unknown and need to be estimated from the data.

Choose K

- Data-based methods for estimating K^* typically examine the within-cluster dissimilarity W_K as a function of the number of clusters K.
- Usually, there will be a sharp decrease in successive differences in criterion value, $W_K - W_{K+1}$, at $K = K^*$. That is, $\{W_K - W_{K+1} | K < K^*\} \gg \{W_K - W_{K+1} | K \geq K^*\}$.
- An estimate $\widehat{K^*}$ for K^* is then obtained by identifying a "kink" in the plot of W_K as a function of K.

Choose K

- The recently proposed Gap statistic compares the curve $\log W_K$ to the curve obtained from data uniformly distributed over a rectangle containing the data.
- It estimates the optimal number of clusters to be the place where the gap between the two curves is largest.

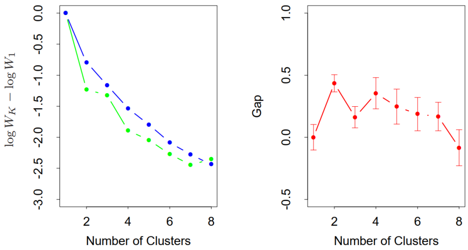


Figure: $K^* = 2$

Hierarchical Clustering Methods

- In hierarchical clustering methods, clusters are formed by iteratively dividing the patterns using top-down or bottom up approach.

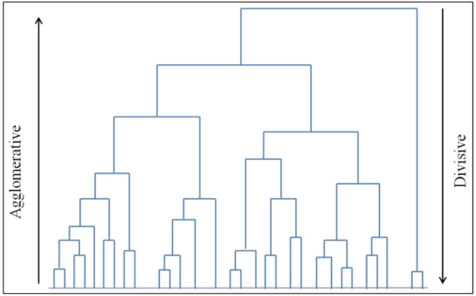


Figure: Hierarchical clustering dendrogram

Agglomerative & Divisive

- The agglomerative (bottom-up) approach builds up clusters starting with single object and then merging these atomic clusters into larger and larger clusters, until all of the objects are finally lying in a single cluster.
- The divisive (top-down) approach breaks up cluster containing all objects into smaller clusters, until each object forms a cluster on its own.

Example (Agglomerative)

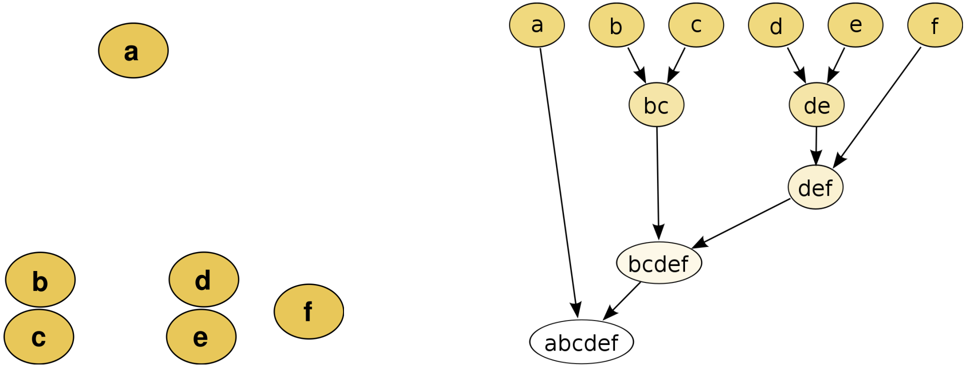


Figure: A simple agglomerative clustering algorithm with single-linkage

Similarity Measure (Linkage)

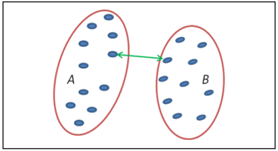


Figure: Single-linkage

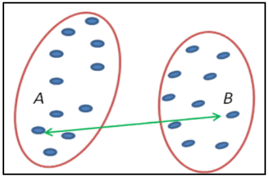


Figure: Complete-linkage

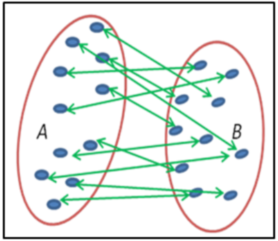


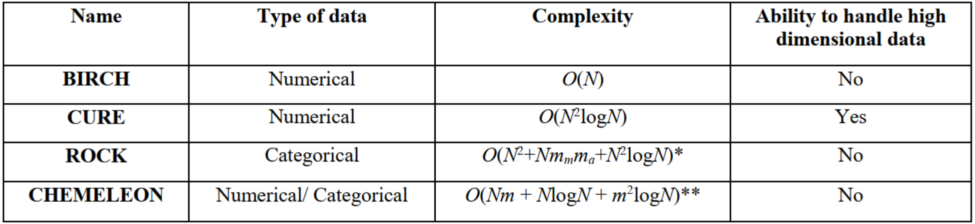
Figure: Average-linkage

Similarity Measure (Linkage)

- Single-linkage:
$$d(A,B)=\min\{d(a,b):a\in A,b\in B\}.$$
- Complete-linkage:
$$d(A,B)=\min\{d(a,b):a\in A,b\in B\}.$$
- Average-linkage:
$$d(A,B)=\frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B}d(a,b).$$

Criticisms and Enhancement

- The classic HC algorithms lack robustness and are sensitive to noise and outliers.
- The computational complexity for most of HC algorithms is $O(N^2)$.



Name	Type of data	Complexity	Ability to handle high dimensional data
BIRCH	Numerical	$O(N)$	No
CURE	Numerical	$O(N^2 \log N)$	Yes
ROCK	Categorical	$O(N^2+Nm_m m_a+N^2 \log N)$*	No
CHEMELEON	Numerical/ Categorical	$O(Nm + N\log N + m^2 \log N)$**	No

Partition Clustering Methods

- Opposite to hierarchical clustering, here data are assigned into K (predefined) clusters without any hierarchical structure by optimizing some criterion (e.g. Minimum Euclidean distance).
- Examples: K-means, PAM, CLARA, CLARANS, Fuzzy C-means, DBSCAN etc.

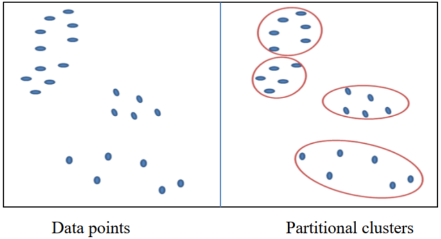


Figure: Partitional clustering approaches

K-means

- K-means algorithm assign N observations to K clusters in such a way that within each cluster the average dissimilarity of the observations from the cluster mean is minimized.
- The loss function (within-point scatter) can be written as

$$L = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \left\| x_i - \bar{x}_k \right\|^2 ,$$

where $N_k = \sum_{i=1}^{N} I(C(i) = k)$.

- Then we need to solve the following optimization problem

$$\min_{C,\{m_k\}_1^K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \left\| x_i - m_k \right\|^2$$

Iterative Descent Algorithm

Algorithm K-means Clustering

```
1. For a given assignment $C$, minimization with respect to $\{m_1, \dots, m_K\}$ yielding the means of the currently assigned clusters.
2. Given a current set of means $\{m_1, \dots, m_K\}$, assign each observation to the closest (current) cluster mean
       $C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} \|x_i - m_k\|^2$.
3. Steps 1 and 2 are iterated until the assignments do not change.
```

Iterative Descent Algorithm

- Each of steps 1 and 2 reduces the value of the criterion, so that convergence is assured.
- We should start the algorithm with many different random choices for the starting means, and choose the solution having smallest loss.

Self-Organizing Map

- Self-organizing map (SOM) is an unsupervised technique used to produce a low-dimensional (typically two-dimensional) representation of a higher dimensional data set while preserving the topological structure of the data.

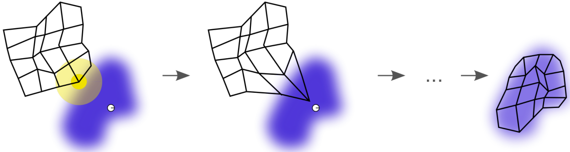


Figure: An illustration of the training of a self-organizing map

Self-Organizing Map

- We consider a SOM with a two-dimensional rectangular grid of K prototypes $m_j \in \mathbb{R}^p$.
- Each of the K prototypes are parametrized with respect to an integer coordinate pair $l_j \in \mathcal{Q}_1 \times \mathcal{Q}_2$. Here $\mathcal{Q}_1 = \{1, 2, \cdots, q_1\}$, similarly $\mathcal{Q}_2$, and $K = q_1 \cdot q_2$.
- The m_j are initialized and need to be updated iteratively.
- Once the model is fit, the observations can be mapped down onto the two-dimensional grid.

Self-Organizing Map

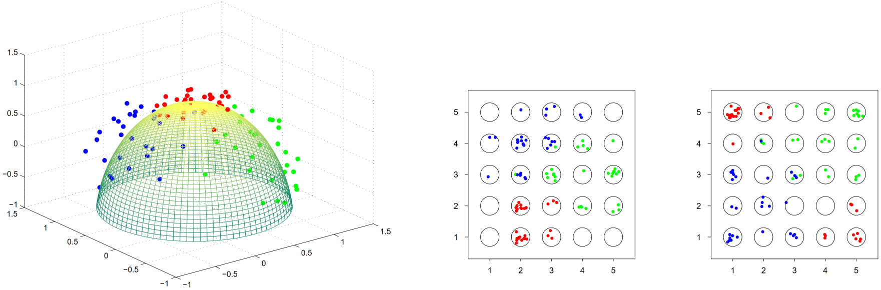


Figure: (Left) Simulated data in three classes. (Right) The 5×5 grid of prototypes.

Self-Organizing Map

- The observations x_i are processed one at a time.
- We find the closest prototype m_j to x_i in Euclidean distance in $\mathbb{R}^p$, and then for all neighbors m_k of m_j, move m_k toward x_i via the update

$$m_k \leftarrow m_k + \alpha(x_i - m_k)$$

.

- The neighbors m_k of m_j are defined with the distance $||l_j - l_k||$ and a threshold r.
- More sophisticated versions modify the update step according to distance:

$$m_k \leftarrow m_k + \alpha h(||l_k - l_j||)(x_i - m_k).$$

Self-Organizing Map

- Typically α is decreased from say 1.0 to 0.0 over a few thousand iterations (one per observation).
- If we take the threshold r small enough so that each neighborhood contains only one point, then the SOM algorithm is an online version of K-means clustering.

Self-Organizing Map

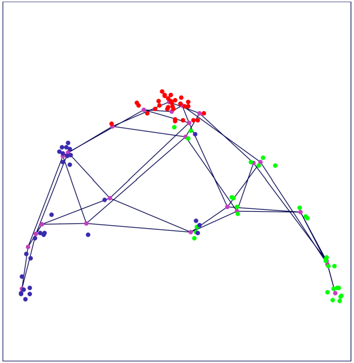


Figure: Wiremesh representation of the fitted SOM model in $\mathbb{R}^3$ where the purple points are the node centers

Convex Clustering

- Lindsten et al. (2011) and Hocking et al (2011). formulate the clustering task as a convex optimization problem.
- Given n points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in $\mathbb{R}^p$, they suggest minimizing the convex criterion

$$F_\gamma(\mathbf{U}) = \frac{1}{2} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{u}_i\|_2^2 + \gamma \sum_{i<j} w_{ij} \|\mathbf{u}_i - \mathbf{u}_j\|$$

  where γ is a positive tuning constant, w_{ij} is a nonnegative weight, and the i th column $\mathbf{u}_i$ of the matrix $\mathbf{U}$ is the cluster center attached to point $\mathbf{x}_i$.
- Different norm can be used here, e.g. Lindsten et al. consider an ℓ_p norm penalty on the differences $\mathbf{u}_i - \mathbf{u}_j$ while Hocking et al. consider ℓ_1, ℓ_2, and ℓ_∞ penalties.

Convex Clustering

- When $\gamma = 0$, the minimum is attained when $\mathbf{u}_i = \mathbf{x}_i$, and each point occupies a unique cluster.
- As γ increases, the cluster centers begin to coalesce. Two points $\mathbf{x}_i$ and $\mathbf{x}_j$ with $\mathbf{u}_i = \mathbf{u}_j$ are said to belong to the same cluster.
- For sufficiently high γ all points coalesce into a single cluster.

Convex Clustering

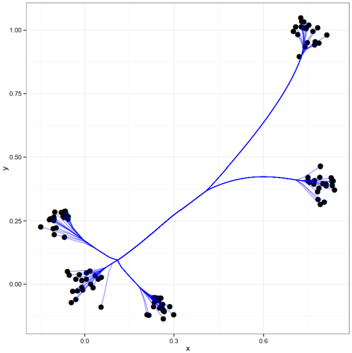


Figure: The blue lines trace the path of the cluster centers as the regularization parameter γ increases.

Convex Clustering

- The benefits of the formulation of convex relaxation are manifold.
- The convex relaxation admits a simple and fast iterative algorithm that is guaranteed to converge to the unique global minimizer, e.g. the alternating direction method of multipliers (ADMM) and alternating minimization algorithm (AMA).
- The convex relaxation performs continuous clustering which is intuitively appealing, globally optimal, and computationally tractable.

Outline

1. Introduction
   - Hierarchical Clustering Methods
   - Partition Clustering Methods
   - Self-Organizing Map
   - Convex Clustering
2. Spectral Clustering
   - Introduction
   - Derivation
   - Summary
   - Practical Details
3. Dirichlet Process
   - Motivation
   - Definition
   - Clustering

Spectral Clustering

Introduction

Given a set of observations $x_1, \ldots, x_n$, suppose we have obtained a similarity matrix:

$$S = (s_{ij})_{n\times n}$$

or a distance matrix

$$D = (d_{ij})_{n\times n},$$

where $s_{ij} = s(x_i, x_j) \geq 0$ and $d_{ij} = d(x_i, x_j) \geq 0$.

$Target$: clustering the observations based on S or D.

$*$ In the following, we focus on clustering based on S.

Spectral Clustering

Introduction

A Graph Cut Perspective

On the basis of S, we define a similarity graph $G = (V, E)$ and its weighted adjacency matrix $W = (w_{ij})_{n\times n}$, where $V = \{1, \dots, n\}$ is the vertex set representing $x_1, \dots, x_n$, $E \subset V \times V$ and $w_{ij} = s_{ij} I((i,j) \in E)$.

- ε-neighborhood: $(i,j) \in E$ if $s_{ij} > \varepsilon$ and $(i,j) \notin E$ otherwise.
- k-nearest neighbor: For each i, only $(i,j_1), \dots, (i,j_k) \in E$ such that $s_{ij_1} \geq \dots \geq s_{ij_k} \geq \dots$
- Fully connected: $(i,j) \in E$ if $s_{ij} > 0$.

Spectral Clustering

Introduction

Given the number k of clusters, our target translates to obtaining a partition:

$$V = \cup_{i=1}^{k} A_i,$$

where A_i's are disjoint.
An intuitive way:

$$\begin{aligned} &\text{Two nodes in the same group} \Rightarrow \text{high similarity;} \\ &\text{Two nodes in different groups} \Rightarrow \text{low similarity.} \end{aligned}$$

Spectral Clustering

Introduction

Cut

For $A, B \subset \{1, \ldots, n\}$, define the between-group weight:

$$W(A,B) = \sum_{\substack{i \in A \\ j \in B}} w_{ij}.$$

A straightforward way is to minimize

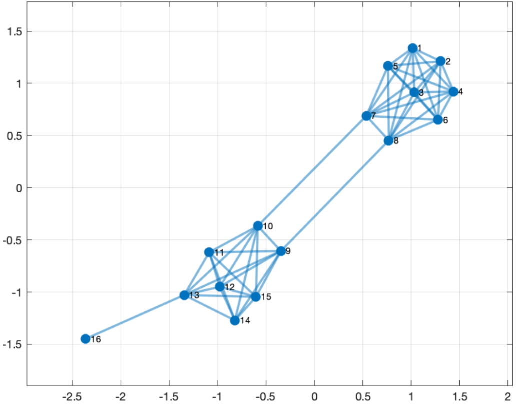
$$cut(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, A_i^c),$$

where $A_i^c = V - A_i$.
Unfortunately, it often yields unsatisfactory partitions in practice.

Spectral Clustering

Introduction

Consider the following unweighted graph. $x_1, \ldots, x_{16} \in \mathbb{R}^2$. E is shown in the figure.



Spectral Clustering

Introduction

Set $k = 2$. Simple calculation yields

$$A_1 = \{1, \dots, 15\},\ A_2 = \{16\} \Rightarrow cut(A_1, A_2) = \frac{1}{2};$$
$$A_1 = \{1, \dots, 8\},\ A_2 = \{9, \dots, 16\} \Rightarrow cut(A_1, A_2) = 1.$$

Which indicates that minimizing cut is likely to yield unbalanced clusters.

Spectral Clustering

Derivation

Ratio Cut

An improved method tries to minimize

$$Ratiocut(A_1,\ldots,A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i,A_i^c)}{|A_i|}. \tag{1}$$

When $k=2$, it simplifies to

$$Ratiocut(A_1,A_2) = cut(A_1,A_2)\Big(\frac{1}{|A_1|}+\frac{1}{|A_2|}\Big).$$

The second term reaches minimum iff $|A_1|=|A_2|$.

$*$ This method minimize the overall between-group similarity, while balancing group sizes.

Spectral Clustering

Derivation

To obtain the minimizor of (1), we introduce two definitions.

- Membership matrix: $H = (h_{ij})_{n\times k}$, where

$$h_{ij} = \begin{cases} 1/\sqrt{|A_j|}, & i \in A_j \\ 0, & \text{otherwise.} \end{cases}$$

- Degree matrix: $D = \mathrm{diag}\{d_1, \ldots, d_n\}$, where

$$d_i = \sum_{j=1}^{n} w_{ij}, \quad i = 1, \ldots, n.$$

Spectral Clustering

Derivation

Denote $H = (h_1^T, \dots, h_n^T)^T$ and let $L := D - W$, we have

$$\frac{W(A_i, A_i^c)}{2|A_i|} = h_i^T L h_i = (H^T L H)_{ii}.$$

Hence, $Ratiocut(A_1, \dots, A_k) = tr(H^T L H)$ and we can write the problem as

$$\begin{aligned} \min_{A_1,\dots,A_k} \; & tr(H^T L H) \\ \text{s.t. } \; & H^T H = I. \end{aligned} \tag{2}$$

Spectral Clustering

Derivation

Solving (2) entails searching over $2^{|V|}$ values. For ease of computation, we relax it to

$$\min_{H\in\mathbb{R}^{n\times k}} tr(H^T LH)$$
$$\text{s.t. } H^T H = I,$$

which is exactly an eigenvalue problem.

Spectral Clustering

Derivation

Normalized Cut

For $A \subset \{1, \dots, n\}$, define another measure of the size of A:

$$\mathrm{vol}(A) = \sum_{i \in A} \sum_{j=1}^{n} w_{ij}.$$

An alternative method seeks to minimize

$$Ncut(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, A_i^c)}{\mathrm{vol}(A_i)}.$$

Spectral Clustering

Derivation

Let $L_{rw} := D^{-1}L = I - D^{-1}W$. Similar procedure shows that the optimization problem is equivalent to

$$\begin{aligned} \min_{A_1,\dots,A_k} \; & tr(H^T L_{rw} H) \\ \text{s.t. } \; & H^T H = I. \end{aligned} \tag{3}$$

Let $L_{sym} := D^{-1/2} L D^{-1/2}$ and substituting in $H = D^{-1/2}T$ transform (3) to

$$\begin{aligned} \min_{A_1,\dots,A_k} \; & tr(T^T L_{sym} T) \\ \text{s.t. } \; & T^T T = I. \end{aligned} \tag{4}$$

As before, we solve their relaxed form in practice.

Spectral Clustering

Derivation

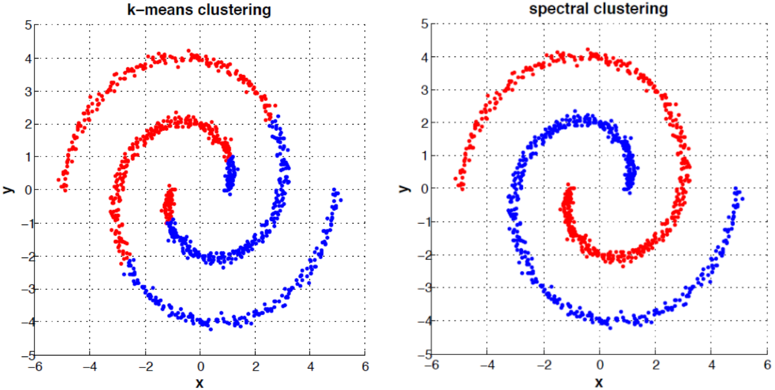
To obtain discrete partitions from the resultant membership matrices, we can apply standard algorithms to cluster the rows of them, e.g., the k-means.

$\ast$ Embedding n-dimensional similarity vectors into a k-dimensional space through Laplacian eigenmap.

Spectral Clustering

Derivation

Compared to k-means, spectral clustering can yield non-convex clusters.



Spectral Clustering
Summary

The Laplacian Matrices
The matrices corresponding to those eigenvalue problems are called the Laplacian matrices in spectral clustering.

- The unnormalized Laplacian matrix

$$L = D - W$$

- The normalized Laplacian matrices

$$L_{sym} = D^{-1/2} L D^{-1/2}$$
$$L_{rw} = D^{-1} L$$

Spectral Clustering

Summary

Properties of L:

- $\forall f \in \mathbb{R}^n,\ f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$.
- $L^T = L,\ L \succeq 0$.
- $\lambda_n(L) = 0$ with eigenvector $\mathbf{1} = (1, \ldots, 1)^T$.

Properties of L_{sym} and L_{rw}:

- $\forall f \in \mathbb{R}^n,\ f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}\left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}}\right)^2$.
- λ is an eigenvalue of L_{rw} with eigenvector $u \Leftrightarrow \lambda$ is an eigenvalue of L_{rw} with eigenvector $D^{1/2}u$.
- $\lambda_n(L_{rw}) = 0$ with eigenvector $\mathbf{1} = (1, \ldots, 1)^T$.
- $L_{rw}, L_{sym} \succeq 0$.

Spectral Clustering

Summary

Unnormalized Spectral Clustering

$Input$: similarity matrix S, number k of clusters.

- Construct a weighted adjacency matrix W.
- Compute the unnormalized Laplacian L.
- Compute the first k eigenvectors $u_1, \dots, u_k$ of L.
- Let $U := (u_1, \dots, u_k) = (y_1^T, \dots, y_n^T)^T$.
- Cluster the points $(y_i)_{i=1,\dots,n}$ with k-means algorithm into clusters $C_1, \dots, C_k$.

$Output$: Clusters $A_1, \dots, A_k$ with $A_i = \{j | y_j \in C_i\}$.

Spectral Clustering

Summary

Normalized Spectral Clustering 1

$Input$: similarity matrix S, number k of clusters.

- Construct a weighted adjacency matrix W.
- Compute the unnormalized Laplacian L.
- Compute the first k eigenvectors $u_1, \dots, u_k$ of the generalized eigenproblem $Lu = \lambda Du$.
- Let $U := (u_1, \dots, u_k) = (y_1^T, \dots, y_n^T)^T$.
- Cluster the points $(y_i)_{i=1,\dots,n}$ with k-means algorithm into clusters $C_1, \dots, C_k$.

$Output$: Clusters $A_1, \dots, A_k$ with $A_i = \{j | y_j \in C_i\}$.

Spectral Clustering

Summary

Normalized Spectral Clustering 2

$Input$: similarity matrix S, number k of clusters.

- Construct a weighted adjacency matrix W.
- Compute the normalized Laplacian L_{sym}.
- Compute the first k eigenvectors $u_1, \ldots, u_k$ of L_{sym}.
- Let $U := (u_1, \ldots, u_k)$, normalize its rows to norm 1 and obtain T, that is set $t_{ij} = u_{ij}/(\sum_k u_{ik}^2)^{1/2}$.
- Let $T = (y_1^T, \ldots, y_n^T)^T$.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ with k-means algorithm into clusters $C_1, \ldots, C_k$.

$Output$: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$.

Spectral Clustering

Practical Details

Number of Clusters

A self-contained method: the eigengap heuristic. Denote the eigenvalues of the Laplacian matrix as

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

We choose k that maximize

$$\gamma_k := \left|\lambda_k - \lambda_{k+1}\right|, \; k = 1, \ldots, n-1.$$

Spectral Clustering

Practical Details

Type of Laplacian Matrix

If all degrees are nearly equal, then three Laplacians are similar. Otherwise, consider the case $k = 2$, recall the objective of clustering:

- Minimize the between-group similarity $\Rightarrow$ minimize $cut(A, A^c)$.
- Maximize the within-group similarity $W(A, A)$ and $W(A^c, A^c)$.

Spectral Clustering

Practical Details

The within-group similarity is

$$W(A,A) = W(A,V) - W(A,A^c) = \mathrm{vol}(A) - cut(A,A^c).$$

Hence,

$$\begin{aligned} &\text{Maximize } W(A,A) \\ \Rightarrow\ & cut(A,A^c) \text{ is small and } \mathrm{vol}(A) \text{ is large} \\ \Rightarrow\ & \text{Minimizing the normalized cut.} \end{aligned}$$

On the contrary, the Ratio Cut maximizes $|A|$ and $|A^c|$, which are not necessarily related to $W(A,A)$.

$*$ Theoretical approval for the normalized Laplacians can be found in the literature. To sum up, the normalized Laplacians are prefered.

Spectral Clustering

Practical Details

To select from the two normalized Laplacians:

- the eigenvectors of L_{rw} are cluster indicators, while the eigenvectors of L_{sym} are additionally multiplied with $D^{1/2}$;
- Using L_{sym} does not have any computational advantages.

Thus, L_{rw} is prefered.

Outline

1. Introduction
   - Hierarchical Clustering Methods
   - Partition Clustering Methods
   - Self-Organizing Map
   - Convex Clustering
2. Spectral Clustering
   - Introduction
   - Derivation
   - Summary
   - Practical Details
3. Dirichlet Process
   - Motivation
   - Definition
   - Clustering

Dirichlet Process

Motivation

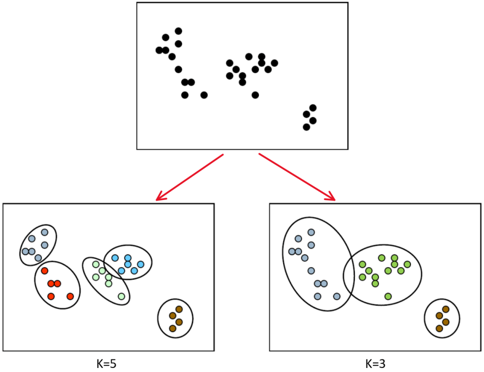
Gaussian mixture (Linear Discriminant Analysis)

Given data $x_1, \ldots, x_n \in \mathbb{R}^2$, we are told that x_i's are drawn from a mixture of K distinct Gaussian populations, not knowing the value of K. How to fit the mixture model?

We can try different values of K, and run EM algorithm to estimate the corresponding parameters.

Dirichlet Process

Motivation



Dirichlet Process

Motivation

What if the sample size is large?

- Sequential fitting plus model selection can be time consuming.
- Both under and over-fitting are possible due to unobserved heterogeneity.

Dirichlet Process

Motivation

Alternative: infer the number of clusters from the data.

$$x_i|\theta_i \sim F_{\theta_i}, \quad i = 1, \ldots, n.$$
$$\theta_i \sim G, \quad i = 1, \ldots, n.$$

$*$ $\{F_\theta : \theta \in \Theta\}$ is a family of parametric distribution.
$*$ $\theta_i = \theta_j \Rightarrow x_i$ and x_j belongs to the same cluster.
$*$ G is the prior for θ.
We introduce the Dirichlet process to define G as the prior distribution of the non-parametric distribution.

Dirichlet Process

Definition

Recall the Dirichlet distribution:

$$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K) \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_K),$$

if

$$P(\boldsymbol{\theta}) = \frac{\Gamma(\sum\limits_{k=1}^{K} \alpha_k)}{\prod\limits_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1},$$

where $\alpha_k > 0$, $\theta_k \geq 0$ for $k = 1, \ldots, K$ and $\sum\limits_{k=1}^{K} \theta_k = 1$.

$*$ A distribution of multinomial distributions with K categories.

Dirichlet Process

Definition

The Dirichlet Process

Definition

Let G_0 be a non-atomic distribution over Θ, $\alpha > 0$ is a real number. We say that G is a Dirichlet process, denoted by $G \sim \mathrm{DP}(\alpha, G_0)$, if

$$(G(\Theta_1), \ldots, G(\Theta_k)) \sim \mathrm{Dir}(\alpha G_0(\Theta_1), \ldots, \alpha G_0(\Theta_k))$$

for every finite partition of Θ: $\Theta = \cup_{i=1}^{k} \Theta_i,\ k = 1, 2, \ldots$

$*$ G_0 is called the base distribution.

$*$ α is a scaling parameter.

Dirichlet Process

Definition

Properties

For any $\Theta_0 \subset \Theta$, it can be shown that

$$E(G(\Theta_0)) = G_0(\Theta_0).$$

$$Var(G(\Theta_0)) = \frac{G_0(\Theta_0)(1 - G_0(\Theta_0))}{\alpha + 1}.$$

$* \ \alpha \to \infty \Rightarrow G \to G_0$ pointwise.

Dirichlet Process

Definition

Existence

Consider the hierachical model:

$$\begin{aligned} \theta_i &\sim G, \\ G &\sim \mathrm{DP}(\alpha, G_0). \end{aligned} \tag{1}$$

It can be shown that the posterior is

$$G|\theta_1, \ldots, \theta_n \sim \mathrm{DP}(\alpha + n, \frac{1}{\alpha + n}(\alpha G_0 + \sum_{i=1}^{n} \delta_{\theta_i}),$$

where δ_θ is the dirac measure at $\theta \in \Theta$.

Dirichlet Process

Definition

Since $\theta_1, \ldots, \theta_{n+1}$ are conditional independent given G, we have

$$\theta_{n+1} | G, \theta_1, \ldots, \theta_n \sim G.$$

For any measurable $A \subset \Theta$, we have

$$\begin{aligned} P(\theta_{n+1} \in A | \theta_1, \ldots, \theta_n) &= E[G(A) | \theta_1, \ldots, \theta_n] \\ &= \frac{1}{\alpha + n} \Big(\alpha G_0(A) + \sum_{i=1}^{n} \delta_{\theta_i}(A) \Big). \end{aligned}$$

Dirichlet Process

Definition

Thus,

$$\theta_{n+1}|\theta_1, \ldots, \theta_n \sim \frac{1}{\alpha+n}\Big(\alpha G_0 + \sum_{i=1}^{n} \delta_{\theta_i}\Big). \tag{5}$$

It follows from (5) and the chain rule that

$$P(\theta_1, \ldots, \theta_n) = P(\theta_{\sigma(1)}, \ldots, \theta_{\sigma(n)}),$$

where $(\sigma(1), \ldots, \sigma(n))$ is an arbitary permutation of $(1, \ldots, n)$. So the existence of G follows from de Finetti's theorem.

Dirichlet Process

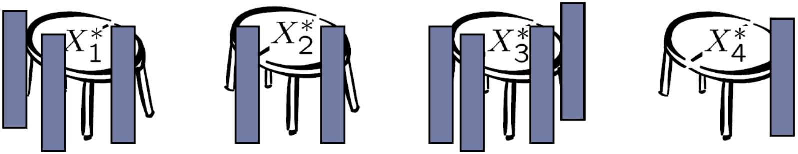
Definition

Construction

Chinese Restraunt Process

The predictive distribution (5) implies a sequential process of generating $\{\theta_i\}_{i=1}^{\infty}$. Let $\theta_1^*, \ldots, \theta_K^*$ be the unique values of $\theta_1, \ldots, \theta_n$, then

$$\theta_{n+1}|\theta_1, \ldots, \theta_n = \begin{cases} \text{a certain } \theta_k^* & \text{with probability } \frac{\#\{\theta_i : \theta_i = \theta_k^*\}}{n+\alpha} \\ \text{new draw from } G_0 & \text{with probability } \frac{\alpha}{n+\alpha}. \end{cases}$$

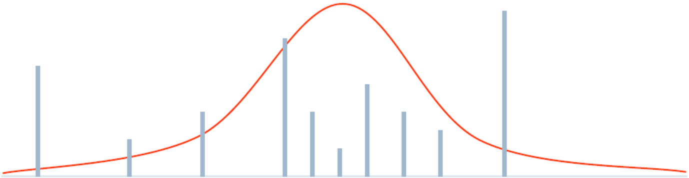


Dirichlet Process

Definition

Stick Breaking

- Draw $\theta_1^*, \ldots, \theta_k^*, \ldots$ from G_0.
- Draw $v_1, \ldots, v_k, \ldots$ from $\mathrm{Beta}(1, \alpha)$.
- Set $\pi_i = v_i \prod_{j=1}^{i-1} (1 - v_j)$.
- Set $G = \sum_{i=1}^{\infty} \pi_i \delta_{\theta_i^*}$, where δ_x is the dirac measure at x.



Dirichlet Process

Clustering

Dirichlet Process Mixture

The complete model:

$$\begin{aligned} x_i|\theta_i &\sim F_{\theta_i}; \\ \theta_i|G &\sim G; \\ G|\alpha, G_0 &\sim \mathrm{DP}(\alpha, G_0). \end{aligned} \tag{6}$$

Denote by z_i the cluster indicators:

$$z_i = k \text{ if } \theta_i = \theta_k^*, \quad i = 1, \ldots, n.$$

Denote by $\pi_k = P(z_i = k), k = 1, \ldots$

Dirichlet Process

Clustering

With $G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*}$, model (6) is equivalent to

$$\begin{aligned} \pi|\alpha &\sim \mathrm{GEM}(\alpha); \\ z_i|\pi &\sim \mathrm{Multinomial}(\pi); \\ \theta_k^*|G_0 &\sim G_0; \\ x_i|z_i, \theta_k^* &\sim F_{\theta_{z_i}^*}. \end{aligned} \tag{7}$$

where $\mathrm{GEM}(\alpha)$ stands for the distribution of $\pi = (\pi_1, \ldots)$ constructed from stick breaking.

Dirichlet Process

Inference

Clustering based on (7): Bayesian inference.

- MCMC.
- Variational inference.

$*$ The Dirichlet process specifies a infinite mixture, while only finite components are active, whose number could increase with sample size.

References I

[1] Khalid El-Arini. "Dirichlet Processes: A gentle tutorial".

[2] David M. Blei and Michael I. Jordan. "Variational inference for Dirichlet process mixtures". In: *Bayesian Analysis* 1.1 (2006), pp. 121–143.

[3] Jianqing Fan et al. *Statistical foundations of data science*. Chapman and Hall/CRC, 2020.

[4] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[5] Ulrike von Luxburg. "A tutorial on spectral clustering". In: *Statistics and Computing* 17.4 (2007), pp. 395–416.

[6] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

References II

[7] Radford M. Neal. "Markov Chain Sampling Methods for Dirichlet Process Mixture Models". In: *Journal of Computational and Graphical Statistics* 9.2 (2000), pp. 249–265.

[8] Amit Saxena et al. "A review of clustering techniques and developments". In: *Neurocomputing* 267 (2017), pp. 664–681.

[9] Yee Whye Teh. "Dirichlet Process".

[10] Juha Vesanto and Esa Alhoniemi. "Clustering of the self-organizing map". In: *IEEE Transactions on neural networks* 11.3 (2000), pp. 586–600.