Quality and Usability Seminar-Clustering

Mohamed Mesto

January 16, 2022

0.1 Supervised and Unsupervised Learning

What are the differences between unsupervised and supervised Learning? What Is Semi-Supervised Learning? Why the massive demand for them? Which domains can they serve?

Towards the desire to improve human life and in conjunction with the growing requirements and needs of consumers over time, the demand has become urgent to develop and employ artificial intelligence and machine learning algorithms to achieve the aspirations of customers in intelligent life.

Unsupervised learning is used to discover patterns from a provided unlabeled dataset. In this method, the algorithms are implemented without human interposition. E.g. (Clustering, Association, Dimensionality reduction). However, **supervised learning** is an algorithm category that specifies a predictive model utilizing data points with known outcomes. In other words, supervised learning is the methodology of training a model by providing input and correct output data. The training data consists of a collection of training instances. E.g. (Classification, Regression). Moreover, **the Semi-Supervised Learning** is a mixed of both types. I.e., It is a training dataset with unlabeled and labeled data.

Ref: * [16]

- [11]

References are created in BibTex APA style (Latex Template form) The References are created in BibTex APA style (Latex form) and can be seen after generating the Latex Template file as Pdf automatically by implementing the last code cell in this colab notebook. Where: * **citations.tplx:** An external file for the Latex Template * **ref.bib:** An external file for the references

0.2 Clustering Methods:

What does the Clustering mean?

Clustering is an unsupervised approach applied on unlabeled datasets. It aims to collect them into combinations depending on their relationships (such as K-means, Gaussian Mixture Models, and Agglomerative Clustering).

0.2.1 K-means Algorithm:

What is K-means Algorithm? when we need it?

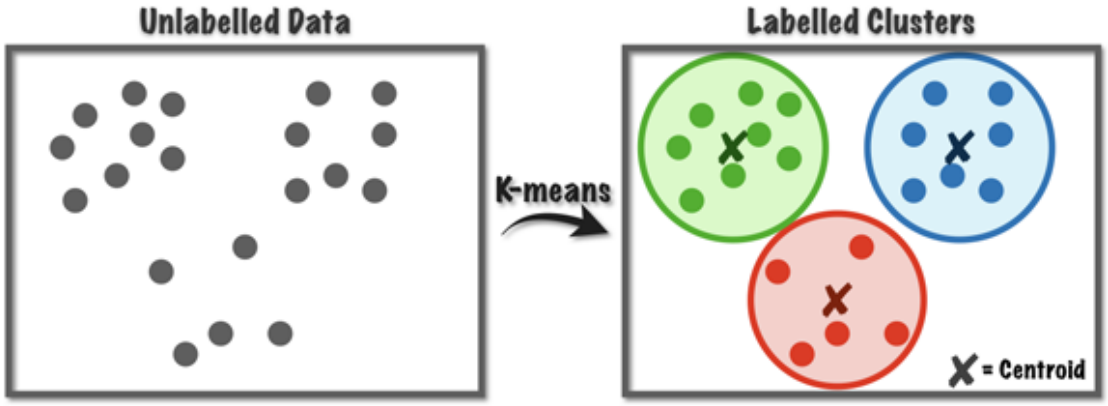
It is an approach used for clustering analysis, primarily in data mining and statistics. It performs the gathering/clustering of unlabeled datasets into groups, where (K) is the number of

clusters. Nevertheless, this method is essential for marketing/customer segmentation, document clustering, image segmentation, compression, and Delivery Store Optimization.

Ref: * [9]

[130]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/K-means/
↪labelled_unlabelled_data.png')
```

[130]:



Ref: * [10]

- [6]

K-means Algorithm steps:

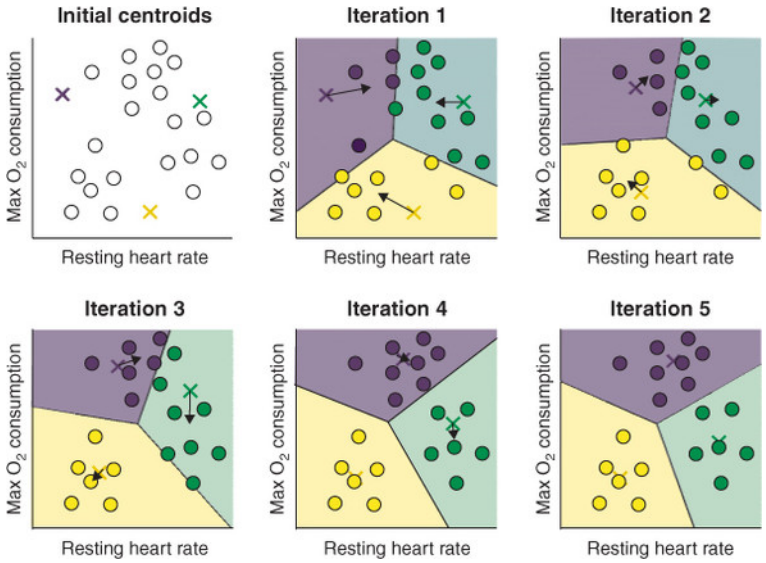
1. Choose the number of clusters "K".
2. Set the initial K-points randomly into the sample data space, representing the beginning combinations of clusters (centroids).
3. Assign each data point to the nearest K (centroid).
4. Calculate the mean of each cluster and move the centroids.
5. Repeat Steps 3 & 4 continuously until the positions of the centroids are no longer changed.

Ref: * [2]

An Example Diagram for K-maens Algorithm steps:

[131]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/K-means/fig16-1_alt.
↪png')
```

[131]:



Ref: * [9]

Hyperparameter Tuning: (Choosing optimal Number of "K")

- **Elbow Method**

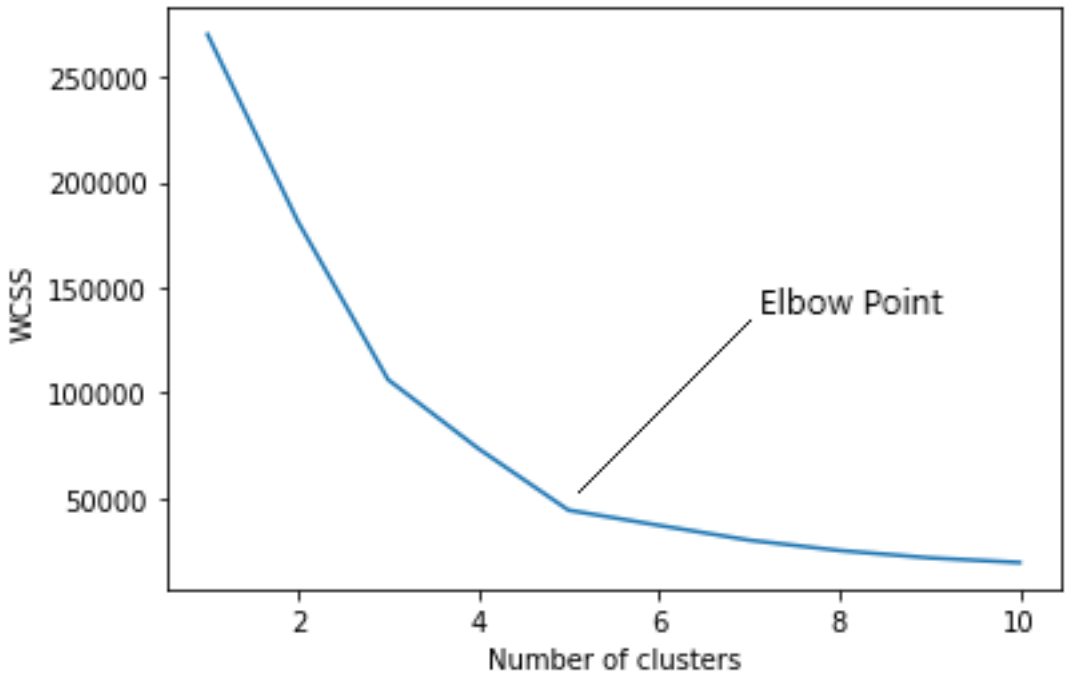
1. We Choose "K" manually, Through visualization.
2. Calculate the distance between points in a cluster (With-in Cluster Sum of Squares "WCSS")
3. If we minimize "WCSS," we have reached the perfect clustering solution.

   Within Cluster Sum of Squares (WCSS), also known as Inertia, is the sum of squared distance between each point and the centroid in a cluster. The plot looks like an Elbow when plotting the WCSS with the "K" value. Other Methods

- **Silhouette Method**

[132]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/K-means/Elbow-method.
 ↪png')
```

[132]:



Ref: * [10]

K-Means Disadvantages: **K-Means** Clustering Algorithm has the following disadvantages-

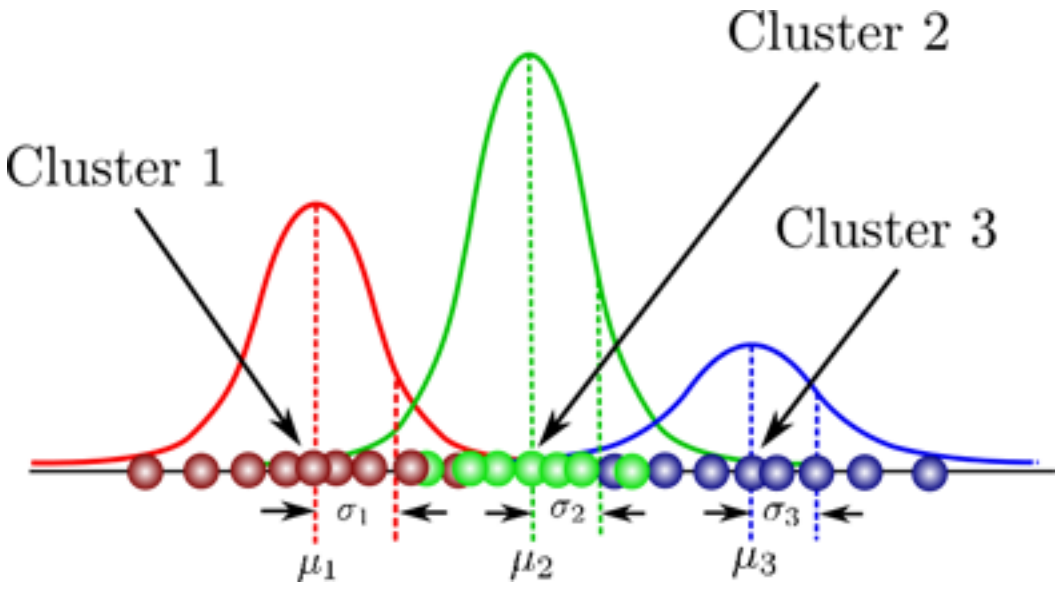
- It needs to determine the number of clusters (k) in advance.
- Unable to deal with noisy data and outliers.
- Difficulties with specifying clusters with non-convex shapes.
- Scaling with number of dimensions: which can solved by Principal component analysis (PCA) or by using "spectral clustering" to modify the clustering algorithm.

Ref: * [5]
Gaussian Mixture Models (GMMs):
What are the Gaussian Mixture Models? What does it relate to the Gaussian Distribution? What makes GMMs a better candidate than K-means?
Before diving into Gaussian Mixture Models, let us look at the "Gaussian Distribution."

Gaussian Distribution is also known as Normal Distribution. The following graph represents a few Gaussian distributions with mean (μ) and variance (σ2) differences. The spread and σ are directly proportional. Nevertheless, It is a bell-shaped curve with the data points harmoniously dispersed around the mean value.

[133]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Gaussian Mixture␣
↪Models/2D_bell_shaped_curve.png')
```

[133]:

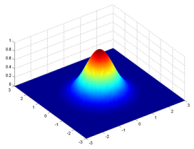


Ref: * [1]

By dealing with two variables, The curve's shape will be a 3D bell curve as displayed below:

[134]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Gaussian Mixture␣
 ↪Models/gd2.png',height=300,width=400
      )
```

[134]:



For the Gaussian distribution's probability density function, we distinguish the following cases: 1. In one-dimensional space, it is given by the following eqaution:

where μ is the mean and $\sigma 2$ is the variance.

[135]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Gaussian Mixture␣
 ↪Models/GD.png' , height=130,width=350 )
```

[135]:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

2. In a two-dimensional space, it is given by:

Where: * x: describes the input vector. * μ: represents the 2D mean vector. * Σ: defines the 2×2 covariance matrix.

[136]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Gaussian Mixture␣
↪Models/GD-3D-.png')
```

[136]:

$$f(x \mid \mu, \Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left[-\tfrac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)\right]$$

3. In a d-dimensional space (multivariate Gaussian model),Where:

- x , μ as vectors of length d.
- Σ: defines the $d\times d$ covariance matrix. it is also possible to generate the equation!

Accordingly, we get the general rule: The method result will be a combination or mixture of k Gaussian distributions if the input is a dataset of d features(where k is equal to that cluster number), where: * Each distribution has a specific mean vector and variance matrix

This guides us to the "Expectation-Maximization (EM)" approach to calculate each Gaussian distribution's mean and variance value.

Ref: * [12]

Expectation-Maximization Algorithm (EM) **Expectation-Maximization** is one of the widely used algorithms in statistical analysis. It depends on an iterative method to find the suitable model parameters by accomplishing maximum likelihood estimation.

In order to empower the statistical algorithms to be able to apply to the dataset (which includes some missing data values), utilizing EM would be the best candidate. These missing variables (unobserved) are named latent variables, e.g. (k) or cluster number. EM's process has the following steps:

1. **Expectation (E)-step:** we employ the supplied data to estimate (expect) the values of the missing parameters(latent variables) as if they were observed. and a probability distribution is constructed.
2. **Optimizing the model**: Recently processed data will provided into the model.
3. **Maximization (M)-step:** The probability distribution parameters from step one are updated to implicate the new model/data.
4. Repeating Steps 2, 3, and 4 until steadiness is achieved (i.e., a distribution values no more changes).

Ref: * [12]

- [15]

Expectation-Maximization (EM) in Gaussian Mixture Models (GMMs) After understanding the EM algorithm, let us use it in GMMs.

To compute the GMMs, we need to find the values of the variables μ, Σ, and Π.

For that we need the following Assumptions at the beginning:

- k: is the number of clusters => k Gaussian distributions.
- Mean Values: $\mu 1$, $\mu 2$, .. μk
- Covariance values : $\Sigma 1$, $\Sigma 2$, .. Σk
- Πi: is the density of the distribution.

The next stage will perform the EM steps 1-4 including the main steps(E-step and M-step):

1. **Expectation (E)-step** : we employ the supplied data to estimate (expect) the values of the missing parameters(latent variables) as if they were observed. and a probability distribution is constructed.

Utilizing the formula below, we can compute the probability that it belongs to cluster/distribution c1, c2, ... ck.

[137]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Gaussian Mixture␣
↪Models/e-step.png')
```

[137]:

$$r_{ic} = \frac{\text{Probability Xi belongs to c}}{\text{Sum of probability Xi belongs to } c_1, c_2, .. c_k} = \frac{\pi_c \mathcal{N}(x_i \ ; \ \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \ ; \ \mu_{c'}, \Sigma_{c'})}$$

As proof of the correctness of the result, the value must be small if the data point is appointed to not the correct cluster and vice versa.

2. **Optimizing the model** update the μ, Σ ,and Π values usinf the following formulas.
3. **Maximization (M)-step:** The probability distribution parameters from step one are updated to implicate the new model.

The new density can be calculated as follows:

[138]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Gaussian Mixture␣
↪Models/m-step-1.png')
```

[138]:

$$\prod = \frac{\text{Number of points assigned to cluster}}{\text{Total number of points}}$$

The new mean and the covariance matrix can be calculated as follows:

[139]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Gaussian Mixture␣
↪Models/m-step-2.png')
```

[139]:

$$\boldsymbol{\mu} = \frac{1}{\text{Number of points assigned to cluster}} \sum_i r_{ic} x_i$$

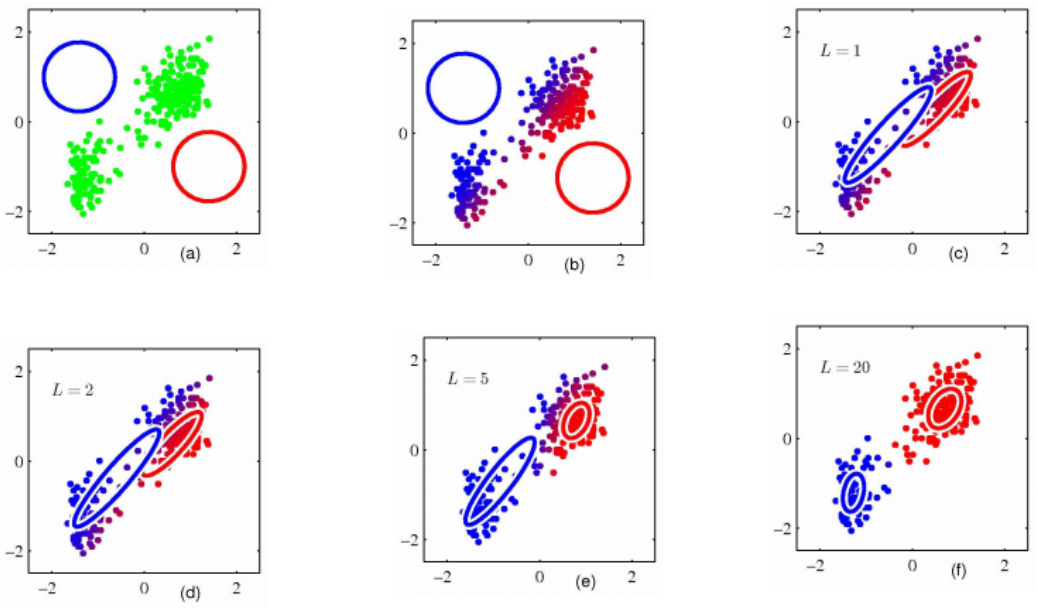
$$\Sigma_c = \frac{1}{\text{Number of points assigned to cluster}} \sum_i r_{ic} (x_i - \boldsymbol{\mu}_c)^T (x_i - \boldsymbol{\mu}_c)$$

4. **Repeating Steps 2, 3, and 4**: Iterate until steadiness is achieved; (i.e., a distribution values no more changes).

An Illustration of Expectation Maximization Algorithm (EM)

[140]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Gaussian Mixture␣
↪Models/EMIterations.png')
```

[140]:



Ref: * [17]

Gaussian Mixture Models (GMMs) After getting to know the **Gaussian Distribution (GD)** and **Expectation-Maximization (EM)** and concepts, we can simply complete discovering the **Gaussian Mixture Models (GMMs)** planet.

Gaussian Mixture Models (GMMs) is one of the most famous clustering algorithms.It uses the Gaussian, which is a method for plotting data. However, it differs from the K-mean algorithm because it considers variance. In order to understand how this method works, let us say we have a certain number of Gaussian distributions, where each distribution represents a cluster. Thus, the Gaussian mixture models algorithm aims to group the data points of each distribution into one set.

Variance is the method for expanding data in the cluster.

E.g. , the following photograph formed 4 Clusters, and it shows us how it expands because the GMMs algorithm works on the mean and variance to form the Clusters, contrariwise the K-mean algorithm that performs better on circular-shaped data.

For a good understanding of the k-mean algorithm, we will look at the following example:

Assuming that we want to apply the k-mean algorithm to a set of data and we have three clusters, then each point belongs to a particular cluster. So the following cases are considered a challenge to the k-average algorithm:

1. If data points belong to the first cluster with a certain probability and to the second cluster with another probability.
2. If we have an overlap between two clusters within the data space, because the data points belong to two different types of data (a mixture)

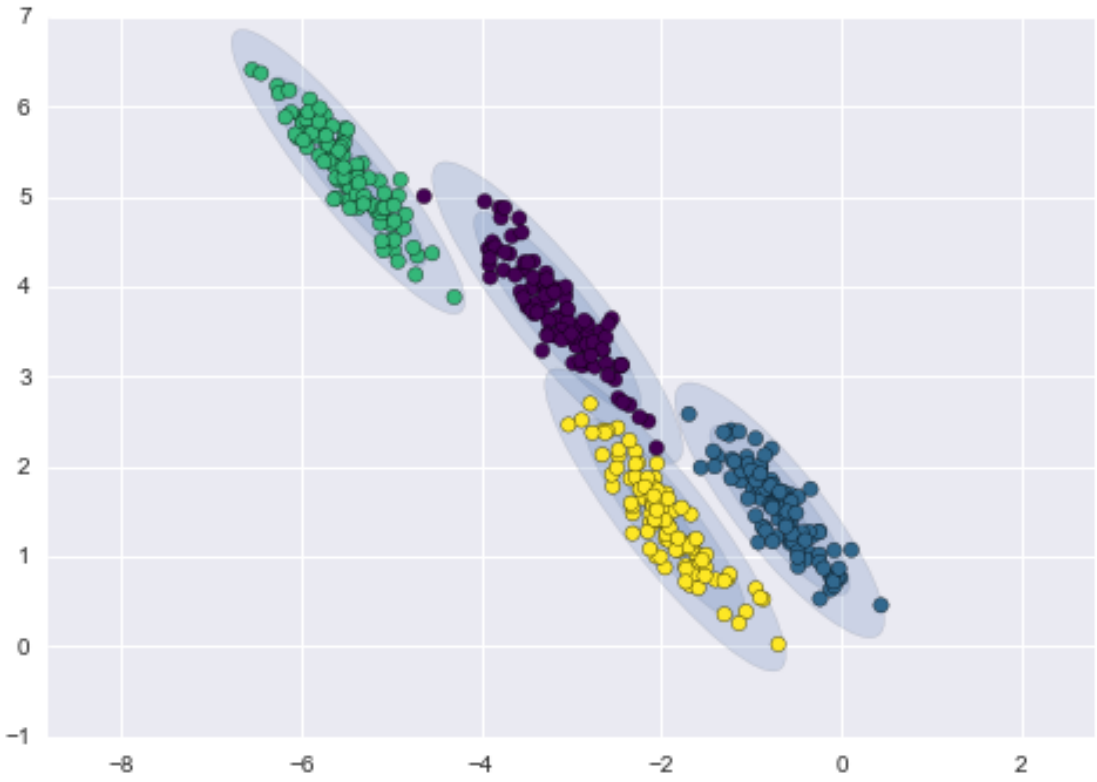
The applicable solution for them is GMMs. because GMMs are probabilistic models and use the soft clustering approach for distributing the points in different clusters. Nevertheless, k-means

considers only the mean to update the centroid while GMMs takes into account the mean as well as the variance of the data. Therefore, the k-means is called a hard assignment.

Ref: * [13] * [12]

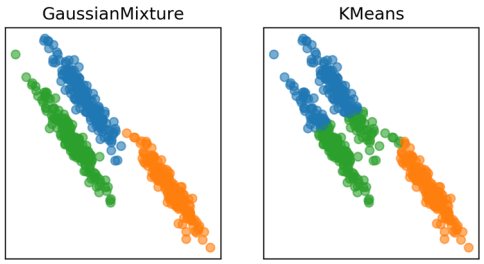
[141]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Gaussian Mixture␣
↪Models/gmm.png')
```

[141]:



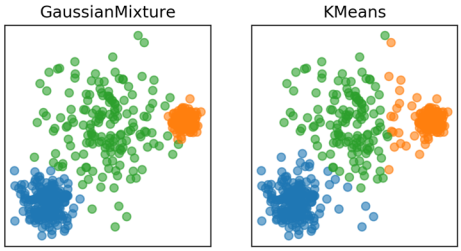
[142]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Gaussian Mixture␣
↪Models/gmm_vs_kmeans_1.png')
```

[142]:



[143]:
```
from IPython.display import Image
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Gaussian Mixture␣
↪Models/gmm_vs_kmeans_2.png')
```

[143]:



Ref: * [8]

0.2.2 Agglomerative Clustering/Hierarchical Clustering:

What is the Agglomerative Clustering Algorithm? what are the similarities and differences between it and the K-means?

Hierarchical Clustering Algorithms: Hierarchical Clustering (HC) or Hierarchical Clustering Analysis (HCA) is a clustering algorithm used in statistical analysis. It aims to analyze and plot the studying data and present the clusters in a hierarchy diagram. It has two different strategies:

1. Agglomerative Clustering :

- A "bottom-up" method
- The analysis or observation concern is building a hierarchical structure of clusters with a specific sequence from bottom to up. I.e., it is performed by plotting/joining the nearest data points to create a cluster on the Dendrogram (Initially, each data point is a cluster of its own). The next step will plot or join the nearest clusters together from the bottom toward the top of the hierarchy.

2. Divisive

- A "top-down" method
- initially, all the data points in the dataset belong to one cluster. The analysis and observation concern is splitting from the top toward down.

Ref: * [14]

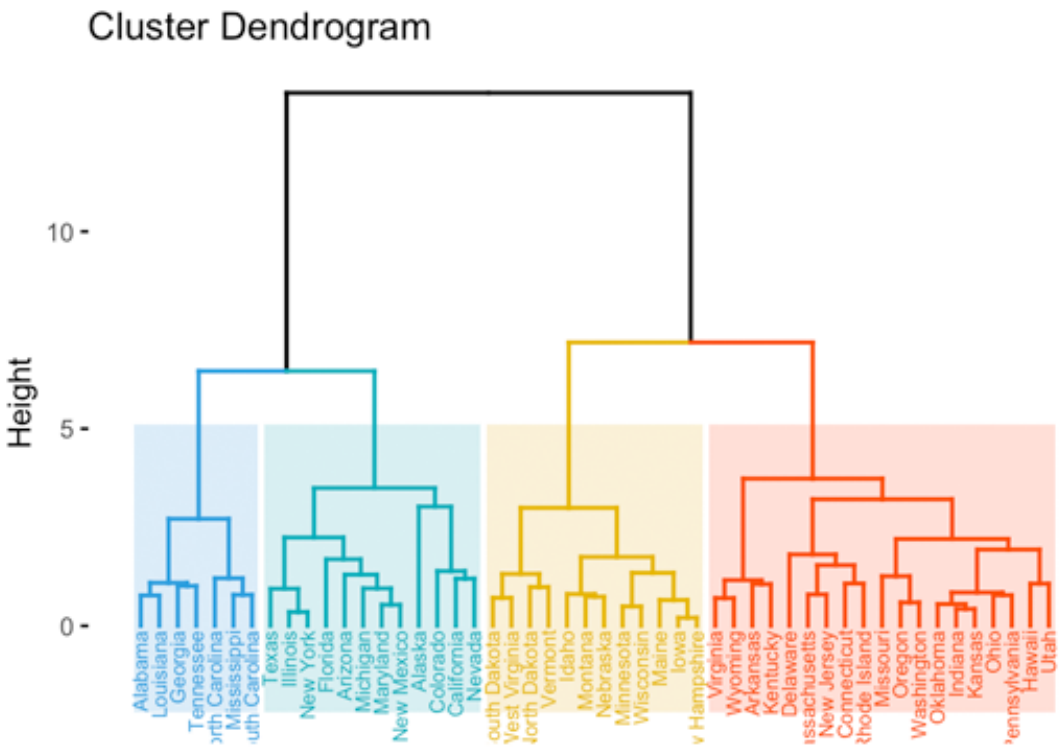
- [7]

An examples for Agglomerative Clustering and Dendrogram

[146]:
```
from IPython.display import Image
```

```
Image('/content/drive/MyDrive/Quality_and_Usability_Seminar/Agglomerative␣
↪Clustering/dendrogram.png')
```

[146]:



Ref: * [4]

Hierarchical Clustering (HC) VS K-Means K-Means meets its goal using the centroid and calculating the distances between the data points. Whereas in Hierarchical Clustering no need to appoint the number of clusters in advance. HC and K-means are both clustering algorithms in the statistical analysis field. However, to be efficient, we should utilize the K-means in the following conditions: * The provided dataset has a particular number of clusters, but they belong to an unknown group. * The provided dataset has a large number of variables. For fast computing, use **K-means.**

Nevertheless, using **Hierarchical Clustering** will be more useful in the following conditions: * The determines are based on previous opinions. **HC** should be used to know the number of clusters. * The demand is high to determine the number of clusters more easily. **HC's dendrogram** is the right decision.

Moreover, the K-means approach's result is unstructured. However, **Hierarchical Clustering** one is more enlightening and interpretable.

Ref: * [3]

Agglomerative Clustering Algorithm steps:

1. Initially, We should consider that every data point represents a cluster on its own.

2. Every two nearest clusters could be joined with each other to form one single cluster.
3. Repeat step 2 until the whole number of clusters are included.

0.3 Libraries/packages to export/generate the report as HTML, and PDF file with the References in BibTex APA style (Latex Template

In order to export/generate the report as HTML or PDF.we need to install the following libraries/packages:

[ ]:
```
# In order to export/generate the report as HTML or PDF
#we need to install the following libraries/packages:
! sudo apt-get update
!sudo apt-get install texlive-latex-extra
!sudo apt-get install texlive-pdflatex-extra
!sudo apt-get install texlive-bibtex-extra
!sudo apt-get install texlive-xetex texlive-fonts-recommended␣
 ↪texlive-plain-generic texlive-generic-recommended
!sudo apt-get install texlive texlive-latex-extra pandoc
! sudo apt-get install texlive-latex-recommended
! sudo apt-get install dvipng
! pip install nbconvert
! sudo apt-get install pandoc
!pip install pdftex
!pip install pdflatex
# !pip install folium==0.2.1
```

0.4 Contents of the Latex Template and Refernces:

* **citations.tplx:** An external file for the Latex Template * **ref.bib:** An external file for the references**

citations.tplx:** An external file for the Latex Template * **ref.bib:** An external file for the references

```
# This is formatted as code ((*- extends 'article.tplx' -*)) ((* block
author *)) \author{Mohamed Mesto} ((* endblock author *)) ((* block title
*)) \title{Quality and Usability Seminar-Clustering} ((* endblock title *))
\usepackage{url} \usepackage{hyperref} \usepackage{graphicx} \usepackage{tcolorbox}
((* block bibliography *)) \bibliographystyle{plain} \bibliography{ref} ((*
endblock bibliography *))
```

[ ]:
```
# export as HTML without the refernces
! ipython nbconvert   --to html  /content/Topic9_Clustering_theory.ipynb
```

[ ]:
```
%%bash
jupyter nbconvert Topic9_Clustering_theory.ipynb --to latex --template citations.
 ↪tplx
pdflatex Topic9_Clustering_theory.tex
bibtex   Topic9_Clustering_theory.aux
```

```
pdflatex Topic9_Clustering_theory.tex
pdflatex Topic9_Clustering_theory.tex
pdflatex Topic9_Clustering_theory.tex
```

[ ]:

References

[1] Oscar Contreras Carrasco. Gaussian Mixture Models Explained. https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95, Jun 3, 2019.

[2] Imad Dabbura. K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks. https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a, Sep 17, 2018.

[3] Vikash Kumar Das. K-Means Clustering vs Hierarchical Clustering. https://www.globaltechcouncil.org/clustering/k-means-clustering-vs-hierarchical-clustering/, 11 October, 2020.

[4] datanovia. Hierarchical Clustering in R: The Essentials. https://www.datanovia.com/en/lessons/examples-of-dendrograms-visualization/, 2008.

[5] developers google. clustering algorithm advantages-disadvantages. https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages, 2021-01-13.

[6] Luigi Fiori. K-Means Clustering using Python. https://medium.com/@luigi.fiori.lf0303/k-means-clustering-using-python-db57415d26e6, Jun 6, 2020·4 min read.

[7] Satyam Kumar. Hierarchical Clustering: Agglomerative and Divisive — Explained. https://towardsdatascience.com/hierarchical-clustering-agglomerative-and-divisive-explained-342e6b20d710, Aug 2, 2020.

[8] Andreas C. Müller. Clustering and Mixture Models. https://amueller.github.io/COMS4995-s18/slides/aml-16-032118-clustering-and-mixture-models/#1, 03/26/18.

[9] Hefin I. Rhys. Chapter 16. Clustering by finding centers with k-means. https://livebook.manning.com/book/machine-learning-for-mortals-mere-and-otherwise/about-this-book/, Manning, 2015.

[10] Basil Saji. In-depth Intuition of K-Means Clustering Algorithm in Machine Learning. https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/, January 20, 2021.

[11] Isha Salian. SuperVize Me: What's the Difference Between Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning? https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/, August 2, 2018.

[12] Aishwarya Singh. Build Better and Accurate Clusters with Gaussian Mixture Models. https://www.analyticsvidhya.com/blog/2019/10/gaussian-mixture-models-clustering/, October 31, 2019.

[13] Jake VanderPlas. *Python Data Science Handbook*. O'Reilly Media, Inc., Released November 2016.

[14] wikipedia. Hierarchical clustering algorithm. https://en.wikipedia.org/wiki/Hierarchical_clustering, 11 November 2021.

[15] wikipedia. Expectation–maximization algorithm. https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm, 7 January 2022,.

[16] Evgenii Zheltonozhskii. Self-Supervised Learning for Large-Scale Unsupervised Image Clustering. https://deepai.org/machine-learning-glossary-and-terms/supervised-learning/, 08/24/2020.

[17] Ivan Zou. Mixture Models and EM Algorithm – Basics of Unsupervised Learning. https://www.sap-techs.com/mixture-models-and-em-algorithm-basics-of-unsupervised-learning/, 15 Jun 2021.