Unsupervised Learning and Clustering

Vibha Mane
The COSINE Lab
Department of Electrical and Computer Engineering

Topics

- Types of Machine Learning Tasks
- Types of Clustering
- Clustering Algorithms
  - k-means
  - Hierarchical
  - DBSCAN
- Clustering Algorithms - Advantages & Limitations
  - k-means
  - Hierarchical
  - DBSCAN

Types of Machine Learning Tasks

- The machine learning tasks are broadly classified into three categories:
- **Supervised Learning:** The data set consists of several example inputs and outputs, and the goal is to find a general rule that maps the inputs into outputs, such as image classification.
- **Unsupervised Learning:** The data set **contains no labels (or target features)**, and the goal is to find interesting or hidden patterns in the data, for example, discovering graph structure.
- **Reinforcement Learning**: The data is given as a feedback and the program take an action so as maximize a reward, for example, playing the mancala game.

Clustering

- Clustering is the task of finding **similar or homogeneous groups** in data.
- When we cluster the observations of a data set, we seek to **partition** them into distinct groups so that the observations within each group are very similar to each other, while observations in different groups are very different from each other.
- Clustering is an example of **unsupervised learning**, and as such there is **no target feature or label**. It is an exploratory data analysis technique.
- As an example, in a **social network**, clustering can be used to recognize communities within large groups of people.
- As another example, in a **digital image**, clustering can be used to divide the image into distinct regions for object recognition.

Clustering, cont.

- As another example, suppose we have a set of n observations, each with p features.
- The n observations could correspond to tissue samples for patients with breast cancer.
- We may have reason to believe that there is some heterogeneity among the tissue samples; for instance, there are a few different unknown types of breast cancer.
- Clustering can be used to **find these subgroups**.
- This is an unsupervised problem, as we are trying to **uncover structure** – in this case, distinct structure, based on the data set.

Types of Clustering

Since there are many different “rules” for defining similarity between objects, there are many clustering algorithms.

- In **centroid-based clustering**, such as *k*-means clustering, objects are assigned to the nearest centroid cluster, such that the squared distances from the cluster are minimized.
- In **hierarchical clustering**, the data is grouped in different scales, by utilizing a cluster tree or dendrogram.
- The **density-based clustering** algorithm groups together objects that are closely packed, marking as outliers points that are in the low-density regions. An example is the **DBSCAN** (density-based spatial clustering of applications with noise) algorithm.

Centroid-based Clustering

- ***k*-means clustering** is an example of centroid-based clustering. The goal is to partition the n data points into ***k* clusters**, where each data point belongs to the cluster with the **closest mean** [3], [4].
- The closest maybe defined in terms of the **Euclidean distance**.
- The mean of a cluster is called the **centroid**. For example, in a data set with three features, namely x_1, x_2, x_3, the mean of a cluster would be computed by averaging over all points in that cluster, **for each feature separately**.

k-means Clustering

- Let $C_1, C_1, \cdots C_k$ denote the k clusters. The idea behind k-means clustering is that a good clustering is one for which **within-cluster variation** is as small as possible.
- The within-cluster variation for cluster $C_{k'}$ is a measure $W(C_{k'})$ of the amount by which the observations within a cluster differ from each other.
- Hence, we want to solve the problem [1]:

$$\underset{C_1, C_2, \cdots C_k}{minimize}\left\{\sum_{k'=1}^{k} W(C_{k'})\right\}.$$

k-means Clustering, cont.

- The above above expression says that we want to partition the observations into k clusters such that the total within-cluster variation , summed over all clusters is as small as possible.
- There are many choices for defining within-cluster variation. The squared Euclidean distance is one such choice. Therefore, we have for within-cluster variation

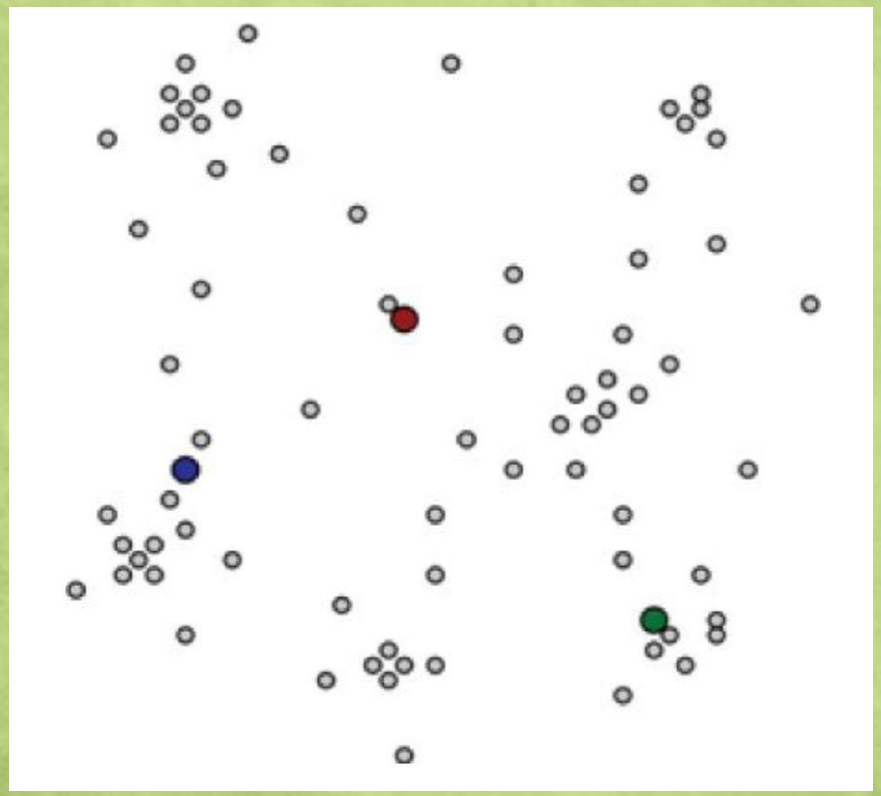
$$W(C_{k'}) = \frac{1}{|C_{k'}|}\sum_{i,i' \in C_{k'}} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2.$$

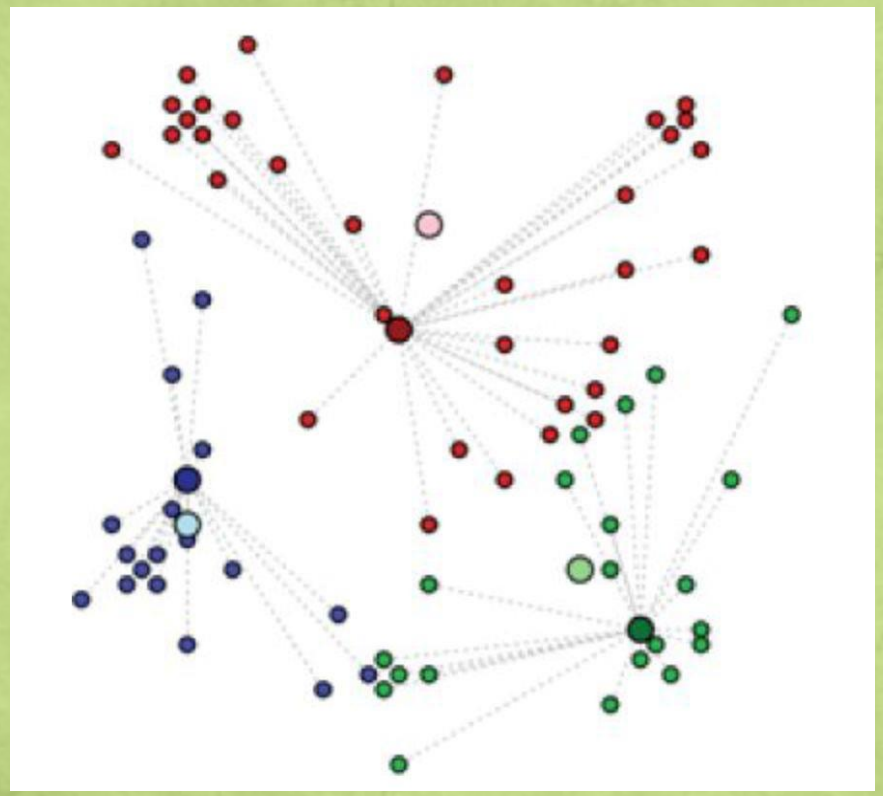
k-means Clustering Algorithm

The algorithm works as follows:

1. Choose the value of k, and initial guesses for the centroids of k clusters.
2. Compute the distance from each sample point to each centroid. Assign each point to the cluster with the closest centroid.
3. After assigning all the data points, compute the **new centroid** for each cluster.
4. Repeat Steps 2 and 3 until the algorithm converges. We reach convergence when the computed centroids do not change.
5. There are methods for determining k, the number of clusters.

k-means Clustering Algorithm, cont.





- The figure on the left shows the data points in black, and **initial centroid** values for $k = 3$ clusters, in three different colors [3].
- The figure on the right shows the data points assigned to the closest clusters, and the new centroids in lighter colors.

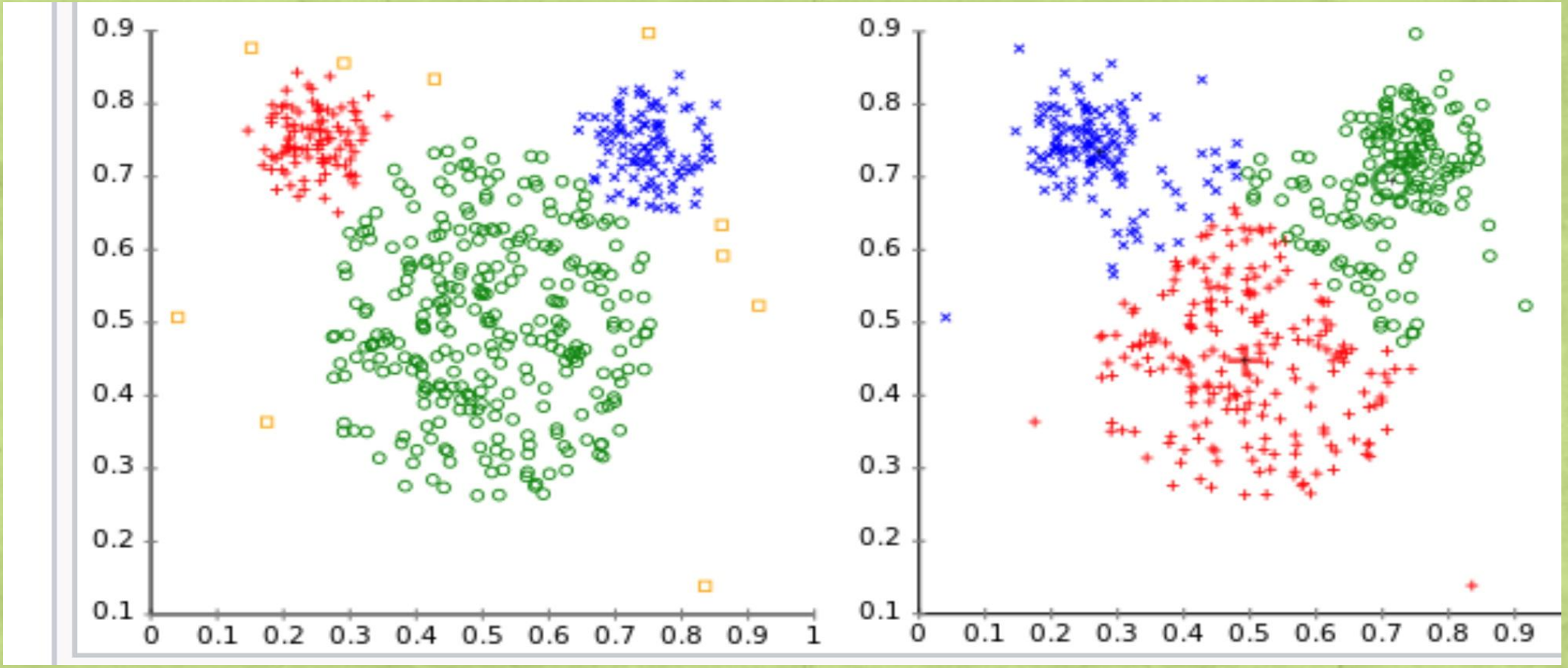
k-means Clustering - Advantages & Limitations

- Advantages
  - The algorithm is easy to implement.
  - It works well for large data sets.
- Limitations
  - The number of clusters *k* is an input parameter. A poor choice of *k* may produce poor results.
  - The algorithm tends to produce spherical and equal sized clusters.
  - The plot on the following page shows original clusters (left) and those produced by *k*-means clustering (right). The tendency to produce equal sized clusters with *k*-means leads to bad results [4].
  - This problem can be remedied with a generalized *k*-means algorithm [5].

k-means Clustering - Advantages & Limitations, cont.

Original Data

k-means Clustering



Hierarchical Clustering

- Hierarchical clustering groups data over a variety of scales by creating a **cluster tree or dendrogram** [6].
- The tree is not a single set of clusters, but a **multilevel hierarchy**.
- In **agglomerative or bottom-up** approach, each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- In order to decide which clusters should be merged, a measure of distance, for example, **Euclidean distance** is used.

Agglomerative Hierarchical Clustering Technique

- In this technique, initially each data point is considered as an individual cluster.
- At each iteration
  - Compute the **similarity matrix** between each pair of clusters.
  - Merge the two closest clusters.
  - Update the similarity matrix.
- The iterative steps are repeated **until a single cluster remains**.
- For N objects, the similarity matrix is an $N \times N$ matrix, which gives similarity between pairs of clusters.
- Some similarity measures are group average or distance between centroids.

Hierarchical Clustering Example - MVN Simulated Data (1 of 5)

- In this example, data is generated from a **multivariate normal distribution (MVN)**, with **three descriptive features** and **three classes** or labels. For more on MVN, see Appendix A.
- The parameters of MVN are a **mean vector** and a **covariance matrix**. For three descriptive features, the size of the mean vector is 3 rows and 1 column, and the size of the covariance matrix is 3 rows and 3 columns.
- In the following page, we give the mean vector and the covariance matrix for the three classes I, II and III.
- The three values 0, 2 and 3 in μ_I are the mean values for the three features, for class I.

Hierarchical Clustering Example - MVN Simulated Data (2 of 5)

- The diagonal values 1.2, 3.0 and 5.0 in Σ_I are the variances (or spread) of the three features in class I.
- The off-diagonal values 0.4, 0.7 and 0.0 give the correlation between the features. The value in row 1 and column 2 is 0.4; the value in row 1 and column 3 is 0.7. This indicates that features 1 and 3 are more strongly correlated compared with features 1 and 2.
- The covariance matrix is symmetric, that is, the value in row 1 and column 3 is the same as the value in row 3 and column 1.
- If you compare the values in μ_I with those in μ_{II} you can see that the two classes I and II are well separated in second and third features.

Hierarchical Clustering Example - MVN Simulated Data (3 of 5)

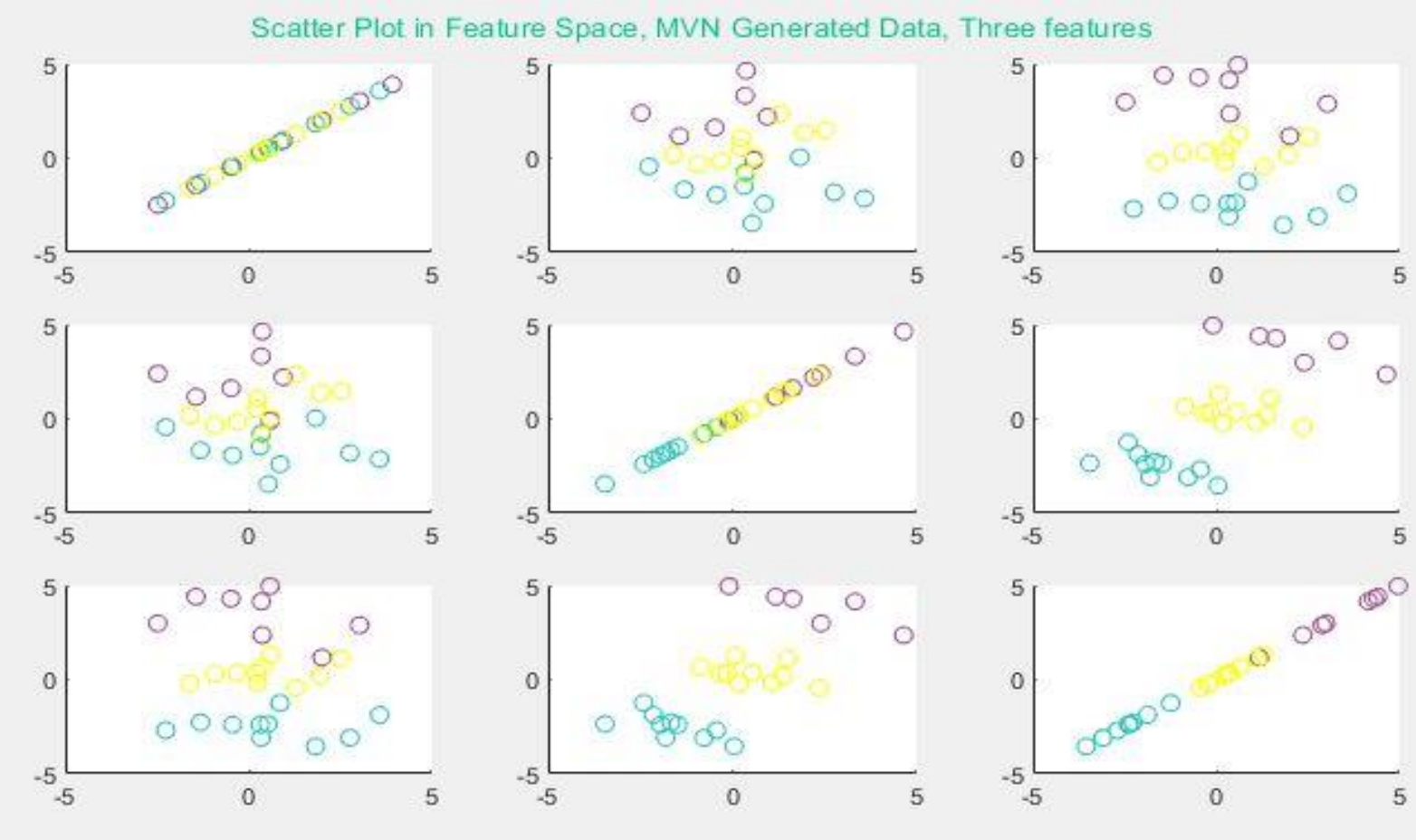
Data generated from Multivariate normal distribution (MVN) distribution, with three descriptive features and three classes or labels.

$$\mu_I = \begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix} \quad \Sigma_I = \begin{bmatrix} 1.2 & 0.4 & 0.7 \\ 0.4 & 3.0 & 0 \\ 0.7 & 0 & 5.0 \end{bmatrix}$$

$$\mu_{II} = \begin{bmatrix} 0 \\ -2 \\ -3 \end{bmatrix} \quad \Sigma_{II} = \begin{bmatrix} 1.0 & -0.4 & 0.2 \\ -0.4 & 1.0 & 0 \\ 0.2 & 0 & 1.0 \end{bmatrix}$$

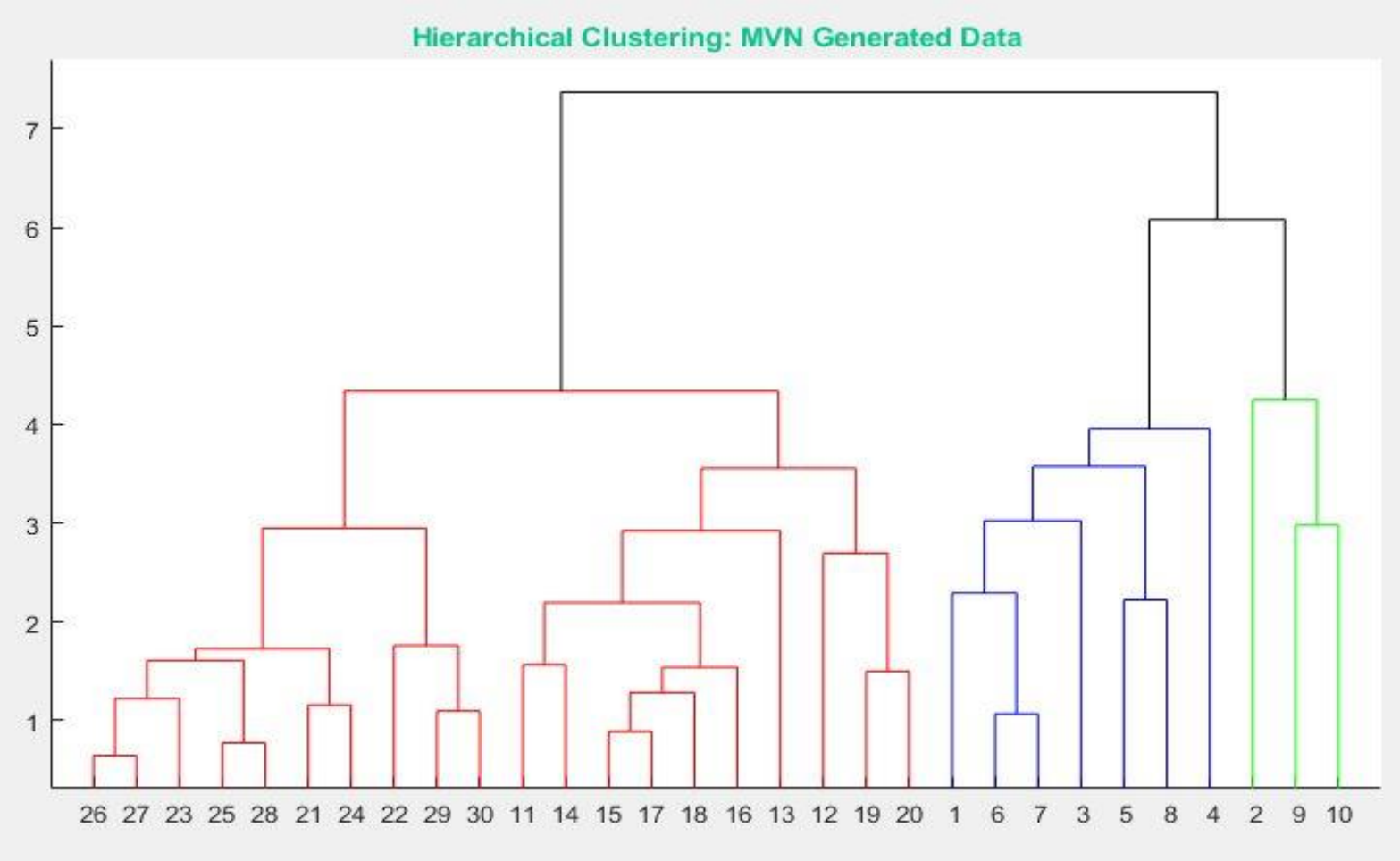
$$\mu_{III} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Sigma_{III} = \begin{bmatrix} 0.5 & 0.1 & 0.2 \\ 0.1 & 0.5 & 0 \\ 0.2 & 0 & 0.5 \end{bmatrix}$$

Hierarchical Clustering Example - MVN Simulated Data (4 of 5)



- The data consists of three classes and three features.
- The scatter plot shows the three classes in three colors.
- The purple, yellow and the green color represent classes I, II and III respectively.
- The classes are well separated in features 2 and 3 (row 2, column 3).

Hierarchical Clustering Example - MVN Simulated Data (5 of 5)



- This is a dendrogram plot of agglomerative hierarchical clustering done for the MVN data.
- The data set consists of 30 points (or objects) labeled on the horizontal axis.
- The links between objects are shown as inverted U-shaped lines.
- The height of the U indicates the distance between the objects.
- Note that even though the data is generated from three classes, there are four clusters at height 4.0, due to large variances in class I compared with II and III.

Hierarchical Clustering - Advantages & Limitations

- Advantages
  - The algorithm is easy to implement.
  - We do not have to specify the number of clusters.
  - The dendrogram produced is very useful in understanding the data.
- Limitations
  - The algorithm does not scale well for large data sets.

Density-based Clustering

- The DBSCAN algorithm was proposed by Martin Ester et al. in 1996 [7].
- DBSCAN is the most common example of density-based clustering algorithm. The algorithm **groups together data points that are closely packed, marking as outliers** points that are in the low-density regions.
- The algorithm requires **two parameters**:
  - **Neighborhood of a point, ε,** defined by a distance.
  - **Minimum number of points, MinPts,** required to form a cluster

Density-based Clustering, cont.

The algorithm works as follows:

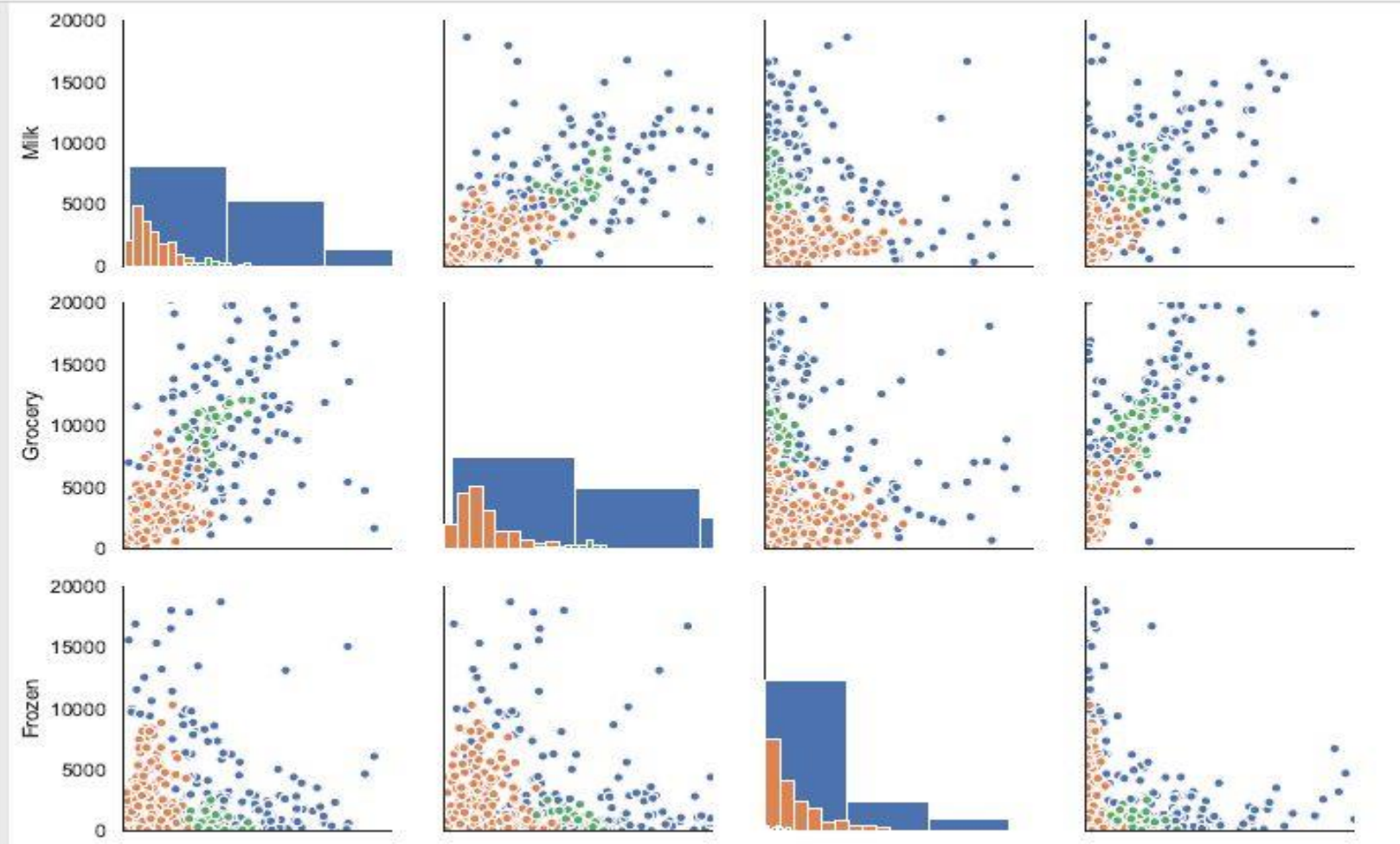
1. Start with an arbitrary data point that has not been visited.
2. Extract all points in the ε-neighborhood of this point.
3. If there are sufficient points in this neighborhood (MinPts), then the clustering process starts, and the points are marked as visited; else the point is labeled as noise (later this point can become part of the cluster).
4. If a point is found to be part of the cluster, then its neighborhood is also part of the cluster and the above procedure from step 2 is repeated.
5. After all the points in the neighborhood are exhausted, a new unvisited point is retrieved and processed.
6. The process continues until all points are marked as visited.

Density-based Clustering Example - Wholesale Customers Data

Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_ Paper	Delicatessen
2	3	12669	9656	7561	214	2674	1338
2	3	7057	9810	9568	1762	3293	1776
2	3	6353	8808	7684	2405	3516	7844
1	3	13265	1196	4221	6404	507	1788
2	3	22615	5410	7198	3915	1777	5185
2	3	9413	8259	5126	666	1795	1451
2	3	12126	3199	6975	480	3140	545
2	3	7579	4956	9426	1669	3321	2566
1	3	5963	3648	6192	425	1716	750
2	3	6006	11093	18881	1159	7425	2098
2	3	3366	5403	12974	4400	5977	1744
2	3	13146	1124	4523	1420	549	497
2	3	31714	12319	11757	287	3881	2931

- The wholesale customer data is taken from the **UCI Machine Learning Repository [10].**
- It shows the customer annual spending on various products, in monetary units.
- The data contains 8 descriptive features, and 440 samples.
- The feature **Milk** gives the annual spending on milk products.
- The feature **Channel** refers to retail vs. restaurant/hotel.
- The data set has **no labels**.

Density-based Clustering Example - Wholesale Customers Data, cont.



- The wholesale customer data is clustered using the **DBSCAN algorithm**.
- Five numerical features, namely, Milk, Grocery, Frozen, Detergents Paper and Delicatessen are used to find similar groups.
- The scatter plot shows **two clusters** in orange and green. The blue data points are the noise points.
- The two clusters are separated in the annual spending on Milk and Grocery.

DBSCAN Clustering - Advantages & Limitations

- Advantages
  - We do not have to specify the number of clusters.
  - The algorithm can find clusters with non-linear separation.
  - The algorithm finds noise points and is therefore robust to outliers.
- Limitations
  - It does not cluster very well data sets with large differences in densities.
  - As with any Euclidean distance measure, it is hard to find an appropriate value of ε in high dimension.

Evaluation of Clustering

- Assessing the quality of clustering is difficult task, as we do not know the ground truth.
- Clustering evaluation can be broadly classified into two approaches:
  - **Internal Evaluation:** When the clustering result is evaluated based on the data that was clustered itself, this is called internal evaluation.
    - These methods usually assign high scores to algorithms that produce clusters with high within cluster similarity and low intra-cluster similarity.
    - Some internal evaluation measures such as Davies-Bouldin Index and Dunn Index are given in [9].
    - Internal evaluation gives some insights into comparison between algorithms.
    - However, a drawback of internal evaluation criteria is that a high score with any of the indices does not necessarily imply an effective model selection or information retrieval

Evaluation of Clustering, cont.

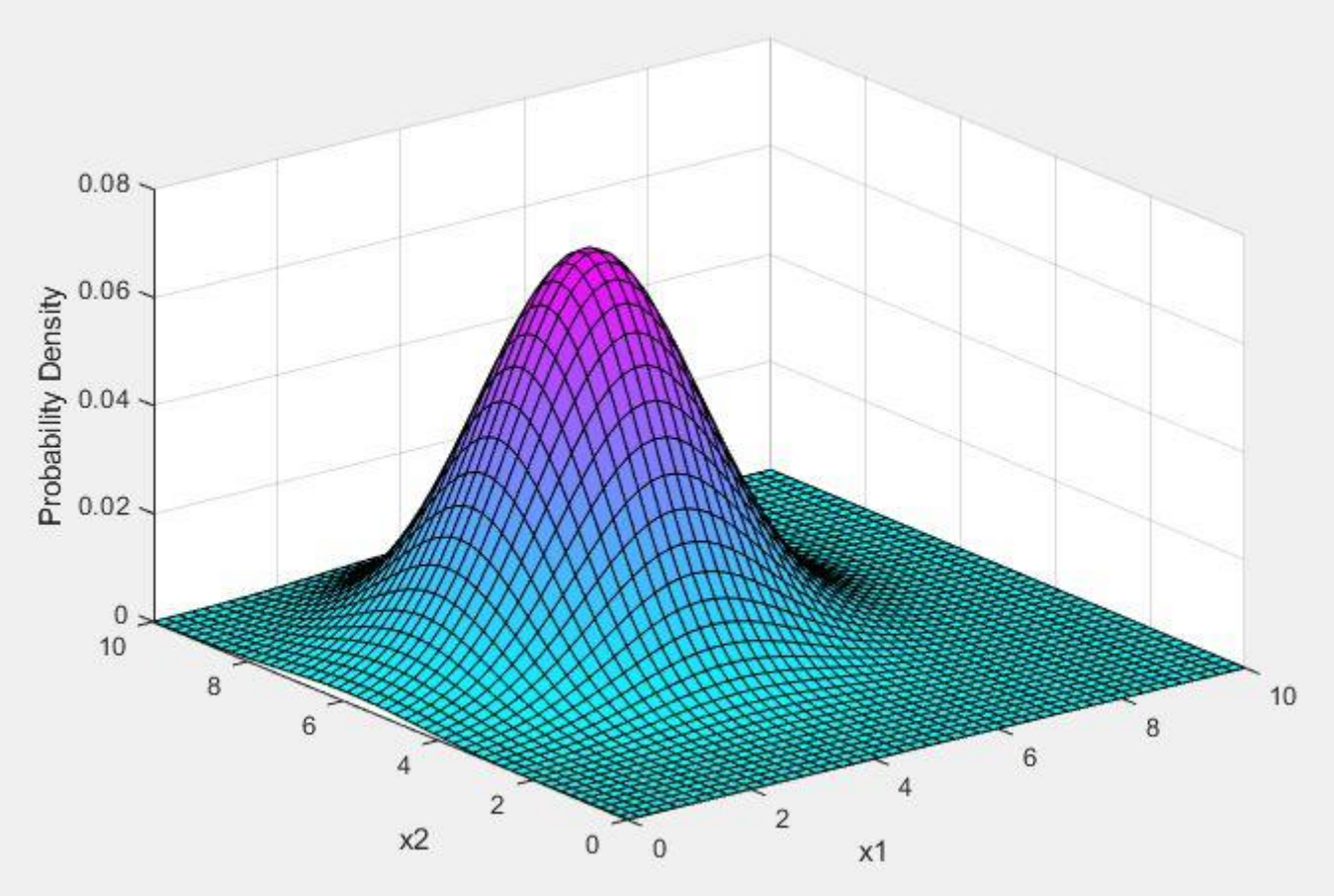
- **External Evaluation:** In external evaluation, clustering results are evaluated based on data that was not used for clustering, such as known class labels and external benchmarks. These benchmarks are usually created by experts and can be thought of as **gold standard**.
  - These evaluation techniques may not necessarily be adequate for real data.
  - From a knowledge discovery point of view, the reproduction of known knowledge may not be the intended result.
  - Some external evaluation measures such as Rand Index and F-measure are given in [9].

Appendix A: Multivariate Normal Distribution (MVN)

- The multivariate normal distribution is a generalization of the one-dimensional normal distribution to higher dimensions.
- For a p-dimensional vector $\boldsymbol{x} = \left(x_1, x_2, \cdots, x_p\right)^T$, it is denoted by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$, the **mean**, is a vector of size p and $\boldsymbol{\Sigma}$, the **covariance matrix**, is a positive definite matrix of size $p \times p$.
- Its probability density function (PDF) is given by

$$f_X(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^p |\boldsymbol{\Sigma}|}} e^{\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right)}$$

Appendix A: Multivariate Normal Distribution (MVN)



- The diagonal elements in the covariance matrix represent the variance of each variable.
- The off-diagonal elements in a covariance matrix represent the correlation between pairs of variables.
- This figure shows a plot of the PDF of a bivariate normal distribution [11].

References

1. Gareth James, et al., *An Introduction to Statistical Learning*, Springer, 2017.
2. Trevor Hastie, et al., *The Elements of Statistical Learning*, Springer, 2016.
3. EMC Education Services, *Data Science and Big Data Analytics*, Wiley, 2015.
4. *k*-means Clustering - https://en.wikipedia.org/wiki/K-means_clustering
5. Yiu-Ming Cheung, *A new generalized k-means clustering algorithm*, Pattern Recognition Letters, 2003.
6. Lior Rokach, et al., *Clustering Methods*, Data Mining and Knowledge Discovery Handbook, Springer, 2005.

References (contd)

7. Martin Ester et al., *A density-based algorithm for discovering clusters in large spatial databases with noise*, KDD, 1996.
8. DBSCAN Clustering Algorithm - https://sites.google.com/site/dataclusteringalgorithms/density-based-clustering-algorithm
9. Cluster Analysis - https://en.wikipedia.org/wiki/Cluster_analysis
10. Dua, D. and Graff, C., UCI Machine Learning Repository, Irvine, CA, University of California, School of Information and Computer Science, 2019.
11. MATLAB, 9.4.0.813654 (R2018A), Natick, MA, 2018.