# Data Mining

## Minería de Datos

Ivan Saavedra, Ph.D.

saavedrai@uninorte.edu.co

Universidad del Norte

División de Ingenierías

Dpto. Ingeniería de Sistemas

202030

# Topics

- Background
- Statistical Concepts
- Markov Chain Monte Carlo
- Parametric Classification
- Non-Parametric Classification
- **Clustering**
- Decision Trees
- Artificial Neural Networks

# Clustering

***Clustering is the process of finding meaningful groups in data.***

- The prospective electoral voters can be clustered into different groups so that candidates can tailor the message to resonate within each group.

- The difference between classification and clustering can be illustrated with an example:
  - Categorizing a given voter as a “soccer mom” (a known user group) or not, based on previously available data, is supervised **Classification**.
  - Segregating a population of electorates into different groups, based on similar demographics is unsupervised learning **Clustering**.

  - The process of identifying whether a data point belongs to a particular known group is **classification**
  - The process of dividing the dataset into meaningful groups is **clustering**.

- The most common application of clustering is to explore the data and find all the possible meaningful groups in the data.
- Clustering a company’s customer records can yield a few groups in such a way that customers within a group are more like each other than customers belonging to a different group.

# Clustering

Some of the common applications:

- **Marketing:** Finding the common groups of customers based on all past customer behaviors, and/or purchase patterns for customer segmentation, and to tailor marketing messages.
- **Document clustering:** One common text mining task is to automatically group documents into groups of similar topics to identify key topics, summarize clustered groups rather than having to read the whole document. Document clustering is used for routing customers support incidents, online content sites, forensic investigations, etc.
- **Image Segmentation:** The object of the image segmentation is to club similar pixels in the image together. One can apply clustering to create clusters having similar pixels in the same group
- **Session grouping:** In web analytics, clustering is helpful to understand common groups of clickstream patterns and to discover different kinds of clickstream profiles. One clickstream profile may be that of a customer who knows what they want and proceeds straight to checkout. Another profile may be that of a customer who has researched the products, read through customer reviews, and then makes a purchase during a later session. Clustering the web sessions by profile helps the e-commerce company provide features fitting each customer profile.

# Clustering

Clustering for preprocessing

1. **Clustering to reduce dimensionality**
   - In a n-dimensional dataset, the computational complexity is proportional to the number of dimensions or “n”.
   - With clustering, n-dimensional attributes can be converted or reduced to one categorical attribute “Cluster ID”. This reduce the complexity, although there will be some loss of information.

2. **Clustering for object reduction**
   - Assume that the number of customers for an organization is in the millions and the number of cluster groups is 100.
   - For each of these 100 clusters, one “poster child” customer can be identified that represents the typical characteristics of all customers in that cluster.
   - The poster child customer can be an actual customer or a fictional customer
   - The prototype of a cluster is the most common representation of all the customers in a group
   - Reducing millions of customers records to 100 prototype records provides an obvious benefit
   - This greatly reduces the record count, and the dataset can be made appropriate for classification or regression where computation complexity is important (i.e. KNN).

# Clustering

A cluster can be:

- **Exclusive or strict partitioning clusters**
  - Each data object belongs to one exclusive cluster
- **Overlapping clusters**
  - The cluster groups are not exclusive, and each data object may belong to more than one cluster.
  - Also known as multi-view clusters.
  - For example, a customer of a company can be grouped in a high-profit customer cluster and a high-volume customer cluster at the same time.
- **Hierarchical clusters**
  - Each child cluster can be merged to form a parent cluster.
  - For example, the most profitable customer cluster can be further divided into a long-term customer cluster and a cluster with new customers with high-value purchases.
- **Fuzzy or probabilistic clusters**
  - Each data point belongs to all cluster groups with varying degrees of membership from 0 to 1.
  - For example, in a dataset with clusters A, B, C, and D, a data point can be associated with all the clusters with degree A=0.5, B=0.1, C=0.4, and D=0.
  - It associates a probability of membership to all the clusters.

# Clustering

Types of Clustering Techniques

- **Prototype-based clustering**
  - Each cluster is represented by a central data object, also called a prototype.
  - The prototype of each cluster is usually the center of the cluster (centroid clustering or center-based).

- **Density clustering**
  - A cluster is defined as a dense region where data objects are concentrated surrounded by a low-density area where objects are sparse.
  - Each dense area can be assigned a cluster and the low-density area can be discarded as noise.

- **Hierarchical clustering**
  - A cluster hierarchy is created based on the distance between data points.
  - The output is a dendrogram: a tree diagram that shows different clusters at any point of precision which is specified by the user.
  - Two approaches: bottom-up and top-down approach.
  - It is useful when the data size is limited.

- **Model-based clustering**
  - Also called distribution-based clustering.
  - A cluster can be thought of as a grouping that has the data points belonging to the same probability distribution.
  - Hence, each cluster can be represented by a distribution model where the parameter of the distribution can iteratively optimize between the cluster data and the model

# k-MEANS Clustering

- k-Means clustering is a prototype-based clustering method where the dataset is divided into k-clusters.
- It is also referred as the Lloyd-Forgy algorithm or Lloyd’s algorithm.
- Is one of the most commonly used clustering algorithms.
- K-Means clustering creates k partitions in n-dimensional space.
- The number of clusters (k) is specified by the user.
- The objective is to find a prototype data point for each cluster.
- All the data points are then assigned to the nearest prototype, which then forms a cluster
- To partition the dataset, a proximity measure must be defined. The most common is the Euclidean distance.
- The prototype is called as the centroid
- The center of the cluster can be the mean of all data objects in the cluster, as in k-means, or the most represented data object
- The centroid does not have to be a real point in the dataset

# k-MEANS Clustering

How it works ?

**Step1:** Initiate Centroids

- Initiate k random centroids

**Step2:** Assign Data Points

- Assign to the "nearest" centroid using a proximity measure

**Step3:** Calculate New Centroids

- $SSE = \sum_{i=1}^{k} \sum_{x_j \in C_i} \left\| x_j - \mu_i \right\|^2$

Where $C_i$ is the cluster $i^{th}$, j are the data points in a given cluster, $\mu_i$ is the centroid for the $i^{th}$ cluster, and $x_j$ is a specific data object.

The centroid with minimal SSE for the given cluster i is the new mean of the cluster.

The mean of the cluster can be calculated as:

$$\mu_i = \frac{1}{j_i} \sum_{x \in C_i} X$$

Where X is the data object vector ($x_1$, $x_2$, …, $x_n$)

**Step4:** Repeat assignment and Calculate New Centroids

**Step5:** Termination

# k-MEANS Clustering

Some key issues to be considered

- **Initiation**
  - The final clustering grouping depends on the random initiator and the nature of the dataset.

- **Empty clusters**
  - One possibility is the formation of empty clusters in which no data objects are associated.
  - This will introduce a centroid with high SSE

- **Outliers**
  - It is susceptible to outliers
  - They drift the centroid away from the representative data point
  - Hence, the prototype is no longer the best representative of the cluster it represents
  - An application with outliers is to identify fraudulent transactions

- **Post-processing**
  - There are a few post-processing techniques to force a new solution that has less SSE
  - One could always increase the number of clusters, but this could start overfitting the dataset
  - Approaches: bisecting the cluster that has the highest SSE and merging two clusters into one even if SSE increases slightly

# k-MEANS Clustering

Evaluation of Clusters

- It is different from regression and classification algorithms because there are no known external labels for comparison
- The evaluation parameter has to be developed from the very dataset

- A good evaluation measure is the total **SSE**
  - Good models will have low SSE within the cluster and low overall SSE among all clusters
- SSE can also be referred to as the average within-cluster distance and can be calculated for each cluster and then averaged for all the clusters

- Another used evaluation measure is the **Davis-Bouldin index**
- The Davis-Bouldin index is a measure of uniqueness of the clusters and takes into consideration both cohesiveness of the cluster (distance between the data points and center of the cluster) and separation between the clusters.
- It is the function of the ratio of within cluster separation to the separation between clusters
- The lower the value of the index, the better the clustering

- Both SSE and the Davies-Bouldin index have the limitation of not guaranteeing better clustering when they have lower scores.

# k-MEANS Clustering

How to choose the right number of clusters ?

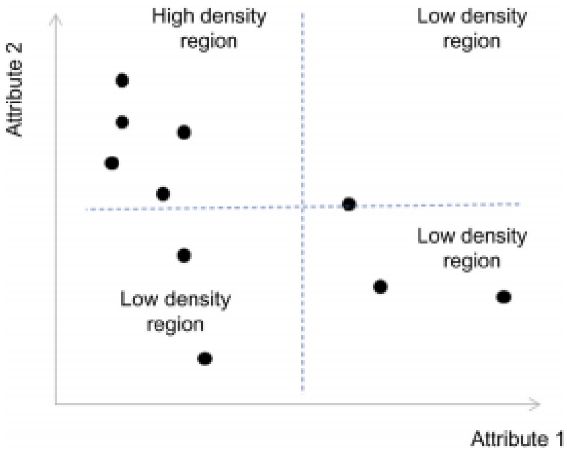
Don’t worry, we can use the elbow curve !

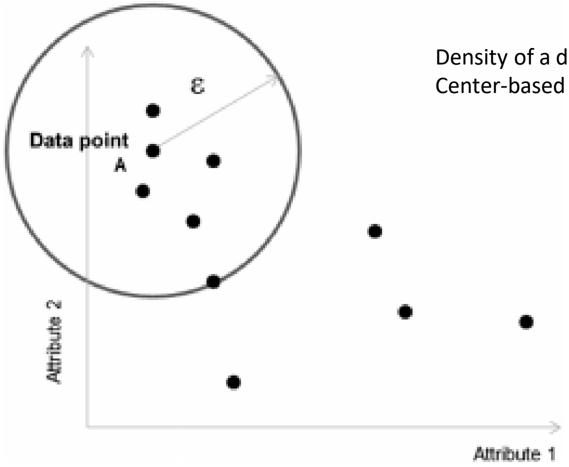
The x-axis will represent the number of clusters and the y-axis will be an evaluation metric

# DBSCAN Clustering

- A cluster is defined as an area of high concentration (or density) of data objects surrounded by areas of low concentration of data objects
- A density-clustering algorithm identifies clusters in the data based on the measurement of the density distribution in n-dimensional space
- Specifying the number of the cluster parameters (k) is not necessary
- It serves as an important data exploration technique

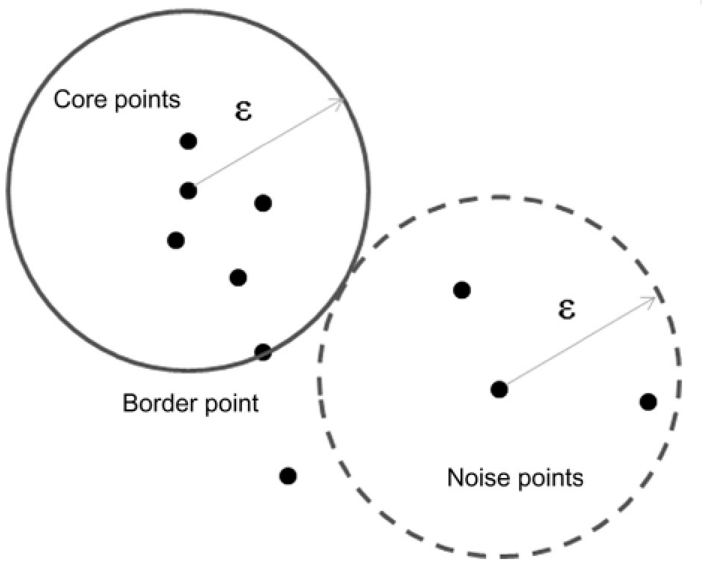
- Density can be defined as the number of data points in a unit n-dimensional space.
- The number of dimensions n is the number of attributes in a dataset
- Consider a 2-dimensional space or a dataset with two numeric attributes





Density of a data point within radius ε.
Center-based density

# DBSCAN Clustering



How it works ?

**Step 1: Defining Epsilon and MinPoints**

- Starts with the calculation of a density for all data points in a dataset, with a fixed radius ε
- A threshold of data points (MinPoints) is defined to determine whether a neighborhood is high-density or low-density

**Step 2: Classification of Data Points**

- Data points can be defined into three buckets:
  - **Core points**: All data points inside the high-density region of at least one data point are considered a core point. A high-density region is a space where there are at least MinPoints data points within a radius of ε for any data point
  - **Border points**: They sit on the circumference of radius ε from a data point. A border point is boundary between high-density and low –density space. Border points are counted within the high-density space calculation.
  - **Noise points**: Any point that is neither a core point nor border point is called a noise point. They form a low-density region around the high-density region.

**Step 3: Clustering**

- Once all data points are classified into density points, we can perform the clustering.
- Groups of core points from distinct clusters
- If two or more core points are within ε of each other, then both core points are in the same cluster
- All noise points form low-density regions and are not classified in any cluster.
- Since DBSCAN is a partial clustering algorithm, a few data points are left unlabeled or associated to a default noise cluster

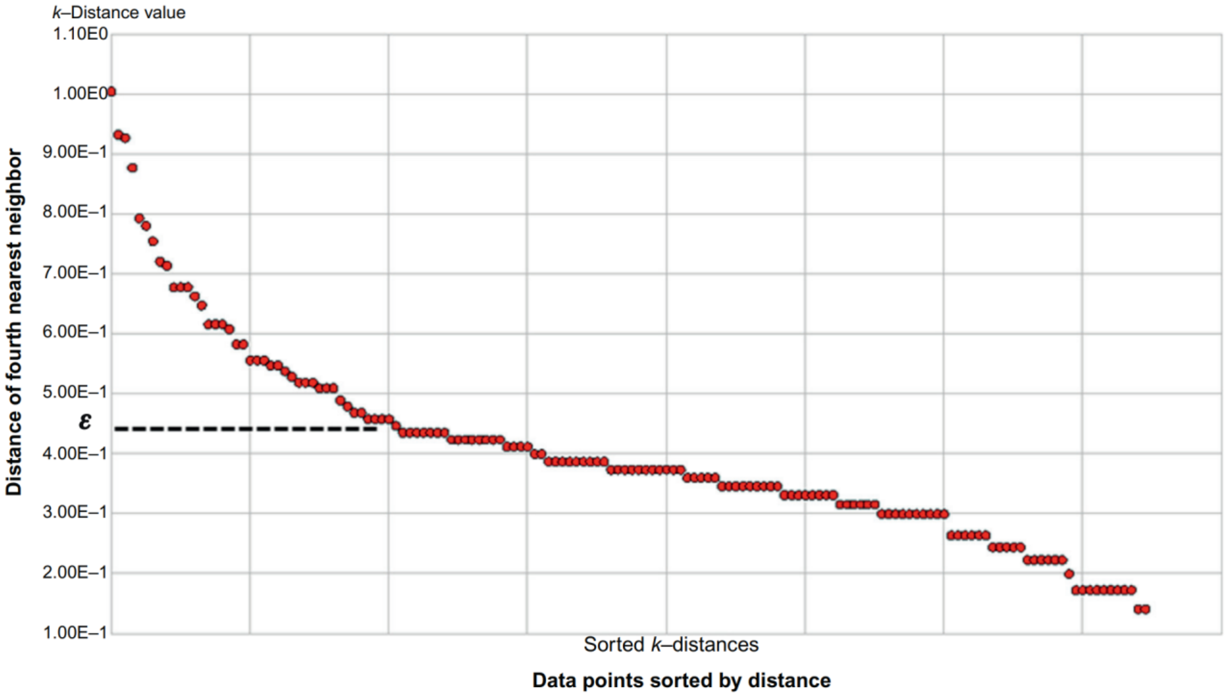
# DBSCAN Clustering

How it works ?

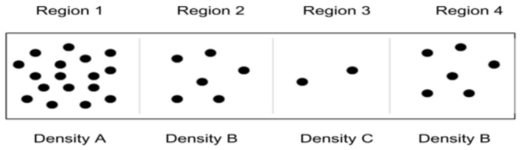
# DBSCAN Clustering

Optimizing Parameters

- There is no need for specifying the number of clusters (k)
- Clusters are automatically found in the dataset
- **Selecting** the distance parameter **ε** and the **MinPoints** can be an issue
- One technique to find an optimal ε relates to k-NN
- It can be estimated by building a **k-distribution graph**
  - Calculated for all data points
  - Arrange all distances values in descending/ascending order
  - Points on the right-hand/left-hand side of the chart will belong to data points inside a cluster (distance is smaller)
  - The distance at which the chart rises will be the optimal valu ε and the value of k is used for MinPoints



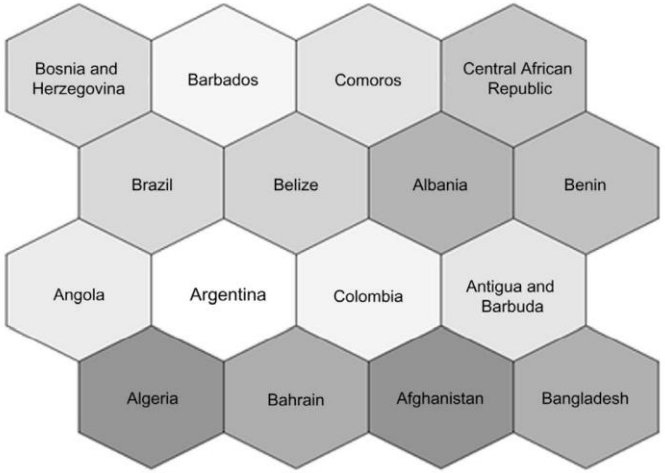
Special Cases: Varying Densities

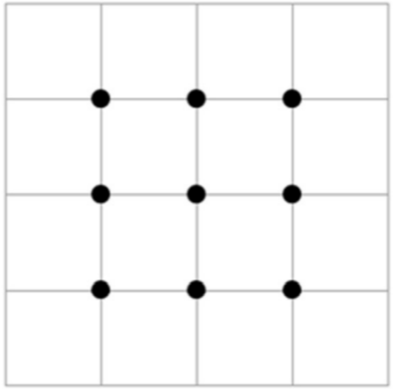


# SELF-ORGINIZING MAPS

SOM

- Is a powerful visual clustering technique that evolved from a combination of NN and prototype-based clustering
- The output is an organized visual matrix, usually a 2d grid
- The objective is to transfer all input data objects with n attributes to the output lattice in such a way that objects next to each other are closely related to each other
- SOM is used as a visual clustering and data exploration technique
- It is also known as Kohonen networks
- Each data point occupies a cell or a node in the output lattice
- Each cell in the SOM grid corresponds to one or a group of data points
- The SOM compares relative gross domestic product (GDP) data from different countries where countries with similar GDP profiles are placed either in the same cells or next to each other
- All similar countries around a particular cell can be considered a grouping
- The individual data objects (countries) do not have a cluster membership





# SELF-ORGINIZING MAPS

How it works ?

**Step 1: Topology Specification**

- Multi-dimensional output is possible
- 2D (rectangular or hexagonal) are commonly used
- The hexagonal can have six neighbors, two more than rectangular
- The number of centroids is the product of the number of rows and columns

**Step 2: Initialize Centroids**

- The initial centroids are values of random data objects from the dataset
- Similar to initialize centroids in k-Means

**Step 3: Assignment of Data Objects**

- Data objects are selected one by one and assigned to the nearest centroid
- It can be calculated using a distance function like Euclidean

**Step 4: Centroid Update**

- Update the closes centroid
- Update all centroids in the grid space neighborhood

**Step 5: Termination**

- It is continued until no significant centroid updates take place in each run or until the specified number of run count is reached

**Step 6: Mapping a New Data Object**

- Any new data object can be quickly given a location on the grid space, based on its proximity to the centroids

# Bibliography

- **Book:** George C. Montgomery and George C. Runger. Applied Statistics and Probability for Engineers.
- **Book:** Vijay Kotu and Bala Deshpande. (2019). Data Science, Concepts and Practice. (Second Edition). Morgan Kaufmann.
- **Book:** Vijay Kotu and Bala Deshpande. Data Science. Concepts and Practice. Second Edition. 2019.
- **Book:** Sebastian Raschka. Python Machine Learning. Packt Publishing. 2015.
- **scikit-learn**. Scikit-learn: Machine Learning in Python, Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. Journal of Machine Learning Research, volume 12, 2011