✅

Part 3 - Unsupervised Learning

- Pages 177 - 218.

Introduction

Descriptive learning tasks, also known as unsupervised learning, refer to **identifying relevant information in data without an external element guiding the learning process**.

> Unsupervised Learning is data-driven, it does not require prior knowledge about its classes or categories.

- These techniques are mainly **used when the goal is to find patterns or trends that help in understanding the data**.
- Given a dataset **X**, an unsupervised machine learning algorithm learns to represent the submitted inputs based on a certain quality criterion.
- Tasks related to unsupervised learning include **summarization**, **association**, and **clustering**.

Main Tasks

Summarization

Summarization **finds a simple and compact description of the data** using statistical measures, advanced visualization techniques, and functional relationship identification between attributes.

Association

Association involves **searching for frequent patterns** of relationships among attributes in a dataset.

Clustering

Clustering **identifies groups in the data based on the similarity** between objects.

- It is appropriate for exploring and verifying structures present in a dataset.

Association

Example - Frequent Pattern Mining

> **Association rules** take the form of **if antecedent then consequent** rules, where both the antecedent and consequent are itemsets.

- One of the objectives of this area in machine learning is to understand which items are frequently purchased together.

Example - Market Basket Analysis (describing customer purchase behavior)

Goal: discover groups of products that are often bought together and, based on these groups, infer which products are likely to be purchased when certain other products have already been bought.

- **A = {a1, ..., am}** is the set of products.
- **Subset I ⊆ A** is a set of items (itemset)
  - Any group of products that can be purchased together.
- **T = {t1, ..., tn}** is a set of **n** transactions, referred to as the transaction database:
  - Each transaction is a pair $<tid_i, itens_i>$. Where:
    - tid_i = transaction ID.

        - $k - item_i \subseteq$ **A** is a set of **k** items (e.g., a set of web pages visited by a user).
    - A transaction **t** $\in$ **I** **supports the itemset** **I** if and only if **I** $\subseteq$ **t**. In other words, transaction **t** contains all the elements of itemset **I**.
        - For example, a transaction where cheese, bread, and butter were purchased **supports** the itemset **{bread, cheese}**. Intuitively, **"support" means strengthening or providing evidence for the itemset.**
        - The set K(I) of transactions that support itemset I is called the support of the itemset.

Apriori Algorithm

The first algorithm developed for mining itemsets and association rules.

- Uses a **breadth-first** (busca em largura) search strategy with a generate-and-test approach.
- **At each level, possible itemsets are generated based on frequent itemsets found in the previous level. After generation, their frequency is tested by scanning the transaction database again.**
- It involves multiple database scans to calculate the support of candidate frequent itemsets.

Apriori Alternatives

Algorithms that **reduce database scans** by generating collections of candidate itemsets using a **depth-first search** (busca em profundidade) strategy. Examples include:

- **Eclat.**
- **FP-growth**
    - Uses a prefix tree to store itemsets, avoiding the combinations required by Apriori for candidate generation.

Association Rules

It is useful to **determine which combinations of terms can be discovered and how interesting those combinations are.**

- Term combinations are represented by rules in the form **A → B**, where **A ∪ B** is a frequent itemset.
- The **level of interest** is represented by the **confidence** of the rules, which measures **the probability of one set of terms occurring given that another set has already occurred.**

$$\text{confidence}(A \rightarrow B) = \frac{P(A \cup B)}{P(A)} = \frac{\text{support}(A \cup B)}{\text{support}(A)}$$

Itemset Summarization

The approach used by an association rule discovery algorithm may result in the generation of a large number of rules, many of which may not be interesting. For instance, **the entire set of frequent itemsets can often be represented more compactly by eliminating all itemsets that are subsets of other frequent itemsets.**

The **monotonicity property of support** suggests a summarized representation of the set of frequent itemsets:

- **Maximal frequent itemsets:** an itemset is *maximal* if it is frequent but none of its proper supersets are frequent.
- **Closed frequent itemsets:** a frequent itemset is *closed* if and only if it does not have any supersets with the same support.

Heuristics for Association Rule Selection

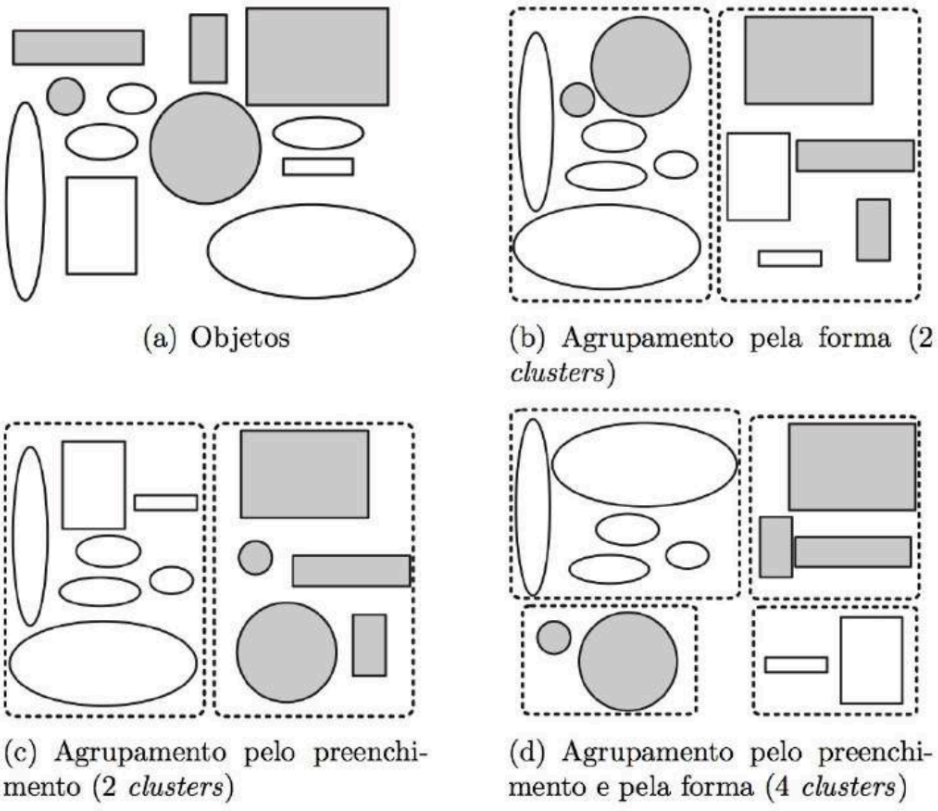
Association rule algorithms tend to generate an excessive number of rules. In recent years, various measures have been proposed to extract **interesting patterns** from large datasets. The idea is to **select a subset of patterns or rules that are somehow more relevant.**

- **Piatetsky-Shapiro Principles (1991)**

- **Interest Coefficient or Lift**
  - Based on the statistical notion of independence between two random variables.
  - Measures the strength of the association.
- **Conviction**
  - Measures how convincing a rule is.

Clustering

Goal: find a **cluster structure** in the data where the **objects belonging to each cluster share some characteristic** or property relevant to the problem domain.



Example of how we can cluster a dataset in different ways.

Cluster

> A cluster is **a collection of objects that are close to each other or satisfy some spatial relationship.** Other definitions of a cluster include:

- **Well-separated cluster**: any point in a given cluster is closer to (or more similar to) every other point in that cluster than to any point outside it.
- **Center-based cluster**: any point in a given cluster is closer to (or more similar to) the center of that cluster than to the center of any other cluster.
- **Contiguous or chained cluster**: any point in a given cluster is closer to (or more similar to) one or more points in that same cluster than to any point outside it.
- **Density-based cluster**: a cluster is a dense region of points, separated from other high-density regions by areas of low density.
- **Similarity-based cluster**: a cluster is a set of points that are similar to each other, while points in different clusters are not similar.

Each of these definitions leads to a **clustering criterion**, which is **the basis for selecting a cluster structure that best fits a given dataset**.

Clustering Criteria

Compactness: compactness or homogeneity of a cluster is generally **associated with low intra-cluster variation**. Algorithms that optimize this criterion tend to be effective at discovering spherical and well-separated clusters, but may fail on more complex structures.

- Example: K-Means.
  - It more easily identifies spherical/globular clusters, but fails in ring-shaped structures.

Chaining or linkage: chaining is based on the idea that **neighboring objects should belong to the same cluster**. This criterion is well-suited for detecting

arbitrarily shaped clusters, but is not robust when there is little spatial separation between clusters.

- Example: hierarchical clustering with average linkage.

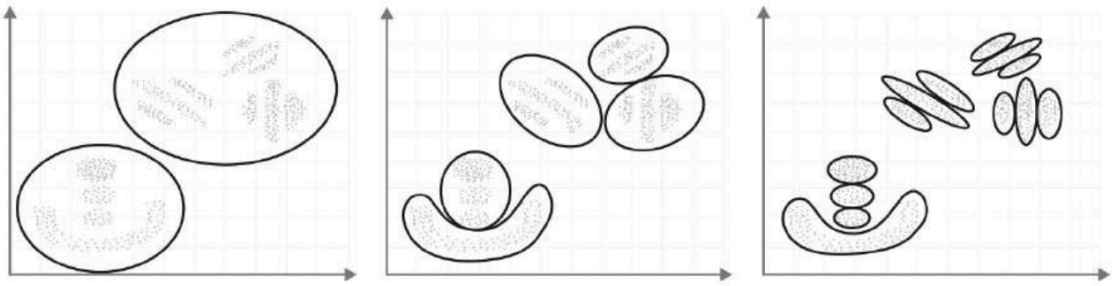
Spatial separation: considers **the distances between clusters** but, by itself, provides little guidance during clustering and can lead to trivial solutions. It is often used in combination with another criteria.

Validation

This stage helps in **selecting the most appropriate algorithm and in tuning its parameters** (e.g., finding the optimal number of clusters). It's important to note that most evaluation techniques are biased toward a specific clustering criterion. Therefore, **multiple validation measures** should be used to select the most consistent results across a variety of algorithms and parameter settings.

Another important aspect is that most algorithms aim to find a **homogeneous structure**, meaning that all clusters follow the same clustering criterion. However, real-world data may exhibit a **heterogeneous structure**, where each cluster aligns with a different clustering criterion.

- It is also possible that **more than one relevant structure** is present in the data. That is, the same dataset may contain multiple structures, each compatible with a different clustering criterion or even heterogeneous in nature.



(a) Estrutura E1 (b) Estrutura E2 (c) Estrutura E3

Clustering Steps

In cluster analysis, there is no prior information about possible data classifications. Therefore, many of the feature selection or extraction techniques described in this chapter do not apply or must be adapted to be used in clustering.

>> Data Preparation

Covers aspects related to preprocessing and the appropriate representation format for use by a clustering algorithm.

- Normalizations, type conversions, reduction of the number of attributes through selection or feature extraction.

>> Proximity

Definition of proximity measures appropriate to the application domain and the type of information to be extracted from the data.

Examples:

- Proximity between objects.
- Proximity between an object and a group of objects.
- Proximity between two groups of objects.

All clustering algorithms consider similarity/dissimilarity between objects. The same algorithm can be implemented using different measures. Similarities between objects and groups, and between two groups, are part of the characterization of each specific algorithm and are generally used to decide the assignment of an object to a cluster or to merge/split clusters, depending on the type of cluster the algorithm aims to identify.

Similarity or dissimilarity measures generally assume that all attributes are equally important, meaning they all contribute to the measurement calculation. Distance measures are directly affected by attribute scales. To minimize this effect, attributes are typically normalized.

- Common dissimilarity measure: Euclidean distance.
- Common similarity measure: correlation.

For distance measures, the smaller the value, the more similar the objects are. All distance measures satisfy the following properties:

1. **d(xi, xi) = 0** for any xi (objects are not different from themselves).
2. **d(xi, xj) = d(xj, xi)** (symmetry).
3. **d(xi, xj) ≥ 0 for all xi and xj** (non-negativity).

Some of these measures, called **metrics**, also satisfy the following properties:

4. **d(xi, xj) = 0** only **if xi = xj**.
5. **d(xi, xl) ≤ d(xi, xj) + d(xj, xl)** for all xi, xj, xl (triangle inequality).

Measures for Quantitative Attributes

- The most used measures for continuous and ratio-scale attributes are distance metrics based on the Minkowski metric, such as **Euclidean distance** (most common), **Manhattan distance**, and **Supremum distance**.
  - When all attributes are binary, Manhattan distance is commonly used, which in this context is known as **Hamming distance**.
- For binary and nominal data, there are various **matching coefficients**, such as the **simple matching coefficient** and **Jaccard coefficient**.
- **Assessing Similarity**
  - Two common ways to evaluate similarity between pairs of objects in clustering are given by the absolute value of **angular separation (cosine)** and **Pearson correlation** (widely used in bioinformatics), which quantify the correlation between objects xi and xj.

Measures for Quantitative Attributes

- These are obtained by summing the individual contributions of all attributes.

- For nominal attributes, a commonly used distance measure is **Hamming distance**.

Measures for Heterogeneous Attributes

- Many datasets used in ML have attributes of different types, both quantitative and qualitative. Measures proposed to assess similarity between objects include:
  - **General similarity coefficient**.

>> Clustering

One or more clustering algorithms are applied to the data to identify possible cluster structures present in the dataset.

>> Validation

This step evaluates the result of clustering and determine whether the clusters are meaningful — that is, whether the solution is representative of the dataset being analyzed. In addition to verifying the solution's validity, it can also help determine the appropriate number of clusters, which is usually not known in advance.

- The definition of proximity measures and clustering criteria used by the algorithms generally depends on certain assumptions about the cluster shapes or the configuration of multiple clusters.
- Another important aspect is that data is rarely "ideally" structured — for example, they often do not form hyperspherical, hyperellipsoidal, or linear configurations — so each clustering algorithm may perform better than others for a specific data structure in the attribute space.

If the goal is to compare algorithms based on different types of criteria, it's essential to know exactly what you want to compare and assess whether such a comparison makes sense, since the objectives of the algorithms may differ. Characteristics for comparing algorithms in the same context include:

- **Algorithm-related**
  - Algorithm complexity.
  - Scalability and efficiency for large datasets.
  - Similarity measures supported by the algorithm.
  - Robustness to noise and outliers.
  - Ability to handle high-dimensional data or to find clusters in subspaces of the original space.
  - Stability (whether different runs yield the same cluster assignments).
  - Ability to incrementally handle the addition of new objects or removal of old objects.

- **Result-related**
  - Types of cluster shapes the algorithm can detect.
  - Interpretability of the results.

- **Data-related**
  - Data types the algorithm supports (continuous, categorical, binary).
  - Sensitivity to the data order.

- **User interaction-related**
  - Whether the algorithm automatically determines the number of clusters or if the user must specify it.
  - Parameters required by the algorithm and the domain knowledge needed by the user.

Considerations

- Clustering algorithms are categorized more by their **models** than by their clustering criteria.

- Researchers should try to mathematically define the models and criteria behind the clustering algorithms they propose, facilitating future investigations and comparisons.
- Clustering validation indexes are direct mathematical formulations of the induction principles underlying clustering criteria. Comparing algorithms using these indexes can provide insights into the contexts where one algorithm performs better than another, but it doesn't necessarily mean that one algorithm produces more valid results than another. Two algorithms applied to a dataset with no real structure will both produce invalid results.
- An algorithm designed for a particular model universe is not suitable for datasets with structures that belong to a radically different family of models. For example, **K-means cannot find non-convex clusters**.

>> Interpretation

The process of examining each cluster with respect to its objects to label them, describing the nature of the cluster.

Cluster Algorithms

Hierarchical

Generates, from a proximity matrix, a sequence of nested partitions. It can be divided into two approaches:

- **Agglomerative**: starts with *n* clusters, each containing a single object, and forms the sequence of partitions by successively merging clusters.
    - Produces a sequence of partitions from *n* objects into *k* clusters, where level 1 has *n* single-object clusters and level *n* has one cluster with all objects. Thus, data is grouped such that if two objects are grouped at a certain level, they remain in the same group at higher levels, building a hierarchy of clusters.
- **Divisive**: starts with one cluster containing all objects and forms the sequence by successively splitting clusters.

Advantages

- Flexibility in terms of granularity level.

- Easy to apply any form of similarity or distance measure.
- Possibility of using any type of attribute.

Disadvantages

- Vague termination criterion.
- Once a cluster is created in the clustering process, it remains until the end, with no reassignment of its objects.

Examples of Hierarchical Algorithms

- BIRCH.
- CURE.
- CHAMELEON.
- OPTICS.
- ROCK.

Squared Error-Based

Optimize the clustering criterion using an iterative technique.

- The first step is to create an initial partition. Then, objects are moved from one cluster to another to improve the clustering criterion. These algorithms are computationally efficient but may converge to an optimum local.
- The clustering criterion used by these partitional algorithms is the **squared error**, which ensures the compactness property of the clusters.
- **Goal**: obtain a partition that minimizes the squared error for a fixed number of clusters. Minimizing the squared error, or within-cluster variance, is equivalent to maximizing between-cluster variance.
- The squared error for a clustering containing *k* clusters is the **sum of the within-cluster variations.**

Examples of SEB Algorithms

- **K-means**:
  - Partitions the dataset into *k* clusters, where *k* is provided by the user.
  - Clusters are formed based on a similarity measure.
  - Minimizes the distance between each object and the centroid of the cluster it belongs to.
  - **Complexity:** O(n).
- PAM.
- CLARA.
- CLARANS.

Density-Based

Assume that clusters are regions with high object density, separated by regions of low density in the object space.

Examples of Density Algorithms

- DENCLUE (most commonly used).
- DBSCAN.
- WaveCluster.

Graph-Based

In clustering using graph-based techniques, data is represented in a **proximity graph**. In the simplest case, each node represents an object and is connected

to the other *n - 1* nodes, resulting in a complete graph.

Examples of Graph-Based Algorithms

- HSC.
- CLICK.

The difference between these two algorithms lies in the similarity graph they construct, the stopping criterion, and the post-processing of the kernels. The **CLICK algorithm** is more recent and currently more widely used.

Neural Network-Based

Examples of Neural Network Algorithms

- SOM.
  - SOM is an unsupervised artificial neural network, often used for clustering and data visualization tasks. It is the most traditional algorithm in this category.
- GCS.
- SOTA.
- HGSOT.
- DGSOT.

Grid-Based

This group of algorithms defines a **grid structure** for the data space and performs all operations within this grid. In general terms, this approach is highly

efficient for large datasets, is capable of identifying clusters with arbitrary shapes, and handles outliers well.

Examples of Grid-Based Algorithms

- CLIQUE.
- MAFIA.
- OptiGrid.
- STING.

Some of these algorithms were designed with a specific functionality in mind. **CLIQUE** and **MAFIA** are techniques specifically developed to handle high-dimensional data.