1. INTRODUCTION TO UNSUPERVISED CLUSTERING LEARNING TECHNIQUES AND ASSOCIATION RULES

The large volumes of data available provide us with an opportunity to discover previously unknown patterns, relationships, or associations hidden in this information. The supervised learning techniques (regression and classification) we saw in previous modules allow us to automatically identify predictive models based on the variables available to us. However, this type of model always assumes that the output variable we want to predict based on the other variables must be specified a priori. In contrast, in this module we present unsupervised learning, which allows us to discover patterns or associations in the data without previously specifying an output variable. In other words, no variable establishes the value to be predicted. We simply let the data speak for itself to discover new relationships between variables or new groups into which the data can be organized.

For instance, the melanoma dataset we are studying in this MOOC contains many variables derived from clinical analysis omics data containing information about the samples and patients. Unsupervised analysis allows us to identify new subgroups of patients from among these, according to the clinical and omics variables available. Researchers hope that these subgroups may have clinical relevance, paving the way for more personalized and effective treatments. The authors of the original melanoma study identified three new groups of patients based on their gene expression and found that these groups had significantly different survival times. In other words, they established a new way of grouping melanoma patients and tested the usefulness of these groups for predicting patient prognosis and survival.
Unsupervised analysis also makes it possible to identify what relationships exist between study variables. For instance, it allows us to study whether there is a connection between gene expression levels, the presence of mutations, and methylation levels and what, if any, this connection is. This is of interest to better understand the molecular basis of a disease and to identify previously unnoticed connections that may lead to new hypotheses for improving disease prevention or treatment. Two of the unsupervised learning techniques most widely used to extract interesting knowledge from datasets containing enormous amounts of information are clustering and association rules. In the following sections, we will briefly introduce the basic concepts of each of these techniques.

2. CLUSTERING

As introduced in Capsule 3 of Module 3 (Unsupervised learning), clustering is a set of unsupervised learning techniques designed to identify groups within the data. A group (or cluster) is a set of instances that resemble each other. Thus, the objective of a clustering algorithm is to group the available instances in such a way that instances within the same group are like each other and different from instances in other groups. It is important to note here that the term ‘similarity’ therefore becomes especially important in defining clusters. In computing, similarity between instances is defined by the following function:

$$f(a, b) \rightarrow R \quad (Equation \ 1)$$

where $a$ and $b$ are instances in our data set, and the function quantifies the distance between these instances. The values of the variables in each instance are used to quantify this magnitude.

Many functions are available to measure similarity (or the distance between instances) however, the use of one or another depends on the type of variables and the problem. We will review some of these measures in section 2.1. In addition to the proposed definition of similarity/distance, a multitude of different clustering algorithms can provide us with different clusters. In Capsule 2 of this module we will review the methodological principles of the most popular of these algorithms.

First, here we present the following example to visually illustrate what clustering is. Table 1 represents a set of patients (instances) and their values for several clinical variables (columns).
Table 1. Data set containing clinical variables obtained from a group of patients in a hospital.

We can consider different options for grouping patients. For instance, using only the ‘gender’ variable we could trivially identify two groups of patients: males and females. Using the ‘age’ and ‘glycemia’ variables we can group the data into other clusters, although it is not so easy to identify these groups. Thus, data visualizations are extremely useful in this task. For example, figure 1 shows a scatterplot of the observations based on some of the variables in our table: age (X-axis), blood glucose (Y-axis), and stress level (color scale).

One of the possible results of a clustering analysis on these data is shown in figure 2.
The brown group represents patients characterized by their young age, low–mid blood glucose, and low–mid stress levels. In turn, the light blue group represents patients characterized by their medium age, high blood glucose, and high stress levels. Finally, the dark blue group represents patients with a medium–high age, low blood glucose, and low stress levels. In this case, it seems that the way the objects are distributed in space invites us to intuitively group them in this way. This is what clustering is all about: letting the data reveal these structures and subsequently identifying the clusters with appropriate scientific techniques.

It is important to note that the outcome of clustering will depend on many factors: the nature of the data, variables considered, metric with which we quantify the similarity between instances, and the clustering algorithm itself. While the groupings proposed here can be seen immediately when adequately represented, as the number of variables and patients increases, the task of identifying groups or clusters in the data will become an increasingly difficult task to solve manually.

2.1. DISTANCE MEASUREMENTS

Numerous functions are available to measure the similarity or distance between instances. The use of one or another depends on the type of variables and the problem at hand. Some of the best known are the Euclidean distance (for numerical variables), the Levenshtein distance (for text-type variables), and the Jaccard coefficient (for integer variables).

Before calculating a distance measure between instances, it is important to consider whether the variables defining them are measured on different scales. This is true for the data in Table 1. For example, a difference of 2 units in the stress variable is maximal,
while the same difference in the blood glucose variable is negligible. If we calculate the
distance between the instances without considering these different scales, the
difference between the blood glucose values will dominate over the difference between
the stress values, with the latter becoming practically irrelevant in the calculation.

Thus, to adequately calculate the distance between instances, the variables must first
be normalized on the same scale (e.g., a range of 0–1 expressed as “[0,1]”), so that they
are all given the same weight in the calculation. Different methods can be used to
normalize the values of a variable, based on a range of simple conversion methods (from
moving the value of the variable proportionally from the range \{min, max\} to \{0,1\} to
more sophisticated methods). We will explain some of these methods in Capsule 2 of
this module.

2.2 CLUSTERING ALGORITHMS

According to the methodology used, we can classify the most popular clustering
algorithms into the following categories:

- Hierarchical clustering algorithms.
- Algorithms that perform clustering via partitions: K-means.
- Other algorithms: e.g., spectral, probabilistic, or density based, etc.

In the following sections we will briefly describe the two main strategies used for
clustering: hierarchical and partitioning clustering.

2.2.1 HIERARCHICAL CLUSTERING

Hierarchical clustering algorithms are the most intuitive and easiest to understand. The
objective of these algorithms is to construct a dendrogram: a tree-like structure that
indicates how to group similar instances to form increasingly larger clusters. Thus, a
dendrogram represents a hierarchy of clusters. The dendrogram concept originates in
the phylogenetic trees used to represent the evolutionary relationship between species
(figure 3).
In this representation, the base of the phylogenetic tree represents a single cluster which groups all species. This group then divides into subgroups as the tree is traversed upwards and its branches spread out. Similarly, a dendrogram is usually used to represent clustering outcomes, as illustrated in figure 2. It should be noted that the length of each branch is proportional to the distance between the two clusters connected to that branch. Thus, the longer the branch length, the greater the difference between the clusters. For example, in the following dendrogram (figure 4), which is associated with the data shown in table 1, because of their high similarity, instances 10 and 1 are directly grouped together with instance 6, thereby constituting the branch colored in red. The green branch represents another cluster, which can be broken down into two subclusters: instances 4, 3, and 9 versus instances 8, 7, 5, and 2.
Two fundamental methodologies are used to construct a dendrogram using hierarchical clustering strategies: **divisive** or **agglomerative**.

- The divisive strategy (also called the downward or top-down strategy) starts from a single group encompassing all the instances and iteratively divides it into two disjointed subgroups, until each instance forms an independent group.
- The agglomerative approach (also referred to as the upward or bottom-up strategy) starts from the premise of the opposite scenario: at the outset, each instance is considered as an independent group and the algorithm must choose, at each step, which two groups to join to form a larger group.

In both cases, decisions regarding how to divide or join these groups at each step are made considering the distance between instances and between groups of instances.

At this point, it is important to note that the distance metrics defined above are functions that apply to individual instances, not to groups of instances. Therefore, how can we measure the distance between two clusters? The metrics usually used for this purpose are shown in table 2 and are calculated using a distance measured between two individual instances:
The distance between two groups is defined as:

<table>
<thead>
<tr>
<th>Name</th>
<th>The distance between two groups is defined as:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum distance (complete linkage)</td>
<td>The distance between the most different instances of these groups.</td>
</tr>
<tr>
<td>Minimum distance (single linkage)</td>
<td>The distance between the most similar instances of these groups.</td>
</tr>
<tr>
<td>Average distance (average linkage, UPGMA)</td>
<td>The average distance between all pairs of instances of one group and another.</td>
</tr>
<tr>
<td>Distance between centroids (UPGMC)</td>
<td>The distance between a representative of each group (centroid).</td>
</tr>
</tbody>
</table>

Table 2. Different means of calculating the distance between two clusters.

For a mathematical definition of each criterion, see:  

In Capsule 2 of this module, we will use hierarchical clustering algorithms with different metrics to measure the distance between instances and between clusters.

2.2.2 CLUSTERING BY PARTITIONING: K-MEANS

K-means is a partitioning clustering algorithm whose objective is to separate the objects into $K$ disjointed sets by minimizing the sum of the distances within the objects of each group. To achieve this goal, an object called a centroid that represents the instances of that group is defined for each of the $K$ groups. Centroids need not be instances of the initial data set; in fact, the most widespread practice is to create a new instance in which the value for each variable is the ‘average’ of the value of the cluster instances. The goal of $K$-means is to divide the instances into $K$ sets by minimizing the squared distance of the instances to their respective centroids. In Capsule 2 of this module, we will apply $K$-means clustering algorithms to identify the centroids in each group.

2.3 RESULTS EVALUATION AND DETERMINATION OF THE NUMBER OF CLUSTERS

Clustering algorithms are often used to determine the best number of groups in the data. Many algorithms require this number as a starting point before they can be run. Hierarchical clustering does not require a priori knowledge of the number of groups we will obtain in our data. However, after obtaining the dendrogram with the hierarchical relationships between groups, it is usual to split the tree at a certain height to obtain a decomposition into groups. Hence, asking the question: at what height do we split the dendrogram? is equivalent to asking: how many groups are there in our data?
Different metrics can allow us to estimate the best number of clusters for our data. One of the most popular metrics is the silhouette index (“silhouette”). This metric estimates how similar an instance is to instances in its own cluster (cohesion) compared to instances in other clusters (separation). The silhouette index takes values between $[-1, 1]$, and the higher the value, the more coherent and better separated the clusters obtained are. In Capsule 2 of this module, we will calculate these metrics and learn how to use them to estimate, a priori, the number of clusters that best reflect the structure of the data.

3. ASSOCIATION RULES

As discussed in Capsule 3 of Module 3 (Unsupervised learning), instances of a dataset are classically called transactions, the values of each instance are called items, and a set of items is called an itemset. Furthermore, association rules are used to represent dependencies between items in a dataset. These rules are expressions of the type $A \rightarrow C$, where $A$ and $C$ are itemsets whose intersection is empty. For instance, this rule represents the fact that when items from $A$ appear in a transaction, there is a high probability that items from $C$ will also appear in it. For instance, the following rule:

$$\text{mask} \rightarrow \text{gloves}$$

could be extracted from a hospital’s repository of medical supplies. This rule tells us that when medical personnel wear a mask, they also usually wear gloves. This type of rule allows us to identify associations between the equipment used, which helps the equipment manager to plan the orders to place to prevent potential product shortages.

Before extracting these rules, we must preprocess the dataset (eliminating confounding variables, etc.) and decide what we will consider as items and transactions. Depending on the appearance of our dataset, there may be several possibilities. For example, we can distinguish the following types of items:

- If each instance of the dataset is a list of items (e.g., a list of products in a supermarket), items are the elements that appear in the instances (e.g., an item could be diapers).
- If the data set contains a fixed number of variables and each instance contains a value for each variable, in this case an item is a “(variable, value)” pair. For a continuous variable, its domain is usually divided into intervals and the numerical value is replaced by the name of the interval to which it belongs.
3.1 CLASSIC QUALITY MEASURES

Association rules are commonly evaluated using the classical measures of support and confidence. The support measure is defined as:

- Support of an itemset \( X \): defined as the frequency with which the itemset \( X \) appears in the data set, as follows:

\[
Sup(X) = \frac{\text{number of occurrences of } X}{\text{total number of transactions}}
\]

- \( A \rightarrow C \) support rule: defined as the frequency with which the rule is satisfied in the data set, calculated as:

\[
Sup(A \rightarrow C) = \frac{\text{number of occurrences of } A \text{ and } C \text{ together}}{\text{total number of transactions}}
\]

This measure takes values in the domain \([0,1]\), where a support of 1 indicates that it appears in all the transactions in the data set while 0 indicates that it appears in none of them. The confidence measure indicates how many transactions in the data set in which the antecedent also appears as the consequent of the rule and is defined as:

\[
Conf(A \rightarrow C) = \frac{Sup(A \rightarrow C)}{Sup(A)} = \frac{\text{number of occurrences of } A \text{ and } C \text{ together}}{\text{umber of occurrences of } A}
\]

This measure takes values in the range \([0,1]\), where a confidence of 1 indicates that whenever \( A \) occurs, \( C \) also occurs, while 0 represents that when \( A \) occurs, \( C \) does not occur.

<table>
<thead>
<tr>
<th>ID</th>
<th>Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(Gender, Female), (Stress, High), (Blood pressure, Low)</td>
</tr>
<tr>
<td>2</td>
<td>(Gender, Female), (Stress, Medium), (Blood pressure, Low)</td>
</tr>
<tr>
<td>3</td>
<td>(Gender, Male), (Stress, Low), (Blood pressure, Low)</td>
</tr>
<tr>
<td>4</td>
<td>(Gender, Male), (Stress, High), (Blood pressure, High)</td>
</tr>
</tbody>
</table>

*Table 3. Data set of clinical variables represented as transactions.*
For instance, consider the data set shown in table 3 and the rule (Gender, Female) \(\rightarrow\) (Blood pressure, Low). The support and confidence values for this rule are as follows:

- \(\text{Sup}(\text{Gender, F}) = \frac{2}{4} = 0.5\)
- \(\text{Sup}(\text{Gender, F}) \land \text{Blood pressure, Low}) = \frac{2}{4} = 0.5\)
- \(\text{Conf}(\text{Gender, F} \rightarrow \text{Blood pressure, Low}) = \frac{0.5}{0.5} = 1\)

Note that, if we interchange the antecedent and consequent in a rule, the rule will have the same support, but may have different confidence levels. In our example, if you are female, you will generally have low blood pressure, but if you have low blood pressure you need not be female:

- \(\text{Sup}(\text{Blood pressure, Low}) = \frac{3}{4} = 0.75\)
- \(\text{Sup}(\text{Blood pressure, Low}) \land \text{Gender, F}) = \frac{2}{4} = 0.5\)
- \(\text{Conf}(\text{Blood pressure, Low} \rightarrow \text{Gender, F}) = \frac{0.5}{0.75} = 0.66\)

### 3.2 CLASSICAL ASSOCIATION RULE EXTRACTION ALGORITHMS

To generate the association rules, the classical approach first generates all the itemsets with support levels equal to or higher than a user-defined threshold, called frequent itemsets. It then uses them to generate all the rules with a confidence level equal to or higher than a user-defined threshold. In the following sections we briefly introduce two of the most widely used classical algorithms: Apriori and FP-growth.

#### 3.2.1 APRIORI ALGORITHM

Apriori was the first algorithm used to obtain association rules from a data set. This algorithm creates a tree with all the possible combinations between the items in the data set. At each level of the tree, the size of the itemsets is increased by 1 item. To make the process more efficient, it eliminates infrequent itemsets from the tree because any other itemset containing them will not be frequent either. Once the tree is generated, the algorithm generates all the association rules with a confidence higher than the indicated threshold based on the frequent itemsets. Figure 5 shows an example of the itemsets that are not generated by eliminating a 2-itemset that is not frequent.
3.2.2 THE FP-GROWTH ALGORITHM

FP-growth initially generates a tree structure, called FP-tree, which allows us to easily know how many times each itemset appears in the dataset. To do this, we go down the tree to the last item containing the itemset and query the node counter. For example, itemset B,C,E appears in only 1 transaction in the dataset according to the FP-tree in figure 6. Once this structure is generated, the algorithm generates the frequent itemsets which are then used to create the association rules with confidence values exceeding the threshold.

![Figure 6. Example of a FP-tree generated from a data set.](image-url)
Both the Apriori and FP-growth algorithms generate exactly the same rules, but the FP-growth algorithm is more efficient because it usually requires less memory and it has a shorter execution time. Therefore, we will use the FP-growth algorithm to perform the analysis in Capsule 3 of this module.

### 3.3 OTHER QUALITY METRICS

Classical association rule extraction algorithms guide their search by using classical support and confidence metrics. However, these metrics present several problems in the process. On the one hand, an item with very high support can be added to any itemset without affecting its support, allowing the creation of many frequent itemsets. This causes algorithms to need more memory and require longer execution times. On the other hand, the confidence measure does not consider the support of the consequent, so if a rule has a very frequent itemset in the consequent, the rule will have a high value for confidence. Any antecedent that the rule has will appear to be a good predictor of the consequent. For instance, consider the following rules:

- Rule 1: Conf (A→B) = 1.0
- Rule 2: Conf (C→D) = 0.8

In principle, according to the confidence measure, it seems that rule 1 is the best. However, let us assume that the support for itemset A is 0.2, for itemset B it is 0.9, for itemset C it is 0.3, and for itemset D it is 0.4. As we can see, itemset B appears in 90% of the transactions; table 4 shows the 10 transactions in the data set.

<table>
<thead>
<tr>
<th>ID</th>
<th>Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>E, F, G</td>
</tr>
<tr>
<td>2</td>
<td>F, H, I</td>
</tr>
<tr>
<td>3</td>
<td>B, G, I</td>
</tr>
<tr>
<td>4</td>
<td>B, E, H</td>
</tr>
<tr>
<td>5</td>
<td>B, C, D</td>
</tr>
<tr>
<td>6</td>
<td>B, C, D</td>
</tr>
<tr>
<td>7</td>
<td>B, C, D</td>
</tr>
<tr>
<td>8</td>
<td>B, C, D</td>
</tr>
<tr>
<td>9</td>
<td>A, B, C</td>
</tr>
<tr>
<td>10</td>
<td>A, B, H</td>
</tr>
</tbody>
</table>

*Table 4. Transactions in which itemsets A, B, C and D appear.*
Table 4 shows how itemset B appears in so many transactions in the data set that itemset A appears to be a good predictor. However, itemset A only appears in a reduced set of examples of those containing itemset B. In contrast, itemset D appears in almost all the transactions in which itemset C appears, meaning that itemset C seems to be a good predictor of itemset D. Therefore, after analyzing the rules, we can conclude that rule 2 is more interesting to us than rule 1.

Because of these problems, over the last few years researchers have proposed other metrics to select and rank rules according to their potential interest to users. Some of the most widely used are:

- **Lift**: This measure represents the ratio between the confidence of the rule and the expected confidence of the rule. Its domain is \([0, \infty]\), where values less than 1 indicate negative dependence, 1 indicates independence, and values greater than 1 indicate positive dependence. Since it is not upper bounded, it is difficult to define a threshold from which to study rules. Because of this, we will study any rule with a lift value exceeding 1, without a higher value for the measure meaning that the rule is better.

- **Leverage**: This metric measures the difference between the observed and expected joint probability of AC, assuming that A and C are independent. Its domain is \([1, 1]\), where values less than 0 represent negative dependence, 0 represents independence, and values greater than 0 represent positive dependence.

- **Conviction**: This measures the expected error of the rule. In other words, how often A appears in a transaction in which C does not appear. Its domain is \([0, \infty]\), where values less than 1 represent negative dependence, 1 represents independence, and values greater than 1 represent positive dependence. This metric is not upper bounded and so it has the same problem as the lift metric. Note that if the confidence of the rule is 1, this metric is \(\infty\) because the expected error is 0.

All these metrics fulfill many of the properties that are desirable in the metrics of interest for the rules, but they all present particular problems. Because of this, it is advisable to analyze the rules by considering several metrics of interest.
REFERENCES

- J. Han, M. Kamber. Data Mining: Concepts and Techniques (3ª ed.). Morgan Kaufmann, 2011.
- P-N. Tan, M. Steinbach, A. Karpatne, V. Kumar. Introduction to Data Mining (2ª ed.). Pearson, 2019

ADDITIONAL REFERENCES

- P. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Alberta (Canada), 2002, 32–41.