An Easy Introduction to Clustering and Classification

Bioinformatics Lab, University of Ljubljana, Slovenia

October 15, 2024

Welcome! In this guide, we'll explore two key types of machine learning: **unsupervised learning** and **supervised learning**. Unsupervised learning helps us discover hidden patterns in data, such as grouping students by their grades or identifying similar dog breeds—this is where **clustering** comes in. Supervised learning, on the other hand, focuses on making predictions from labeled data—this is the realm of **classification**, where we can predict outcomes like whether a student will pass or fail based on their grades.

1 The Data

Before we can do any clustering or classification, we need data. Imagine we have the grades of five students in two subjects: Math and English (Table 1). Each student has a score in both subjects, and we can think of this as a little snapshot of their academic performance. We'll represent each student by their scores, like Alice with 85 in Math and 78 in English. By turning these scores into data, we can start exploring relationships between students—like who has similar scores or who might be struggling in certain subjects.

Table 1: A sample data set of student grades in Math and English.

Student	Math	English
Alice	85	78
Bob	90	88
Carol	95	92
David	50	65
Eve	55	60

In this data table, each row represents a student, and the two columns show their grades in Math and English. With this data, we can start looking for patterns, like which students have similar academic profiles or what subjects they excel in.

2 Distances

Once we have data we can asses how similar or different these data instances, that is, in our example, students, are. One way to do this is by calculating the **distance** between them. Think of distance as a way of comparing two things. For example, if Alice and Bob have very similar grades, the distance between them will be small. We use different methods for measuring distance, such as:

- **Euclidean distance**: This is like measuring the straight line between two points on a map. The more different the grades, the bigger the distance.

- **Cosine distance**: This looks at the angle between two points. This method is great when we care more about the direction of the data rather than the actual values, and we should use it when the data has many features (our student data has only two features).

Let's compute the **Euclidean distance** between Alice and Bob, based on their grades in Math and English:

- Alice's grades: Math = 85, English = 78
- Bob's grades: Math = 90, English = 88

The Euclidean distance between them is calculated using the formula:

$$d = \sqrt{(85-90)^2 + (78-88)^2} = \sqrt{(-5)^2 + (-10)^2} = \sqrt{25+100} = \sqrt{125} \approx 11.18$$

So, the Euclidean distance between Alice and Bob is approximately 11.18. The larger the distance, the more different their grades are. We can now compute all the distances and show them in the distance matrix (Figure 1). According to our data, Bob and Carol have the smallest distance, which means they have similar grades.

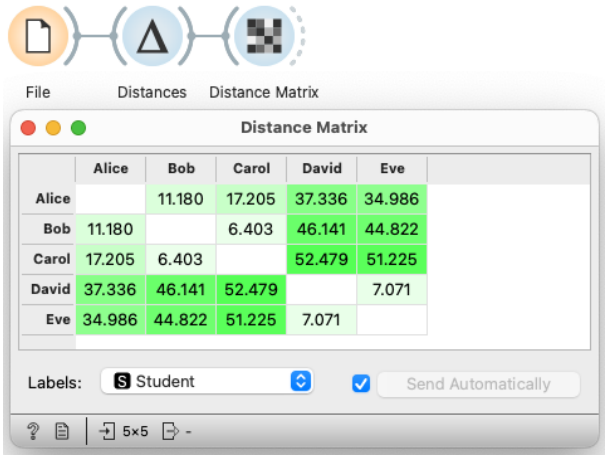

Figure 1: Distance matrix for the student data as displayed by Orange Data Mining software.

Euclidean distance can easily be extended to more dimensions, making it a versatile tool for comparing data points in many different contexts. The distance between two points is then simply the square root of the sum of the squared differences in each dimension.

3 Hierarchical Clustering

Now that we can measure distances, we can start grouping students who are close together. This brings us to **hierarchical clustering**, which is like building a family tree. It starts by treating each student as their own cluster, then gradually merges the closest students together until everyone is in one big group. To estimate the distance of two clusters, two popular metrixs are **average linkage** that finds the average distance between all points in two clusters, and **ward linkage**, which merges the clusters in a way that keeps them as compact as possible.

The result of hierarchical clustering can be visualized using a **dendrogram**—a tree-like diagram that shows how the clusters were formed (Figure 2). Here is the dendrogram for our student data, which we have also cut to expose the two clusters. Note, however, that our data set is very small and to find someting statistically significant, we would need a larger data set:

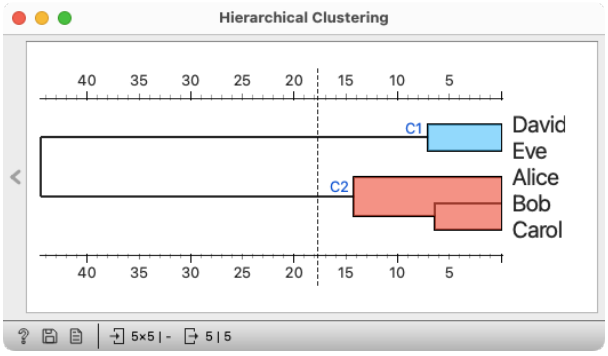

Figure 2: Dendrogram showing the hierarchical clustering of students based on their grades in Math and English.

4 Other Clustering Techniques

There are other clustering methods you might hear about, each with its own special tricks:

- **k-means**: This method tries to split the data into a set number of groups by finding the best way to divide the data into clusters. It's fast, can be applied to very large data sets, and works well when we know how many groups we want.
- **DBSCAN**: This method is great when we don't know how many groups there are and when we want to ignore outliers (points that don't fit into any group). It finds clusters based on how dense the data is.
- **Gaussian Mixture Models**: This method assumes the data comes from overlapping groups and identifies these groups. Each data point can belong to multiple clusters, with a probability assigned to its membership in each cluster.

5 Classification and Accuracy

While clustering helps us group data, **classification** is all about making predictions. For example, imagine we want to predict whether a student will pass or fail based on their grades. In this case, we already know the outcome for each student (pass or fail), and we want to train a model to predict the outcome for new students.

In classification, we train the model on a set of labeled data (where we already know the answers), and then use it to predict the class (or category) for new data points. One important concept in classification is **accuracy**, which tells us how often the model gets the predictions right. The higher the accuracy, the better the model is at making correct predictions.

Some popular classification techniques include:

- **Logistic regression**: A simple but effective method for binary classification problems (where there are two possible outcomes, like pass/fail). Logistic regression build a linar model, a hyperplane, that separates the data into two classes. In two dimensions, this hyperplane is simply a line that separates points of different classes.
- **Classification trees**: These models split the data into branches based on certain features, eventually leading to a prediction.
- **Random forests**: A more advanced method that builds many classification trees and combines their predictions to improve accuracy. While random forests are usually more accurate than a single tree, they are also more complex and hard or even impossibe to interpret.

6 Clustering of Unstructured Data

Clustering doesn't just work on numbers or grades—it can also be applied to unstructured data like images and text. To do this, we use something called **embedding**, which turns images or text into numbers that a computer can work with (we use embedding in Figure 3). For example:

- **Images**: A deep learning model like a neural network can look at an image and turn it into a series of numbers that represent what's in the picture.
- **Text**: A model like BERT can take a sentence or document and turn it into numbers based on the meaning of the words.

Once these images or texts are turned into numbers, we can apply clustering to group similar items together. For example, we can cluster dog breeds based on their images or articles based on their topics.

7 t-SNE for Dimensionality Reduction

Sometimes data can have many dimensions or attributes, making it hard to visualize. This is where **t-SNE** (t-Distributed Stochastic Neighbor Embedding) comes in handy. It's a method that reduces high-dimensional data to just two or three dimensions, making it easier to see and understand. For example, if we're working with images of dogs or articles from the web, t-SNE helps us create a map of the data, where similar items are placed closer together (Figure 3).

Figure 3: A t-SNE plot of images of dogs of various breeds. We selected four neighboring data points (dog images) and they indeed look similar. Notice also that we first need to embed the images into a vector space before applying t-SNE.

8 Conclusion

Clustering and classification are powerful ways to find patterns and make predictions in data, whether it's grouping students, predicting exam results, or analyzing images and text. Each method has its own strengths, and the choice depends on the structure of the data and the goals of the analysis. With these foundational techniques, you're ready to start exploring and applying machine learning to real-world problems.