Making Sense of Unstructured Data
Week 3

Learning Objectives of the Session

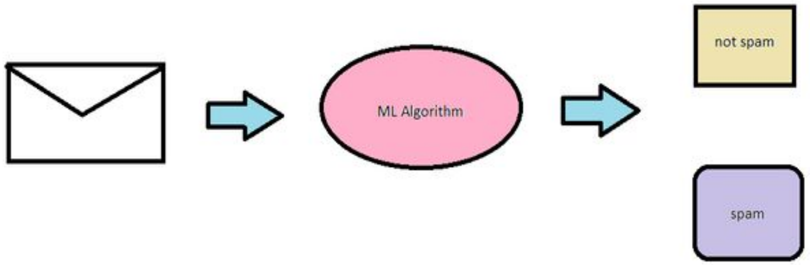
- Introduction to Supervised and Unsupervised Learning
- Clustering
- K-means Clustering
- K-medoids Clustering
- Gaussian Mixture Models (GMMs)
- Spectral Clustering
- Principal Component Analysis

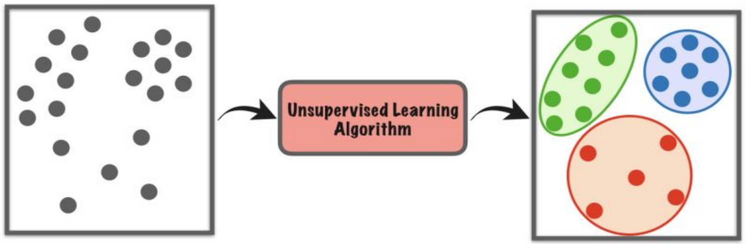
Discussion Questions

1. What is the difference between Supervised Learning and Unsupervised Learning?
2. What is clustering and what are the most common clustering techniques?
3. Why and when should you use clustering?
4. How does clustering help in finding hidden groupings and patterns?
5. What is K-means Clustering?
6. Why is K-means Clustering so popular? What are its assumptions?
7. Why is converting features into continuous values and scaling the data important while performing K-means Clustering?
8. How do you determine the value of K in K-means Clustering?
9. What is Spectral Clustering?
10. How does PCA help in Dimensionality Reduction?

Supervised vs Unsupervised Learning

Supervised learning algorithms are trained using labeled data. Example - Predict whether a new email is SPAM or NOT SPAM, predict whether a customer will churn or not churn, etc.





Unsupervised learning algorithms are trained using unlabeled data. It learns on itself and finds patterns in the data to form different groups within the dataset. For example: Customer segmentation, image segmentation etc.

What is clustering?

Cluster analysis, or clustering, is an unsupervised machine learning technique. It involves automatically discovering natural groupings in data (groups data according to the notion of similarity).

The most common types of clustering are K-means clustering, LDA Clustering, Spectral Clustering, Modularity Clustering, Hierarchical Clustering, DBSCAN clustering etc.

Image source

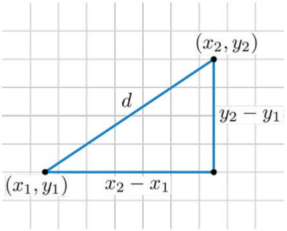
Unsupervised Learning: Clustering

Cluster analysis, or clustering, is an unsupervised machine learning technique. It involves automatically discovering natural groupings in the data (it groups data points according to the notion of similarity).

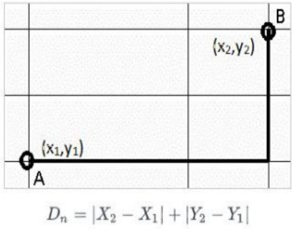
Similarity: In clustering, we can find the similarity between two data points based on some kind of distance metric between them, and group them together.

There are different distance measures which can be used to find similarity. Some of these are:

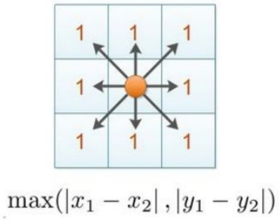
1. Euclidean distance
2. Manhattan distance
3. Chebyshev distance



Euclidean distance



Manhattan distance



Chebyshev distance

Why and When to Use Clustering

Why

Retail, finance, and marketing are some of the key domains that use clustering methods to analyze their data. Clustering techniques help decision-makers in these domains gain further customer insights.
The factors analyzed through clustering can have a big impact on sales and customer satisfaction, making it an invaluable tool to improve overall business performance.

When

- When you are starting out with a large and unlabeled dataset.
- When you do not know how many or which classes your data is divided
- When annotating (labeling) your data can be very expensive.

How does clustering help in finding hidden groupings and patterns?

Clustering methods simply try to group similar patterns into clusters whose members are more similar to each other (according to some **distance measure**) than to members of other clusters.

Image source

K-means Clustering

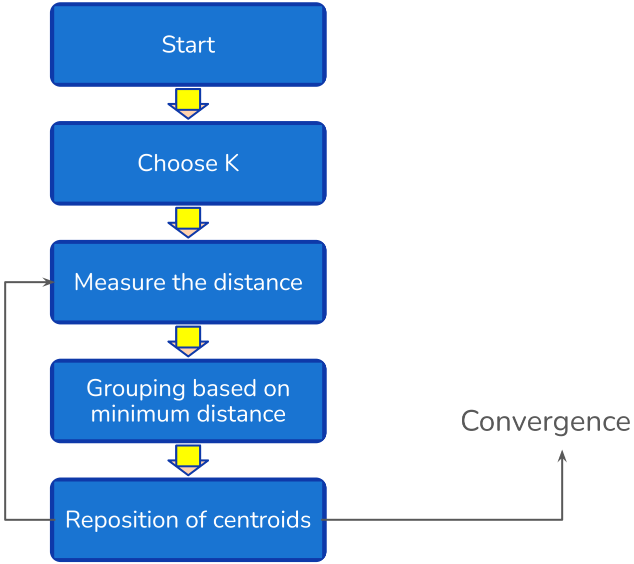
K-means Clustering is an **iterative algorithm** that divides the unlabeled dataset into **K different clusters**, in such a way that each point in the dataset belongs to only one group that has similar properties.
The algorithm starts with initial estimates for the **K** centroids, which can be randomly generated or randomly selected from the data set.

The algorithm then iterates between two steps:

1. The Data Assignment step:
Each centroid defines one of the clusters. In this step, each data point based on the squared Euclidean distance is assigned to its nearest centroid. If ci is the collection of centroids in set C, then each data point x is assigned to a cluster based on where dist(·) is the standard (L2) Euclidean distance. **Min dist(C,x)**

2. The Centroid Update step:
Centroids are recomputed by taking the mean of all data points assigned to that centroid cluster.
The algorithm iterates between step 1 and 2 until a stopping criteria is met.
This algorithm may converge on a local optimum. Assessing more than one run of the algorithm with randomized starting centroids may give a better outcome.

K-means Clustering - Steps



If cluster is not stable

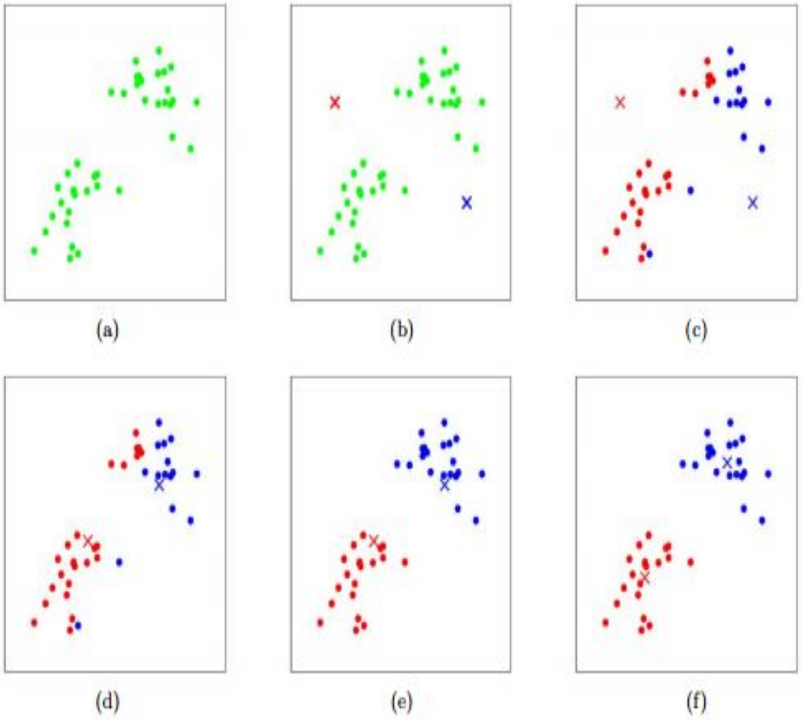


Image source

K-means Clustering - Advantages and Disadvantages

Advantages:

- K-means Clustering is relatively simple to implement
- It scales well to large datasets
- It guarantees convergence
- It can easily adapt to new examples

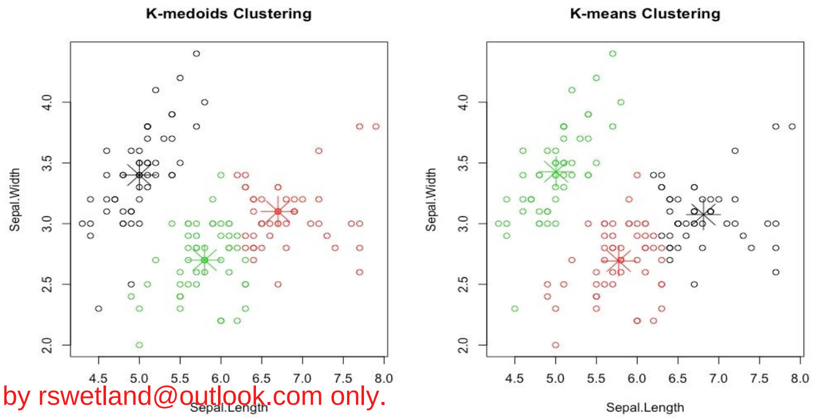
Disadvantages:

- It is not always straightforward to identify the correct value of K
- K-means Clustering has trouble clustering the data when clusters are of varying sizes and density
- It can easily get affected by outliers
- It assumes the cluster shapes to be spherical in nature, and does not perform well on arbitrary data
- It depends on the initial values assigned to the centroids, and may give different results for different initializations

K-medoids Clustering (PAM) - Alternative to K-means

- One problem with K-means Clustering is that the final centroids are not interpretable i.e. The centroids are not actual points, but the means of the points present in the cluster.
- The idea behind K-medoids Clustering is to make sure the final centroids are themselves actual data points so that they are interpretable, and somewhat representative of their cluster.
- In K-medoids, we only change one step from K-means which is the step around updating the centroids. In this process, if there are m points in a cluster, swap the previous centroids with all other (m-1) points from the cluster and finalize the point as new centroid which has minimum loss.

- Because of this, unlike K-means, K-medoids is robust to outliers and converges fast.
- You can see in this image that the centroids in K-medoids are the actual data points represented as the cross, unlike K-means.



Case Study - Clustering

Dimensionality Reduction

What is Dimensionality Reduction?

- Dimensionality reduction is the process by which we reduce the number of dimensions or features in our dataset.

The Need for Dimensionality Reduction

- In modern-day machine learning, we tend to collect and utilize many features (information) about each data point, in an attempt to get more accurate results. However, as we start increasing the number of features (columns) in our dataset, after a certain point, the performance and robustness of the model starts decreasing in addition to the obvious increased computational complexity. This is sometimes called “The Curse of Dimensionality”.
- In addition, not all of these multiple features may be equally relevant to our machine learning problem, meaning the value added by some of them may be minimal and they can be be removed.
- Hence, we use Dimensionality Reduction techniques to remove features and transform the data into lower dimensions, while at the same time, keeping most of our original information intact. This also helps us with data visualization, as it is easier to visualize data in lower-dimensional space.

Principal Component Analysis

Principal Component Analysis, or PCA, is a popular method for dimensionality reduction. It is used to transform your data into a lower-dimensional space by projecting the data onto new lower-dimensional axes.

These axes are called **Principal Components**.

The selection of the principal components is such that they retain the maximum variation present in the original variables on the first principal component, and the variation decreases as we move down the order. All Principal Components are orthogonal axes to each other.

Steps for PCA:

- Begin by standardizing the data
- Generate the covariance matrix
- Perform eigenvalue decomposition
- Sort the eigen pairs in descending order
- Order and select the largest one

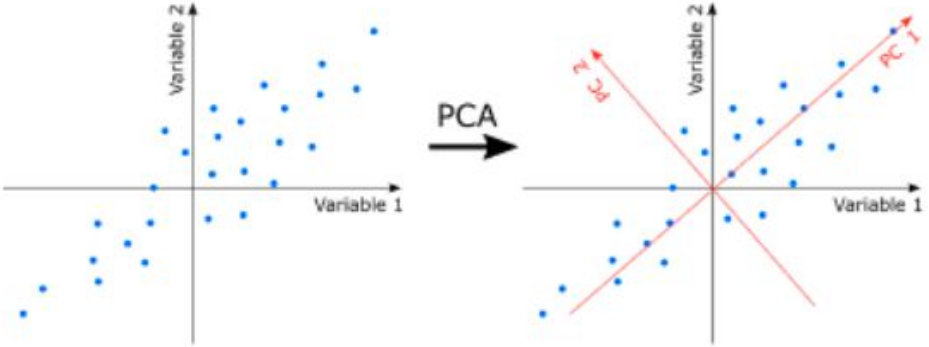


Image source

Case Study - PCA

Optional Section - Additional Reading

Gaussian Mixture Models - Expectation Maximization

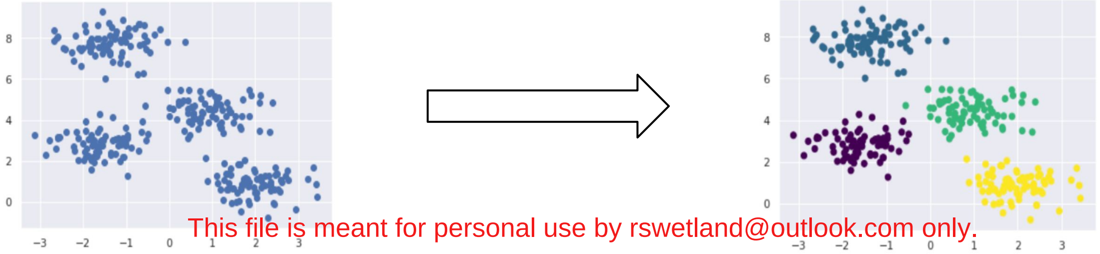
In GMMs, we need the parameters of each Gaussian (variance, mean etc.) in order to cluster our data, but we need to know which sample belongs to what Gaussian in order to estimate those same parameters.

That is where we need the Expectation Maximization (EM) algorithm. There are two steps involved in this algorithm:

1. **The E-step:** It estimates the probability for a given observation to be in a cluster / distribution. This value will be high when the point is assigned to the right cluster and lower otherwise.
2. **The M-step:** In this step we want to maximize the likelihood that each observation came from the distribution.

After that we reiterate these two steps and updates the probabilities of an observation to be in a cluster.

An example of GMM clustering



Spectral Clustering

In spectral clustering, **data points are treated as nodes of a graph**. Thus, spectral clustering is a graph partitioning problem. The nodes are then mapped to a low-dimensional space that can be easily segregated to form clusters. No assumption is made about the shape/form of the clusters. **The goal** of spectral clustering is to cluster data that is connected but not necessarily compact or clustered within convex boundaries.

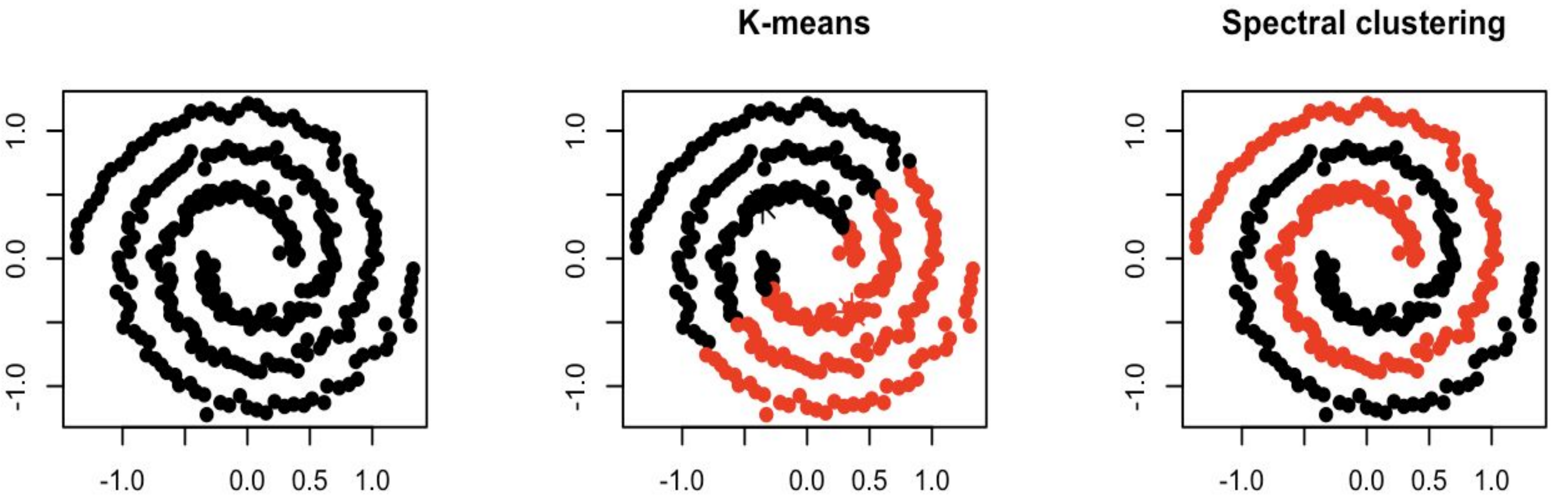
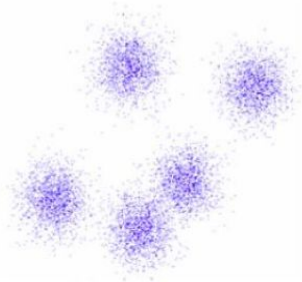


Image source

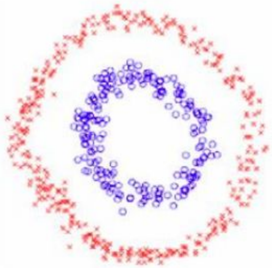
K-means Clustering vs Spectral Clustering

- **Compactness:** Points that lie close to each other fall in the same cluster and are compact around the cluster center. The closeness can be measured by the distance between the observations. E.g.: K-means Clustering
- **Connectivity:** Points that are connected or immediately next to each other are put in the same cluster. Even if the distance between 2 points is less, if they are not connected, they are not clustered together. Spectral Clustering is a technique that follows this approach. K-means Clustering will fail to effectively cluster these, even when the true number of clusters K is known to the algorithm.

K-means, as a *data-clustering* algorithm, is ideal for discovering globular clusters where all the members of each cluster are in close proximity to each other (in the Euclidean sense).



Compactness



Connectivity

Image source

Great Learning

Happy Learning !