Clustering
K-Means Clustering
Dimensionality Reduction

Lecture # 13

Definition

The process of finding meaningful groups in data is Clustering. The objective here is not to predict a target class variable, but to capture the possible natural groupings in the data.

Examples:

The customers of a company can be grouped based on purchase behavior. The prospective electoral voters can be clustered into different groups so that candidates can tailor their message to resonate within each group.

Clustering vs. Classification

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

In clustering one would not know ahead what groups to look for and therefore, the inferred groups might be difficult to explain.

Clustering is used in two different classes of applications:

To describe a given dataset and as a preprocessing step for other data science algorithms.

Clustering of the Data

A common application of clustering is to explore the data and find all possible meaningful groups in the data. For example, clustering of the customers of a store can yield a few groups. The customers within a certain group have more in common with one another as compared to the customers in a different group. The number of groups or clusters are either user-defined or automatically determined by the algorithm from the dataset.

Clustering and Dimensionality Reduction

Clustering process considers all the attributes of the dataset and reduces the information to a cluster, which is another attribute (i.e. the ID of the cluster to which a record belongs). Therefore, clustering can be used as a data compression technique. The output of clustering is the cluster ID for each record and it can be used as input variable for another data science task. Clustering can be used for two types of processing:

Dimensionality Reduction: In an N-dimensional dataset (N being the number of attributes), the computational complexity is proportional to the number of dimensions, N. In clustering, the N-dimensional attributes can be reduced to one categorical attribute- i.e. Cluster ID. This reduces the complexity with some loss in information.

Object Reduction: Consider the number of customers for an organization (i.e. millions) with a much smaller number of cluster groups (hundreds). For each of these cluster groups, a customer can be identified that represents the characteristics of all the customers in that group. This greatly reduces the record counts and the dataset can be made appropriate by classification by other techniques (i.e. K-NN).

Types of Clustering Techniques

Based on the data point memberships, a cluster can be:

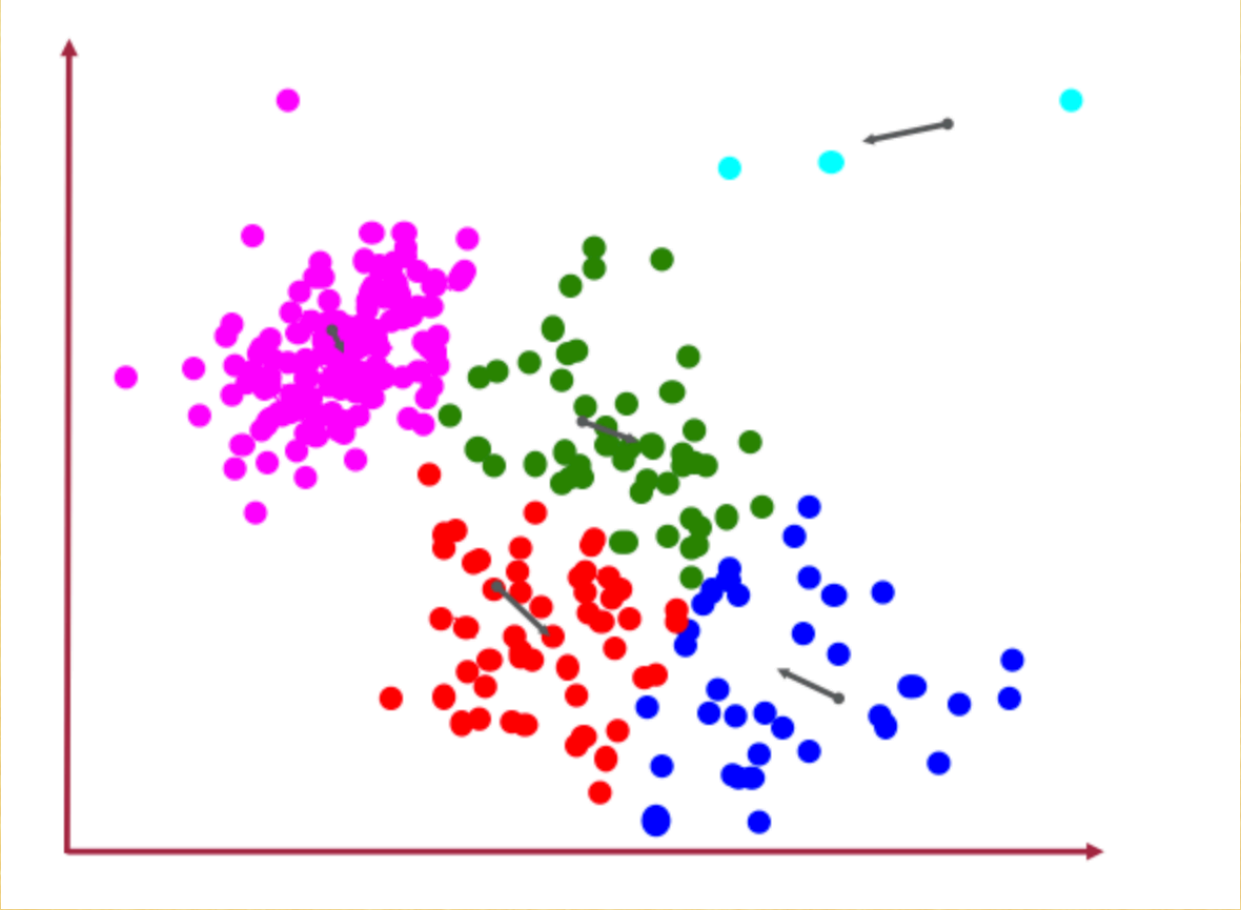
Exclusive: Each data point belongs to one exclusive cluster

Overlapping: the cluster groups are not exclusive, with each data point belonging to more than one cluster.

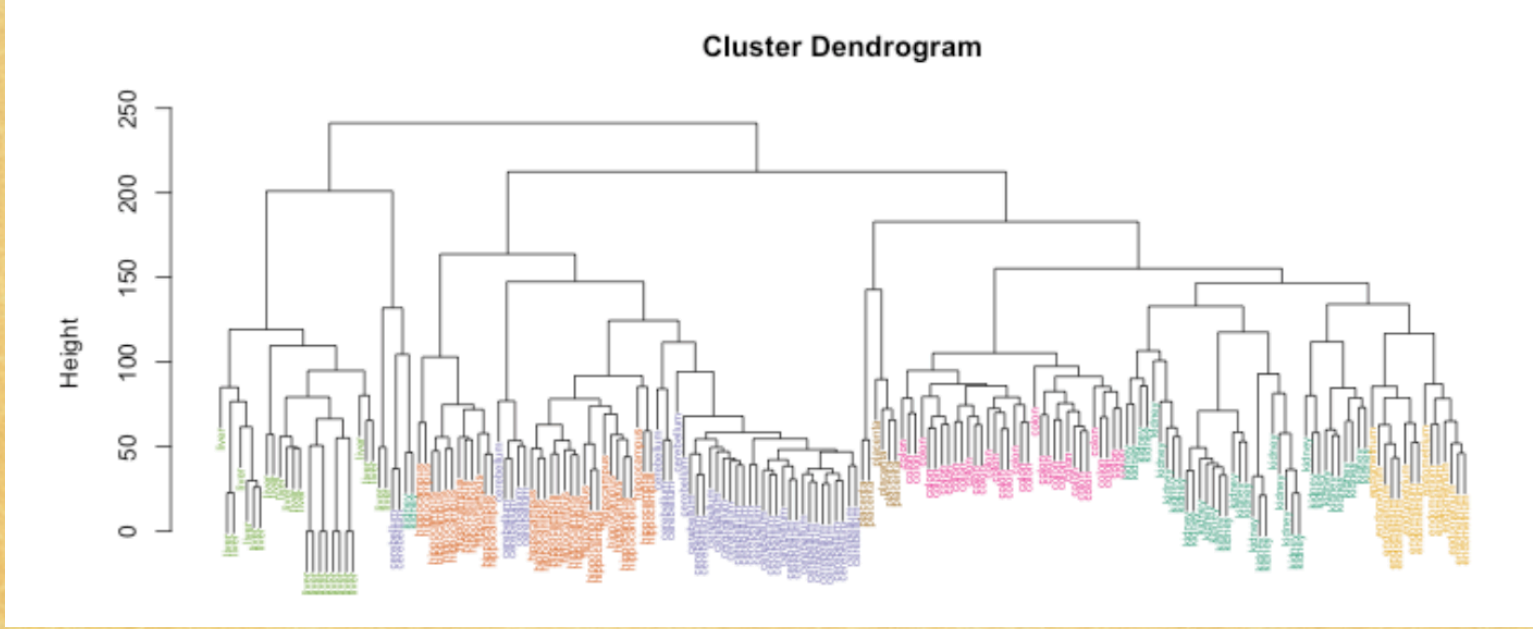
Hierarchical: Each child cluster can be merged to form a parent cluster.

Fuzzy or Probabilistic: Each data set belongs to all cluster groups with varying degrees of membership from 0 to 1. Here, instead of a definite association of a dat to a cluster, we associate a probability membership to all the clusters.

Clustering



Hierarchical Clustering



K-Means Clustering

K-means clustering is a method where the dataset is divided into k-clusters. In this technique the user specifies the number of clusters (k) that need to be grouped in the dataset. 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. The prototype is called the centroid, the center of the cluster. The center of the cluster can be the mean of all data objects in the cluster. The centroid represents the characteristic of all the data points within the cluster.

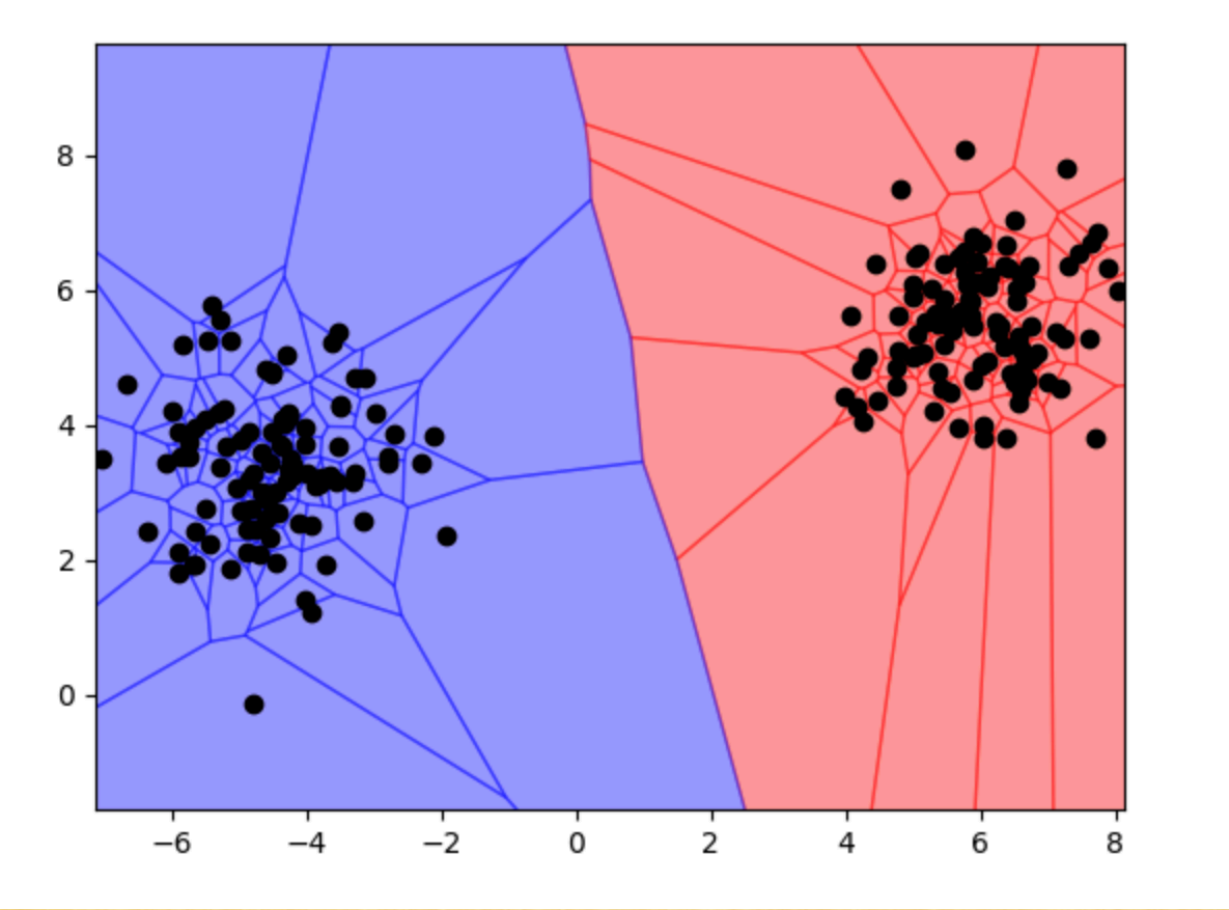
The k-means algorithm divides the data space in k partitions or boundaries where the centroid in each partition is the prototype of the cluster. The data points inside the partition belong to the cluster. The partitions are called Voronoi partitions and each prototype is a seed in a Voronoi partition.

A Voronoi partition is the process of segmenting a space into regions around a set of points called seeds. All the other points are then associated to the nearest seed and the points associated with that seed.

K-means clustering is one of the simplest and most commonly used clustering algorithms.

K-means clustering creates k partitions in N-dimensional space, where N is the number of attributes in a given dataset. To partition the dataset, a proximity measure needs to be defined. The most commonly used measure for a numeric attribute is the Euclidean distance.

Voronoi Clustering



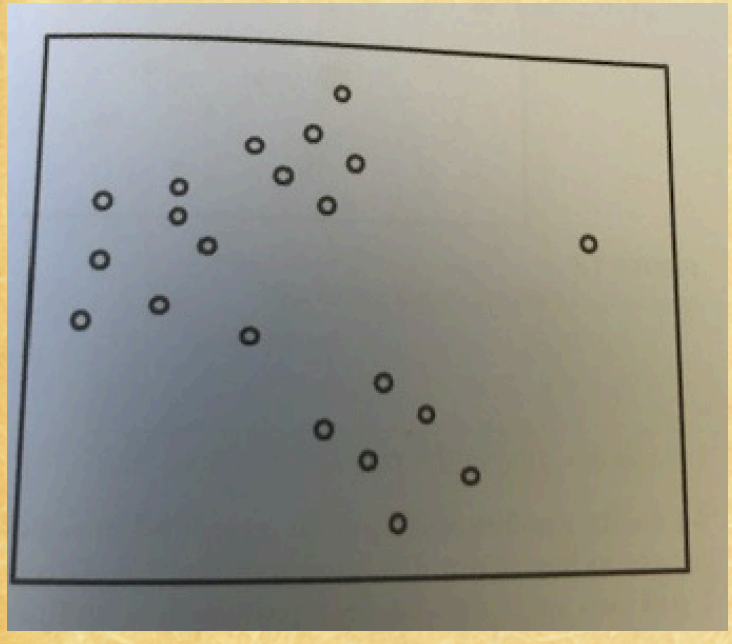
Implementing K-Means Clustering

The process of k-meadns clustering is similar to Voronoi iteration where the objective is to divide a space to cells around points. The difference is that in the Voronoi the space is divided into cells whereas k-means clustering partitions the points in data space. Here is a step-by-step instructions of K-means clustering:

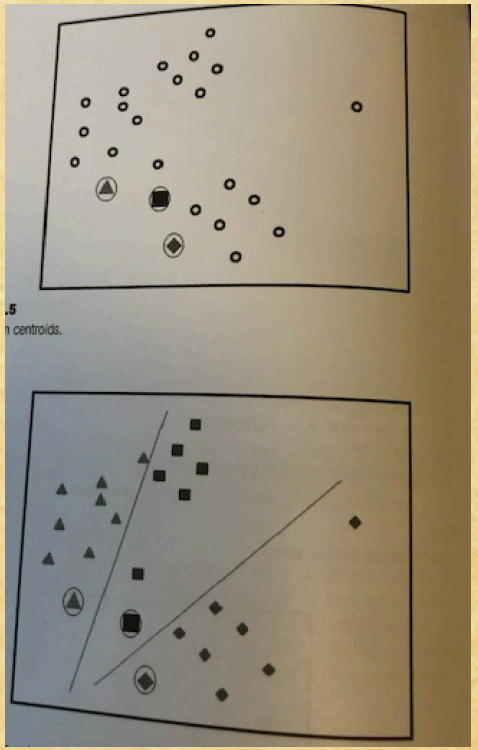
1. Initiate Centroids: we initiate k random centroids. The number of clusters, k, should be specified by the user.

2. 2. Assign data points: Once centroids are initiated, all the data points are assigned to the nearest centroid to form a cluster. Euclidean distance measurement is applied to measure distances (proximity). The Euclidean distance between two data points $X(x_1, x_2, \ldots, x_n)$ and $C\ (c_1, c_2,\ldots,c_n)$ with N attributes is: $d=[(x_1 - c_1)^2 + (x_2 - c_2)^2 + \ldots + (x_n - c_n)^2]^{1/2}$ This step also leads to partitioning of data space into Voronoi partitions. The data with similar attributes are now separated into groups or clusters.

Search for Clustering



Optimizing the centroids



3. Calculate New Centroid: For each cluster a new centroid can now be calculated. The new centroid is the most representative of all the data points in the cluster. This step is done by minimizing the sum of the squared errors (SSE) of all data points in a cluster from the centroid of the cluster. The aim of this step is to minimize the SSEs of individual clusters. This is calculated as

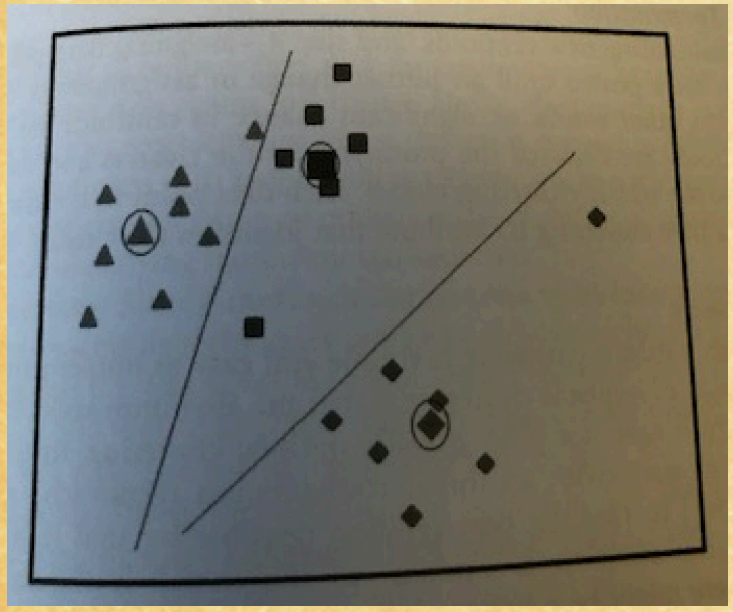
$$SSE = \sum_{i=1}^{k} \sum_{x_j \in C_i} |x_j - \mu_i|^2$$

Where C_i is the i^{th} cluster, j are the data points in a given cluster, μ_i is the centroid for i^{th} cluster and x_j is a specific data point. The centroid with minimal SSE for the given cluster I is the new mean of the cluster. The mean of the cluster is calculated as

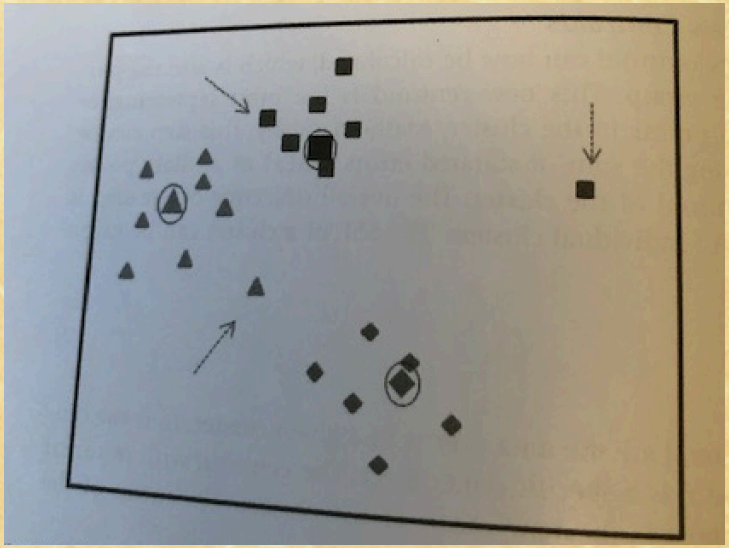
$$\mu_i = \frac{1}{j_i} \sum_{x \in C_i} X$$

Where X is the data object vector (x_1, x_2, …, x_n). In the case of k-means clustering the new centroid will be the mean of all the data points. K-Medoid clustering is a variation of K-means clustering where the median is calculated instead of the mean.

Determination of Centroid



Final Centroid



4. Repeat assignment and calculate new centroid: Once new centroids are identified, assigning data points to the nearest centroid is repeated until all the data points are reassigned to new centroids.

5. Steps 3 (calculating a new centroid) and 4 (assigning data points) to new centroids are repeated until no further change in assignment of data points happens- i.e. no significant change in centroids is noted. The final centroids are the prototypes of the clusters and used to describe the clustering model.

Density Based Clustering

Density Based Spatial Clustering and Application with Noise (DBSCAN) is a density-based clustering algorithm used to identify clusters of any shape in data set containing noise and outliers.

Advantages of DBSCAN are:

- Unlike K-means, DBSCAN does not require the user to specify the number of clusters to be generated
- DBSCAN can find any shape of clusters. The cluster doesn't have to be circular.
- DBSCAN can identify outliers

How DBSCAN Technique Works

- For each point x_i , compute the distance between x_i and the other points. Finds all neighbor points within distance **ε** of the starting point x_i. Each point, with a neighbor count greater than or equal to **MinPts**, is marked as **core point** or **visited**.
- For each **core point**, if it's not already assigned to a cluster, create a new cluster. Find recursively all its density connected points and assign them to the same cluster as the core point.
- Iterate through the remaining unvisited points in the dataset.

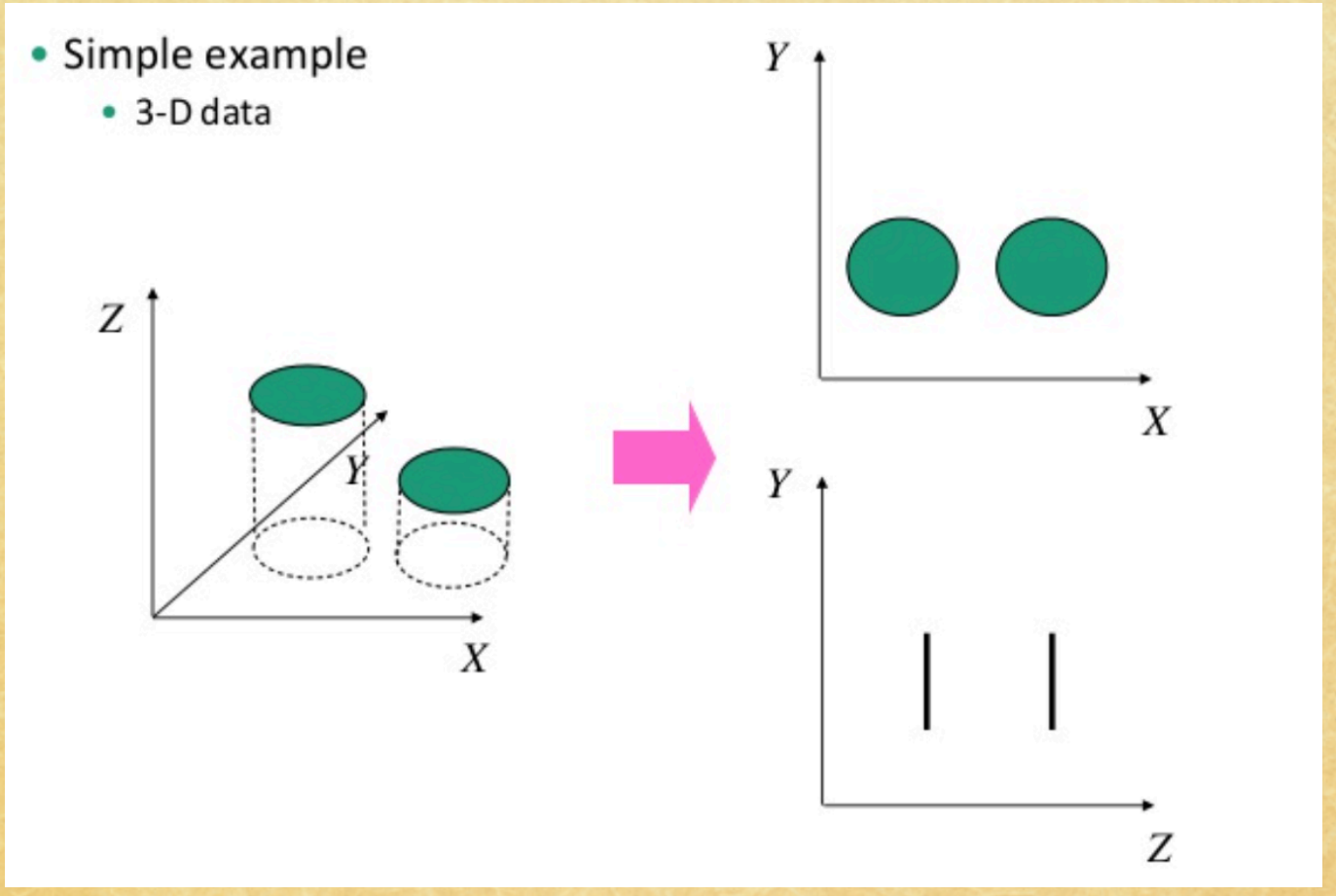
Dimensionality Reduction

Dimensionality Reduction

In machine learning classification problems, there are often too many factors on the basis of which the final classification is done. These factors are basically variables called features. The higher the number of features, the harder it gets to visualize the training set and then work on it. Sometimes, most of these features are correlated, and hence redundant. This is where dimensionality reduction algorithms come into play. Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables. It can be divided into feature selection and feature extraction.

A classification problem that relies on both humidity and rainfall can be collapsed into just one underlying feature, since both of the aforementioned are correlated to a high degree. Hence, we can reduce the number of features in such problems. A 3-D classification problem can be hard to visualize, whereas a 2-D one can be mapped to a simple 2 dimensional space, and a 1-D problem to a simple line. The below figure illustrates this concept, where a 3-D feature space is split into two 1-D feature spaces, and later, if found to be correlated, the number of features can be reduced even further.

From GeeksforGeeks https://www.geeksforgeeks.org/dimensionality-reduction/



Components of Dimensionality Reduction

Dimensionality reduction has two components:

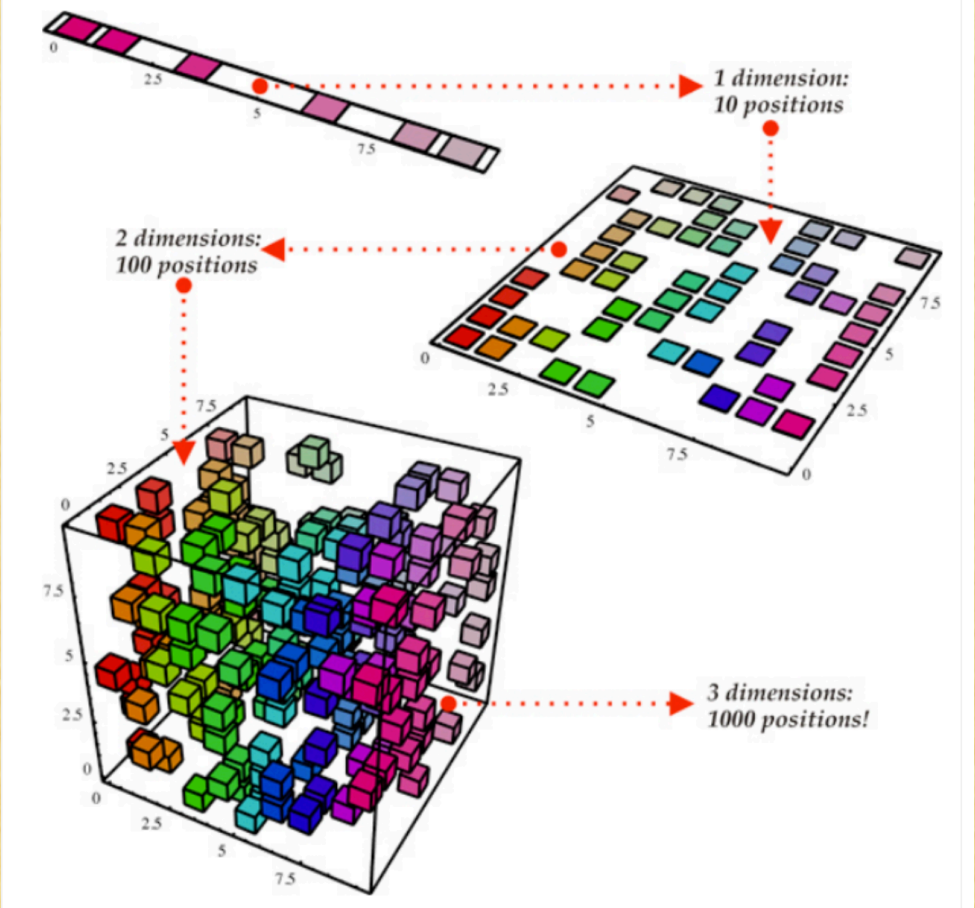
Feature selection: In this, we try to find a subset of the original set of variables, or features, to get a smaller subset which can be used to model the problem. This is hand selecting features which are highly discriminative. It requires an understanding of what aspects of the dataset are important in whatever predictions one is making, and which aren't.

Feature extraction: This reduces the data in a high dimensional space to a lower dimension space, i.e. a space with lesser no. of dimensions. It involves a transformation of the features, which often is not reversible because some information is lost in the process of dimensionality reduction.

an increase in the dimensionality of a data set results in exponentially more data being required to produce a representative sample of that data set. To combat the curse of dimensionality, numerous linear and non-linear dimensionality reduction techniques must be used. These aim to reduce the number of dimensions (variables) in the dataset through feature selection or feature extraction without significant loss of information.

Feature extraction is the process of transforming the original dataset into a dataset with fewer dimensions. Two well known feature extraction techniques are Principle Component Analysis (PCA) and Self-Organizing Maps (SOMs) .

Number of Features increase exponentially with dimension



Dimensionality reduction techniques are also used to reduce two undesired characteristics in data namely noise (variance) and redundancy (highly correlated variables).

Advantages and Disadvantages

Advantages of Dimensionality Reduction

It helps in data compression, and hence reduced storage space.

It reduces computation time.

It remove redundant features

By reducing the dimension of the data it allows visualization

Disadvantages of Dimensionality Reduction

It may lead to some amount of data loss.

PCA tends to find linear correlations between variables, which is sometimes undesirable.

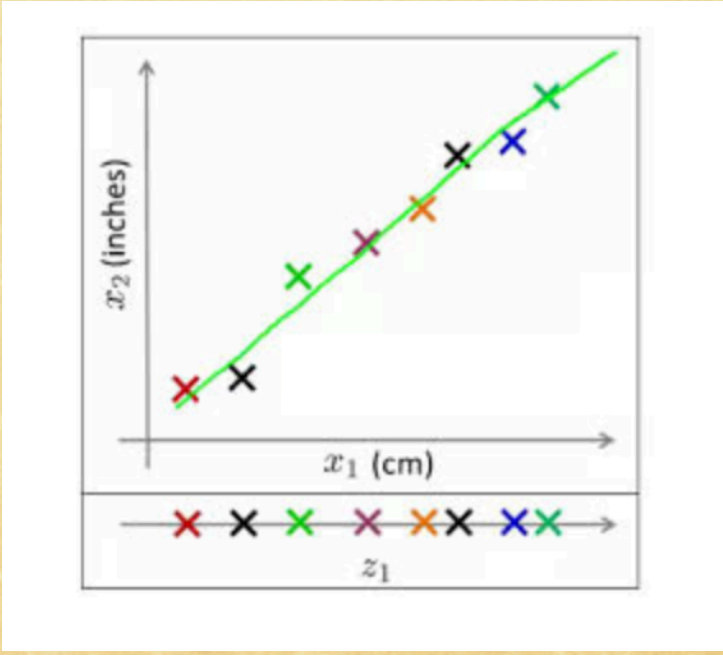
PCA fails in cases where mean and covariance are not enough to define datasets.

We may not know how many principal components to keep- in practice, some thumb rules are applied.

Dimension Reduction refers to the process of converting a set of data having vast dimensions into data with lesser dimensions ensuring that it conveys similar information concisely. These techniques are typically used while solving **machine learning problems** to obtain better features for a classification or regression task.

Dimensionality Reduction

The image shows 2 dimensions x_1 and x_2, which are measurements of several object in cm (x_1) and inches (x_2). Now, if you were to use both these dimensions in machine learning, they will convey similar information and introduce a lot of noise in system, so you are better of just using one dimension. Here we have converted the dimension of data from 2D (from x_1 and x_2) to 1D (z_1), which has made the data relatively easier to explain.



Dimension Reduction Techniques*

Low Variance: In case of high number of dimensions, to reduce dimensions, we should drop variables having low variance compared to others because these variables will not explain the variation in target variables.

High Correlation: Dimensions exhibiting higher correlation can lower the performance of model. It is redundant to have multiple variables with similar information or variation. One may identify variables with high correlation and select one of them.

Backward Feature Elimination: Here, we start with all n dimensions. Compute the sum of square of error (SSR) after eliminating each variable (n times). Then, identifying variables whose removal has produced the smallest increase in the SSR and removing it, leaving us with n-1 input features. Repeat this process until no further variables can be dropped.

* Taken from Analytics Vidhya

https://www.analyticsvidhya.com/blog/2015/07/dimension-reduction-methods/

Factor Analysis: In this case some variables are highly correlated. These variables can be grouped by their correlations – i.e. all variables in a particular group can be highly correlated among themselves but have low correlation with variables of other groups. Here each group represents a single underlying factor. These factors are small in number as compared to large number of dimensions.

Principle Component Analysis: In this technique, variables are transformed into a new set of variables, which are linear combination of original variables. These new set of variables are known as principle components. They are obtained in such a way that first principle component accounts for most of the possible variation of original data after which each succeeding component has the highest possible variance.

The second principal component must be orthogonal to the first principal component. In other words, it does its best to capture the variance in the data that is not captured by the first principal component. For two-dimensional dataset, there can be only two principal components. Below is a snapshot of the data and its first and second principal components. You can notice that second principle component is orthogonal to first principle component.

Self Organizing Maps: This is an unsupervised method where each data point in the data set recognizes itself by competing for representation. It starts with initializing the weight vectors. Then a sample vector is selected randomly and the map of weight vectors is searched to fid which weight best represents that sample. Each weight vector has neighboring weights that are close to it. SOM is a type of artificial neural network.

Principle Component Analysis

Definition

Principle Component Analysis (PCA) is a technique to study the internal structure of the data in a way that best explains the variance in the data. If a multivariate dataset is visualised as a set of coordinates in a high dimensional data space (1 axis per variable), PCA can supply the user with a lower-dimensional picture, a projection of this object when viewed from its most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced.

PCA can be thought of as fitting an n-dimensional ellipsoid to the data, where each axis of the ellipsoid represents a principal component. If some axis of the ellipsoid is small, then the variance along that axis is also small, and by omitting that axis and its corresponding principal component from our representation of the dataset, we lose only a small amount of information (Figure 1).

Principle Components

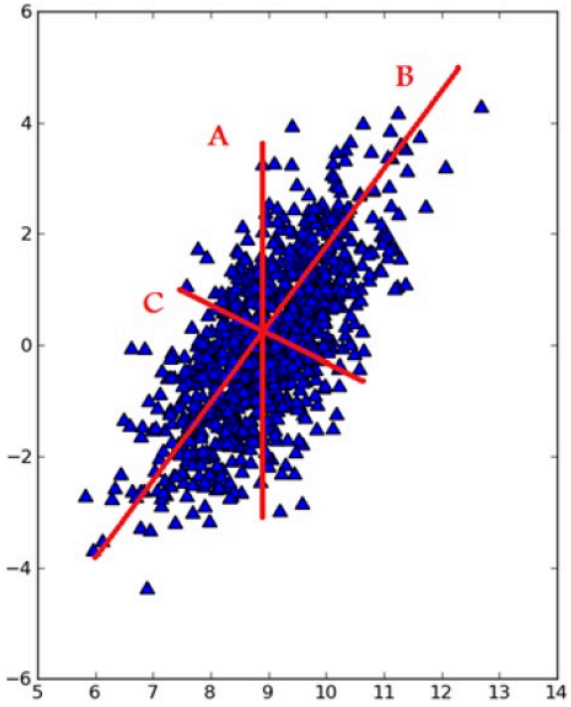


Figure 13.1 Three choices for lines that span the entire dataset. Line B is the longest and accounts for the most variability in the dataset.

To find the axes of the ellipsoid, we must first subtract the mean of each variable from the dataset to center the data around the origin. Then, we compute the covariance matrix of the data, and calculate the eigenvalues and corresponding eigenvectors of this covariance matrix. Then we must normalize each of the orthogonal eigenvectors to become unit vectors. Once this is done, each of the mutually orthogonal, unit eigenvectors can be interpreted as an axis of the ellipsoid fitted to the data. This choice of basis will transform our covariance matrix into a diagonalised form with the diagonal elements representing the variance of each axis. The proportion of the variance that each eigenvector represents can be calculated by dividing the eigenvalue corresponding to that eigenvector by the sum of all eigenvalues (Figure 2).

PCA is defined as an orthogonal linear transformation that transforms data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate (called first principle component), the second greatest variance on the second coordinate and so on. (Figure 1).

Source used for this study and further reading

Data Science: Concepts and Practice

By Vijay Kotu Bala Deshpande