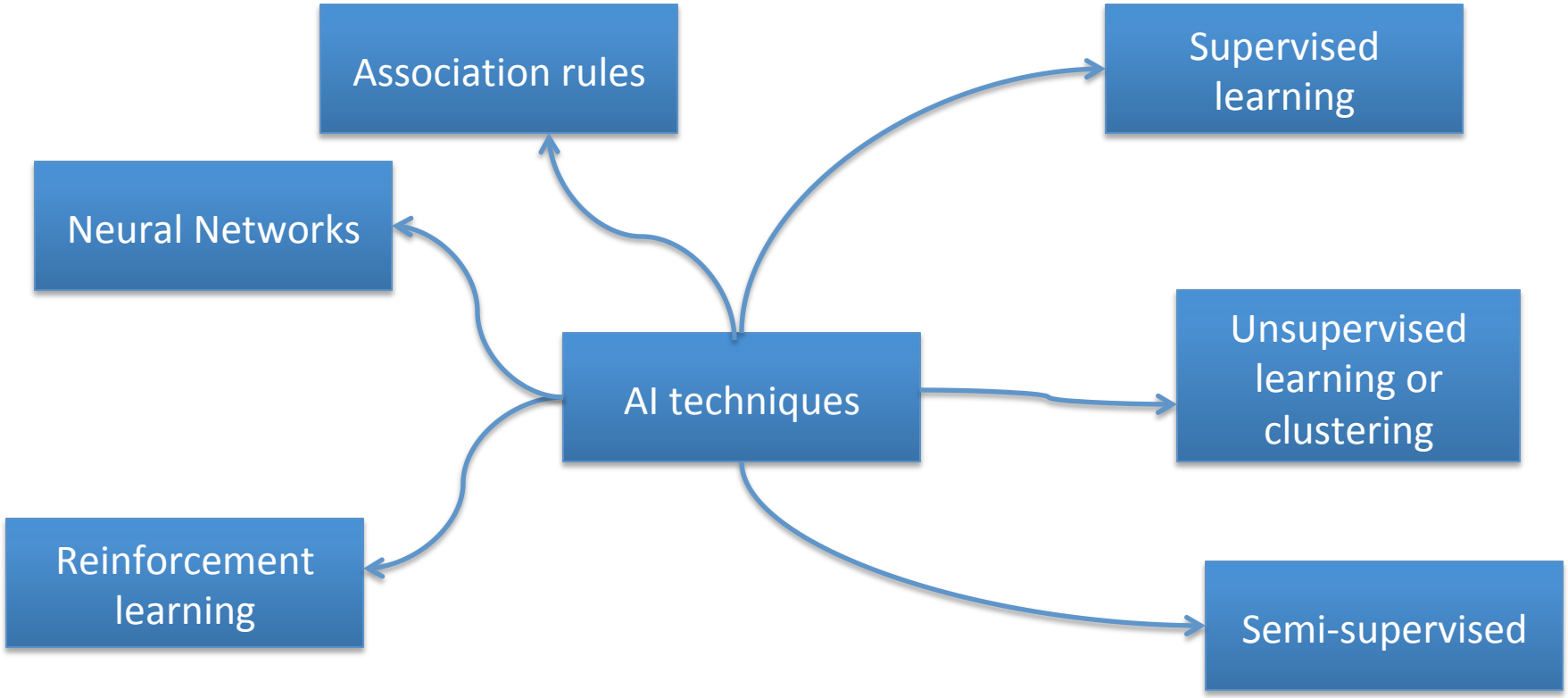
Clustering

What is cluster analysis?



Clustering

In data science, we often think about how to use data to make predictions on new data points. This is called “supervised learning.” Sometimes, however, rather than ‘making predictions’, we instead want to categorize data into buckets. This is termed “unsupervised learning.”

In case of unsupervised learning we don’t make predictions, we are merely categorizing the data into groups.

Clustering

To illustrate the difference, let's say we're at a major pizza chain and we've been tasked with creating a feature in the order management software that will predict delivery times for customers. In order to achieve this, we are given a dataset that has delivery times, distances traveled, day of week, time of day, staff on hand, and volume of sales for several deliveries in the past. From this data, we can make predictions on future delivery times. This is a good example of supervised learning.

Clustering

Now, let's say the pizza chain wants to send out targeted coupons to customers. It wants to segment its customers into 4 groups: large families, small families, singles, and college students. We are given prior ordering data (e.g. size of order, price, frequency, etc) and we're tasked with putting each customer into one of the four buckets. This would be an example of "unsupervised learning" since we're not making predictions; we're merely categorizing the customers into groups.

Example

A bank wants to give credit card offers to its customers. Currently, they look at the details of each customer and based on this information, decide which offer should be given to which customer.

Now, the bank can potentially have millions of customers. Does it make sense to look at the details of each customer separately and then make a decision? Certainly not! It is a manual process and will take a huge amount of time.

Introduction

So what can the bank do? One option is to segment its customers into different groups. For instance, the bank can group the customers based on their income:

The bank can now make three different strategies or offers, one for each group. Here, instead of creating different strategies for individual customers, they only have to make 3 strategies. This will reduce the effort as well as the time.

Introduction

The groups shown above are known as clusters and the process of creating these groups is known as clustering. Formally, we can say that:

Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.

It groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.

In clustering, don't have a target to predict. We try to club similar observations and form different groups. Hence it is an unsupervised learning problem.

What is data clustering

What is a cluster?

A cluster is …

- Comprised of a number of similar objects collected and grouped together
- A set of entities which are alike, and entities from different clusters are not alike.
- Clustering is an unsupervised machine learning task that automatically divides the data into groups or clusters of similar items. It does without having been told how the clusters should look. Clustering is used for knowledge discovery rather than prediction.

What is data clustering

- Without advance knowledge of what comprises a cluster, how can a computer possibly know where one group ends and another begins? The answer is simple. Clustering is guided by the principle that items inside a cluster should be very similar to each other, but very different from those outside. The definition of similarity might vary across applications, but the basic idea is always the same – group the data so that related elements are placed together.

What is data clustering

- Clustering is somewhat different from the classification, prediction, and pattern detection. In each of these cases, the result is a model that maps input to an outcome or features to other features; conceptually the model describes the existing patterns within data. In contrast, clustering creates new data. Unlabeled examples are given a cluster label that has been inferred entirely from the relationships within the data. For this reason, clustering task are referred to as unsupervised classification because, in a sense, it classifies unlabeled examples.

What is data clustering

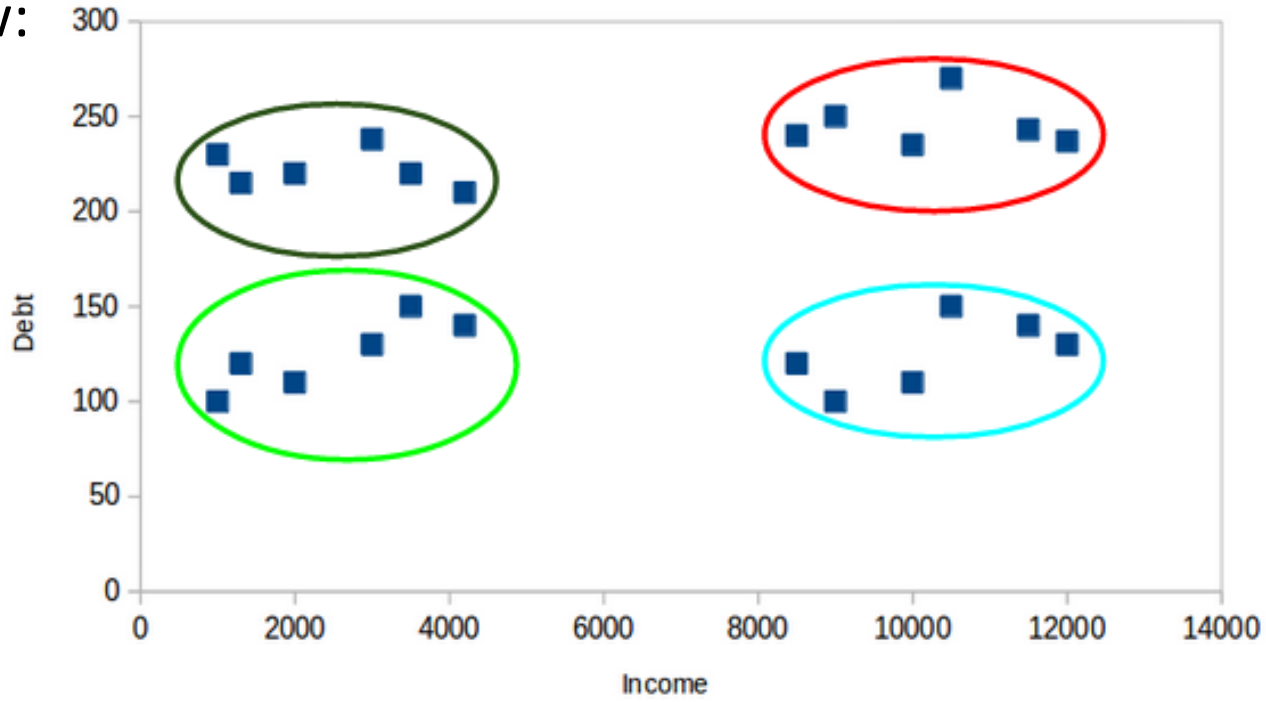
- The catch is that the class labels obtained from a unsupervised classifier are without intrinsic meaning. Clustering will tell you which groups of examples are closely related − but it' up to you to apply an actionable and meaningful label.

What is cluster analysis

- Clusters should exhibit high internal homogeneity and high external heterogeneity.
- What this means?
  - When plotted geometrically, objects within cluster should be very close together and cluster will be far apart.

What is cluster analysis

Let's say the bank only wants to use the income and debt to make segmentation. On the X-axis, we have the income of the customer and the y-axis represents the amount of debt. Here, we can clearly visualize that these customers can be segmented into 4 different clusters as shown below:



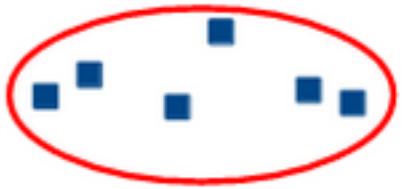
Source: analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/

What is cluster analysis

This is how clustering helps to create segments (clusters) from the data. The bank can further use these clusters to make strategies and offer discounts to its customers. So let's look at the properties of these clusters.

Property 1: All the data points in a cluster should be similar to each other. Let us illustrate it using the above example

What is cluster analysis

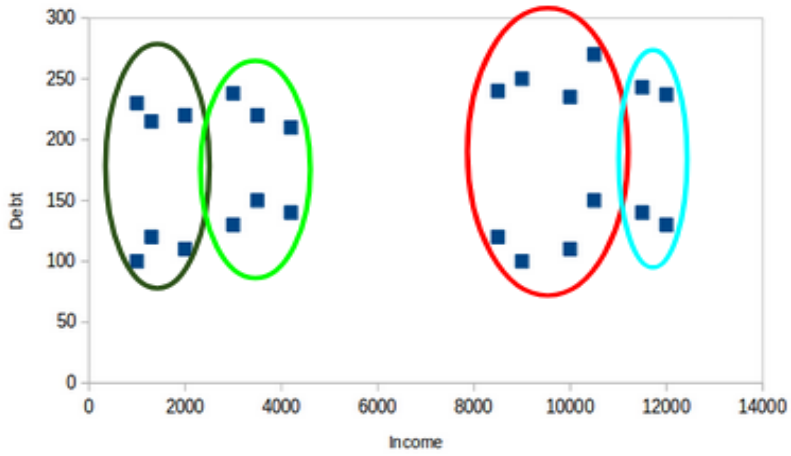


If the customers in a particular cluster are not similar to each other, then their requirements might vary, right? If the bank gives them the same offer, they might not like it and their interest in the bank might reduce. Not ideal.

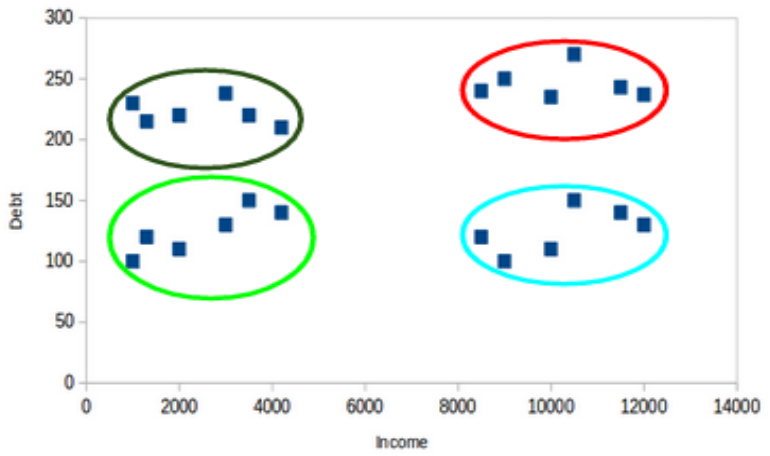
Having similar data points within the same cluster helps the bank to use targeted marketing. You can think of similar examples from your everyday life and think about how clustering will (or already does) impact the business strategy.

What is cluster analysis

Property 2: The data points from different clusters should be as different as possible. This will intuitively make sense if you grasped the above property. Let's again take the same example to understand this property:



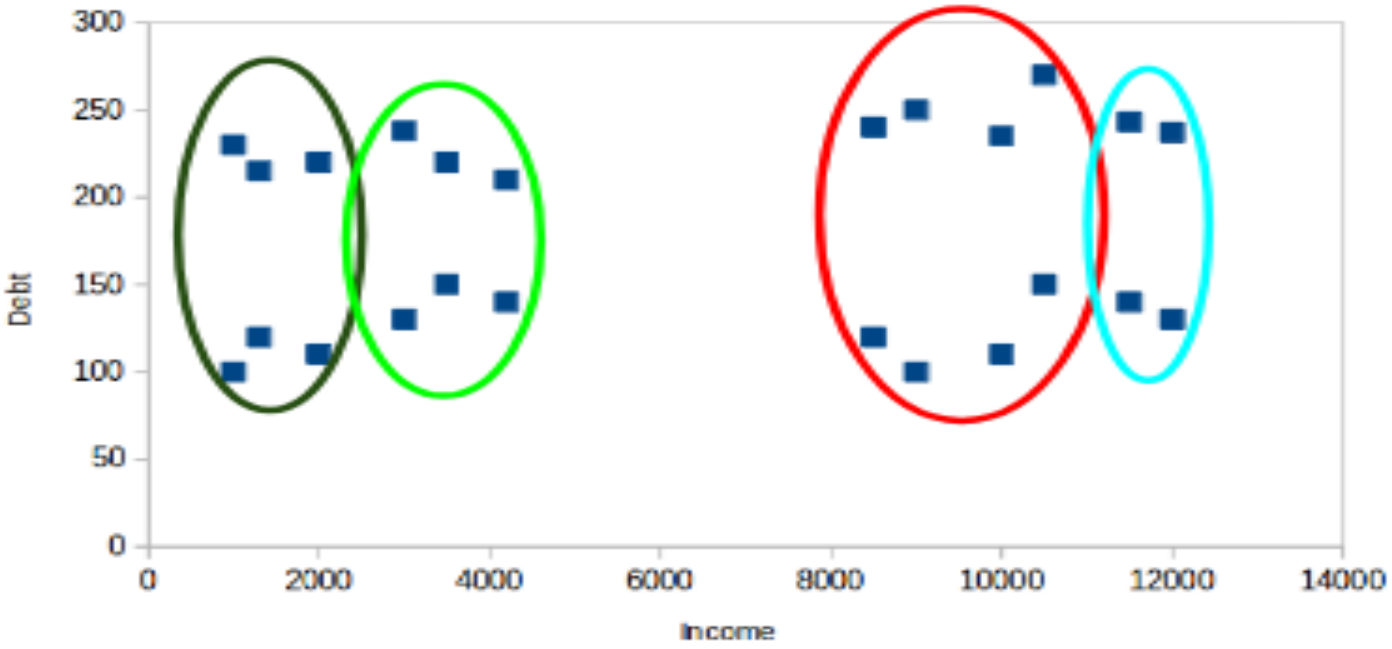
Case - I



Case - II

Which of these cases do you think will give us the better clusters? If you look at case I:

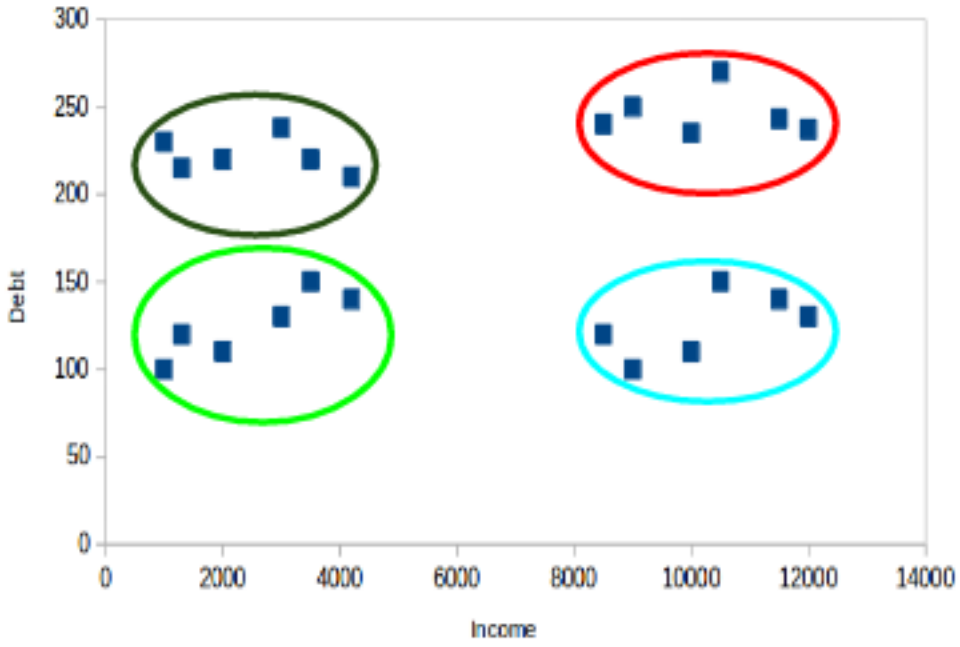
What is cluster analysis



Case - I

Customers in the red and blue clusters are quite similar to each other. The top four points in the red cluster share similar properties as that of the top two customers in the blue cluster. They have high income and high debt value. Here, we have clustered them differently. Whereas, if you look at case II:

What is cluster analysis



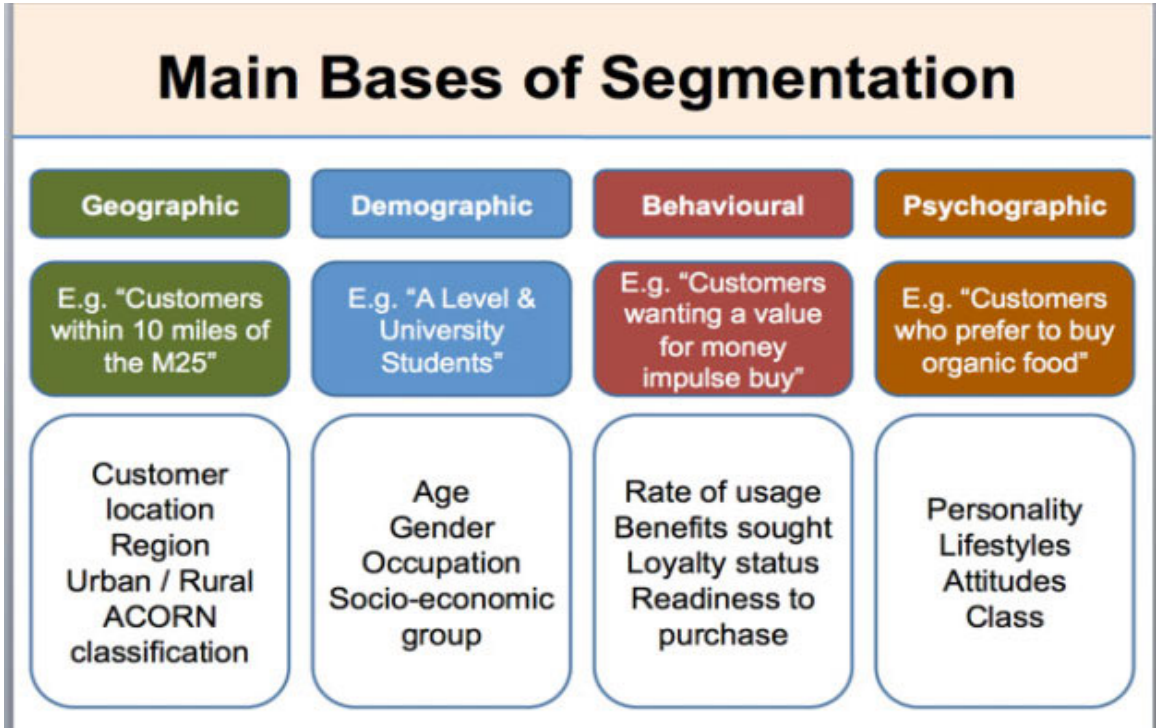
Case - II

Points in the red cluster are completely different from the customers in the blue cluster. All the customers in the red cluster have high income and high debt and customers in the blue cluster have high income and low debt value. Clearly we have a better clustering of customers in this case.

What is Clustering for? Applications

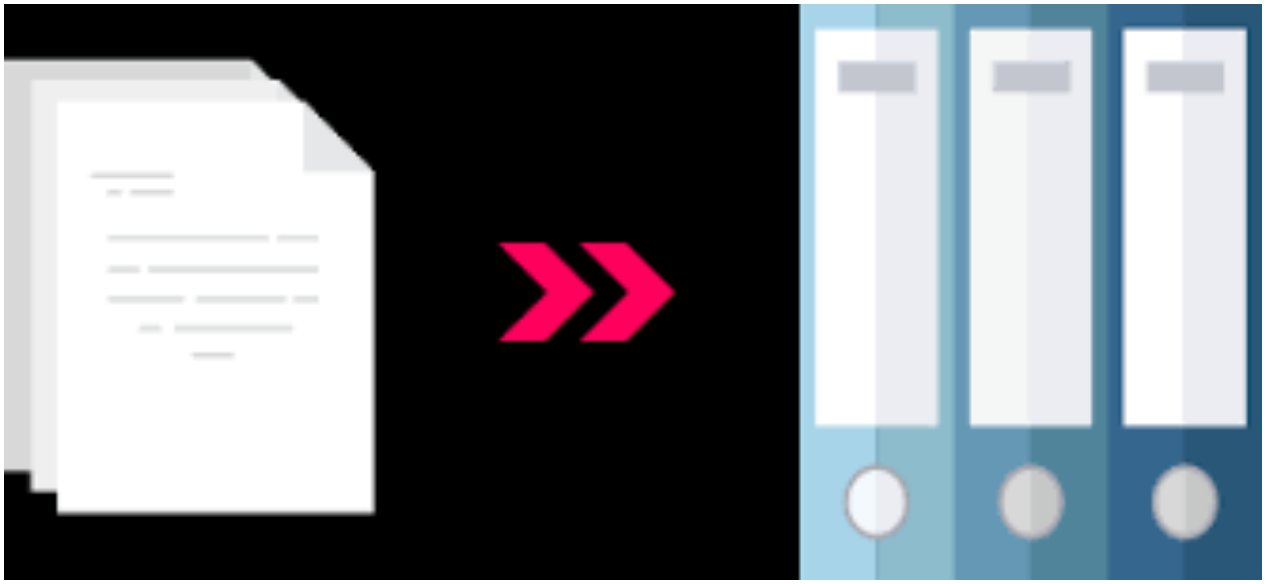
Clustering is widely used technique in the industry. It is actually being used in almost every domain, ranging from banking to recommendation engines, document clustering to image segmentation.

Customer segmentation: segment customers according to their similarities to do targeted marketing. This strategy is across functions, including telecom, e-commerce, sports, merketing etc.



What is Clustering for? Applications

Document Clustering: given a collection of text documents, we want to organize them according to their content similarities oo produce a topic hierarchy



What is Clustering for? Applications

Field of psychiatry: where the characterization of patients on the basis of clusters of symptoms can be useful in the identification of an appropriate form of therapy.

Biology: used to find groups of genes that have similar functions

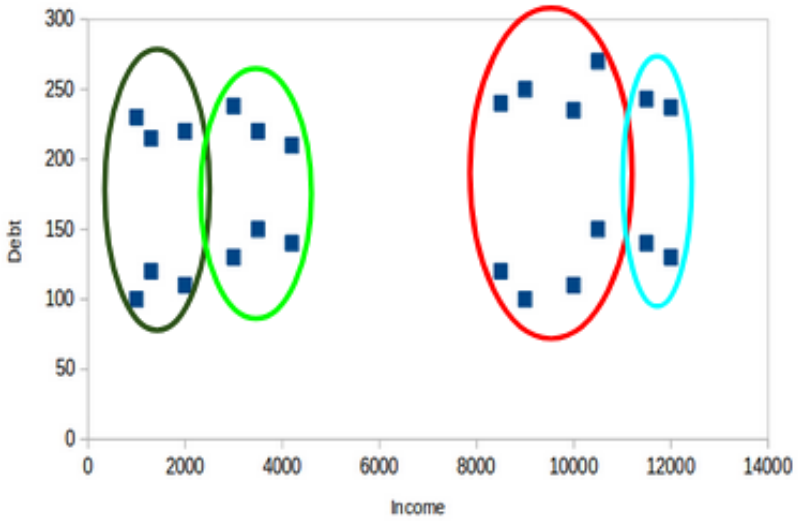
In fact, clustering is one of the most utilized data mining techniques.

It has a long history, and used in almost every field, e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc.

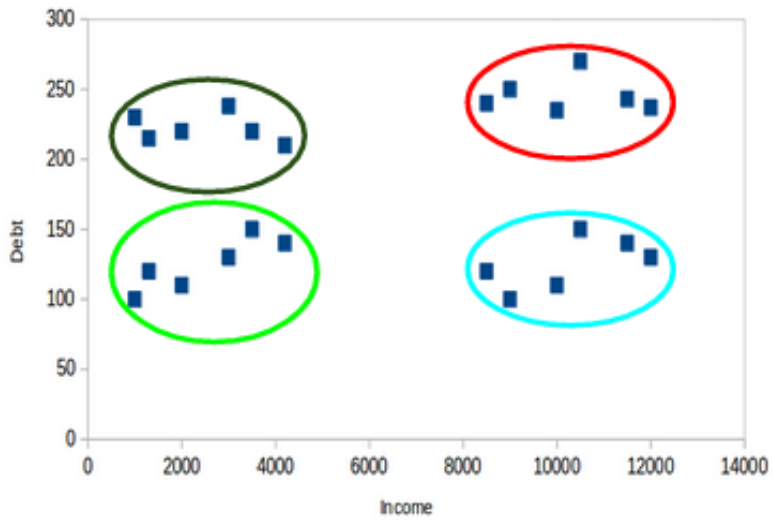
In recent years, due to the rapid increase of online documents, text clustering becomes important.

Different Evaluation Metrics for Clustering

The primary aim of clustering is not just to make clusters, but to make good and meaningful ones. We saw this in the below example:



Case - I



Case - II

multiple clusters

Here, we used only two features and hence it was easy for us to visualize and decide which of these clusters is better.

What is Clustering for? Applications

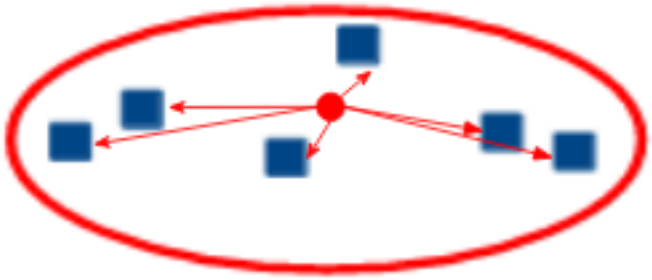
Unfortunately, that's not how real-world scenarios work. We will have a ton of features to work with. Let's take the customer segmentation example again – we will have features like customer's income, occupation, gender, age, and many more. Visualizing all these features together and deciding better and meaningful clusters would not be possible for us.

This is where we can make use of evaluation metrics. Let's discuss a few of them and understand how we can use them to evaluate the quality of our clusters.

Inertia

Recall the first property of clusters we covered above. This is what inertia evaluates. It tells us how far the points within a cluster are. So, inertia actually calculates the sum of all the points within a cluster from the centroid of that cluster.

We calculate this for all the clusters and the final inertial value is the sum of all these distances. This distance within the clusters is known as intracluster distance. So, inertia gives us the sum of intracluster distances:



Intra cluster distance

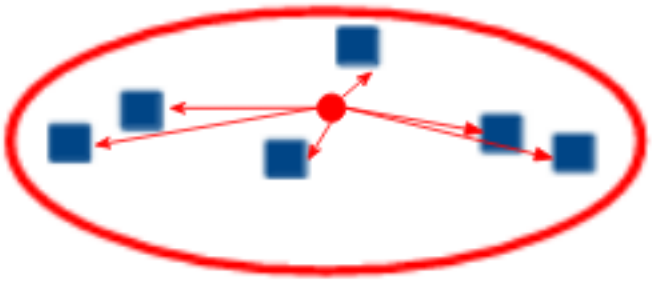
Now, what do you think should be the value of inertia for a good cluster? Is a small inertial value good or do we need a larger value? We want the points within the same cluster to be similar to each other, right? Hence, the distance between them should be as low as possible.

Dunn Index

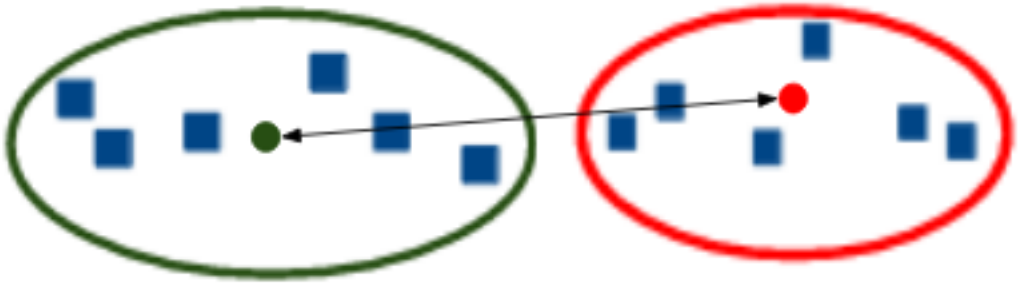
We now know that inertia tries to minimize the intracluster distance. It is trying to make more compact clusters.

Let me put it this way – if the distance between the centroid of a cluster and the points in that cluster is small, it means that the points are closer to each other. So, inertia makes sure that the first property of clusters is satisfied. But it does not care about the second property – that different clusters should be as different from each other as possible.

This is where Dun Index can come into rescue



Intra cluster distance



Inter cluster distance

Along with the distance between the centroid and points, the Dunn index also takes into account the distance between two clusters. This distance between the centroids of two different clusters is known as inter-cluster distance. Let's look at the formula of the Dunn index:

Dunn index = min(Inter cluster distance)/max(Intra cluster distance)

We want to maximize the Dunn index. The more the value of the Dunn index, the better will be the clusters.

Aspects of Clustering

A clustering algorithm
- Partitional clustering
- Hierarchical clustering
- …

A distance (similarity, or dissimilarity) function

Clustering quality
- Inter-clusters distance ⇒ maximized
- Intra-clusters distance ⇒ minimized

The quality of a clustering result depends on the algorithm, the distance function, and the application.

Common Roles Cluster Analysis can play

Data Reduction

The goal of clustering is to reduce the amount of data by categorization or grouping similar data items together. A researcher may be faced with a large amount of data that can be meaningless unless classified into manageable group. Clustering can perform this data reduction procedure objectively by reducing the information from an entire population to information about specific group.

Such grouping is pervasive in the way humans process information, and one of the motivations for using clustering algorithms is to provide automated tools to help in constructing categories.

Common Roles Cluster Analysis can play

Hypothesis Generation

Clustering analysis is also useful when a researcher wishes to develop hypothesis concerning the nature of the data or to examine previously stated hypotheses.

Characteristics of clustering technique

What makes a clustering algorithm efficient and effective? The answer is not clear. A specific method can perform well on one data set, but very poorly on another, depending on the size and dimensionality of the data as well as the objective function and structures used.

Regardless of the method, researchers agree that characteristics of a good clustering technique are:

Scalability: The ability of the algorithm to perform well with large number of data objects

Analyze mixture of attribute types: The ability to analyze single as well as mixtures of attributes types

Characteristics of clustering technique

Find arbitrary-shaped clusters: The shape usually corresponds to the kinds of clusters an algorithm can find and we should consider this as a very important thing when choosing a method, since we want to be as general as possible. Different types of algorithms will be biased towards finding different types of cluster structures/shapes and it is not always an easy task to determine the shape or the corresponding bias. Especially when categorical attributes are present we may not be able to talk about cluster structures.

Characteristics of clustering technique

Minimum requirements for input parameters: Many clustering algorithms require some user-defined parameters, such as the number of clusters, in order to analyze the data. However, with large data sets and higher dimensionalities, it is desirable that a method require only limited guidance from the user, in order to avoid bias over the result.

Handling the noise: Clustering algorithms should be able to handle deviations, in order to improve cluster quality. Deviations are defined as data objects that depart from generally accepted norms of behavior and are also referred to as outliers.

Characteristics of clustering technique

Sensitivity to the order of input records: The same data set, when presented to certain algorithms in different orders, may produce dramatically different results. The order of input mostly affects algorithms that require a single scan over the data set, leading to locally optimal solutions at every step. Thus, it is crucial that algorithms be insensitive to the order of input.

High dimensionality of data: The number of attributes/dimensions in data sets is large, and many clustering algorithms can not handle more than a small number (8-10) of dimensions. It is a challenge to cluster high dimensional data set such as US census data set which contains 138 attributes

Characteristics of clustering technique

The appearance of large number of attributes is often termed as the **curse of dimensionality**. This has to do with the following:

1 As the number of attributes become larger, the amount of resources required to store or represent them grows

2. The distance of a given point from the nearest and furthest neighbor is almost the same, for a wide variety of distributions and distance functions.

Both of the above highly influence the efficiency of a clustering algorithm, since it would need more time to process the data, while at the same time the resulting clusters would be of very poor quality.

Characteristics of clustering technique

Interpretability and usability: Most of the times, it is expected that the clustering algorithms produce usable and interpretable results. But when it comes to comparing the results with preconceived ideas or constraints, some techniques fail be be satisfactory. Therefore, easy to understand results are highly desirable.

How does Cluster Analysis work?

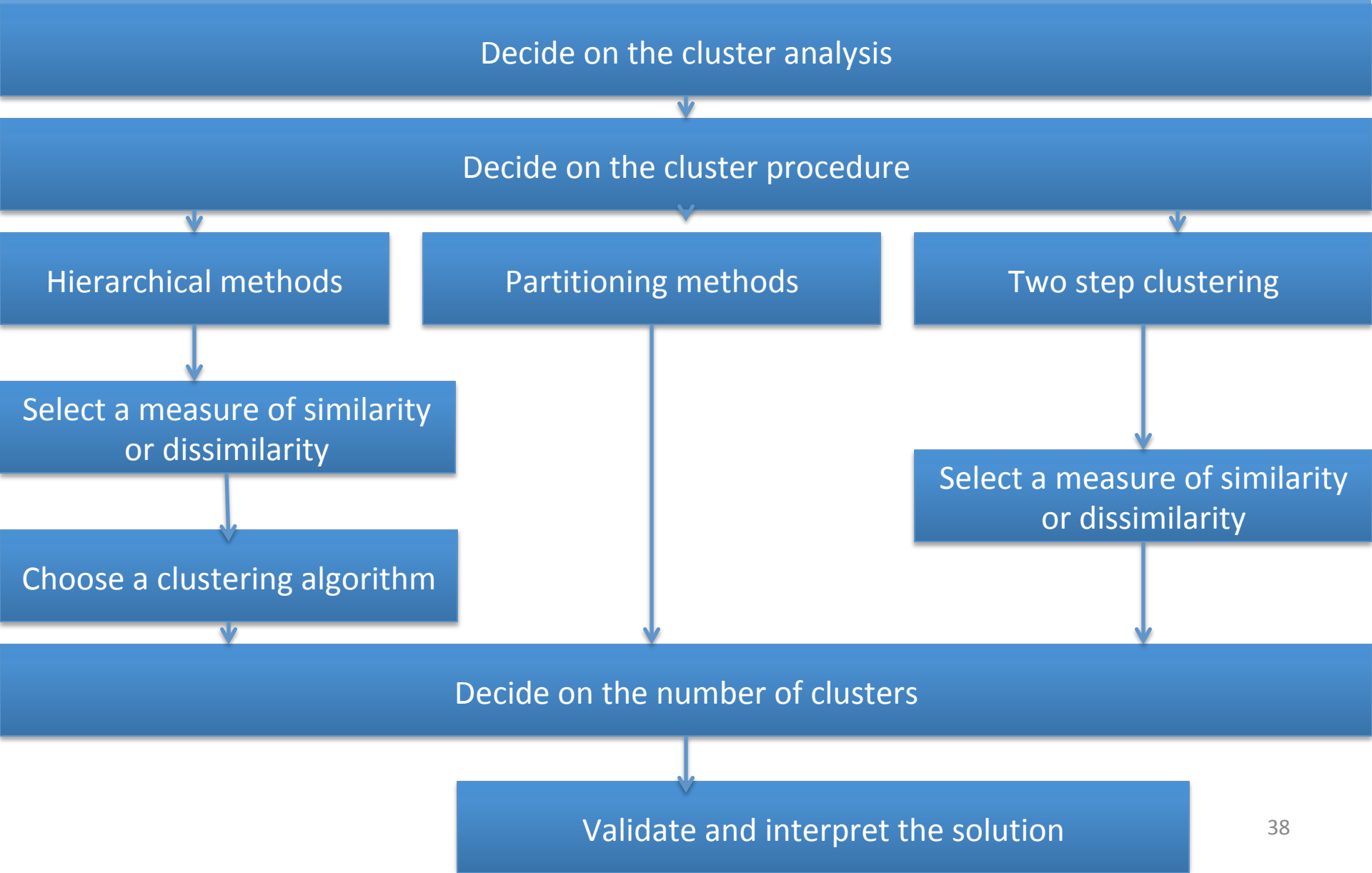
The primary objective of cluster analysis is to define the structure of the data by placing the most similar observations into groups. To accomplish this task, we must address three basic questions:

How do we measure similarity?

How do we form clusters?

How many clusters/groups do we form?

Step in a cluster analysis



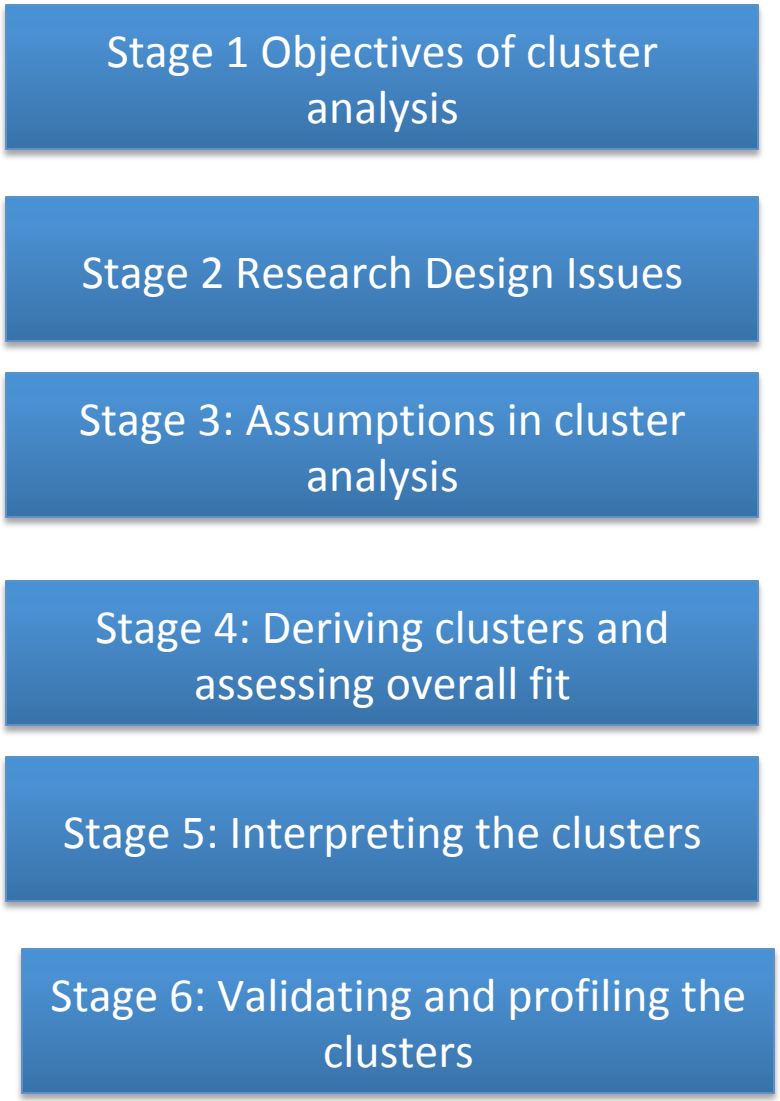
Types of clustering algorithms

The clustering methods can be divided into two basic types: **hierarchical** and **partitional clustering**. Within each of the types there exists a wealth of subtypes and different algorithms for finding the clusters.

Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones, or by simply splitting larger clusters. The clustering methods differ in the rule by which it is decided which two small clusters are merged or which large cluster is split. The end result of the algorithm is a tree of clusters called a **dendrogarm,** which shows how the clusters are related. By cutting the **dendrogram** at a desired level a clustering of the data items into disjoint groups is obtained

Partitional clustering, on the other hand, attempts to directly decompose the data set into a set of disjoint clusters. The criterion function that the clustering algorithm tries to minimize may emphasize the local structure of the data, as by assigning clusters to peaks in the probability density function, or the global structure. Typically the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters

Cluster Analysis Diagram



Objectives

Select objectives:

Taxonomy description: for exploratory purposes and the formation of a taxonomy

Data simplification – a researcher could face a large number of observations that are meaningless unless classified into manageable groups

Hypothesis testing or generation – a researcher wishes to develop hypothesis concerning the nature of the data or to examine previously stated hypothesis

Relationship identification – a researcher wishes to reveal relationships among observations that are not possible with individual observations

Research Design Issues

Five questions to be asked before starting:

1. What variables are relevant
2. Is the sample size adequate
3. Can outliers be detected and if so should they be removed
4. How should object similarity be measured
5. Should data be standardized?

Research Design Issues

1. Is the sample size adequate?

- The sample size must be large enough to provide sufficient representation of small groups within the population and represent the underlying structure

- A researcher should ensure the sample size is sufficiently large enough to adequately represent all relevant groups

- Specify the group sizes necessary for relevance for the questions being asked

Remarks: 1 Interest is focus on the identification of small groups – large sample size
2 Interest is focus on the identification of large groups – small sample size

Research Design Issues

Can outliers be detected and if so should they be removed?

What outliers can be?

Truly aberrant observation not representative for the population

distort the actual structure and result in unrepresentative clusters – should be removed

Representative observations of small or insignificant groups

should be removed so that the resulting clusters represent more accurately relevant groups

An under sampling of the actual group in the population that causes poor representation of the group

They represent valid and relevant groups – should be included in the clustering solution.

Research Design Issues

How should object similarity be measured?

Three ways to measure inter-objects similarities

- Correlation measures
- Distance measures
- Associations measures

Distance functions

There are numerous distance functions for

- Different types of data
  - Numeric data
  - Nominal data
- Different specific applications

Distance functions for numeric attributes

- Most commonly used functions are
  - Euclidean distance and
  - Manhattan (city block) distance
- We denote distance with: $dist(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are data points (vectors)
- They are special cases of Minkowski distance. h is positive integer.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = ((x_{i1} - x_{j1})^h + (x_{i2} - x_{j2})^h + \ldots + (x_{ir} - x_{jr})^h)^{\frac{1}{h}}$$

Euclidean distance and Manhattan distance

- If $h = 2$, it is the Euclidean distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2}$$

- If $h = 1$, it is the Manhattan distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ir} - x_{jr}|$$

- Weighted Euclidean distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \ldots + w_r(x_{ir} - x_{jr})^2}$$

Squared distance and Chebychev distance

- Squared Euclidean distance: to place progressively greater weight on data points that are further apart.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2$$

- Chebychev distance: one wants to define two data points as "different" if they are different on any one of the attributes.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max(|x_{i1} - x_{j1}|, |x_{i2} - x_{j2}|, \ldots, |x_{ir} - x_{jr}|)$$

Distance functions for binary and nominal attributes

- Binary attribute: has two values or states but no ordering relationships, e.g.,
  - Gender: male and female.
- We use a confusion matrix to introduce the distance functions/measures.
- Let the *i*th and *j*th data points be $\mathbf{x}_i$ and $\mathbf{x}_j$ (vectors)

Confusion matrix

Data point i \ Data point j	1	0	
1	a	b	$a+b$
0	c	d	$c+d$
	$a+c$	$b+d$	$a+b+c+d$

(10)

a: the number of attributes with the value of 1 for both data points.

b: the number of attributes for which $x_{if} = 1$ and $x_{jf} = 0$, where x_{if} (x_{jf}) is the value of the fth attribute of the data point $\mathbf{x}_i$ ($\mathbf{x}_j$).

c: the number of attributes for which $x_{if} = 0$ and $x_{jf} = 1$.

d: the number of attributes with the value of 0 for both data points.

Symmetric binary attributes

- A binary attribute is **symmetric** if both of its states (0 and 1) have equal importance, and carry the same weights, e.g., male and female of the attribute Gender
- Distance function: Simple Matching Coefficient, proportion of mismatches of their values

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{b + c}{a + b + c + d}$$

Symmetric binary attributes: example

$\mathbf{x}_1$	1	1	1	0	1	0	0
$\mathbf{x}_2$	0	1	1	0	0	1	0

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{2+1}{2+2+1+2} = \frac{3}{7} = 0.429$$

Asymmetric binary attributes:

- Asymmetric: if one of the states is more important or more valuable than the other.
  - By convention, state 1 represents the more important state, which is typically the rare or infrequent state.
  - Jaccard coefficient is a popular measure

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{b + c}{a + b + c}$$

  - We can have some variations, adding weights

Nominal attributes

- Nominal attributes: with more than two states or values.
  - the commonly used distance measure is also based on the simple matching method.
  - Given two data points $\mathbf{x}_i$ and $\mathbf{x}_j$, let the number of attributes be *r*, and the number of values that match in $\mathbf{x}_i$ and $\mathbf{x}_j$ be *q*.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{r - q}{r}$$

Distance function for text documents

- A text document consists of a sequence of sentences and each sentence consists of a sequence of words.
- To simplify: a document is usually considered a "bag" of words in document clustering.
  - Sequence and position of words are ignored.
- A document is represented with a vector just like a normal data point.
- It is common to use similarity to compare two documents rather than distance.
  - The most commonly used similarity function is the **cosine similarity**.

Research Design Issues

Should data be standardized

Distance measures used to estimate inter-objects similarities are sensitive to different scales or magnitudes among the variables

In general, variable with a larger dispersion (standard deviation) will have a bigger impact on the clustering

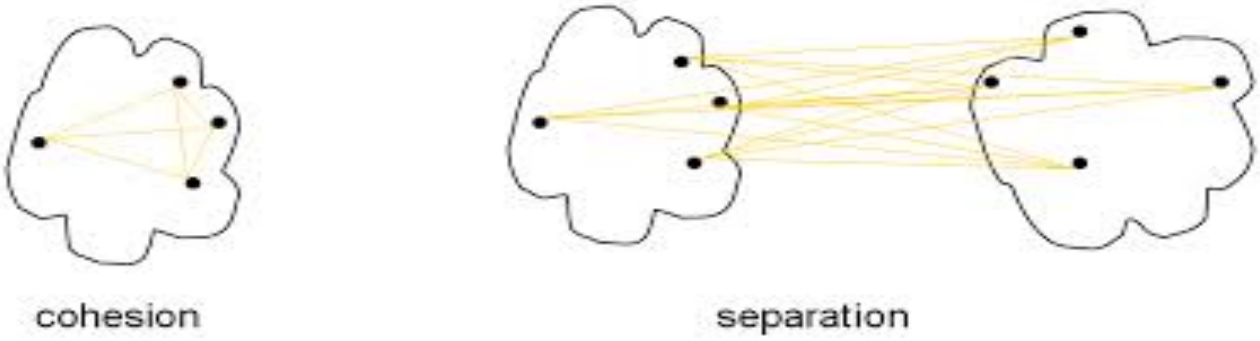
Clustering variables that are not all of the same scale should be standardized

Internal indices – example: silhouette coefficient

$$sc = 1-c/s$$

c – cluster cohesion is the mean value of the distances of all pairs of points within cluster. Smaller is it better.

S – cluster separation is the mean value of the distances between the points in the cluster and points outside the cluster. Bigger it is better it is.



K-means clustering

- K means clustering algorithm was developed by J. MacQueen (1967) and then by J.A. Hartigan and M.A. Wong around 1975.
- Simply speaking k-mean clustering is an algorithm to classify or to group your objects based on attributes/features into K number of group.
- The first property of clusters states that the points within a clustre should be similar to each other. So our aim is to minimize the distance between the points within a cluster.
- There is an algorithm that tries to minimize the distance of the points in a cluster with their centroid − k-mean clustering technique

- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Each cluster has a cluster **center**, called **centroid**.

K-means clustering

K-means is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid.

The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.

K is the number of clusters and is specified by the user.

K-means clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K must be specified
- The basic algorithm is very simple

Algorithm

Given *k*, the *k-means* algorithm works as follows:

1)Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers

2)Assign each data point to the closest centroid

3)Re-compute the centroids using the current cluster memberships.

4)If a convergence criterion is not met, go to 2).

K-means clustering

- Initial centroids are often chosen randomly
  - Clusters produced vary from one run to another
- The centroid is (typically) the mean of the points in the cluster

- Closeness is measured by Euclidean distance, cosine similarity, correlation etc.

- K-means will converge for common similarity measures mentioned above

Stopping Criteria for K-means clustering

Most of the convergence happens in the first few iterations. Essentially, there are essentially three stopping criteria that can be adopted to stop the K-means algorithm:

- Centroids of newly formed clusters do not change
- Points remain in the same cluster
- Maximum number of iterations are reached

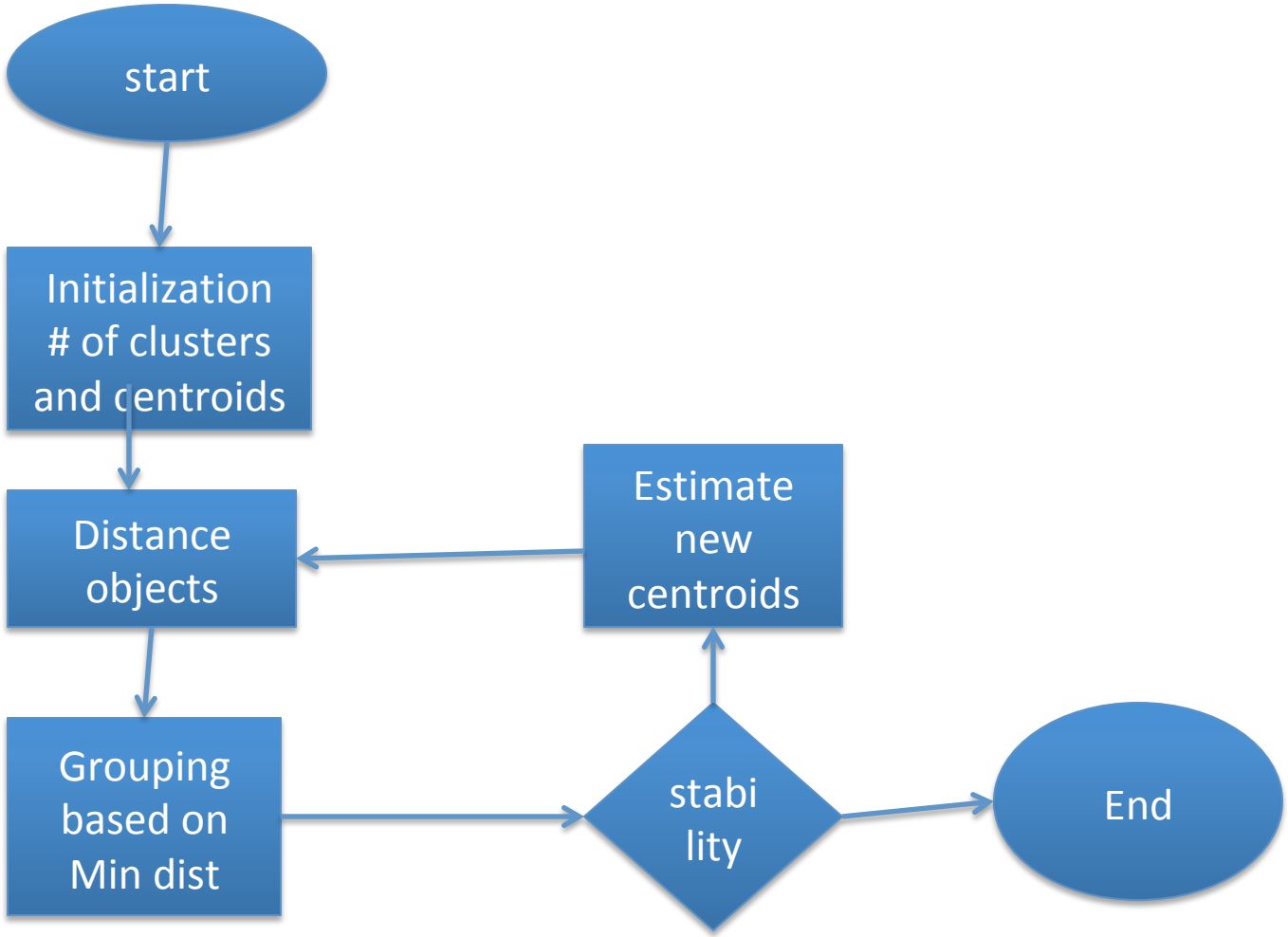
We can stop the algorithm if the centroids of newly formed clusters are not changing. Even after multiple iterations, if we are getting the same centroids for all the clusters, we can say that the algorithm is not learning any new pattern and it is a sign to stop the training.

Another clear sign that we should stop the training process if the points remain in the same cluster even after training the algorithm for multiple iterations.

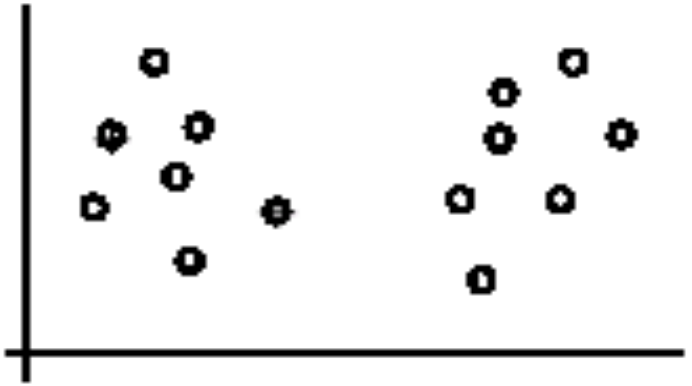
Stopping Criteria for K-means clustering

Finally, we can stop the training if the maximum number of iterations is reached. Suppose if we have set the number of iterations as 100. The process will repeat for 100 iterations before stopping.

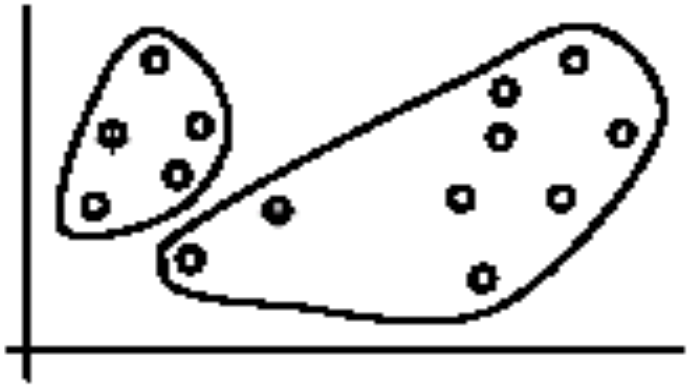
K-means flow


start
Initialization
# of clusters
and centroids
Distance
objects
Grouping
based on
Min dist
stabi
lity
Estimate
new
centroids
End

Example



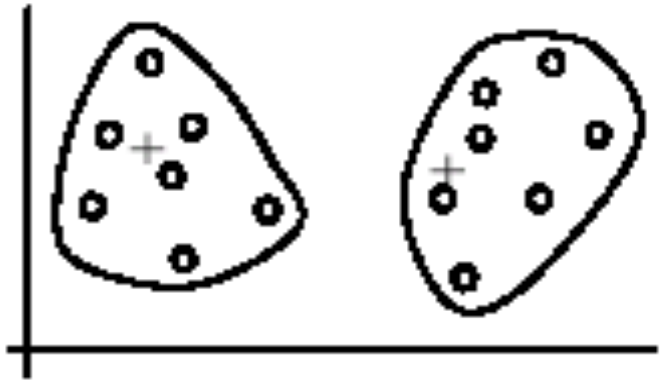
(A). Random selection of k centers



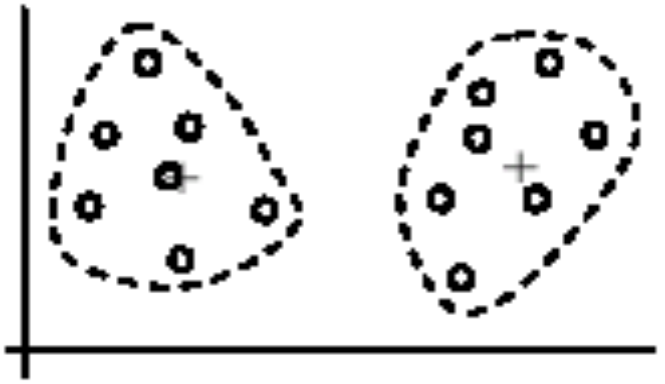
Iteration 1: (B). Cluster assignment

(C). Re-compute centroids

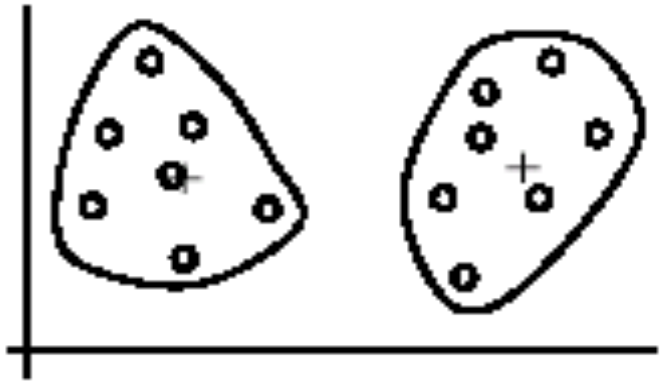
Example



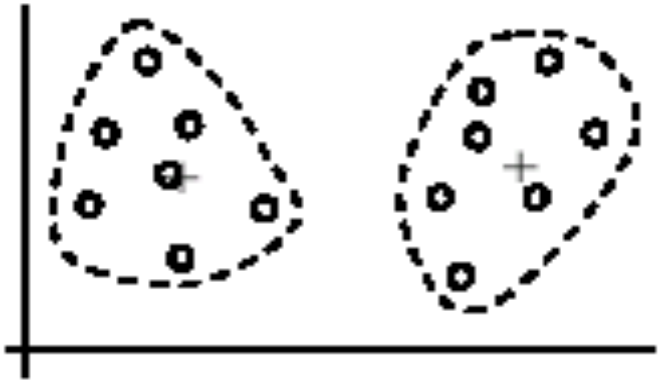
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

Strength of K-means

Strengths:

- Simple: easy to understand and to implement
- Efficient: Time complexity: $O(tkn)$,
  where n is the number of data points,
  k is the number of clusters, and
  t is the number of iterations.
  Since both k and t are small. k-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.

Note that: it terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.

Weaknesses of K-means

- The algorithm is only applicable if the mean is defined.
  For categorical data, *k*-mode - the centroid is represented by most frequent values.
- We never know the real cluster, using the same data, if it is inputted in a different order may produce different cluster if the number of data is a few
- The user needs to specify *k*.
- We never know which attribute contributes more to the grouping process since we assume that each attribute has the same weight.
- The algorithm is sensitive to **outliers.** Very far data from the centroid may pull the centroid away from the real one. To overcome outliers problem, we can use median instead of mean.
- Sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the local optimum.

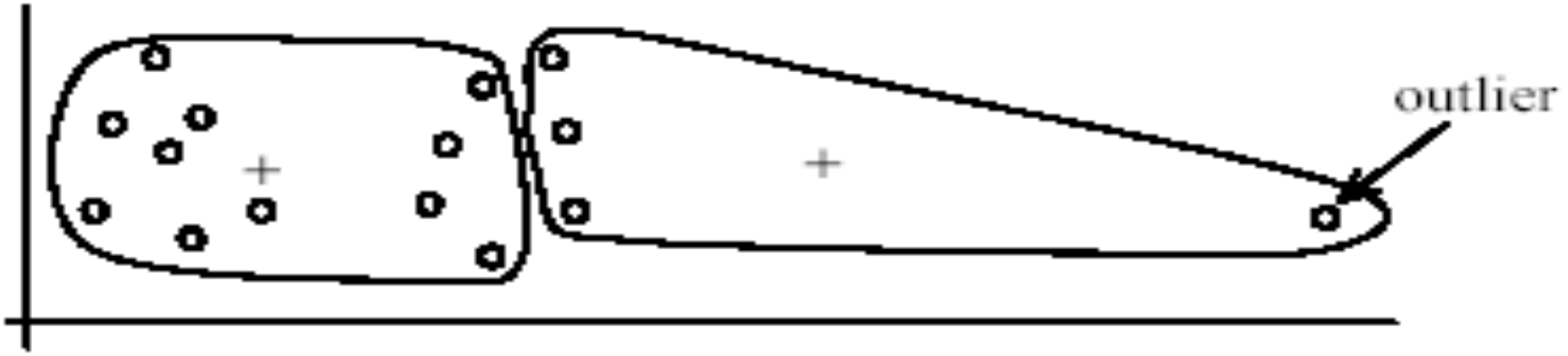
Pre-processing and Post processing

- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post- processing
  - Eliminate small clusters that may represent outliers
  - Split “loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE

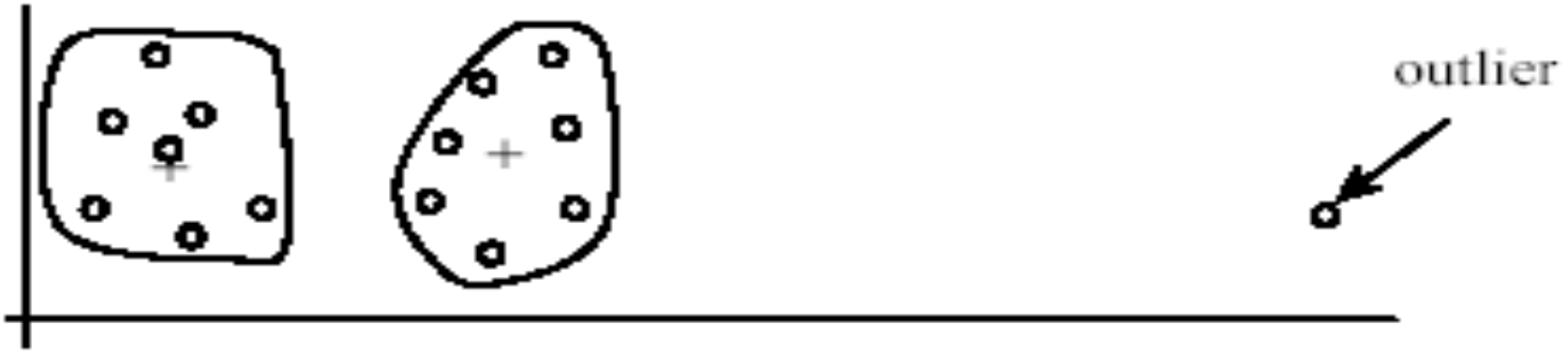
Initial Centroids Problem

- If there are K real clusters then the chance of selecting one centroid for each cluster is small. The following steps are taken to avoid initial centroid problem
  - Multiple runs
    - Helps, but probability is not on your side
  - Sample and use hierarchical clustering to determine initial centroids
  - Select more than K initial centroids and then select among these initial centroids
    - Select most widely separated

Weaknesses of K-means: Problems with outliers



(A): Undesirable clusters



(B): Ideal clusters

Weaknesses of K-means: To deal with outliers

- One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.
- To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Another method is to perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.
- Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

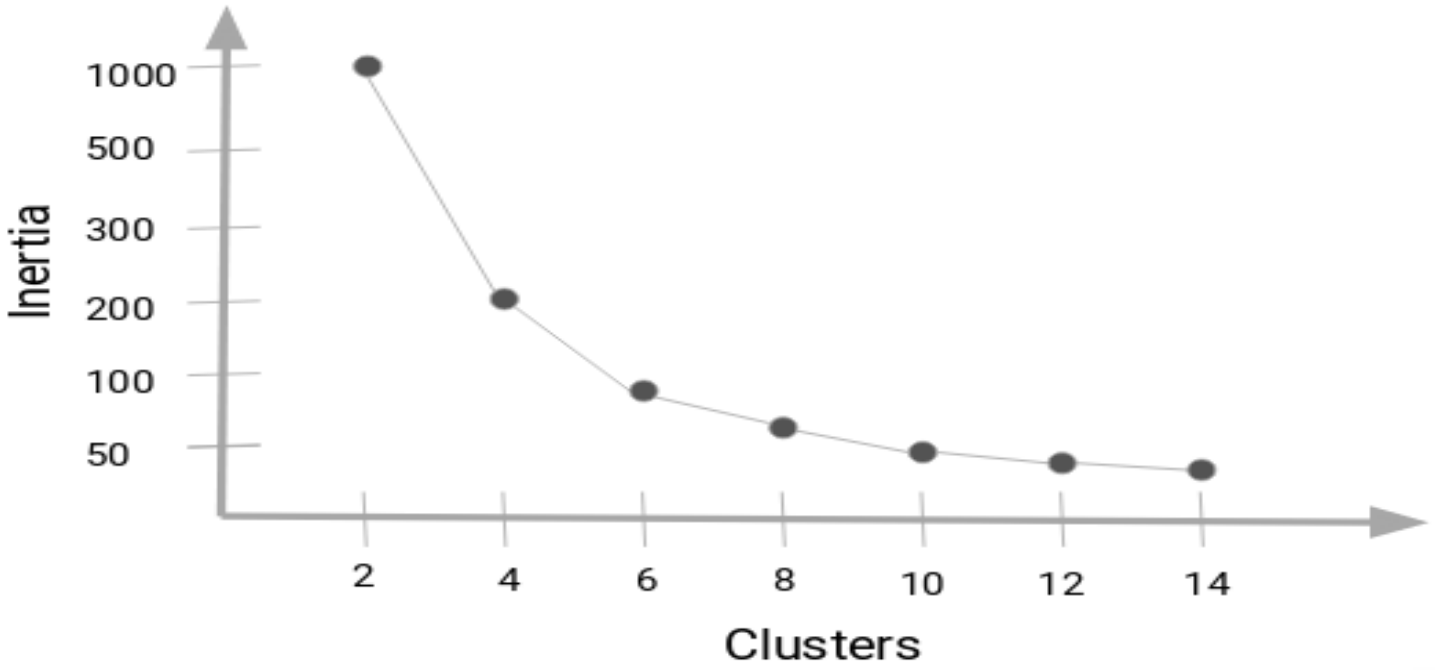
Note that we may end with different clusters. This is the result of the random initialization trap. Essentially, our starting centroids can dictate the location of our clusters in k-mean clustering.

This isn't the result we wanted, but one way to combat this is with the k-means ++ algorithm, which provides better initial seeding in order to find the best clusters. Fortunately, this is automatically done in k-means implementation we'll be using in Python.

How to choose the right number of clusters

One of the most common doubts everyone has while wokring with k-means is selecting the right number of clusters

One thing we can do is plot a graph, also known as an elbow curve, where the x-axis will represent the number of clusters and the y-axis will be an evaluation metric. Let's say inertia for now.



How to choose the right number of clusters

When we changed the cluster value from 2 to 4, the inertia value reduced very sharply. This decrease in the inertia value reduces and eventually becomes constant as we increase the number of clusters further.

So, the cluster value where this decrease in inertia value becomes constant can be chosen as the right cluster value for our data.

Here, we can choose any number of clusters between 6 and 10. We can have 7, 8, or even 9 clusters. You must also look at the computation cost while deciding the number of clusters. If we increase the number of clusters, the computation cost will also increase. So, if you do not have high computational resources, my advice is to choose a lesser number of clusters.

Summary K-means:

- Despite weaknesses, *k*-means is still the most popular algorithm due to its simplicity, efficiency and other clustering algorithms have their own lists of weaknesses.
- No clear evidence that any other clustering algorithm performs better in general although they may be more suitable for some specific types of data or applications.
- Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

Hierarchical Clustering

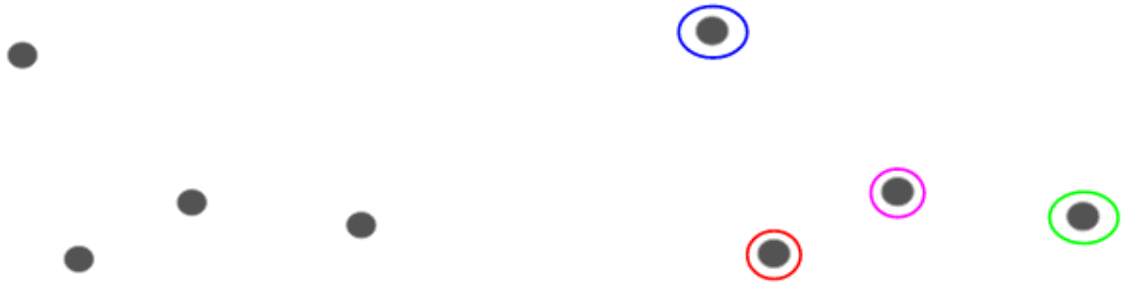
K-means is an iterative process. It will keep on running until the centroids of newly formed clusters do not change or the maximum number of iterations are reached.

But there are certain challenges with K-means. It always tries to make clusters of the same size. Also, we have to decide the number of clusters at the beginning of the algorithm. Ideally, we would not know how many clusters should we have, in the beginning of the algorithm and hence it a challenge with K-means.

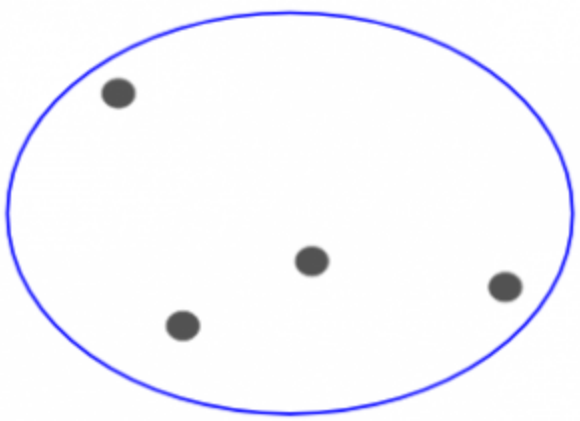
This is a gap hierarchical clustering bridges. It takes away the problem of having to pre-define the number of clusters. So, let's see what hierarchical clustering is and how it improves on K-means.

Hierarchical Clustering

Let's say we have the below points and we want to cluster them into groups:



We can assign each of these points to a separate cluster. Based on the similarity, we can combine the most similar clusters together and repeat the process until only a single cluster is left



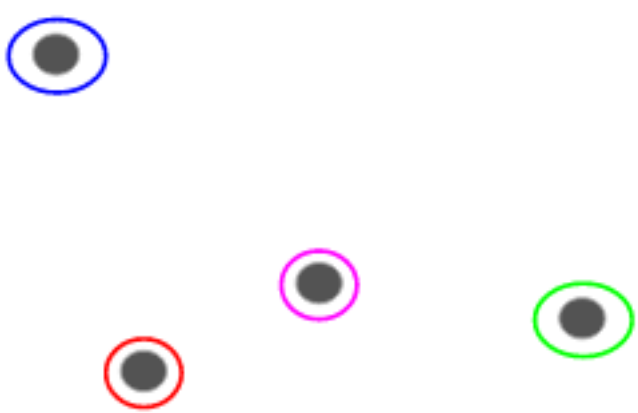
Types of Hierarchical clustering:

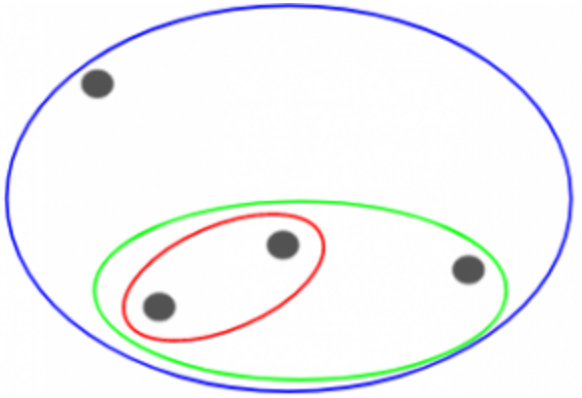
Two main types of hierarchical clustering

Agglomerative (bottom up) clustering: It builds the dendrogram (tree) from the bottom level, and

merges the most similar (or nearest) pair of clusters

stops when all the data points are merged into a single cluster (i.e., the root cluster).





Steps to Perform Agglomerative Hierarchical clustering:

Following are the steps involved in agglomerative clustering:

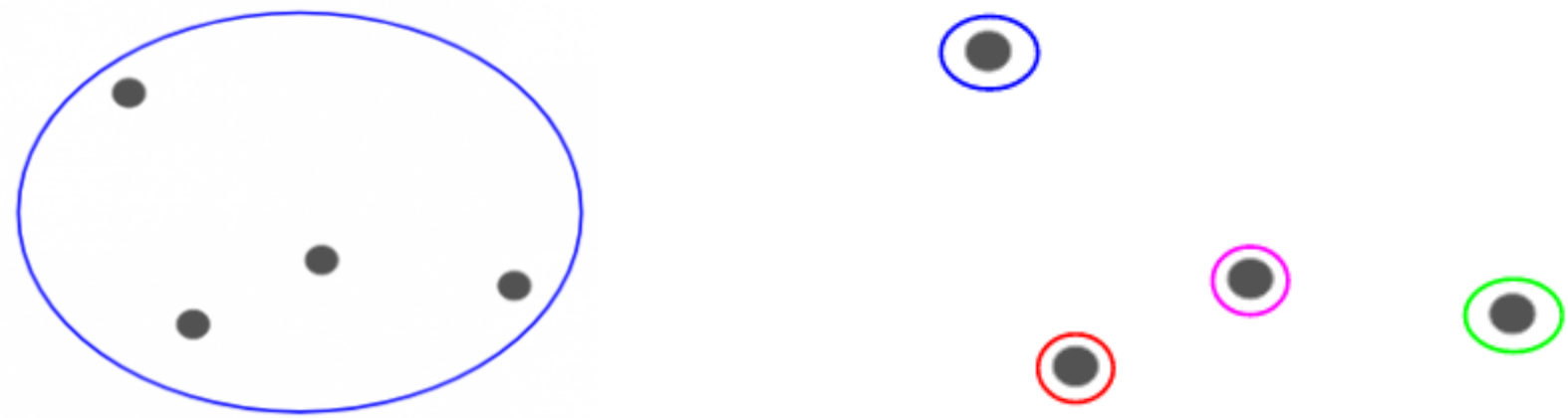
1. At the start, treat each data point as one cluster. Therefore, the number of clusters at the start will be K, while K is an integer representing the number of data points.
2. Form a cluster by joining the two closest data points resulting in K-1 clusters.
3. Form more clusters by joining the two closest clusters resulting in K-2 clusters.
4. Repeat the above three steps until one big cluster is formed.
5. Once single cluster is formed, dendrograms are used to divide into multiple clusters depending upon the problem.

Types of Hierarchical clustering:

Divisive (top down) clustering: It works in the opposite way. It starts with all data points in one cluster, the root.

Splits the root into a set of child clusters (the farthest point in the cluster). Each child cluster is recursively divided further

stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point



Agglomerative clustering:

It is more popular then divisive methods.

At the beginning, each data point forms a cluster (also called a node).

Merge nodes/clusters that have the least distance.

Go on merging

Eventually all nodes belong to one cluster

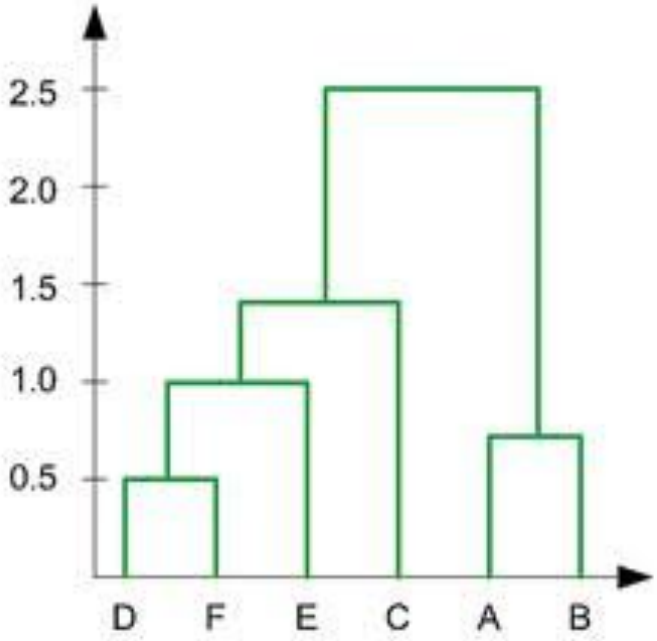
Agglomerative clustering:

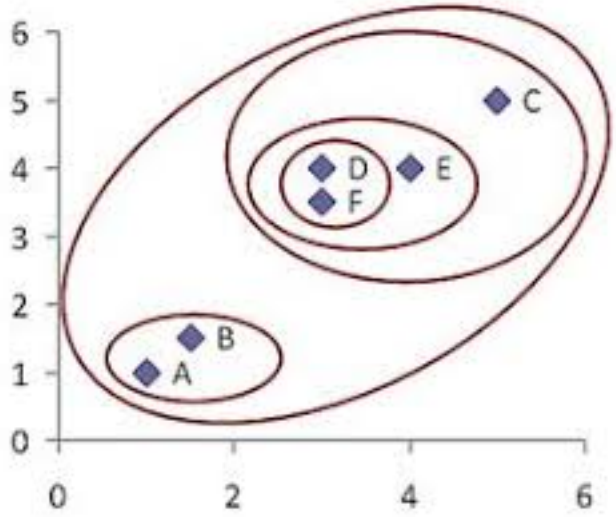
Algorithm Agglomerative(D)

1 Make each data point in the data set D a cluster,
2 Compute all pair-wise distances of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in D$;
2 **repeat**
3 find two clusters that are nearest to each other;
4 merge the two clusters form a new cluster c;
5 compute the distance from c to all other clusters;
12 **until** there is only one cluster left

Hierarchical clustering:

- Produces a set of nested sequence of clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





Hierarchical clustering:

- Dendrogram: Graphical representation (tree graph) of the results of a hierarchical procedure. Starting with each object as a separate cluster, the dendrogram shows graphically how the clusters are combined at each step of the procedure until all are contained in a single cluster

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by cutting the dendrogram

- Can be visualized as a dendrogram at the proper level

- Whenever two clusters are merged, we will join them in dendrogram and the height of the join will be the distance between these points.

Hierarchical clustering:

- More the distance of the vertical lines in the dendrogram, more the distance between those clusters.
- We can set a threshold distance and draw a horizontal line (generally, we try to set the threshold in such a way that it cuts the tallest vertical line without any horizontal line passing through).
- The number of clusters will be the number of vertical lines which are being intersected by the line drawn using the threshold.
- Basically the horizontal line is a threshold, which defines the minimum distance required to be a separate cluster.
- This is how we can decide the number of clusters using the dendrogram.

Measuring the distance of two clusters

A few ways to measure distances of two clusters.

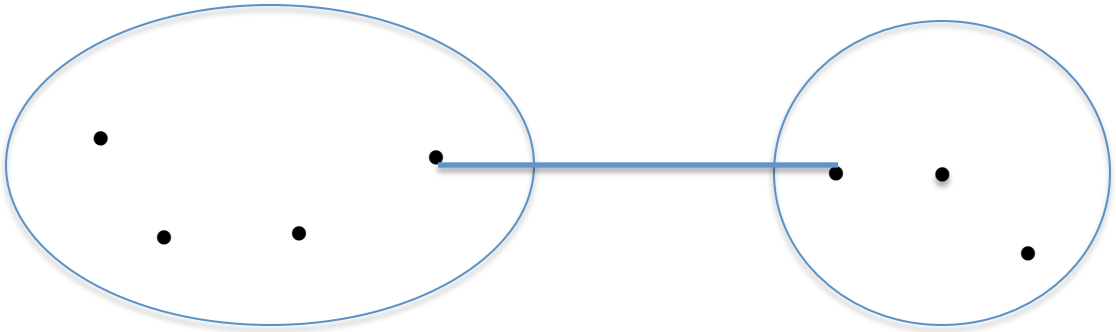
Results in different variations of the algorithm.

- Single link
- Complete link
- Average link
- Centroids
- …

Single Link method

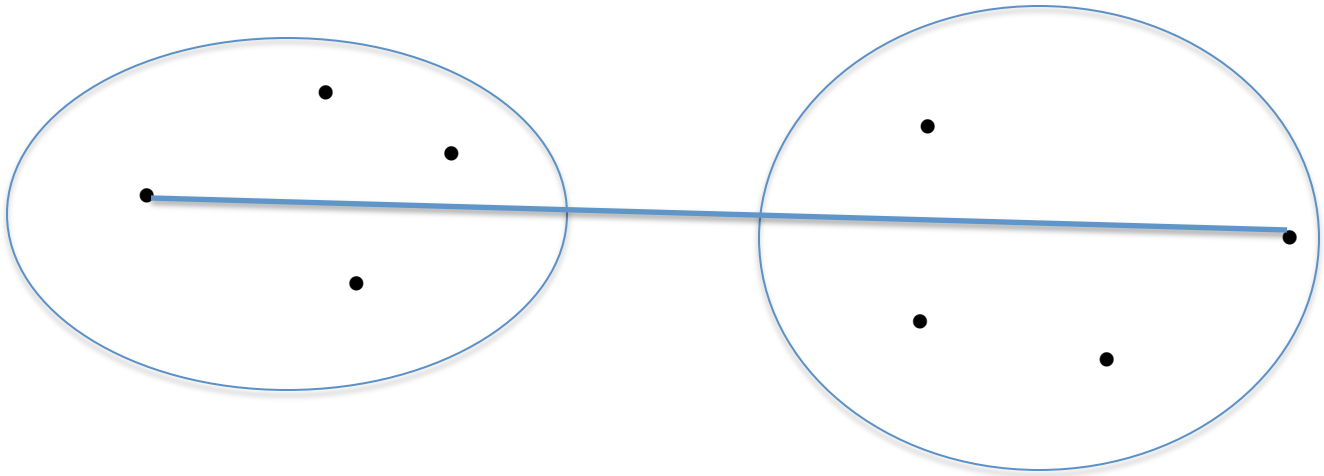
The distance between two clusters is the distance between two closest data points in the two clusters, one data point from each cluster.

It can find arbitrarily shaped clusters, but it may cause the undesirable "chain effect" by noisy points



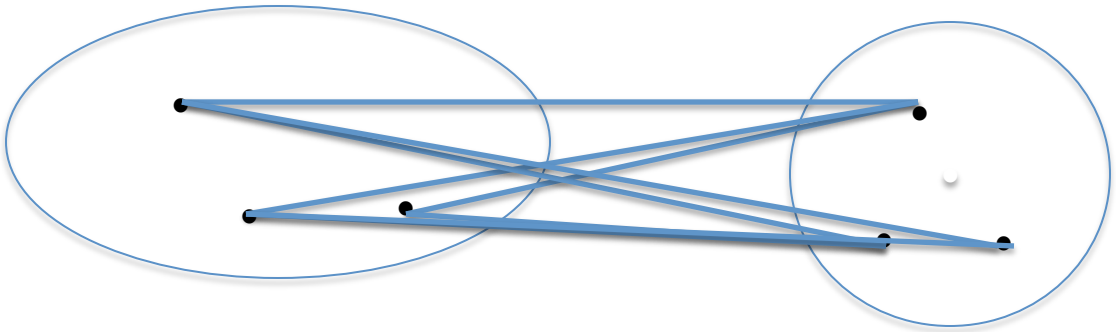
Complete link method

The distance between two clusters is the distance of two furthest data points in the two clusters.
It is sensitive to outliers because they are far away



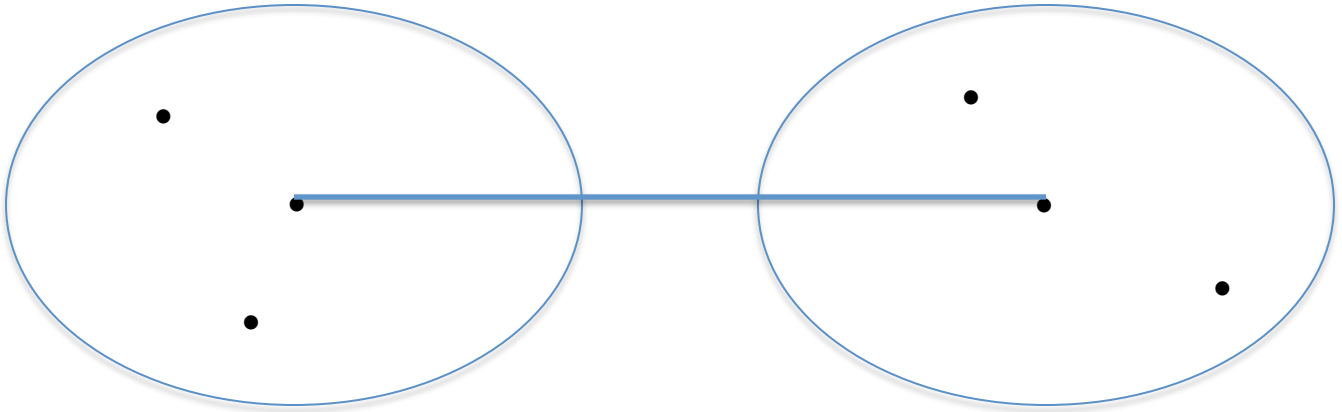
Average link and centroid methods

- Average link: A compromise between
  - the sensitivity of complete-link clustering to outliers and
  - the tendency of single-link clustering to form long chains that do not correspond to the intuitive notion of clusters as compact, spherical objects.
  - In this method, the distance between two clusters is the average distance of all pair-wise distances between the data points in two clusters. •



Average link and centroid methods

Centroid method: In this method, the distance between two clusters is the distance between their centroids



The complexity

- All the algorithms are at least $O(n^2)$. n is the number of data points.
- Single link can be done in $O(n^2)$.
- Complete and average links can be done in $O(n^2 \log n)$.
- Due the complexity, hard to use for large data sets.

Advantages and Disadvantages

Advantages

- No apriori information about the number of clusters required
- Easy to implement and gives best results in some cases

Disadvantages

- Algorithm can never undo what was done previously
- No objective function is directly minimized
- Sometimes it is difficult to identify the correct number of clusters by the dendrogram
- Based on the type of distance matrix chosen for merging
- different algorithms can suffer with one or more of the following:
  - Sensitivity to noise and outliers
  - Breaking large clusters
  - difficulty handling different sized clusters and convex shapes.

How to choose a clustering algorithm

- Clustering research has a long history. A vast collection of algorithms are available.
  - We only introduced several main algorithms.
- Choosing the "best" algorithm is a challenge.
  - Every algorithm has limitations and works well with certain data distributions.
  - It is very hard, if not impossible, to know what distribution the application data follow. The data may not fully follow any "ideal" structure or distribution required by the algorithms.
  - One also needs to decide how to standardize the data, to choose a suitable distance function and to select other parameter values.

How to choose a clustering algorithm

- Due to these complexities, the common practice is to
  - run several algorithms using different distance functions and parameter settings, and
  - then carefully analyze and compare the results.
- The interpretation of the results must be based on insight into the meaning of the original data together with knowledge of the algorithms used.
- Clustering is highly application dependent and to certain extent subjective (personal preferences).

K-means vs. Hierarchical

	Hierarchical	K-means
Running time	$O(N^{2)}$	Fastest, each iter is linear
Assumptions	Requires a similarity/ dissimilarity matrix	Strong assumption
Input parameters	None	K # of clusters
clusters	Subjective(only tree is returned)	Exactly k clusters

Practical Issues in clustering

- Should the observation or features first be standardized I some way. For instance, may be the variables should be centered to have mean zero and scaled to have standard deviation one.
- In the case of K-means clustering, how many clusters should look for in the data? Not obvious what is good K. The results strongly depends on the initial guess of centroids. Only numerical attributes are covered.
- In the case of hierarchical clustering
  - What dissimilarity measure should be used?
  - What type of linkage should be used?
  - Where should we cut the dendrogram in order to obtain clusters?

Each of these decisions can have a strong impact on the results obtained. In practice, we try several different choices, and look for the one with the most useful or interpretable solution. With these methods, there is no single right answer.

Summary

- Clustering is has a long history and still active
  - There are a huge number of clustering algorithms
  - More are still coming every year.
- We only introduced main algorithms. There are many others, e.g.,
  - density based algorithm, sub-space clustering, scale-up methods, neural networks based methods, fuzzy clustering, co-clustering, etc.
- Clustering is hard to evaluate, but very useful in practice. This partially explains why there are still a large number of clustering algorithms being devised every year.
- Clustering is highly application dependent and to some extent subjective.