MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

IN COLLABORATION WITH MIT MANAGEMENT SLOAN SCHOOL | UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

MASTER MEIM 2021-2022

Machine Learning: Unsupervised techniques

LESSON 1

prof. Antonino Staiano

M.Sc. In Applied Computer Science of University Parthenope of Naples

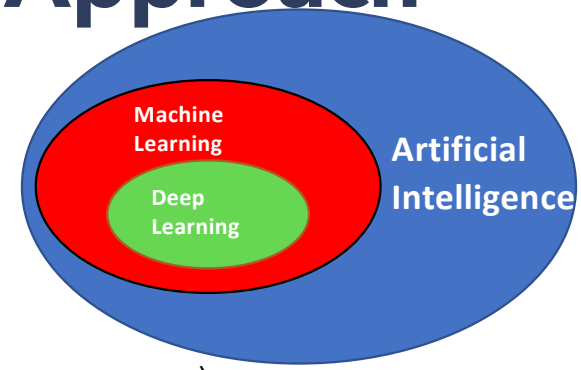
Lessons plan

- **Clustering algorithms** (Lesson 1) - July 21, 09:00 – 13:00
  - Prof. Antonino Staiano
- Hands-on Clustering (Lesson 2) - July, 21, 14:00 – 18:00
  - Prof. Alessio Ferone
- Dimensionality reduction (Lesson 3) - July 22, 09:00 – 13:00
  - Prof. Antonino Staiano
- Hands-on Dimensionality reduction (Lesson 4) - July 22, 14:00 – 18:00
  - Prof. Alessio Ferone

Machine learning

- Autonomously identify patterns in data that generate knowledge and help make better decisions and accurate predictions
- Pervasively employed in daily life to make critical decisions in medical diagnosis, the stock market, energy load forecasting and much more
- Multimedia service providers (Spotify, Amazon Music, Netflix, …) rely on machine learning to sift through millions of options for giving advice on songs or movies
  - E-commerce companies, such as Amazon, use it to gain insight into their customers' buying behavior

Machine Learning: A Data-Driven Approach



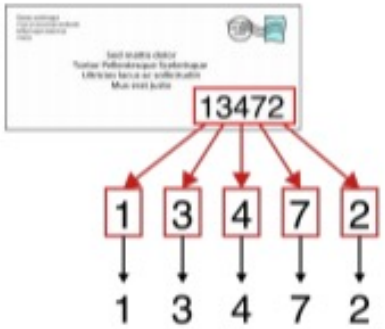
- Machine Learning (ML)
  - Discover meaningful information from data
- Data
  - Anything that can be recorded and measured
    - Raw numbers (e.g., stock prices, planet masses, heights of people visiting a museum)
    - Sounds (e.g., words someone speaks into their cell phone)
    - Pictures (e.g., photographs of flower or cats)
    - Words (e.g., the text of a newspaper article or novel)
    - … (anything else you want to investigate)
- Deep Learning
  - Approaches to Machine Learning that use specialized layers of computation, stacked up one after the next

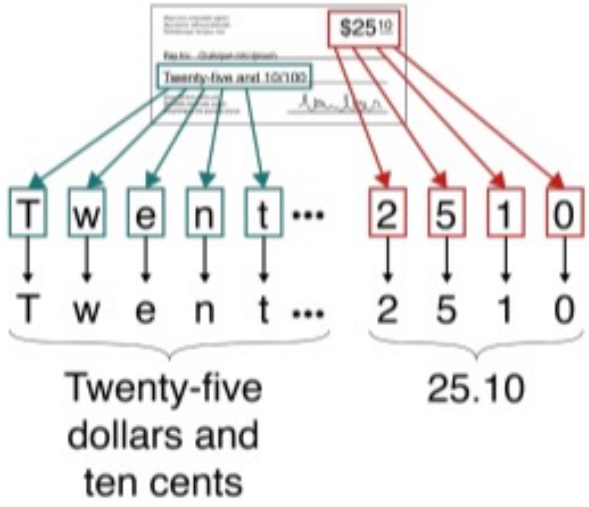
Machine Learning

- The basic idea is to equip computers with the same ability as humans and animals: To learn from experience
  - It is focused on learning
- Computational techniques are used to learn the information directly from data without relying on a predetermined equation as a model
- ML algorithms adaptively improve their performance from time to time as more examples are available for learning

Extracting Meaning from Data

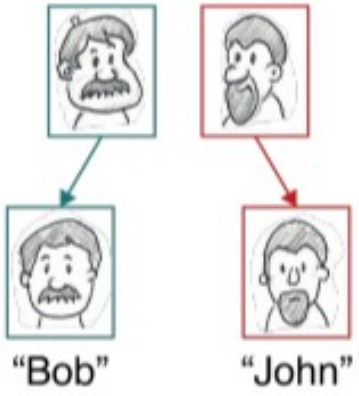
Getting a zip code from an envelope





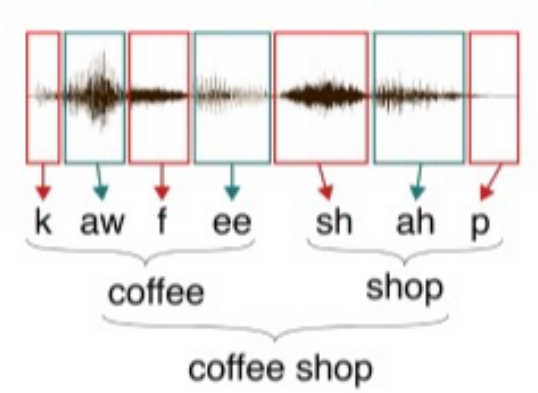
Reading numbers and letters on a check

Recognizing faces from photos

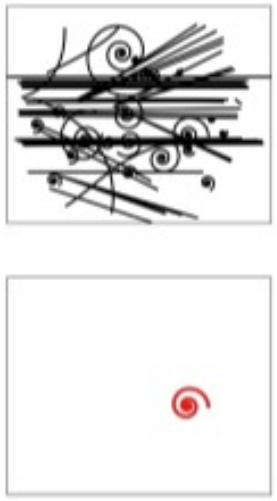


Extracting Meaning from Data (cont'd)

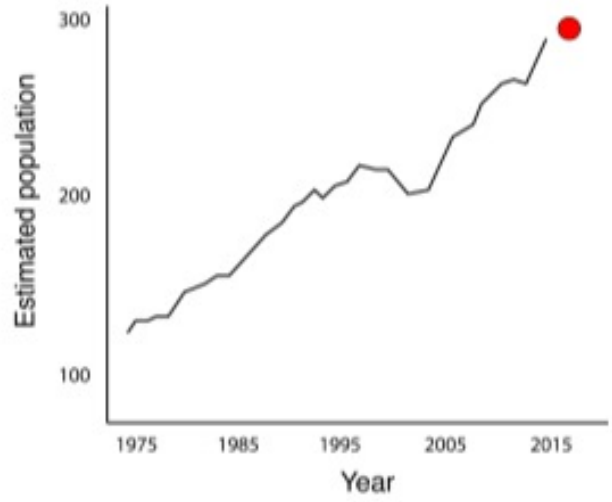
Turning a recording into sounds, then words, and ultimately a complete utterance



Finding one unusual event in a particle accelerator's output full of similar-looking trails



Predicting the population of the northern resident Orca whale population off Canada's west coast



MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

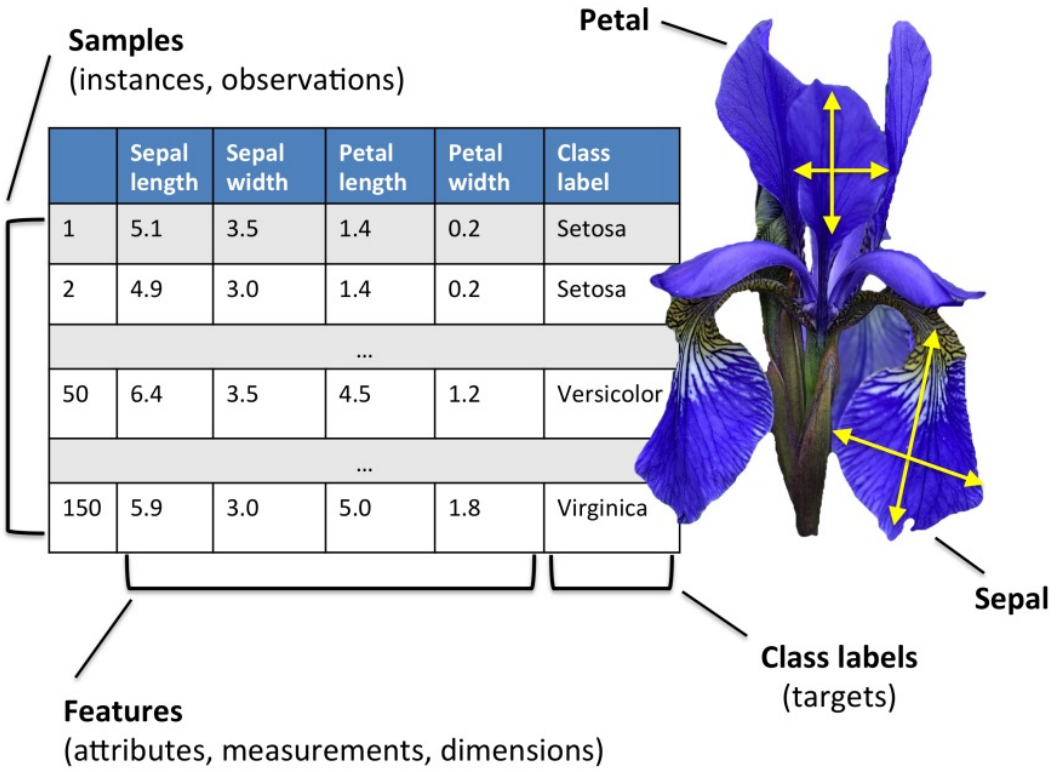
IN COLLABORATION WITH MIT MANAGEMENT SLOAN SCHOOL | UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

Type of Data and their representation

Preliminaries

Data Terminology

- There are several synonyms to describe data, a matrix formed by *samples (rows)* and *variables (columns)*
  - Sample
    - *Record, row, instance, observation, example*
  - Variable
    - *Feature, attribute, column, measurement, field*

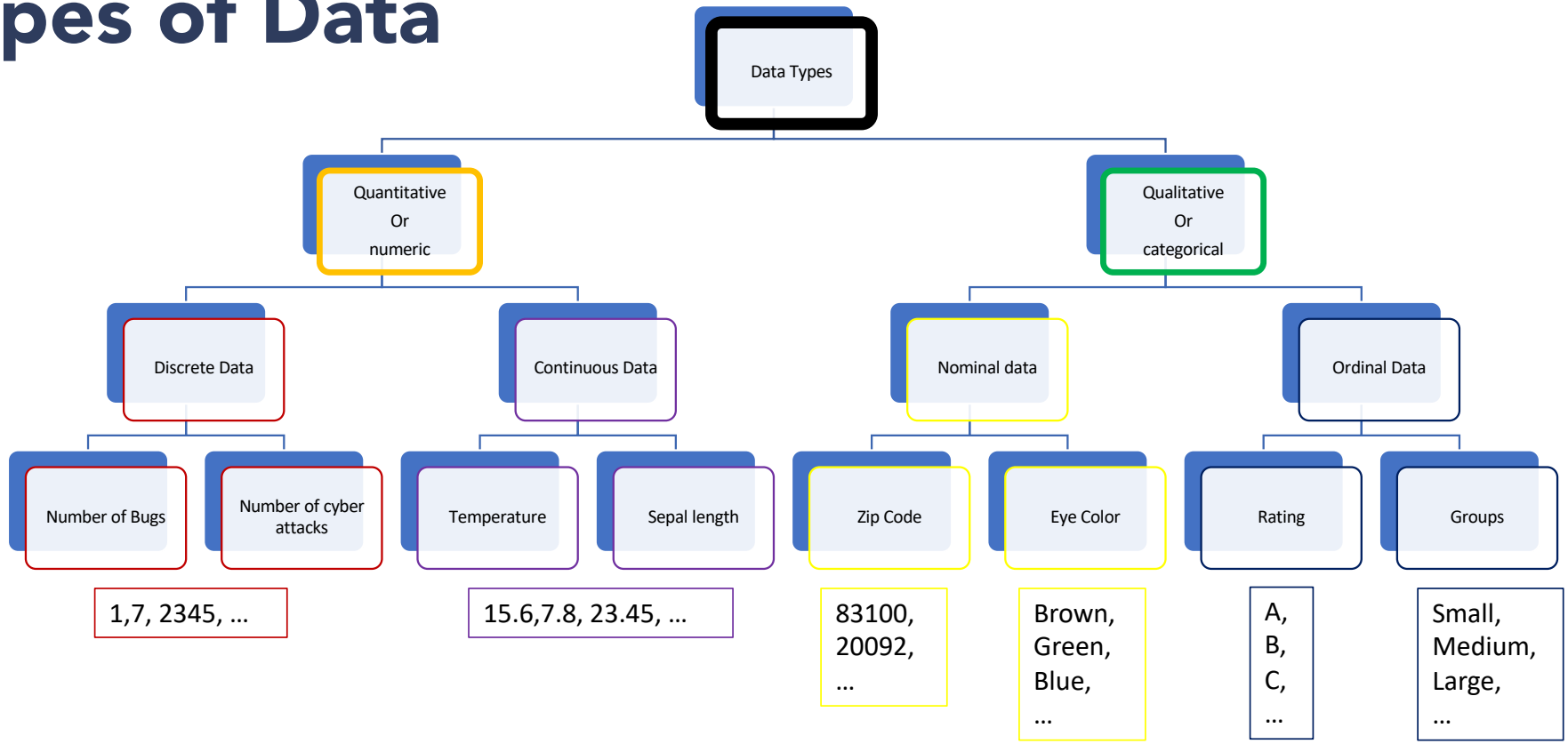


	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

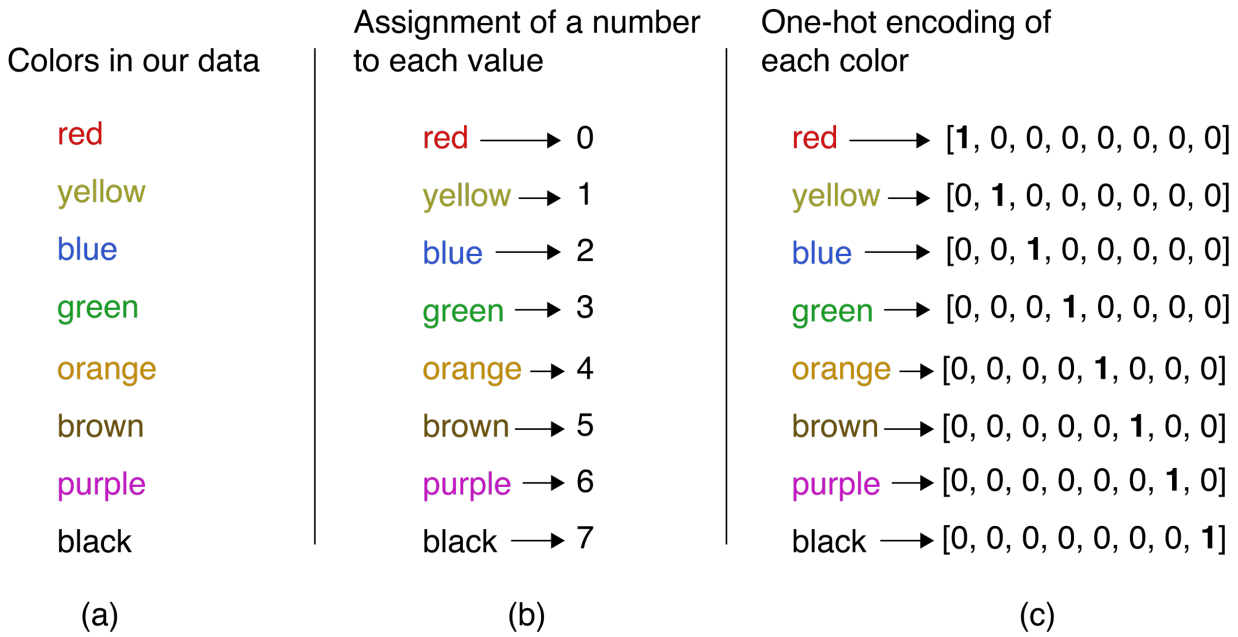
Types of Data

- A sample is a list of values, each of which is called a feature
- Each feature can be either of two general types
  - Numerical
    - A number, either floating-point or integer
    - Quantitative data
    - Can be sorted using its values
  - Categorical
    - Is a string that describes a label or category, such as “cow” or “tiger”
      - Ordinal
        - Has a known order, so it can be sorted (e.g., rainbow colors)
      - Nominal
        - Does not have a natural ordering (e.g., desktop items)

Types of Data



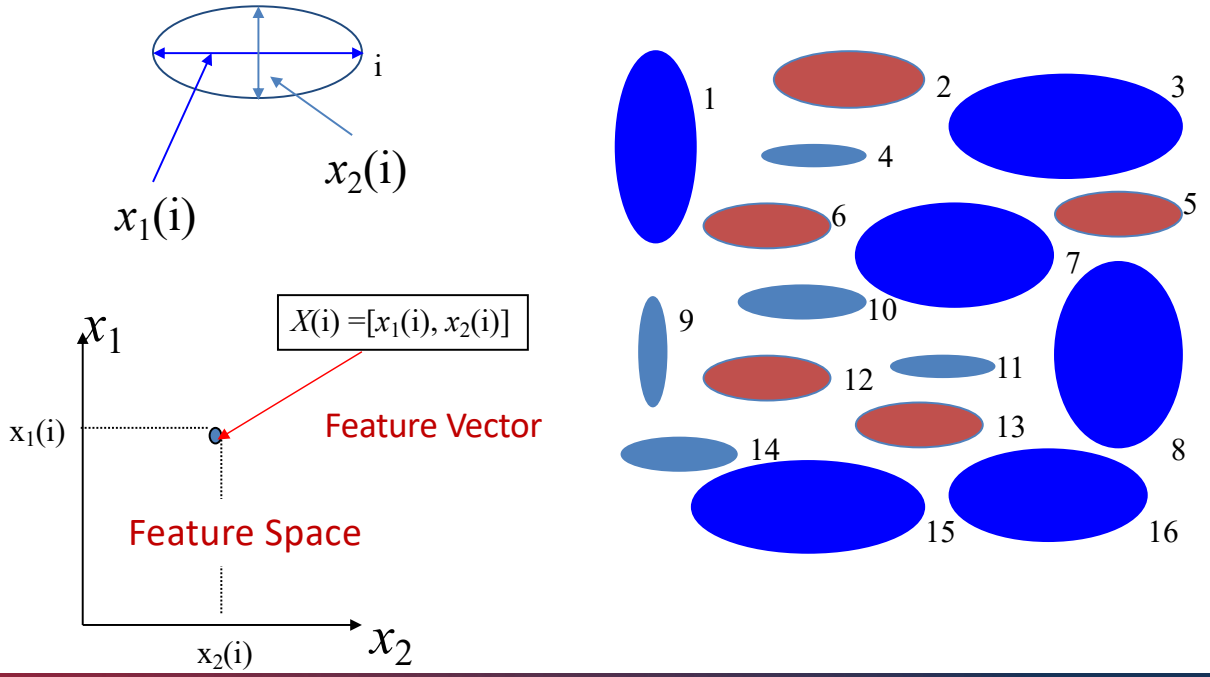
Data Encoding



- Often more convenient to work with a list of numbers than a single value

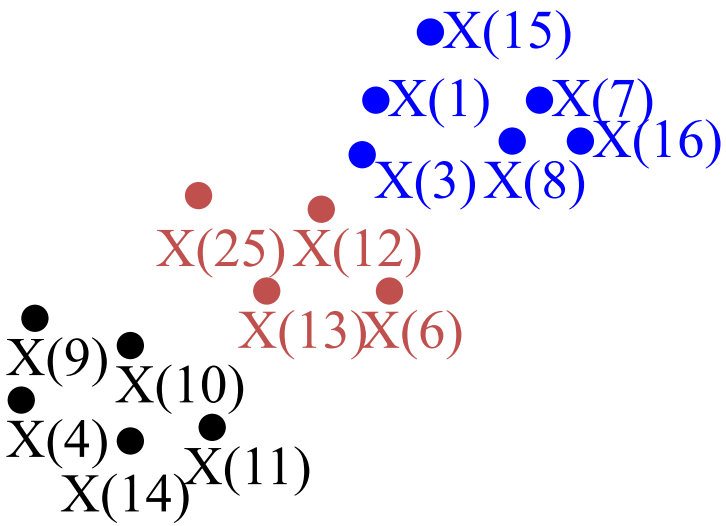
Feature Space

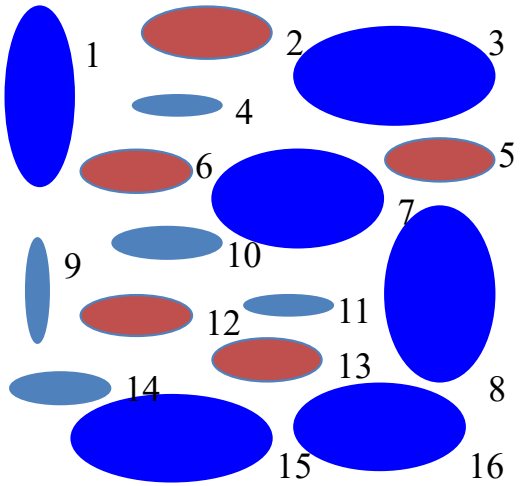
- Any entity of the real-world is represented by a feature vector



Feature space

- From Objects to Feature Vectors to Points in the Feature Spaces





Elliptical balls (objects)

Representing general objects

- Feature vectors for
  - Faces
  - Cars
  - Fingerprints
  - Gestures
  - Emotions (a smiling face, a sad expression, …)
  - …

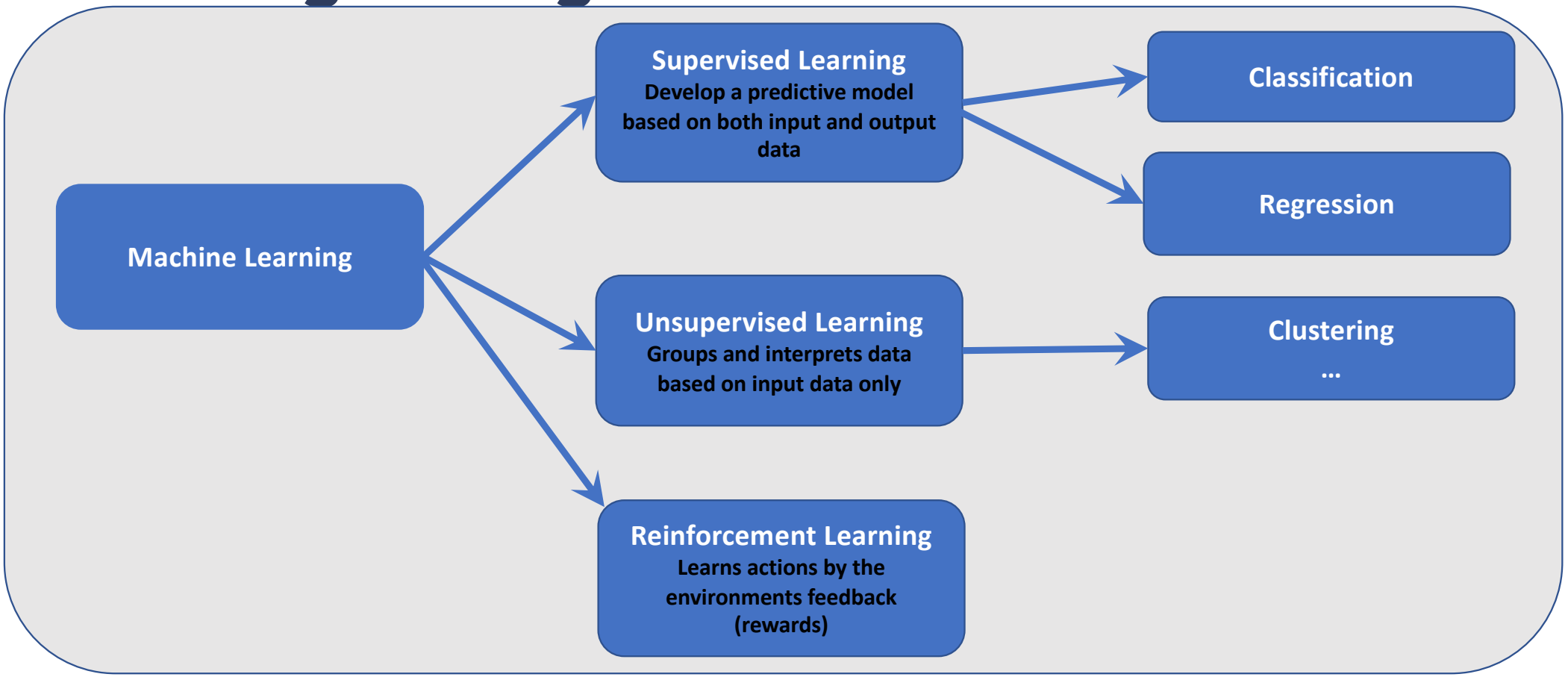
MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

IN COLLABORATION WITH MIT MANAGEMENT SLOAN SCHOOL | UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

Types of learning

Learning paradigms

Learning strategies



Preparing a learning task

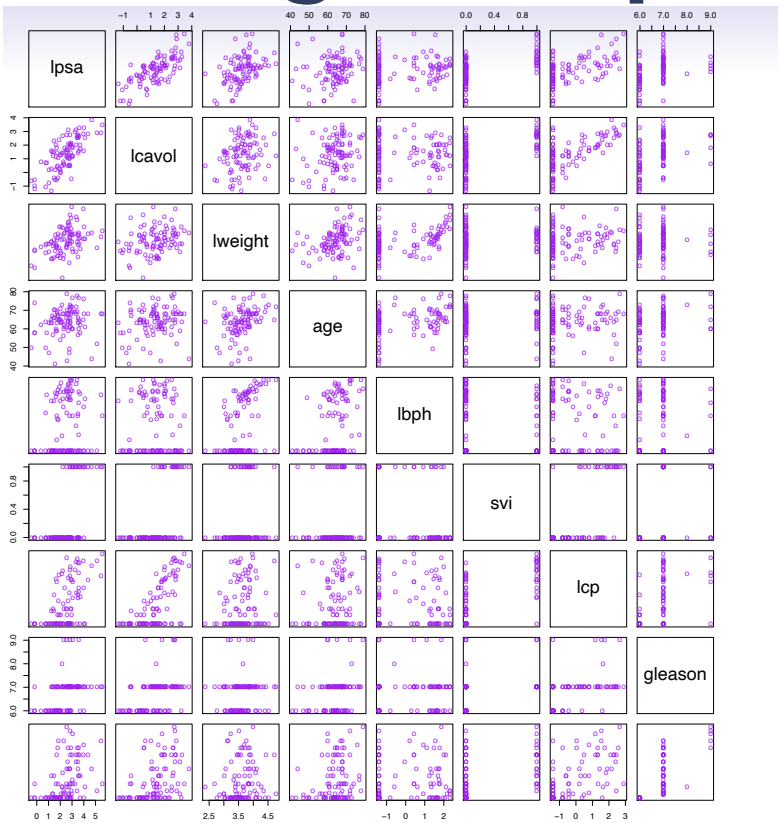
- Starts by collecting the facts we're going to teach
  - Collecting as much data as we can get
- Sample
  - Each piece of observed data (for instance, the weather at a given moment)
- Features
  - Names of the measurements that make it up (e.g., the temperature, wind speed, humidity)
  - Each feature has an associated value, typically stored as a number
- Each sample could be provided with a label
  - E.g., if our sample is a photo, the label could be the name of the person in the photo, or a type of an animal it shows

Supervised learning

- When our samples come with pre-assigned labels we talk about **supervised learning**
  - The supervision is provided by labels
- Two general types of supervised learning exist
  - **Classification**
    - Looking through a given set of categories to find the one the best describes a particular input
  - **Regression**
    - Taking a set of measurements and predicting some other value

Supervised learning: examples

Identify prostate cancer risk factors by predicting PSA level



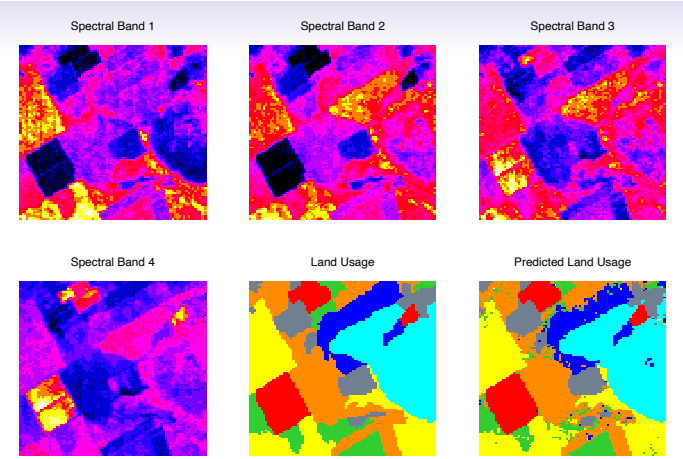
Supervised learning: examples

- Build a spam email detection system
  - Data from 4061 emails, each one labeled as **spam** or **email**
  - Goal: to build a custom spam filter
  - Input (features): relative frequencies of 57 most commonly occurring words and punctuation symbols in such email messages

	george	you	hp	free	!	edu	remove
spam	0.00	2.26	0.02	0.52	0.51	0.01	0.28
email	1.27	1.27	0.90	0.07	0.11	0.29	0.01

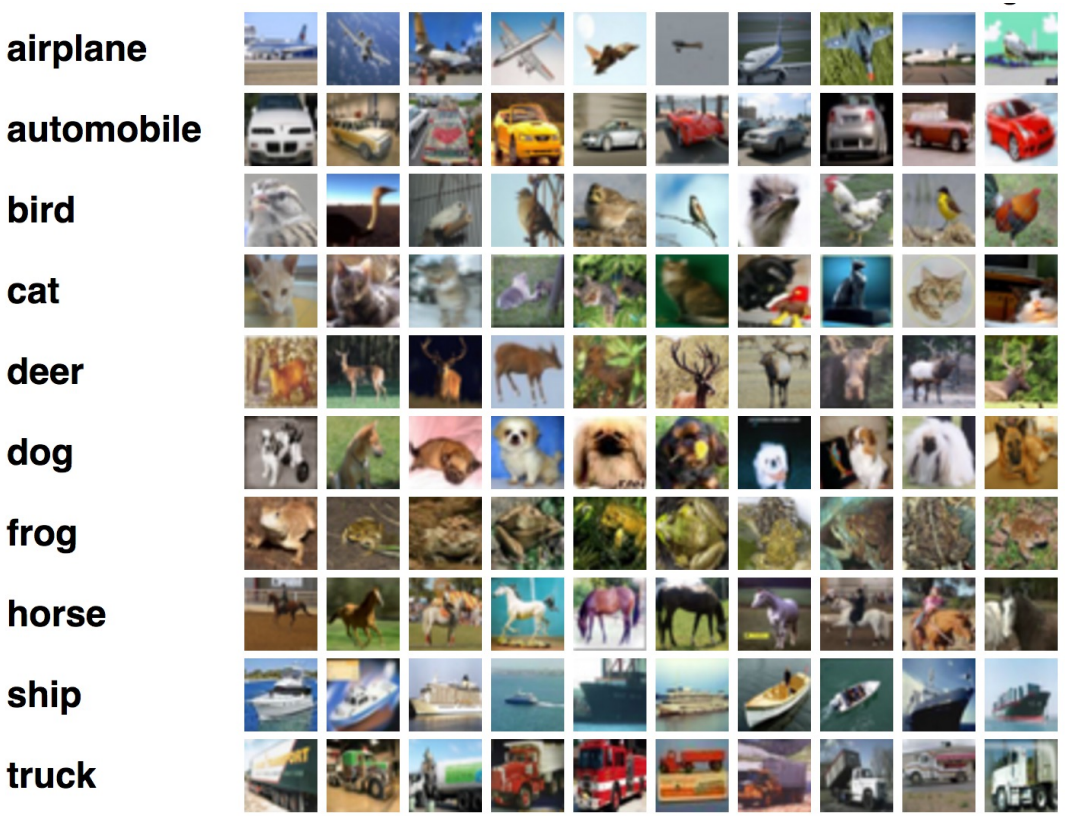
Supervised learning: examples

- Identify the pixels in a LANDSAT image by category of use:
  - {red soil, cotton, vegetation stubble, mixed, gray soil, damp gray soil}



Supervised learning: examples

- Organize a set of photos of everyday objects based on what they show



Supervised learning

- The examples just described have many things in common
- For each we have a set of variables (**features**) that can be denoted as inputs, whose measured values are available
  - Their values affect one or more outputs in some way
- For each example, the goal is to use inputs to predict outputs
- This is what we call **supervised learning**

Quantitative and qualitative output variables

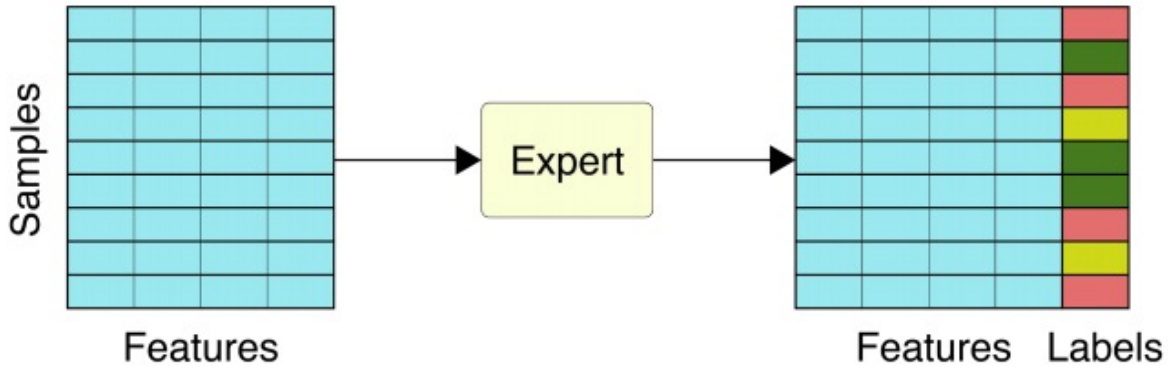
- The type of the output variable (s) can be different in various real-world problems
  - In the example of PSA prediction (for prostate cancer), the output is a **quantitative measure**
  - In the LANDSAT example, the output is **qualitative** (different land uses) and takes values in a finite set {red soil, cotton, vegetation stubble, mixed, gray soil, damp gray soil} of categories
    - There is no ordering between the categories, which is why descriptive labels are often used to denote them instead of numbers
    - Qualitative variables are called categorical or discrete

Classification and regression

- For each type of output variable, it makes sense to use inputs to predict outputs
- The distinction in the type of output determines the specific prediction task
  - Regression: when we want to predict quantitative outputs
  - Classification: when you want to predict qualitative values
- Both can be seen as methods for the approximation of functions

A Learning strategy for the computer

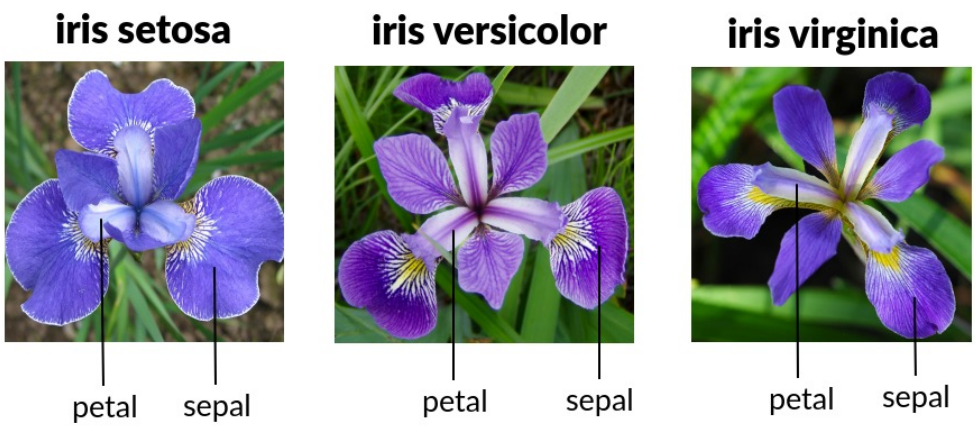
- Weather measurements on a mountain
  - The expert's opinion provides the confidence the day's weather conditions are good for hiking

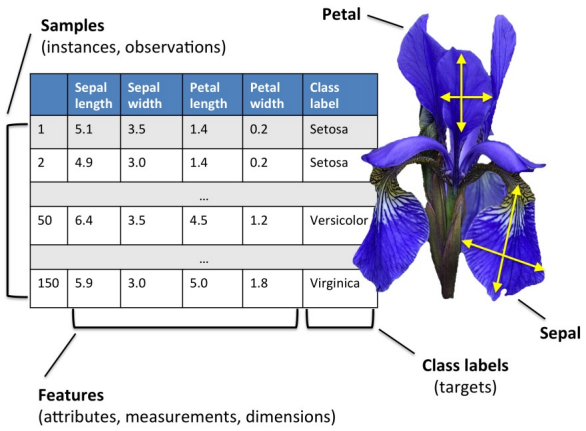


  - Labeled data are given to a computer for it to produce the right label for each input
    - We do not tell it how to do this, rather we give it an algorithm with many parameters it can adjust
  - The algorithm is run to produce an output which is the computer prediction of what it thinks the expert's label is for that sample

An example of classification problem

- Characteristic measures of three species of Iris flower
  - 150 flowers
  - 4 features: $\mathbf{X} = (x_1, x_2, x_3, x_4)^T$
    - $\mathbf{X}$ = (Petal_L, Petal_W, Sepal_L, Sepal_W)
  - 3 outputs: Iris species, C_1 = Versicolor, C_2 = Virginica, C_3 =Setosa
- The goal is to correctly classify the three species of Iris

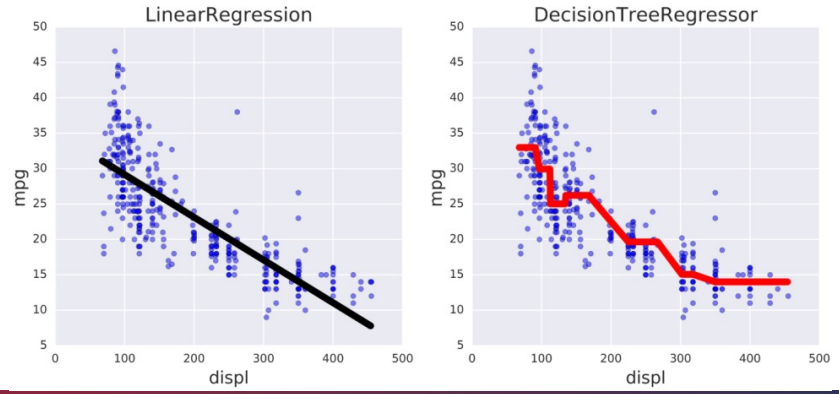




	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

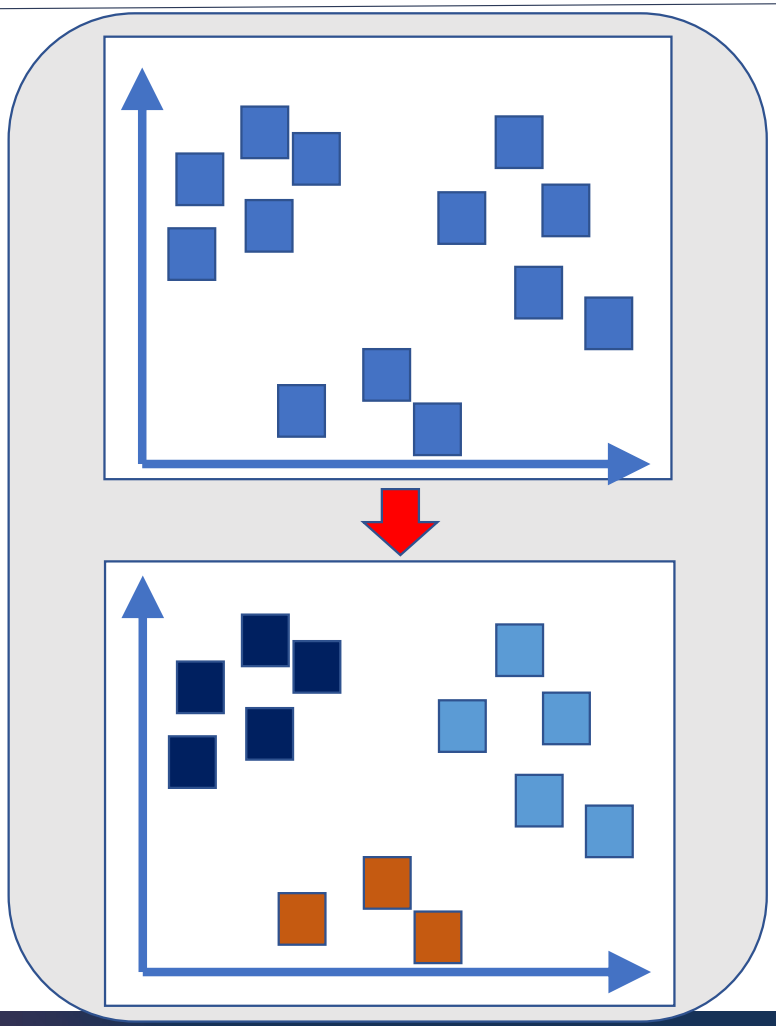
An example of regression problem

- Data relating to different car models, from 1970 to 1982
  - 406 cars
    - 7 features, $\mathbf{X} = (x_1, x_2, \ldots, x_7)^T$
      - X = (Acceleration, Cylinders, Displacement, Horsepower, Model_year, Weight, Origin)
    - 1 output: Y = MPG
- Predict fuel consumption (expressed in miles per gallon)



Unsupervised learning

- Unsupervised learning finds hidden patterns or intrinsic structures in data
- It is used to derive inferences from datasets that consists of input data without labeled answers
- Clustering is the most common form of unsupervised learning
  - Used for exploratory data analysis to find hidden patterns or groupings in data
  - Applications of clustering include gene expression analysis, market research, object recognition …
- Now, it's time to dive in the *unsupervised sea* …



MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

IN COLLABORATION WITH MIT MANAGEMENT SLOAN SCHOOL | UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

Unsupervised learning

Unsupervised Learning

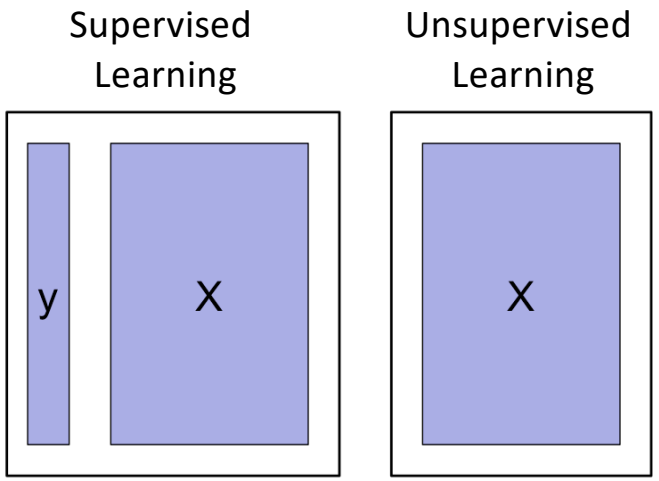
- We will investigate two kinds of unsupervised learning
  - Clustering (today)
  - Data transformation (tomorrow)
    - Dimensionality reduction

Unsupervised Learning tasks

- Typical examples of unsupervised learning tasks include the problem of image and text segmentation and the task of novelty detection in process control
- In unsupervised learning we are given a training sample of objects (e.g., images) with the aim of extracting some *structure* from them
  - For instance, identifying indoor or outdoor images or extracting face pixels in an image

Unsupervised vs Supervised learning

- With supervised learning we observe both a set of features $x_1, x_2, \ldots, x_p$ for each object and the corresponding value of the output variable Y. Then the goal is to predict Y through $x_1, x_2, \ldots, x_p$
- Now let's shift our attention to unsupervised learning, where we only observe the features $x_1, x_2, \ldots, x_p$
  - We are not interested in prediction, as we do not have any output variables

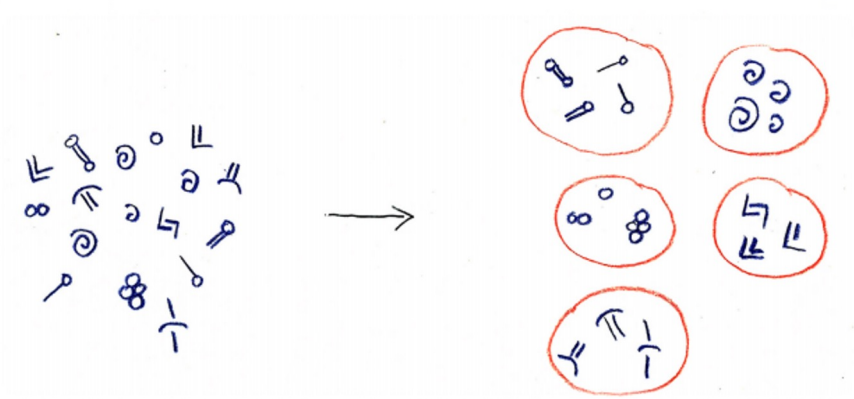


Unsupervised learning

- When input data does not have labels, the learning algorithm comes from the category of **unsupervised learning**
  - In other words, we are not supervising the learning process by feeding labels
  - Rather, the system must figure out everything out on its own, with no help from us
- Unsupervised learning is often used to solve problems called **clustering**, **noise reduction**, **dimensionality reduction**

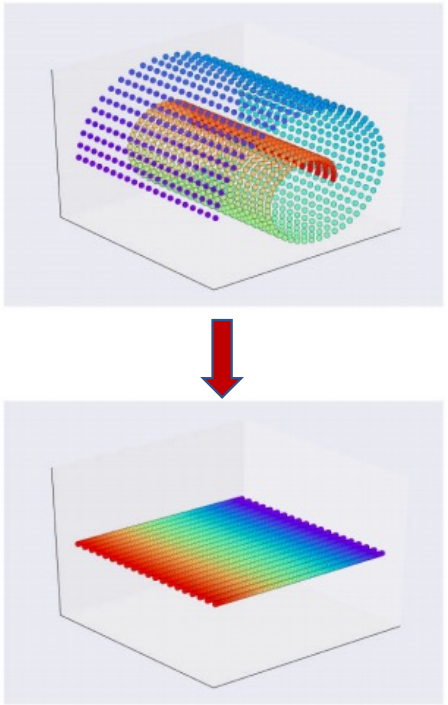
Unsupervised learning: clustering

- Data are grouped automatically by the algorithm based on their similarity
- The problem is called a clustering problem and the algorithm finding the groupings is called a clustering algorithm
- Because the inputs are unlabeled the clustering is performed using an unsupervised learning algorithm
- There is a plethora of clustering algorithms in literature to choose from



Unsupervised learning: Dimensionality reduction

- The data used in machine learning processes often have many variables (features)
  - If your dataset has two features, then it is two-dimensional data. If it has three features, then it is three-dimensional data and so on
- One aims at using as many features as possible to capture the characteristics of data, but avoiding the dimension to be too high
- Most of these dimensions may or may not matter in the context of our application with the questions we are asking
- Reducing such high dimensions to a more manageable set of related and useful variables improves the performance and accuracy of our analysis



Unsupervised learning: noise reduction

- In many real-world applications, samples are corrupted by noise
- A de-noising algorithm might help to clean our noisy data up

- Because we don't have labels for our data (e.g., in a noisy photo we've just pixels), de-nosing is a form of unsupervised learning
  - By learning the statistics of the samples, the algorithm estimates what part of each sample is noise and then removes it, leaving us with cleaned-up data that's easier to learn from and interpret

Challenges of unsupervised learning

- Unsupervised learning is more subjective than supervised learning
  - There is no simple objective for the analysis, such as the prediction of an output variable
- Since unsupervised learning algorithms are usually applied to data that does not contain any label information, we don't know what the right output should be
- Example
  - A hypothetical clustering algorithm could have grouped together all the pictures that show faces in profile and all the full-face pictures
  - This would certainly be a possible way to divide a collection of pictures of people's faces, but it's not the one we were looking for

Unsupervised learning advantage

- It is often easier to obtain unlabeled data from the instrumentation of a laboratory or from a computer than labeled data that requires human intervention
- For example, it's hard to automatically gauge the overall sentiment of a film review: is it favorable or not?

MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

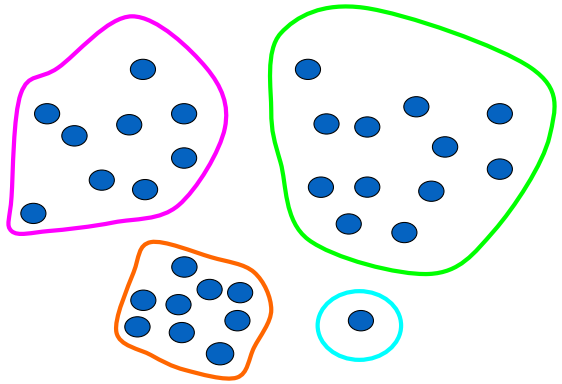
IN COLLABORATION WITH MIT MANAGEMENT SLOAN SCHOOL | UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

Clustering

Unsupervised learning

Clustering

- Clustering involves a set of techniques for identifying subgroups, or clusters, in a data set
- We look for a partition of the data into distinct groups so that the observations within each group are quite similar to each other
  - Data points in the same cluster should have a small distance from one another
  - Data points in different clusters should be at a large distance from one another
- For all of this to be of practical use, we need to define what it means for a set of two or more observations to be similar or different
  - It is a necessary consideration, specific to the domain and based on the knowledge of the data to be studied



Examples of clustering applications

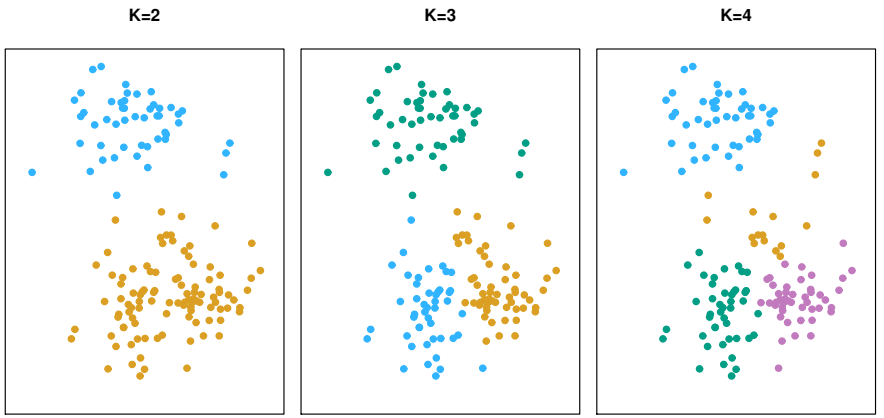
- Breast Cancer Patient Data
  - Suppose we collect **p** measurements (features) from each of **N** breast cancer patients. There could be several unknown cancers that we could discover by clustering the data
- Market segmentation
  - Segment the market by identifying sub-groups of people who may be more receptive to a particular form of advertising or more likely to buy a particular product based on many measurements (e.g. *average household income*, *employment*, *distance to the nearest urban area*, and so on) for many people
- Land use
  - Identification of areas of similar land use in an earth observation database
- Insurance
  - Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning
  - Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies
  - Observed earthquake epicenters should be clustered along continent faults

Clustering techniques

- There are many different types of clustering
- Let's focus on three among the most popular
  - Partitive clustering
    - K-means
      - We try to partition the observations into a predefined number of clusters
  - Hierarchical Clustering
    - Agglomerative
      - We don't know the number of clusters we are looking for
      - Provides a tree representation of N observations, called a dendrogram, which allows you to see the groupings obtained for each possible number of clusters from 1 to N
  - Density-based clustering
    - DBScan
      - Works by detecting areas where points are concentrated and where they are separated by areas that are empty or sparse

K-means clustering

- Let's consider a simulated data set with 150 observations in a 2-dimensional space
- K-means algorithm first requires you to specify the number K of desired clusters
- K-means results with different values of the number K of clusters
  - The color of each observation indicates the cluster to which it was assigned by the K-means algorithm
  - We observe that there is no ordering between clusters, so the coloring of the clusters is arbitrary
  - The colors are not labels used by the algorithm but are the result of the clustering procedure



Details of the K-means algorithm

- Let $C_1, \ldots, C_K$ be sets containing the indices of the observations in each cluster. These sets satisfy two properties
  1. $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \ldots, N\}$ → each observation belongs to at least one of the K clusters
  2. $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$ → i clusters do not overlap, i.e., no observation belongs to more than one cluster
     - For instance, if the *i-th* observation is in the *k-th* cluster, then i ∈ C_k and i ∉ $C_{k'}$ for each k′ ≠ k

Details of the K-means algorithm

- The idea of the K-means algorithm is that a good clustering is the one for which the variability within each cluster is as small as possible
- The intra-cluster variability for cluster C_k is a measure, VIC (C_k), expressing how much the observations in a cluster differ from each other
- So, we want to solve the problem

$$min_{C_{1,\dots,C_K}} \left\{ \sum_{k=1}^{K} VIC(C_k) \right\} \qquad (1)$$

- The formula tells us that we want to partition the observations into K clusters so that the total intra-cluster variety, summed over all K clusters, is as small as possible

How do we define intra-cluster variability?

- Euclidean distance is usually used

$$VIC(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \qquad (2)$$

where $|C_k|$ denotes the number of observations in the k-th cluster

- Combining (1) and (2) we obtain the optimization problem that defines the K-means clustering algorithm:

$$min_{C_{1,\dots,CK}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

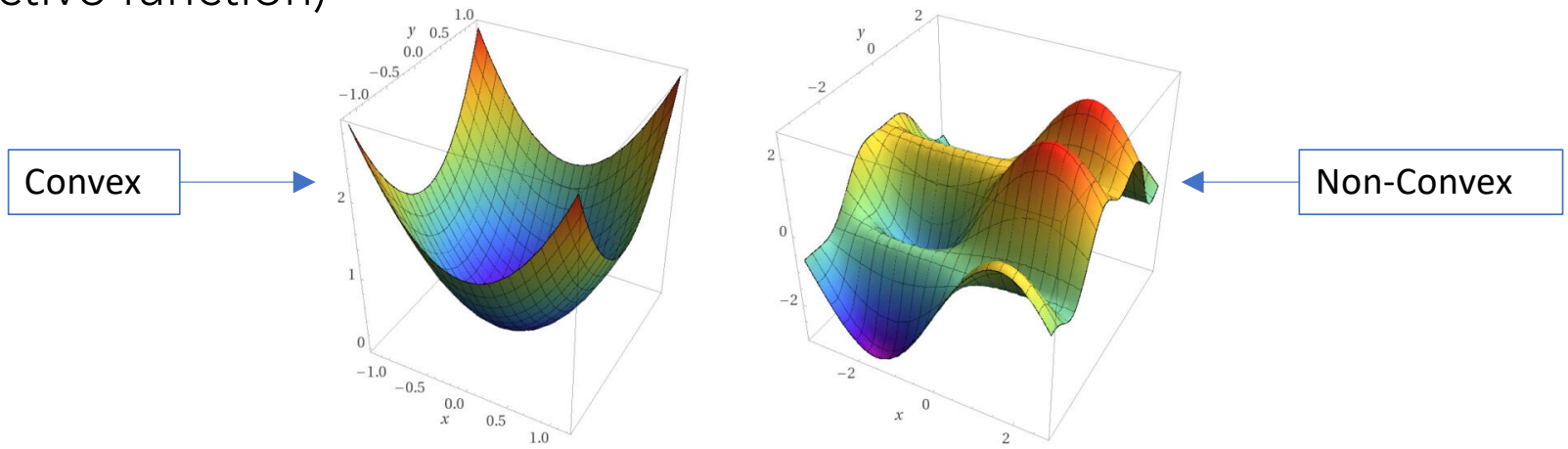
Objective Function

K-means clustering algorithm

1. Randomly assign a number, from 1 to K, to each observation. This serves as an initial assignment of observations to clusters
2. Iterate until the assignment to the clusters remains stable (it no longer changes)
   - 2.1 For each of the K clusters, calculate the centroid
     - The centroid of the k-th cluster is the vector of the means of the p features of the observations in the k-th cluster
   - 2.2 Assign each observation to the cluster whose centroid is *closest*
     - closest is defined using Euclidean distance

Properties of k-means algorithm

- At each step it is guaranteed that the value of the objective function decreases
  - Since each step of the alternating optimization scheme decreases the objective function
- However, it is not guaranteed that it gives the global minimum (non-convex objective function)



K-means at work

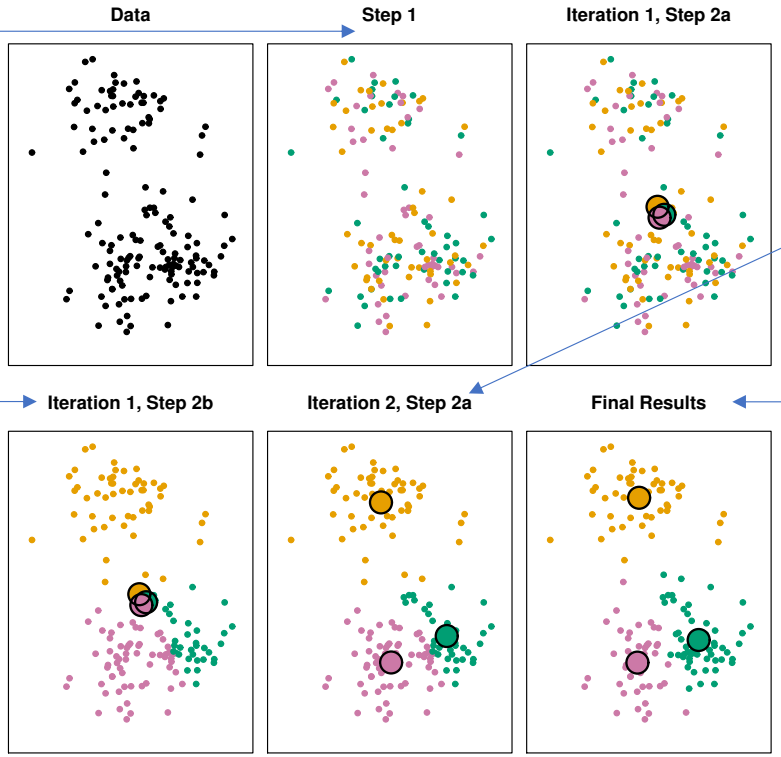
Random assignment of points

Computation of cluster centers from initial assignment

Calculation of the new cluster centroids

Assignment of points to the centroid of the closest cluster

No further changes, so stop



Details for previous figure

- The progress of the K-means algorithm with K = 3
- Top left: Observations (data set)
- Center top: Step 1, each observation is randomly assigned to a cluster
- Top right: Step 2.1, the centroids of the clusters (colored disks) are calculated. At first the centroids are almost completely overlapped due to the initial random assignment
- Bottom left: Step 2.2, each observation is assigned to the nearest centroid
- Bottom center: Step 2.1 is performed again to determine the new cluster centroids
- Bottom right: The results obtained after 10 iterations

Example: different starting values

- K-means can get stuck in local optima and not find the best solution
- It is important to run the algorithm several times, starting with a different random initial assignment each time

Details for the previous figure

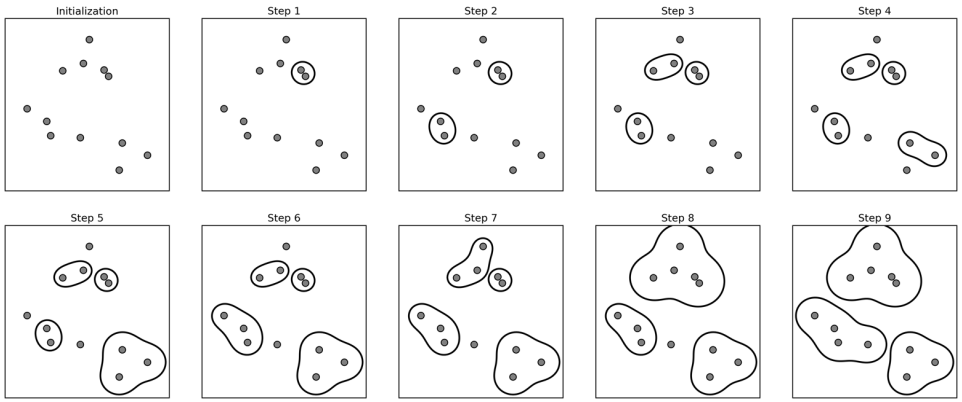
- K-means performed 6 times on data of the previous figure with K = 3, each time with a different random assignment of the observations (**step 1** of the algorithm)
- Each plot shows the value of the objective function
- Three different local optimal are obtained, one of which has the smallest value of the objective function and provides better separation between clusters
- Those labeled red all achieve the same best solution, with an objective function value of 235.8

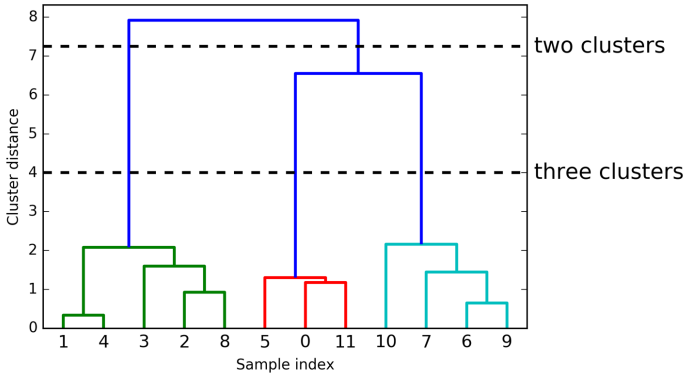
Hierarchical Clustering

- K-means clustering requires you to specify in advance the number of K clusters. This can be a disadvantage
- Hierarchical clustering is an alternative approach that does not require you to commit to a particular choice of K
- We describe bottom-up or agglomerative clustering
  - This is the most common type of hierarchical clustering and refers to the fact that a dendrogram is constructed starting from the leaves and combining clusters all the way to the root

Agglomerative clustering algorithm

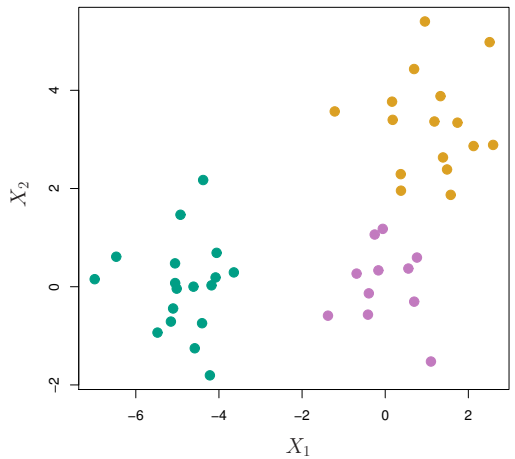
- Let each data point be a cluster
- Repeat
  - Merge the two closest clusters
  - Update the proximity between clusters
- Until only a single cluster remains
- The key operation is the computation of the proximity of two clusters
- Different approaches to defining the distance between clusters distinguish the different algorithms

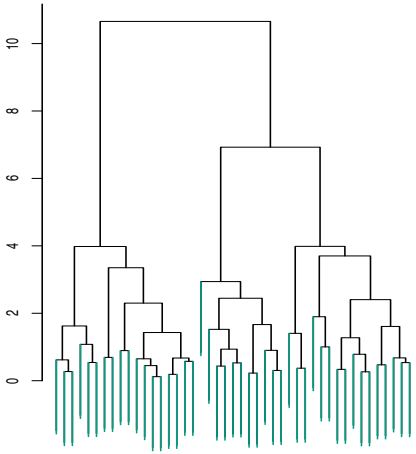




Example of Hierarchical clustering

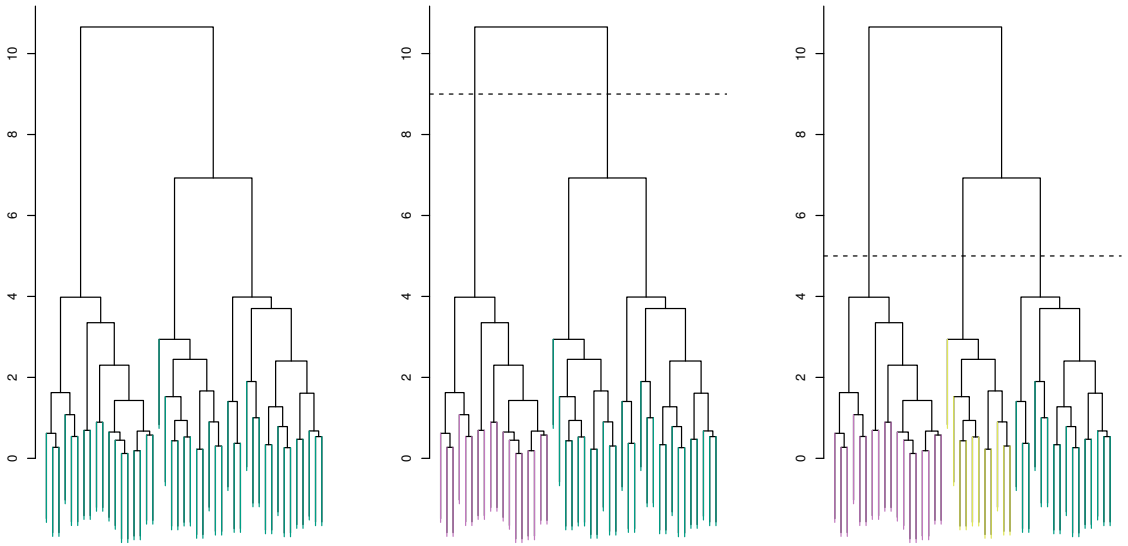
- 45 observations generated in two-dimensional space
- There are three distinct classes, shown in separate colors
- However, we will treat these class labels as unknown and try to group the observations in order to discover the classes from the data





Choice of clusters

- To choose the clusters we draw a line that crosses the dendrogram
- We can form any number of clusters depending on where we " cut "



1 Cluster **2 Clusters** **3 Clusters**

How do you read the dendrogram

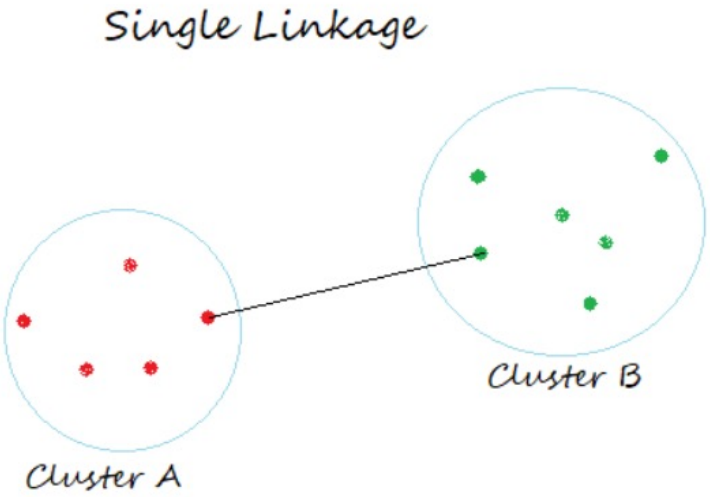
- Left: Dendrogram obtained from the hierarchical grouping of data from the previous slide
- Middle: Dendrogram from the left panel, cut at a height 9 (indicated by the dotted line). This cut results in two distinct clusters, shown in different colors
- Right: Dendrogram from the left panel, now cut to a height 5. This cut produces three distinct groups, shown in different colors
  - Note that the colors were not used in clustering, but are simply used for display purposes in this figure

How do we define dissimilarity?

- To implement hierarchical clustering, a question must be resolved
- How do we define the dissimilarity, or linkage, between the merged clusters (5,7) and 8, for instance?
- There are four different options
  - Complete linkage
  - Single linkage
  - Average linkage
  - Centroid linkage

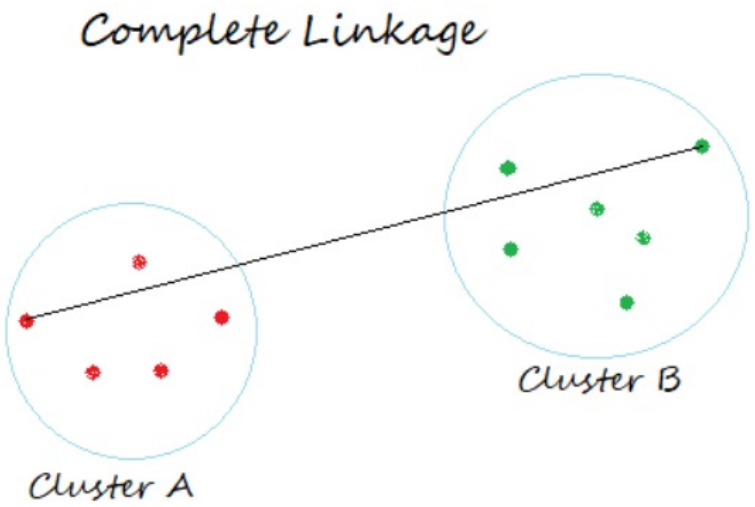
How do we define similarity?

- Single linkage
  - smallest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \min(t_{i,p}, t_{j,q})$



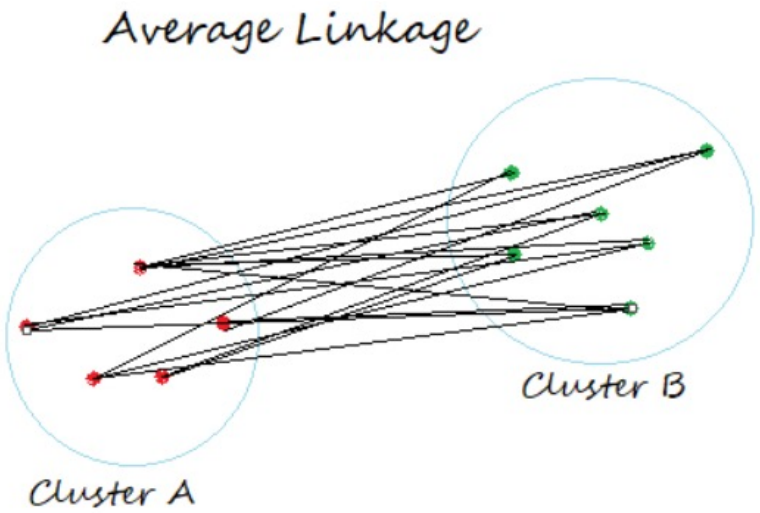
How do we define similarity?

- Complete linkage
  - largest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \max(t_{i,p}, t_{j,q})$



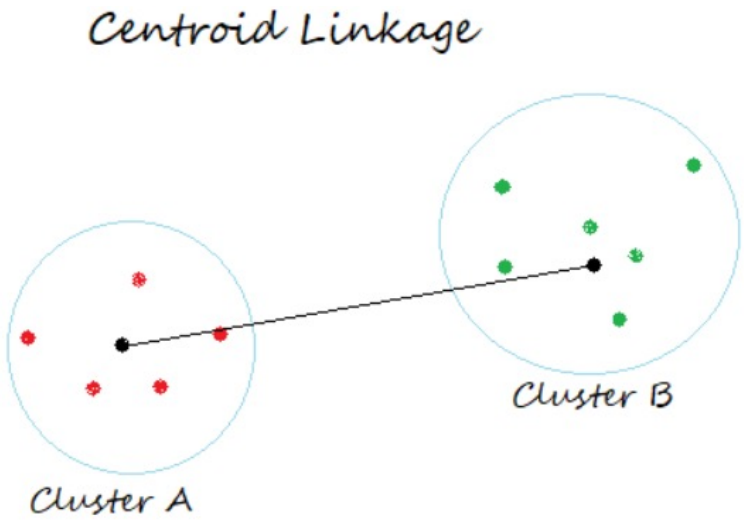
How do we define similarity?

- **Average** (or group average)
  - average distance between an element in one cluster and an element in the other, i.e., $d(Ci, Cj) = avg(d(t_{i,p}, t_{j,q}))$



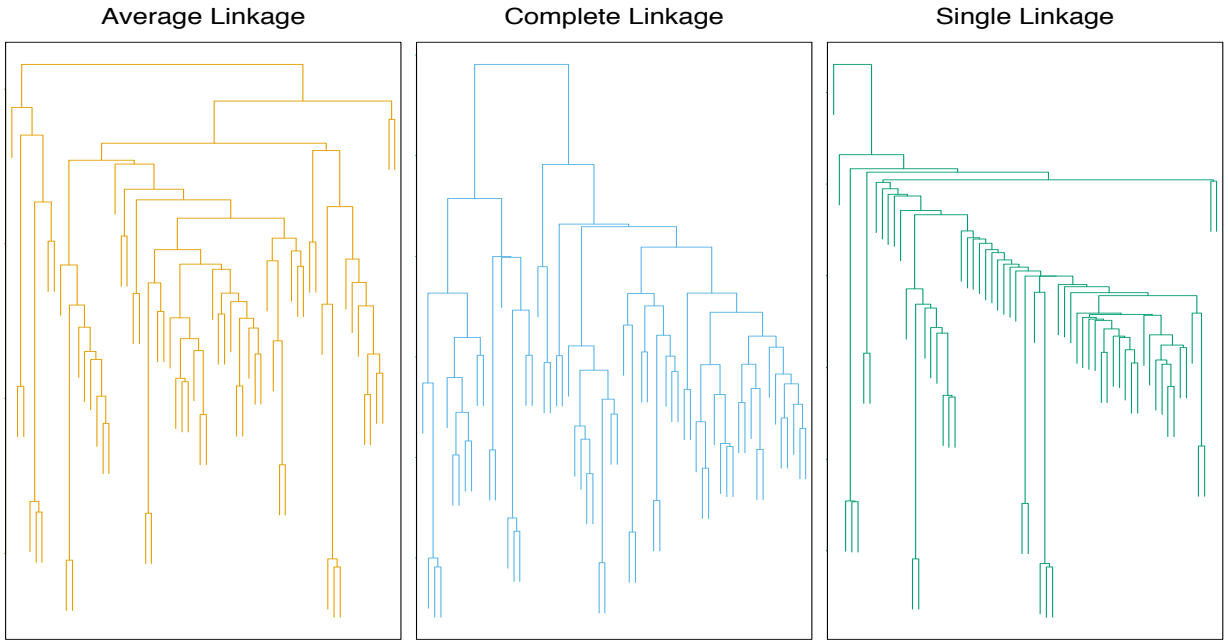
How do we define similarity?

- Centroid linkage
  - Distance between the centroids of two clusters, i.e., $d(C_i, C_j) = d(\mu_i, \mu_j)$
    - μ_i and μ_i are the centroids



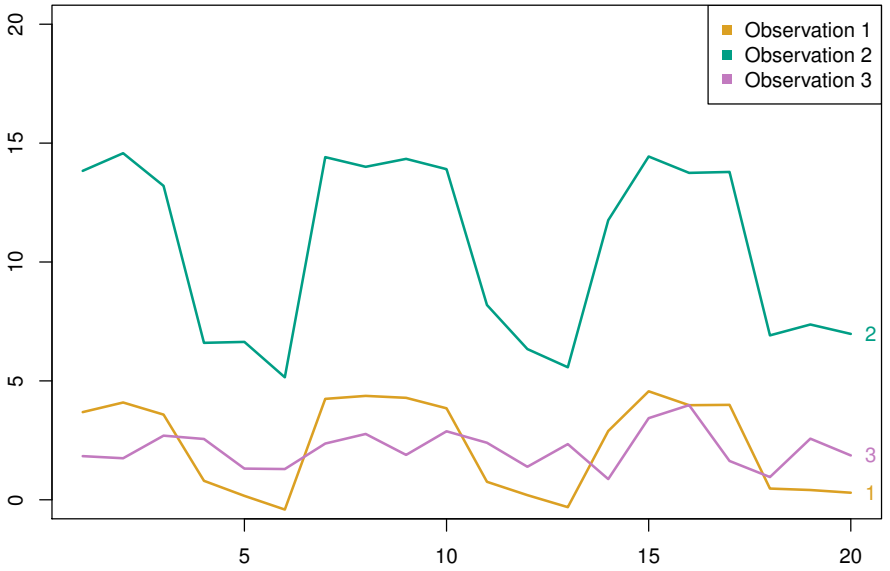
Linkage might be important

- We have three clustering results for the same data set
- The only difference is the linkage method, and the results are very different
- Complete and average linkages tend to build clusters of equal size
- Single linkage leads to extended clusters to which the individual leaves are merged one by one



Choice of dissimilarity measure

- Until now the Euclidean distance has been used
- An alternative is correlation-based distance, which considers two observations to be similar if their features are highly correlated
- In this example, we have 3 observations and p = 20 variables
  - In terms of Euclidean distance, 1 and 3 are similar
  - However, 1 and 2 are highly correlated, so they would be considered similar in terms of correlation measure



Distance measures

L_r-norm

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = (\sum_{i=1}^{n} |x_i - y_i|^r)^{1/r}$$

Euclidean distance (r=2)

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Manhattan distance (r=1)

$$d([x_1, x_2, \ldots, x_n], [y_1, y_2, \ldots, y_n]) = \sum_{i=1}^{n} |x_i - y_i|$$

L_∞-norm

$$d([x_1, x_2, \ldots, x_n], [y_1, y_2, \ldots, y_n]) = \max_{i=1}^{n} |x_i - y_i|$$

Cosine distance

- The cosine distance between x, y is the angle that the vectors to those points make

$$d(x,y) = \arccos \frac{\sum_1^n x_i y_i}{\sqrt{\sum_i^n x_i^2}\sqrt{\sum_i^n y_i^2}}$$

- This angle will be in the range 0 to 180 degrees, regardless of how many dimensions the space has.
- Example
  - given x = (1,2,-1) and y = (2,1,1) the angle between the two vectors is 60

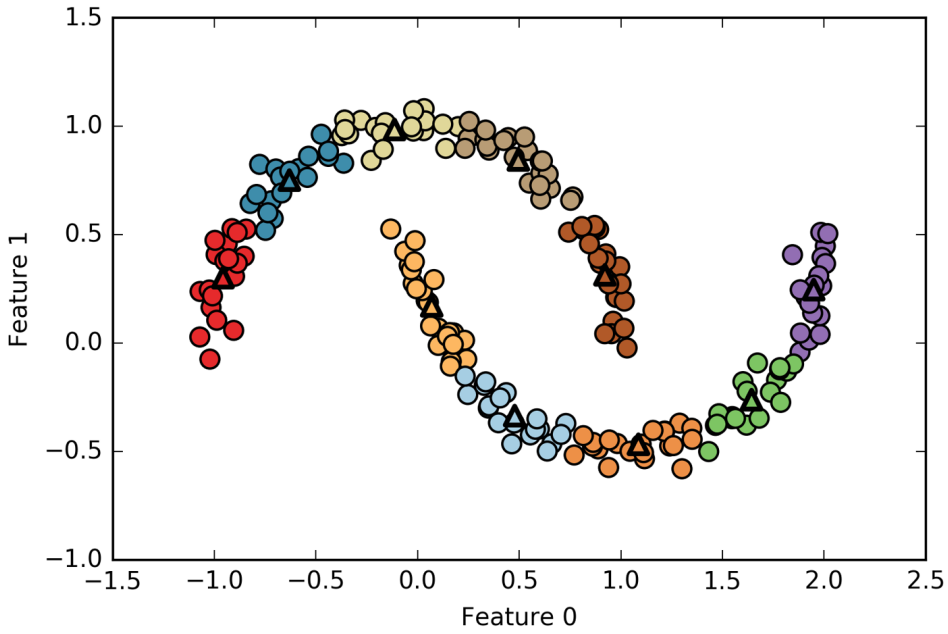
Jaccard Distance

- Jaccard distance is defined as
  - d(x,y) = 1- SIM(x,y)
  - SIM(x,y) = $\frac{x \cap y}{x \cup y}$ -> Jaccard similarity
- Can also be interpreted as the percentage of identical attributes

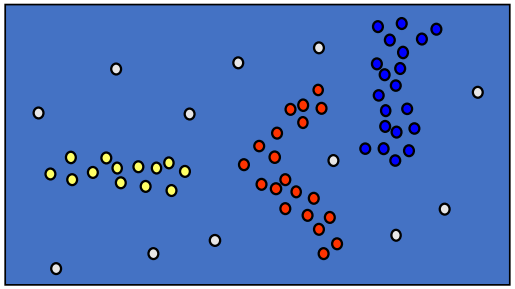
Practical considerations for clustering

- To cluster a set of data, several non-trivial decisions must be made
  - Should features be standardized first? that is, having the variables centered to have zero mean and standard deviation to one
  - In case of hierarchical clustering :
    - What dissimilarity measure should be used?
    - What kind of linkage should be used?
    - Where should we cut the dendrogram to get the clusters?
  - In case of K-means clustering :
    - How many clusters should we look for in the data?
- In practice, we try different choices and look for the one with the most useful or interpretable solution. There is no single correct answer!

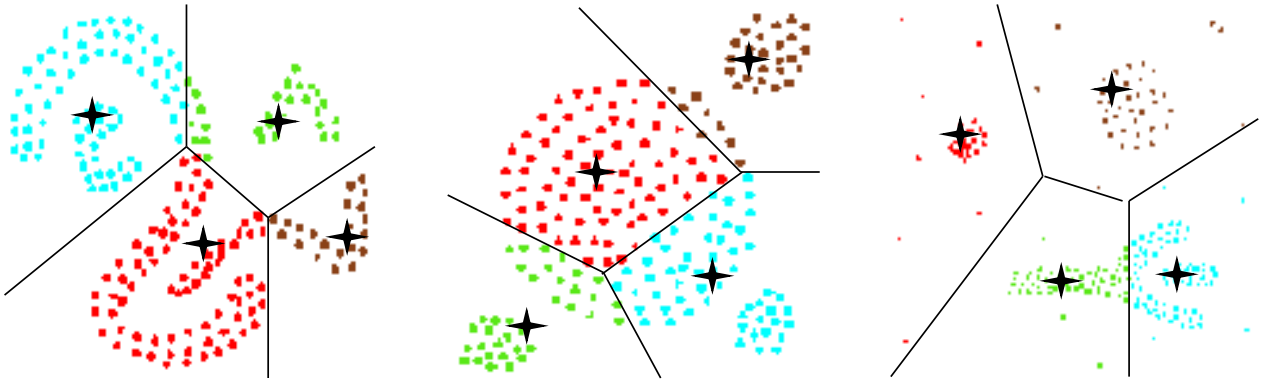
Limitations of k-means and agglomerative clustering



Density-Based Clustering



- Why Density-Based Clustering?
  - Basic Idea
    - Clusters are dense regions in the data space, separated by regions of lower object density



DBSCAN

- Density-Based Spatial Clustering of Applications with Noise
  - does not require the user to set the number of clusters *a priori*
  - can capture clusters of complex shapes, and
  - can identify points that are not part of any cluster
  - slower than k-means and agglomerative clustering but scales to relatively large datasets
- DBSCAN works by identifying points that are in "crowded" regions of the feature space, where many data points are close together
- These regions are referred to as *dense* regions in feature space
- The idea behind DBSCAN is that clusters form dense regions of data, separated by regions that are relatively empty

DBSCAN parameters

- Points that are within a dense region are called *core samples* (or core points)
  - Two parameters in DBSCAN
    - *eps* for the neighborhood of point p:
      $N(P) := \{Q \text{ in data set } D \mid dist(P, Q) \leq \text{eps}\}$
    - *min_samples* minimum number of points in the given neighbourhood N(P)
  - If there are at least **min_samples** data points within a distance of **eps** to P, that data point is classified as a core sample
  - Core samples that are closer to each other than the distance eps are put into the same cluster

DBSCAN algorithm

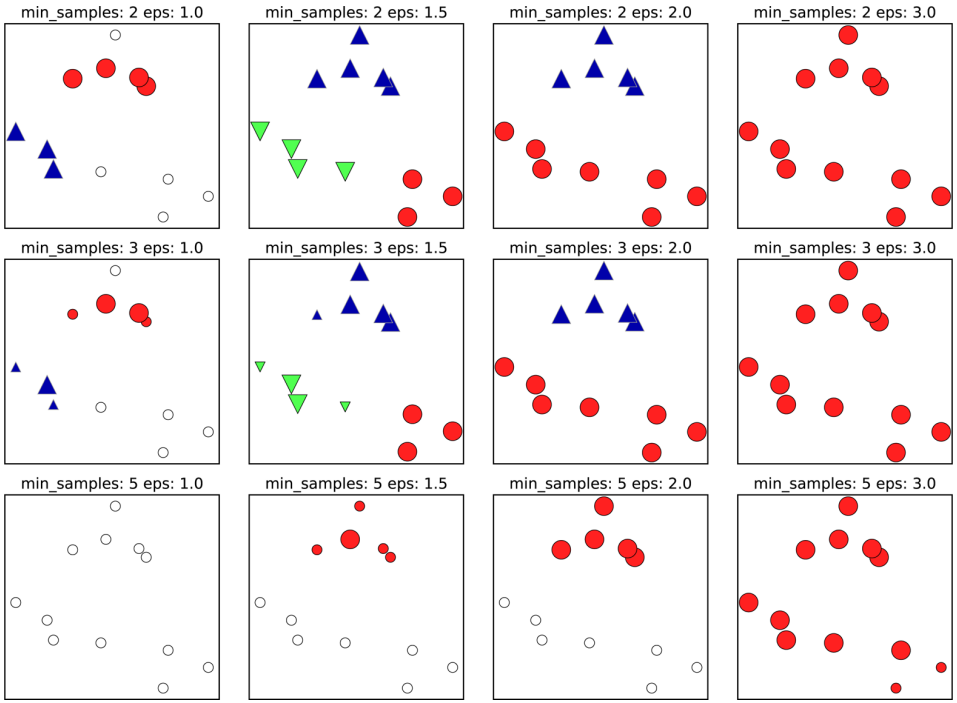
- Picks an arbitrary point, P, to start with
- Finds the set S of points Q with $dist(P, Q) \leq eps$
  - If $|S| < min\_samples$, P is labeled as *noise*
    - it doesn't belong to any cluster
  - If $|S| \geq min\_samples$, P is labeled a core sample and assigned a new cluster label
- All neighbors (within eps) of P are visited
  - If they have not been assigned a cluster yet, they are assigned the new cluster label that was just created
  - If they are core samples, their neighbors are visited in turn, and so on
  - The cluster grows until there are no more core samples within distance eps of the cluster
    - Then another point that hasn't yet been visited is picked, and the same procedure is repeated

DBSCAN algorithm

- Eventually, there are three kinds of points:
  - core points, points that are within distance eps of core points (called *boundary points*), and noise
- When the DBSCAN algorithm is run on a particular dataset multiple times, the clustering of the core points is always the same, and the same points will always be labeled as noise
- A boundary point might be neighbor to core samples of more than one cluster
  - The cluster membership of boundary points depends on the order in which points are visited. Usually there are only few boundary points, and this slight dependence on the order of points is not important

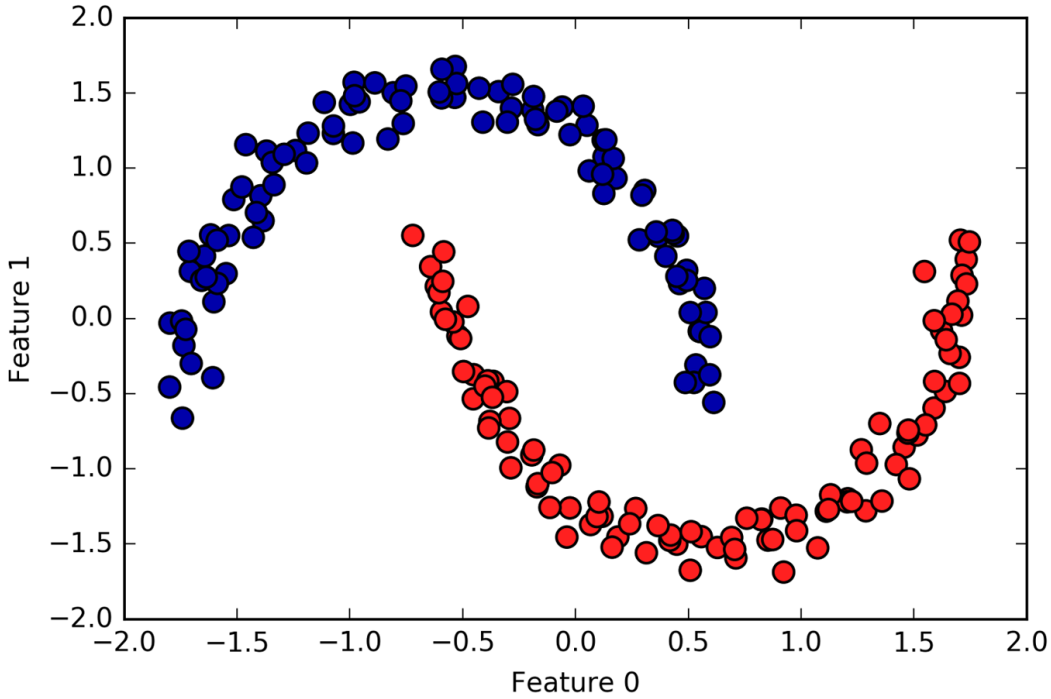
DBSCAN with different parameter settings

- Points that belong to clusters are solid
  - noise points are shown in white
- Core samples are shown as large markers, while boundary points are displayed as smaller markers
- Increasing eps more points will be included in a cluster
  - This makes clusters grow, but might also lead to multiple clusters joining into one
- Increasing min_samples fewer points will be core points, and more points will be labeled as noise



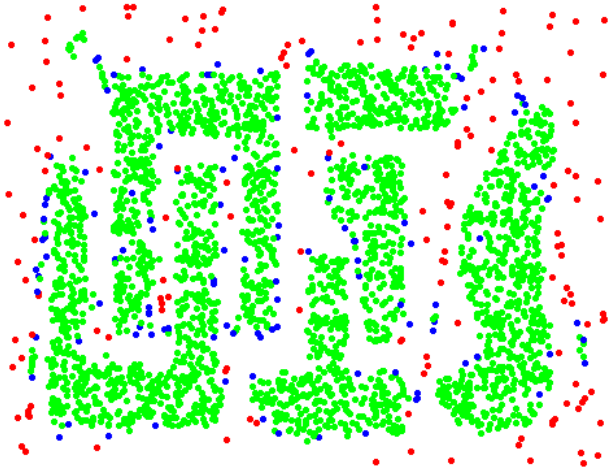
DBSCAN running result

- eps = 0.5
- min_samples =5



Yet another example

Original Points



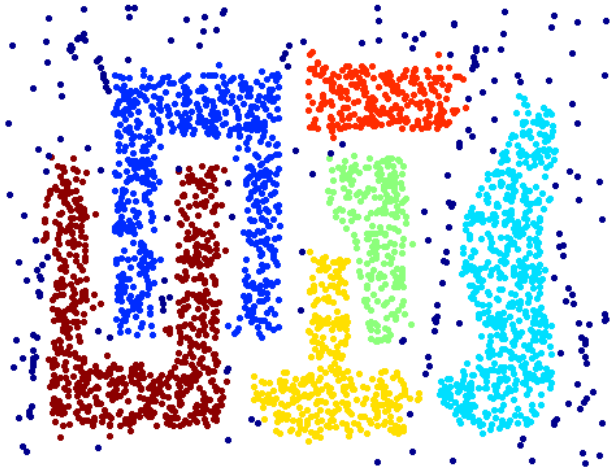
Point types: core, border and outliers

eps = 10, min_samples = 4

When DBSCAN Works Well

- Resistant to Noise
- Can handle clusters of different shapes and sizes

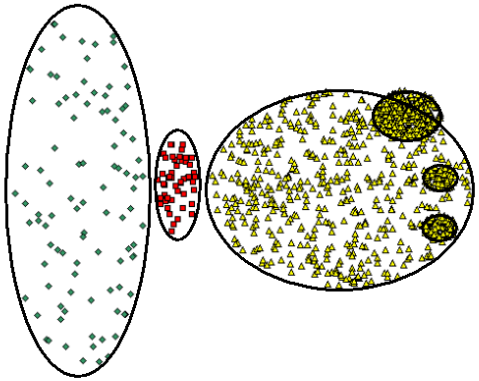
Original Points



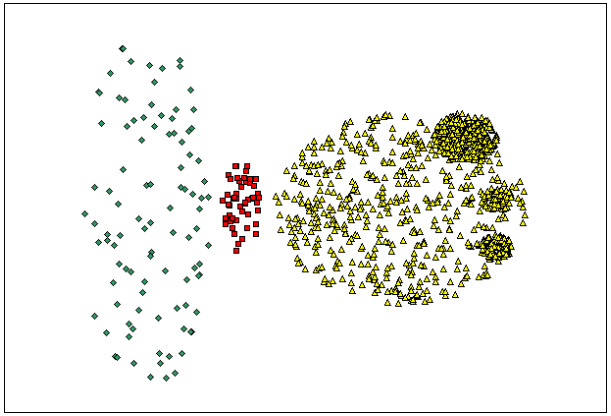
Clusters

When DBSCAN Does NOT Work Well

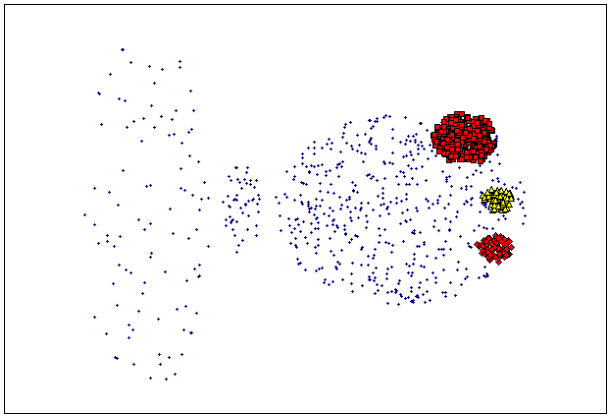
- Cannot handle Varying densities
- Sensitive to parameters



Original Points



(min_samples = 4, eps = 9.92)



(min_samples = 4, eps=9.75)

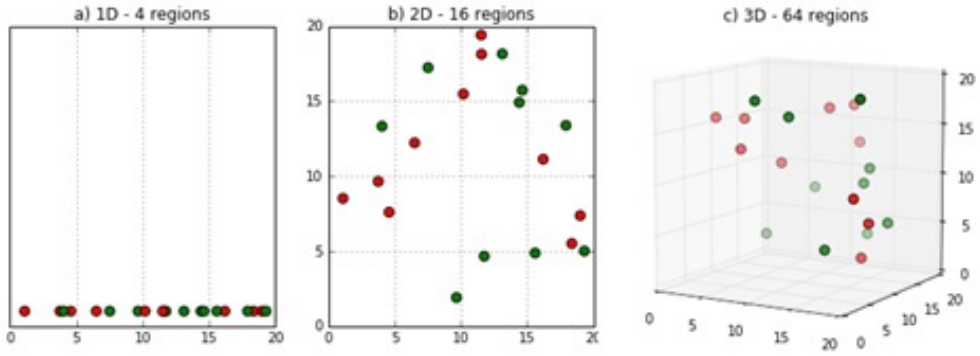
What is a Good clustering ?

- Very hard to assess how well an algorithm worked, and to compare outcomes between different algorithms
- A good clustering consists of high-quality clusters with
  - High intra-class similarity
  - Low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns
- Evaluation
  - Various measure of intra/inter cluster similarity
  - Manual inspection
  - Benchmarking on existing labels

Curse of dimensionality and clustering

- In high dimensions, almost all pairs of points are equally far away from one another
- Data in only one dimension is relatively packed
- Adding a dimension "stretch" the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart – high dimensional data is extremely sparse
- Distance measure becomes meaningless – due to equidistance

Data gets increasingly sparse



Analysis results degrade

How do we assess clustering results ?

- Several validity measures
  - Clustering evaluation
    - Assess the goodness or quality of the clustering
  - Clustering stability
    - Sensitivity of the clustering result to various algorithmic parameters

Validation measures

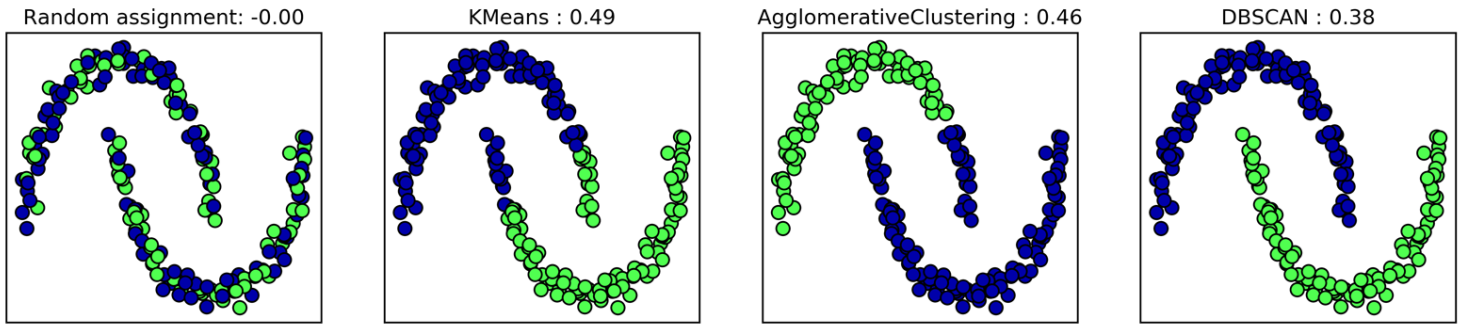
- Based on the notions of intra-cluster similarity or compactness contrasted with the notions of inter-cluster separation
- They typically propose a trade-off to maximizing these two competing measures
- They are computed from the distance (or proximity) matrix
- The internal measures are based on various functions over the intra-cluster and inter-cluster weights

Validation measures

- Many indexes exist
  - Dunn
  - Davies-Bouldin
  - Silhouette
  - …

Objectively comparing clustering is hard

- Silhouette score for comparisons
  - DBSCAN's result more intuitive but lower Silhouette index



Relative measures

- Compare different clustering obtained by varying different parameters for the same algorithm, e.g., the number of clusters k

Within-cluster sum of squares

$$\mathrm{WSS}(C) = \sum_{i=1}^{k} \sum_{x_j \in C_i} ||x_j - \mu_i||^2$$

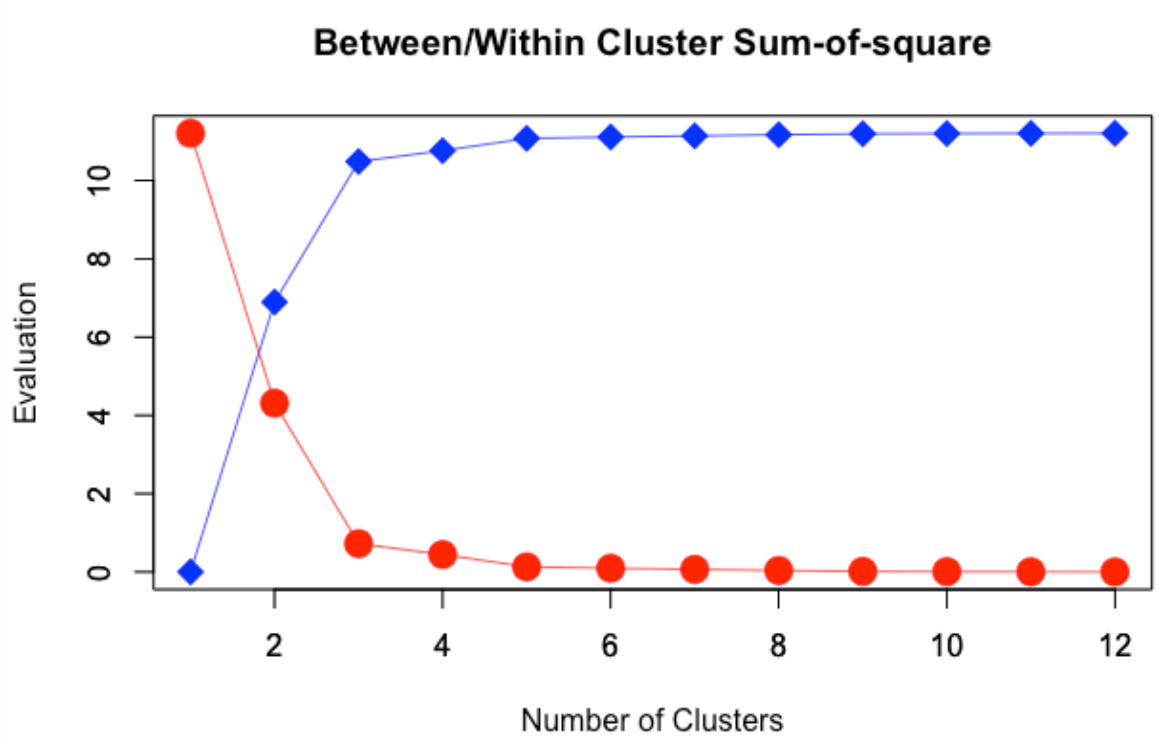
where μ_i is the centroid of cluster C_i (in case of Euclidean spaces)

Between-cluster sum of squares

$$\mathrm{BSS}(C) = \sum_{i=1}^{k} |C_i| \cdot ||\mu - \mu_i||^2$$

where μ is the centroid of the whole dataset

Knee/Elbow Analysis of Clustering



Summarizing …

- We looked at three clustering algorithms: *k*-means, hierarchical (agglomerative), and DBSCAN
- All three have a way of controlling the granularity of clustering
  - *k*-means and agglomerative clustering allow you to specify the number of desired clusters
  - DBSCAN lets you define proximity using the eps parameter, which indirectly influences cluster size
- All three methods can be used on large, real-world datasets, and allow for clustering into many clusters
- Each of the algorithms has somewhat different strengths
  - *k*-means allows for a characterization of the clusters using the cluster means
  - Agglomerative clustering can provide a whole hierarchy of possible partitions of the data, which can be easily inspected via dendrograms
  - DBSCAN allows for the detection of "noise points" that are not assigned to any cluster, and automatically determine the number of clusters
    - It allows for complex cluster shapes, and might produce clusters of very differing size, which can be a strength or a weakness