# Introduction to Machine Learning

**Lecture 8 - Unsupervised Learning 1**

**Guang Bing Yang, PhD**





## Unsupervised Learning

- Introduction to unsupervised learning.
- Introduction to clustering.
- K-means algorithm





## What is Unsupervised Learning

- It is one of three categories of machine learning.
- It is about to find underlying patterns in data and is often used in exploratory data analysis.
- The data used in unsupervised learning have no labels, only X. $D = \{(x_i,)\}_{i=1}^{N}$.
- It focuses on the data's features.
- The goal of the unsupervised learning is to find relationships or patterns within the data;
- Then group or cluster data points based on the input data alone.
- Not like the Supervised learning which makes predictions, the unsupervised learning has no prediction tasks but has tasks to discover patterns or group data into clusters.





## Examples of Unsupervised Learning Applications

- Examples of unsupervised learning application
  - Image segmentation
  - Data compression
  - Data mining -- discovery of association rules
  - Density estimation (e.g. texture synthesis), image compression
  - level set estimation
  - clustering/mode finding e.g. spike sorting; algorithm Mixture of Gaussians
  - metric learning e.g. 3D visualization; algorithm multidimensional scaling
  - feature extraction - or representation learning, e.g. whitening; algorithm principle component analysis.

# Unsupervised Learning : Clustering

- Clustering is the most common applications of unsupervised learning.
- It is a technique for finding similarity groups in data, called clusters. For instance,
  - Grouping data points that are similar to (close to) each other in one cluster;
  - Splitting (un-grouping) data points that are very different (far away) from each other into different clusters.
  - Clustering is an unsupervised learning task as no class values are given, unlike the case in supervised learning.
  - Association rule mining is also unsupervised.


# Unsupervised Learning : Clustering

- An illustration

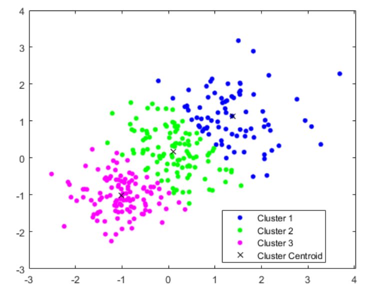



Image copied from https://www.mathworks.com/


# Unsupervised Learning : Clustering

- What is clustering useful in real-life?
- For example:
  - groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.
  - personalize tailor-made is very expensive and the amount of clothes is very limit.
  - Another example in marketing business, segment customers according to their similarities to do targeted marketing, etc.
  - Another example, like building topic hierarchy for organizing a collection of text documents, like libraries, etc.
  - Recommendation systems, such as giving you better Amazon suggestions


# Unsupervised Learning : Clustering

- Categories of clustering
  - A clustering algorithm
    - Partitional clustering — each data point in a dataset can only belong to one cluster
    - Hierarchical clustering — clusters within clusters
- 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.

# Unsupervised Learning : Clustering

- The theorem of clustering is;
  - data points that are in the same group should have similar properties and/or features, while those in different groups should have highly dissimilar properties and/or features. The similarity between points is usually quantified by a distance metric based on some sort of feature variable set.

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 19, 2021 



# Clustering: K-means

- K-means is a partition clustering algorithm
- In k-means clustering, the goal is to partition, or divide, the data into a predetermined value for K, the number of clusters.
- Each data point will fall into only one cluster of the K clusters, and therefore the clusters will not overlap like they would in hierarchical clustering.
- Given a set of data points in N samples as: $\{x_1, x_2, \ldots, x_n\}$,
  - where $x_{i1}, \ldots x_{iD}$ is a D-dimensional vector in a real-valued space $X \in R^D$.
- The k-means algorithm partitions the given data into k clusters.
  - Each cluster has a cluster center, called centroid, or cluster prototype $\mu_k$.
  - k is known.

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 19, 2021 



# Clustering: K-means Algorithm

- Define a binary indicator variable, use 1-of-K coding scheme
  - $r_{nk} \in \{0,1\}$
  - $r_{nk} = 1, \;\; r_{nj} = 0 \;\; \forall j \neq k$
  - Distortion measure (objective function): $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} || x_n - \mu_k ||^2$
- Given k, the k-means algorithm works as follows:
  1. Randomly choose k data points (seeds) to be the initial centroids, cluster centres
  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).

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 19, 2021 



# K-means Clustering: Expectation Maximization

- Find values for $\{r_{nk}\}$ and $\{\mu_k\}$ to minimize:
  - $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} || x_n - \mu_k ||^2$
- Iterative procedure:
  - E-step:
    - Minimize J w.r.t. $r_{nk}$, keep $\mu_k$ fixed.
    - $r_{nk} = 1 \;$ if $\; k = \text{argmin}_j || x_n - \mu_k ||^2$ or, $r_{nk} = 0 \;$ otherwise
  - M-step:
    - minimize J w.r.t. $\mu_k$, keep $r_{nk}$ fixed
    - $2 \sum_{n=1}^{N} r_{nk}(x_n - \mu_k) = 0$, then $\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$

yguangbing@gmail.com, Guang.B@chula.ac.th  Mar 19, 2021 

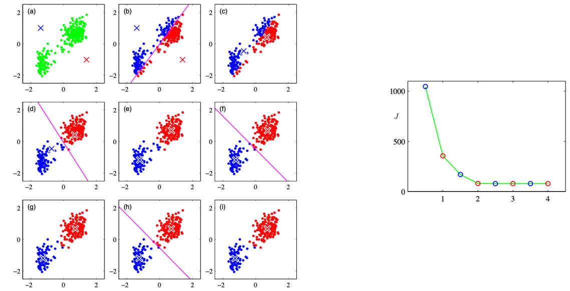
# K-means Clustering: Expectation Maximization

- Stopping/convergence criterion
  - no (or minimum) re-assignments of data points to different clusters,
  - no (or minimum) change of centroids, or
  - minimum decrease in the sum of squared error (SSE) — J function,
    - $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathrm{dist}(x_n - \mu_k)$
  - $k_j$ is the jth cluster, $\mu_k$ is the centroid of cluster $k_j$ (the mean vector of all the data points in $k_j$), and $\mathrm{dist}(x_n, \mu_k)$is the distance between data point x and centroid $\mu_k$.





# K-means Clustering: Example



K-means clustering example in OldFaithful data

- Each E or M step reduces the value of the objective function J
- Convergence to a global or local maximum





**K-means clustering**

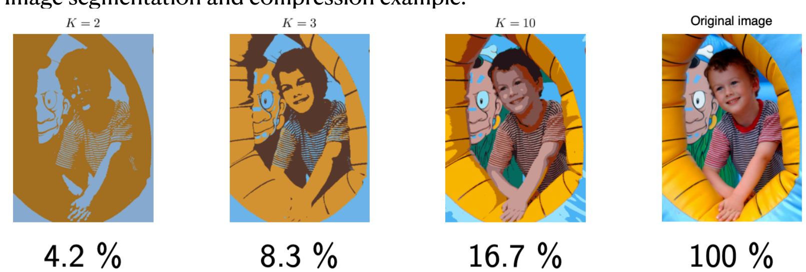
- Direct implementation of K-means can be slow.
- Online version: $\mu_k^{new} = \mu_k^{old} + \eta(x_n - \mu_k^{old})$.
- K-mediods, general distortion measure: $\tilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} V(x_n, \mu_k)$,
  - where $v(\,.\,,\,.\,)$ is any kind of dissimilarity measure.





**Examples of K-means clustering applications**

- Image segmentation and compression example:




Cited from the book: Christopher M. Bishop, Pattern Recognition and Machine Learning, 2006

## An example distance function

- In the **Euclidean space**, the mean of a cluster is computed with:
  - $\mu_j = \frac{1}{|K_j|} \sum_{x_i \in K_j} x_i,$
  - where $|K_j|$ is the number of data points in cluster $K_j$
  - The distance from one data point $x_i$ to a mean (centroid) $\mu_j$ is computed with
    - $\text{dist}(x_i, \mu_j) = ||x_i - \mu_j|| = \sqrt{(x_{i1} - \mu_{j1})^2 + (x_{i2} - \mu_{j2})^2 + \ldots + (x_{iD} - \mu_{jD})^2}$



## K-means Clustering: 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.



## K-means Clustering: Weaknesses

- The algorithm is only applicable if the mean is defined.
- For categorical data, k-mode - the centroid is represented by most frequent values.
- The user needs to specify k.
- The algorithm is sensitive to outliers
- Outliers are data points that are very far away from other data points.
- Outliers could be errors in the data recording or some special data points with very different values.



## K-means Clustering: Weaknesses

- To deal with outliers
  - Remove some data points in the clustering process that are much further away from the centroids than other data points.
  - Perform random sampling. In sampling the chance of selecting an outlier is very small because of a small set of data points be selected.
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

# Clustering: Represent clusters

- Use the centroid of each cluster to represent the cluster.
  - compute the radius and
  - standard deviation of the cluster to determine its spread in each dimension
  - the centroid representation alone works well if the clusters are of the hyper-spherical shape.
  - if clusters are elongated or are of other shapes, centroids are not sufficient

yguangbing@gmail.com, Guang.B@chula.ac.th




# Clustering: Represent clusters

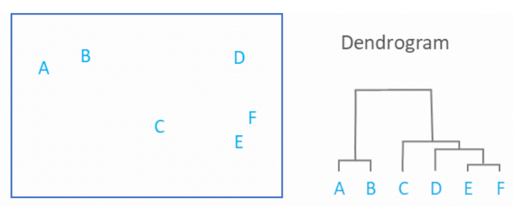
- Use frequent values to represent cluster
  - This method is mainly for clustering of categorical data (e.g., k-modes clustering).
  - Main method used in text clustering, where a small set of frequent words in each cluster is selected to represent the cluster.

yguangbing@gmail.com, Guang.B@chula.ac.th




# Hierarchical Clustering

- Hierarchical clustering finds clusters within clusters by a system of hierarchies.
- Not like the partition clustering, every data points can belong to multiple clusters, some clusters will contain smaller clusters within it.
- This hierarchy system can be organized as a tree diagram.




Hierarchical Clustering Dendrogram —copied from displayr.com

yguangbing@gmail.com, Guang.B@chula.ac.th




# Hierarchical Clustering

- This hierarchy system can be organized as a tree diagram.
  - Agglomerative algorithms find clusters with a bottom-up approach. These algorithms start with each data point as a cluster, then progressively “zoom out” and combine smaller clusters into larger clusters.
  - Divisive algorithms take the opposite approach: top-down. Divisive algorithms start out by looking at the entire dataset as one cluster, then “zooming in” to divide the dataset into smaller clusters.
- Unlike k-means clustering, with hierarchical clustering, the number of clusters in unknown beforehand.

yguangbing@gmail.com, Guang.B@chula.ac.th

# Measure the distance of two clusters

- A few ways to measure distances of two clusters. Results in different variations of the algorithm.
  1. **Single link** -- The distance between two clusters is the distance between two closest data points in the two clusters, one data point from each cluster
  2. **Complete link** -- 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
  3. **Average link** -- A compromise between single and complete link. The distance between two clusters is the average distance of all pairwise distances between the data points in two cluster
  4. **Centroids** -- In this method, the distance between two clusters is the distance between their centroids





# Distance Functions

- For Key to clustering. “similarity” and “dissimilarity” can also be commonly used terms.
- 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: $\text{dist}(x_i, x_j)$, where $x_i$ and $x_j$ are data points (vectors)
- They are special cases of Minkowski distance. h is positive integer.
- $\text{dist}(x_i, x_j) = ((x_{i1} - x_{j1})^h + (x_{i2} - x_{j2})^h + \ldots + (x_{iD} - x_{jD})^h)^{1/h}$





## **Euclidean distance** and **Manhattan distance**

- If h = 2, it is the Euclidean distance:
  - $\text{dist}(x_i, x_j) = \sqrt{((x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{iD} - x_{jD})^2)},$
- If h = 1, it is the Manhattan distance:
  - $\text{dist}(x_i, x_j) = |(x_{i1} - x_{j1})| + |(x_{i2} - x_{j2})| + \ldots + |(x_{iD} - x_{jD})|$
- Weighted Euclidean distance
  - $\text{dist}(x_i, x_j) = \sqrt{w_1((x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \ldots + w_D(x_{iD} - x_{jD})^2)}$

### Squared distance and Chebyshev distance

- Squared distance and Chebyshev distance
  - Squared Euclidean distance: to place progressively greater weight on data points that are further apar
  - $\text{dist}(x_i, x_j) = ((x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{iD} - x_{jD})^2)$
  - Chebyshev distance: one wants to define two data points as "different" if they are different on any one of the attributes.
  - $\text{dist}(x_i, x_j) = max(|(x_{i1} - x_{j1})| + |(x_{i2} - x_{j2})| + \ldots + |(x_{iD} - x_{jD})|)$





### Distance function for binary and nominal attributes

- Use 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 ith and jth data points be xi and xj (vectors)
- 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
  - $\text{dist}(x_i, x_j) = \dfrac{b + c}{a + b + c + d}$





### Distance function for binary and 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 xi and xj, let the number of attributes be r, and the number of values that match in xi and xj be q.
  - $\text{dist}(x_i, x_j) = \dfrac{r - q}{r}$





### Recap

- Unsupervised learning is one of three categories of machine learning
- Clustering is the main focus of the unsupervised learning.
- Data mining is another technique of the unsupervised learning.
- Clustering is has along history and still active
  - There are a huge number of clustering algorithms
  - More are still coming every year.
- We only introduced several main algorithms.
  - partitional-clustering and hierarchical clustering
  - K-means is a partitional-clustering

## Recap

- The 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.





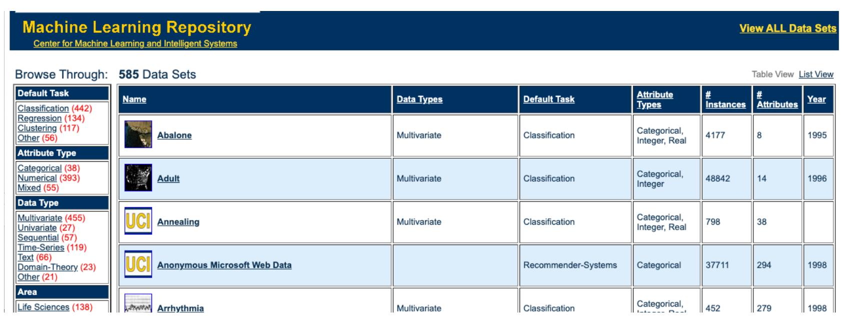
## Questions?

Datasets can be used in your project:
UCI Machine Learning Repository
https://archive.ics.uci.edu/ml/datasets.php





Datasets can be used in your project:
UCI Machine Learning Repository
https://archive.ics.uci.edu/ml/datasets.php



Machine Learning Repository
Center for Machine Learning and Intelligent Systems
View ALL Data Sets

Browse Through: 585 Data Sets
Table View List View

Default Task
Classification (442)
Regression (134)
Clustering (117)
Other (56)
Attribute Type
Categorical (38)
Numerical (393)
Mixed (55)
Data Type
Multivariate (455)
Univariate (27)
Sequential (57)
Time-Series (119)
Text (66)
Domain-Theory (23)
Other (21)
Area
Life Sciences (138)

| Name | Data Types | Default Task | Attribute Types | # Instances | # Attributes | Year |
|---|---|---|---|---|---|---|
| Abalone | Multivariate | Classification | Categorical, Integer, Real | 4177 | 8 | 1995 |
| Adult | Multivariate | Classification | Categorical, Integer | 48842 | 14 | 1996 |
| Annealing | Multivariate | Classification | Categorical, Integer, Real | 798 | 38 | |
| Anonymous Microsoft Web Data | | Recommender-Systems | Categorical | 37711 | 294 | 1998 |
| Arrhythmia | Multivariate | Classification | Categorical, [illegible] | 452 | 279 | 1998 |

UCI Machine Learning Repository