Machine Learning (CE 40477)

Fall 2024

Ali Sharifi-Zarchi

CE Department
Sharif University of Technology

October 13, 2024

1. Unsupervised Learning Overview
2. K-Means
3. Clustering
4. Conclusion

Unsupervised Learning

Unsupervised Learning involves working with **unlabeled data**, where the goal is to **infer the natural structure** present within a set of data points.

- Learning from unlabeled data.
- Two of the most common tasks:
  - **Clustering**: Grouping data points into clusters based on similarity towards user need.
  - **Dimensionality Reduction**: Reducing the number of features under consideration and keeping (perhaps approximately) the most informative features.

Music Recommendation Systems

- When you like a song you probably like other "similar" songs.
- Fun little exercise to build a simple system, after finishing this chapter.

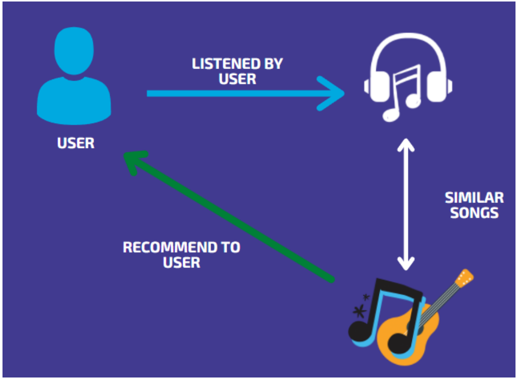
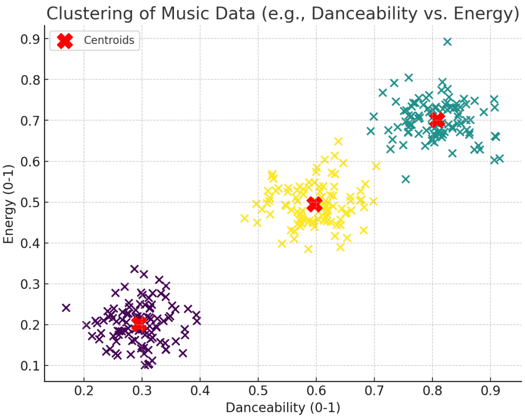


Figure adapted from machinelearninggeek.com

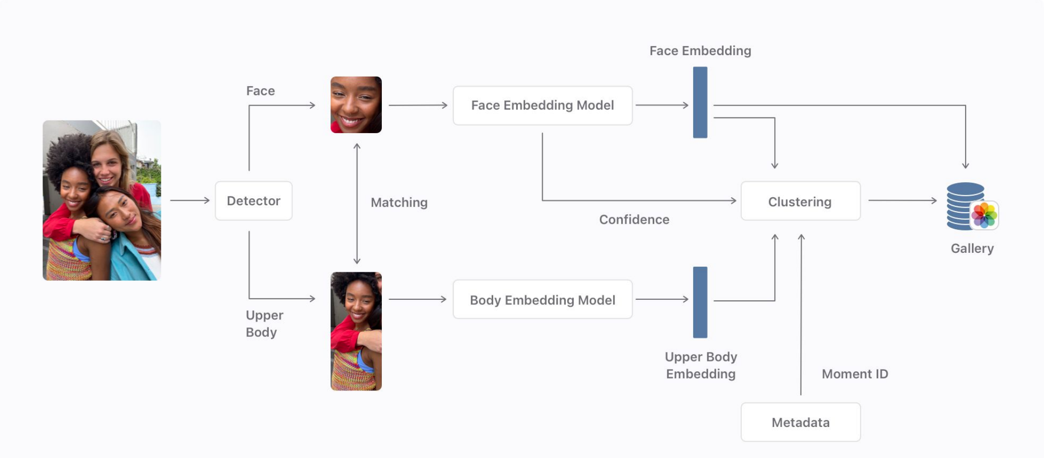
Music Recommendation Systems

- When you like a song you probably like other "similar" songs.
- Fun little exercise to build a simple system, after finishing this chapter.



Organizing Photos on Smartphones

- All pictures with that one friend
- All pictures where you looked "cool"



K-Means overview

- A popular **clustering** algorithm
- **Similar** data in the same cluster.
- K-Means suggests an **iterative** process to find these centers.

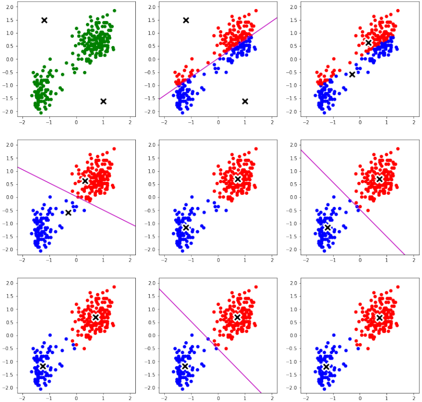


Figure From mlbhanuyerra.github.io.

Problem Intuition

- One of the most straightforward tasks we can perform on a data set without labels.
- finding groups of data in our dataset which are "similar" to one another –**clusters**.
- How many cluster? Can we cluster new unseen data? What is similar data ?

Problem definition

- Formally: We have $X_{\text{train}} = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\} \subseteq \mathbb{R}^d$
- Assume we know there are K clusters, or we want K clusters.
- We are learning:
  1. a function or mapping $f : \mathbb{R}^d \to \{1, 2, \ldots, K\}$ that assigns a cluster to each data point.
  2. a set of K prototypes $\mu = \{\mu_1, \mu_2, \ldots, \mu_K\} \subseteq \mathbb{R}^d$ as the **cluster representatives**.
- data assigned to the same $i \in \{1, 2, \ldots, K\}$ are in the same cluster i.

Objective Function

- Create objective function like the loss we had before.
- We want data in the same cluster to be closer and data from different clusters to be further. more on this later.
- in K-Means, this is expressed as:

$$\sum_{\mathbf{x} \in X_{\text{train}}} ||x - \mu_{f(x)}||^2$$

Objective Function (cont.)

- We can express f by defining $r_k(\mathbf{x}) = 1$ if $f(\mathbf{x}) = k$ and 0 otherwise, we can write this objective as below:

$$J = \sum_{\mathbf{x} \in X_{\text{train}}} \sum_{k=1}^{K} r_k(x) ||x - \mu_k||^2$$

- called*distortion measure.*
- chose f and μ to minimize this.
- Its NP-hard. what does K-Means suggest ?

Observation

- If we fix the set of **centroids** or representatives μ, we could minimize each term by **assigning**:

$$f(x) := argmin_k \, ||x - \mu_k||^2$$

.

Observation (cont.)

If we fix the assignments f we can optimize for μ by taking the derivative as:

$$\frac{\partial J}{\partial \mu_k} = 0 \implies 2\sum_{i=1}^{N} r_k(x_i)\left(x_i - \mu_k\right) = 0$$

and **updating** μ as:

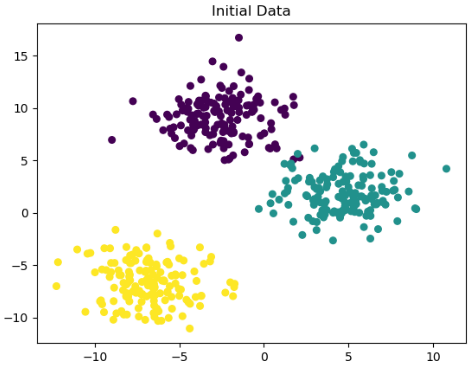
$$\mu_k = \frac{\sum_{i=1}^{N} r_k(x_i) x_i}{\sum_{i=1}^{N} r_k(x_i)}$$

K-Means Process

- K-Means uses an iterative process that:
  1. **Assigns** each point to the **nearest** centroid. Optimizing for f.
  2. **Updates** each centroid as the **mean** of the points in its cluster. Optimizing for μ.

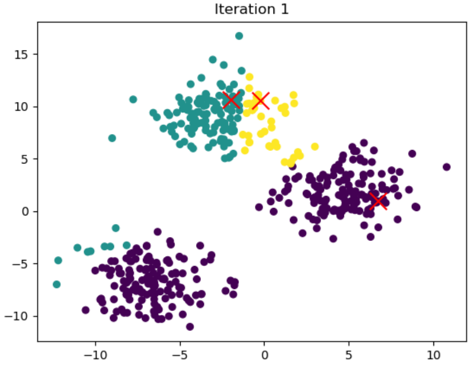
K-Means in action

original blobs

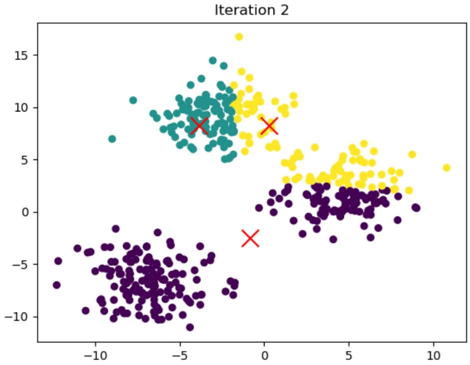


K-Means in action

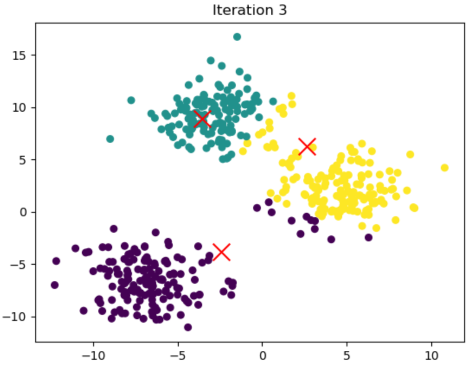
random initialization



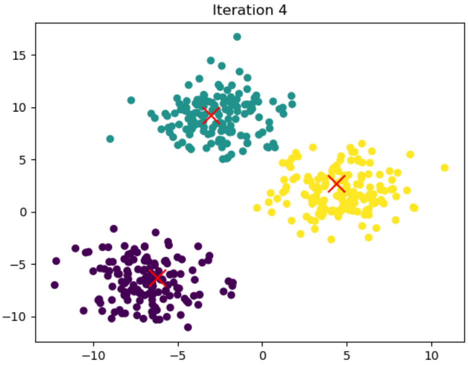
K-Means in action (cont.)



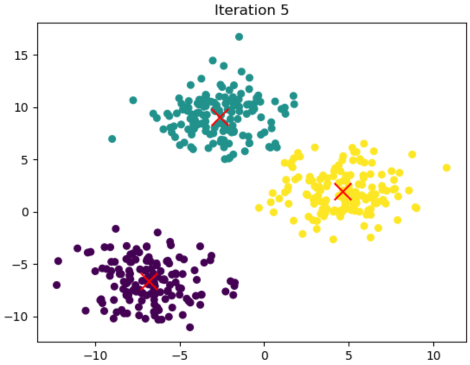
K-Means in action (cont.)



K-Means in action (cont.)



K-Means in action (cont.)



K-Means Algorithm

K-Means Algorithm (cont.)

Algorithm Overview

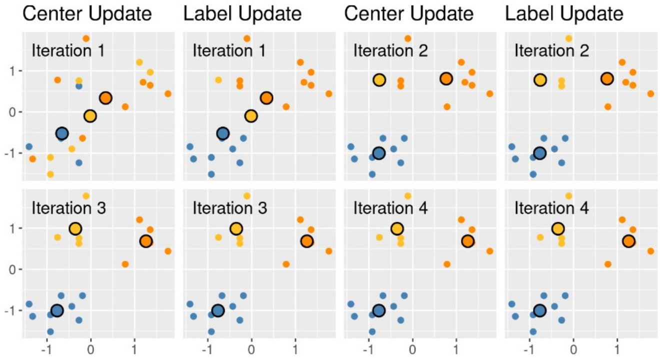


Figure 1: picture from datasciencebook.ca

Convergence

- How do we know K-Means will converge in a finite number of steps ?

Convergence (cont.)

- In Assignment step:
  - we optimize J with respect to $r_k(x)$.
  - In this step J is a linear combination of $r_k(x)$.
  - We need each x to be at least in some cluster and terms involving different xs are independent.
  - So for each x we chose one of the the K distance expressions that is the minimum. i.e.

$$r_k(x) = \begin{cases} 1 & k = \text{argmin}_j ||x - \mu_j||_2^2 \\ 0 & O.W \end{cases}$$

  - This will definitely not decrease J.

Convergence (cont.)

- Now with r_ks fixed, J is a quadratic function of μ_k (like SSE) and by taking derivative we can minimize as:

$$\frac{\partial J}{\partial \mu_k} = 0 \implies 2\sum_{i=1}^{N} r_k(x_i)\left(x_i - \mu_k\right) = 0$$

then we set:

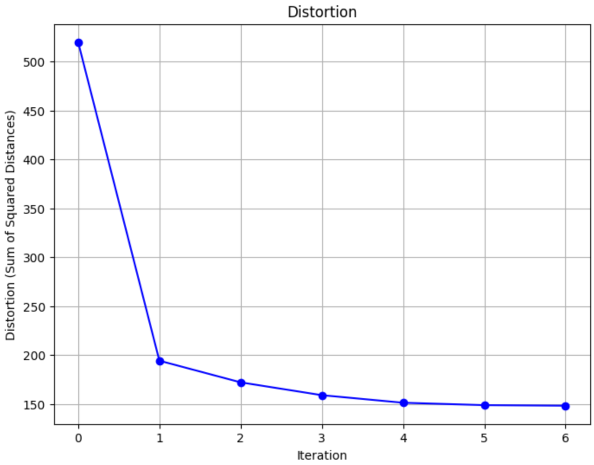
$$\mu_k = \frac{\sum_{i=1}^{N} r_k(x_i) x_i}{\sum_{i=1}^{N} r_k(x_i)}$$

- This will also definitley not increase J.

Convergence (cont.)

- We know each step will not increase the J objective from its current value.
- Also and J is non-negative, and there are a finite number of partitions so there is a minimum.
- Therefore we must converge at some point, where the J does not decrease anymore.
- The convergence properties of the K-means algorithm were studied by MacQueen (1967).

K-Means convergence (cont.)



Optional Adventure

Each Assignment and Updating step in K-Means corresponds respectively to the E (expectation) and M (maximization) steps of the EM algorithm.

One can prove that k-means is equivalent to running EM on a particular Naive Bayes Model.

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.

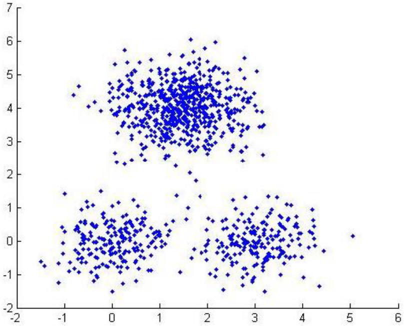
Some Issues

- k-Means always converges. What could go wrong ?
- K-means algorithm is a **heuristic**
- It requires initial centroids, and the choice is important. It could affect the t in $O(tkn)$.

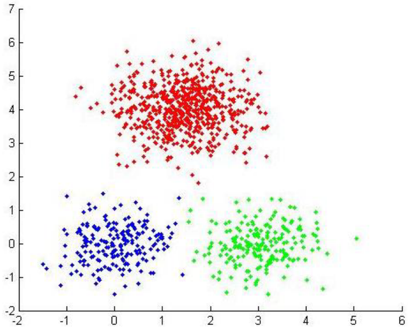
Local optimum

- The algorithm finds a local Minimum but it does not guarantee global minimum.
- This is highly affected by the initialization.
- Whats the solution ? some suggestions are:
  - variance-based split / merge
  - Random centers from the data points with Multiple runs and select the best ones.
  - initialization heuristics (k-means++ , Furthest Traversal)
  - Initializing with the suggested results of another method

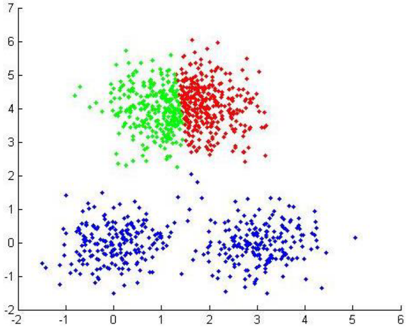
Local optimum



Local optimum (cont.)



Optimal clustering



Possible clustering

Defined Mean

- In the begging we assumed $x_i \in \mathbb{R}^d$, which is not always the case. K-Means requires a space where sample **mean** is defined.
  - A simple case is when we have categorical data.
  - A suggested solution: k-mode - the centroid is represented by most frequent values.

How many clusters?

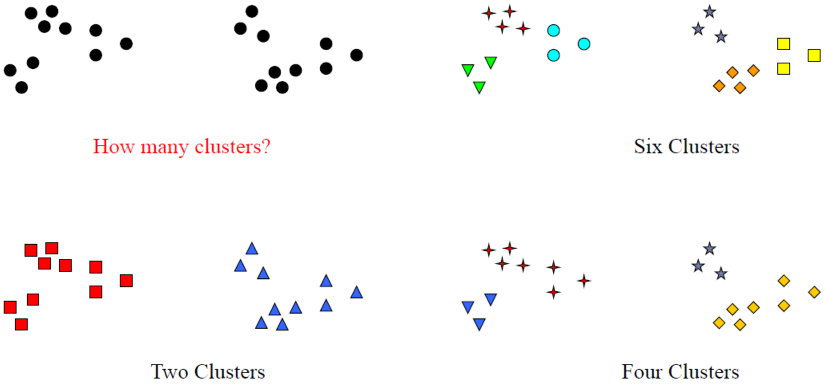


Figure adapted from slides of Dr. Soleymani, Modern Information Retrieval Course, Sharif University of technology.

How many clusters? (cont.)

- Number of clusters is usually given in advance in the problem of clustering. However; finding the **right** number of clusters is also a problem.
- **Elbow Method** and **Silhouette Score** can help.
- There is a trade-off between having better focus within each cluster or having too many clusters.
- Don't want one-element clusters.
- **Optimization problem:** penalize having too many clusters

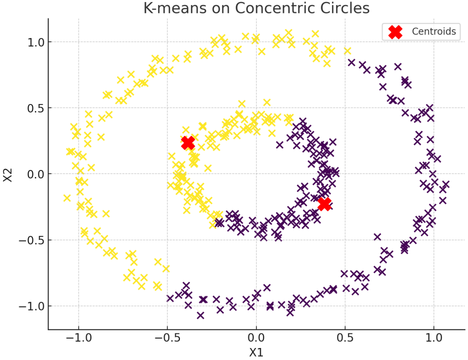
$$K^* = \ arg\ min_k\ J(k) + \lambda k$$

Outliers

- 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-medoids and DBSCAN are more robust to outliers.

Definition Issue

- Perhaps the most important problems is how k-means defines clusters.
- K-means assumes clusters are spherical and separated by equal variance, which limits its effectiveness on non-spherical or complex-shaped clusters.
- So lets take a closer look at clustering.



Clustering

- Assume we have a set of unlabeled data points $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$.
- We intend to organize data into **groups** of **similar** objects.
  - group and similar should be with respect to our need.
  - For example all data points having most similar number of buys in a market.
- A cluster is a collection of data items which are similar between them, and dissimilar to data items in other clusters.
- Clustering could also help to compress and reduce data. (???)

Clustering (cont.)

From another point of view, clusters are regions of high density that are separated from one another with regions of low density.

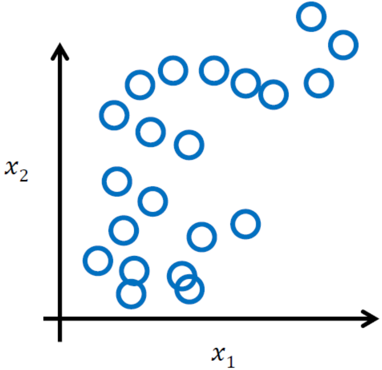


Figure adap

Historic application of clustering

- John Snow, a London physician, plotted the location of cholera deaths on a map during an outbreak in the 1850s.
- The locations indicated that cases were clustered around certain intersections where there were polluted wells – thus exposing both the problem and the solution.

Modern applications of clustering

- Clustering is the origin of many unsupervised learning applications.
- Customer Segmentation (Marketing)
- Image Segmentation and Object Detection (Computer Vision)
- Anomaly Detection (Cybersecurity, Finance)
- Genomics and Bioinformatics
- Social Network Analysis and Community Detection
- Vector Quantization
- ...

Analysing the task

- first lets define a way to measure and show similarity. Two general ways would be:
  - a similarity function $s(x_i, x_j)$ that is larger when x_i and x_j are more similar
  - a dissimilarity or distance function $d(x_i, x_j)$ that is smaller the more simialr to points are.
- a criterion to evaluate (and use to determine) a clustering. notion of "good" and "bad" clustering.
- Algorithm to use the above and compute clustering.
- Extra Note: Most algorithms require a distance function to be a **proper metric** and the similarity measure to create a **PSD matrix** for all pairs of a finite number of data points.

Common similarity and distance measures

- Assume p and q are two data points from $\mathbb{R}^D$. most common similarity and distance measures in the problem of clustering are as follows:
  - **Euclidean distance:** Most common measure of distance between two vectors:

$$d(p,q) = \sqrt{\sum_{i=1}^{D} (p_i - q_i)^2}$$

    it is translation invariant.
  - **Manhattan distance:** Most common measure of distance when dimensions are not equally important

$$d(p,q) = \sqrt{\sum_{i=1}^{D} |p_i - q_i|}$$

  - **Cosine similarity:** Most common measure of similarity when the magnitude of vectors does not change the similarity

$$s(p,q) = \frac{p^T q}{||p|| \, . \, ||q||}$$

Hard clustering vs Soft clustering

- **Hard Clustering:** Each data point belongs to exactly one cluster
  - more common and easier to do
- **Soft Clustering**

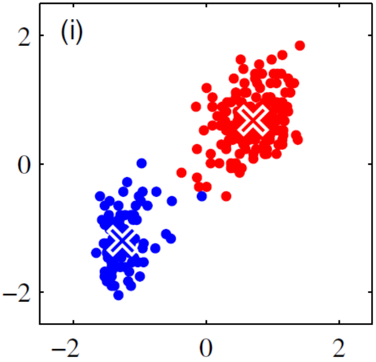


Figure adap

Hard clustering vs Soft clustering (cont.)

- **Hard Clustering**
- **Soft Clustering:** Each data point can belong to multiple clusters.
    - data point belongs to each cluster with a probability
- **From now on, we will focus on problem of hard clustering**

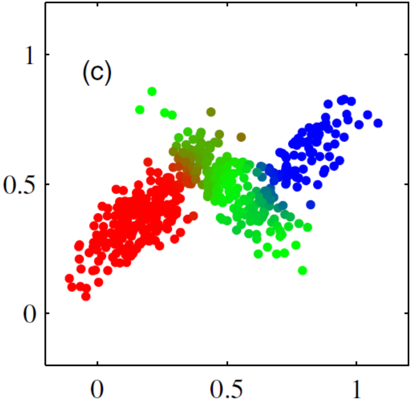


Figure adap

Cluster Evaluation

- Intra-cluster cohesion (compactness)
  - Cohesion measures how near the data points in a cluster are to the cluster centroid.
  - Sum of squared error (SSE) is a commonly used measure.
- Inter-cluster separation (isolation):
  - Separation means that different cluster centroids should be far away from one another.
  - Sum of squared error (SSE) is a commonly used measure.

Clustering Algorithms

- The Traditional algorithms for clustering are usually categorized as:
  - **Hierarchical** algorithms find successive clusters using previously established clusters. These algorithms can be either agglomerative ("bottom-up") or divisive ("top-down"):
    - *Agglomerative* algorithms begin with each element as a separate cluster and merge them into successively larger clusters;
    - *Divisive* algorithms begin with the whole set and proceed to divide it into successively smaller clusters.
  - **Partitional** algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.
  - **Bayesian** algorithms try to generate a posteriori distribution over the collection of all partitions of the data.

Clustering Algorithms (cont.)

- But modern approaches leverage advances in deep learning, self-supervised learning, and representation learning.
- As it is a common idea in ML, these methods transform data vectors, so the traditional clustering concepts can be applied.
- For example, with the same "curse of dimensionality" we had in supervised learning, for high dimensional vectors, using raw distance metrics will lose most of its functionality. So a Neural Network learns to transform data into a low dimensional space where our distance measure is more effective.
- Or when the data clusters are not centeric, they can be transformed into a space where the clusters are separated with respect to distance and in a centeric manner.

Unsupervised Learning Review

- **Objective**: To find hidden structures or underlying distributions in the data.
- **Input**: A dataset $X = \{x_1, x_2, ..., x_n\} \subseteq \mathbb{R}^d$, where the data points $x_i \in \mathbb{R}^d$ are unlabeled.
- **Goal**: Learn a mapping $f : \mathbb{R}^d \to \mathbb{R}^m$ to describe underlying structure, in a way that is useful for a downstream task.
- Common tasks:
  - Clustering: The mapping $f(X) = Z$ where $Z \in \{1, 2, \ldots, K\}$ represents the cluster assignments.
  - Dimensionality Reduction: The mapping $f(X) = Z$, where $Z \in \mathbb{R}^k$ represents a lower-dimensional representation with $k < d$.
  - Density Estimation: Estimate the probability distribution $P(X)$.
  - Anomaly detection
  - Generative modeling

References

- [1]
- [2]
- [3]
- [4]

Contributions

- **This slide has been prepared thanks to:**

[1] C. M., *Pattern Recognition and Machine Learning.*
Information Science and Statistics, New York, NY: Springer, 1 ed., Aug. 2006.

[2] M. OpenCourseWare, “Class 13: Machine learning and cognitive neuroscience.”
`http://www.mit.edu/~9.54/fall14/slides/Class13.pdf`, 2014.
Accessed: 2024-10-09.

[3] D. Sontag, “Lecture 14: Structured prediction.” `https://people.csail.mit.edu/dsontag/courses/ml12/slides/lecture14.pdf`, 2012.
Accessed: 2024-10-09.

[4] M. Soleymani Baghshah, “Machine learning.” Lecture slides.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics).*
Springer, 1 ed., 2007.

[6] A. Ng and T. Ma, *CS229 Lecture Notes.*

[7] T. Mitchell, *Machine Learning.*
McGraw-Hill series in computer science, New York, NY: McGraw-Hill Professional, Mar. 1997.

[8] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data: A Short Course.*
New York, NY: AMLBook, 2012.

[9] S. Goel, H. Bansal, S. Bhatia, R. A. Rossi, V. Vinay, and A. Grover, "CyCLIP: Cyclic Contrastive Language-Image Pretraining," *ArXiv*, vol. abs/2205.14459, May 2022.