CSC 461: Machine Learning

Fall 2024

Clustering, K-means

Prof. Marco Alvarez, Computer Science
University of Rhode Island

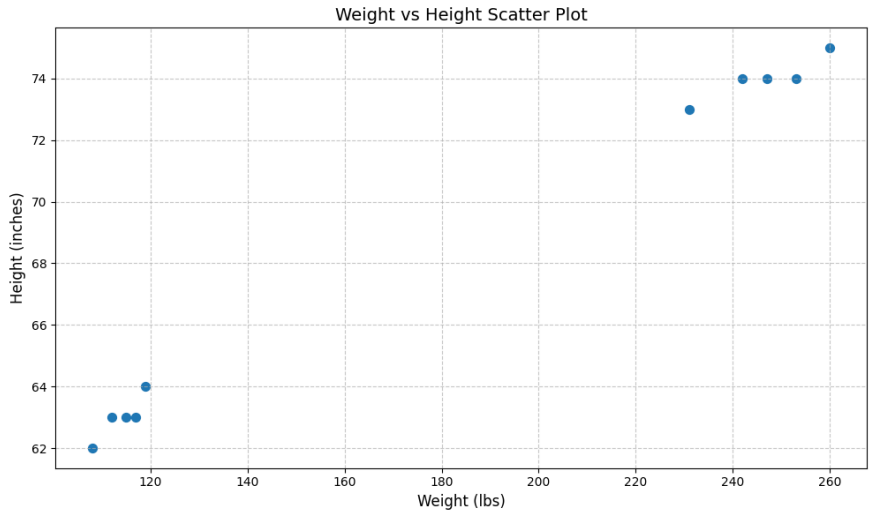
Unsupervised learning

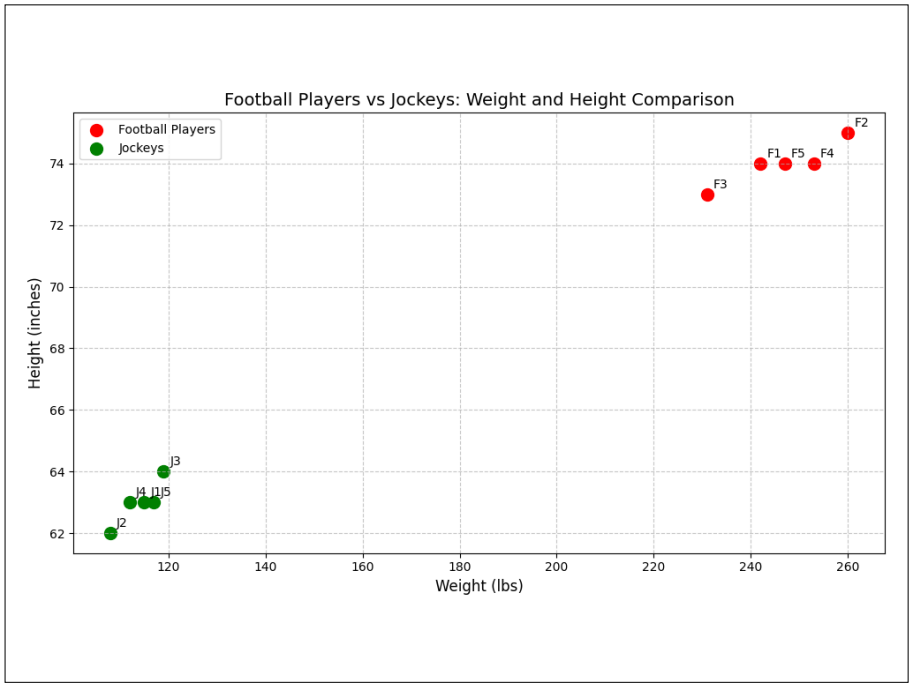
- Algorithms/methods designed to uncover latent structure within **unlabeled data**
  - discover patterns/groupings in the data
- Dataset
  - a set of observation (a.k.a., data instances or data points)

$$\mathscr{D} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}, \qquad \text{usually } \mathbf{x_i} \in \mathbb{R}^d$$

`Labels` may be available with the data, however, they should be ignored if unsupervised learning is applied

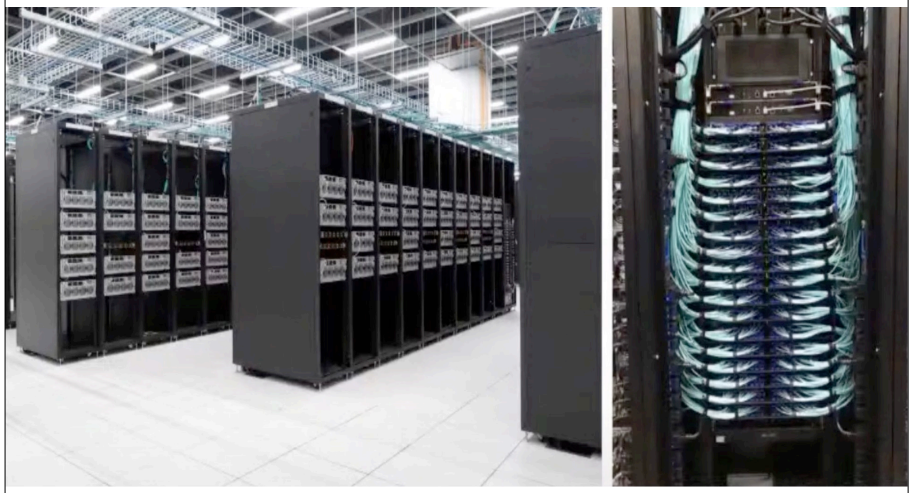
Weight (lbs)	Height (inches)	Class
242	74	Football Player
260	75	Football Player
231	73	Football Player
253	74	Football Player
247	74	Football Player
115	63	Jockey
108	62	Jockey
119	64	Jockey
112	63	Jockey
117	63	Jockey





Cluster Analysis

Not this type of clusters …



https://blogs.nvidia.com/blog/2021/06/22/tesla-av-training-supercomputer-nvidia-a100-gpus/

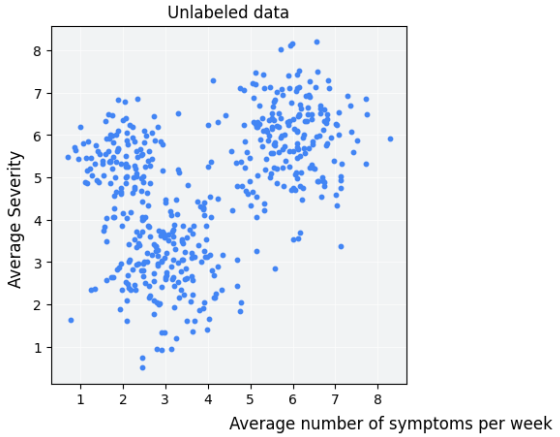
Clustering

- **Grouping** a set of *n* observations into *K* clusters such that:
  - objects within the same cluster exhibit **high intra-cluster similarity**
  - objects from different clusters exhibit **low inter-cluster similarity**
- Scale of clustering tasks
  - typically, *K* ranges from a few to hundreds
  - *n* can range from hundreds to billions
  - each observation usually represented as a **high-dimensional** vector
- Challenges:
  - optimization problem => NP-hard in many formulations
  - trade-off between computational complexity and clustering quality

Clustering in modern ML/AI

- Fundamental unsupervised learning technique
  - critical in data preprocessing and feature engineering
    - increasingly important in big data and high-dimensional spaces
  - example applications: anomaly detection, recommender systems, image segmentation
- Goal
  - discover inherent structure in unlabeled data

Clustering unlabeled data

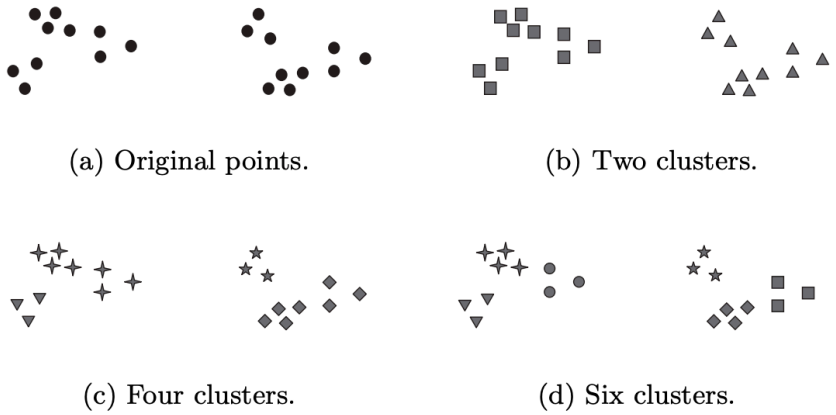


given n feature vectors

group them into k clusters based on pairwise similarities

https://developers.google.com/machine-learning/clustering/overview

Subjective nature



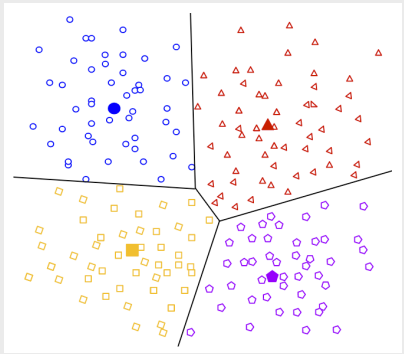
(a) Original points. (b) Two clusters.

(c) Four clusters. (d) Six clusters.

https://www-users.cse.umn.edu/~kumar001/dmbook/index.php

Types of clustering

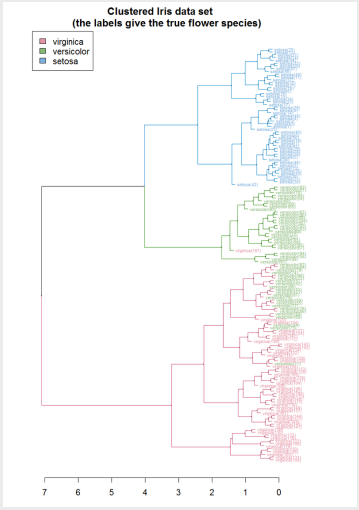
Partition clustering (centroid-based)



- each **observation** belongs to exactly one **partition**
- each **partition** is represented by a **centroid**
- **observations** are iteratively assigned to the **cluster** with the nearest **centroid**

https://developers.google.com/machine-learning/clustering/clustering-algorithms

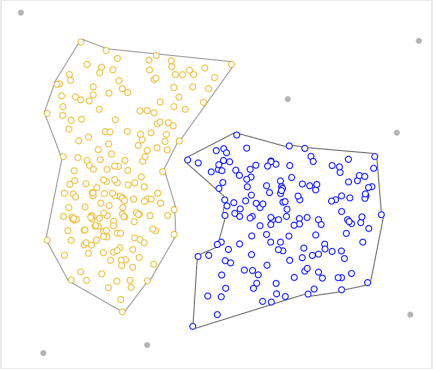
Hierarchical clustering



- defines a tree-like hierarchy (dendrogram) of clusters
- doesn't require the number of clusters in advance
- sensitive to the choice of distance metric between observations

https://en.wikipedia.org/wiki/Hierarchical_clustering

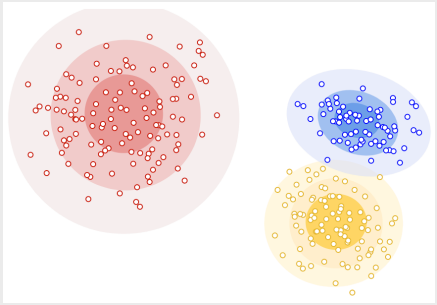
Density-based



- forms clusters in high-density regions of the data space
- can discover arbitrarily shaped clusters
- points in low-density regions are treated as outliers
- challenges: varying densities and high-dimensional data

https://developers.google.com/machine-learning/clustering/clustering-algorithms

Distribution-based



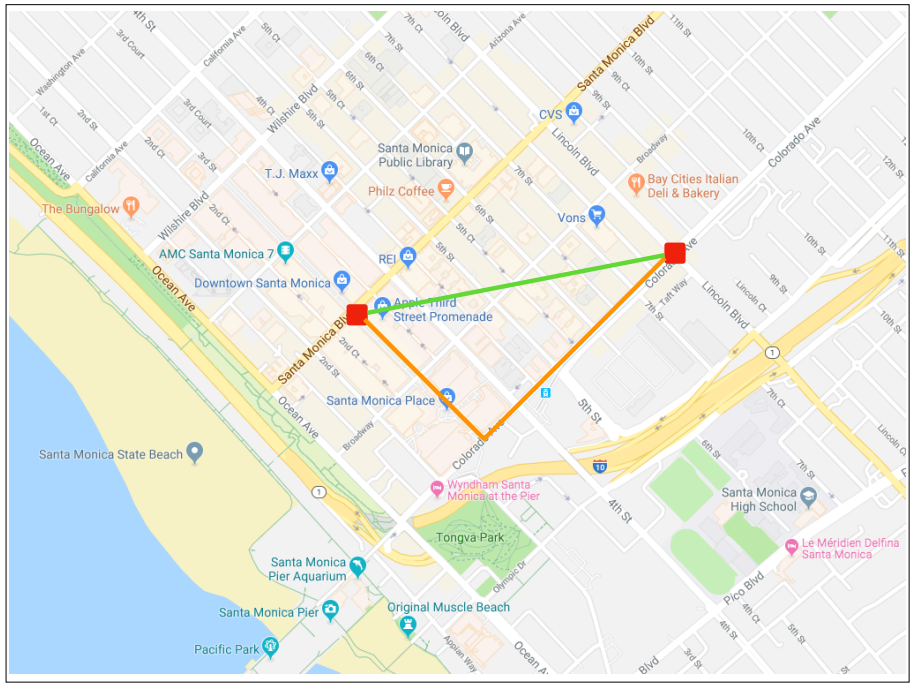
- assumes clusters are generated by underlying probability distributions
- requires specification of an appropriate distribution
- instead of assigning points to a single clusters (hard assignment), it assigns a probability that a point belongs to each cluster (soft assignment)

https://developers.google.com/machine-learning/clustering/clustering-algorithms

Distances and norms

Distance metrics

- Clustering algorithms rely heavily on the concept of **similarity** or **distance** between data points
- Common distance metrics include:
  - Euclidean distance $$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$
  - Manhattan distance $$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$$
  - Cosine similarity $$\text{sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}||\mathbf{y}|}$$
  -



Vector norms

- Vector norms measure the "**length**" of a vector
  - closely related to distance metrics
  - L1 norm (Manhattan norm) $$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$$
  - L2 norm (Euclidean norm) $$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

The L2 norm of $\mathbf{x} - \mathbf{y}$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$

K-means

K-means

- One of the most popular and widely used clustering algorithms
  - due to its simplicity and efficiency
- Given n observations and a desired number of clusters K
  - partition the data into K clusters such that the within-cluster-distance is minimized for all clusters

More formally …

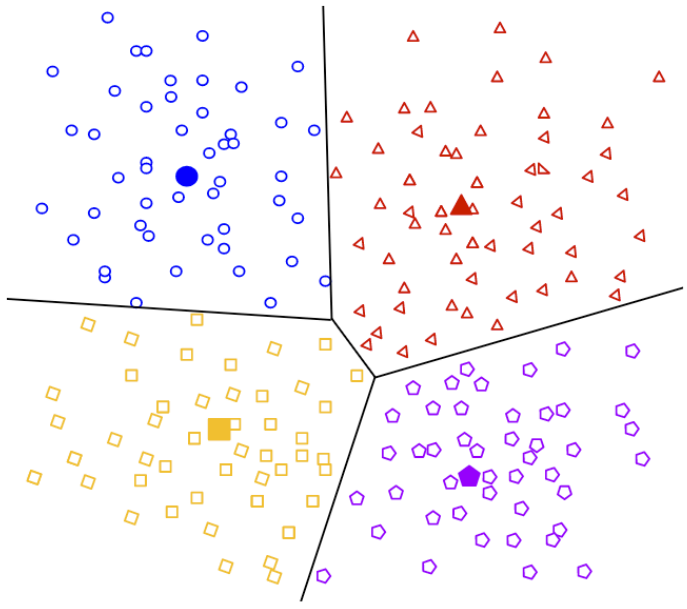
- Given a dataset $\mathcal{D} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$ where $\mathbf{x_i} \in \mathbb{R}^d$, and a desired number of clusters K:

  each cluster is a set denoted by C_k, for $k = 1,\ldots,K$

  if $\mathbf{x_i}$ is assigned to cluster C_k then $i \in C_k$

  subject to: $\bigcup_k C_k = \{1,\ldots,n\}$ and $C_i \cap C_j = \emptyset$, for $i \neq j$

- Cluster centroids
  - group representatives $\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_K}$ where $\mathbf{z_i} \in \mathbb{R}^d$
    - each centroid $\mathbf{z_k}$ is the **mean of all observations** in cluster C_k
    - we want the quantities $\|\mathbf{x_i} - \mathbf{z_k}\|$, where $i \in C_k$, to be small



Clustering objective

- Cost function J denotes the sum of square distances from observations to centroids
  - minimizing J encourages all points to be near their centroids (better clustering)
    - note that J depends on the cluster assignments and centroids

$$J = \underset{C_1,\ldots,C_K}{\arg\min} \sum_{k=1}^{K} \sum_{i \in C_k} \|\mathbf{x_i} - \mathbf{z_k}\|_2^2$$

Given a function $f(x)$, $\arg\min_x f(x)$ denotes the value(s) of x for which $f(x)$ is minimized

How to minimize the goal?

- Trying a brute-force approach for an **optimal solution** would be computationally infeasible
  - i.e., exhaustively enumerating all possible partitions of observations into clusters
- Relaxing our minimization goal
  - instead of an optimal we can settle with an **approximate solution**, using an efficient iterative method (Lloyd's algorithm, 1957)

K-means algorithm

- Randomly **initialize** K centroids
- Repeat until convergence
  - **partition:** assign each $\mathbf{x_i}$ to their closest centroid
  - **update:** recompute all cluster centroids
    - set $\mathbf{z_k}$ to be the mean of all observations in group C_k

Ties in group assignment are broken by selecting the group with the smallest index

Empty groups can occur and are simply dropped

The algorithm converges when group assignments remain unchanged in successive iterations, though it's often stopped earlier when improvement becomes minimal

Initial representatives can be chosen randomly from the data points or by random assignment

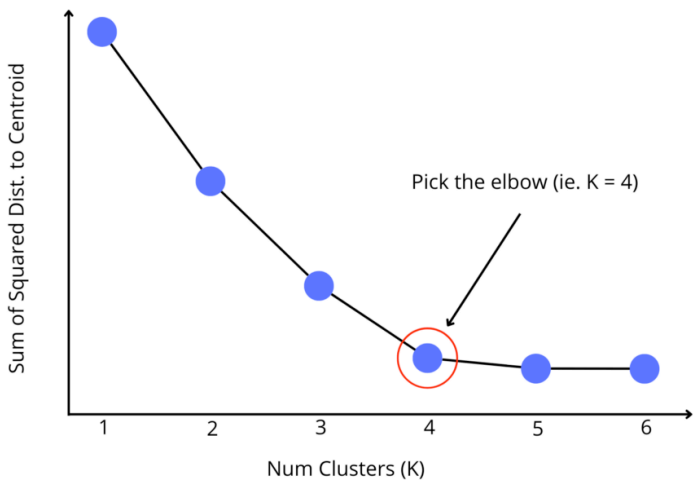
K-Means Clustering Demo

https://user.ceng.metu.edu.tr/~akifakkus/courses/ceng574/k-means

Issues

- Convergence
  - J decreases in each step, ensuring finite convergence
  - **heuristic nature**: may converge to different partitions based on initial centroids
    - **NOT guaranteed** to converge to an optimal (global) solution
- Initialization sensitivity
  - different results for different values of K (provided by user)
  - results highly dependent on initial centroid placement
- Practical approach
  - run algorithm multiple times with different initializations and values of K, then select partition with smallest J among the multiple runs
  - can also use the "elbow" method (to choose best K)
  - use k-means++ (better initialization)
- Widely used and effective in practical applications despite limitations

The "elbow" method



https://towardsdatascience.com/unsupervised-learning-k-means-clustering-27416b95af27

K-Means Notebook

https://colab.research.google.com/drive/1tw1zTdDO4Abd57-jb0SmECtSUJxIgugF