MATH412/COMPSCI434/MATH713
Fall 2025
Topological Data Analysis

Topic 11: TDA + Machine Learning

Instructor: Ling Zhou

- We have seen different topological objects that can be potentially used for data analysis
- They can be used to potentially augment / strengthen machine learning approaches

- Today
  - Examples of how they connect or can be combined with ML pipelines.

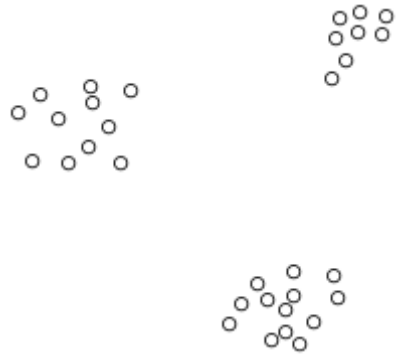
Overview

- Hierarchical clustering tree (HTC)
  - Connection to join/merge tree
- Graph Classification
  - Kernel methods
  - Weisfeiler-Lehman + persistent homology
- Topological constraints / priors
  - Optimizing topological loss function
  - Topological layer in NN

Hierarchical clustering tree (HCT) for density distribution

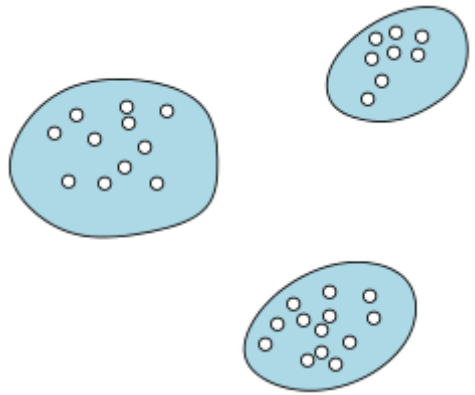
Introduction

- Clustering



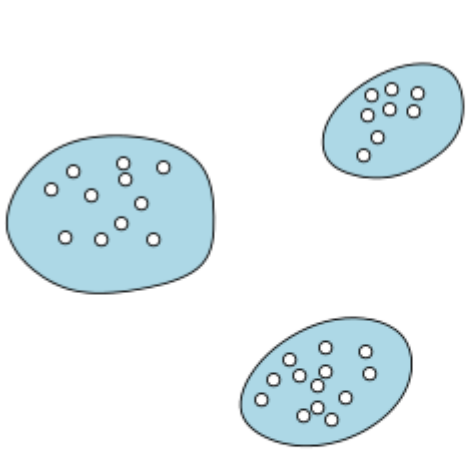
Introduction

- Clustering



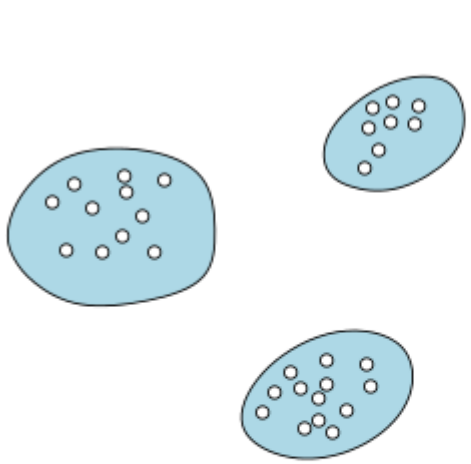
Introduction

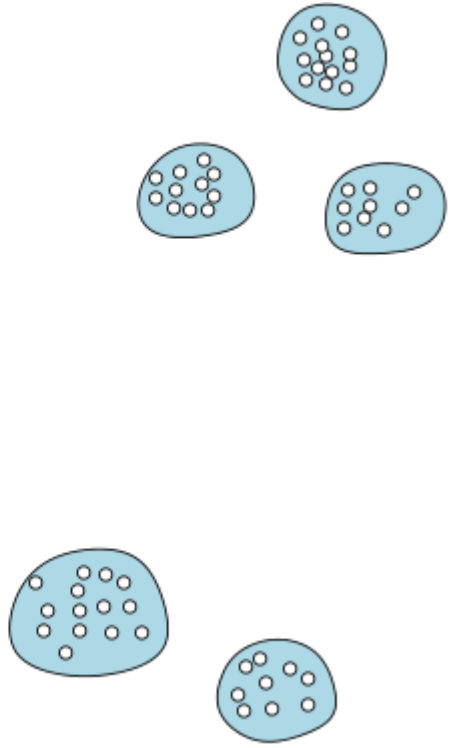
- Hierarchical Clustering



Introduction

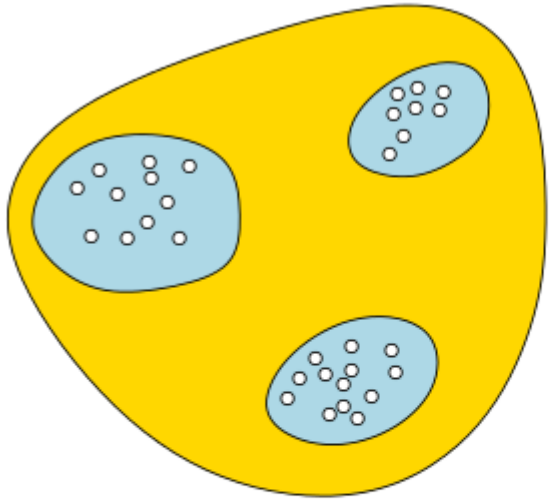
- Hierarchical Clustering

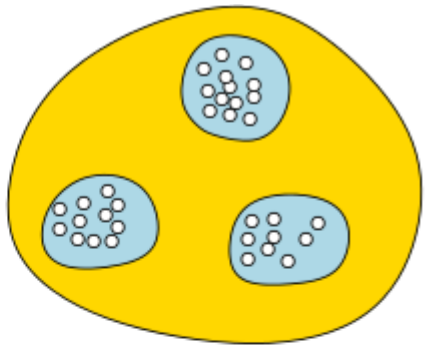


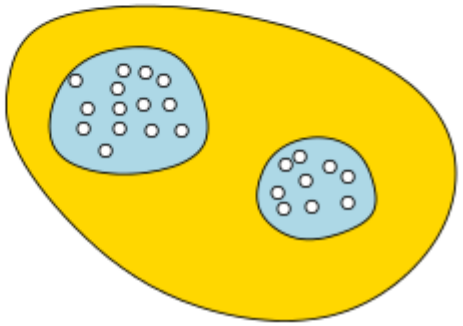


Introduction

- Hierarchical Clustering

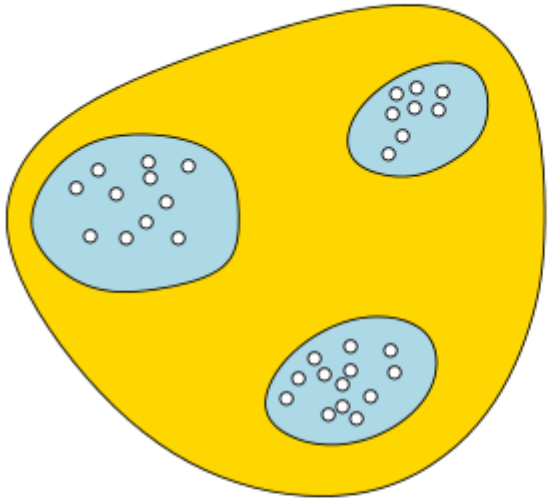


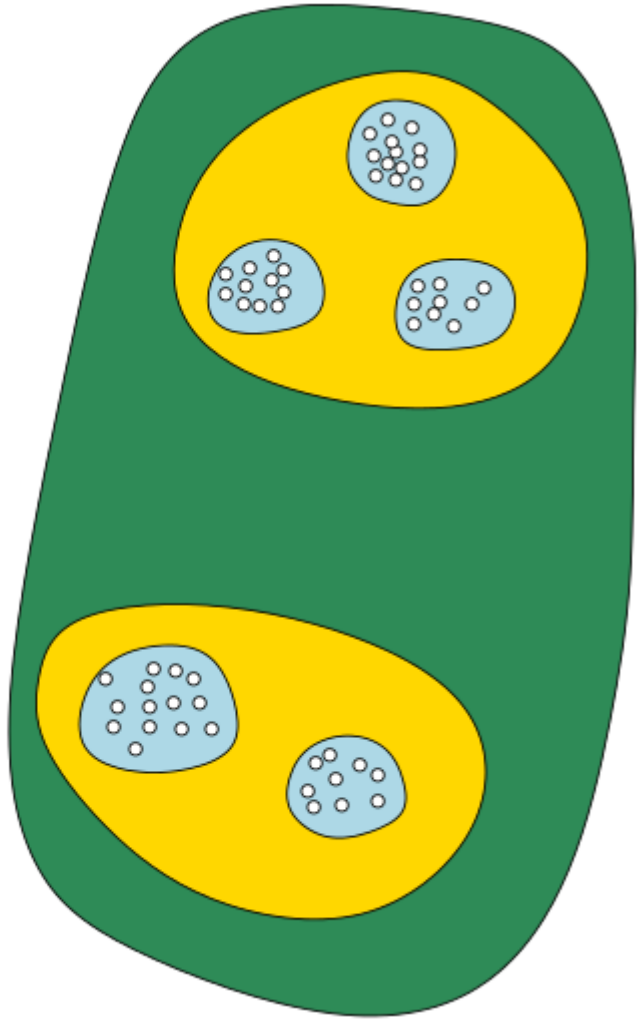




Introduction

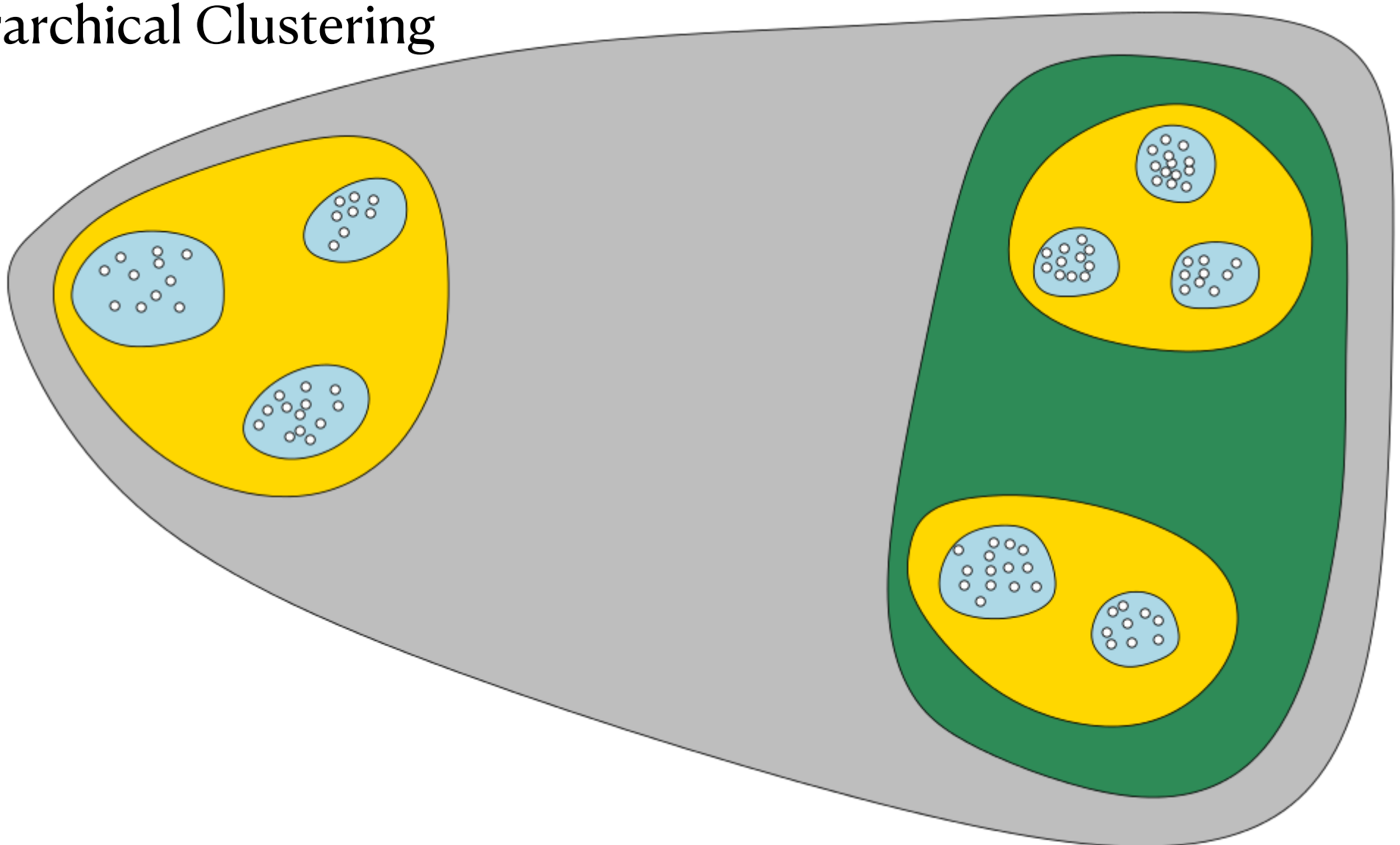
- Hierarchical Clustering



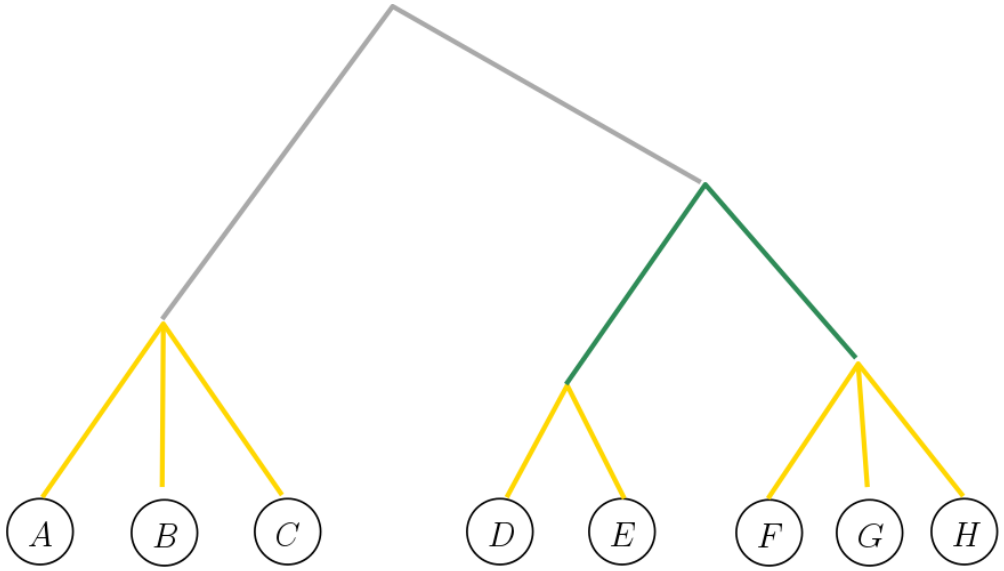


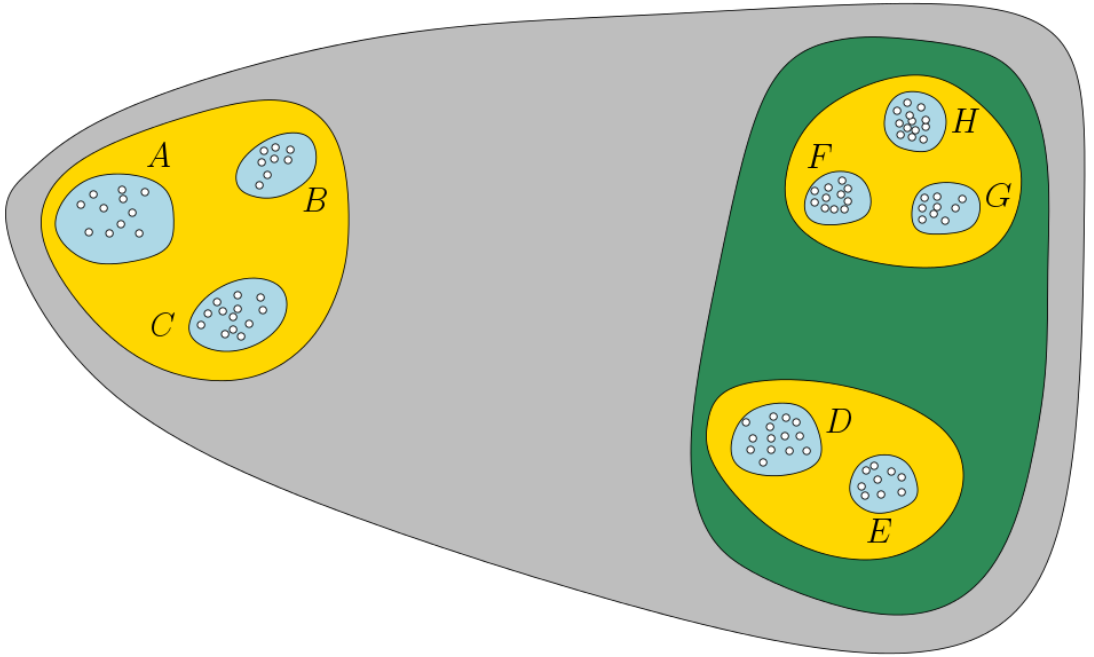
Introduction

- Hierarchical Clustering



Hierarchical Clustering Tree (HCT)

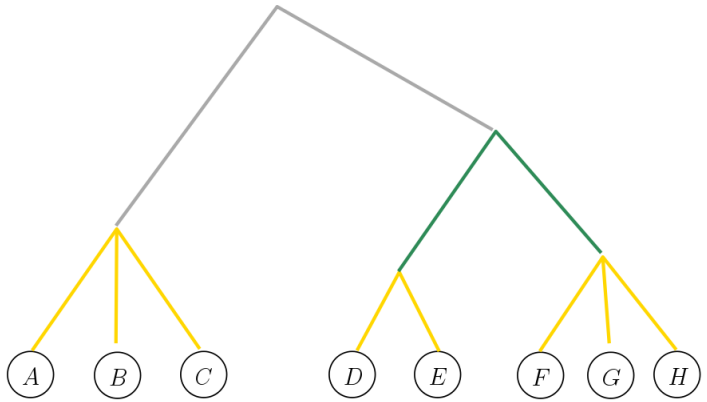




- Each internal tree node indicates a cluster, containing all leaves in subtree
- Ancestor/descendent indicates containment relation
- Height at each tree node corresponds to certain *cost* of the corresponding cluster (e.g, tightness of cluster)

Hierarchical Clustering Tree (HCT)

- A *hierarchical clustering tree (HCT)* of a set X is a collection $\mathscr{C}$ of subsets of X s.t. $X \in \mathscr{C}$ and $\mathscr{C}$ has hierarchical structure. That is, if $C_1, C_2 \in \mathscr{C}$ such that $C_1 \neq C_2$, then $C_1 \cap C_2 = \emptyset$, or $C_1 \subset C_2$ or $C_2 \subset C_1$.
- Each element C of $\mathscr{C}$ is a *cluster*. Each cluster C is a node in the tree. The descendants of C are those clusters $C' \in \mathscr{C}$ such that $C' \subset C$.
- Every cluster in the tree except for X itself is a descendant of X. Hence X is called the *root* of the HCT.



One Example: Single Linkage Clustering

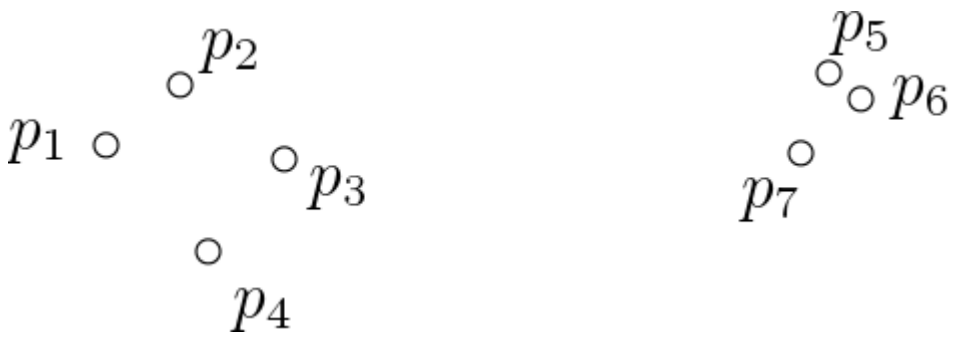
- One of the family of agglomerative clustering methods
  - Input: A discrete n-point metric (P, d_P)
  - Output: A hierarchical clustering tree T, with points in P being leaves

- Starting with each data point as a single cluster
- Keep merging clusters based on nearest distance between points from their members

P does not need to be embedded in Euclidean space.

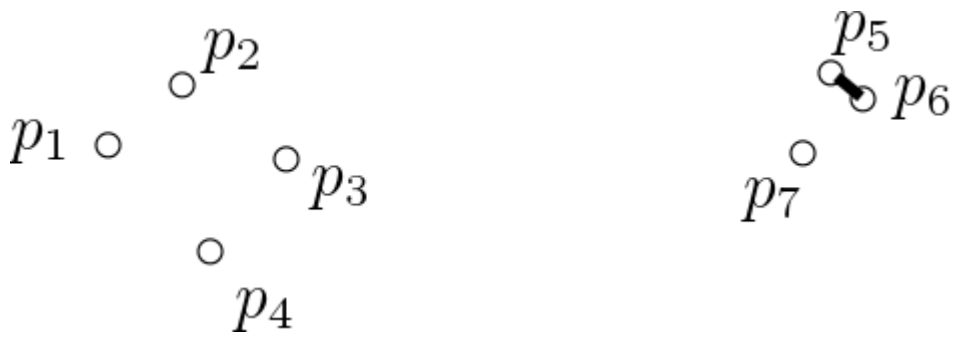
One Example: Single Linkage Clustering

- One of the family of agglomerative clustering methods
- Starting with each data point as a single cluster
- Keep merging clusters based on nearest distance between points from their members



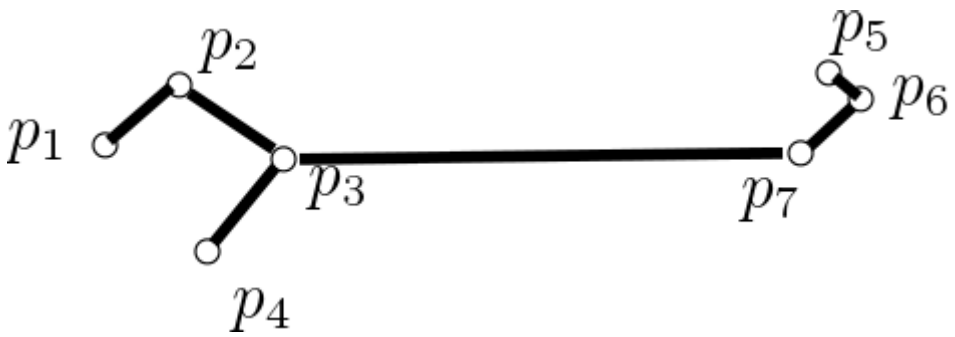
One Example: Single Linkage Clustering

- One of the family of agglomerative clustering methods
- Starting with each data point as a single cluster
- Keep merging clusters based on nearest distance between points from their members



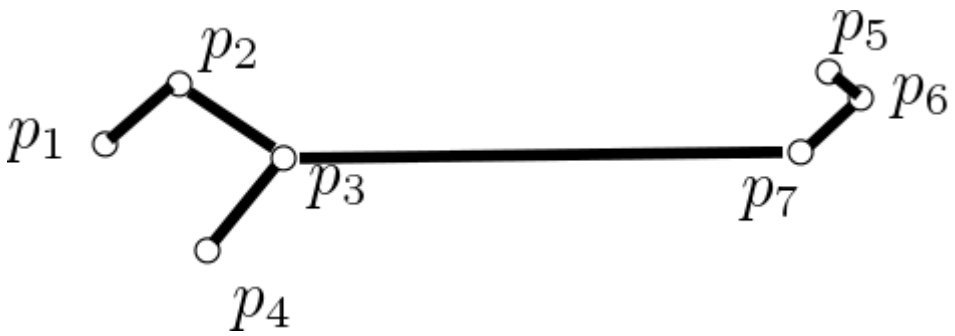
One Example: Single Linkage Clustering

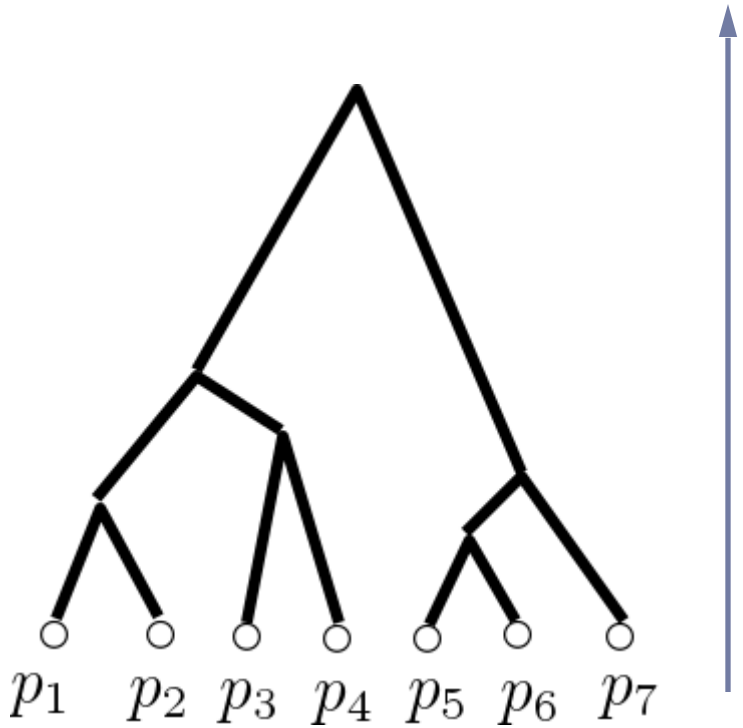
- One of the family of agglomerative clustering methods
- Starting with each data point as a single cluster
- Keep merging clusters based on nearest distance between points from their members



One Example: Single Linkage Clustering

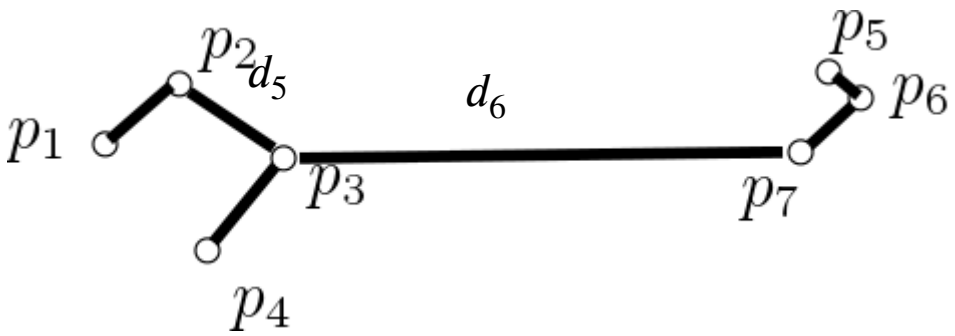
- One of the family of agglomerative clustering methods
- Starting with each data point as a single cluster
- Keep merging clusters based on nearest distance between points from their members

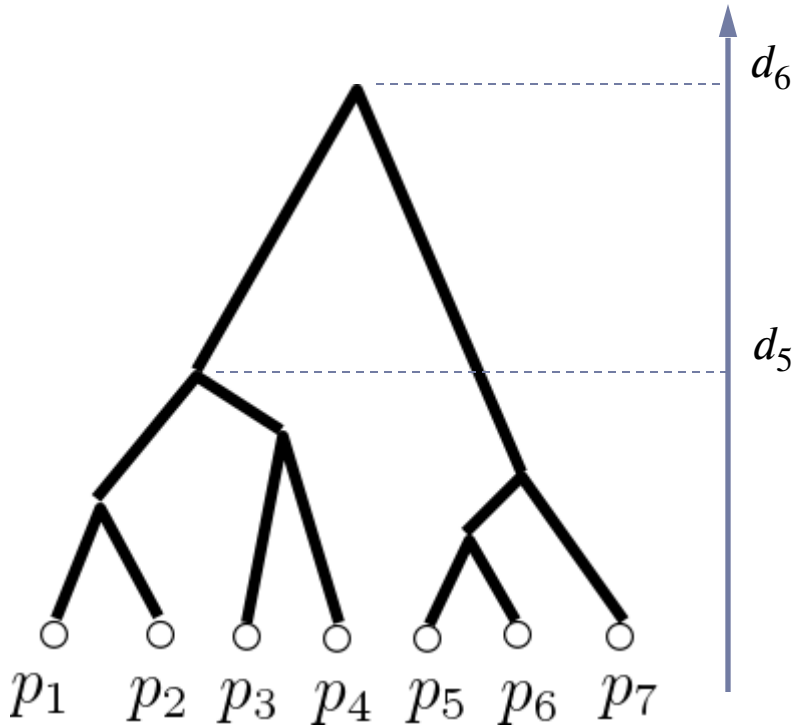




One Example: Single Linkage Clustering

- One of the family of agglomerative clustering methods
- Starting with each data point as a single cluster
- Keep merging clusters based on nearest distance between points from their members

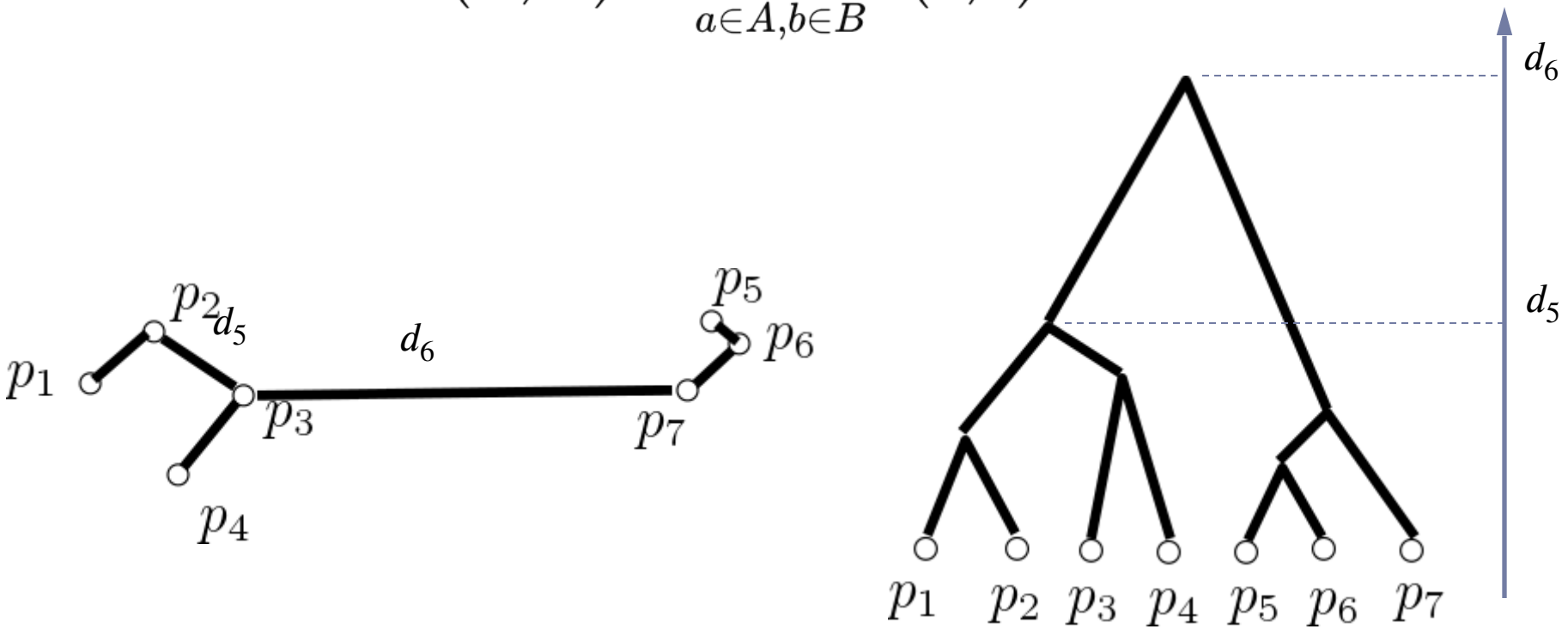




One Example: Single Linkage Clustering

- Given two subsets A and B of the P, single linkage clustering constructs a tree such that the distance between the two clusters A and B is

$$D(A, B) = \min_{a \in A, b \in B} d(a, b)$$



- Some other agglomerative clustering methods
  - Average linkage → $D(A,B) = \frac{1}{|A|\,|B|} \sum_{a \in A} \sum_{b \in B} d(a,b)$
  - Complete linkage → $D(A,B) = \max_{a \in A, b \in B} d(a,b)$
  - ...
- Top-down methods:
  - Hierarchical spectral clustering
  - Iterative k-means
  - ...

- Okay, we can define different HCTs for the same discrete data produced by different algorithms

- How do we talk about “faithfulness” or even consistency of HCTs for data sampled from a hidden distribution
  - First, need a model of space where data are sampled from
  - Next, need to model “true” HCT
  - Then, measure distance and obtain statistical consistency etc.

How to model the space behind data

- Some common scenarios:
  - The input is a set of points sampled from a density distribution
  - The input is a graph sampled from a graph generative model

- In what follows, we will consider the first scenario, but the work has been extended to the second scenario

Problem Setting

Beyond Hartigan Consistency: Merge Distortion Metric for Hierarchical Clustering

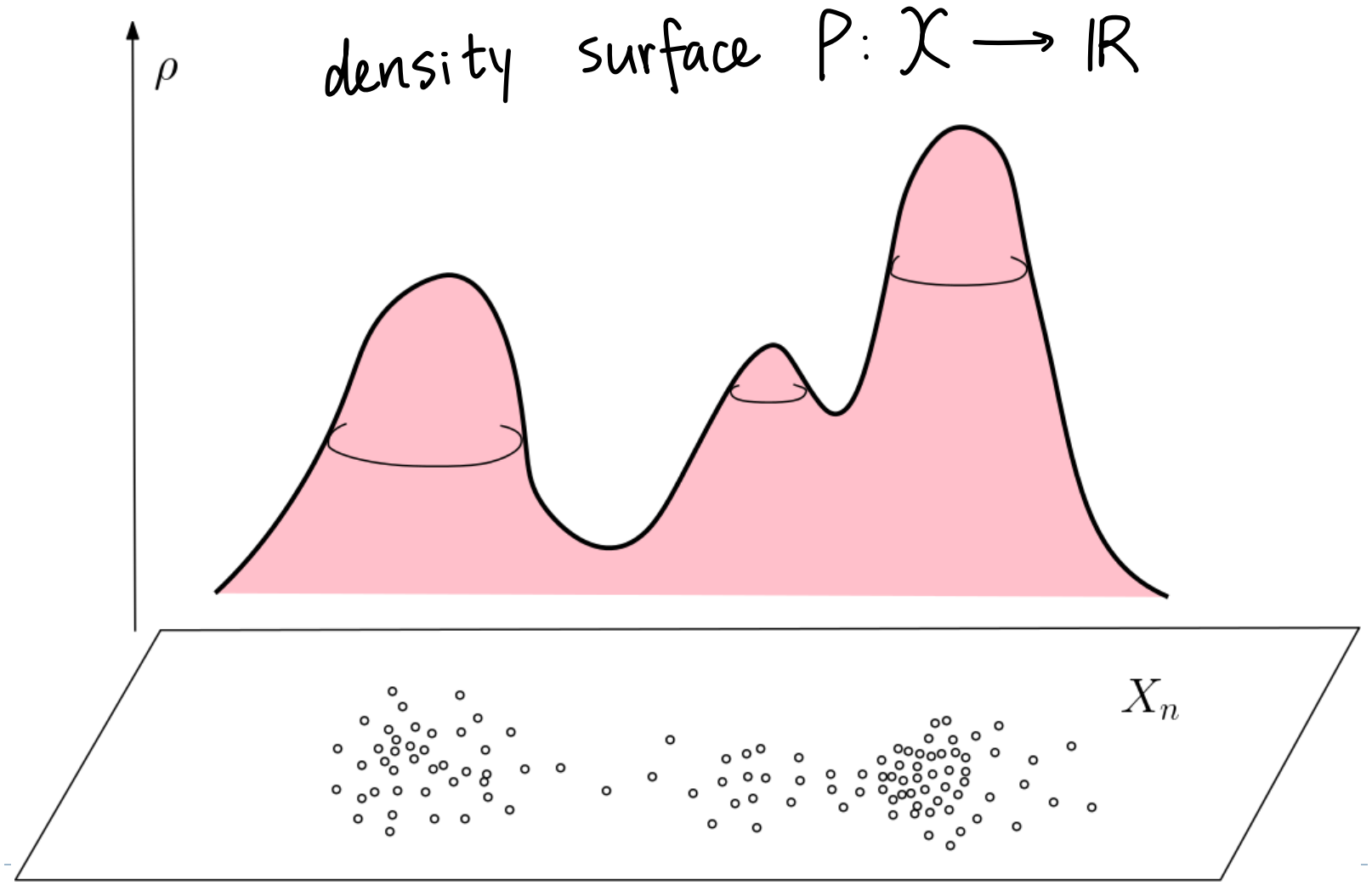
Justin Eldridge, Mikhail Belkin, Yusu Wang *Proceedings of The 28th Conference on Learning Theory*, PMLR 40:588-606, 2015.

Graphons, mergeons, and so on!

Justin Eldridge **Mikhail Belkin** **Yusu Wang**
The Ohio State University
{eldridge, mbelkin, yusu}@cse.ohio-state.edu

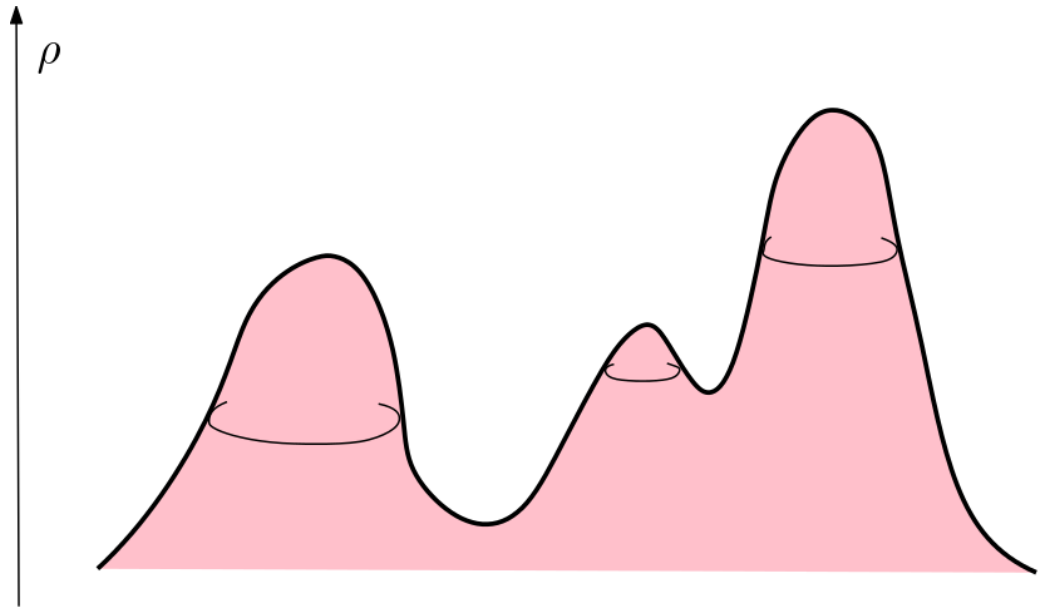
- Input:
  - A set of n points X_n sampled from a ``nice'' density distribution $\rho: \mathcal{X} \to R$ with $\mathcal{X} \subset R^d$
- Output:
  - A hierarchical clustering tree T_n constructed from X_n

- Question:
  - As $n \to \infty$, does T_n converge to the ``*true hierarchical clustering tree*'' T_ρ for the density ρ ?
  - [Eldridge, Belkin, W., 2015, 2018]


density surface $P: X \to \mathbb{R}$
ρ
X_n

True HCT T_ρ

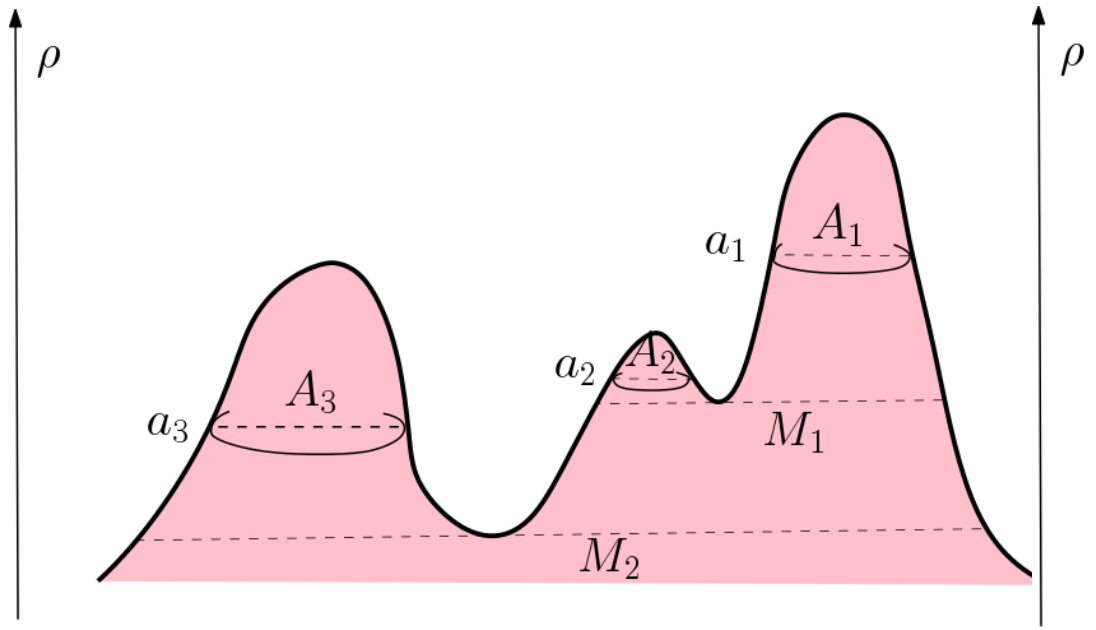
- Given $\rho: \mathcal{X} \to R$ with $\mathcal{X} \subset R^d$, the *density cluster tree* of ρ, defined as T_ρ, is the HCT whose nodes (clusters) are the connected components of
  - $f^{-1}(\lambda) = \{x \in \mathcal{X} : f(x) \geq \lambda\}$, for any $\lambda \geq 0$.

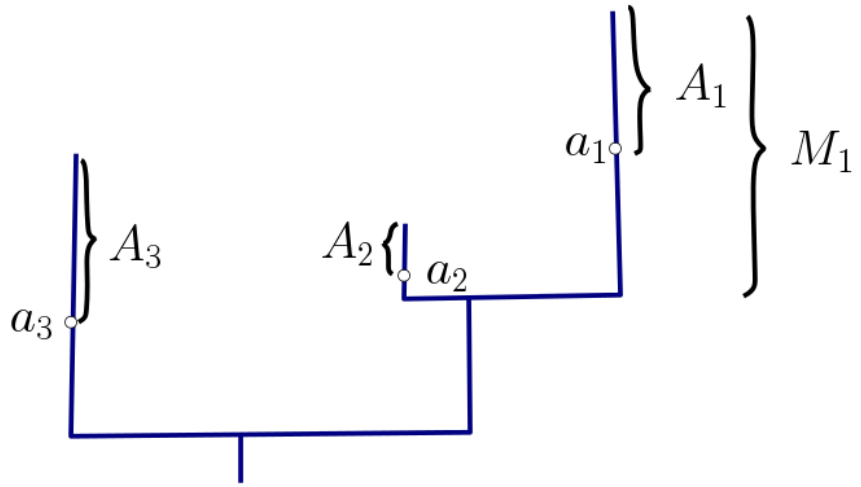


True HCT T_ρ

- Given $\rho: \mathcal{X} \to \boldsymbol{R}$ with $\mathcal{X} \subset \boldsymbol{R}^d$, the *density cluster tree* of ρ, defined as T_ρ, is the HCT whose nodes (clusters) are the connected components of
  - $f^{-1}(\lambda) = \{x \in \mathcal{X} : f(x) \geq \lambda\}$, for any $\lambda \geq 0$.

T_ρ is the merge tree w.r.t the density function ρ !

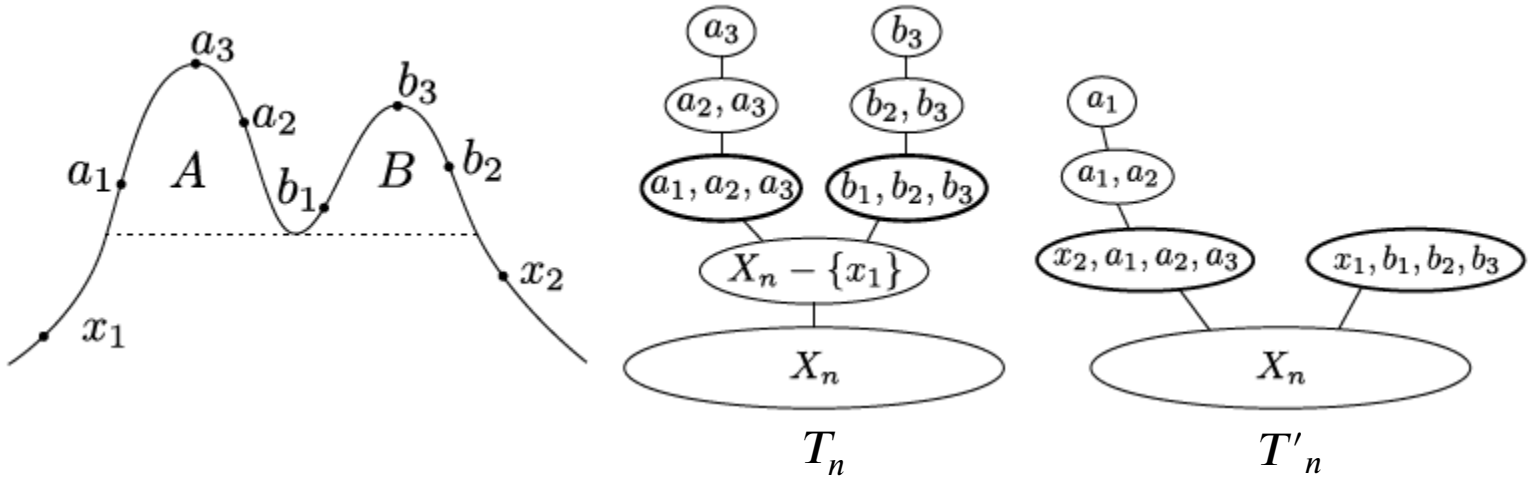




Distance for HCTs

- Equip a HCT T_n with **height** $h: T_n \to R$ with $h(C) := \min_{x \in C} \rho(x)$, for any cluster C of T_n
- Given two points $x,\ x' \in X_n$, and a HCT $(T_n,\ h)$, the **merge height** of $x,\ x'$ is

$$m_h(x,\ x') := \sup_{\substack{x,\ x' \in C \\ C \text{ cluster of } T}} h(C)$$



Distance for HCTs

- Given a set of n points X_n sampled from a density function $\rho: \mathcal{X} \to R$
- (T_n, h): empirical cluster tree for X_n equipped with height h
- (T_ρ, ρ): true density cluster tree equipped with height ρ
- *Merge-distortion distance*
  - $D\left(T_n, T_\rho\right) = \sup_{x, x' \in X_n} |m_\rho(x, x') - m_h(x, x')|$

Intuitively, an empirical HCT T_n is close to T_ρ if it merges different elements at about the same height.

Ideally, for a given clustering algorithm $\mathcal{A}$, we want that its output HCT T_n converges to T_ρ (in probability) as $n \to \infty$.

A Consistent Clustering Algorithm

- Given a set of n points X_n sampled from a density function $\rho: \mathcal{M} \to R$ on a manifold $\mathcal{M} \subset R^d$
  - Estimate empirical density $\hat{\rho} : X_n \to R$
  - Choose a parameter $r > 0$ and construct the 1-skeleton of Rips complex $K_r = Rips^r(P)$
  - Compute the merge tree (variant of the contour tree) T_n for the PL-function $\hat{\rho}$ on K_r

- Theorem:
  - For any fixed $r > 0$, we have with probability 1 as $n \to \infty$, that $D\left(T_n, T_\rho\right) = O(r)$.

There are other consistent HCT algorithms, including the Robust single linkage clustering algorithm by *[Chaudhuri and Dasgupta, 2010]*

Remarks

- HCT essentially can be viewed as a 0-dimensional topological summarization

- What we learned in class also provides higher dimensional summaries as well
  - e.g, topological metaphor such as Mapper structure

Machine Learning

Neural network

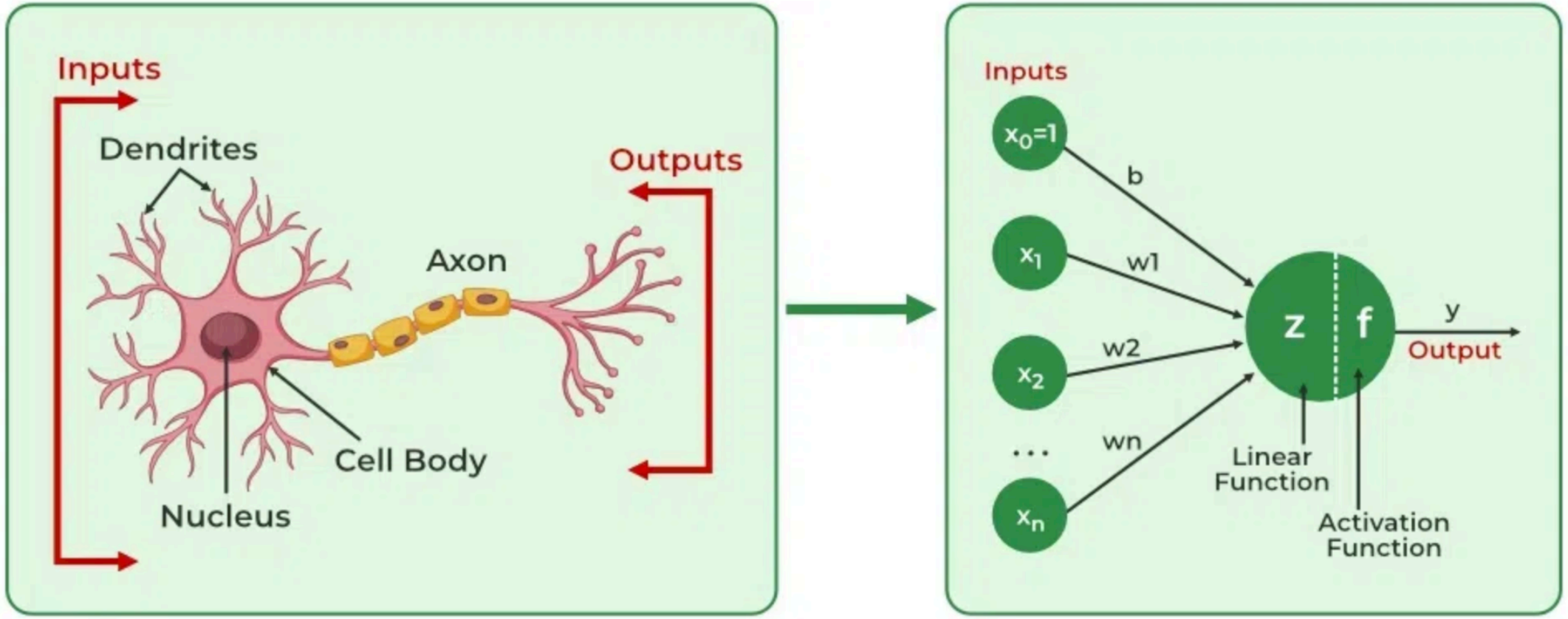
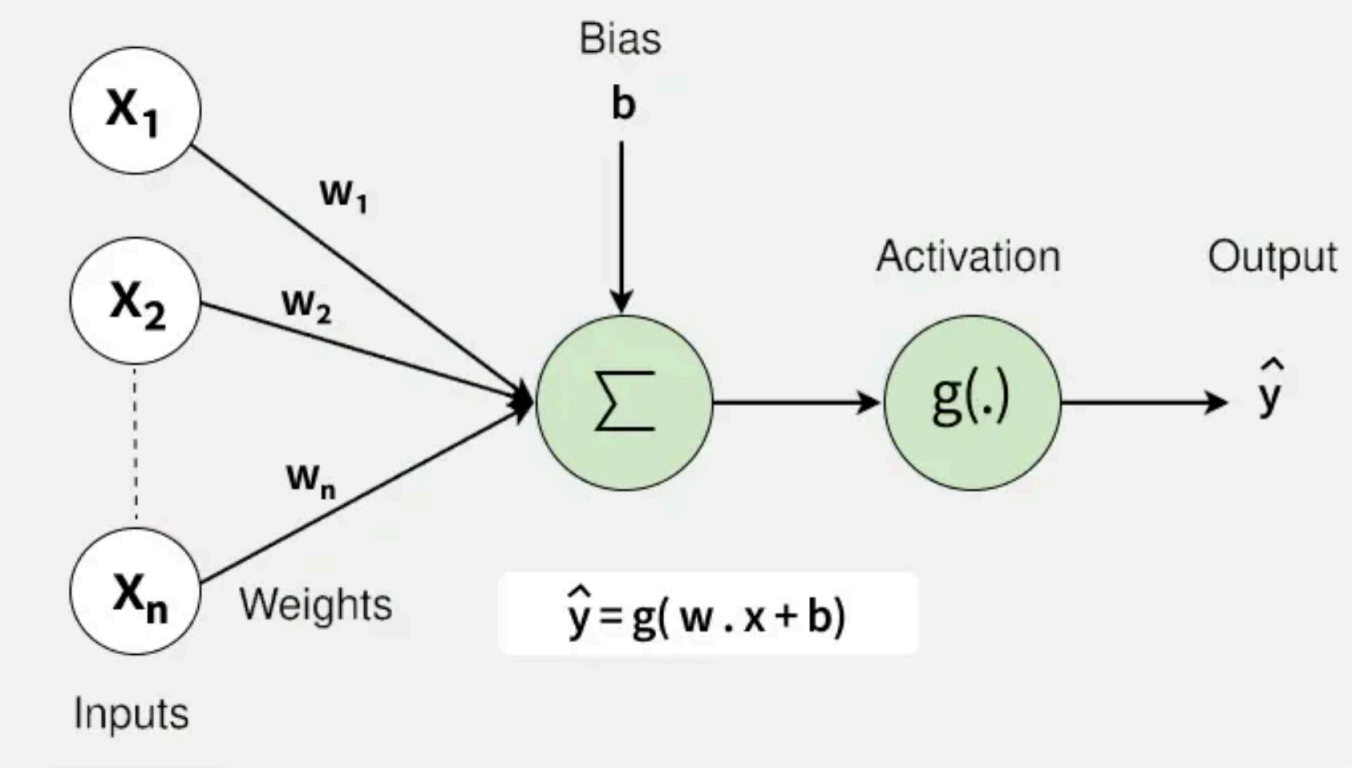


Figure taken from https://www.geeksforgeeks.org/machine-learning/neural-networks-a-beginners-guide/

Neural network

$$\hat{y} = g(Wx + b)$$

- x & b : vectors
- W : matrix
- g : activation function



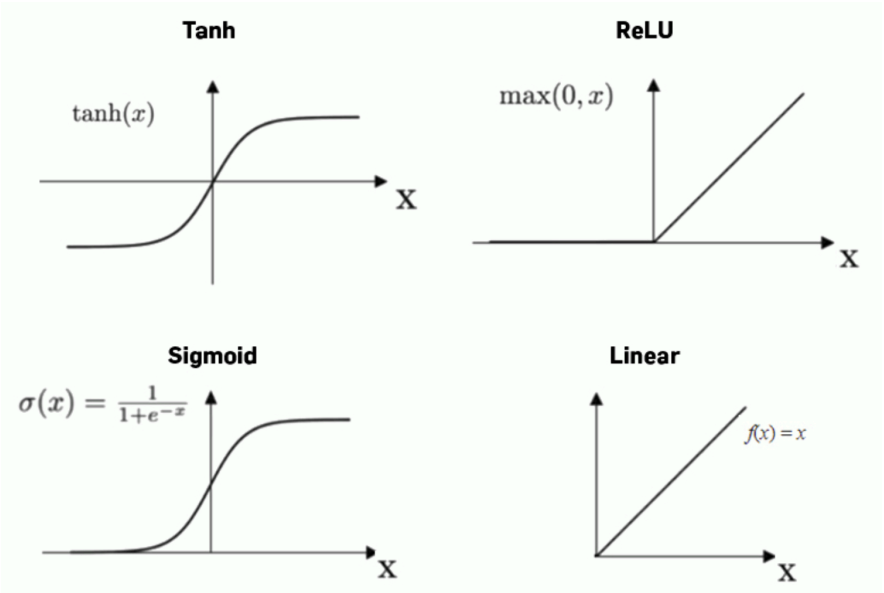


Figure taken from https://www.geeksforgeeks.org/machine-learning/neural-networks-a-beginners-guide/