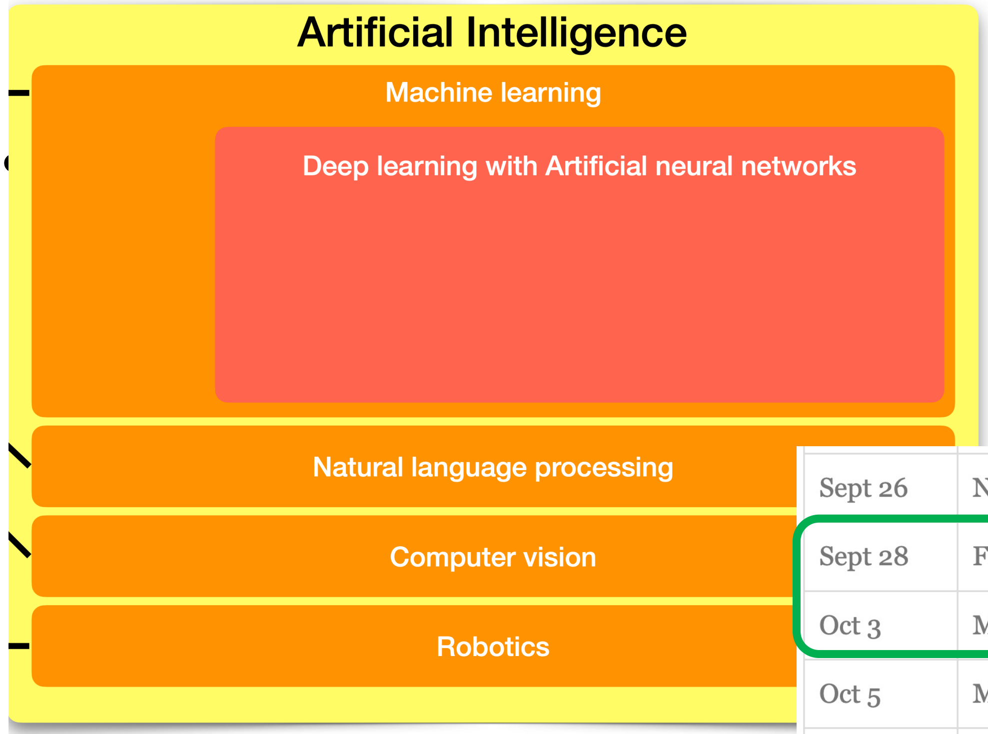
CS 540 Introduction to Artificial Intelligence
ML Intro / Unsupervised Learning I

University of Wisconsin-Madison
Fall 2023



Sept 26	Natural Language Processing (NLP)
Sept 28	Finish NLP; Machine Learning: Introduction
Oct 3	Machine Learning: Unsupervised Learning I
Oct 5	Machine Learning: Unsupervised Learning II
Oct 10	Machine Learning: Linear Regression
Oct 12	Machine Learning: K-Nearest Neighbors & Naive Bayes

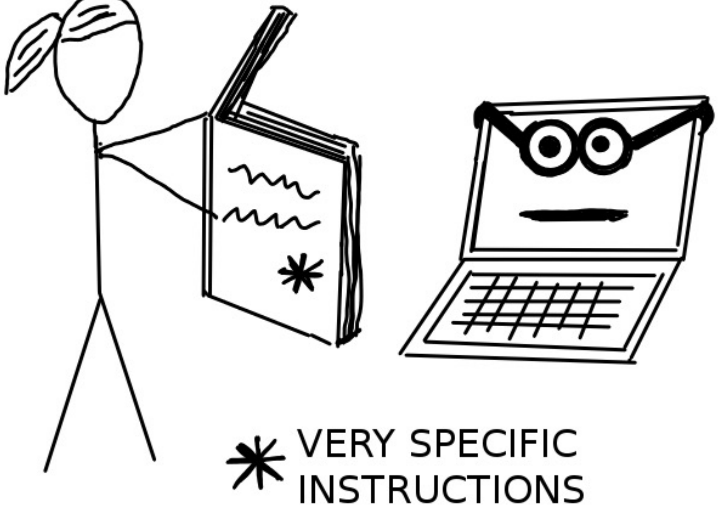
Outline

- Machine Learning Overview
  - Supervised learning, unsupervised learning, reinforcement learning
- Unsupervised Learning: Clustering
  - Hierarchical Clustering
    - Divisive, agglomerative, linkage strategies
  - Centroid-based, K-Means

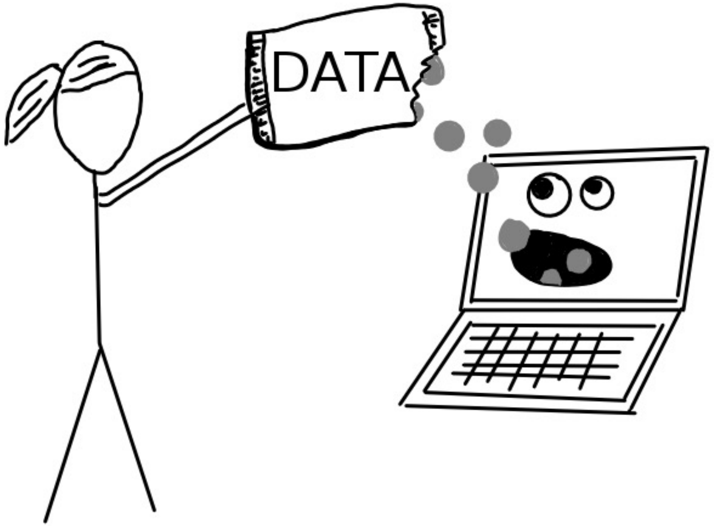
What is machine learning?

- Arthur Samuel (1959): the field of study that gives the computer the ability to learn **without being explicitly programmed**.
- Tom Mitchell (1997): A computer program is said to learn from **experience E** with respect to some class of **tasks T** and **performance measure P**, if its performance at tasks in T as measured by P, improves with experience E.

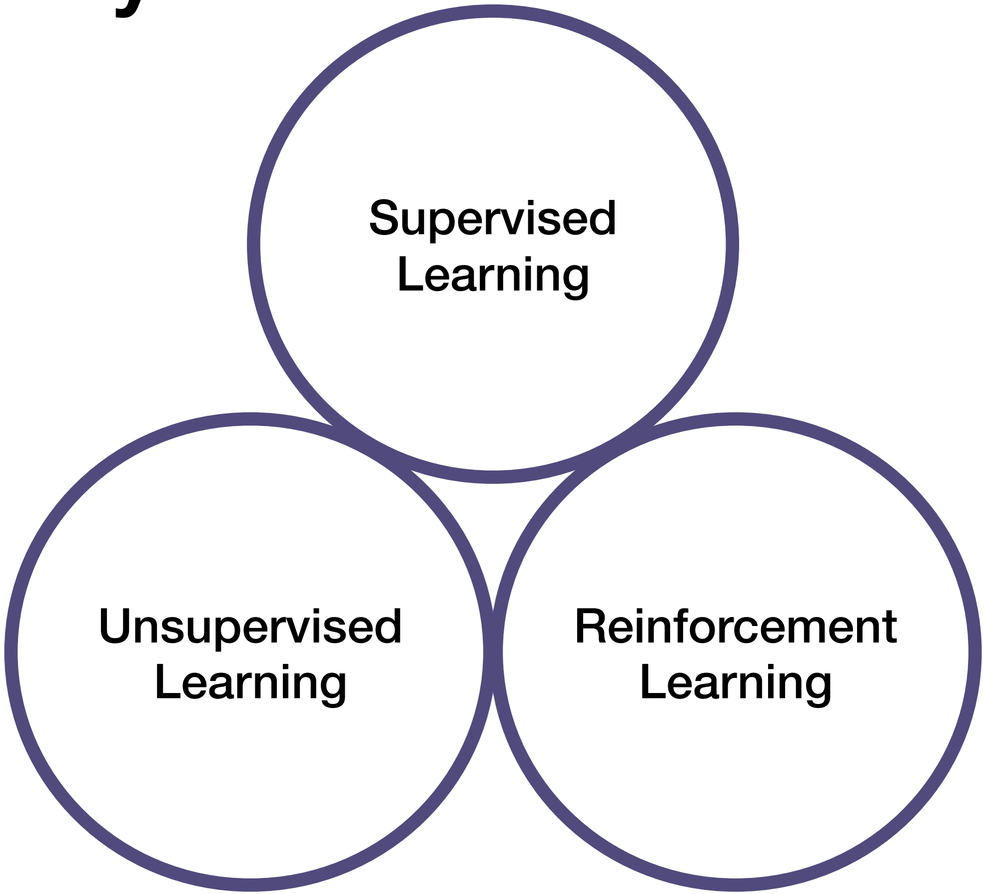
Without Machine Learning



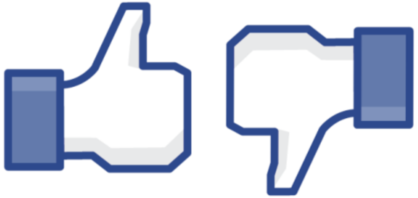
With Machine Learning



Taxonomy of ML



Supervised Learning

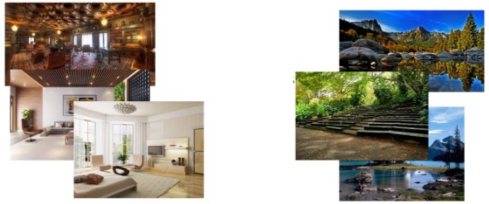


Supervised learning:

- Learn from labelled data.
- Dataset: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

Features / Covariates / Input

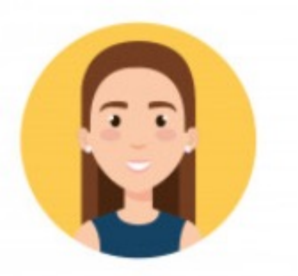
Labels / Outputs



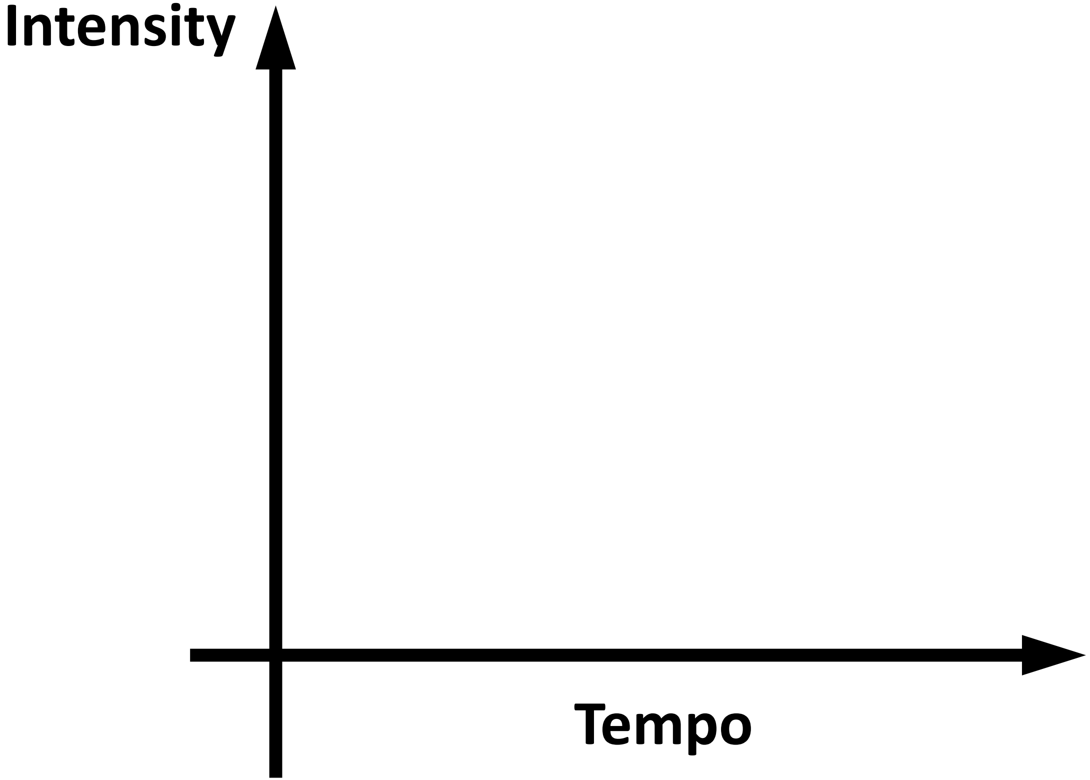
indoor outdoor

- Goal: find function $f : X \to Y$ to predict label on **new** data
- Labels can be discrete ("classification") or real-valued ("regression").

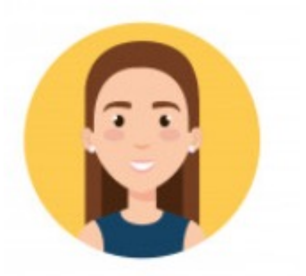
Example 1: Predict whether a user likes a song or not



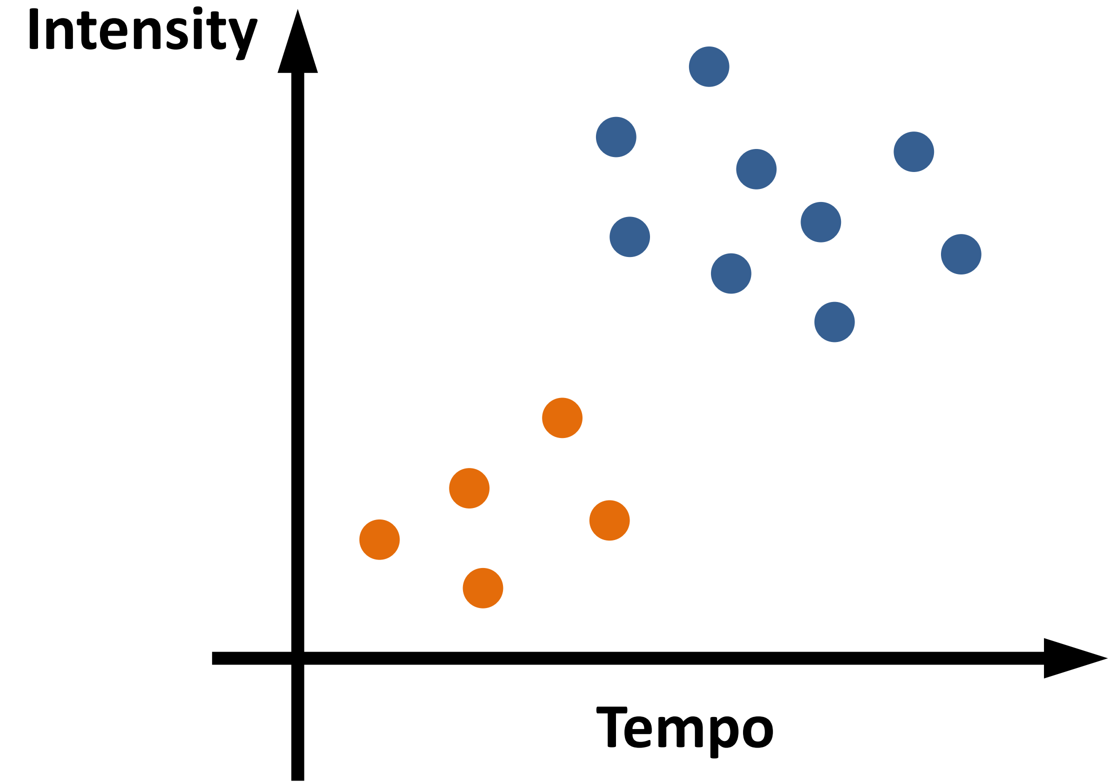
User Sharon



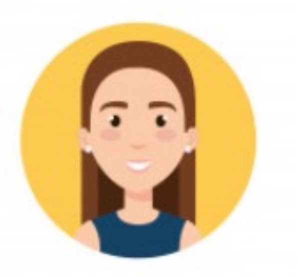
Example 1: Predict whether a user likes a song or not



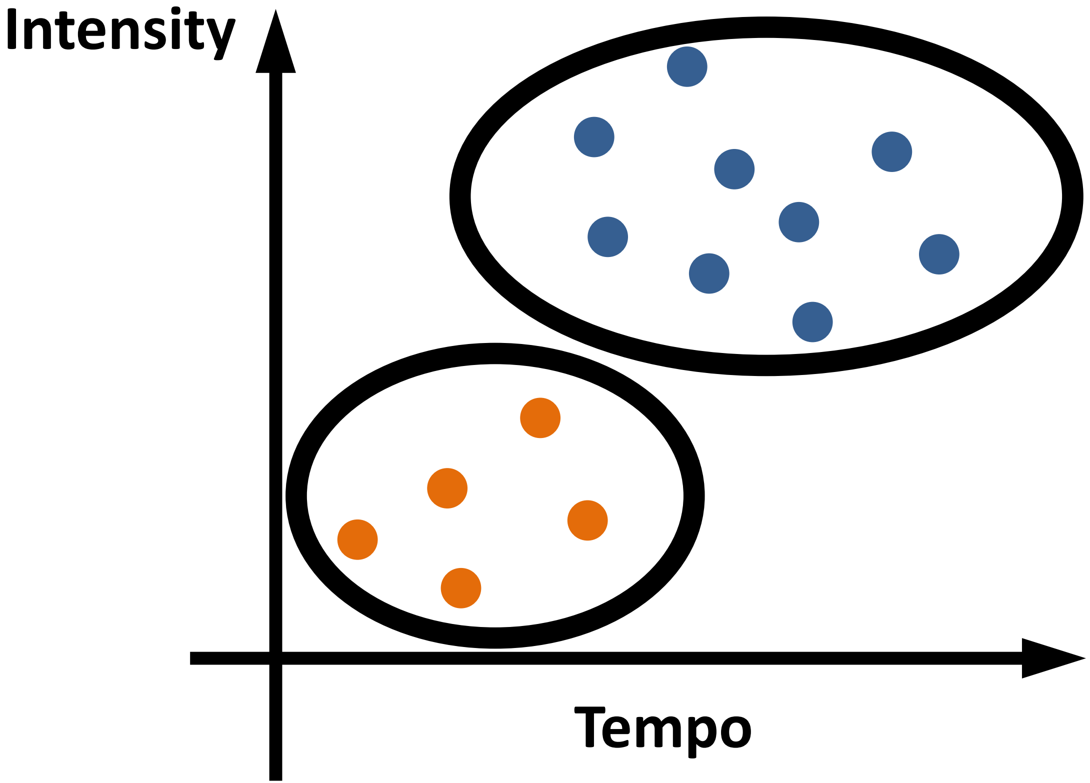
User Sharon



Example 1: Predict whether a user likes a song or not

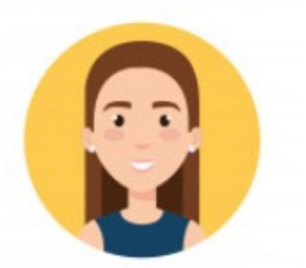


User Sharon



- Dislike
- Like

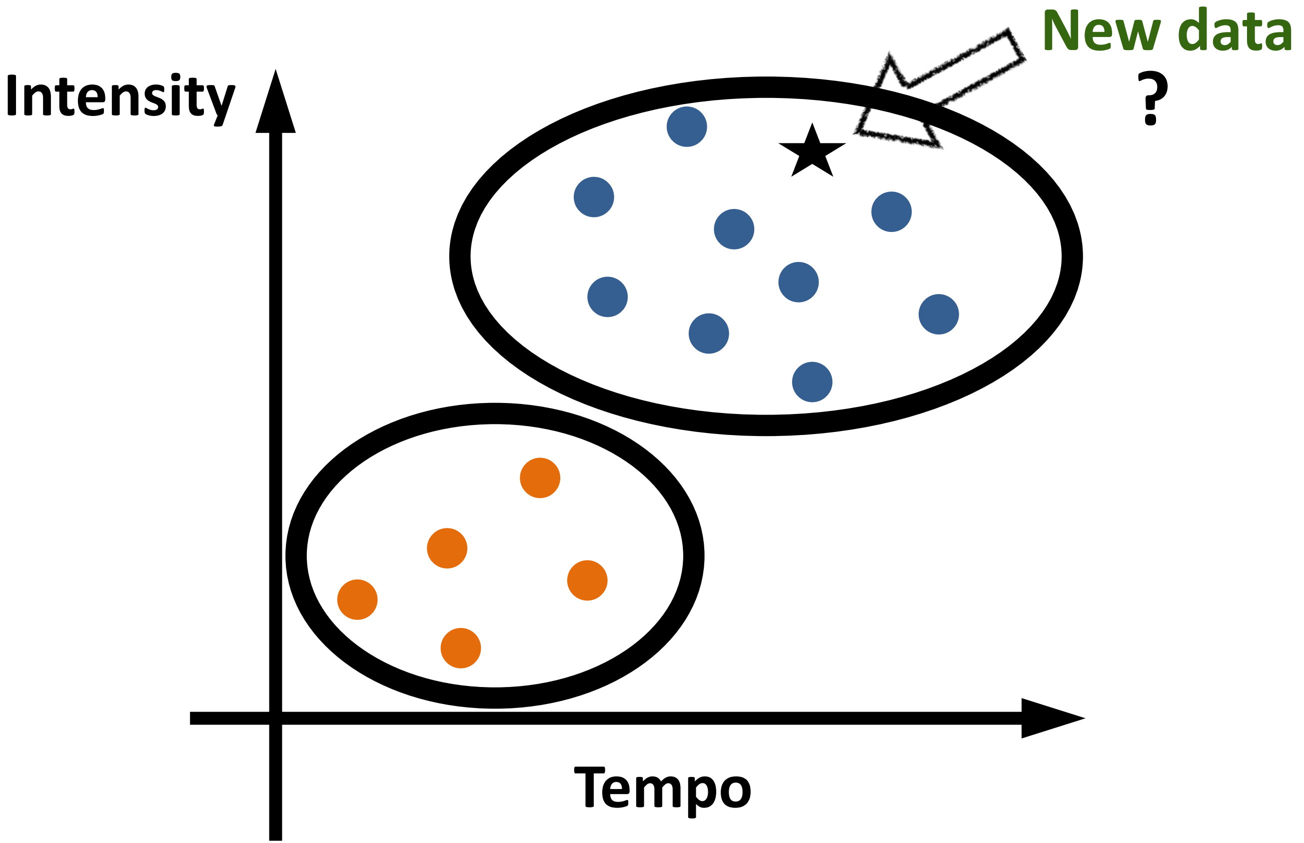
Example 1: Predict whether a user likes a song or not



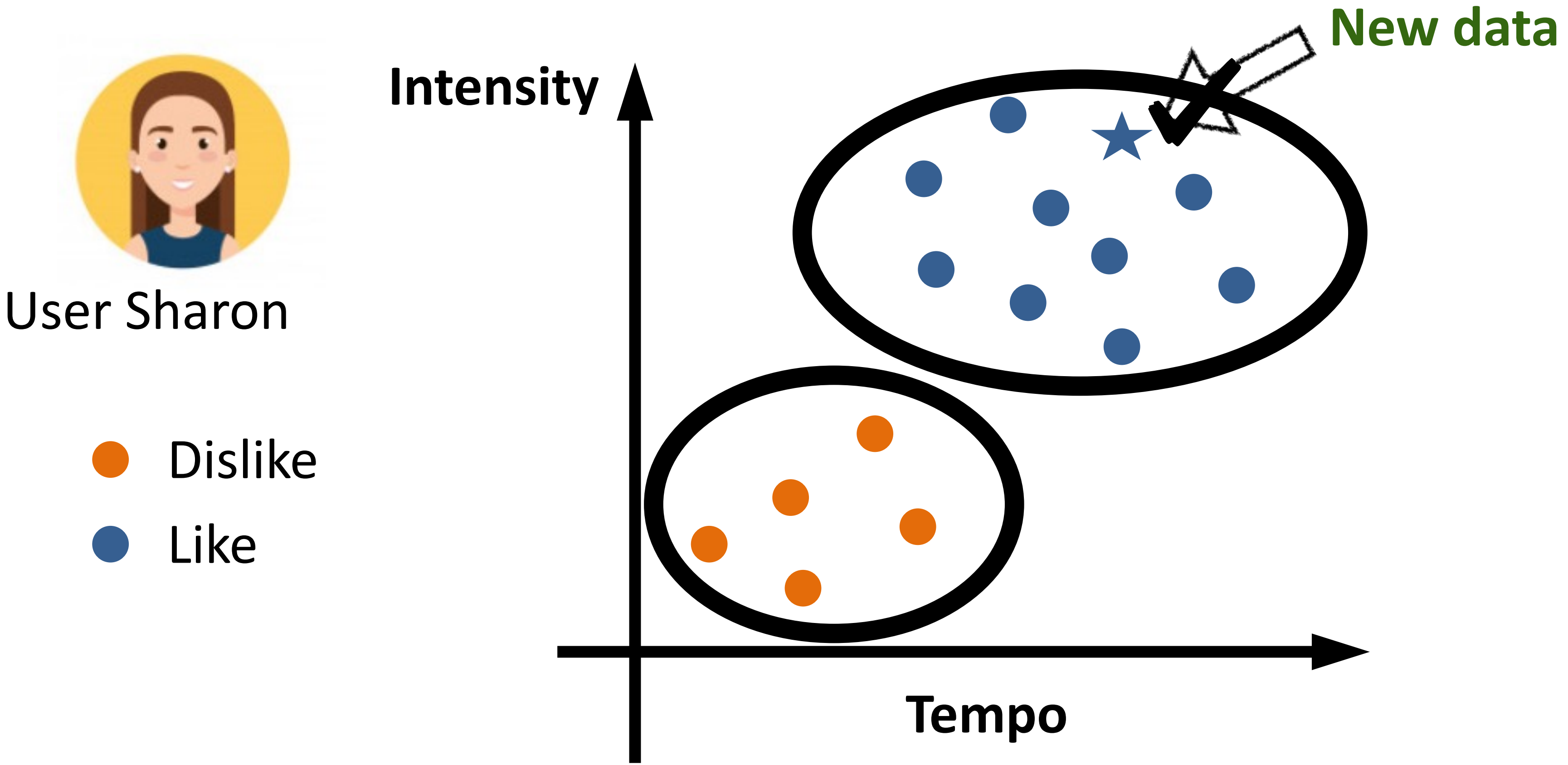
User Sharon

Dislike

Like

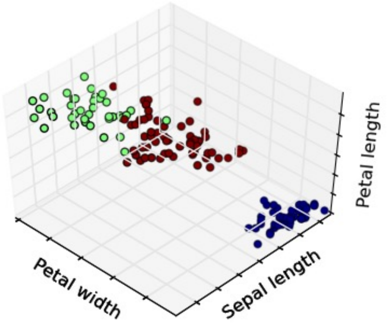


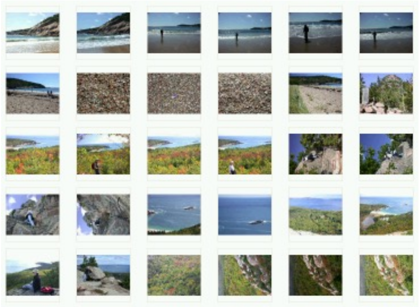
Example 1: Predict whether a user likes a song or not

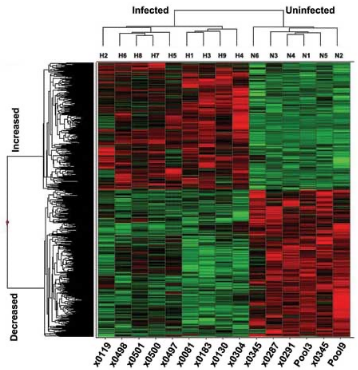


Unsupervised Learning

- No labels; generally won't be making predictions
- Dataset: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$
- Goal: find patterns/structures that help better understand data
  - E.g., dimension reduction, clustering, …

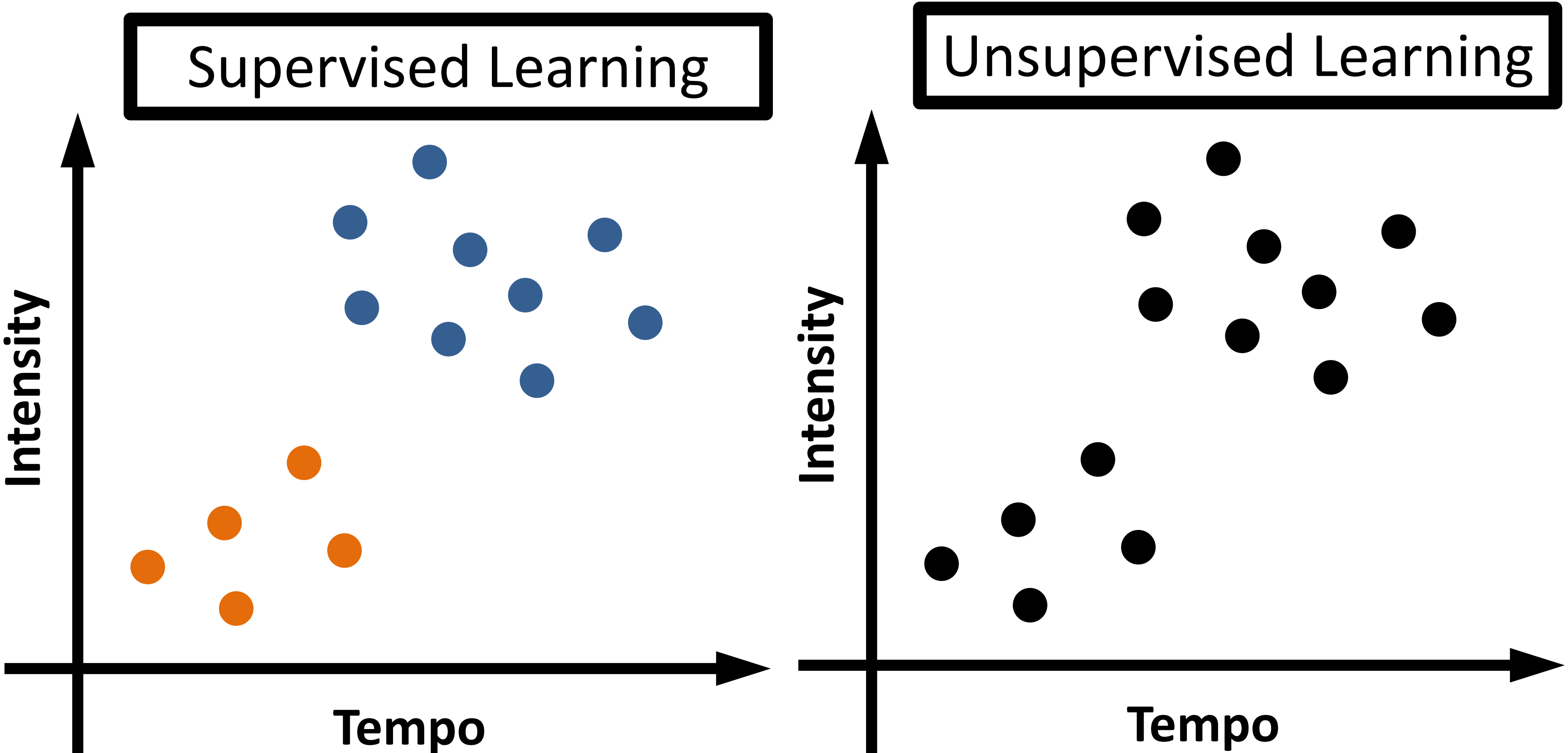






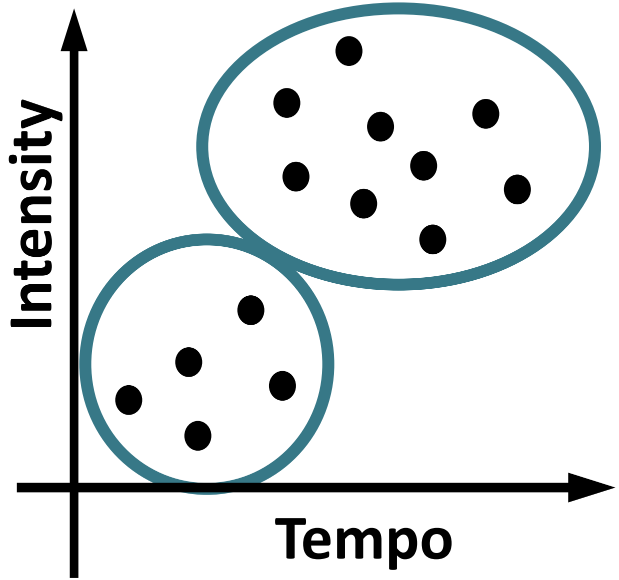
Mulvey and Gingold

Unsupervised Learning



Clustering

- Given: dataset contains **no label** $x_1, x_2, \ldots, x_n$
- **Output:** divides the data into clusters such that there are intra-cluster similarity and inter-cluster dissimilarity



Unsupervised Learning (**UL**)

- Clustering is just one type of unsupervised learning
  - PCA is another unsupervised algorithm
  - So is language modelling.
- Estimating probability distributions also UL (GANs)
- Clustering is popular & useful!

StyleGAN2 (Kerras et al ’20)

Reinforcement Learning

Reinforcement Learning

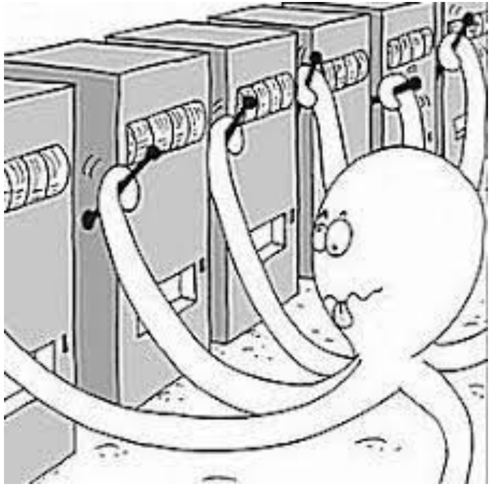
- Given: an agent that can take actions and a reward function specifying how good an action is.
- **Goal:** learn to choose actions that maximize future reward total.

Google Deepmind

Reinforcement Learning Key Problems

1. Problem: actions may have delayed effects.
   – Requires **credit-assignment**
2. Problem: maximal reward action is unknown
   – Exploration-exploitation trade-off

"..the problem [exploration-exploitation] was proposed [by British scientist] to be dropped over Germany so that German scientists could also waste their time on it."
- Peter Whittle



Multi-armed Bandit

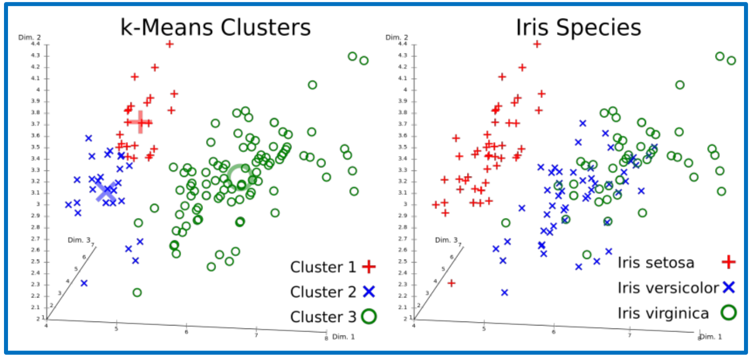
Today: Clustering

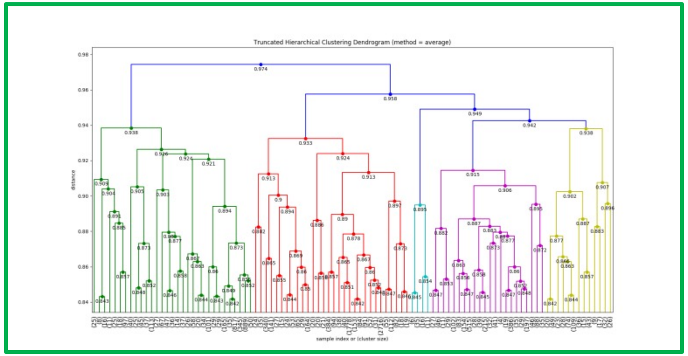
- Several types of clustering

Partitional
- Center-based
- Graph-theoretic
- Spectral

Hierarchical
- Agglomerative
- Divisive

Bayesian
- Decision-based
- Nonparametric

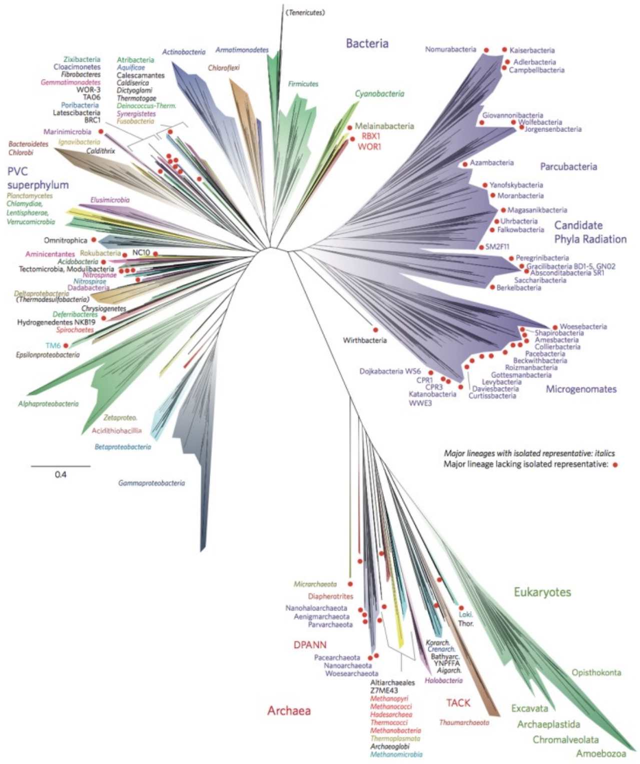




Hierarchical Clustering

Basic idea: build a “hierarchy”

- Want: arrangements from specific to general
- One advantage: no need for k, number of clusters.
- **Input**: points. **Output**: a hierarchy
  - A binary tree

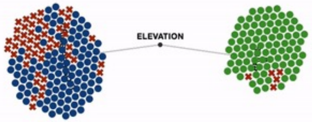


Credit: Wikipedia

Agglomerative vs Divisive

Two ways to go:

- **Agglomerative**: bottom up.
  - Start: each point a cluster. Progressively merge clusters
- **Divisive**: top down
  - Start: all points in one cluster. Progressively split clusters



Credit: r2d3.us

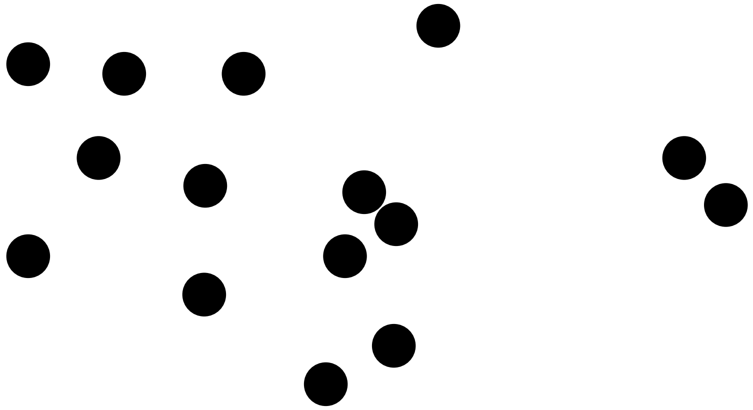
Hierarchical Agglomerative Clustering (HAC)

Input: data points $x_1, \ldots, x_n \in R^m$, cluster distance function $d(A, B)$

1. Initialize n clusters, one data point each
2. While (number of clusters > 1)
3. find the closest clusters $c_1, c_2 = \operatorname{argmin}_{A,B} d(A, B)$ over all cluster pairs A, B
4. merge c_1, c_2 into a new cluster, remove c_1, c_2

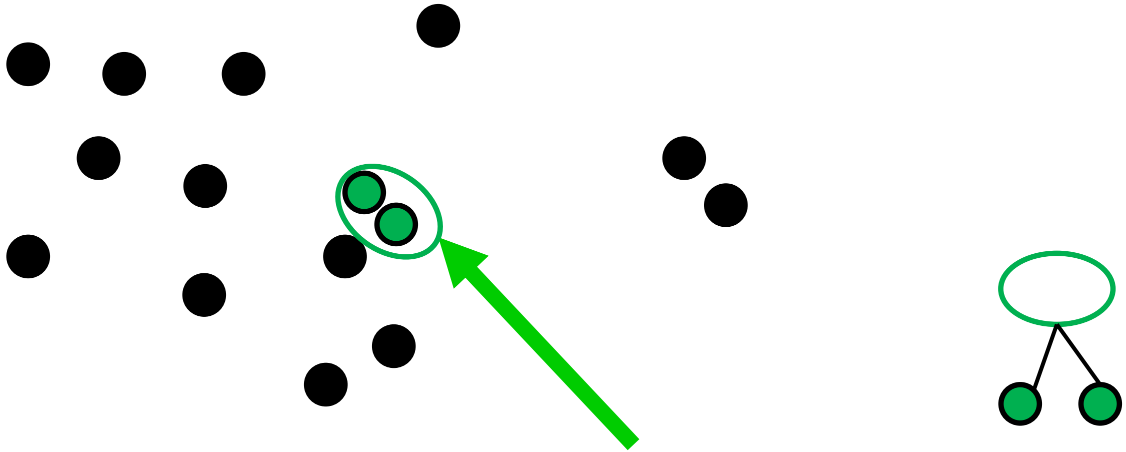
Agglomerative Clustering Example

Agglomerative. Start: every point is its own cluster



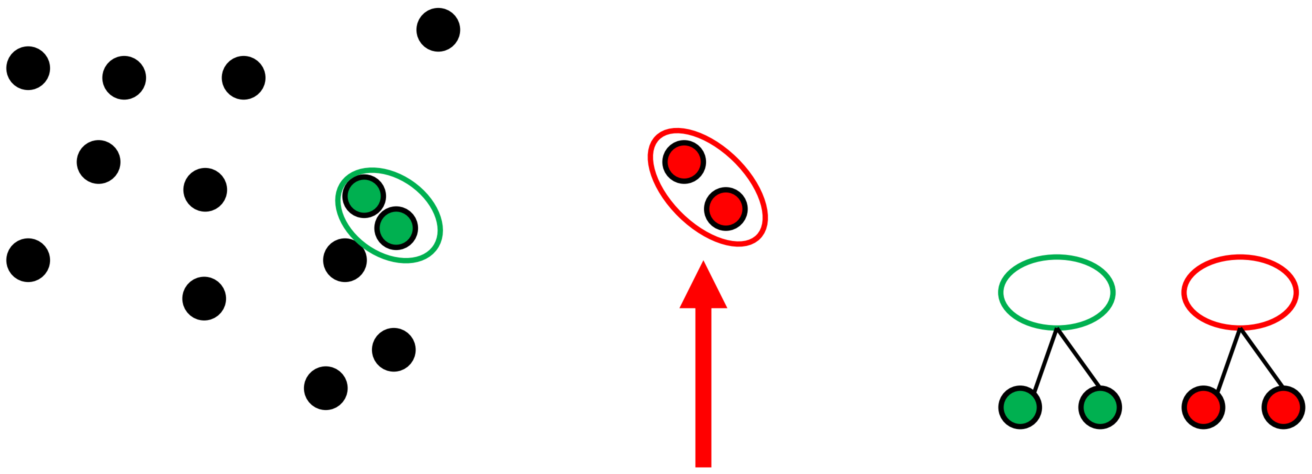
Agglomerative Clustering Example

Get pair of clusters that are closest and merge



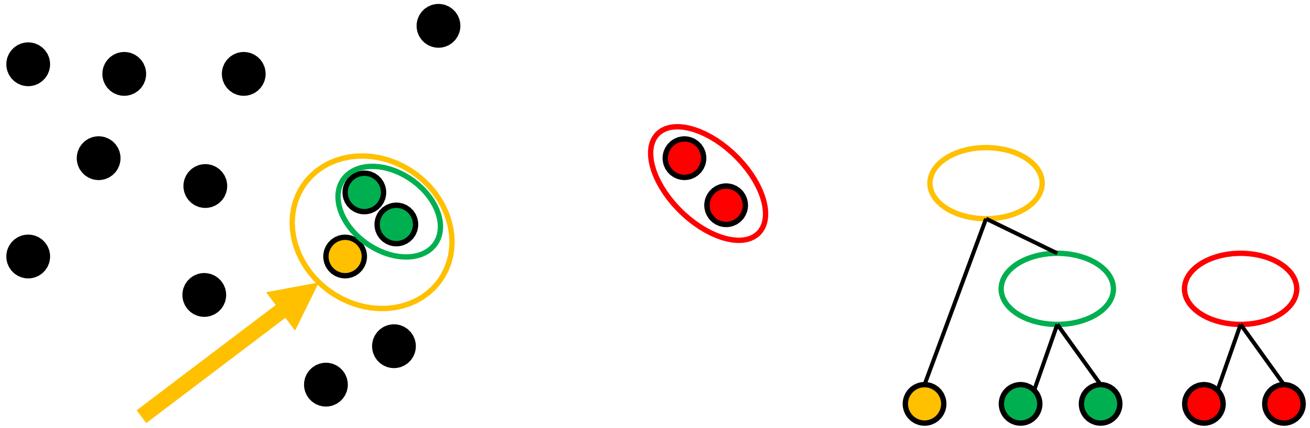
Agglomerative Clustering Example

Repeat: Get pair of clusters that are closest and merge



Agglomerative Clustering Example

Repeat: Get pair of clusters that are closest and merge



Cluster Distance Function

Merge: use closest clusters. Define closest?

- Single-linkage

$$d(A, B) = \min_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$

- Complete-linkage

$$d(A, B) = \max_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$

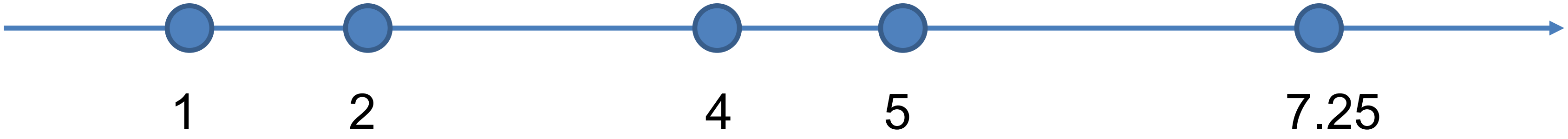
- Average-linkage

$$d(A, B) = \frac{1}{|A||B|} \sum_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$

Single-linkage Example

We'll merge using single-linkage

- 1-dimensional vectors.
- Initial: all points are clusters

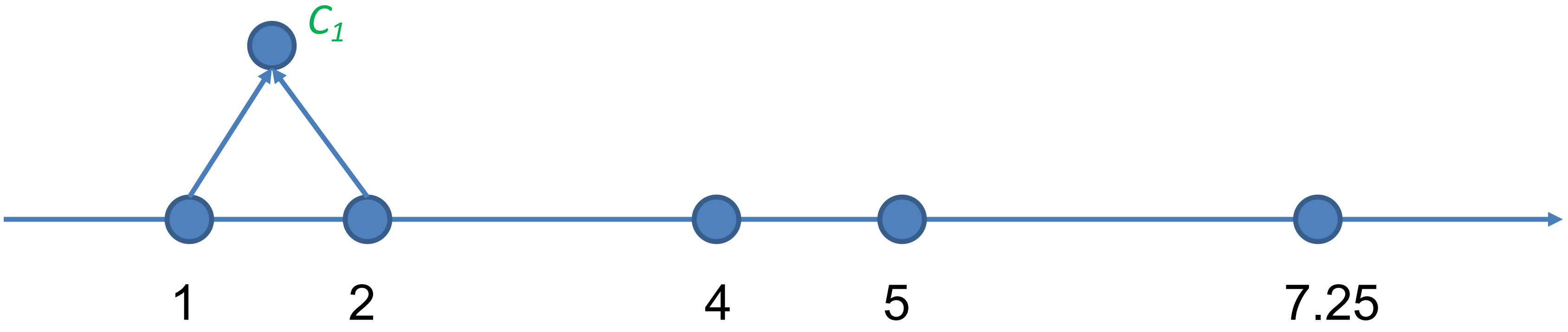


Single-linkage Example

We'll merge using single-linkage

$$d(C_1, \{4\}) = d(2,4) = 2$$

$$d(\{4\}, \{5\}) = d(4,5) = 1$$

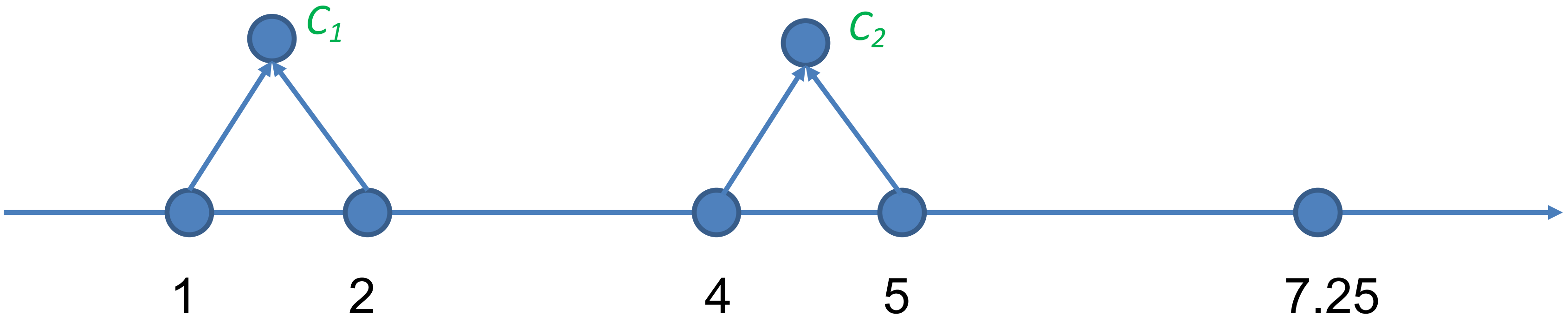


Single-linkage Example

Continue…

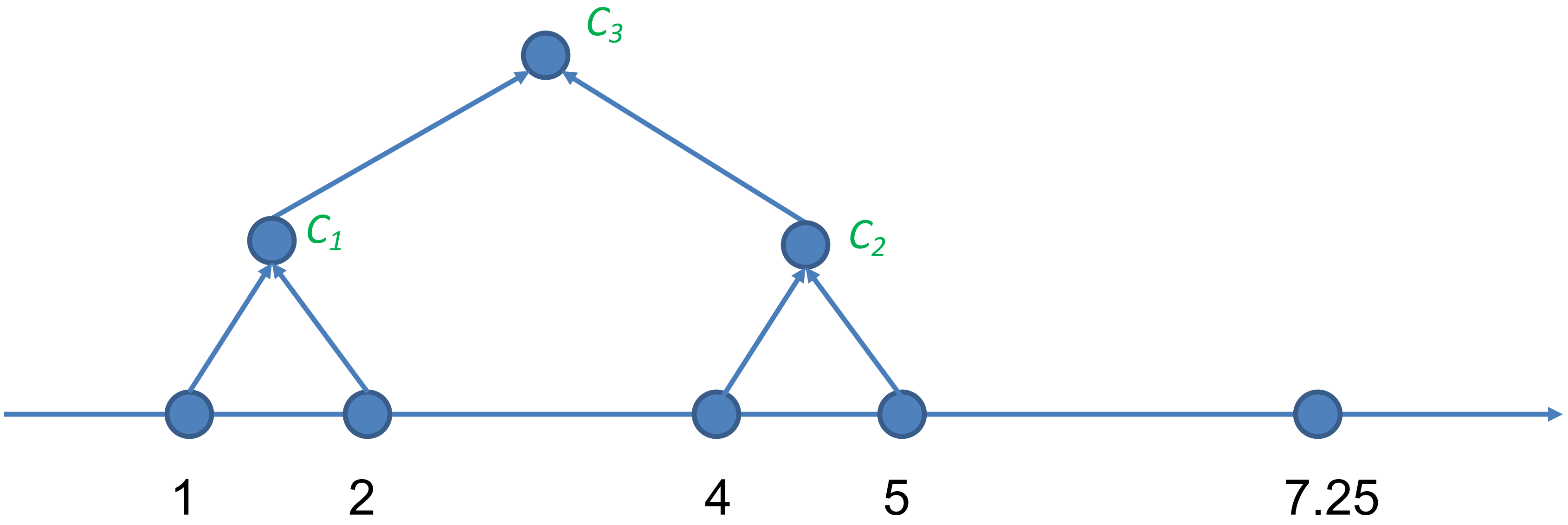
$$d(C_1, C_2) = d(2, 4) = 2$$

$$d(C_2, \{7.25\}) = d(5, 7.25) = 2.25$$

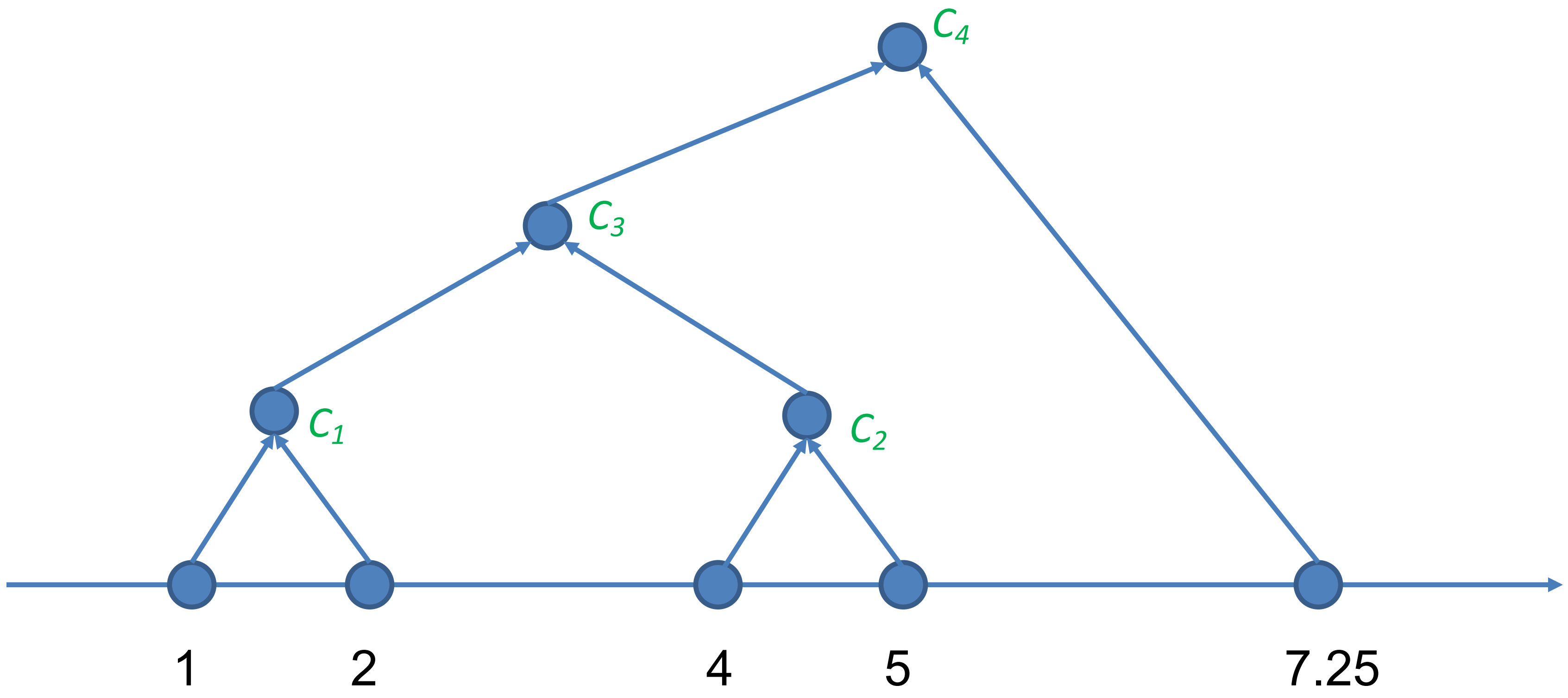


Single-linkage Example

Continue…



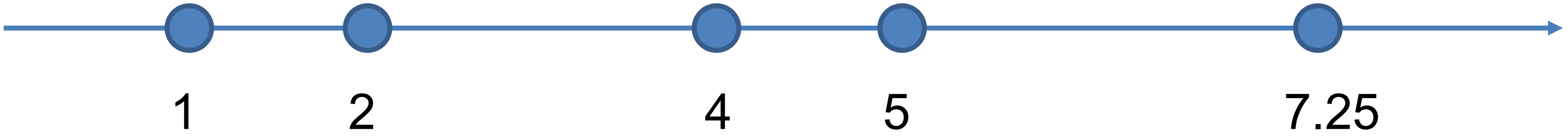
Single-linkage Example



Complete-linkage Example

We'll merge using complete-linkage

- 1-dimensional vectors.
- Initial: all points are clusters

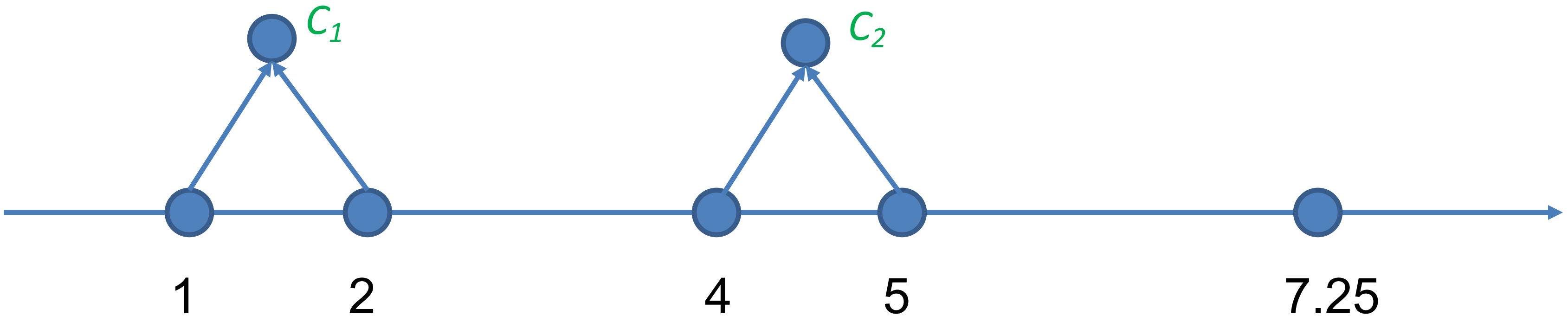


Complete-linkage Example

Beginning is the same…

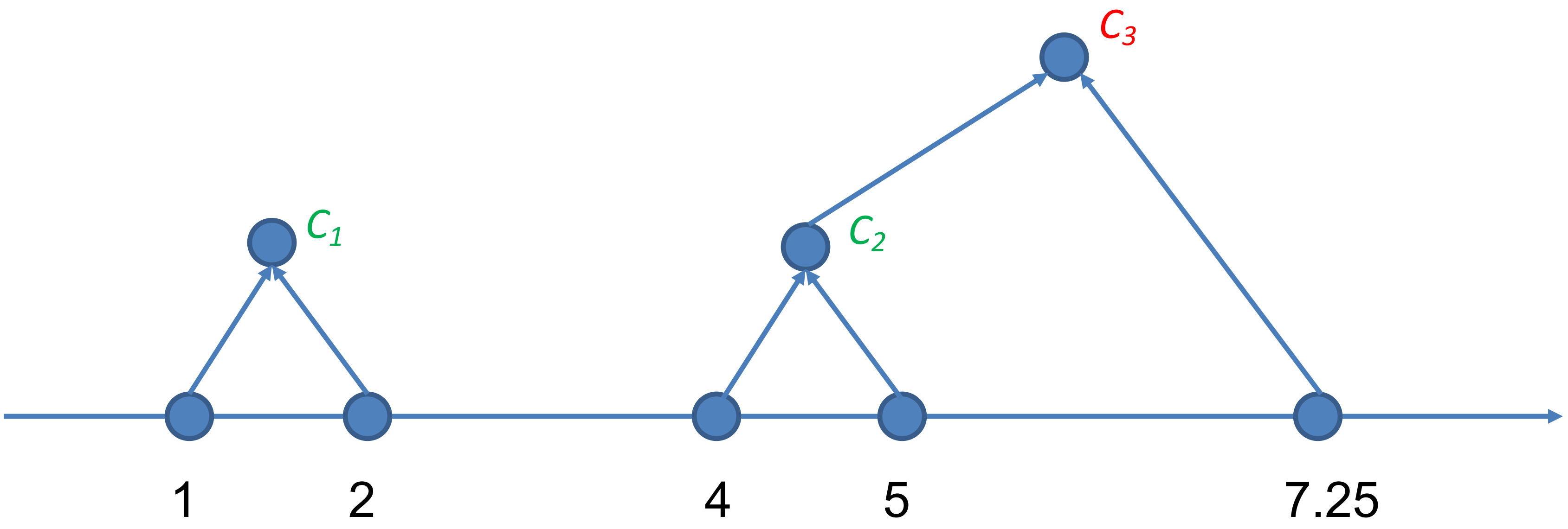
$$d(C_1, C_2) = d(1, 5) = 4$$

$$d(C_2, \{7.25\}) = d(4, 7.25) = 3.25$$

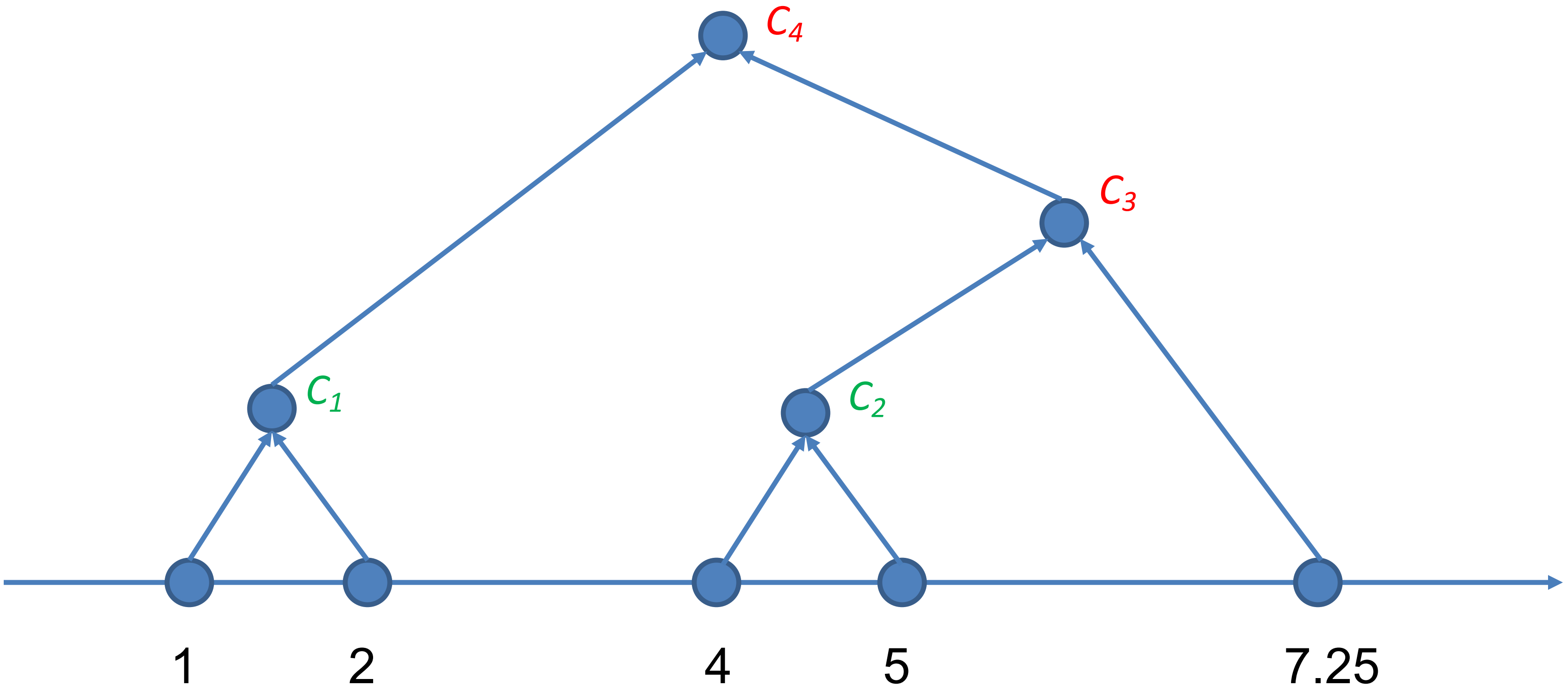


Complete-linkage Example

Now we diverge:



Complete-linkage Example

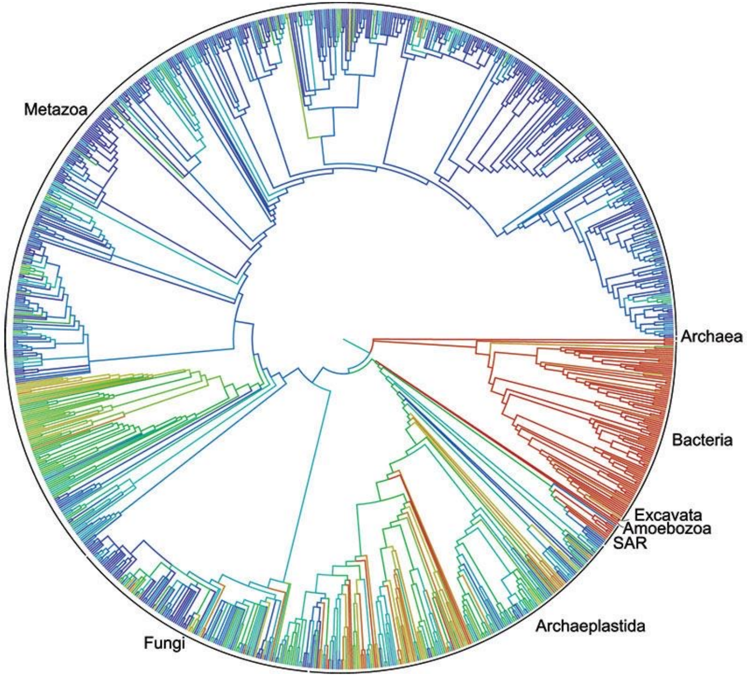


When to Stop?

No simple answer:

- Use the binary tree (a **dendrogram**)

- Cut at different levels (get different heights/depths)

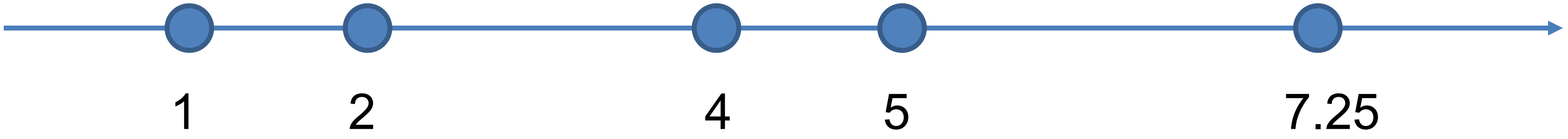


http://opentreeoflife.org/

Break & Quiz

Q 1.1: Let's do hierarchical clustering for two clusters with average linkage on the dataset below. What are the clusters?

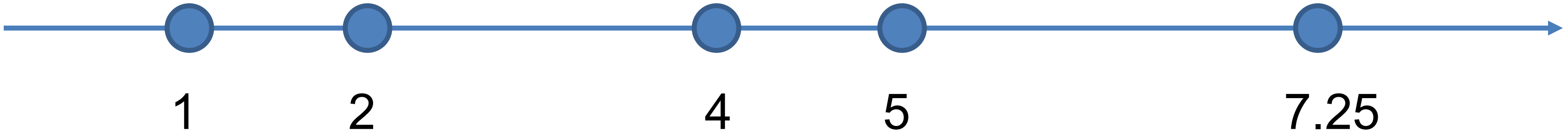
- A. {1}, {2,4,5,7.25}
- B. {1,2}, {4, 5, 7.25}
- C. {1,2,4}, {5, 7.25}
- D. {1,2,4,5}, {7.25}



Break & Quiz

Q 1.1: Let’s do hierarchical clustering for two clusters with average linkage on the dataset below. What are the clusters?

- A. {1}, {2,4,5,7.25}
- **B. {1,2}, {4, 5, 7.25}**
- C. {1,2,4}, {5, 7.25}
- D. {1,2,4,5}, {7.25}



Break & Quiz

Q 1.2: If we do hierarchical clustering on n points, the maximum depth of the resulting tree is

- A. 2
- B. log *n*
- C. *n*/2
- D. *n*-1

Break & Quiz

Q 1.2: If we do hierarchical clustering on n points, the maximum depth of the resulting tree is

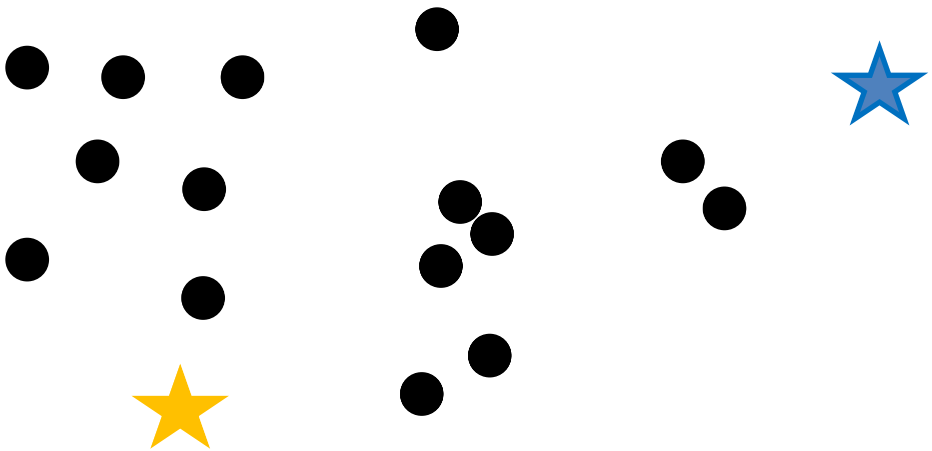
- A. 2
- B. log *n*
- C. *n*/2
- **D. *n*-1**

Center-based Clustering

- k-means is an example of a partitional, **center-based clustering algorithm.**
- Specify a desired number of clusters, k; run k-means to find k clusters.

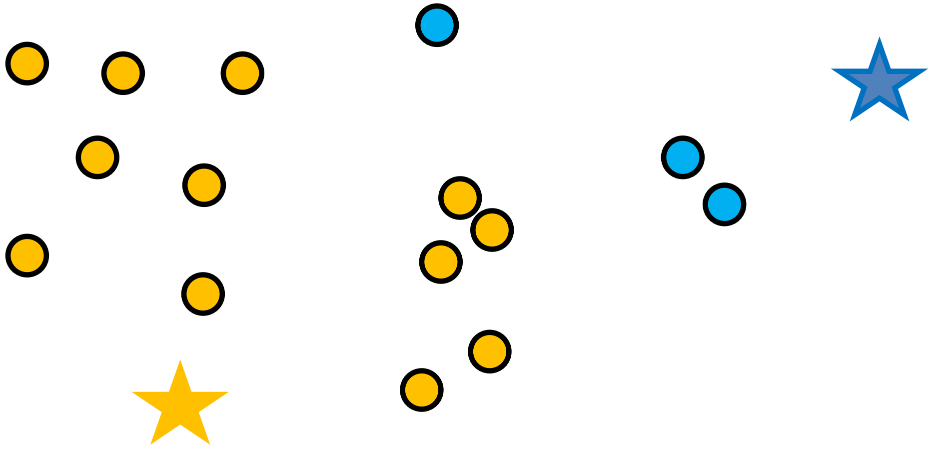
Center-based Clustering

- Steps: **1.** Randomly pick k cluster centers



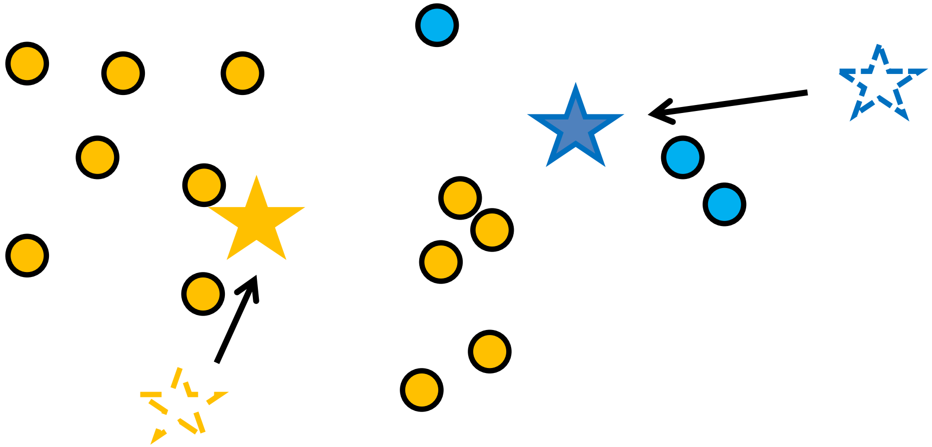
Center-based Clustering

- **2.** Find closest center for each point



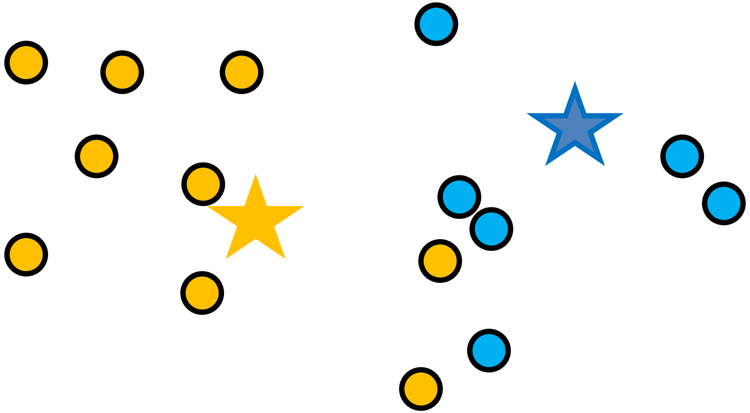
Center-based Clustering

- **3.** Update cluster centers by computing centroids



Center-based Clustering

- Repeat Steps 2 & 3 until convergence



K-means algorithm

- Input: $x_1, x_2, \ldots, x_n, k$
- Step 1: select k cluster centers $c_1, c_2, \ldots, c_k$
- Step 2: for each point x_i, assign it to the closest center in Euclidean distance:

$$y(x_i) = \mathrm{argmin}_j \, ||x_i - c_j||$$

- Step 3: update all cluster centers as the centroids:

$$c_j = \frac{\sum_{x:y(x)=j} x}{\sum_{x:y(x)=j} 1}$$

- Repeat Step 2 and 3 until cluster centers no longer change

Break & Quiz

Q 2.1: You have seven 2-dimensional points. You run 3-means on it, with initial clusters

$$C_1 = \{(2,2),(4,4),(6,6)\}, C_2 = \{(0,4),(4,0)\}, C_3 = \{(5,5),(9,9)\}$$

Cluster centroids are updated to?

- A. C_1: (4,4), C_2: (2,2), C_3: (7,7)
- B. C_1: (6,6), C_2: (4,4), C_3: (9,9)
- C. C_1: (2,2), C_2: (0,0), C_3: (5,5)
- D. C_1: (2,6), C_2: (0,4), C_3: (5,9)

Break & Quiz

Q 2.1: You have seven 2-dimensional points. You run 3-means on it, with initial clusters

$$C_1 = \{(2,2),(4,4),(6,6)\}, C_2 = \{(0,4),(4,0)\}, C_3 = \{(5,5),(9,9)\}$$

Cluster centroids are updated to?

- **A. C_1: (4,4), C_2: (2,2), C_3: (7,7)**
- B. C_1: (6,6), C_2: (4,4), C_3: (9,9)
- C. C_1: (2,2), C_2: (0,0), C_3: (5,5)
- D. C_1: (2,6), C_2: (0,4), C_3: (5,9)

Break & Quiz

Q 2.1: You have seven 2-dimensional points. You run 3-means on it, with initial clusters

$$C_1 = \{(2,2),(4,4),(6,6)\}, C_2 = \{(0,4),(4,0)\}, C_3 = \{(5,5),(9,9)\}$$

Cluster centroids are updated to?

- **A. C_1: (4,4), C_2: (2,2), C_3: (7,7)**
- B. C_1: (6,6), C_2: (4,4), C_3: (9,9)
- C. C_1: (2,2), C_2: (0,0), C_3: (5,5)
- D. C_1: (2,6), C_2: (0,4), C_3: (5,9)

The average of points in C1 is (4,4).
The average of points in C2 is (2,2).
The average of points in C3 is (7,7).

Break & Quiz

Q 2.2: We are running 3-means again. We have 3 centers, C_1 (0,1), C_2, (2,1), C_3 (-1,2). Which cluster assignment is possible for the points (1,1) and (-1,1), respectively? Ties are broken arbitrarily:

(i) C_1, C_1 (ii) C_2, C_3 (iii) C_1, C_3

- A. Only (i)
- B. Only (ii) and (iii)
- C. Only (i) and (iii)
- D. All of them

Break & Quiz

Q 2.2: We are running 3-means again. We have 3 centers, C_1 (0,1), C_2, (2,1), C_3 (-1,2). Which cluster assignment is possible for the points (1,1) and (-1,1), respectively? Ties are broken arbitrarily:

(i) C_1, C_1 (ii) C_2, C_3 (iii) C_1, C_3

- A. Only (i)
- B. Only (ii) and (iii)
- C. Only (i) and (iii)
- **D. All of them**

Break & Quiz

Q 2.2: We are running 3-means again. We have 3 centers, C_1 (0,1), C_2, (2,1), C_3 (-1,2). Which cluster assignment is possible for the points (1,1) and (-1,1), respectively? Ties are broken arbitrarily:

(i) C_1, C_1 (ii) C_2, C_3 (iii) C_1, C_3

- A. Only (i)
- B. Only (ii) and (iii)
- C. Only (i) and (iii)
- **D. All of them**

Break & Quiz

Q 2.2: We are running 3-means again. We have 3 centers, C_1 (0,1), C_2, (2,1), C_3 (-1,2). Which cluster assignment is possible for the points (1,1) and (-1,1), respectively? Ties are broken arbitrarily:

(i) C_1, C_1 (ii) C_2, C_3 (iii) C_1, C_3

- A. Only (i)
- B. Only (ii) and (iii)
- C. Only (i) and (iii)
- **D. All of them**

For the point (1,1): square-Euclidean-distance to C1 is 1, to C2 is 1, to C3 is 5
So it can be assigned to C1 or C2

For the point (-1,1): square-Euclidean-distance to C1 is 1, to C2 is 9, to C3 is 1
So it can be assigned to C1 or C3

Break & Quiz

Q 2.3: If we run K-means clustering twice with random starting cluster centers, are we guaranteed to get same clustering results? Does K-means always converge?

- A. Yes, Yes
- B. No, Yes
- C. Yes, No
- D. No, No

Break & Quiz

Q 2.3: If we run K-means clustering twice with random starting cluster centers, are we guaranteed to get same clustering results? Does K-means always converge?

- A. Yes, Yes
- **B. No, Yes**
- C. Yes, No
- D. No, No

Break & Quiz

Q 2.3: If we run K-means clustering twice with random starting cluster centers, are we guaranteed to get same clustering results? Does K-means always converge?

- A. Yes, Yes
- **B. No, Yes**
- C. Yes, No
- D. No, No

The clustering from k-means will depend on the initialization. Different initialization can lead to different outcomes.

K-means will always converge on a finite set of data points:

1. There are finite number of possible partitions of the points
2. The assignment and update steps of each iteration will only decrease the sum of the distances from points to their corresponding centers.
3. If it run forever without convergence, it will revisit the same partition, which is contradictory to item 2.