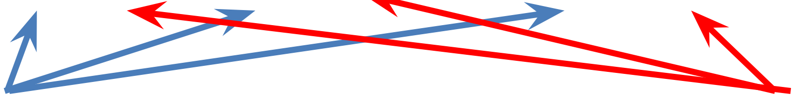
CS 540 Introduction to Artificial Intelligence
Unsupervised Learning I
University of Wisconsin-Madison

Spring 2022

Recap of Supervised/Unsupervised

Supervised learning:

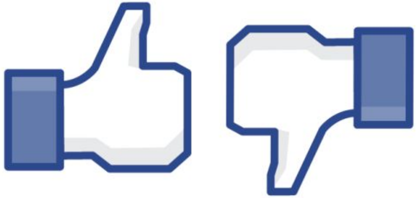
- Make predictions, classify data, perform regression
- Dataset: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$



Features / Covariates / Input

Labels / Outputs

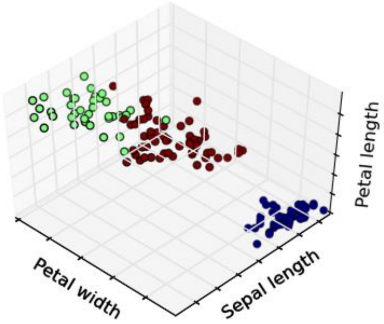
- Goal: find function $f : X \to Y$ to predict label on **new** data

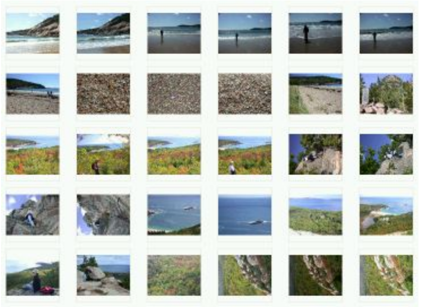


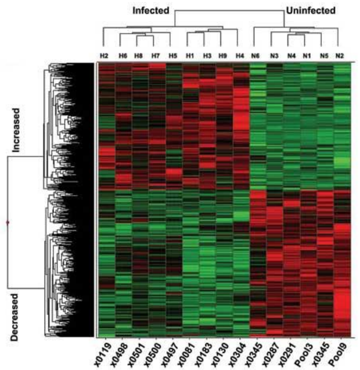
Recap of Supervised/Unsupervised

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.







Mulvey and Gingold

Outline

- Intro to Clustering
- K-means clustering
- Hierarchical Agglomerative Clustering
- Other Clustering Types

Recap of Supervised/Unsupervised

Note that there are **other kinds** of ML:

- Mixtures: semi-supervised learning, self-supervised
  - Idea: different types of “signal”
- Reinforcement learning
  - Learn how to act in order to maximize rewards
  - Later on in course…

DeepMind

Unsupervised Learning & Clustering

- Note that clustering is just one type of unsupervised learning (**UL**)
  - PCA is another unsupervised algorithm
- Estimating probability distributions also UL (GANs)
- Clustering is popular & useful!

StyleGAN2 (Kerras et al '20)

There are many clustering algorithms

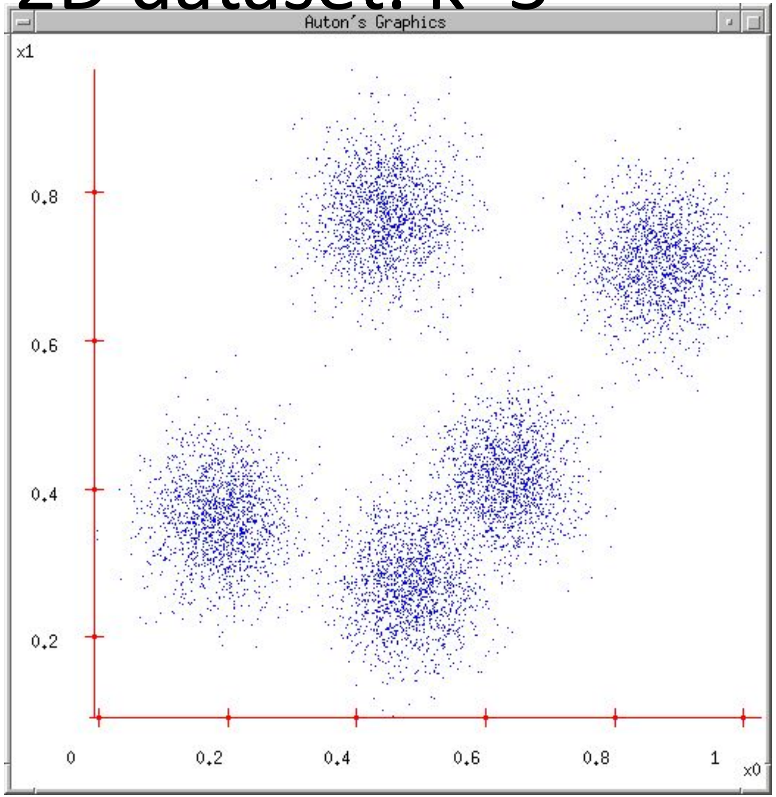
- K-means algorithm
- HAC (Hierarchical Agglomerative Clustering) algorithm
- Spectral clustering algorithm
- t-SNE (t-distributed stochastic neighbor embedding)
- …

K-means clustering

- Input:
  - A dataset $x_1, \ldots, x_n$, each point is a feature vector
  - Assume the number of desired clusters, k, is given

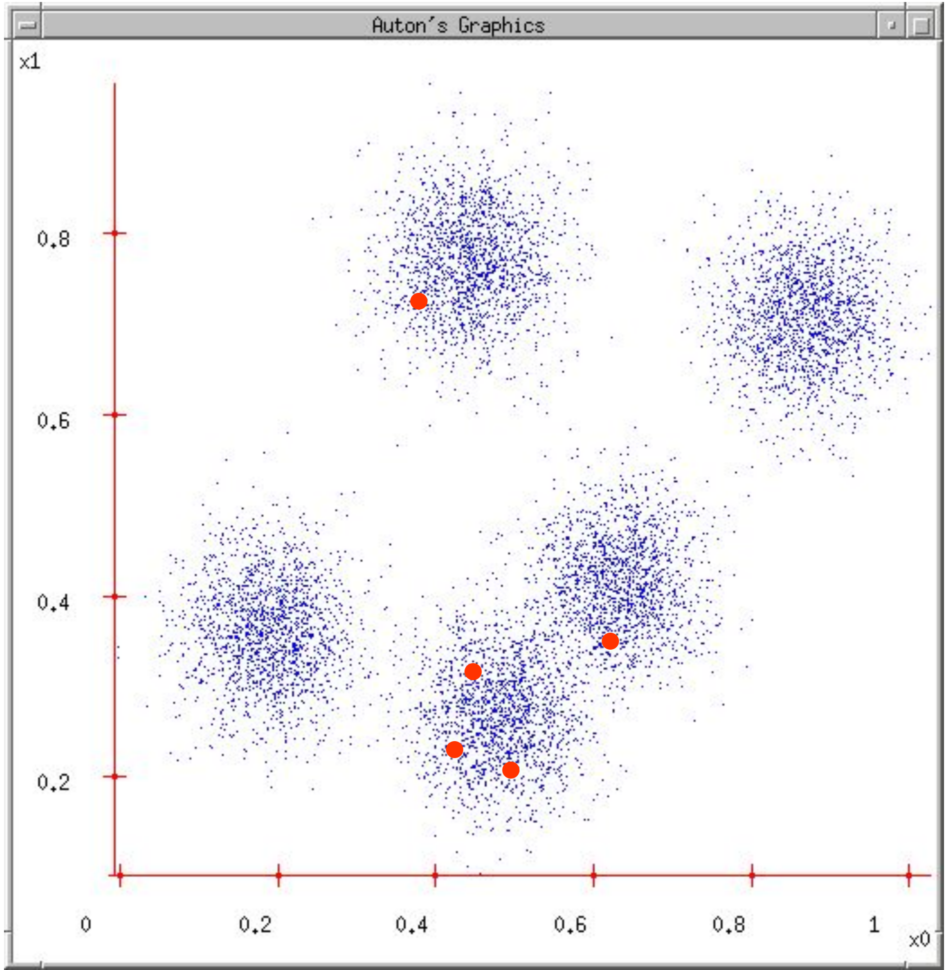
K-means clustering demo

- The 2D dataset. k=5



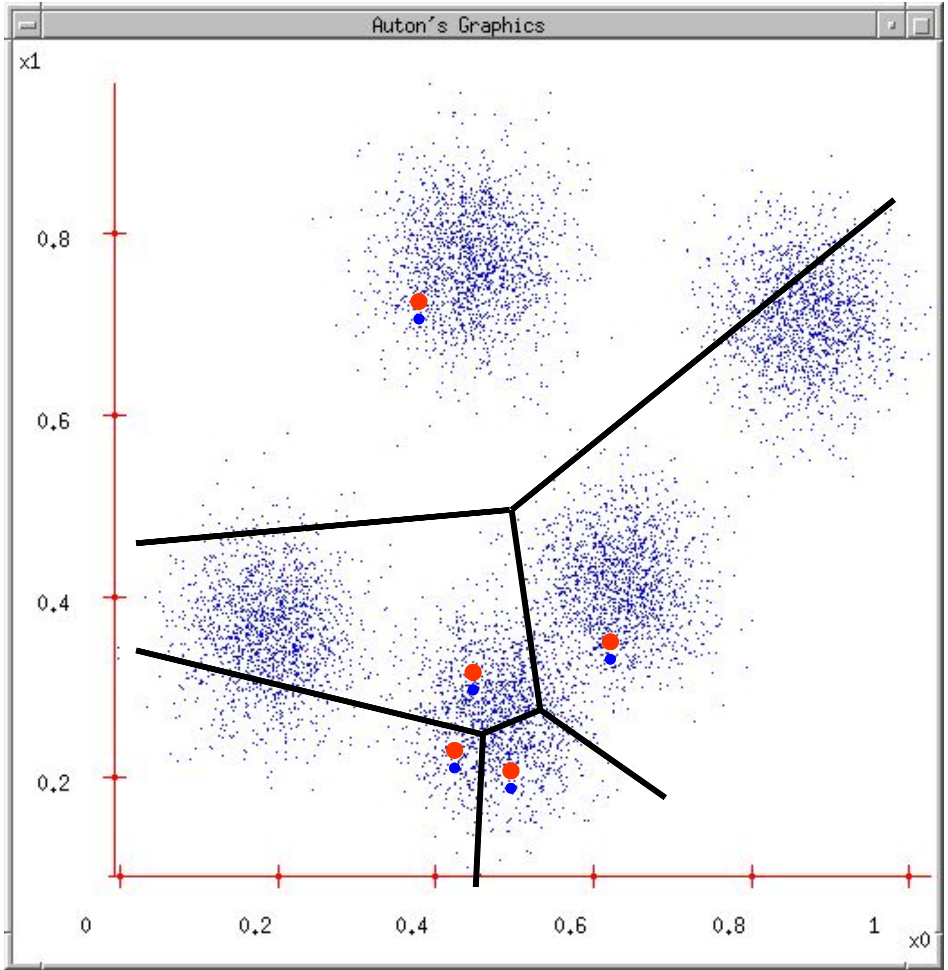
K-means clustering

- Randomly picking 5 positions as initial cluster centers (not necessarily a data point)



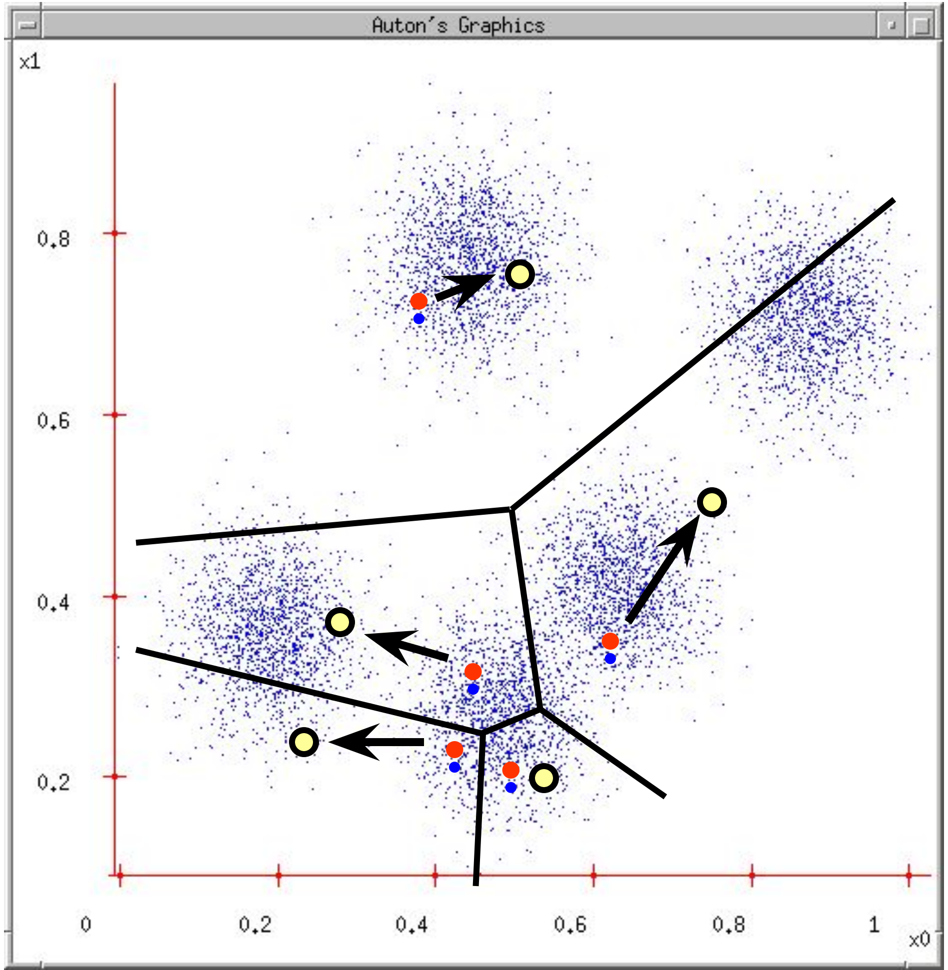
K-means clustering

- Each point finds which cluster center it is closest to (very much like 1NN). The point belongs to that cluster.



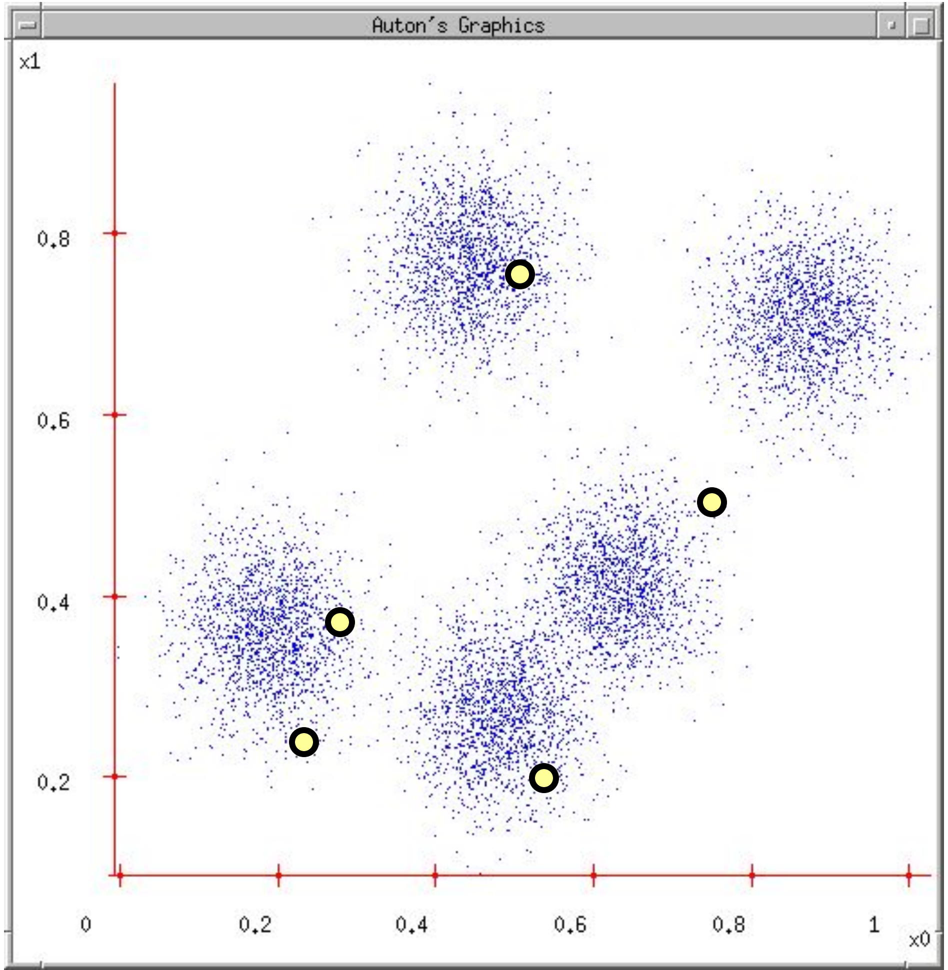
K-means clustering

- Each cluster computes its new centroid, based on which points belong to it

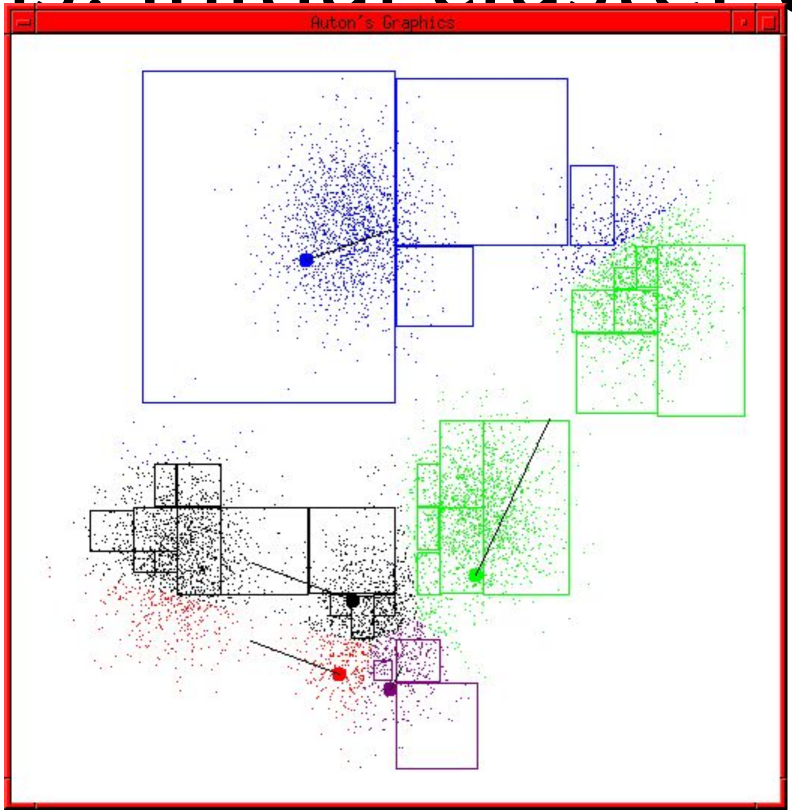


K-means clustering

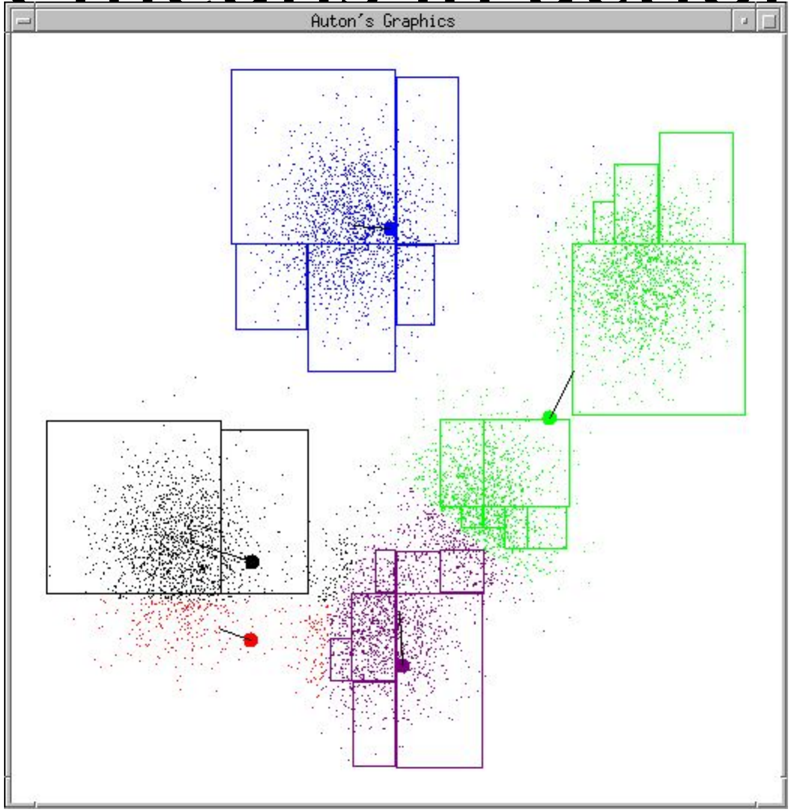
- Each cluster computes its new centroid, based on which points belong to it
- And repeat until convergence (cluster centers no longer move)…



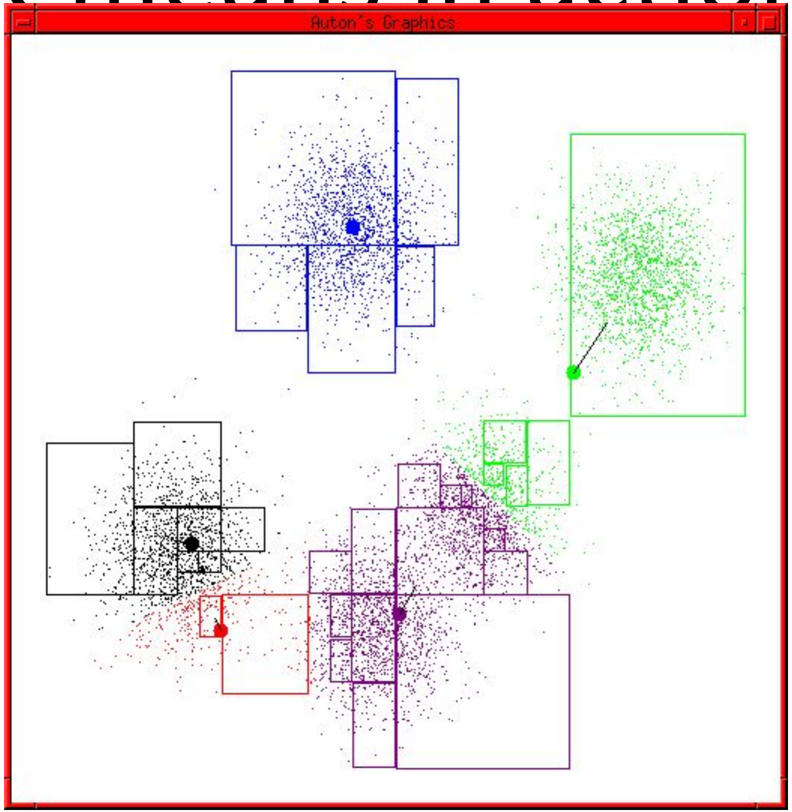
K-means: initial cluster centers



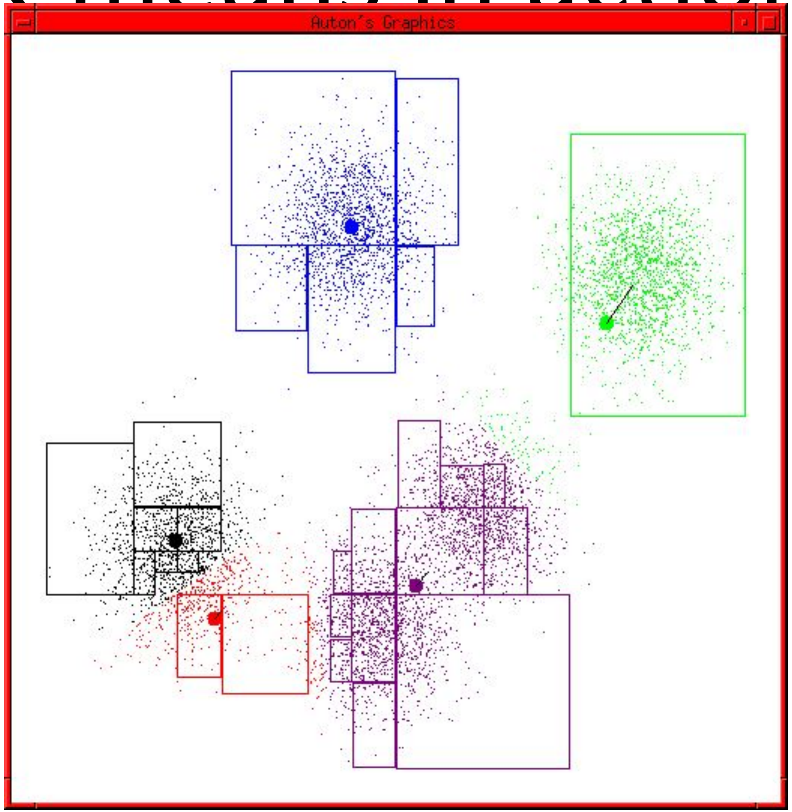
K-means in action



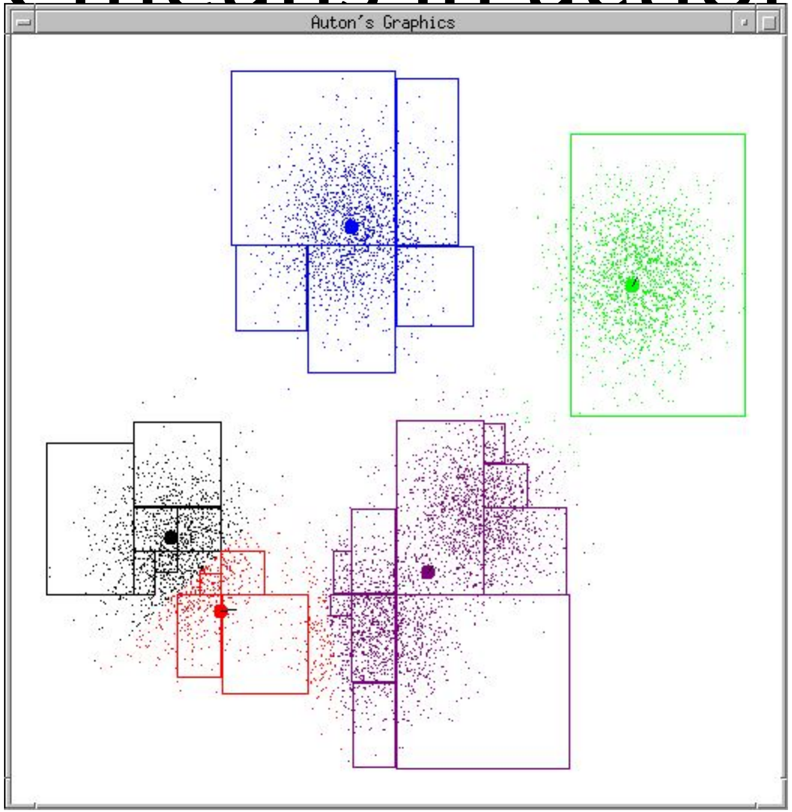
K-means in action



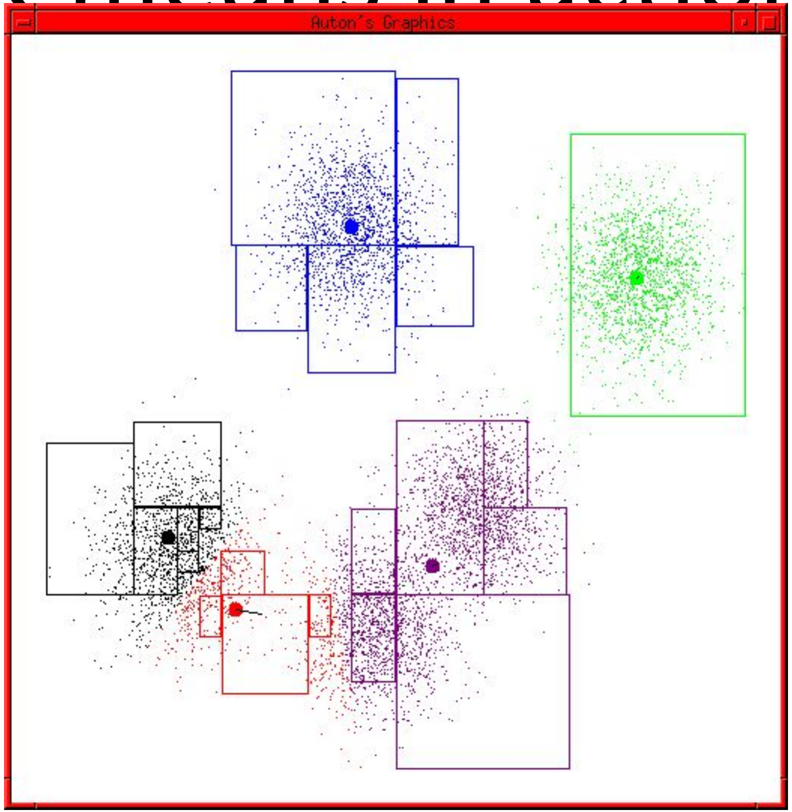
K-means in action



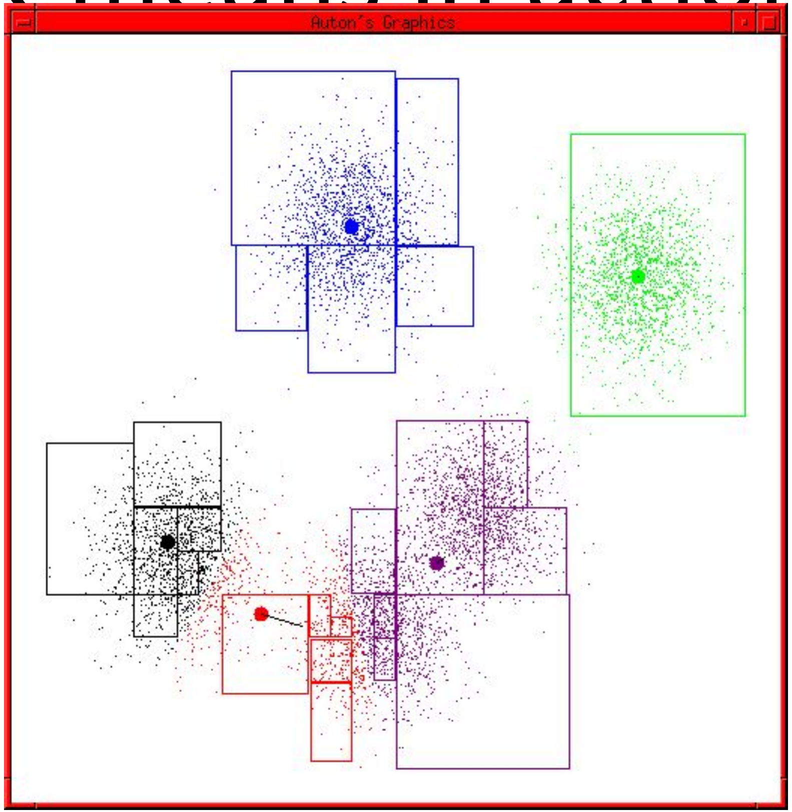
K-means in action



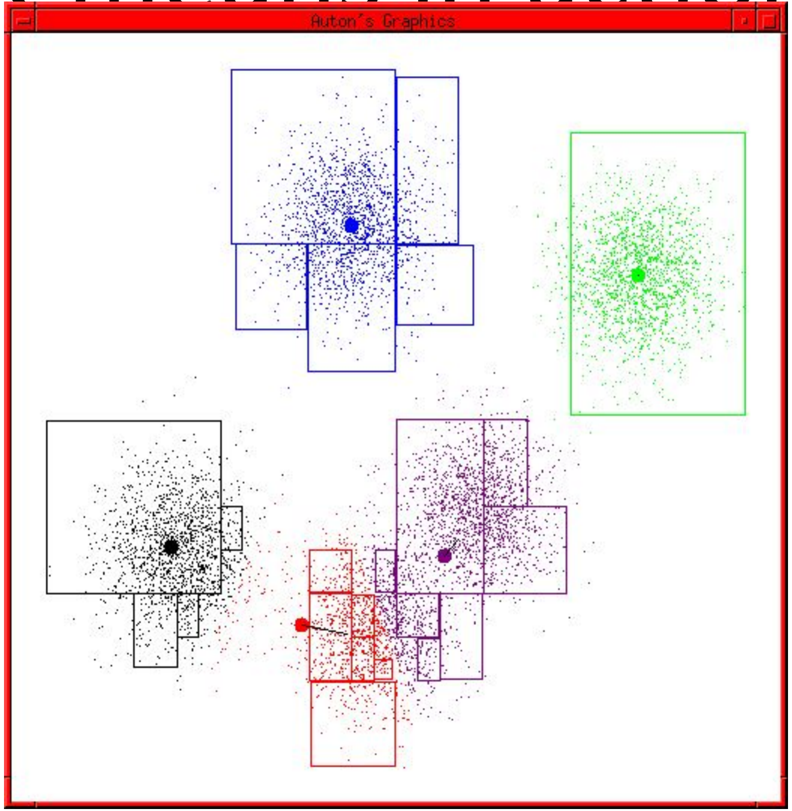
K-means in action



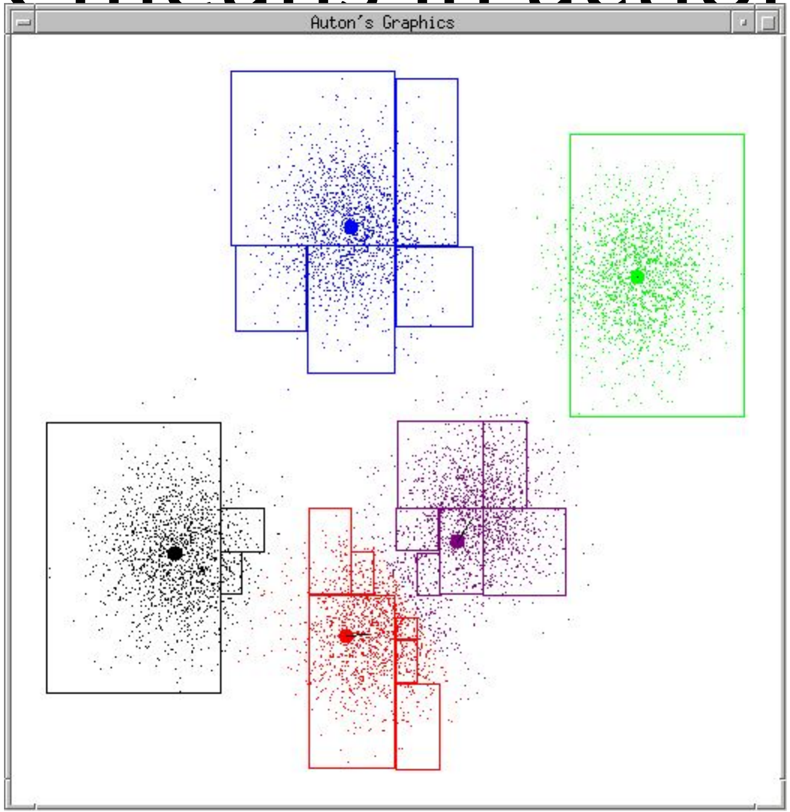
K-means in action



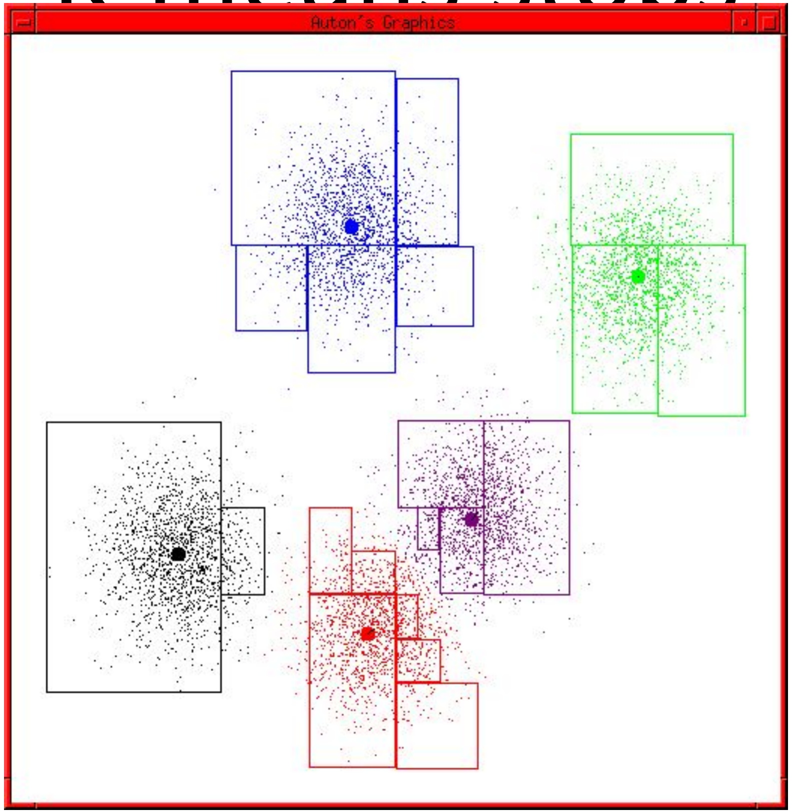
K-means in action



K-means in action



K-means stops



K-means algorithm

- Input: $x_1 \ldots x_n$, k
- **Step 1**: select k cluster centers $c_1 \ldots c_k$
- **Step 2**: for each point x, determine its cluster assignment: find the closest center in Euclidean distance

$$y(x) = argmin_{i=1:k} ||x - c_i||$$

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

$$c_i = \frac{\sum_{x:y(x)=i} x}{\sum_{x:y(x)=i} 1}$$

- Repeat step 2, 3 until cluster centers no longer change

Questions on k-means

- What is k-means trying to optimize?
- Will k-means stop (converge)?
- Will it find a global or local optimum?
- How to pick starting cluster centers?
- How many clusters should we use?

Distortion

- Suppose for a point x, you replace its coordinates by the cluster center $c_{y(x)}$ it belongs to (lossy compression)
- How far are you off? Measure it with squared Euclidean distance: $||x - c_{y(x)}||^2$

- This is the distortion of a single point x. For the whole dataset, the distortion is $\sum_{i=1}^{n} ||x_i - c_{y(x_i)}||^2$

The optimization problem of k-means

$$\min_{c,y} \sum_{i=1}^{n} ||x_i - c_{y(x_i)}||^2$$

Step 1

- For fixed cluster centers, if all you can do is to assign x to some cluster, then assigning x to its closest cluster center y(x) minimizes distortion

$$\Sigma_{d=1\ldots D} [x(d) - c_{y(x)}(d)]^2$$

- Why? Try any other cluster z≠y(x)

$$\Sigma_{d=1\ldots D} [x(d) - c_z(d)]^2$$

Step 2

- If the assignment of x to clusters are fixed, and all you can do is to change the location of cluster centers
- Then this is an optimization problem!
- Variables? $c_1(1), \ldots, c_1(D), \ldots, c_k(1), \ldots, c_k(D)$

$$\min \sum_x \sum_{d=1\ldots D} [x(d) - c_{y(x)}(d)]^2$$

$$= \min \sum_{z=1..k} \sum_{y(x)=z} \sum_{d=1\ldots D} [x(d) - c_z(d)]^2$$

- Unconstrained.

$$\partial/\partial c_z(d) \sum_{z=1..k} \sum_{y(x)=z} \sum_{d=1\ldots D} [x(d) - c_z(d)]^2 = 0$$

Step 2

- The solution is

$$c_z(d) = \sum_{y(x)=z} x(d) \Big/ |n_z|$$

- The d-th dimension of cluster z is the average of the d-th dimension of points assigned to cluster z
- Or, update cluster z to be the centroid of its points. This is exact what we did in step 2.

Repeat (step1, step2)

- Both step1 and step2 minimizes the distortion
  $$\Sigma_x \Sigma_{d=1...D} [x(d) - c_{y(x)}(d)]^2$$
- Step1 changes x assignments y(x)
- Step2 changes c(d) the cluster centers
- However there is no guarantee the distortion is minimized over all… need to repeat
- This is hill climbing (coordinate descent)
- Will it stop?

Repeat (step1, step2)

- Will it stop?

There are finite number of points

Finite ways of assigning points to clusters

In step1, an assignment that reduces distortion has to be a new assignment not used before

Step1 will terminate

So will step 2

So k-means terminates

What optimum does K-means find

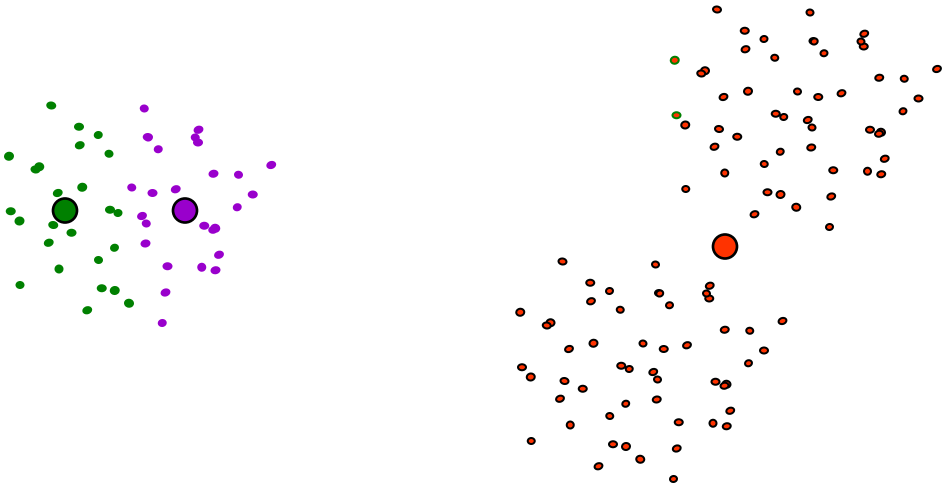
- Will k-means find the global minimum in distortion? Sadly no guarantee…
- Can you think of one example?

What optimum does K-means find

- Will k-means find the global minimum in distortion? Sadly no guarantee…
- Can you think of one example? (Hint: try k=3)

What optimum does K-means find

- Will k-means find the global minimum in distortion? Sadly no guarantee…
- Can you think of one example? (Hint: try k=3)



Picking starting cluster centers

- Which local optimum k-means goes to is determined solely by the starting cluster centers
  - Be careful how to pick the starting cluster centers. Many ideas. Here's one neat trick:
    1. Pick a random point x1 from dataset
    2. Find the point x2 farthest from x1 in the dataset
    3. Find x3 farthest from the closer of x1, x2
    4. … pick k points like this, use them as starting cluster centers for the k clusters
  - Run k-means multiple times with different starting cluster centers (hill climbing with random restarts)

Picking the number of clusters

- Difficult problem
- Domain knowledge?
- Otherwise, shall we find k which minimizes distortion?

Picking the number of clusters

- Difficult problem
- Domain knowledge?
- Otherwise, shall we find k which minimizes distortion? k = N, distortion = 0
- Need to regularize. A common approach is to minimize the Schwarz criterion

$$\text{distortion} + \lambda\ (\#\text{param}) \log N$$
$$= \text{distortion} + \lambda\ D\ k \log N$$

D: #dimensions

k: #clusters

N: #points

Break & Quiz

Q 1.1: 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 at the next iteration are?

- 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 1.1: 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 at the next iteration are?

- **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 1.2: 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 1.2: 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 1.3: 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

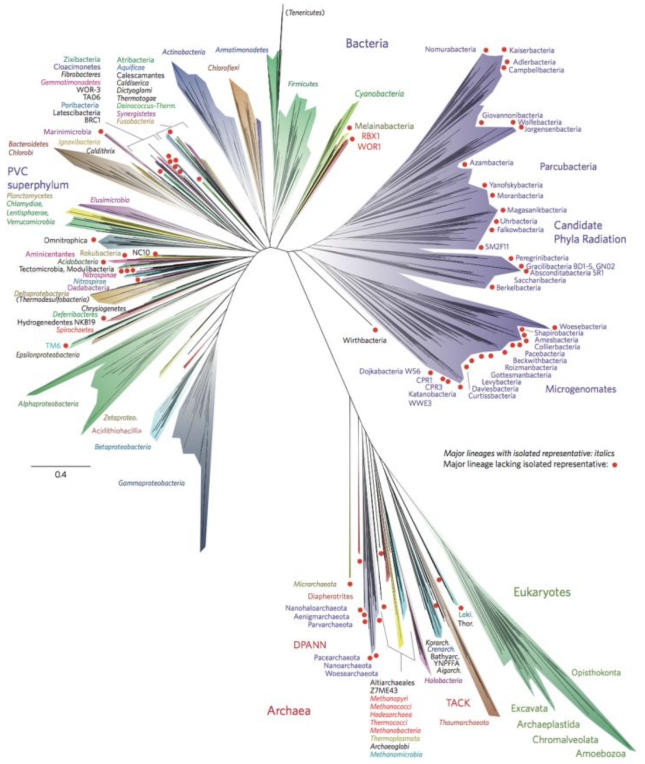
Q 1.3: If we run K-means clustering twice with random starting cluster centers, are we guaranteed to get same clustering results? Does K-means always converge?

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

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



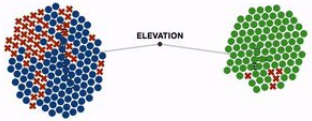
Credit: Wikipedia

Agglomerative vs Divisive

Two ways to go:

- **Agglomerative**: bottom up.
  - Start: each point a cluster. Progressively merge clusters

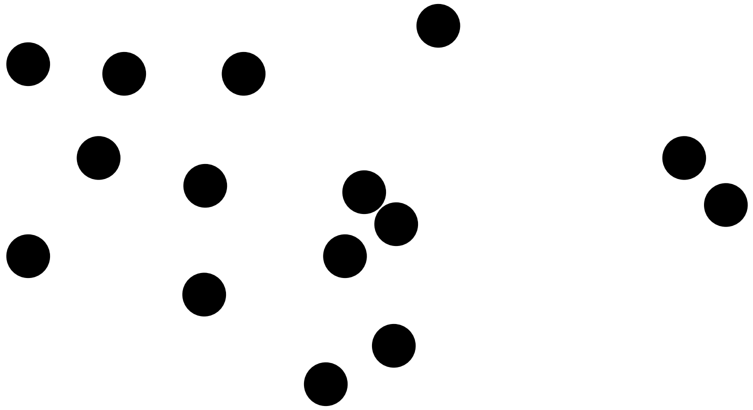
- **Divisive**: top down
  - Start: all points in one cluster. Progressively split clusters



Credit: r2d3.us

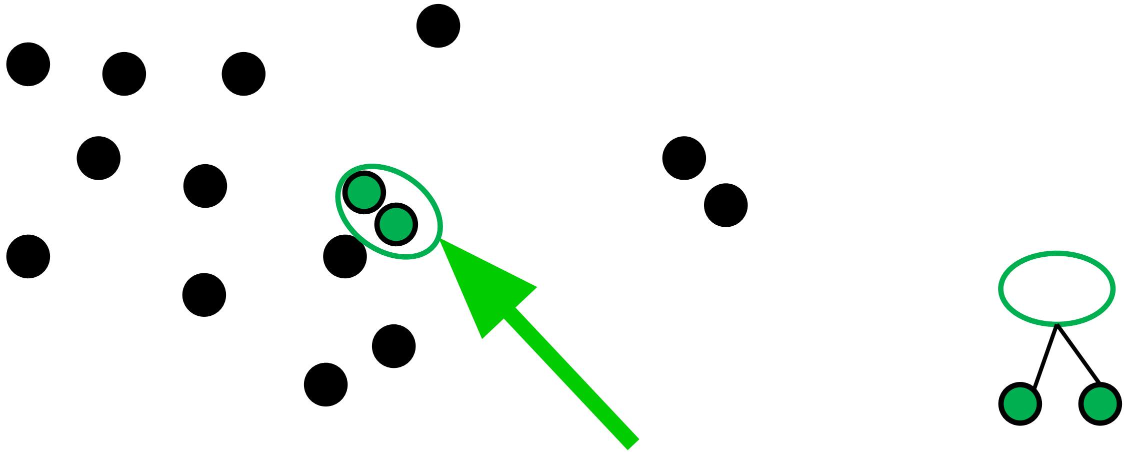
Agglomerative Clustering Example

Agglomerative. Start: every point is its own cluster



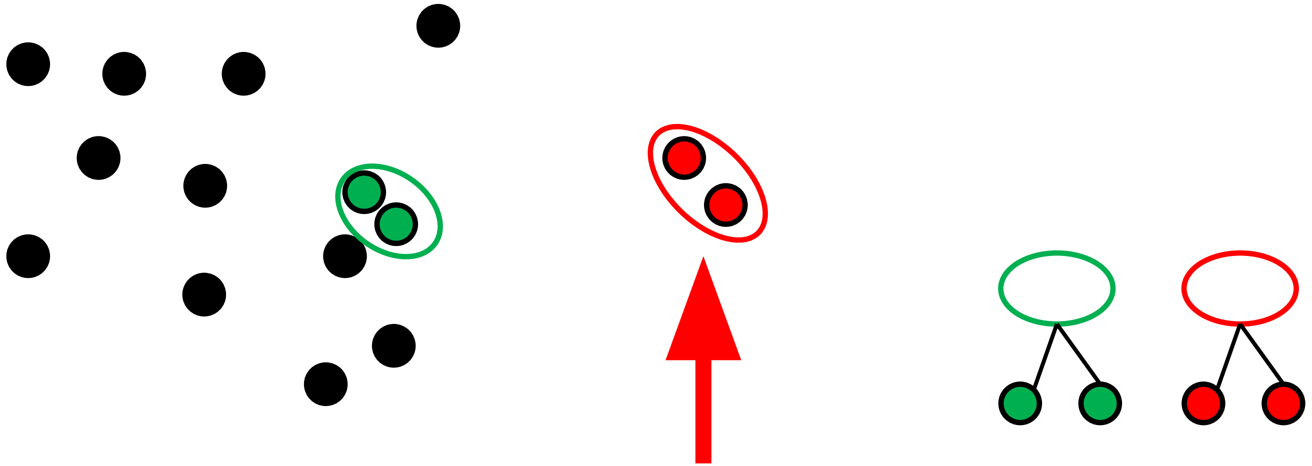
Agglomerative Clustering Example

Get pair of clusters that are closest and merge



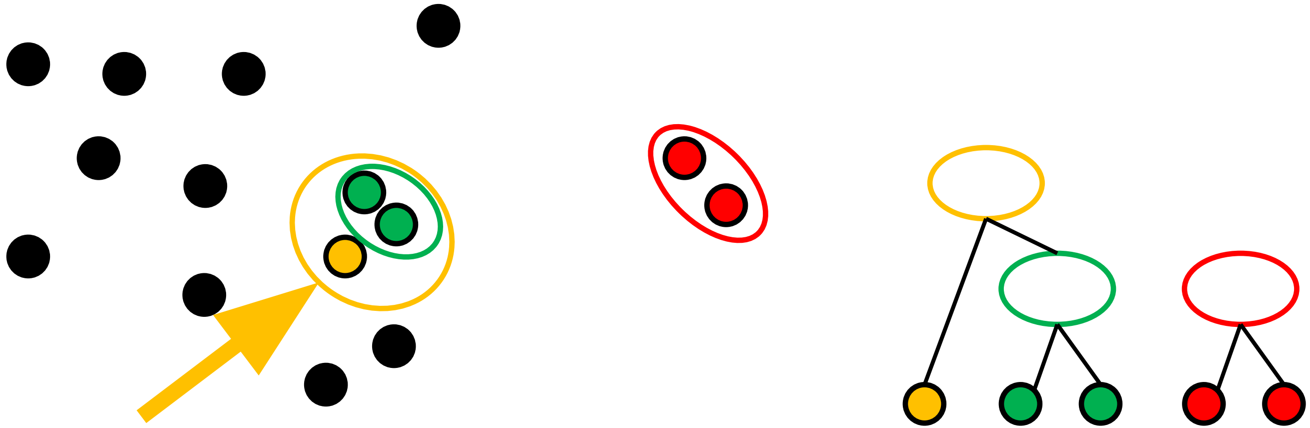
Agglomerative Clustering Example

Repeat: Get pair of clusters that are closest and merge



Agglomerative Clustering Example

Repeat: Get pair of clusters that are closest and merge



Merging Criteria

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

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

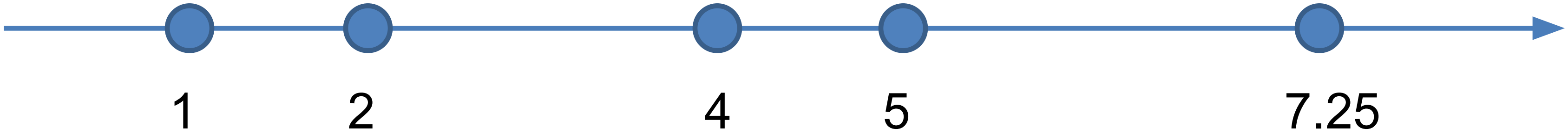
- Average-linkage

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

Single-linkage Example

We’ll merge using single-linkage

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

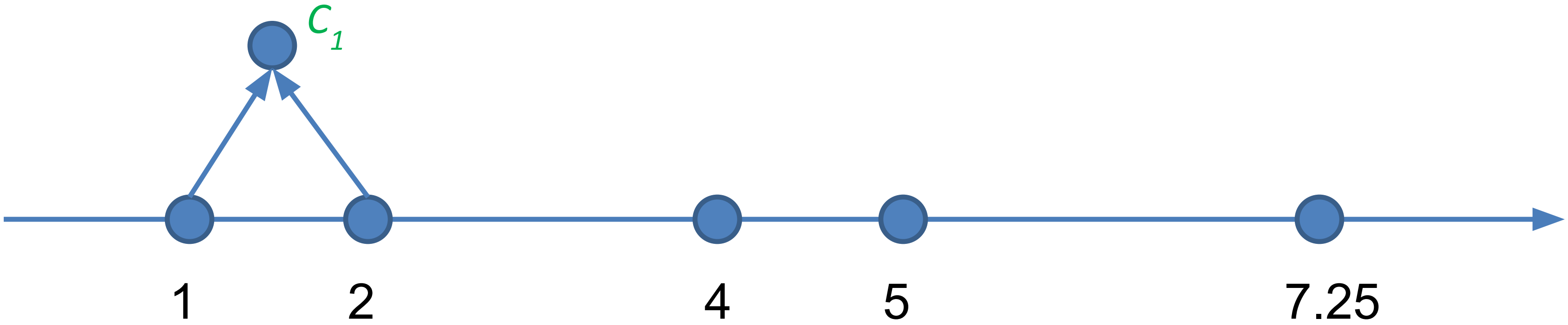


Single-linkage Example

We'll merge using single-linkage

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

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

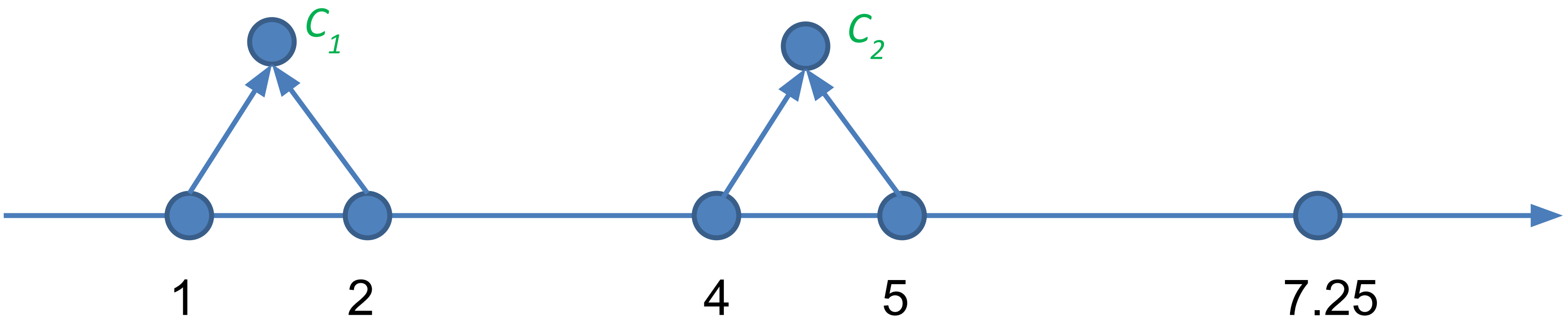


Single-linkage Example

Continue…

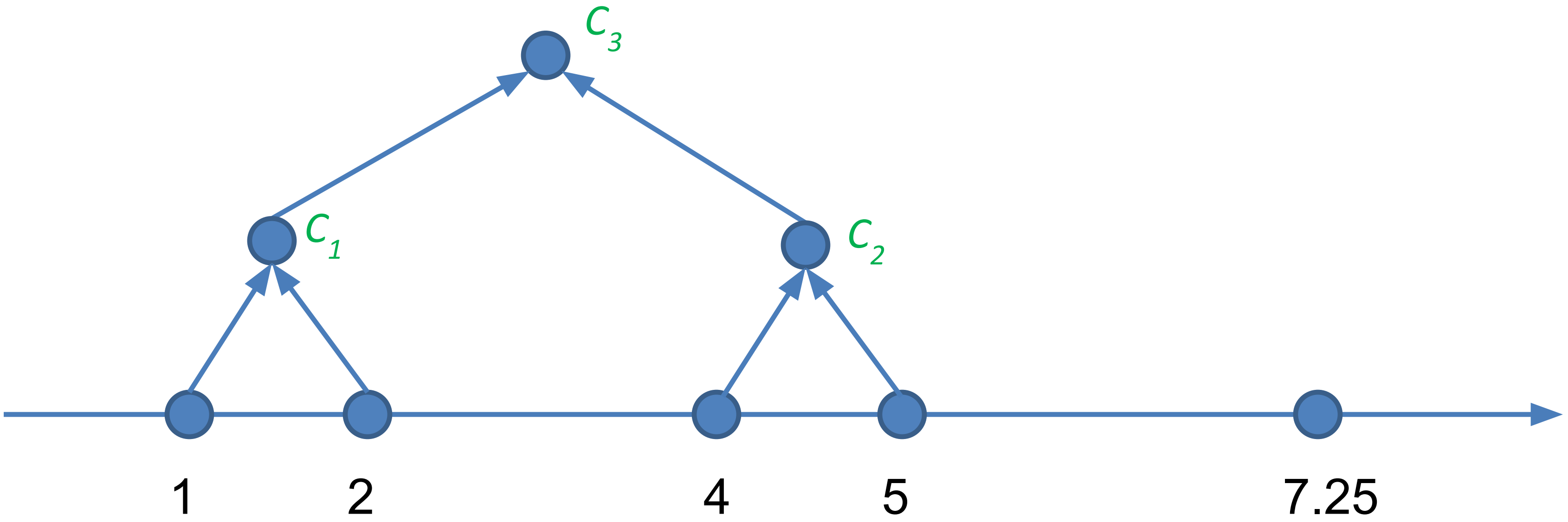
$$d(C_1, C_2) = d(2, 4) = 2$$

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

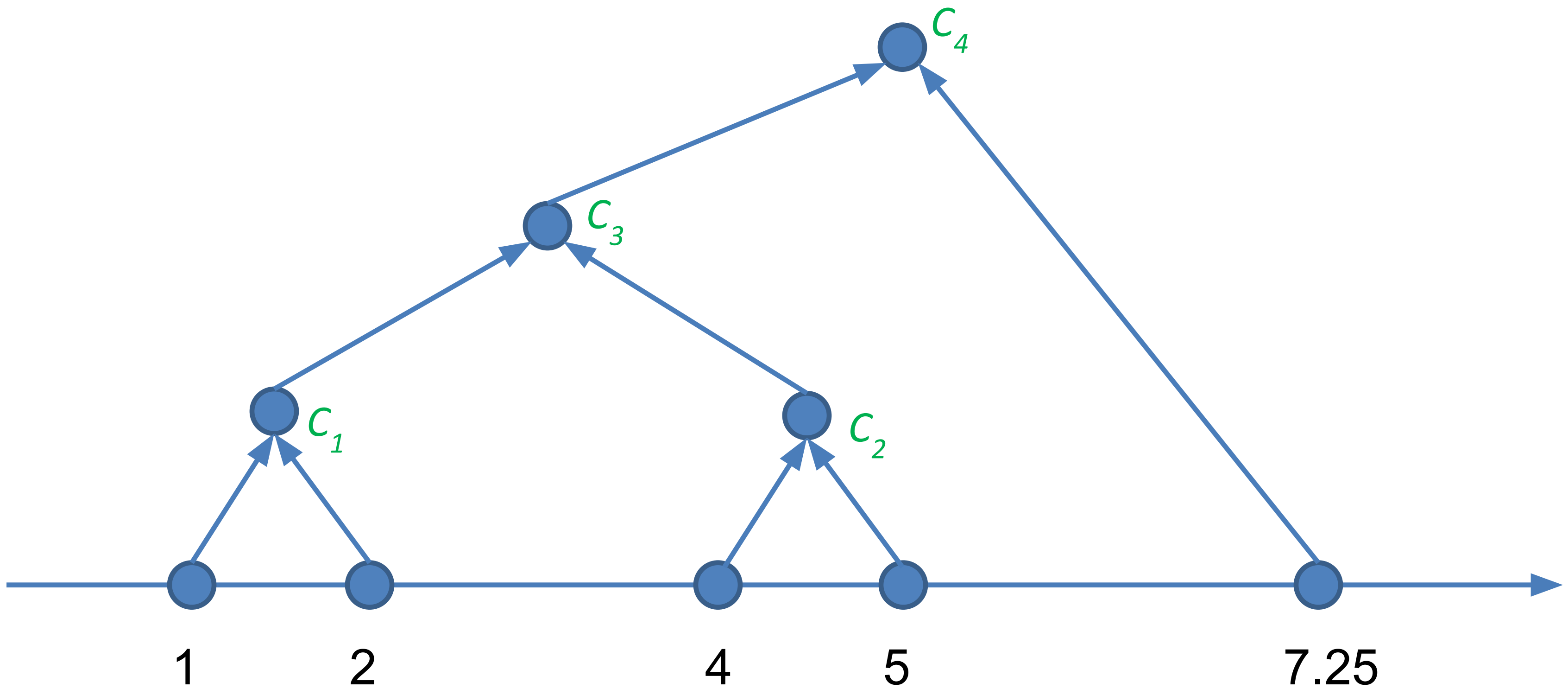


Single-linkage Example

Continue…



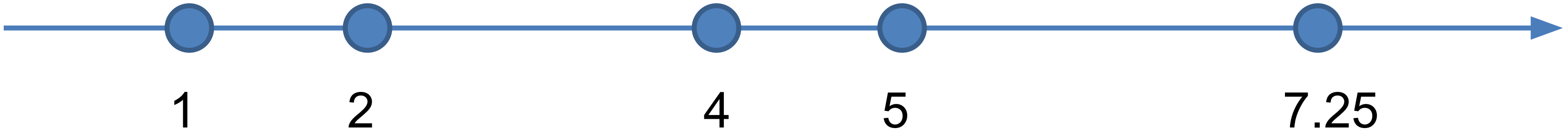
Single-linkage Example



Complete-linkage Example

We'll merge using complete-linkage

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

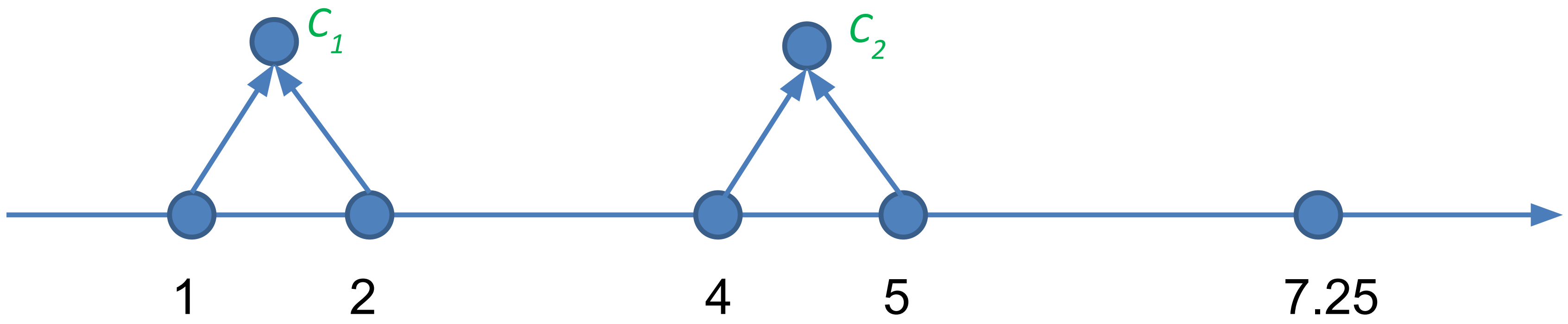


Complete-linkage Example

Beginning is the same…

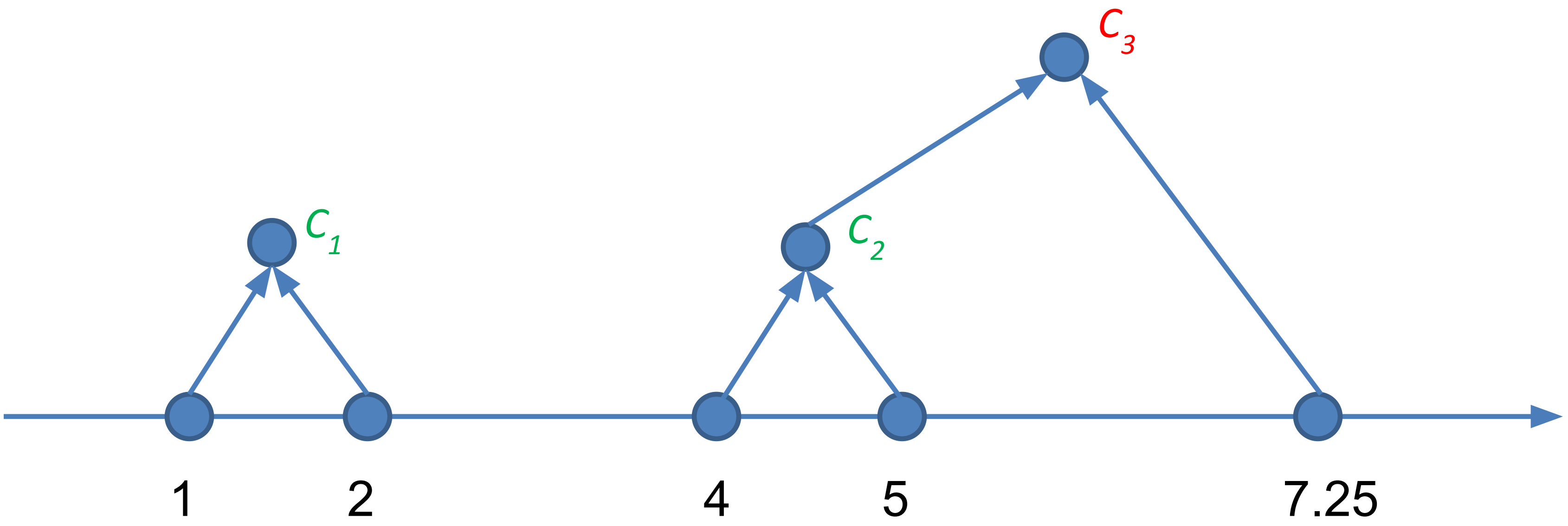
$$d(C_1, C_2) = d(1, 5) = 4$$

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

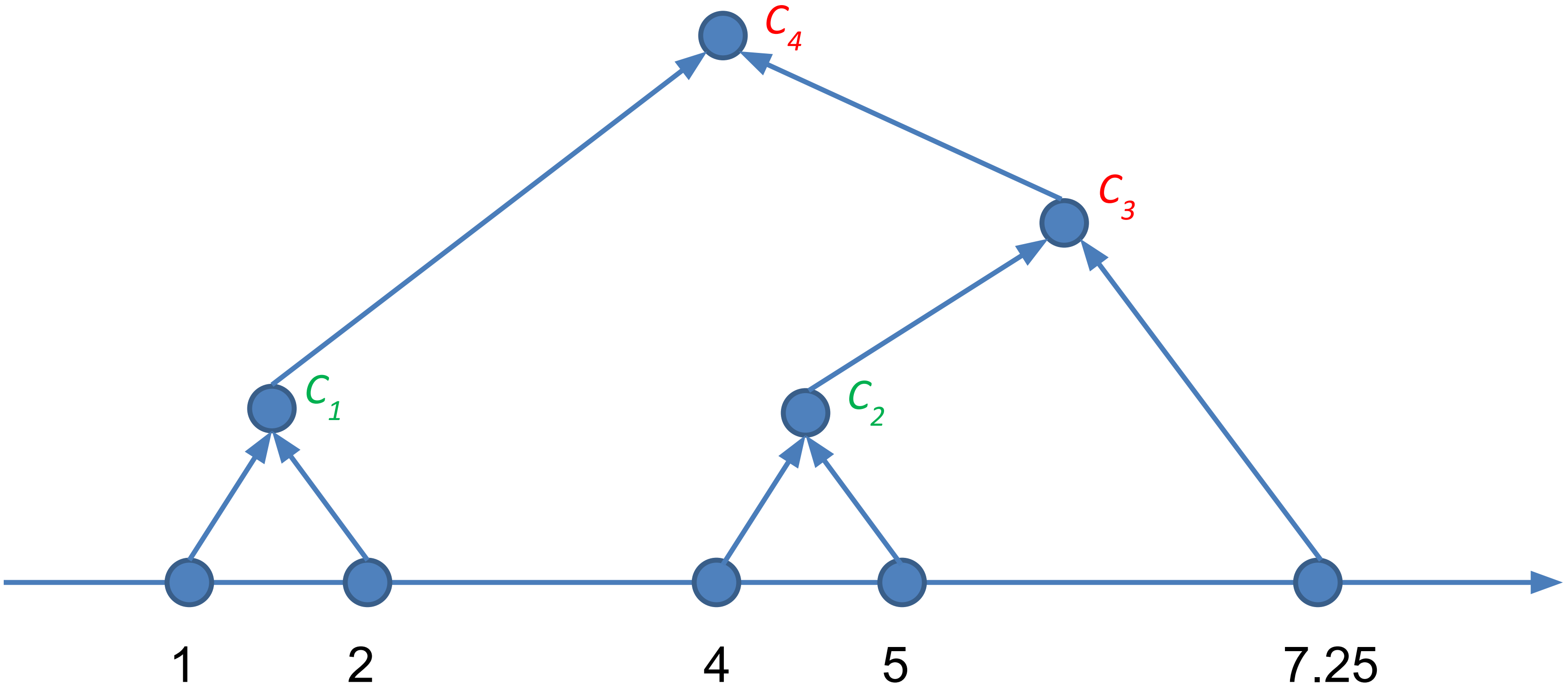


Complete-linkage Example

Now we diverge:



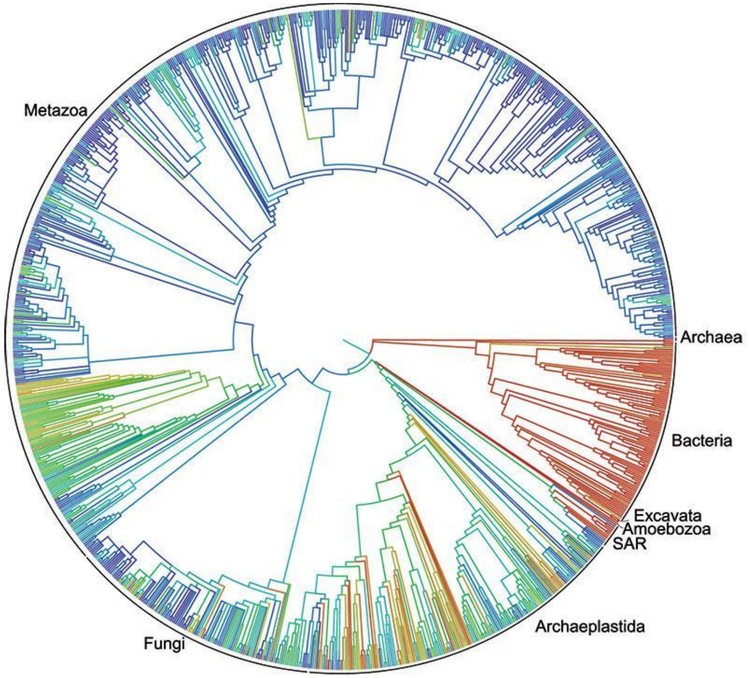
Complete-linkage Example



When to Stop?

No simple answer:

- Use the binary tree (a **dendogram**)
- Cut at different levels (g different heights/depth

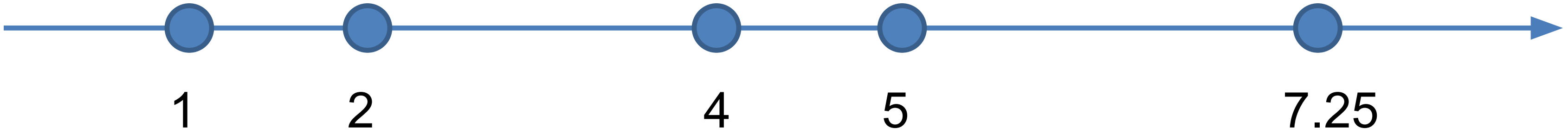


http://opentreeoflife.org/

Break & Quiz

Q 2.1: Let's do hierarchical clustering for two clusters with average linkage on the dataset below. What are the clusters?

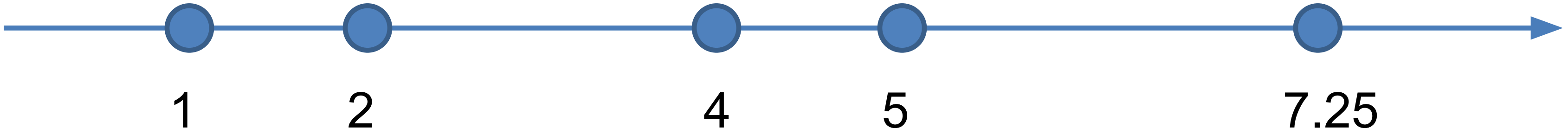
- A. {1}, {2,4,5,7.25}
- B. {1,2}, {4, 5, 7.25}
- C. {1,2,4}, {5, 7.25}
- D. {1,2,4,5}, {7.25}



Break & Quiz

Q 2.1: 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 2.2: 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 2.2: 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**