# K-means Clustering

**Yingyu Liang**

yliang@cs.wisc.edu

**Computer Sciences Department**

**University of Wisconsin, Madison**

[Partially Based on slides from Jerry Zhu and Mark Craven]

# Review: clustering

given

- training set of instances $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(m)}$

output

- model $h$ that divides the training set into clusters such that there is intra-cluster similarity and inter-cluster dissimilarity

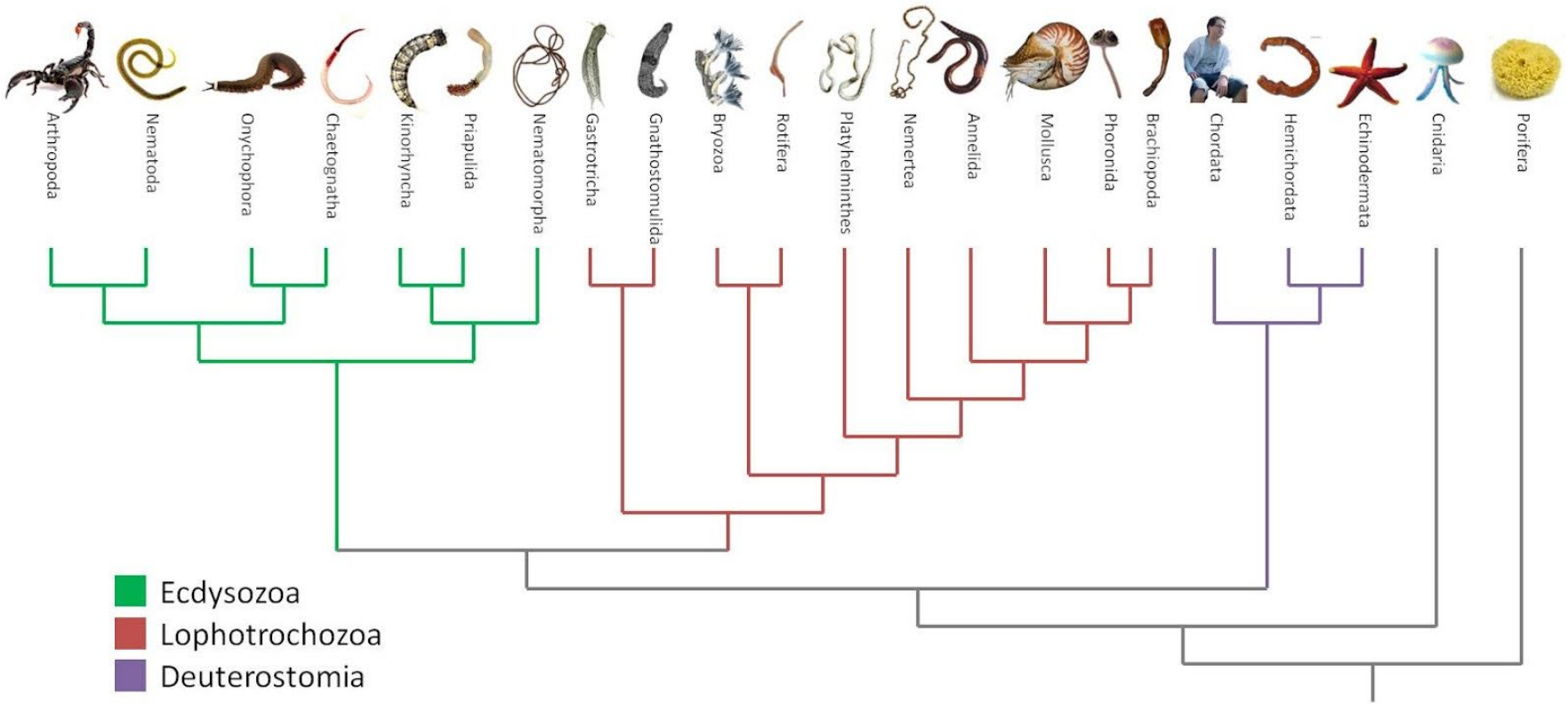
# Review: clustering

- Many clustering algorithms. We'll look at the two most frequently used ones:
  - Hierarchical clustering

    Where we build a binary tree over the dataset
  - K-means clustering

    Where we specify the desired number of clusters, and use an iterative algorithm to find them

# Hierarchical clustering



# K-means clustering

- Can get $k$ clusters
- and one center/ prototype for each cluster







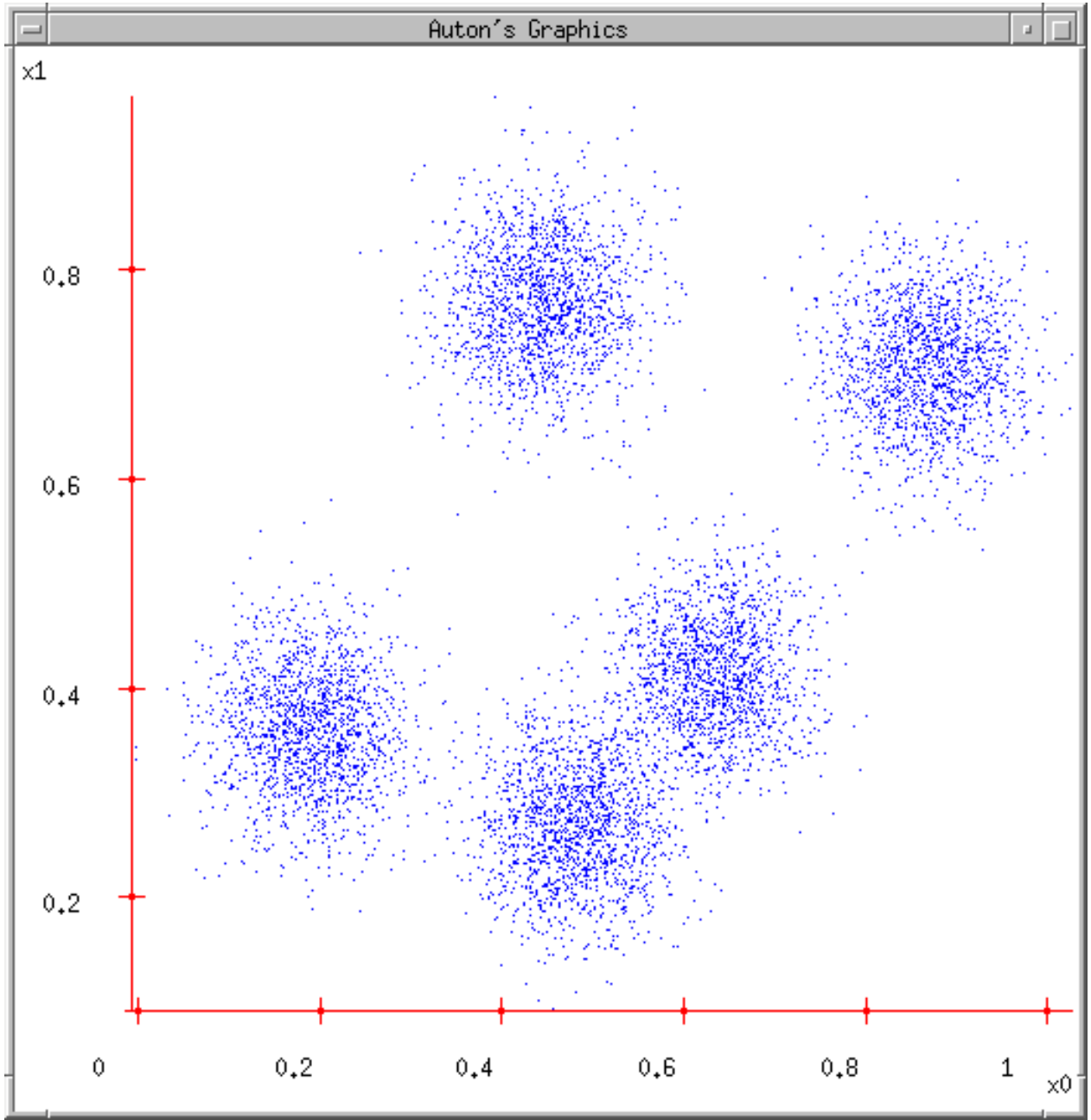

# K-means clustering

- Very popular clustering method

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

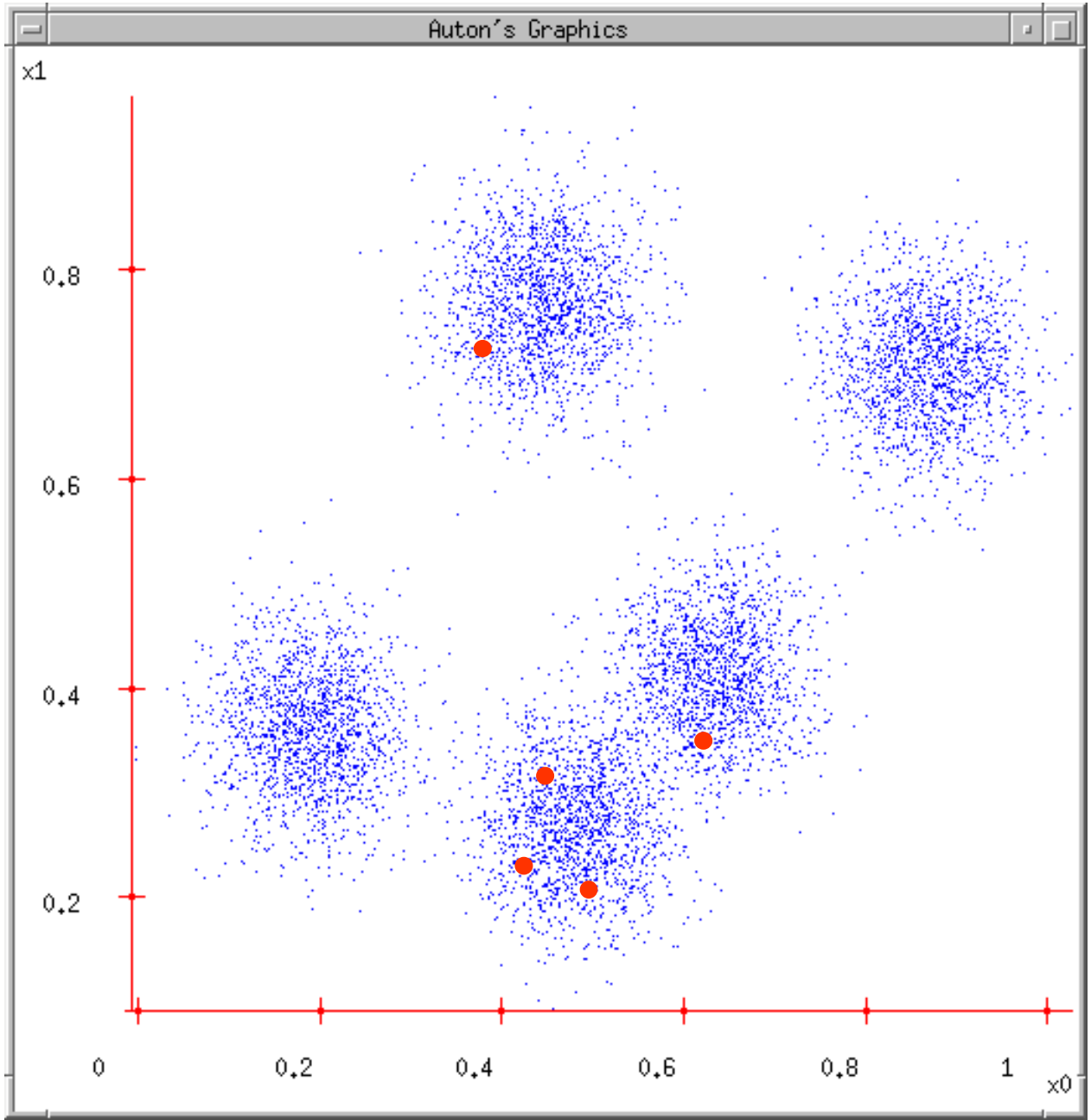
# K-means clustering

- Input: dataset, $k = 5$



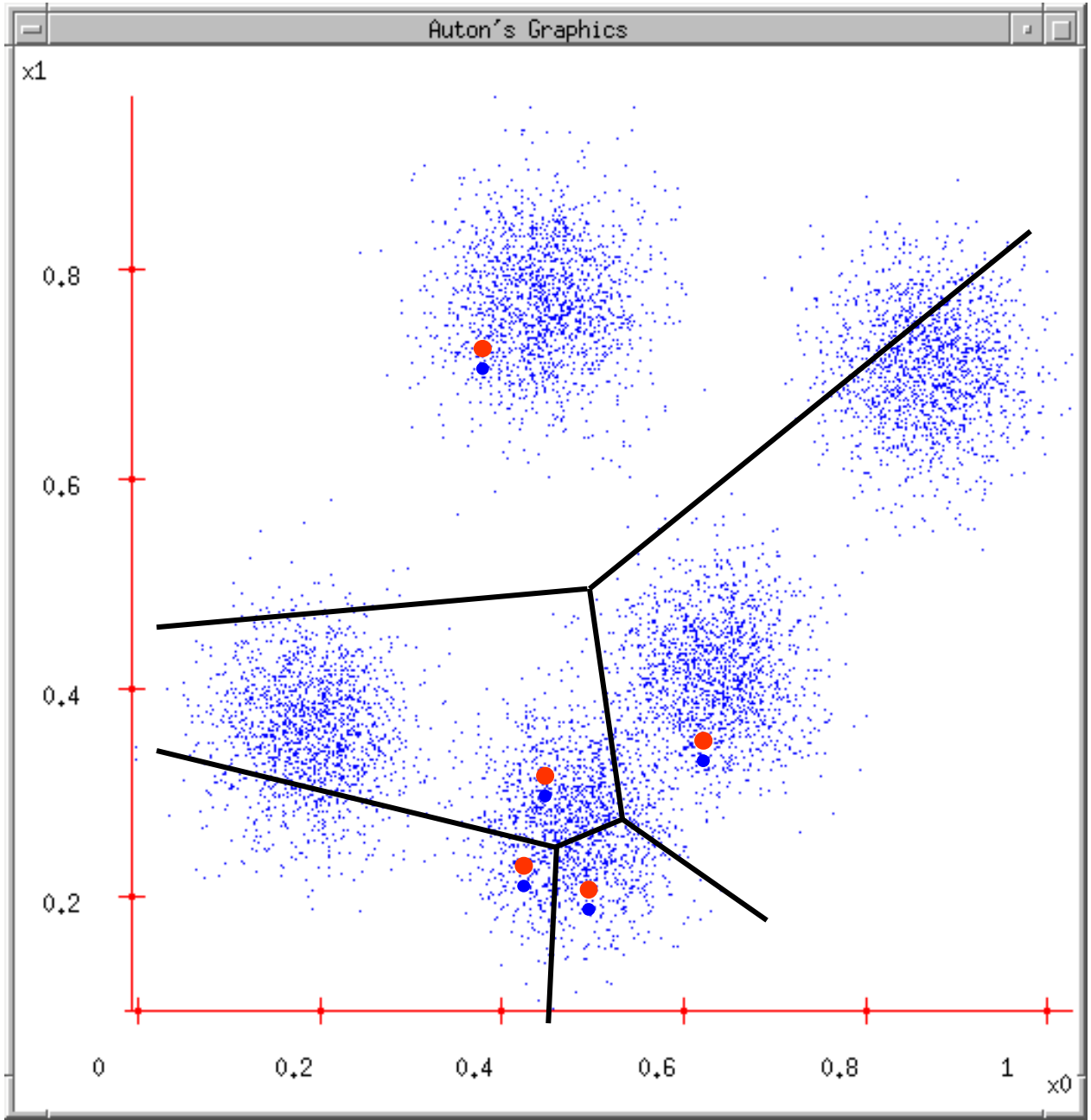
# K-means clustering

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



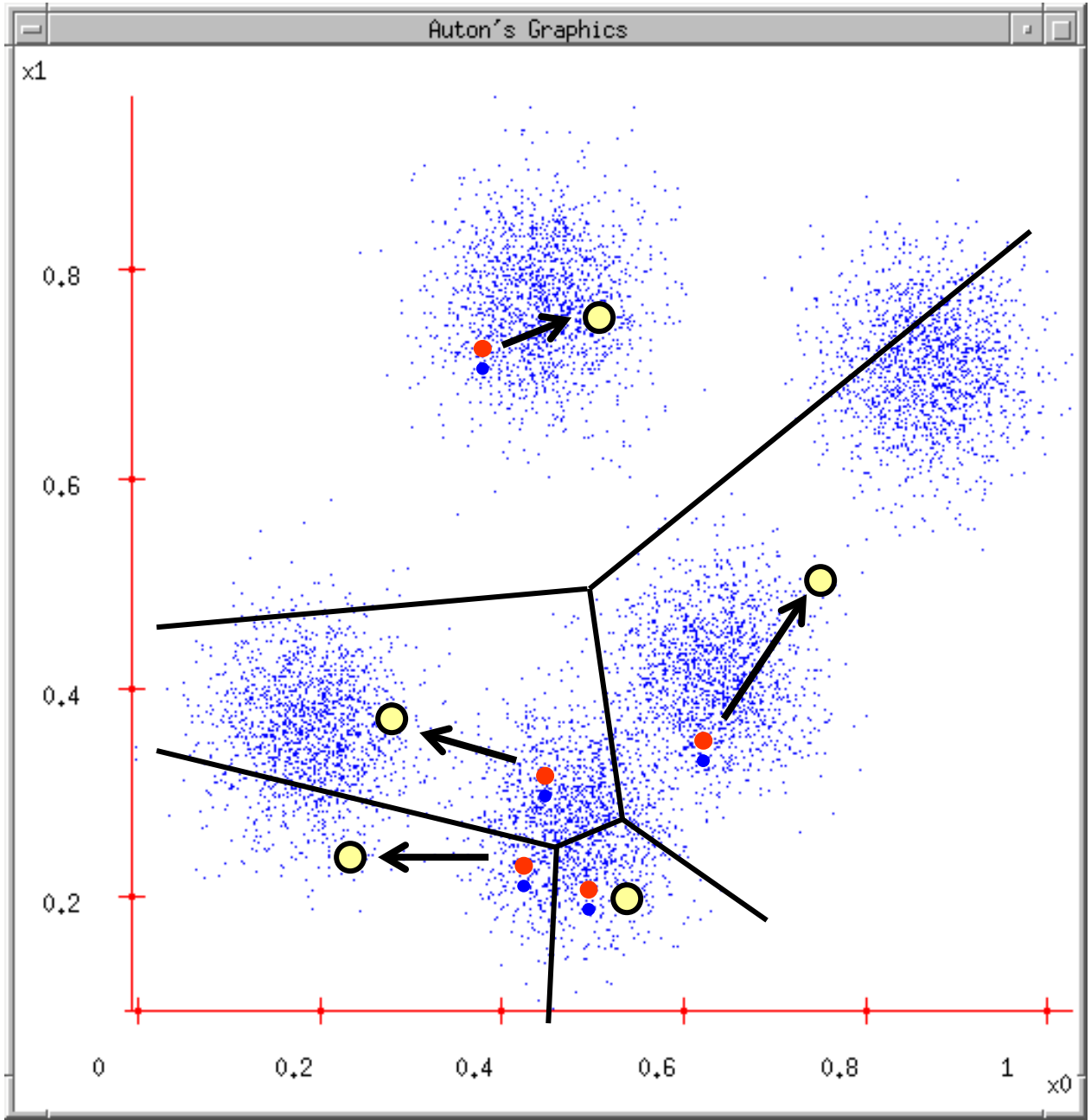
# K-means clustering

- Each point finds which cluster center it is closest to. The point is assigned to that cluster.



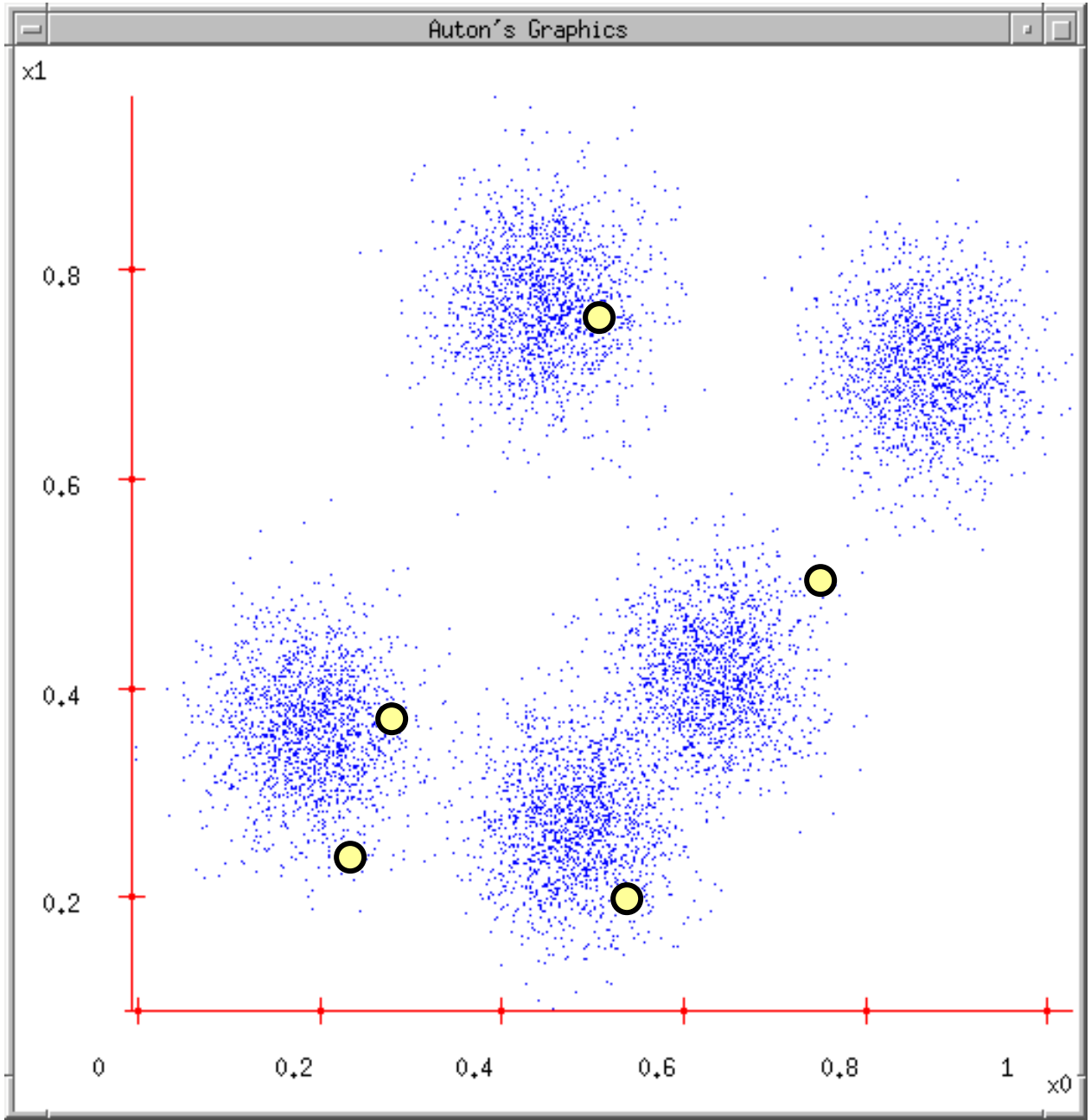
# K-means clustering

- Each cluster computes its new center, based on which points belong to it. The new center is the centroid (the average of points in the cluster).



# 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 algorithm

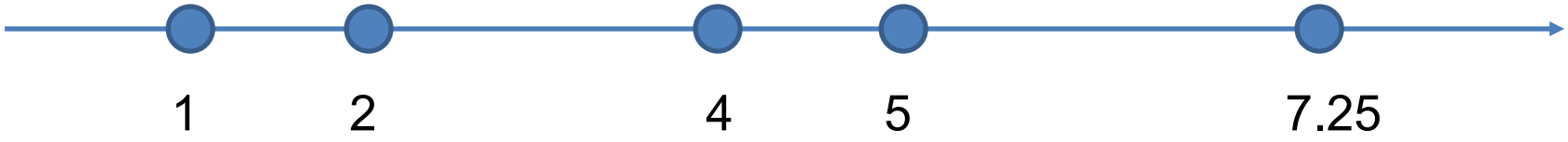
- Input: points $x_1, \ldots, x_n$, number of clusters $k$
- Select $k$ centers $c_1, \ldots, c_k$
- **Step 1**: for each point $x$, determine its cluster: find the closest center in Euclidean distance
- **Step 2**: update all cluster centers as the centroids

$$c_i = \sum_{x \text{ in cluster } i} x \,/\text{SizeOf}(\text{cluster } i)$$

- Repeat step 1, 2 until the centers don't/slightly change

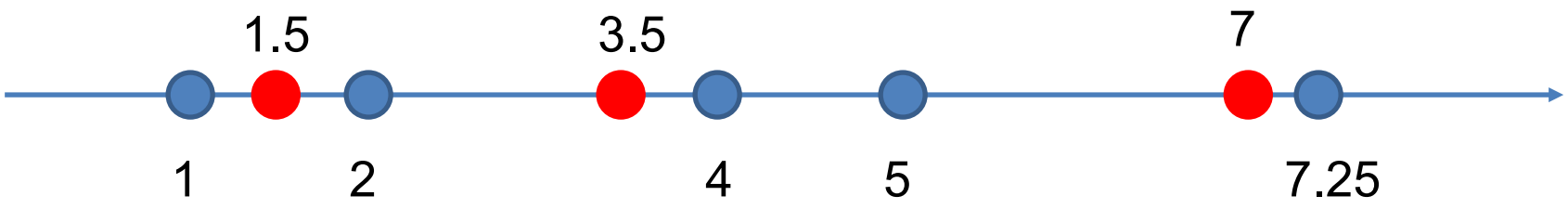
# Example 1

Data set, $k = 3$



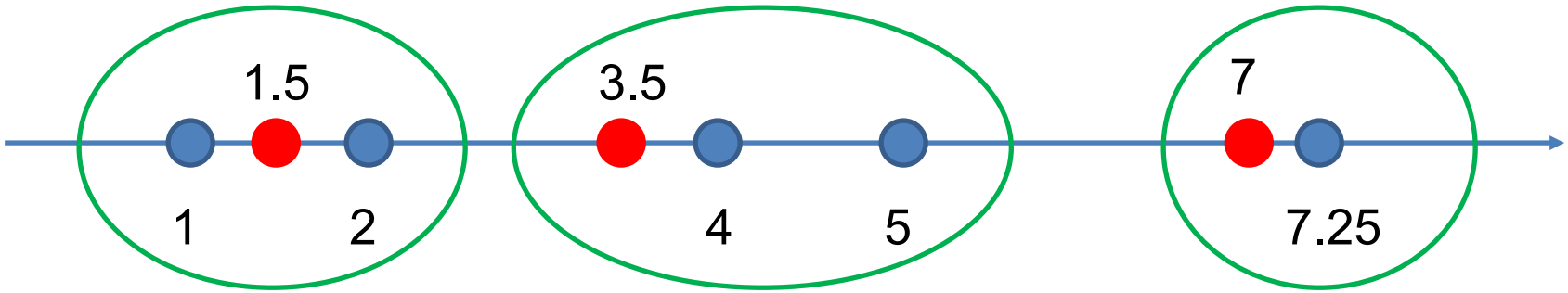
# Example 1

Initialize centers



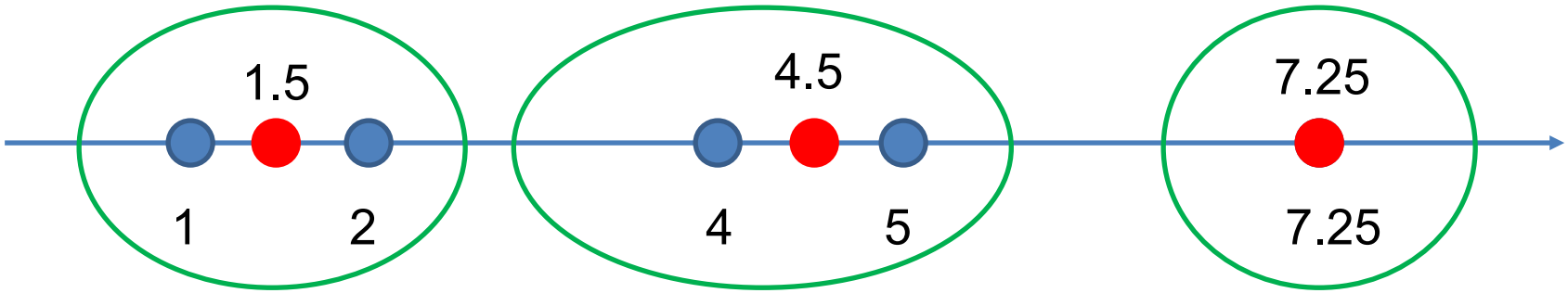
# Example 1

Assign the points to centers



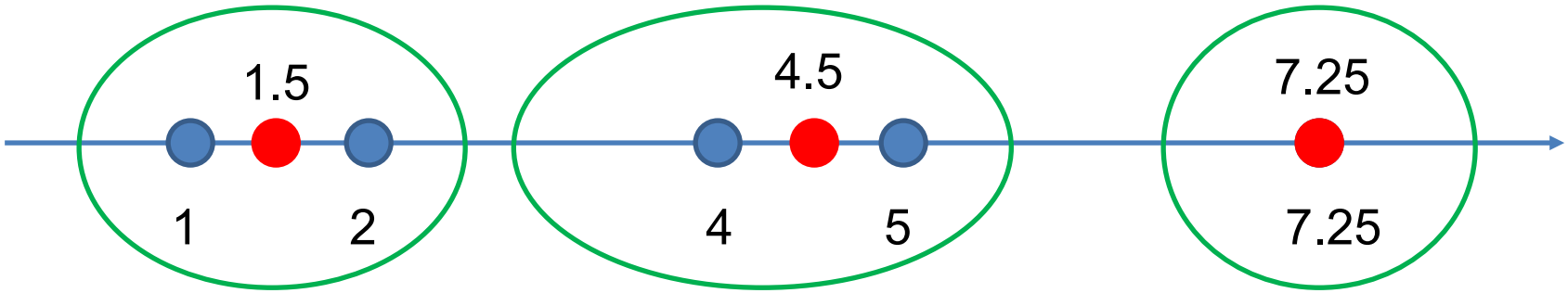
# Example 1

Update the centers



# Example 1

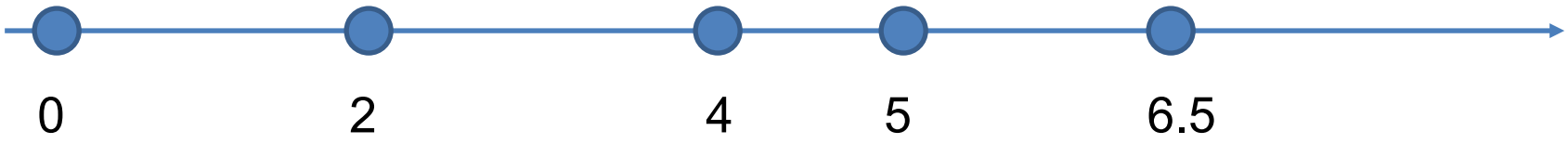
Assign the points to centers




At this time point, the clusters don't change, and the centers don't change. Stop.

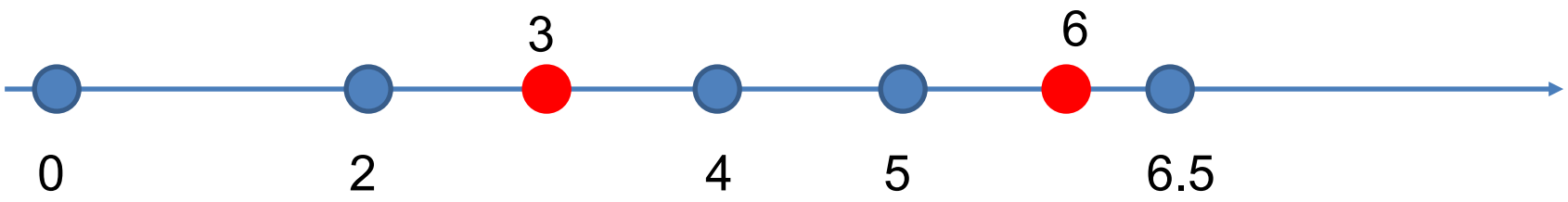
# Example 2

Data set, $k = 2$



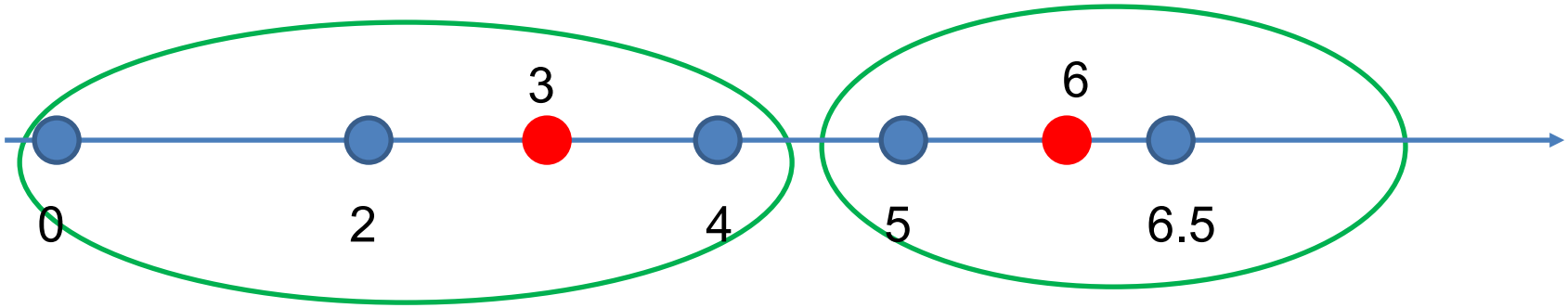
# Example 2

Initialize centers



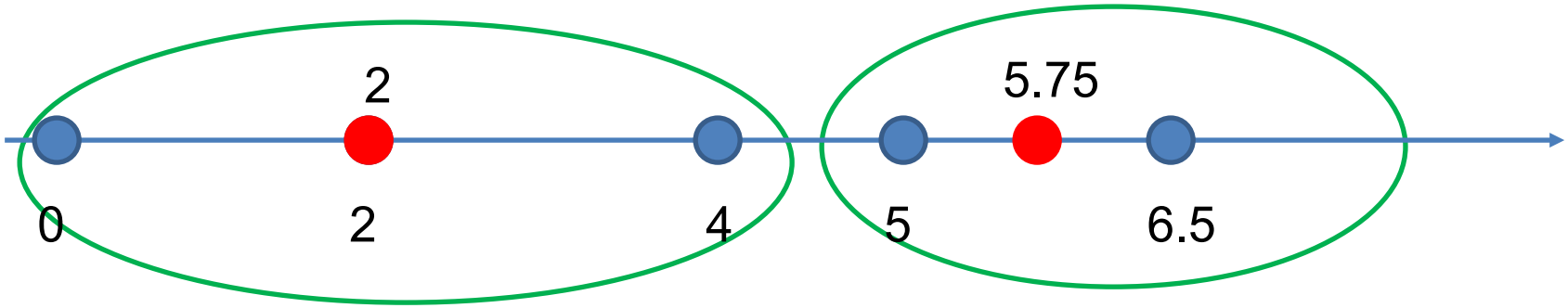
# Example 2

Assign the points to centers



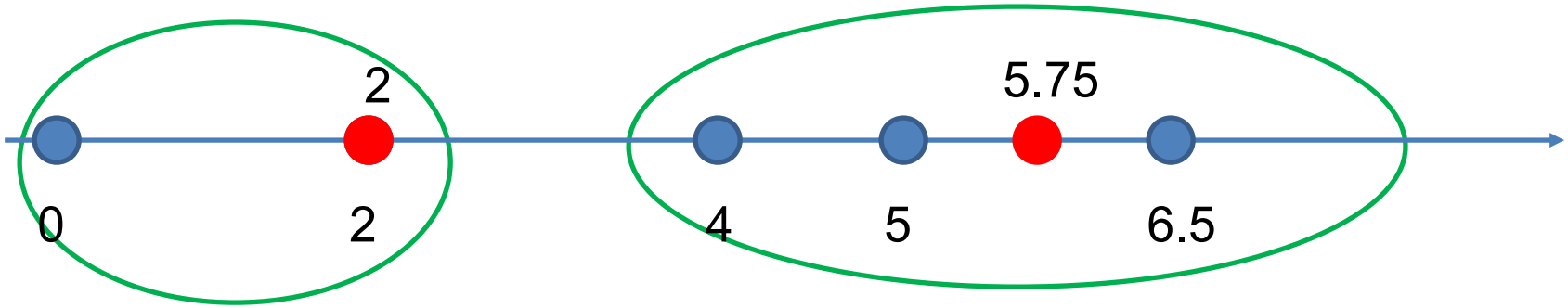
# Example 2

Update the centers



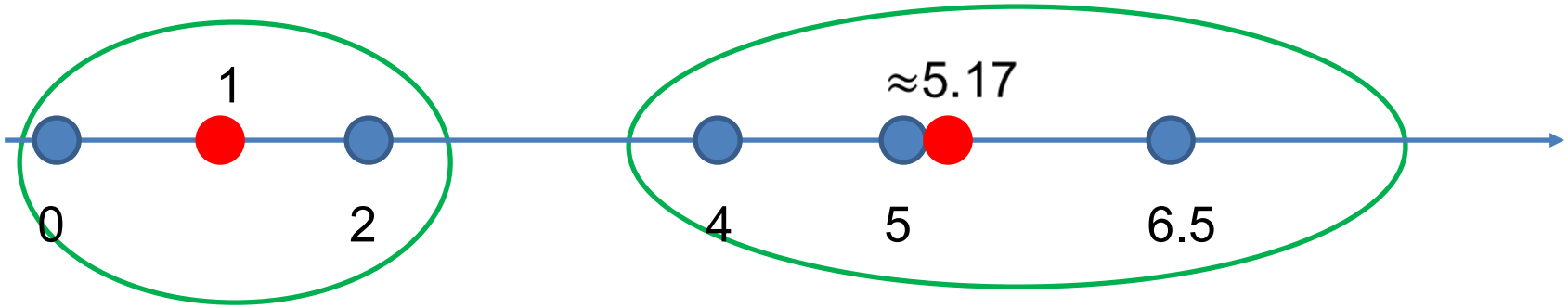
# Example 2

Assign the points to centers



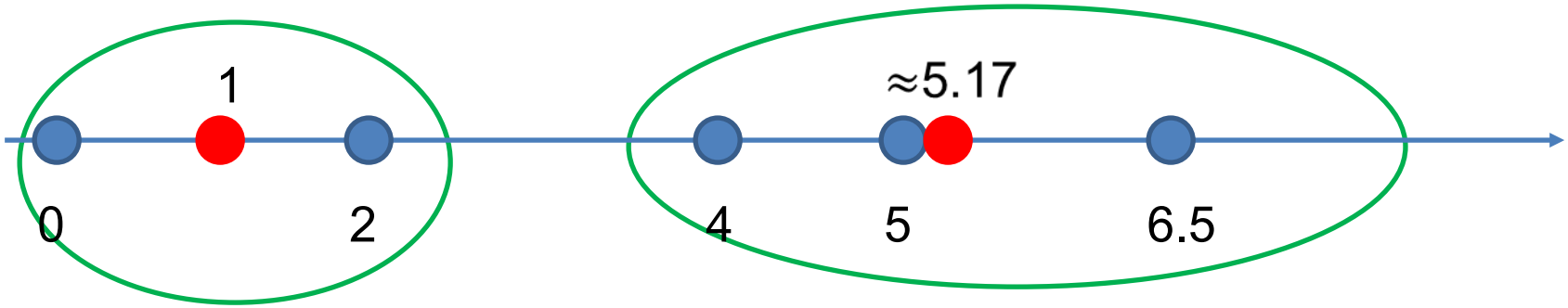
# Example 2

Update the centers



# Example 2

Assign the points to centers




At this time point, the clusters don't change, and the centers don't change. Stop.

# 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?

# K-MEANS OBJECTIVE

# Distortion

- Clustering as summarization: replace a point $x$ with its center $c_{y(x)}$. How far are you off?
- The distortion of $x$ is measured by squared Euclidean distance:

$$\left\|x - c_{y(x)}\right\|^2 = \sum_{i=1}^{d} \left[x_i - \left(c_{y(x)}\right)_i\right]^2$$

- The distortion of the whole dataset is

$$\sum_{x} \left\|x - c_{y(x)}\right\|^2$$

# The optimization objective

- Minimize the distortion of the dataset

$$\min_{\substack{y(x_1),\ldots,y(x_n) \\ c_1,\ldots,c_k}} \sum_x \left\| x - c_{y(x)} \right\|^2$$

# Step 1

- Suppose we fix the cluster centers
- Assigning $x$ to its closest cluster center $y(x)$ minimizes the distortion

$$\left\| x - c_{y(x)} \right\|^2$$

# Step 2

- Suppose we fix the assignment of points. All you can do is to change the cluster centers
- This is a continuous optimization problem!

$$\min_{c_1,\ldots,c_k} \sum_{x} \left\| x - c_{y(x)} \right\|^2$$

# Step 2

- Suppose we fix the assignment of points. All you can do is to change the cluster centers
- This is a continuous optimization problem!

$$\min_{c_1,\ldots,c_k} \sum_x \left\| x - c_{y(x)} \right\|^2$$

- Set the gradient to 0 leads to

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

# Step 2

- Computing the best centers (assignments fixed)

$$\sum_{x} \|x - c_{y(x)}\|^2 = \sum_{i=1}^{k} \sum_{y(x)=i} \|x - c_i\|^2$$

- Set the gradient w.r.t. $c_i$ to 0:

$$\sum_{y(x)=i} 2(x - c_i) = 0$$

- So the optimal $c_i$ is average(points in the cluster)

# Repeat (step1, step2)

- Both step1 and step2 minimizes the distortion
- Step1 changes the assignments $y(x)$
- Step2 changes the cluster centers $c_z$

- However there is no guarantee the distortion is minimized over all… need to repeat
- This is hill climbing (coordinate descent)

# PROPERTIES OF K-MEANS CLUSTERING

# 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?

# Repeat (step1, step2)

- Both step1 and step2 minimizes the distortion
- Step1 changes the assignments $y(x)$
- Step2 changes the cluster centers $c_z$

- Will it stop?

# Repeat (step1, step2)

- Both step1 and
- Step1 changes
- Step2 changes

- 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

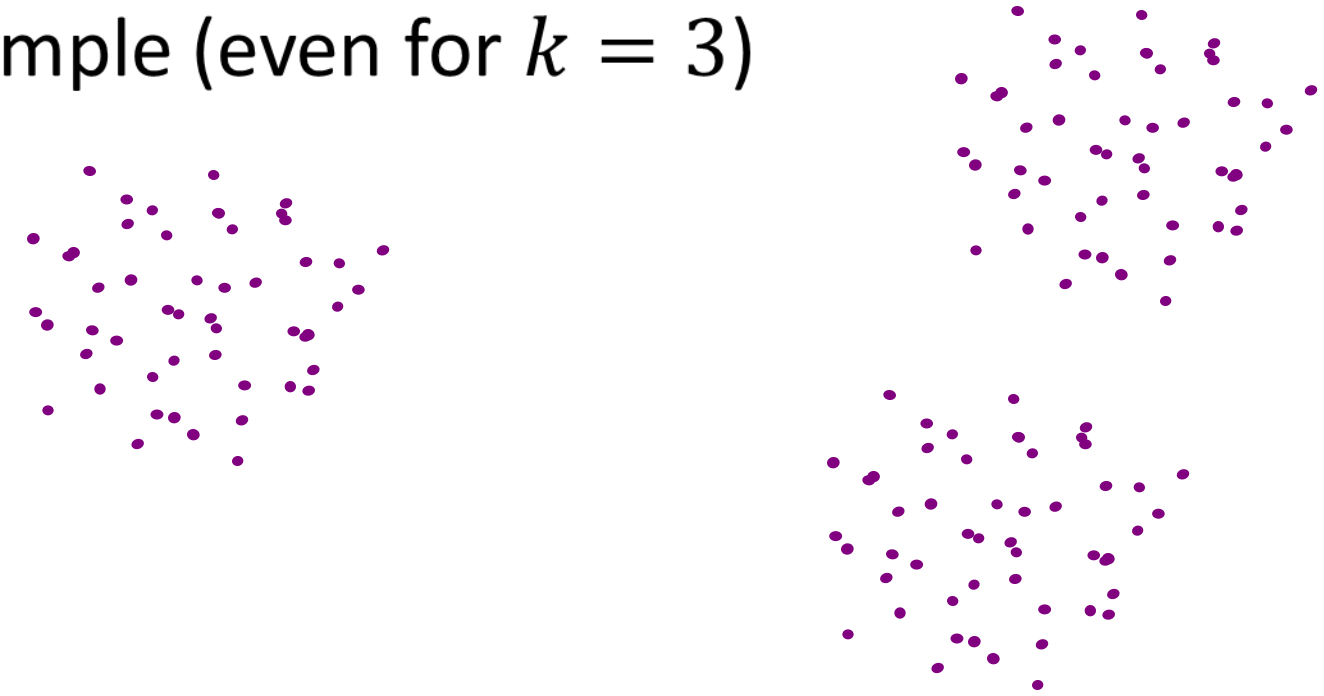
So k-means terminates

# Will find global optimum?

- Sadly no guarantee

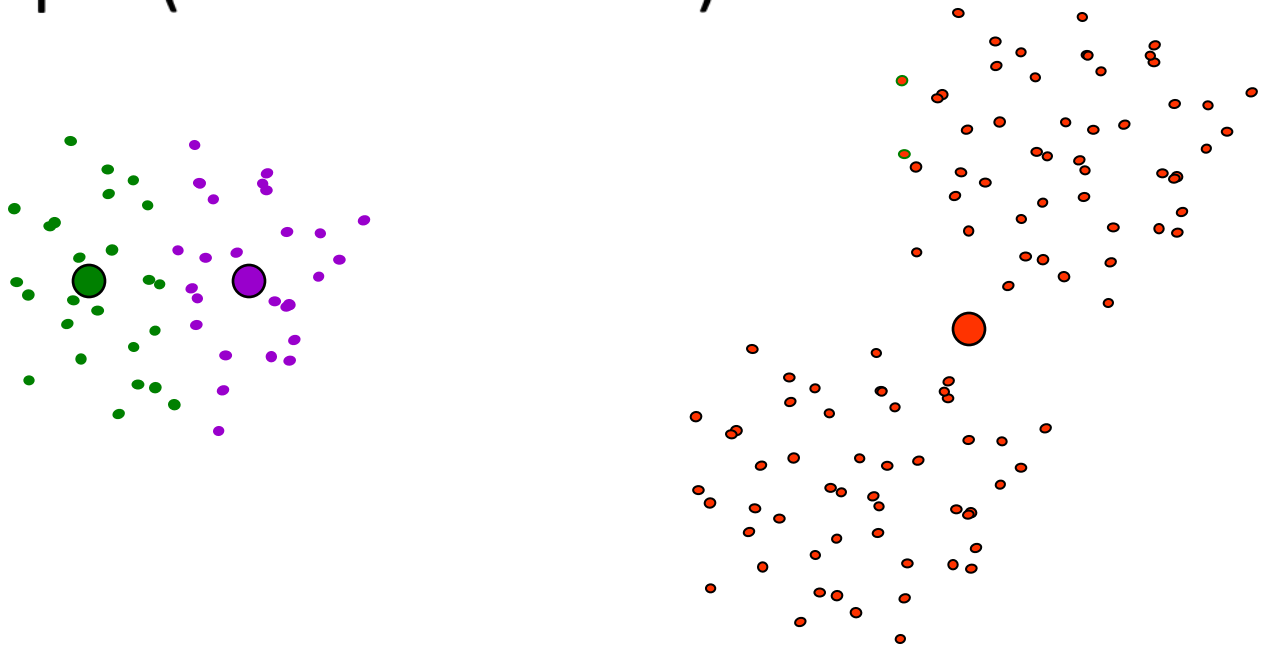
# Will find global optimum?

- Sadly no guarantee
- Example (even for $k = 3$)



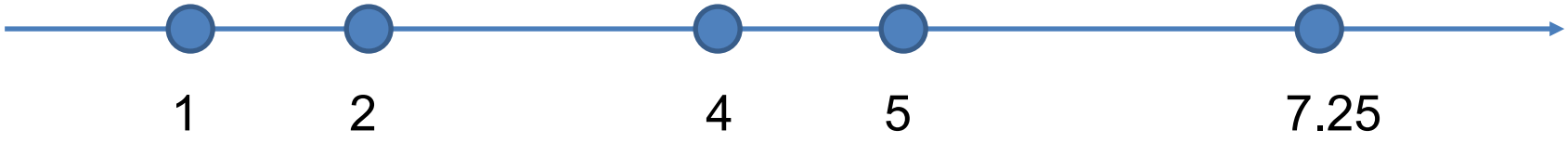
# Will find global optimum?

- Sadly no guarantee
- Example (even for $k = 3$)



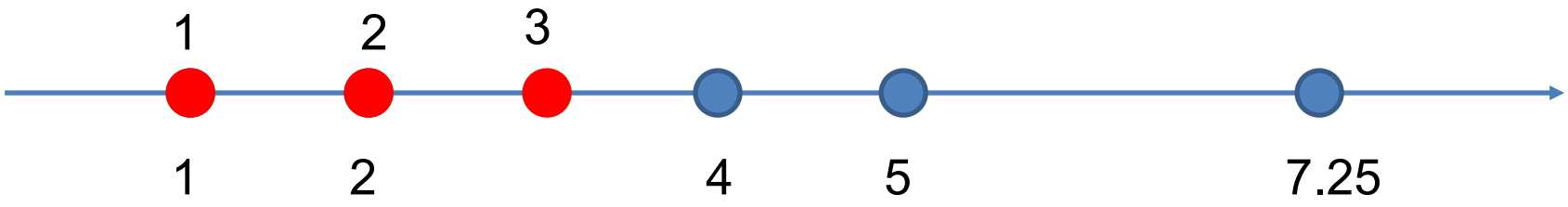
# Example 3

Same data set as Example 1, $k = 3$



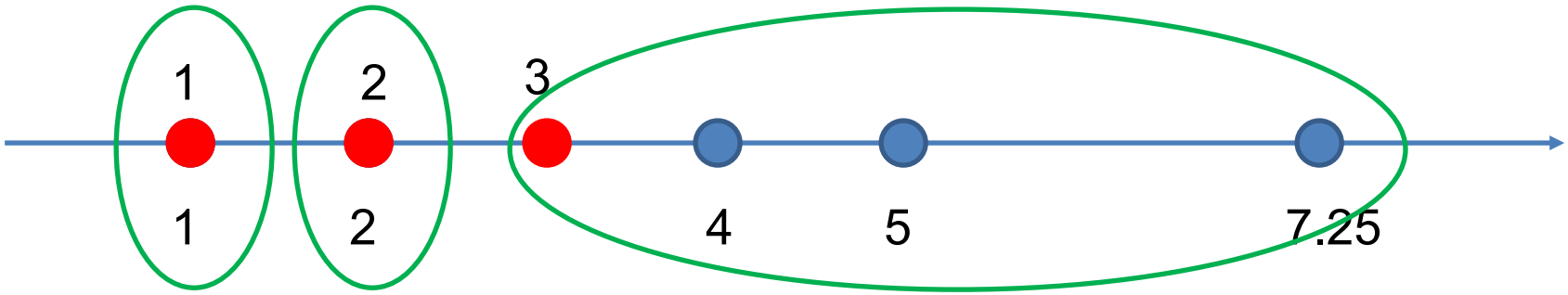
# Example 3

Initialize centers different from Example 1



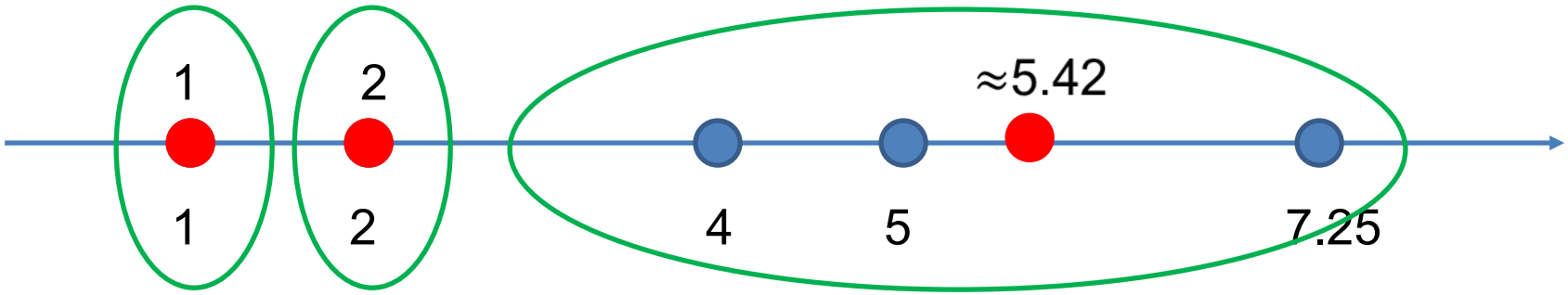
# Example 3

Assign the points to centers



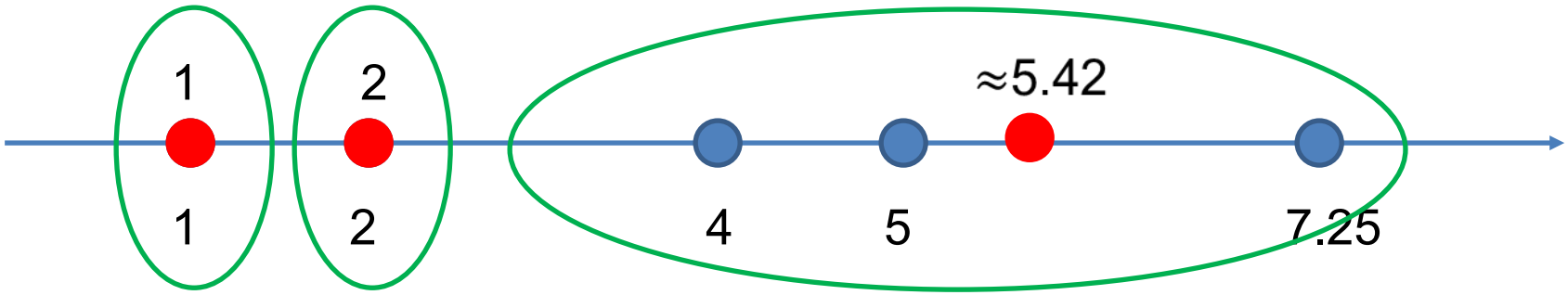
# Example 3

Update the centers



# Example 3

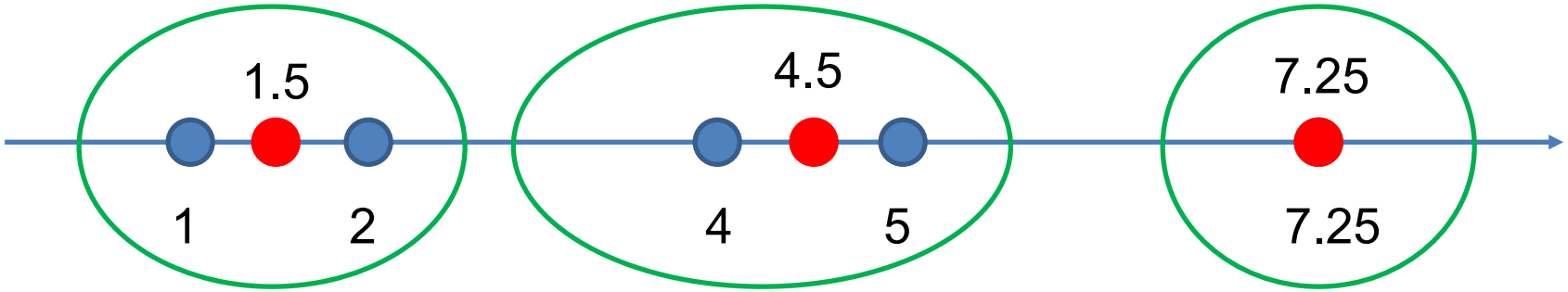
Assign the points to centers



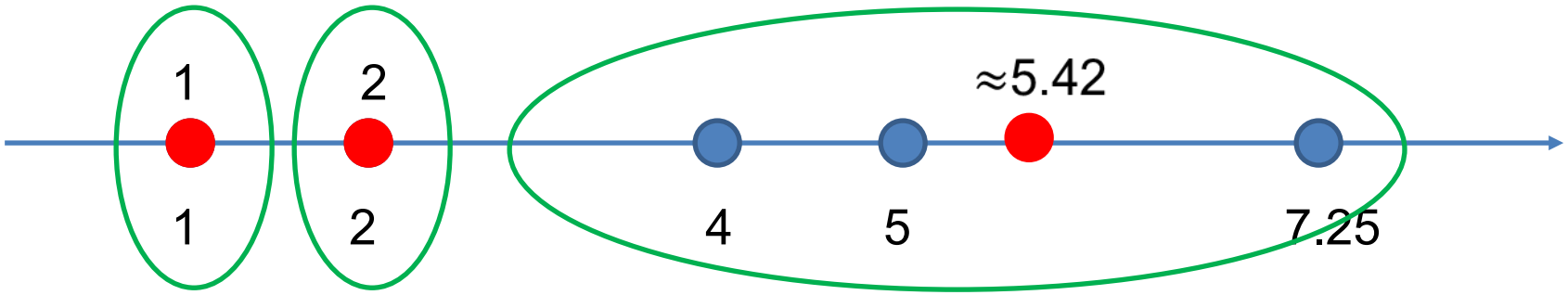

At this time point, the clusters don't change, and the centers don't change. Stop.

# Example 1 v.s. Example 3

Example 1




Example 3



# Picking starting cluster centers

- Which local optimum k-means goes is solely determined by the starting cluster centers

# 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 $x_1$ from dataset
    2. Find the point $x_2$ farthest from $x_1$ in the dataset
    3. Find $x_3$ farthest from the closer of $x_1, x_2$
    4. … pick $k$ points like this, use them as starting centers
  - 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. E.g., the Schwarz criterion

distortion + $\lambda(\#param) \log n$ = distortion + $\lambda dk \log n$