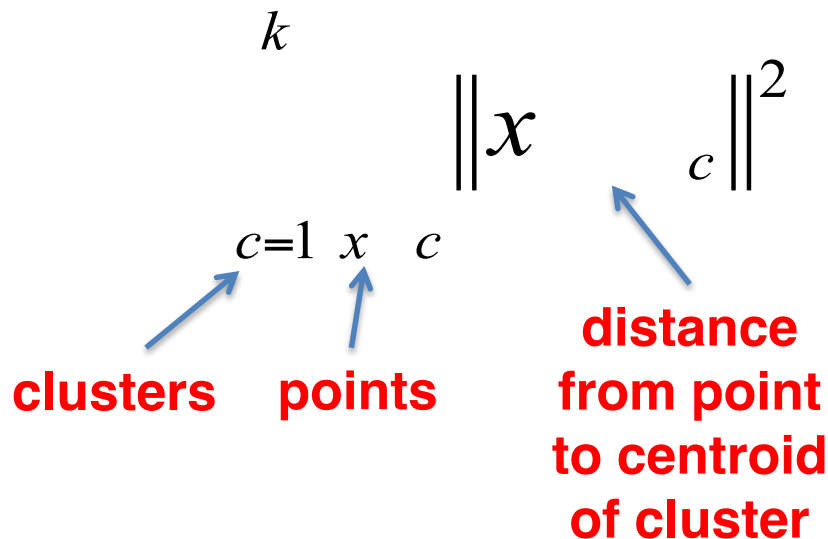


# **A Quick Introduction to Machine Learning (K-means Clustering)**

Lecturer: John Guttag

# K-means Clustering

**Given a set of points  $X$ , and a positive integer  $k$ , partition  $X$  into  $k$  clusters such that it approximately minimizes the objective function**



The diagram illustrates the K-means objective function. It features the mathematical expression  $\sum_{c=1}^k \sum_{x \in c} \|x - \mu_c\|^2$ . The term  $k$  is positioned above the summation symbol. The term  $c=1$  is placed below the first summation symbol, with a blue arrow pointing from the word "clusters" below it. The term  $x$  is placed between the two summation symbols, with a blue arrow pointing from the word "points" below it. The term  $c$  is placed below the second summation symbol. The term  $\|x - \mu_c\|^2$  is placed to the right of the second summation symbol, with a blue arrow pointing from the red text "distance from point to centroid of cluster" below it.

$$\sum_{c=1}^k \sum_{x \in c} \|x - \mu_c\|^2$$

**clusters**    **points**    **distance from point to centroid of cluster**

Minimizing the sum of the mean square differences

# K-means Algorithm

randomly choose k examples as centroids

while true:

    create k clusters by assigning each  
        example to closest centroid

    compute k new centroids by averaging  
        examples in each cluster

    if centroids don't change:

        break

(re-)assign points

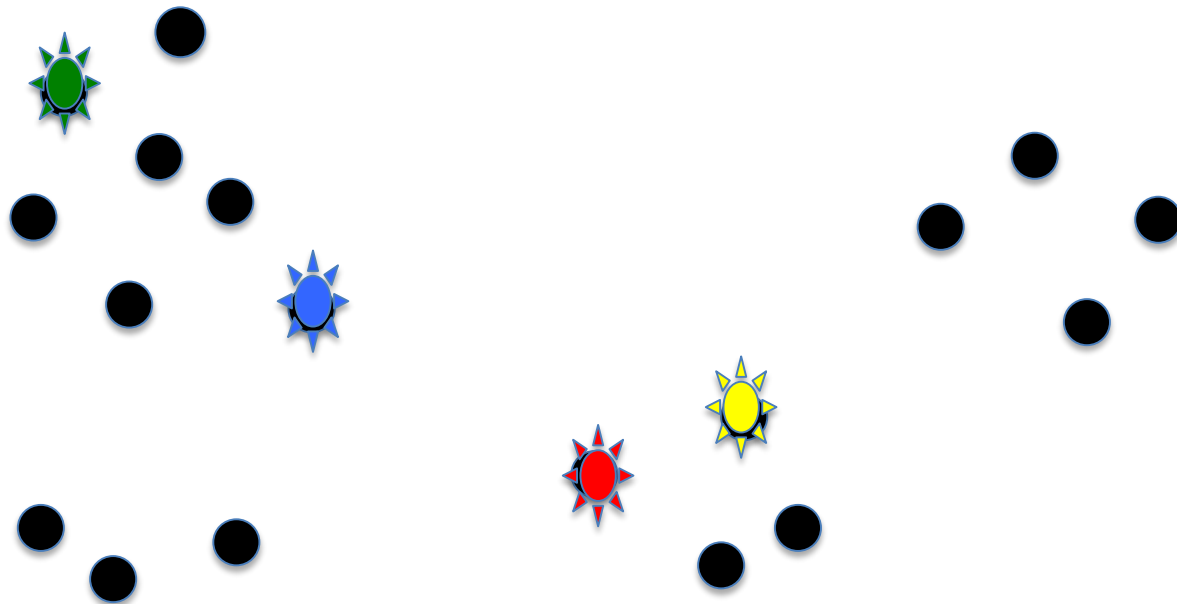
compute centroids

check if no points change

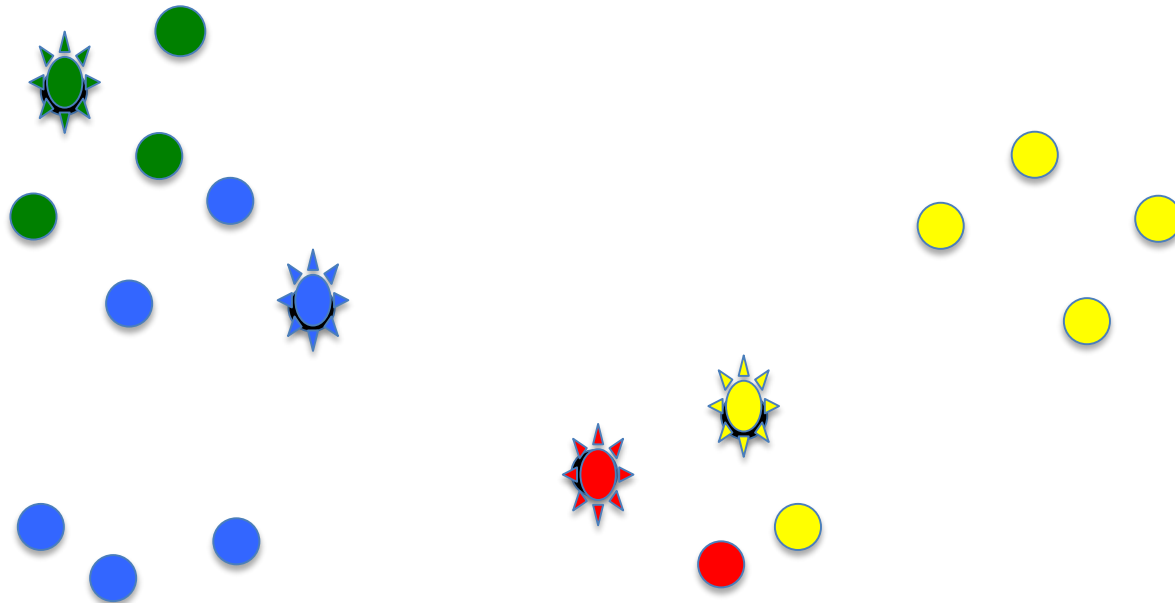
# Example



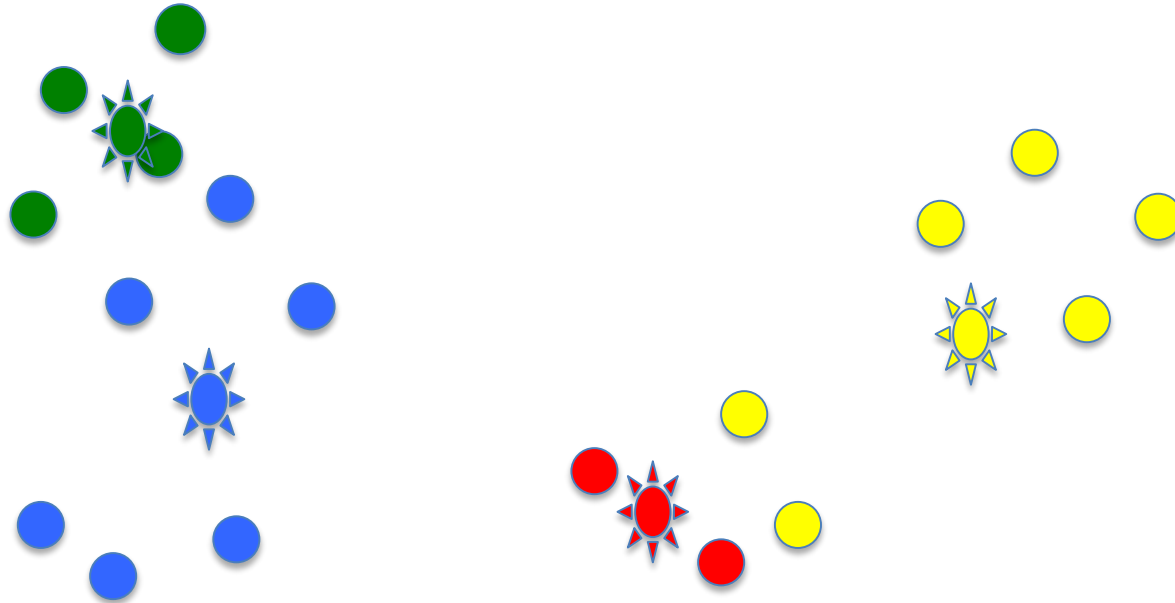
# Choose Initial Centroids ( $k = 4$ )



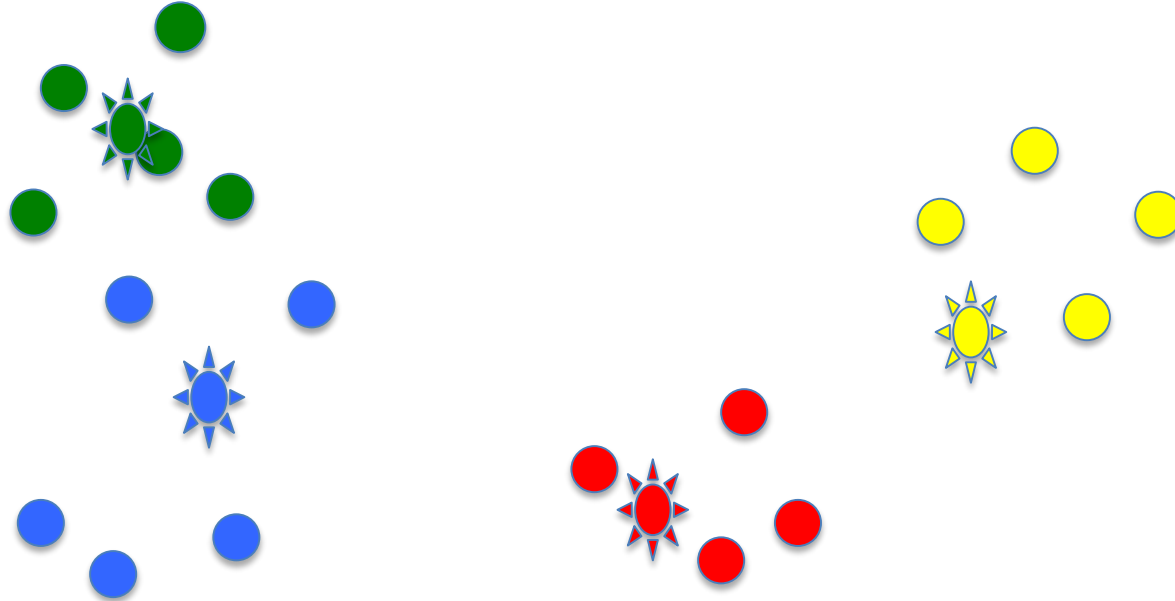
# Assign Points to Clusters



# Compute New Centroids

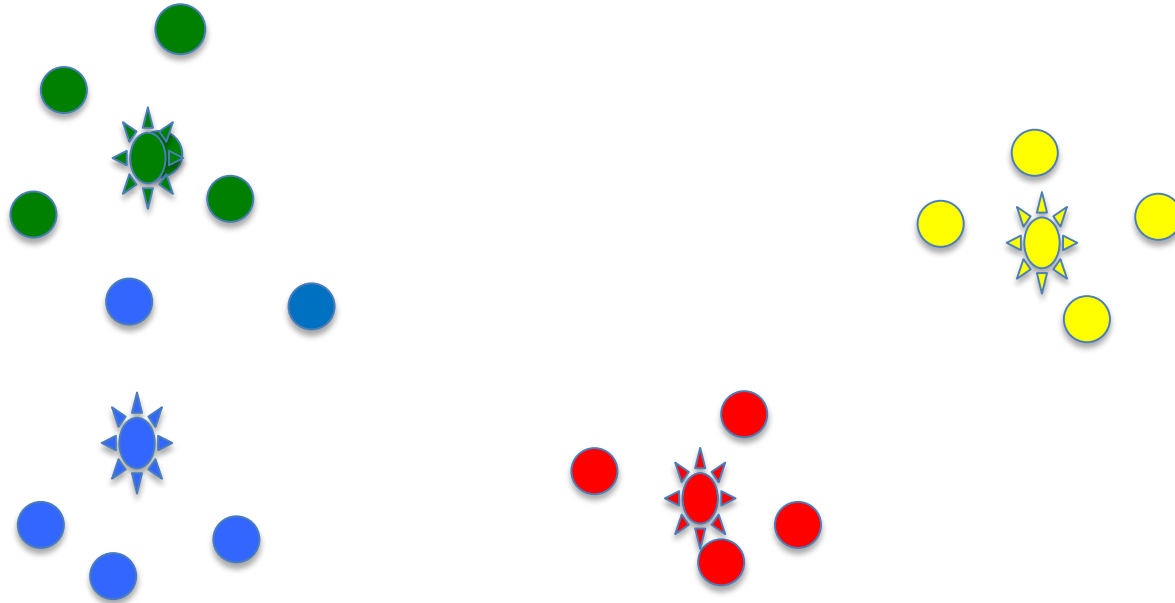


# Reassign Points to Clusters

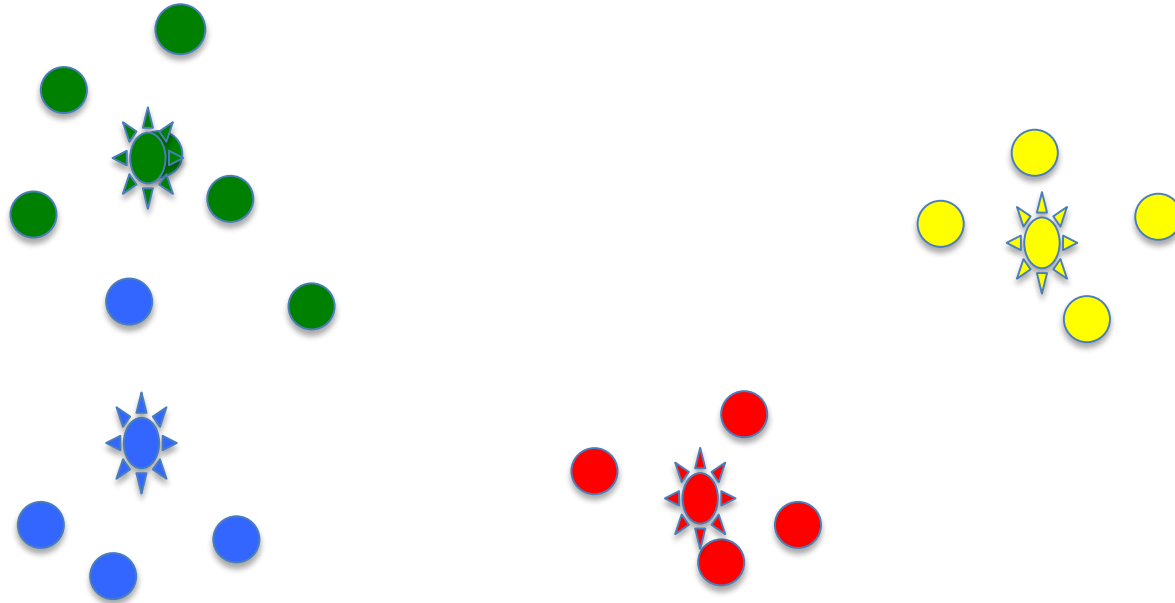




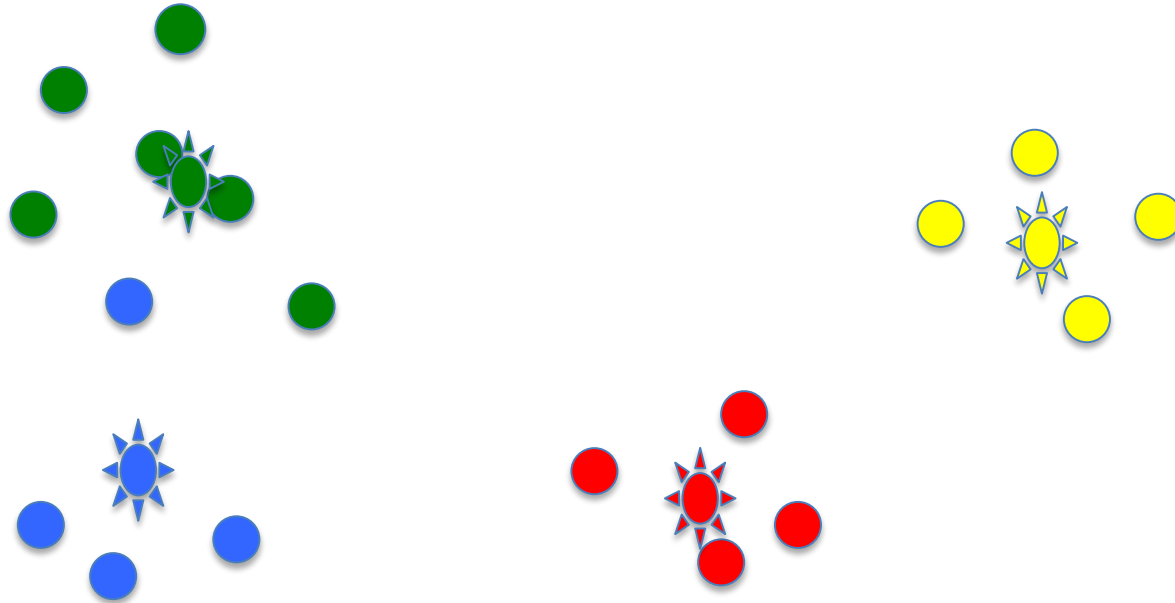
# Compute New Centroids



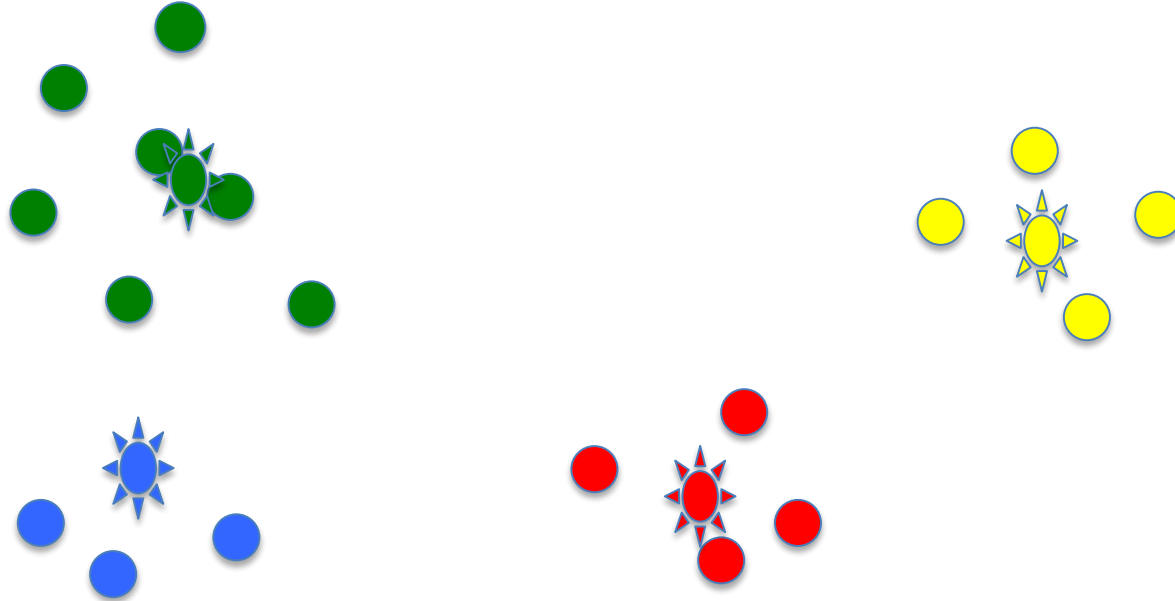
# Reassign Points



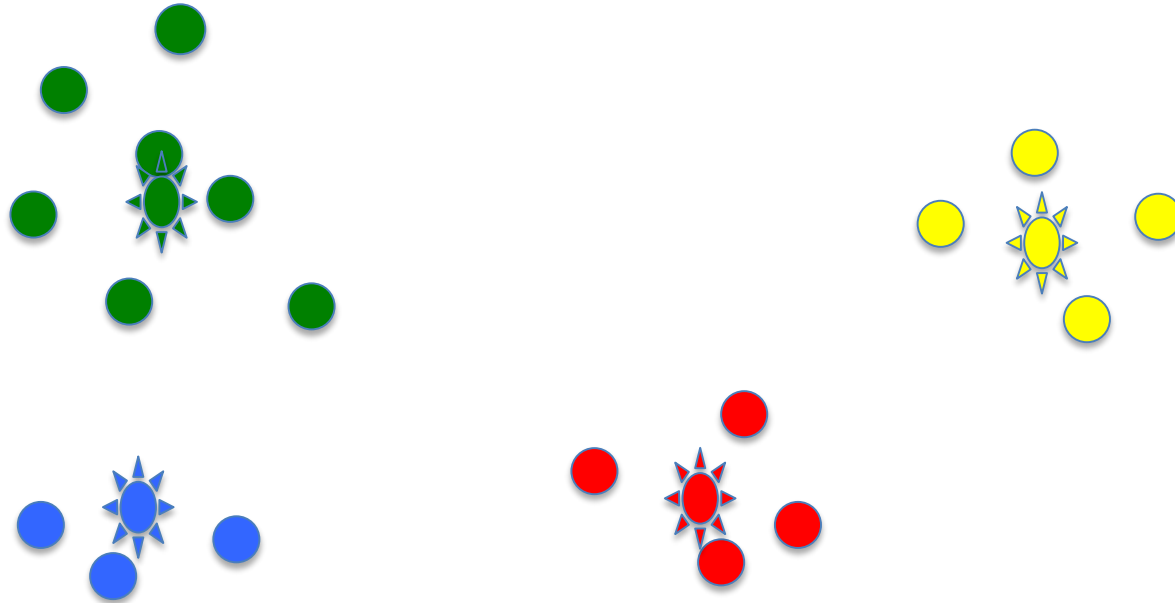
# Compute New Centroids



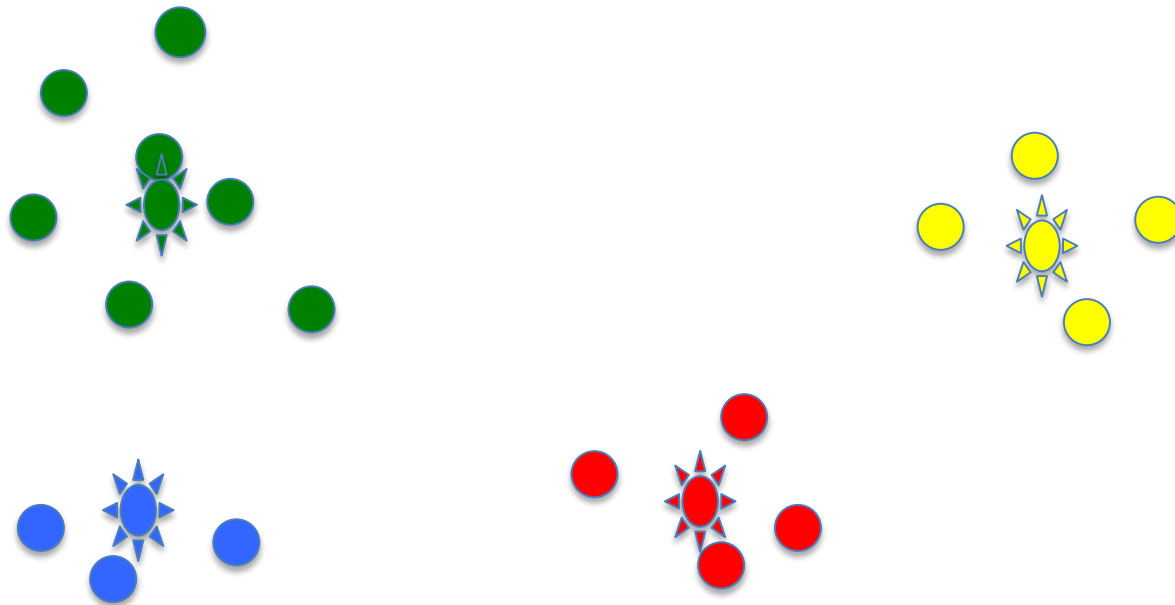
# Reassign Points



# Compute New Centroids



# No Points Move

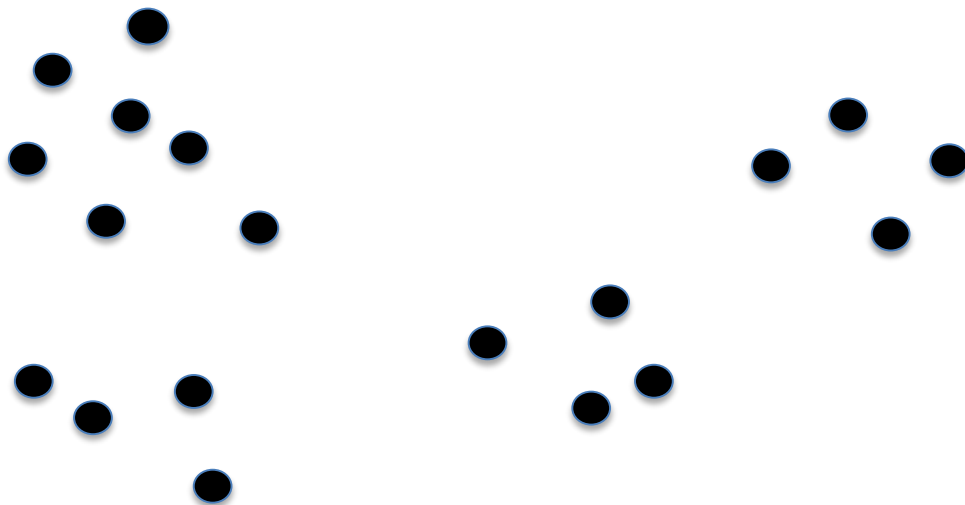


# Issues with K-means

**Final result can depend upon initial centroids**

**Greedy algorithm can find different local optima**

**Choosing the “wrong”  $k$  can lead to nonsense**



# Choosing K

## **A priori knowledge about application domain**

There are five different kinds of bacteria:  $k = 5$

There are two kinds of people in the world:  $k = 2$

## **Search for a good k**

Try different values of  $k$ , and evaluate quality of results



# Choosing Centroids

**Try multiple random choices and choose best**

# Finding the “Best” Solution

```
best = kMeans(points)
for t in range(numTrials):
    C = kMeans(points)
    if badness(C) < badness(best):
        best = C
```

$$V(c) = \sum_{\substack{x \in c \\ \text{属于}}} (\text{mean}(c) - x)^2 \quad \text{badness}(C) = \sum_{\substack{c \in C \\ \text{属于}}} V(c)$$

# Hierarchical vs. K-means

**Hierarchical looks at different numbers of clusters**  
**From 1 to n**

**K-means looks at many ways of creating k clusters**

**Hierarchical is slow**

**K-means is fast**

**Hierarchical is deterministic**

**K-means is non-deterministic**



CC-BY Image Courtesy of MrGuilt