DSC 40A

Theoretical Foundations of Data Science I

Announcements

- Homework 7 due 12/6.
- SET
- Final exam

Question

Answer at q.dsc40a.com

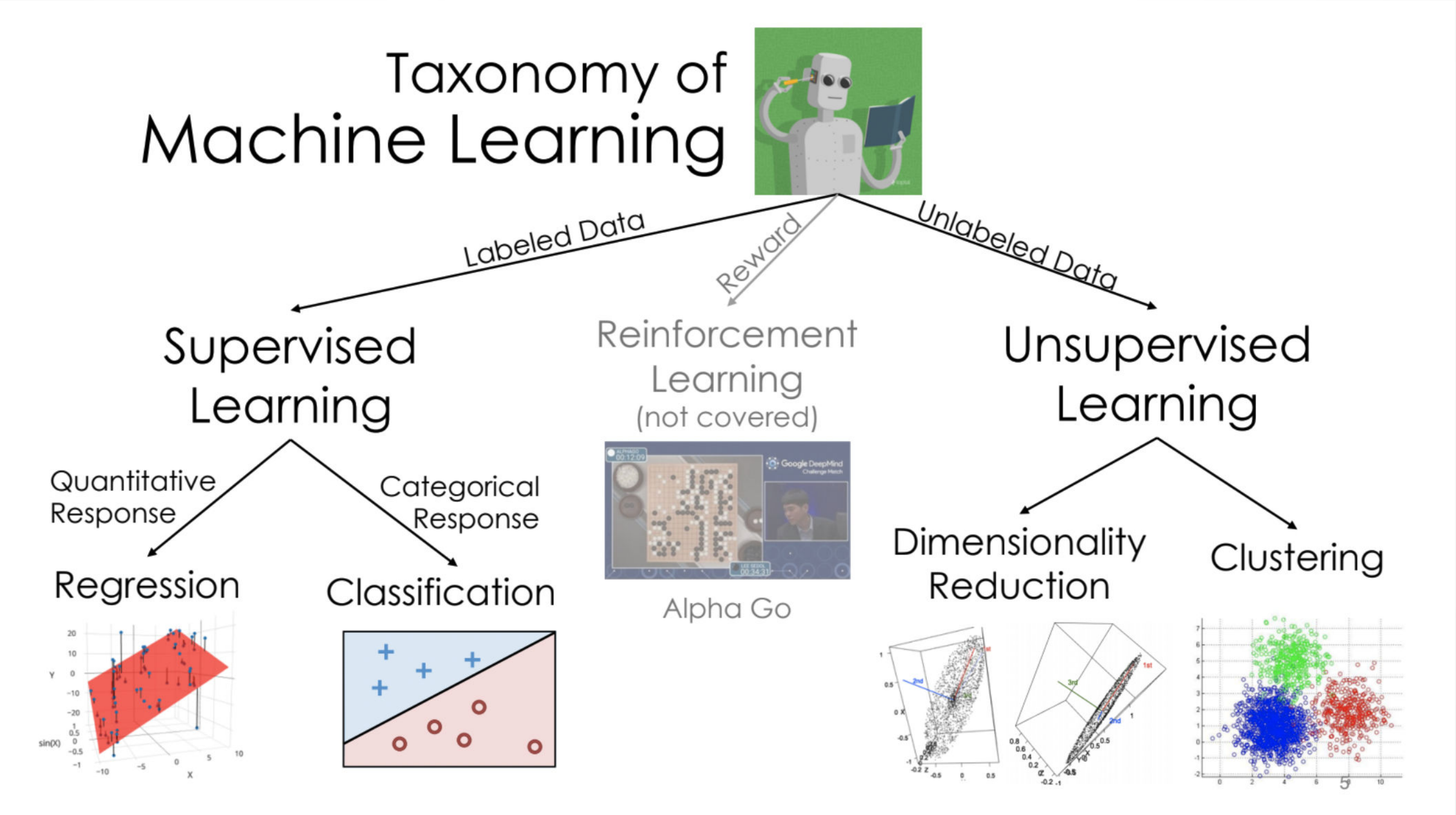
Remember, you can always ask questions at q.dsc40a.com!

If the direct link doesn't work, click the "Lecture Questions" link in the top right corner of dsc40a.com.

Outline

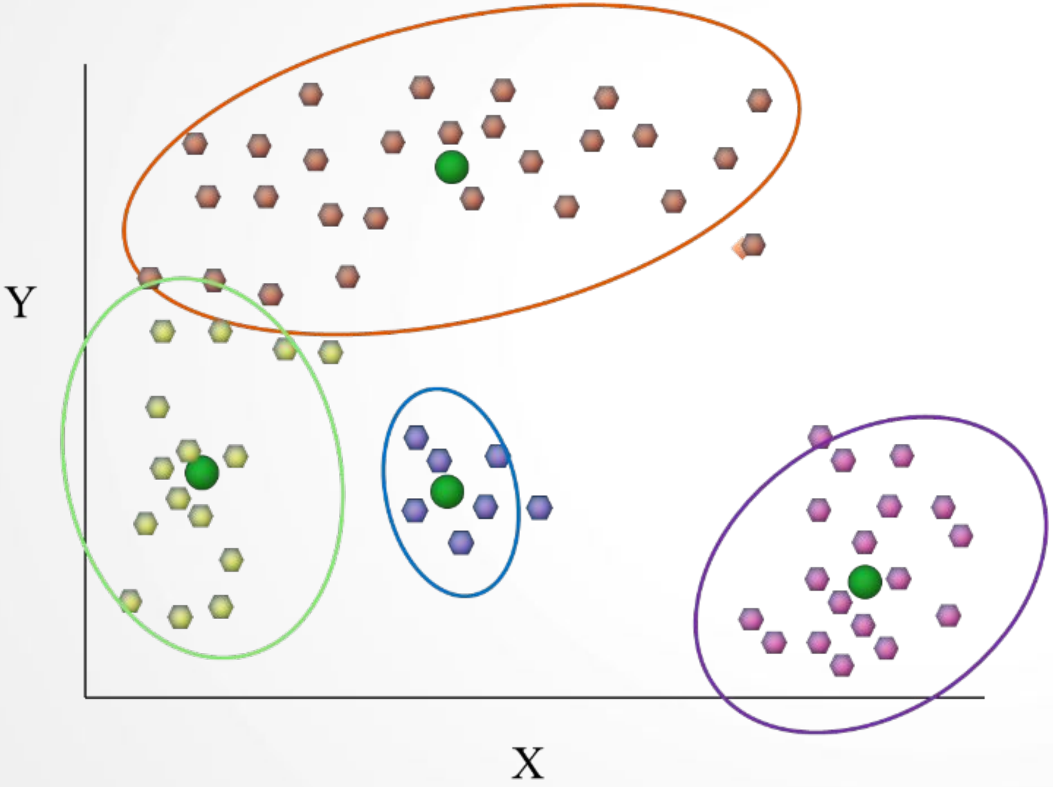
- We'll look at the clustering problem in machine learning and an algorithm that solves this problem.
- Look out for connections to loss functions and risk minimization!

Today



Taxonomy of
Machine Learning
Labeled Data
Reward
Unlabeled Data
Supervised Learning
Reinforcement Learning
(not covered)
Unsupervised Learning
Quantitative Response
Categorical Response
Regression
Classification
Alpha Go
Dimensionality Reduction
Clustering

Clustering: Applications



- Bot detection
- Marketing to different subpopulations
- Discovering structure:
  - strains of viruses
  - new species
  - communities in a social network
  - chemicals properties

Clustering: Problem Statement

Given a list of n data points (or vectors) in R^d

$$x_1, x_2, \ldots, x_n$$

and a positive integer, k,

group the data points into k groups (clusters) of nearby points.

Clustering: Problem Statement

Given a list of n data points (or vectors) in R^d

$$x_1, x_2, \ldots, x_n$$

and a positive integer, k,

group the data points into k groups (clusters) of nearby points.

Which of these inequalities should be true?

A. $d < n$
B. $n < d$
C. $k < n$
D. $n < k$

How to define groups?

Pick k cluster centers (centroids),

$$\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$$

These k centroids define the k groups, by placing each data point in the group corresponding to the nearest centroid.

How to define centroids?

Choose the k cluster centers (centroids) to minimize a cost function.

Cost($\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, ... , $\boldsymbol{\mu}_k$) = total squared distance of each data point x_i to its nearest centroid $\boldsymbol{\mu}_j$

Lloyds Algorithm, or k-Means Clustering

1. Randomly initialize the k centroids.
2. Keep centroids fixed. Update groups.

   Assign each point to the nearest centroid.
3. Keep groups fixed. Update centroids.

   Move each centroid to the center of its group.
4. Repeat steps 2 and 3 until done.

Step 1: Randomly initialize the k centroids.

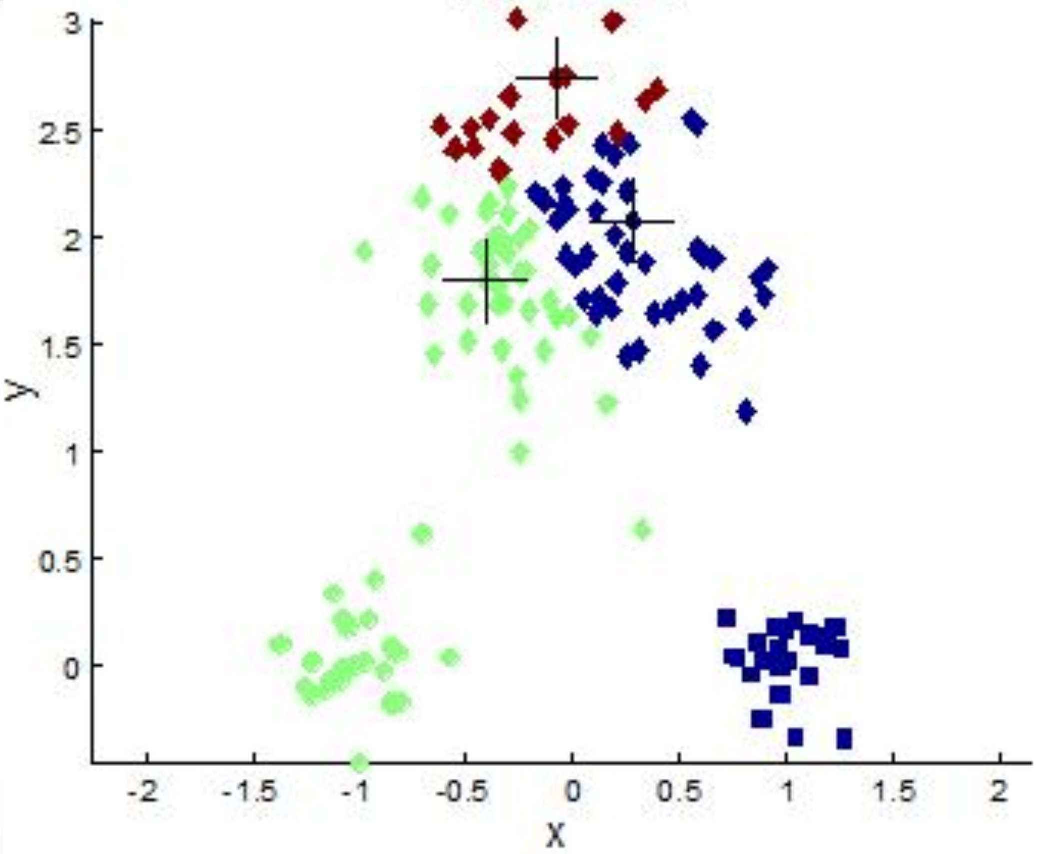
Two common strategies:

- Randomly select k of the data points x_i.
- Randomly assign each data point to one of k groups. Set the centroid of each group to be the center of the points assigned to that group.

Step 2: Keep centroids fixed. Update groups.

For each point,

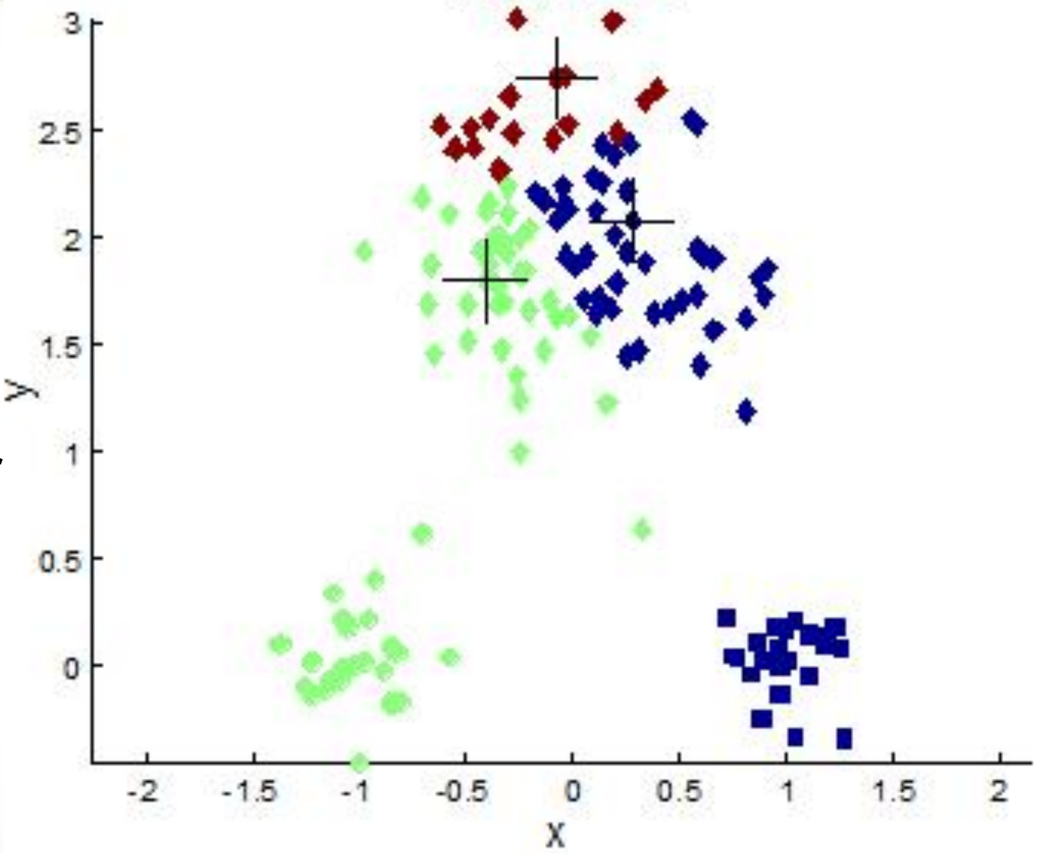
- find the nearest centroid and
- add the point to a group corresponding to that nearest centroid.



Step 3: Keep groups fixed. Update centroids.

For each centroid,

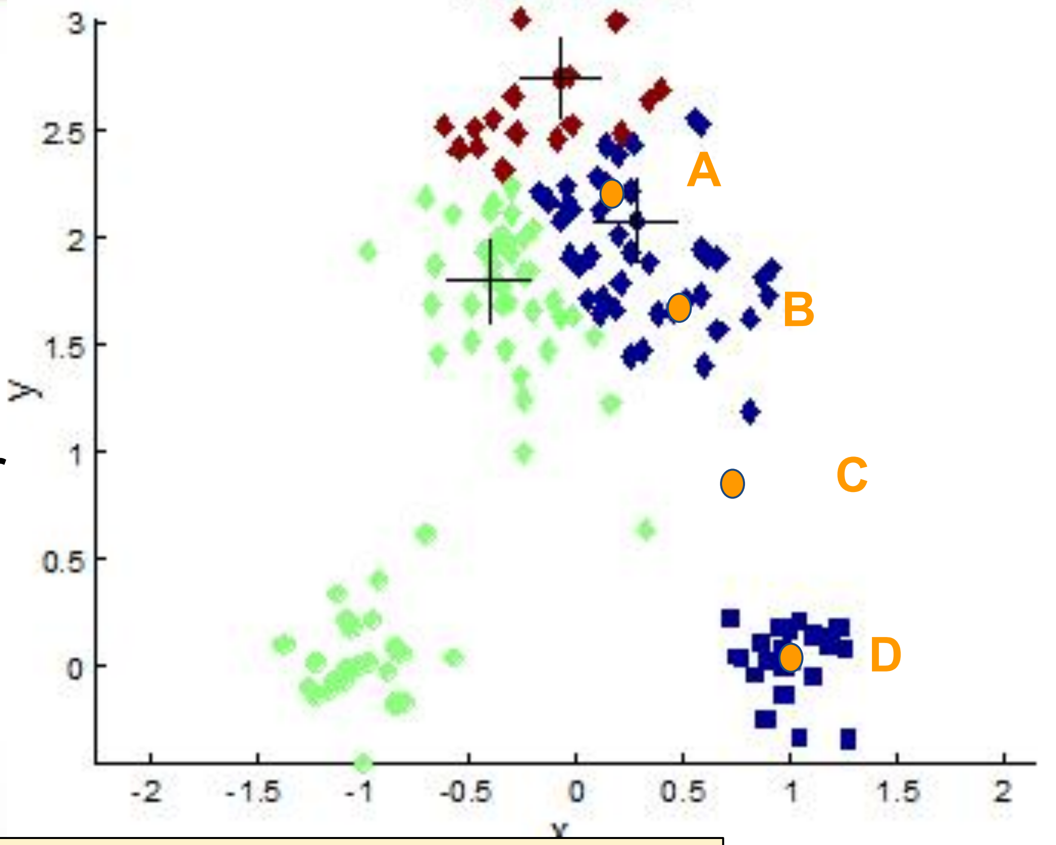
- average the coordinates of all data points in the group, and
- move the centroid to this center point with average coordinates.



Step 3: Keep groups fixed. Update centroids.

For each centroid,

- average the coordinates of all data points in the group, and
- move the centroid to this center point with average coordinates.



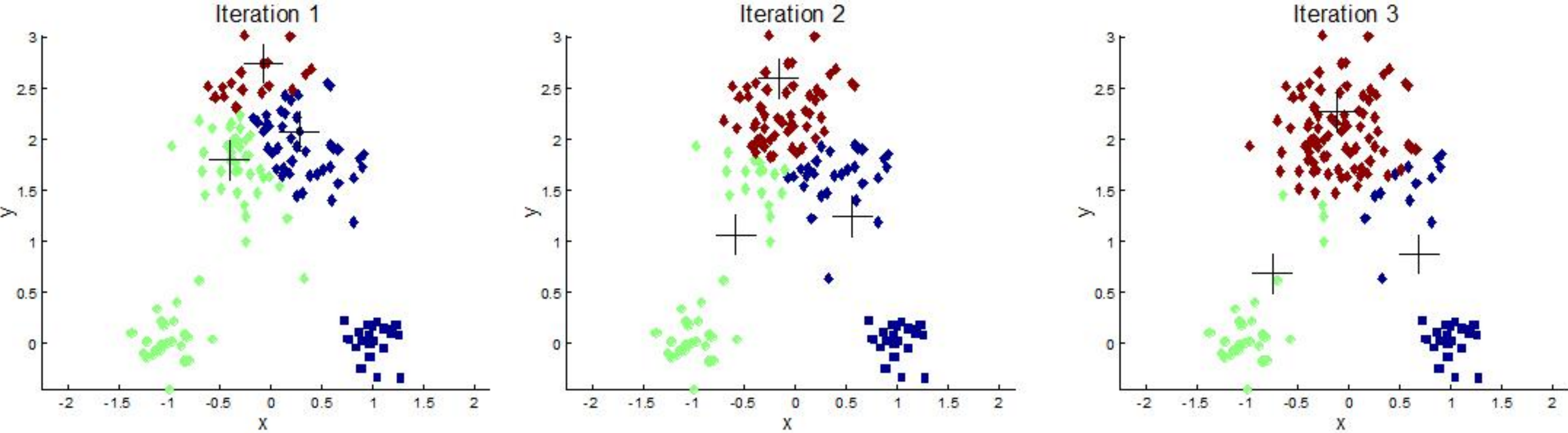
For the blue group of points, approximately where will the centroid move to?

Step 4: Repeat steps 2 and 3 until done.

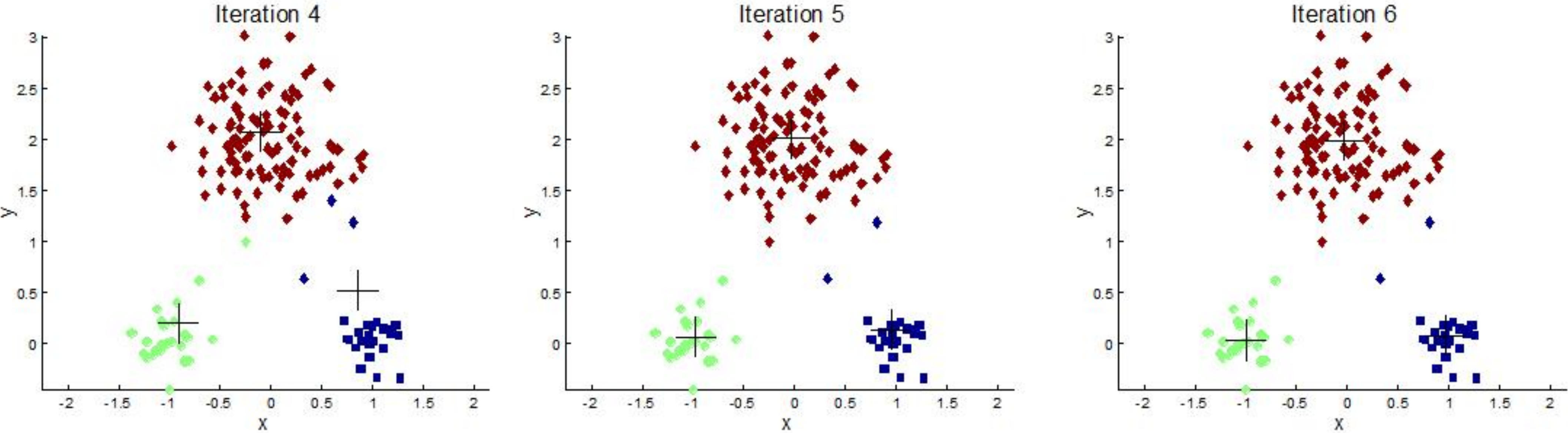
Done when:

- max number of iterations is reached, or
- centroids don’t move (at all, or very much), or
- groups don’t change (at all, or very much)

k-Means Clustering Example



k-Means Clustering Example



Summary

- We described the clustering problem and the k-means algorithm, which solves this problem.
- **Next time:** We'll see that updating the centroids according to this algorithm reduces the cost with each iteration.

Cost($\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, ... , $\boldsymbol{\mu}_k$) = total squared distance of each data point x_i to its nearest centroid $\boldsymbol{\mu}_j$