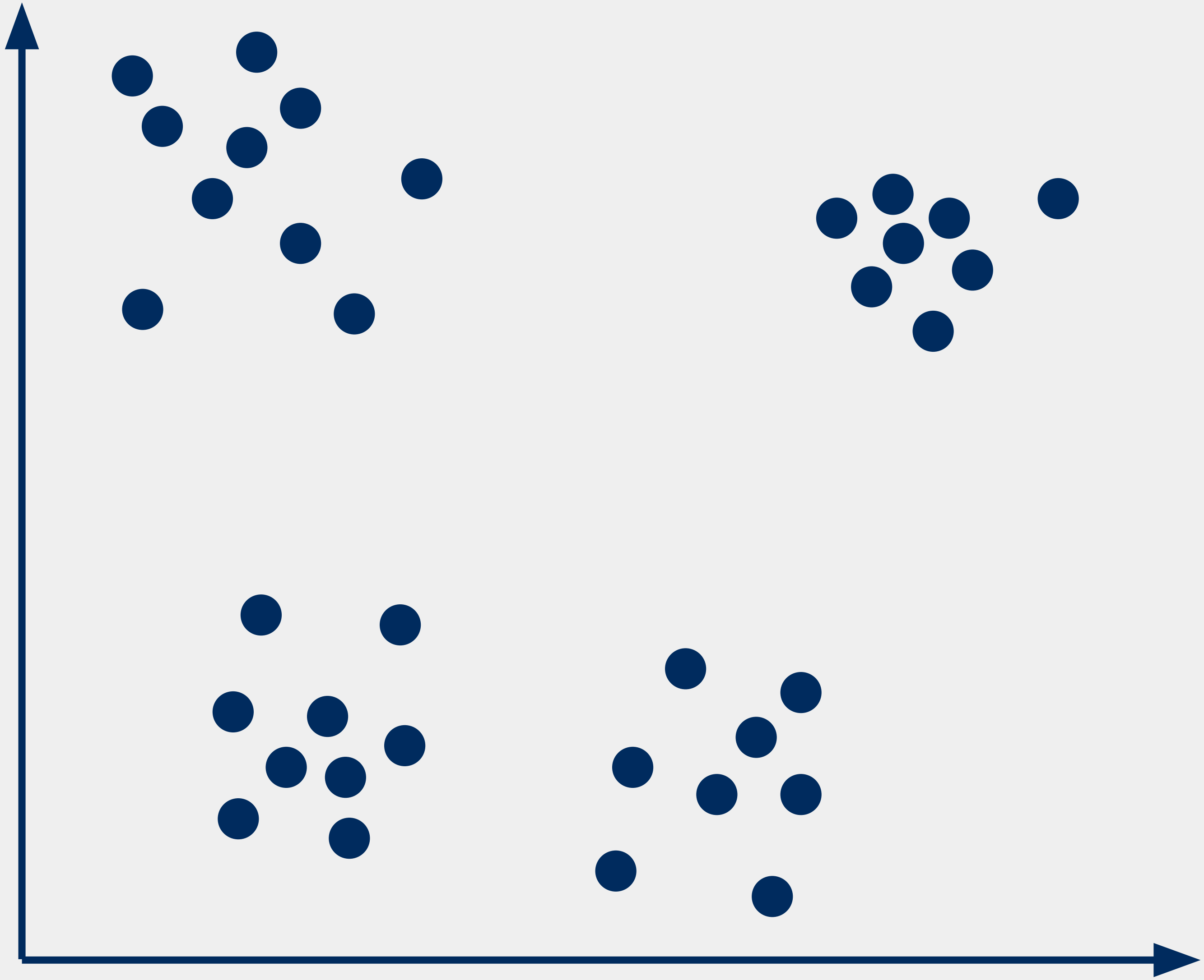
CS 1501

Clustering

The Clustering Problem

Given a collection of examples, group them into classes based on how similar they are to each other

Clustering example



Machine learning approaches

- Supervised learning
  - Use a dataset of labelled examples (training) to produce a model that can output an appropriate label for new inputs
- Unsupervised learning
  - Use an unlabelled dataset to produce a model to map inputs onto useful values or vectors
- Semi-supervised learning
  - Same goal as supervised learning, but input also contains a (usually much larger) set of unlabelled examples
- Reinforcement learning

Machine learning terms

- Our data set is a collection of *examples*, each of which has attributes (or *features*) in common with other examples
  - Each example with *d* attributes is described in the input by a *d*-dimensional *feature vector*
- A *hyperparameter* is set before running the algorithm
  - In contrast to a *parameter* that helps to define the model learned by the learning algorithm
  - For our clustering examples, the number of clusters desired could be a hyperparameter

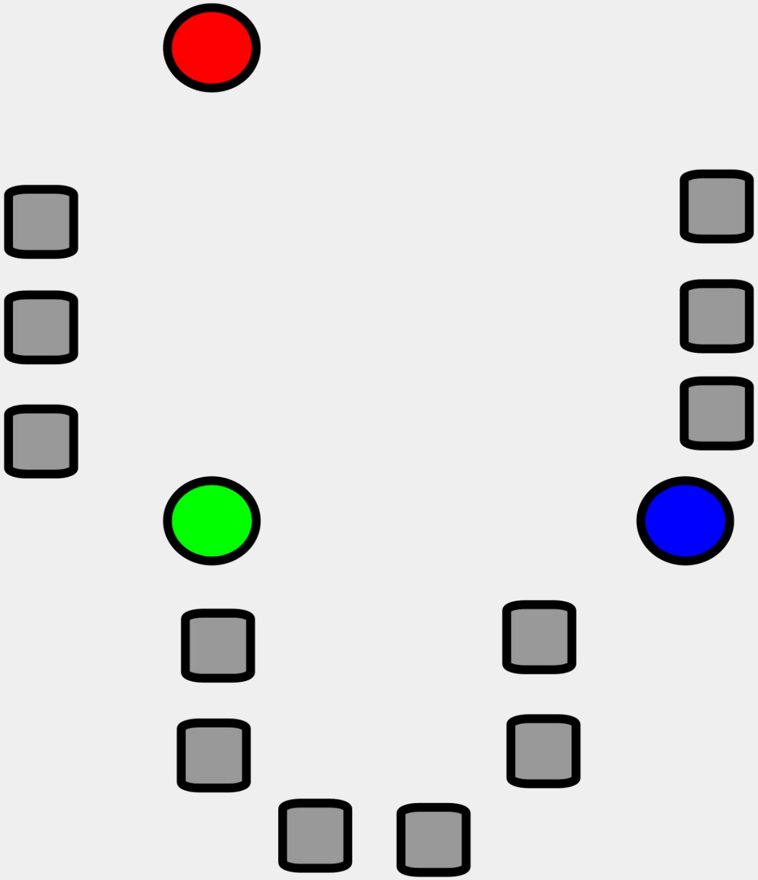
The k-means Clustering Problem

Given a collection of *d*-dimensional feature vectors, group them into clusters that minimize the sum of distances from each example to the centroid of its cluster

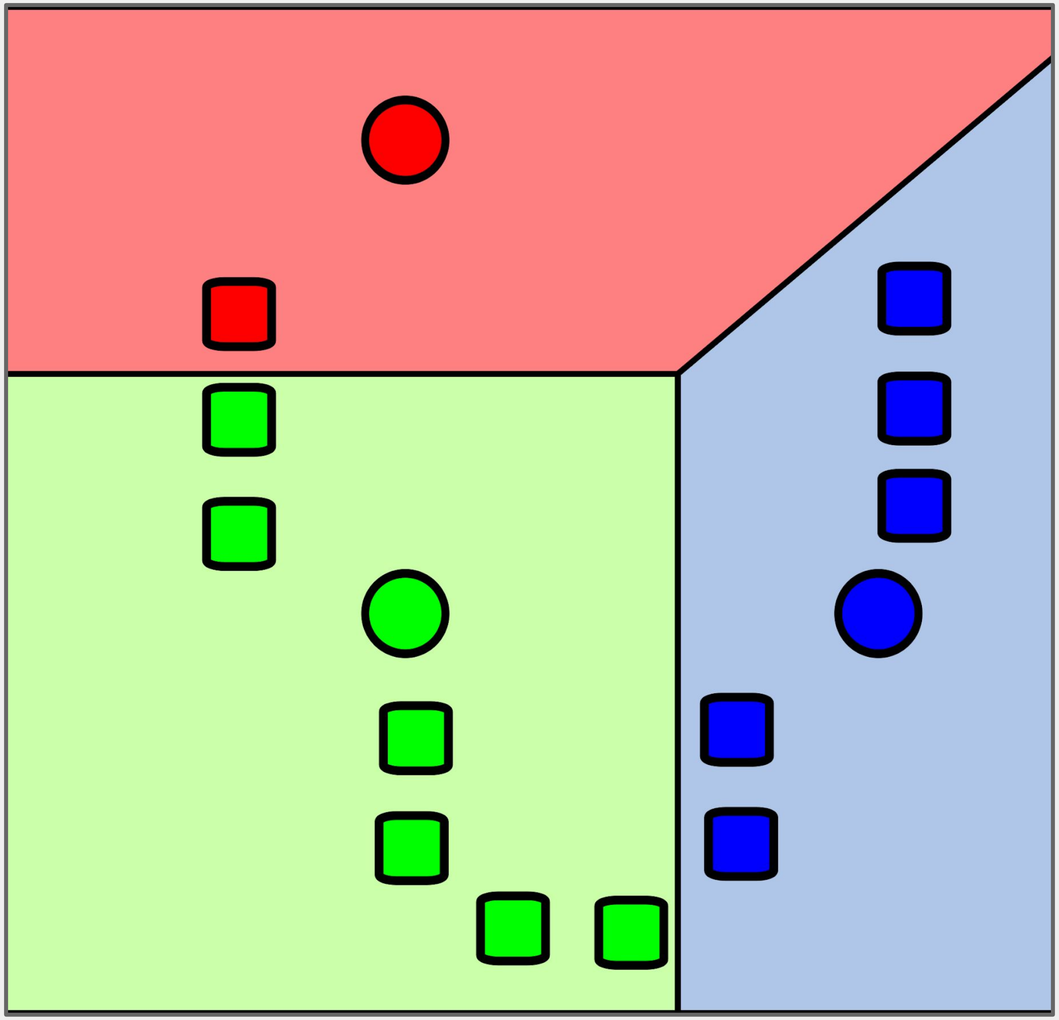
Naive k-means (Lloyd's)

- Pick starting "means" (initial centroids)
  - E.g., randomly pick examples from the input set
  - E.g., randomly pick points in *d*-dimensional space
- Assign all examples to a cluster based on the centroid they are closest to by some distance metric (e.g., Euclidean)
- For each cluster, compute the new centroid as the mean of all feature vectors in that cluster
- Redo assignments/centroid calculation until convergence
  - I.e., no examples are assigned to new clusters after new centroids are computed

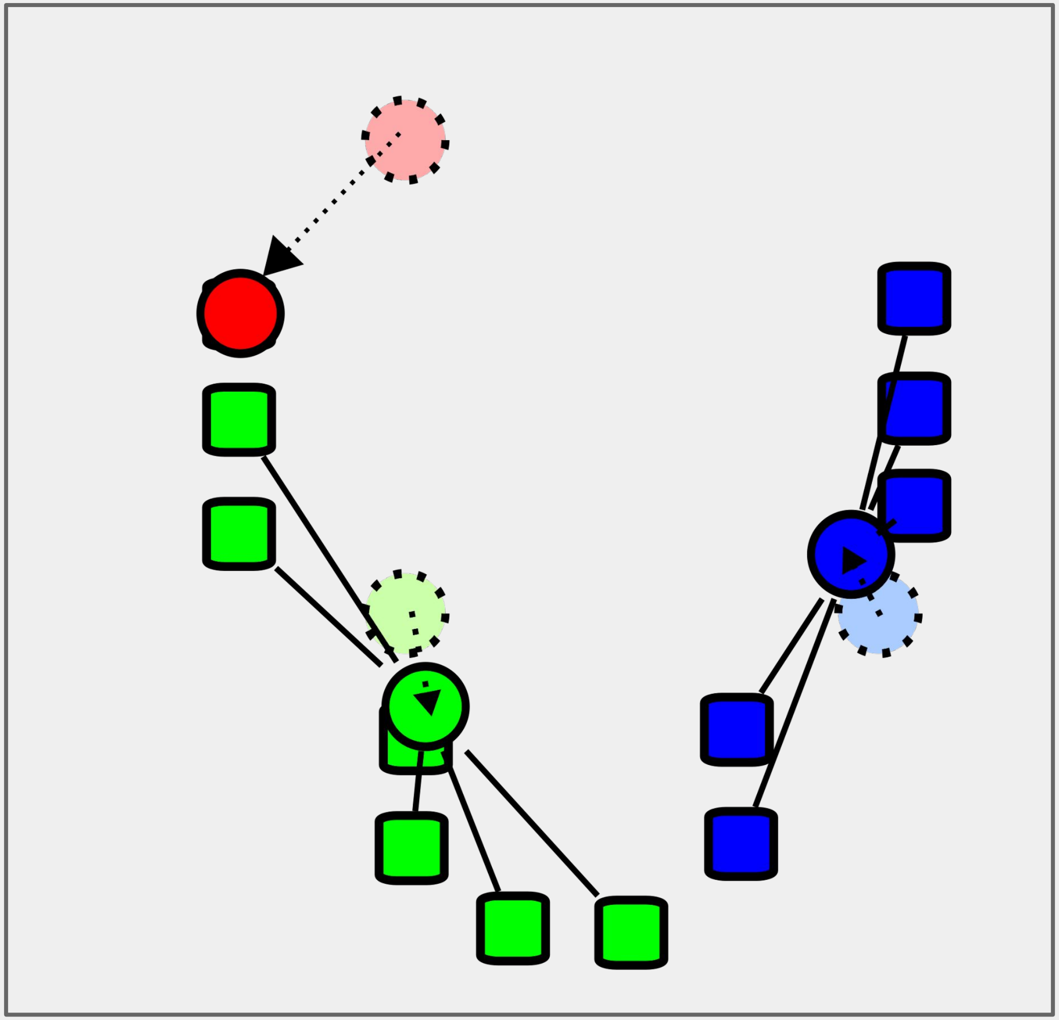
k-means Example 1



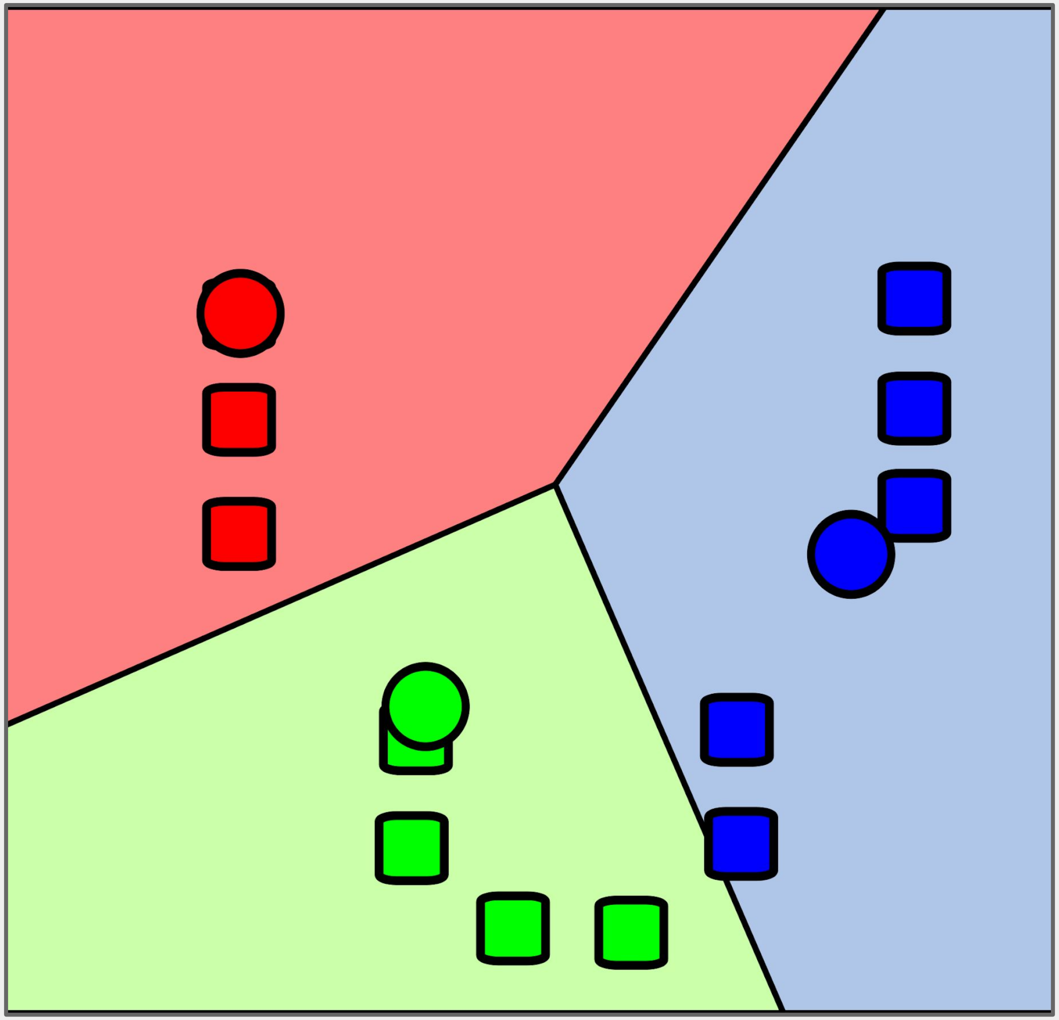
k-means Example 1



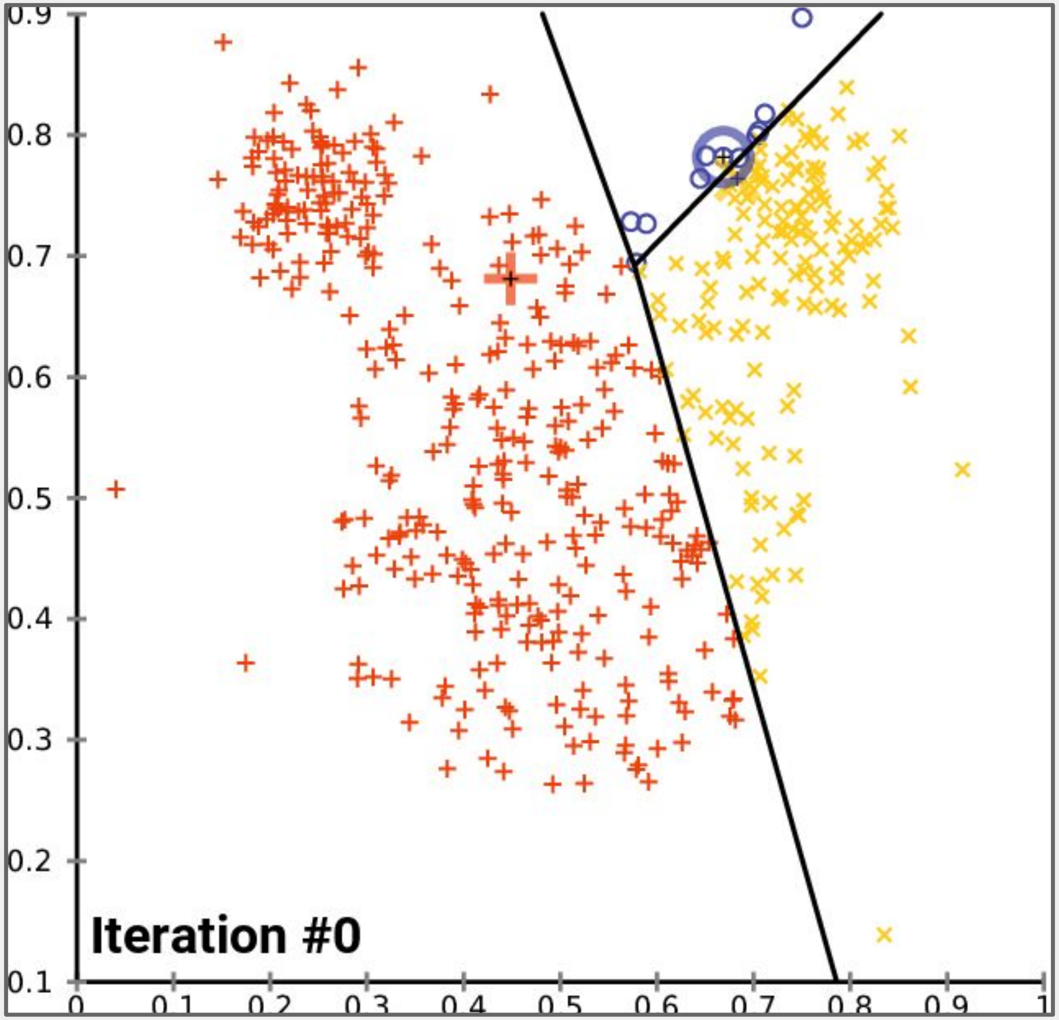
k-means Example 1



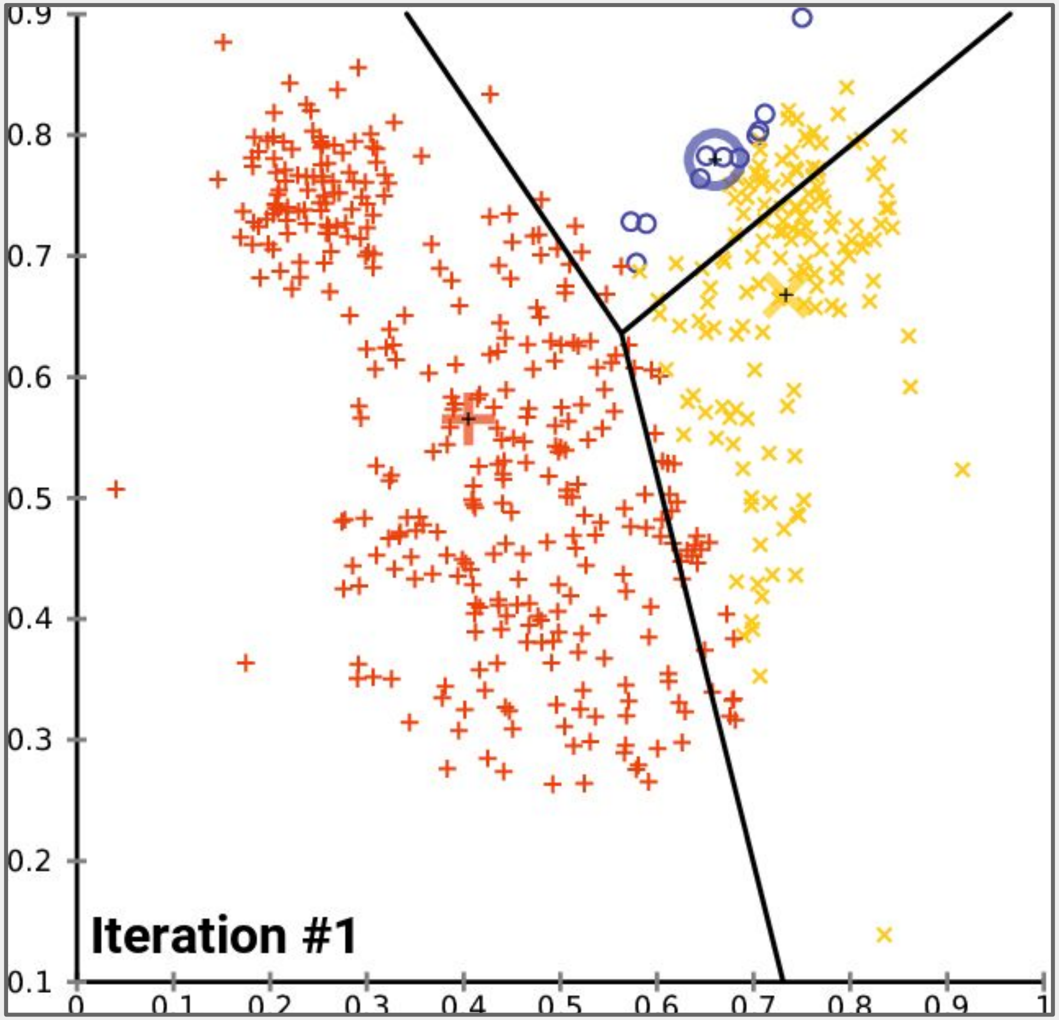
k-means Example 1



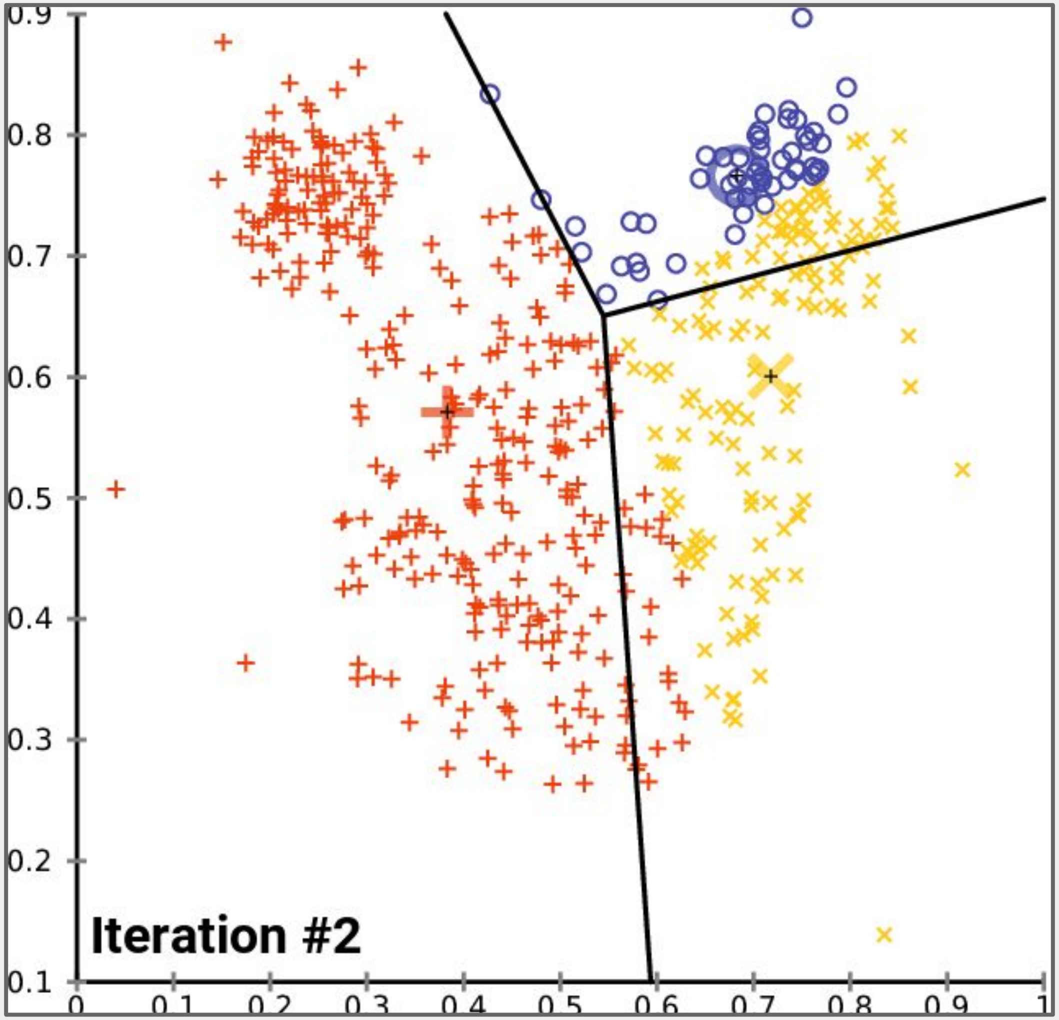
k-means Example 2



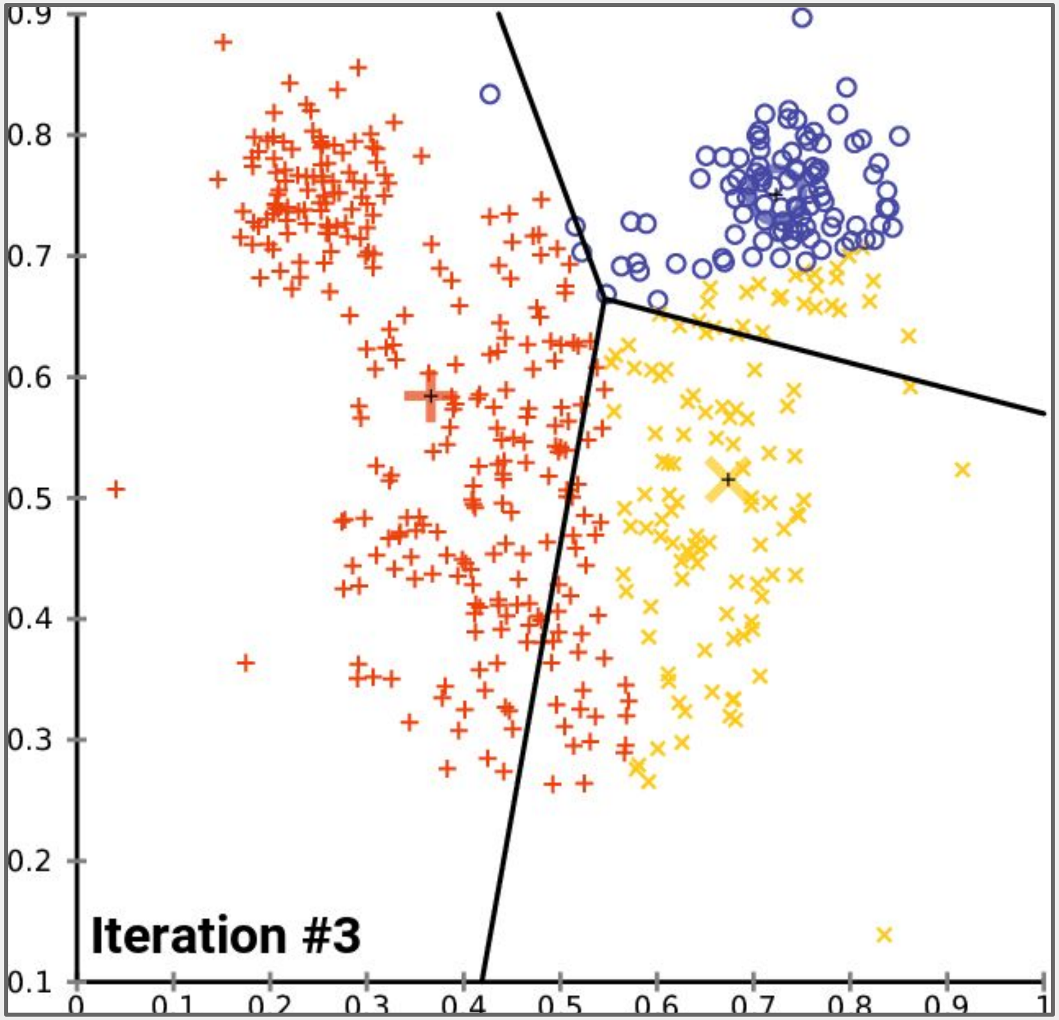
k-means Example 2



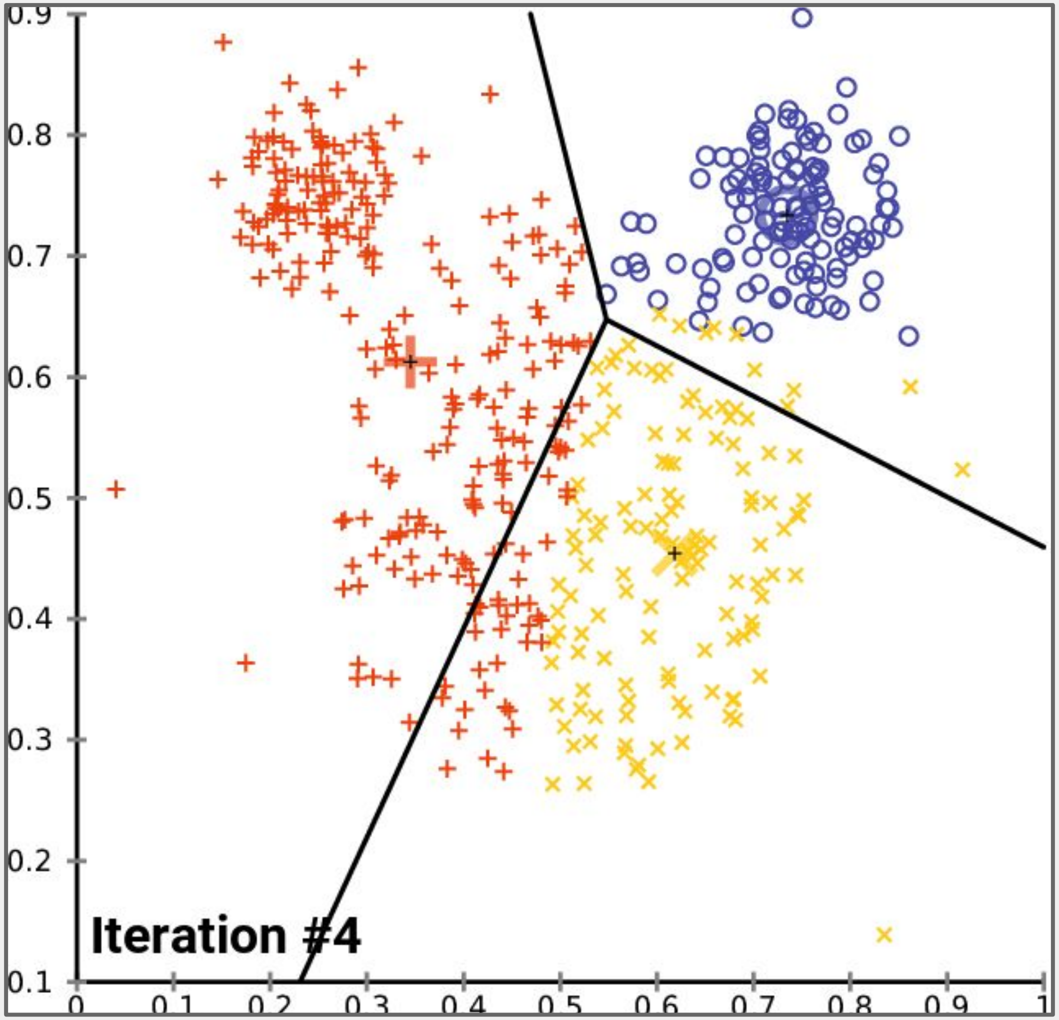
k-means Example 2



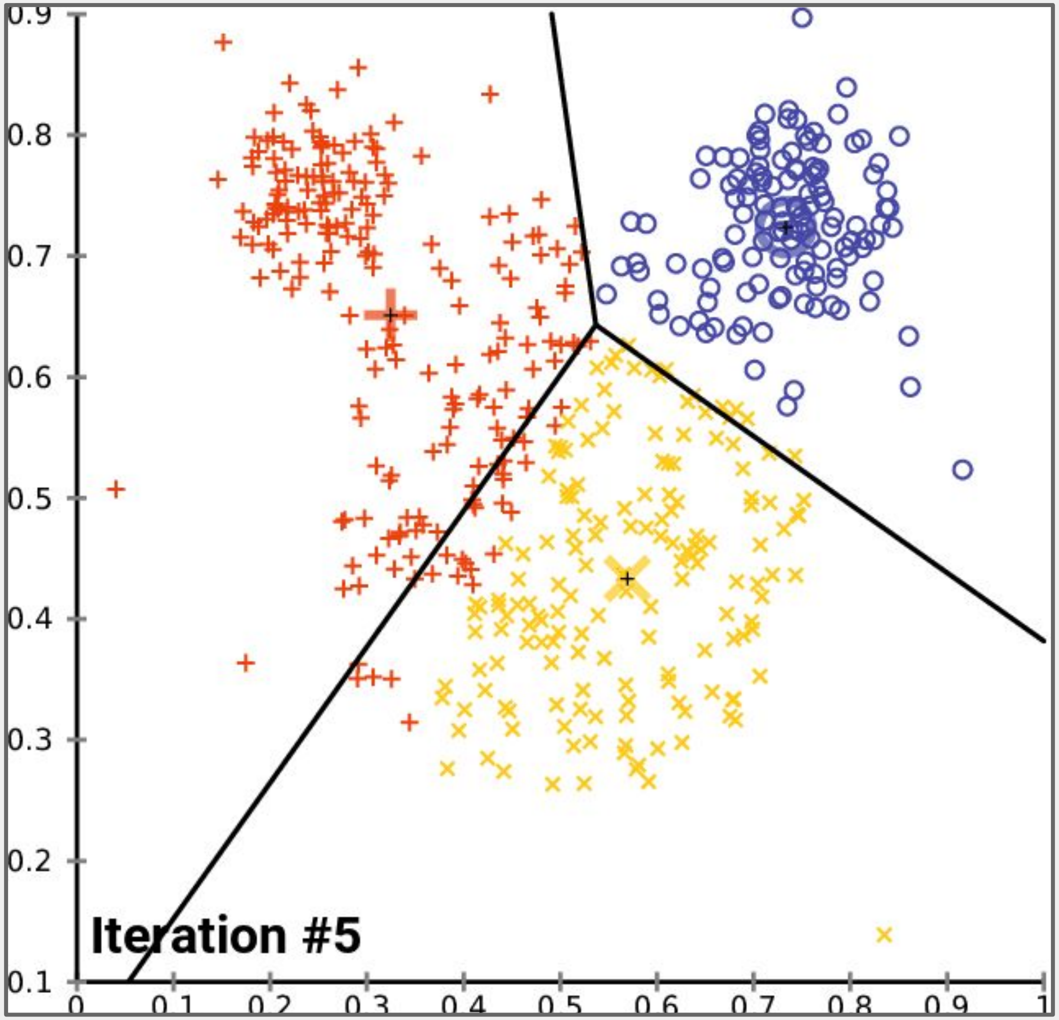
k-means Example 2



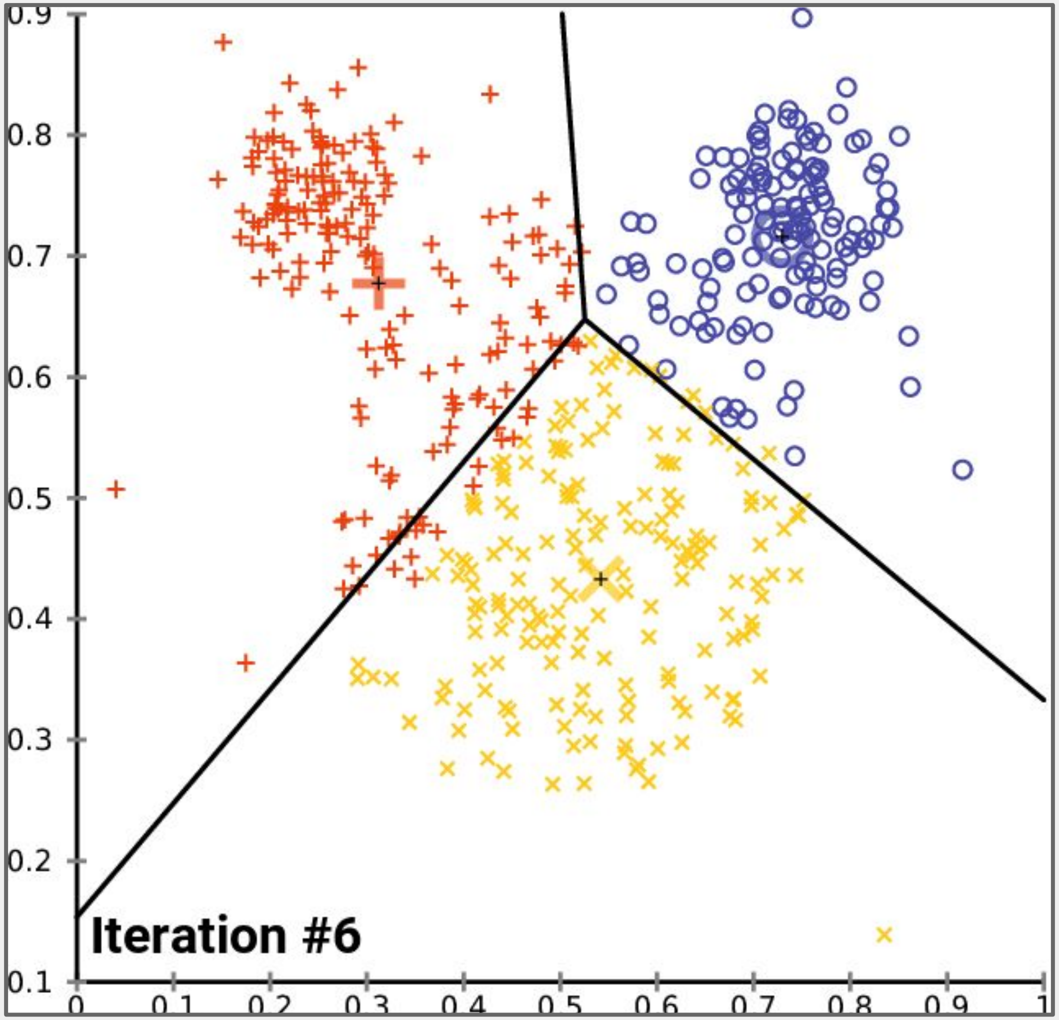
k-means Example 2



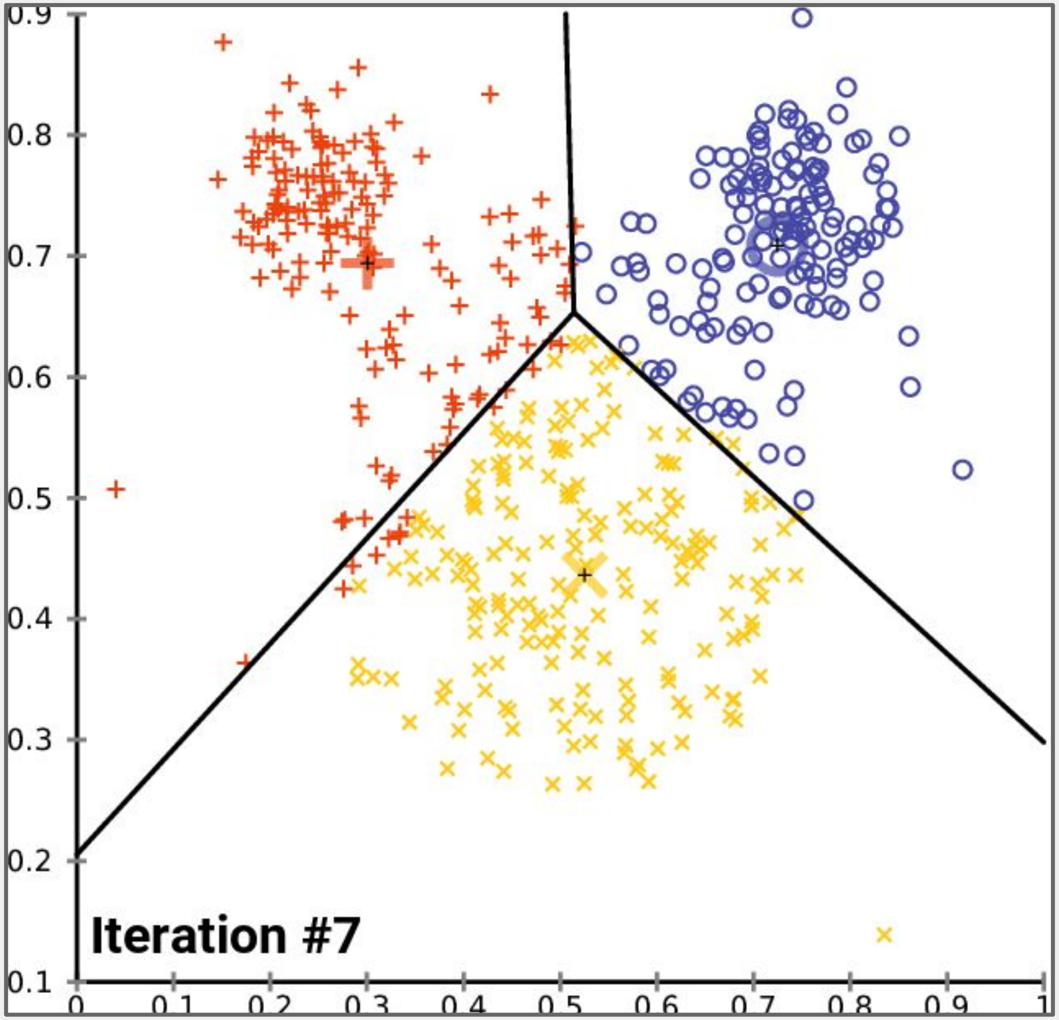
k-means Example 2



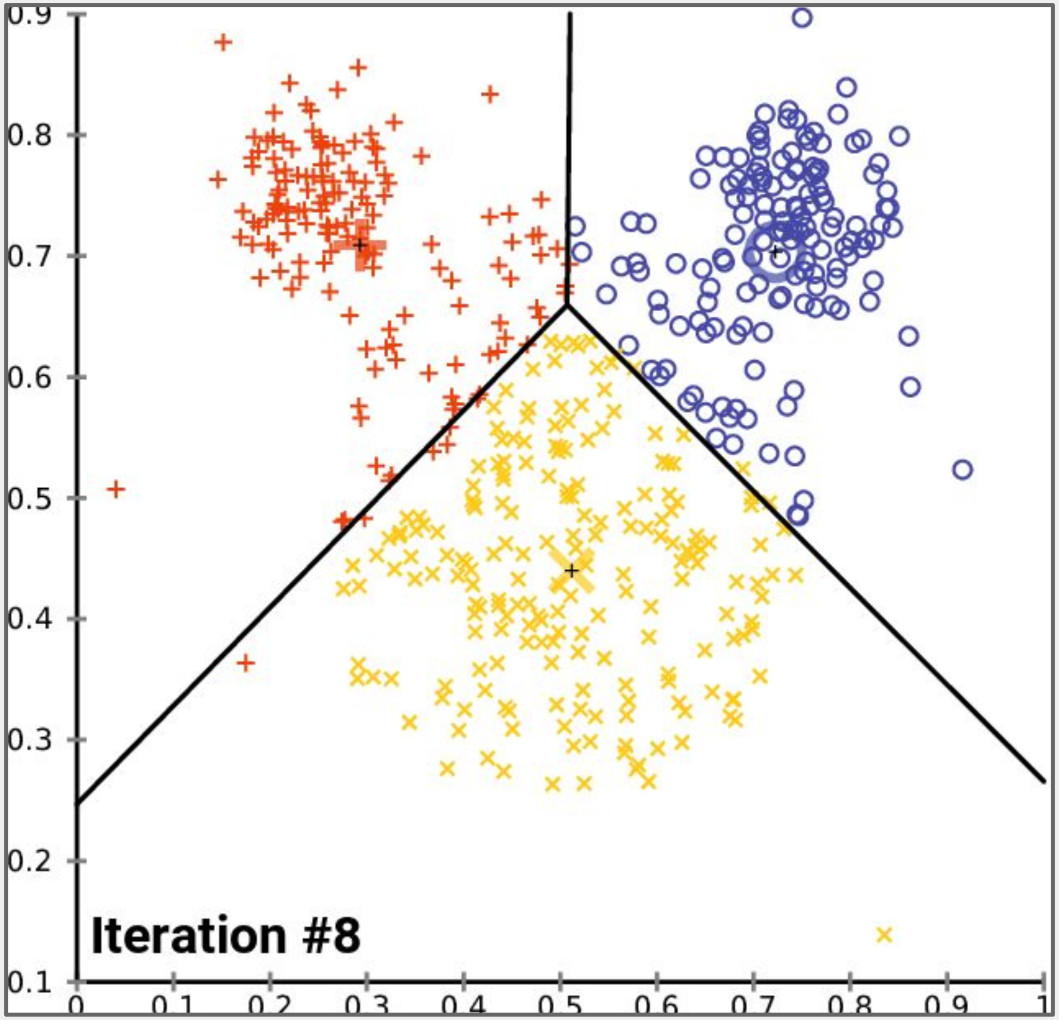
k-means Example 2



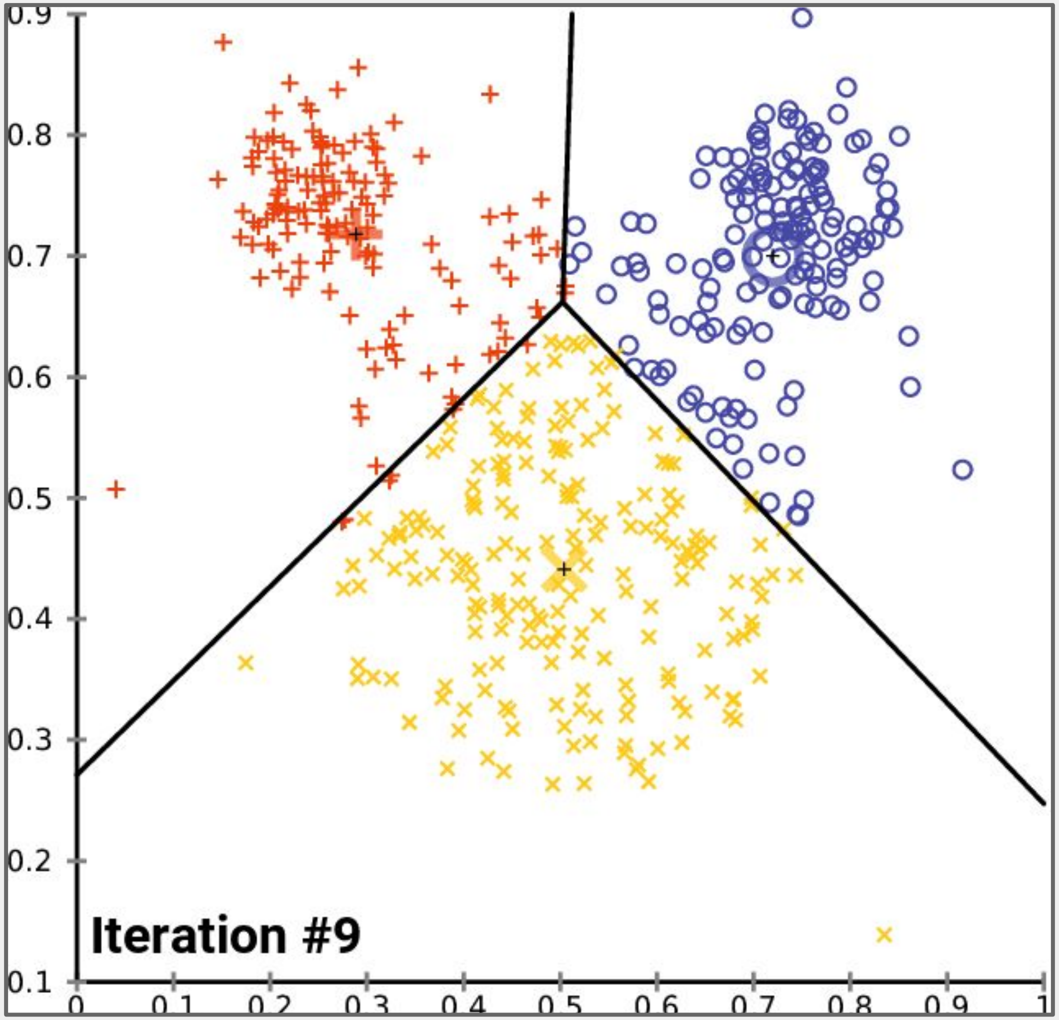
k-means Example 2



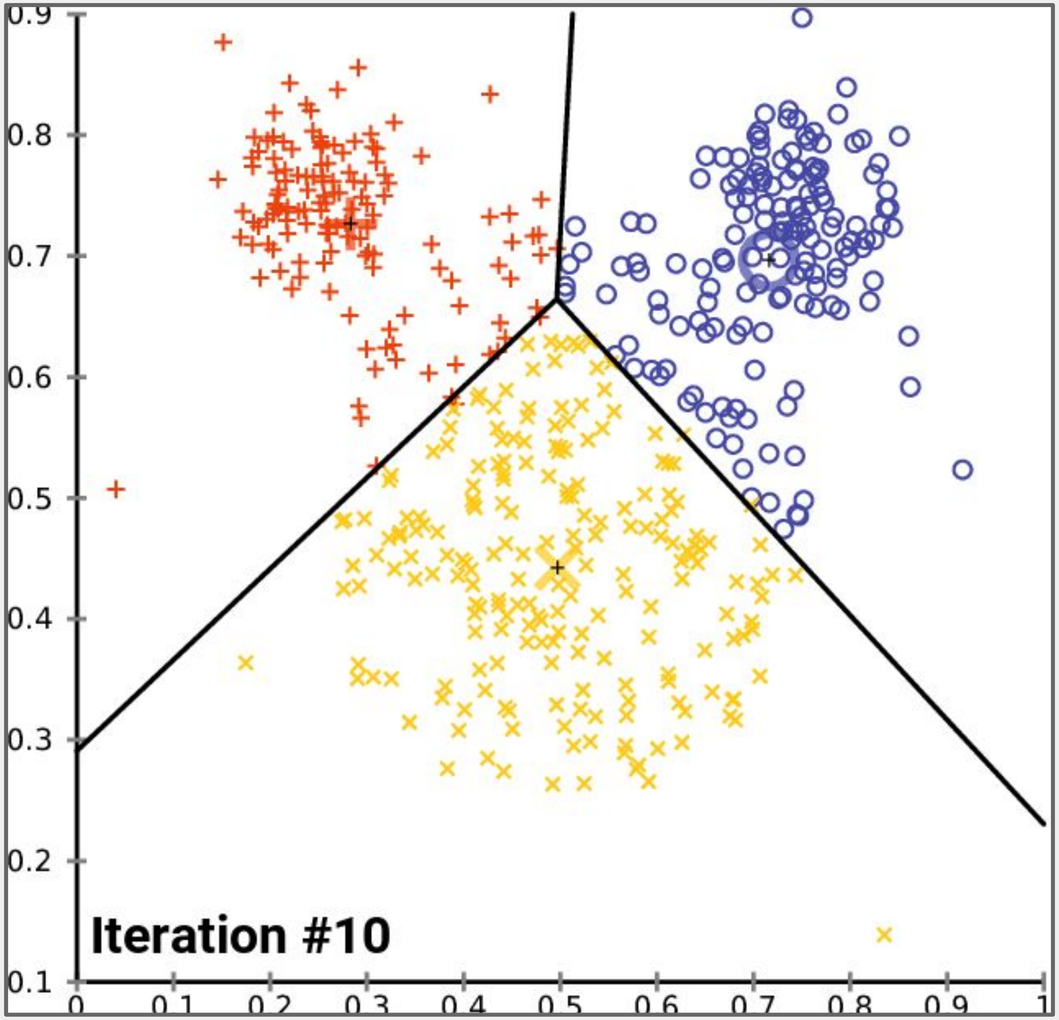
k-means Example 2



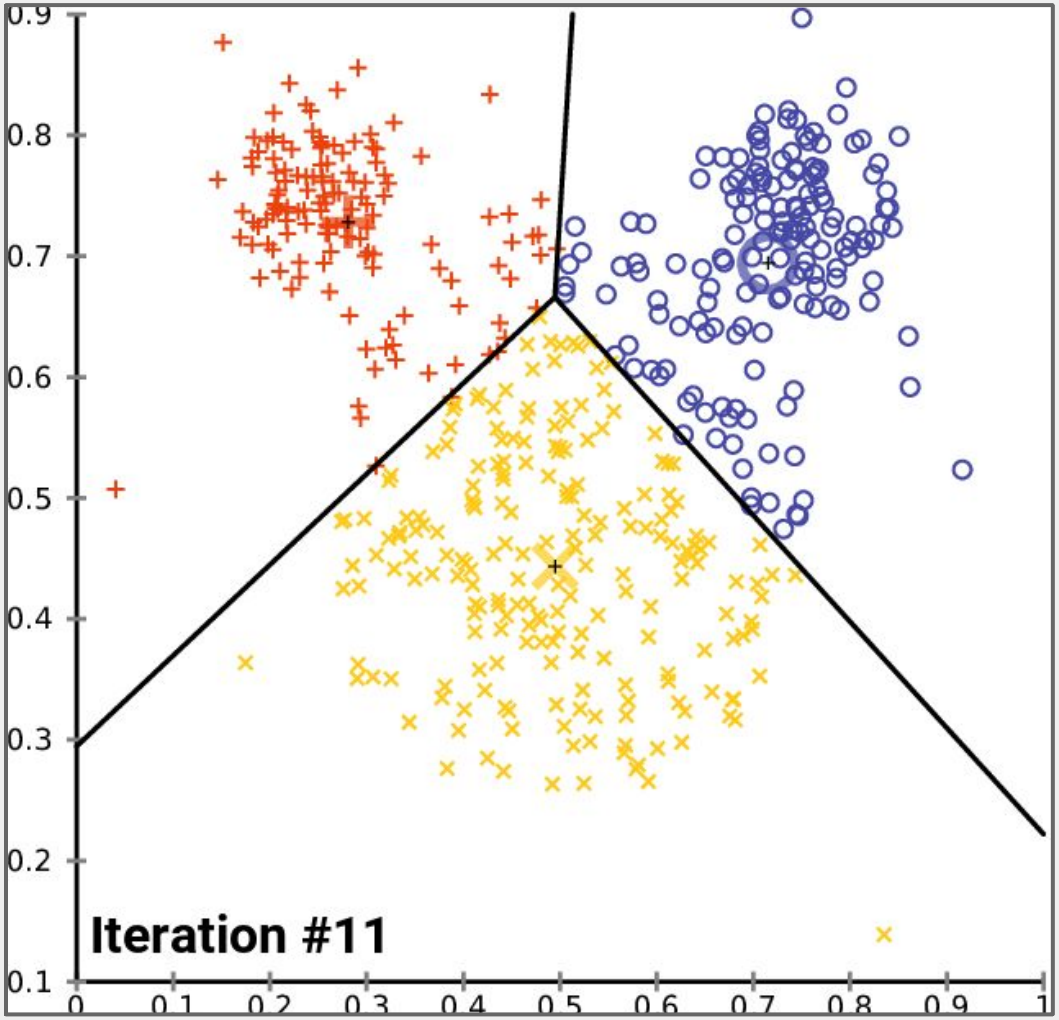
k-means Example 2



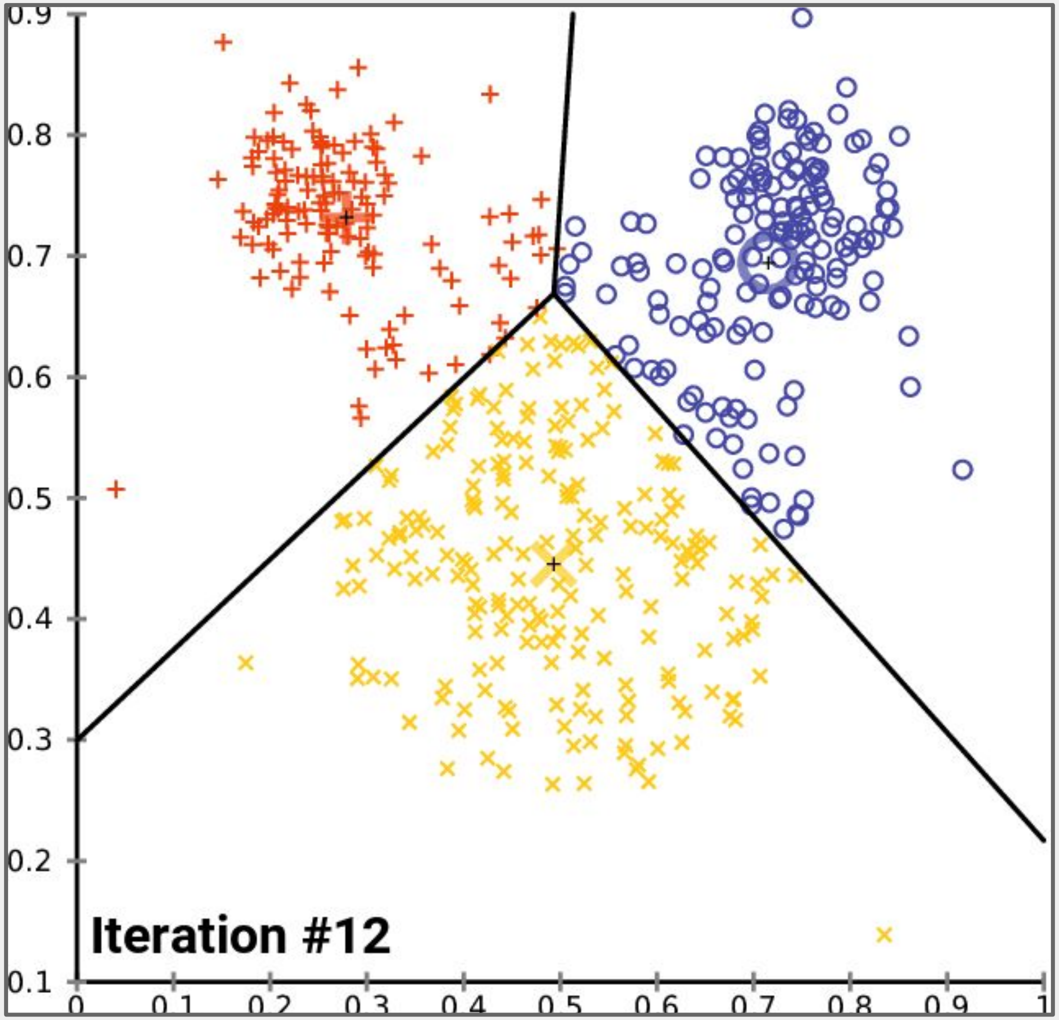
k-means Example 2



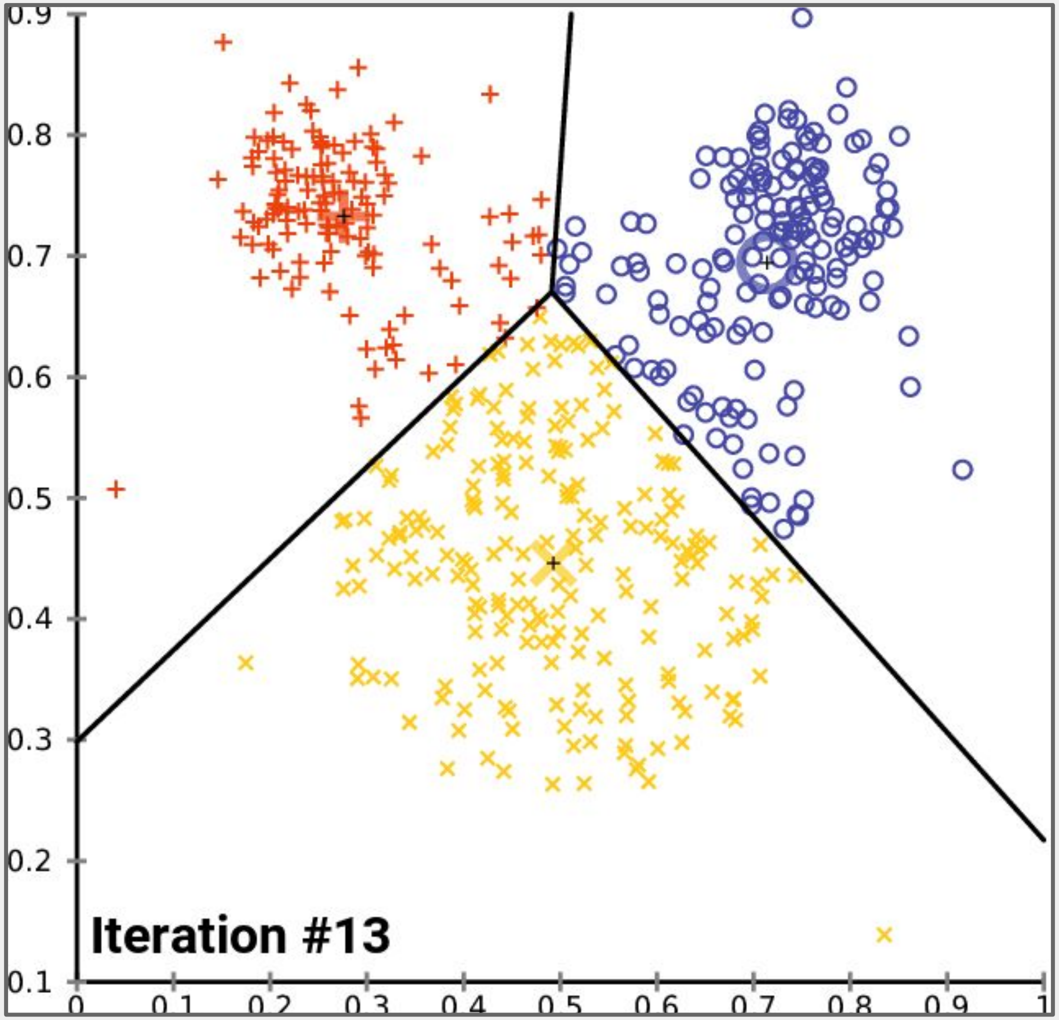
k-means Example 2



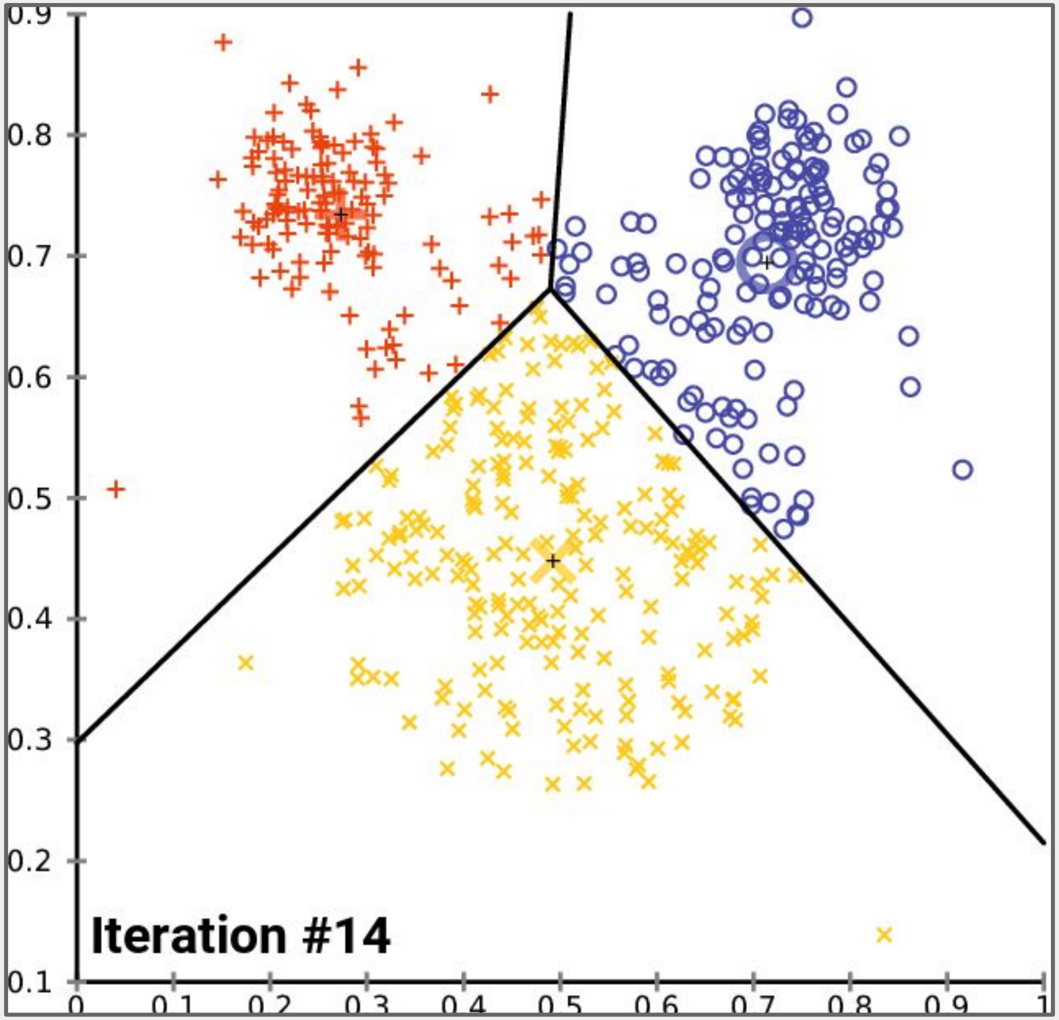
k-means Example 2



k-means Example 2



k-means Example 2

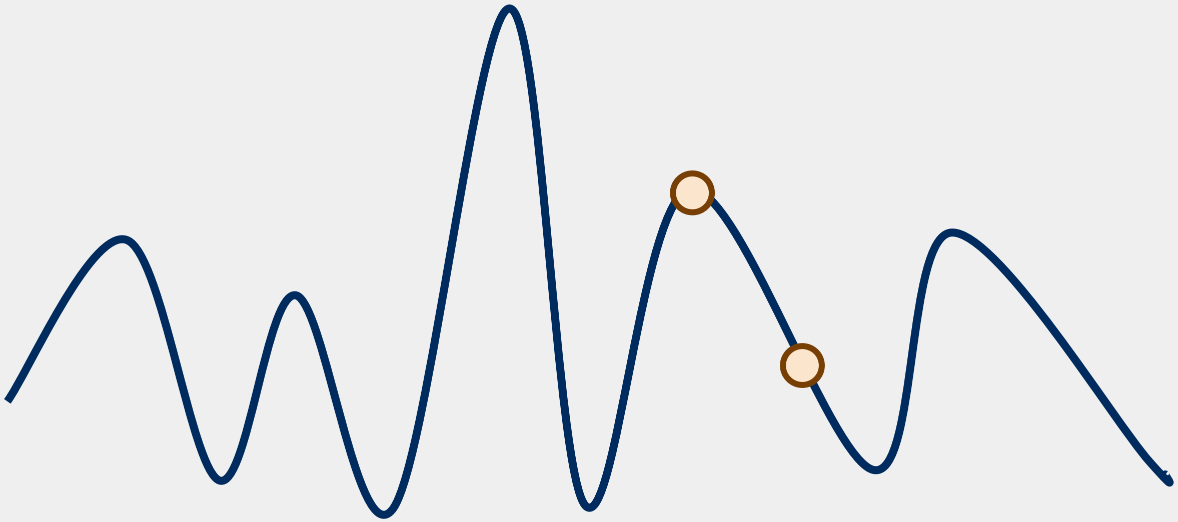


So…

- Runtime?
- To assign to clusters…
  - For each of the *n* examples
    - Compute the distance to each of the *k* centroids
      - Which will take O(*d*) time
- …
- And we need to do that each iteration!
  - How many iterations will we need??

Wait…

- Does this even solve the problem?
  - Nope!
  - We could end up stuck in a *local optimum*
    - Optimal only within neighboring possible solutions
    - Not *globally optimal* across all possible solutions



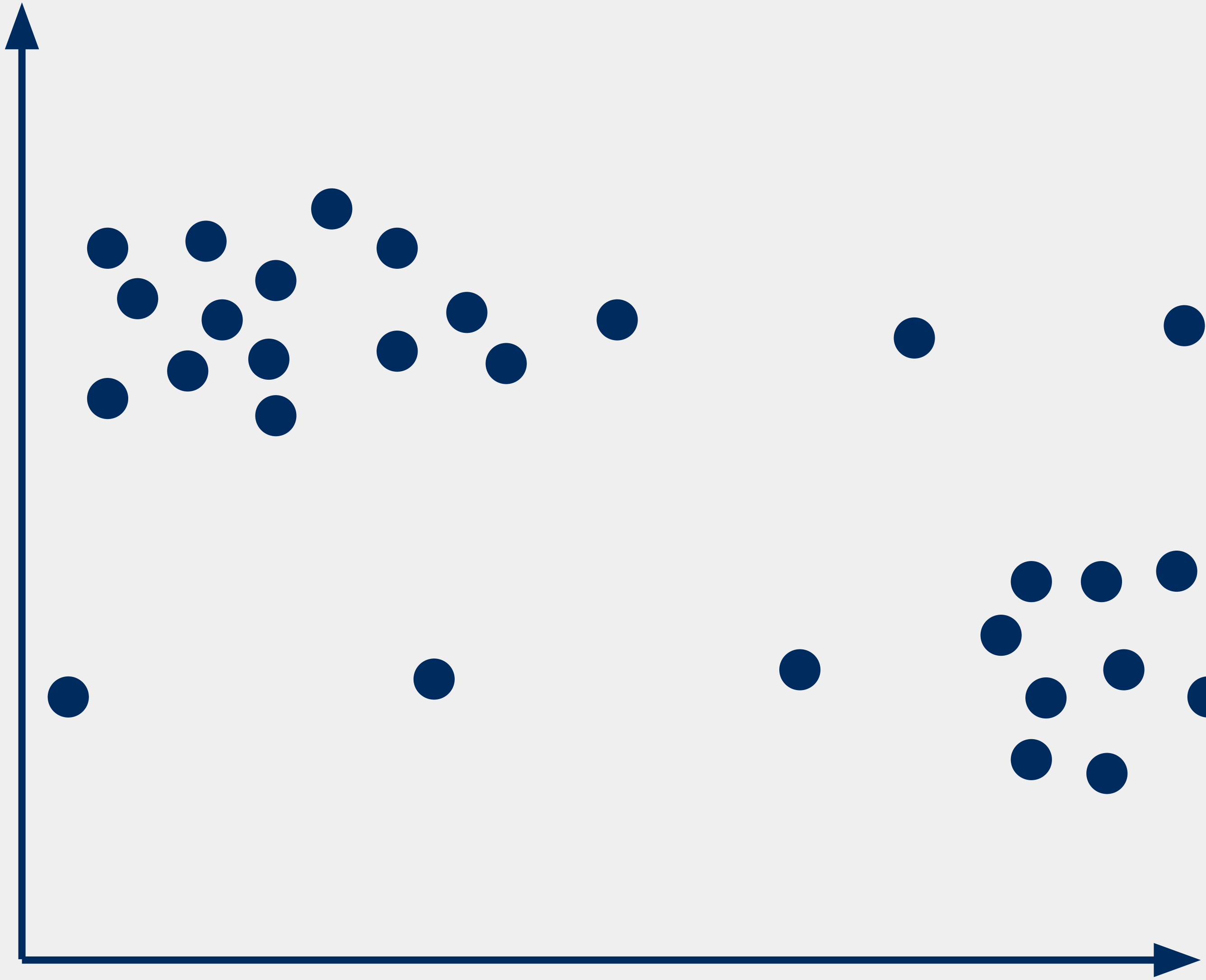
k-means clustering is NP-Hard

- What does NP-Hard mean?
  - Informally: At least as computationally expensive to solve as the most computationally expensive problems in NP
  - Even more informally: Probably takes way too long to run
- So what can we do, on average, get better clustering solutions more quickly?

k-means++

- Goal: spread out the initial clusters as much as possible
- For the *initial assignment*:
  - Pick the first initial centroid uniformly at random
  - The for the next *k*-1 centroids:
    - For each example not already picked:
      - Find the distance to its closest centroid
    - Select the next centroid at random but weighted by those computed distances
      - Further distance means more likely to be selected
- Then proceed with Lloyds as before

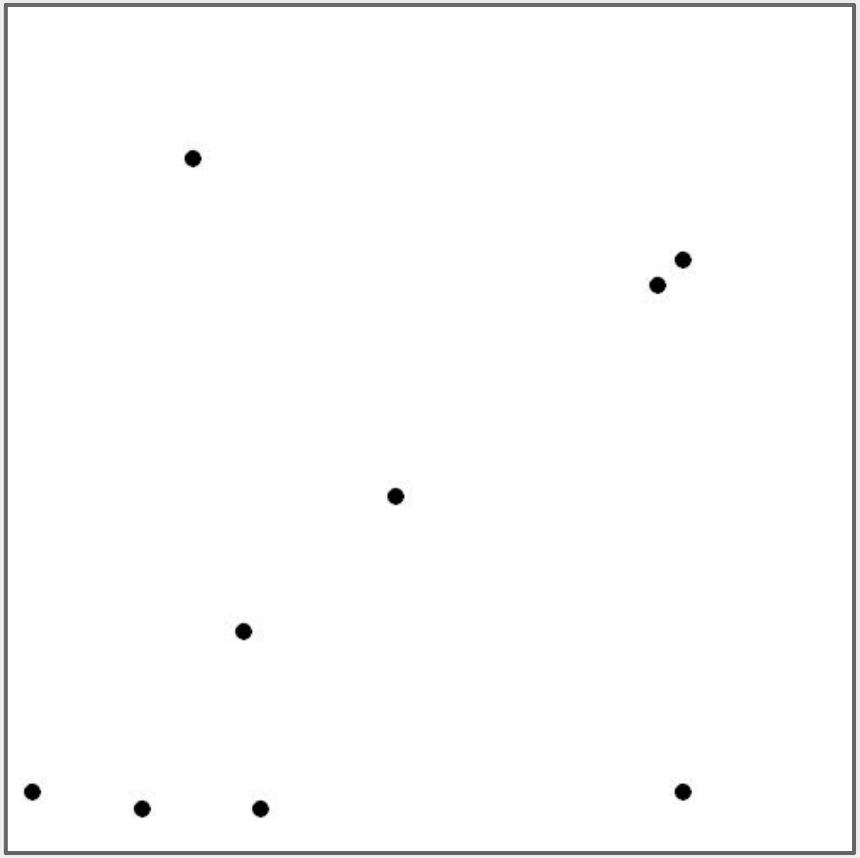
Is k-means always the clustering we want?



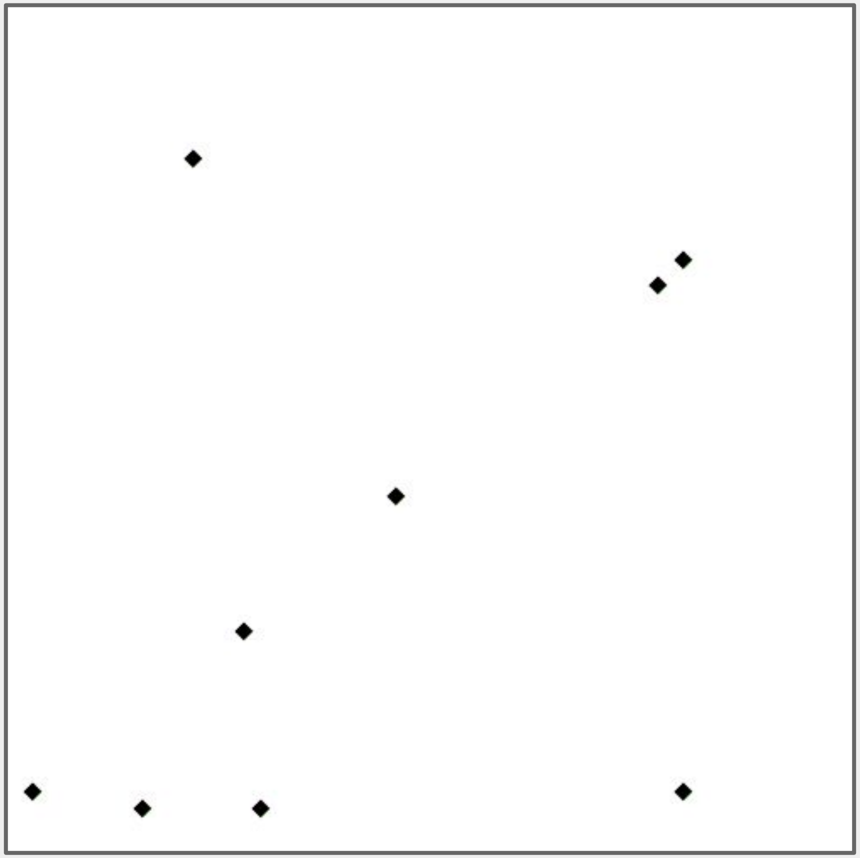
Voronoi diagrams

- Partition a plane to group all points closest to each of a given set of objects together

Euclidean distance Voronoi diagram



Manhattan distance Voronoi diagram



There's alot more to clustering

- Many more approaches to tackle k-means
- Can we do anything to try and figure out what k should be set to?
- And many other clustering algorithms out there!