Clustering: K-means and Nearest Neighbors

Foundations of Data Analysis

February 17, 2022

Clustering Example

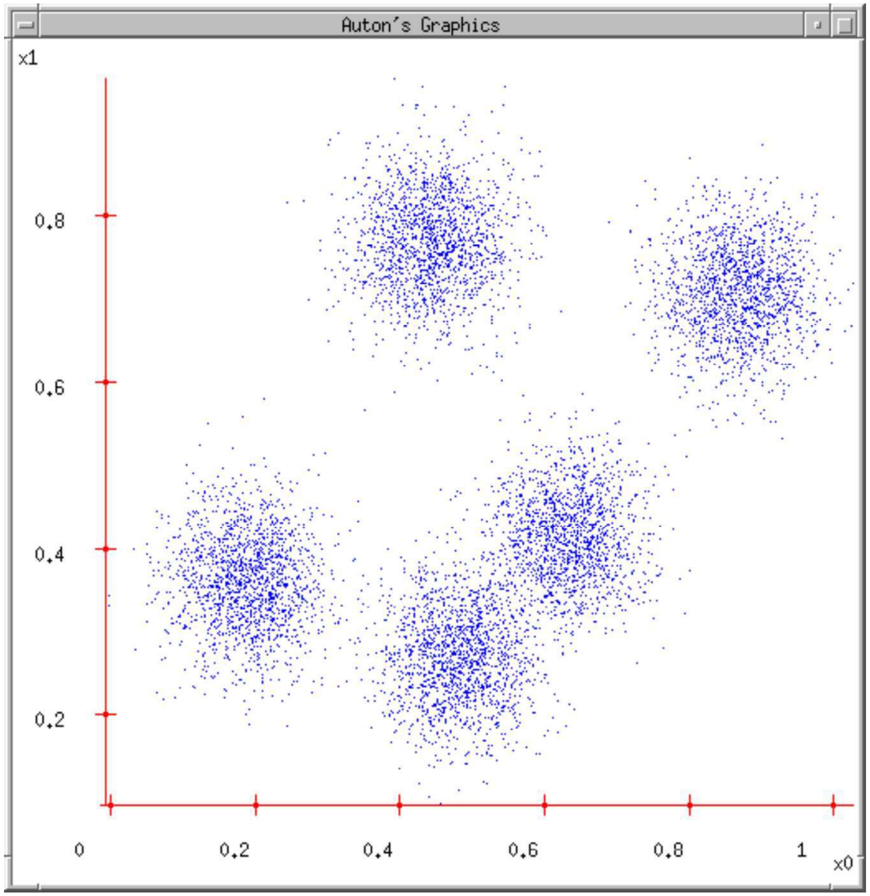
Original image

Segmented image

Divide data into different groups

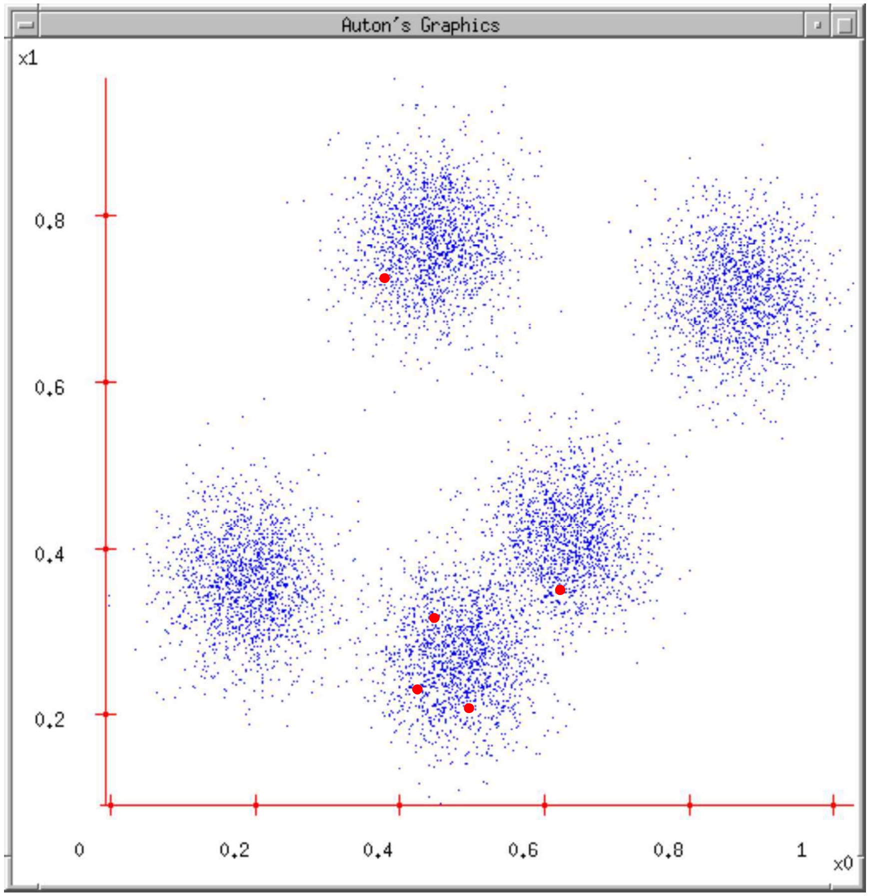
Clustering: K-means

1. Ask user how many clusters they'd like (e.g. k=5)



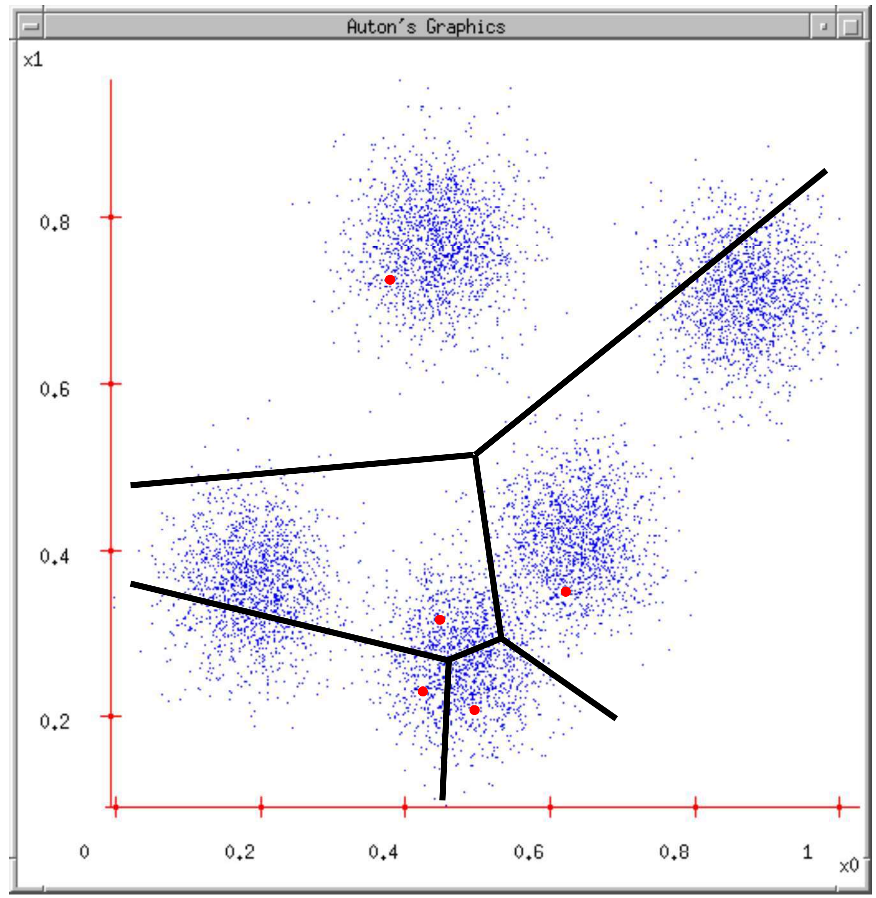
Clustering: K-means

1. Ask user how many clusters they'd like (e.g. k=5)
2. Randomly guess k cluster Center locations



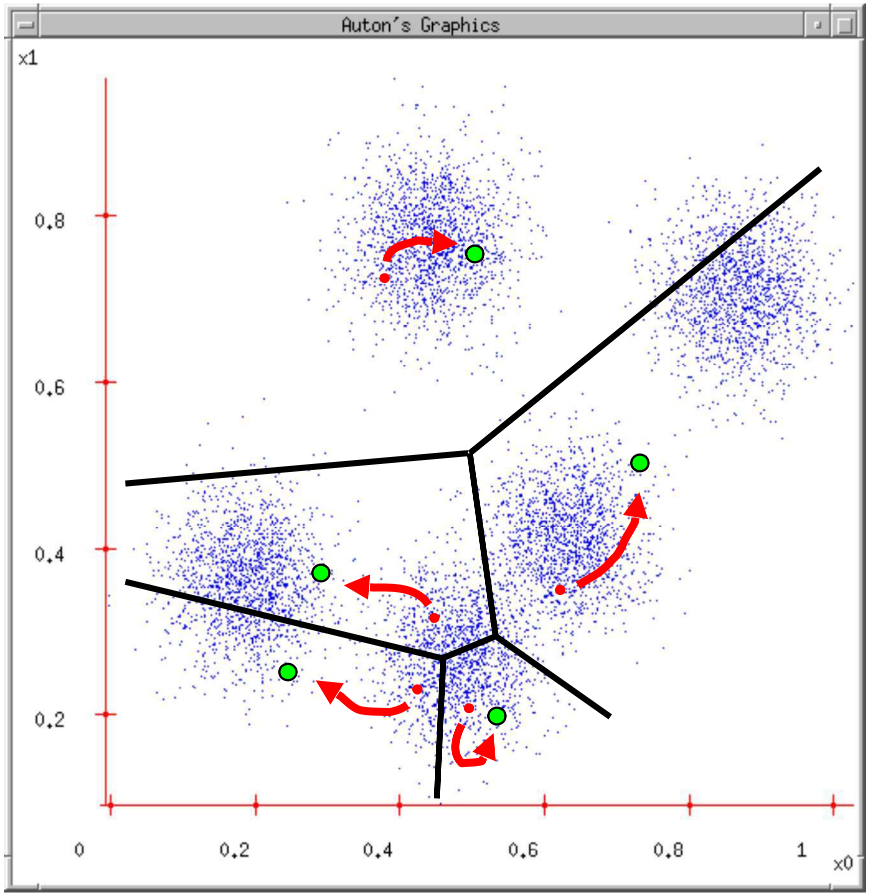
Clustering: K-means

1. Ask user how many clusters they'd like (e.g. k=5)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.



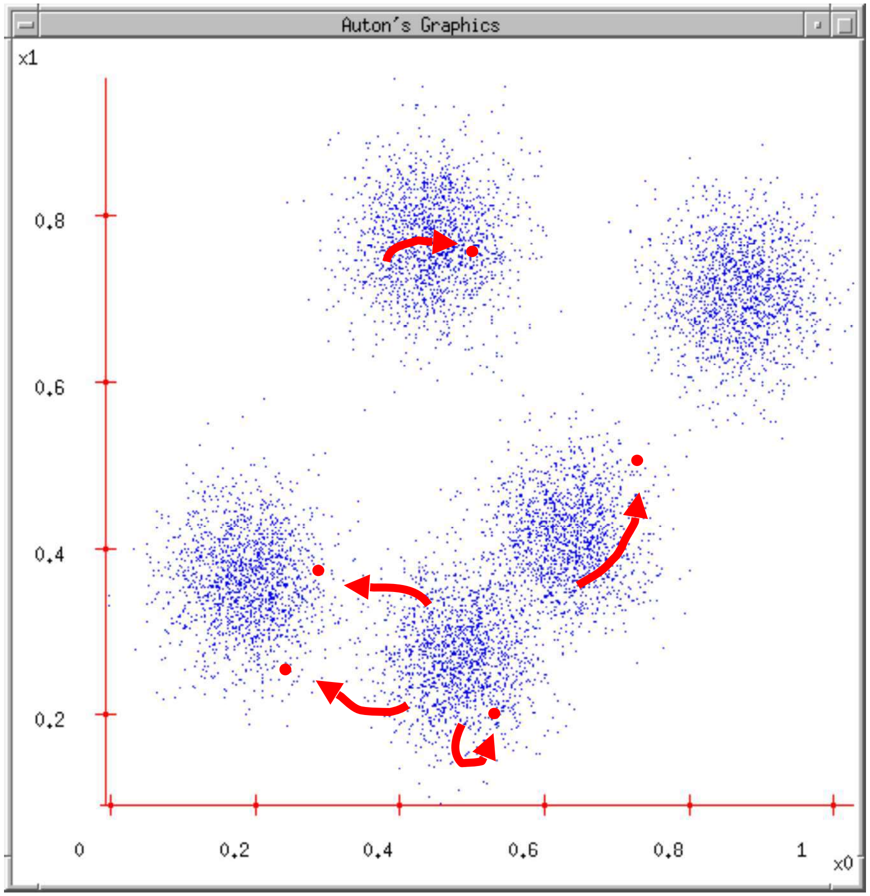
Clustering: K-means

1. Ask user how many clusters they'd like (e.g. k=5)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center re-finds the centroid of the points it owns…



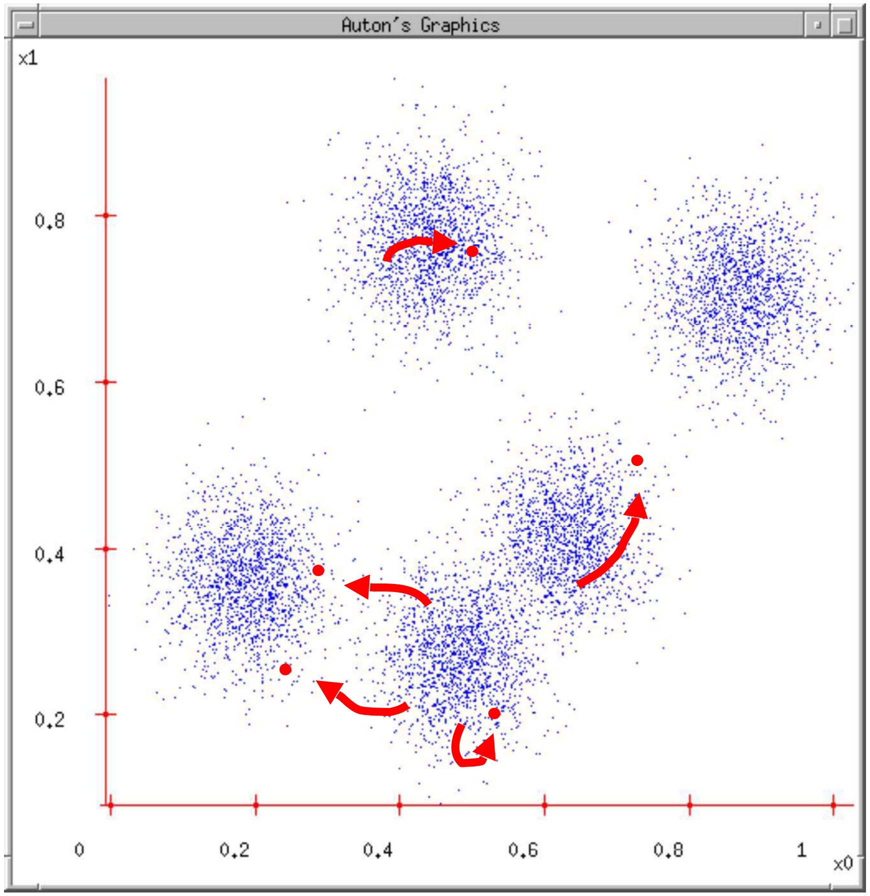
Clustering: K-means

1. Ask user how many clusters they'd like (e.g. k=5)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center re-finds the centroid of the points it owns…
5. …and jumps there



Clustering: K-means

1. Ask user how many clusters they'd like (e.g. k=5)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center re-finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat steps 3-5 until terminated!



Disadvantages of K-means

Disadvantages of K-means

- Does not work efficiently with complex structured data (mostly non-linear)

Disadvantages of K-means

- Does not work efficiently with complex structured data (mostly non-linear)

- Hard assignment for labels might lead to misgrouping

Disadvantages of K-means

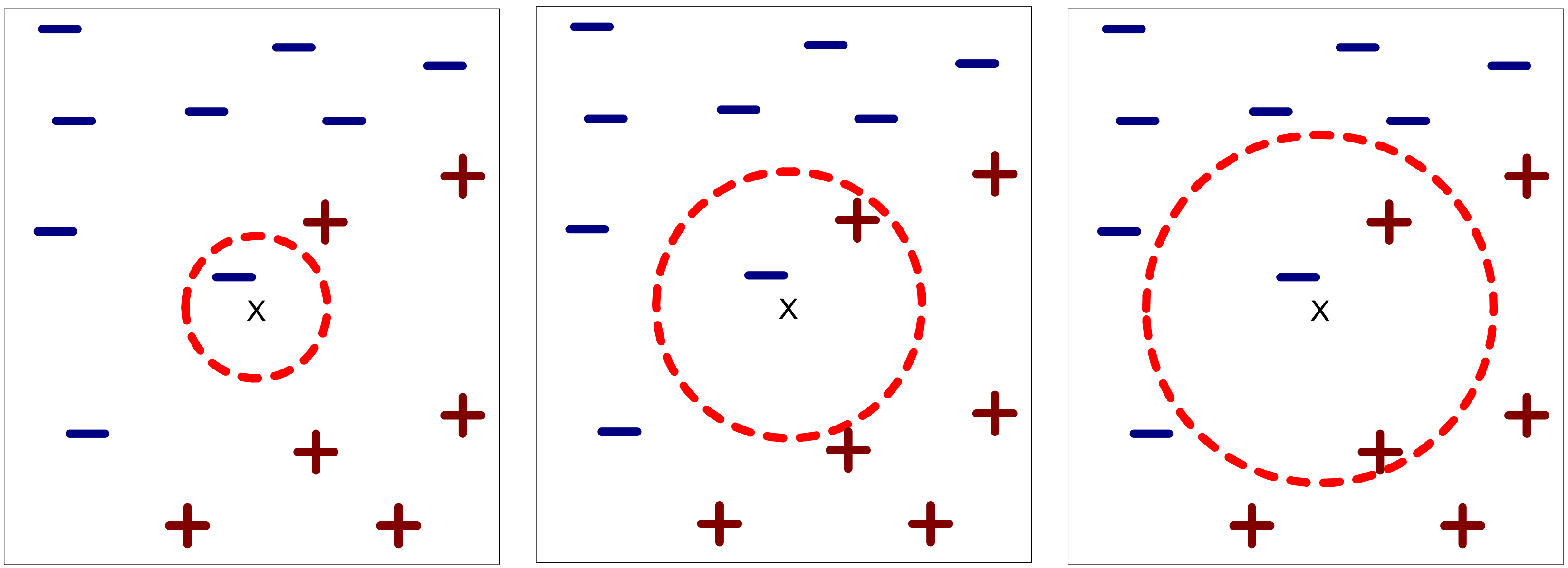
- Does not work efficiently with complex structured data (mostly non-linear)

- Hard assignment for labels might lead to misgrouping

- Random guess for initialization might be a hassle

- Nearest Neighbors: (Un)supervised Learning (non-parametric model)

Nearest Neighbors

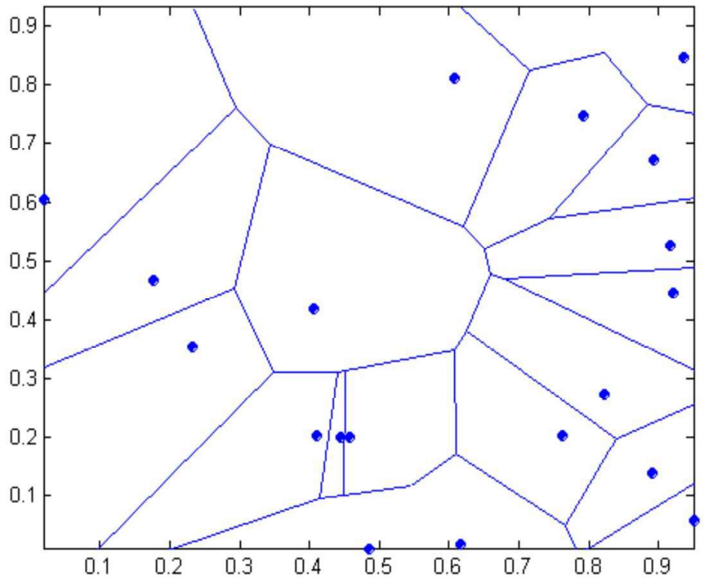


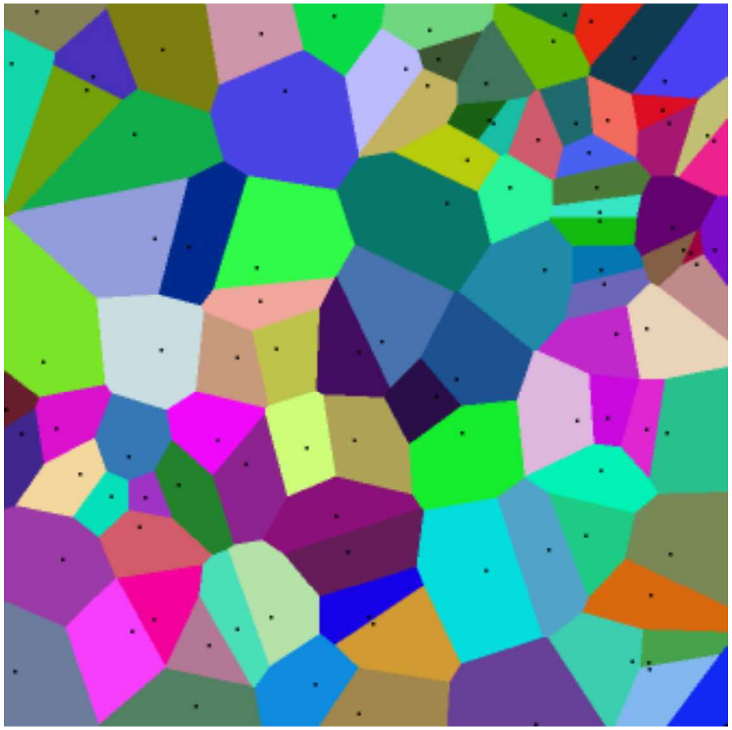
(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of seed x: data points that have the k smallest distance to x.

Nearest Neighbor

Voronoi Diagram

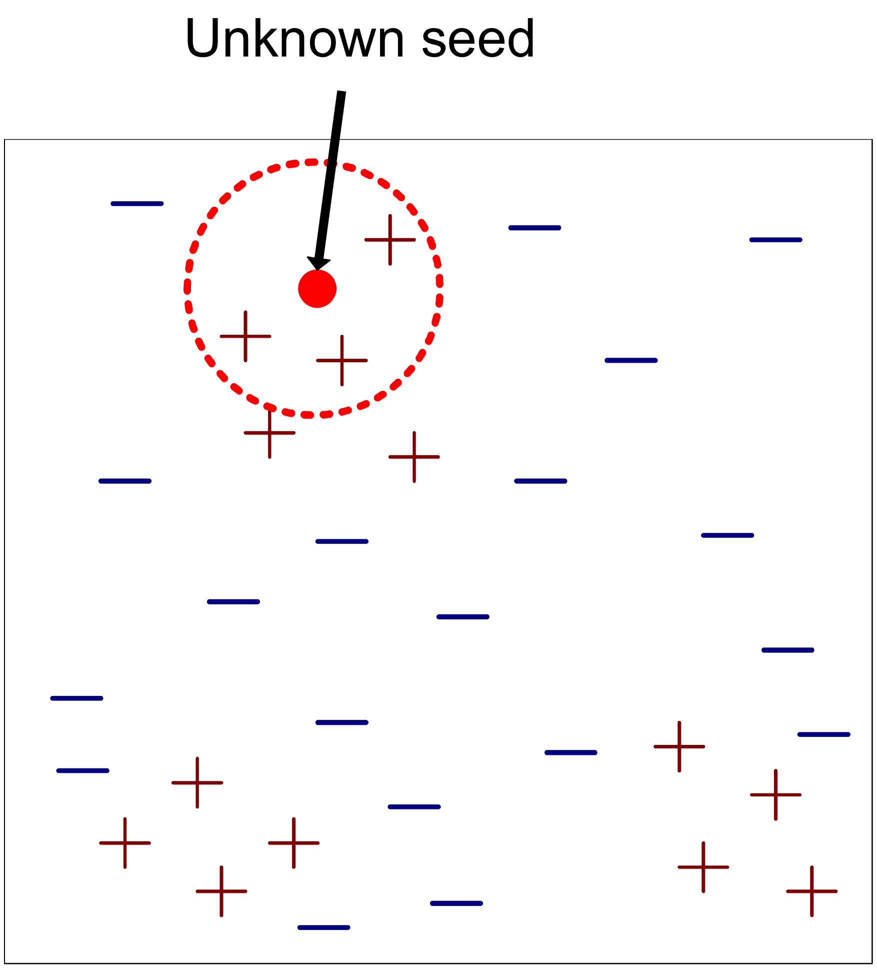




- Partitions space into regions
- boundary: points at the same distance from two different training examples

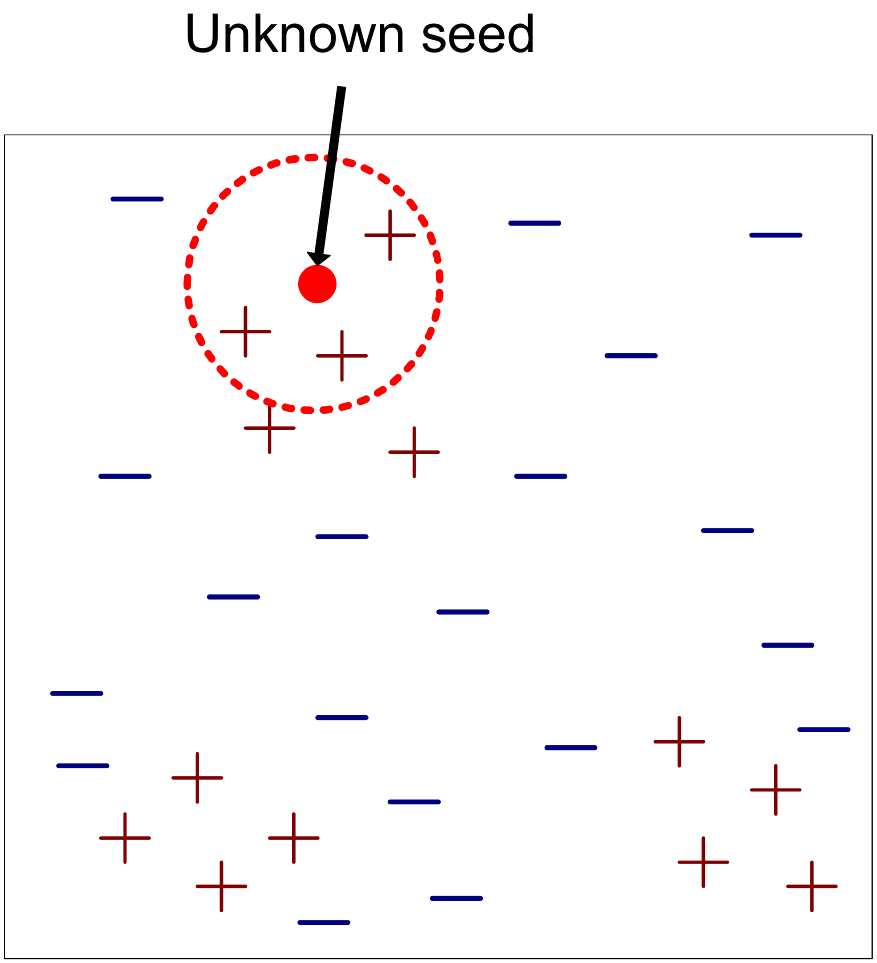
- K-Nearest Neighbor (KNN) classification - supervised learning

KNN Classifiers



- Requires three things
  - The set of stored records
  - Distance metric
  - The value of *k*, the number of nearest neighbors to retrieve

KNN Classifiers



- Requires three things
  - The set of stored records
  - Distance metric
  - The value of *k*, the number of nearest neighbors to retrieve

- To classify an unknown seed:
  - Compute distance to other training seeds
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown seed (e.g., by taking majority vote)

Nearest Neighbor Classification

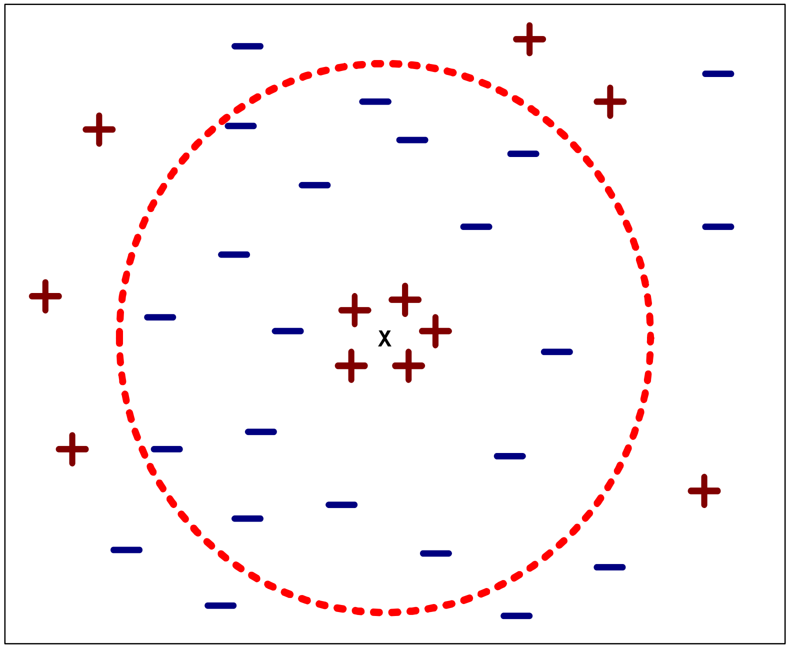
- Compute distance between two points:
  - Euclidean distance (L2 norm)

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from nearest neighbor list
  - take the majority vote of class labels among the k-nearest neighbors
  - Weight the vote according to distance
    - weight factor, $w = 1/d^2$

Nearest Neighbor Classification…

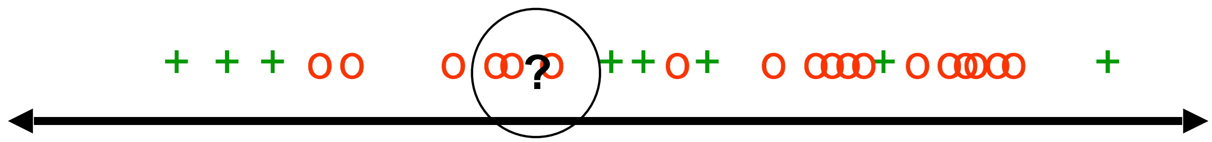
- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes



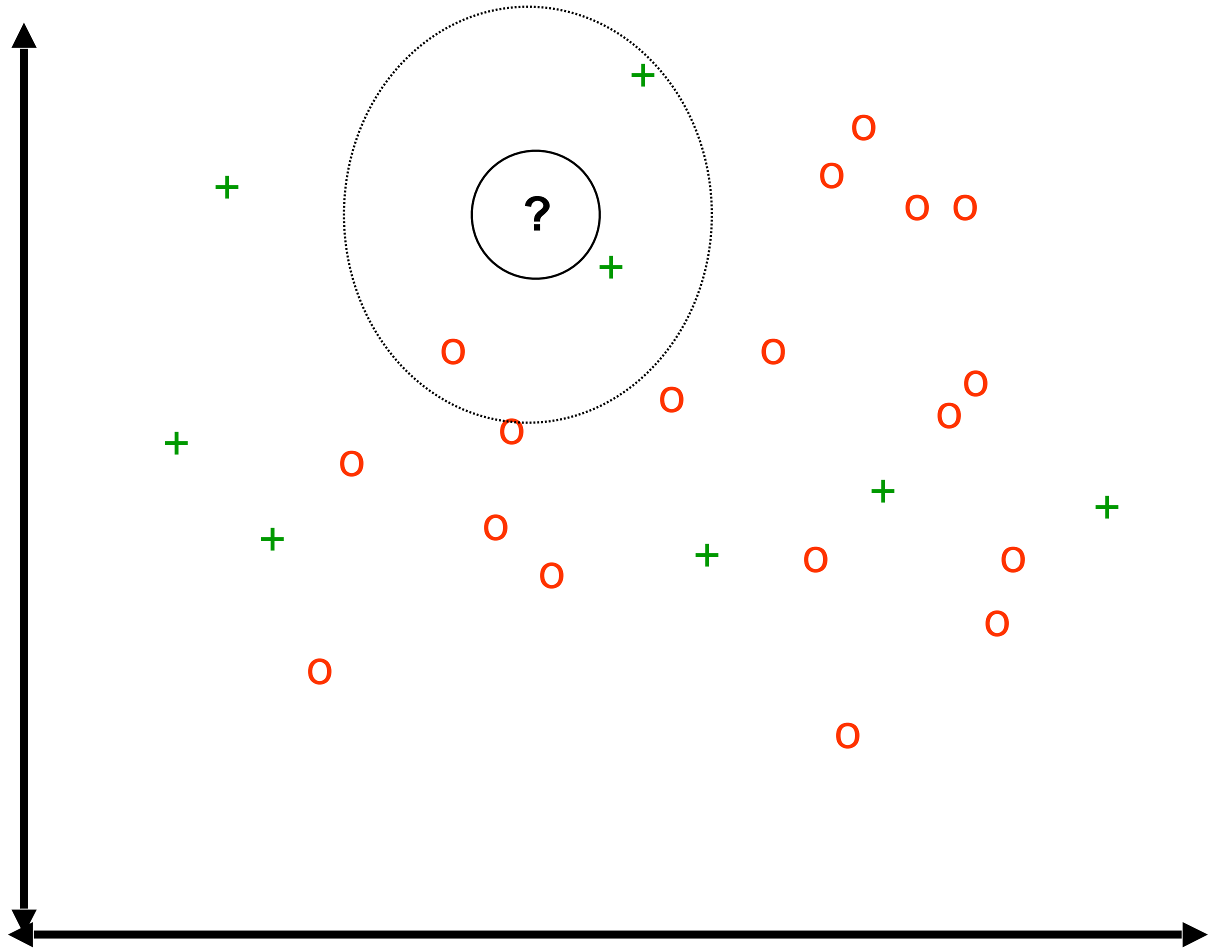
Issues of Nearest Neighbor Classification

- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5m to 1.8m
    - weight of a person may vary from 90lb to 300lb
    - income of a person may vary from \$10K to \$1M

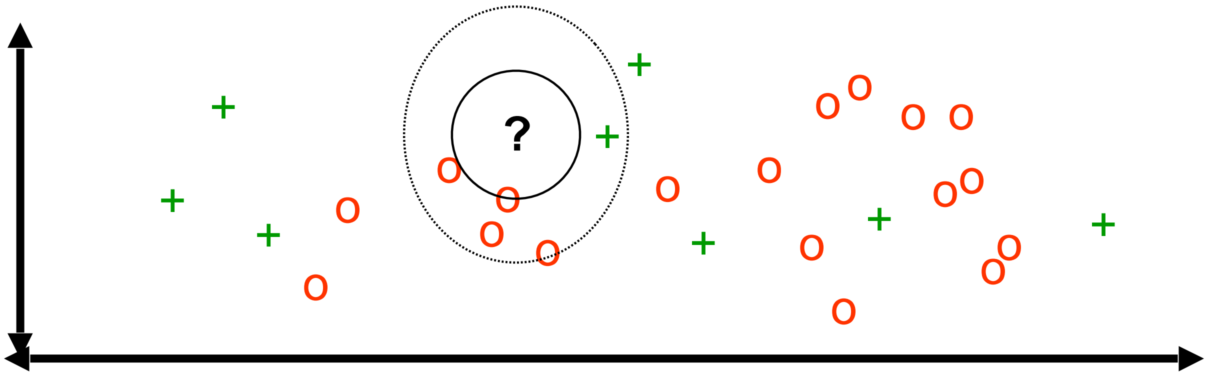
K-NN and Irrelevant Features



K-NN and Irrelevant Features



K-NN and Irrelevant Features



Issues of Nearest Neighbor Classification

- Problem with Euclidean measure:
  - High dimensional data
    - curse of dimensionality
  - Can produce counter-intuitive results

1 1 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1

vs

1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1

d = 1.4142

d = 1.4142

Solution: Normalize the vectors to unit length.

K-NN Algorithm

- Training:
  - Save the training examples

- At prediction:
  - Find the *k* training examples $(x_1,y_1),\ldots(x_k,y_k)$ that are closest to the test example x
  - Predict the most frequent class among those y_i's.

K-NN Algorithm

- Training:
  - Save the training examples
- At prediction:
  - Find the *k* training examples $(x_1,y_1),\ldots(x_k,y_k)$ that are closest to the test example x
  - Predict the most frequent class among those y_i's.

- Improvements:
  - Weighting examples from the neighborhood
  - Measuring "closeness"
  - Finding "close" examples in a large training set quickly