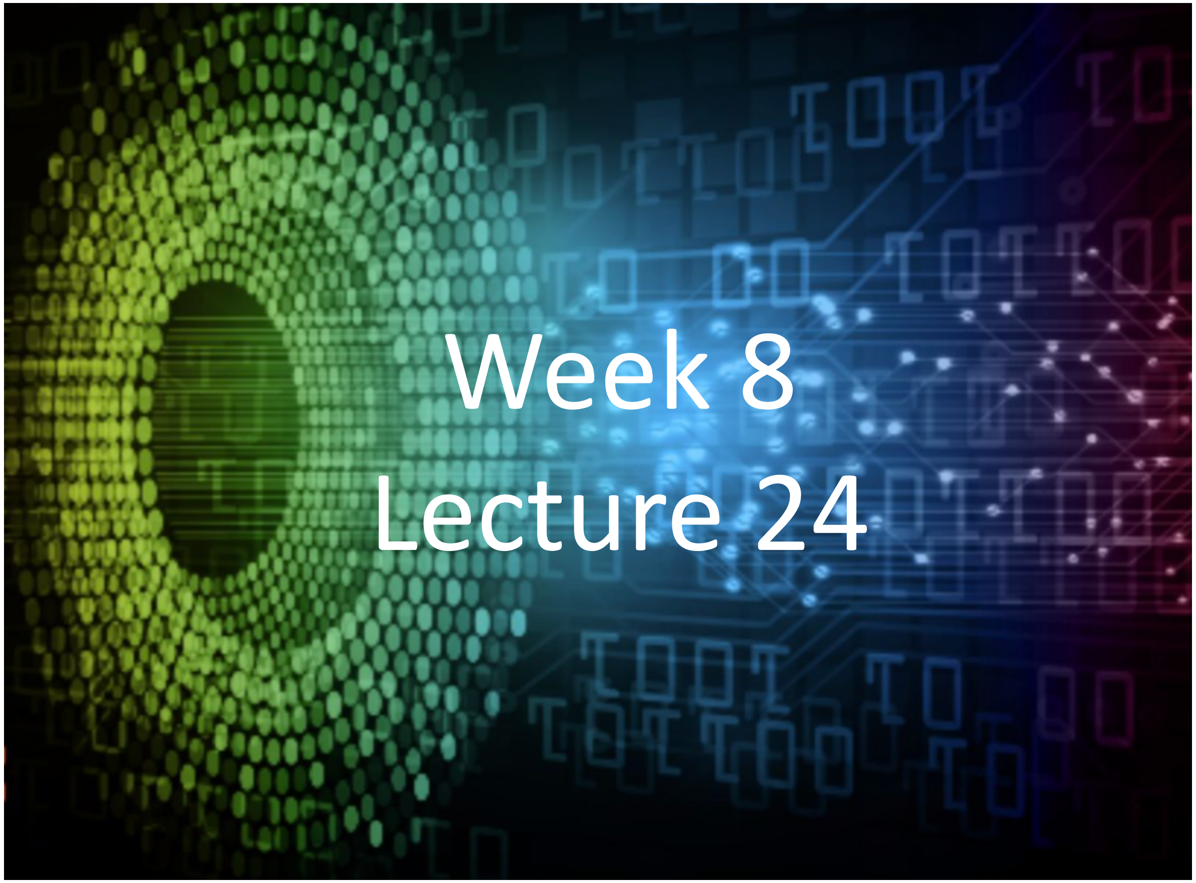
Introduction to Deep Learning Applications and Theory



AMATH 563

Previous Lecture:
Manifold Learning and Representation

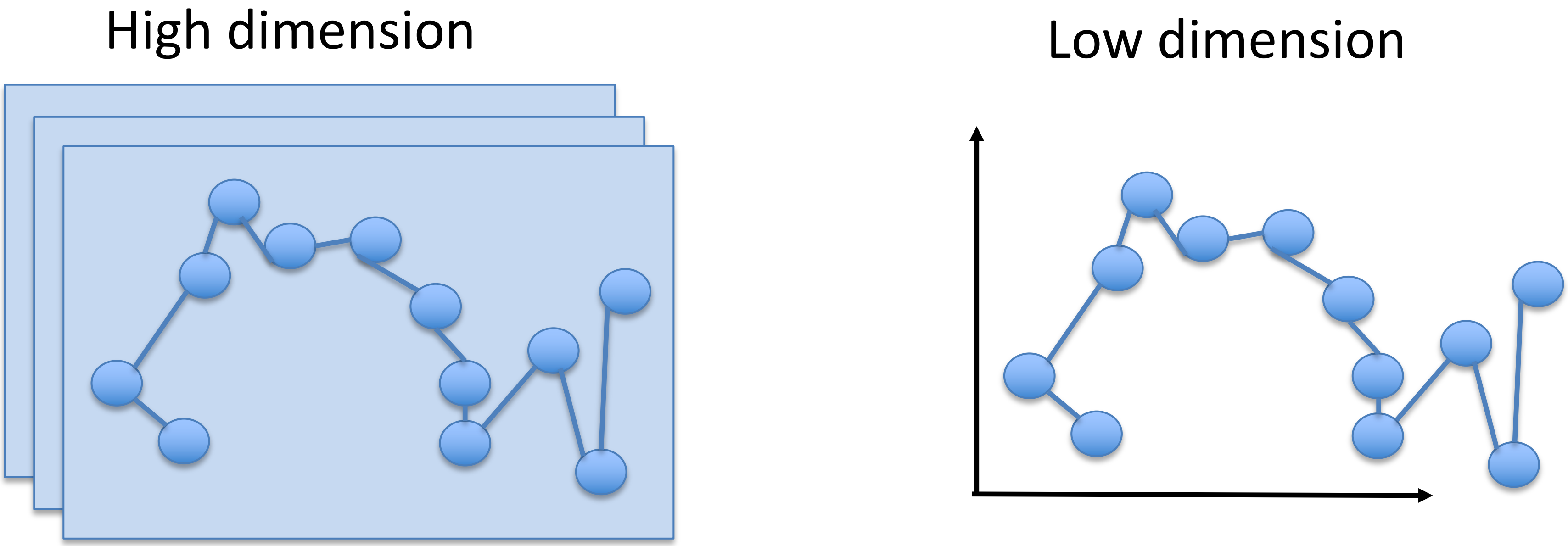
- Multi-Dimensional Scaling (MDS)
- ISOMAP
- t-SNE
- Force Directed graphs

This Lecture:
Embedding and Clustering

- UMap
- Clustering
- K Means
- K Nearest Neighbors (KNN)

UMAP

Uniform **M**anifold **A**pproximation and **P**rojection



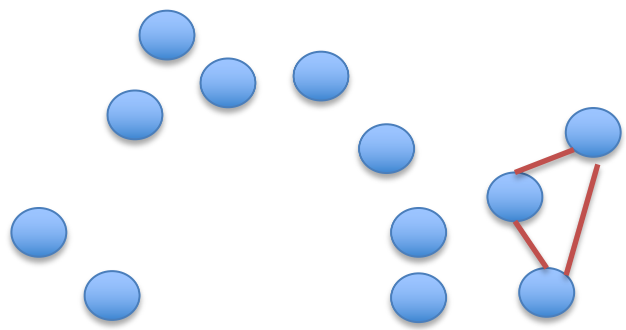
Neighbor Graph

McInnes, Leland, John Healy, and James Melville.
Umap: Uniform manifold approximation and projection for dimension reduction.
arXiv preprint arXiv:1802.03426 (2018).

UMAP

Uniform **M**anifold **A**pproximation and **P**rojection

High dimension

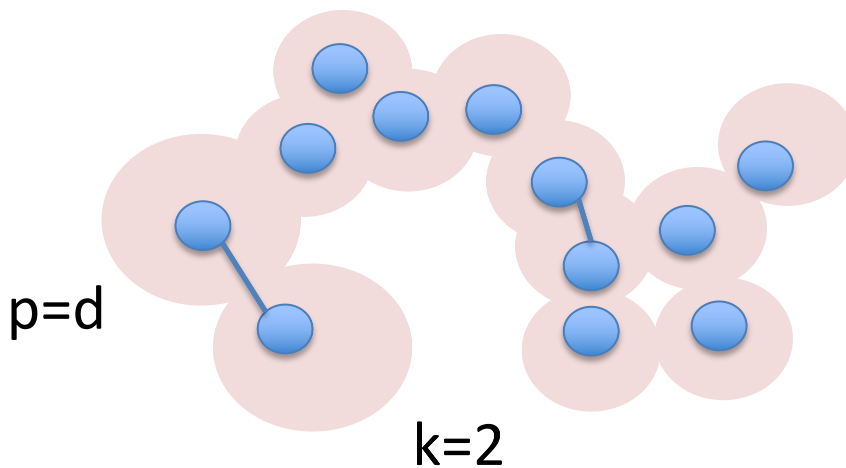


2-simplex

Neighbor Graph

UMAP

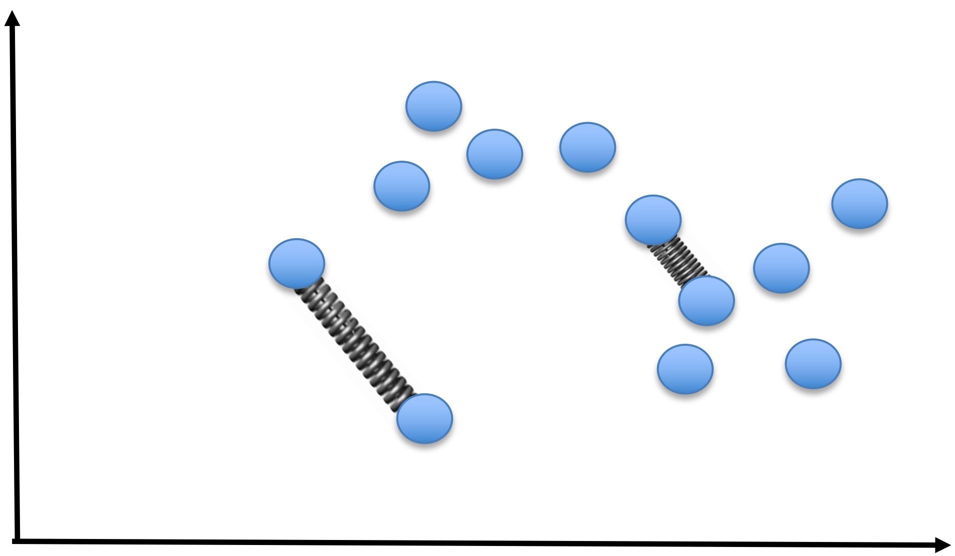
High dimension



Neighbor Graph

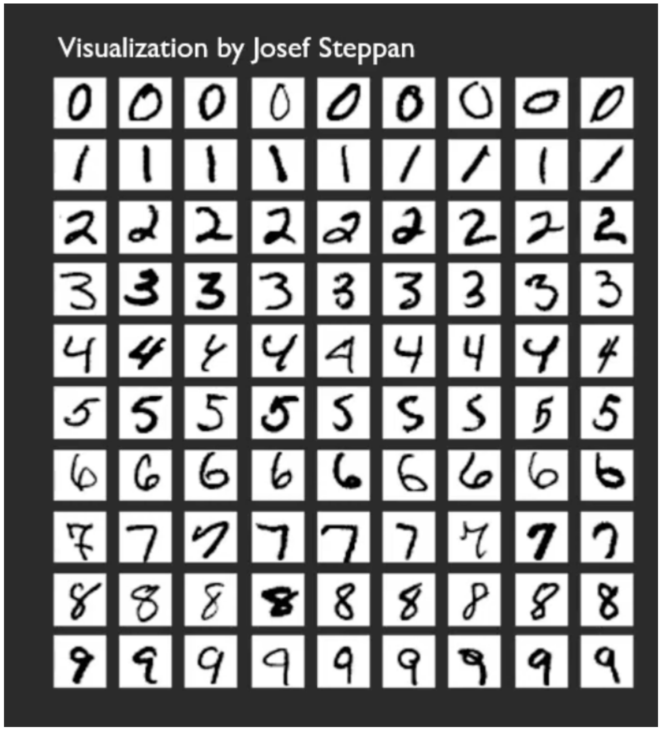
UMAP

Low dimension

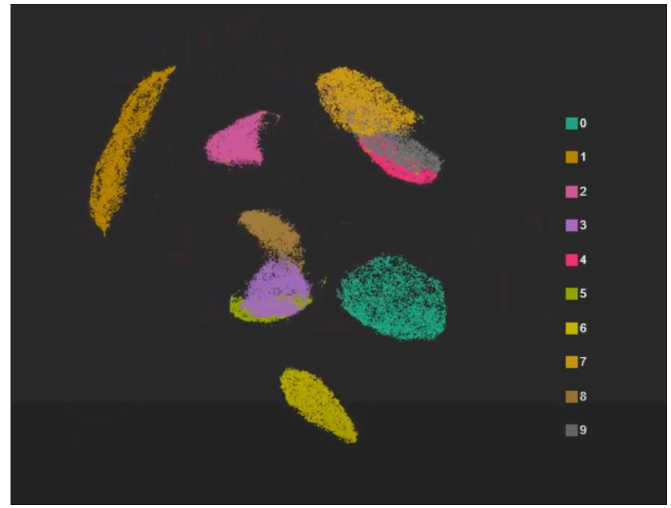


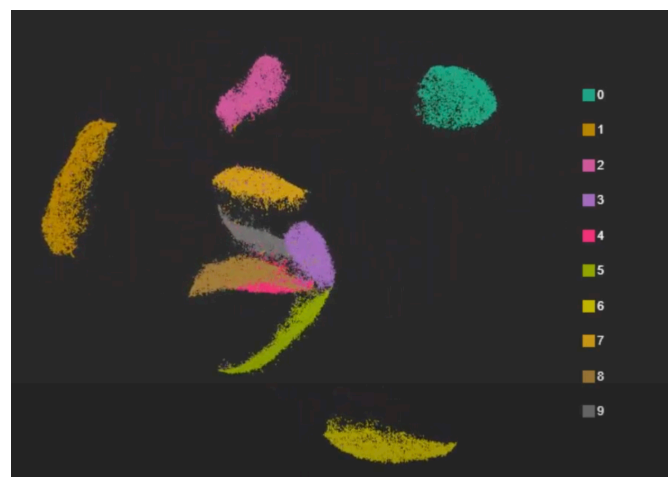
Graph Projection

UMAP



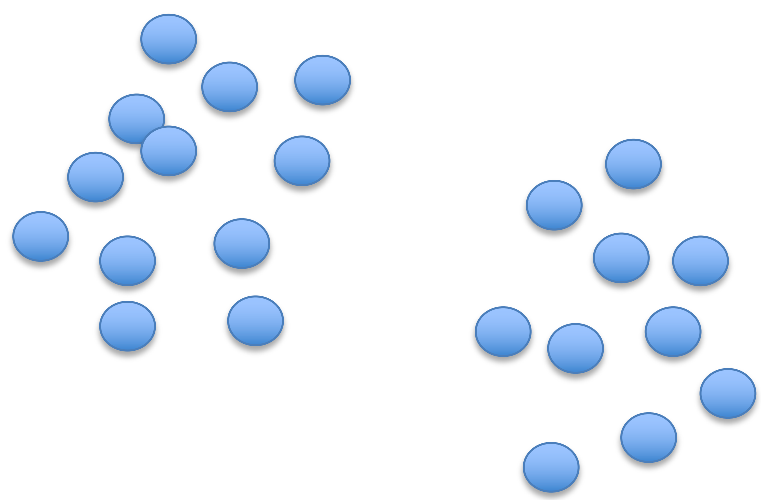
6000*28*28





Youtube: AI Coffee Break

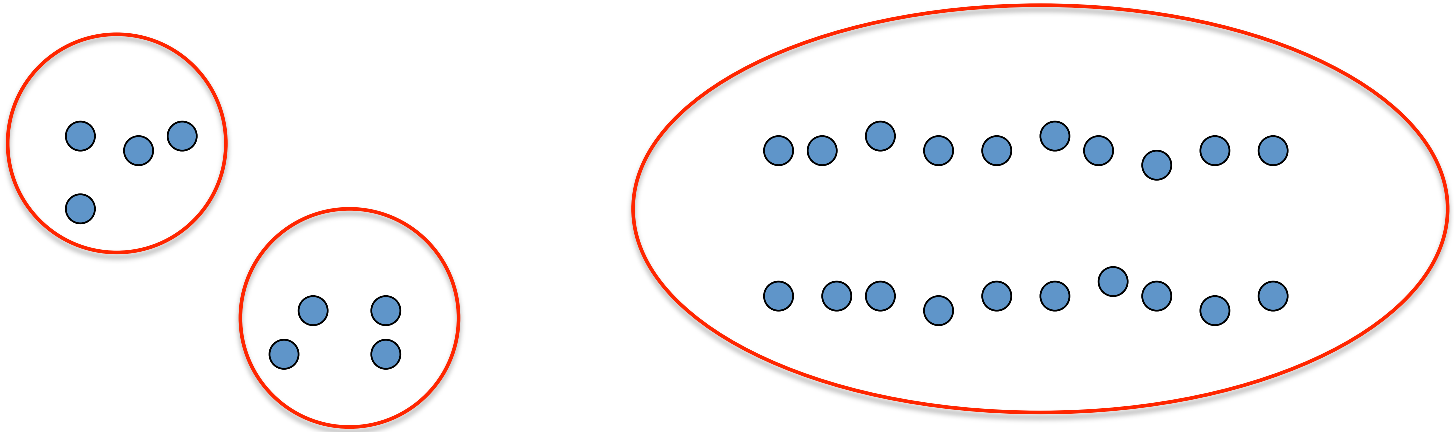
Clustering



Slides adapted from David Sontag

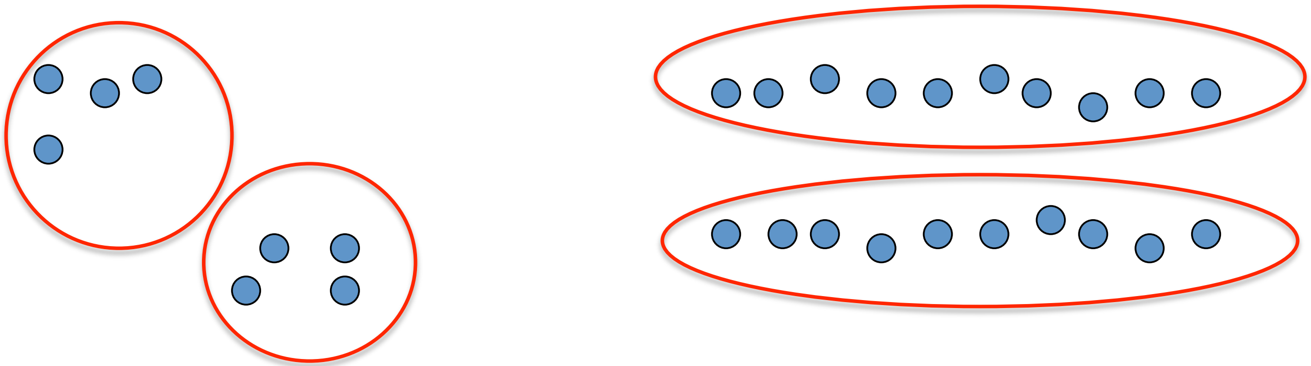
Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns



Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns



- What could "similar" mean?
  - One option: small Euclidean distance (squared)

$$\mathrm{dist}(\vec{x}, \vec{y}) = ||\vec{x} - \vec{y}||_2^2$$

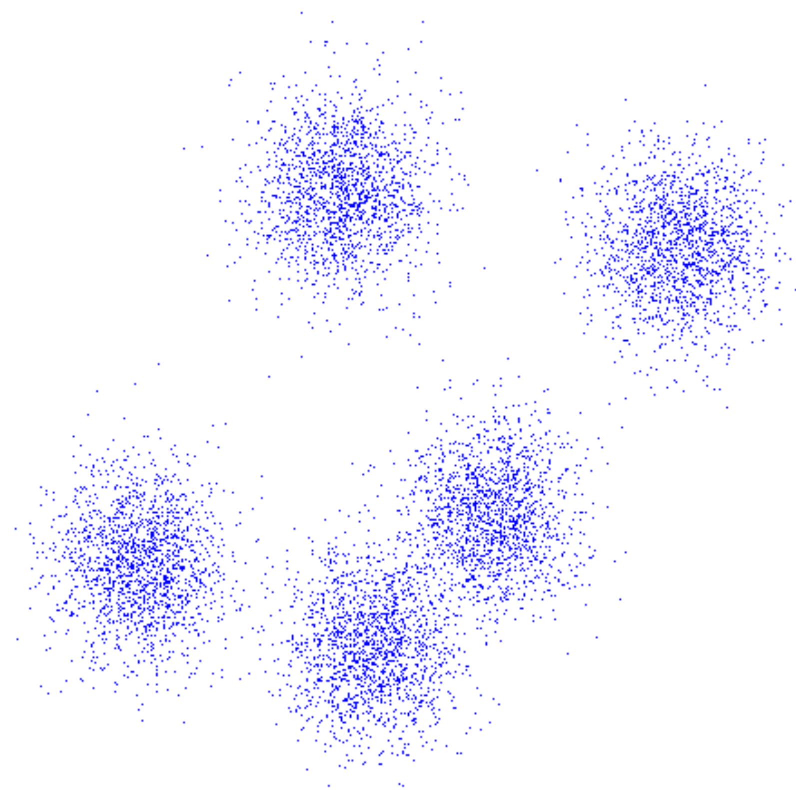
  - Clustering results are crucially dependent on the measure of similarity (or distance) between "points" to be clustered

K-Means

- Unsupervised
- K – number of clusters (centers)
- Euclidean Distance
- Iterative Process

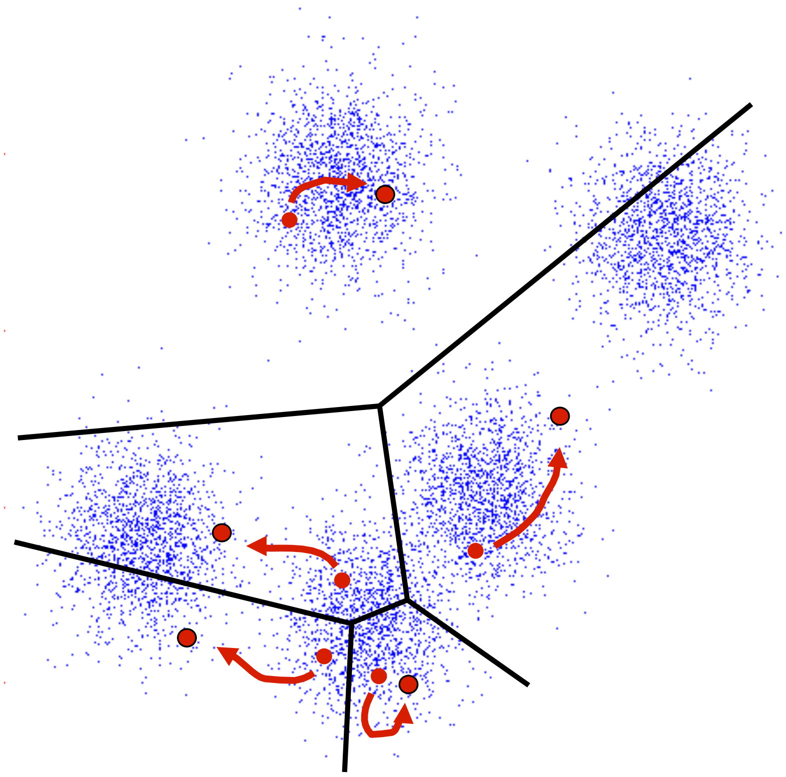
K-Means

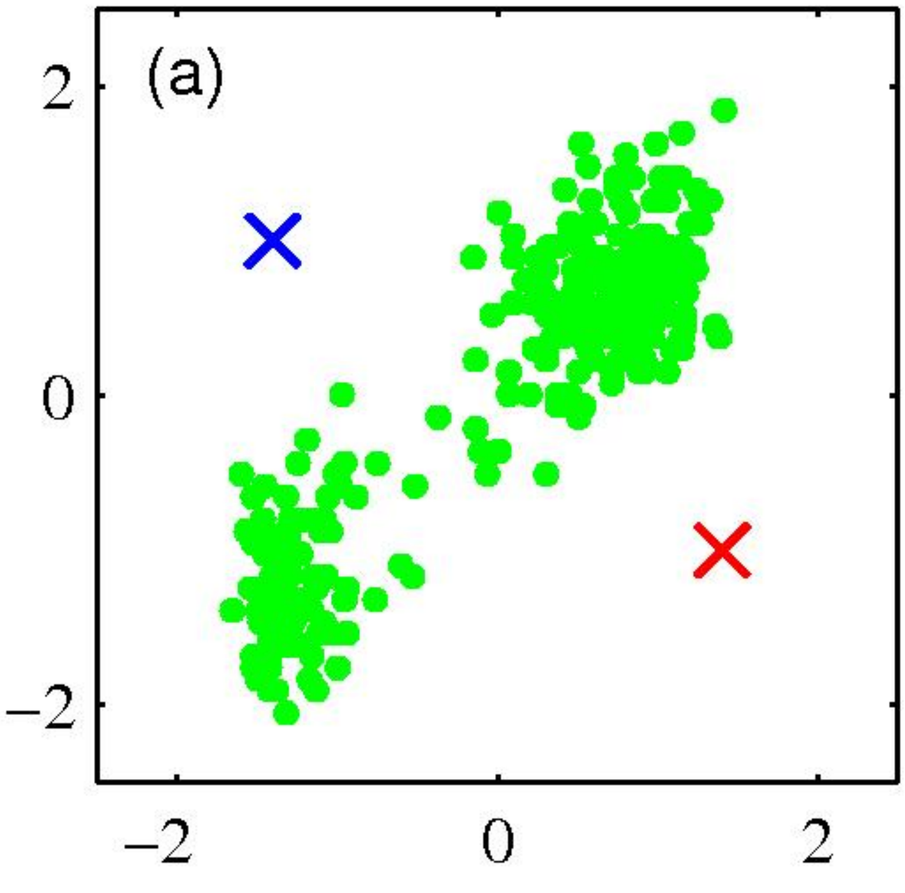
- An iterative clustering algorithm
  - Initialize: Pick *K* random points as cluster centers
  - Alternate:
    1. Assign data points to closest cluster center
    2. Change the cluster center to the average of its assigned points
  - Stop when no points' assignments change



K-Means

- An iterative clustering algorithm
  - Initialize: Pick *K* random points as cluster centers
  - Alternate:
    1. Assign data points to closest cluster center
    2. Change the cluster center to the average of its assigned points
  - Stop when no points' assignments change

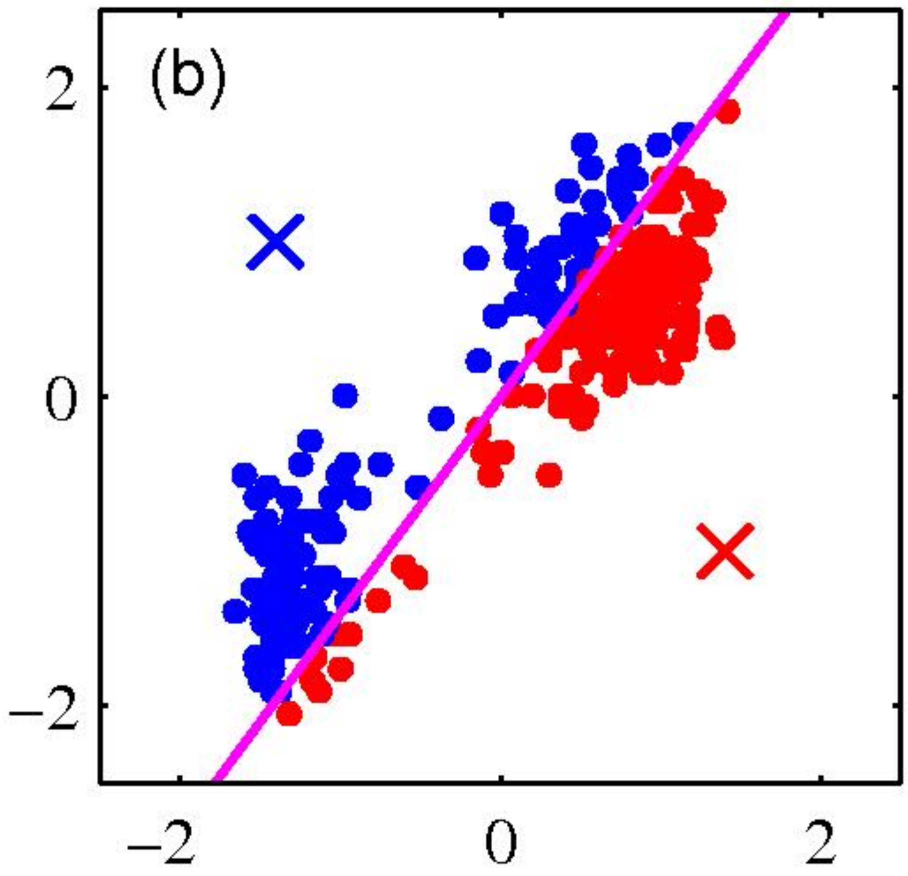




- Pick K random points as cluster centers (means)

Shown here for K=2

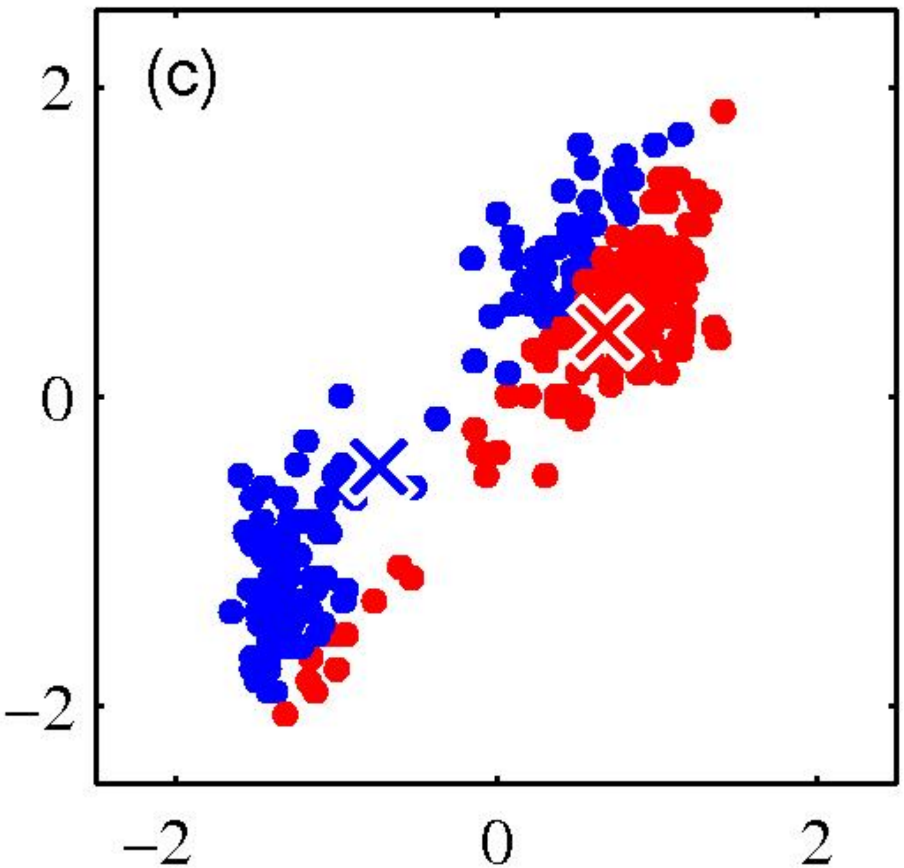
K-means clustering: Example



Iterative Step 1

- Assign data points to closest cluster center

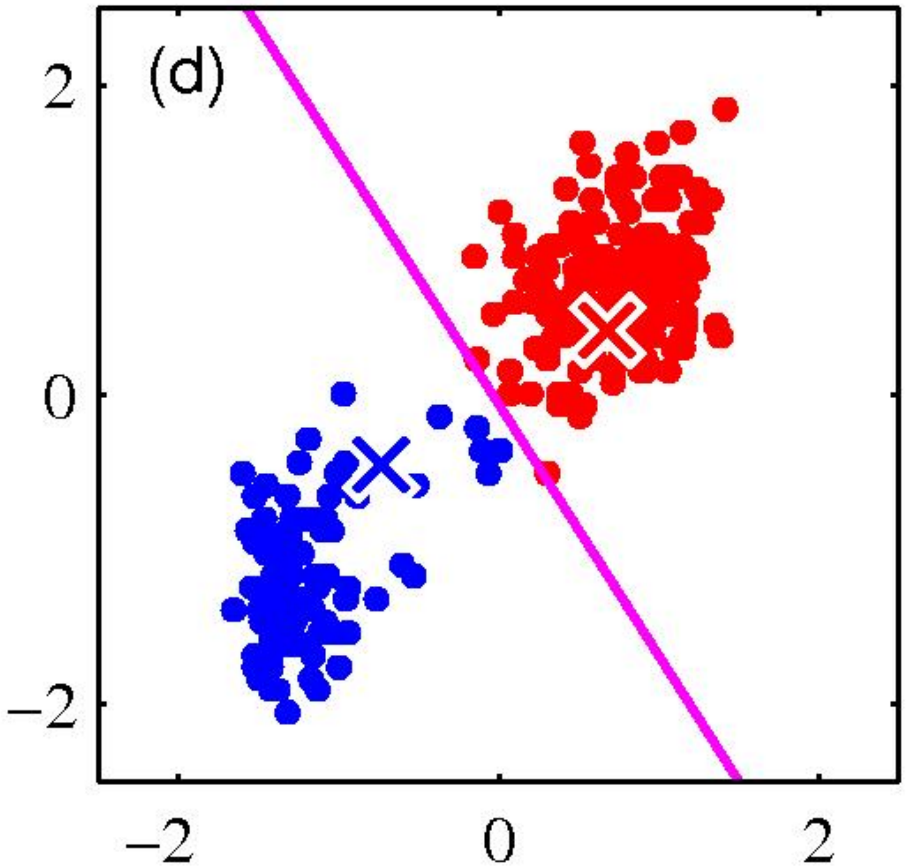
K-means clustering: Example



Iterative Step 2

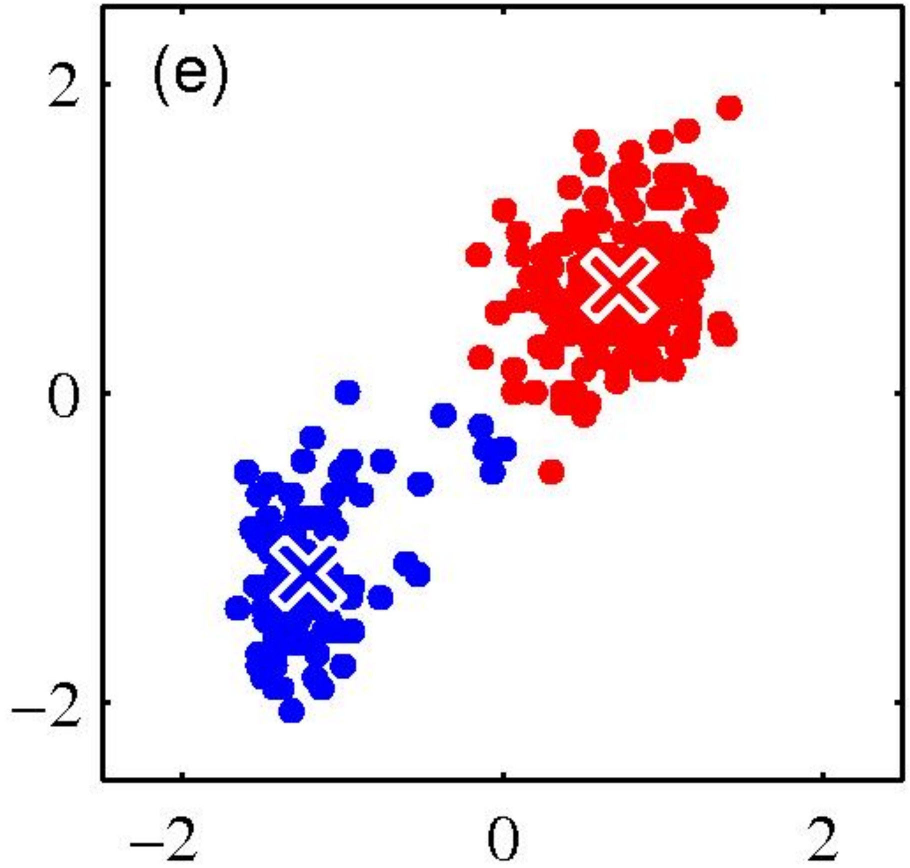
- Change the cluster center to the average of the assigned points

K-means clustering: Example



- Repeat until convergence

K-means clustering: Example



Properties of K-means **algorithm**

- Guaranteed to converge in a finite number of iterations

- Running time per iteration:
    1. Assign data points to closest cluster center

       O(KN) time

    2. Change the cluster center to the average of its assigned points

       O(N)

What properties should a distance measure have?

- Symmetric
  - D(A,B)=D(B,A)
  - Otherwise, we can say A looks like B but B does not look like A
- Positivity, and self-similarity
  - $D(A,B) \geq 0$, and $D(A,B)=0$ iff $A=B$
  - Otherwise there will different objects that we cannot tell apart
- Triangle inequality
  - $D(A,B)+D(B,C) \geq D(A,C)$
  - Otherwise one can say “A is like B, B is like C, but A is not like C at all”

[Slide from Alan Fern]

Kmeans Convergence

Objective

$$\min_{\mu}\min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2$$

1. Fix μ, optimize *C:*

Step 1 of kmeans

$$\min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2 = \min_{c} \sum_{i}^{n} \left|x_i - \mu_{x_i}\right|^2$$

2. Fix *C,* optimize μ:

$$\min_{\mu} \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2$$

   – Take partial derivative of μ_i and set to zero, we have

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

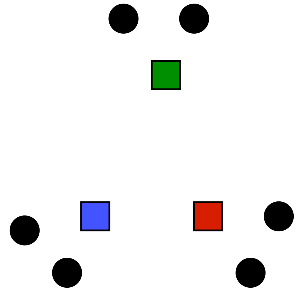
Step 2 of kmeans

Kmeans takes an alternating optimization approach, each step is guaranteed to decrease the objective – thus guaranteed to converge

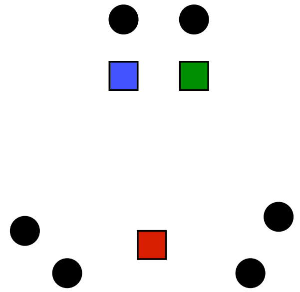
[Slide from Alan Fern]

K-Means Initialization

- K-means **algorithm** is a heuristic
  - Requires initial means
  - It does matter what you pick!



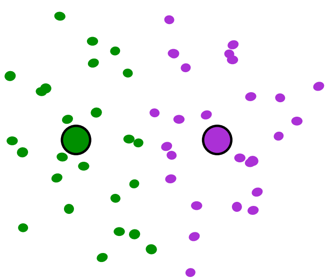
  - What can go wrong?



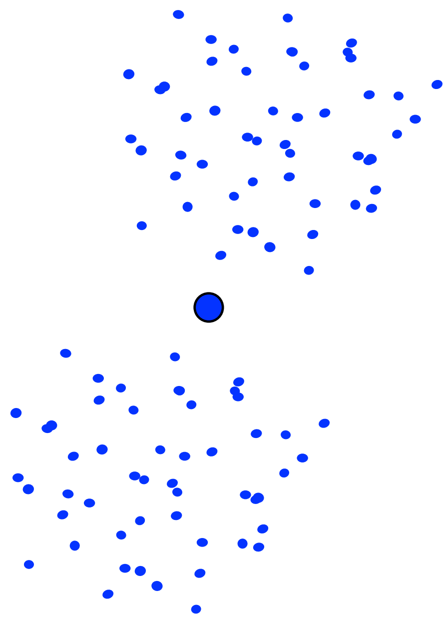
  - Various schemes for preventing this kind of thing: variance-based split / merge, initialization heuristics

K-Means Getting Stuck

A local optimum:

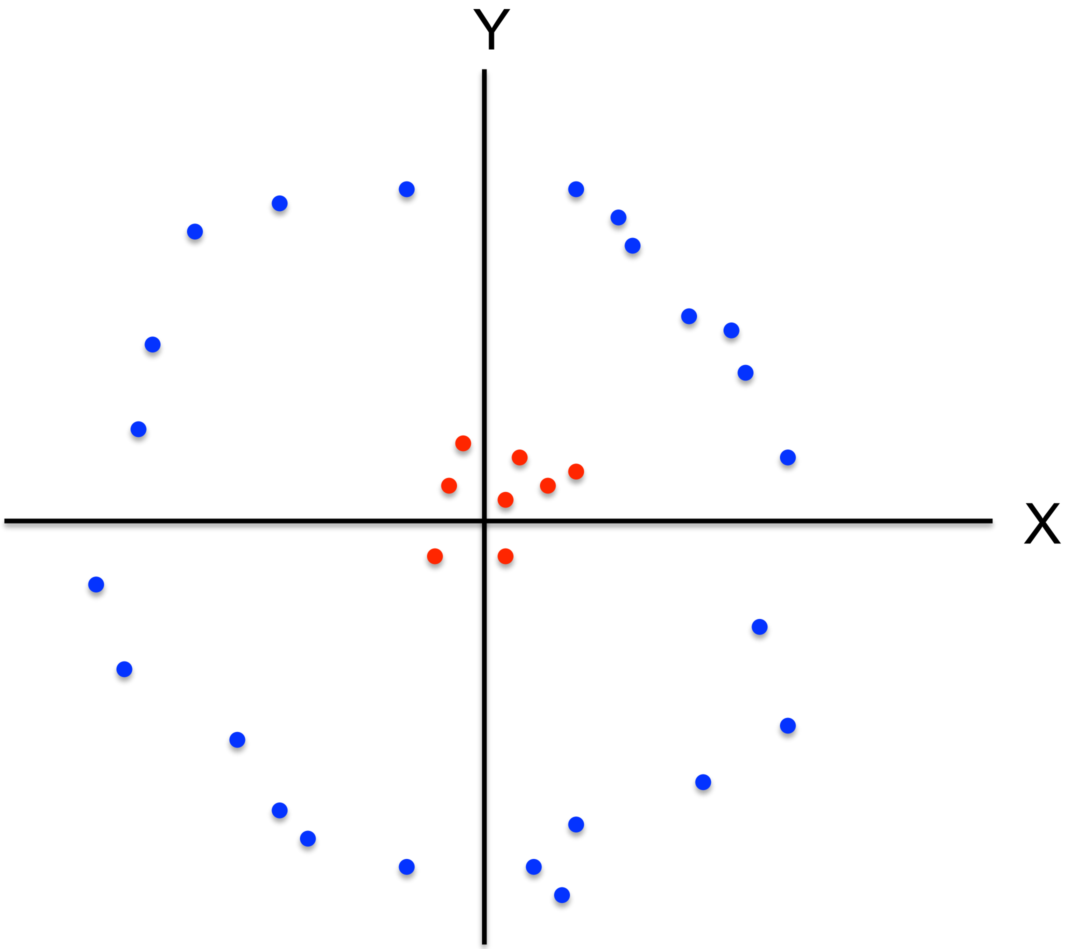


Would be better to have
one cluster here

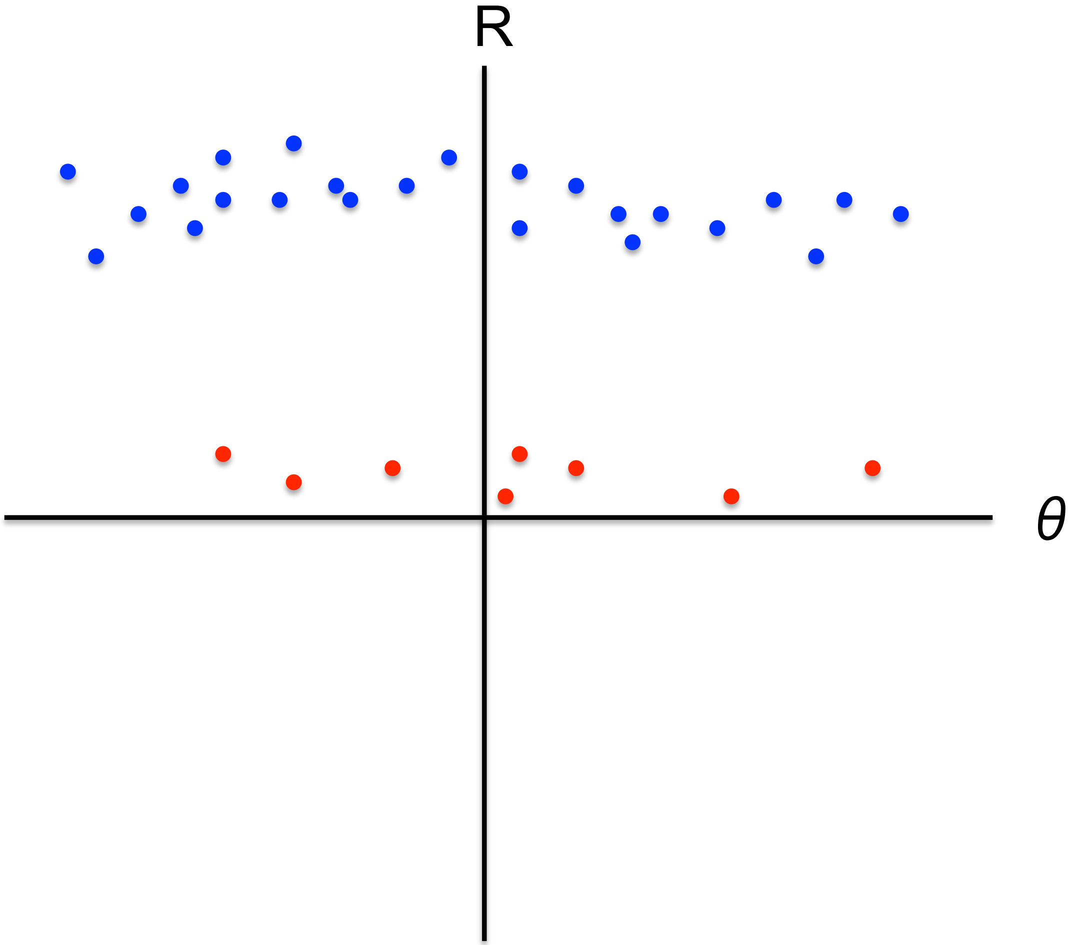


… and two clusters here

K-Means Failures



Distance in K-Means

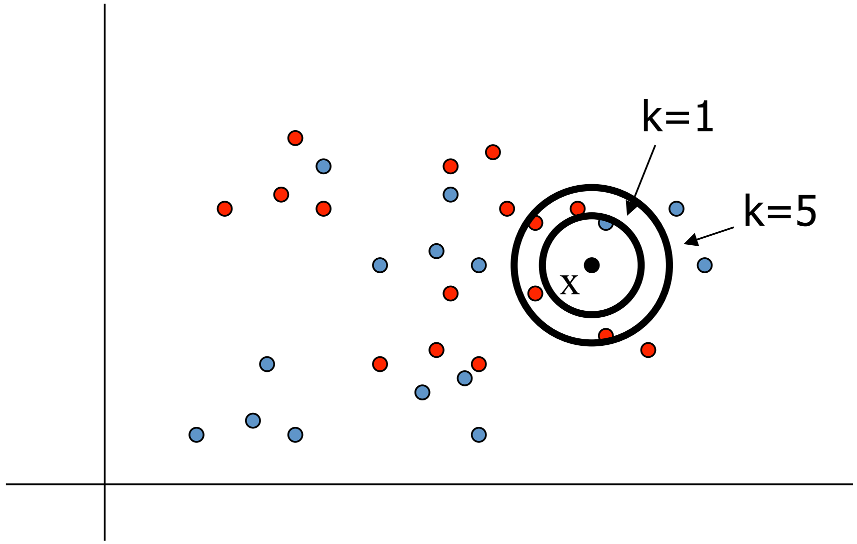


K-Nearest Neighbors (KNN)

- Learning Algorithm:
  - Store training examples
- Prediction Algorithm:
  - To classify a new example **x** by finding the training example ($\mathbf{x}^i$,y^i) that is *nearest* to **x**
  - Guess the class $y = y^i$

K-Nearest Neighbors (KNN)

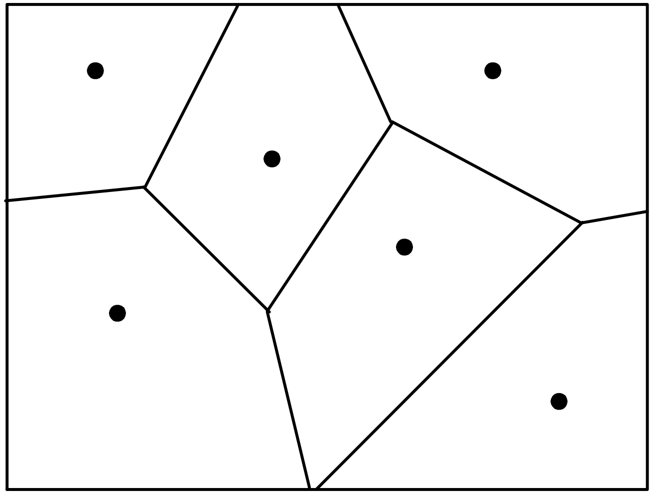
- To classify a new input vector x, examine the k-closest training data points to x and assign the object to the most frequently occurring class



common values for k: 3, 5

K-Nearest Neighbors (KNN)

1-NN Decision Surf ace



- The more examples that are stored, the more complex the decision boundaries can become