Clustering and Classification

Lesson 13

Machine Learning Paradigm

- Observe set of examples: training data
- Infer something about process that generated that data
- Use inference to make predictions about previously unseen data: test data
- Supervised: given a set of feature/label pairs, find a rule that predicts the label associated with a previously unseen input
- *Unsupervised*: given a set of feature vectors (without labels) group them into "natural clusters"

Clustering Is an Optimization Problem

$$variability(c) = \sum_{e \in c} distance(mean(c), e)^2$$

$$dissimilarity(C) = \sum_{c \in C} variability(c)$$

- Why not divide variability by size of cluster?
  - Big and bad worse than small and bad
- Is optimization problem finding a C that minimizes *dissimilarity(C)*?
  - No, otherwise could put each example in its own cluster
- Need a constraint, e.g.,
  - Minimum distance between clusters
  - Number of clusters

Two Popular Methods

- Hierarchical clustering
- K-means clustering

Hiearchical Clustering

1. Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item.

2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one fewer cluster.

3. Continue the process until all items are clustered into a single cluster of size N.

What does distance mean?

Linkage Metrics

- *Single-linkage:* consider the distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster
- *Complete-linkage*: consider the distance between one cluster and another cluster to be equal to the greatest distance from any member of one cluster to any member of the other cluster
- *Average-linkage:* consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster

Example of Hierarchical Clustering

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{BOS} {NY} {CHI} {DEN} {SF} {SEA}

Example of Hierarchical Clustering

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{BOS} {NY} {CHI} {DEN} {SF} {SEA}

{BOS, NY} {CHI} {DEN} {SF} {SEA}

Example of Hierarchical Clustering

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{BOS} {NY} {CHI} {DEN} {SF} {SEA}

{BOS, NY} {CHI} {DEN} {SF} {SEA}

{BOS, NY, CHI} {DEN} {SF} {SEA}

Example of Hierarchical Clustering

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{BOS} {NY} {CHI} {DEN} {SF} {SEA}

{BOS, NY} {CHI} {DEN} {SF} {SEA}

{BOS, NY, CHI} {DEN} {SF} {SEA}

{BOS, NY, CHI} {DEN} {SF, SEA}

Example of Hierarchical Clustering

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{BOS} {NY} {CHI} {DEN} {SF} {SEA}

{BOS, NY} {CHI} {DEN} {SF} {SEA}

{BOS, NY, CHI} {DEN} {SF} {SEA}

{BOS, NY, CHI} {DEN} {SF, SEA}

{BOS, NY, CHI, DEN} {SF, SEA} Single linkage

or

{BOS, NY, CHI} {DEN, SF, SEA} Complete linkage

Clustering Algorithms

- Hierarchical clustering
  - Can select number of clusters using dendogram
  - Deterministic
  - Flexible with respect to linkage criteria
  - Slow
    - Naïve algorithm n^3
    - n^2 algorithms exist for some linkage criteria
- K-means a much faster greedy algorithm
  - Most useful when you know how many clusters you want

K-means Algorithm

```
randomly chose k examples as initial 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
```

What is complexity of one iteration?

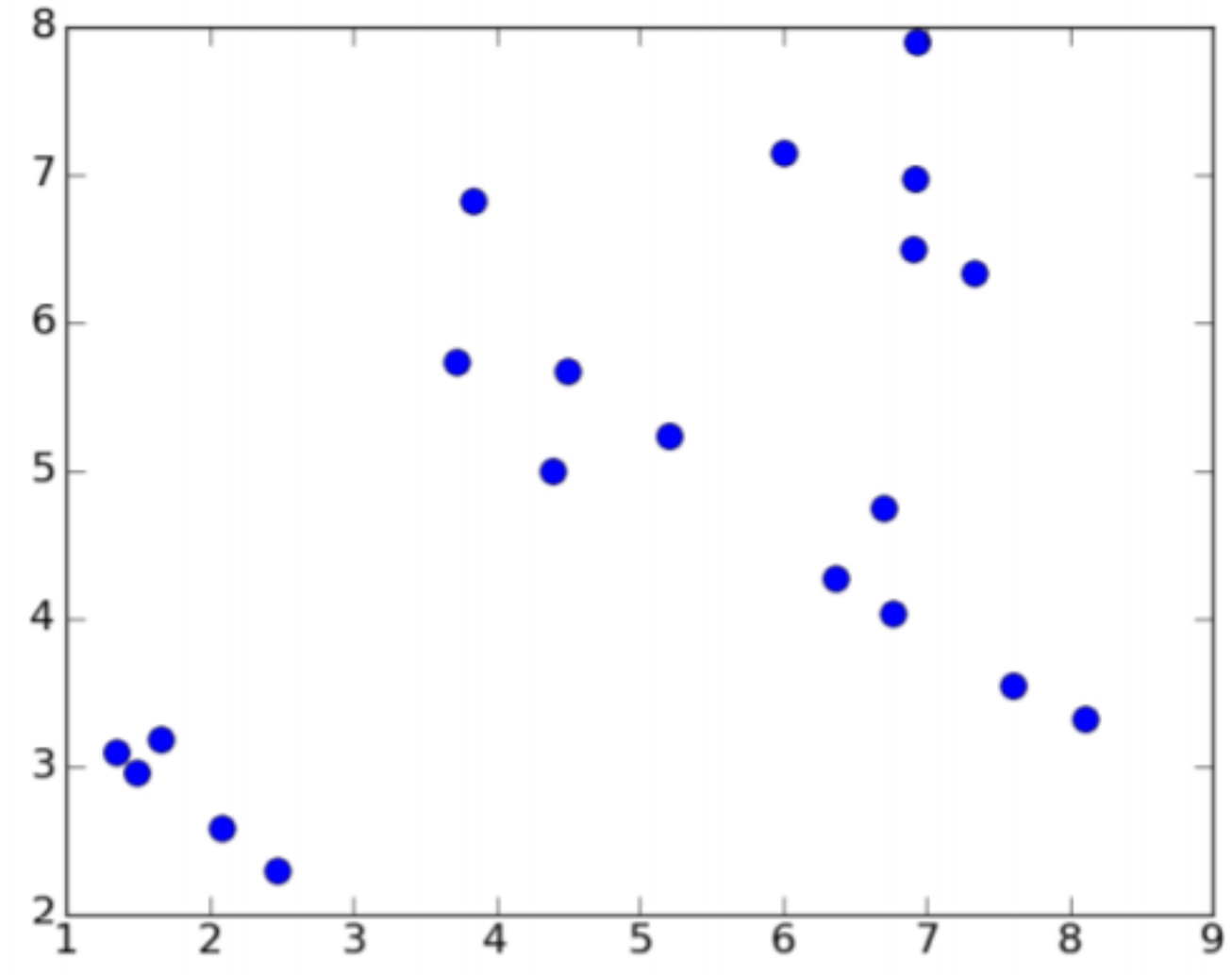
K-means Algorithm

```
randomly chose k examples as initial 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
```

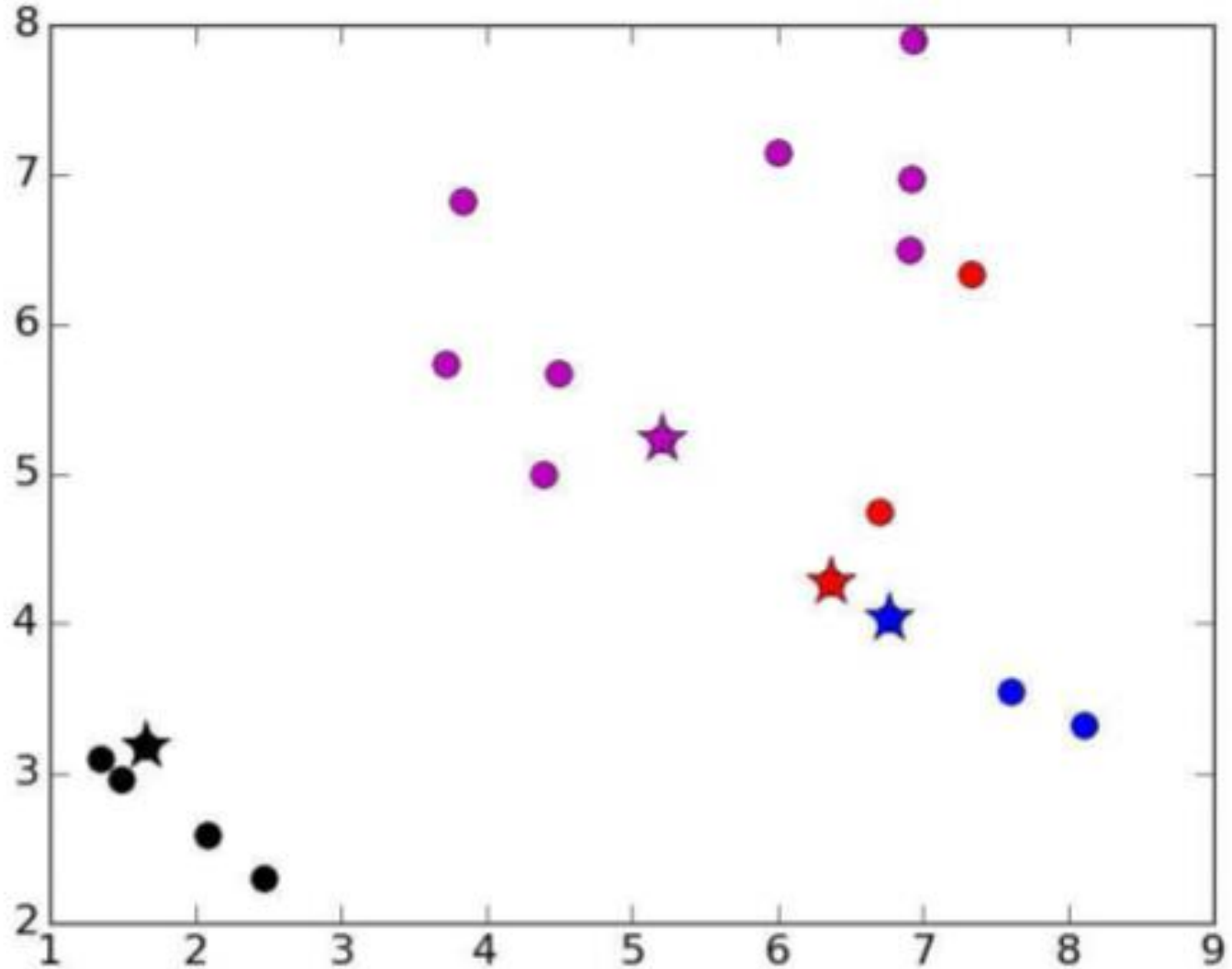
What is complexity of one iteration?

k*n*d, where n is number of points and d time required to compute the distance between a pair of points

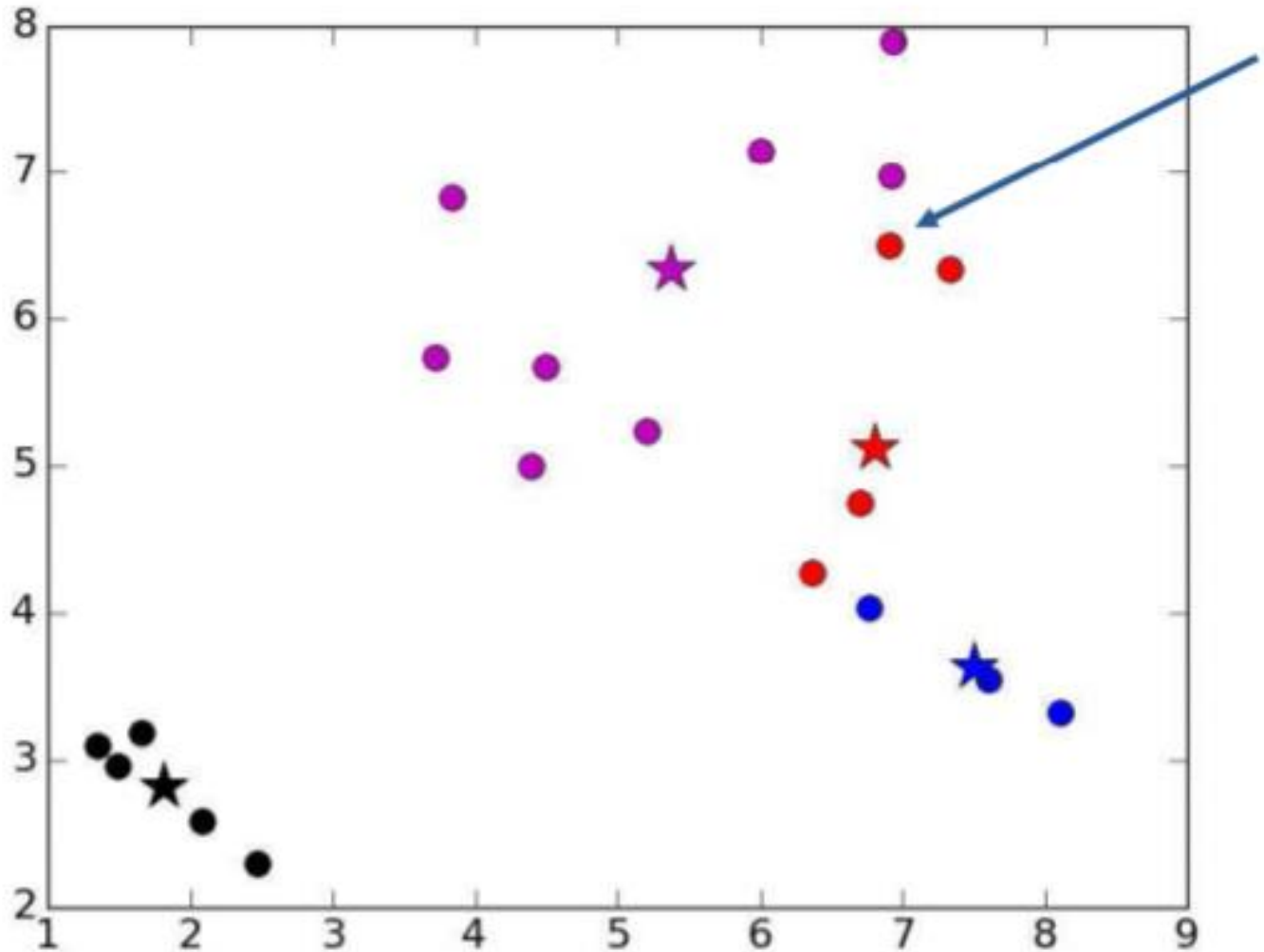
An Example



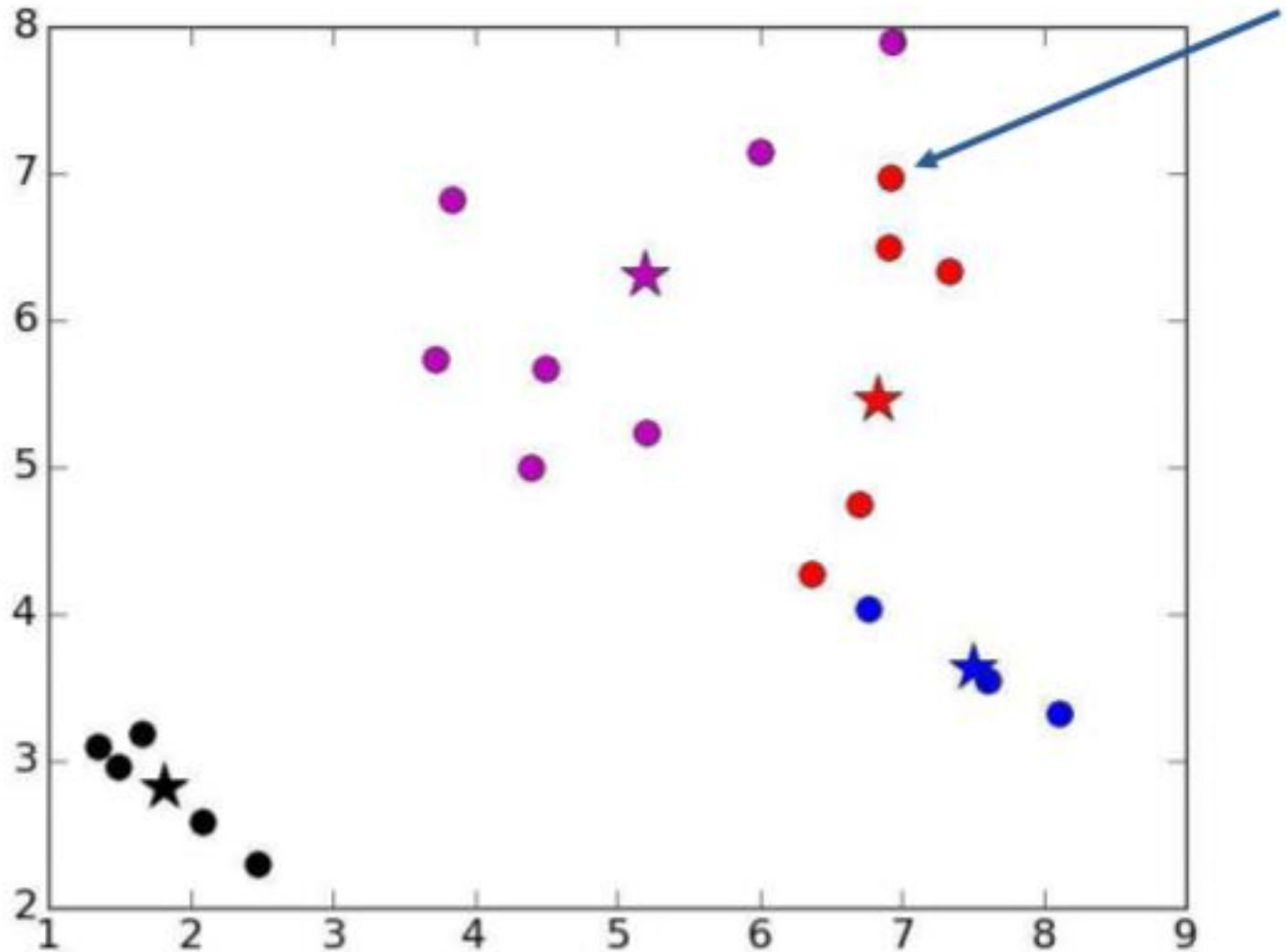
K = 4, Initial Centroids



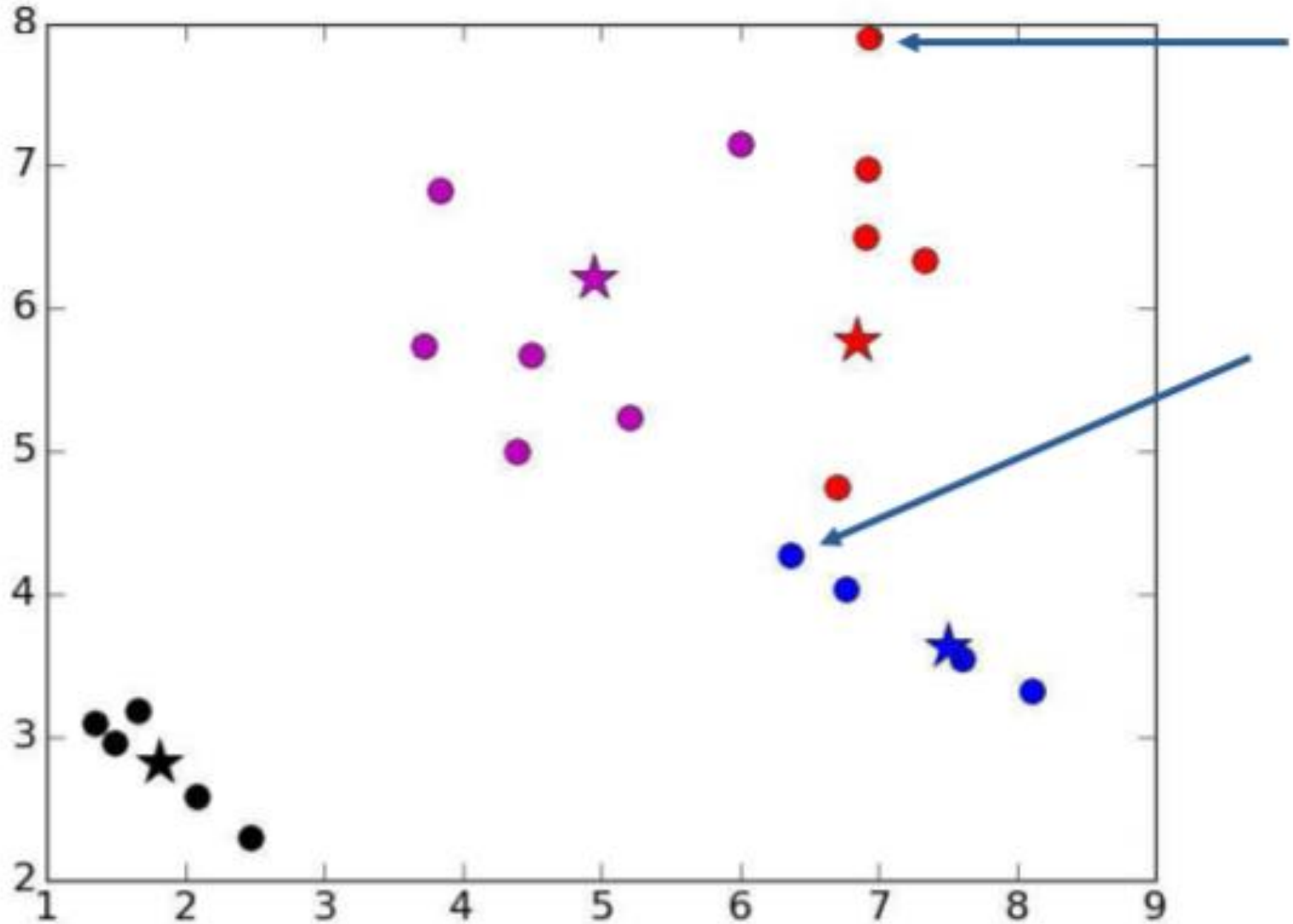
Iteration 1



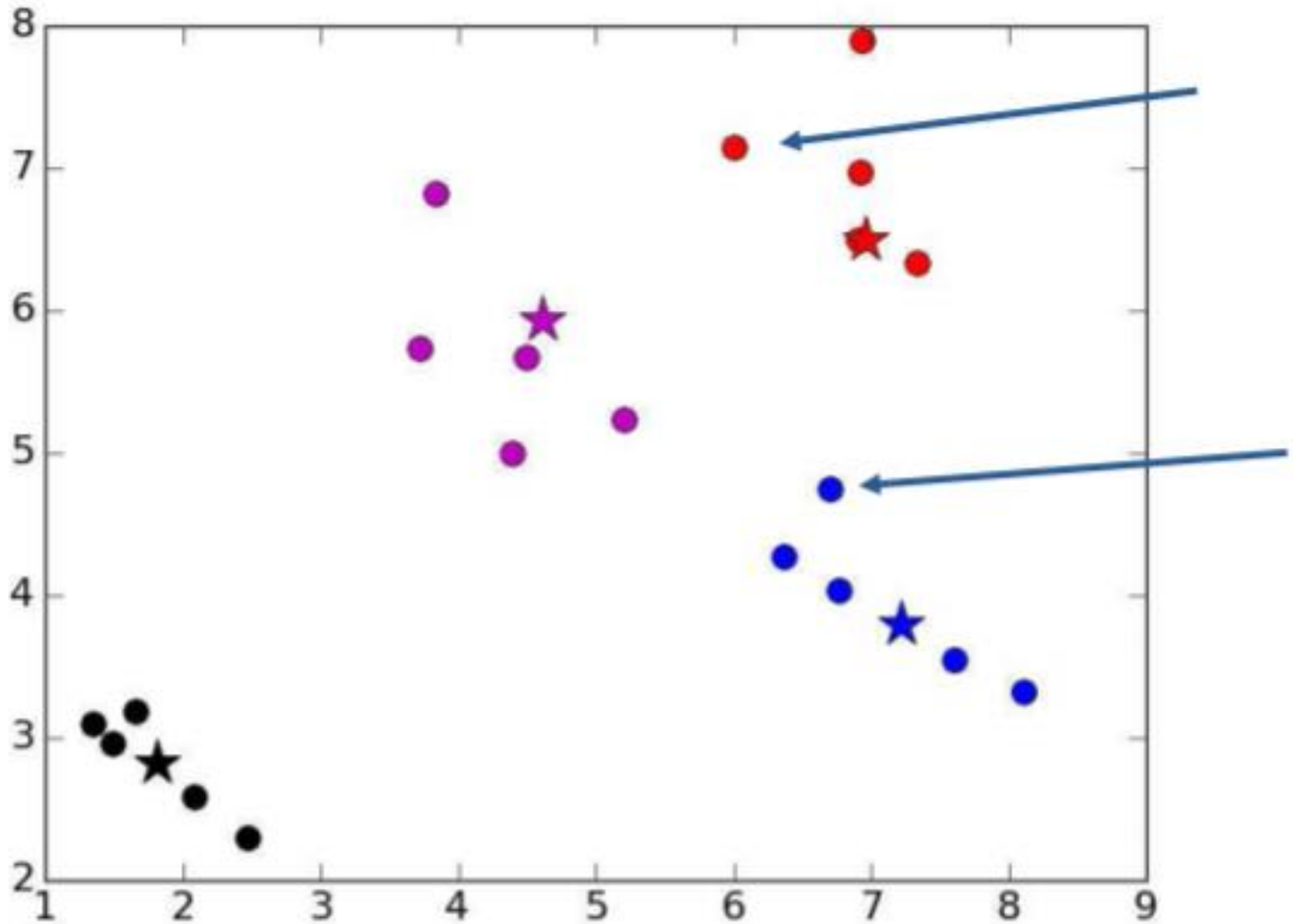
Iteration 2



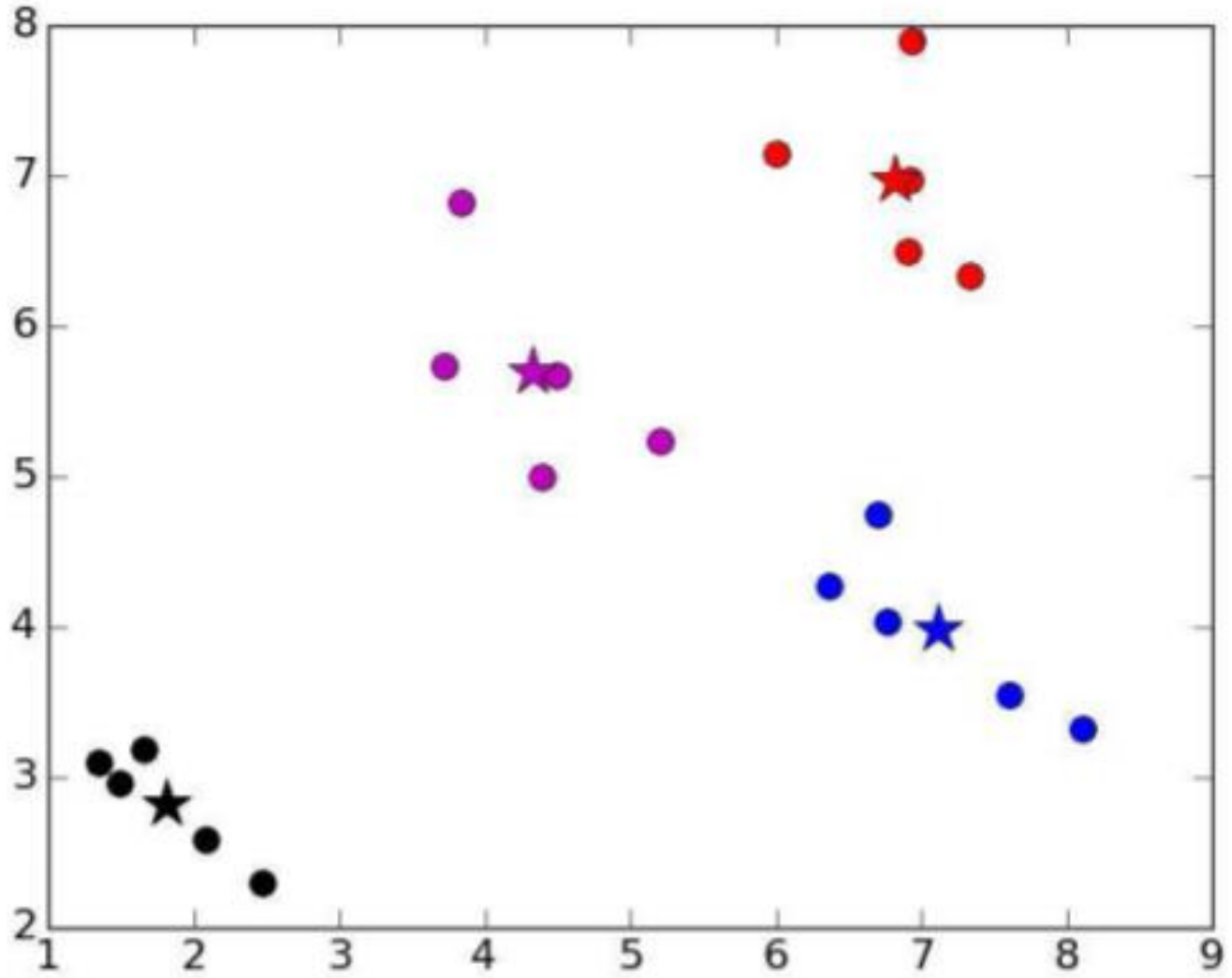
Iteration 3



Iteration 4

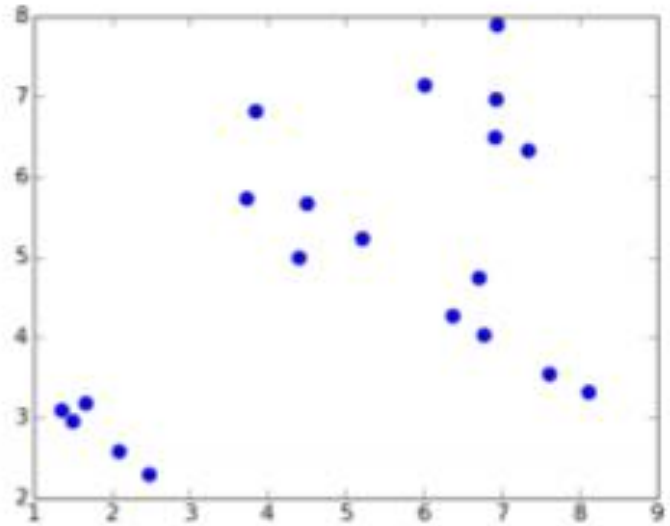


Iteration 5



Issues with k-means

- Choosing the "wrong" k can lead to strange results
  - Consider k = 3

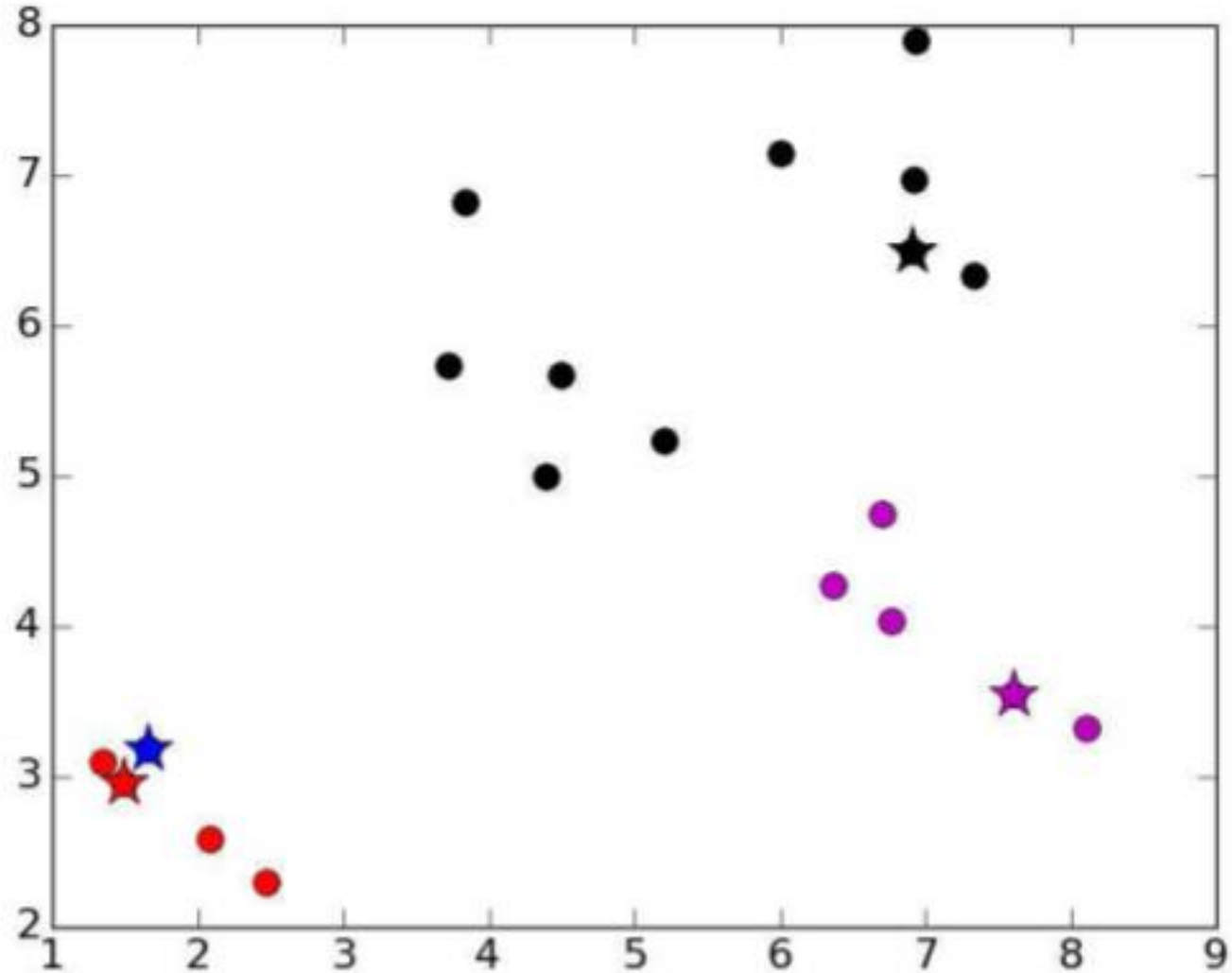


- Result can depend upon initial centroids
  - Number of iterations
  - Even final result
    - Greedy algorithm can find different local optimas

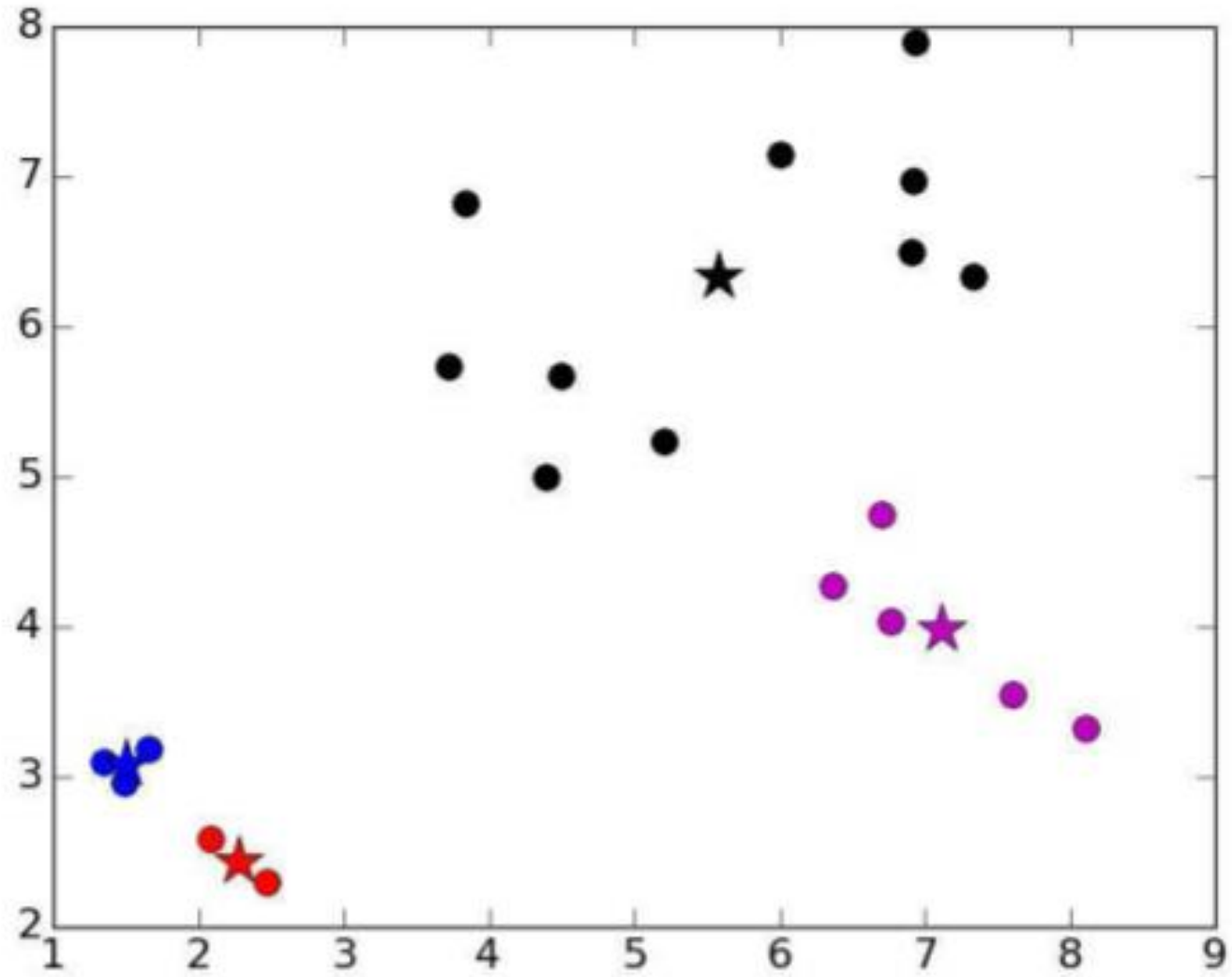
How to Choose K

- *A priori* knowledge about application domain
  - There are two kinds of people in the world: k = 2
  - There are five different types of bacteria: k = 5
- Search for a good k
  - Try different values of k and evaluate quality of results
  - Run hierarchical clustering on subset of data

Unlucky Initial Centroids



Converges On



Mitigating Dependence on Initial Centroids

Try multiple sets of randomly chosen initial centroids

Select “best” result

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

Evaluating a Clustering

```
def dissimilarity(clusters):
    """Assumes clusters a list of clusters
       Returns a measure of the total dissimilarity of the
       clusters in the list"""
    totDist = 0
    for c in clusters:
        totDist += c.variability()
    return totDist
```

Classification

Lesson 13b

Supervised Learning

- Regression
  - Predict a real number associated with a feature vector
  - E.g., use linear regression to fit a curve to data
- *Classification*
  - Predict a discrete value (label) associated with a feature vector

	Features					Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	Number legs	Reptile
Cobra	1	1	1	1	0	1
Rattlesnake	1	1	1	1	0	1
Boa constrictor	0	1	0	1	0	1
Chicken	1	1	0	1	2	0
Guppy	0	1	0	0	0	0
Dart frog	1	0	1	0	4	0
Zebra	0	0	0	0	4	0
Python	1	1	0	1	0	1
Alligator	1	1	0	1	4	1

Distance Matrix

	cobra	rattlesnake	boa constrictor	chicken	guppy	dart frog	zebra	python	alligator
cobra	–	0.0	1.414	2.236	1.732	1.732	2.236	1.0	1.414
rattlesnake	0.0	–	1.414	2.236	1.732	1.732	2.236	1.0	1.414
boa constrictor	1.414	1.414	–	2.236	1.0	2.236	1.732	1.0	1.414
chicken	2.236	2.236	2.236	–	2.449	2.0	2.0	2.0	1.0
guppy	1.732	1.732	1.0	2.449	–	2.0	1.414	1.414	1.732
dart frog	1.732	1.732	2.236	2.0	2.0	–	1.414	2.0	1.732
zebra	2.236	2.236	1.732	2.0	1.414	1.414	–	2.0	1.732
python	1.0	1.0	1.0	2.0	1.414	2.0	2.0	–	1.0
alligator	1.414	1.414	1.414	1.0	1.732	1.732	1.732	1.0	–

Using Distance Matrix for Classification

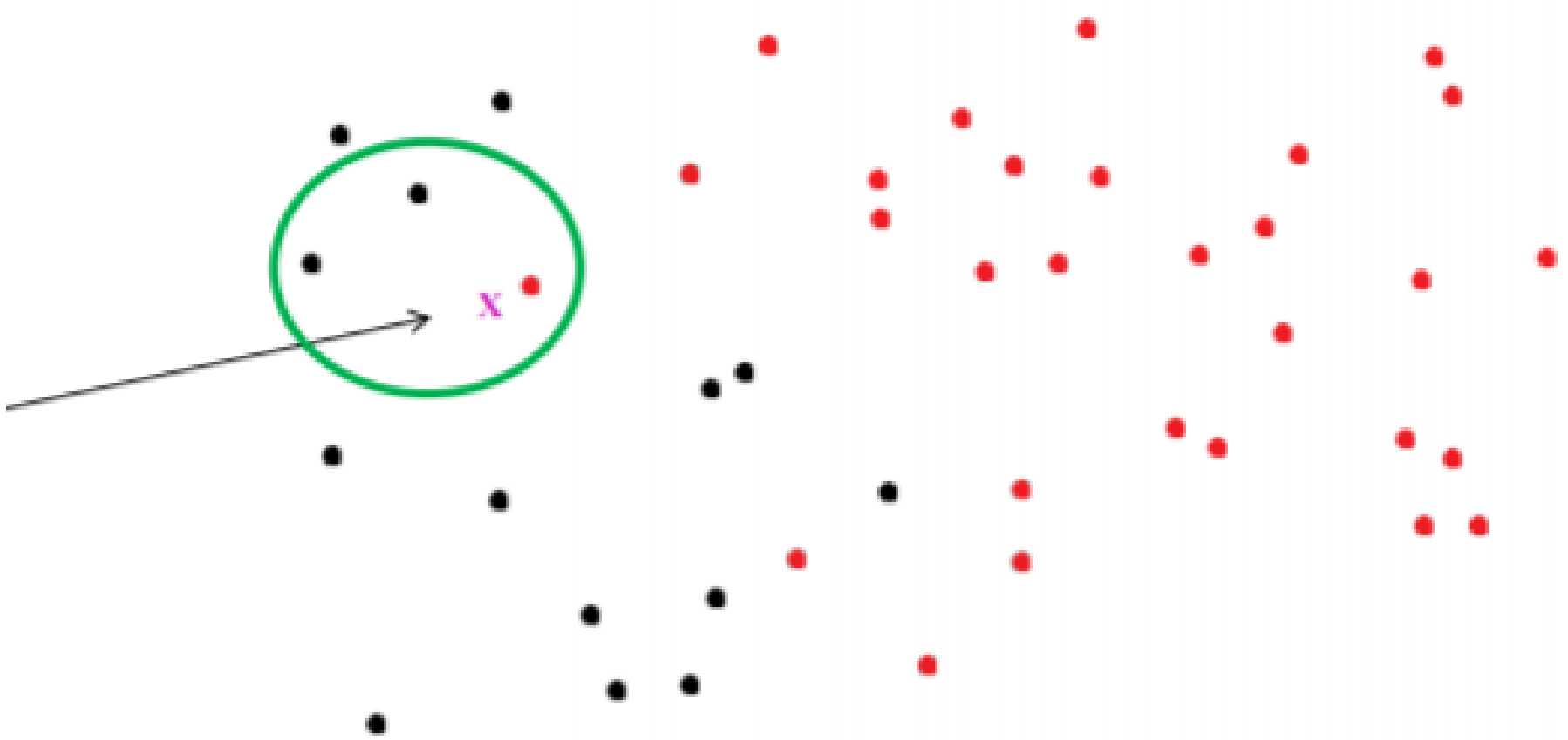
- Simplest approach is probably nearest neighbor
- Remember training data
- When predicting the label of a new example
  - Find the nearest example in the training data
  - Predict the label associated with that example

Distance Matrix

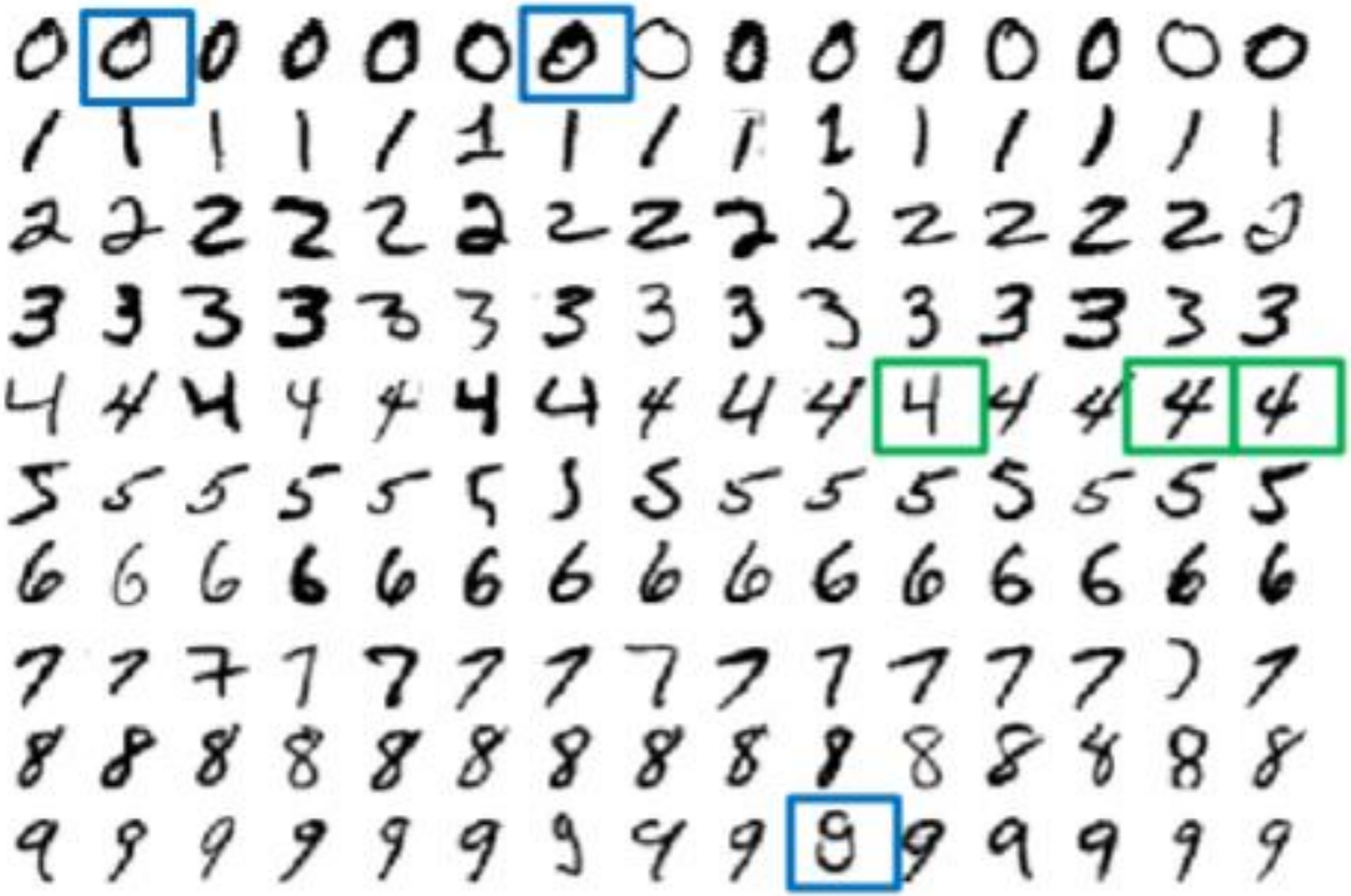
	cobra	rattlesnake	boa constrictor	chicken	guppy	dart frog	zebra	python	alligator	Label
cobra	–	0.0	1.414	2.236	1.732	1.732	2.236	1.0	1.414	R
rattlesnake	0.0	–	1.414	2.236	1.732	1.732	2.236	1.0	1.414	R
boa constrictor	1.414	1.414	–	2.236	1.0	2.236	1.732	1.0	1.414	R
chicken	2.236	2.236	2.236	–	2.449	2.0	2.0	2.0	1.0	~R
guppy	1.732	1.732	1.0	2.449	–	2.0	1.414	1.414	1.732	~R
dart frog	1.732	1.732	2.236	2.0	2.0	–	1.414	2.0	1.732	~R
zebra	2.236	2.236	1.732	2.0	1.414	1.414				
python	1.0	1.0	1.0	2.0	1.414	2.0				
alligator	1.414	1.414	1.414	1.0	1.732	1.732				

An Example

K-nearest Neighbors



An Example



Advantages and Disadvantages of KNN

- Advantages
  - Learning fast, no explicit training
  - No theory required
  - Easy to explain method and results
- Disadvantages
  - Memory intensive and predictions can take a long time
    - Are better algorithms than brute force
  - No model to shed light on process that generated data

The Titanic Disaster

- RMS Titanic sank in the North Atlantic the morning of 15 April 1912, after colliding with an iceberg. Of the 1,300 passengers aboard, 812 died. (703 of 918 crew members died.)
- Database of 1046 passengers
  - Cabin class
    - 1^{st}, 2^{nd}, 3^{rd}
  - Age
  - Gender

Is Accuracy Enough

- If we predict “died”, accuracy will be >62% or passenger and >76% for crew members
- Consider a disease that occurs in 0.1% of population
  - Predicting disease-free has an accuracy of 0.999

Other Metrics

$$sensitivity = \frac{true\ positive}{true\ positive + false\ negative}$$

$$specificity = \frac{true\ negative}{true\ negative + false\ positive}$$

$$positive\ predictive\ value = \frac{true\ positive}{true\ positive + false\ positive}$$

$$negative\ predictive\ value = \frac{true\ negative}{true\ negative + false\ negative}$$

sensitivity = recall
specificity = precision

Testing Methodology Matters

- Leave-one-out
- Repeated random subsampling

Repeated Random Subsampling

```
def split80_20(examples):
    sampleIndices = random.sample(range(len(examples)),
                                  len(examples)//5)
    trainingSet, testSet = [], []
    for i in range(len(examples)):
        if i in sampleIndices:
            testSet.append(examples[i])
        else:
            trainingSet.append(examples[i])
    return trainingSet, testSet
```

Repeated Random Subsampling

```
def randomSplits(examples, method, numSplits, toPrint = True):
    truePos, falsePos, trueNeg, falseNeg = 0, 0, 0, 0
    random.seed(0)
    for t in range(numSplits):
        trainingSet, testSet = split80_20(examples)
        results = method(trainingSet, testSet)
        truePos += results[0]
        falsePos += results[1]
        trueNeg += results[2]
        falseNeg += results[3]
    getStats(truePos/numSplits, falsePos/numSplits,
             trueNeg/numSplits, falseNeg/numSplits, toPrint)
    return truePos/numSplits, falsePos/numSplits,\
             trueNeg/numSplits, falseNeg/numSplits
```

To do.

- Assignment: Clustering and Classification
  - due 8 June 17:00
- Quiz 04: Curve Fitting
  - due 1 June 17:00

- FSR: complete the survey in IVY that will inform me about your topic/concept
- FE: complete the assignments and quiz. Topics to be tested:
  - Hierachical Clustering, Classification, Clustering, Curve Fitting, DFS, BFS and DP, monte carlo simulation (short answer and practical)