



Department of Computer Science
UNIVERSITY OF COLORADO BOULDER



Machine Learning: Yoshinari Fujinuma

University of Colorado Boulder
LECTURE 23

Slides adapted from Chenhao Tan, Jordan Boyd-Graber, Chris Ketelsen

Logistics

- HW4 is due on Wednesday.
- In-class quiz 2 on Friday...?

Learning objectives

- Learn about general clustering
- Learn about the K-means algorithm

Outline

Recap of Unsupervised Learning

Clustering

K-means

Supervised vs. unsupervised learning

Data

X

Hidden
Structure

Z

- **Unsupervised methods** find **hidden structure** in data, structure that we can never formally observe.

When do we need unsupervised learning?

- Acquiring labels is expensive
- You may not even know what labels to acquire

When do we need unsupervised learning?

- Exploratory data analysis
- Learn patterns/representations that can be useful for supervised learning (representation learning)
- Generate data
- ...

Outline

Recap of Unsupervised Learning

Clustering

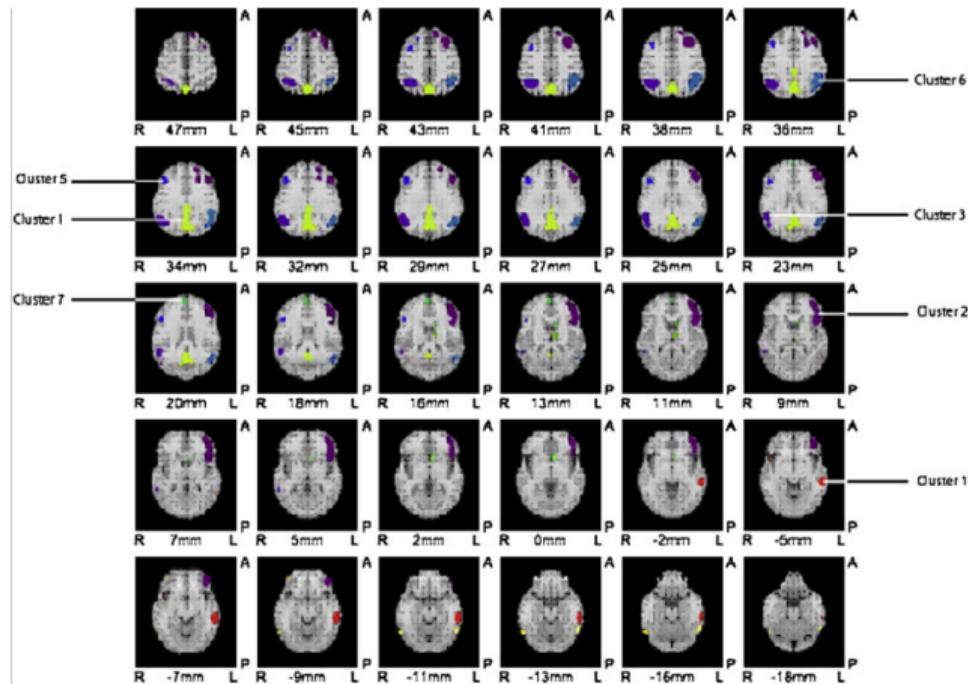
K-means

Clustering

- One important unsupervised method is clustering
- Goal: Organize data in classes

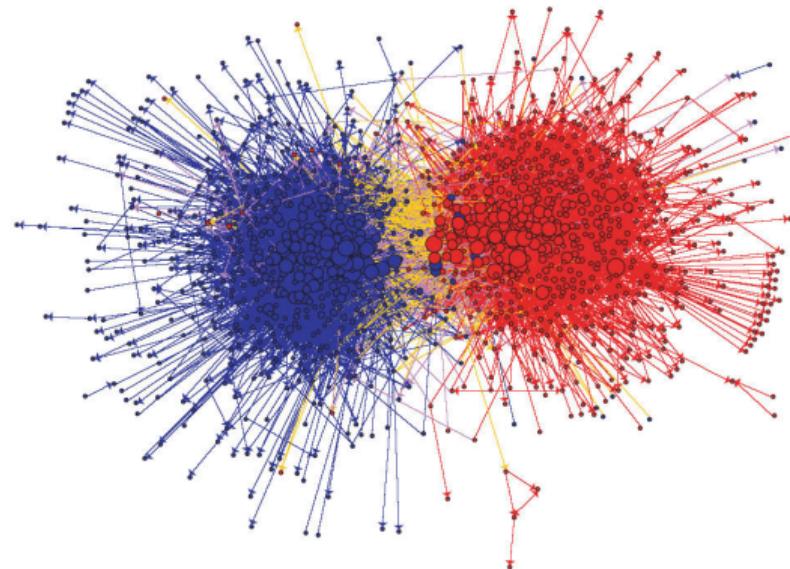
Clustering Applications

Medical imaging



Clustering Applications

Community detection



Adamic & Glance (2005)

Clustering

- One important unsupervised method is clustering
- Goal: Organize data in classes
 - Classes are hard to define
 - Different data representation may lead to different clusterings

Clustering

- One important unsupervised method is clustering
- Goal: Organize data in classes
 - Data have high in-class similarity
 - Data have low out-of-class similarity

Clustering — Similarity

We'll call $d(\mathbf{x}, \mathbf{y})$ the similarity measure of \mathbf{x} and \mathbf{y}

Examples:

Euclidean Distance: $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$

Edit Distance: $d(\mathbf{x}, \mathbf{y}) = \# \text{ replace, insert, deletes to turn } \mathbf{x} \text{ into } \mathbf{y}$

$$d(\text{kitten}, \text{sitting}) = 3$$

kitten → sitten → sittin → sitting

What properties make a good similarity measure?

Clustering — Similarity

We'll call $d(\mathbf{x}, \mathbf{y})$ the similarity measure of \mathbf{x} and \mathbf{y}

Properties:

Symmetry $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$

Self-Consistency $d(\mathbf{x}, \mathbf{x}) = 0$

Positivity $d(\mathbf{x}, \mathbf{y}) = 0$ iff $\mathbf{x} = \mathbf{y}$

Triangle Inequality $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$

OK, so say we have a good similarity measure.

How do we find clusters in the data?

Outline

Recap of Unsupervised Learning

Clustering

K-means

K-means

- Simplest clustering method
- Iterative in nature
- Reasonably fast
- Very popular in practice (though with more bells and whistles)
- Requires real-valued data

K-means

General Idea:

pick K initial cluster means

do until convergence ...

- associate examples closest to mean k with cluster k
- update cluster means with current examples in cluster k

Stop when:

- cluster assignments don't change
- cluster means don't change (too much)

K-means example

Input data



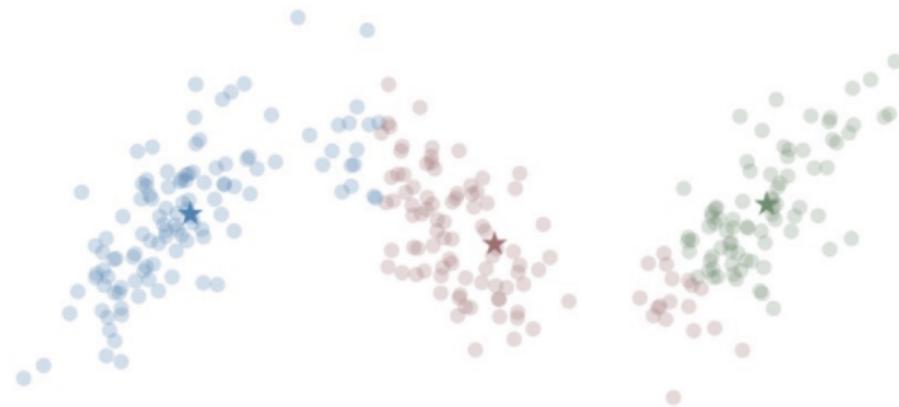
K-means example

Randomly pick centroids, assign each data point to the closest centroid



K-means example

Recompute centroids



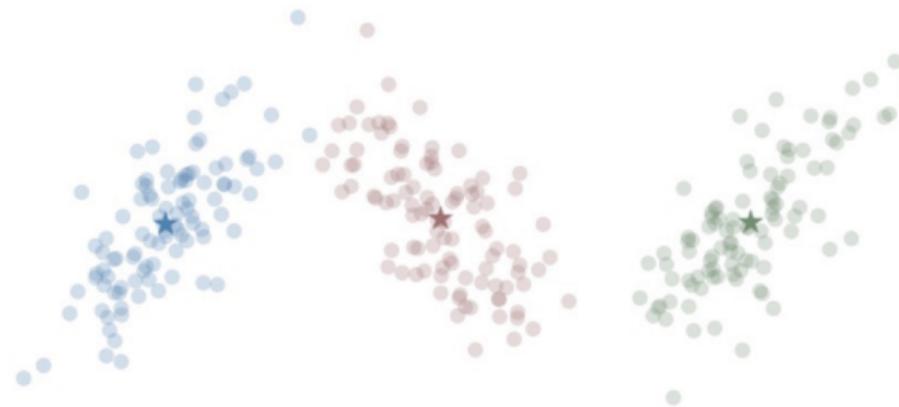
K-means example

Assign each data point to the closest centroid



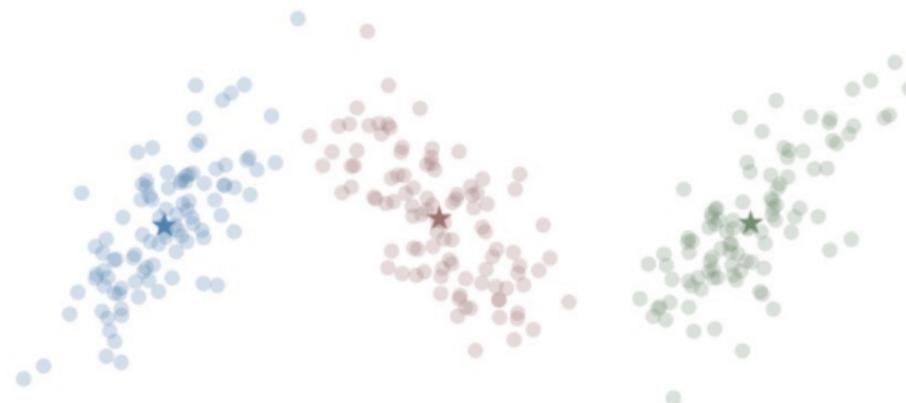
K-means example

Recompute centroids



K-means example

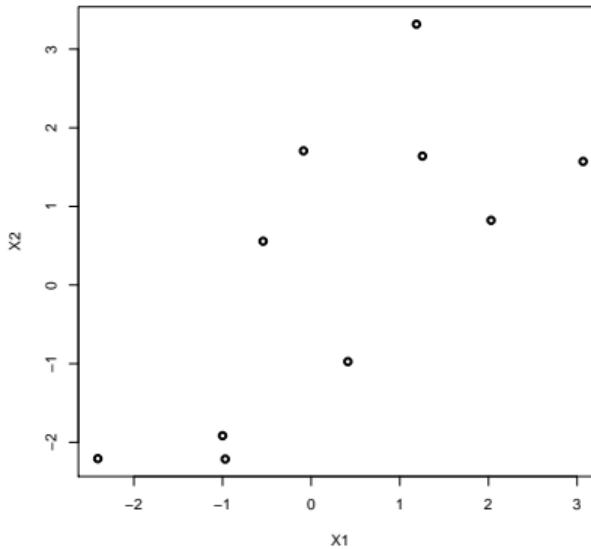
Converged



K-means example in numbers

Data:

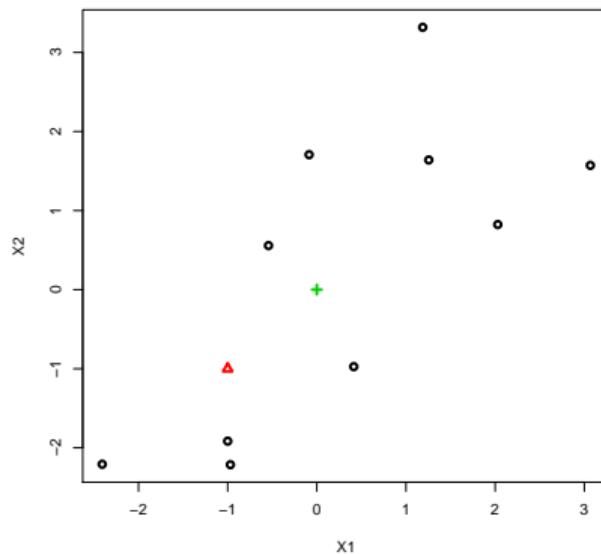
x_1	x_2
0.4	-1.0
-1.0	-2.2
-2.4	-2.2
-1.0	-1.9
-0.5	0.6
-0.1	1.7
1.2	3.3
3.1	1.6
1.3	1.6
2.0	0.8



K-means example in numbers

Pick K centers (randomly):

$$(-1, -1) \text{ and } (0, 0)$$



K-means example in numbers

Calculate distance between points and those centers:

x_1	x_2	(-1, -1)	(0, 0)
0.4	-1.0	1.4	1.1
-1.0	-2.2	1.2	2.4
-2.4	-2.2	1.9	3.3
-1.0	-1.9	0.9	2.2
-0.5	0.6	1.6	0.8
-0.1	1.7	2.9	1.7
1.2	3.3	4.8	3.5
3.1	1.6	4.8	3.4
1.3	1.6	3.5	2.1
2.0	0.8	3.5	2.2

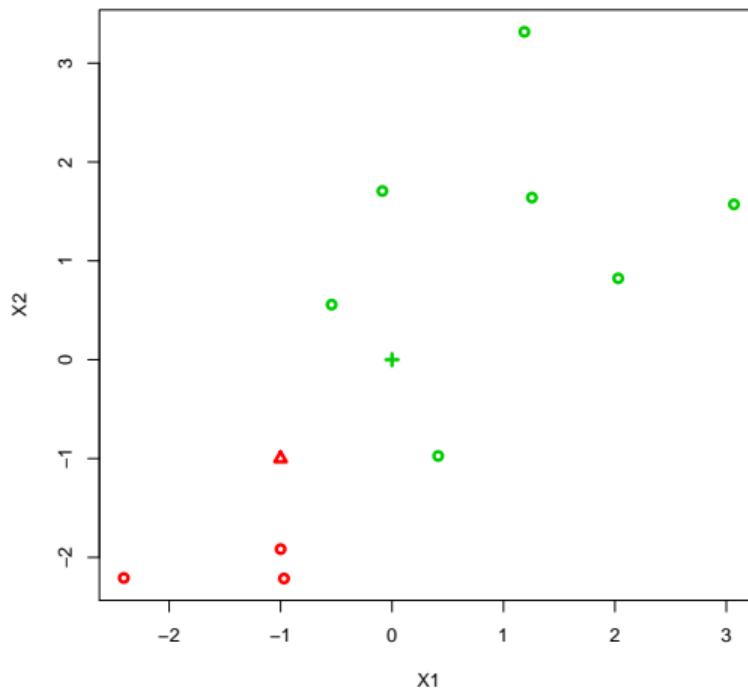
K-means example in numbers

Choose mean with smaller distance:

x_1	x_2	(-1, -1)	(0, 0)
0.4	-1.0	1.4	1.1
-1.0	-2.2	1.2	2.4
-2.4	-2.2	1.9	3.3
-1.0	-1.9	0.9	2.2
-0.5	0.6	1.6	0.8
-0.1	1.7	2.9	1.7
1.2	3.3	4.8	3.5
3.1	1.6	4.8	3.4
1.3	1.6	3.5	2.1
2.0	0.8	3.5	2.2

K-means example in numbers

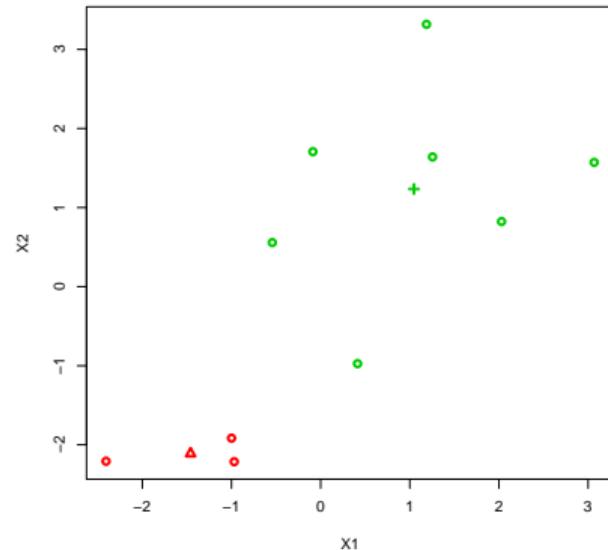
New clusters:



K-means example in numbers

Refit means for each cluster:

- cluster 1: $(-1.0, -2.2)$, $(-2.4, -2.2)$,
 $(-1.0, -1.9)$
- new mean:** $(-1.5, -2.1)$
- cluster 2: $(0.4, -1.0)$, $(-0.5, 0.6)$,
 $(-0.1, 1.7)$, $(1.2, 3.3)$, $(3.1, 1.6)$,
 $(1.3, 1.6)$, $(2.0, 0.8)$
- new mean:** $(1.0, 1.2)$



K-means example in numbers

Recalculate distances for each cluster:

x_1	x_2	(-1.5, -2.1)	(1.0, 1.2)
0.4	-1.0	2.2	2.3
-1.0	-2.2	0.5	4.0
-2.4	-2.2	1.0	4.9
-1.0	-1.9	0.5	3.8
-0.5	0.6	2.8	1.7
-0.1	1.7	4.1	1.2
1.2	3.3	6.0	2.1
3.1	1.6	5.8	2.0
1.3	1.6	4.6	0.5
2.0	0.8	4.6	1.1

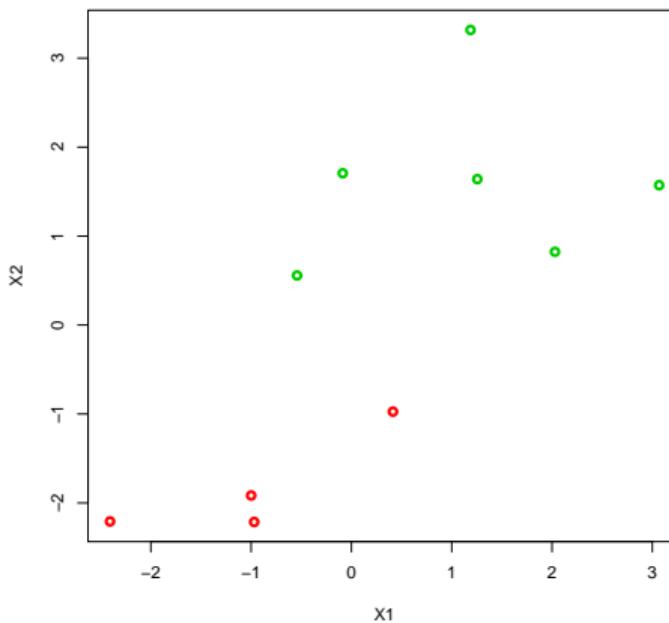
K-means example in numbers

Choose mean with smaller distance:

x_1	x_2	(-1.5, -2.1)	(1.0, 1.2)
0.4	-1.0	2.2	2.3
-1.0	-2.2	0.5	4.0
-2.4	-2.2	1.0	4.9
-1.0	-1.9	0.5	3.8
-0.5	0.6	2.8	1.7
-0.1	1.7	4.1	1.2
1.2	3.3	6.0	2.1
3.1	1.6	5.8	2.0
1.3	1.6	4.6	0.5
2.0	0.8	4.6	1.1

K-Means: Example

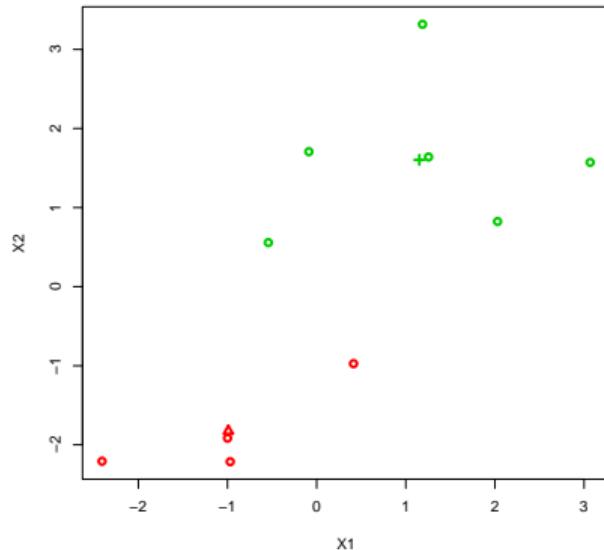
New clusters:



K-Means: Example

Refit means for each cluster:

- cluster 1: $(0.4, -1.0)$, $(-1.0, -2.2)$,
 $(-2.4, -2.2)$, $(-1.0, -1.9)$
- new mean: $(-1.0, -1.8)$
- cluster 2: $(-0.5, 0.6)$, $(-0.1, 1.7)$,
 $(1.2, 3.3)$, $(3.1, 1.6)$, $(1.3, 1.6)$, $(2.0, 0.8)$
- new mean: $(1.2, 1.6)$



K-means example in numbers

Recalculate distances for each cluster:

x_1	x_2	(-1.0, -1.8)	(1.2, 1.6)
0.4	-1.0	1.6	2.7
-1.0	-2.2	0.4	4.4
-2.4	-2.2	1.5	5.2
-1.0	-1.9	0.1	4.1
-0.5	0.6	2.4	2.0
-0.1	1.7	3.6	1.2
1.2	3.3	5.6	1.7
3.1	1.6	5.3	1.9
1.3	1.6	4.1	0.1
2.0	0.8	4.0	1.2

K-means example in numbers

Select smallest distance and compare these clusters with previous:

Table: New Clusters

x_1	x_2	(-1.0, -1.8)	(1.2, 1.6)
0.4	-1.0	1.6	2.7
-1.0	-2.2	0.4	4.4
-2.4	-2.2	1.5	5.2
-1.0	-1.9	0.1	4.1
-0.5	0.6	2.4	2.0
-0.1	1.7	3.6	1.2
1.2	3.3	5.6	1.7
3.1	1.6	5.3	1.9
1.3	1.6	4.1	0.1
2.0	0.8	4.0	1.2

Table: Old Clusters

(-1.5, -2.1)	(1.0, 1.2)
2.2	2.3
0.5	4.0
1.0	4.9
0.5	3.8
2.8	1.7
4.1	1.2
6.0	2.1
5.8	2.0
4.6	0.5
4.6	1.1

K-means

Strengths:

- Simple to understand
- Efficient - time complexity $\mathcal{O}(dNKT)$ for $\mathbf{x} \in \mathbb{R}^d$
 - N : number of points
 - K : number of clusters
 - T : number of iterations
- Simple to implement

K-means

```
def KMeans(X, K, max_it=500):

    # Initialize cluster means to K samples from data
    rstart = choice(range(X.shape[0]), size=(K), replace=False)
    mu, muold = X[rstart,:], 1000*np.ones(X[rstart,:].shape)

    its = 0
    while its < max_it and np.linalg.norm(mu-muold) > 1e-4 :

        # compute distance b/w each point and centroid
        dist = np.array([[np.linalg.norm(x-m) for m in mu] for x in X])

        # compute new cluster assignments
        z = np.array([np.argmin(d) for d in dist])

        # move centroids
        muold = mu
        mu = np.array([np.mean(X[z==k, :], axis=0) for k in range(K)])
        its += 1

    return mu, z
```

K-means

Weaknesses:

- Usually only converges to local minimum
- Have to determine number of clusters
- Can be sensitive to outliers
- Only generates convex clusters

K-means weaknesses

Usually only converges to local minimum

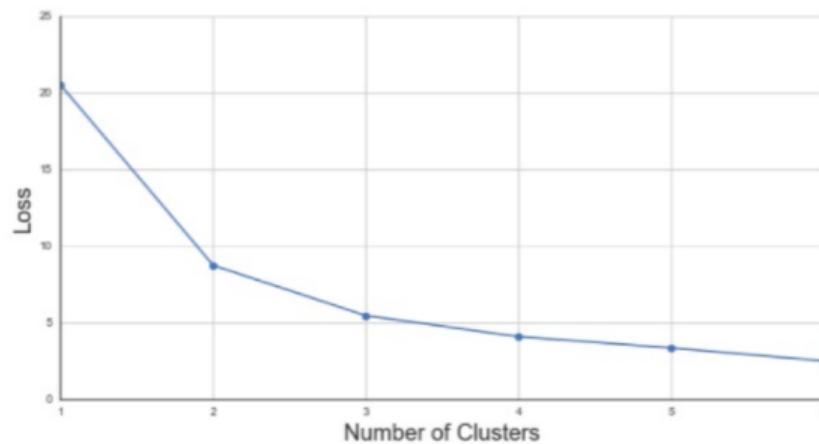
Fix: Do several runs with random initializations and choose best

K-means weaknesses

Have to determine number of clusters

Fix: Use the elbow method.

Run K-Means for different values of k and look at loss function



K-means weaknesses

Sensitive to outliers



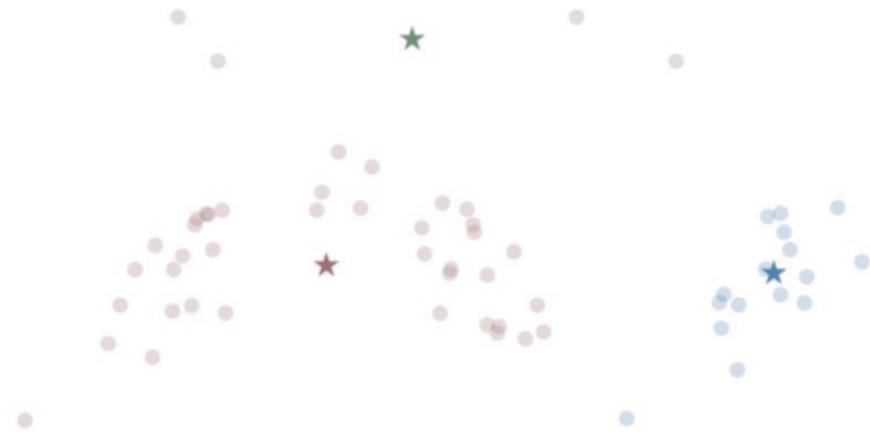
K-means weaknesses

Sensitive to outliers



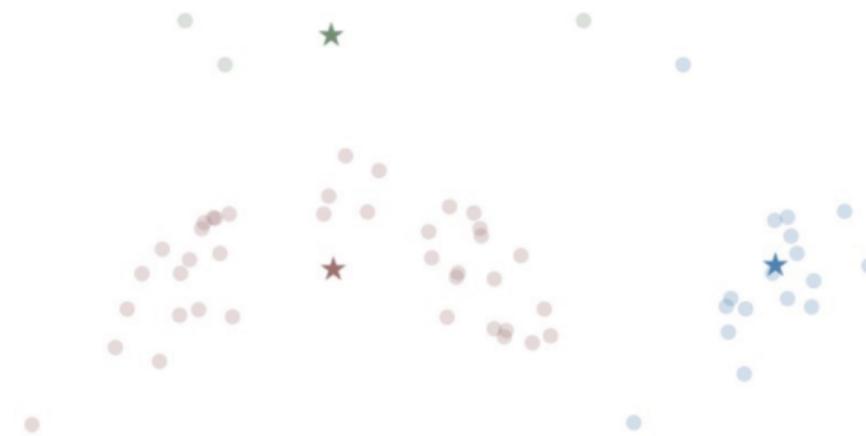
K-means weaknesses

Sensitive to outliers



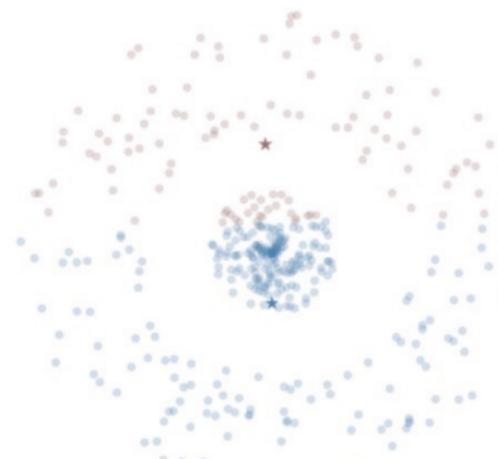
K-means weaknesses

Sensitive to outliers



K-means weaknesses

Convex clusters



Wrap up

- Clustering requires additional assumptions to identify classes
- K-means represent a simple yet strong clustering method