Unsupervised Learning 1
Clustering and Segmentation

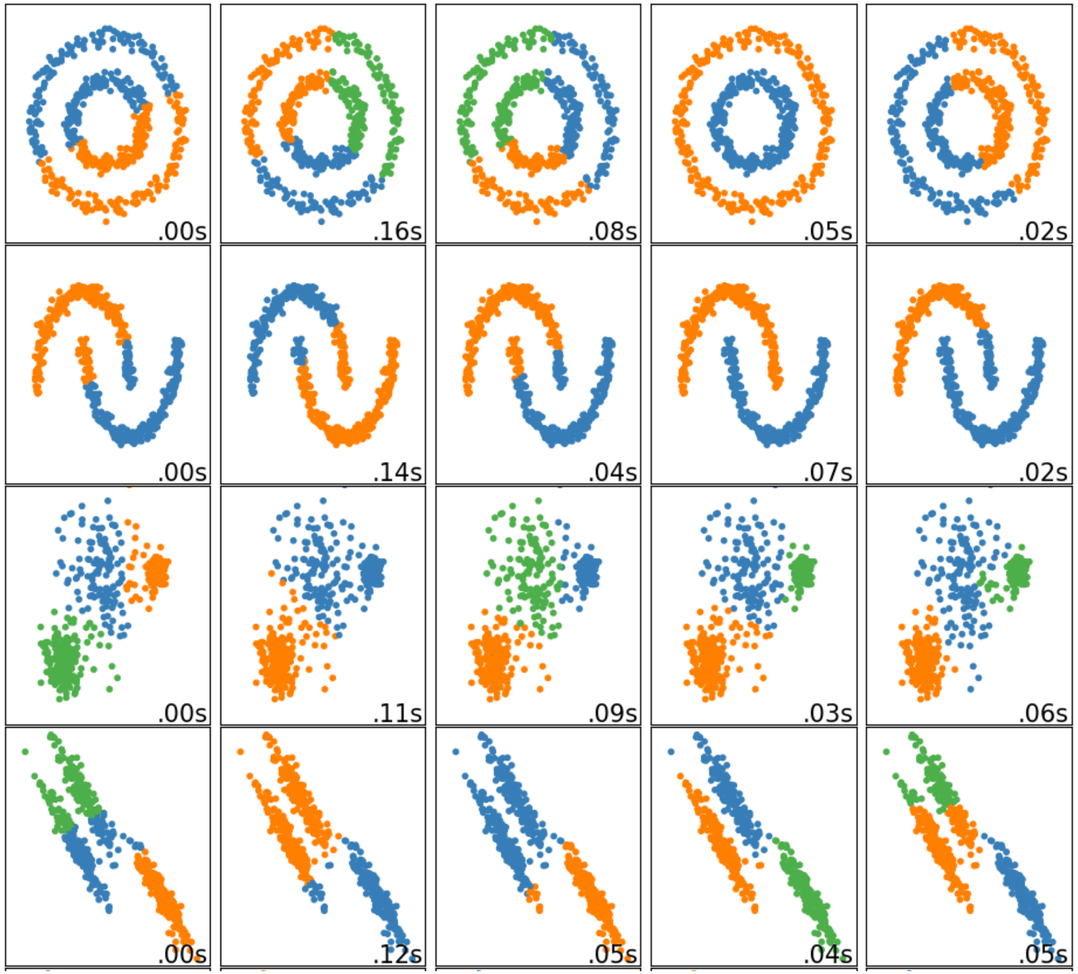
DSCC 251/451: Machine Learning with Limited Data

C.M. Downey

Spring 2026

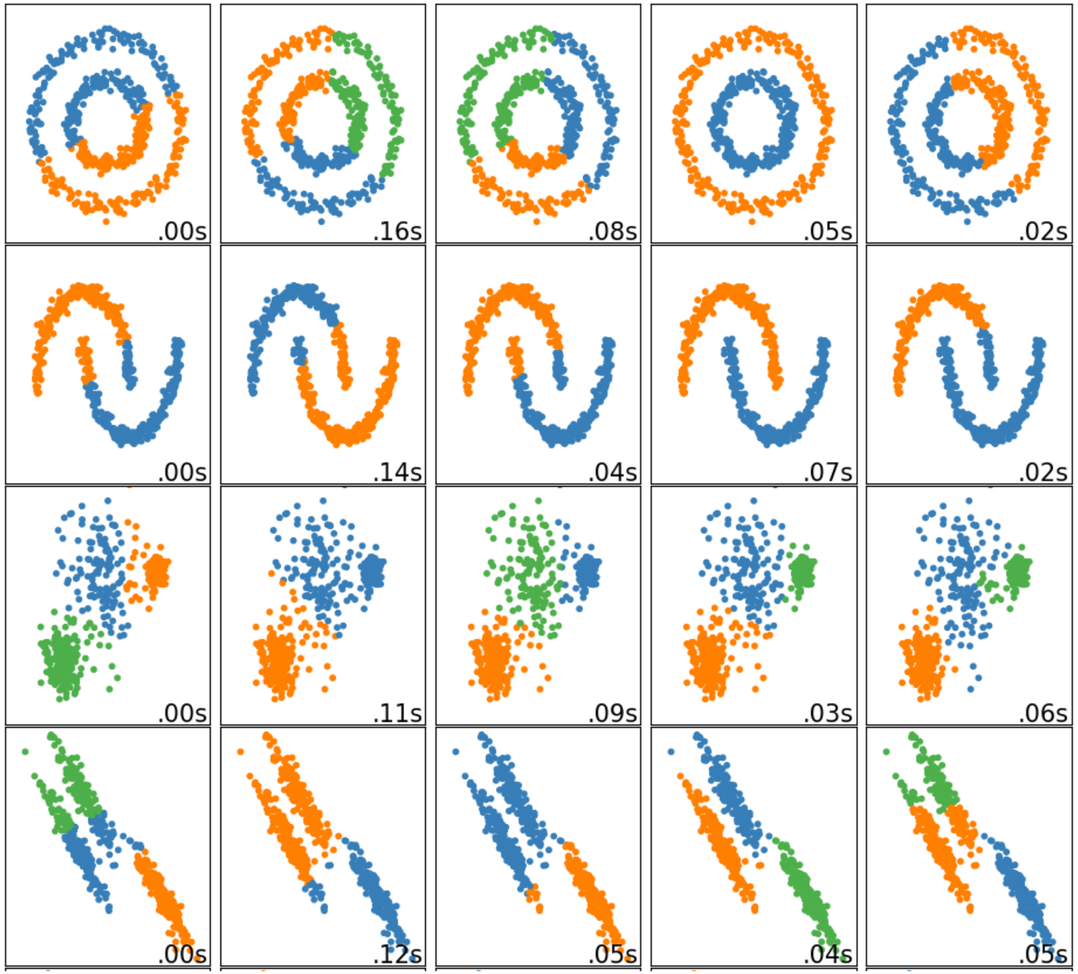
What is Unsupervised Learning?

Unsupervised Learning



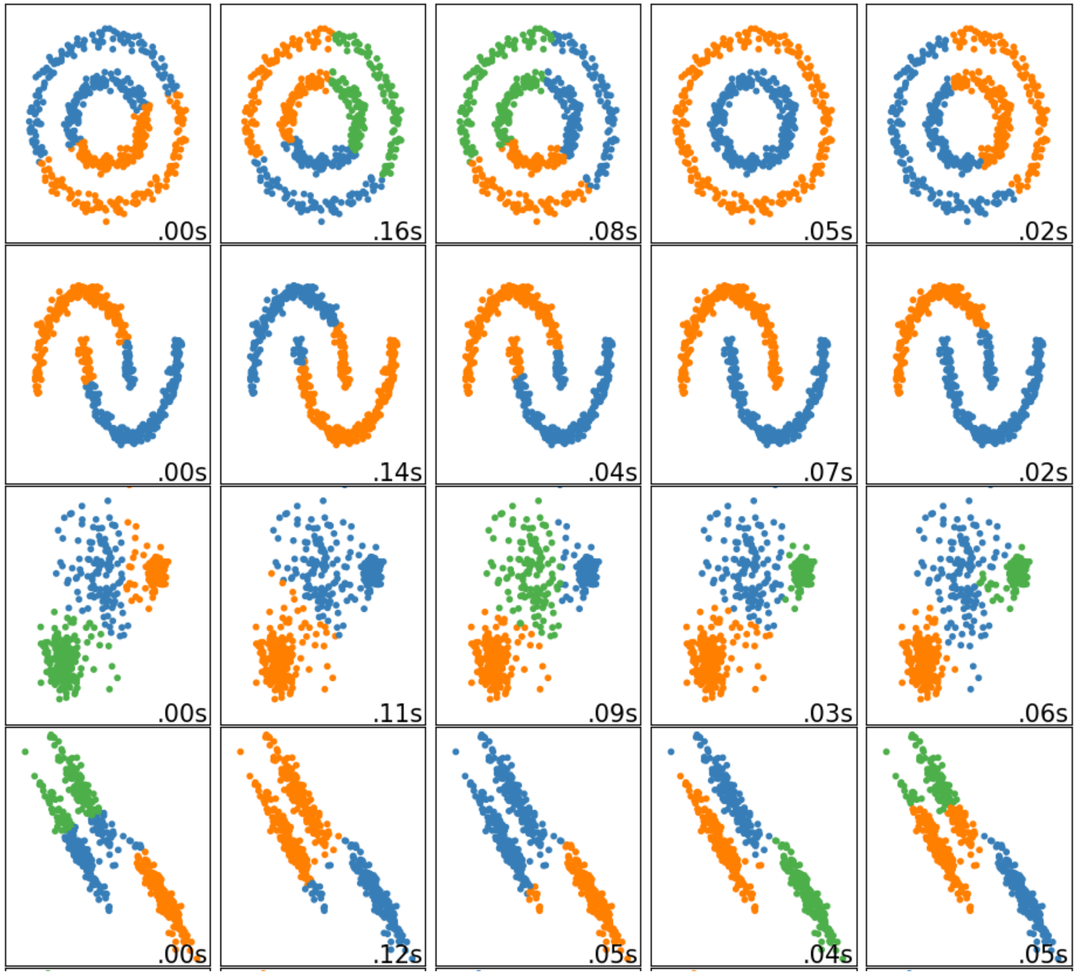
Unsupervised Learning

- **Negative definition:** learning without labels (X but no Y)
  - True, but not very informative



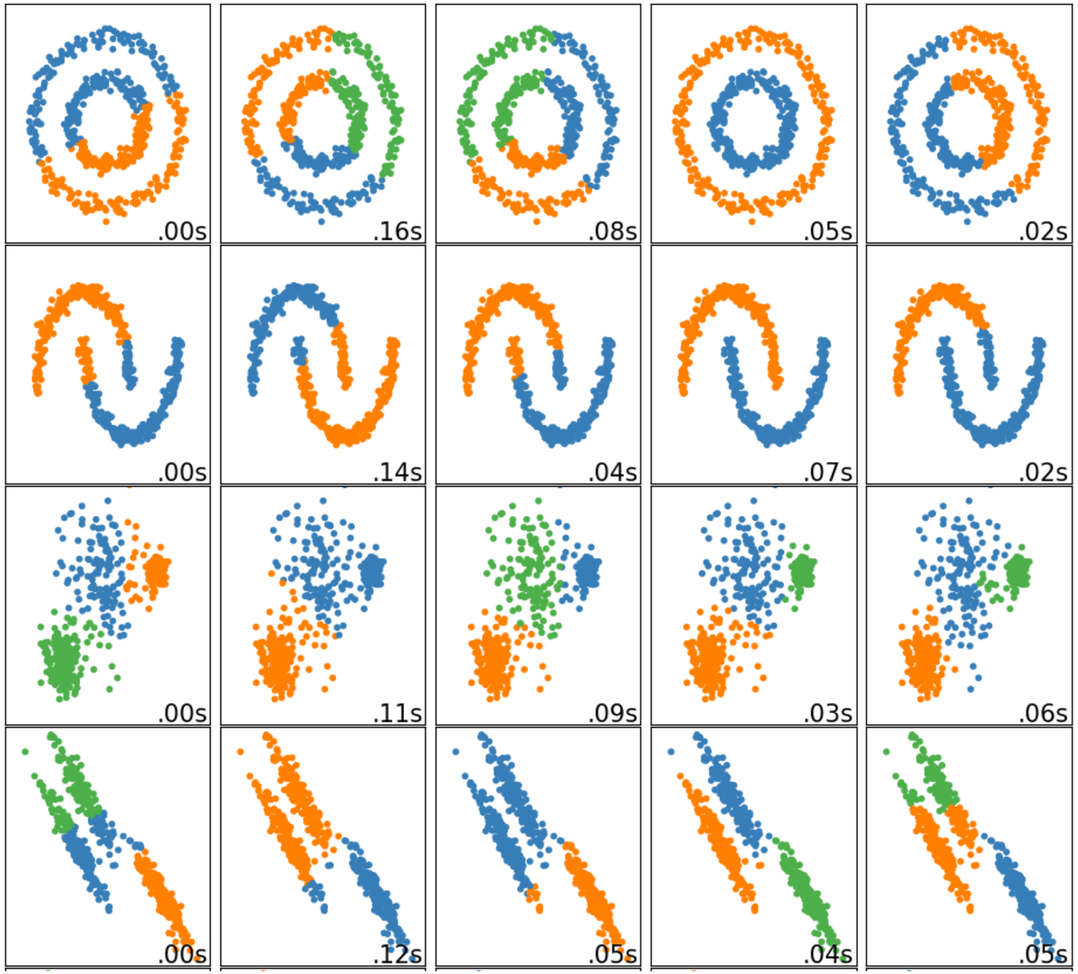
Unsupervised Learning

- **Negative definition:** learning without labels (X but no Y)
  - True, but not very informative
- **Positive definition:** discovering **structure in data**
  - Especially structure that's **useful for tasks** that you **can't directly optimize**



Unsupervised Learning

- **Negative definition:** learning without labels (X but no Y)
  - True, but not very informative
- **Positive definition:** discovering **structure in data**
  - Especially structure that's **useful for tasks** that you **can't directly optimize**
- Why do unsupervised learning? Robust labeled data is **scarce/expensive**
  - But **structure still exists** in unlabeled data
  - If we can model it, we **reduce dependence on labels**



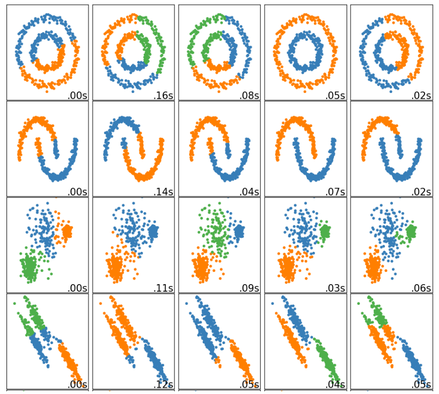
Types of Unsupervised Learning

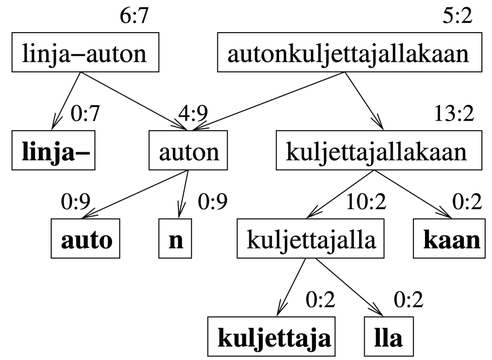
Types of Unsupervised Learning

- We'll see **two distinct "flavors"**

Types of Unsupervised Learning

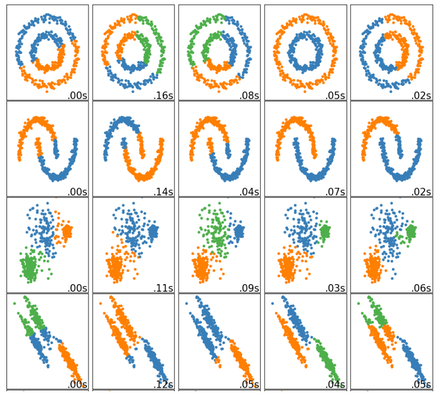
- We'll see **two distinct "flavors"**
- Today: discovering **discrete/categorical structure**
  - I.e. groupings, segments, boundaries, categories
  - **Approximates outputs you don't actually have**
  - Examples: clustering, segmentation, quantization

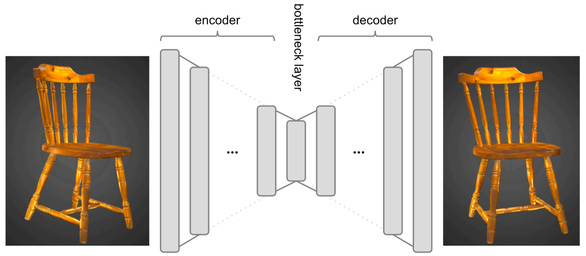


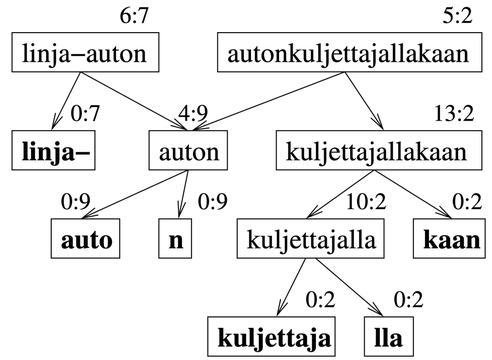


Types of Unsupervised Learning

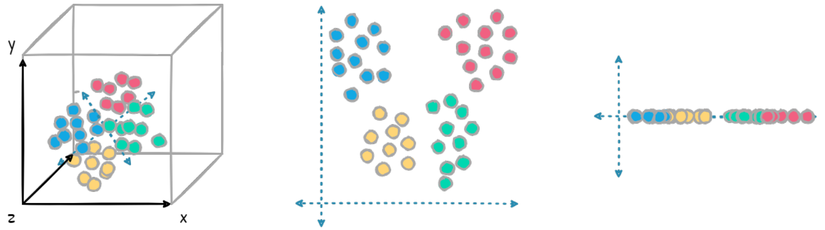
- We'll see **two distinct "flavors"**
- Today: discovering **discrete/categorical structure**
  - I.e. groupings, segments, boundaries, categories
  - **Approximates outputs you don't actually have**
  - Examples: clustering, segmentation, quantization
- Next time: discovering **continuous structure**
  - I.e. representations, projections, embeddings
  - Typically used as **inputs to downstream tasks**
  - Examples: dimensionality reduction, autoencoders



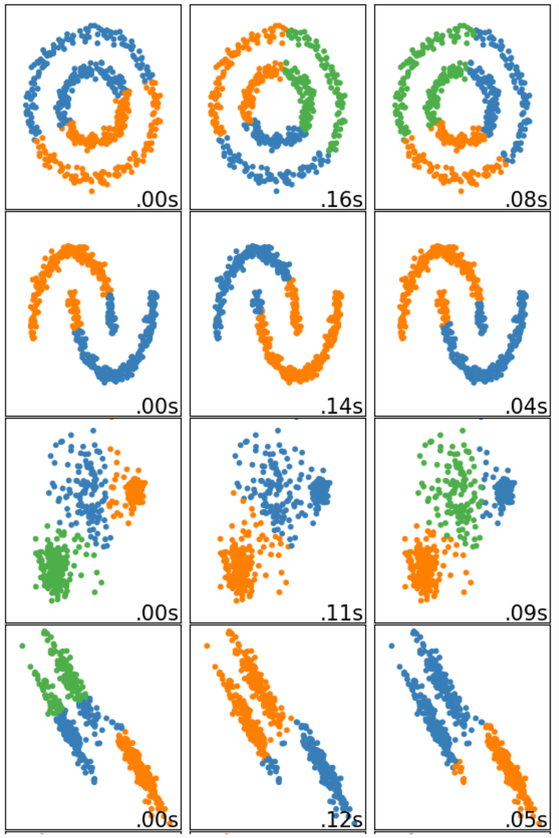




Dimensionality Reduction

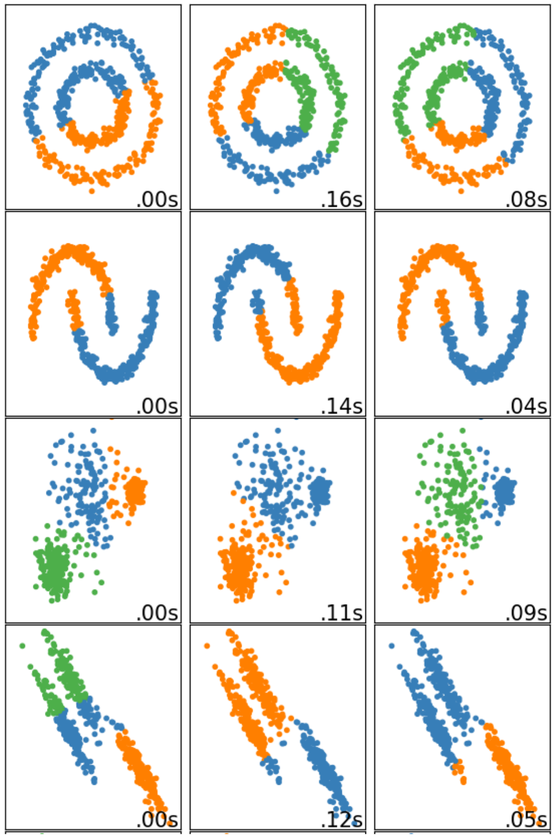


The Unsupervised Pattern



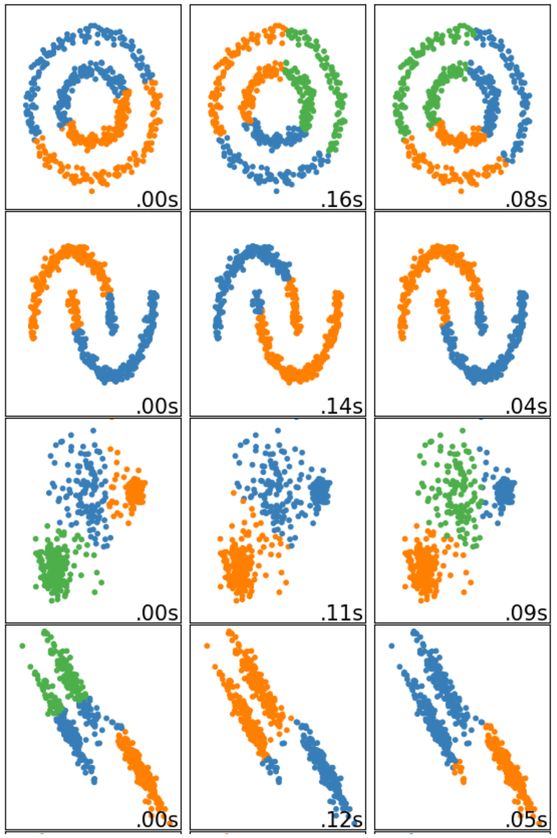
The Unsupervised Pattern

- For both flavors, you **can't optimize the real objective** (i.e. labels, outputs)
  - You often optimize a **surrogate objective** instead
  - You then **hope/hypothesize** that the surrogate **correlates with the real objective**



The Unsupervised Pattern

- For both flavors, you **can't optimize the real objective** (i.e. labels, outputs)
  - You often optimize a **surrogate objective** instead
  - You then **hope/hypothesize** that the surrogate **correlates with the real objective**
- "Cake" connection: unsupervised learning is the **body of the cake**
  - i.e. the **bulk of what we can learn**
  - Supervised learning is only the icing



Surrogate Objectives

Surrogate Objectives

- Situation: you want to **map raw data** to **meaningful categories/decisions**
  - But you **don't have labels** to learn that mapping directly

Surrogate Objectives

- Situation: you want to **map raw data** to **meaningful categories/decisions**
  - But you **don't have labels** to learn that mapping directly
- Solution: optimize for something that you **can learn/measure**
  - Should be something that **correlates with your real goal**
  - Accept that the **correlation is not guaranteed**

Surrogate Objectives

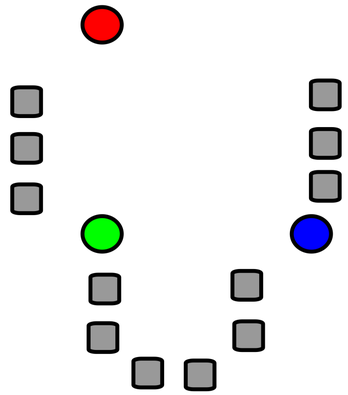
- Situation: you want to **map raw data** to **meaningful categories/decisions**
  - But you **don't have labels** to learn that mapping directly
- Solution: optimize for something that you **can learn/measure**
  - Should be something that **correlates with your real goal**
  - Accept that the **correlation is not guaranteed**
- Example: business thinks it has distinct **"categories" of customers**
  - How do you **discover** those categories from raw buying data?
  - Maybe **clustering:** try to find groups of data that are **distinct from each other**

K-Means Clustering

K-Means Clustering (overview)

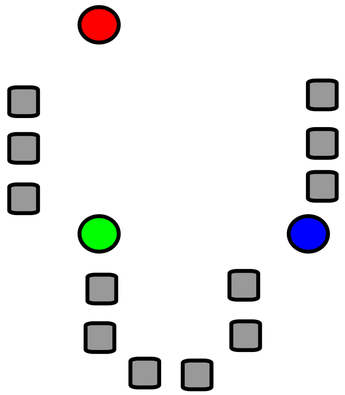
K-Means Clustering (overview)

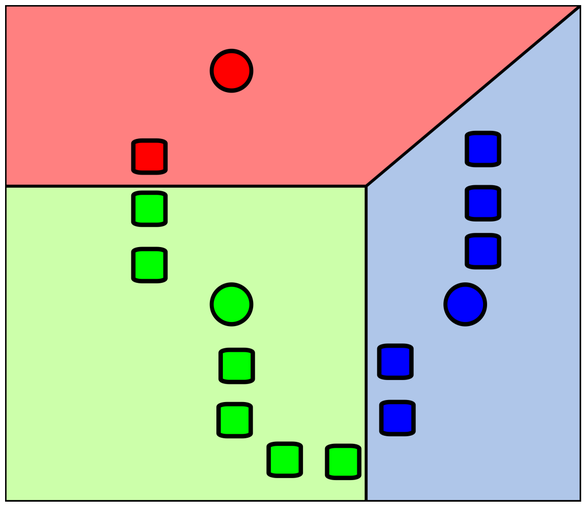
1. initialize "centroids" randomly in data space



K-Means Clustering (overview)

1. initialize "centroids" randomly in data space

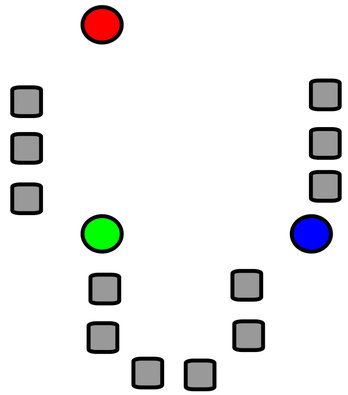


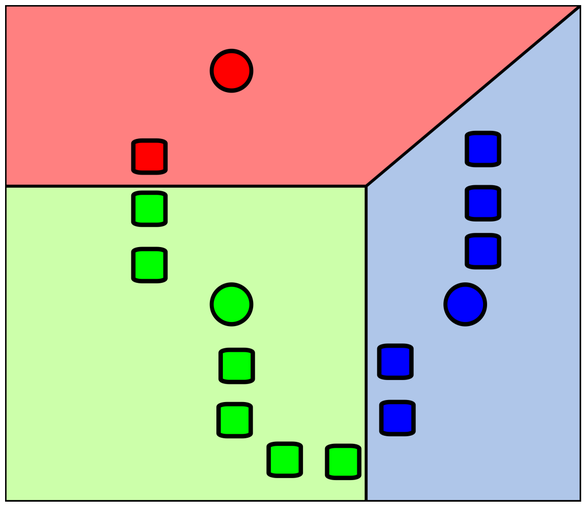


2. assign each point to **nearest centroid**

K-Means Clustering (overview)

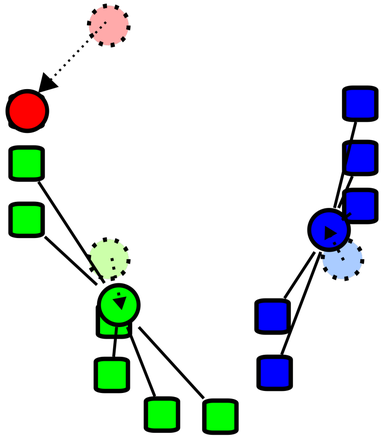
1. initialize "centroids" randomly in data space





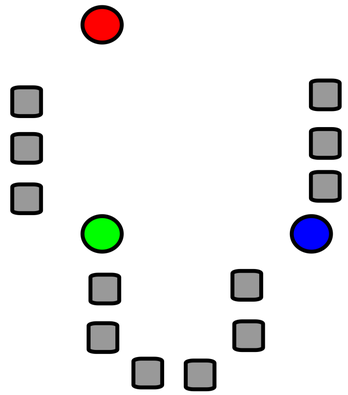
2. assign each point to **nearest centroid**

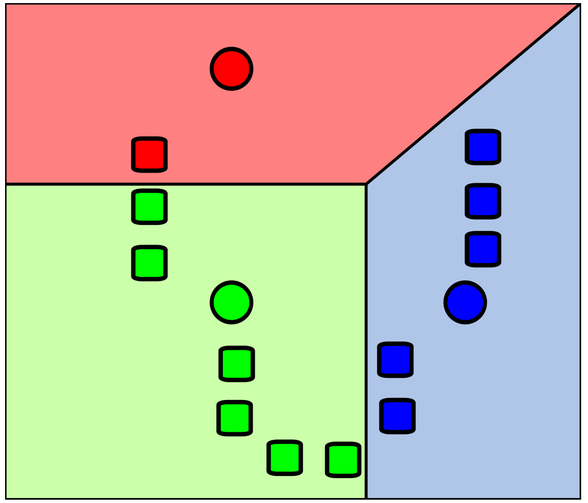
3. shift the centroids to be the **mean** of their points



K-Means Clustering (overview)

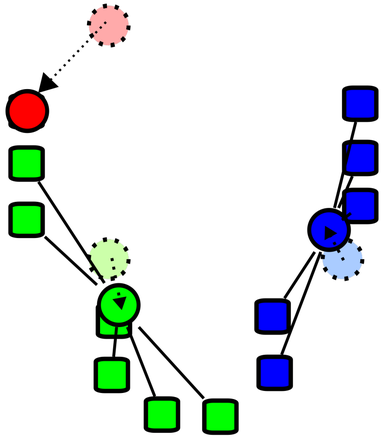
1. initialize "centroids" randomly in data space

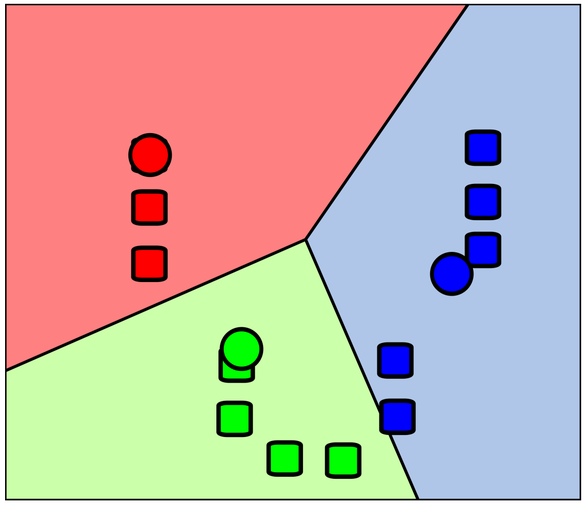




2. assign each point to **nearest centroid**

3. shift the centroids to be the **mean** of their points

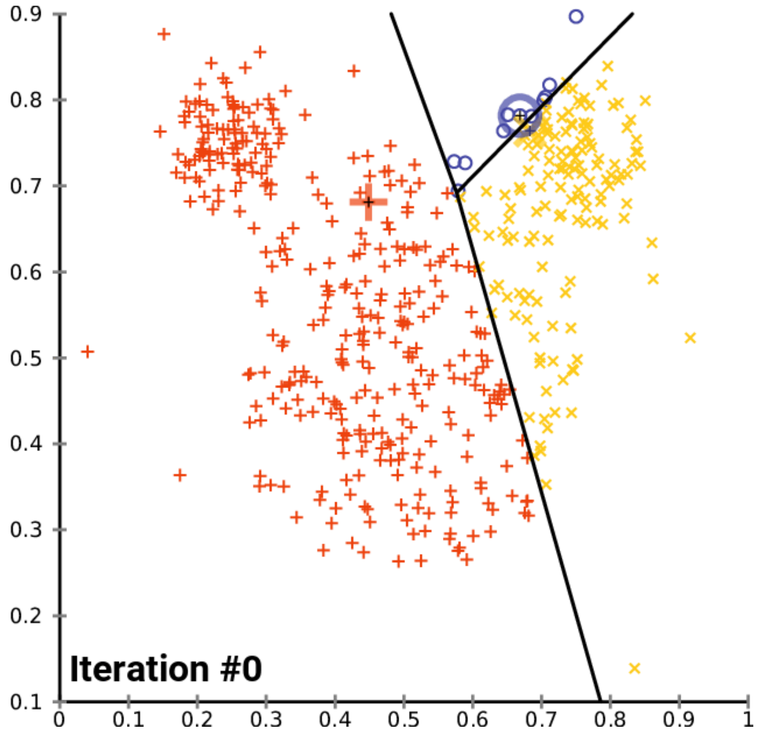




4. **re-assign** points to nearest centroid, and **repeat**

K-Means Formally

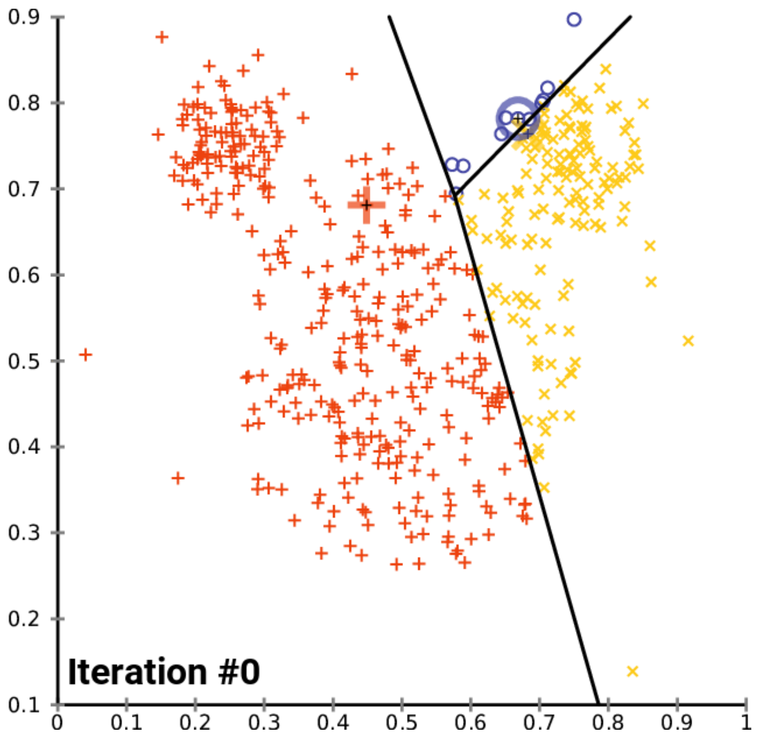
$$\sum_{i} \sum_{x \in C_i} ||x - \mu_i||^2$$



K-Means Formally

- This process minimizes the **Within-Cluster Sum of Squares** (right)
  - I.e. minimize the **total distance from points to centroids**
  - This is the **surrogate objective**

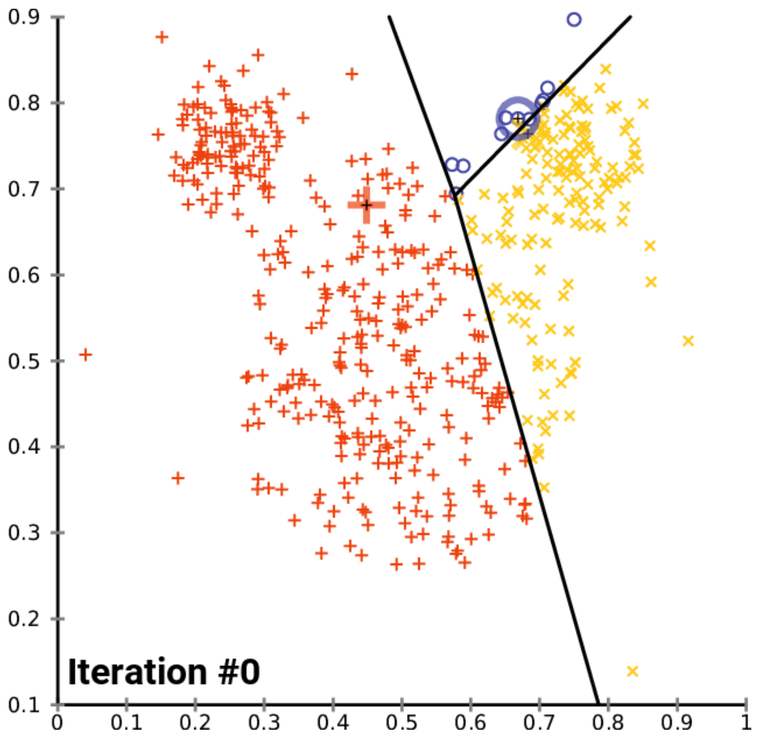
$$\sum_{i} \sum_{x \in C_i} ||x - \mu_i||^2$$



K-Means Formally

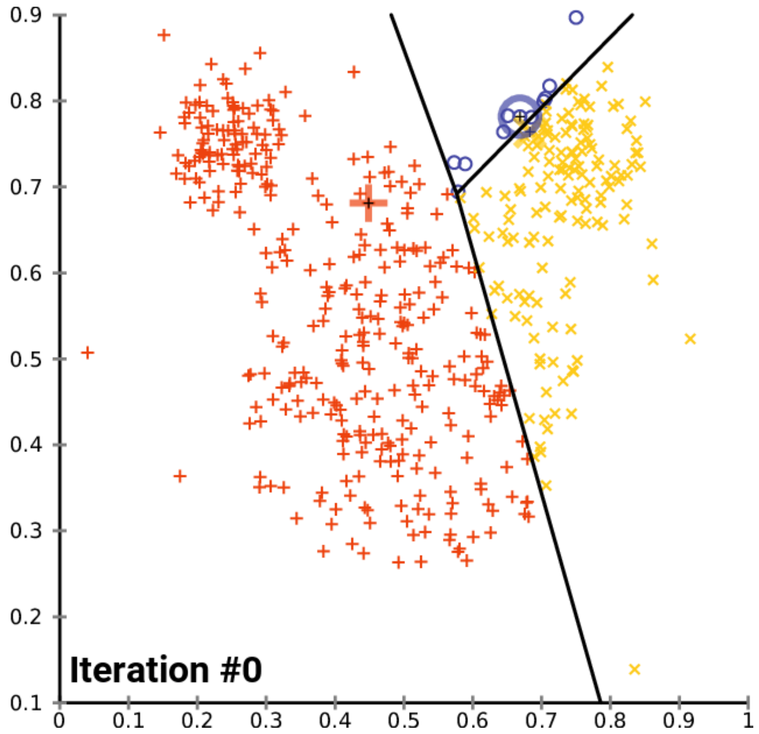
- This process minimizes the **Within-Cluster Sum of Squares** (right)
  - I.e. minimize the **total distance from points to centroids**
  - This is the **surrogate objective**
- Reasoning: points **near each other** are **probably similar**, and thus may be the **same class**
  - This introduces **inductive biases!**

$$\sum_{i} \sum_{x \in C_i} ||x - \mu_i||^2$$



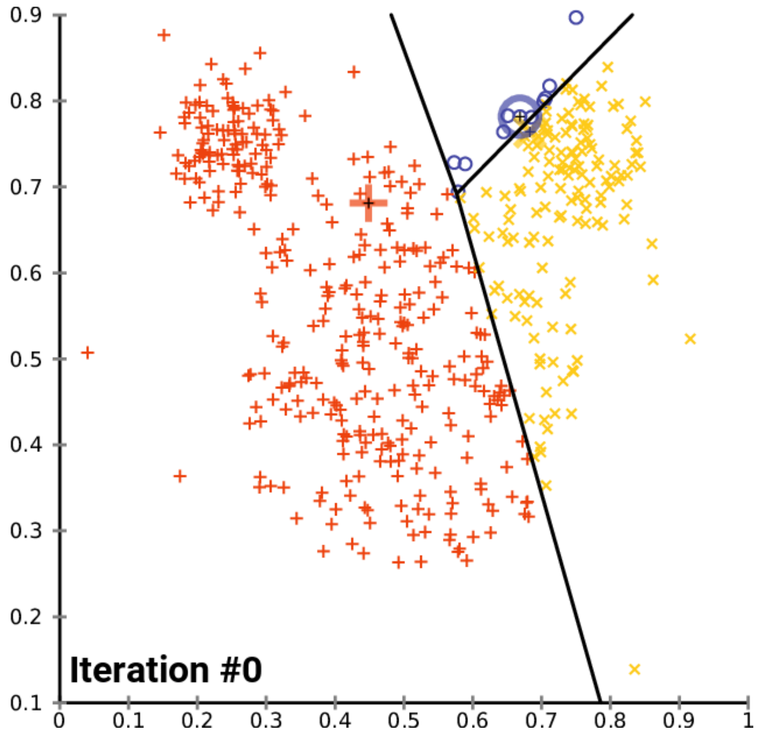
Inductive Biases

$$\sum_{i} \sum_{x \in C_i} ||x - \mu_i||^2$$



Inductive Biases

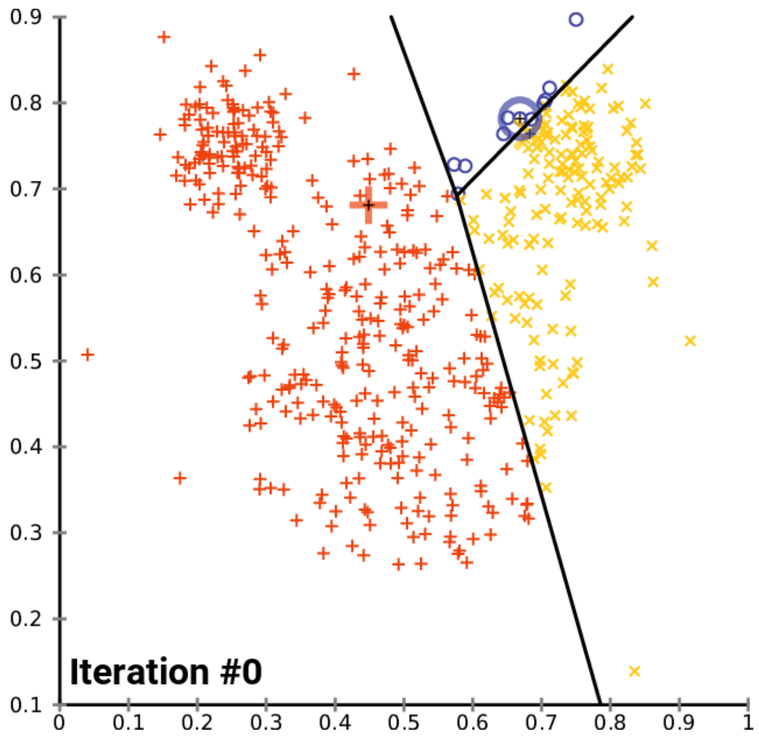
$$\sum_{i} \sum_{x \in C_i} ||x - \mu_i||^2$$



Inductive Biases

- Tacitly assumes clusters are **spherical** (equal direction variance)

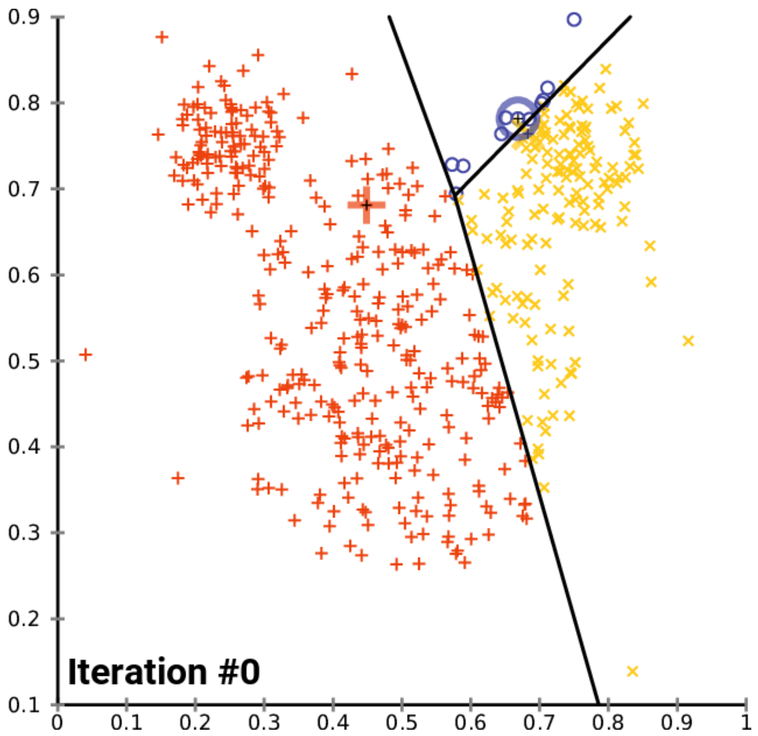
$$\sum_{i} \sum_{x \in C_i} ||x - \mu_i||^2$$



Inductive Biases

- Tacitly assumes clusters are **spherical** (equal direction variance)
- Assumes clusters are **similar size**

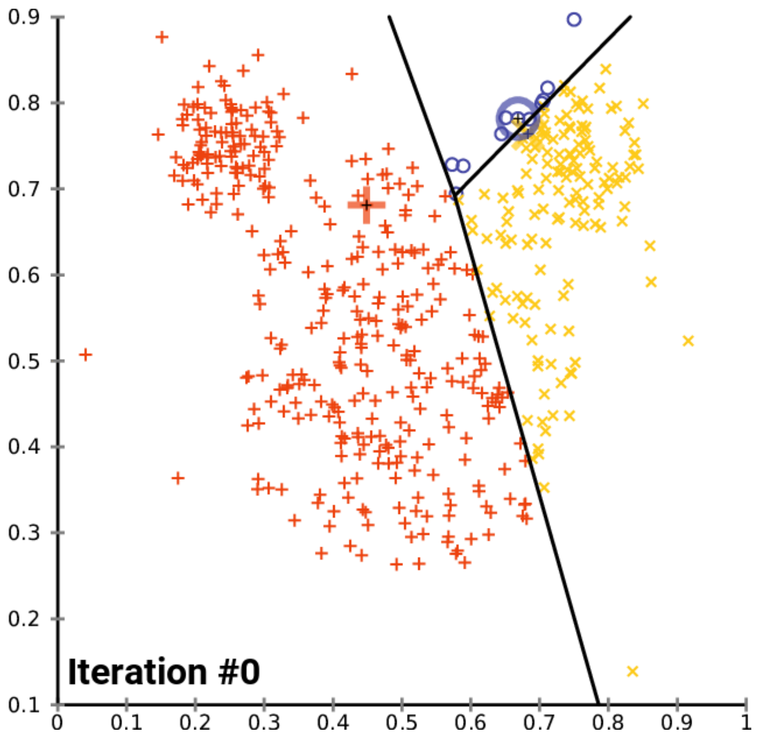
$$\sum_i \sum_{x \in C_i} ||x - \mu_i||^2$$



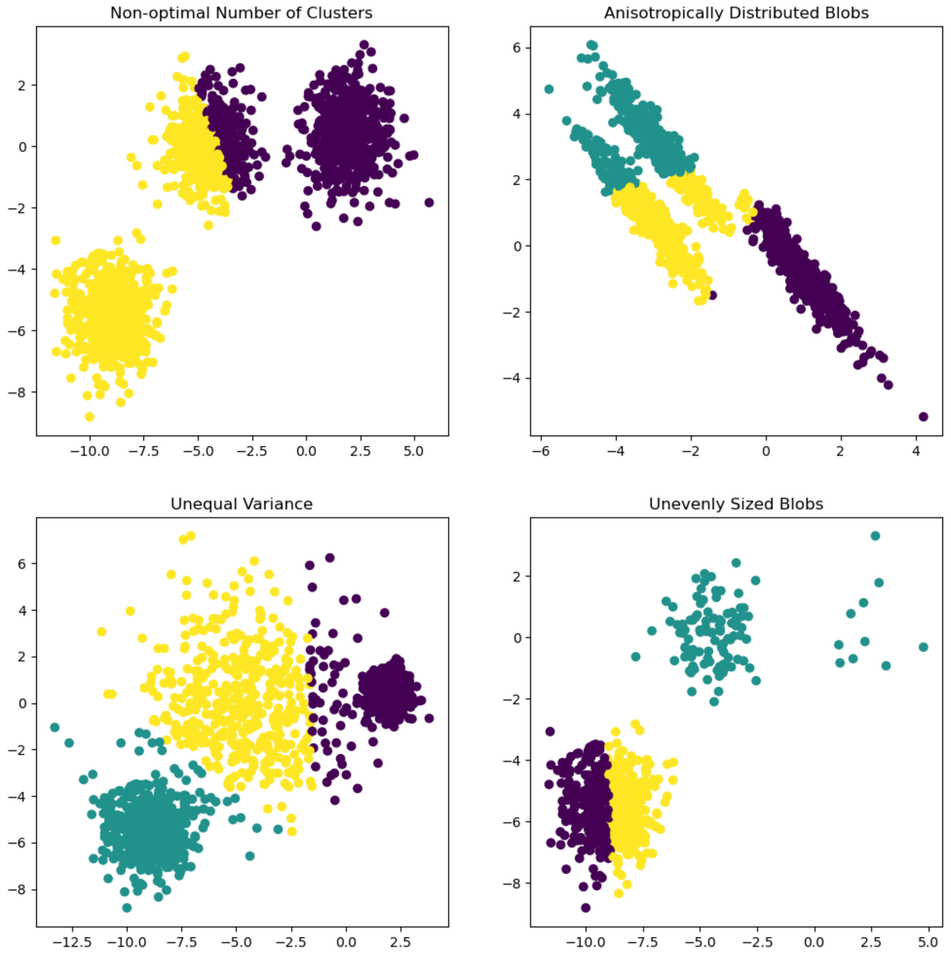
Inductive Biases

- Tacitly assumes clusters are **spherical** (equal direction variance)
- Assumes clusters are **similar size**
- Big one: assumes you know the **number of clusters in advance!**
  - The "K" in K-Means
  - Usually **unlikely**; but there are ways to select the **optimal number**

$$\sum_{i} \sum_{x \in C_i} ||x - \mu_i||^2$$

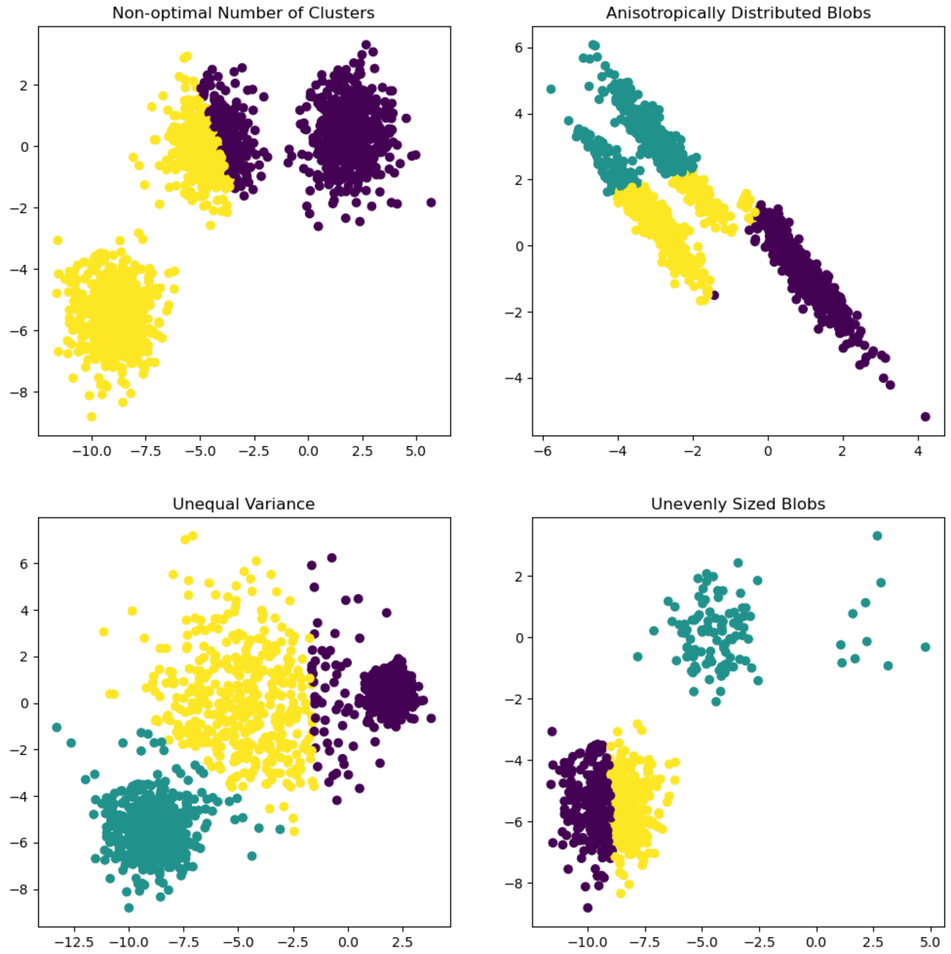


K-Means Failure Cases



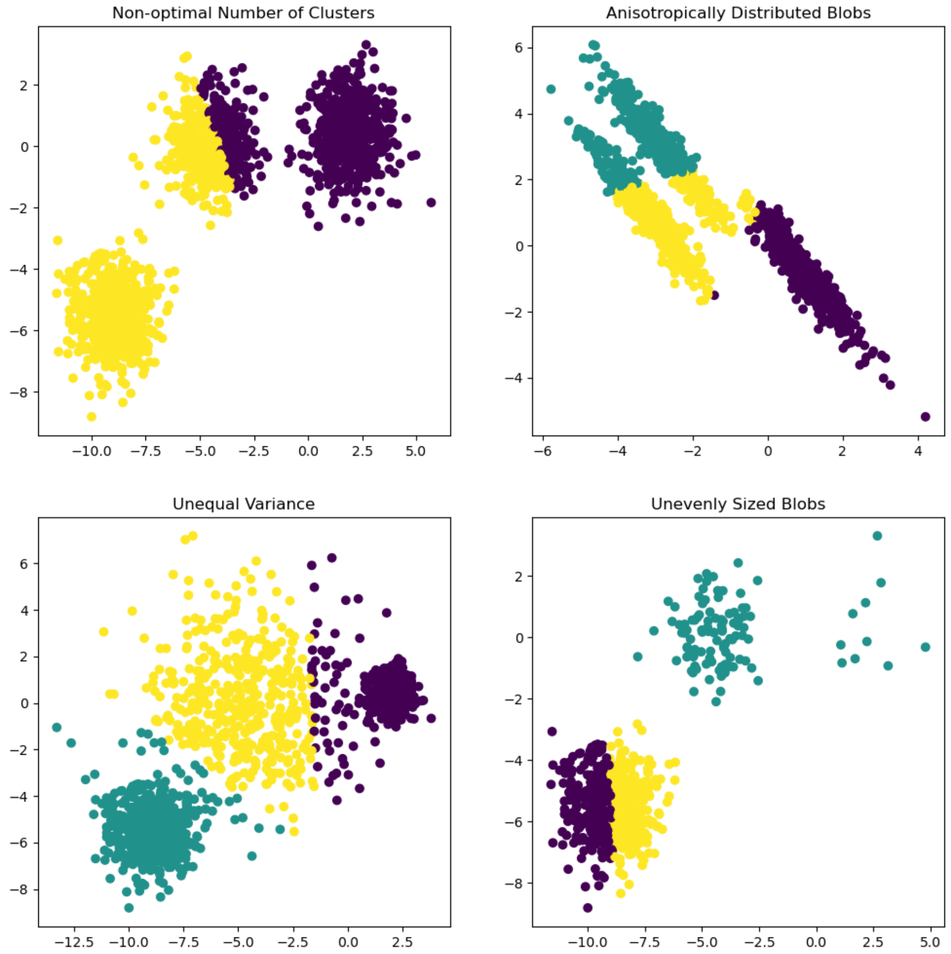
K-Means Failure Cases

- Here are some **common failure cases**



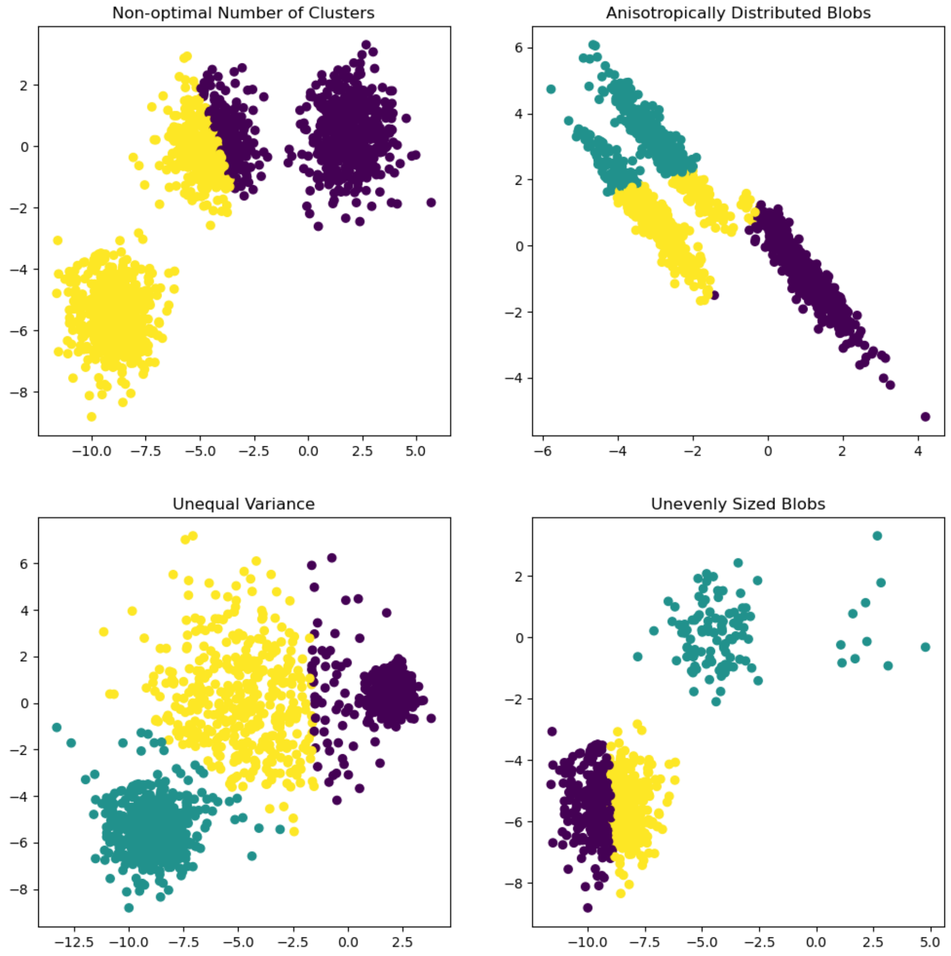
K-Means Failure Cases

- Here are some **common failure cases**
  - **Non-optimal K** (wrong number of clusters)



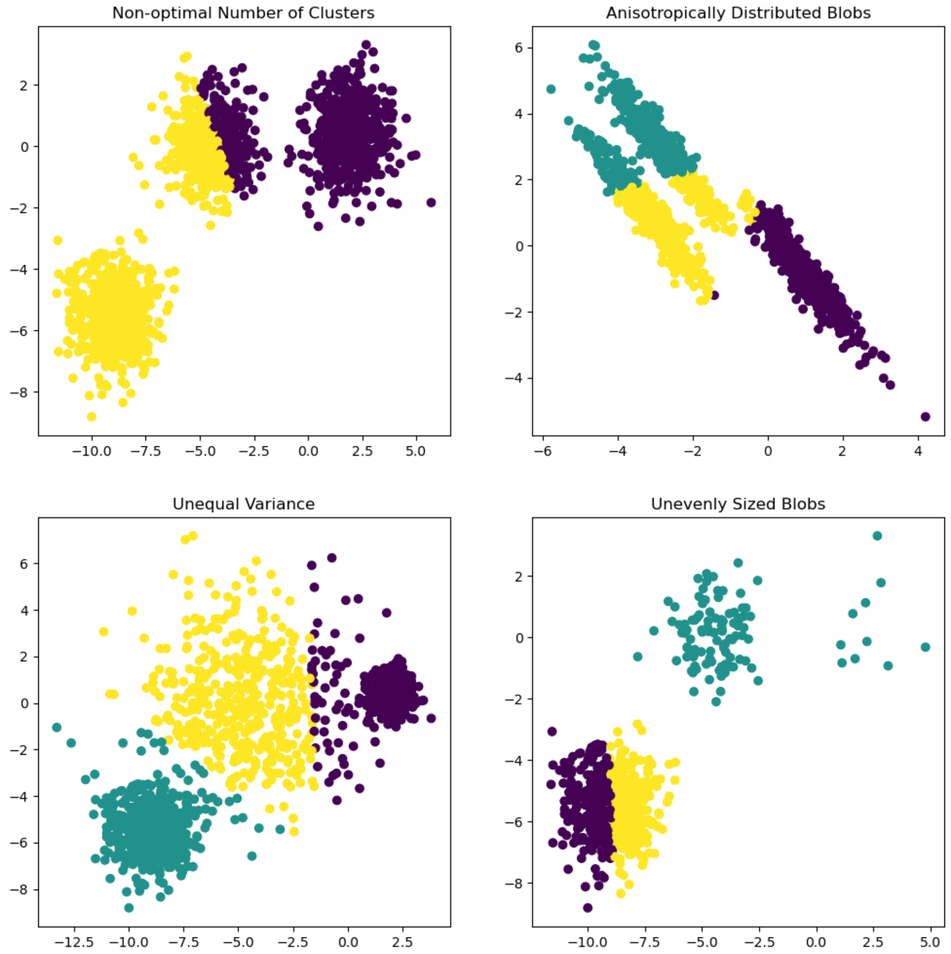
K-Means Failure Cases

- Here are some **common failure cases**
  - **Non-optimal K** (wrong number of clusters)
  - **Elongated blobs** (dimensions have covariance effects)



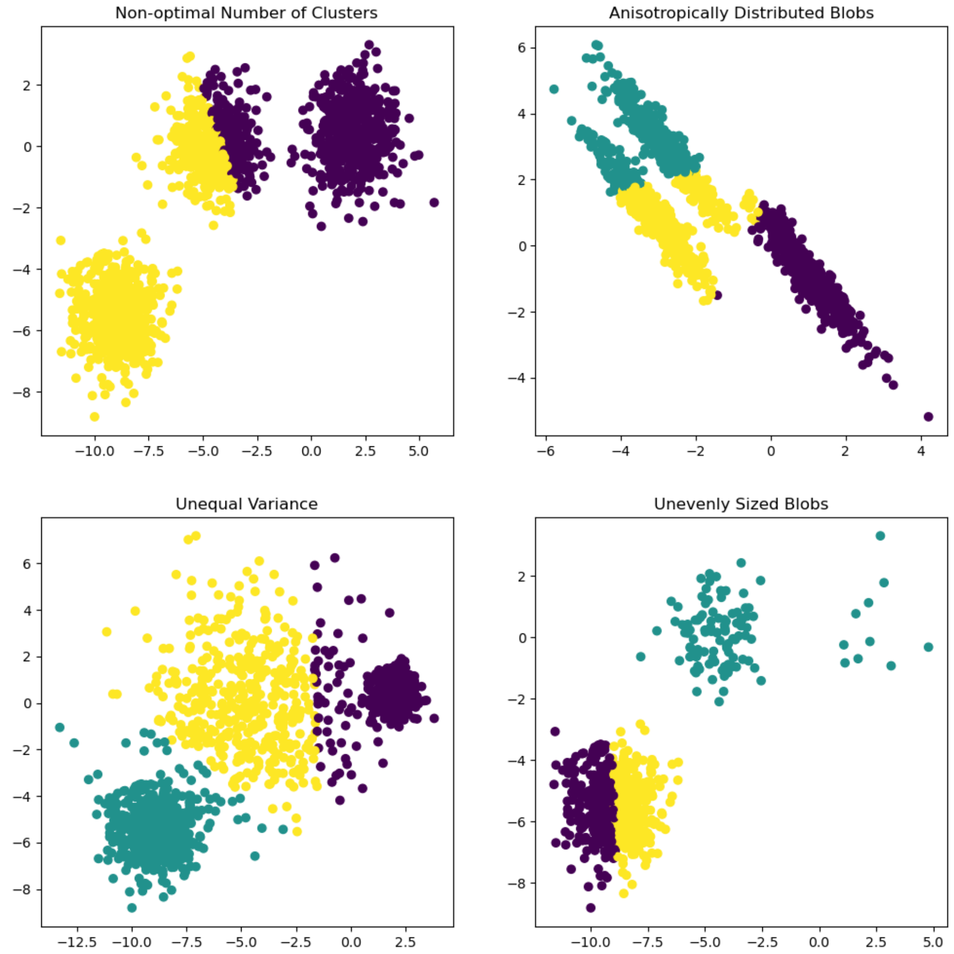
K-Means Failure Cases

- Here are some **common failure cases**
  - **Non-optimal K** (wrong number of clusters)
  - **Elongated blobs** (dimensions have covariance effects)
  - **Unequal variance** (clusters have different spreads)



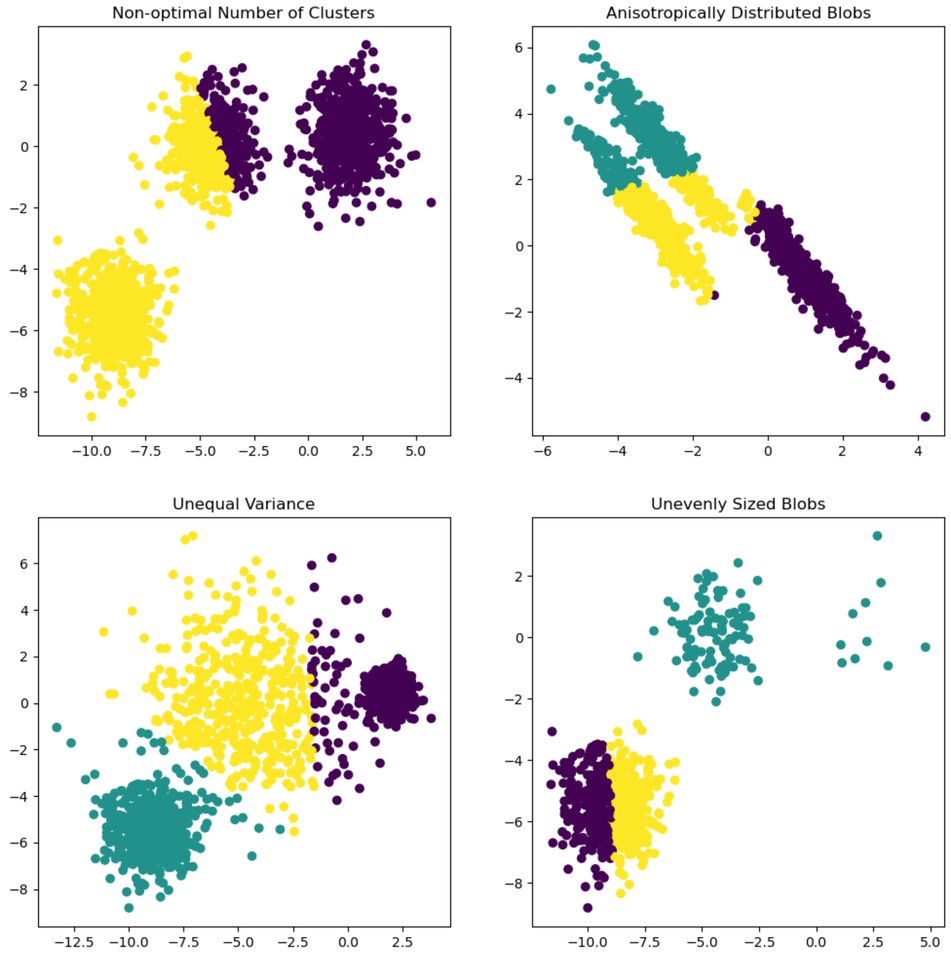
K-Means Failure Cases

- Here are some **common failure cases**
  - **Non-optimal K** (wrong number of clusters)
  - **Elongated blobs** (dimensions have covariance effects)
  - **Unequal variance** (clusters have different spreads)
  - **Uneven size** (different number of data points)



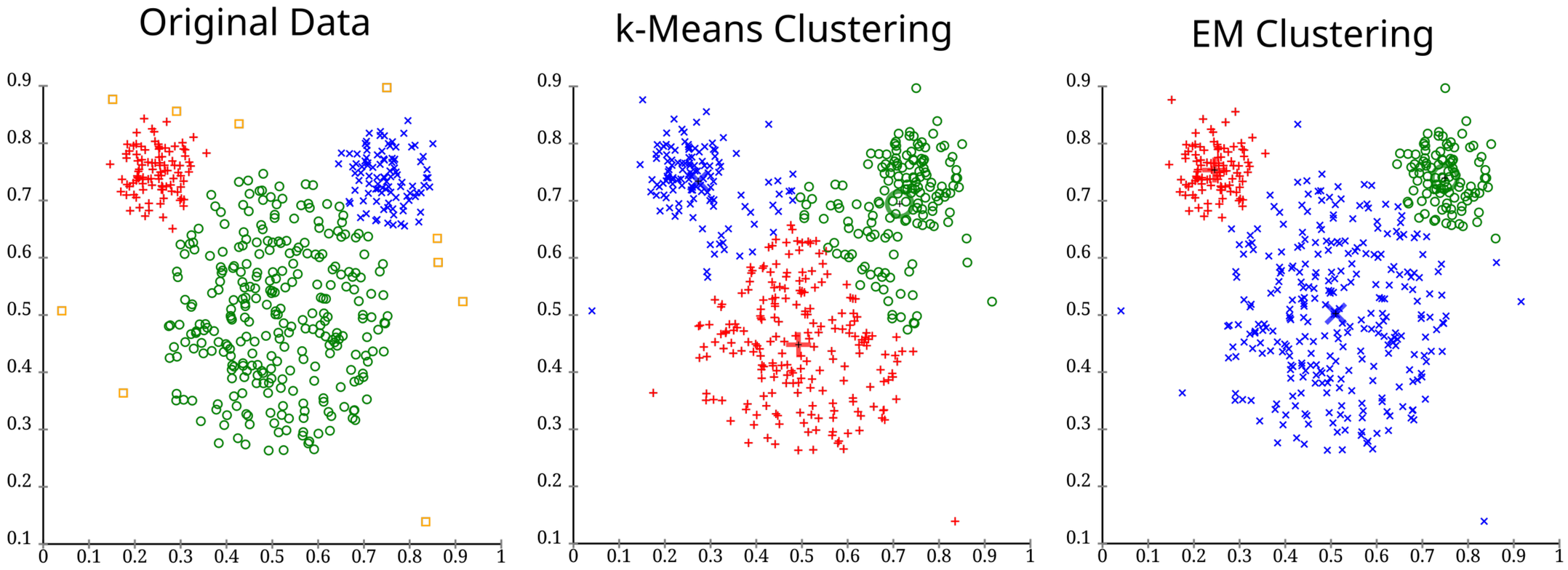
K-Means Failure Cases

- Here are some **common failure cases**
  - **Non-optimal K** (wrong number of clusters)
  - **Elongated blobs** (dimensions have covariance effects)
  - **Unequal variance** (clusters have different spreads)
  - **Uneven size** (different number of data points)
- All of these demonstrate **inductive biases**!



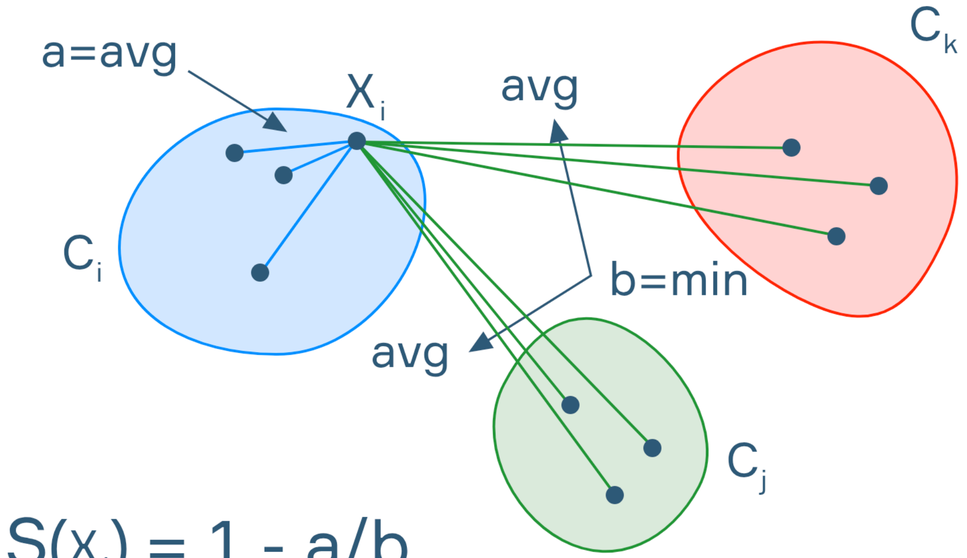
"Mouse" Dataset

Different cluster analysis results on "mouse" data set:



Evaluating Clusters

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

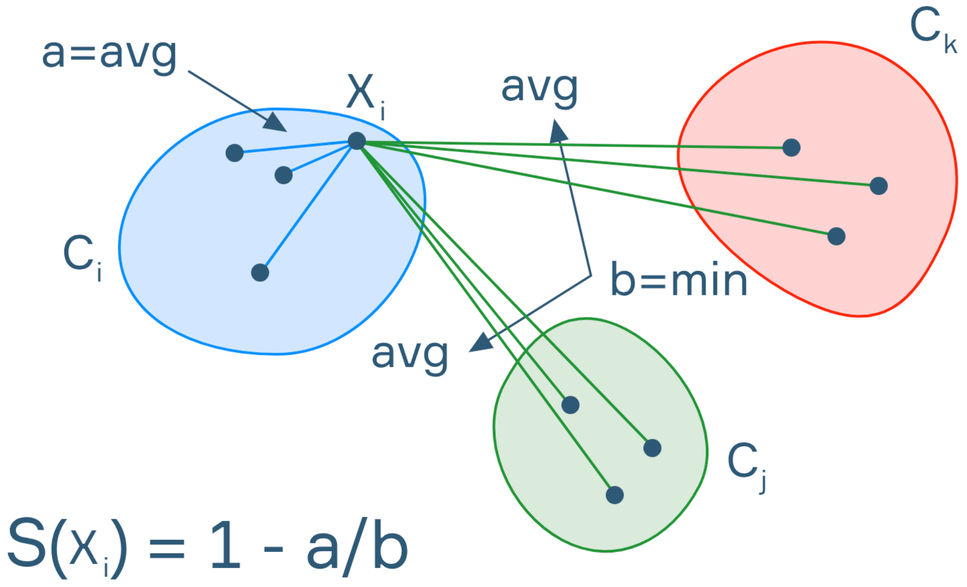


$S(x_i) = 1 - a/b$

Evaluating Clusters

- How do we know how "good" the clusters are? **Another surrogate metric!**

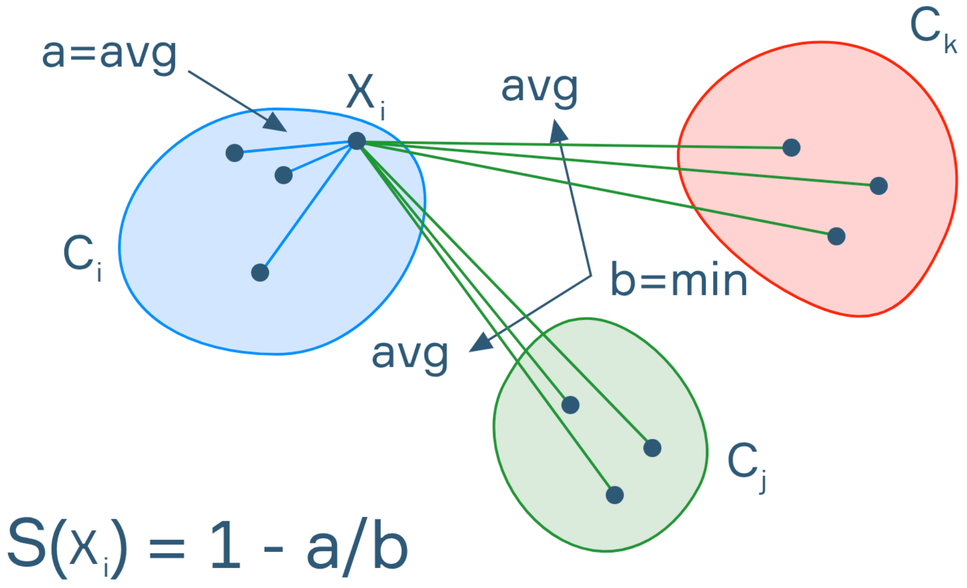
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$



Evaluating Clusters

- How do we know how "good" the clusters are? **Another surrogate metric!**
- Common method uses the **Silhouette Score** (right)
  - $a(i)$: average distance to **other points** in the **same cluster**
  - $b(i)$: average distance to points in the **nearest other cluster**
  - Range: $[-1,1]$ (**higher** is better)

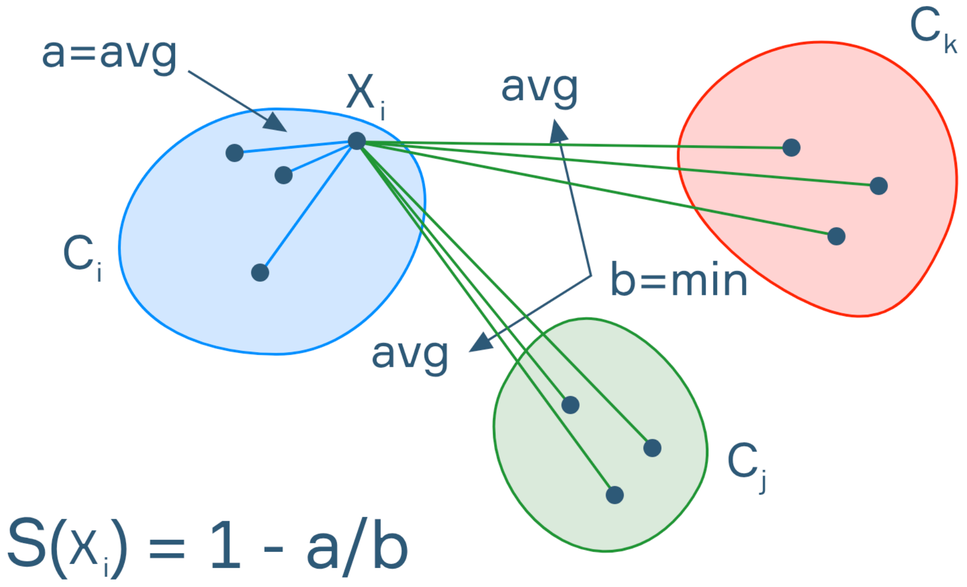
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$



Evaluating Clusters

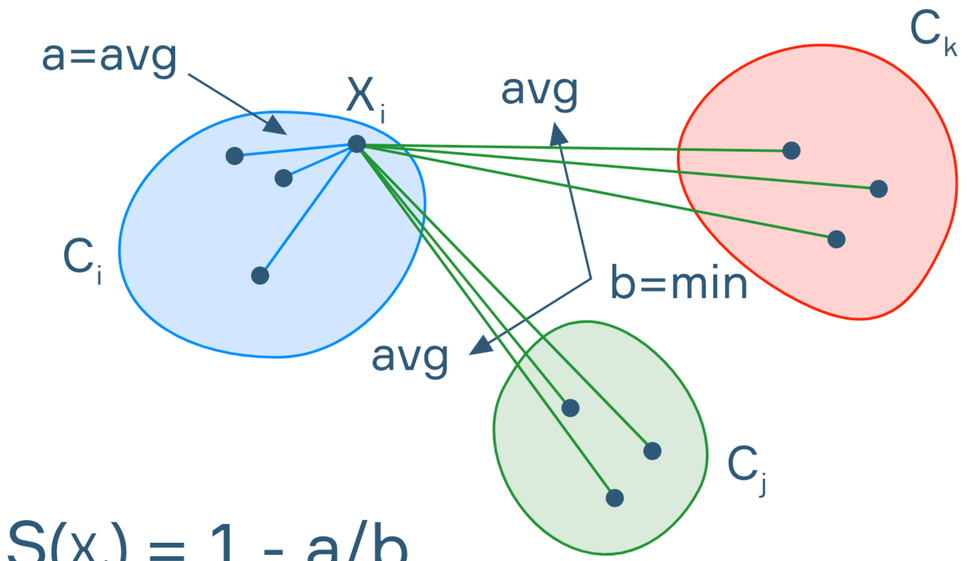
- How do we know how "good" the clusters are? **Another surrogate metric!**
- Common method uses the **Silhouette Score** (right)
  - $a(i)$: average distance to **other points** in the **same cluster**
  - $b(i)$: average distance to points in the **nearest other cluster**
  - Range: $[-1,1]$ (**higher** is better)
- How **well-separated** are the clusters?

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$



Evaluating Clusters

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

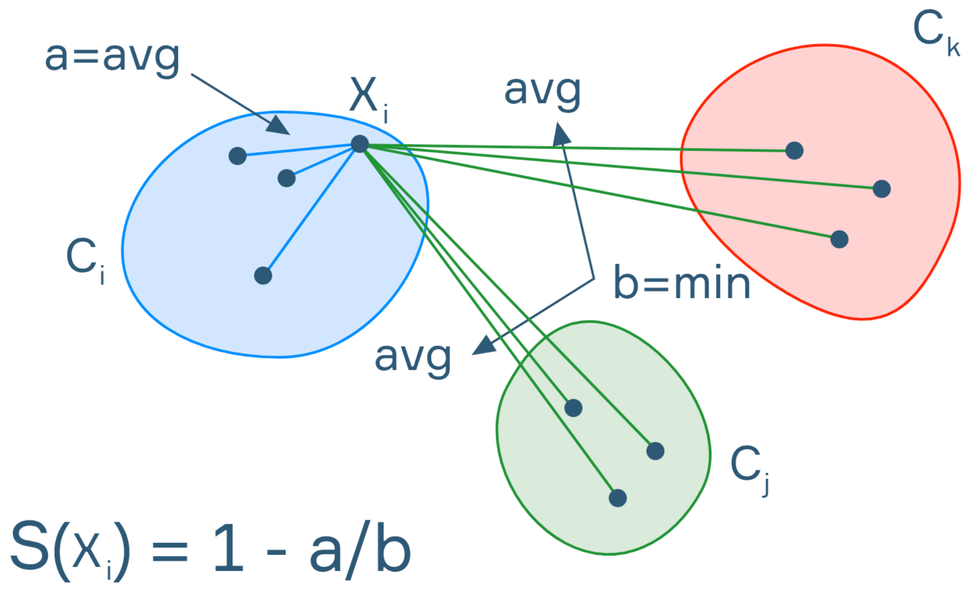


$S(x_i) = 1 - a/b$

Evaluating Clusters

- Can use Silhouette to determine the **optimal number** of clusters

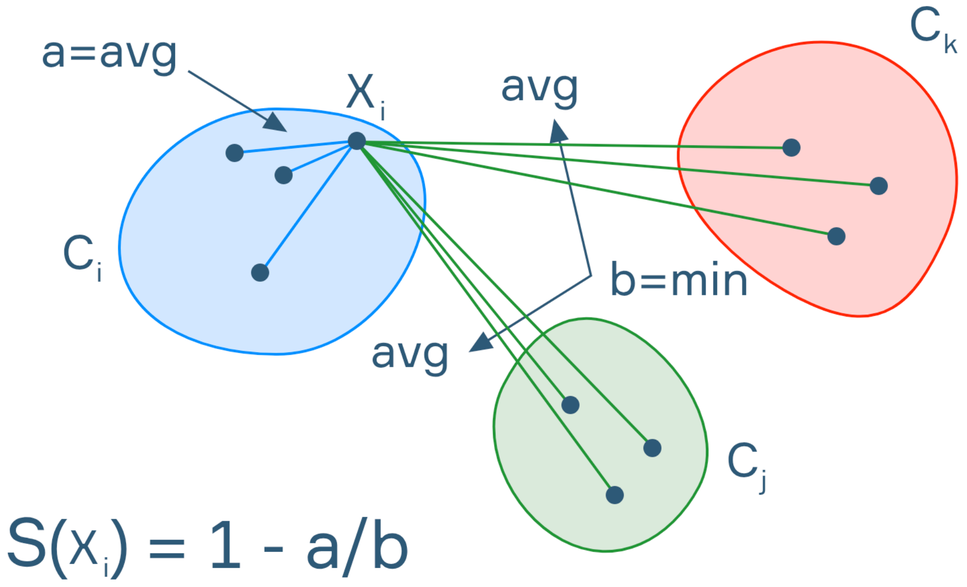
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$



Evaluating Clusters

- Can use Silhouette to determine the **optimal number** of clusters
- Run K-Means for $K = 2,3,4...$
  - Computer Silhouette score for each
  - Chose K with the **highest score**

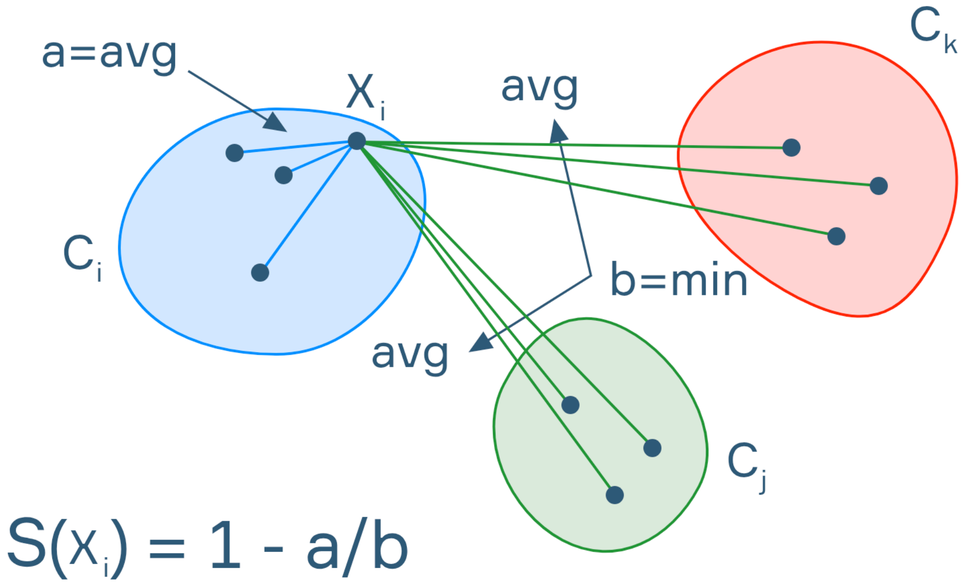
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$



Evaluating Clusters

- Can use Silhouette to determine the **optimal number** of clusters
- Run K-Means for $K = 2,3,4...$
  - Computer Silhouette score for each
  - Chose K with the **highest score**
- Caveat: still **no guarantee** this matches the "ground truth"

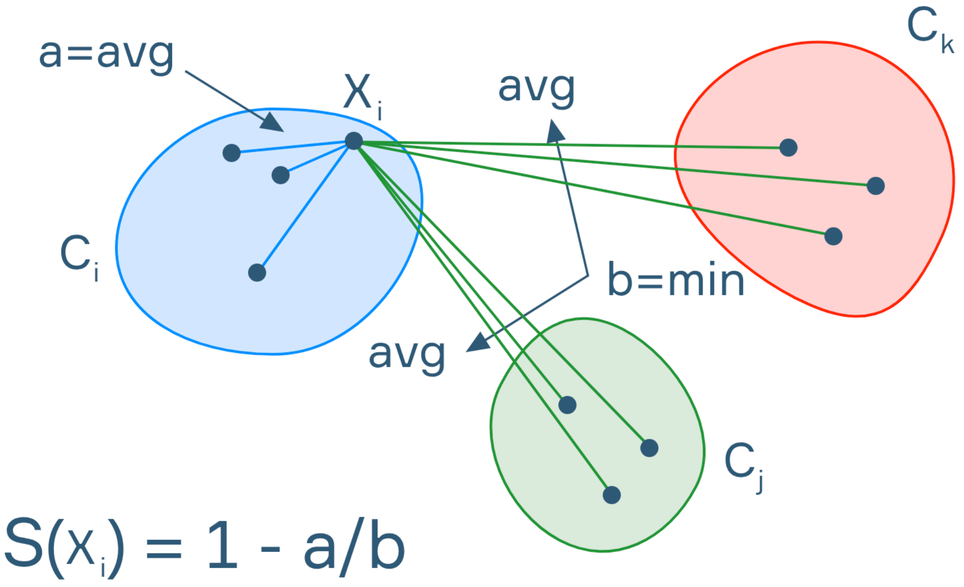
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$



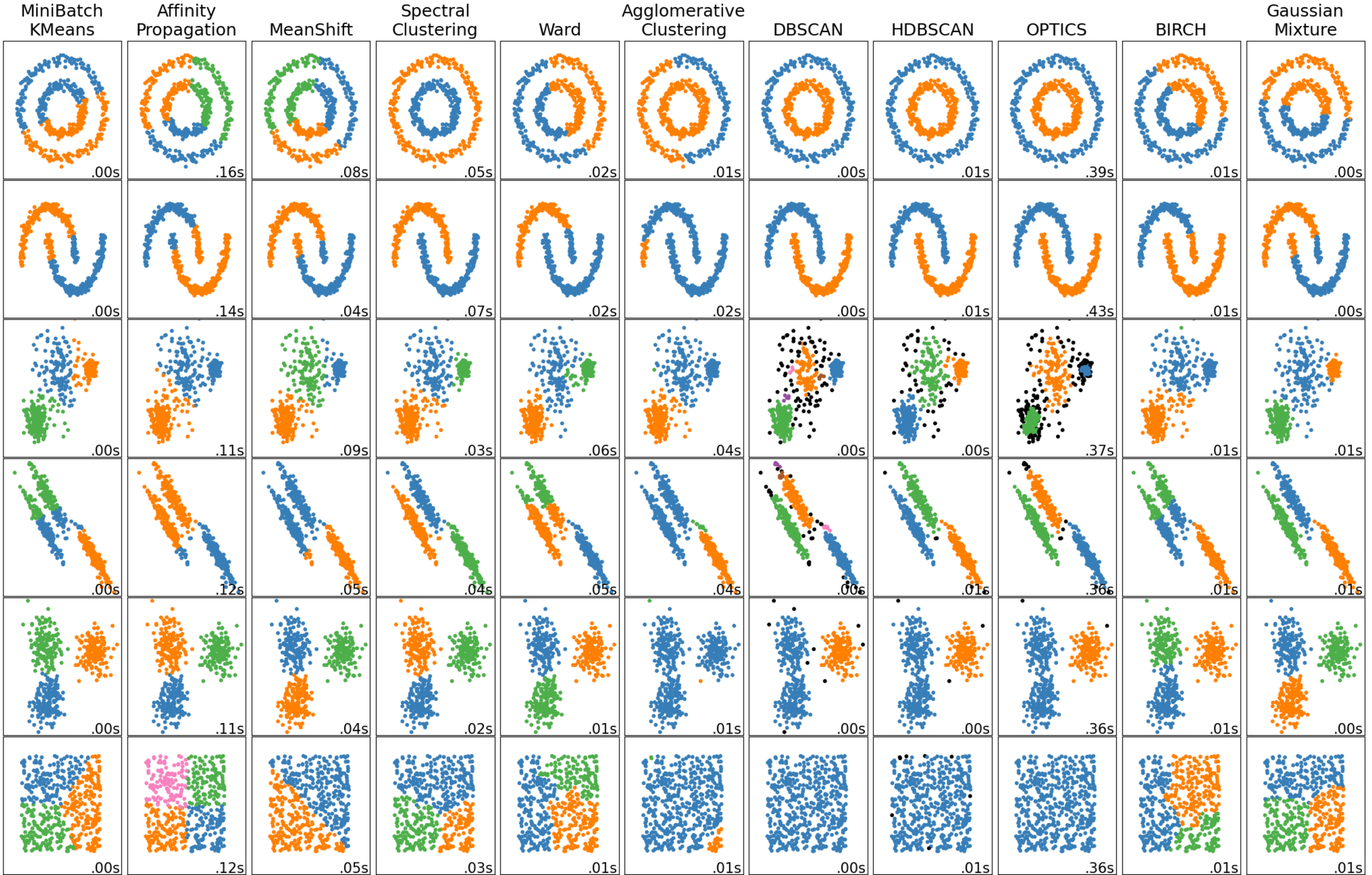
Evaluating Clusters

- Can use Silhouette to determine the **optimal number** of clusters
- Run K-Means for $K = 2,3,4...$
  - Computer Silhouette score for each
  - Chose K with the **highest score**
- Caveat: still **no guarantee** this matches the "ground truth"
- Still a surrogate measure!

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

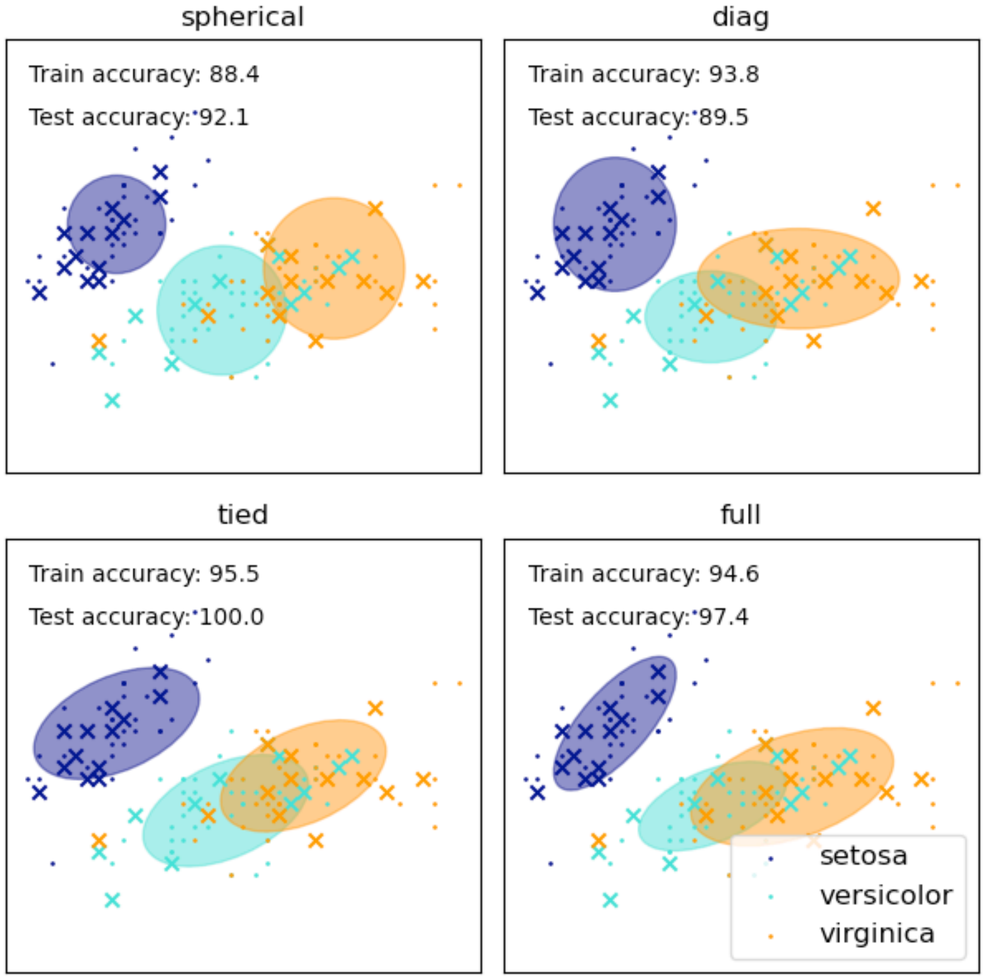


Other Clustering Techniques


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
HDBSCAN
OPTICS
BIRCH
Gaussian Mixture
.00s .16s .08s .05s .02s .01s .00s .01s .39s .01s .00s
.00s .14s .04s .07s .02s .02s .00s .01s .43s .01s .00s
.00s .11s .09s .03s .06s .04s .00s .00s .37s .01s .01s
.00s .12s .05s .04s .05s .04s .00s .01s .36s .01s .01s
.00s .11s .04s .02s .01s .01s .00s .00s .36s .01s .00s
.00s .12s .05s .03s .01s .01s .00s .01s .36s .01s .01s

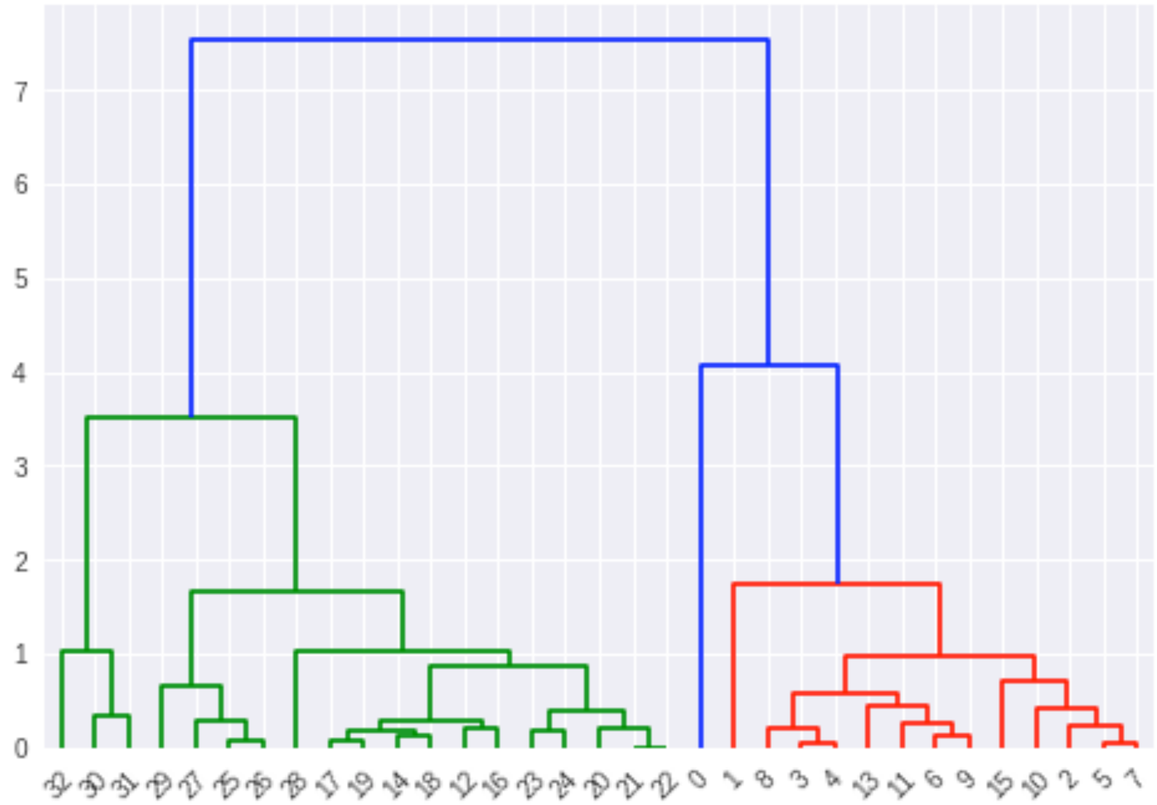
Gaussian Mixture Models

- Models data points as **generated** by multiple **Guassian (Normal) Distributions**
- Adds a **probabilistic interpretation** (what's the probability this data point comes from this cluster?)
- Each cluster has its **own variance** (addresses differing spread)
- Also handles **elliptical clusters** (with dimension co-variance)



Hierarchical Clustering

- **Iteratively merges** datapoints to address **hierarchical structure**
- **Don't need to know K in advance!**
- Can be visualized as a **"dendrogram"** (tree structure)
- Good for **data exploration**
- Con: might **imply more structure than there is**



Sequence Segmentation

Morpheme Segmentation

Inuktitut:
"**pusikaarjuakuluqaqtuŋa"**

English: "I have a big cat"

Pieces: *pusikaa - jua - kulu - qaq - tuŋa*

Morpheme Segmentation

- Problem in NLP/Linguistics

Inuktitut:
"**pusikaarjuakuluqaqtuŋa"**

English: "I have a big cat"

Pieces: *pusikaa - jua - kulu - qaq - tuŋa*

Morpheme Segmentation

- Problem in NLP/Linguistics
- Many languages have **complex structure within words**
  - Traditional techniques just **split into words**
  - How do we **discover this structure?**

Inuktitut:
"**pusikaarjuakuluqaqtuŋa"**

English: "I have a big cat"

Pieces: *pusikaa - jua - kulu - qaq - tuŋa*

Morpheme Segmentation

- Problem in NLP/Linguistics
- Many languages have **complex structure within words**
  - Traditional techniques just **split into words**
  - How do we **discover this structure?**
- Supervised data is **very rare**
  - Usually don't bother to make it unless you're doing NLP/ML

Inuktitut:
"**pusikaarjuakuluqaqtuŋa"**

English: "I have a big cat"

Pieces: *pusikaa - jua - kulu - qaq - tuŋa*

Minimal Description Length (MDL)

- MDL: try to **jointly minimize the "cost"** of the **data** and the **"description"**
  - Intuitively: try to **"compress"** the system to its **most efficient form**
- MDL **tradeoff:**
  - Undersegmentation: data is **compressed** but codebook is **large**
  - Oversegmentation: codebook is **compressed** but data is **large**

$$\begin{aligned} C &= \text{Cost}(\text{Source text}) + \text{Cost}(\text{Codebook}) \\ &= \sum_{\text{tokens}} -\log p(m_i) + \sum_{\text{types}} k * l(m_j) \end{aligned}$$

m_i : a morpheme

$l(m_i)$: its length

$-\log p(m_i)$: its **negative log probability** (frequency)

Hierarchical Counting