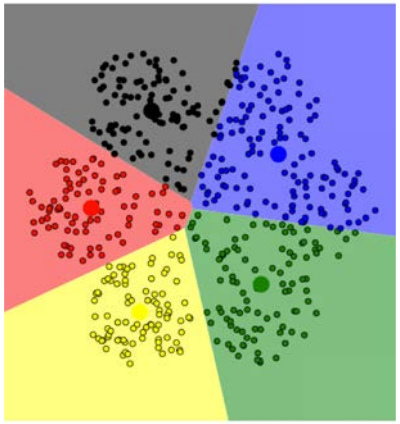
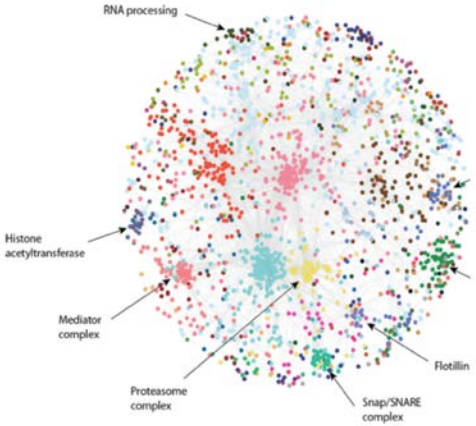
Tufts COMP 135: Introduction to Machine Learning

https://www.cs.tufts.edu/comp/135/2019s/

Clustering: K-Means & Mixture models





Prof. Mike Hughes

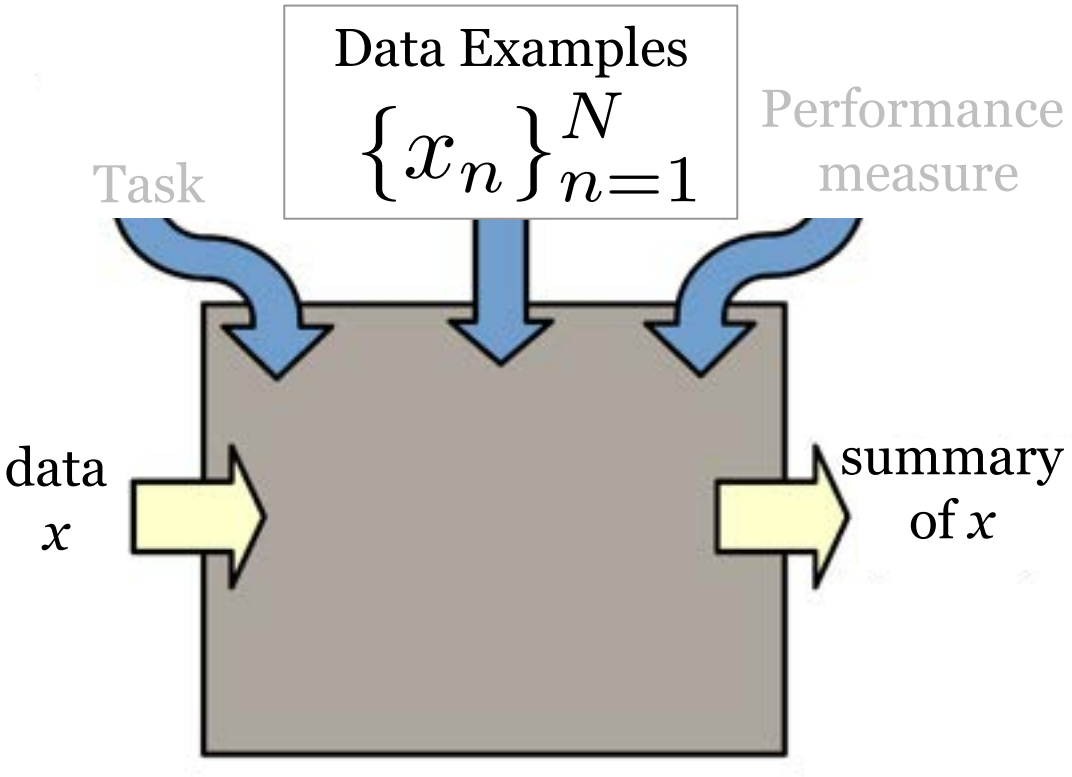
Many ideas/slides attributable to:
Emily Fox (UW), Erik Sudderth (UCI)

What will we learn?

Supervised Learning

Unsupervised Learning

Reinforcement Learning



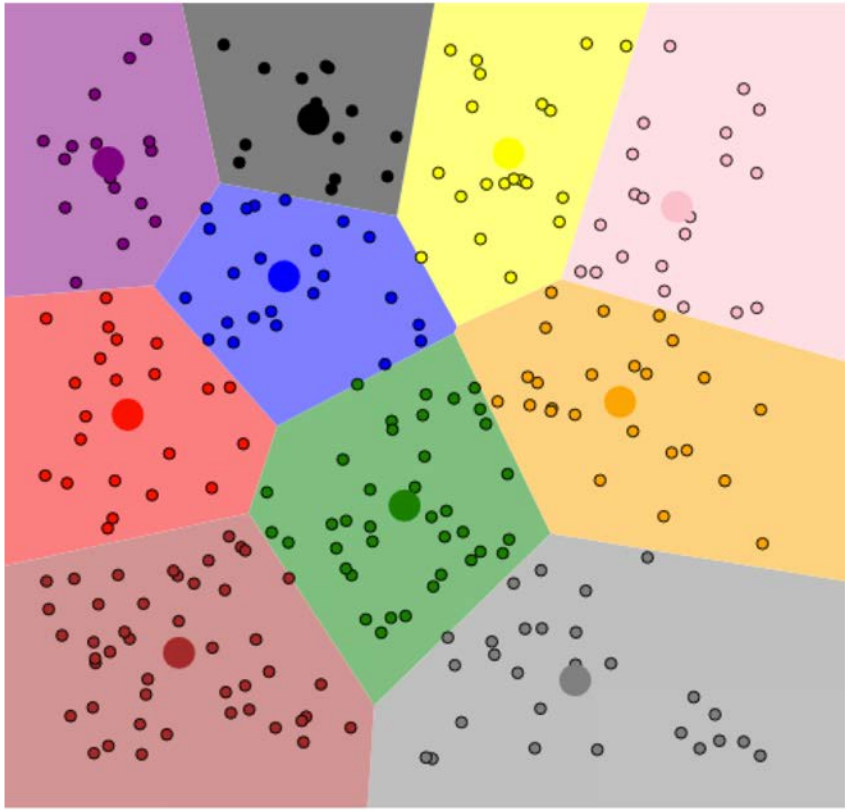
Task: Clustering

Supervised Learning

Unsupervised Learning

clustering

Reinforcement Learning



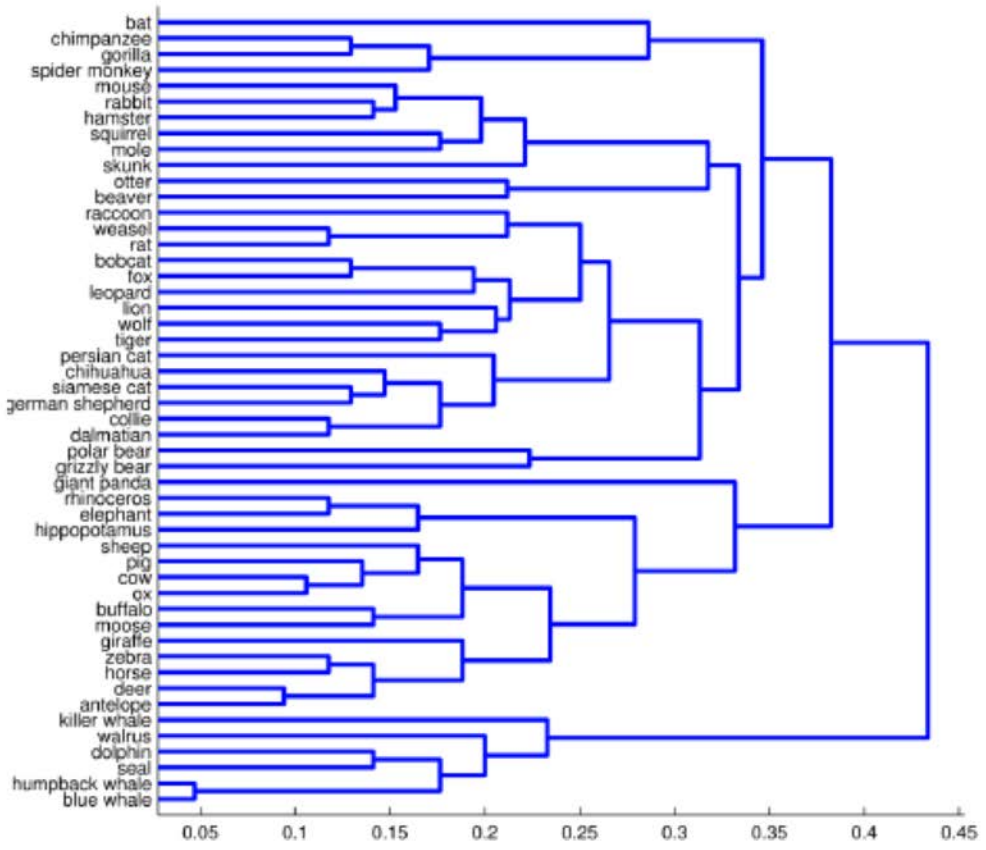
Clustering: Unit Objectives

- Understand key challenges
  - How to choose the number of clusters?
  - How to choose the shape of clusters?
- K-means clustering (deep dive)
  - Shape: Linear Boundaries (nearest Euclidean centroid)
  - Explain algorithm as instance of "coordinate descent"
    - Update some variables while holding others fixed
  - Need smart init and multiple restarts to avoid local optima
- Mixture models (primer)
  - Advantages of soft assignments and covariances

Examples of Clustering

Clustering Animals by Features

Data set of 50 animals, 85 binary features (e.g., longneck, water, smelly)



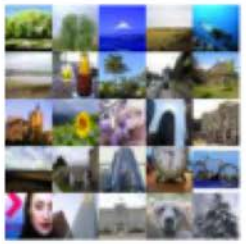
Clustering Images

(a) Cluster Centers

(b) Cluster 1

(c) Cluster 2

(d) Cluster 3



(e) Cluster 4

(f) Cluster 5

(g) Cluster 6

(h) Cluster 7

(i) Cluster 8

(j) Cluster 9

(k) Cluster 10

(l) Cluster 11

(m) Cluster 12

(n) Cluster 13

(o) Cluster 14

(p) Cluster 15

(q) Cluster 16

Image Compression

Original Image

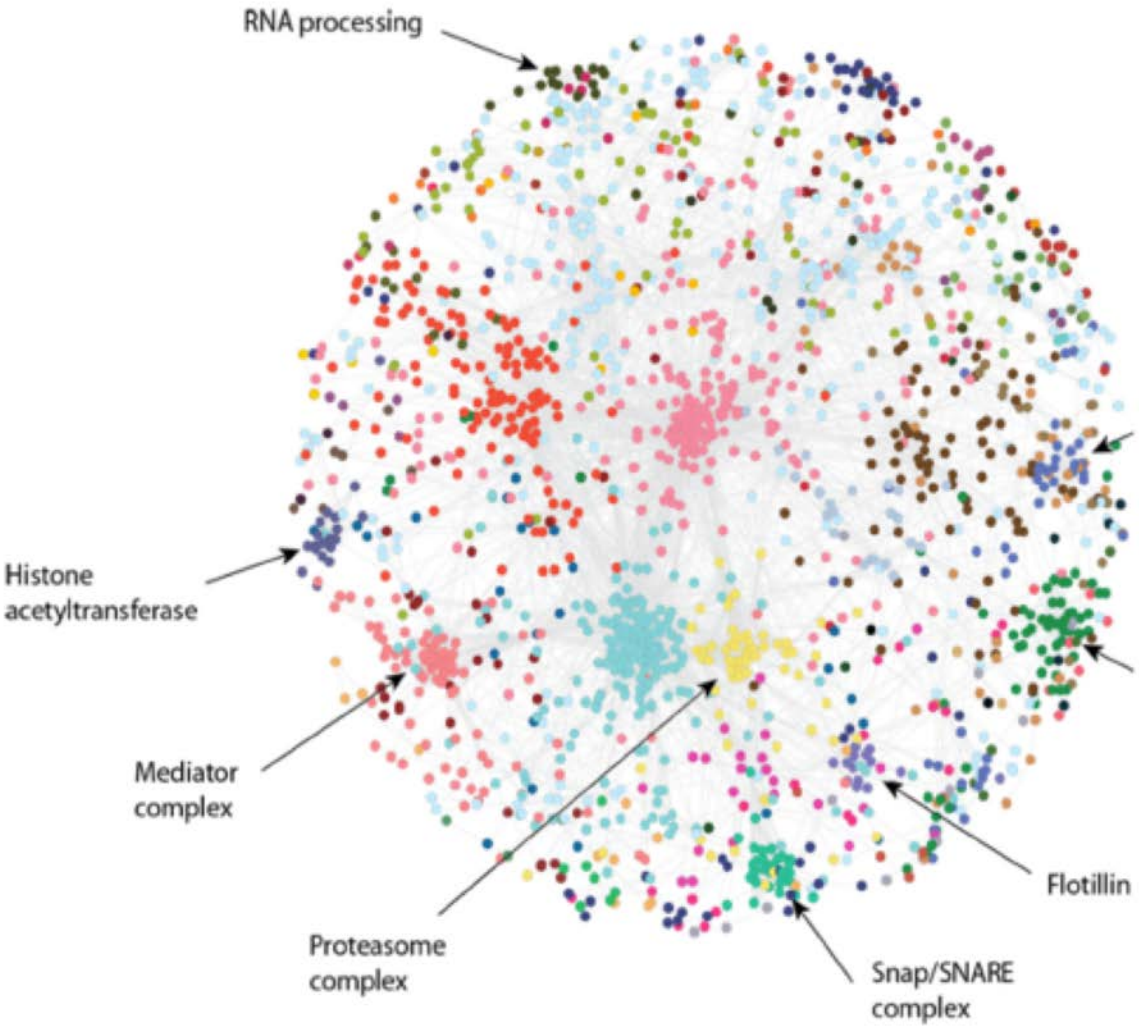
Possible pixel values (R, G, B):
255 * 255 * 255 = 16 million

16-color Image

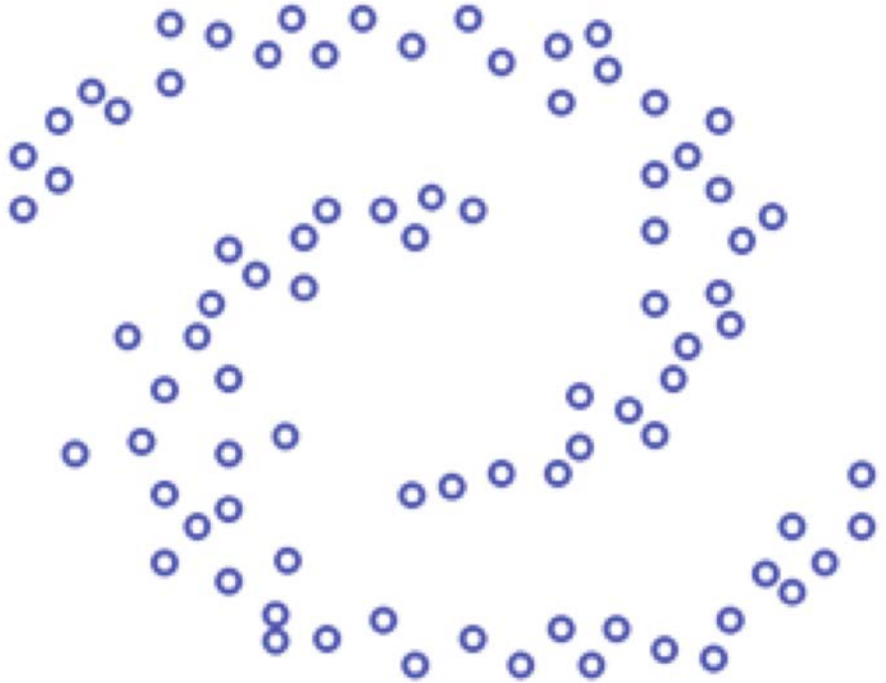
Possible pixel values:
One of 16 fixed (R,G,B) values

This image on the right achieves a compression factor of around 1 million!

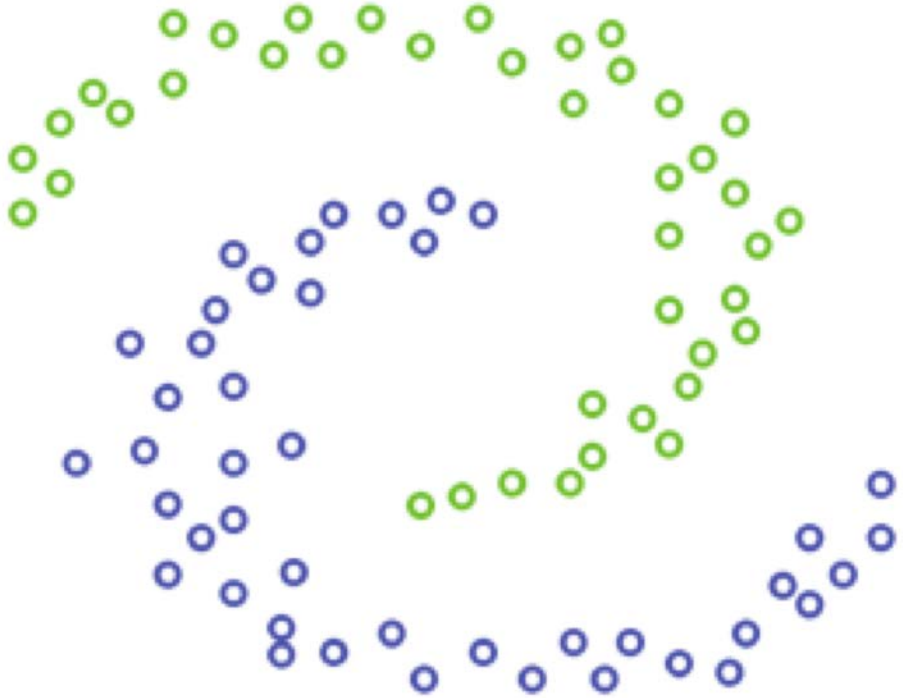
Understanding Genes



How to cluster these points?



How to cluster these points?



Key Questions

how many?

what shape?

K-Means

Input:

- Dataset of N example feature vectors $\{x_n\}_{n=1}^{N}$

$$x_n = [x_{n1} \; x_{n2} \; \ldots \; x_{nF}]$$

- Number of clusters K

K-Means Goals

- Assign each example to one of K clusters
  - Assumption: Clusters are exclusive
- Minimize Euclidean distance from examples to cluster centers
  - Assumption: Isotropic Euclidean distance (all features weighted equally, no covariance modeled) is a good metric for your data

K-Means output

- Centroid Vectors (one per cluster k in 1, … K)

$$\mu_k = [\mu_{k1} \ \mu_{k2} \ \ldots \mu_{kF}]$$

Length = # features F
Real-valued

- Assignments (one per example n in 1 … N)

$$z_n = [z_{n1} \ z_{n2} \ \ldots z_{nK}]$$
$$= [0 \ 1 \ 0 \ 0 \ 0]$$
$$= [0 \ 0 \ 0 \ 1 \ 0]$$

One-hot vector indicates which of K clusters example n is assigned to

Use Euclidean distance

$$\text{dist}(x_n, \mu_k) = ||x_n - \mu_k||_2 = \sqrt{\sum_{f=1}^{F} (x_{nf} - \mu_{kf})^2}$$

K-means Optimization Problem

$$\text{cost}(z, \mu)$$

$$\min_{z, \mu} \quad \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \text{dist}(x_n, \mu_k)$$

K-Means Algorithm

Initialize cluster means

Repeat until converged

1) Update per-example assignment

For each n in 1:N:

Find cluster k* that minimizes $\text{dist}(x_n, \mu_k)$

Set z_n to indicate k*

2) Update per-cluster centroid

For each k in 1:K:

Set μ_k to mean of data vectors assigned to k

K-Means Algorithm

Initialize cluster means

Repeat until converged

1) Update per-example assignment

$$z_{nk} = \begin{cases} 1 & \text{if } \text{dist}(x_n, \mu_k) \leq \text{dist}(x_n, \mu_j) \quad \forall j \neq k \\ 0 & \text{o.w.} \end{cases}$$

2) Update per-cluster centroid

$$\mu_k = \frac{\sum_n z_{nk} x_n}{\sum_n z_{nk}}$$

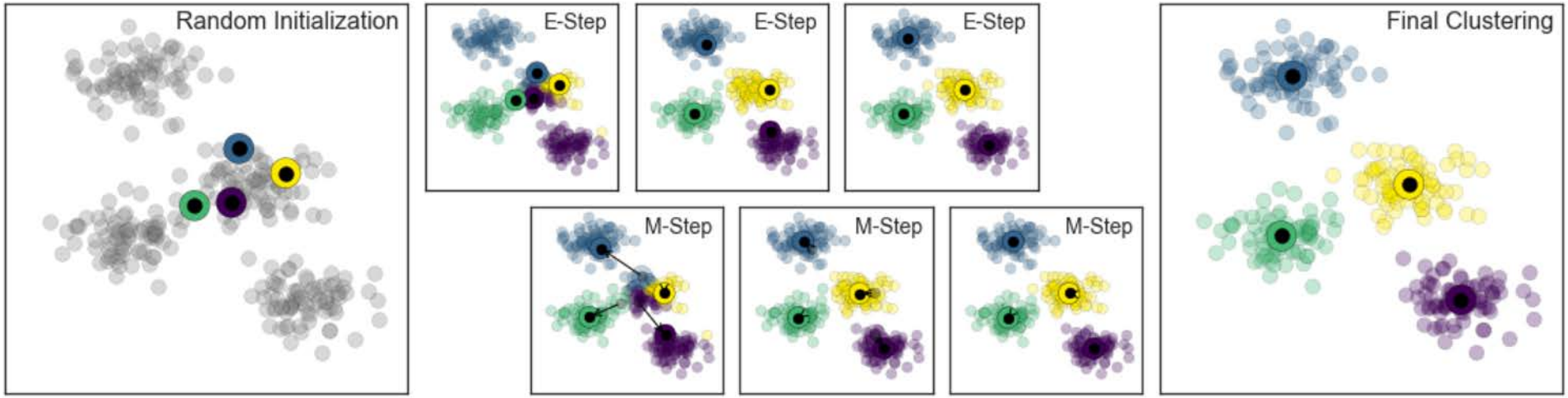
Updates each improve cost

$$\min_{z,\mu} \quad \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \text{dist}(x_n, \mu_k)$$

$$\text{cost}(z, \mu)$$

K-Means Algo: Coordinate Ascent

Credit: Jake VanderPlas



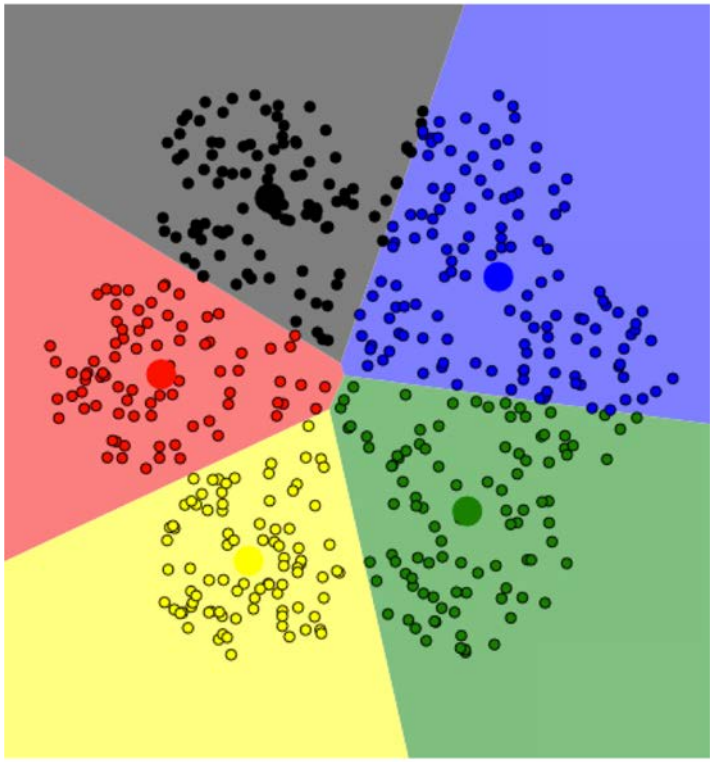
E-step or per-example step: Update Assignments
M-step or per-centroid step: Update Centroid Locations
Each step yields cost equal or lower than before

Demo!

http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

Demo 2 (Choose initial clusters)

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

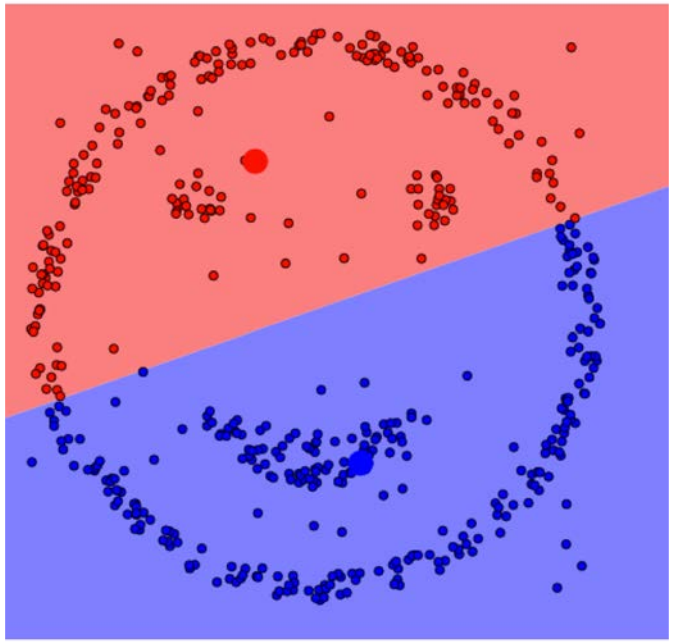


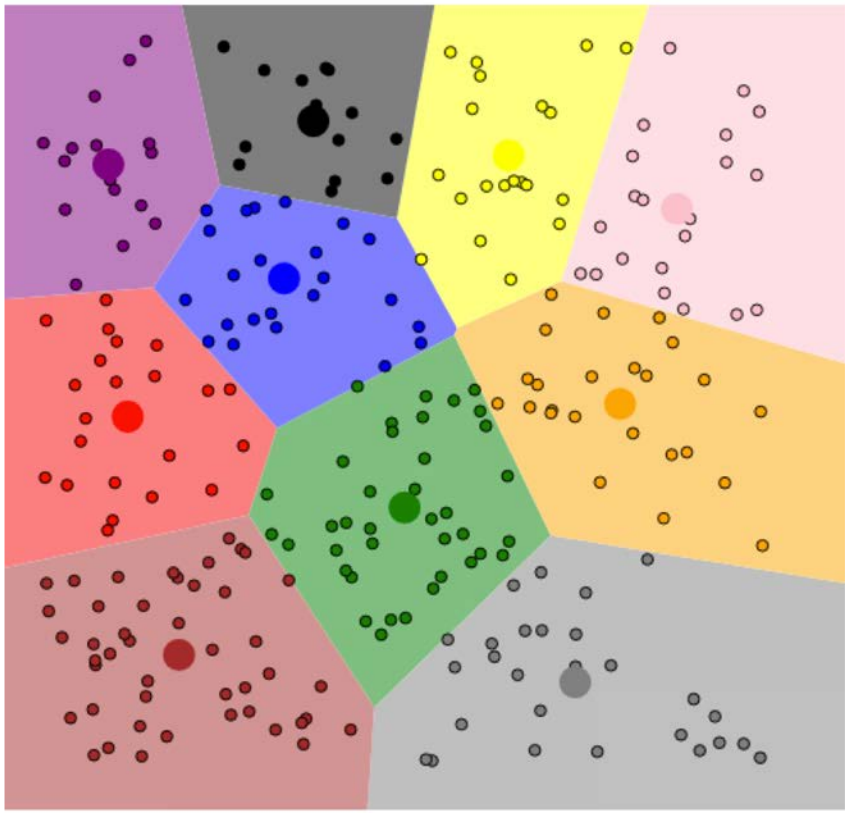
Pick a dataset and fix a K value (e.g. 2 clusters)

Can you find a different fixed point solution from your neighbor?

What does this mean about the objective?

K-means Boundaries are Linear





Decisions when applying k-means

- How to initialize the clusters?

- How to choose K?

Initialization: *K-means++*

Possible Initializations

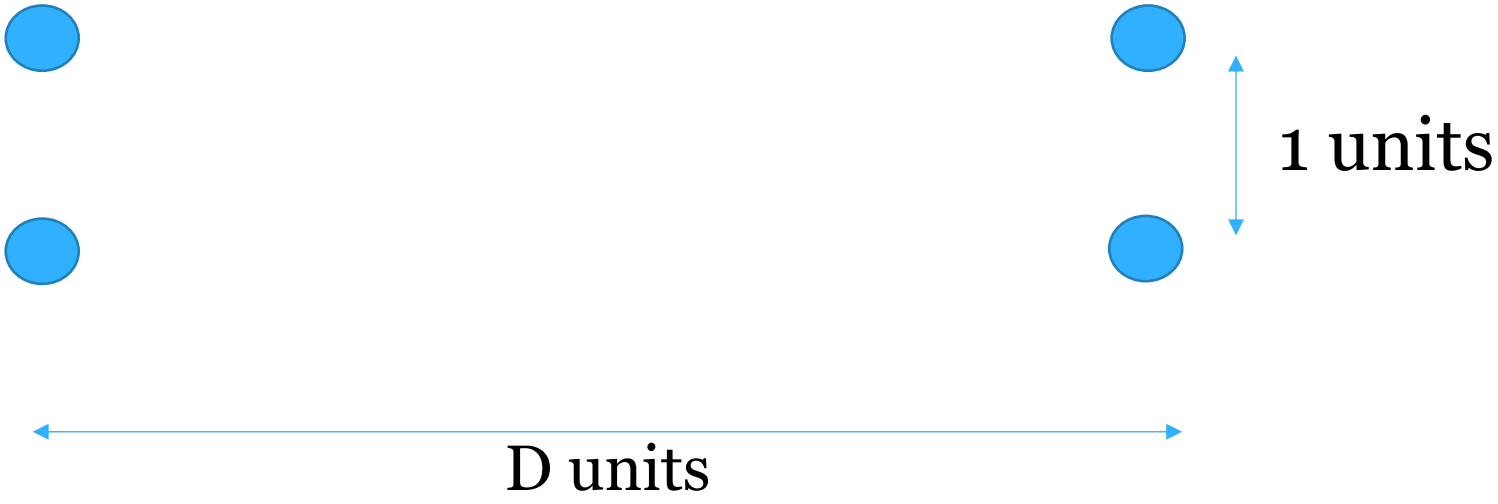
- Draw K random centroid locations

- Choose K data vectors as centroids
  - Uniformly at random

What can go wrong?

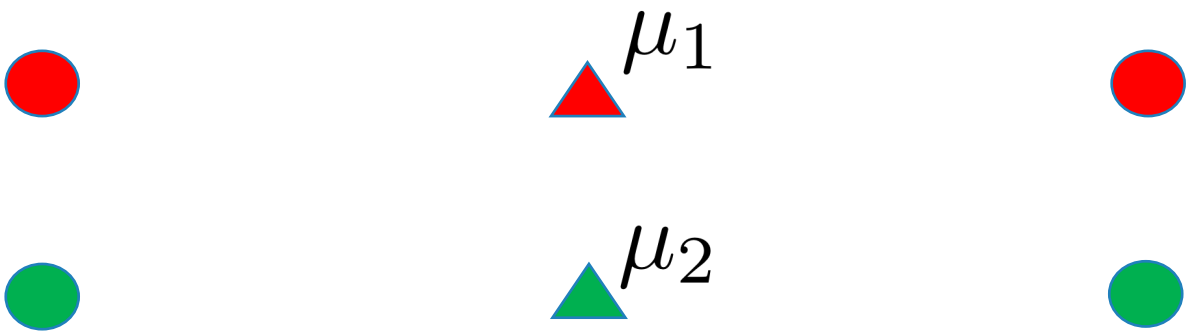
Example

- Toy Example: Cluster these 4 points with K=2



No Guarantees on Cost!

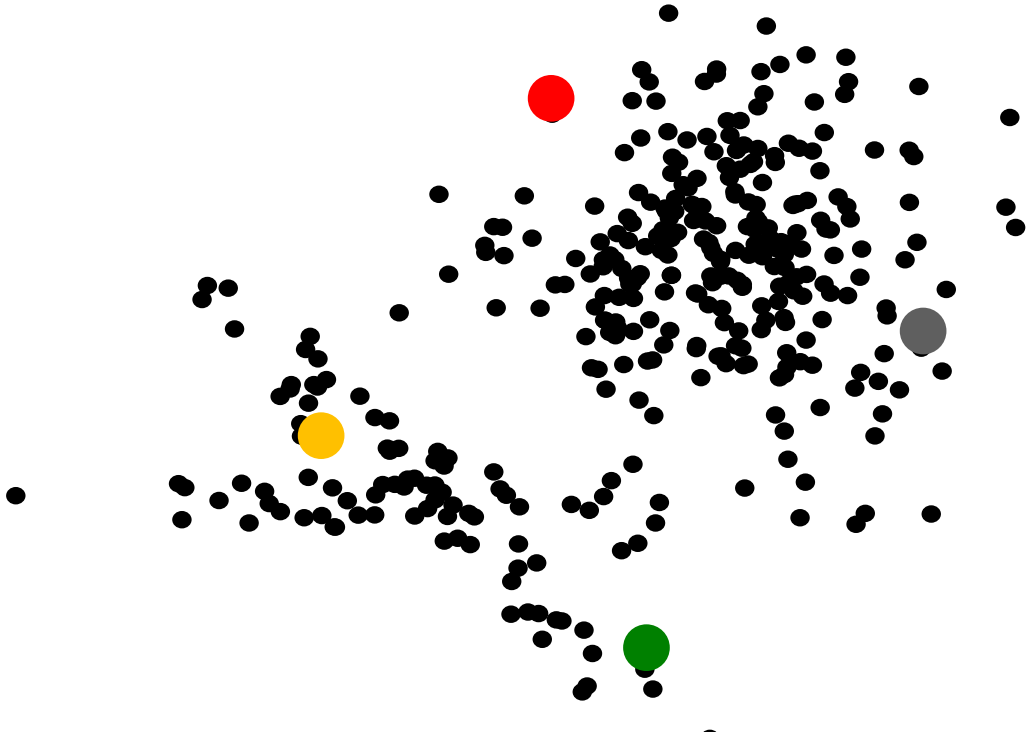
BAD solution. Cost scales with distance D, which could be arbitrarily larger than 1



OPTIMAL solution. Cost scales will be O(1)

Better init: k-means++

Arthur & Vassilvitskii SODA '07



Step 1: choose an example uniformly at random as first centroid

Repeat for k = 2, 3, … K:

Choose example based on distance from nearest centroid

$$Pr(\mu_k = x_n) \propto \min_{j \in \{1,2,\ldots k-1\}} \text{dist}(x_n, \mu_j)$$

k-means++: Guarantees on Quality

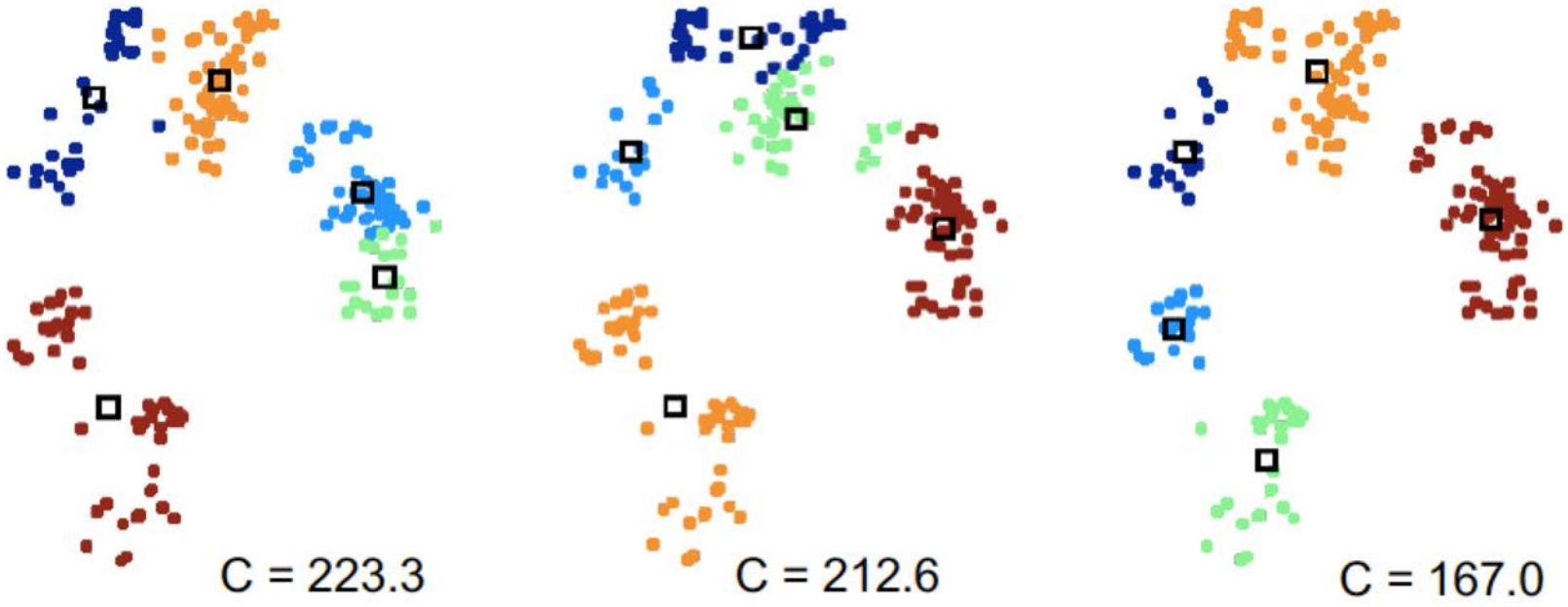
Arthur & Vassilvitskii SODA '07

Step 1: choose an example uniformly at random as first centroid
Repeat for k = 2, 3, … K:
Choose with probability proportional to distance from nearest centroid

$$Pr(\mu_k = x_n) \propto \min_{j \in \{1,2,\ldots k-1\}} \text{dist}(x_n, \mu_j)$$

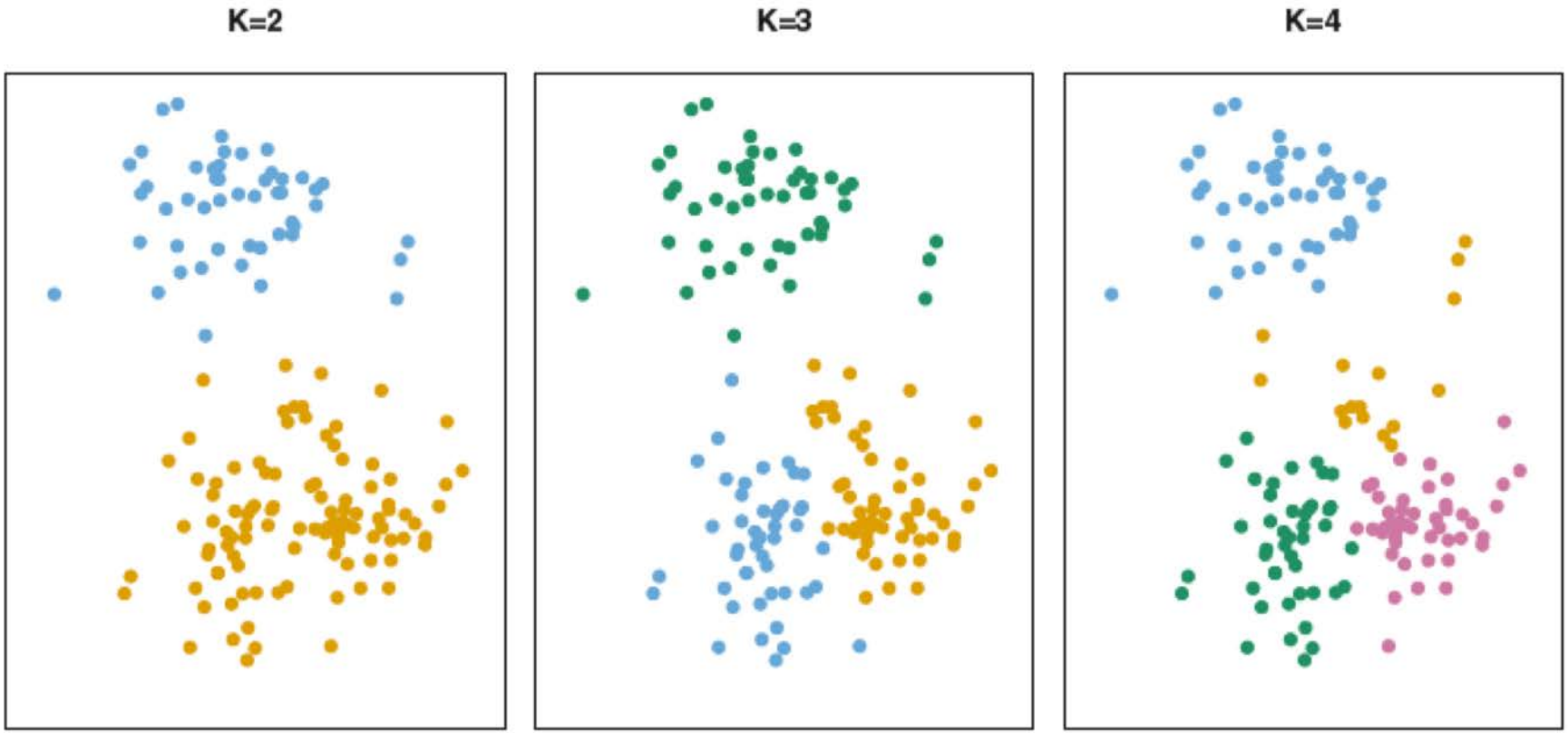
Theorem: This initialization will achieve score that is *O(log K)* of optimal score.

Use cost to decide among multiple runs of k-means

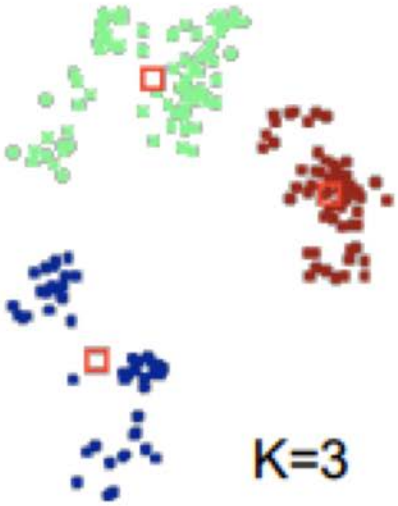


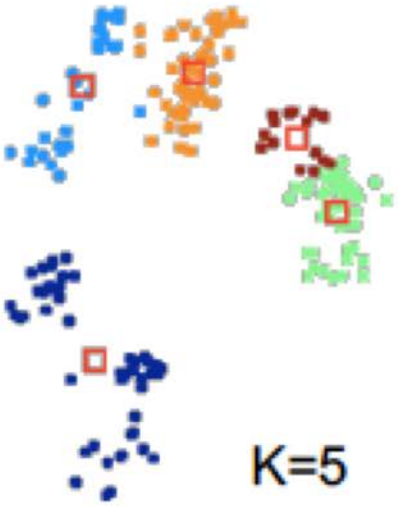
How to pick K in K-means?

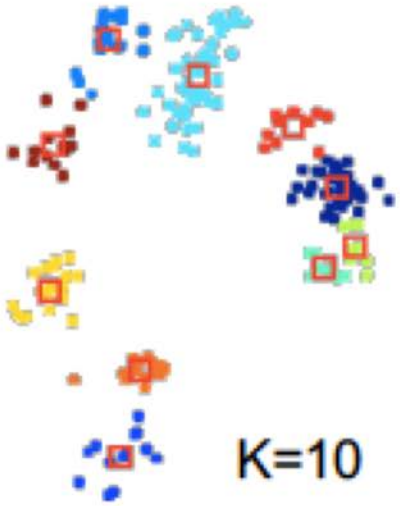
Same data. Which K is best?

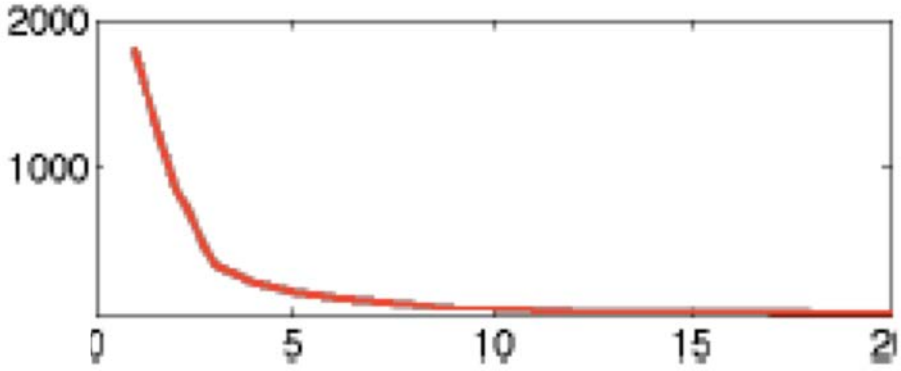


Use cost function? **No!**









At each K, the global optimal cost always decreases. (Local optima may not)

Limit as K -> N, cost is **zero**.

Add complexity penalty!

Want adding additional clusters to increase cost, if don't help "enough"

$$J(z, \mu) = \text{cost}(z, \mu) + \lambda \text{penalty}(K)$$

Computation Issues

K-Means Computation

- Most expensive step: Updating assignments
  - N x K distance calculations

- Scalable?
  - Don't need to update all examples, just grab a minibatch
  - Can do stochastic learning rate updates too
- Parallelizable?
  - Yes. Given fixed centroids, can process minibatches of examples (the assignment step) in parallel

Improved clustering: Gaussian mixture model

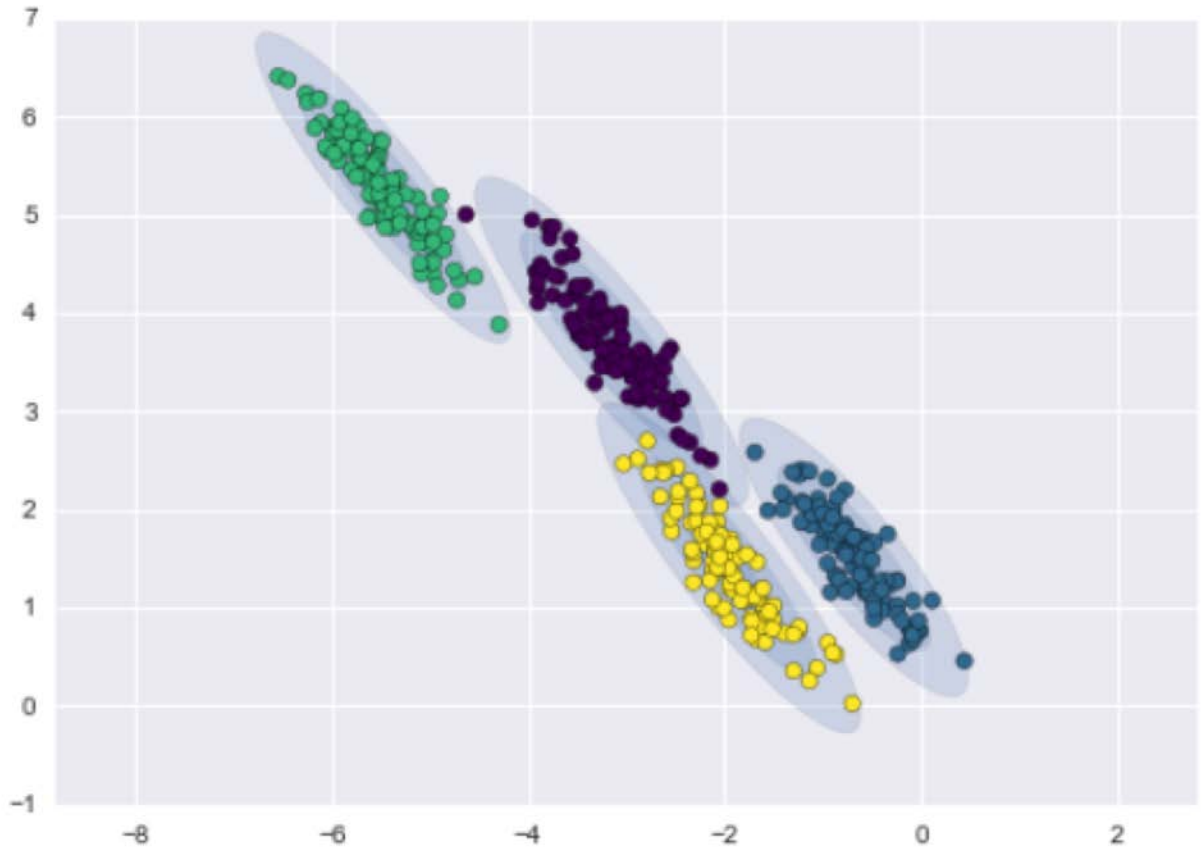
Improving K-Means

- Assign each example to one of K clusters
  - Assumption: Clusters are exclusive
  - Improvement: Soft probabilistic assignment

- Minimize Euclidean distance from examples to cluster centers
  - Assumption: Isotropic Euclidean distance (all features weighted equally, no covariance modeled) is a good metric for your data
  - Improvement: Model cluster covariance

Gaussian Mixture Model

```
gmm = GMM(n_components=4, covariance_type='full', random_state=42)
plot_gmm(gmm, X_stretched)
```



Gaussian Mixture Model

- Mean Vectors (one per cluster k in 1, … K)

$$\mu_k = [\mu_{k1} \ \mu_{k2} \ \ldots \mu_{kF}]$$

Length = # features F
Real-valued

- **Covariance** Matrix (one per cluster k in 1 … K)

$$\Sigma_k$$

F x F square symmetric matrix
Positive definite (invertible)

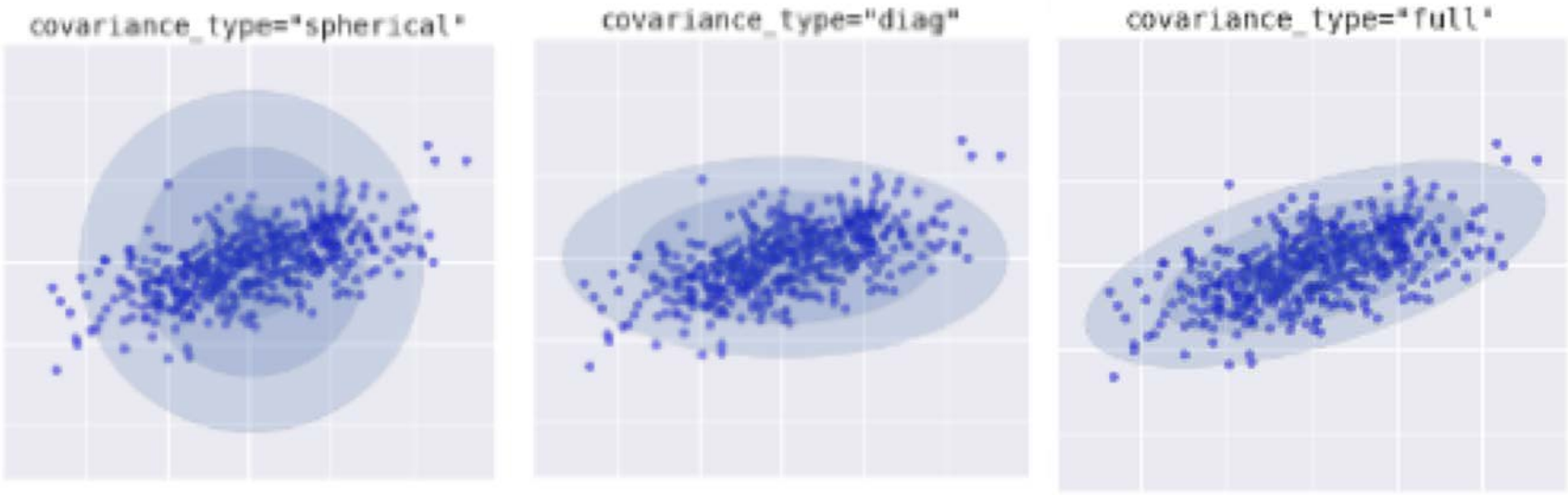
- **Soft** assignments (one per example n in 1 … N)

$$r_n = [r_{n1} \ r_{n2} \ \ldots \ r_{nK}]$$

Probabilistic!
Vector sums to one

Covariance Models

Credit: Jake VanderPlas



Most similar to k-means

More flexible

GMM Training

Maximize the likelihood of the data

$$\max_{\mu, \Sigma} \quad \sum_{n=1}^{N} \log p(x_n | \mu, \Sigma)$$

Beyond this course: Can show this looks a lot like K-means' simplified objective

Algorithm: Coordinate ascent!

E-step : Update soft assignments r

M-step: Update means and covariances

Special Case

- K-means is a GMM with:
  - Hard winner-take-all assignments
  - Spherical covariance constraints

Clustering: Unit Objectives

- Understand key challenges
  - How to choose the number of clusters?
  - How to choose the shape of clusters?
- K-means clustering (deep dive)
  - Shape: Linear Boundaries (nearest Euclidean centroid)
  - Explain algorithm as instance of "coordinate descent"
    - Update some variables while holding others fixed
  - Need smart init and multiple restarts to avoid local optima
- Mixture models (primer)
  - Advantages of soft assignments and covariances