Unsupervised Learning & Data Clustering

Problem Set-up

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Define the set-up
of unsupervised
learning

Learning from Unlabeled Data

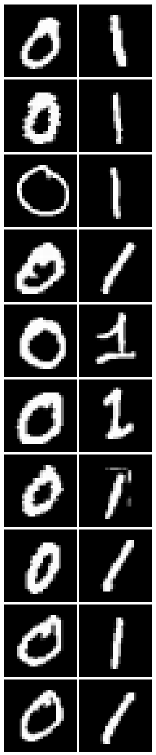
Given a training set of *n* unlabeled samples $\{x^{(i)}\}$

What can we learn from the samples?

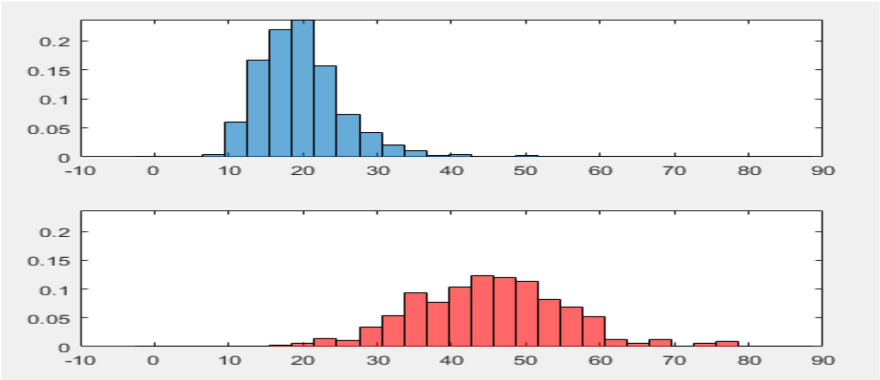
- We could estimate the overall distribution of the data without knowing their label.
- We could figure out the groupings of the samples (if any).
- We could identify some features that may be more important than others.

An Example

Illustrating structures/groupings of unlabeled samples may relate to the (unknown) labels of the samples



→ If we know the labels, we may find the densities of the classes →

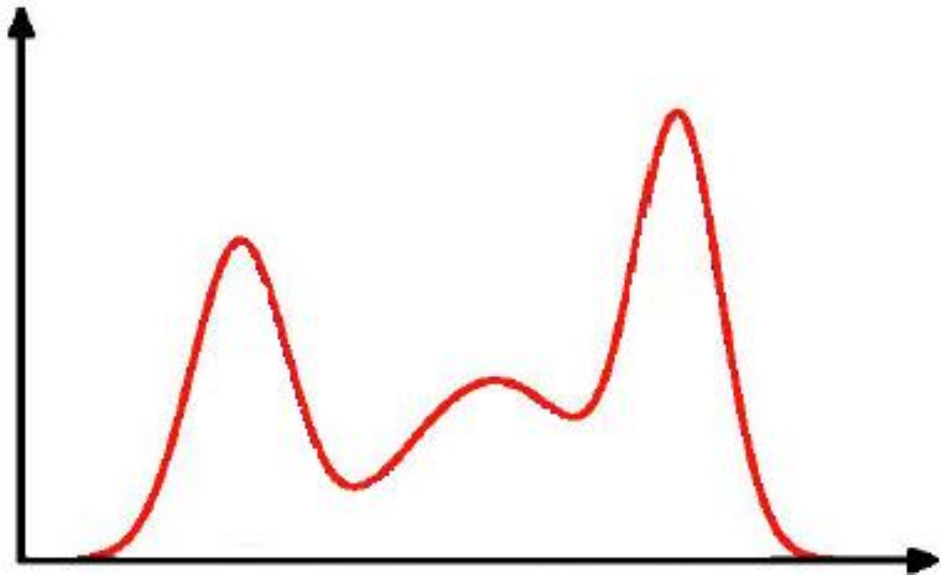


→ What may we see if we have no label for the data samples?

Another Example

- A density estimated from unlabeled samples may help us to identify densities of different classes

- If we know there are three classes in the data, each having a normal distribution …



A Mixture-Density Model

- **Assume a parametric model like this:**
  - The samples come from C classes.
  - The prior probabilities $P(\omega_j)$ for each class are known, for $j = 1, \ldots, C$.
  - The form of $p(\mathbf{x} \mid \omega_j, \theta_j)$ ($j = 1, \ldots, C$) are known.
  - The C parameter vectors $\theta_1, \theta_2, \ldots, \theta_C$ are unknown.
- **Samples from this distribution are given, but the labels of the training samples are *unknown*.**

A Mixture-Density Model (cont'd)

- What is the PDF of the unlabeled samples?

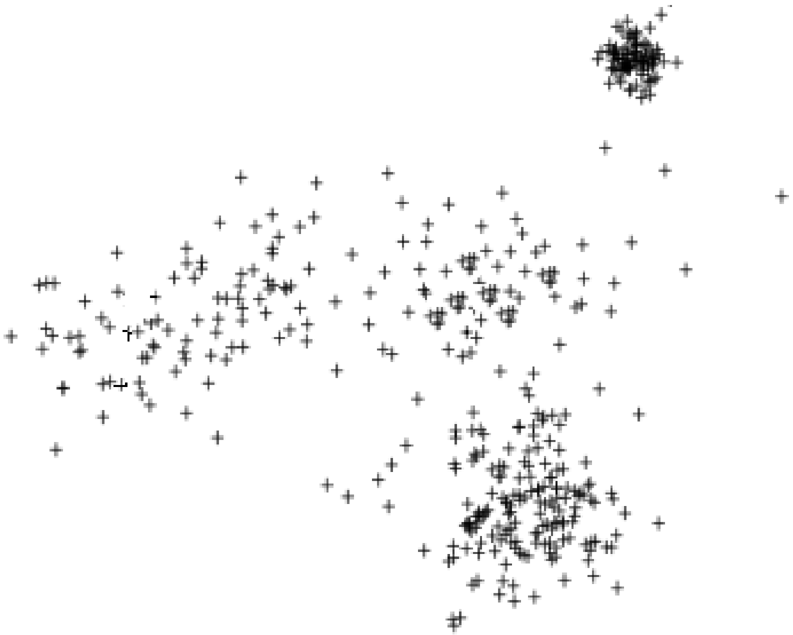
$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{j=1}^{C} p(\mathbf{x} \mid \omega_j, \boldsymbol{\theta}_j) P(\omega_j)$$

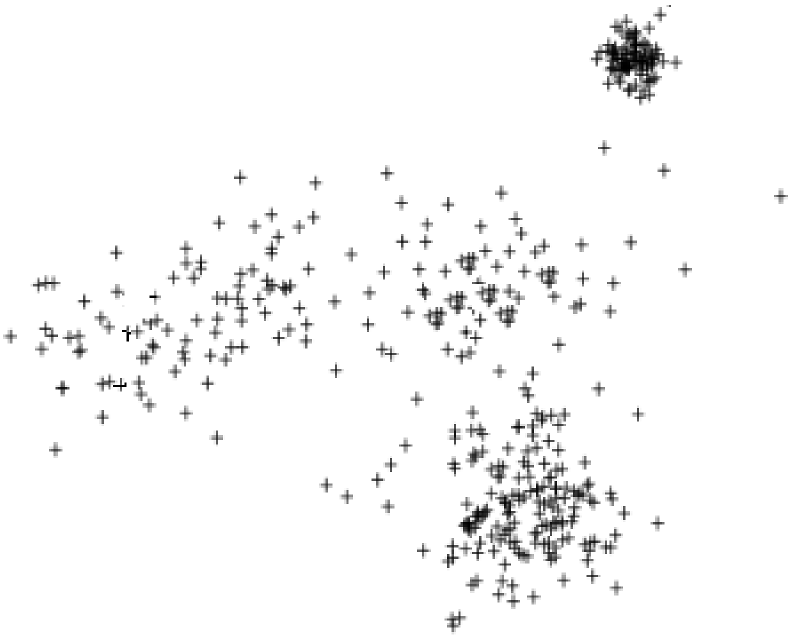
where $\theta = (\theta_1, \theta_2, \ldots, \theta_C)$

- Can we learn θ from unlabeled samples from this mixture density?

Illustrating Mixture-Density Model

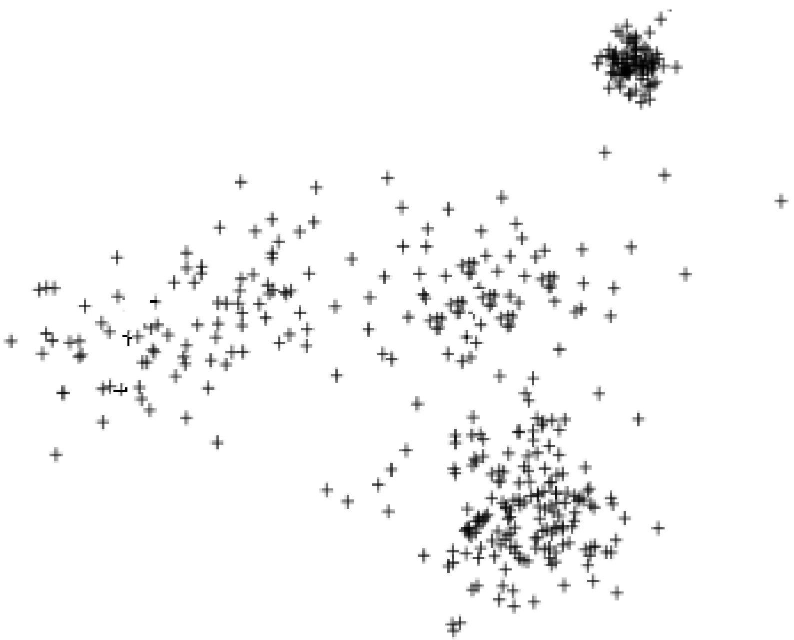
An example: with the assumption of 4 classes

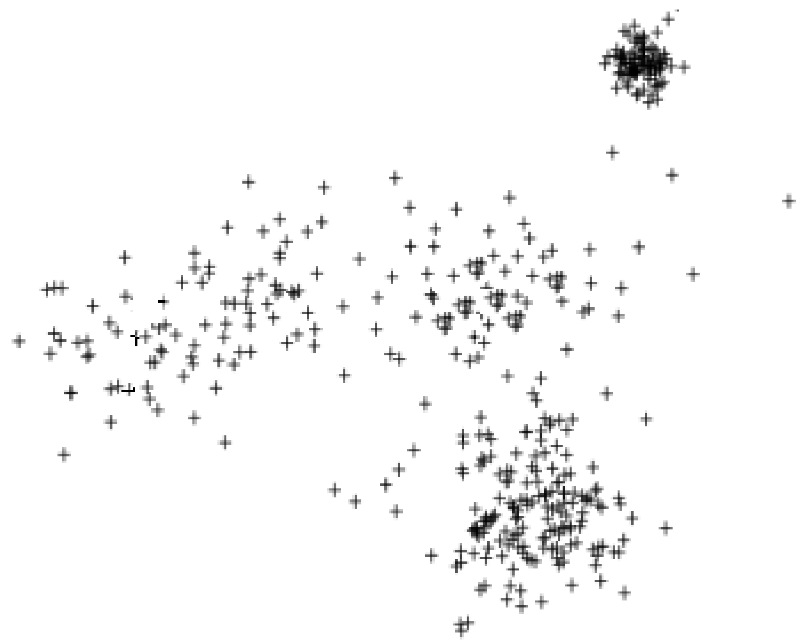




Illustrating Mixture-Density Model

An example: with the assumption of 2 classes





The Question of Identifiability

- **Can we learn a unique θ from a set of unlabeled samples from a mixture density?**
  - For continuous features (with PDFs), the answer is often "Yes".
- **An example in discrete case (with PMF).**
  - Two coins with P(head) being p & q respectively.
  - Randomly pick one and toss it; Record the outcome.
  - With only the outcomes of N tosses, but not knowing which coin was used each time (→ unsupervised), can we figure out p and q?

Unsupervised Learning

Gaussian Mixture Models and the EM Algorithm

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Define the Gaussian Mixture Model

Objective

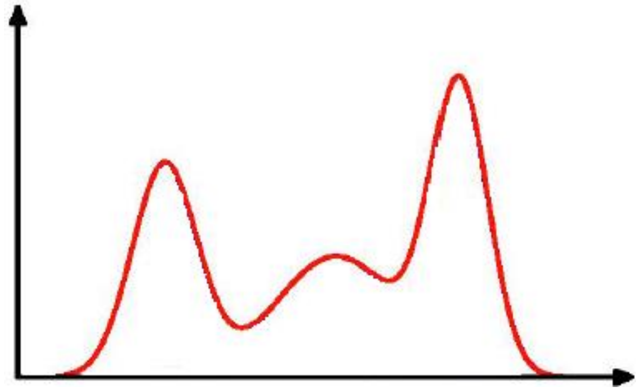
Illustrate the Expectation-Maximization Algorithm

The Gaussian Mixture Models

- **The mixture model:**

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{j=1}^{C} \underline{p(\mathbf{x} \mid \omega_j, \boldsymbol{\theta}_j)} P(\omega_j)$$

- **GMM: each component density is a Gaussian distribution.**
  - Can be a good approximation to many real data distributions.



If We Do Have Labels…

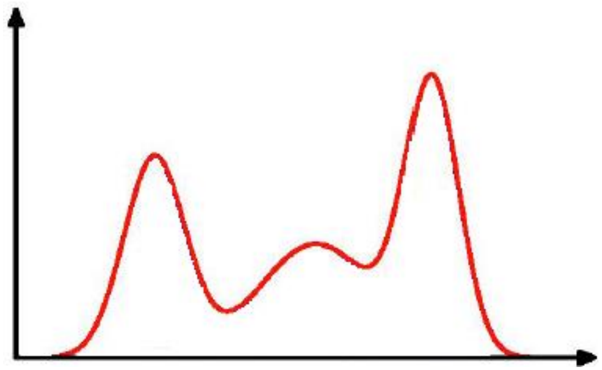
$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{j=1}^{C} \underline{p(\mathbf{x} \mid \omega_j, \boldsymbol{\theta}_j)} P(\omega_j)$$

- This becomes supervised learning for each component (class).

- It is more difficult without labels.

Unsupervised Case

- Consider an iterative method using the *maximum likelihood estimation* concept.
- Consider a 3-component 1-d example.



- What are the parameters in this case?
- We might have some initial (imprecise) guesses for the parameter, e.g., vs the *k-means algorithm*.
  - *How to improve the initial guesses?*

Unsupervised Case (cont'd)

Iterate on t

Given parameter estimates at iteration t-1

An example of Expectation-Maximization Algorithm.

Step 1. For a sample j, compute its probability of being from class *k*

$$P[y_j = k \mid x_j, \theta^{(t-1)}] \propto P_k^{(t-1)} p(x_j \mid \mu_k^{(t-1)}, \sigma_k^{(t-1)}), \ \forall k = 1, 2, 3$$

Step 2. Update the estimates of the parameters

$$\mu_k^{(t)} = \frac{\sum_j x_j P[y_j = k \mid x_j, \theta^{(t-1)}]}{\sum_j P[y_j = k \mid x_j, \theta^{(t-1)}]}$$

$$P_k^{(t)} = \frac{\sum_j P[y_j = k \mid x_j, \theta^{(t-1)}]}{\text{total \# of samples}}$$

Unsupervised Learning

The k-Means Algorithm

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Discuss the basics of data clustering

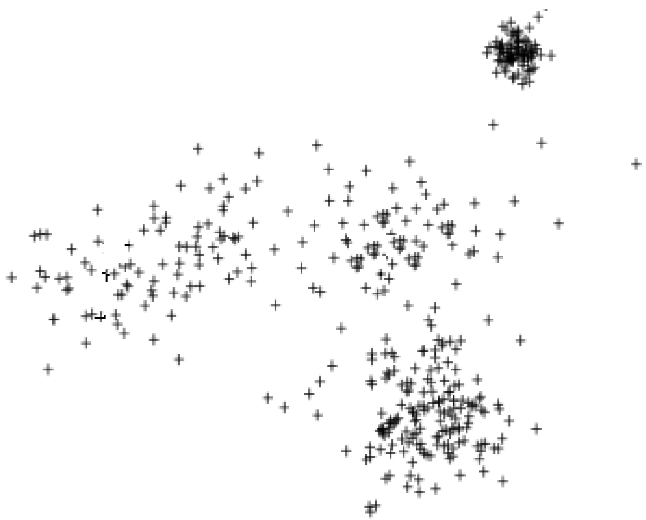
Objective

Illustrate the k-Means Algorithm

Finding the Clusters/Groupings of the Samples

A few basic questions to answer

- How to represent the clusters?
  - ➔ We will use the centroid to represent a cluster.
- Which cluster a sample should be assigned to (e.g., membership)?
  - ➔ We will use the similarity to the centroid to determine the membership.
- What similarity measure to use?
  - E.g., Euclidean distance



More on Similarity Measures

- **If we use Euclidean distance as the measure:**
  - It is invariant to translations & rotations of the feature space.
  - But not to more general transformations.
- **E.g., if one feature is scaled.**

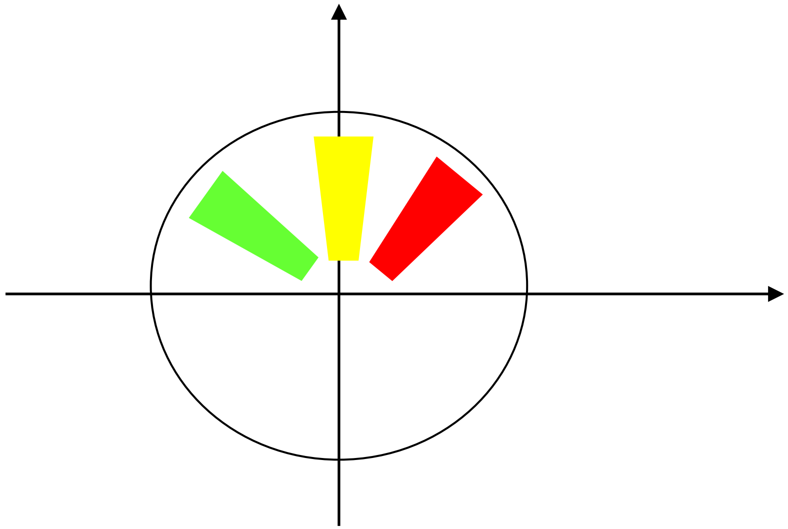
More on Similarity Measures (cont'd)

- Other types of similarity measures

$$s(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^t \mathbf{x}'}{\|\mathbf{x}\| \|\mathbf{x}'\|}$$

- E.g., cosine similarity

- E.g., distance on a graph, like shortest path.



Clustering as Optimization

The sum-of-squared-error criterion/cost

- Let D_i be the subset of samples from class i.
- Let n_i be the number of samples in D_i, and $\mathbf{m}_i$ the mean of those samples

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{x}$$

- The sum of squared error is:

$$J_e = \sum_{i=1}^{C} \sum_{\mathbf{x} \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2$$

➔ Well-separated, compact data “clouds” tend to give small errors when the clusters coincide with the clouds.

Clustering as Optimization (cont'd)

$$J_e = \sum_{i=1}^{C} \sum_{\mathbf{x} \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2$$

- **An optimization problem to solve for finding a "good" clustering: to find the partition of the data that minimizes J_e**

- If the membership of a sample is determined by the distance to the means $\mathbf{m}_i$
  - Then the task is to find the optimal set of $\{m_i\}$
  - The problem is NP-hard.

k-Means Clustering

- **Input: Given n data samples**

- **Goal: Partition them into *k* clusters/sets D_i, with respective center/mean vectors $\mu_1, \mu_2, \ldots, \mu_k$, so as to minimize**

$$\sum_{i=1}^{k} \sum_{\mathbf{x} \in D_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

- **Comparing with the mixture models:**
  - Here we do “hard” assignment of the membership to a sample (simply based on its distance to the cluster center).

The Basic k-Means Algorithm

Given: n samples, a number k.

<u>Begin</u>

initialize μ_1, μ_2, …, μ_k (randomly selected)

<u>do</u> classify n samples according to nearest μ_i

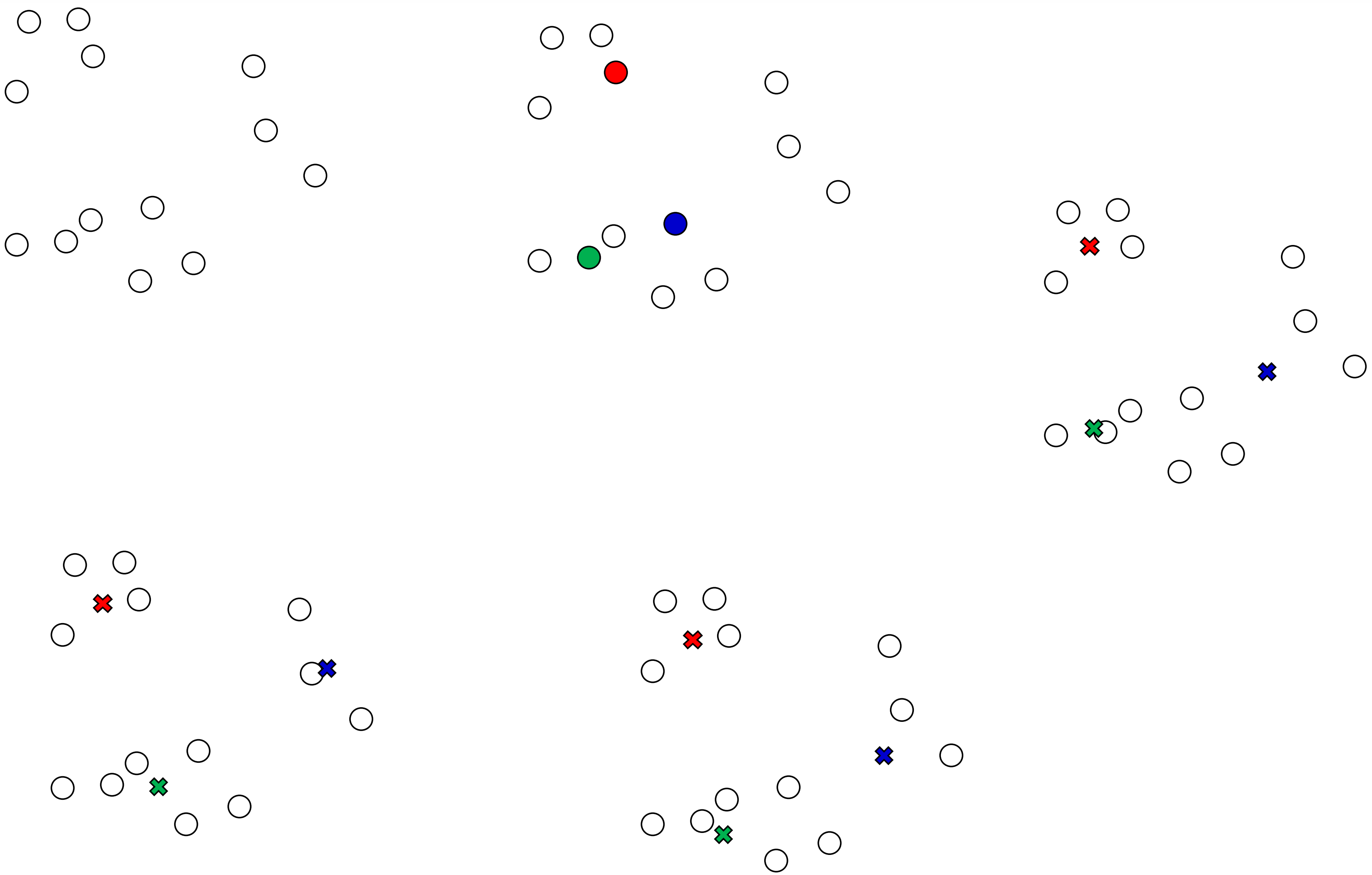
recompute μ_i

<u>until</u> no change in μ_i

<u>return</u> μ_1, μ_2, …, μ_k

<u>End</u>

Illustrating the Algorithm



Unsupervised Learning
Analyzing the k-Means Algorithm

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Discuss the weaknesses of the k-means algorithm

Objective

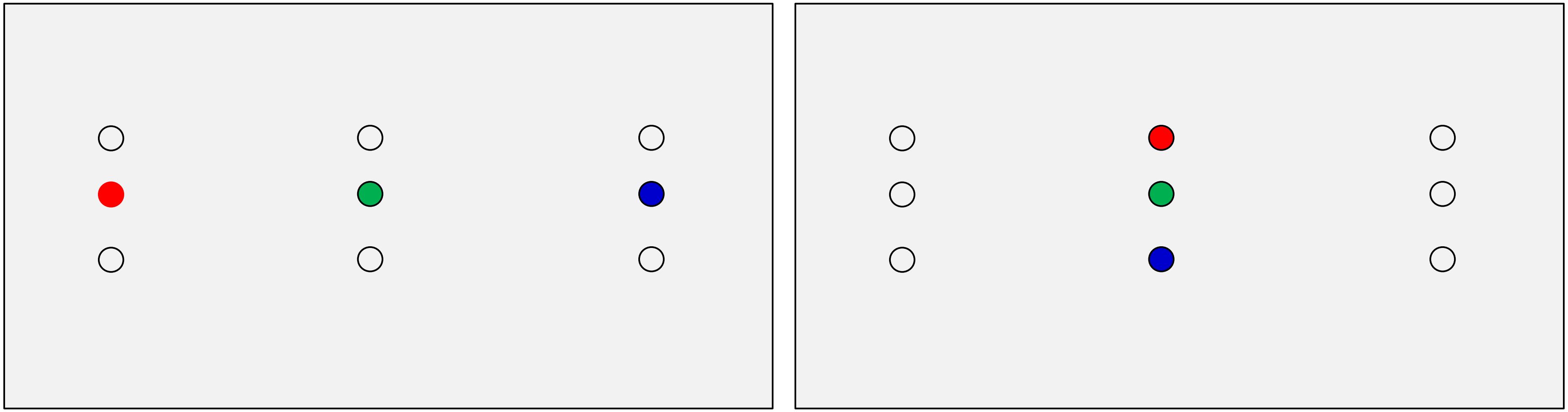
Discuss a few common techniques for potential improvement

Properties of the k-Means Algorithm

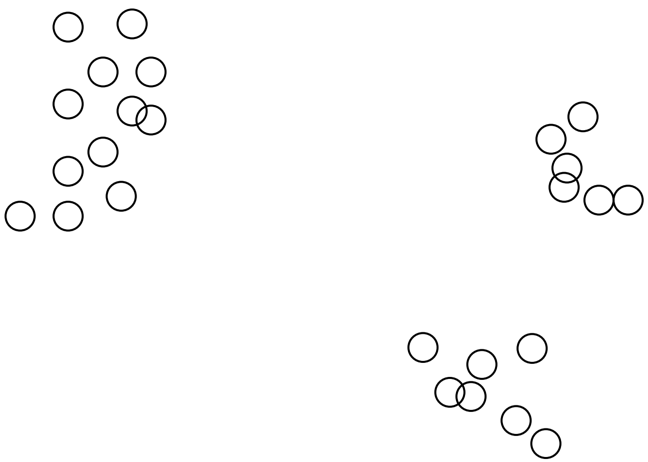
- The algorithm will converge when the cluster centers no longer change.

- But the results may not be an optimal solution.

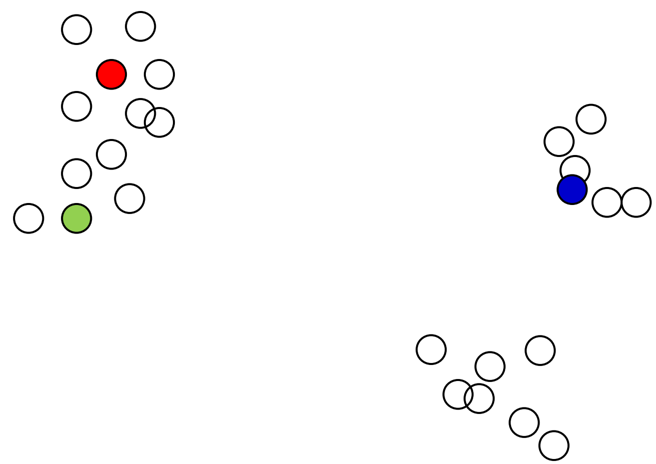
➔ **Sensitivity to initialization**



Another Example



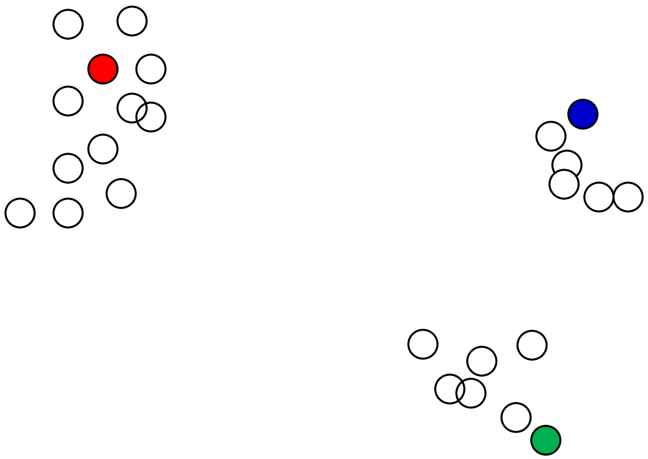
- The natural grouping seems to be so well defined.
- For k=3, what will be the clusters?



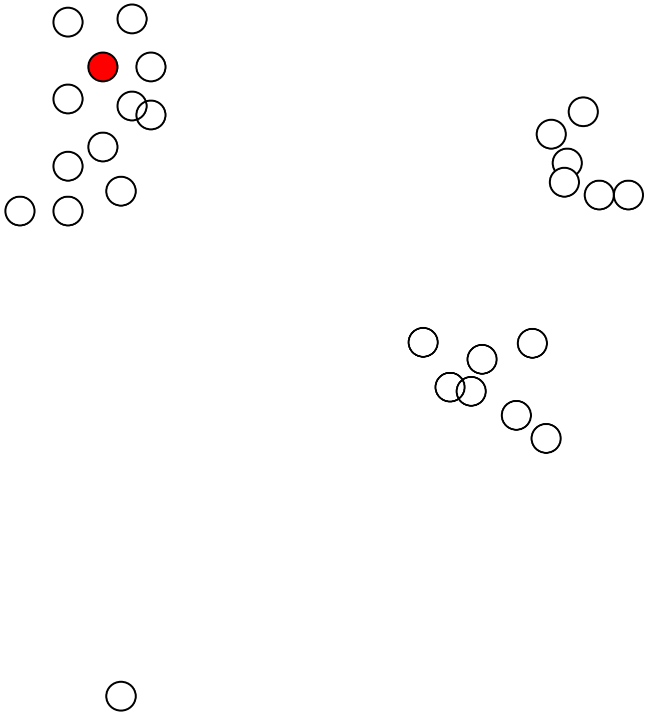
What can we do to improve?

A Few Common “Tricks”

- Choosing the point furthest from the previous centers.
  - Drawback: might be sensitive to “outlies”.



- Multiple runs with different initial centers.



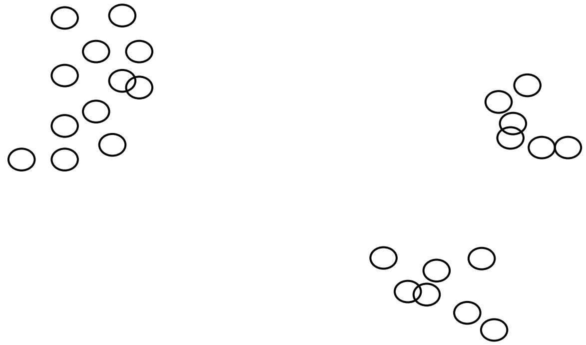
Other Variants of Basic k-Means

k-Means++:

- New centers are chosen with probabilities (as a function of distance to closest prior centers).
- Kind of between “random” and “furthest point” techniques.

Hierarchical approaches

- Agglomerative vs divisive.



The Question of Choosing k

Two trivial extremes

- If k=1, the error is the variance of the samples.
- If k=n, the error can become 0.

What is a proper 1<k<n for capturing the structure of the samples?

Some tricks

- Trick 1: Will the cost function drop dramatically at some point?
- Trick 2: Cross-validation (on, e.g., a classification task)