ECEN 758 Data Mining and Analysis: Lecture 6, Representative Clustering I

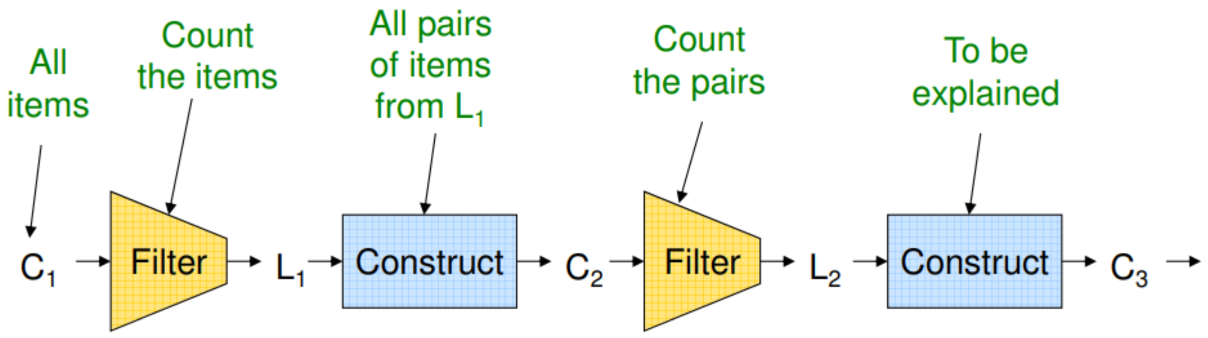
Joshua Peeples, Ph.D.
Assistant Professor
Department of Electrical and Computer Engineering

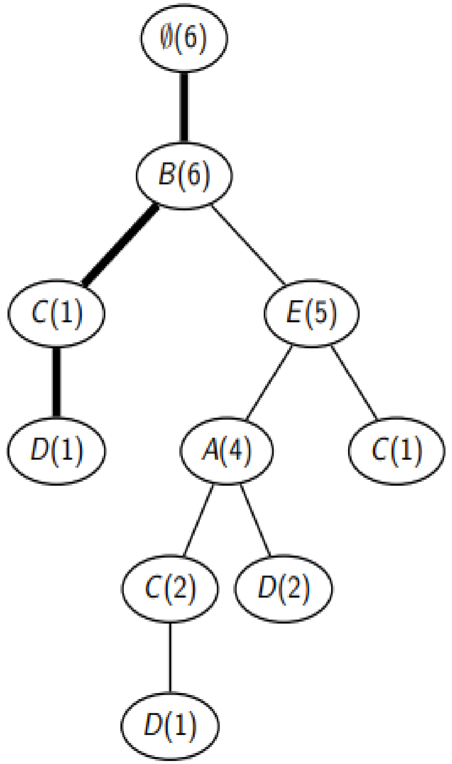
Announcements

- Solutions for Assignment #1 will be available Wednesday
- For future assignments
  - Please upload submission as single PDF
  - Please upload Python code (.py, ipynb)
  - **Do not include screenshots of code in submission**

Last Lecture

- Frequent itemset mining and association rules





Today

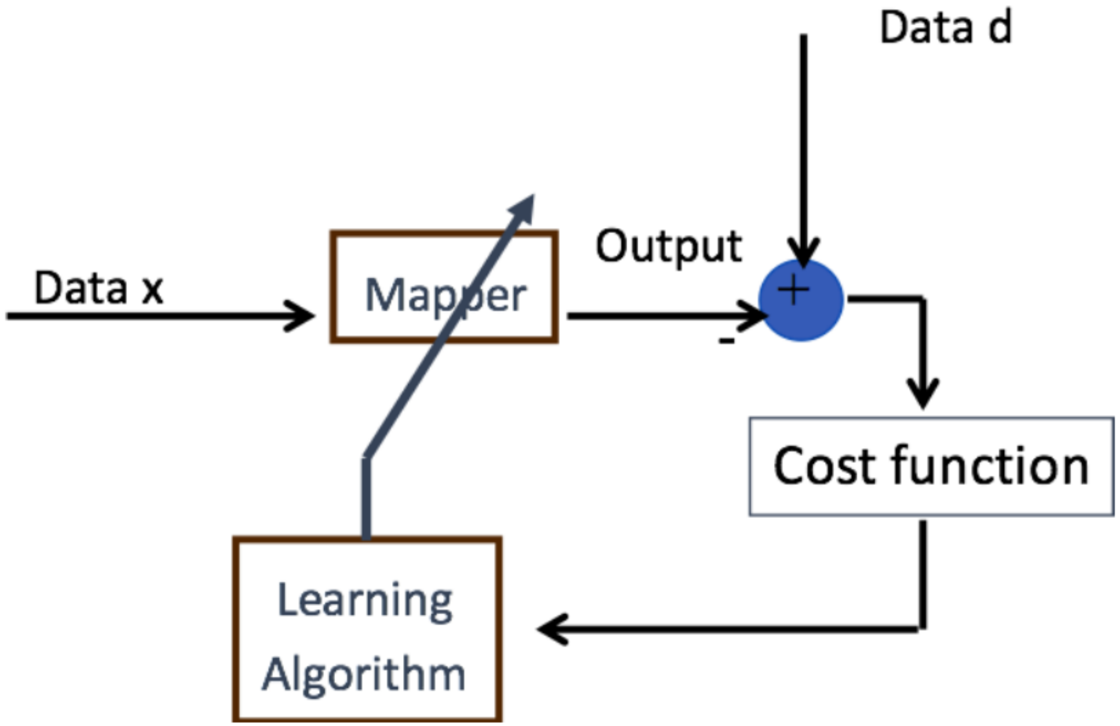
TEXAS A&M UNIVERSITY Engineering

- Representative Clustering I
- Reading: MMDS Chapter 7
- Supplemental reading: ZM Chapter 13 and 17

Review of Machine Learning

Machine Learning Model

- In machine learning the model is derived from the data (observations)
- As a learning machine, the model can be modified over time, with additional data (observations), with the goal of improving outcomes



Many Sub-areas in Machine Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Semi-supervised Learning
- Self-supervised Learning
- Multiple Instance Learning
- Active Learning
- Transfer Learning
- ….

Many Sub-areas in Machine Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Semi-supervised Learning
- Self-supervised Learning
- Multiple Instance Learning
- Active Learning
- Transfer Learning
- ….

Types of Learning

- Supervised learning
  - We “coach” the computer
  - Uses knowledge already learned
- Unsupervised learning
  - “We’re free!!”

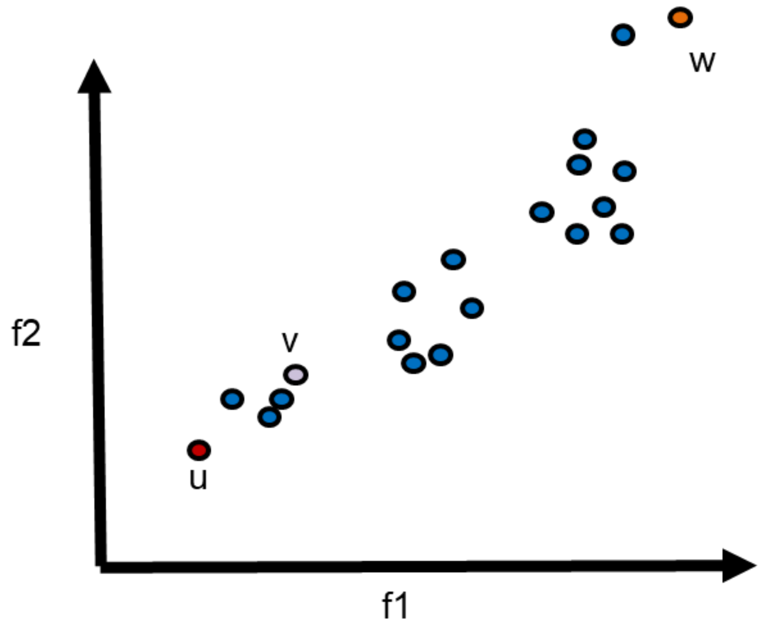
Unsupervised Learning: Clustering

Clustering Overview

- Clustering:
  - Unsupervised learning – just data, no labels
  - Similarity/Dissimilarity in the data
  - Can provide insights when we have no preconception of data

Clustering Overview

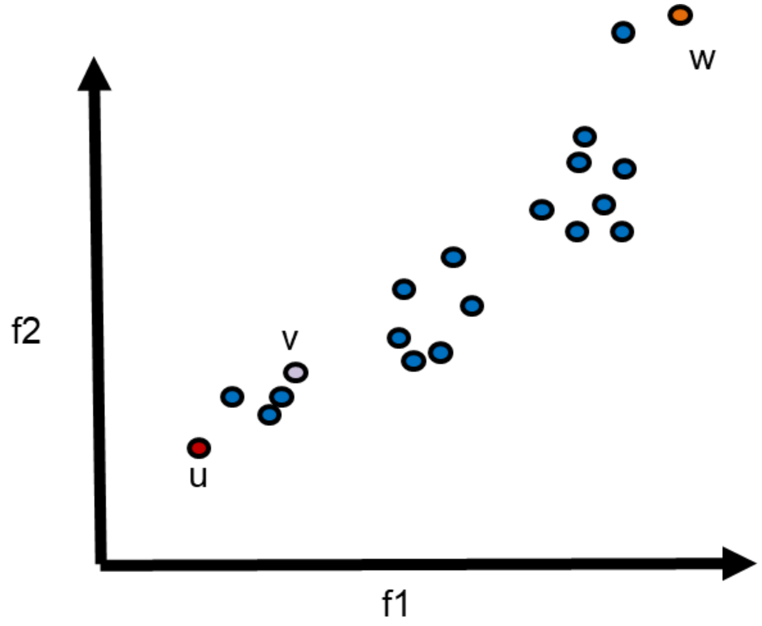
- Basic idea: group together similar instances
- Example: Use total squared Euclidean distance as similarity (or dissimilarity) metric for 2D point patterns



Clustering Overview

- Is observation u more similar to v or w?
- Similarity (or dissimilarity) of u to v is square distance between them:

$$SqDist(\boldsymbol{u},\boldsymbol{v}) = \left(v_{f1} - u_{f1}\right)^2 + \left(v_{f2} - u_{f2}\right)^2$$



Clustering Overview

- Clustering may appear straightforward
  - 2D
  - Small amounts of data
- Typically have large feature space
  - Difficult to tell differences between data points

Images from: Sloan Digital Sky Survey

Clustering Overview

- We will discuss several variants of clustering
  - Representative-based Clustering
  - Hierarchical Clustering
  - Density-Based Clustering

Clustering Overview

- We will discuss several variants of clustering
  - **Representative-based Clustering**
  - Hierarchical Clustering
  - Density-Based Clustering

Representative-based Clustering

Representative-based Clustering

- Goal: partition data into *k* groups or clusters
- Clusters:
  - Representative of data points in group (also called centroid)
  - Common choice is mean
- Brute force solution not ideal
  - Generate all possible partitions

$$\boldsymbol{D} = \begin{pmatrix} X_1 & X_2 & \cdots & X_d \\ \hline x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

$$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$$

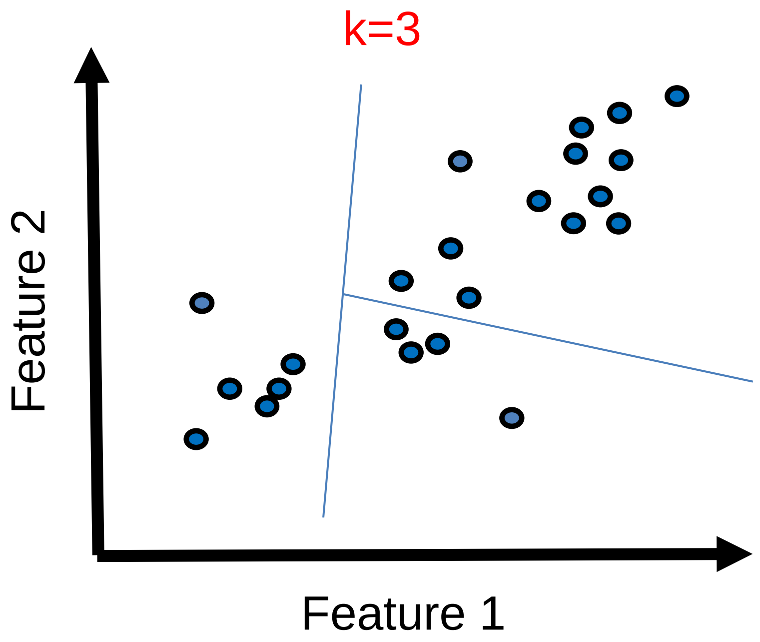
$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{x_j \in C_i} \boldsymbol{x}_j$$

k-Means Clustering

k-Means Clustering

TEXAS A&M UNIVERSITY Engineering

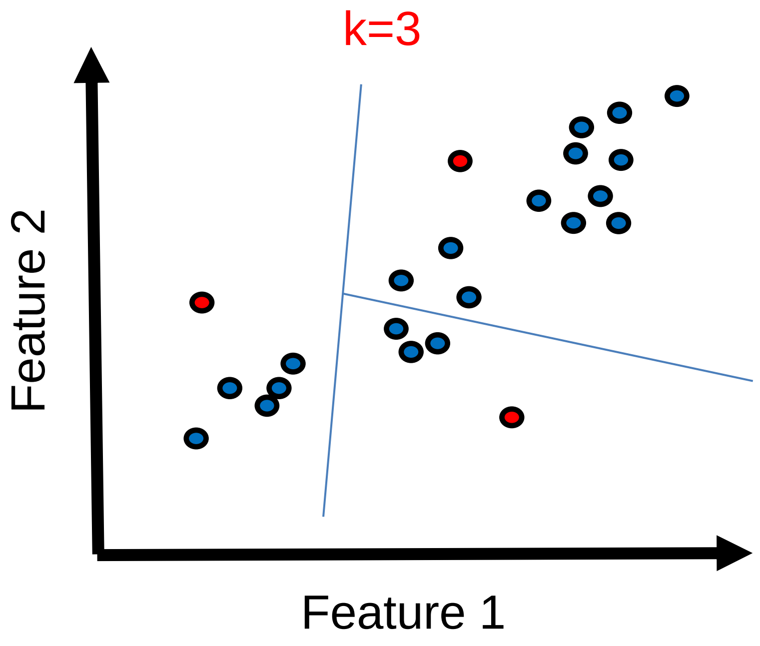
- Basic idea: use distance to group together similar instances
- k-means clustering algorithm
  - Choose k



k-Means Clustering

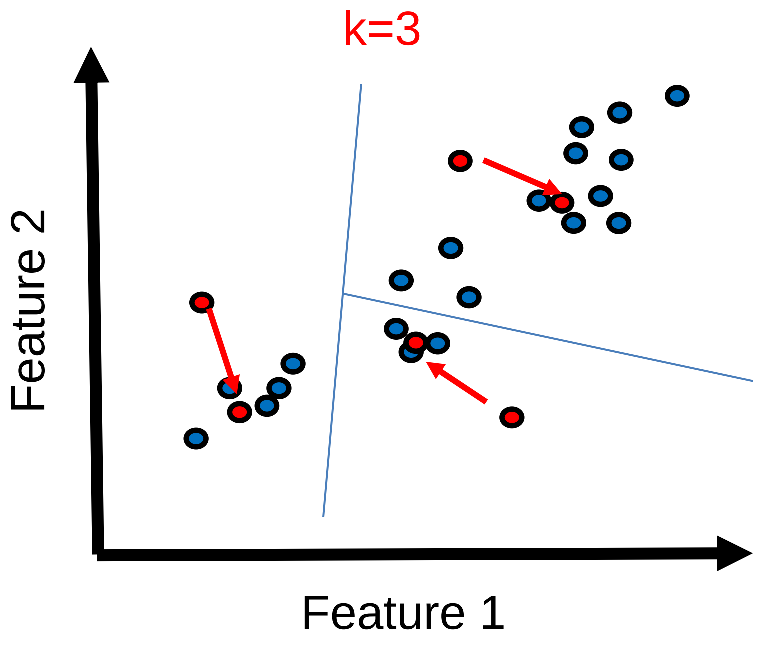
TEXAS A&M UNIVERSITY Engineering

- Basic idea: use distance to group together similar instances
- k-means clustering algorithm
  - Choose k
  - Assign k random points as estimate of cluster centers, c_k (means)

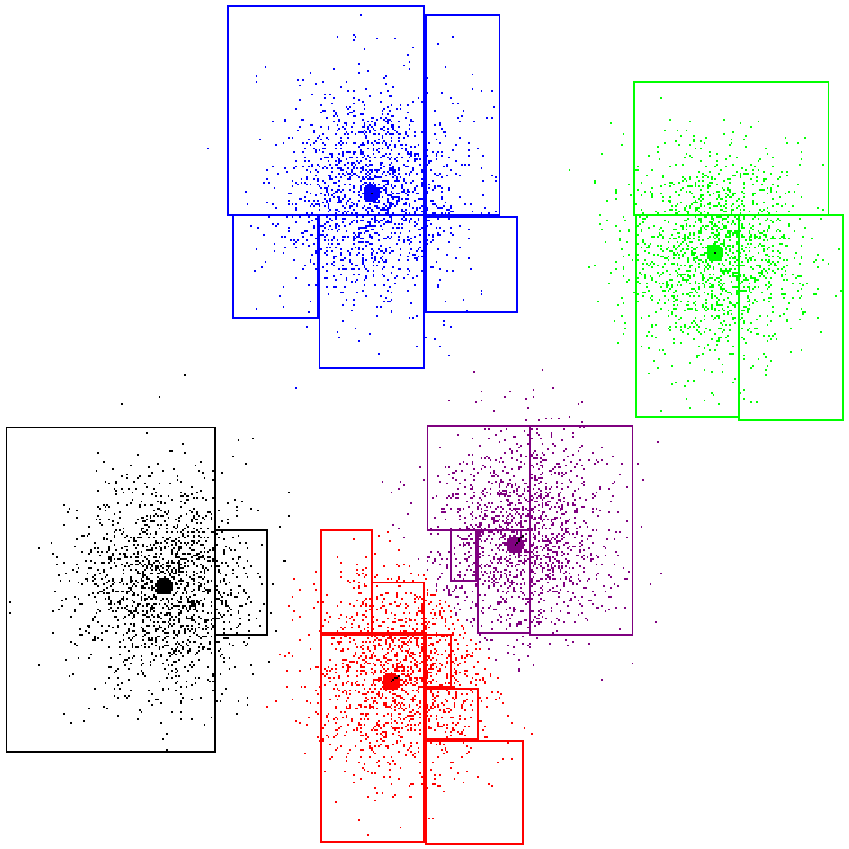


k-Means Clustering

- Basic idea: use distance to group together similar instances
- k-means clustering algorithm
  - Choose k
  - Assign k random points as estimate of cluster centers, c_k (means)
  - Alternate between:
    - 1) Assign data instances to closest mean
    - 2) Reassign each mean to the average of its newly assigned points
  - Stop when no points' assignments change



k-Means Example



k-Means Clustering Algorithm

k-Means Pseudocode

K-means $(\boldsymbol{D}, k, \epsilon)$:

$t = 0$

Randomly initialize k centroids: $\boldsymbol{\mu}_1^t, \boldsymbol{\mu}_2^t, \ldots, \boldsymbol{\mu}_k^t \in \mathbb{R}^d$

repeat

  $t \leftarrow t + 1$

  $C_j \leftarrow \emptyset$ for all $j = 1, \cdots, k$

  // Cluster Assignment Step

  foreach $\boldsymbol{x}_j \in \boldsymbol{D}$ **do**

    $i^* \leftarrow \arg\min_i \left\{ \|\boldsymbol{x}_j - \boldsymbol{\mu}_i^t\|^2 \right\}$ // Assign $\boldsymbol{x}_j$ to closest centroid

    $C_{i^*} \leftarrow C_{i^*} \cup \{\boldsymbol{x}_j\}$

  // Centroid Update Step

  foreach $i = 1$ *to* k **do**

    $\boldsymbol{\mu}_i^t \leftarrow \frac{1}{|C_i|} \sum_{\boldsymbol{x}_j \in C_i} \boldsymbol{x}_j$

until $\sum_{i=1}^k \left\| \boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1} \right\|^2 \leq \epsilon$

k-Means Algorithm: Objective

- Sum of squared errors (SSE) objective function
- Goal: find clustering to minimize SSE
- Greedy iterative approach
  - Can converge to a local optima
- Two steps to achieve minima

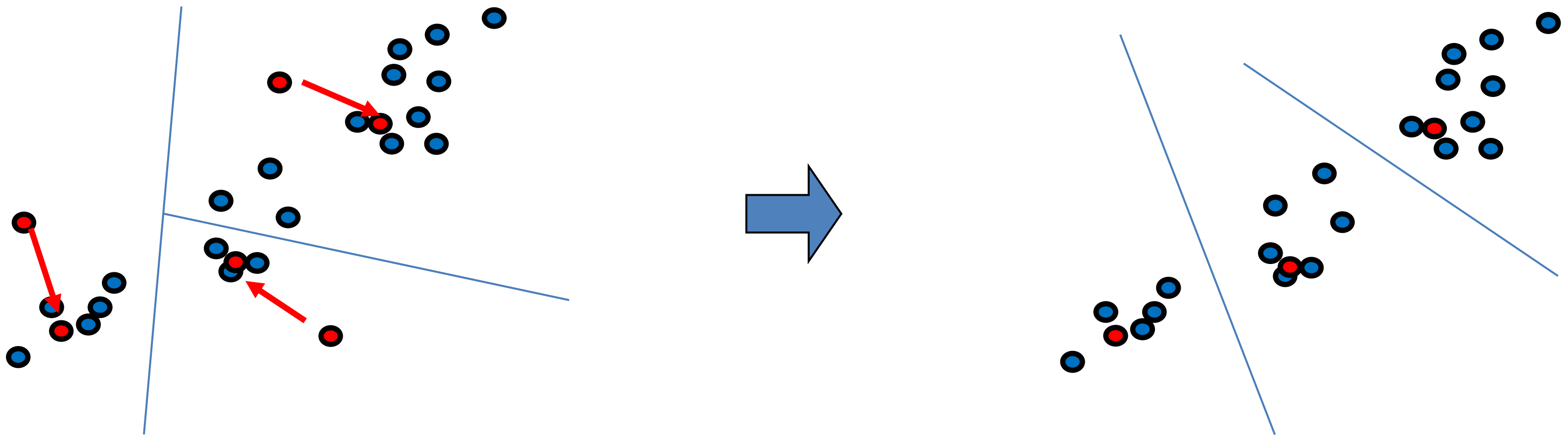
$$SSE(\mathcal{C}) = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \{SSE(\mathcal{C})\}$$

Phase I: Update Assignments

- For each point, re-assign to closest mean: $a_{ij} = \underset{k}{argmin}\, dist(x_i, c_k)$

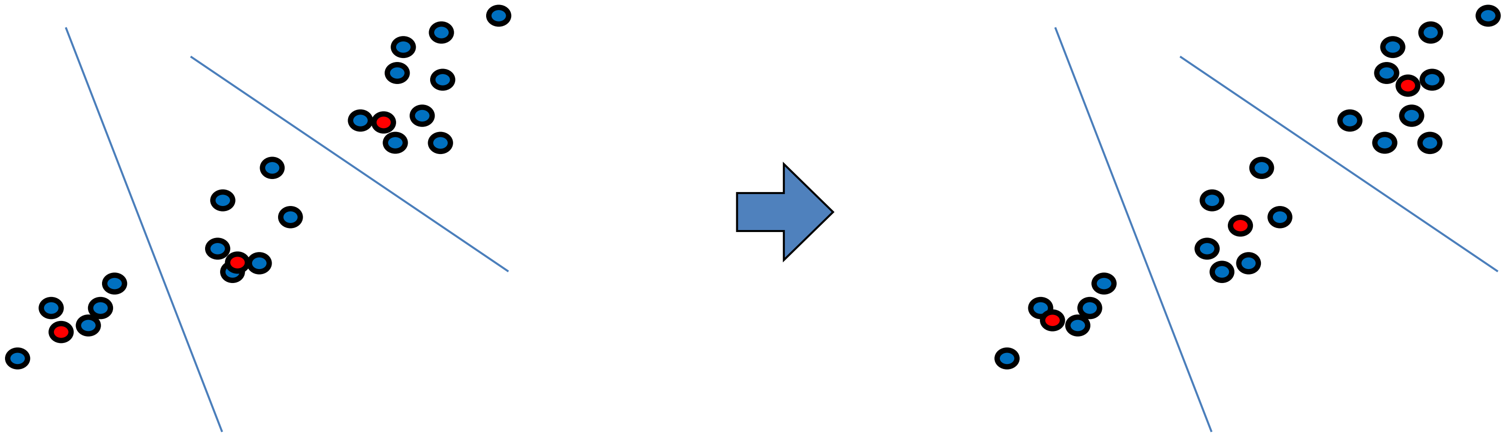
- Choose among $[c_1, .. \, c_k]$ the mean which minimizes the distance between x_i and c_k, and assign that value of [1..k] to a_{ij}



Phase II: Update Means

- Move each mean to the average of its assigned points:
- Select the points which are assigned to the mean point c_k (i.e. those with $a_{ij} = k$.) Average these points and assign that new value to c_k

$$c_k = \frac{1}{|\{i: a_{ij} = k\}|} \sum_{i:a_{ij}=k} \boldsymbol{x}_i$$



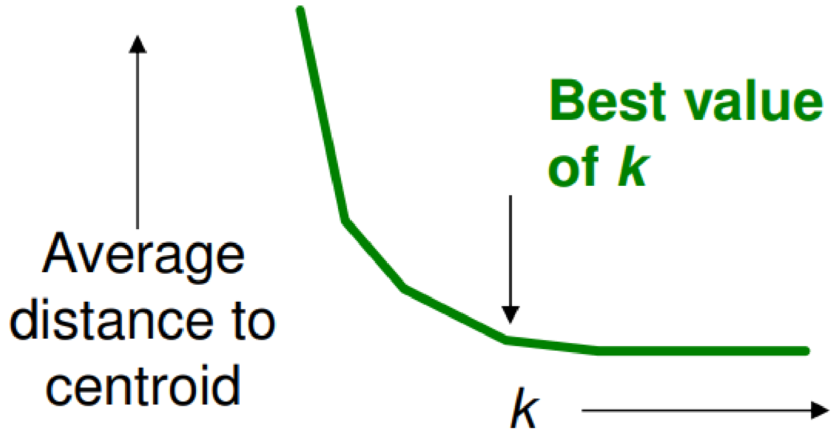
k-Means Algorithm Choices

k-Means Consideration

- **Selecting number of clusters (*k*)**
- Initialization

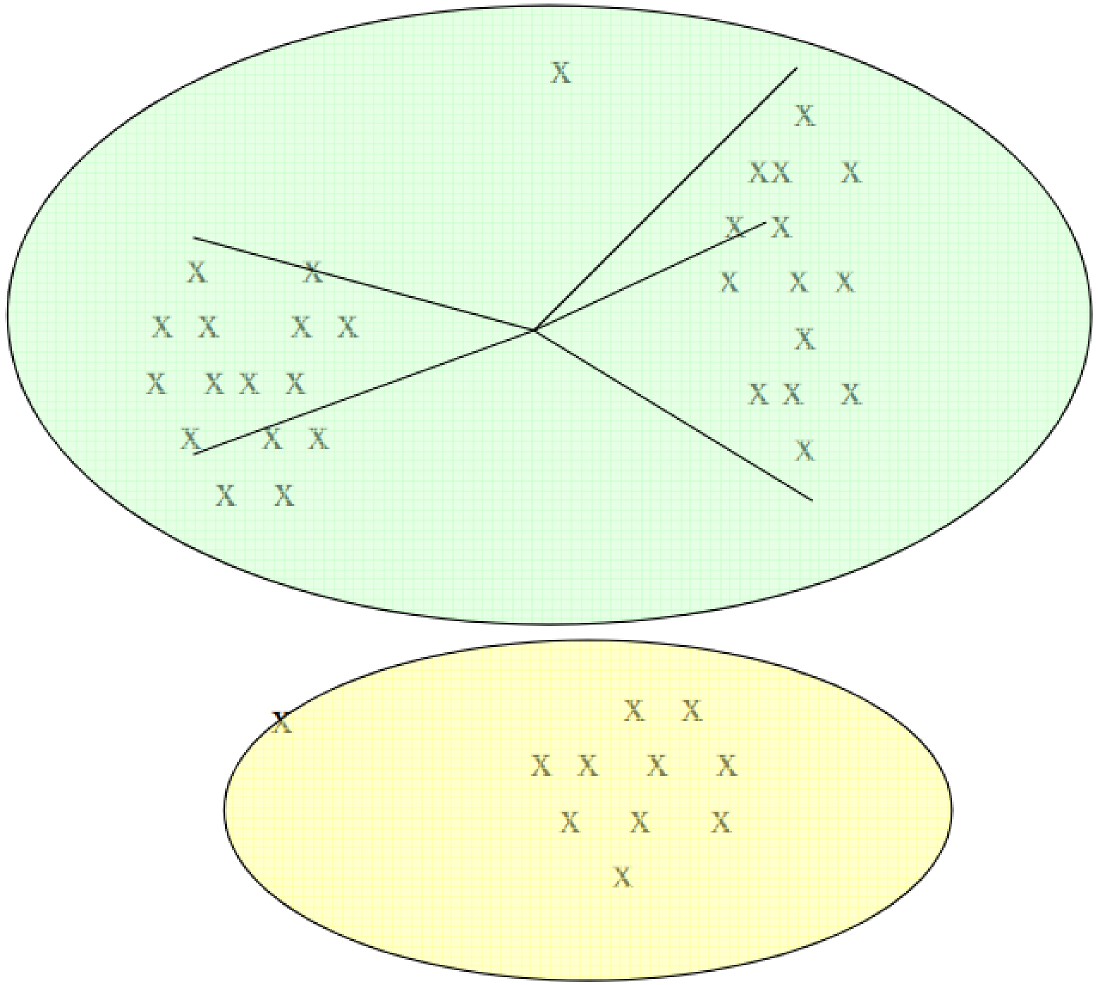
Choosing *k*

- *k* is hyperparameter to determine number of clusters
- Results heavily dependent on *k*
- Selecting *k*
  - Try different values and look at change in average distance to centroid
    - Average falls rapidly until right *k*, then changes little (“elbow method”)



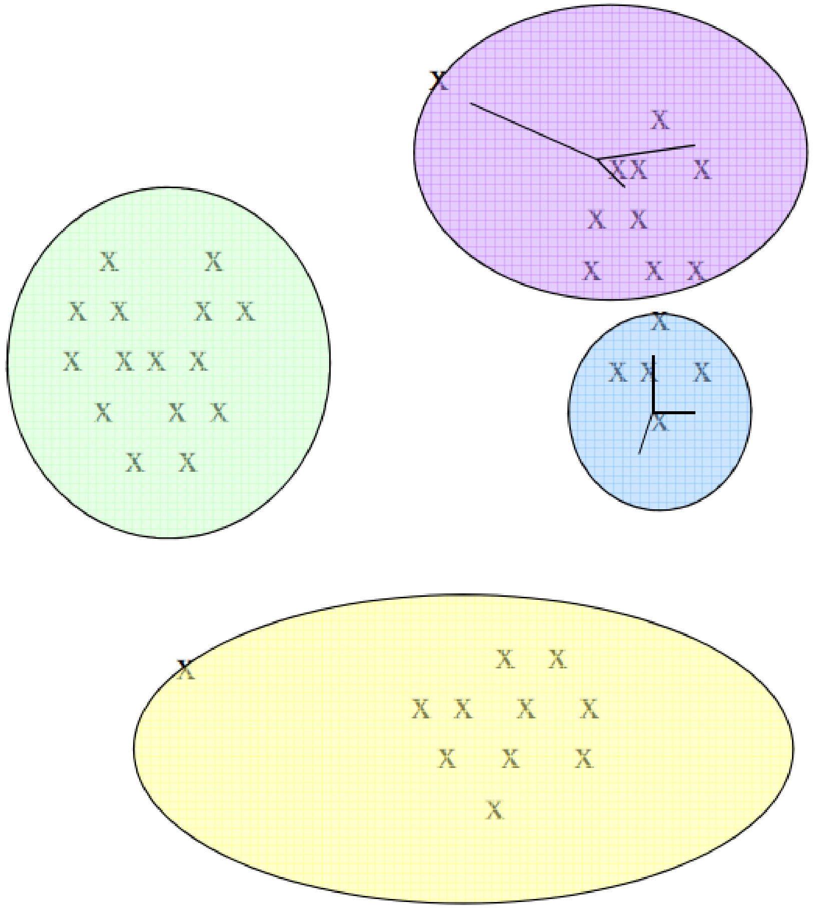
Too few *k*

Too few;
many long
distances
to centroid.



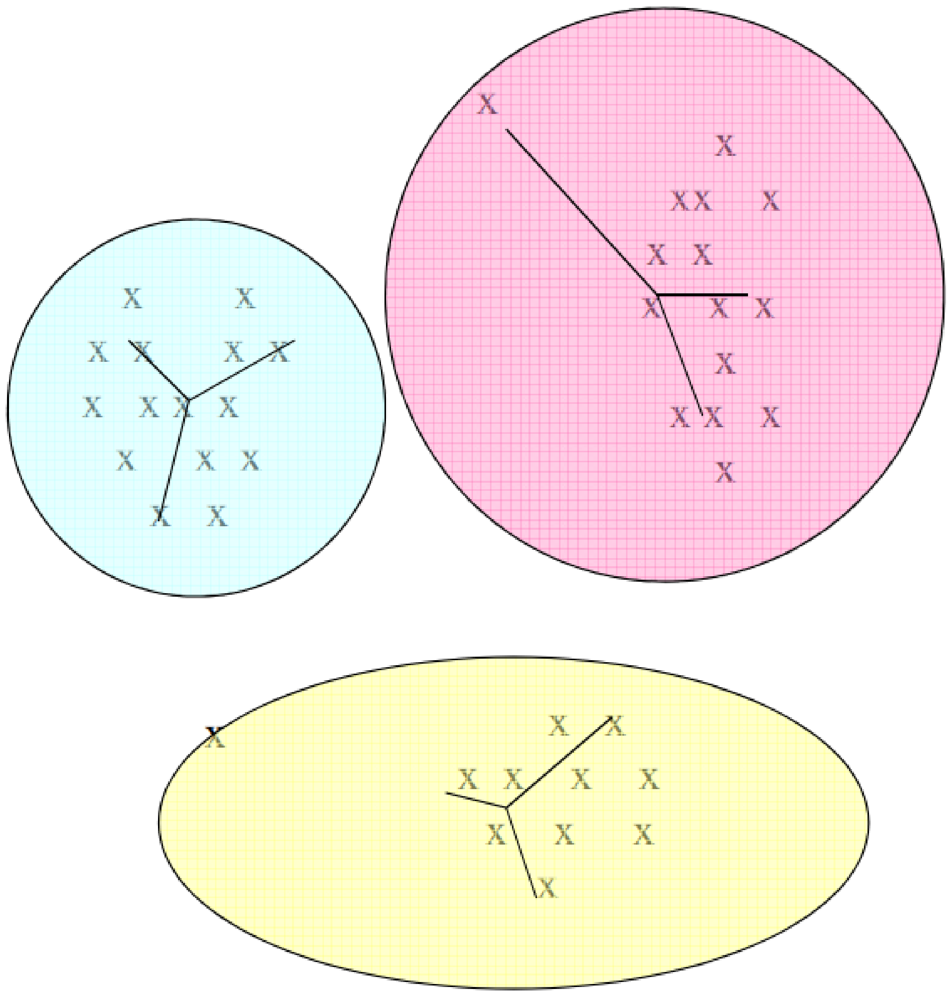
Too many *k*

Too many;
little improvement
in average
distance.



Optimal *k*

Just right;
distances
rather short.

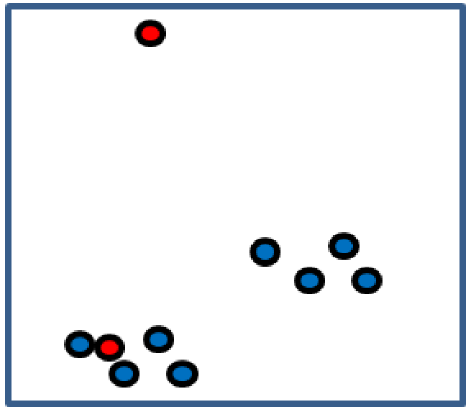


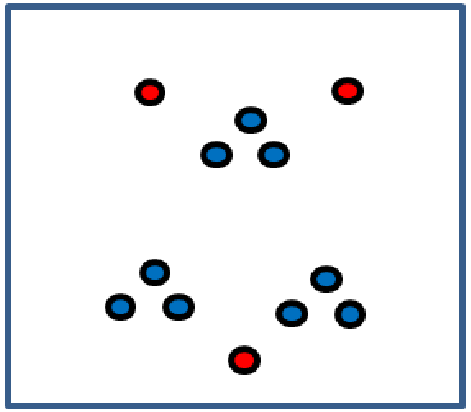
k-Means Consideration

- Selecting number of clusters (*k*)
- **Initialization**

Initialization

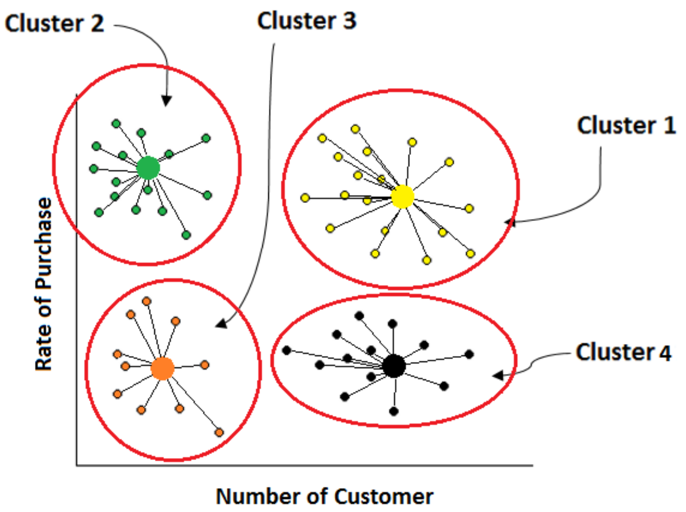
- Result depends on initial location of the k-means
  - What can go wrong?
  - Most k-means solvers include initialization heuristics to minimize these issues

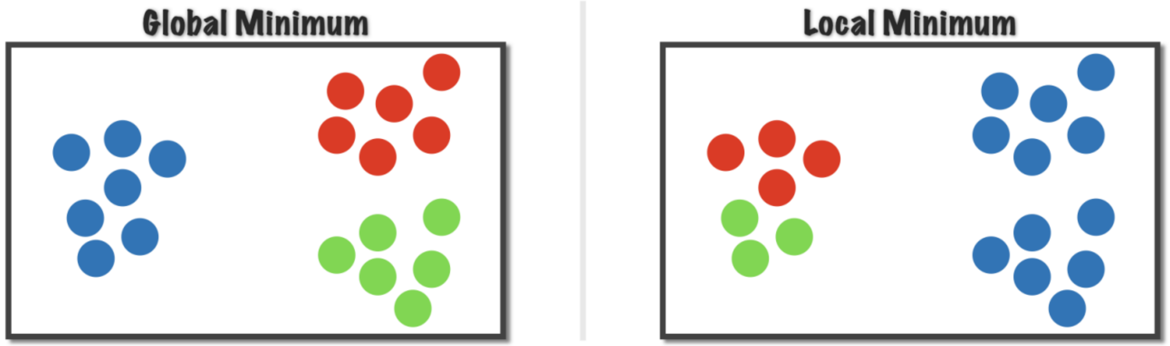




Initialization

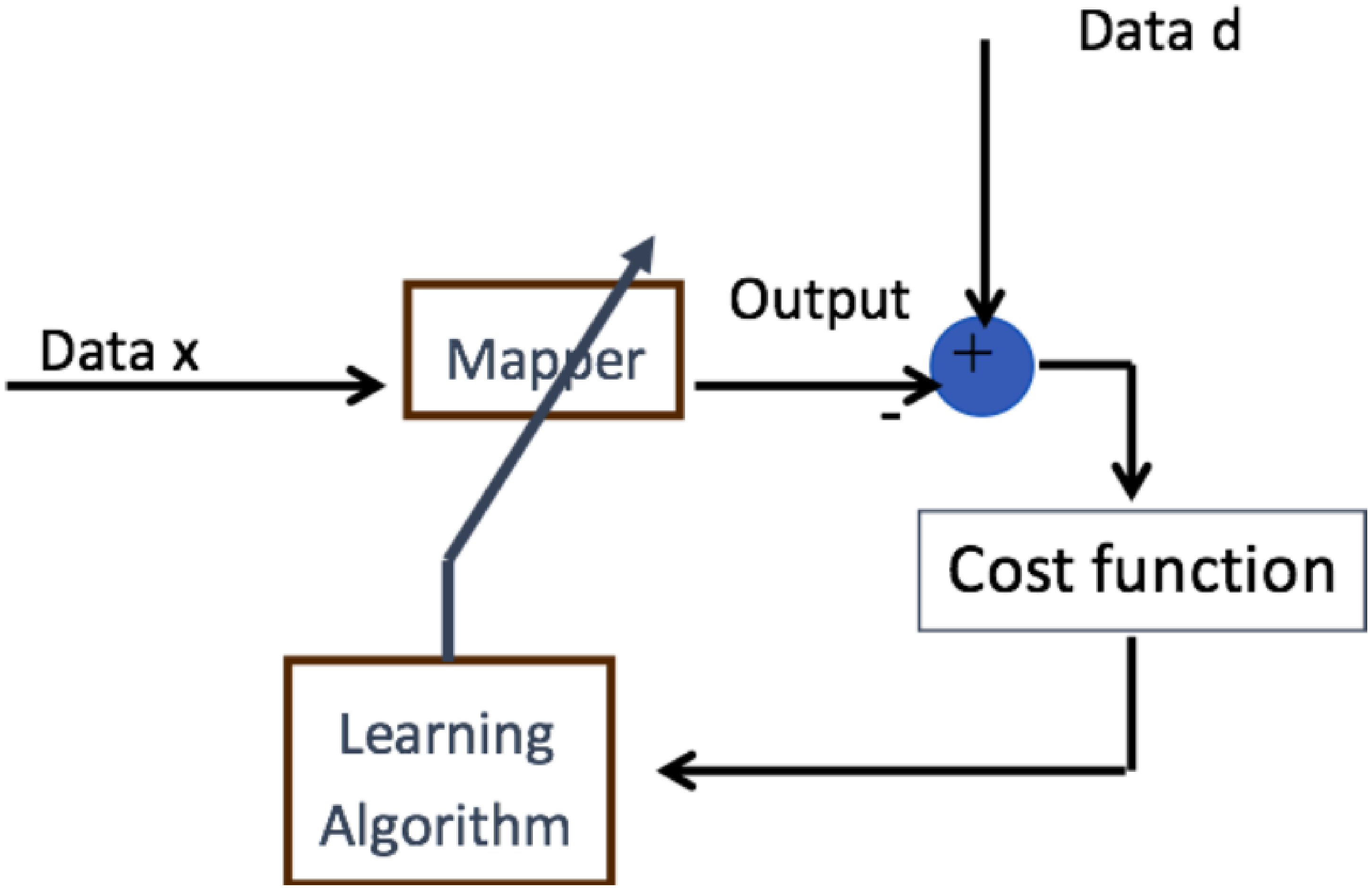
- Will it always locate the clusters in the same location and assign the same data to each mean? i.e. deterministic?
- Is k-means guaranteed to find the solution with the lowest total distance to the means? i.e. global optimum solution?





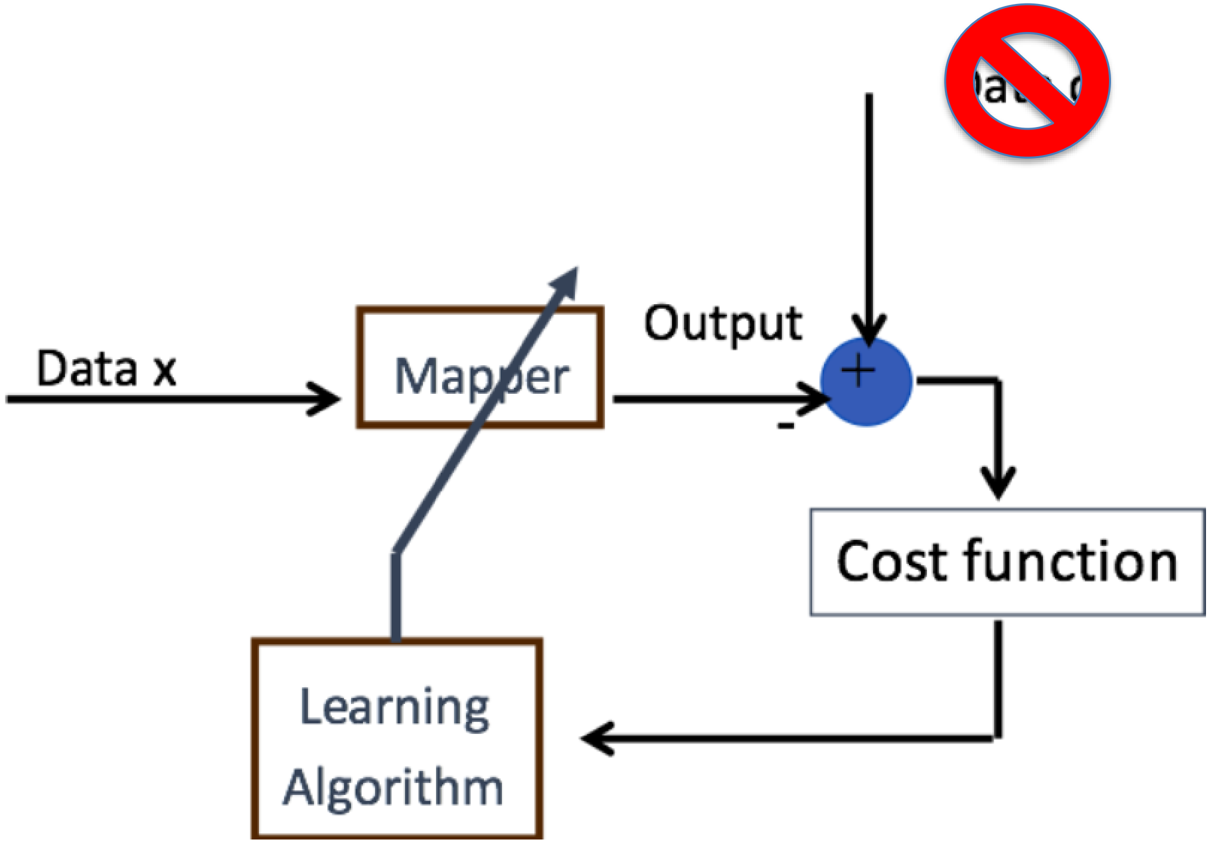
Images from: Amit Chauchan, Fully Explained K-means Clustering with Python and Alan Jeffares, K-means: A Complete Introduction

k-Means Machine Learning Model



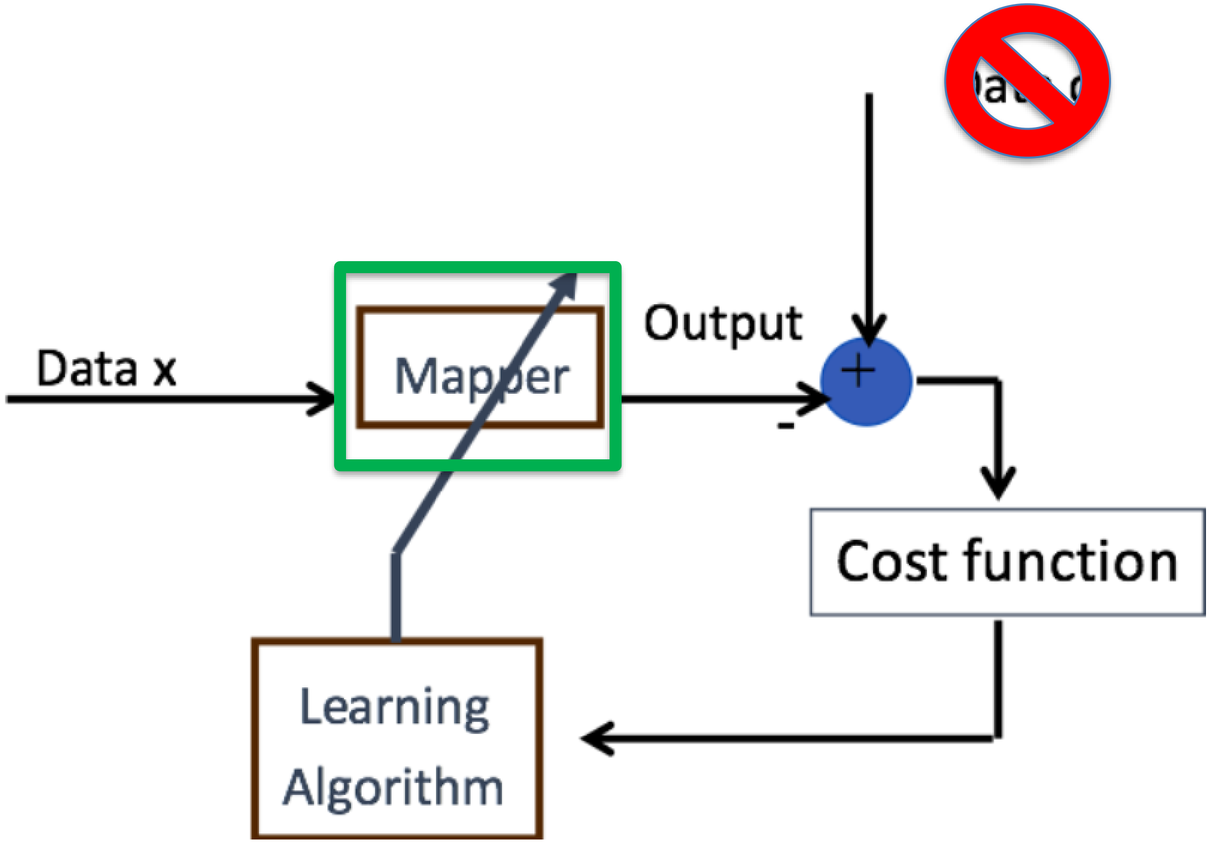
k-Means Machine Learning Model

- Unsupervised: No labels, *d*



k-Means Machine Learning Model

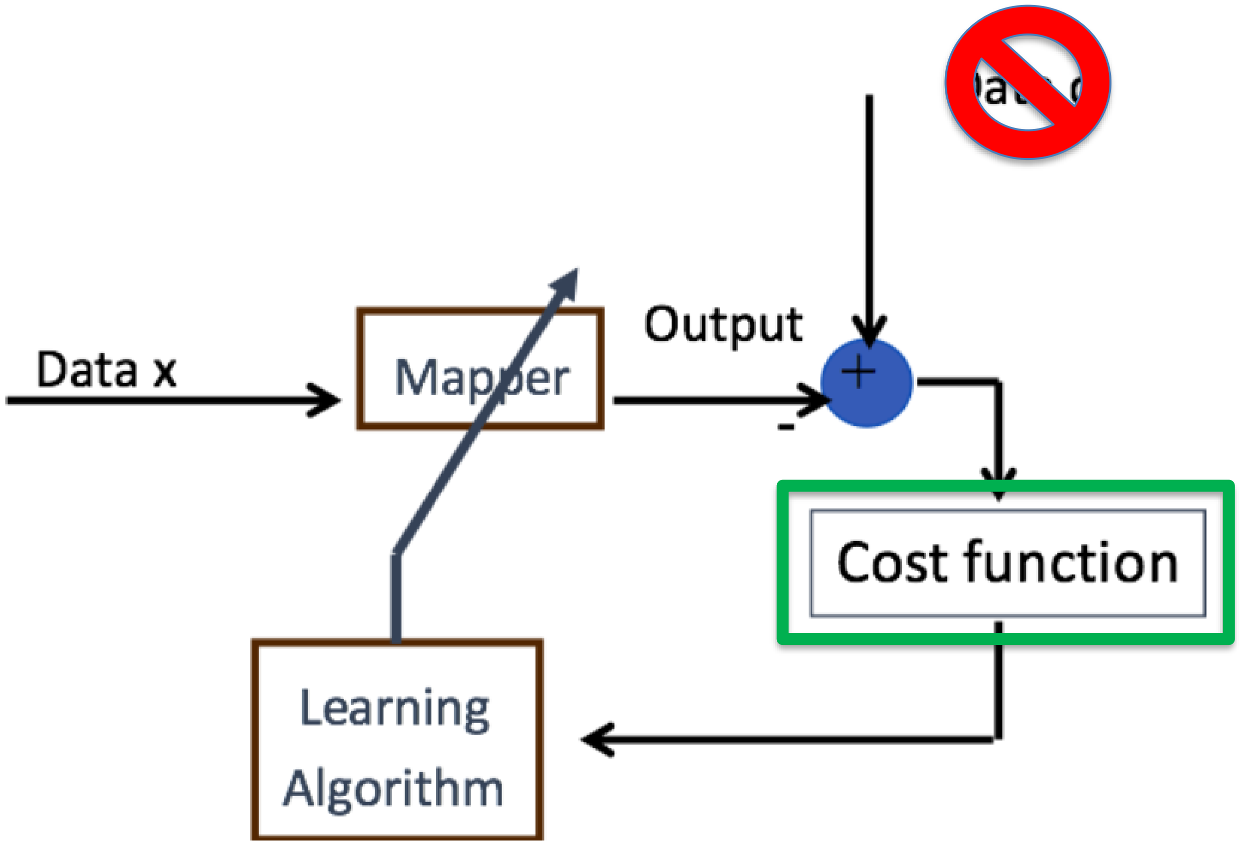
- Unsupervised: No labels, *d*
- **Mapper:**
  - k-means algorithm
  - Takes input data and groups into *k* clusters



k-Means Machine Learning Model

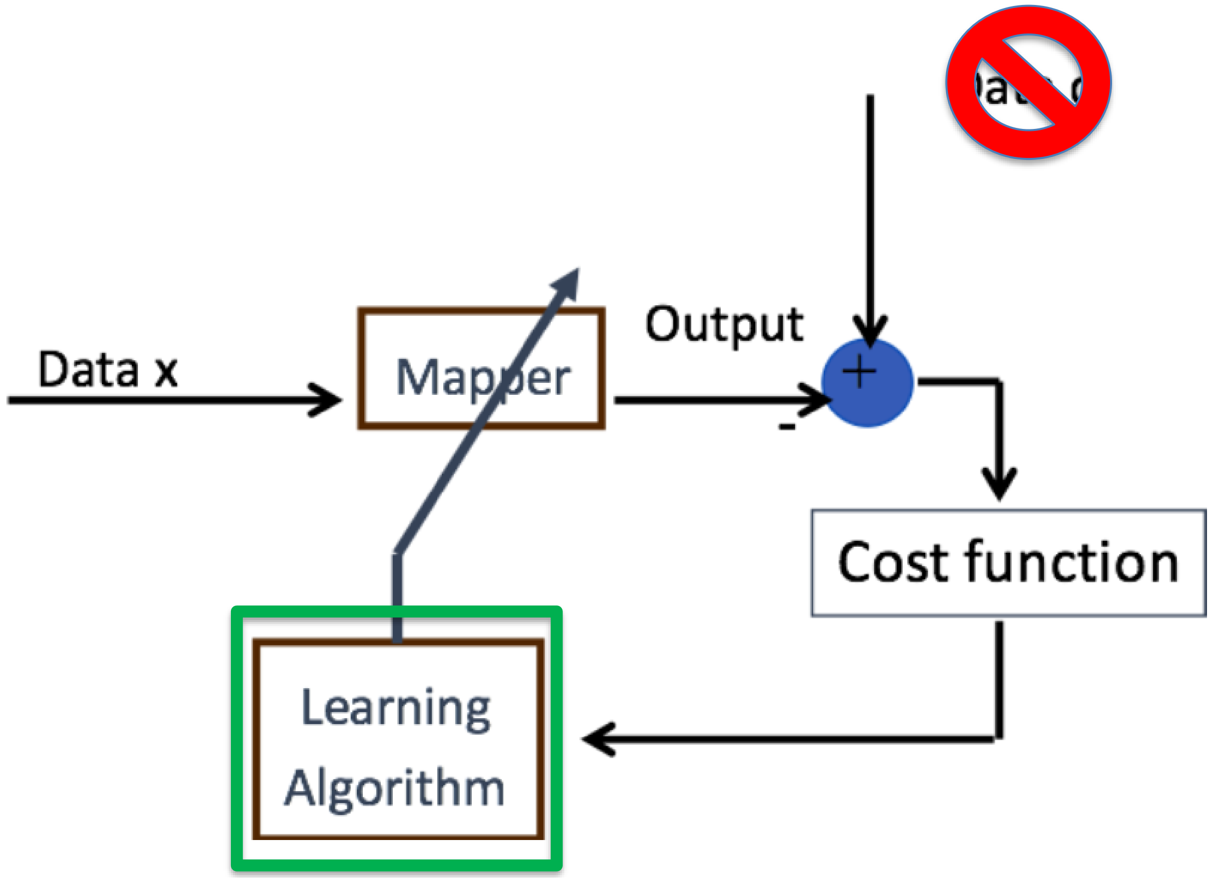
- Unsupervised: No labels, *d*
- Mapper:
  - k-means algorithm
  - Takes input data and groups into *k* clusters
- **Cost function:**
  - Sum of squared errors (SSE)

$$SSE(\mathcal{C}) = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

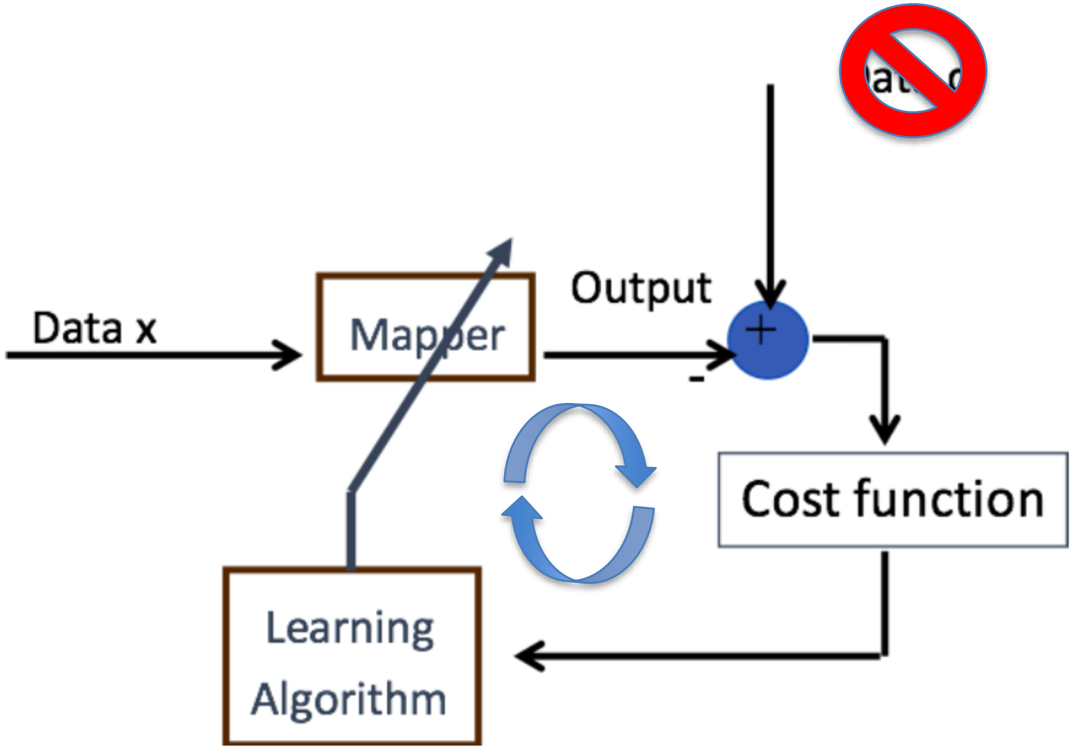


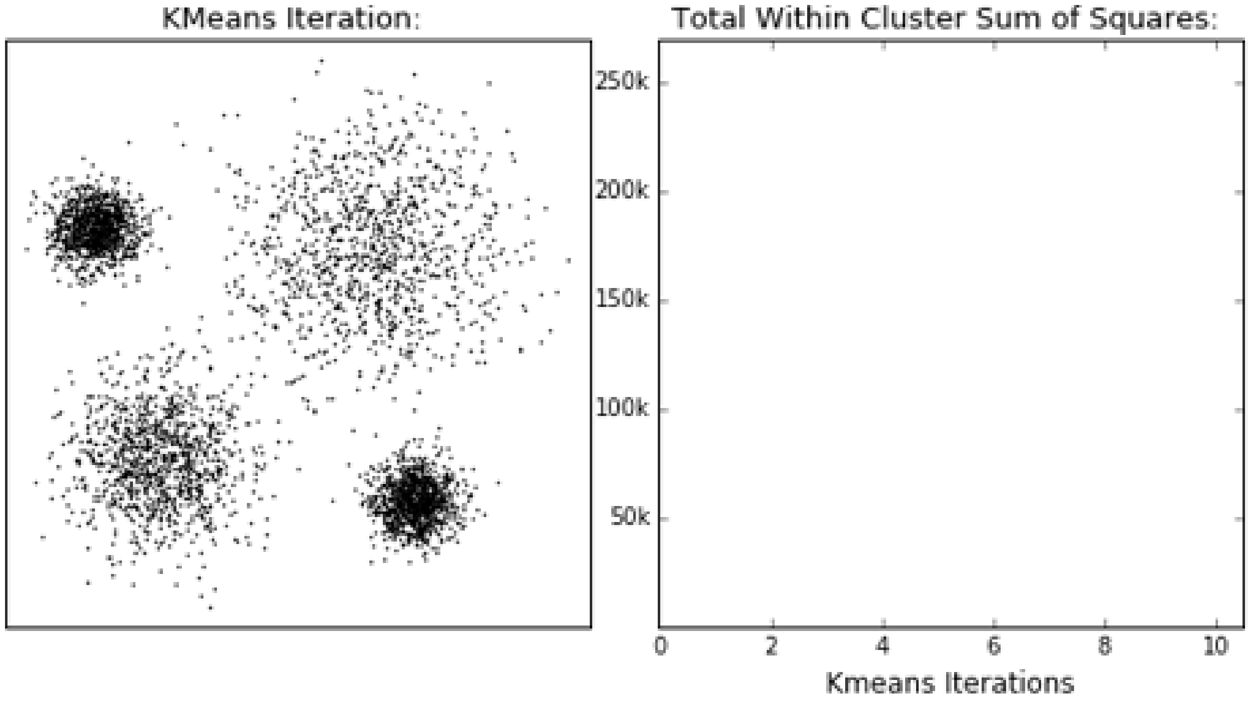
k-Means Machine Learning Model

- Unsupervised: No labels, *d*
- Mapper:
  - k-means algorithm
  - Takes input data and groups into *k* clusters
- Cost function:
  - Sum of squared errors (SSE)
- **Learning algorithm**
  - Update cluster assignments
  - Update centroids



k-Means Machine Learning Model





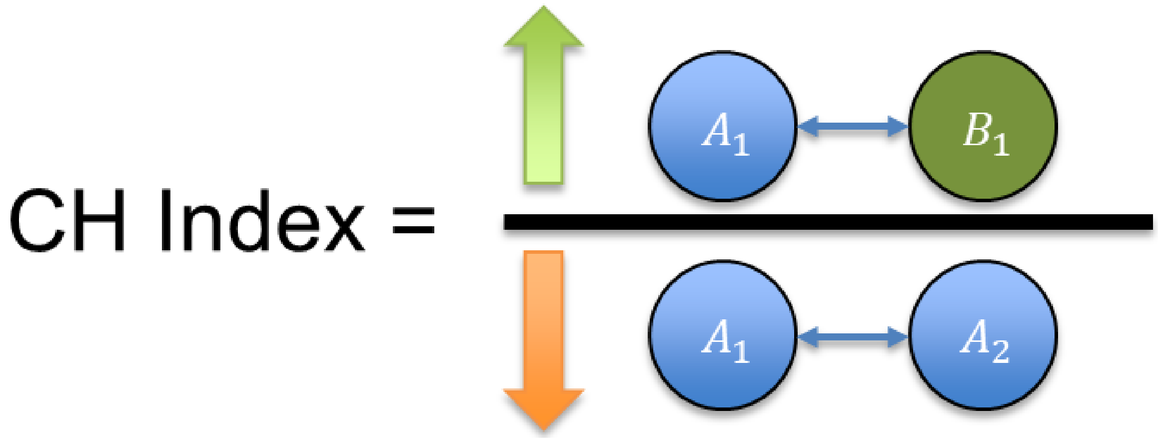
Gif from: D. Sheehan, Clustering with Scikit with GIFs

Cluster Evaluation

Evaluating Clustering

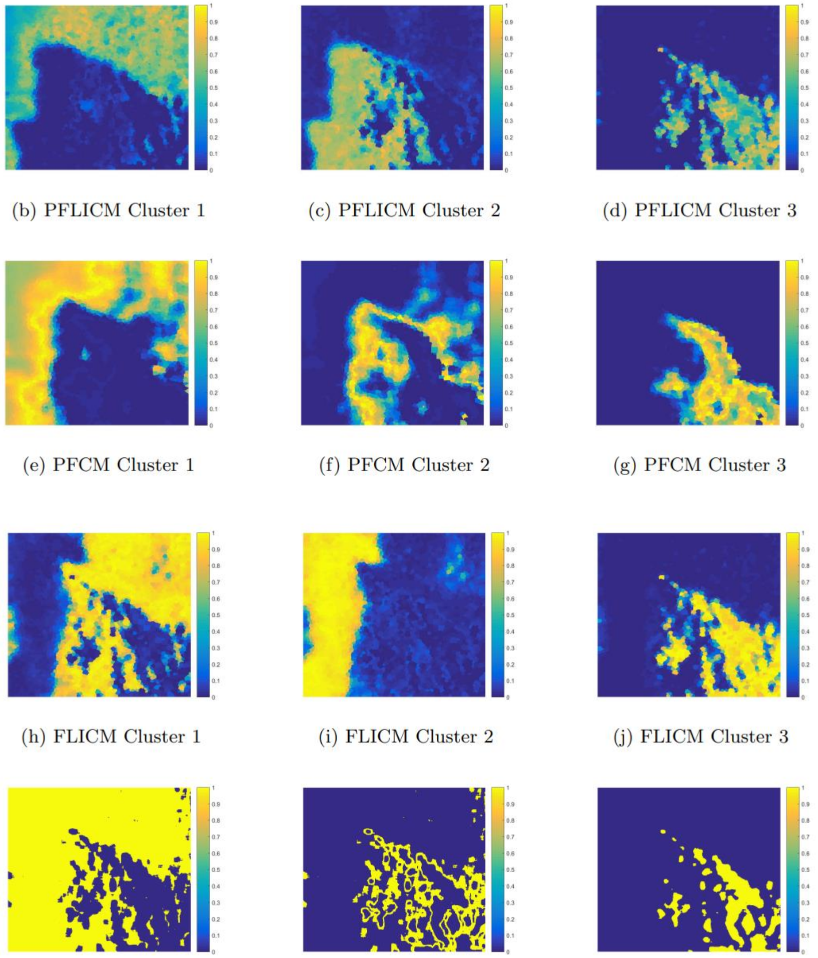
TEXAS A&M UNIVERSITY Engineering

- Two important measures:
  - Intra-cluster compactness
  - Inter-cluster separability
- Various indices to capture metrics
  - Silhouette index
  - **Calinski-Harabasz (CH) index**
  - Davie-Bouldin (DB) index
  - Dunn index
- More details in ZM Chapter 17!



Sonar Image Segmentation

Input Image



(b) PFLICM Cluster 1 (c) PFLICM Cluster 2 (d) PFLICM Cluster 3

(e) PFCM Cluster 1 (f) PFCM Cluster 2 (g) PFCM Cluster 3

(h) FLICM Cluster 1 (i) FLICM Cluster 2 (j) FLICM Cluster 3

(k) K-Means Cluster 1 (l) K-Means Cluster 2 (m) K-Means Cluster 3

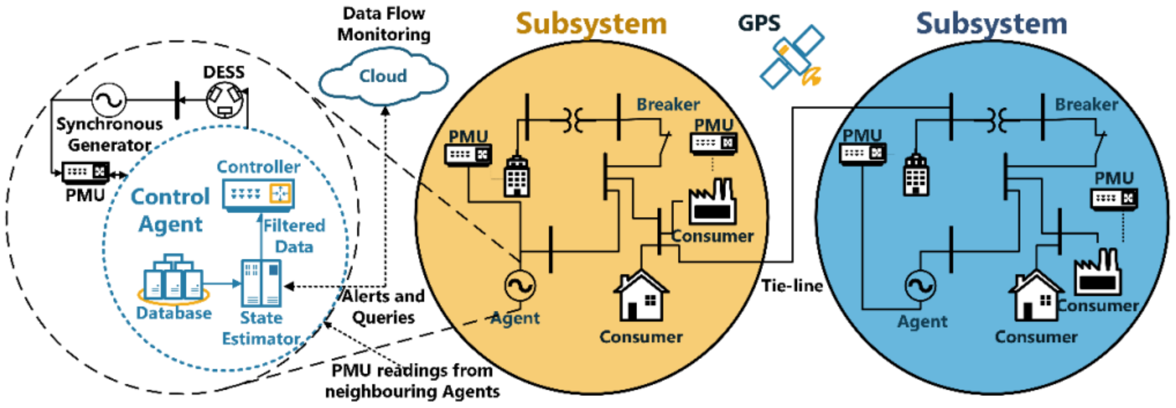
J. Peeples, et al. Possibilistic fuzzy local information C-means with automated feature selection for seafloor segmentation. SPIE, 2018.

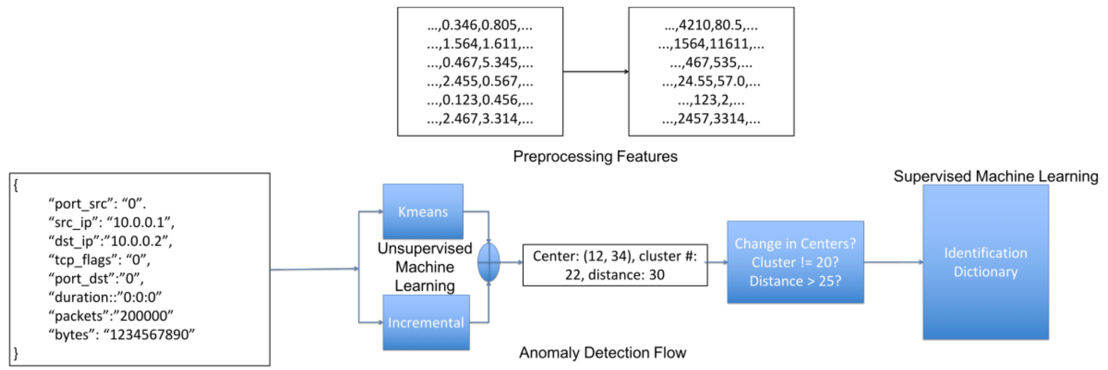
Semi-supervised Learning

k-Means for Semi-supervised learning

TEXAS A&M UNIVERSITY Engineering

- Cluster initial data
  - Unsupervised
- “Labels” clusters
- Take new data and assign to clusters
  - Supervised



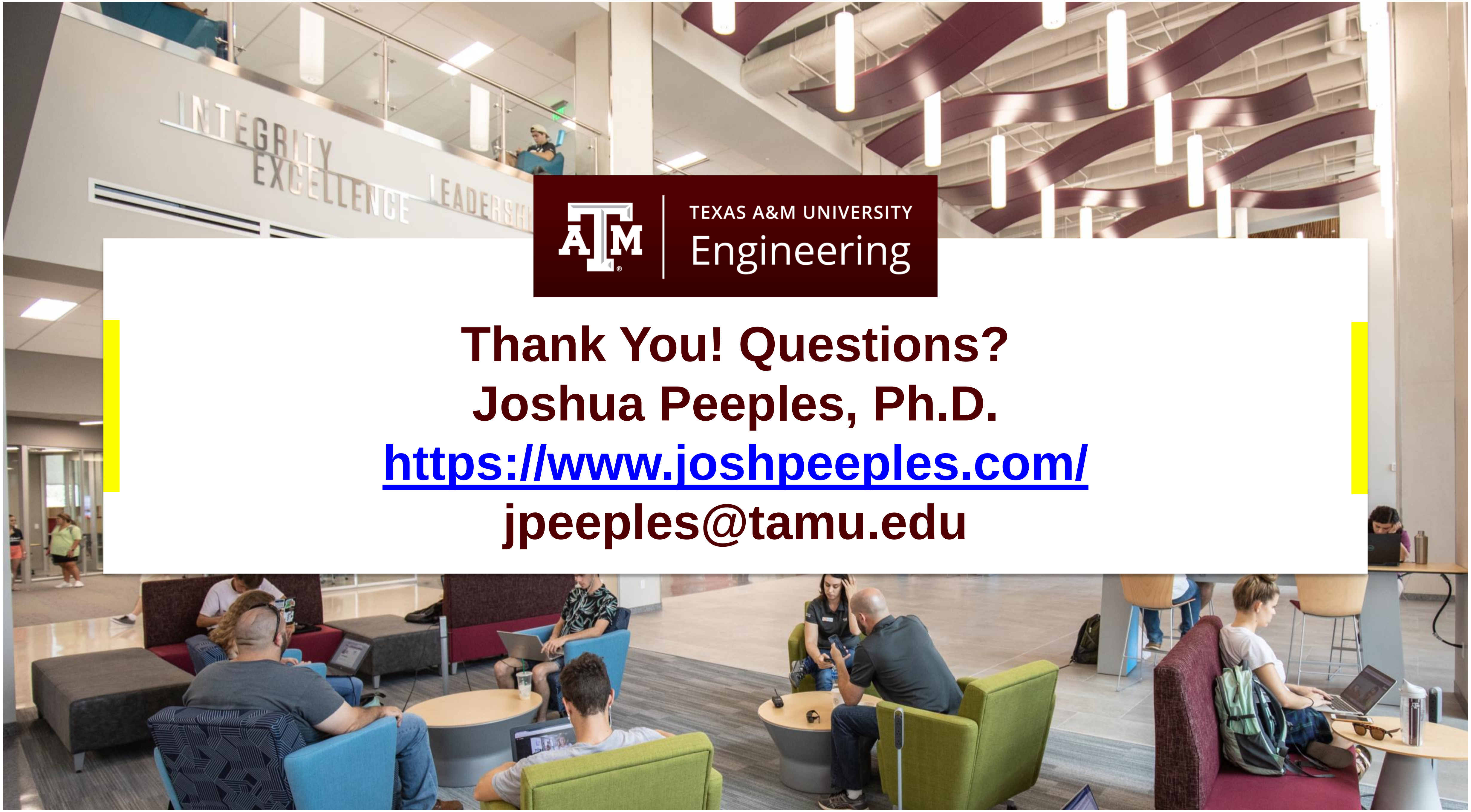


A. Starke, et al. Toward Resilient Smart Grid Communications Using Distributed SDN with ML-Based Anomaly Detection. WWIC 2018.

Next class

TEXAS A&M UNIVERSITY Engineering

- Representative Clustering II



TEXAS A&M UNIVERSITY Engineering

Thank You! Questions?

Joshua Peeples, Ph.D.
https://www.joshpeeples.com/
jpeeples@tamu.edu

Supplemental Slides

Useful Links

- Sklearn k-means
- StatQuest: k-means clustering
- k-mean Google Colab Notebook
- k-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks