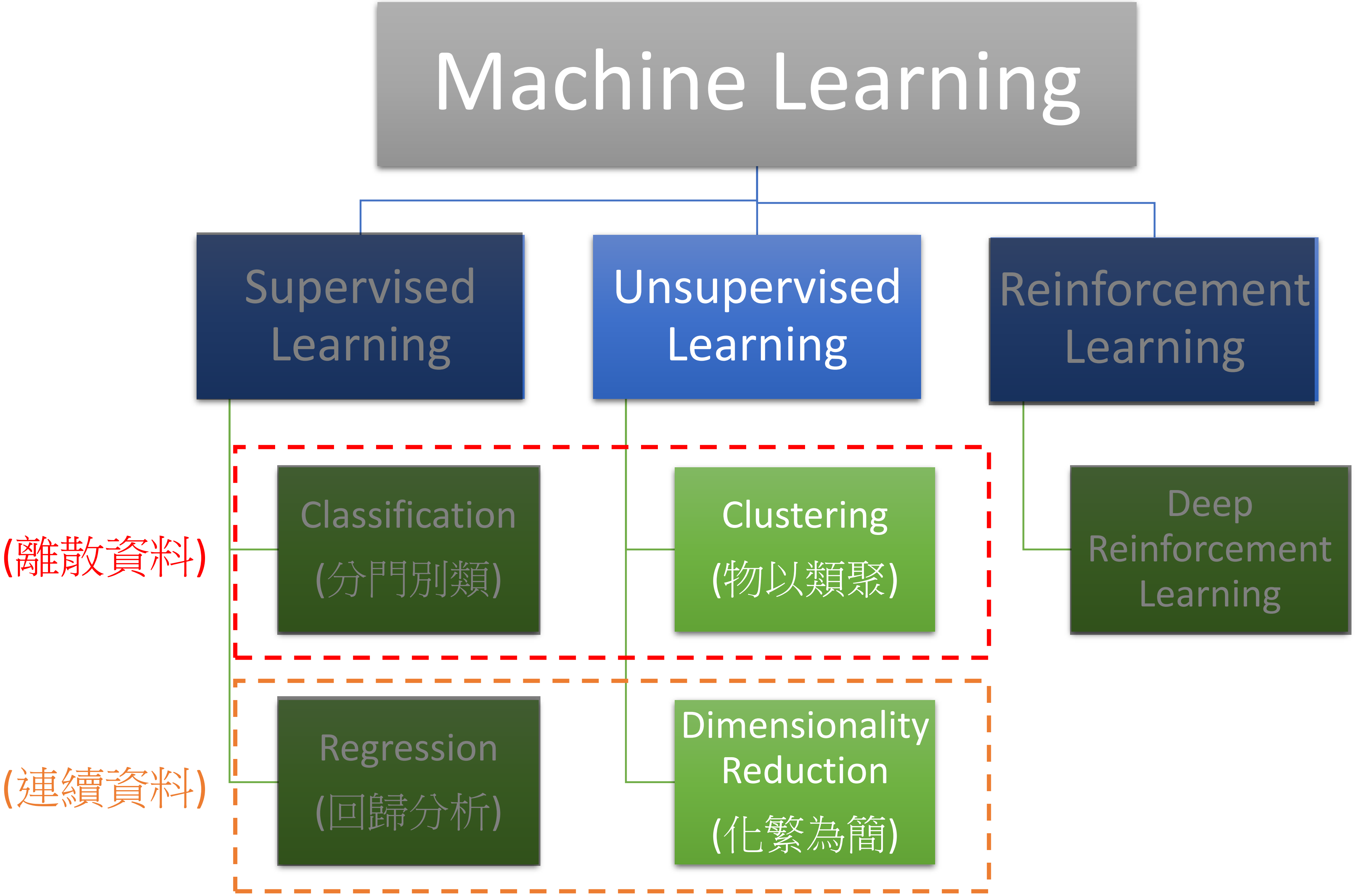
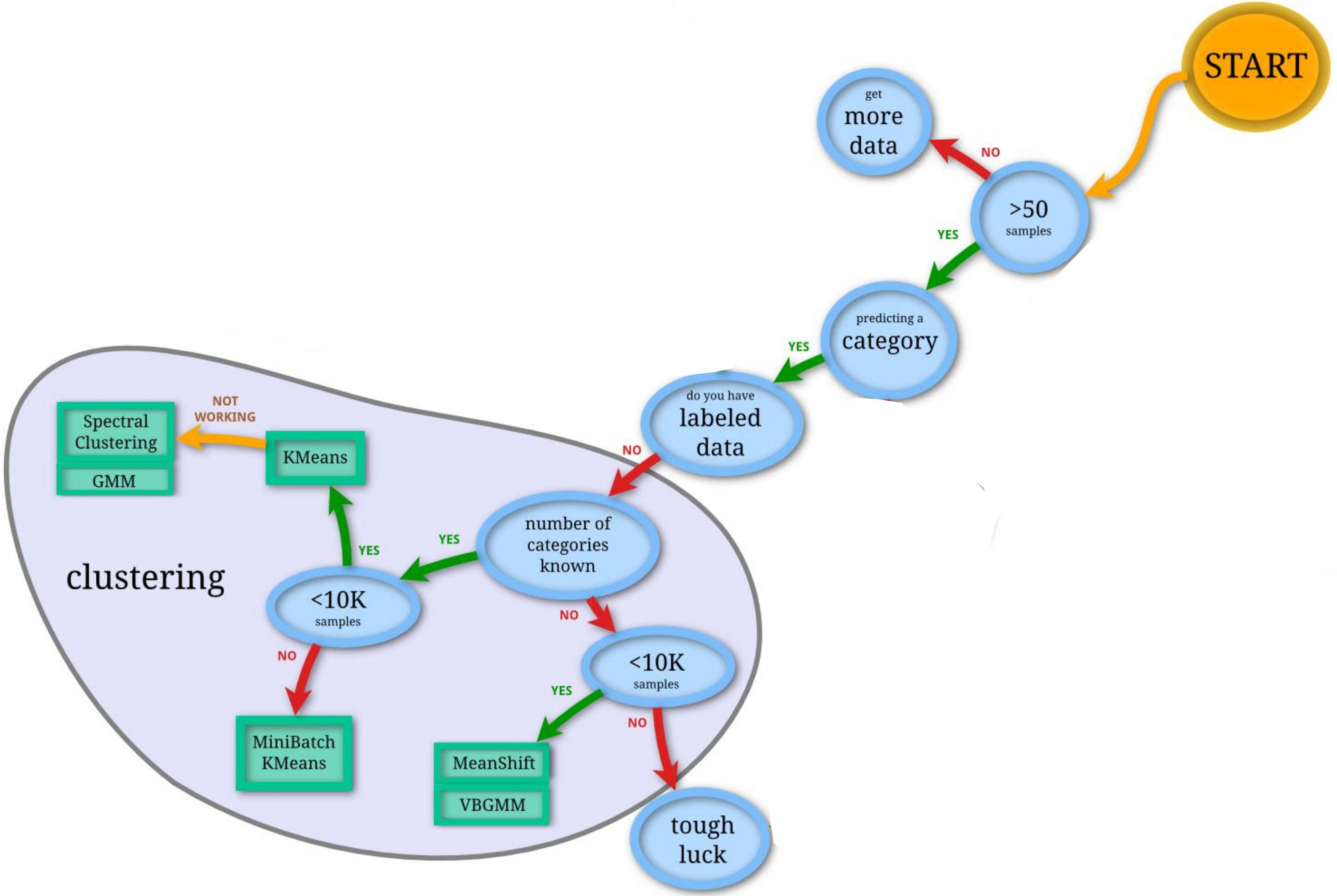
Unsupervised Learning

Prof. Kuan-Ting Lai

2020/4/23


Machine Learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning
(離散資料)
Classification
(分門別類)
Clustering
(物以類聚)
Deep Reinforcement Learning
(連續資料)
Regression
(回歸分析)
Dimensionality Reduction
(化繁為簡)

scikit-learn algorithm cheat-sheet



K-means

- Partition *n* samples into *k* clusters
- Each sample belongs to the cluster with the nearest mean (cluster centroid)

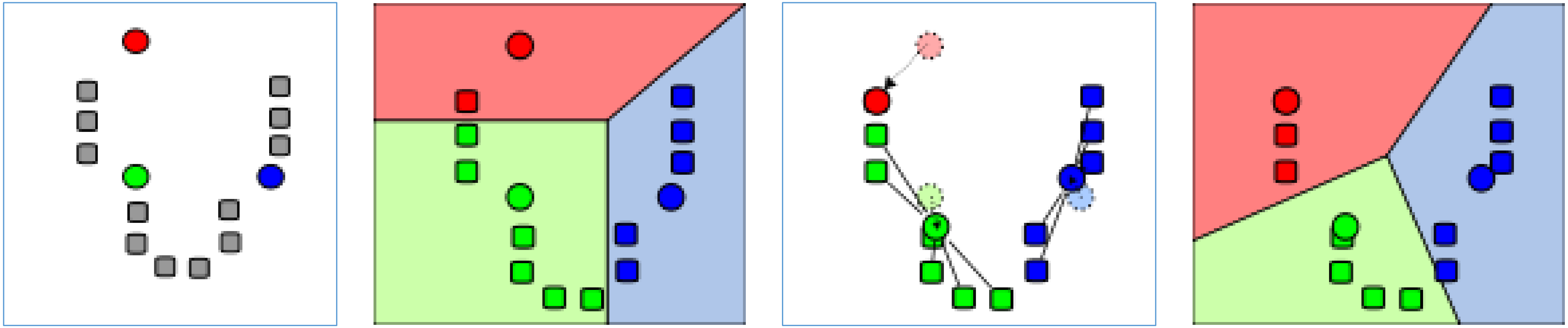
$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

https://en.wikipedia.org/wiki/K-means_clustering

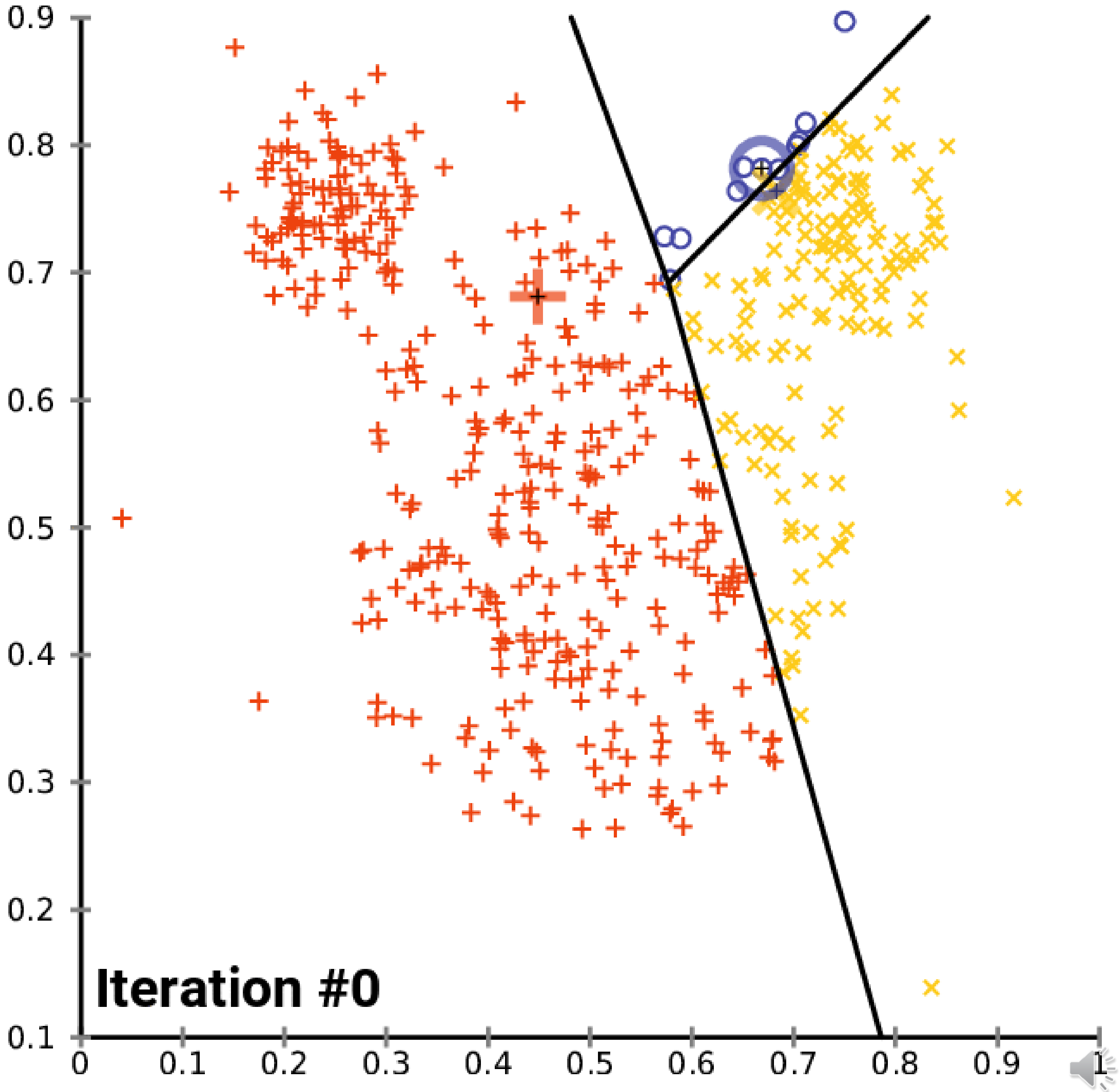
Naïve K-means

1. Assignment step: Assign each sample to a cluster
2. Update step: Recalculate the means

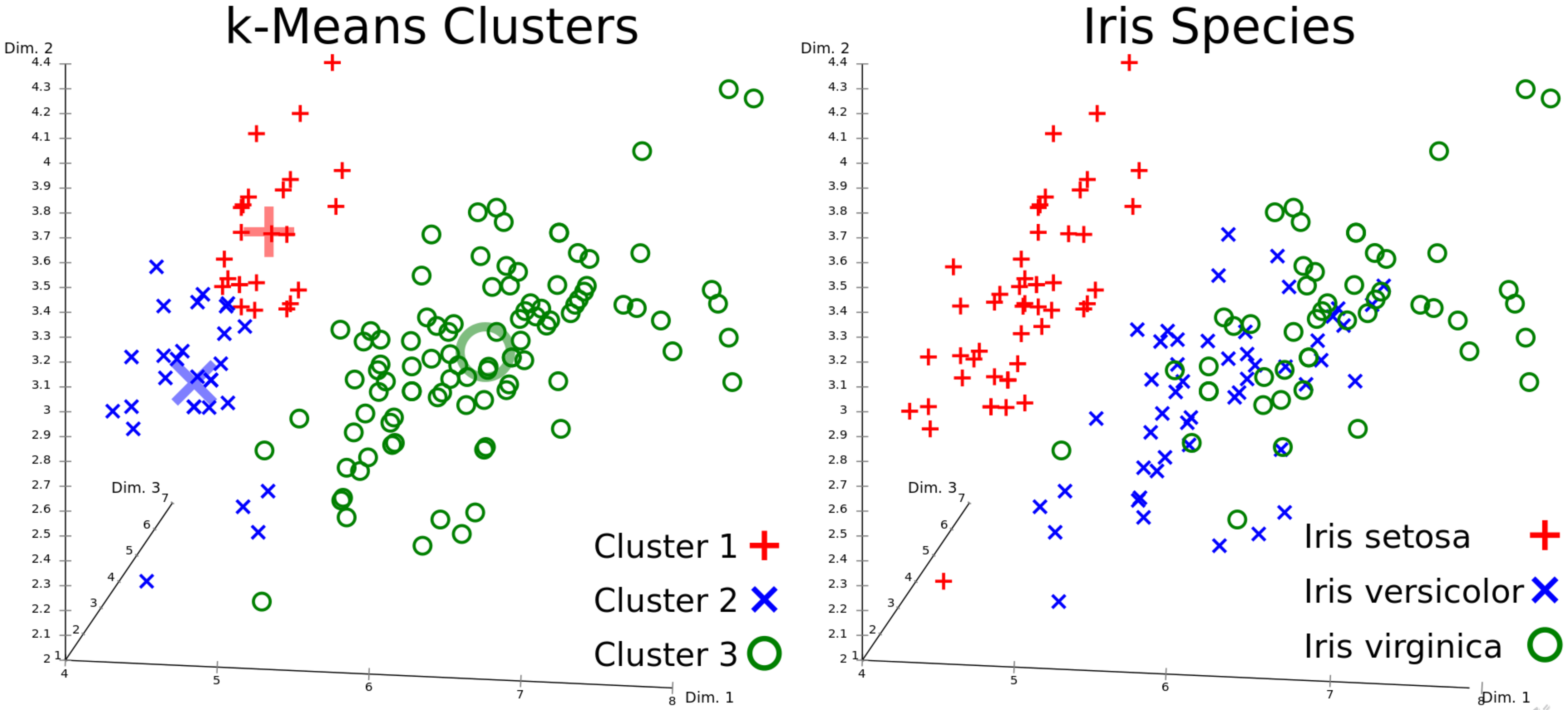
- Iteratively perform the two steps until convergence



K-means Convergence



K-means for Iris Flower Dataset



https://en.wikipedia.org/wiki/File:Iris_Flowers_Clustering_kMeans.svg

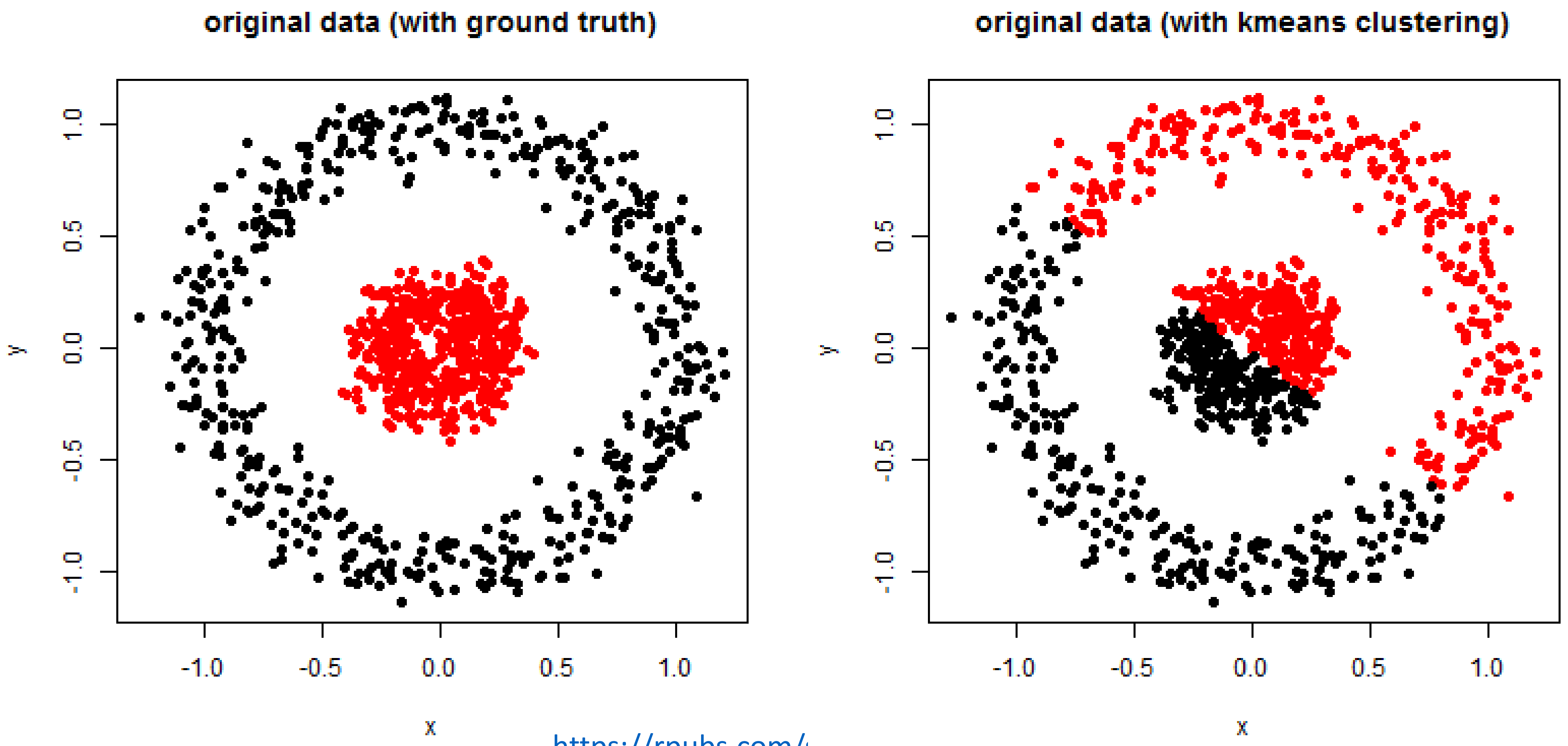
K-means Applications

- Vector Quantization
  - Color quantization, image segmentation,…
- Cluster Analysis
  - Market segmentation
- Feature learning

Problems of k-means

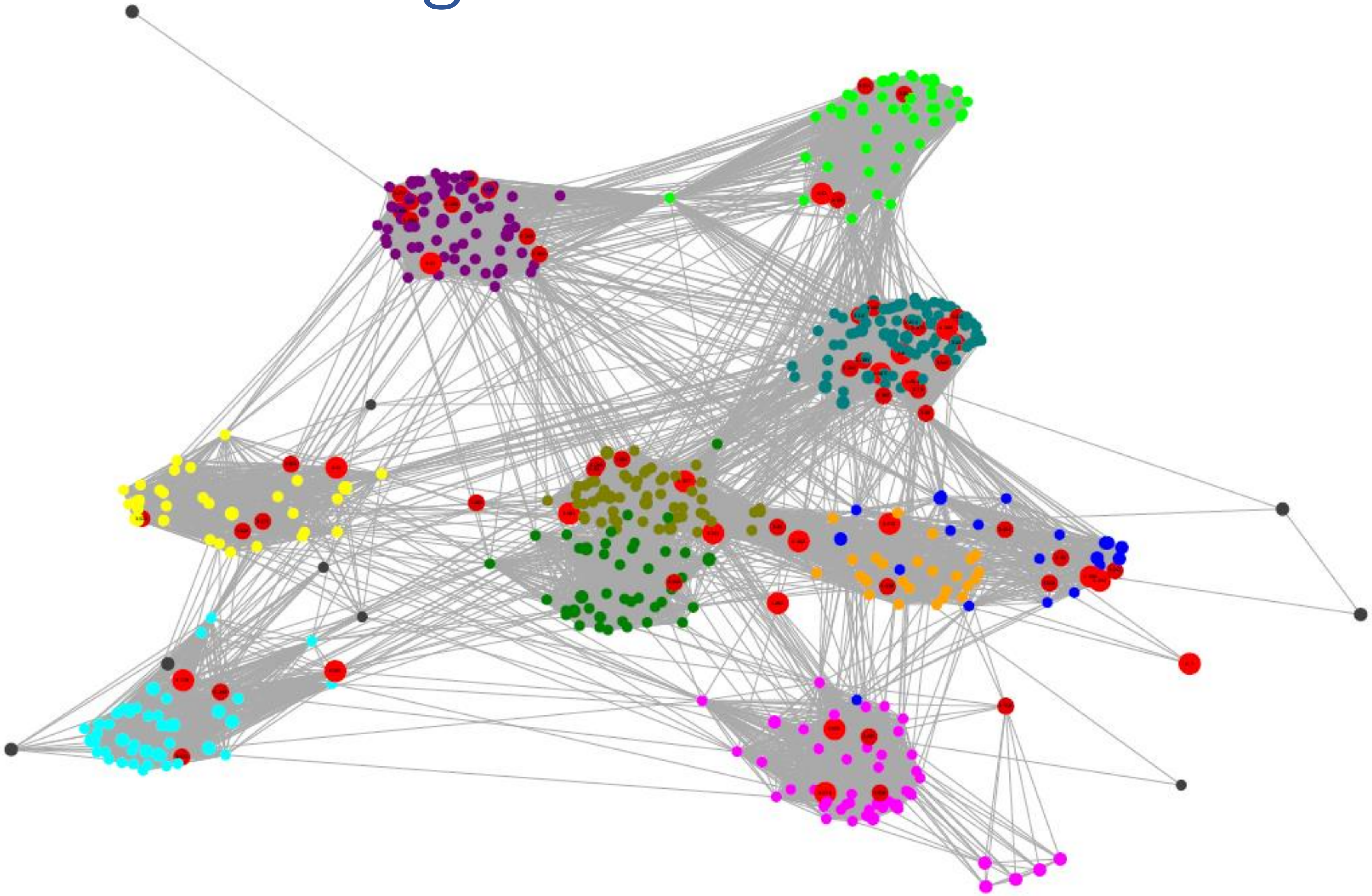
- Sensitive to initialization
- Sensitive to outliers
  - Solution: using k-medians
- Make hard assignments
  - Gaussian Mixture Models (GMM) allows soft assignments
- Works well only for round shaped, equal sizes clusters

K-means vs. Spectral Clustering



https://rpubs.com/

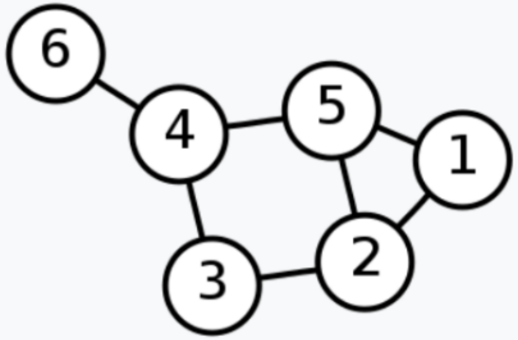
Graph Clustering



Laplacian Matrix

- $L = D - A$, $L_{i,j} := \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases}$

where $\deg(v_i)$ is the degree of the vertex i.

Labelled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$

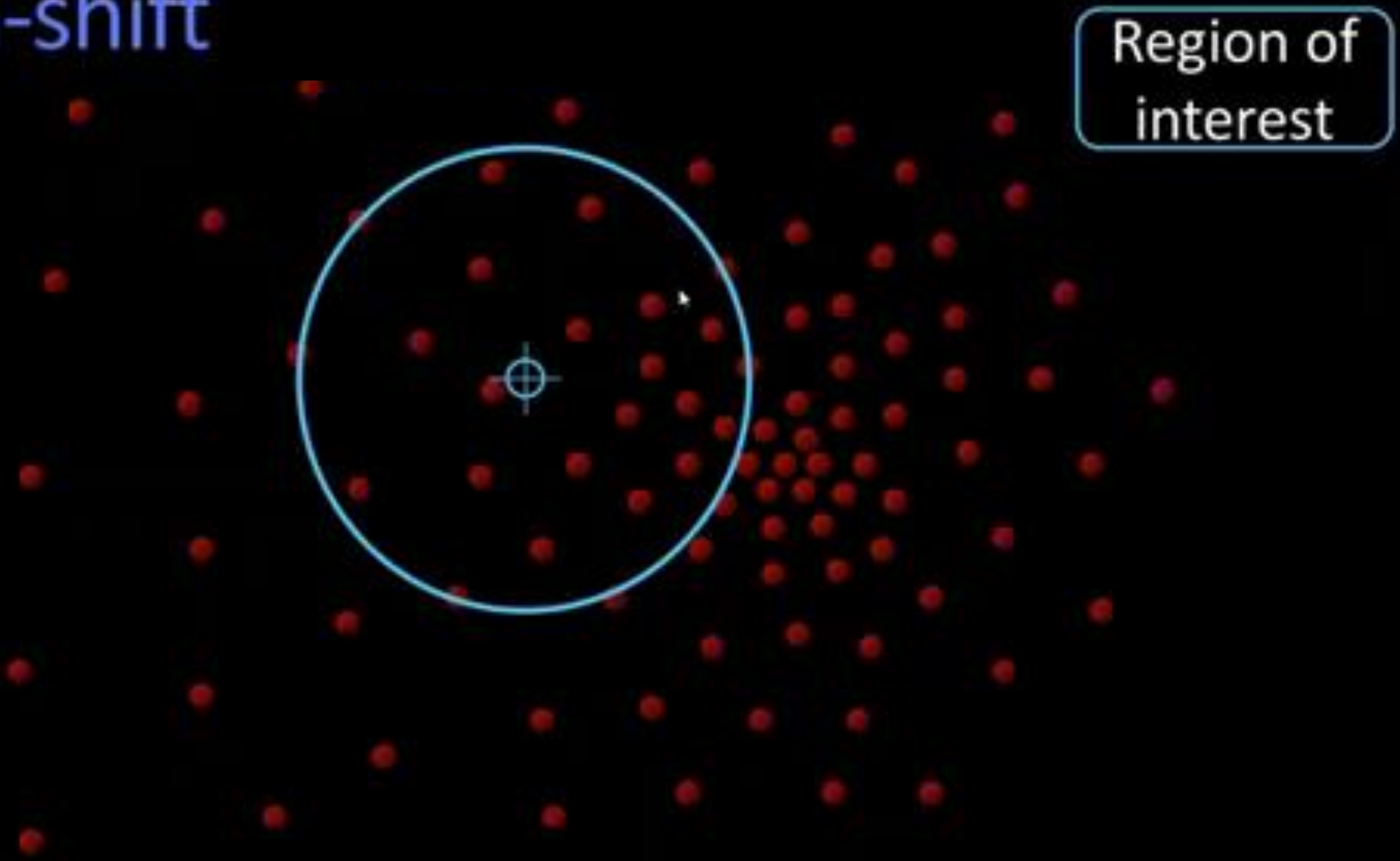
Spectral Clustering

- Utilize the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction

$$L^{\text{sym}} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

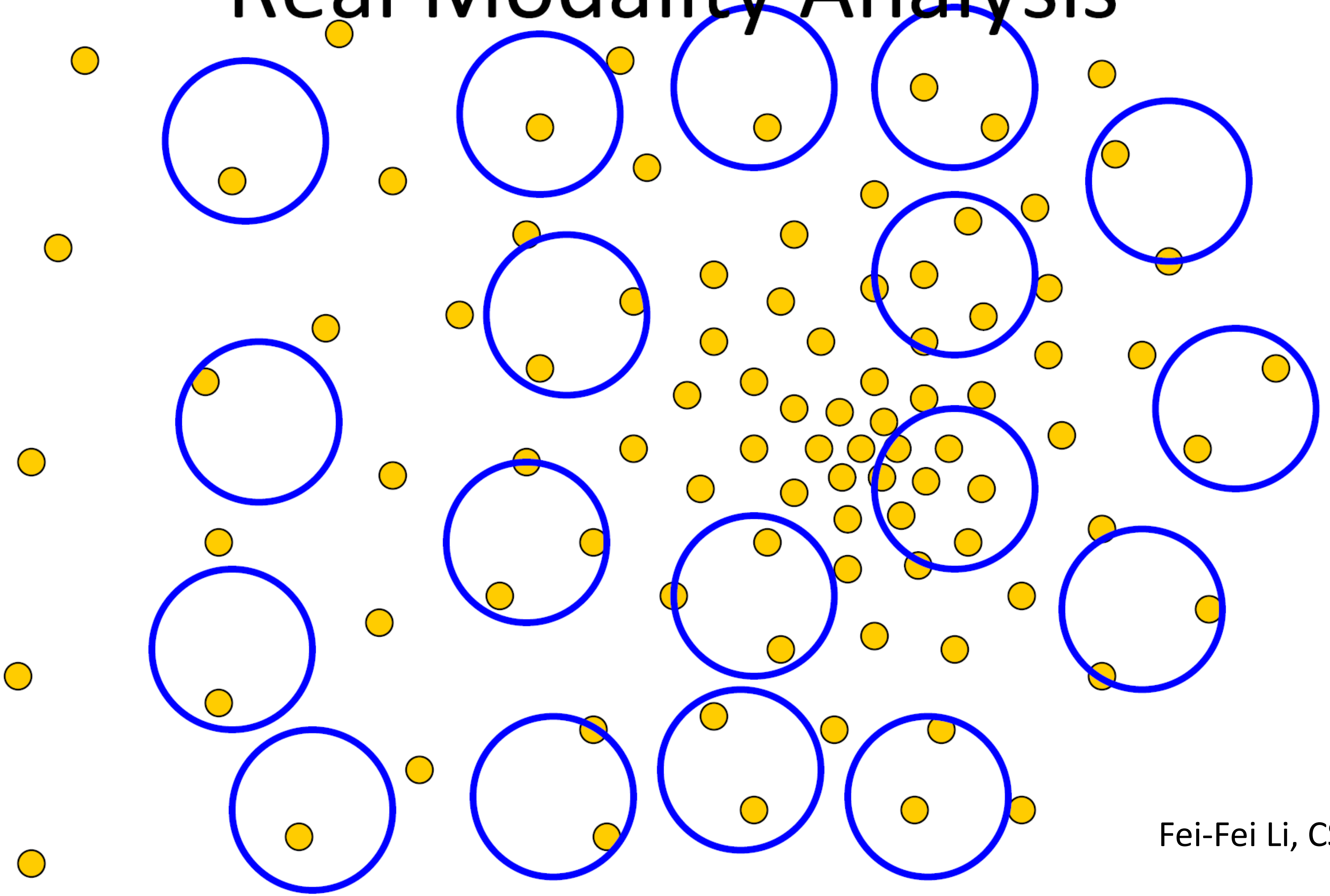
$$L^{\text{sym}}_{i,j} := \begin{cases} 1 & \text{if } i = j \text{ and } \deg(v_i) \neq 0 \\ -\frac{1}{\sqrt{\deg(v_i)\deg(v_j)}} & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise.} \end{cases}$$

Mean-shift



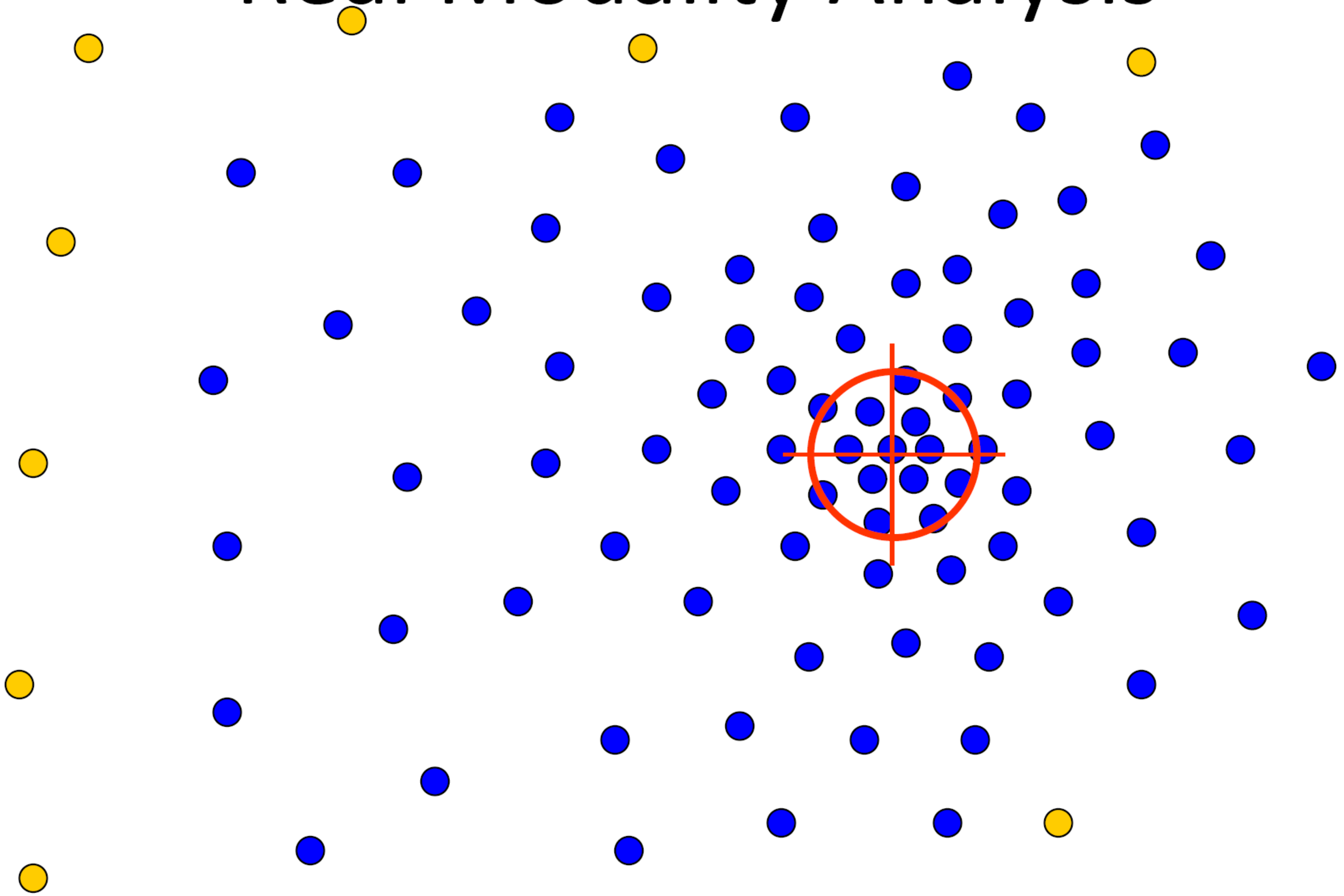
https://www.youtube.com/watch?v=TMPEujQrY70

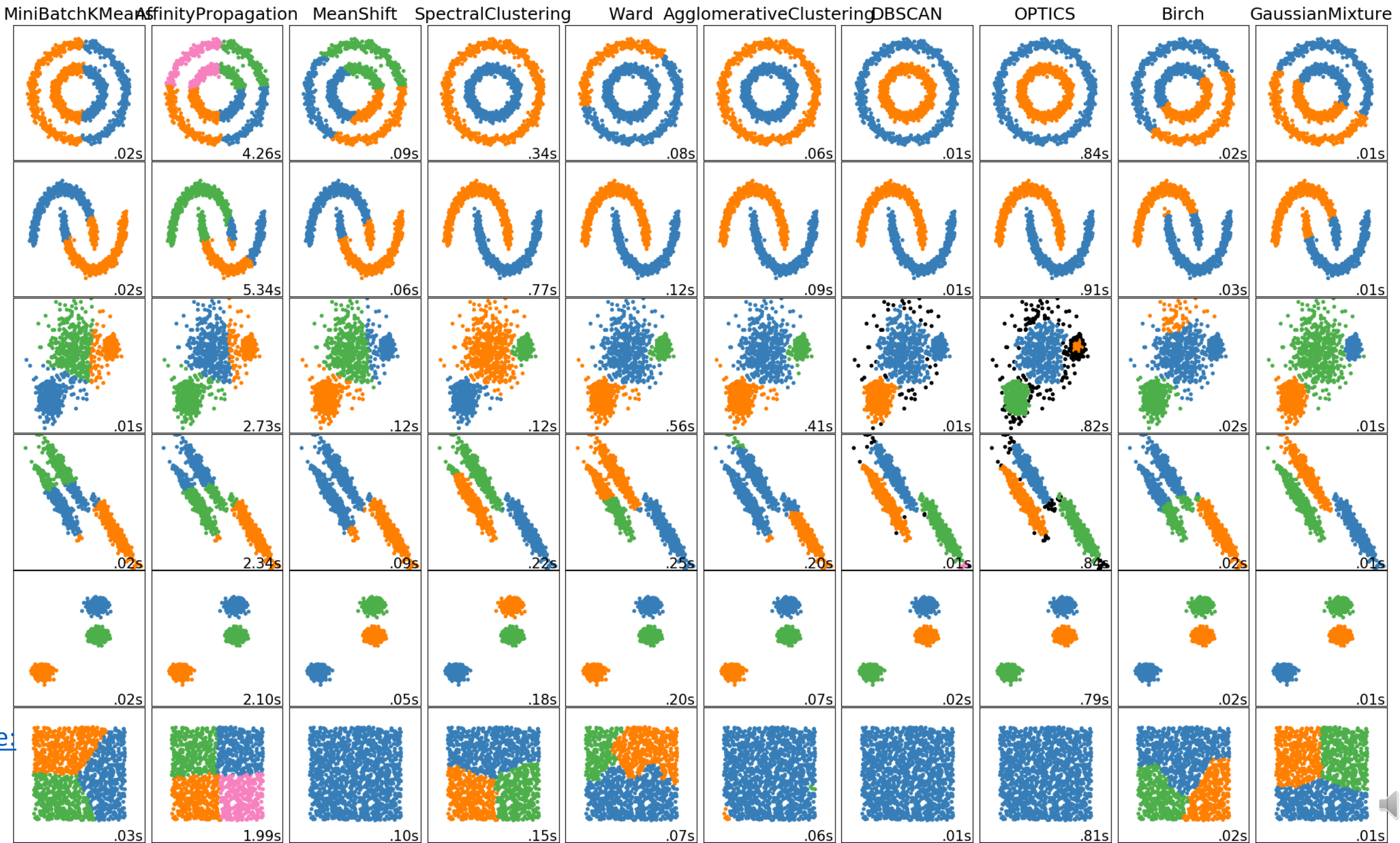
Real Modality Analysis



Fei-Fei Li, CS131 2016

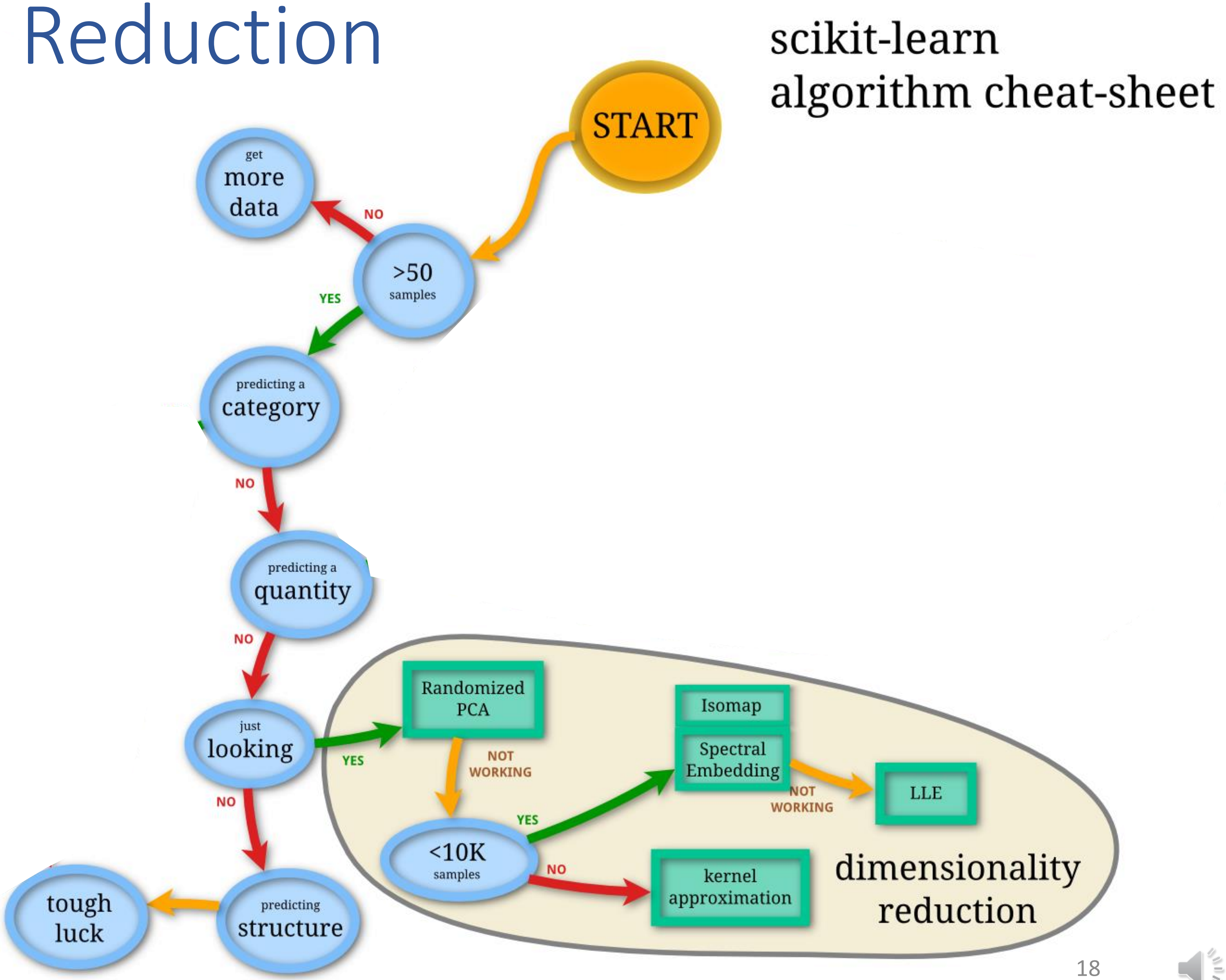
Real Modality Analysis





[source: scikit-learn]

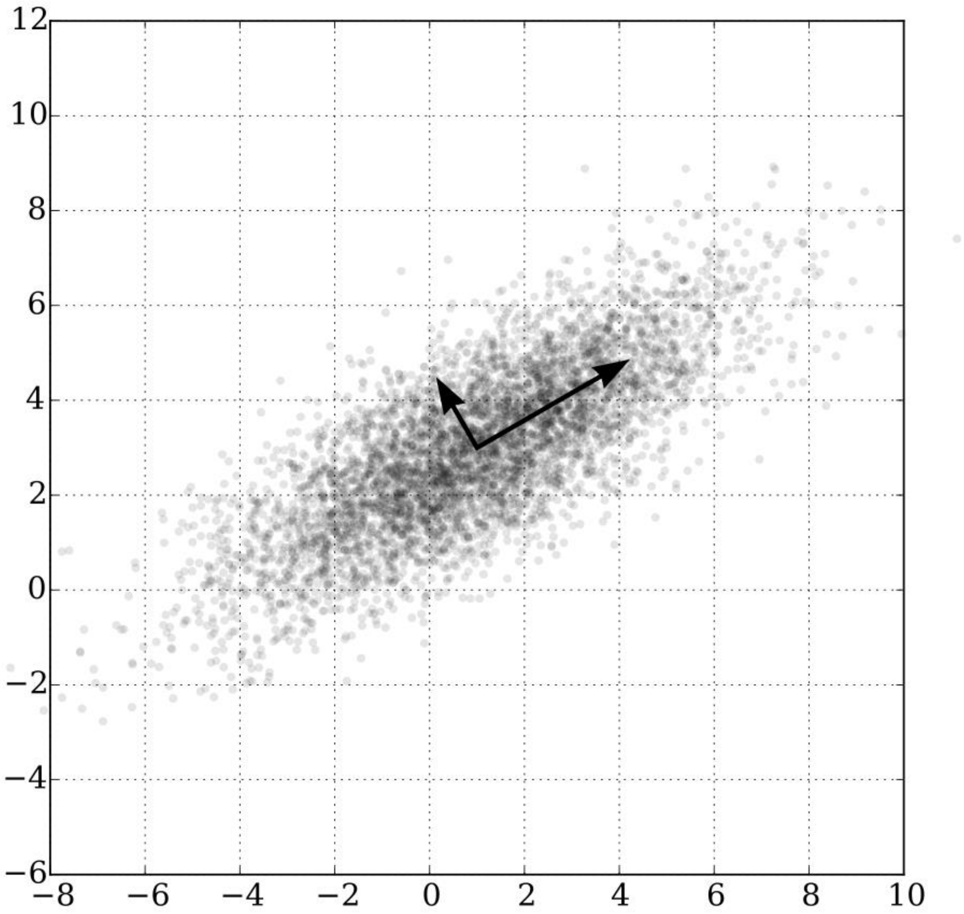
Dimensionality Reduction



Principle Component Analysis (PCA)

- Assumptions
  - *Linearity*
  - *Mean and Variance are sufficient statistics*
  - *The principal components are orthogonal*
- Calculate PCA
  - $Y = W \cdot X$

$$\text{max.} \quad \operatorname{cov}(\mathbf{Y}, \mathbf{Y})$$
$$s.b.t \quad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}$$

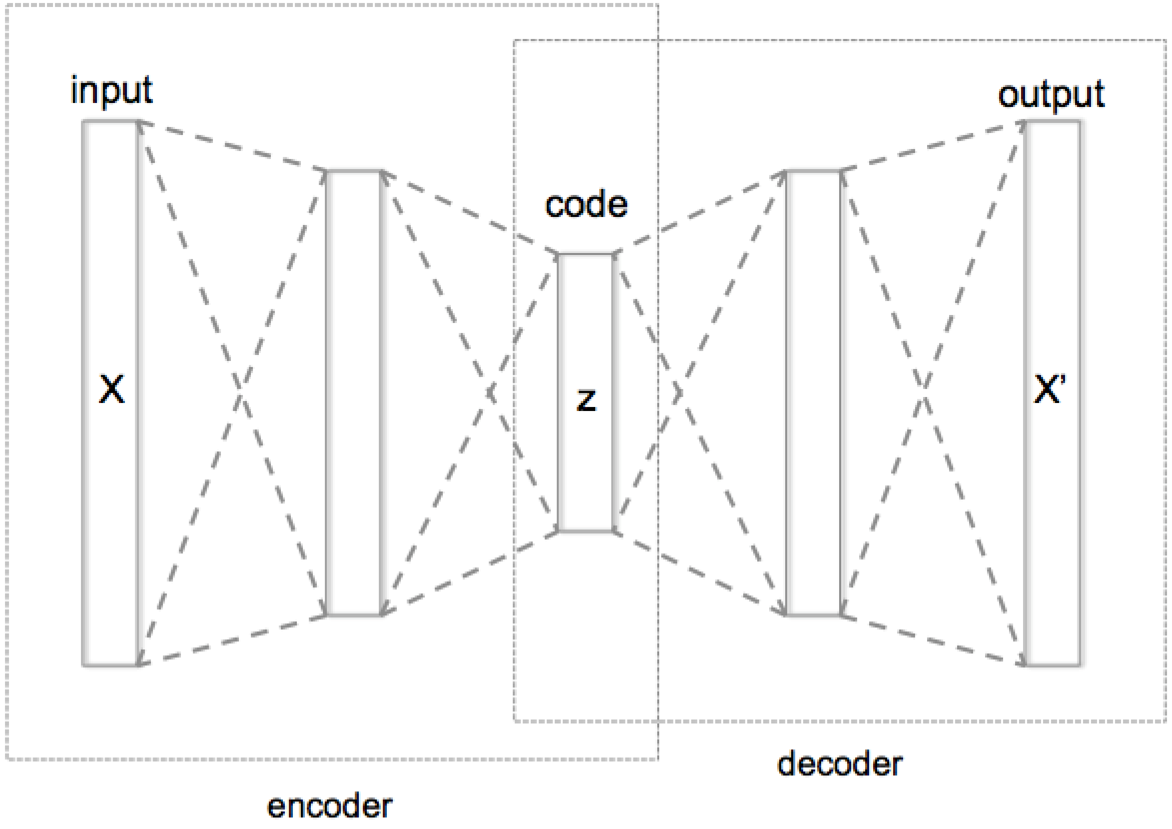


https://en.wikipedia.org/wiki/Principal_component_analysis

Eigenfaces

DL Unsupervised Method: Autoencoder

- An *autoencoder* is a neural network that learns to copy its input to its output.
- It learns a *hidden* layer to describes a *code* used to represent the input



https://en.wikipedia.org/wiki/Autoencoder

References

- https://en.wikipedia.org/wiki/K-means_clustering
- https://en.wikipedia.org/wiki/Principal_component_analysis
- https://www.geeksforgeeks.org/ml-mean-shift-clustering/
- https://en.wikipedia.org/wiki/Autoencoder