

# CS247: ADVANCED DATA MINING

## 3: K-Means and Mixture Model

---

**Instructor: Yizhou Sun**

[yzsun@cs.ucla.edu](mailto:yzsun@cs.ucla.edu)


April 8, 2019

# Methods to Learn

	Vector Data	Text Data	Graph & Network	Recommender Systems
Classification	Naïve Bayes; Logistic Regression; NN		Label Propagation	
Clustering	K-means; kernel k-means; Mixture Models	PLSA; LDA	Spectral Clustering	Matrix Factorization
Prediction	NN			Collaborative Filtering; Factorization machine; Hybrid CF; Recommendation with graph regularization
Ranking			PageRank	
Similarity Search			P-PageRank	
Representation Learning		Word embedding	Network embedding	Deep collaborative learning

# Clustering

---

- Clustering 
- K-means
- Kernel K-means
- Mixture Model and EM algorithm
- Summary

# What is Cluster Analysis?

---

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation*, ...)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- **Unsupervised learning**: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a **stand-alone tool** to get insight into data distribution
  - As a **preprocessing step** for other algorithms


# Applications of Cluster Analysis

---

- Data reduction
  - Summarization: Preprocessing for regression, PCA, classification, and association analysis
  - Compression: Image processing: vector quantization
- Prediction based on groups
  - Cluster & find characteristics/patterns for each group
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection: Outliers are often viewed as those “far away” from any cluster

# Clustering

---

- Clustering
- K-means 
- Kernel K-means
- Mixture Model and EM algorithm
- Summary

# Recall K-Means

---

- Objective function

- $J = \sum_{j=1}^k \sum_{C(i)=j} \|x_i - c_j\|^2$

- Total within-cluster variance

- Re-arrange the objective function

- $J = \sum_{j=1}^k \sum_i w_{ij} \|x_i - c_j\|^2$

- $w_{ij} \in \{0,1\}$

- $w_{ij} = 1$ , if  $x_i$  belongs to cluster  $j$ ;  $w_{ij} = 0$ , otherwise

- Looking for:

- The best assignment  $w_{ij}$

- The best center  $c_j$

# Solution of K-Means

---

- Iterations

$$J = \sum_{j=1}^k \sum_i w_{ij} \|x_i - c_j\|^2$$

- Step 1: Fix centers  $c_j$ , find assignment  $w_{ij}$  that minimizes  $J$

- $\Rightarrow w_{ij} = 1$ , if  $\|x_i - c_j\|^2$  is the smallest

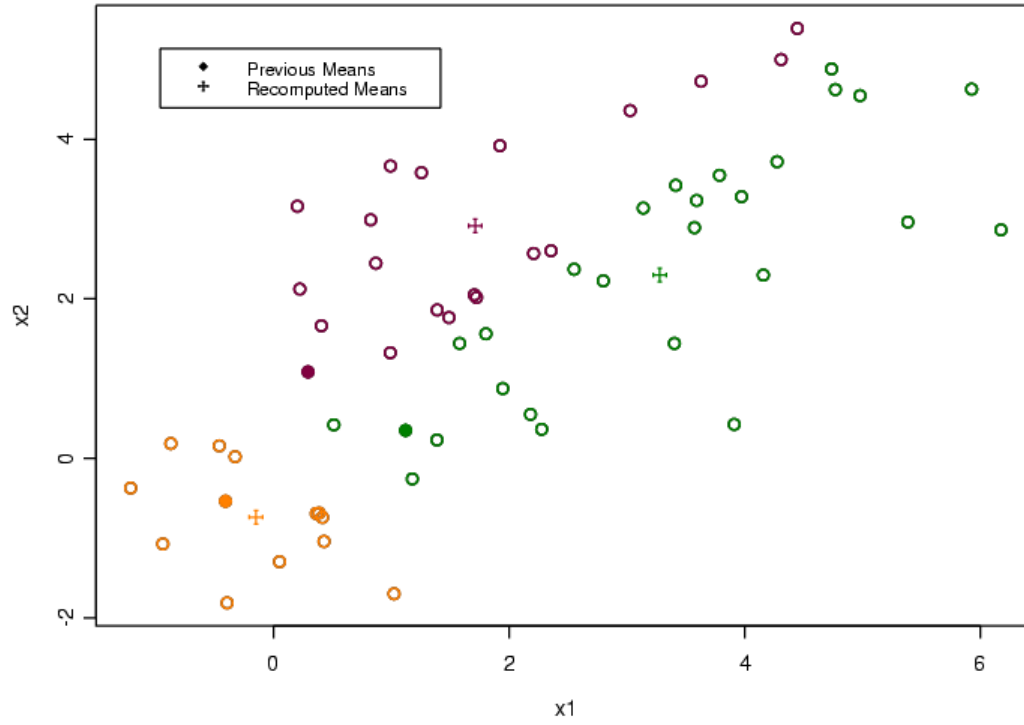
- Step 2: Fix assignment  $w_{ij}$ , find centers that minimize  $J$

- $\Rightarrow$  first derivative of  $J = 0$

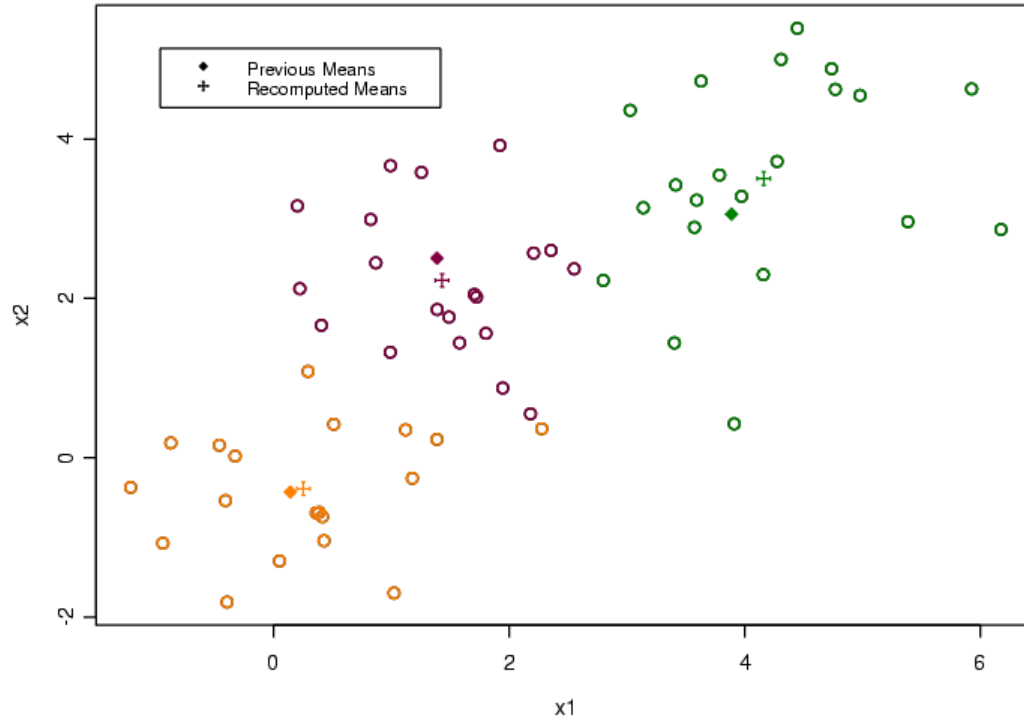
- $\Rightarrow \frac{\partial J}{\partial c_j} = -2 \sum_i w_{ij} (x_i - c_j) = 0$

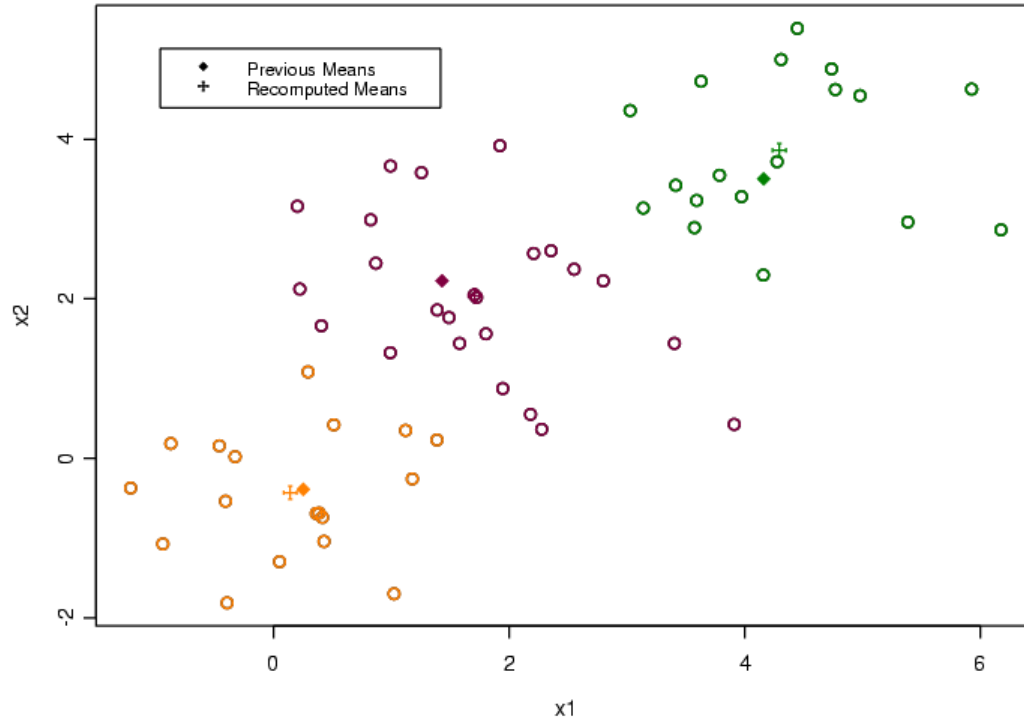
- $\Rightarrow c_j = \frac{\sum_i w_{ij} x_i}{\sum_i w_{ij}}$

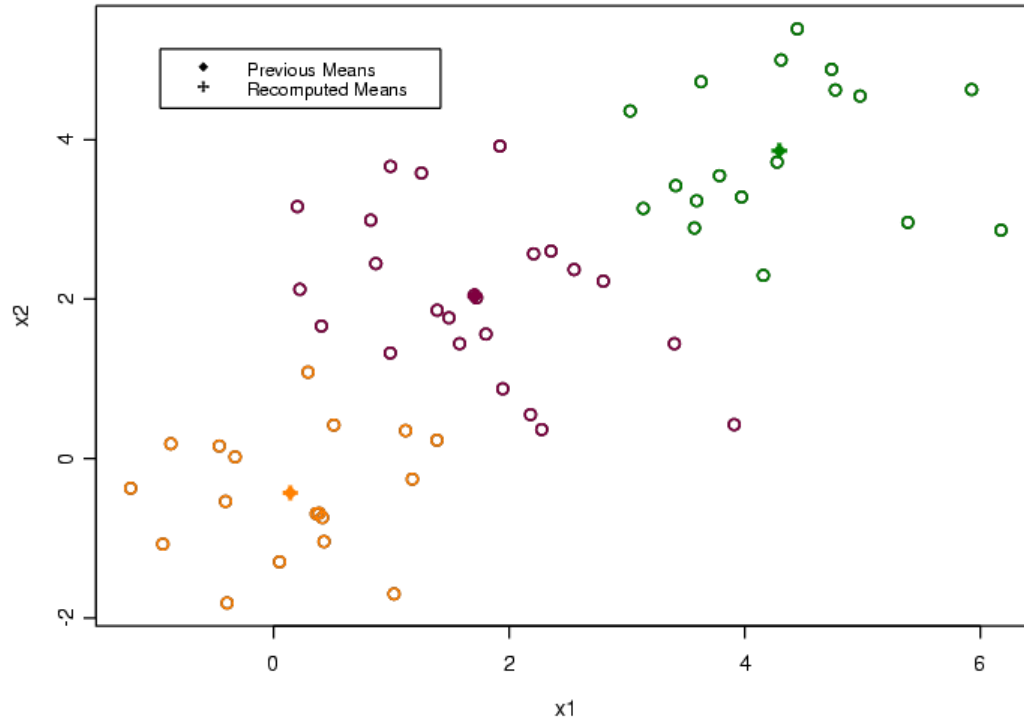
- Note  $\sum_i w_{ij}$  is the total number of objects in cluster  $j$

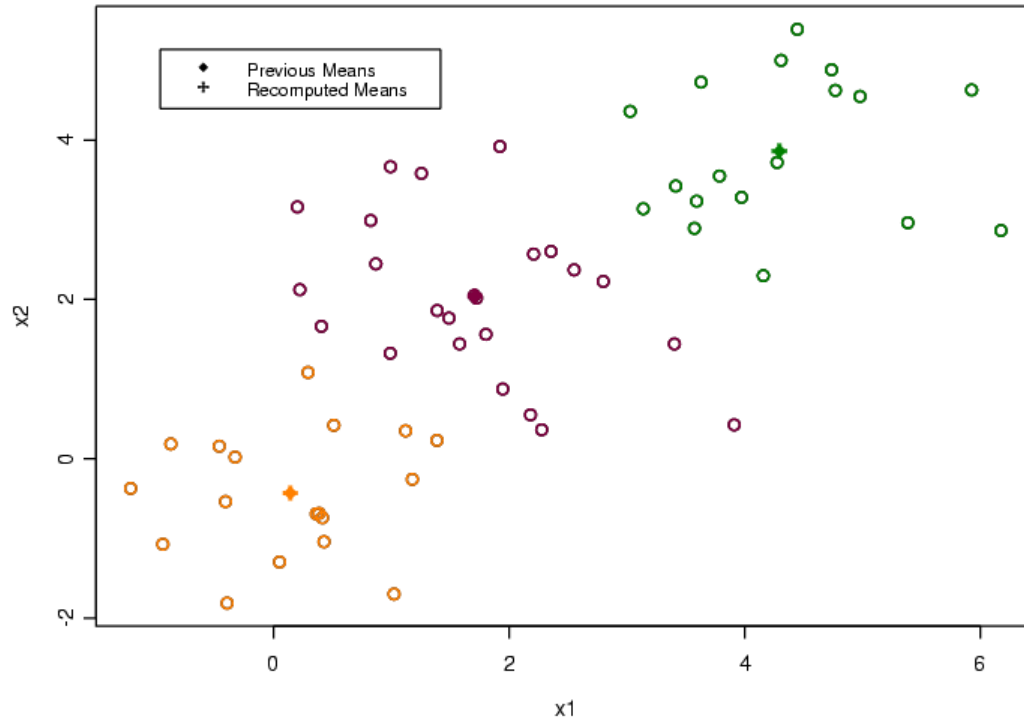












**Converges! Why?**

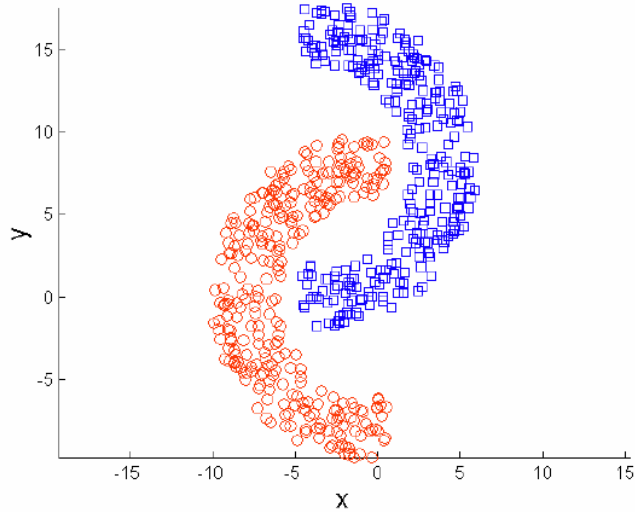
# Limitations of K-Means

---

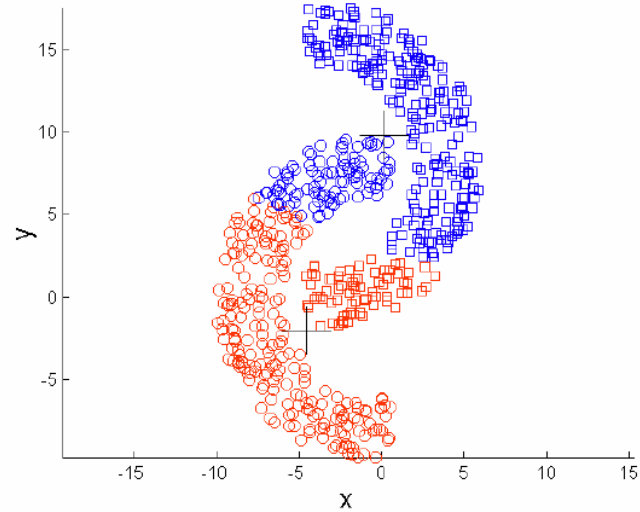
- K-means has problems when clusters are of
  - Non-Spherical Shapes
  - Different Sizes and density

# Limitations of K-Means: Non-Spherical Shapes

---

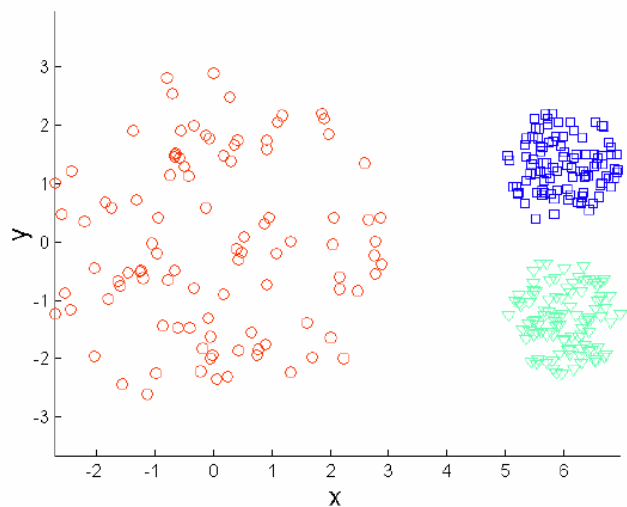


Original Points

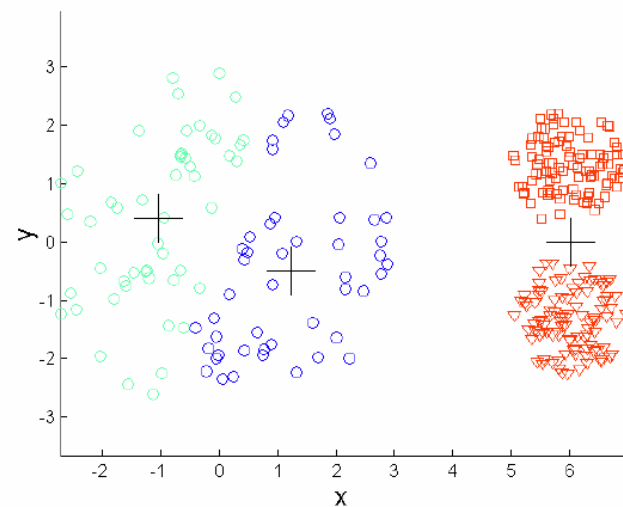


K-means (2 Clusters)

# Limitations of K-Means: Different Sizes and Variances



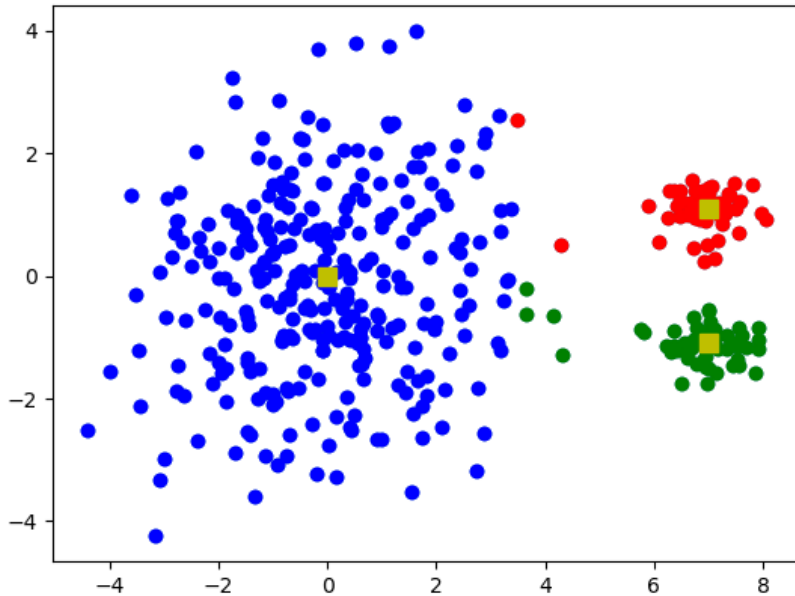
Original Points



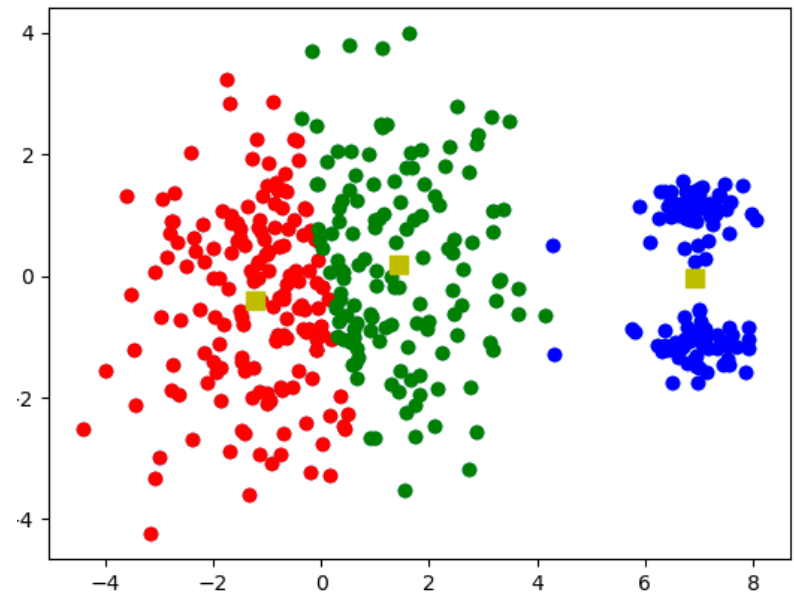
K-means (3 Clusters)

# Example

- Consider the cost of K-means in two cases



Cost:  $J = 1560.86$

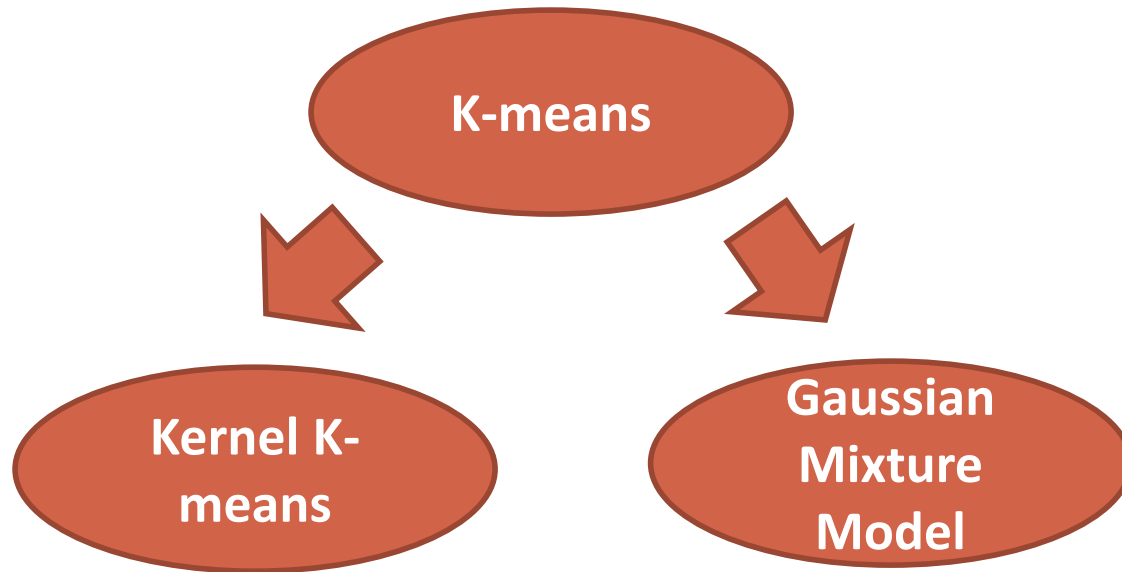


Cost:  $J = 1147.42$

$$\text{Recall: } J = \sum_{j=1}^k \sum_{C(i)=j} \|x_i - c_j\|^2$$


# Connections of K-means to Other Methods

---



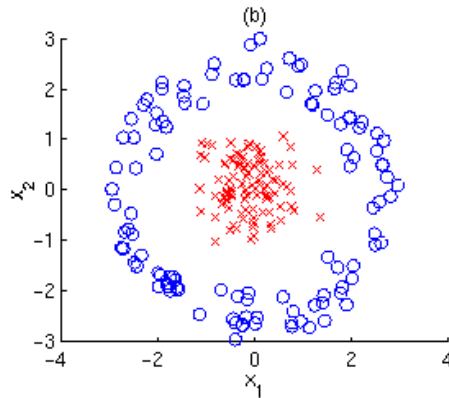
# Clustering

---

- Clustering
- K-means
- Kernel K-means 
- Mixture Model and EM algorithm
- Summary

# Kernel K-Means

- How to cluster the following data?



- A non-linear map:  $\phi: R^n \rightarrow F$ 
  - Map a data point into a higher/infinite dimensional space
  - $x \rightarrow \phi(x)$
- Dot product matrix  $K_{ij}$ 
  - $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$

# Typical Kernel Functions

---

- Recall kernel SVM:

Polynomial kernel of degree  $h$  :  $K(\mathbf{X}_i, \mathbf{X}_j) = (\mathbf{X}_i \cdot \mathbf{X}_j + 1)^h$

Gaussian radial basis function kernel :  $K(\mathbf{X}_i, \mathbf{X}_j) = e^{-\|\mathbf{X}_i - \mathbf{X}_j\|^2 / 2\sigma^2}$

Sigmoid kernel :  $K(\mathbf{X}_i, \mathbf{X}_j) = \tanh(\kappa \mathbf{X}_i \cdot \mathbf{X}_j - \delta)$

# Solution of Kernel K-Means

- Objective function under new feature space:
  - $J = \sum_{j=1}^k \sum_i w_{ij} \|\phi(x_i) - c_j\|^2$
- Algorithm
  - By fixing assignment  $w_{ij}$ 
    - $c_j = \sum_i w_{ij} \phi(x_i) / \sum_i w_{ij}$
  - In the assignment step, assign the data points to the closest center

$$\bullet d(x_i, c_j) = \left\| \phi(x_i) - \frac{\sum_{i'} w_{i'j} \phi(x_{i'})}{\sum_{i'} w_{i'j}} \right\|^2 = \phi(x_i) \cdot \phi(x_i) - \frac{2 \sum_{i'} w_{i'j} \phi(x_i) \cdot \phi(x_{i'})}{\sum_{i'} w_{i'j}} + \frac{\sum_{i'} \sum_l w_{i'j} w_{lj} \phi(x_{i'}) \cdot \phi(x_l)}{(\sum_{i'} w_{i'j})^2}$$

Do not really need to know  $\phi(x)$ , but only  $K_{ij}$

# Advantages and Disadvantages of Kernel K-Means

---

- **Advantages**

- Algorithm is able to identify the non-linear structures.

- **Disadvantages**


- Number of cluster centers need to be predefined.
- Algorithm is complex in nature and time complexity is large.

- **References**

- Kernel k-means and Spectral Clustering by Max Welling.
- Kernel k-means, Spectral Clustering and Normalized Cut by Inderjit S. Dhillon, Yuqiang Guan and Brian Kulis.
- An Introduction to kernel methods by Colin Campbell.

# Clustering

---

- Clustering
- K-means
- Kernel K-means
- Mixture Model and EM algorithm 
- Summary

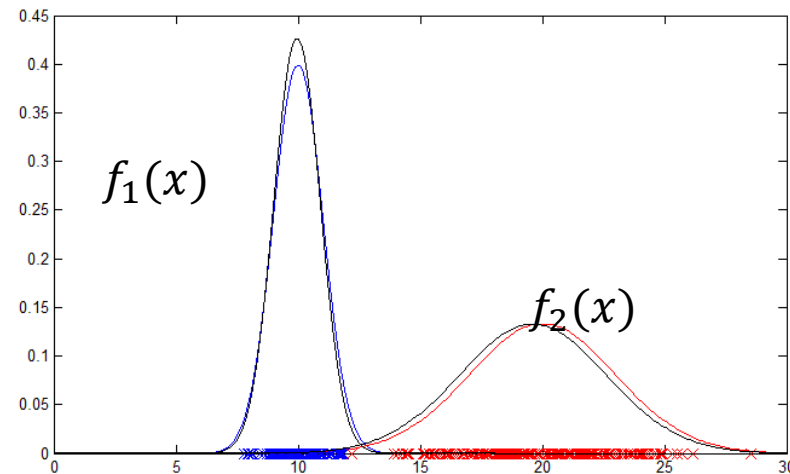
# Hard Clustering vs. Soft Clustering

---

- Hard Clustering
  - Every object  $i$  is assigned to one cluster  $j$ , e.g., k-means
    - $w_{ij} = \{0,1\}$  and  $\sum_j w_{ij} = 1$
- Soft Clustering
  - Every object  $i$  is assigned with a probability to different clusters
    - $w_{ij} \in [0,1]$  and  $\sum_j w_{ij} = 1$

# Mixture Model-Based Clustering

- A set  $C$  of  $k$  probabilistic clusters  $C_1, \dots, C_k$ 
  - probability density functions:  $f_1, \dots, f_k$ ,
  - Cluster prior probabilities:  $w_1, \dots, w_k, \sum_j w_j = 1$
- Joint Probability of an object  $i$  and its cluster  $C_j$  is:
  - $p(x_i, z_i = C_j) = w_j f_j(x_i)$
  - $z_i$ : hidden random variable
- Probability of  $i$  is:
  - $p(x_i) = \sum_j w_j f_j(x_i)$



# Maximum Likelihood Estimation

---


- Since objects are assumed to be generated independently, for a data set  $D = \{x_1, \dots, x_n\}$ , we have,

$$p(D) = \prod_i p(x_i) = \prod_i \sum_j w_j f_j(x_i)$$

$$\Rightarrow \log p(D) = \sum_i \log p(x_i) = \sum_i \log \sum_j w_j f_j(x_i)$$

- Task: Find  $k$  probabilistic clusters s.t.  $p(D)$  is maximized

# The EM (Expectation Maximization) Algorithm

- **The (EM) algorithm:** A framework to approach maximum likelihood or maximum a posteriori estimates of parameters in statistical models.
- **E-step** assigns objects to clusters according to the current soft clustering or parameters of probabilistic clusters
  - $w_{ij}^{t+1} = p(z_i = j | \theta_j^t, x_i) \propto p(x_i | z_i = j, \theta_j^t) p(z_i = j | \theta_j^t)$ 
- **M-step** finds the new clustering or parameters that maximize the expected likelihood, with respect to conditional distribution  $p(z_i = j | \theta_j^t, x_i)$ 
  - $\theta^{t+1} = \operatorname{argmax}_{\theta} \sum_i \sum_j w_{ij}^{t+1} \log p(x_i, z_i = j | \theta)$

# Gaussian Mixture Model

---

- Generative model
  - For each object:
    - Pick its cluster, i.e., a distribution component:  
 $Z \sim \text{Categorical}(w_1, \dots, w_k)$
    - Sample a value from the selected distribution:  
 $X|Z \sim N(\mu_Z, \sigma_Z^2)$
- Overall likelihood function
  - $L(D | \theta) = \prod_i \sum_j w_j p(x_i | \mu_j, \sigma_j^2)$   
s.t.  $\sum_j w_j = 1$  and  $w_j \geq 0$
  - Q: What is  $\theta$  here?

# Apply EM algorithm: 1-d

- An iterative algorithm (at iteration  $t+1$ )

- **E(expectation)-step**

- Evaluate the weight  $w_{ij}$  when  $\mu_j, \sigma_j, w_j$  are given

- $$w_{ij}^{t+1} = \frac{w_j^t p(x_i | \mu_j^t, (\sigma_j^2)^t)}{\sum_k w_k^t p(x_i | \mu_k^t, (\sigma_k^2)^t)}$$

$$f_j^t(x_i)$$

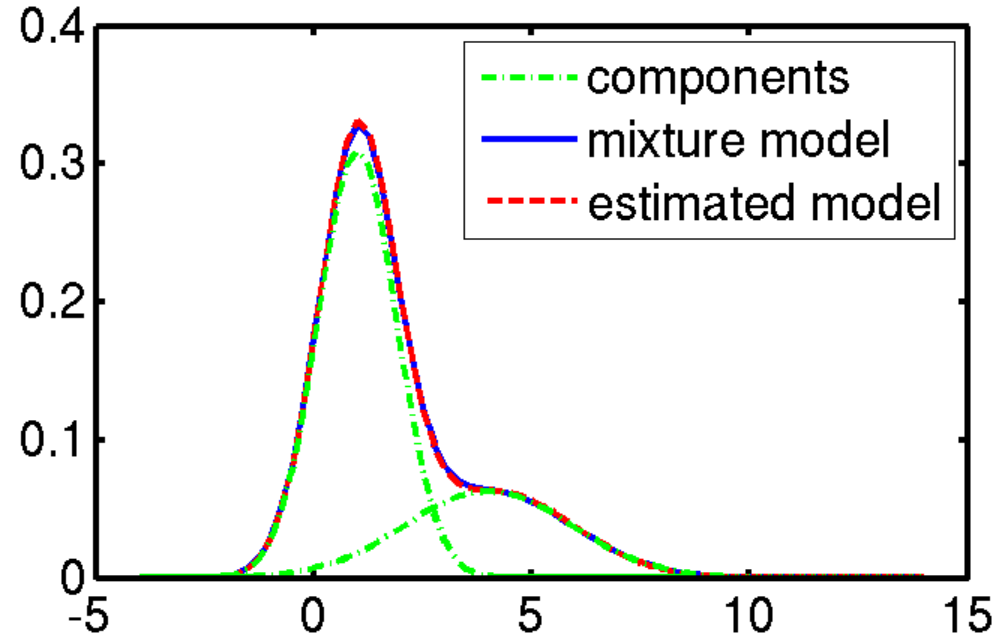
- **M(maximization)-step**

- Find  $\mu_j, \sigma_j, w_j$  that maximize the weighted log likelihood, where  $w_{ij}$ 's are the weights:  $\sum_{ij} w_{ij}^{t+1} \log w_j p(x_i | \mu_j, \sigma_j^2)$
- It is equivalent to Gaussian distribution parameter estimation when each point has a weight belonging to each distribution

- $$\mu_j^{t+1} = \frac{\sum_i w_{ij}^{t+1} x_i}{\sum_i w_{ij}^{t+1}}; (\sigma_j^2)^{t+1} = \frac{\sum_i w_{ij}^{t+1} (x_i - \mu_j^{t+1})^2}{\sum_i w_{ij}^{t+1}}; w_j^{t+1} = \sum_i w_{ij}^{t+1} / n$$

# Example: 1-D GMM

- Blue curve: ground truth distribution
- Sample data points from blue curve
- Red curve: estimated distribution



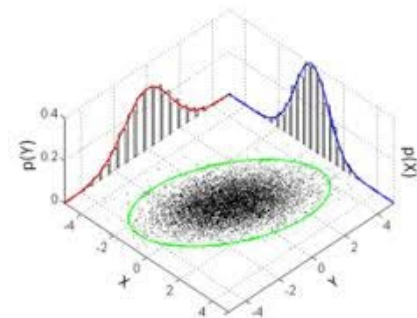
# 2-d Gaussian

- Bivariate Gaussian distribution

- Two dimensional random variable:  $X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left(\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Sigma = \begin{pmatrix} \sigma_1^2 & \sigma(X_1, X_2) \\ \sigma(X_1, X_2) & \sigma_2^2 \end{pmatrix}\right)$$

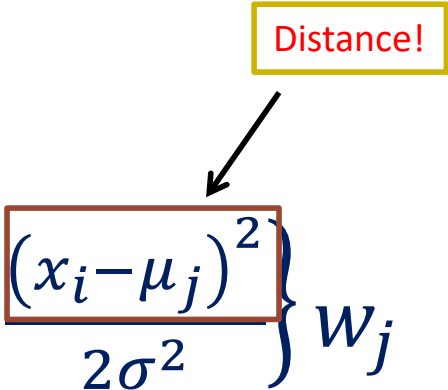
- $\mu_1$  and  $\mu_2$  are means of  $X_1$  and  $X_2$
- $\sigma_1$  and  $\sigma_2$  are standard deviations of  $X_1$  and  $X_2$
- $\sigma(X_1, X_2)$  is the covariance between  $X_1$  and  $X_2$ ,  
i.e.,  $\sigma(X_1, X_2) = E(X_1 - \mu_1)(X_2 - \mu_2)$



# Apply EM algorithm: 2-d

- An iterative algorithm (at iteration  $t+1$ )
  - E(expectation)-step
    - Evaluate the weight  $w_{ij}$  when  $\mu_j, \Sigma_j, w_j$  are given
    - $w_{ij}^{t+1} = \frac{w_j^t p(x_i | \mu_j^t, \Sigma_j^t)}{\sum_j w_j^t p(x_i | \mu_j^t, \Sigma_j^t)}$
  - M(maximization)-step
    - Find  $\mu_j, \Sigma_j, w_j$  that maximize the weighted likelihood, where  $w_{ij}$ 's are weights:  $\sum_{ij} w_{ij}^{t+1} \log w_j p(x_i | \mu_j, \Sigma_j)$
    - It is equivalent to Gaussian distribution parameter estimation when each point has a weight belonging to each distribution
    - $\mu_j^{t+1} = \frac{\sum_i w_{ij}^{t+1} x_i}{\sum_i w_{ij}^{t+1}}; (\sigma_{j,1}^2)^{t+1} = \frac{\sum_i w_{ij}^{t+1} \|x_{i,1} - \mu_{j,1}^{t+1}\|^2}{\sum_i w_{ij}^{t+1}}; (\sigma_{j,2}^2)^{t+1} = \frac{\sum_i w_{ij}^{t+1} \|x_{i,2} - \mu_{j,2}^{t+1}\|^2}{\sum_i w_{ij}^{t+1}};$
    - $(\sigma(X_1, X_2)_j)^{t+1} = \frac{\sum_i w_{ij}^{t+1} (x_{i,1} - \mu_{j,1}^{t+1})(x_{i,2} - \mu_{j,2}^{t+1})}{\sum_i w_{ij}^{t+1}}; w_j^{t+1} \propto \sum_i w_{ij}^{t+1}$

# K-Means: A Special Case of Gaussian Mixture Model

- When each Gaussian component with covariance matrix  $\sigma^2 I$ , and with the same size  $w_j$ 
  - Soft K-means
    - $w_{ij} \propto p(x_i | \mu_j, \sigma^2) w_j \propto \exp \left\{ - \frac{(x_i - \mu_j)^2}{2\sigma^2} \right\} w_j$ 
- When  $\sigma^2 \rightarrow 0$ 
  - Soft assignment becomes hard assignment
  - $w_{ij} \rightarrow 1$ , if  $x_i$  is closest to  $\mu_j$  (why?)

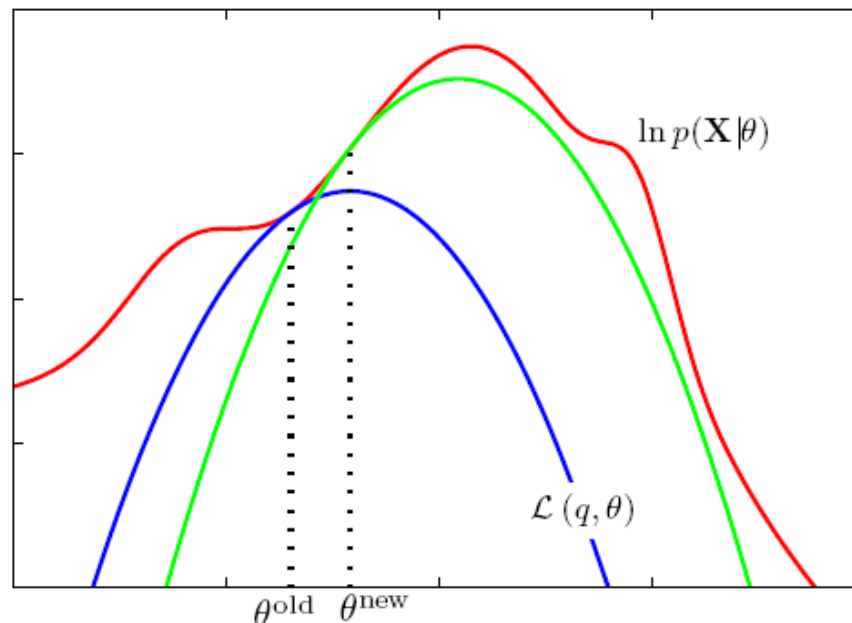
# Mapping Soft Clustering to Hard Clustering

---

- For evaluation purpose
  - $j^* = \operatorname{argmax}_j w_{ij}$
  - $w_{ij^*} = 1; w_{ij} = 0$  for all other  $j \neq j^*$
- Example:
  - $K = 3$ ; the output of GMM for object  $i$  is
    - $w_{i1} = 0.7, w_{i2} = 0.2, w_{i3} = 0.1$
    - $\Rightarrow$  mapping result: assign  $i$  to cluster 1

# Why EM Works?

- E-Step: computing a **tight** lower bound  $L$  of the original objective function  $l$  at  $\theta_{old}$
- M-Step: find  $\theta_{new}$  to maximize the lower bound
- $l(\theta_{new}) \geq L(\theta_{new}) \geq L(\theta_{old}) = l(\theta_{old})$



# How to Find Tight Lower Bound?

- $$\begin{aligned}\ell(\theta) &= \log \sum_h p(d, h; \theta) \\ &= \log \sum_h \frac{q(h)}{q(h)} p(d, h; \theta) \\ &= \log \sum_h q(h) \frac{p(d, h; \theta)}{q(h)}\end{aligned}$$

*q(h): the key to tight lower bound  
we want to get*

- Jensen's inequality

- $$\log \sum_h q(h) \frac{p(d, h; \theta)}{q(h)} \geq \sum_h q(h) \log \frac{p(d, h; \theta)}{q(h)}$$

*the tight lower bound*

- When “=” holds to get a tight lower bound?

- $q(h) = p(h|d, \theta)$  (why?)

# In GMM Case

---

$$L(D; \theta) = \sum_i \log \sum_j w_j p(x_i | \mu_j, \sigma_j^2)$$

$$\geq \sum_i \sum_j w_{ij} \left( \underbrace{\log w_j p(x_i | \mu_j, \sigma_j^2)}_{\log L(x_i, z_i = j | \theta)} - \underbrace{\log w_{ij}}_{\text{Does not involve } \theta, \text{ can be dropped}} \right)$$

$\log L(x_i, z_i = j | \theta)$

Does not involve  $\theta$ ,  
can be dropped


# Advantages and Disadvantages of GMM

---

- **Strength**
  - Mixture models are more general than partitioning: different densities and sizes of clusters
  - Clusters can be characterized by a small number of parameters
  - The results may satisfy the statistical assumptions of the generative models
- **Weakness**
  - Converge to local optimal (overcome: run multi-times w. random initialization)
  - Computationally expensive if the number of distributions is large
  - Hard to estimate the number of clusters
  - Can only deal with spherical clusters

# Clustering

---

- Clustering
- K-means
- Kernel K-means
- Mixture Model and EM algorithm
- Summary 

# Summary

---

- Revisit k-means
  - Objective function, Limitations
- Kernel k-means
  - Distance in higher-dimensional space
- Mixture models
  - Gaussian mixture model; multinomial mixture model; EM algorithm; Connection to k-means