CS247: ADVANCED DATA MINING

3: Basics: K-Means and Mixture Model

Instructor: Yizhou Sun

yzsun@cs.ucla.edu

January 22, 2024

Announcement

- Midterm time and location is still pending
  - Evening exam might not be approved
  - If that is the case, we need to take an in-class exam on 2/21
    - Schedule needs to be slightly adjusted
- HW1 will be out today, due in 1 week
  - Start working early!

Question from last lecture

- In logistic regression, what if we change the class labels? Will that affect the model?

Clustering

- Clustering 
- 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

Problem Formalization

- Given a dataset $D = \{\boldsymbol{x}_i\}_{i=1}^{N}$
  - Output the latent clustering structure $p(z_i|\boldsymbol{x}_i)$

slido

Join at slido.com
#584205

ⓘ Start presenting to display the joining instructions on this slide.

slido

Write down the clustering algorithms that you have learned.

ⓘ Start presenting to display the poll results on this slide.

Applications of Cluster Analysis

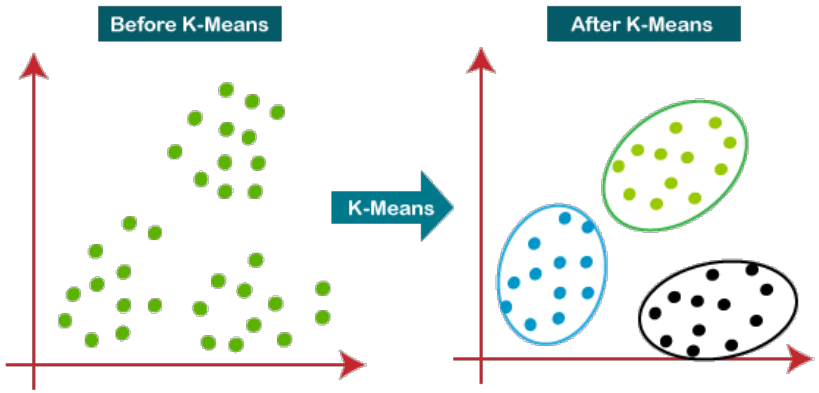
- Data reduction
  - Summarization: Preprocessing for regression, PCA, classification, and association analysis
  - Compression: Image processing: vector quantization
- Prediction based on groups
  - Cluster & find characteristics/patterns for each group
  - E.g., build a recommendation model for each subgroup of customers.
- 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
- Mixture Model and EM algorithm
- Summary

Recall K-Means

- Given a dataset $D = \{\boldsymbol{x}_i\}_{i=1}^{N}$, and the cluster number K
  - Partition the data points into K clusters, such that the total within-cluster variance is minimized
  - Objective function
    - $J = \sum_{j=1}^{k} \sum_{C(i)=j} ||x_i - c_j||^2$
    - $C(i) = j$: ith cluster label is j; c_j is the cluster center for jth cluster



Re-arrange the objective function

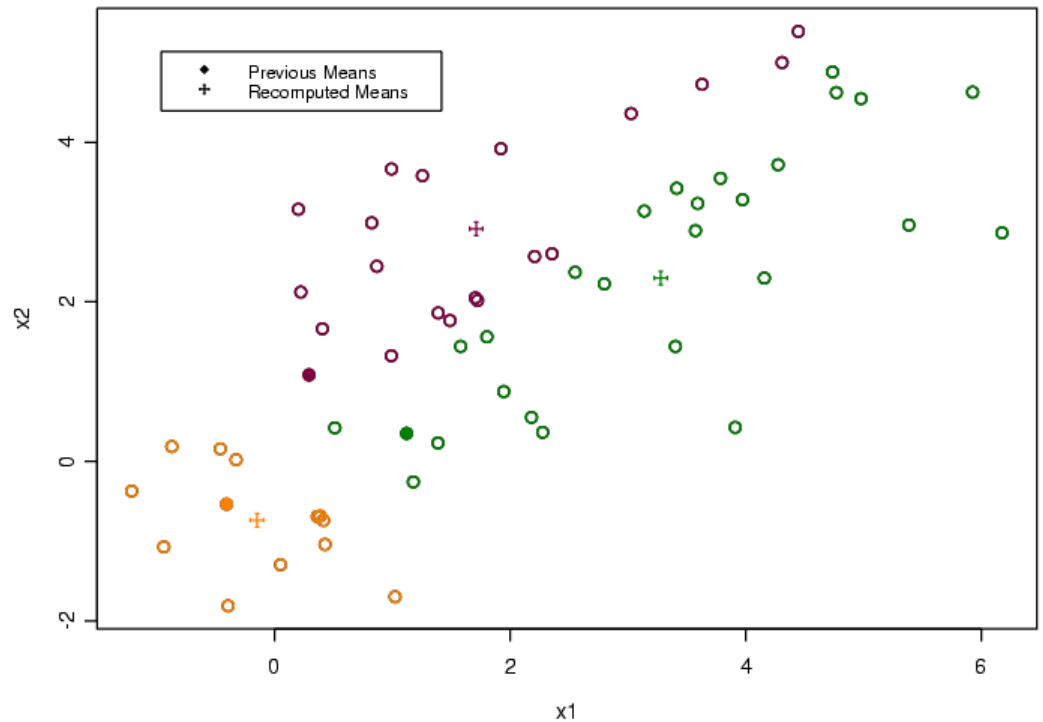
- Objective function
  - $J = \sum_{j=1}^{k} \sum_{C(i)=j} ||x_i - c_j||^2$
- Re-arrange the objective function
  - $J = \sum_{j=1}^{k} \sum_{i} w_{ij} ||x_i - c_j||^2$
    - $w_{ij} \in \{0,1\}$
    - $w_{ij} = 1, if\ x_i\ belongs\ to\ cluster\ j; w_{ij} = 0, otherwise$
  - Looking for:
    - The best assignment w_{ij}
    - The best center c_j

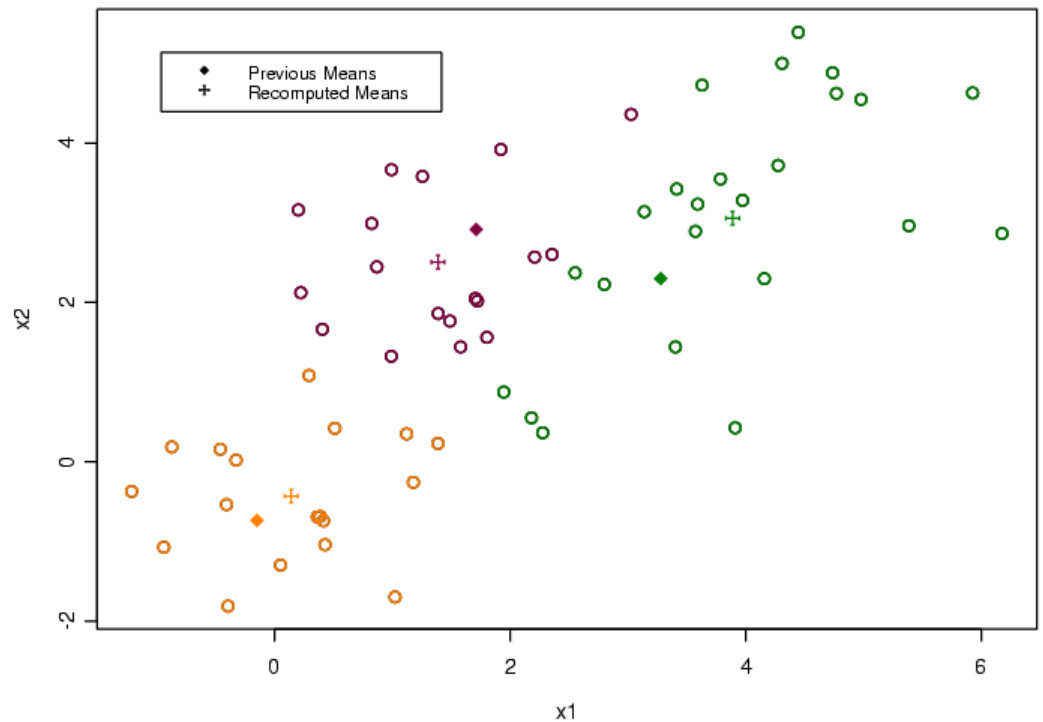
Solution of K-Means

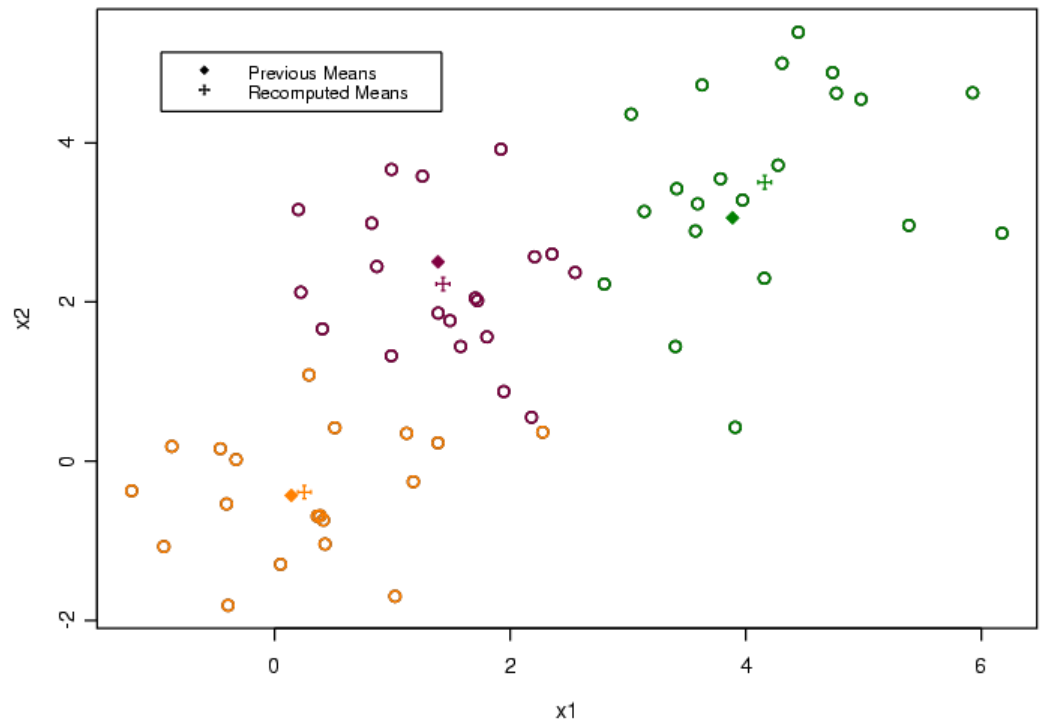
- Iterations

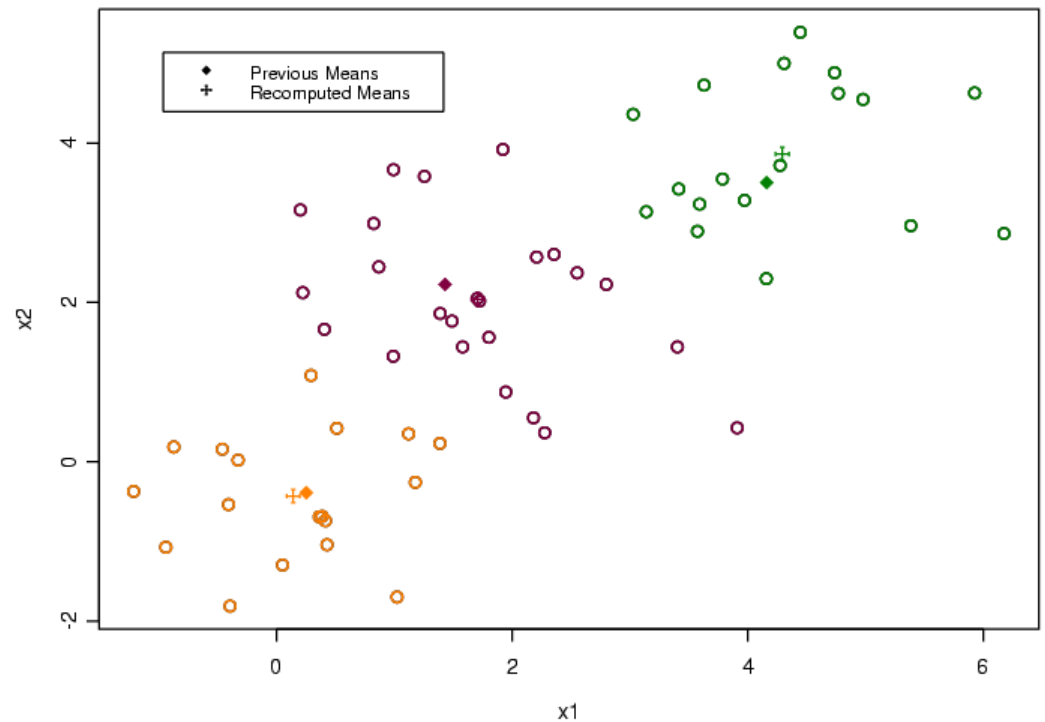
$$J = \sum_{j=1}^{k} \sum_{i} w_{ij} ||x_i - c_j||^2 = \sum_{i} \sum_{j} w_{ij} ||x_i - c_j||^2$$

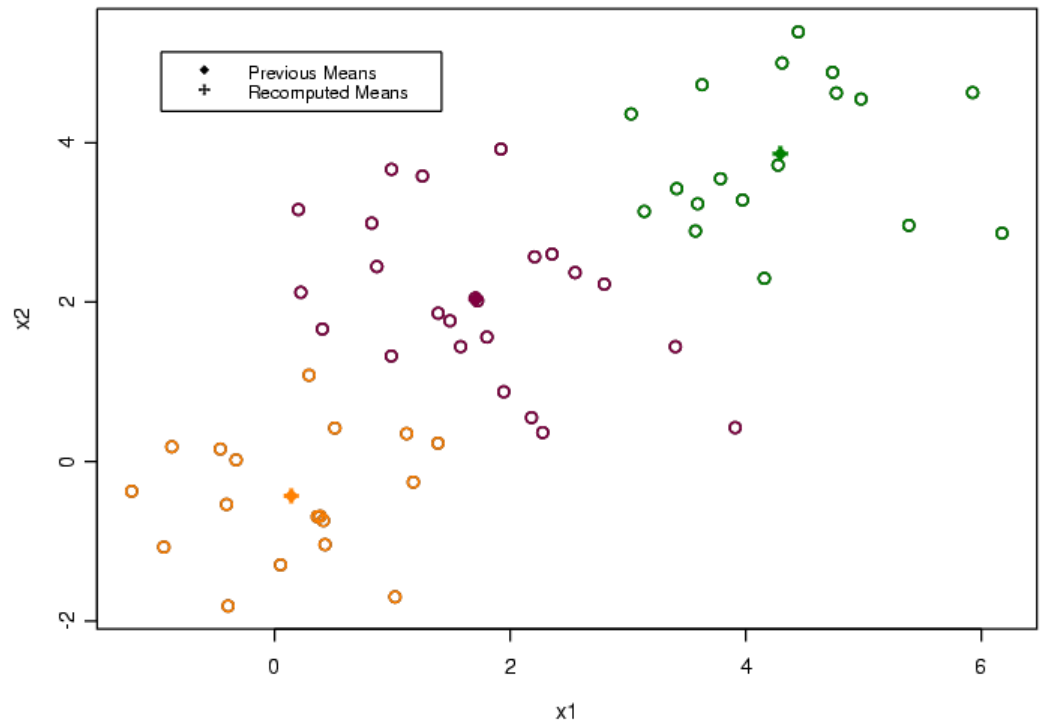
  - Step 1: Fix centers c_j, find assignment w_{ij} that minimizes J for each i
    - => $w_{ij} = 1, if\ ||x_i - c_j||^2$ is the smallest
  - Step 2: Fix assignment w_{ij}, find centers that minimize J
    - => first derivative of J = 0
    - => $\frac{\partial J}{\partial c_j} = -2 \sum_i w_{ij}(x_i - c_j) = 0$
    - => $c_j = \frac{\sum_i w_{ij} x_i}{\sum_i w_{ij}}$
      - Note $\sum_i w_{ij}$ is the total number of objects in cluster j

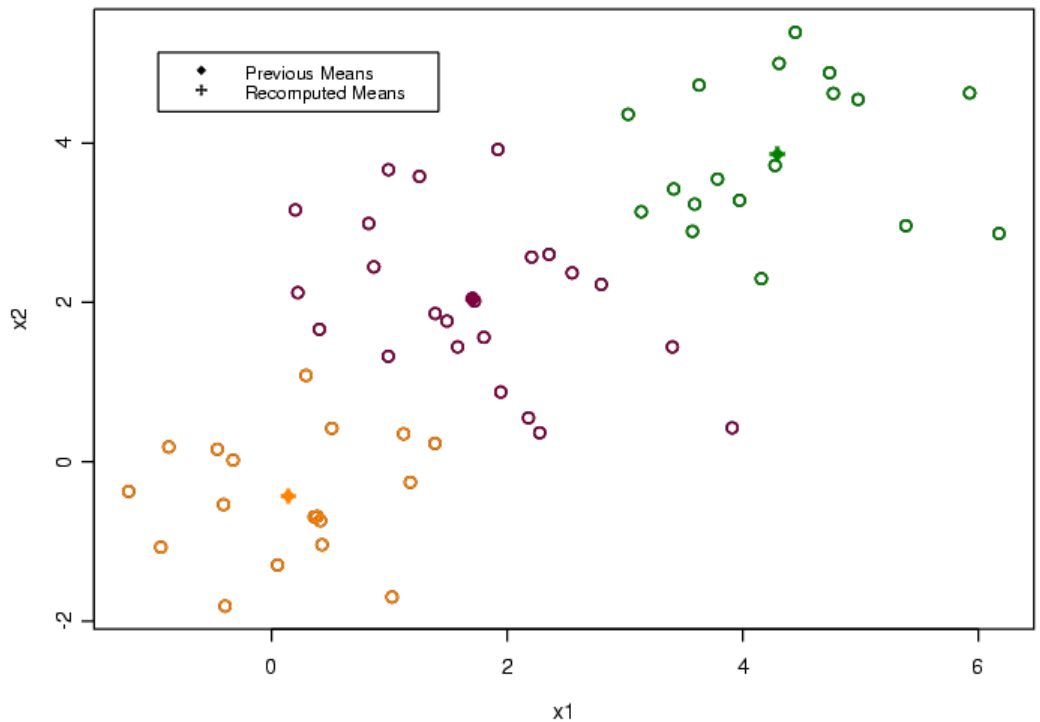

Previous Means
Recomputed Means
x2
x1
-2
0
2
4
0
2
4
6


Previous Means
Recomputed Means
x2
x1
4
2
0
-2
0
2
4
6


Previous Means
Recomputed Means
x2
x1
-2
0
2
4
0
2
4
6


Previous Means
Recomputed Means
x2
x1
4
2
0
-2
0
2
4
6


Previous Means
Recomputed Means
x2
x1
-2
0
2
4
0
2
4
6



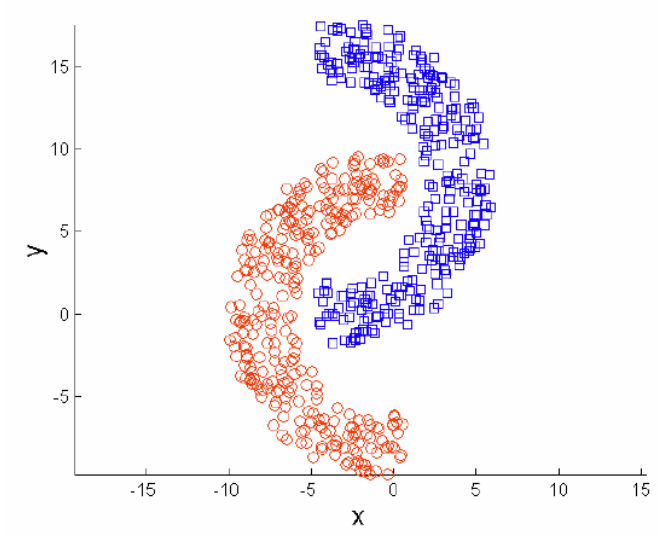
Converges! Why?

Q&A

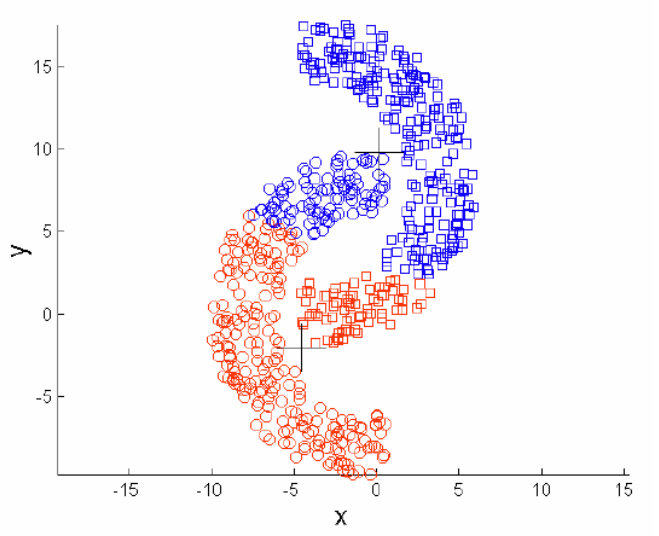
Limitations of K-Means

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

Limitations of K-Means: Non-Spherical Shapes

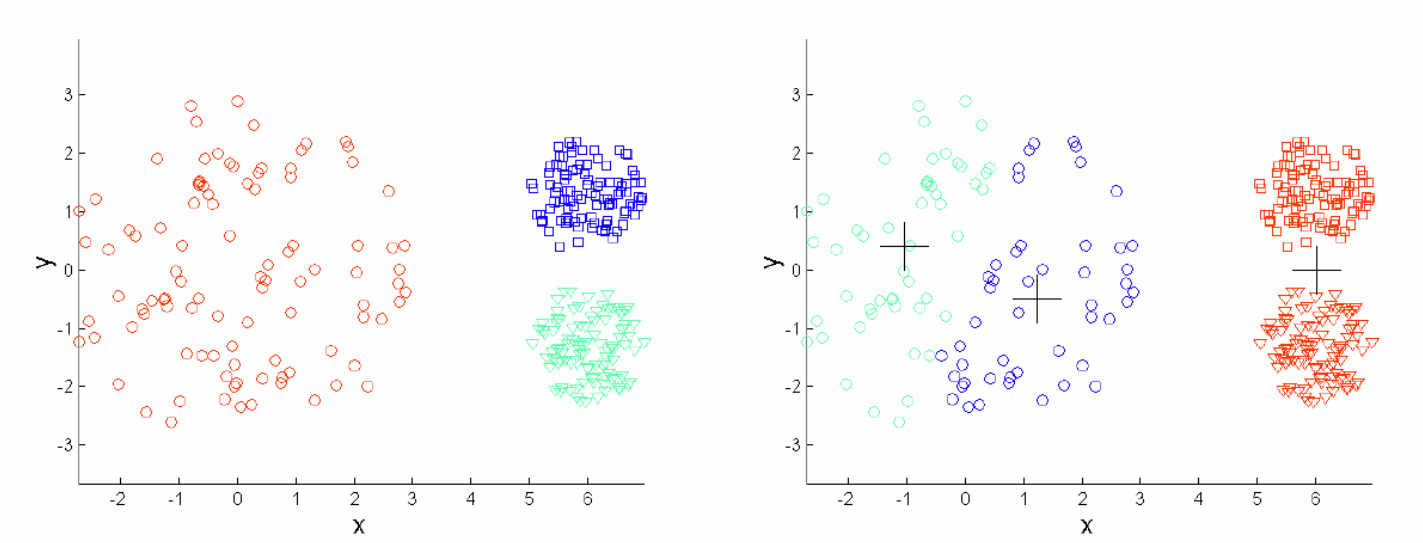


Original Points



K-means (2 Clusters)

Limitations of K-Means: Different Sizes and Variances

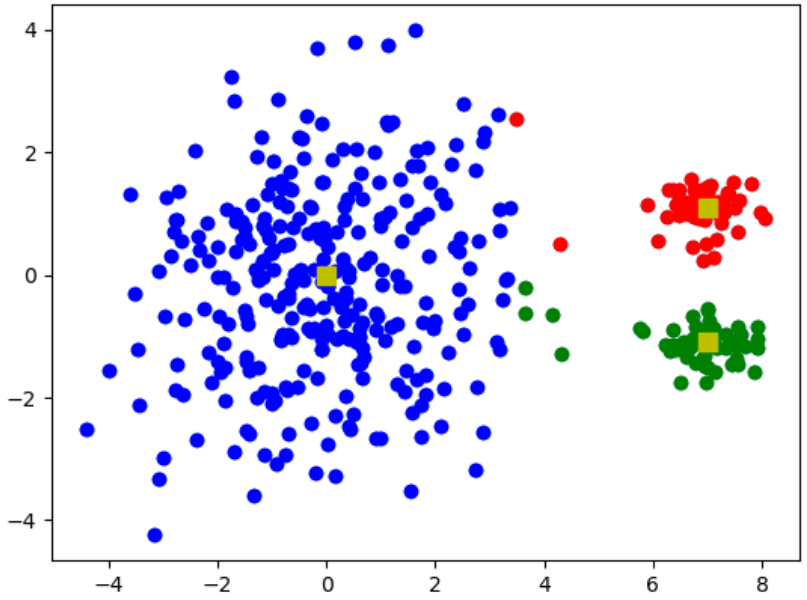


Original Points

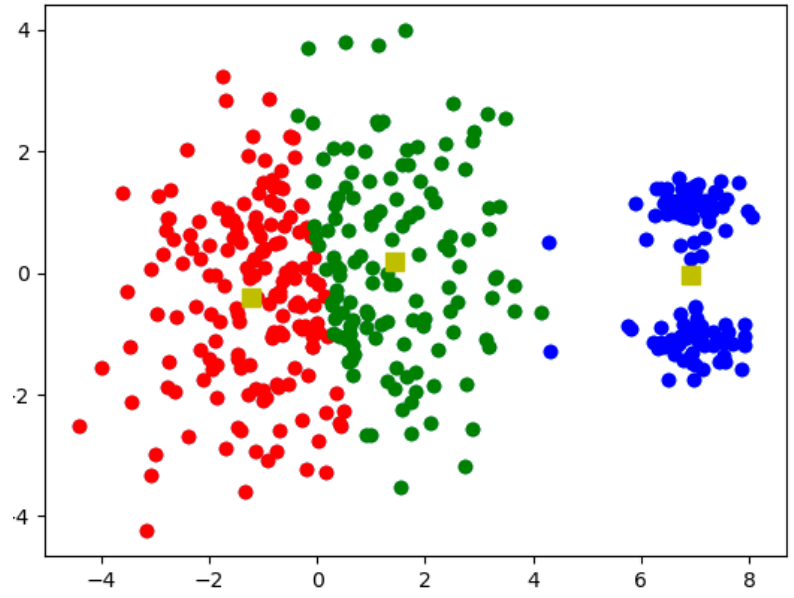
K-means (3 Clusters)

Example

- Consider the cost of K-means in two cases



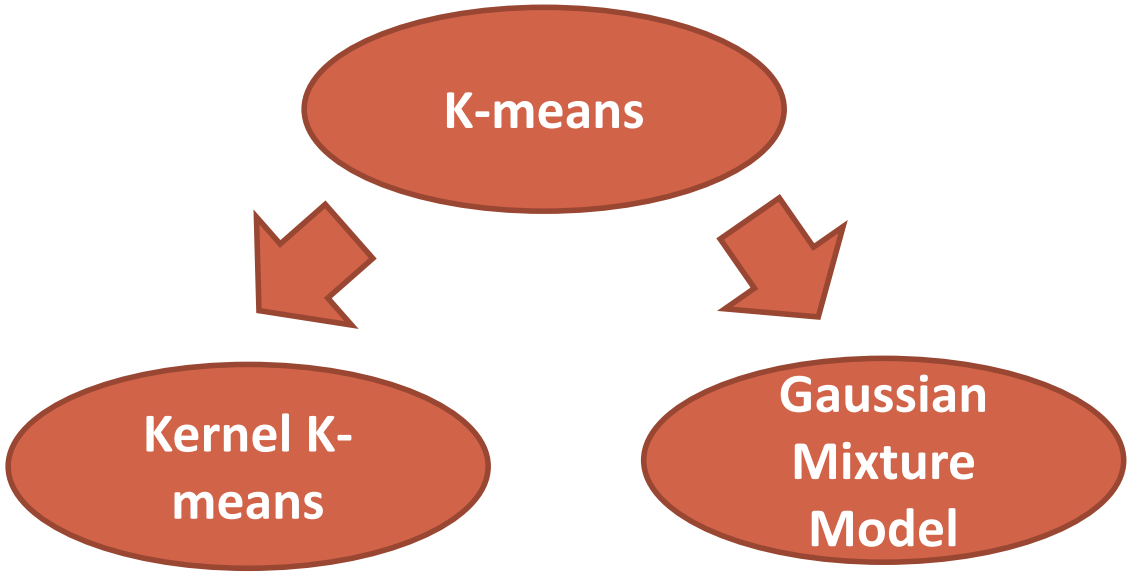
Cost: J = 1560.86



Cost: J = 1147.42

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

Connections of K-means to Other Methods



Clustering

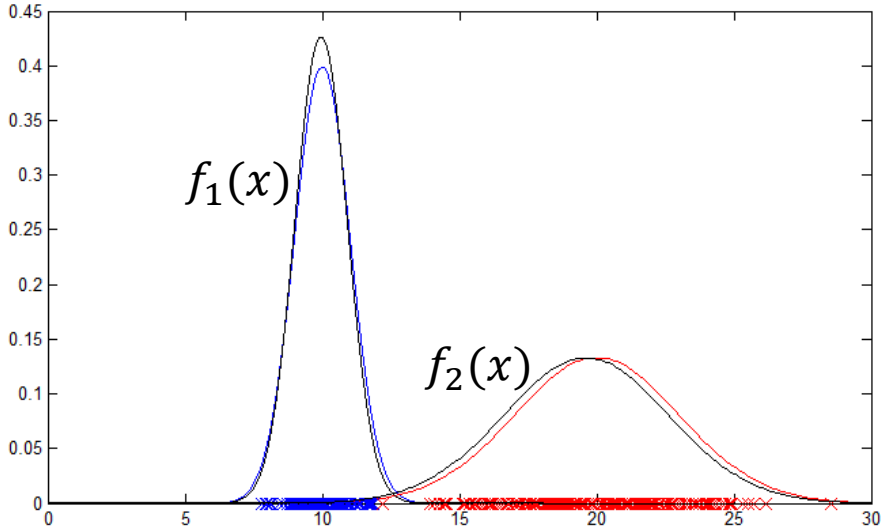
- Clustering
- K-means
- Mixture Model and EM algorithm 
- Summary

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
    - $w_{ij} \in [0,1]\ and\ \sum_j w_{ij} = 1$

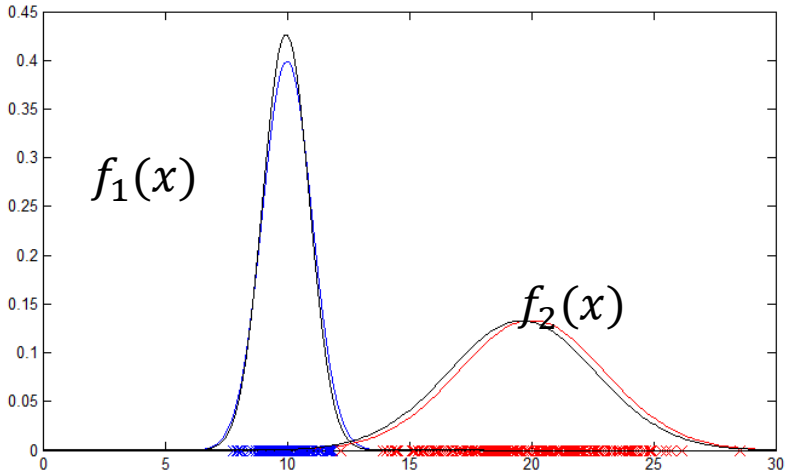
Mixture Model-Based Clustering

- Each cluster C_j *corresponds to a distribution over data points*
  - probability density/mass functions: $f_j(x; \theta_j)$
- Relative cluster size:
  - prior probabilities: $w_1, \ldots, w_K, \sum_j w_j = 1$



Likelihood of the dataset

- Under I.I.D assumption
  - $L(\Theta|D) = P(D|\Theta) = \prod_i p(x_i|\Theta)$
  - $\Theta = \{\theta_1, \ldots, \theta_K; w_1, \ldots, w_K\}$
- Joint Probability of an object *i and its* cluster C_j is:
  - $p(x_i, z_i = j|\Theta) = w_j f_j(x_i|\theta_j)$
  - z_i: hidden random variable
- Probability of *i* is:
  - $p(x_i|\Theta) = \sum_j w_j f_j(x_i|\theta_j)$



The inference problem

- Which cluster does x_i belong to?
  - $p(z_i = j|x_i) = p(x_i, z_i)/p(x_i)$

Maximum Likelihood Estimation

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

$$L(\Theta|D) = \prod_i p(x_i) = \prod_i \sum_j w_j f_j(x_i|\theta_j)$$

$$\Rightarrow logL(\Theta|D) = \sum_i logp(x_i|\Theta) = \sum_i log \sum_j w_j f_j(x_i|\theta_j)$$

- The learning task: Find *best* Θ 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
    - $w_{ij}^{(t+1)} = p(z_i = j|\Theta^{(t)}, x_i) \propto p(x_i|z_i = j, \Theta^{(t)})p(z_i = j|\Theta^{(t)})$
      $f_j(x_i)$ w_j
  - **M-step** finds the new clustering or parameters that maximize the expected log complete likelihood, with respect to conditional distribution $p(z_i = j|\Theta^{(t)}, x_i)$
    - $\Theta^{(t+1)} = argmax_\Theta \sum_i \sum_j w_{ij}^{(t+1)} \log p(x_i, z_i = j|\Theta)$

Gaussian Mixture Model

- Generative model
  - For each object:
    - Pick its cluster, i.e., a distribution component: $Z \sim Categorical(w_1, \ldots, w_K)$
    - 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 Θ here?

Estimating Parameters

- $L(D;\Theta) = \sum_i \log \sum_j w_j p(x_i|\mu_j, \sigma_j^2)$
  - Considering the first derivative of μ_j:
    - $$\frac{\partial L}{\partial u_j} = \sum_i \frac{w_j}{\sum_{j'} w_{j'} p(x_i|\mu_{j'}, \sigma_{j'}^2)} \frac{\partial p(x_i|\mu_j, \sigma_j^2)}{\partial \mu_j}$$

$$= \sum_i \frac{w_j p(x_i|\mu_j, \sigma_j^2)}{\sum_{j'} w_{j'} p(x_i|\mu_{j'}, \sigma_{j'}^2)} \frac{1}{p(x_i|\mu_j, \sigma_j^2)} \frac{\partial p(x_i|\mu_j, \sigma_j^2)}{\partial \mu_j}$$

$$= \sum_i \boxed{\frac{w_j p(x_i|\mu_j, \sigma_j^2)}{\sum_{j'} w_{j'} p(x_i|\mu_{j'}, \sigma_{j'}^2)}} \boxed{\frac{\partial \log p(x_i|\mu_j, \sigma_j^2)}{\partial u_j}}$$

$w_{ij} = P(Z = j|X = x_i, \theta)$ $\quad$ $\partial l(x_i)/\partial \mu_j$

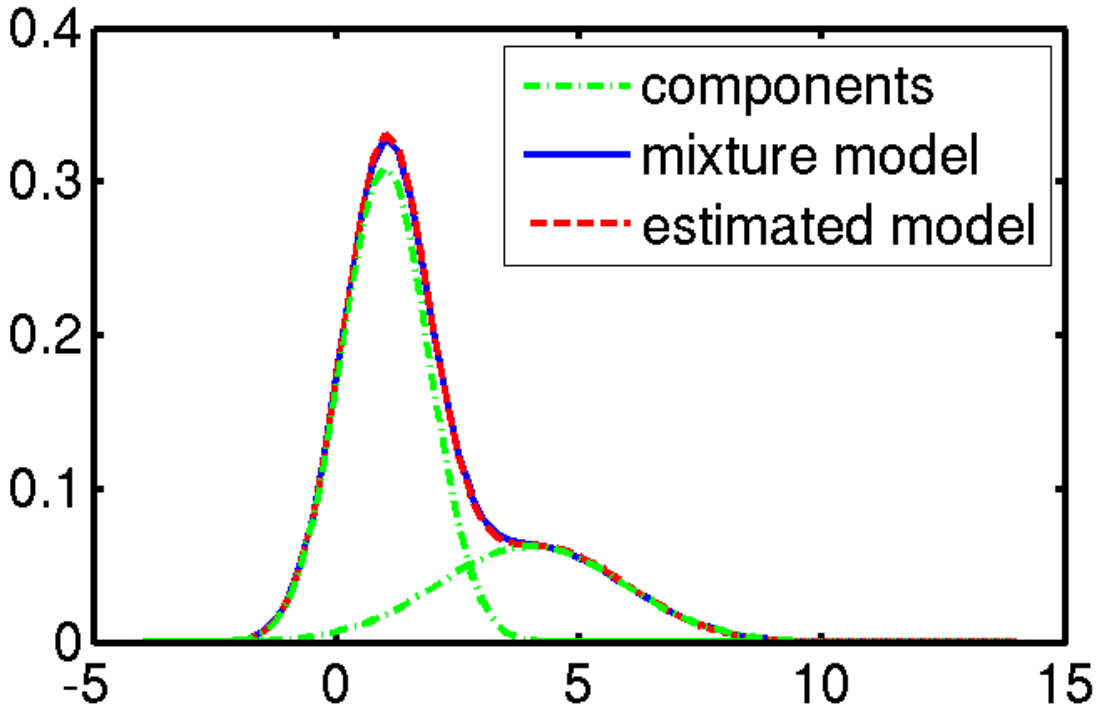
Like weighted likelihood estimation;
But the weight is determined by the parameters!

Apply EM algorithm: 1-d

- An iterative algorithm (at iteration t+1)
  - E(expectation)-step
    - Evaluate the weight w_{ij} when μ_j, σ_j, w_j are given
    - $w_{ij}^{(t+1)} = \frac{w_j^{(t)} p(x_i|\mu_j^{(t)},(\sigma_j^2)^{(t)})}{\sum_k w_k^{(t)} p(x_i|\mu_k^{(t)},(\sigma_k^2)^{(t)})}$, where $p(x_i|\mu_j^{(t)},(\sigma_j^2)^{(t)}) = f_j^{(t)}(x_i)$
  - M(maximization)-step
    - Find μ_j, σ_j, w_j that maximize the weighted log complete 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
      - $\mu_j^{(t+1)} = \frac{\sum_i w_{ij}^{(t+1)} x_i}{\sum_i w_{ij}^{(t+1)}}; (\sigma_j^2)^{(t+1)} = \frac{\sum_i w_{ij}^{(t+1)} \left(x_i - \mu_j^{(t+1)}\right)^2}{\sum_i w_{ij}^{(t+1)}}; w_j^{(t+1)} = \sum_i w_{ij}^{(t+1)} / n$

Example: 1-D GMM

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



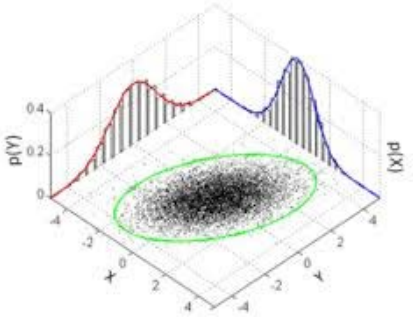
https://www.mathworks.com/matlabcentral/fileexchange/24867-gaussian_mixture_model-m

2-d Gaussian

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

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

  - $\mu_1 \; and \; \mu_2$ 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 $\boldsymbol{\mu}_j, \Sigma_j, w_j$ are given
    - $w_{ij}^{(t+1)} = \frac{w_j^{(t)} p(\boldsymbol{x}_i | \boldsymbol{\mu}_j^{(t)}, \Sigma_j^{(t)})}{\sum_j w_j^{(t)} p(\boldsymbol{x}_i | \boldsymbol{\mu}_j^{(t)}, \Sigma_j^{(t)})}$
  - M(maximization)-step
    - Find $\boldsymbol{\mu}_j, \Sigma_j, w_j$ that maximize the weighted log complete likelihood, where w_{ij}'s are weights: $\sum_{ij} w_{ij}^{(t+1)} log w_j p(\boldsymbol{x}_i | \boldsymbol{\mu}_j, \Sigma_j)$
    - It is equivalent to Gaussian distribution parameter estimation when each point has a weight belonging to each distribution
      - $\boldsymbol{\mu}_j^{(t+1)} = \frac{\sum_i w_{ij}^{(t+1)} \boldsymbol{x}_i}{\sum_i w_{ij}^{(t+1)}}; (\sigma_{j,1}^2)^{t+1} = \frac{\sum_i w_{ij}^{(t+1)} \left\| x_{i,1} - \mu_{j,1}^{(t+1)} \right\|^2}{\sum_i w_{ij}^{t+1}}; (\sigma_{j,2}^2)^{(t+1)} = \frac{\sum_i w_{ij}^{(t+1)} \left\| x_{i,2} - \mu_{j,2}^{(t+1)} \right\|^2}{\sum_i w_{ij}^{t+1}};$
      - $(\sigma(X_1, X_2)_j)^{(t+1)} = \frac{\sum_i w_{ij}^{(t+1)} (x_{i,1} - \mu_{j,1}^{(t+1)})(x_{i,2} - \mu_{j,2}^{(t+1)})}{\sum_i w_{ij}^{(t+1)}}; w_j^{(t+1)} \propto \sum_i w_{ij}^{(t+1)}$

K-Means: A Special Case of Gaussian Mixture Model

- When each Gaussian component with covariance matrix $\sigma^2 I$, and with the same size $w_j = 1/K$
  - Soft K-means
  - 1D case: $w_{ij} \propto p(x_i|\mu_j, \sigma^2) w_j \propto \exp\left\{-\frac{(x_i-\mu_j)^2}{2\sigma^2}\right\} w_j$

Distance!

- When $\sigma^2 \to 0$
  - Soft assignment becomes hard assignment
  - $w_{ij} \to 1, if\ x_i$ is closest to μ_j (why?)

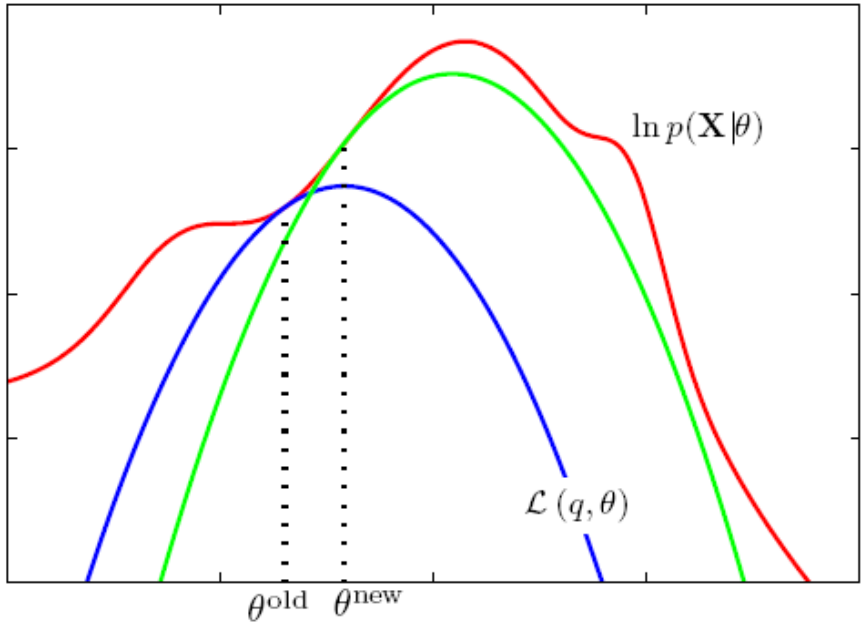
Q&A

Mapping Soft Clustering to Hard Clustering

- For evaluation purpose
  - $j^* = \text{argmax}_j w_{ij}$
  - $w_{ij^*} = 1; w_{ij} = 0 \; for \; all \; other \; j \neq j^*$
- 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 θ_{old}
- M-Step: find θ_{new} to maximize the lower bound
- $l(\theta_{new}) \geq L(\theta_{new}) \geq L(\theta_{old}) = l(\theta_{old})$



How to Find a Tight Lower Bound?

- $l(\theta) = logp(x|\theta) = log\sum_z p(z,x|\theta)$

$$= log\sum_z q(z)\frac{p(z,x|\theta)}{q(z)}$$

$q(z)$: a distribution defined over z
the key to tight lower bound we want to get

- Jensen's inequality
  - $log\sum_z q(z)\frac{p(z,x|\theta)}{q(z)} \geq \sum_z q(z)log\frac{p(z,x|\theta)}{q(z)}$

a lower bound of $l(\theta)$

  - When "=" holds to get a tight lower bound?
    - When $q(z) = p(z|x,\theta_{old})$, tight at $\theta = \theta_{old}$ (why?)

Derivation

- $log \sum_z q(z) \frac{p(z,x|\theta)}{q(z)} = \sum_z q(z) log \frac{p(z,x|\theta)}{q(z)}, when\ q(z) = p(z|x, \theta_{old})$

In GMM Case

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

$$\geq \sum_i \sum_j w_{ij} \left(\log w_j p(x_i|\mu_j, \sigma_j^2) - log w_{ij}\right)$$

$q(z_i) = p(z_i|x_i, \theta_{old})$

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

Does not involve θ, can be dropped

Q&A

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
- Summary

Summary

- Revisit k-means
  - Objective function, Limitations
- Mixture models
  - Gaussian mixture model; EM algorithm; Connection to k-means