Gaussian Mixture Models For Clustering Data

Soft Clustering and the EM Algorithm

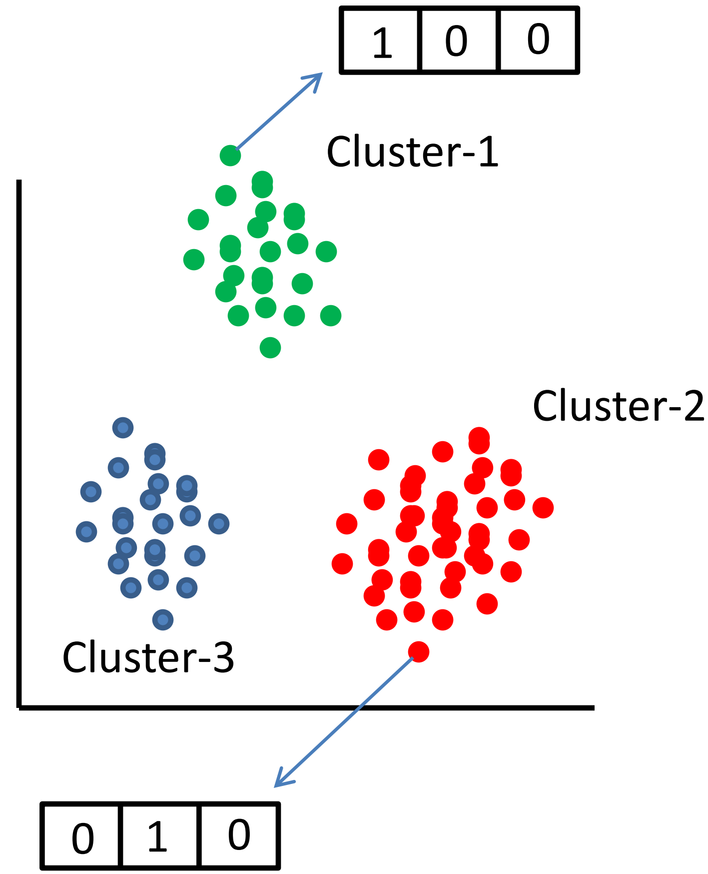
K-Means Clustering

- Input:
  - Observations: $x_i \in \mathbb{R}^d \qquad \forall i \in \{1, \ldots., N\}$
  - Number of Clusters: k

- Output:
  - Cluster Assignments.
  - Cluster Centroids: $\mu_j \in \mathbb{R}^d \qquad \forall j \in \{1, \ldots., k\}$

K-Means Clustering

- Let z_i be a binary vector of dimension ***'k'*** associated with each observation.
- If the i^{th} observation belongs to the j^{th} cluster then $z_{ij} = 1$ and all other components of ***z*** are zero.
- Thus, **z** can be considered as a cluster label vector associated with each observation.



1-of-k representation for cluster assignment.

K-Means Clustering

- We can now cast k-means as a minimization problem with the objective function:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \underbrace{\left\| x_n - \mu_k \right\|^2}_{\text{Squared Distance}}$$

- We need to find z_{nk} and μ_k that minimize J.

K-Means Clustering

- Minimizing this function w.r.t z_{nk} :
  - All ***n*** points are independent can optimize each one independently.
  - Choose z_{nk} to be ***1*** for whichever value ***k*** gives the minimum value of the squared distance.
  - Assign the current observation to the nearest cluster center.

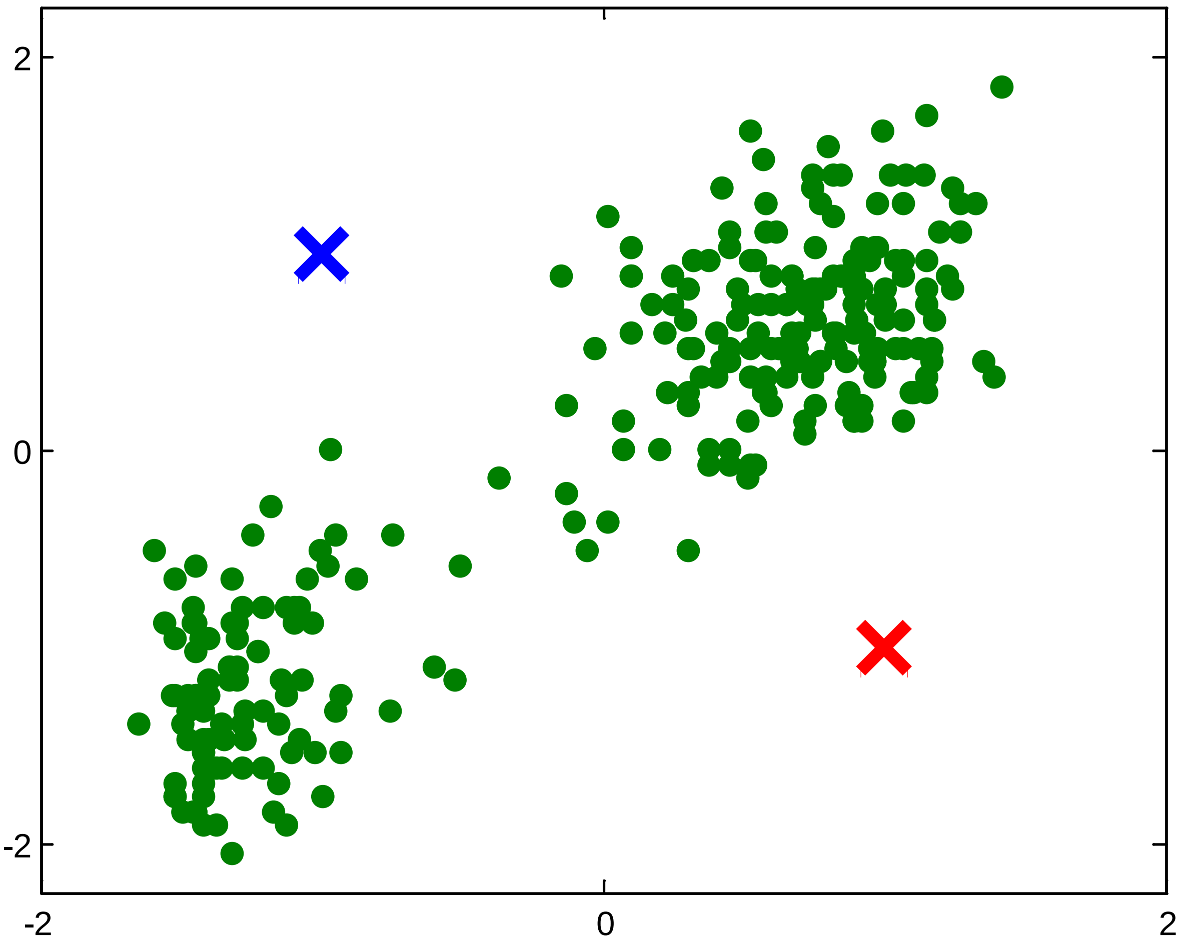
- Minimizing this function w.r.t μ_k :
  - Take the derivative of J w.r.t μ_k and equate to zero.

$$\mu_k = \frac{\sum_{n=1}^{N} z_{nk} x_n}{\sum_{n=1}^{N} z_{nk}}$$

Finally!

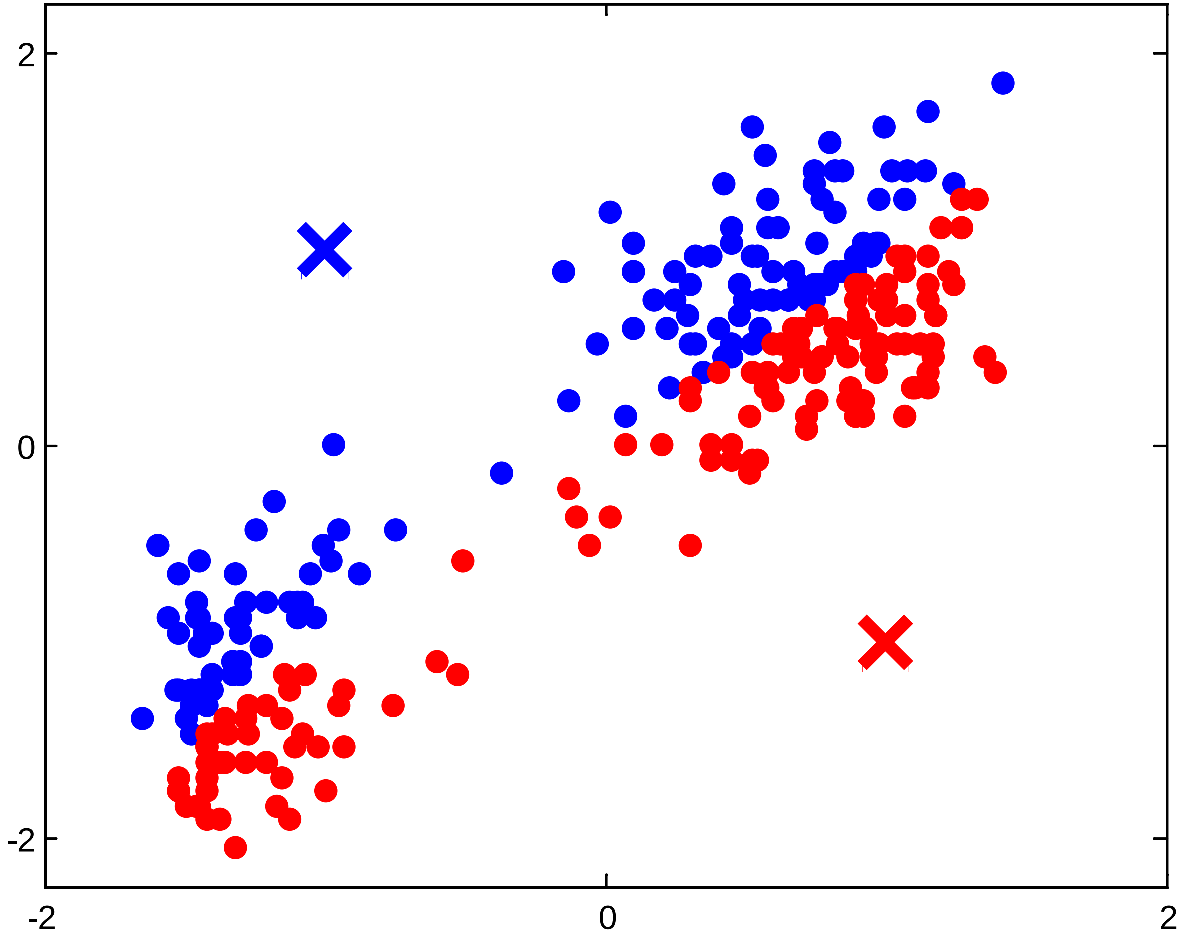
- Iterative Algorithm For K-Means:
  - Initialize ***k*** centroids.
  - Repeat till <u>*convergence:*</u>
    - Calculate z_{nk}
    - Update μ_k

Example



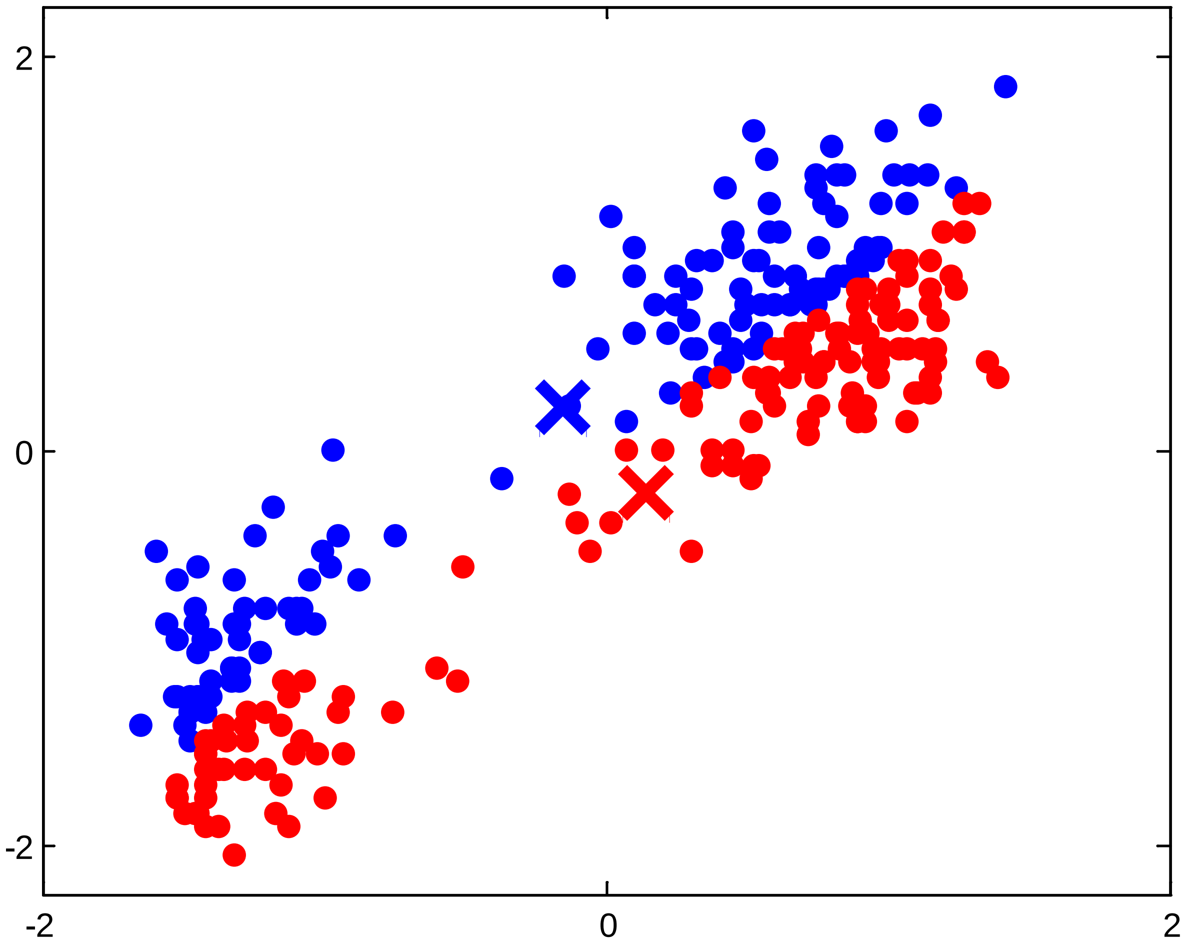
- Looking for two clusters.
- Initialize the centroids. (The blue and the red crosses).

Example



- Calculate the cluster assignments.

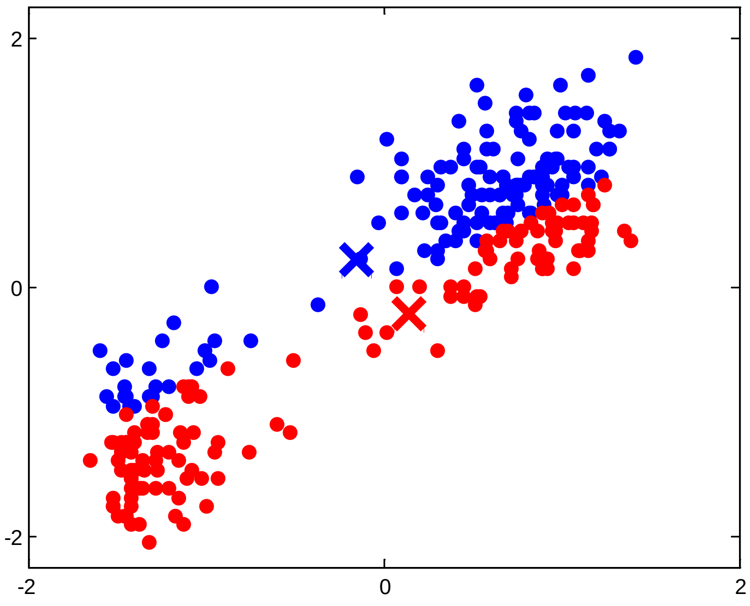
Example



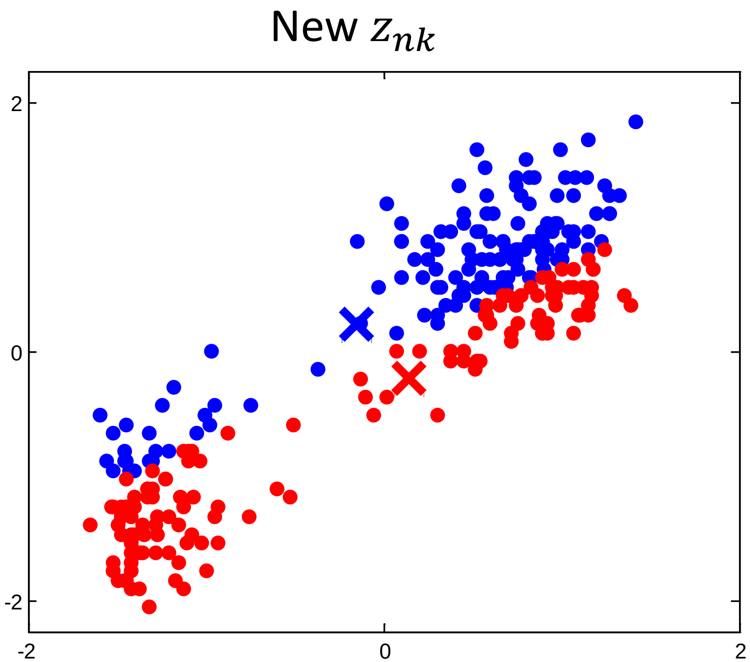
- Re-calculate the centroids based on the new cluster assignments.

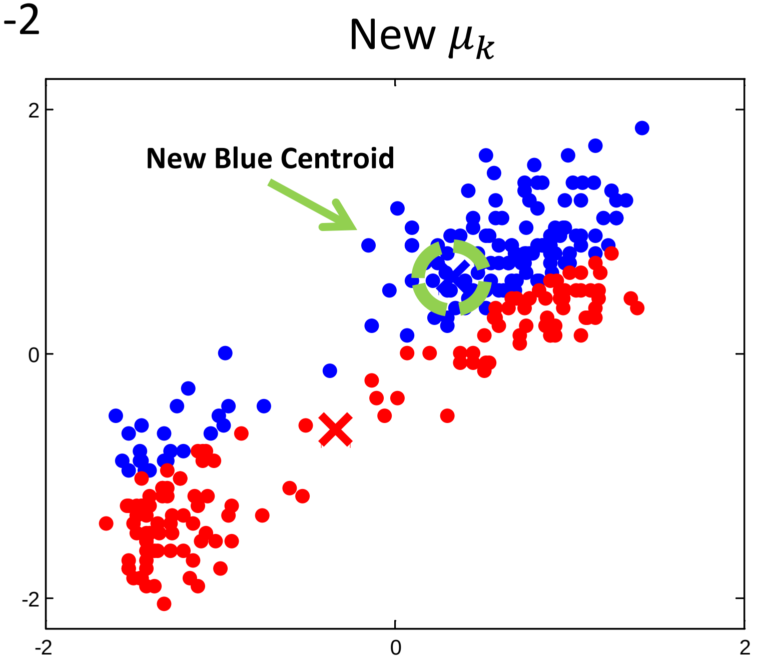
New z_{nk}

Iteration -2



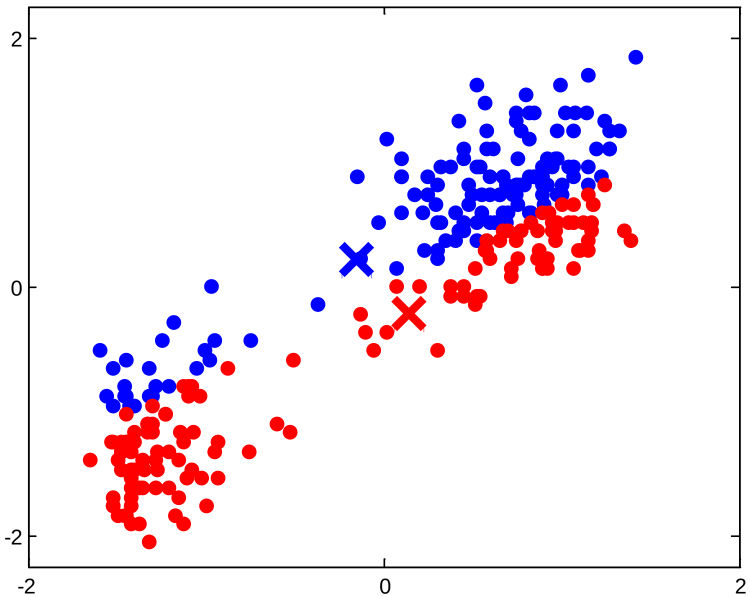
Iteration -2



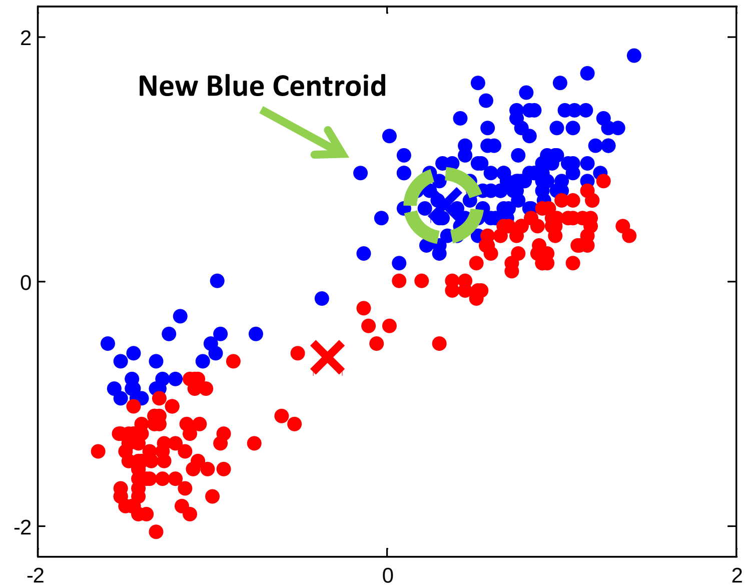


Iteration -2

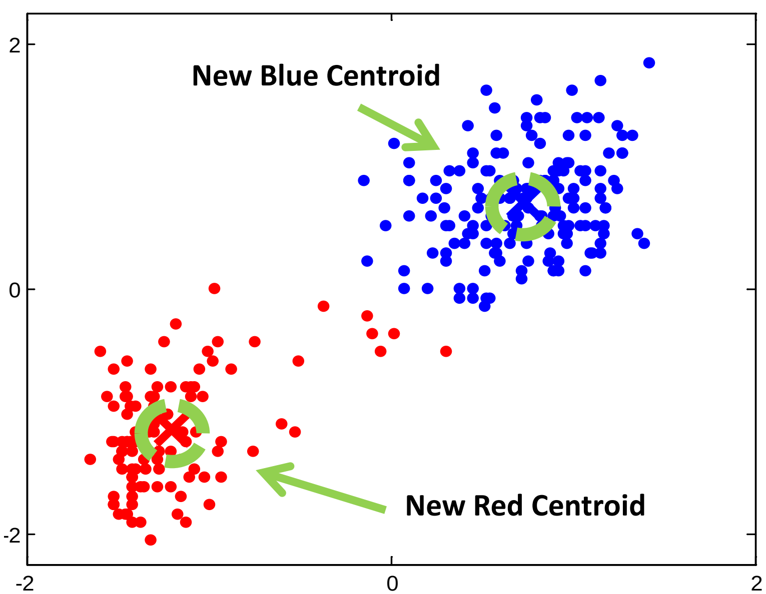
New z_{nk}



New μ_k

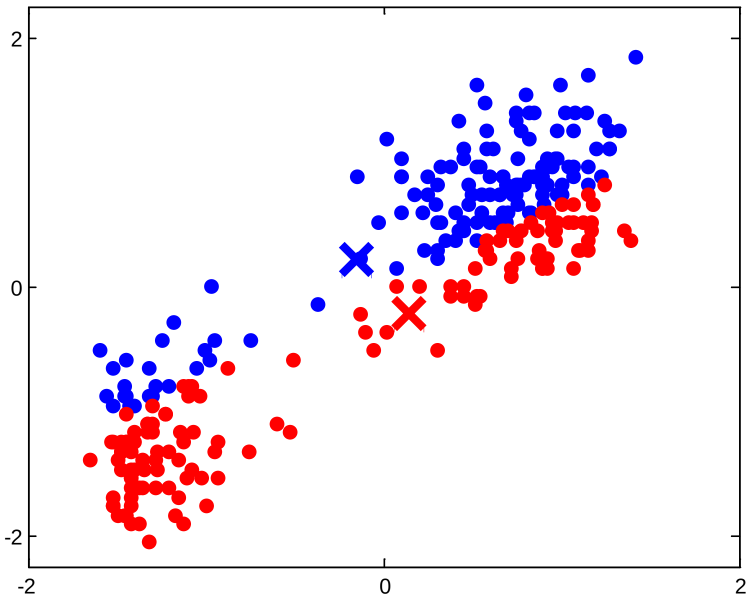


Iteration -3

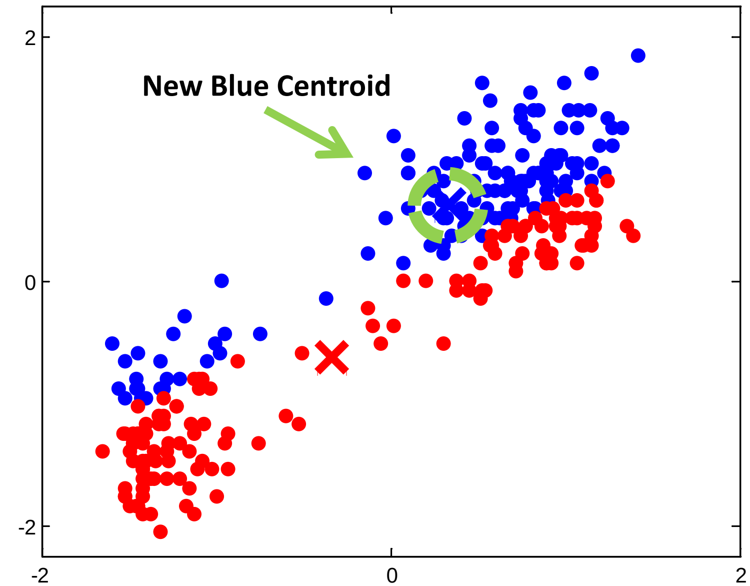


Iteration -2

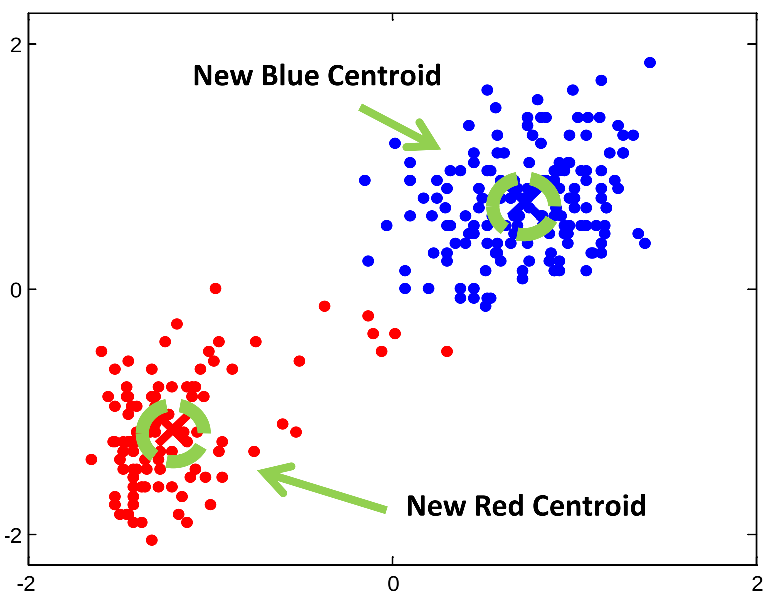
New z_{nk}



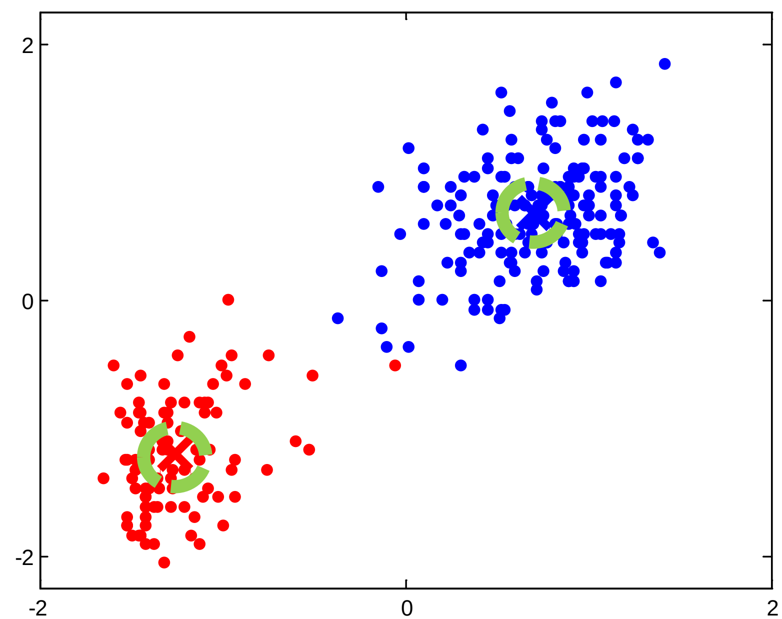
New μ_k



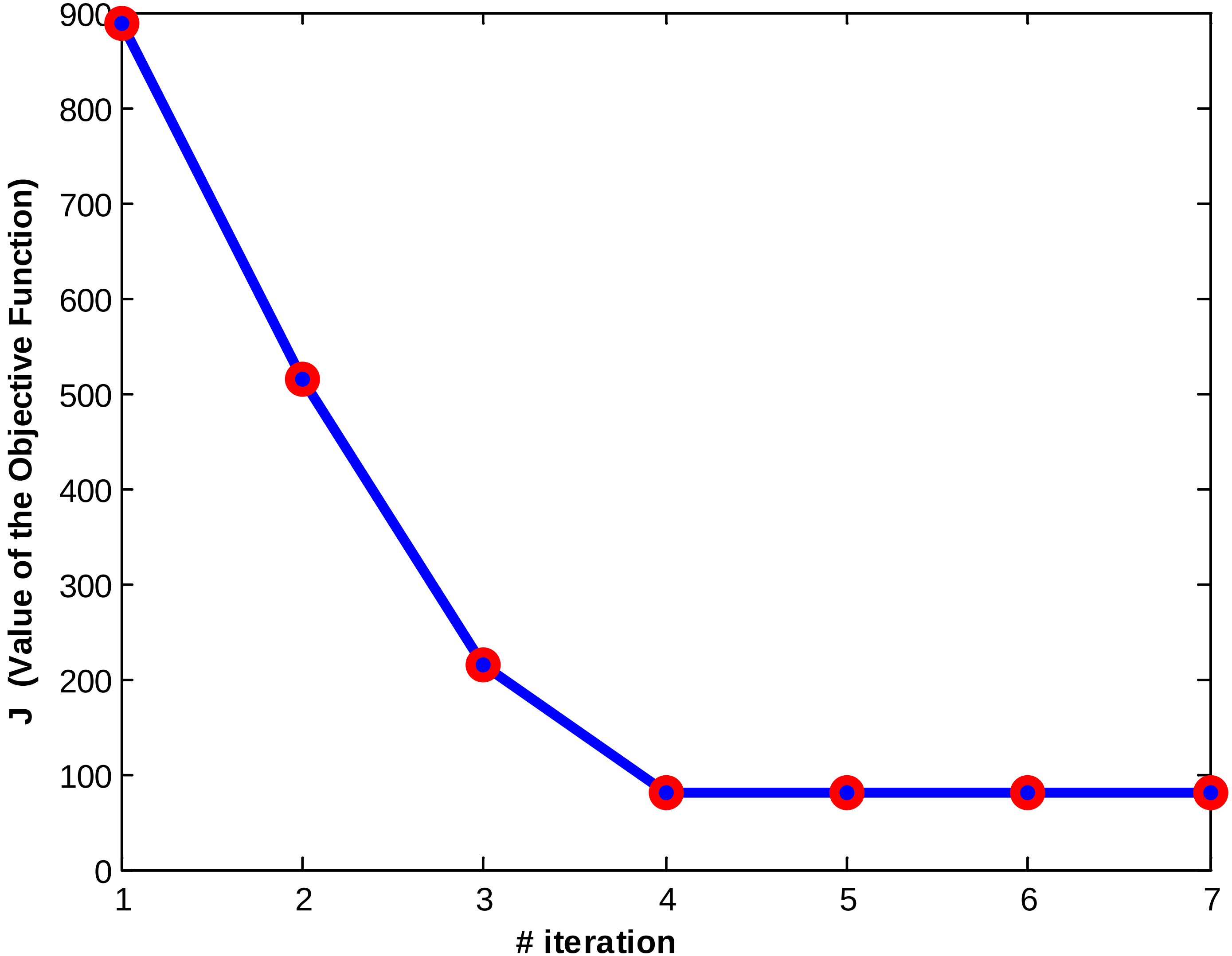
Iteration -3



Final Results



Minimizing the Objective Function

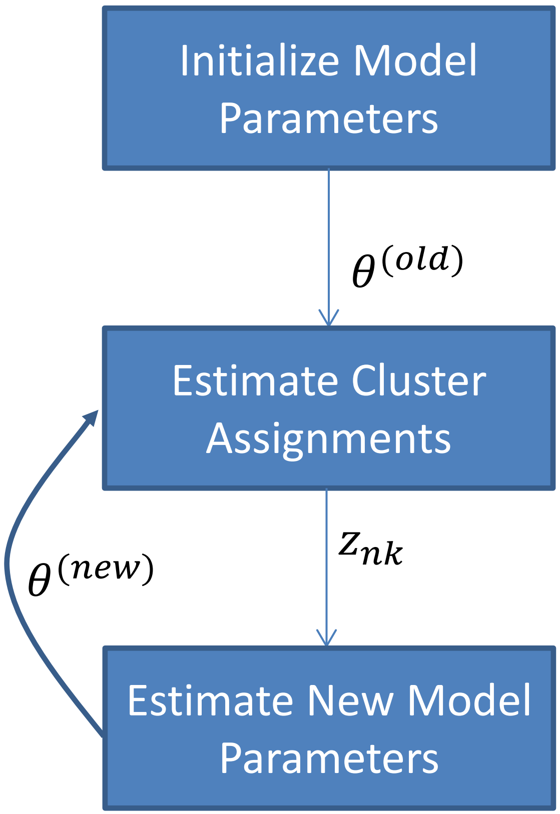


Terminology

- Model Parameters ($\theta = \{\mu_{1....k}\}$)
  - The centroids.
- Complete Data ($y = \{x_{1..n}, z_{1..n}\}$)
  - The observations along with the cluster assignments.
- Incomplete Data ($x_{1..n}$)
  - Only the observations.

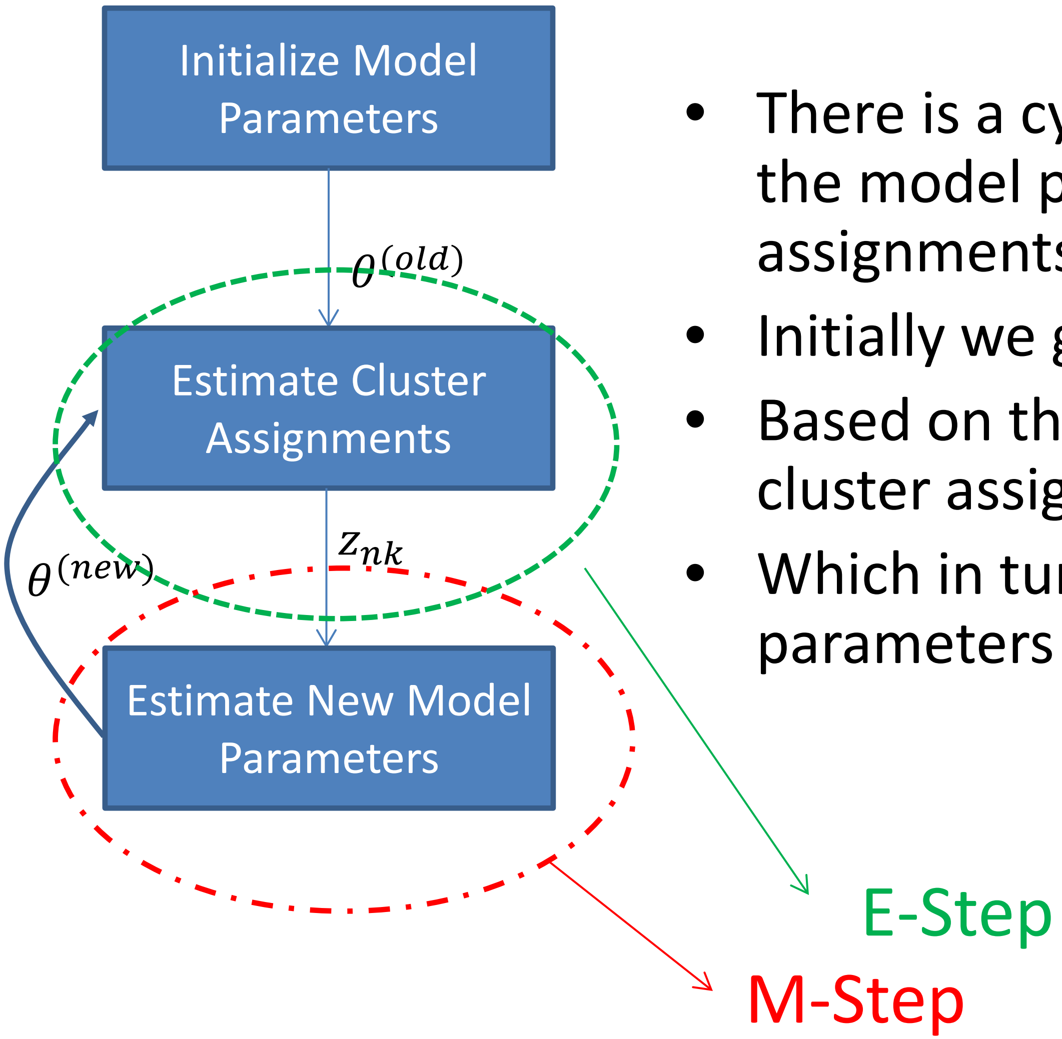
- In clustering problems we only have incomplete data and need to find estimates for both θ and $z_{1..n}$.

Dissecting the K-Means Algorithm



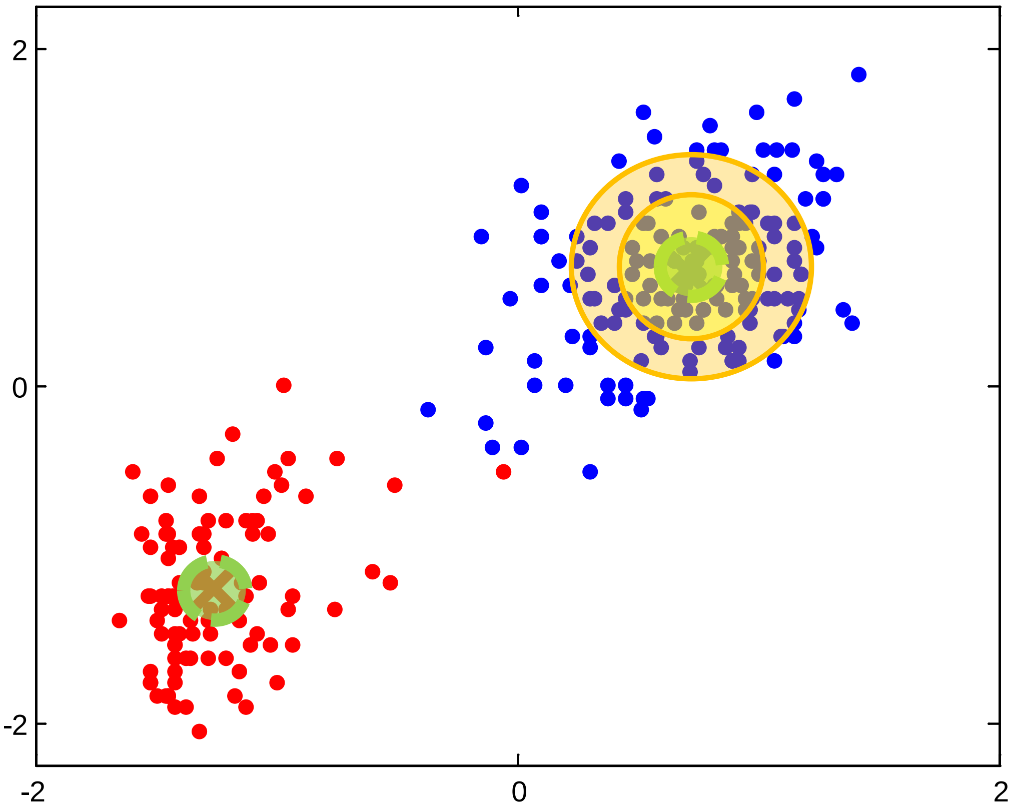
- There is a cyclic dependency between the model parameters and the cluster assignments.
- Initially we guess the model parameters.
- Based on this guess we estimate new cluster assignments.
- Which in turn impacts the model parameters which are then re-estimated.

Dissecting the K-Means Algorithm



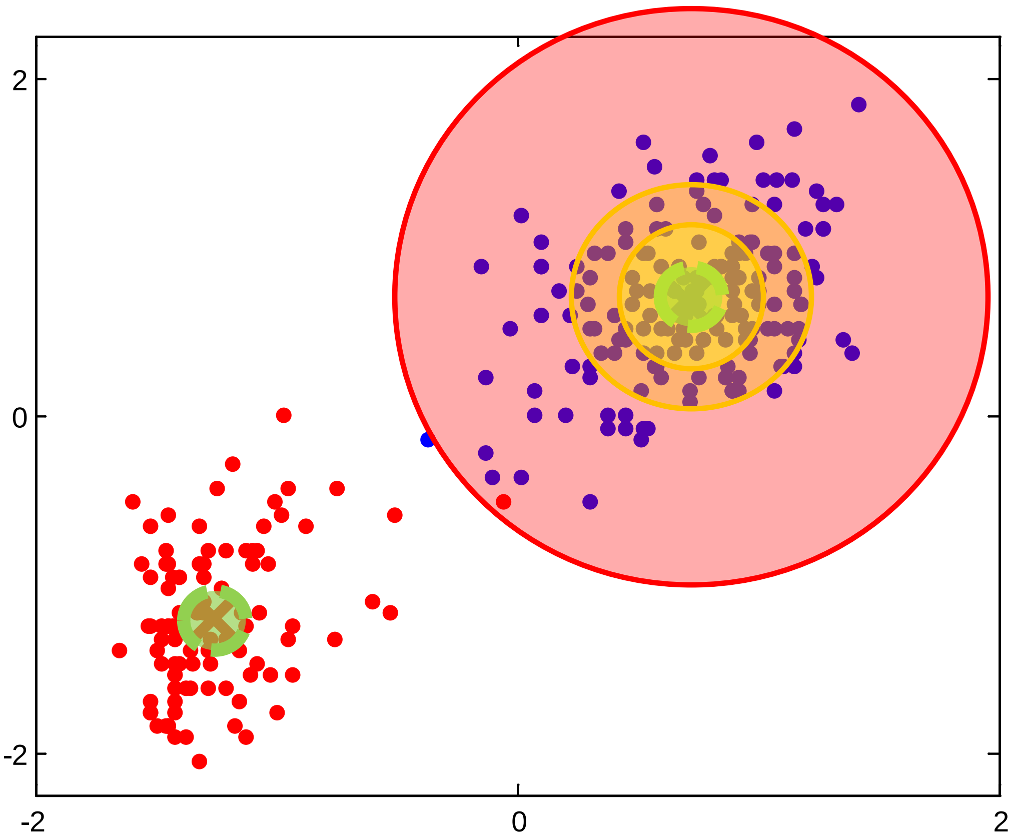
- There is a cyclic dependency between the model parameters and the cluster assignments.
- Initially we guess the model parameters.
- Based on this guess we estimate new cluster assignments.
- Which in turn impacts the model parameters which are then re-estimated.

Another Problem



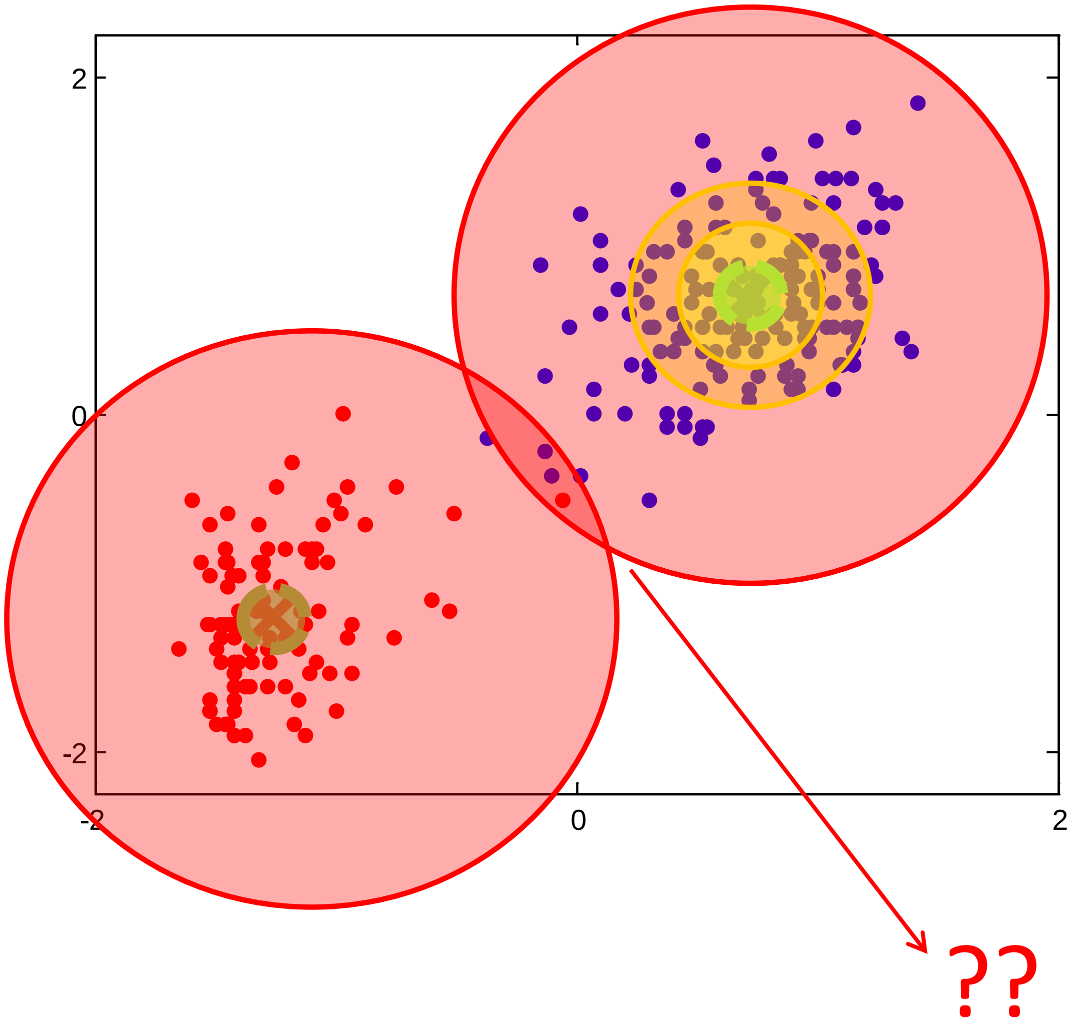
- K-Means makes hard guesses for cluster assignment.
- For some cases our model may not be sure about exact cluster assignment.
- Can we make this probabilistic so that z_{nk} defines the probability that the n^{th} observation belongs to the k^{th} cluster?

Another Problem



- K-Means makes hard guesses for cluster assignment.
- For some cases our model may not be sure about exact cluster assignment.
- Can we make this probabilistic so that z_{nk} defines the probability that the n^{th} observation belongs to the k^{th} cluster?

Another Problem



- K-Means makes hard guesses for cluster assignment.
- For some cases our model may not be sure about exact cluster assignment.
- Can we make this probabilistic so that z_{nk} defines the probability that the n^{th} observation belongs to the k^{th} cluster?

Probabilistic Clustering

- Lets place a Gaussian centered at each of the means discovered by K-Means (assume we know the covariance).

- Since we have run the k-means algorithm we have access to complete data i.e. $y = \{x_{1..n}, z_{1..n}\}$

- The probability of the complete data is:

$$P(X, Z|\theta) = \prod_{n} \prod_{k} \{\pi_k \cdot \mathcal{N}(\mu_k, \Sigma_k)\}^{z_{nk}}$$

Complete Data Likelihood

Probabilistic Clustering

- We don't know the value of ***Z*** for our data, they are missing/hidden/latent. Need to get rid of ***Z*** to calculate the data likelihood:

$$P(X|\theta) = \sum_Z P(X,Z|\theta)$$ (Marginalize it out)

- Lets see what happens to our complete data likelihood when we marginalize out ***Z***.

$$\sum_{z_i} P(x_i, z_i|\theta) = \sum_{z_i} \left\{ \prod_k \{\pi_k N(x_i|\mu_k, \Sigma_k)\}^{z_{ik}} \right\}$$

$$P(x_i|\theta) = \sum_k \pi_k N(x_i|\mu_k, \Sigma_k)$$

Gaussian Mixture Model

- Data generated from a mixture distribution:
  - $P(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k)$
  - Linear superposition of ***k*** Gaussians.
  - Added constraints:
    - $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$ (Multinomial Distribution).

- Generating Data:
  - Pick one of the Gaussian randomly with probability π_k.
  - Sample the value from the Gaussian centered at μ_k.

- Parameters of GMM:
  - $\theta = \{\pi_{1..k}, \mu_{1..k}, \Sigma_{1..k}\}$.

Estimating the Parameters

- We want to estimate our model parameters such that the probability of the data being generated by the model is maximized.

$$\theta = \arg\max_{\theta} P(X|\theta)$$

  which is equivalent to:

$$\theta = \arg\max_{\theta} \log(P(X|\theta))$$

- Lets apply this to our incomplete-data likelihood:

$$P(X|\theta) = \prod_{n} \left\{ \sum_{k} \pi_k N(x_n|\mu_k, \Sigma_k) \right\}$$

$$\log(P(X|\theta)) = \sum_{n} \log \left\{ \sum_{k} \pi_k N(x_n|\mu_k, \Sigma_k) \right\}$$

STUCK!!

Estimating the Parameters

- Make it a bit simpler, assume we know ***Z***. Now we can maximize the complete data log likelihood and estimate the model parameters.

$$P(X, Z|\theta) = \prod_{n} \prod_{k} \{\pi_k \cdot \mathcal{N}(\mu_k, \Sigma_k)\}^{z_{nk}}$$

$$\log\big(P(X, Z|\theta)\big) = \sum_{n} \sum_{k} z_{nk} \{\log(\pi_k) + \log(\mathcal{N}(x_n|\mu_k, \Sigma_k))\}$$

- Much more easier to work with, the parameters are decoupled and we can maximize easily.

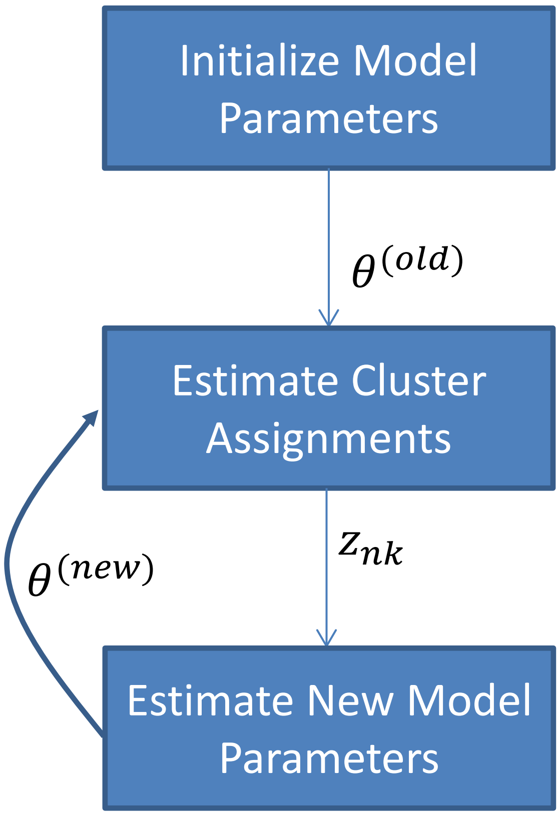
Estimating the Parameters

- If we maximize the complete data log likelihood we get the following estimates:

- $\mu_k = \frac{\sum_n z_{nk} x_n}{\sum_n z_{nk}} = \frac{\sum_n z_{nk} x_n}{N_k}$
- $\Sigma_k = \frac{1}{N_k} \sum_n z_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$
- $\pi_k = \frac{\sum_n z_{nk}}{N}$

- Are we done??
- What about the z_{nk}? we assumed they are known, but they are not!

What about z_{nk}?



- Recall, the game we played while using k-means.
- Guess the parameters, estimate the z_{nk}!!
- Fixing the parameters to some values, we now get a distribution over the missing Z i.e. $P(Z|X,\theta)$.
- OK! But this is a distribution, how do I get individual values for z_{nk}?

What about z_{nk}?

- Lets evaluate the expected value of each z_{nk}, under $P(Z|X,\theta)$.

$$\mathbb{E}_{P(Z|X,\theta)}[z_{nk}] = 1 \times P(z_{nk} = 1|x_n, \theta_k) + 0 \times P(z_{nk} = 0|x_n, \theta_k)$$
$$= P(z_{nk} = 1|x_n, \theta_k).$$

- Using Bayes Theorem we have:

Probability that the k-th component was chosen to generate x_n

$$P(z_{nk} = 1|x_n, \theta_k) = \frac{P(x_n|z_{nk} = 1, \theta_k) \cdot P(z_{nk} = 1|\theta_k)}{P(x_n|\theta_k)}$$

Probability of generating x_n using the k-th component.

Incomplete Data Likelihood for x_n

What about z_{nk}?

- So,

$$\mathbb{E}_{P(Z|X,\theta)}[z_{nk}] = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n|\mu_j, \Sigma_j)}$$
$$= \gamma(z_{nk})$$

- This quantity can be viewed as the "responsibility" that the k^{th} component takes for "explaining" the observation x_n.

- Finally, we can substitute this value for z_{nk} in our parameter estimates as our best guesses for the values of z_{nk} given our current model parameters.

EM for GMM based clustering

1. Initialize the model parameters $\theta^{(0)}$
2. **E-Step:** Evaluate the responsibilities using current parameter estimates:

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n|\mu_j, \Sigma_j)}$$

3. **M-Step:** Re-estimate the parameters using the current responsibilities:

- $\mu'_k = \frac{\sum_n \gamma(z_{nk}) x_n}{N_k}$
- $\Sigma'_k = \frac{1}{N_k} \sum_n \gamma(z_{nk})(x_n - \mu'_k)(x_n - \mu'_k)^T$
- $\pi'_k = \frac{\sum_n \gamma(z_{nk})}{N}$

where, $N_k = \sum_n \gamma(z_{nk})$.

4. If convergence criterion is not satisfied go back to step-2.

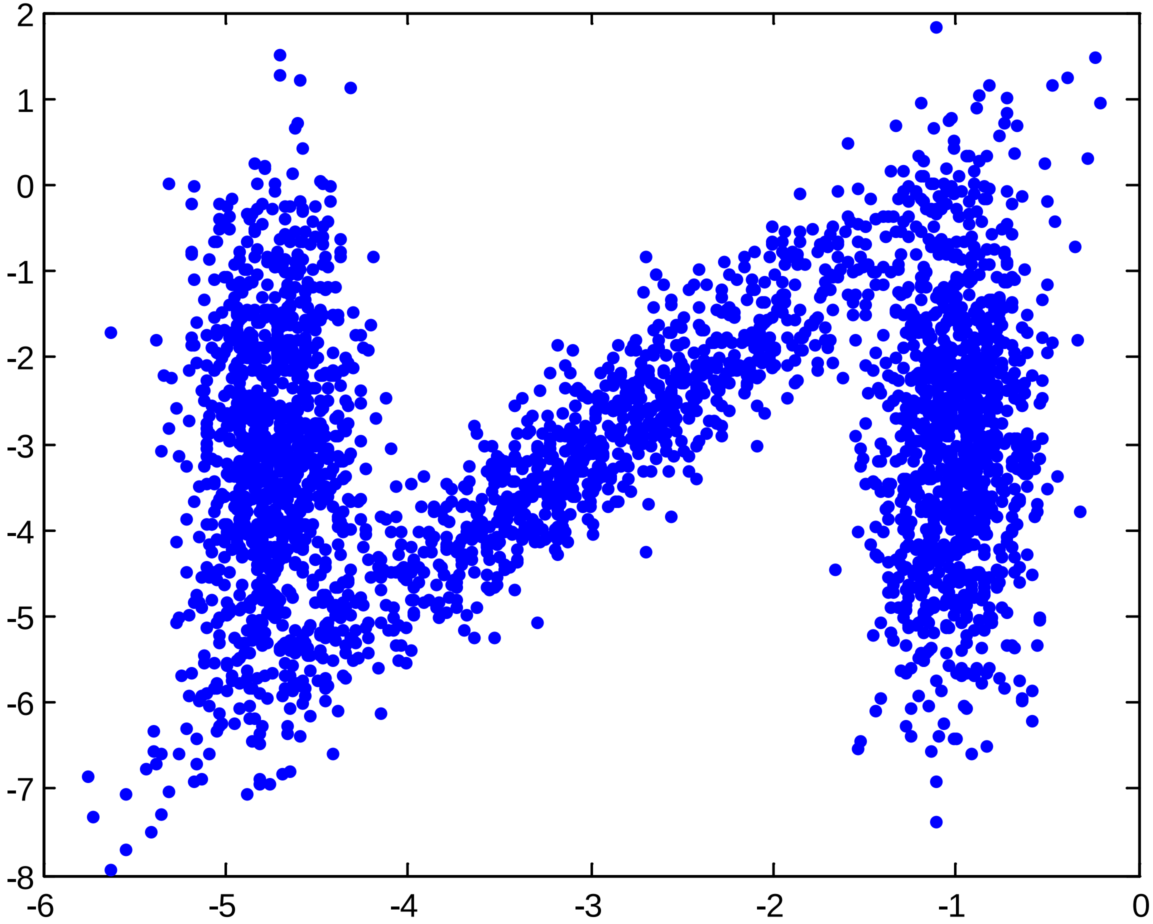
EM for GMM based clustering

- Convergence Criterion:
  - Check for the change in the values of the parameters.
  - Calculate the incomplete data log likelihood:

$$\log(P(X|\theta)) = \sum_{n} \log\left\{\sum_{k} \pi_k N(x_n|\mu_k, \Sigma_k)\right\}$$

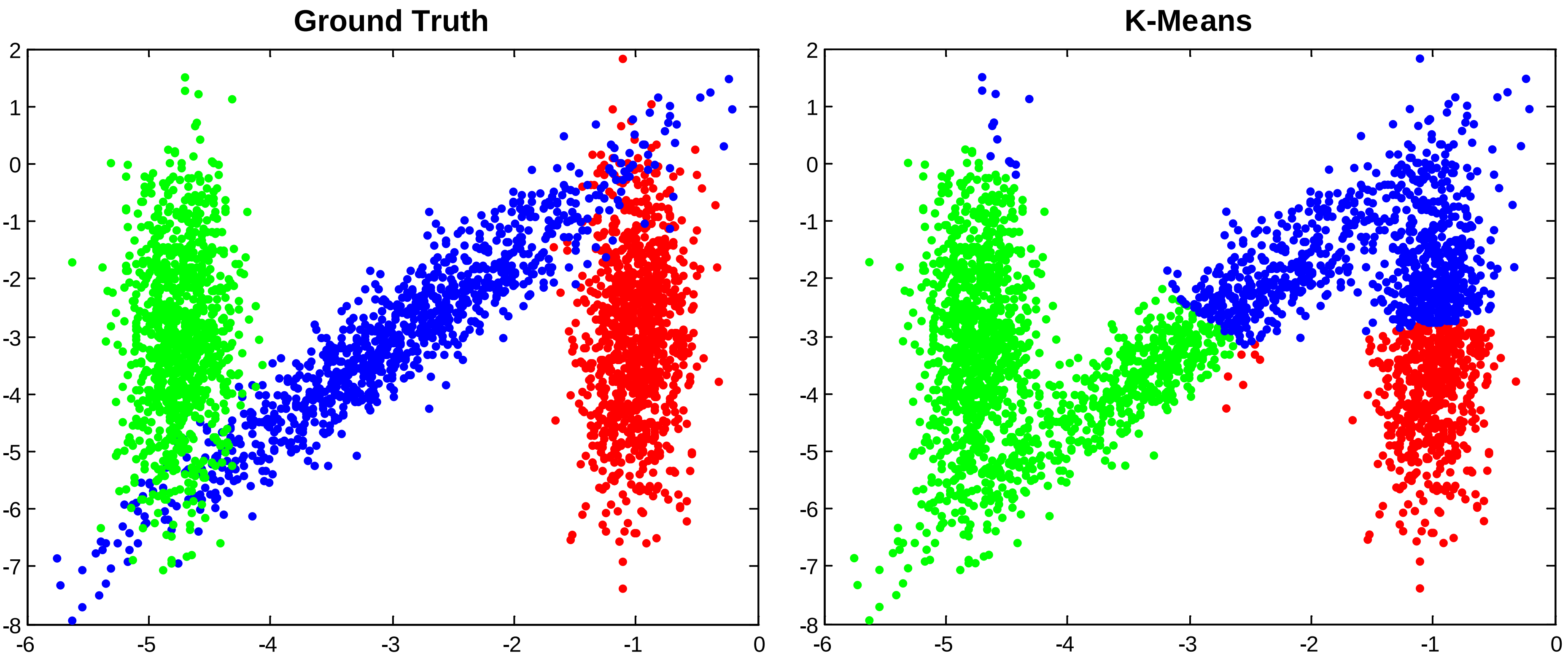
and if the value on current iteration has not changed from the previous value, or the change is negligible (below a preset tolerance), stop.

Example



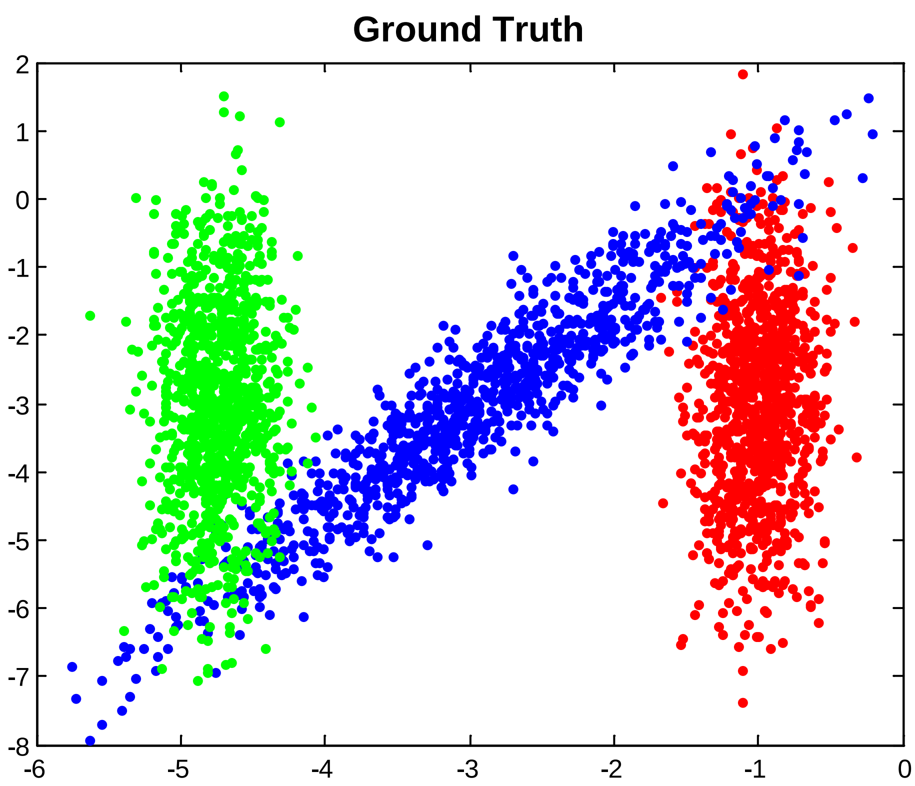
- Number of Clusters??
- Data sampled from three Gaussians centered at:
  - [-1,-3]
  - [-3,-3]
  - [-4.75,-3]

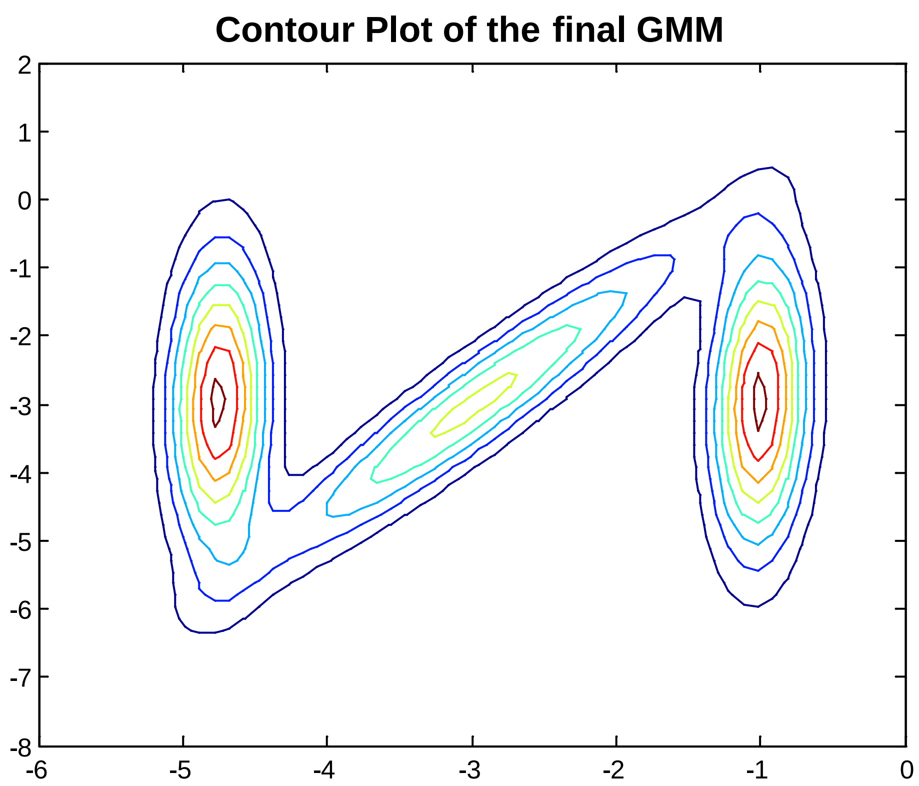
Example



- Run K-Means with 50 random starting points. Select the solution that has the minimum sum of squared distances.

Example





- Soft Clustering using a three component Gaussian Mixture Model with random starting point.

Example

- Original Means:
  - [-1,-3]
  - [-3,-3]
  - [-4.75,-3]
- K-Means Centroids:
  - [-1.0335,-4.057]
  - [-1.5821, -1.6458]
  - [-4.3681, -3.4009]
- Means of the Three Gaussians Discovered by GMM:
  - [-1.0006, -2.9663]
  - [-2.9747, -2.9921]
  - [-4.7488, -2.9717]

EM Algorithm

- A very powerful method for dealing with probabilistic models that involve latent/missing variables.
- Each iteration of the EM is guaranteed to maximize the data log likelihood.
- Guaranteed to converge to a local maxima.
- Sensitive to starting points.
- We have applied it to Gaussian Mixture Models, which can model any arbitrary shaped densities. Can be used for data density estimation aside from clustering.