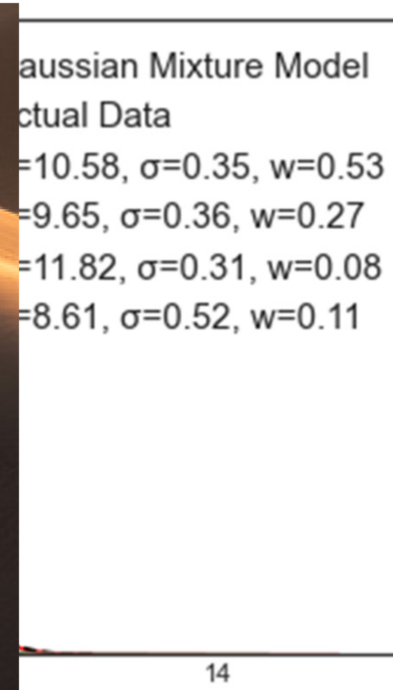
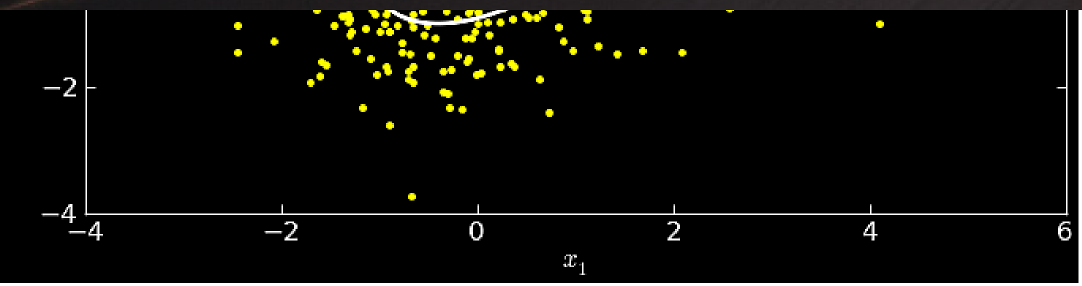
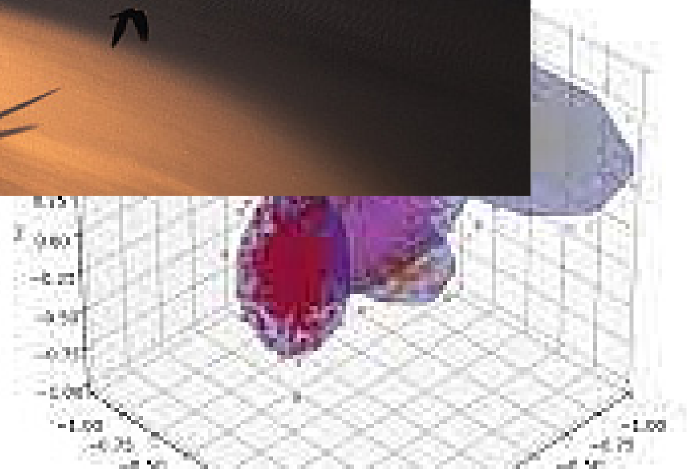
Gaussian Mixture Model (GMM) using Expectation Maximization (EM) Technique

Book : C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006


aussian Mixture Model
ctual Data
=10.58, σ=0.35, w=0.53
=9.65, σ=0.36, w=0.27
=11.82, σ=0.31, w=0.08
=8.61, σ=0.52, w=0.11
14


−2
−4
−4
−2
0
2
4
6
x_1



The Gaussian Distribution

- Univariate Gaussian Distribution

$$G(\mathrm{x} \mid \underbrace{\mu}_{\text{mean}}, \underbrace{\sigma}_{\text{variance}}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Multi-Variate Gaussian Distribution

$$\mathcal{N}(\mathbf{x} \mid \underbrace{\boldsymbol{\mu}}_{\text{mean}}, \underbrace{\Sigma}_{\text{covariance}}) = \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})\right\}$$

We need to estimate these parameters (Σ, μ) of a distribution:

One method – Maximum Likelihood (ML) Estimation.

$$\ln p\left(\mathbf{x}|\mu,\sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi).$$

Which is MAP, and which one MLE ??

$$= \arg\max_{\theta} \frac{f(x \mid \theta)\, g(\theta)}{\int_{\Theta} f(x \mid \vartheta)\, g(\vartheta)\, d\vartheta}$$

$$= \arg\max_{\theta} f(x \mid \theta)\, g(\theta).$$

ML Method for estimating parameters

- Consider log of Gaussian Distribution

$$\ln p(\mathbf{x} \mid \mu, \Sigma) = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)$$

- Take the derivative and equate it to zero

$$\frac{\partial \ln p(\mathbf{x} \mid \mu, \Sigma)}{\partial \mu} = 0 \qquad \frac{\partial \ln p(\mathbf{x} \mid \mu, \Sigma)}{\partial \Sigma} = 0$$

$$\Downarrow \qquad \Downarrow$$

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \qquad \Sigma_{ML} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{x}_n - \mu_{ML})(\mathbf{x}_n - \mu_{ML})^T$$

Where, N is the number of samples or data points

Gaussian Mixtures

- Linear super-position of Gaussians

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

Number of Gaussians

Mixing coefficient: weightage for each Gaussian dist.

- Normalization and positivity require:

$$\mathbf{0 \leq \pi_k \leq 1, \qquad \sum_{k=1}^{K} \pi_k = 1}$$

- Consider log-likelihood:

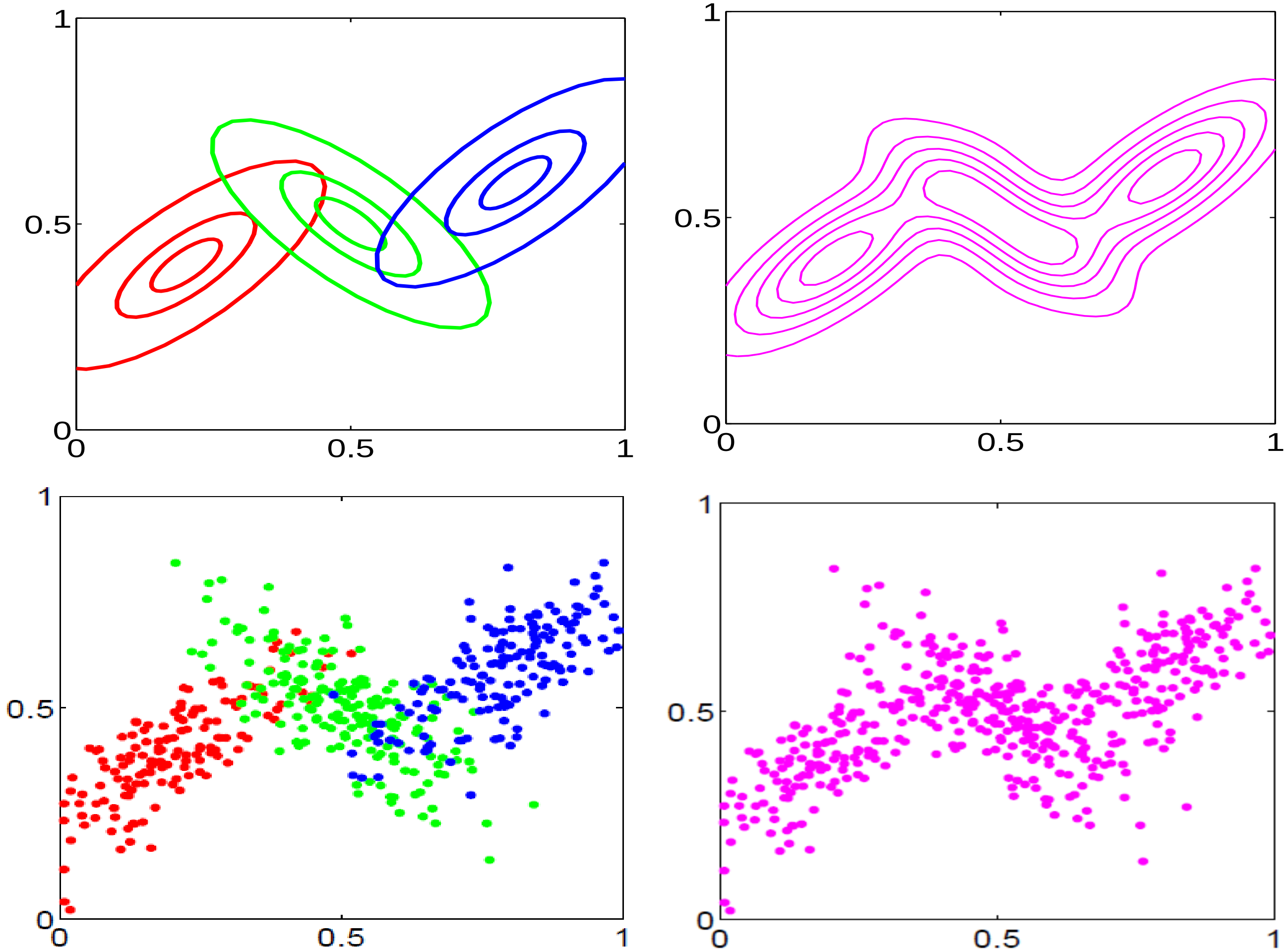
$$\ln p(X \mid \mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln p(x_n) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \right\}$$

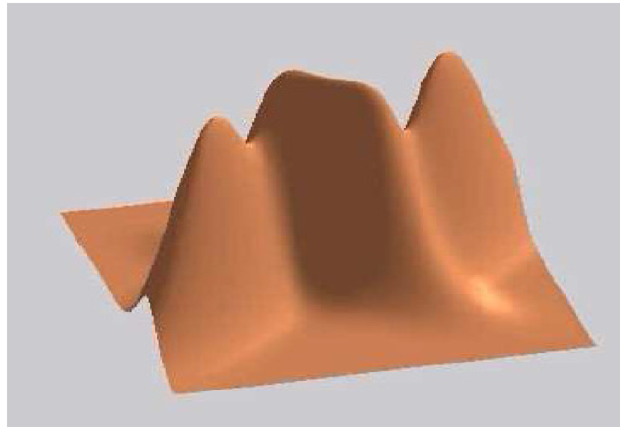
ML does not work here as there is no closed form solution

Parameters can be calculated using -

Expectation Maximization (EM) technique

Example: Mixture of 3 Gaussians





Latent variable: posterior prob.

- We can think of the mixing coefficients as prior probabilities for the components

- For a given value of ‘x’, we can evaluate the corresponding posterior probabilities, called responsibilities

- From Bayes rule

$$\gamma_k(\mathbf{x}) = p(k \mid \mathbf{x}) = \frac{p(k)p(\mathbf{x} \mid k)}{p(\mathbf{x})}$$

Latent Variable

$$= \frac{\pi_k \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x} \mid \mu_j, \Sigma_j)} \qquad \text{where,} \qquad \pi_k = \frac{N_k}{N}$$

Interpret N_k as the effective no. of points assigned to cluster k.

Expectation Maximization

- EM algorithm is an iterative optimization technique which is operated locally
- Estimation step: for given parameter values we can compute the expected values of the latent variable.
- Maximization step: updates the parameters of our model based on the latent variable calculated using ML method.

EM Algorithm for GMM

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters comprising the means and covariances of the components and the mixing coefficients.

1. Initialize the means $\boldsymbol{\mu}_j$, covariances Σ_j and mixing coefficients $\boldsymbol{\pi}_j$, and evaluate the initial value of the log likelihood.

2. E step. Evaluate the responsibilities using the current parameter values

$$\gamma_k(x) = \frac{\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x \mid \mu_j, \Sigma_j)}$$

EM Algorithm for GMM

3. M step. Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

$$\Sigma_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n)(x_n - \mu_j)(x_n - \mu_j)^T}{\sum_{n=1}^{N} \gamma_j(x_n)}$$

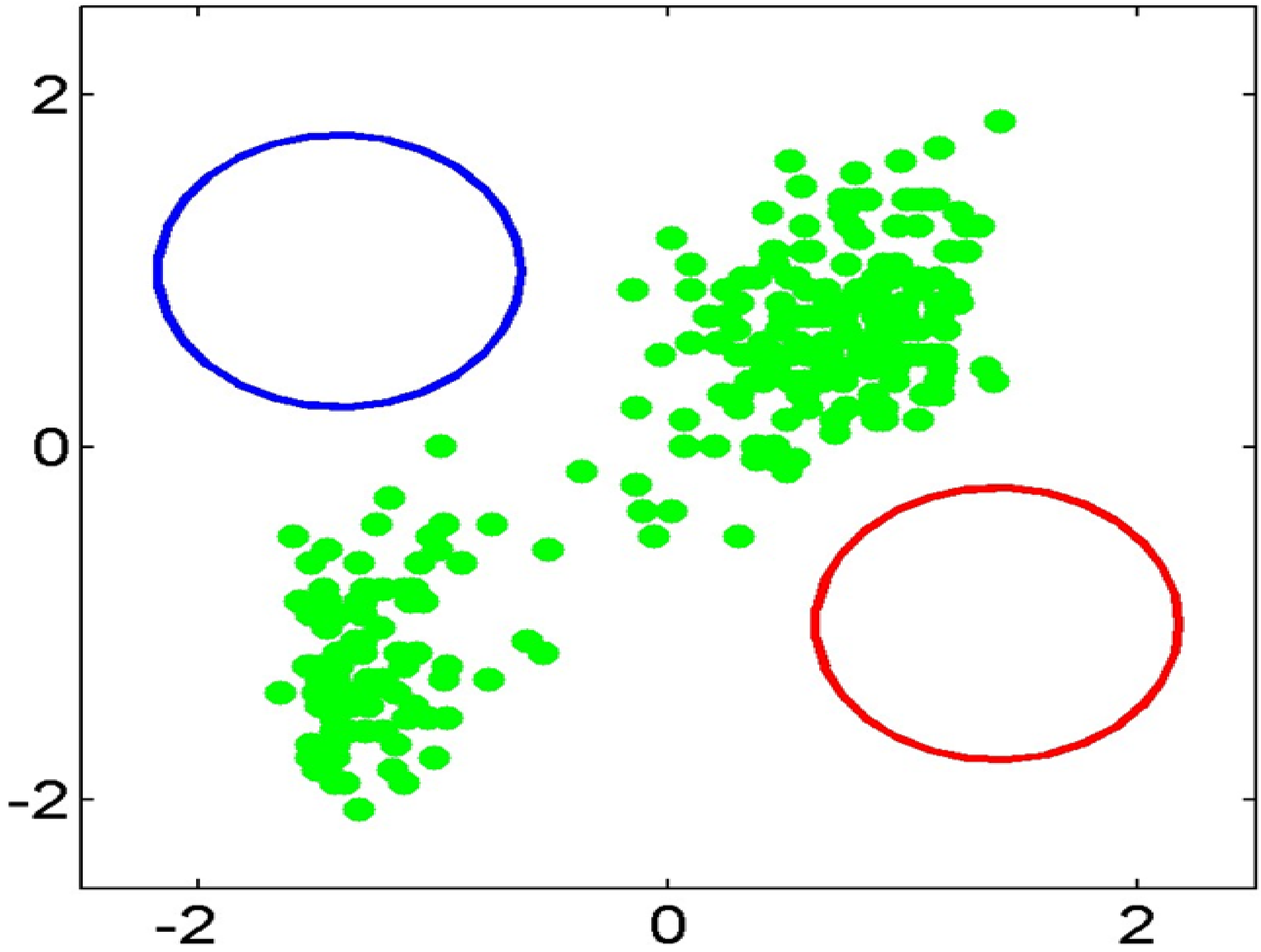
$$\boldsymbol{\pi}_j = \frac{1}{N}\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)$$

4. Evaluate log likelihood

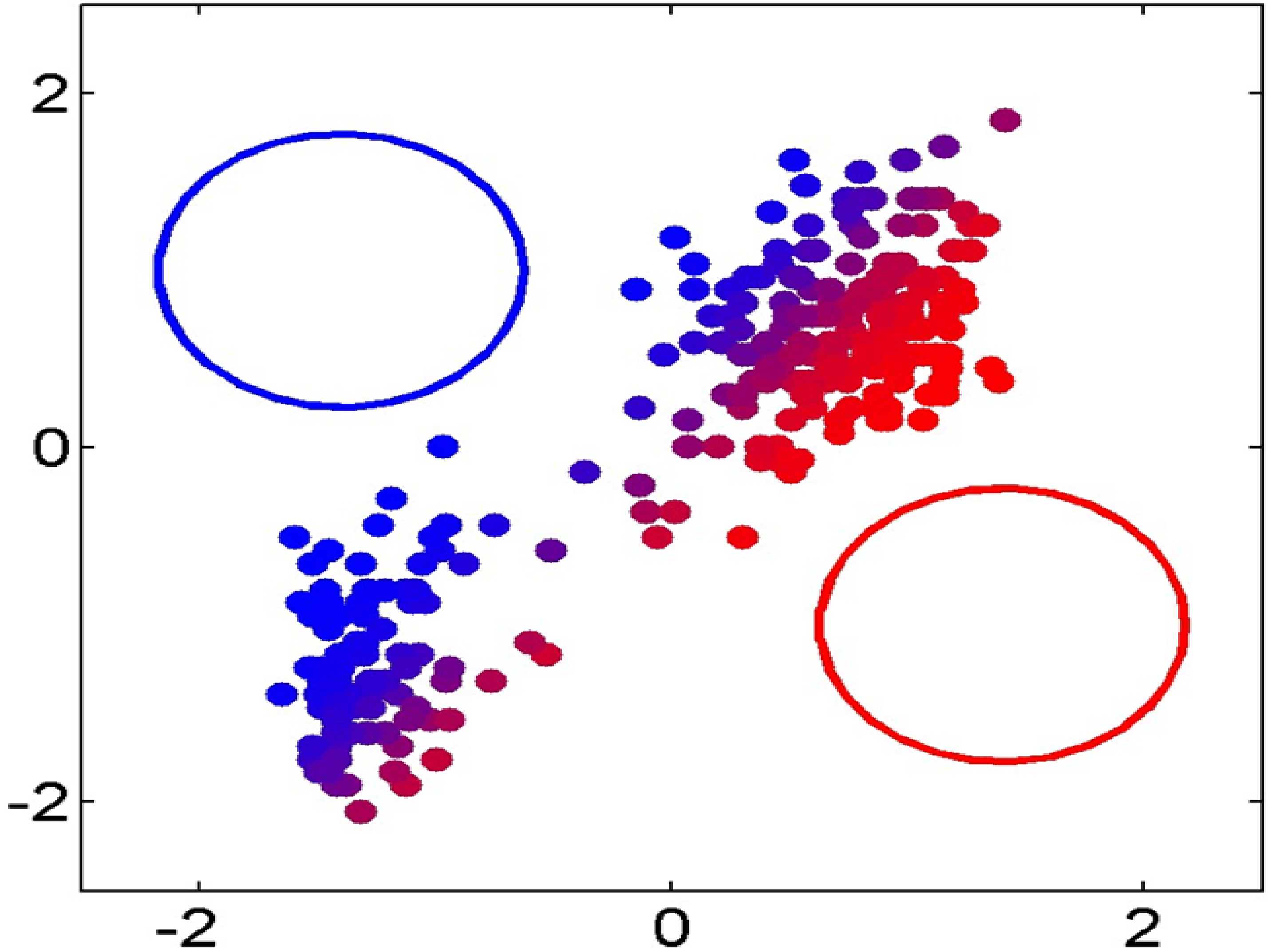
$$\ln p(\mathbf{X} \mid \boldsymbol{\mu}, \Sigma, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \boldsymbol{\pi}_k N(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k) \right\}$$

If there is no convergence, return to step 2.

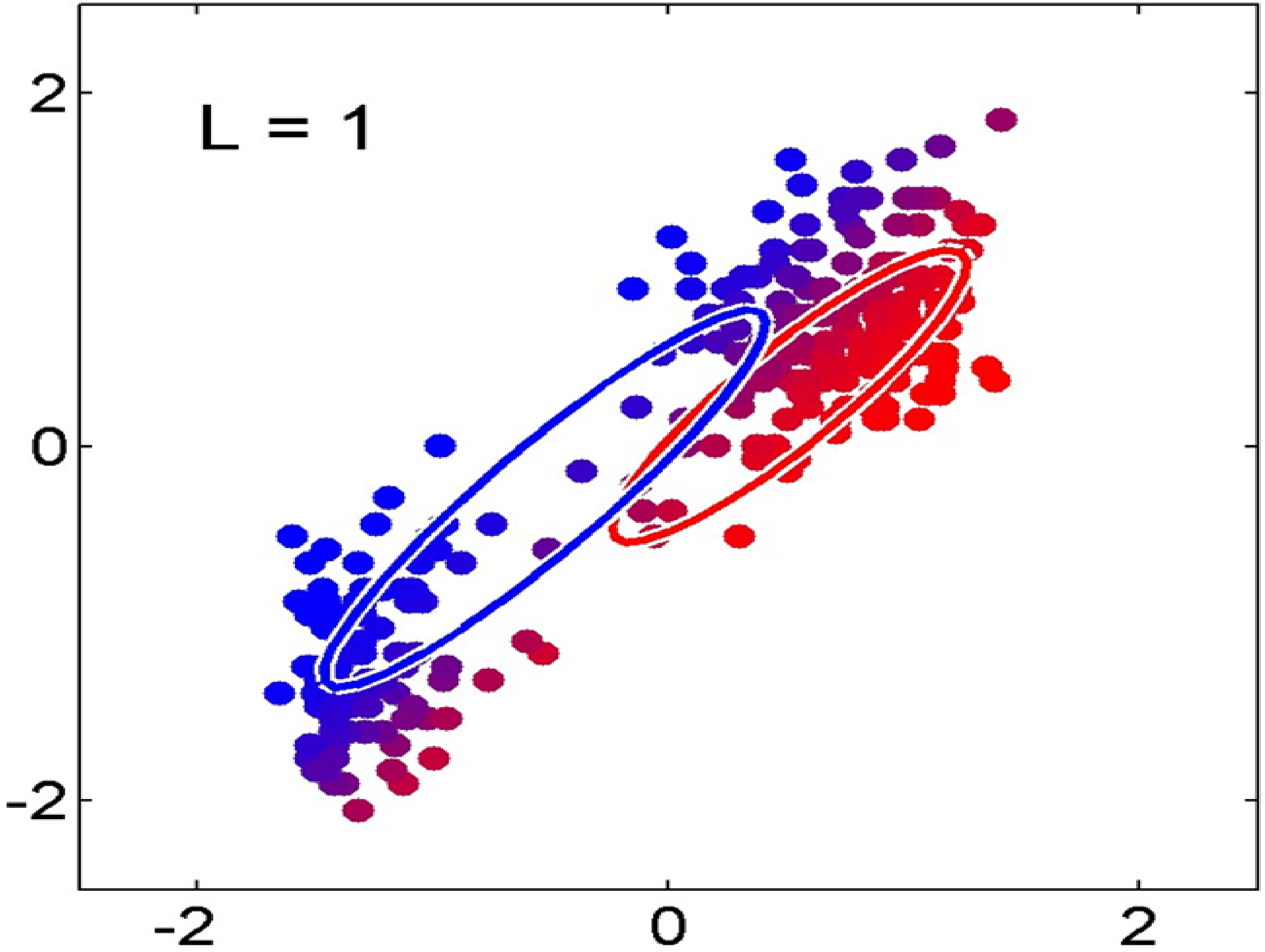
EM Algorithm : Example



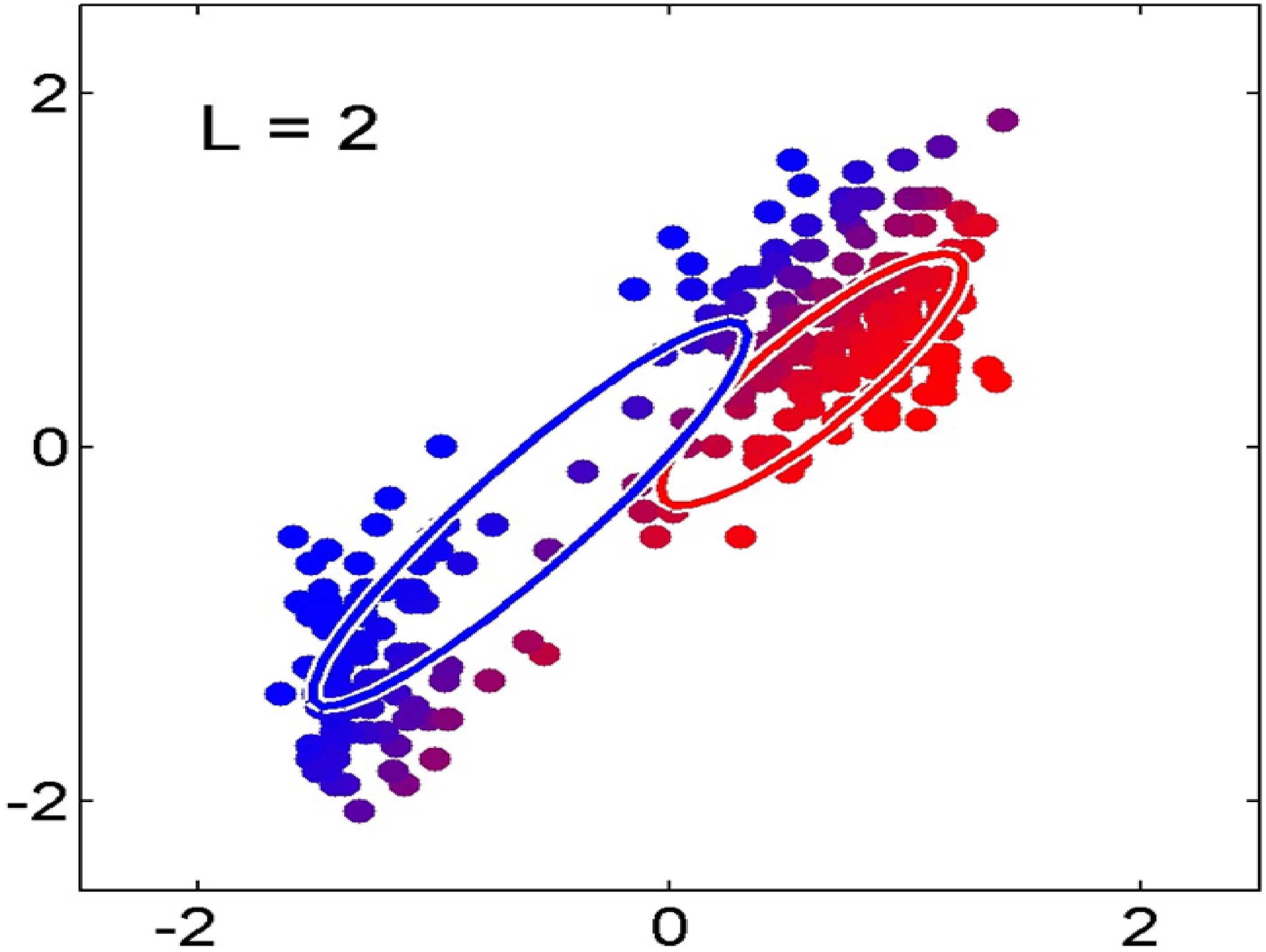
EM Algorithm : Example



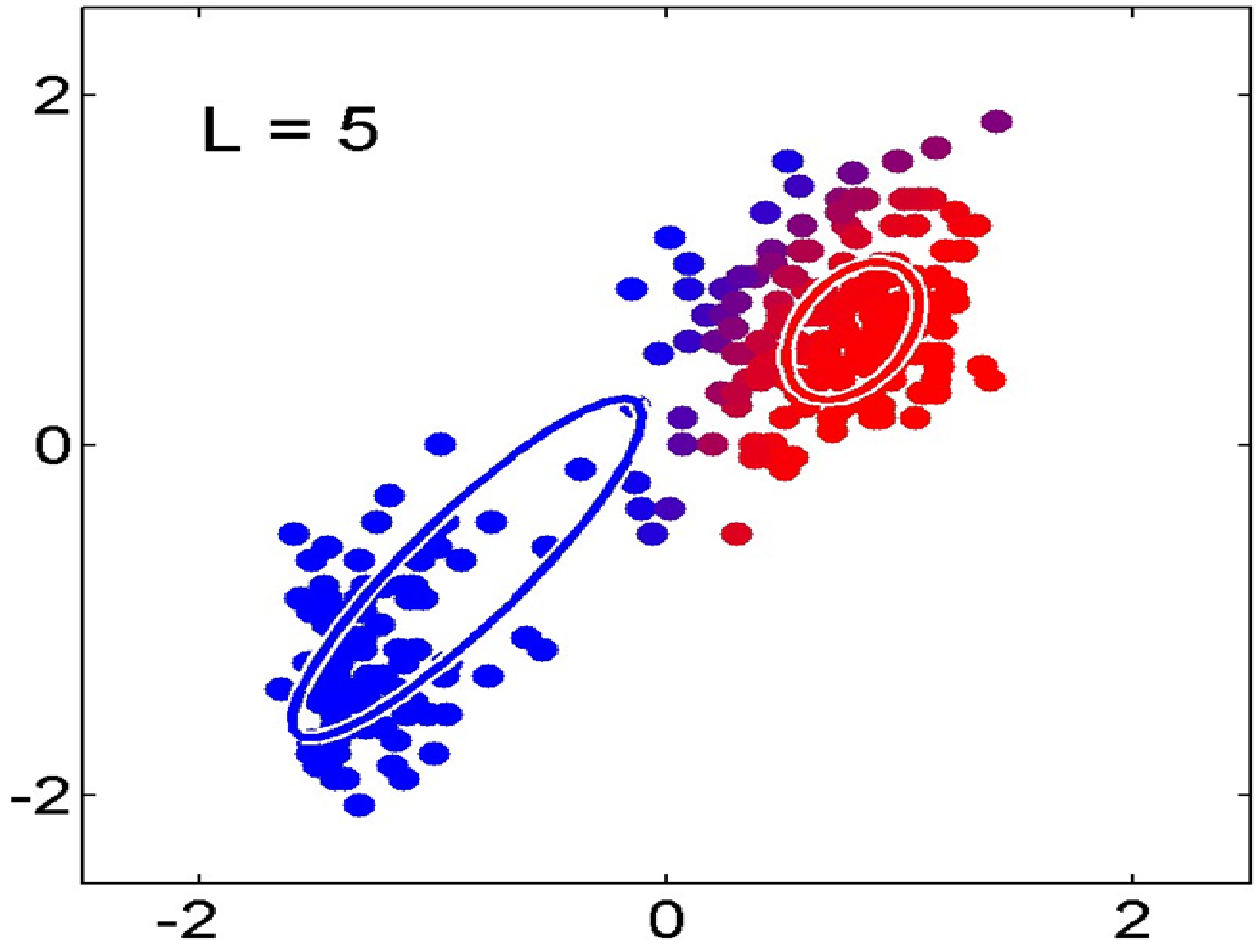
EM Algorithm : Example



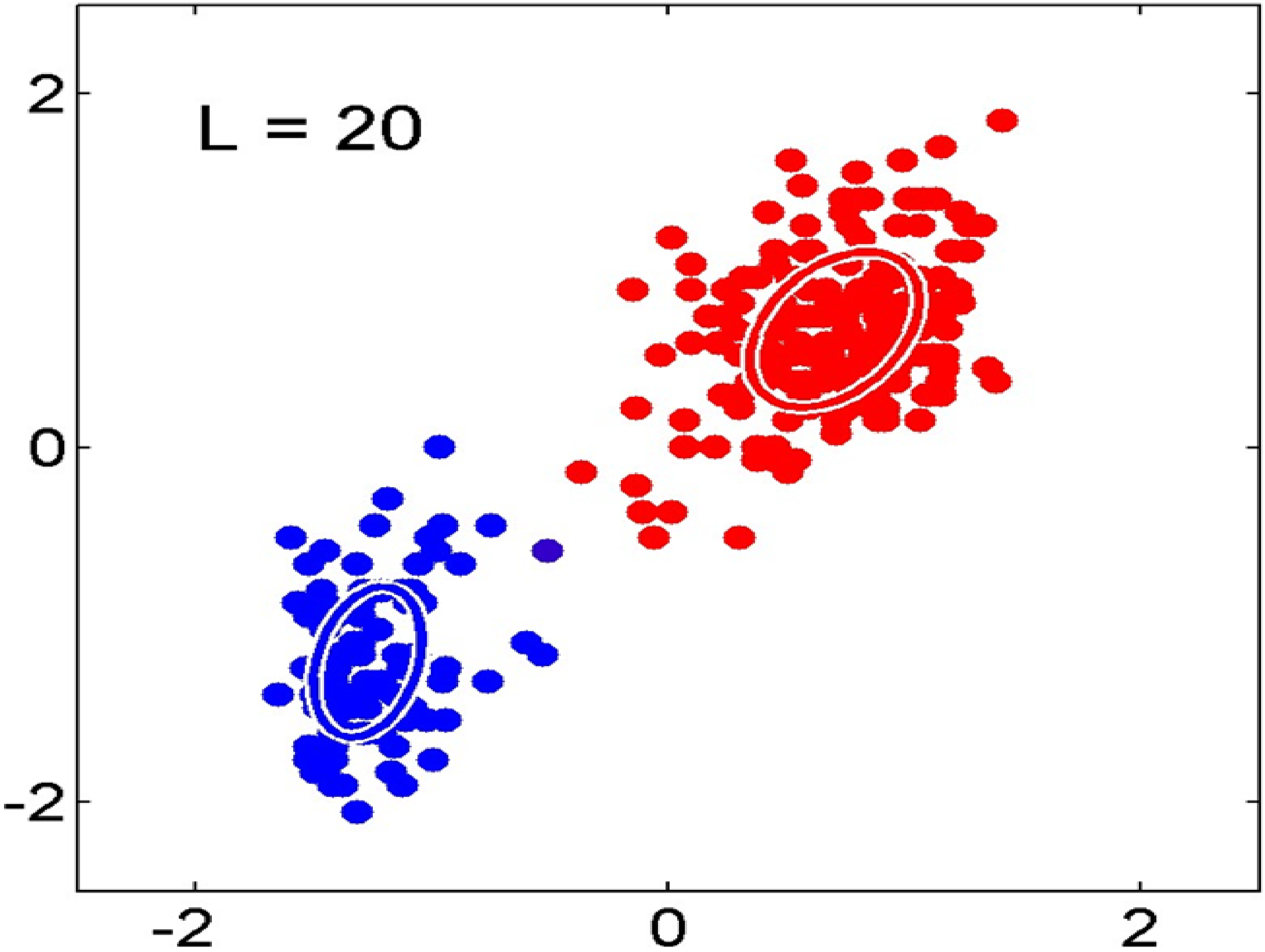
EM Algorithm : Example

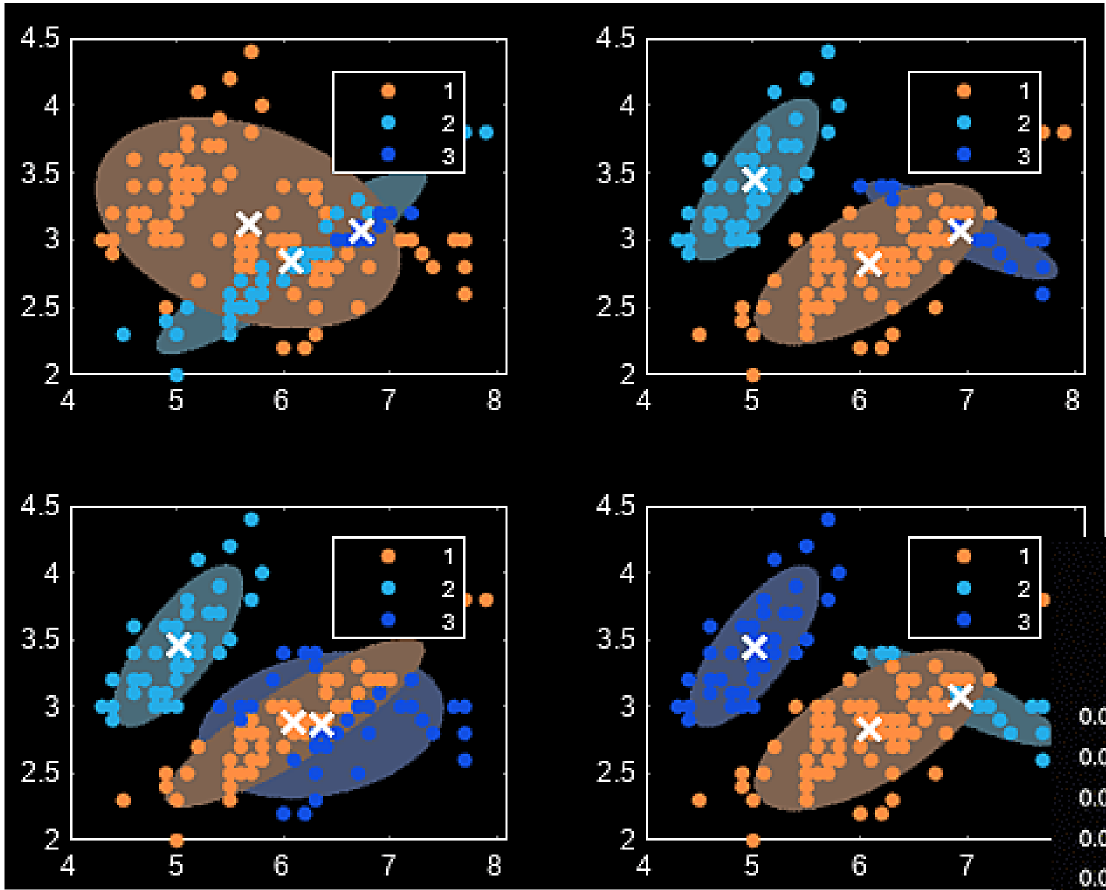


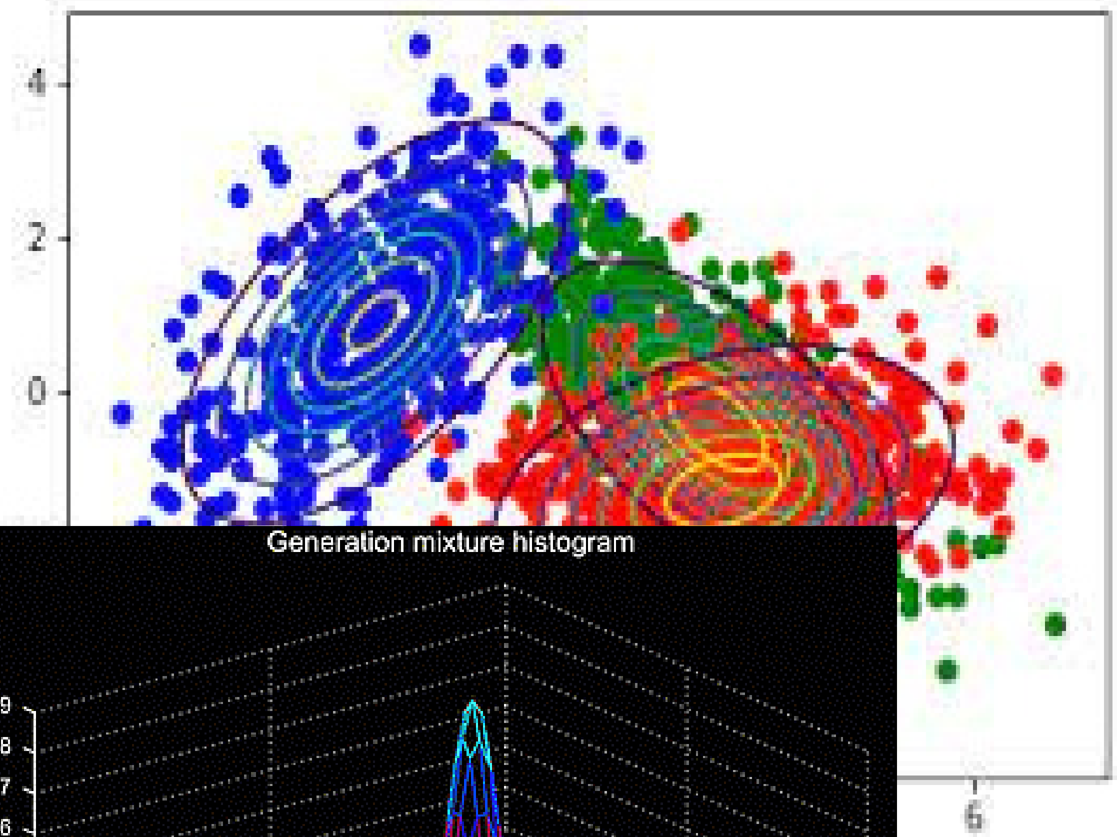
EM Algorithm : Example

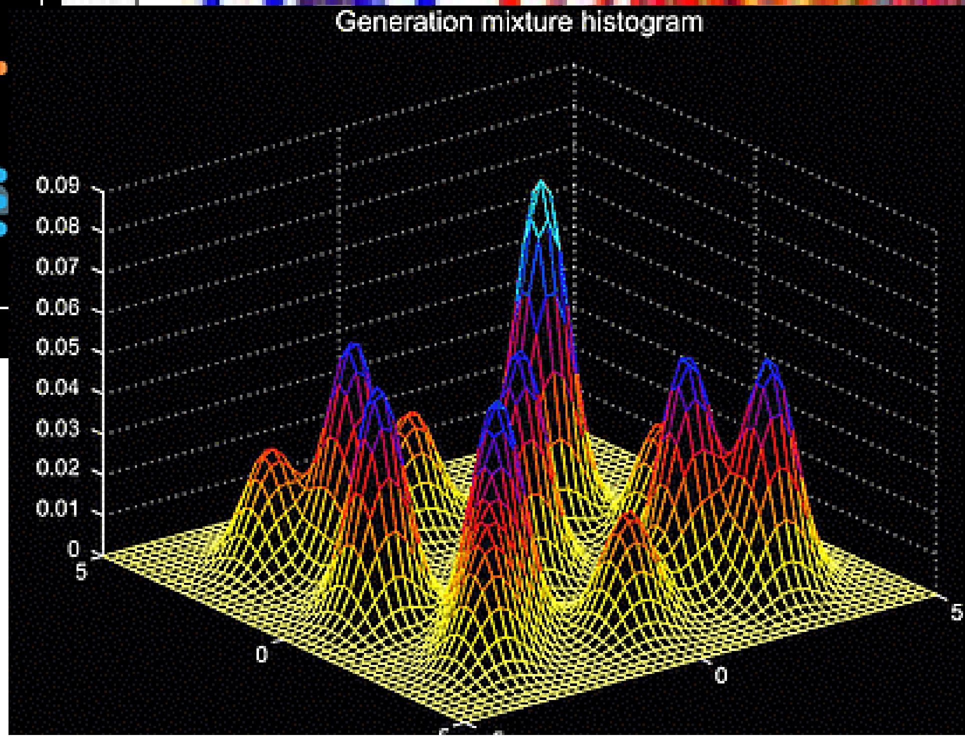


EM Algorithm : Example



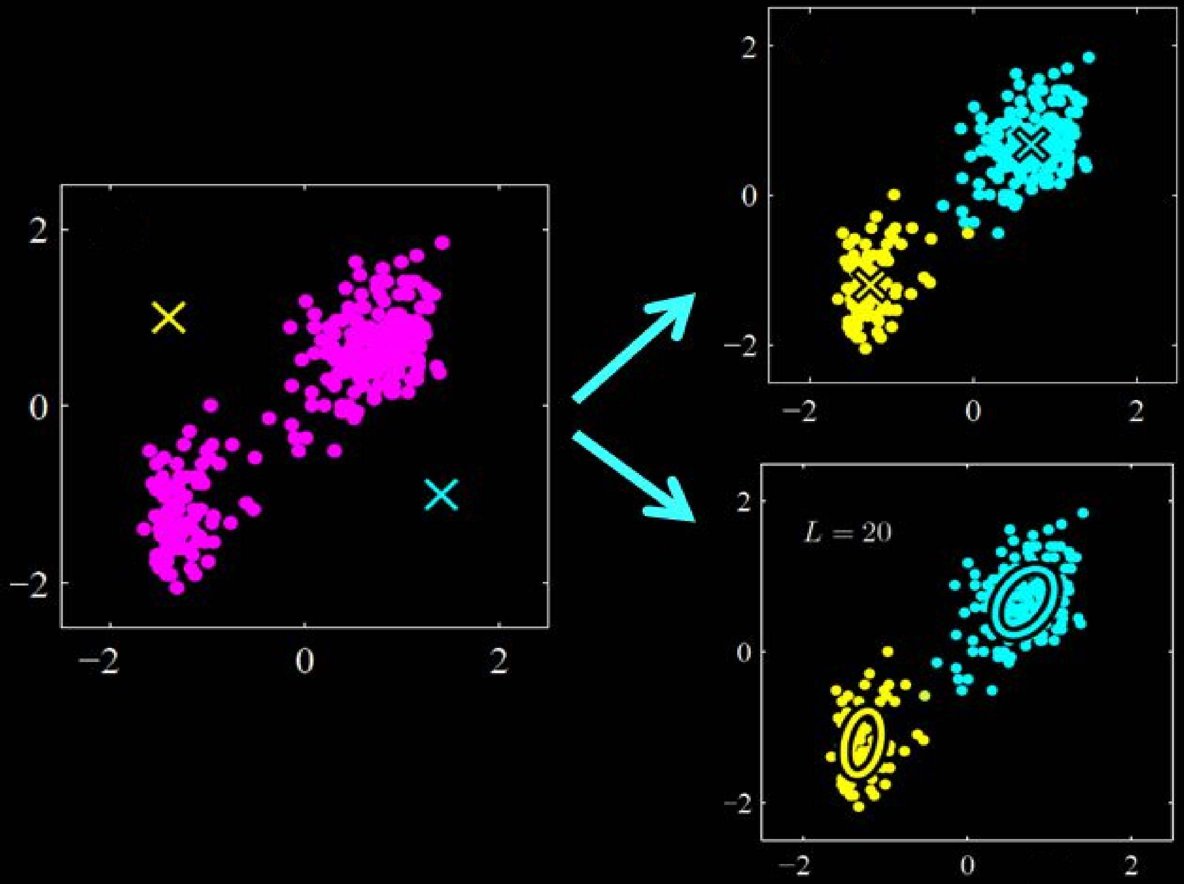

1
2
3




Generation mixture histogram

K-means vs GMM

Two representative techniques are k-means and Gaussian Mixture Model (GMM). K-means assigns data points to the nearest clusters, while GMM assigns data to the Gaussian densities that best represent the data.

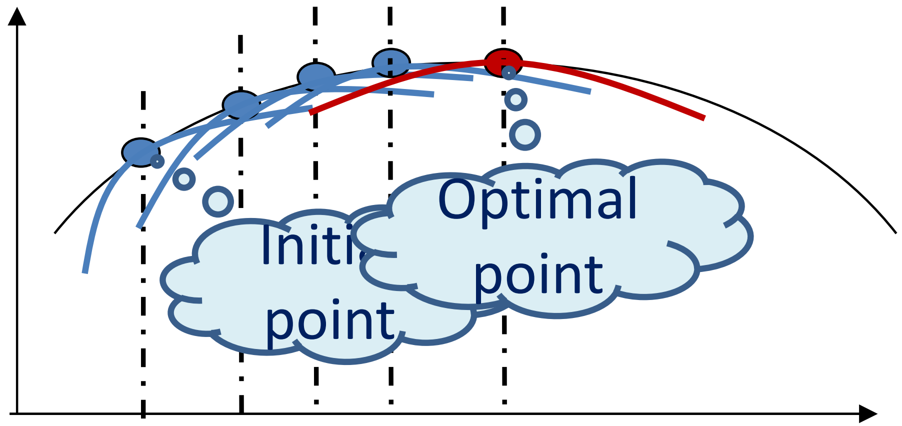


Hard clustering: a data point is assigned only one cluster.

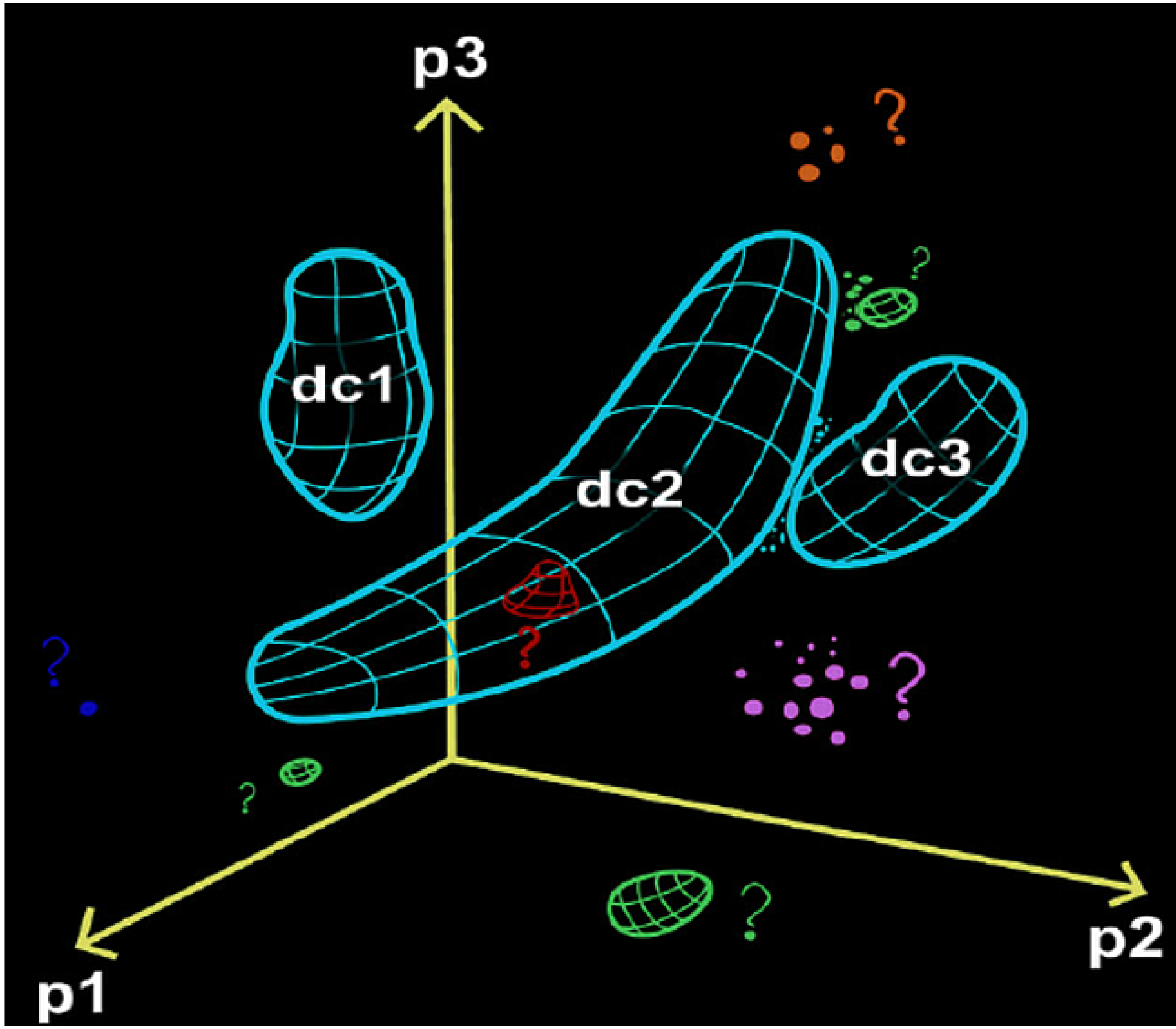
Soft clustering: a data point is assigned multiple Gaussians probabilistically.

Expectation Maximization

- EM algorithm is an iterative optimization technique which is operated locally



- Estimation step: for given parameter values we can compute the expected values of the latent variable.
- Maximization step: updates the parameters of our model based on the latent variable calculated using ML method.



Other Applications of Latent Variable:

- HMM, PGM, LDA (latent Dirichlet Allocation),
any mixture models (e.g. multi-variate Bernoulli);
Bayesian Learning with mixed graph models (DAG, G-DMG etc.)