Gaussian Mixture Models

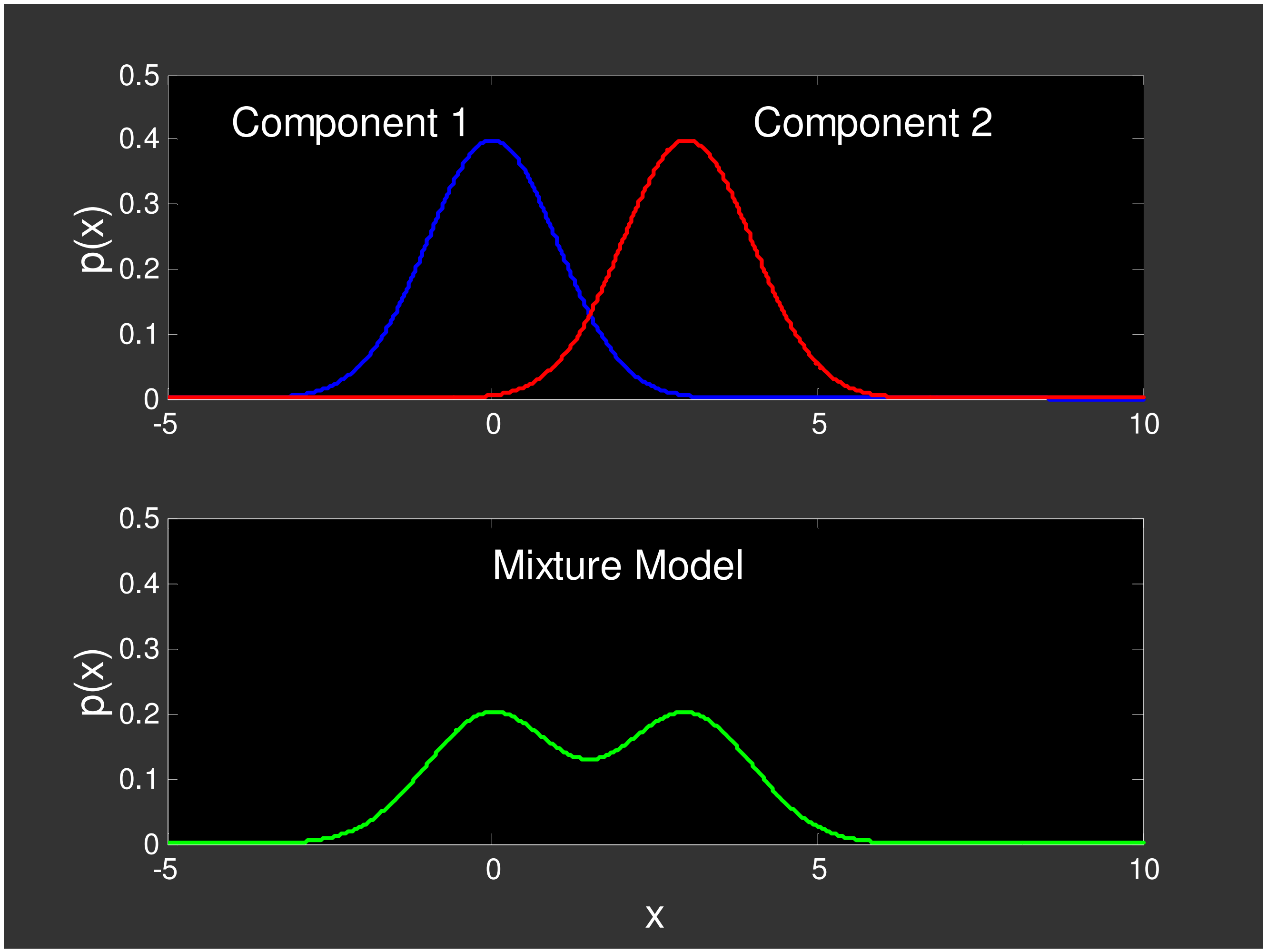
Mixture Models

$$p(\mathbf{x} \mid \omega) = \sum_{k=1}^{K} p(\mathbf{x}, c_k \mid \omega)$$

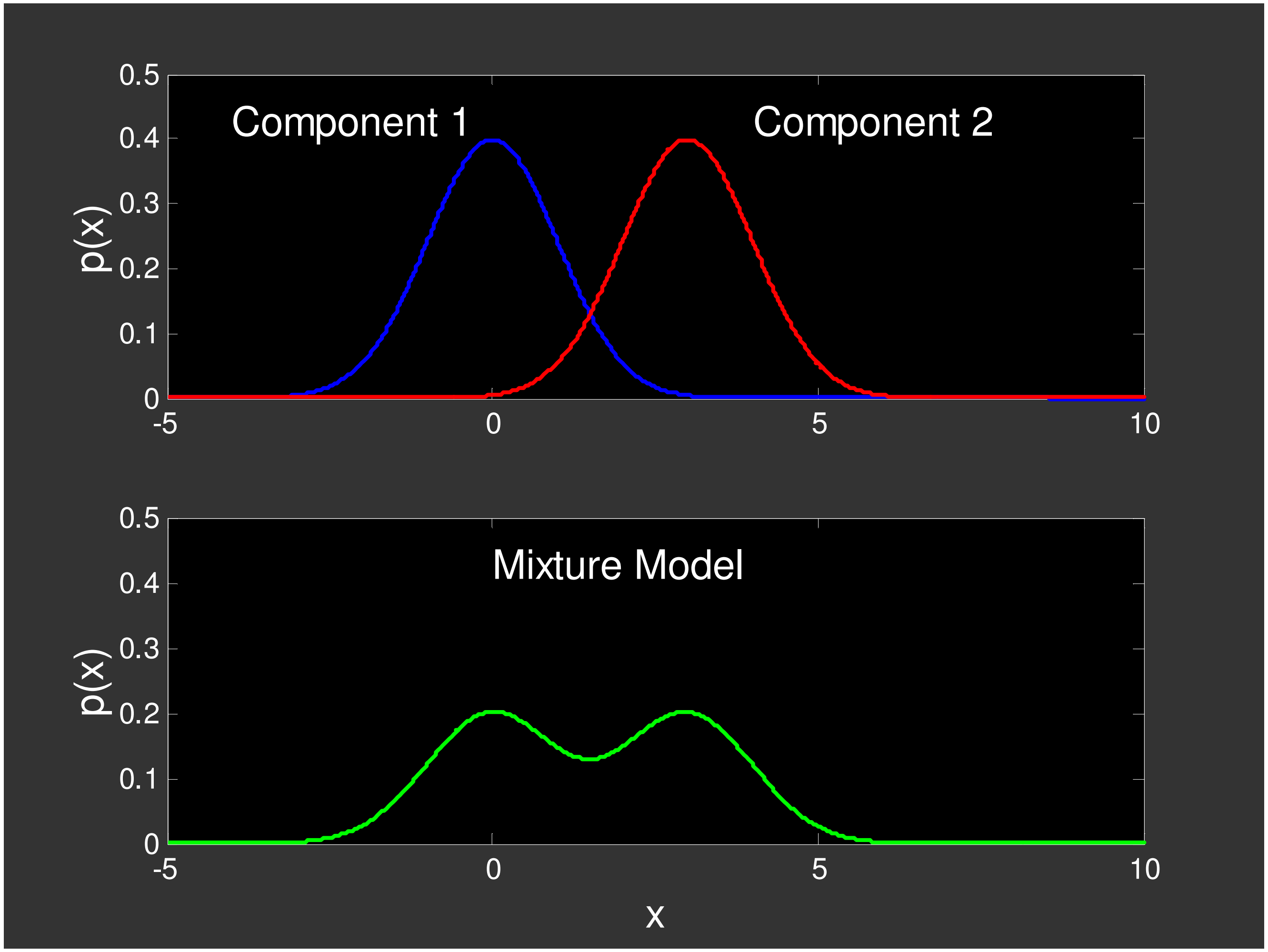
$$= \sum_{k=1}^{K} p(\mathbf{x} \mid c_k) p(c_k)$$

$$= \sum_{k=1}^{K} \pi_k \; p(\mathbf{x} \mid c_k, \boldsymbol{\theta}_k)$$

Gaussian Mixture Models



Gaussian Mixture Models



$$p(\mathbf{x} \mid \omega) = \sum_{k=1}^{K} \pi_k \; p(\mathbf{x} \mid c_k, \boldsymbol{\theta}_k)$$

$$= \sum_{k=1}^{K} \pi_k \; N(\mathbf{x} \mid \boldsymbol{\theta}_k)$$

$$= \sum_{k=1}^{K} \pi_k \; N(\mathbf{x} \mid \boldsymbol{\mu}_{k,} \Sigma_k)$$

Where $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_{k,} \Sigma_k)$ are parameters of component c_k,

_

Gaussian Mixture Models

Gaussian Mixture Model comprising K Gaussian

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

Log likelihood →
note that we have a sum of logarithm functions

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

Gaussian Mixture Models

- Linear super-position of Gaussians

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

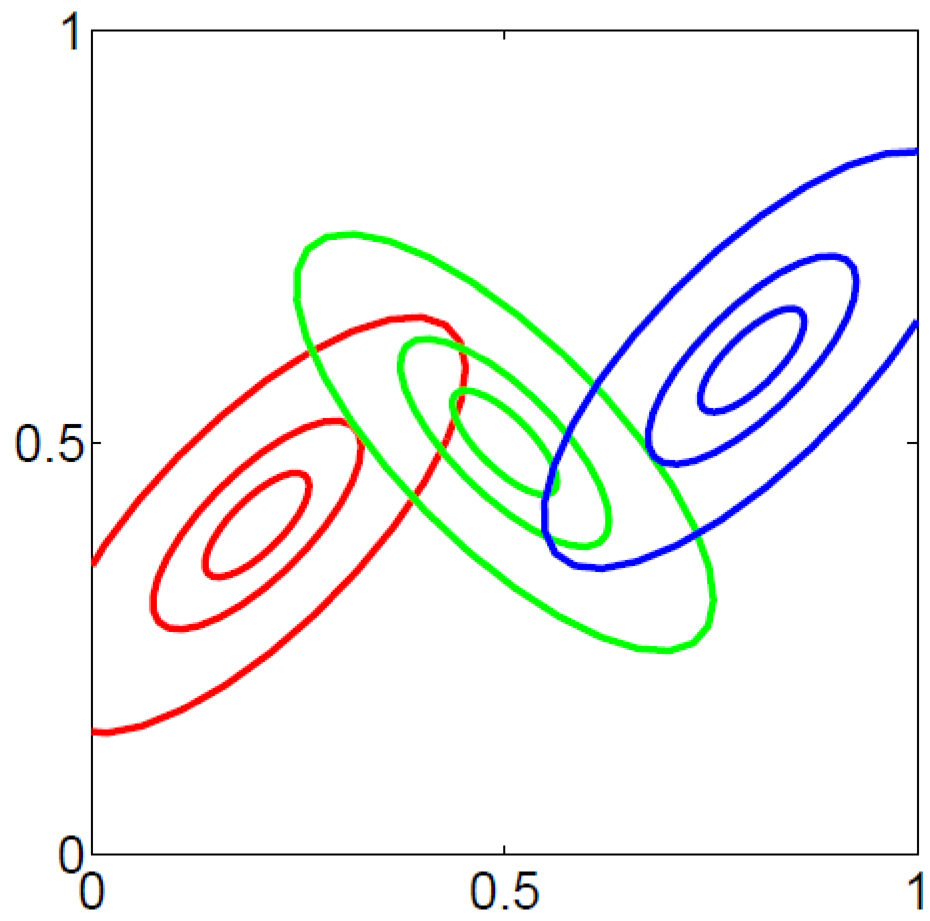
- Normalization and positivity require

$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leqslant \pi_k \leqslant 1$$

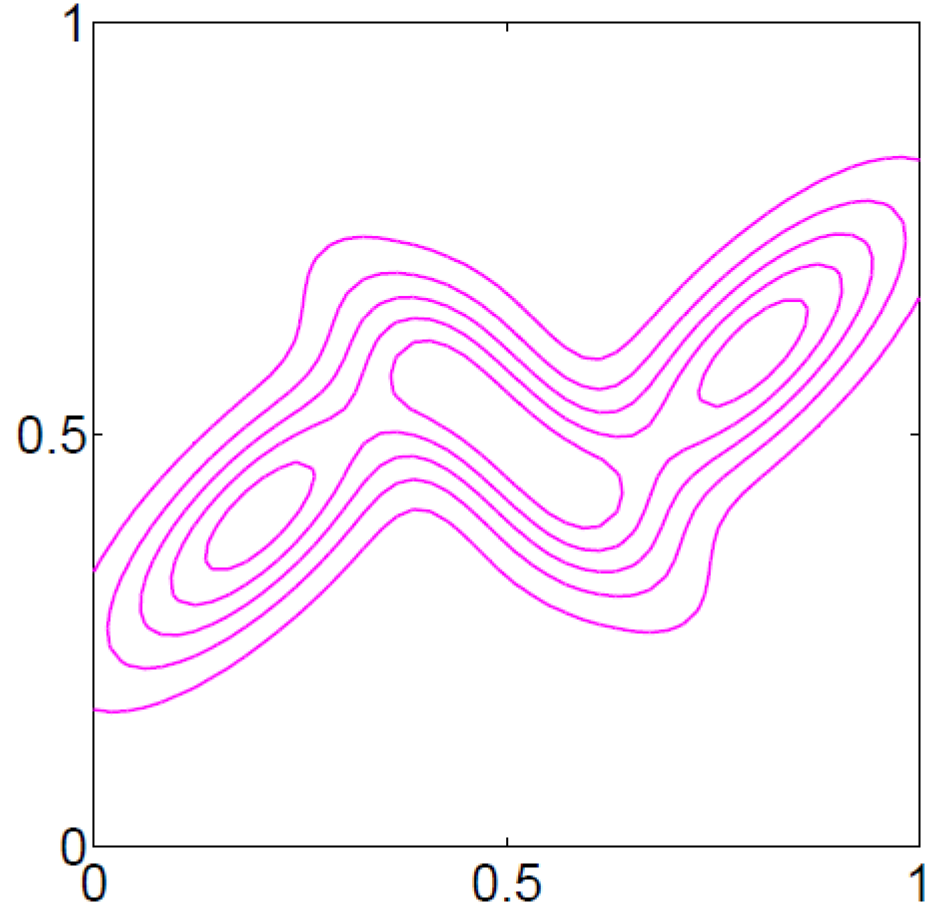
- Can interpret the mixing coefficients as prior probabilities

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k) p(\mathbf{x}|k)$$

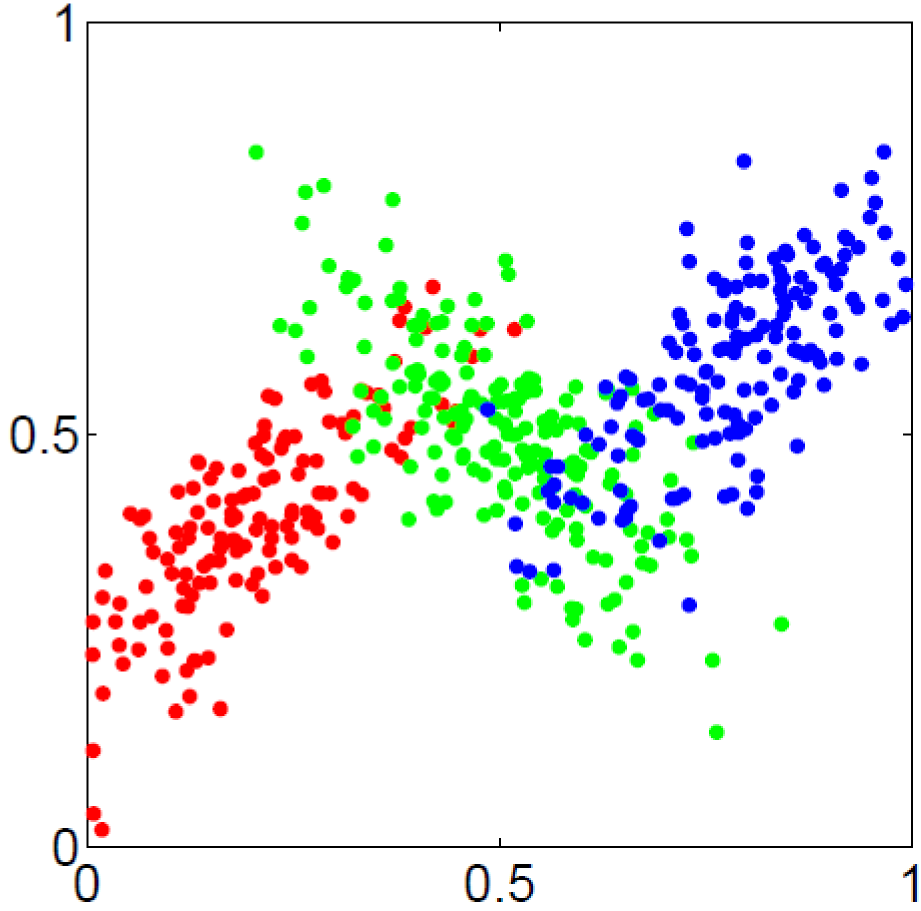
Gaussian Mixture Models



Gaussian Mixture Models



Gaussian Mixture Models



Sampling from the Gaussian

- To generate a data point:
  - first pick one of the components with probability π_k
  - then draw a sample $\mathbf{x}_n$ from that component
- Repeat these two steps for each new data point

Fitting the Gaussian Mixture

- We wish to invert this process – given the data set, find the corresponding parameters:
  - mixing coefficients
  - means
  - covariances
- If we knew which component generated each data point, the maximum likelihood solution would involve fitting each component to the corresponding cluster
- Problem: the data set is unlabelled
- We shall refer to the labels as *latent* (= hidden) variables

Gaussian Mixture Models

- We can think of the mixing coefficients as prior probabilities for the components
- For a given value of $\mathbf{x}$ we can evaluate the corresponding posterior probabilities, called *responsibilities*

- These are given from Bayes' theorem by

$$\gamma_k(\mathbf{x}) \equiv p(C_k \mid \mathbf{x}) = \frac{p(C_k)\,p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

$$= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Estimation of means

- Let us proceed by simply differentiating the log likelihood
- Setting derivative with respect to $\boldsymbol{\mu}_j$ equal to zero gives

$$- \sum_{n=1}^{N} \underbrace{\frac{\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_k \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}}_{\gamma_j(\mathbf{x}_n)} \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_j) = 0$$

giving

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

which is simply the weighted mean of the data

Estimation of covariance matrices

The expression of covariance matrix for j^{th} Gaussian component is given by: (Derivation not necessary !!)

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

Estimation of weights

Lagrange multipliers : Form Lagrange function

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

which gives

$$0 = \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda$$

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

Taking derivative with respect to π_1 gives

$$0 = \sum_{n=1}^{N} \left(\frac{N(\mathbf{x}_n \mid \boldsymbol{\mu}_{1,} \Sigma_1)}{\sum_j \pi_j \, N(\mathbf{x}_n \mid \boldsymbol{\mu}_{j,} \Sigma_j)} \right) + \lambda$$

Multiplying entire equation with π_1 gives

$$0\pi_1 = (\pi_1) \sum_{n=1}^{N} \left(\frac{N(\mathbf{x}_n \mid \boldsymbol{\mu}_{1,} \Sigma_1)}{\sum_j \pi_j \, N(\mathbf{x}_n \mid \boldsymbol{\mu}_{j,} \Sigma_j)} \right) + \lambda \pi_1$$

$$0 = \sum_{n=1}^{N} \left(\frac{\pi_1 N(\mathbf{x}_n \mid \boldsymbol{\mu}_{1,} \Sigma_1)}{\sum_j \pi_j \, N(\mathbf{x}_n \mid \boldsymbol{\mu}_{j,} \Sigma_j)} \right) + \lambda \pi_1 \qquad (1)$$

Similarly taking derivative with respect to π_2 and then multiplying with π_2 gives

$$0 = \sum_{n=1}^{N} \left(\frac{\pi_2 N(\mathbf{x}_n \mid \boldsymbol{\mu}_{2,} \Sigma_2)}{\sum_j \pi_j \, N(\mathbf{x}_n \mid \boldsymbol{\mu}_{j,} \Sigma_j)} \right) + \lambda \pi_2 \qquad (2)$$

Similarly taking derivative with respect to π_3 and then multiplying with π_3 gives

$$0 = \sum_{n=1}^{N} \left(\frac{\pi_3 N(\mathbf{x}_n \mid \boldsymbol{\mu}_{3,} \Sigma_3)}{\sum_j \pi_j \, N(\mathbf{x}_n \mid \boldsymbol{\mu}_{j,} \Sigma_j)} \right) + \lambda \pi_3 \qquad (3)$$

$$0 = \sum_{n=1}^{N} \left(\frac{\pi_M N(\mathbf{x}_n \mid \boldsymbol{\mu}_{M,} \Sigma_M)}{\sum_j \pi_j \, N(\mathbf{x}_n \mid \boldsymbol{\mu}_{j,} \Sigma_j)} \right) + \lambda \pi_M \qquad (M)$$

–

Summing equations (1) to (M) gives

$$0 = \sum_{n=1}^{N} \left(\frac{\sum_{k} \pi_k N(\mathbf{x}_n \mid \boldsymbol{\mu}_{k,} \Sigma_k)}{\sum_{j} \pi_j \; N(\mathbf{x}_n \mid \boldsymbol{\mu}_{j,} \Sigma_j)} \right) + \lambda \sum_{k} \pi_k$$

$$0 = \sum_{n=1}^{N} 1 + \lambda (\sum_{k} \pi_k)$$

$$0 = N + \lambda$$

$$\boxed{\lambda = -N}$$

$$\left(\frac{1}{\pi_k}\right)\sum_{n=1}^{N}\left(\frac{\pi_k N(\mathbf{x}_n \mid \boldsymbol{\mu}_{k,} \Sigma_k)}{\sum_j \pi_j \; N(\mathbf{x}_n \mid \boldsymbol{\mu}_{j,} \Sigma_j)}\right) - N = 0$$

$$\left(\frac{1}{\pi_k}\right)\sum_{n=1}^{N}\left(\gamma_k(\mathbf{x}_n)\right) - N = 0$$

Let $$\sum_{n=1}^{N}\left(\gamma_k(\mathbf{x}_n)\right) = N_k \neq 1$$

$$\frac{N_k}{\pi_k} - N = 0$$

$$\boxed{\pi_k = \frac{N_k}{N}}$$

Iterative solution

- The solutions are not closed form since they are coupled
- Suggests an iterative scheme for solving them:
  - Make initial guesses for the parameters
  - Alternate between the following two stages:
    1. E-step: evaluate responsibilities
    2. M-step: update parameters using ML results

GMM Algorithm

EM for Gaussian Mixtures

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 $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.
2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{9.23}$$

Here $\gamma(z_{nk})$ is the same as $\gamma_k(\mathbf{x}_n)$

GMM Algorithm

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

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n \tag{9.24}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \tag{9.25}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{9.26}$$

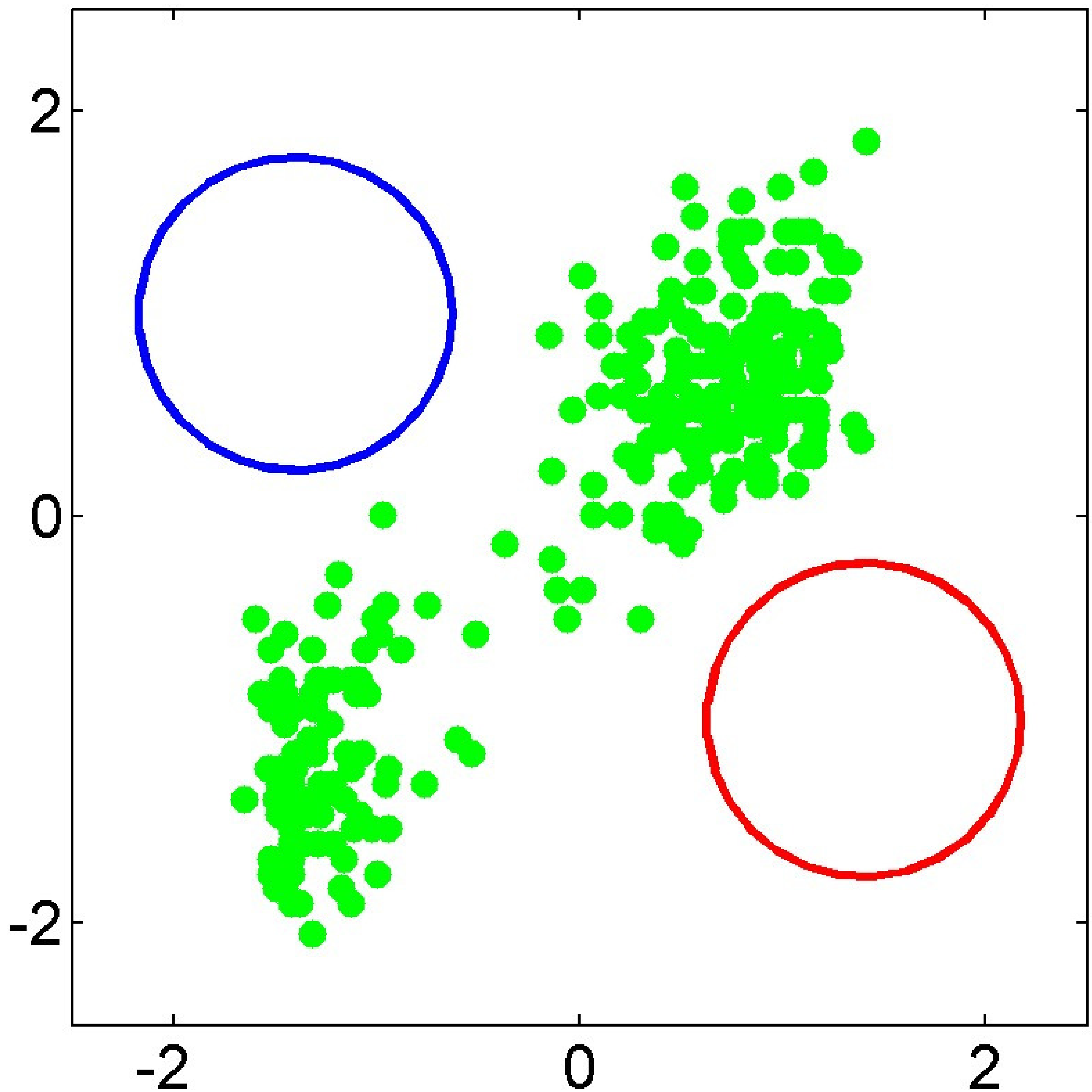
where

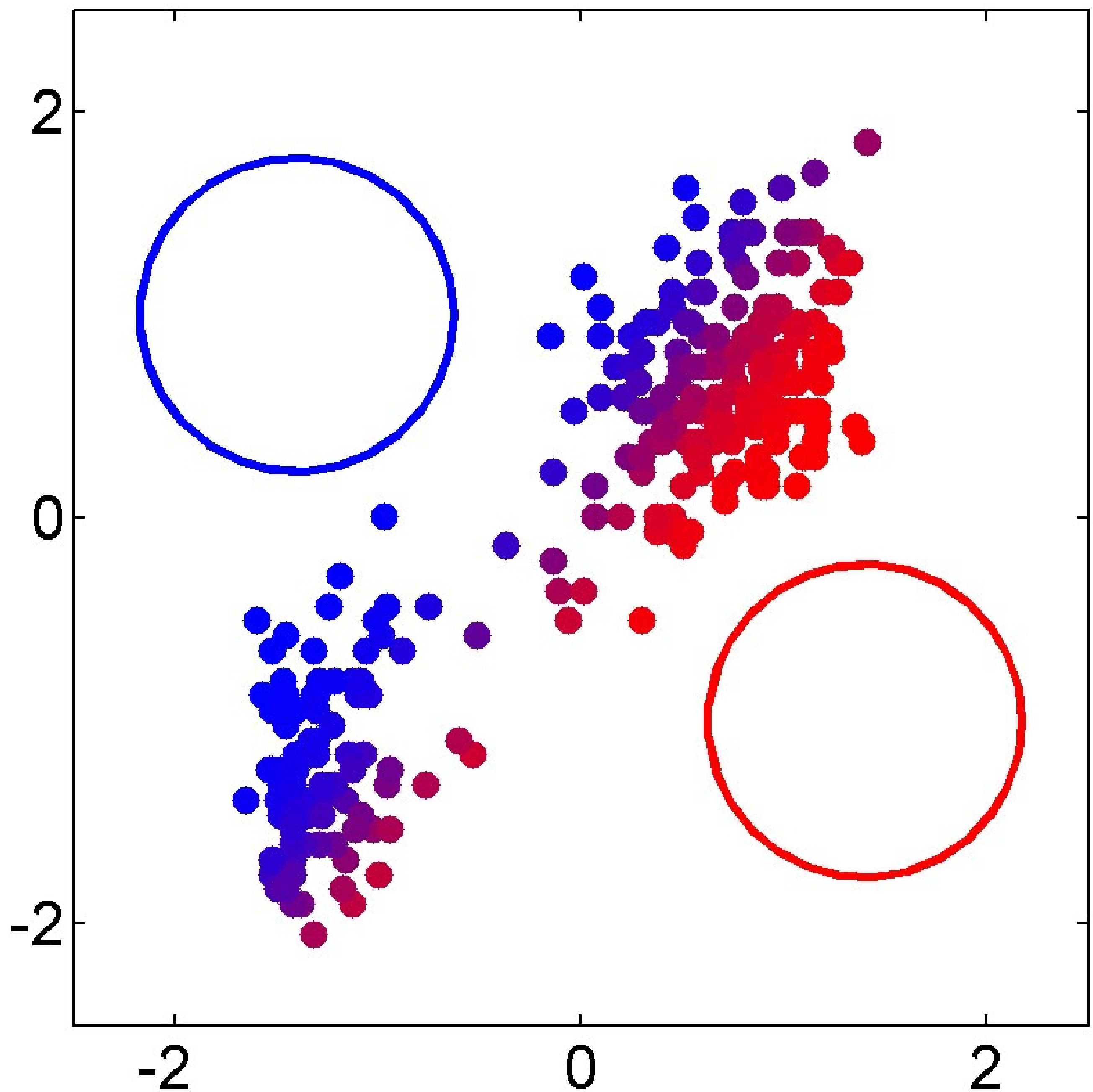
$$N_k = \sum_{n=1}^{N}\gamma(z_{nk}). \tag{9.27}$$

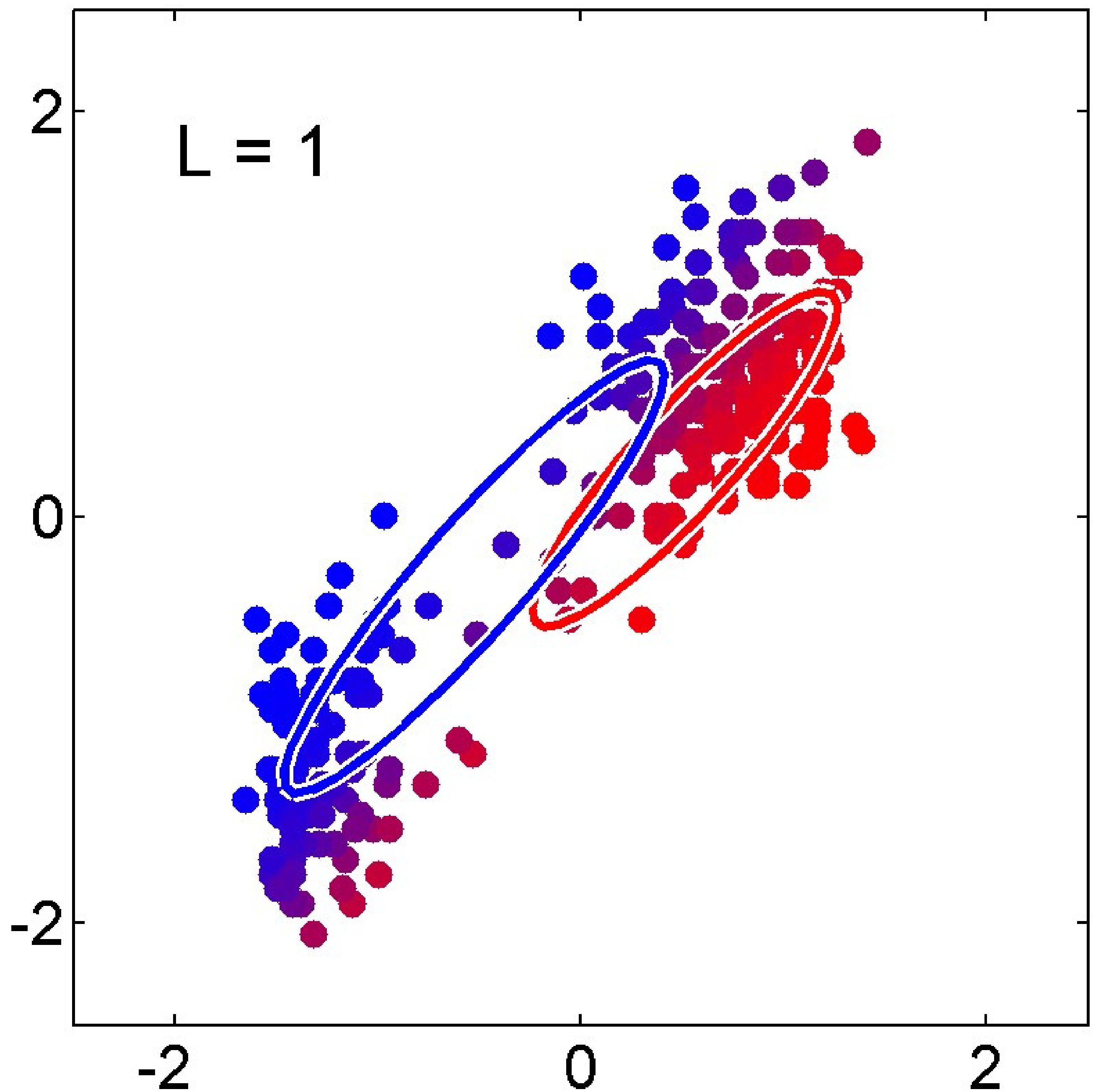
4. Evaluate the log likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi}) = \sum_{n=1}^{N}\ln\left\{\sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right\} \tag{9.28}$$

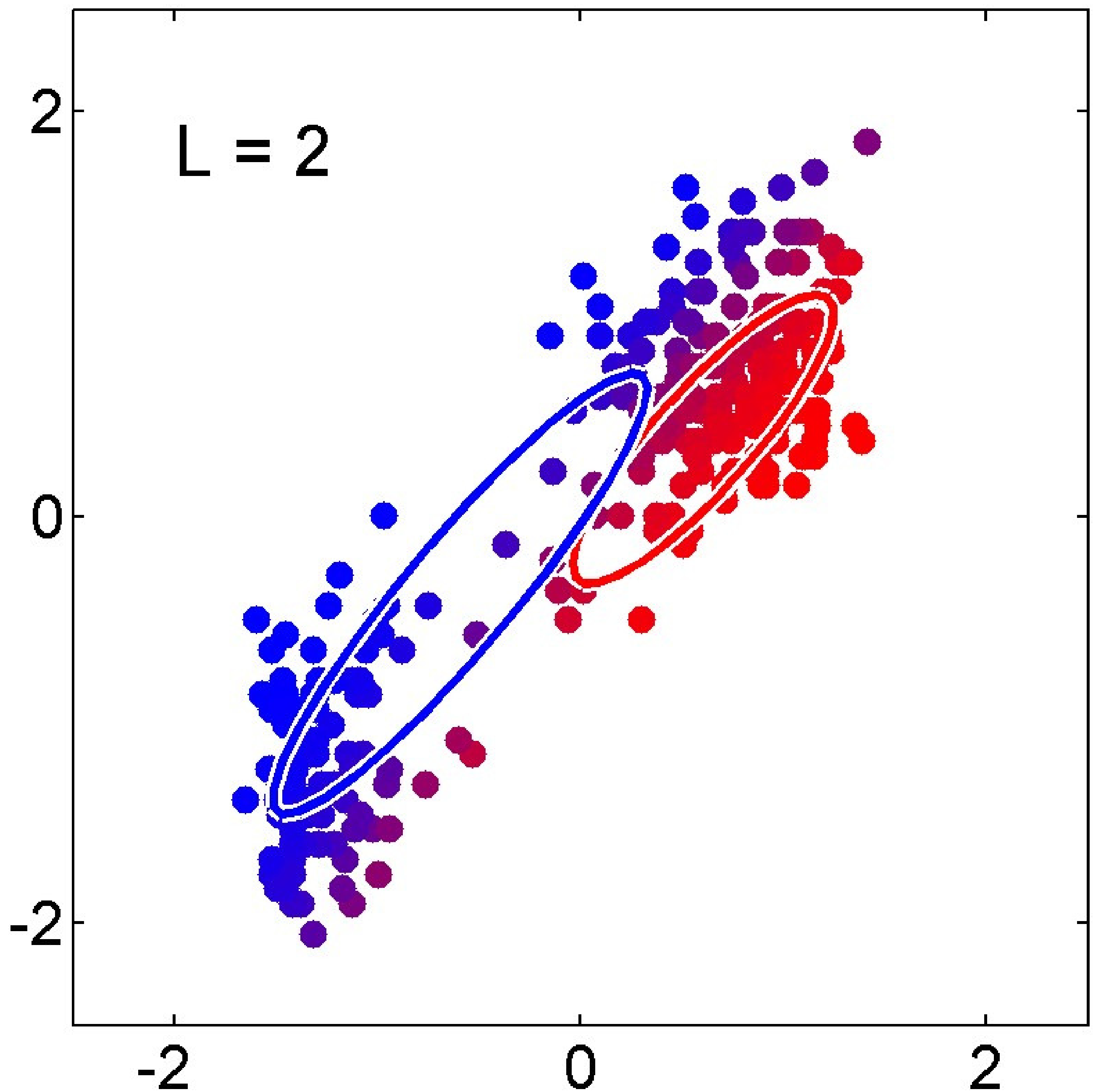
and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.


2
0
-2
-2
0
2

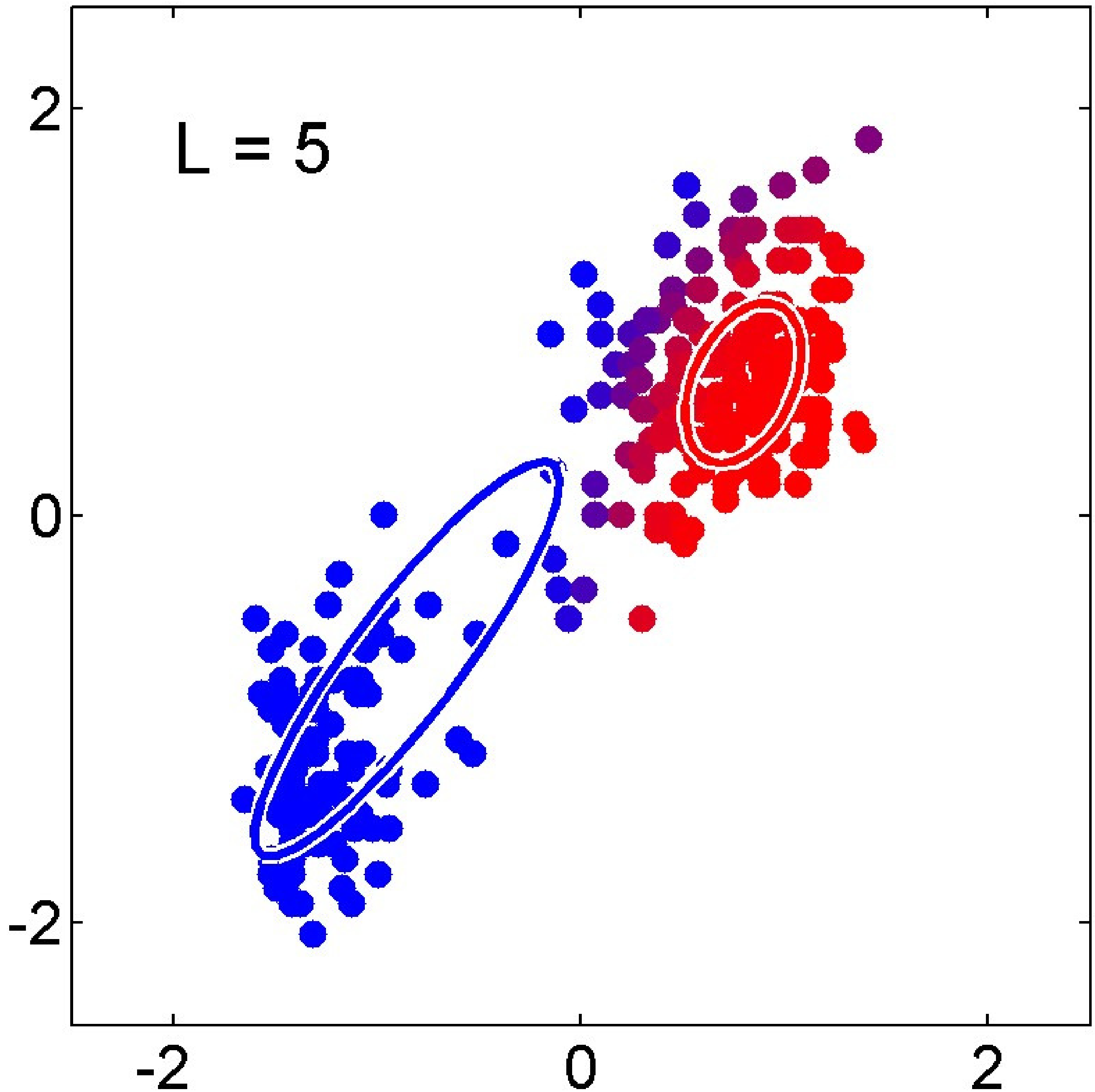

2
0
-2
-2
0
2

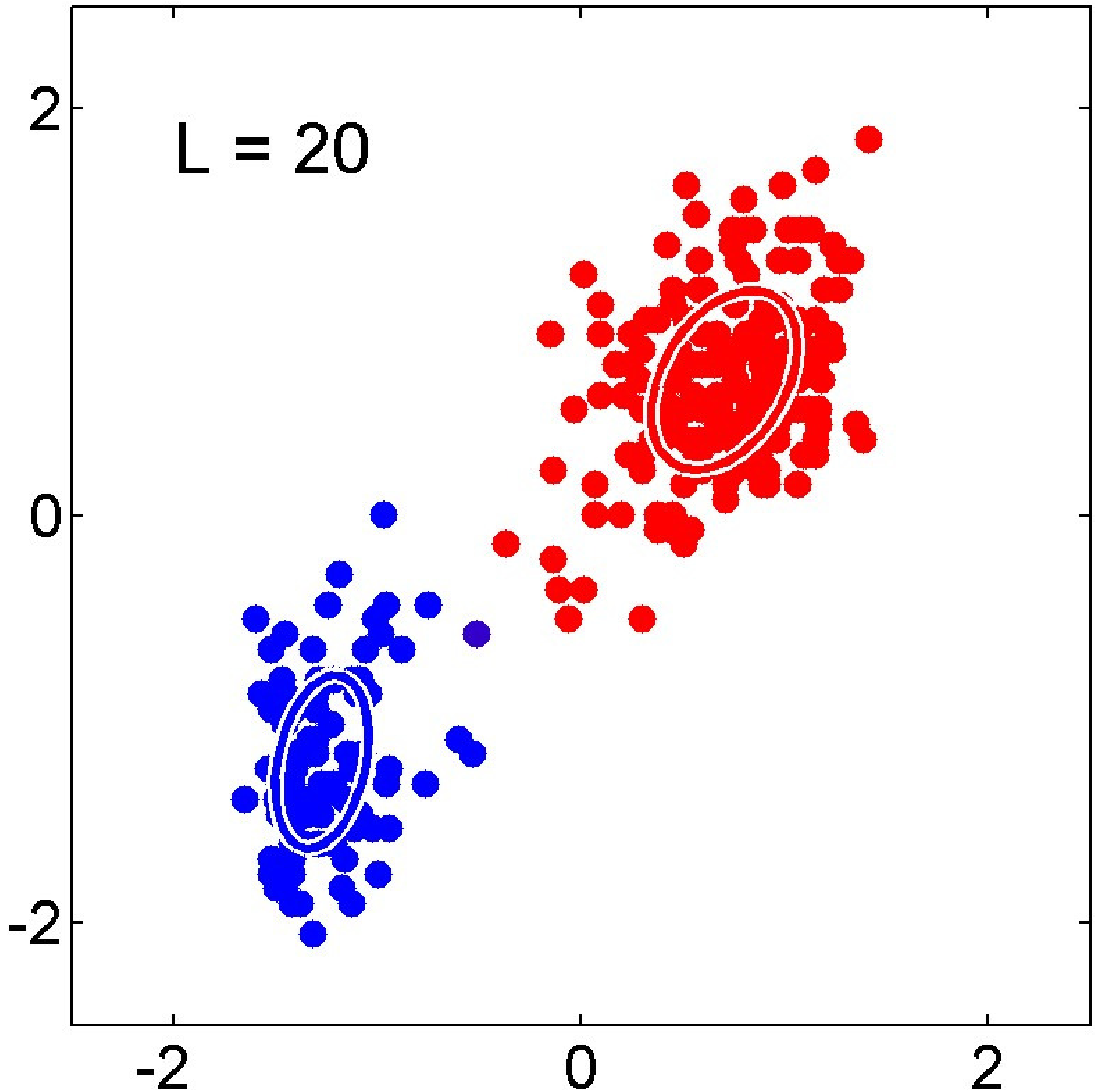


L = 1
2
0
-2
-2
0
2



L = 2
2
0
-2
-2
0
2


L = 5
2
0
-2
-2
0
2

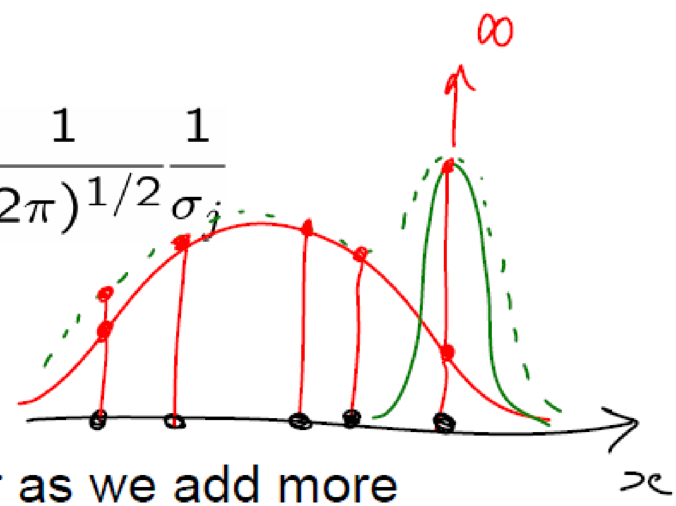

L = 20
2
0
-2
-2
0
2

Overfitting in GMMs

Over-fitting in Gaussian Mixture Models

- Singularities in likelihood function when a component 'collapses' onto a data point:

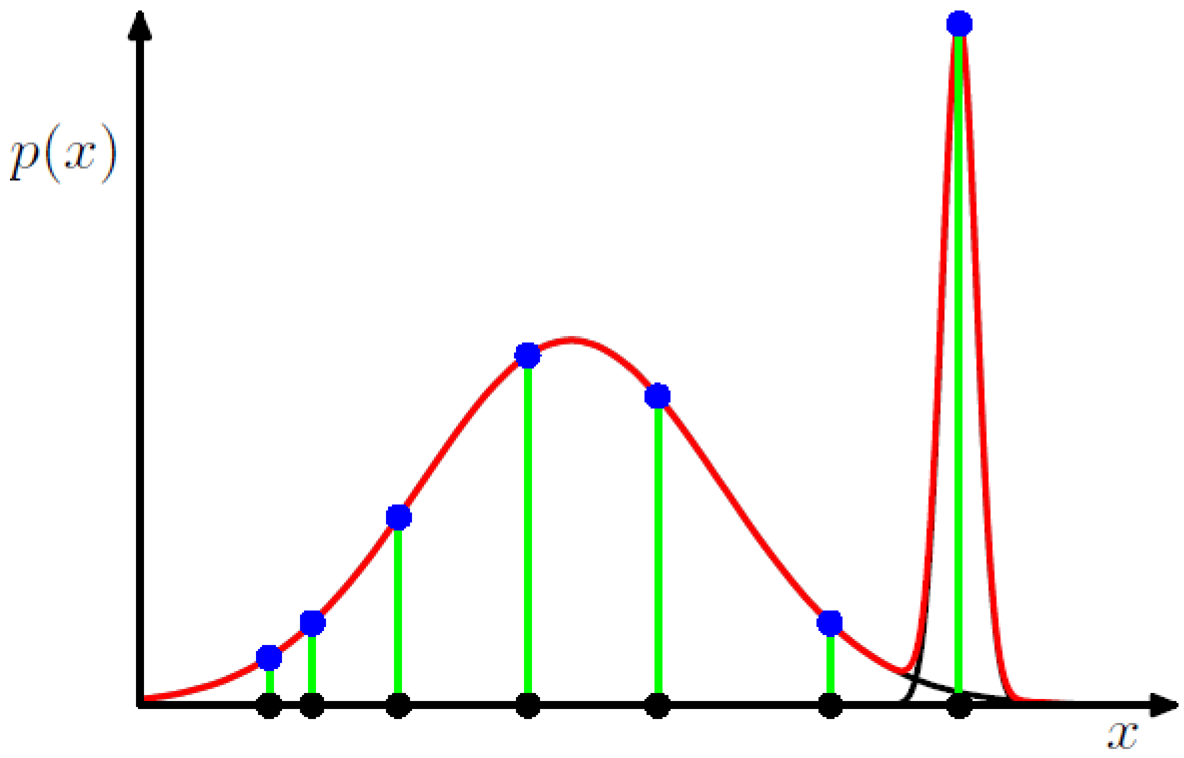
$$\mathcal{N}(\mathbf{x}_n|\mathbf{x}_n, \sigma_j^2\mathbf{I}) = \frac{1}{(2\pi)^{1/2}}\frac{1}{\sigma_j}$$



then consider $\sigma_j \to 0$

- Likelihood function gets larger as we add more components (and hence parameters) to the model
  - not clear how to choose the number K of components

Overfitting in GMMs



Relation to k means

- Consider GMM with common covariances
- Take limit $\epsilon \to 0$
- Responsibilities become binary

$$\gamma_i(\mathbf{x}_n) = \frac{\pi_i \exp\left\{-\|\mathbf{x}_n - \boldsymbol{\mu}_i\|^2/2\epsilon\right\}}{\sum_j \pi_j \exp\left\{-\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2/2\epsilon\right\}} \to r_{ni} \in \{0, 1\}$$

- EM algorithm is precisely equivalent to K-means

Relation to k means

$$\gamma_i(\mathbf{x}_n) = \frac{\pi_i \exp\left\{-\|\mathbf{x}_n - \boldsymbol{\mu}_i\|^2/2\epsilon\right\}}{\sum_j \pi_j \exp\left\{-\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2/2\epsilon\right\}} \rightarrow r_{ni} \in \{0, 1\}$$

If we consider the limit $\epsilon \to 0$, we see that in the denominator the term for which $\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2$ is smallest will go to zero most slowly, and hence the responsibilities $\gamma(z_{nk})$ for the data point $\mathbf{x}_n$ all go to zero except for term j, for which the responsibility $\gamma(z_{nj})$ will go to unity. Note that this holds independently of the values of the π_k so long as none of the π_k is zero. Thus, in this limit, we obtain a hard assignment of data points to clusters, just as in the K-means algorithm, so that $\gamma(z_{nk}) \to r_{nk}$ where r_{nk} is defined by (9.2). Each data point is thereby assigned to the cluster having the closest mean.