# Mixture of Gaussians
# Expectation Maximization (EM)

## Part 1

Most of the slides are due to Christopher Bishop

BCS Summer School, Exeter, 2003.

The rest of the slides are based on lecture notes by A. Ng

# Limitations of K-means

- Hard assignments of data points to clusters – small shift of a data point can flip it to a different cluster
- Not clear how to choose the value of K
- Solution: replace ‘hard’ clustering of K-means with ‘soft’ probabilistic assignments
- Represents the probability distribution of the data as a *Gaussian mixture model*

# The Gaussian Distribution

- Multivariate Gaussian

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

mean covariance

- Define precision to be the inverse of the covariance

$$\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$$

- In 1-dimension

$$\tau = \frac{1}{\sigma^2}$$

# Gaussian Mixtures

- 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

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

# Sampling from the Gaussian

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

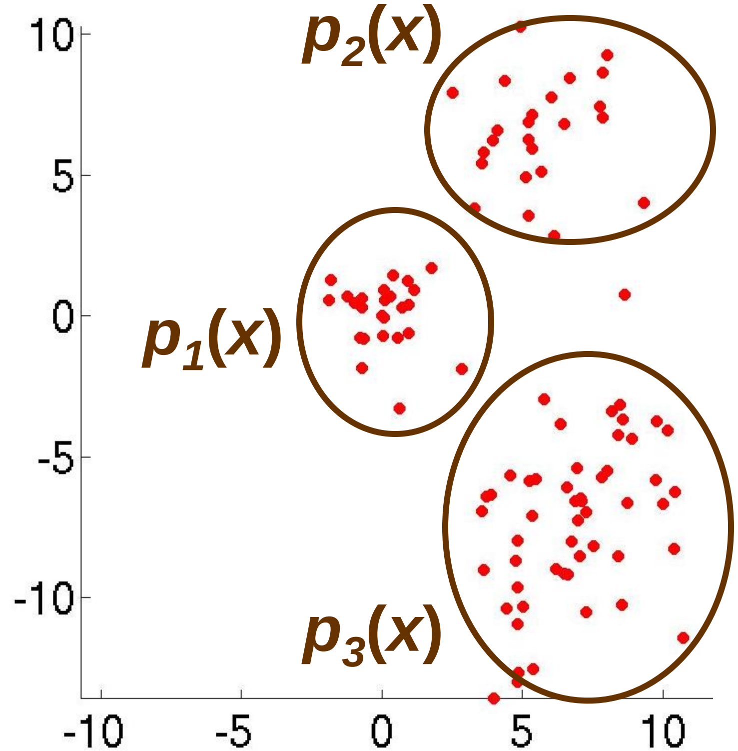
# Example: Gaussian Mixture Density

- Mixture of 3 Gaussians

$$p_1(x) \cong N\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$$

$$p_2(x) \cong N\left([6,6], \begin{bmatrix}4 & 0\\0 & 4\end{bmatrix}\right)$$

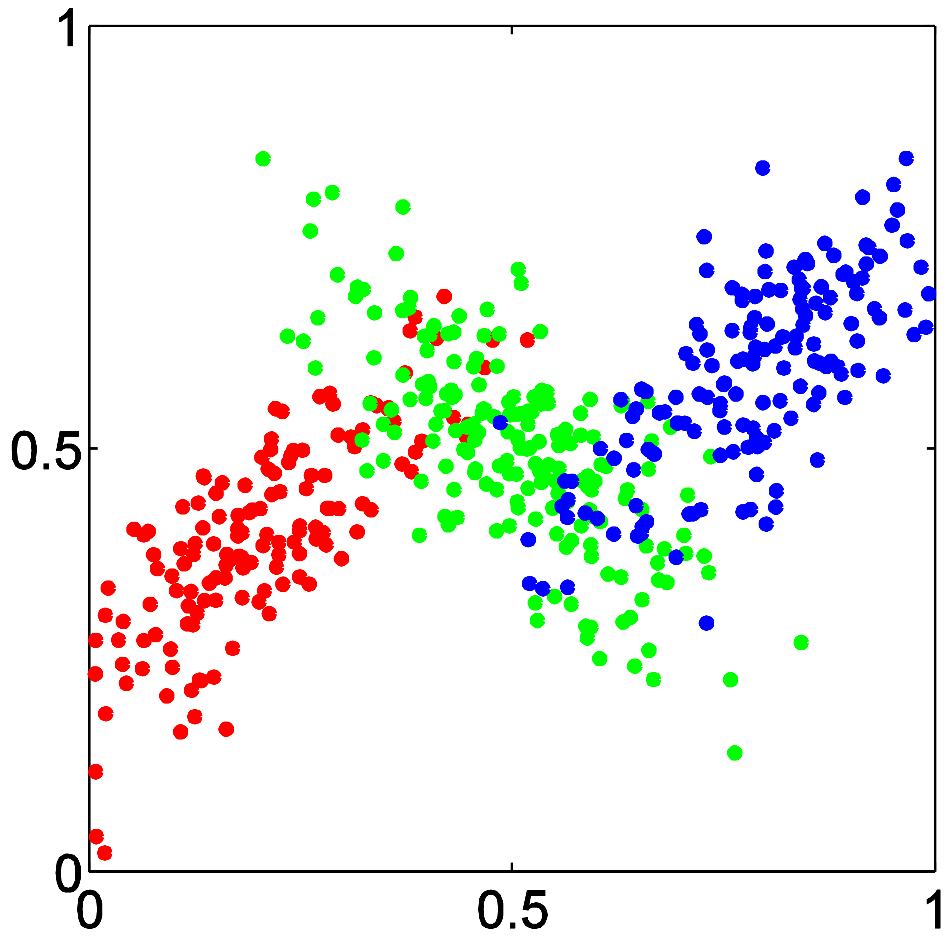
$$p_3(x) \cong N\left([7,-7], \begin{bmatrix}6 & 0\\0 & 6\end{bmatrix}\right)$$




$$p(x) = 0.2\,p_1(x) + 0.3\,p_2(x) + 0.5\,p_3(x)$$

# Synthetic Data Set

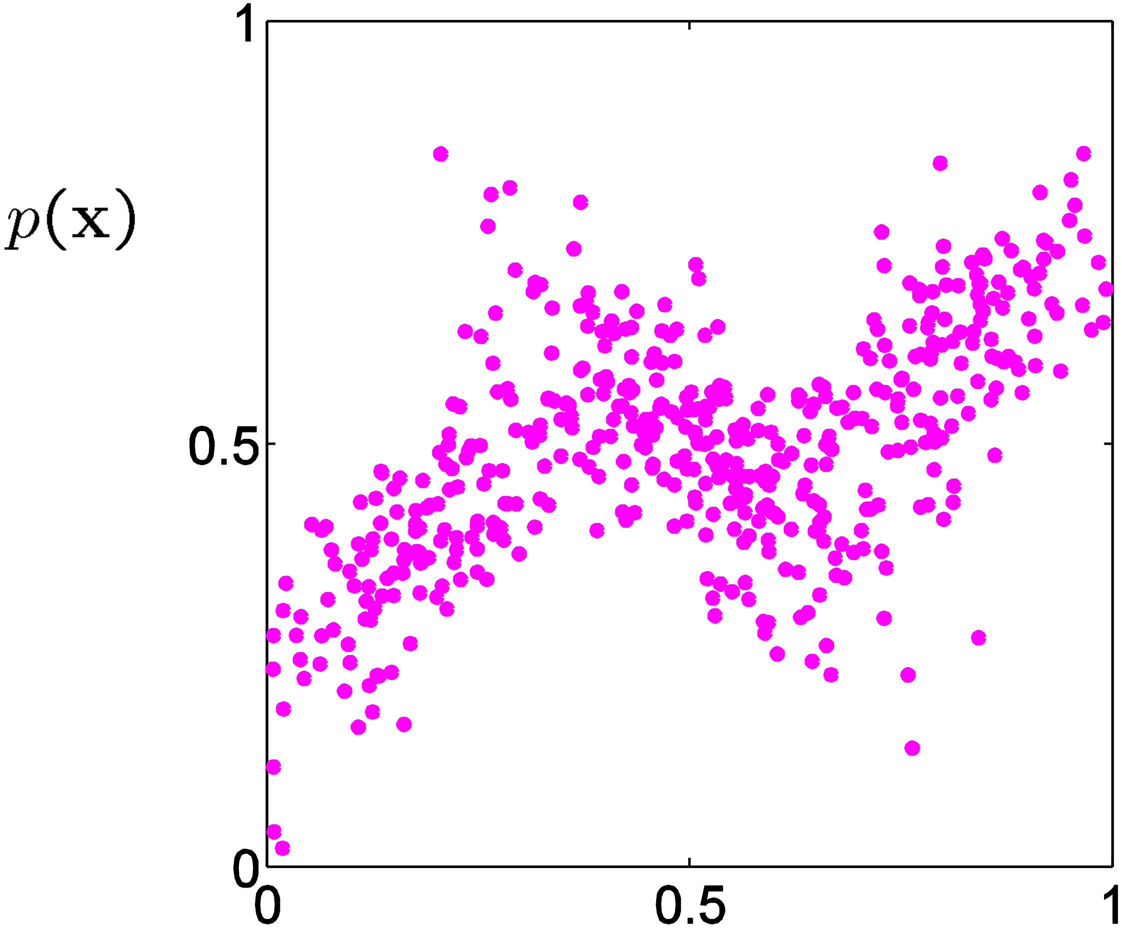
$p(\mathbf{x}|k)$



# 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

# Synthetic Data Set Without Labels



# Posterior Probabilities

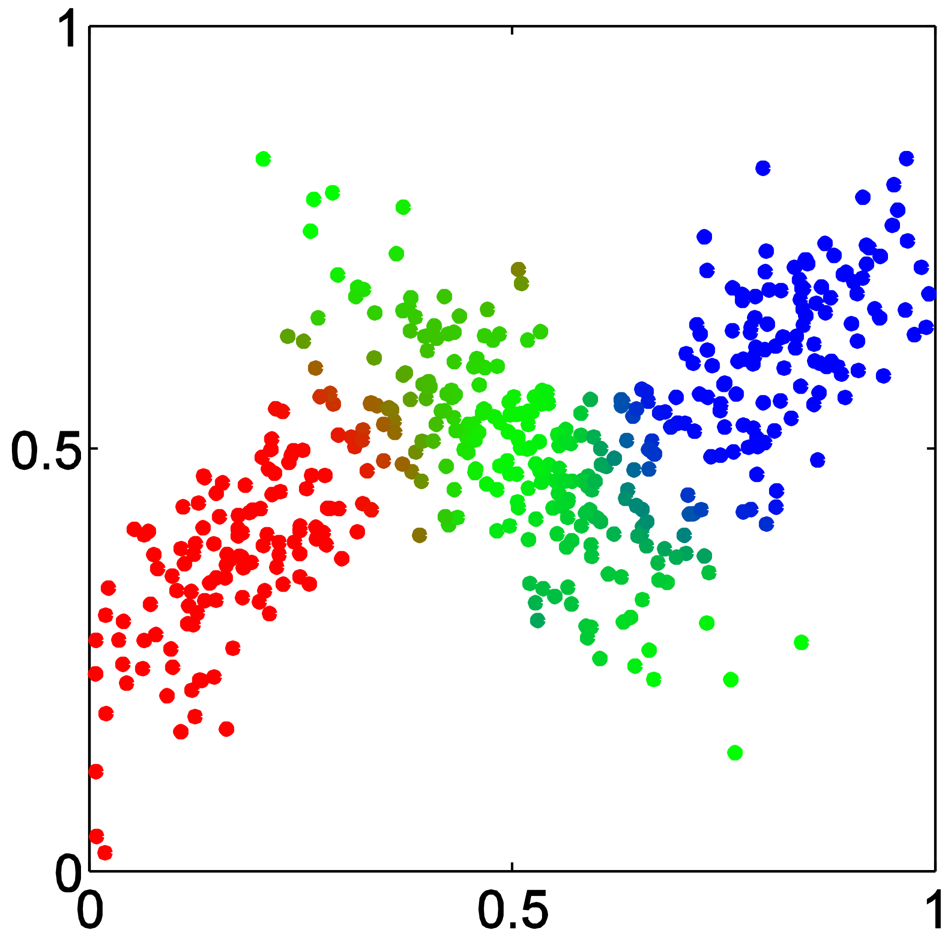
- We can think of the mixing coefficients as prior probabilities for the components
- For a given value of $\mathbf{x}$ we can evaluate the corresponding posterior probabilities, called *responsibilities*
- These are given from Bayes' theorem by

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

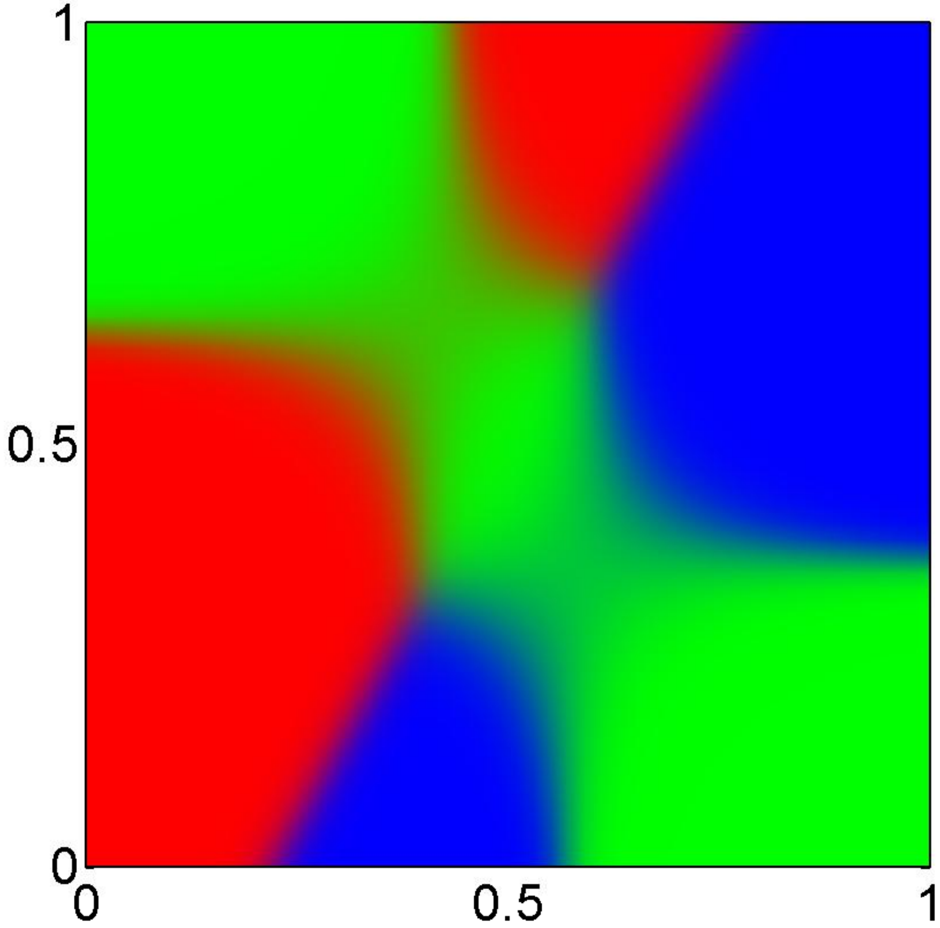
$$= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

# Posterior Probabilities (colour coded)

$p(k \mid x)$



# Posterior Probability Map



# Maximum Likelihood for the GMM

- The log likelihood function takes the form

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

- Note: sum over components appears *inside* the log
- There is no closed form solution for maximum likelihood
- How to maximize the log likelihood
  – solved by expectation-maximization (EM) algorithm

# EM Algorithm – Informal Derivation

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

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

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

# EM Algorithm – Informal Derivation

- Similarly for the covariances

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

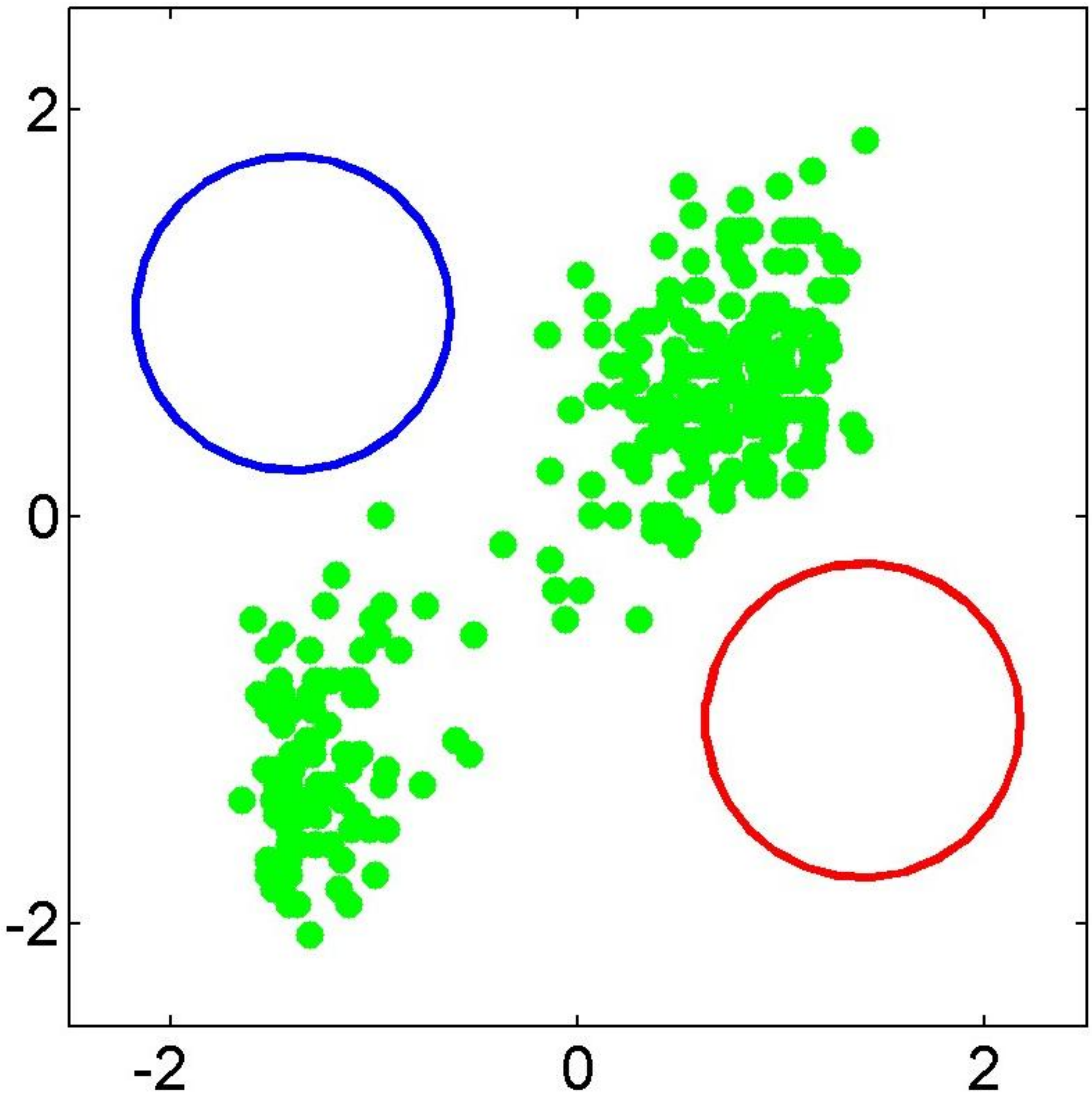
- For mixing coefficients use a Lagrange multiplier to give

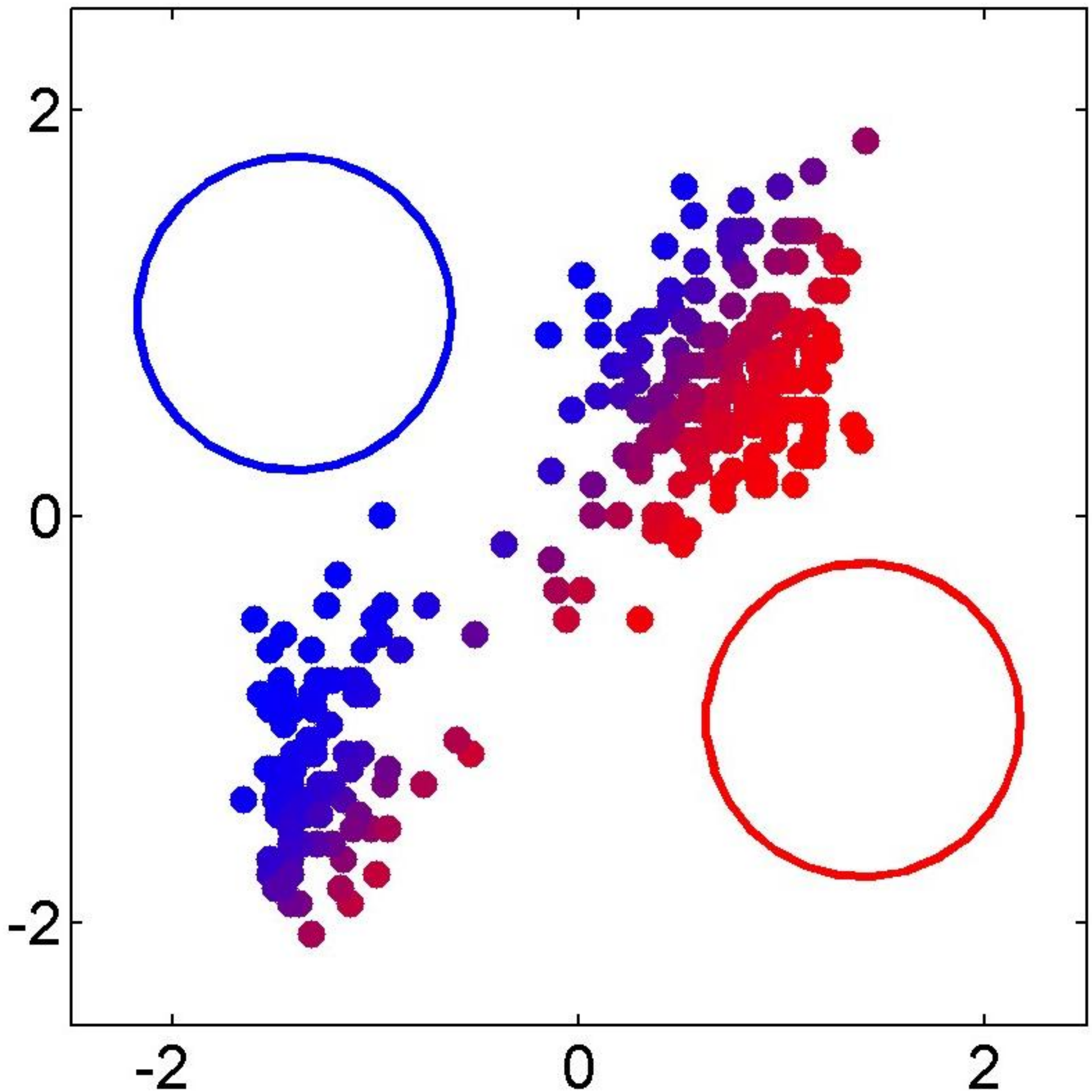
$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)$$

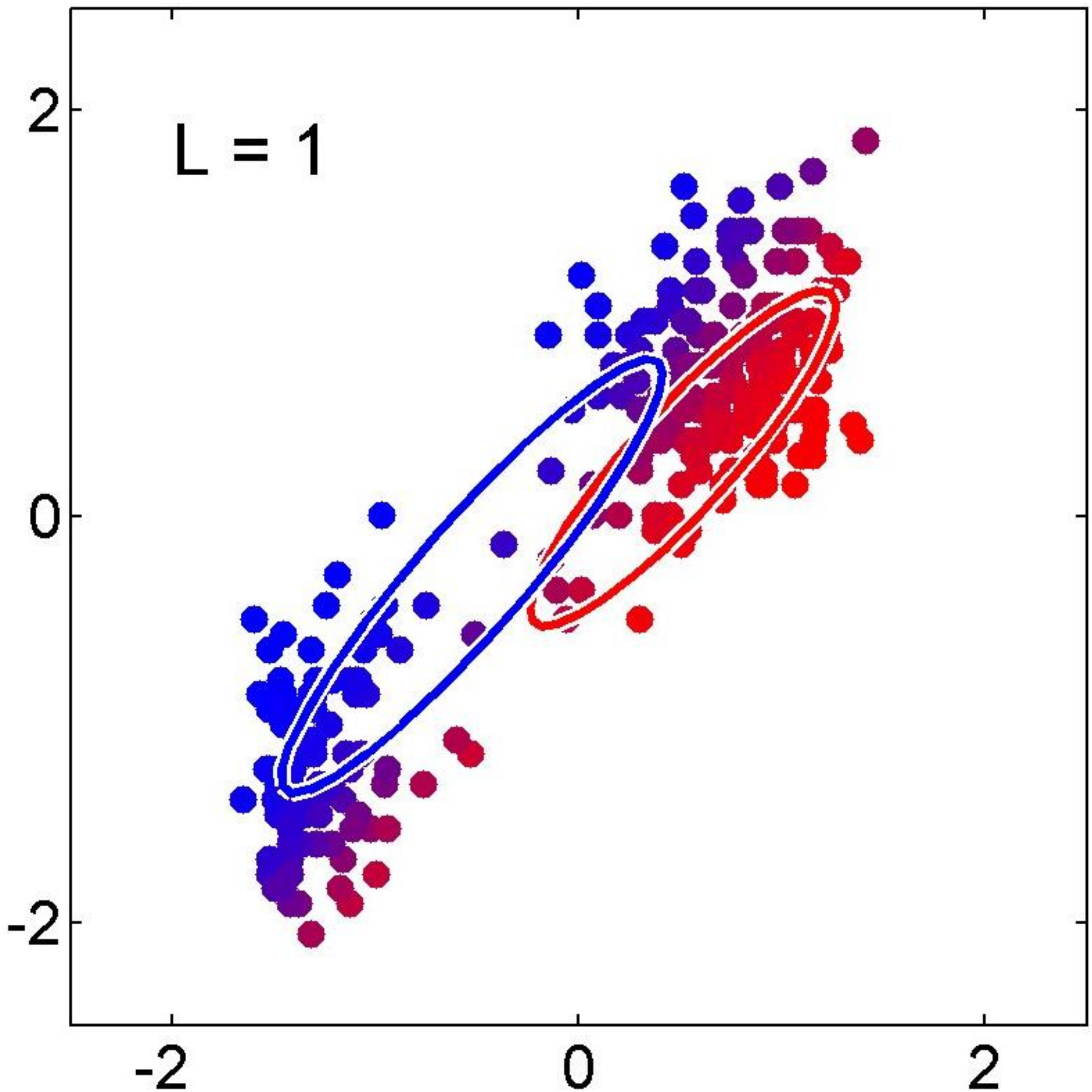
Average responsibility which component j takes for explaining the data points.

# EM Algorithm – Informal Derivation

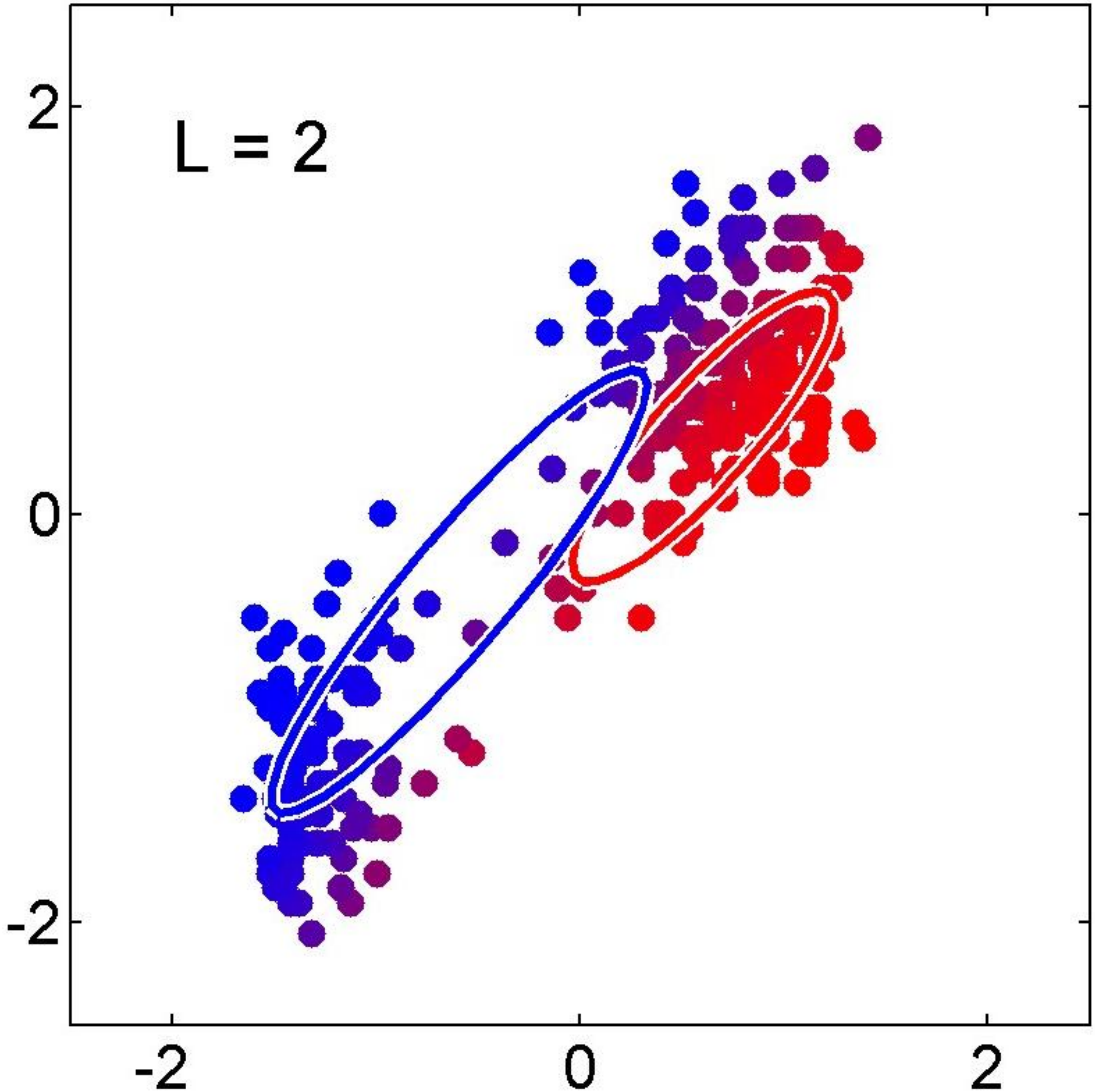
- The solutions are not closed form since they are coupled
- Suggests an iterative scheme for solving them:
  - Make initial guesses for the parameters
  - Alternate between the following two stages:
    1. E-step: evaluate responsibilities
    2. M-step: update parameters using ML results

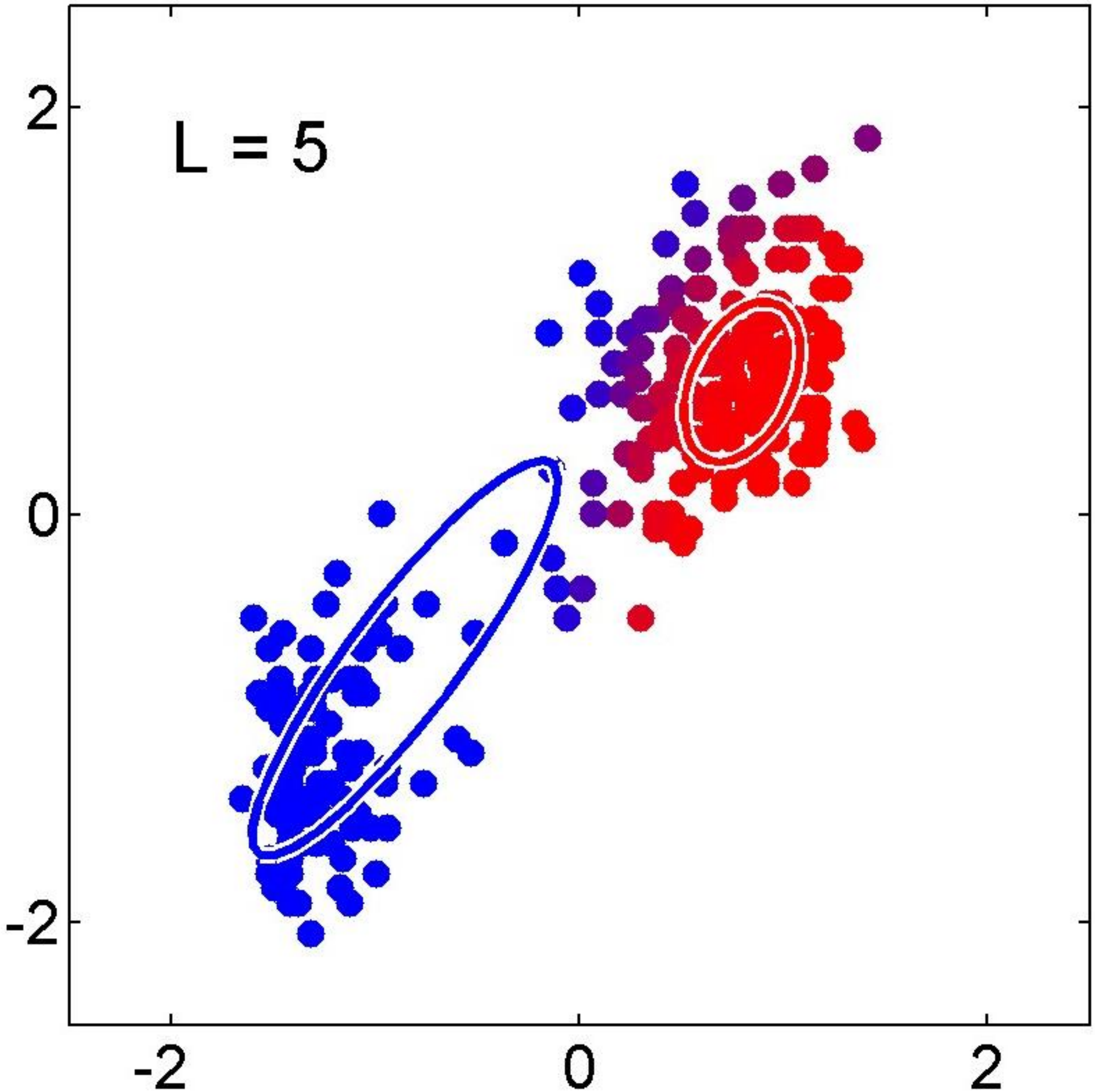

2
0
-2
-2
0
2


2
0
-2
-2
0
2

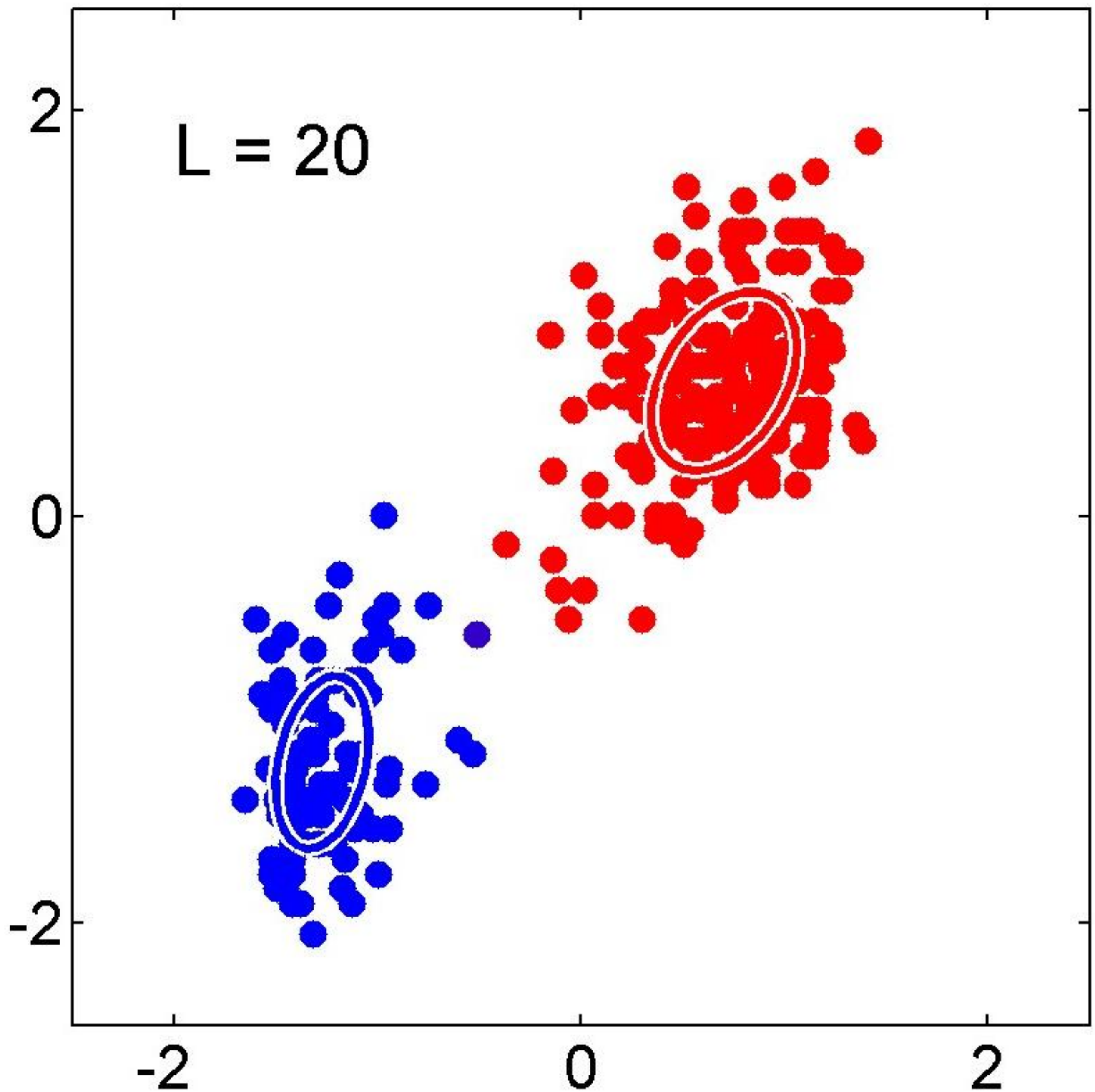


L = 1
2
0
-2
-2
0
2


L = 2
2
0
-2
-2
0
2



L = 5
2
0
-2
-2
0
2



L = 20
2
0
-2
-2
0
2