Machine Learning

CSE 6363 (Fall 2016)

Lecture 16 K-means and EM

Heng Huang, Ph.D.

Department of Computer Science and Engineering

General Idea: Expectation Maximization

- Start by devising a noisy channel
  - Any model that predicts the corpus observations via some hidden structure (tags, parses, …)
- Initially **guess** the parameters of the model!
  - Educated guess is best, but random can work

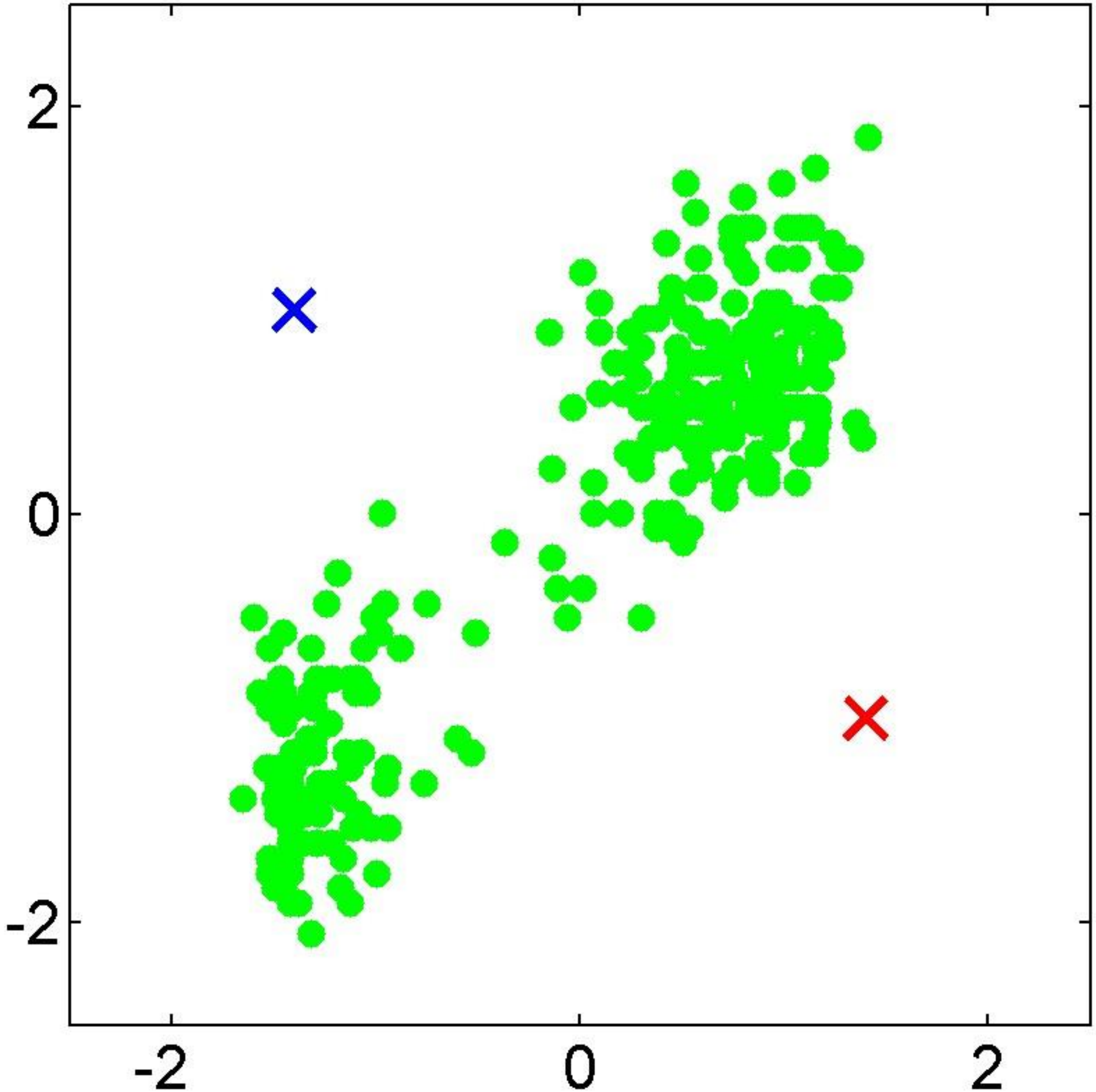
- **Expectation step:** Use current parameters (and observations) to reconstruct hidden structure
- **Maximization step:** Use that hidden structure (and observations) to reestimate parameters

Repeat until convergence!

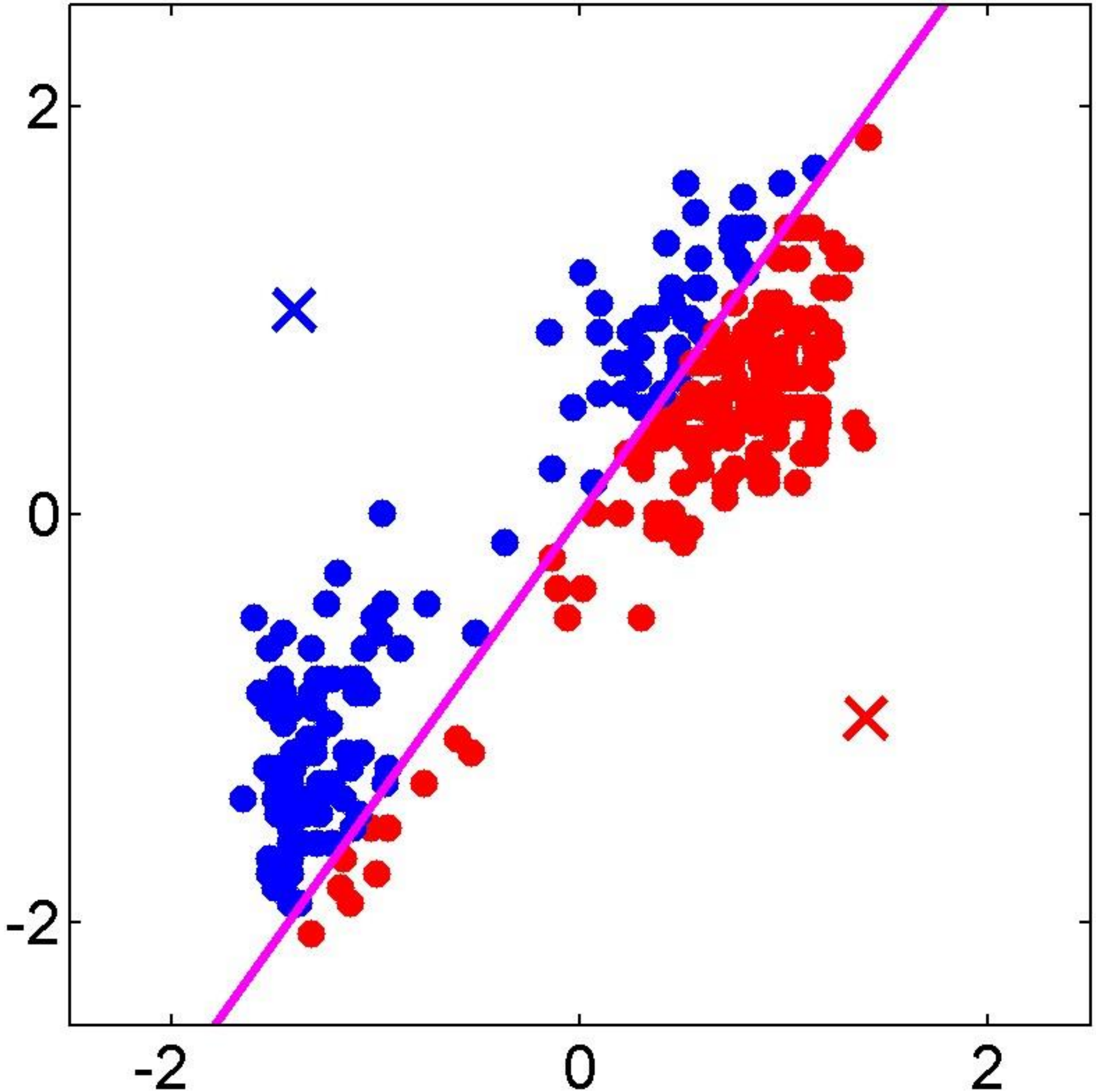
K-means Algorithm

- Goal
  - represent a data set in terms of K clusters each of which is summarized by a point-learner $\boldsymbol{\mu}_k$
- Initialize prototypes, then iterate between two phases:
  - E-step: assign each data point to nearest learner
  - M-step: update learners to be the cluster means
- Simplest version is based on Euclidean distance

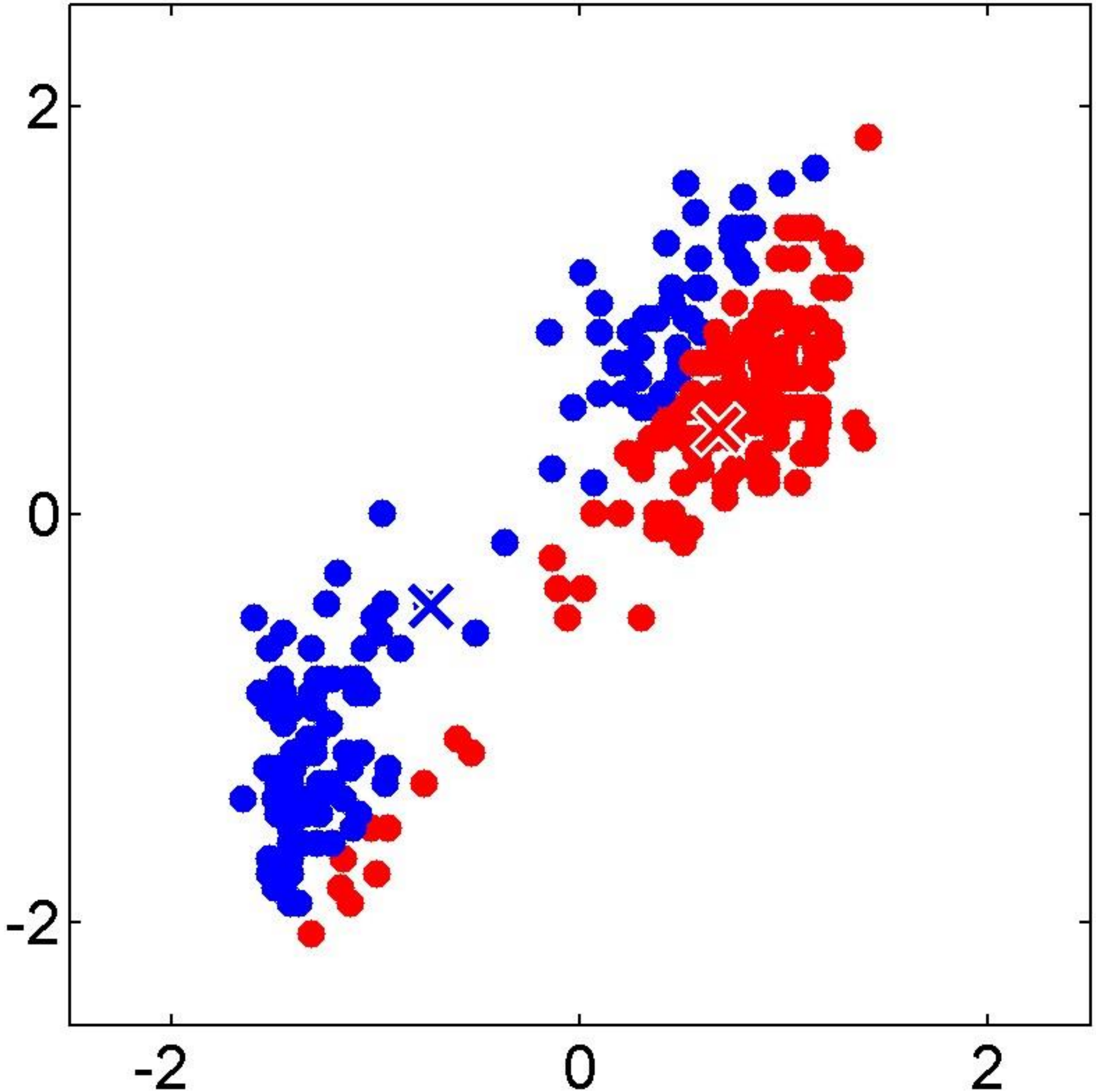
$$Dist(X,Y) = \sqrt{\sum_{i=1}^{m}(X_i - Y_i)^2}$$



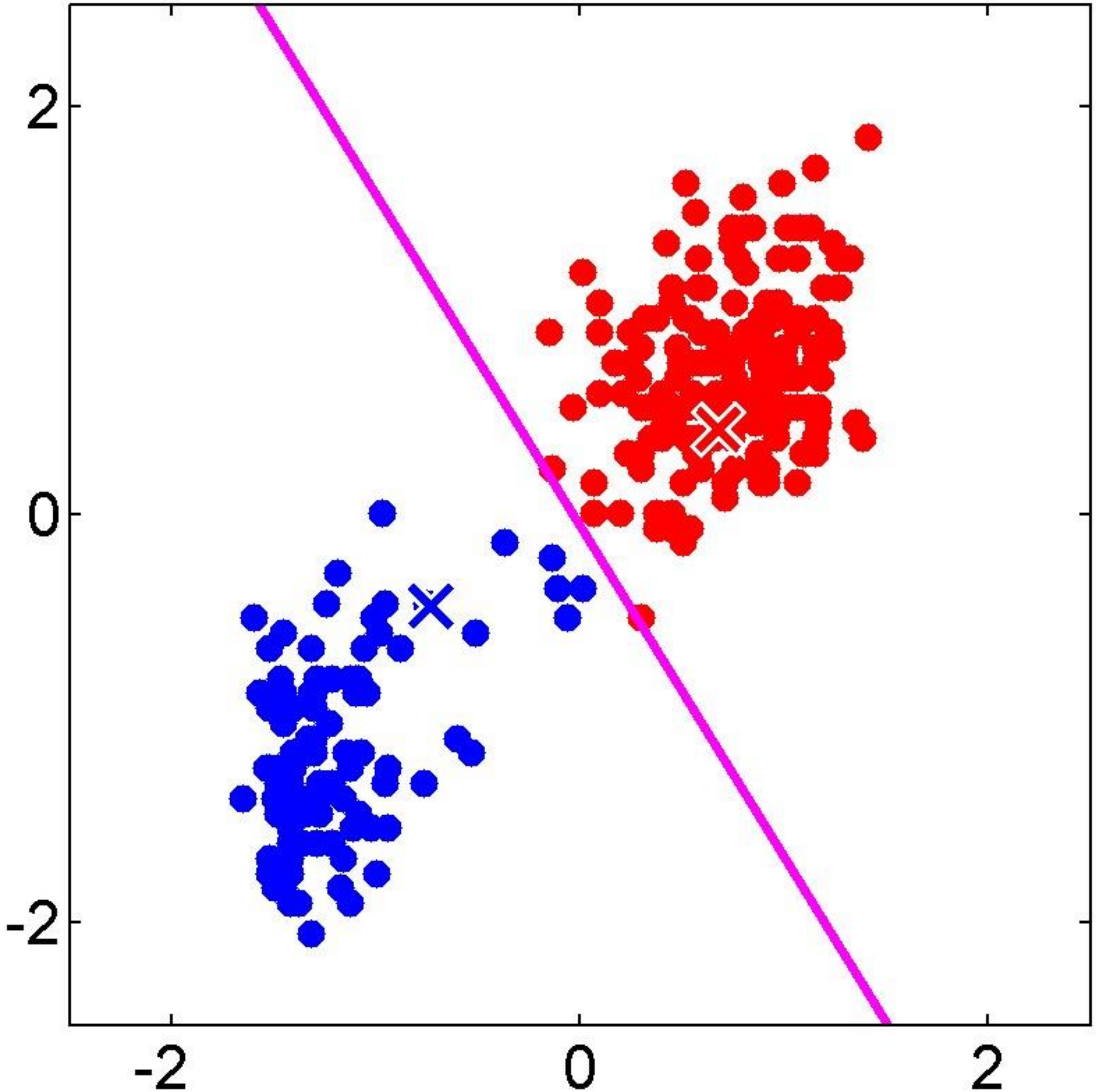
2
0
-2
-2
0
2



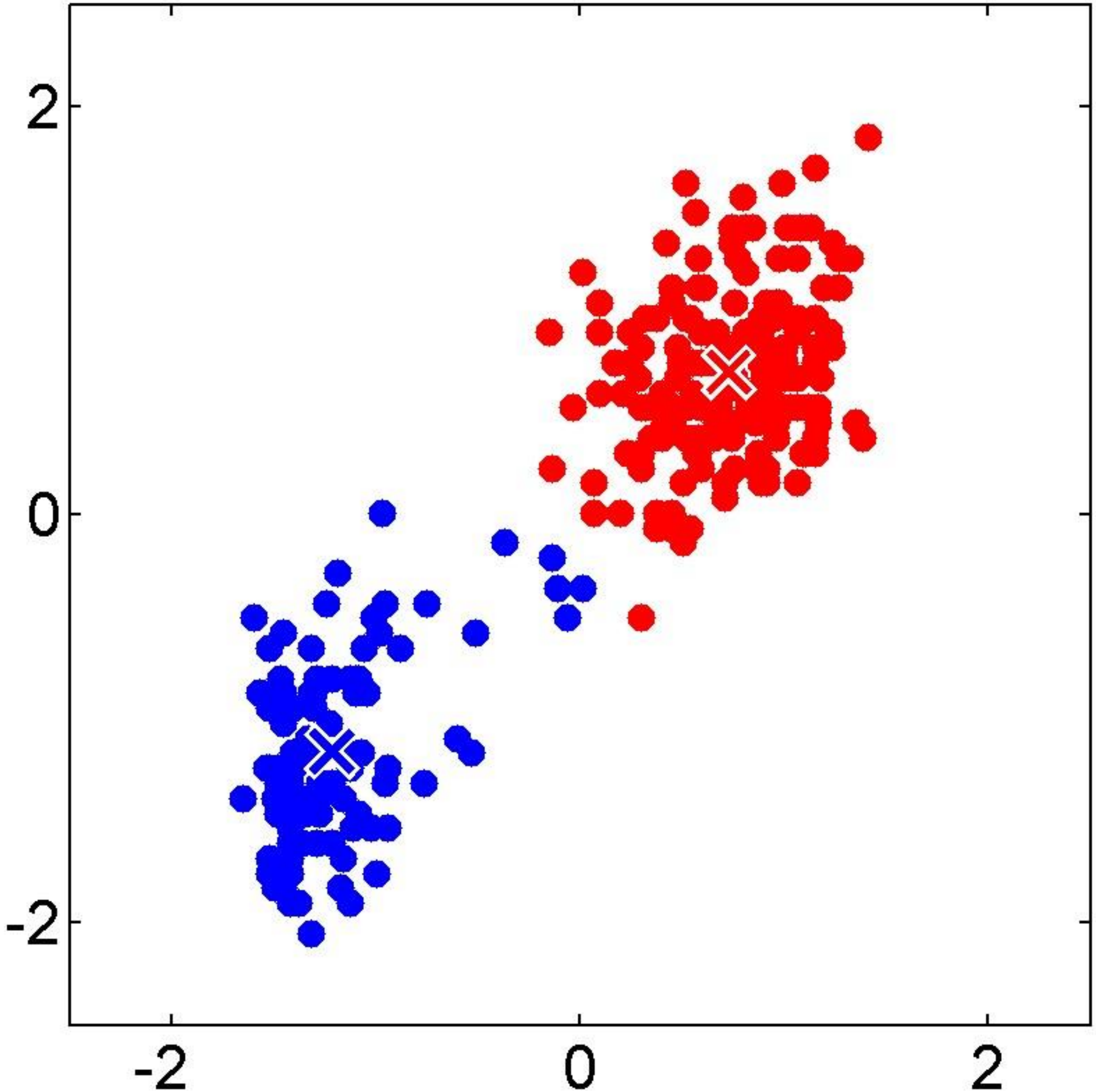
2
0
-2
-2
0
2

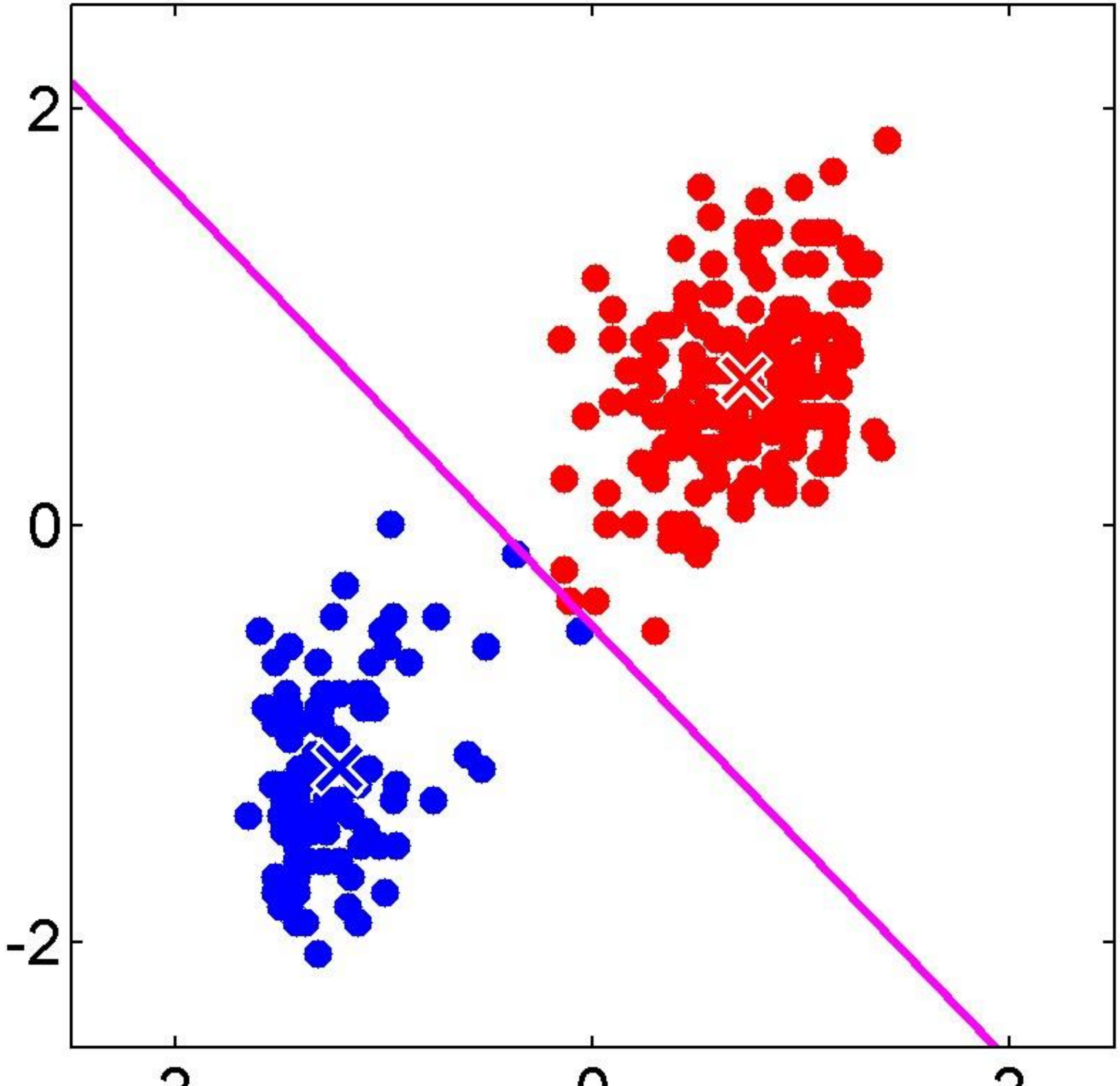


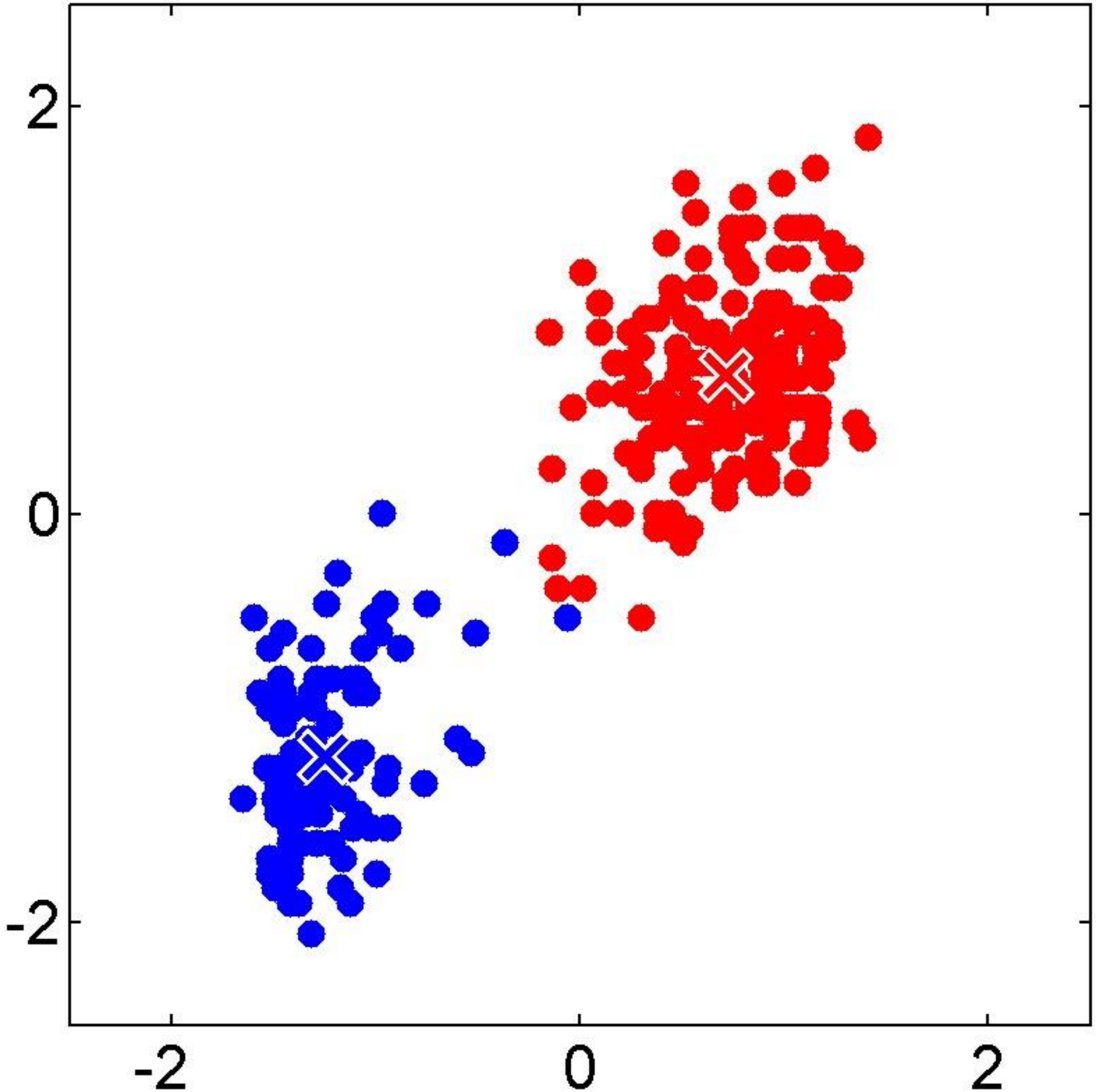
2
0
-2
-2
0
2



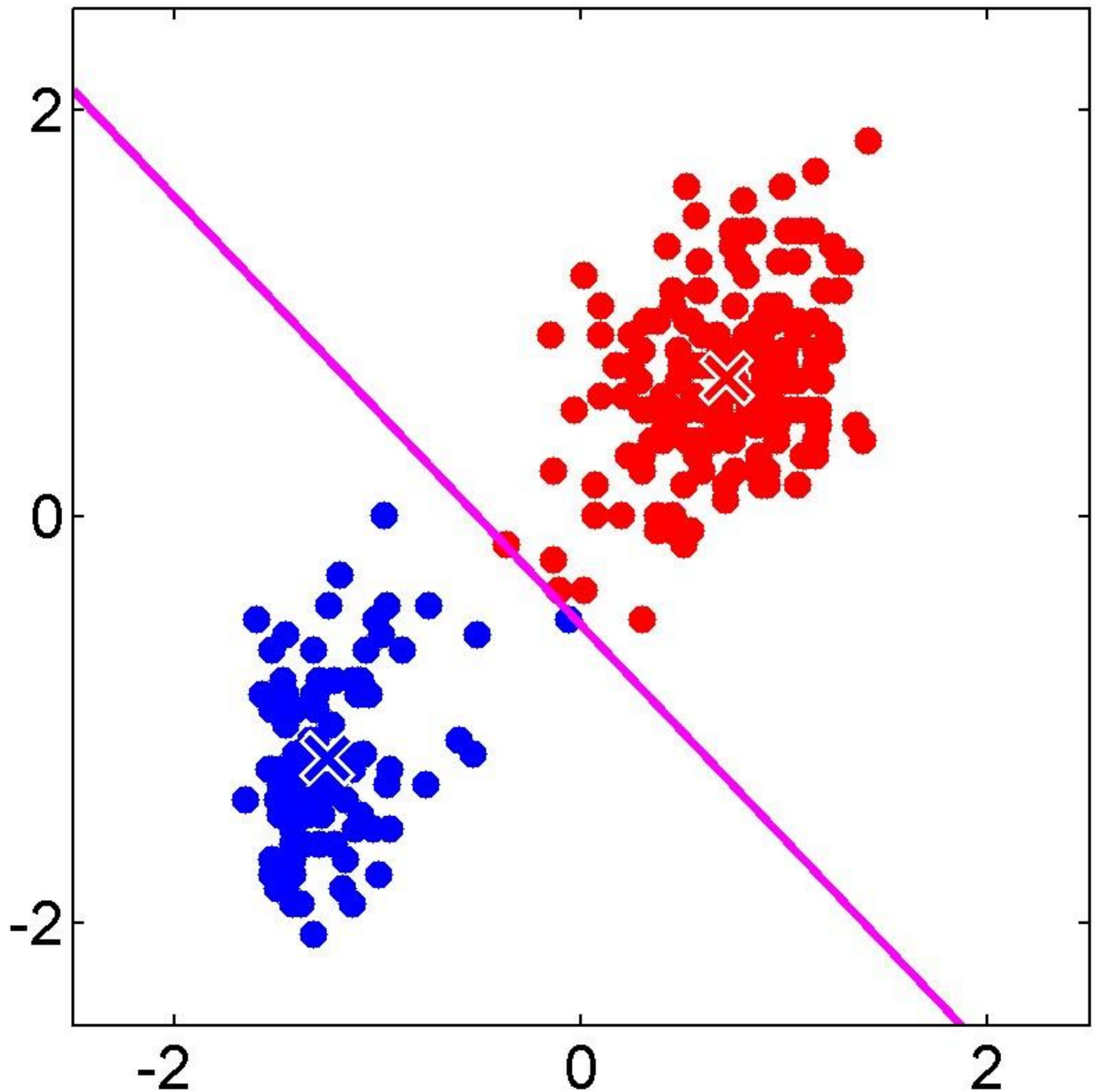
2
0
-2
-2
0
2

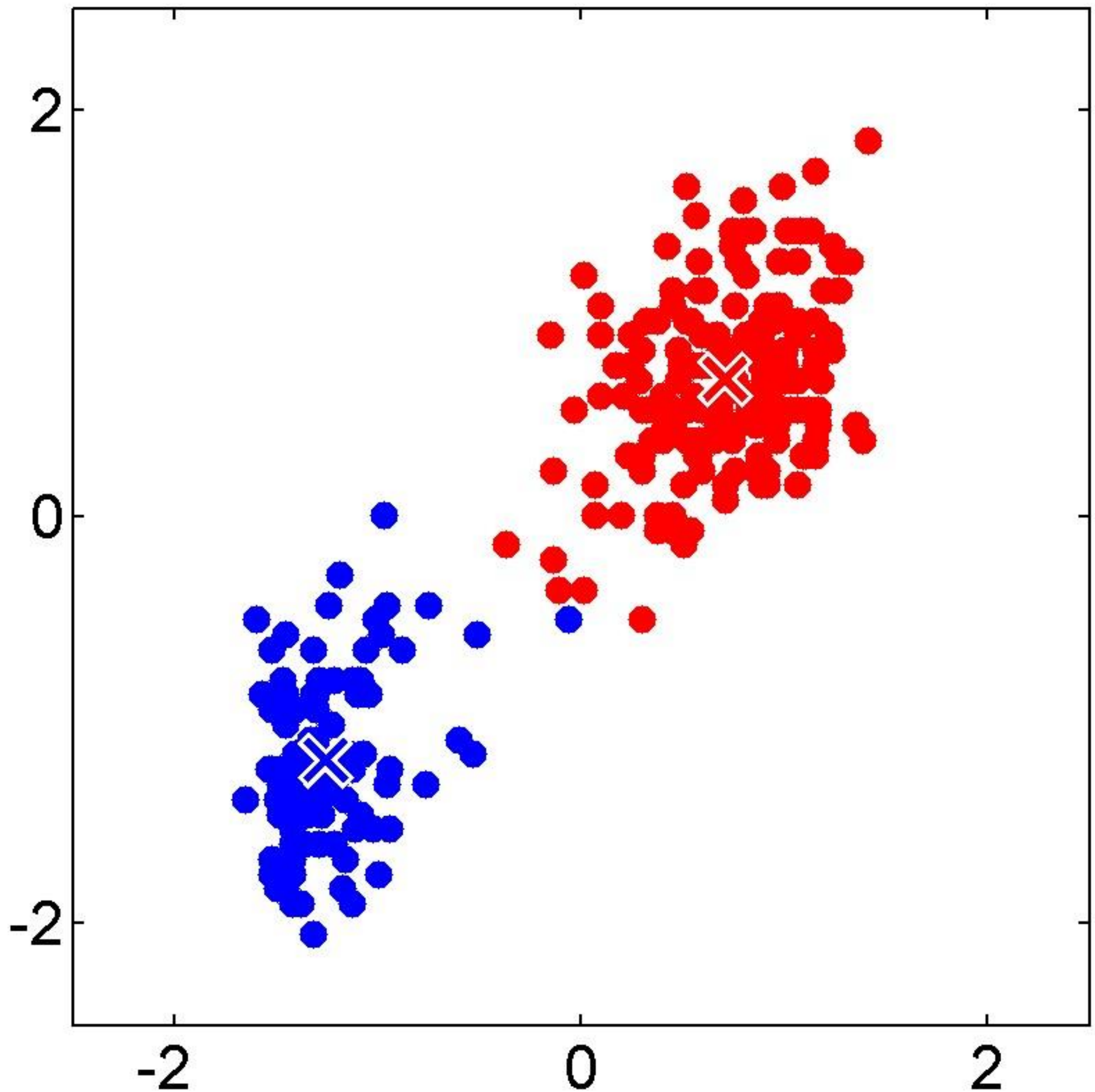

2
0
-2
-2
0
2


2
0
-2
-2
0
2



2
0
-2
-2
0
2


2
0
-2
-2
0
2


2
0
-2
-2
0
2

Responsibilities

- *Responsibilities* assign data points to clusters

$$r_{nk} \in \{0, 1\}$$

such that

$$\sum_k r_{nk} = 1$$

- Example: 5 data points and 3 clusters

$$(r_{nk}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

K-means Cost Function

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

data

responsibilities

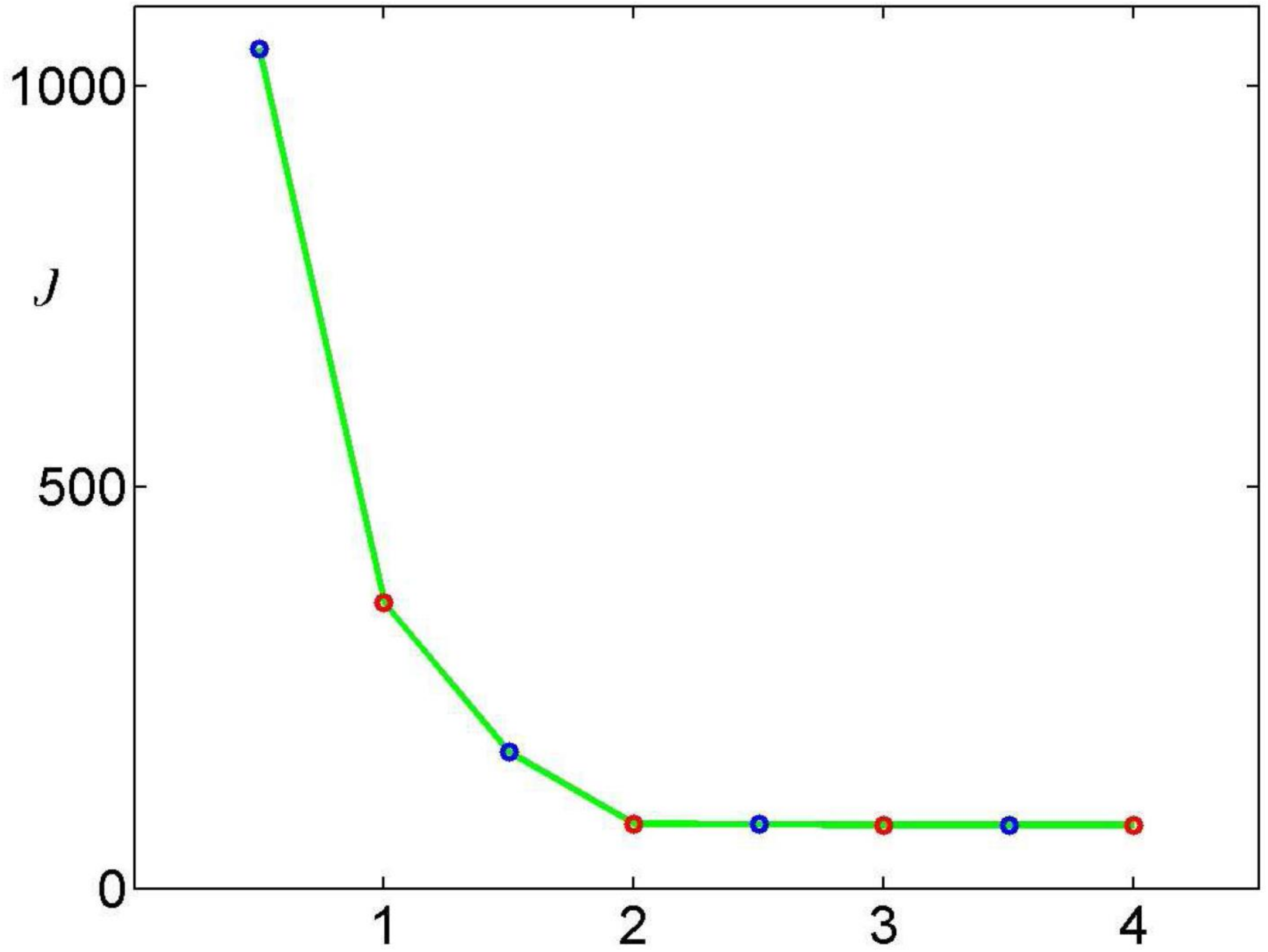
prototypes

Minimizing the Cost Function

- E-step: minimize J w.r.t. r_{nk}
  - assigns each data point to nearest learner
- M-step: minimize J w.r.t. $\boldsymbol{\mu}_k$
  - gives

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{kn} \mathbf{x}_n}{\sum_n r_{kn}}$$

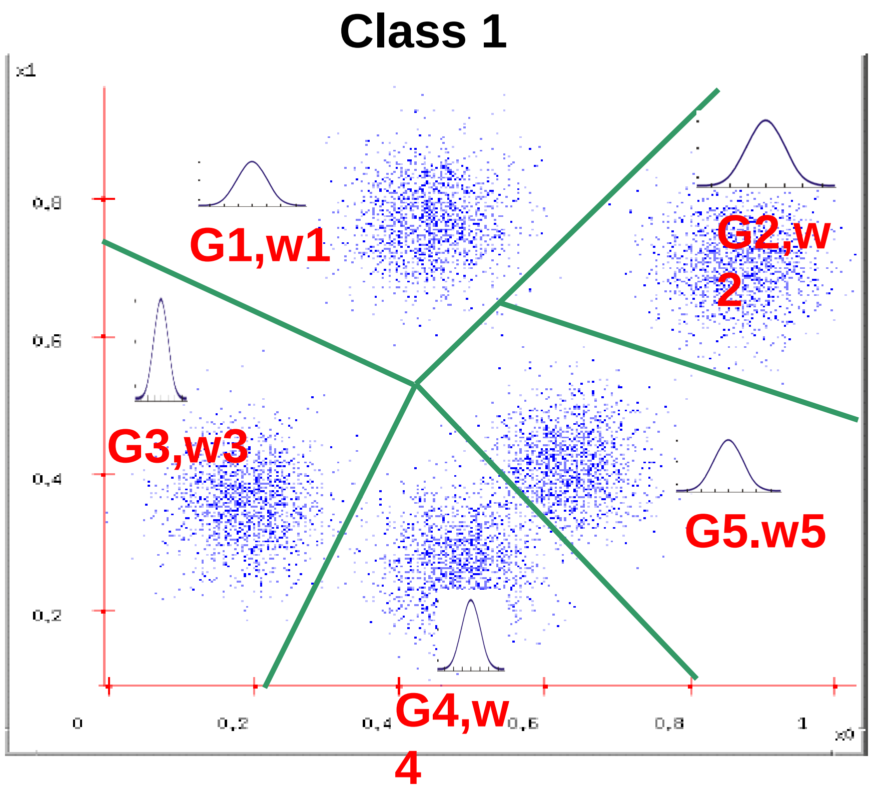
  - each learner set to the mean of points in that cluster
- Convergence guaranteed since there are a finite number of possible responsibility settings.



- How to evaluate K-means clustering results?

Limitations of K-means

- Hard assignments of data points to clusters
  - small shift of a data point can flip it to a different cluster
- Solution: replace 'hard' clustering of K-means with 'soft' probabilistic assignments
- Represents the probability distribution of the data as a *Gaussian Mixture Model (GMM)*



Maximum Likelihood Principle

- To describe the problem in a "probability" way
- Remind: what is probability?

$$p(x) \geq 0 \qquad \int_{-\infty}^{+\infty} p(x)dx = 1$$

- Mapping from distance to probability:

$$p = 0 \quad \Longleftrightarrow \quad ||\mathbf{x}_t - \mathbf{m}|| = +\infty$$
$$p = 1 \quad \Longleftrightarrow \quad ||\mathbf{x}_t - \mathbf{m}|| = 0$$

$$[0, +\infty) \Longleftrightarrow [0, 1]$$

- But not all positive monotonic functions are ok, why?
  – One function: Gaussian distribution

Gaussian Distribution

- Multivariate Gaussian

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

where $\boldsymbol{\Sigma}$ is the covariance matrix, and $\boldsymbol{\mu}$ is the mean vector.

d is the dimension.

- In 1-dimension case:

$$G(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$$

where σ^2 is the variance, and μ is the mean value,

dimension $d = 1$.

Recall: Likelihood Function

- Data set

$$D = \{\mathbf{x}_n\} \qquad n = 1, \ldots, N$$

- Consider first a single Gaussian
- Assume observed data points generated independently

$$p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Viewed as a function of the parameters, this is known as the *likelihood function*

Recall: Maximum Likelihood Solution

Set the parameters by maximizing the likelihood function

Equivalently maximize the log likelihood

$$\ln p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{Nd}{2}\ln(2\pi) - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})$$

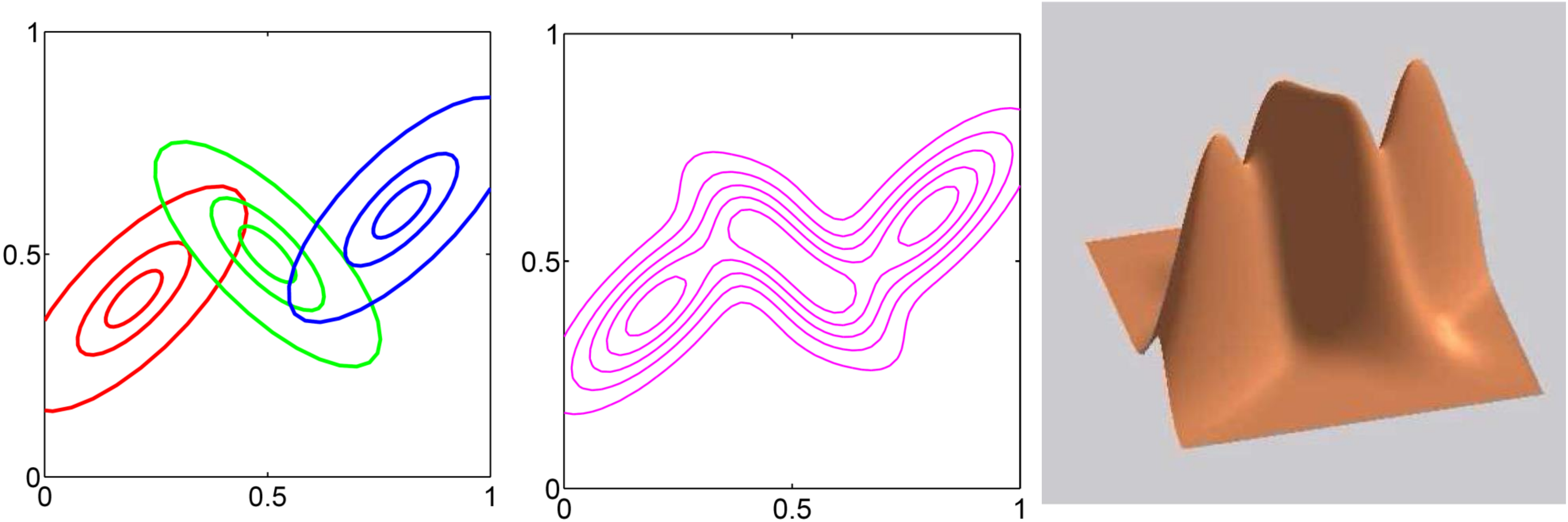
- Maximizing w.r.t. the mean gives the *sample mean*

$$\boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

- Maximizing w.r.t covariance gives the *sample covariance*

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

Example: Mixture of 3 Gaussians



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

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

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

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

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

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

Maximum Likelihood for the GMM

- 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

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

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
- Each EM cycle guaranteed not to decrease the likelihood

- Initialize $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and π_k and evaluate log-likelihood with these
- E Step: $\gamma(z_{nk}) = \dfrac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$

- M Step: $\boldsymbol{\mu}_k^{\text{new}} = \dfrac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n,$

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

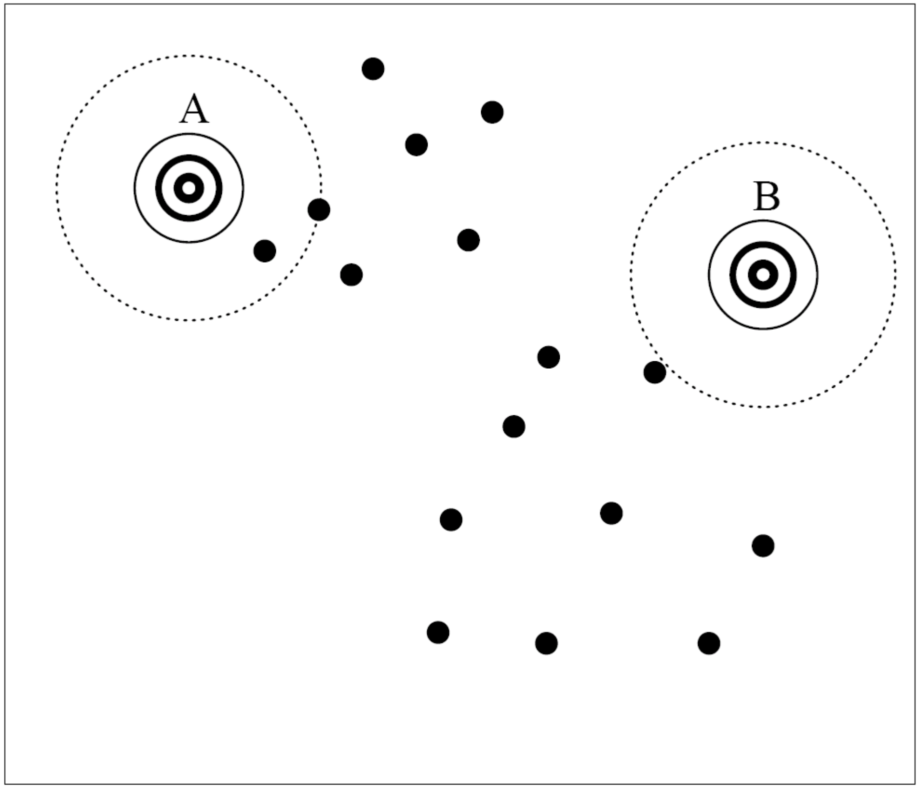
$$\pi_k^{\text{new}} = \frac{N_k}{N} \quad \text{with } N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

- Evaluate log-likehood

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

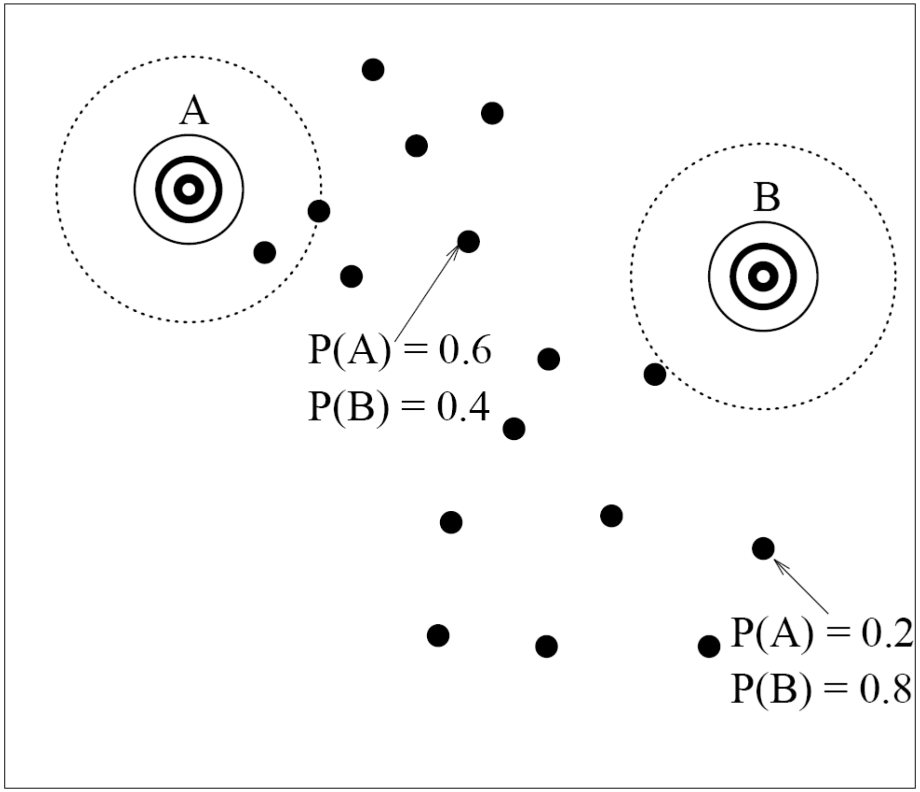
Processing : EM Initialization

- Initialization :
  - Assign random value to parameters



Processing : the E-Step (1/2)

– Expectation :
  - Pretend to know the parameter
  - Assign responsibilities of Gaussians to each data point

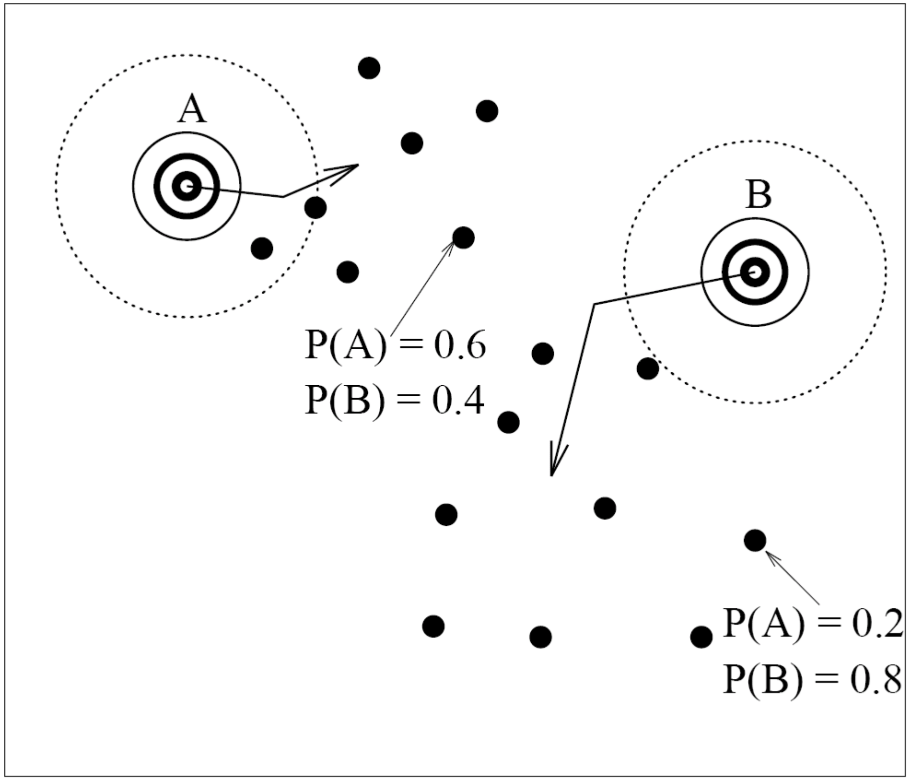


Processing : the E-Step (2/2)

- Competition of Hypotheses
  - Compute the expected values of Pij of hidden *indicator variables*.
- Each gives membership weights to data point
- Normalization
- Weight = relative likelihood of class membership

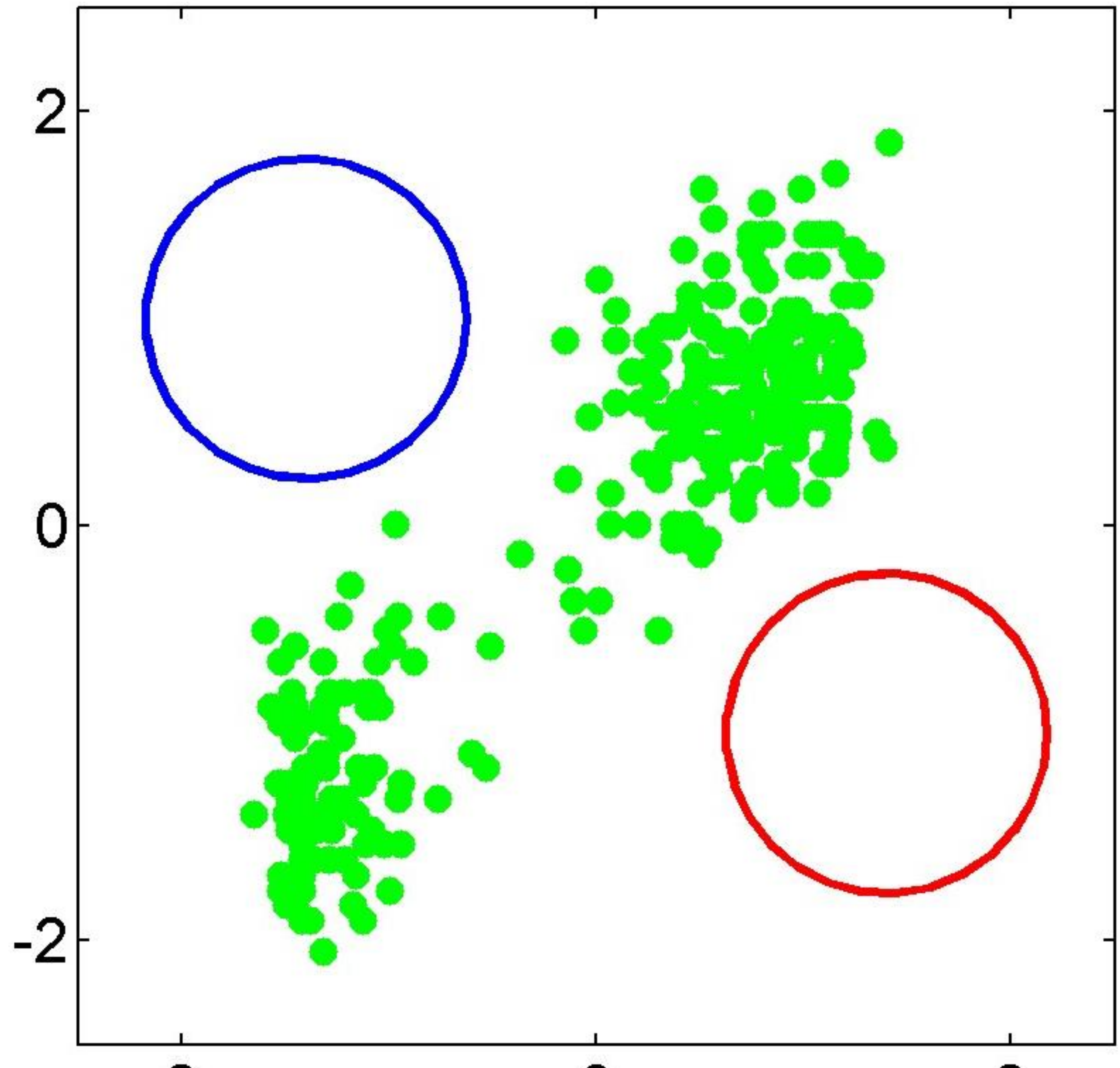
Processing : the M-Step (1/2)

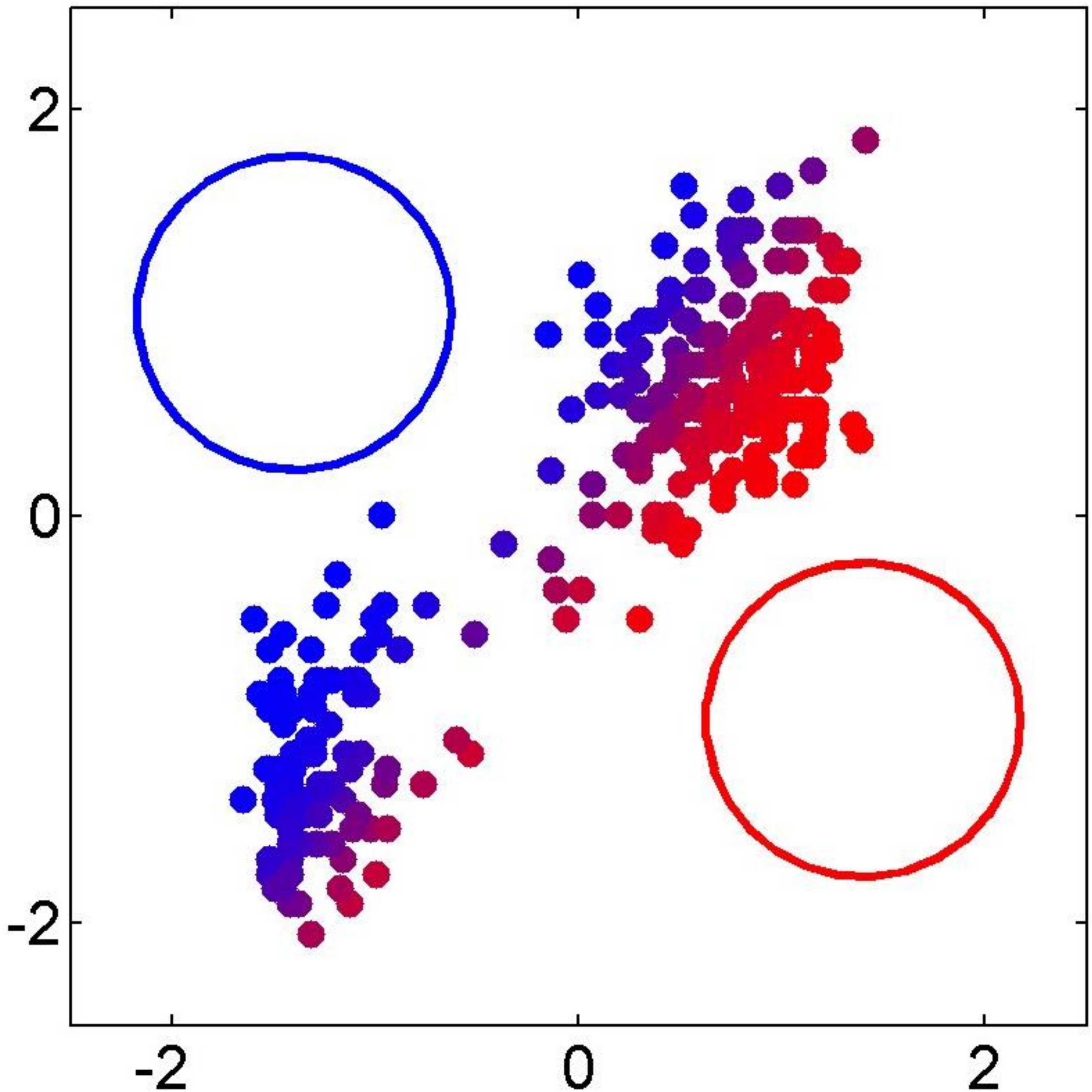
– Maximization :
  - Fit the parameter to its set of points

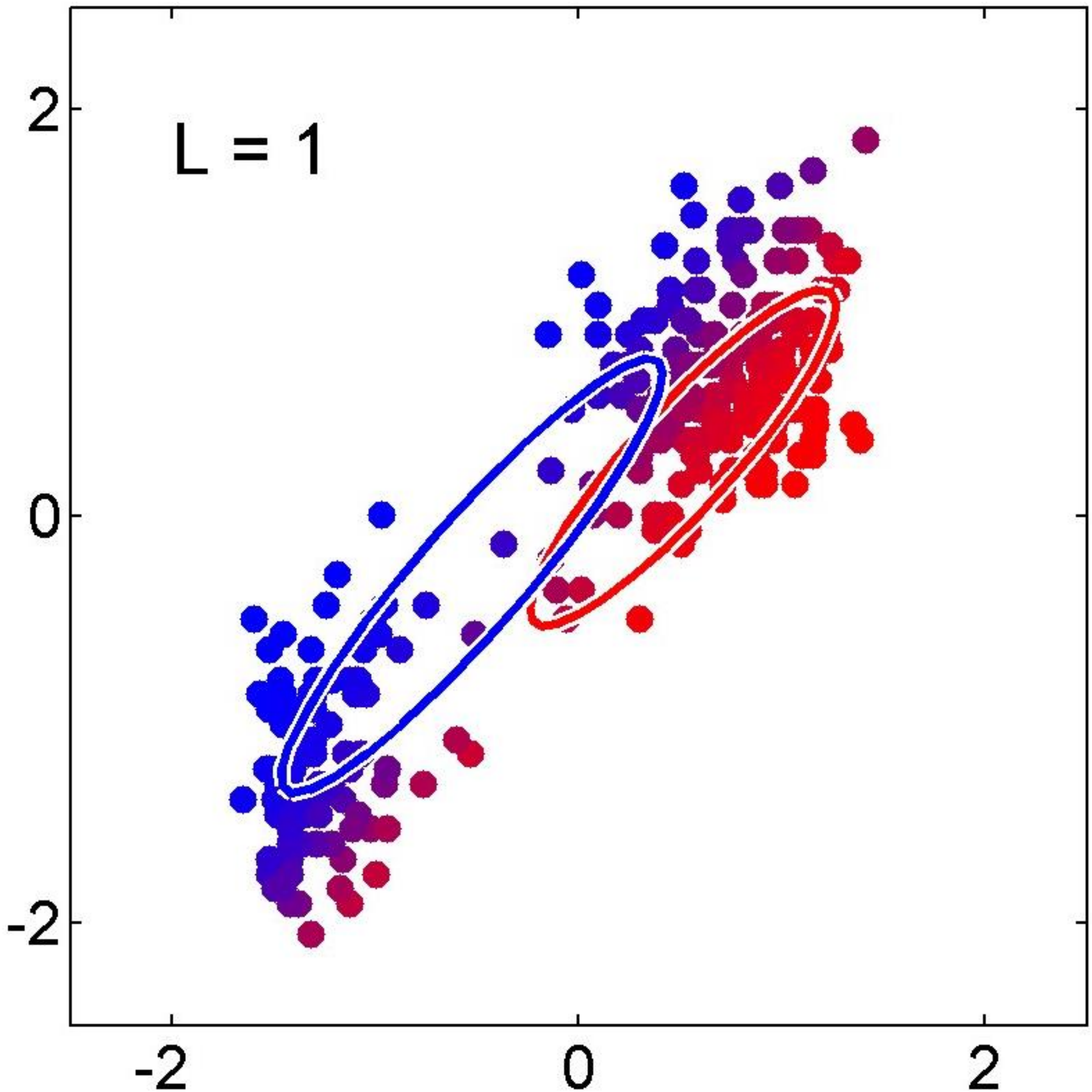


Processing : the M-Step (2/2)

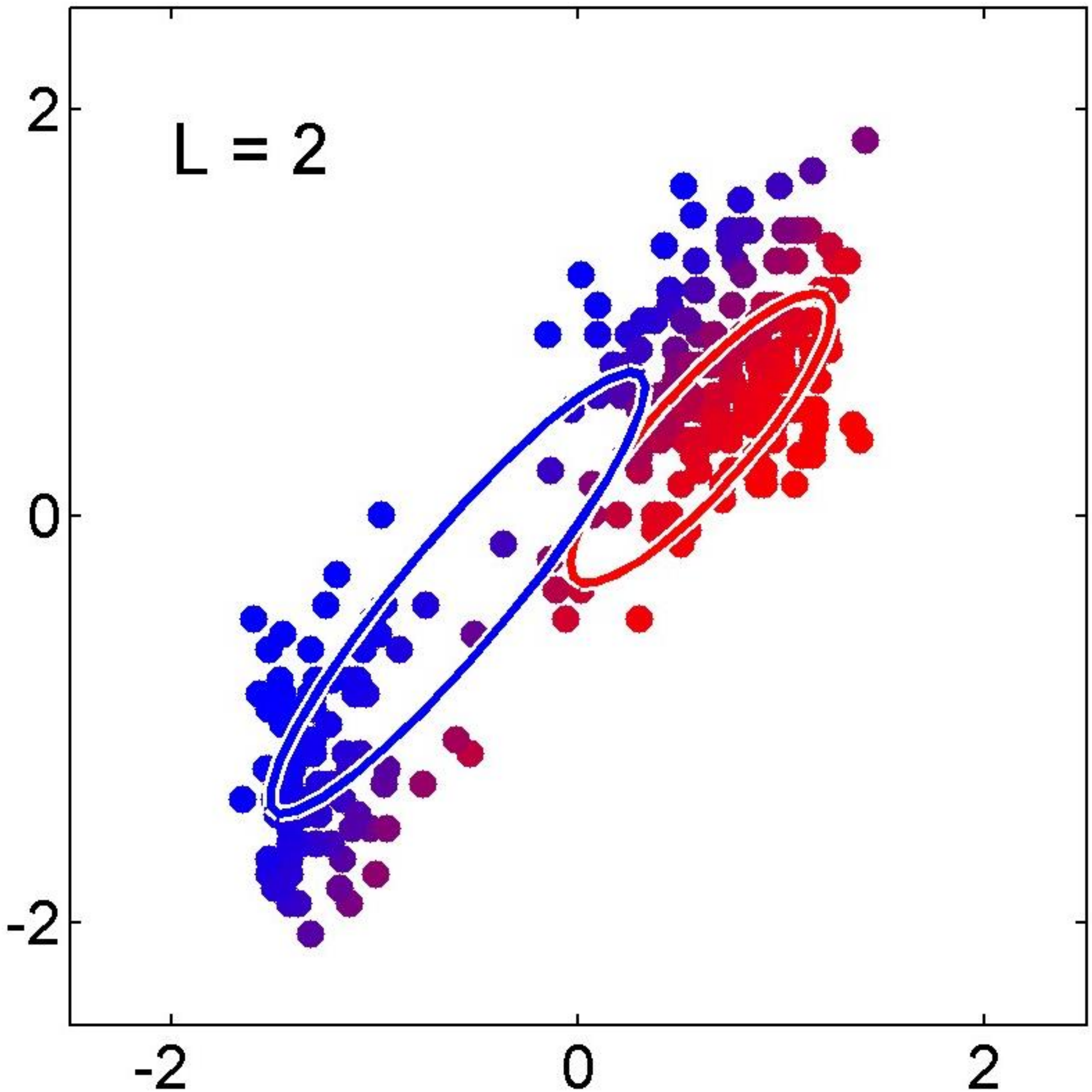
- For each Gaussian learner
  - Find the new value of parameters to maximize the *log likelihood*
  - Based on
    - Weight of points in the class
    - Location of the points
  - Gaussians are *pulled* toward data

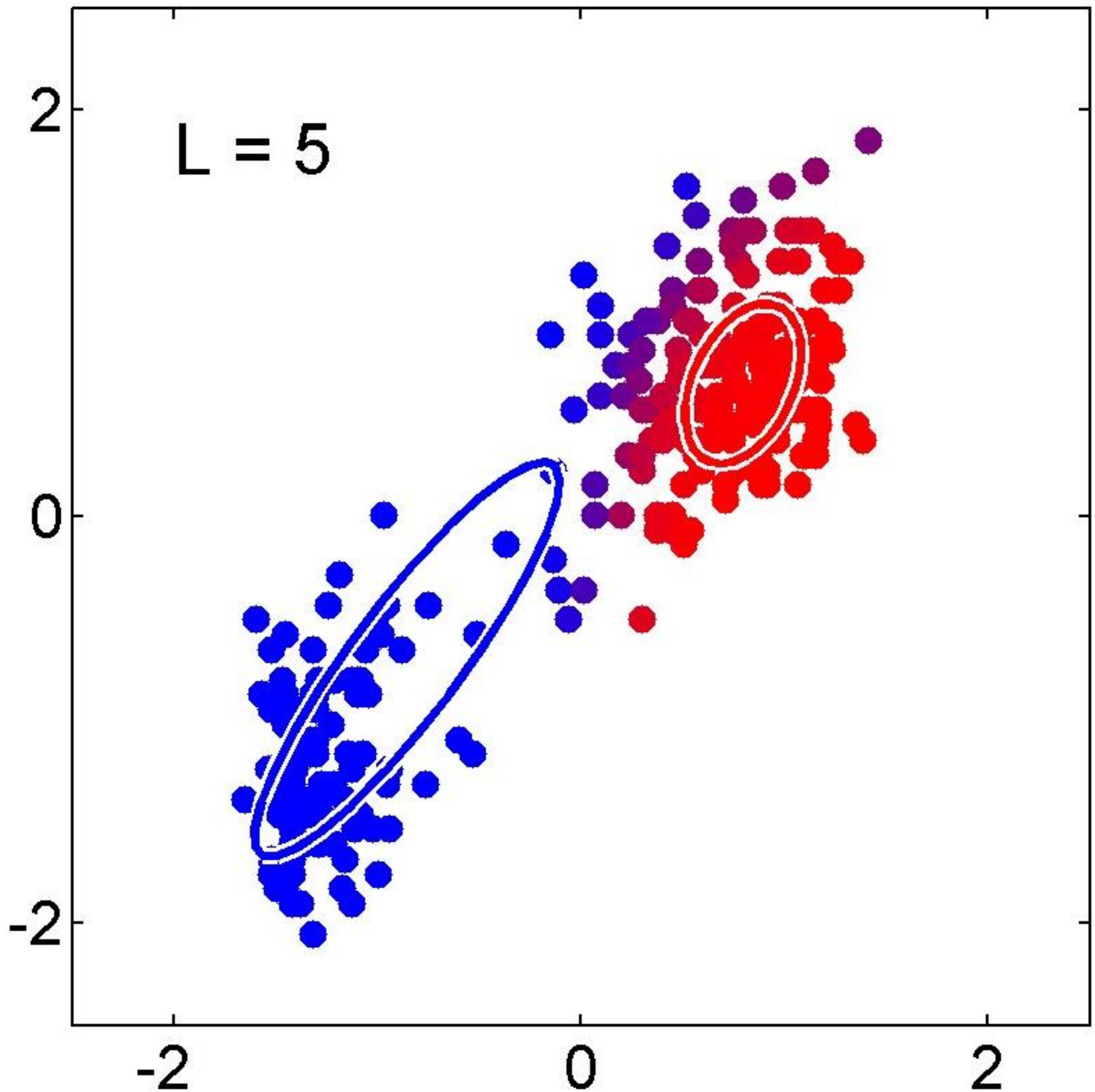

2
0
-2
-2
0
2


2
0
-2
-2
0
2

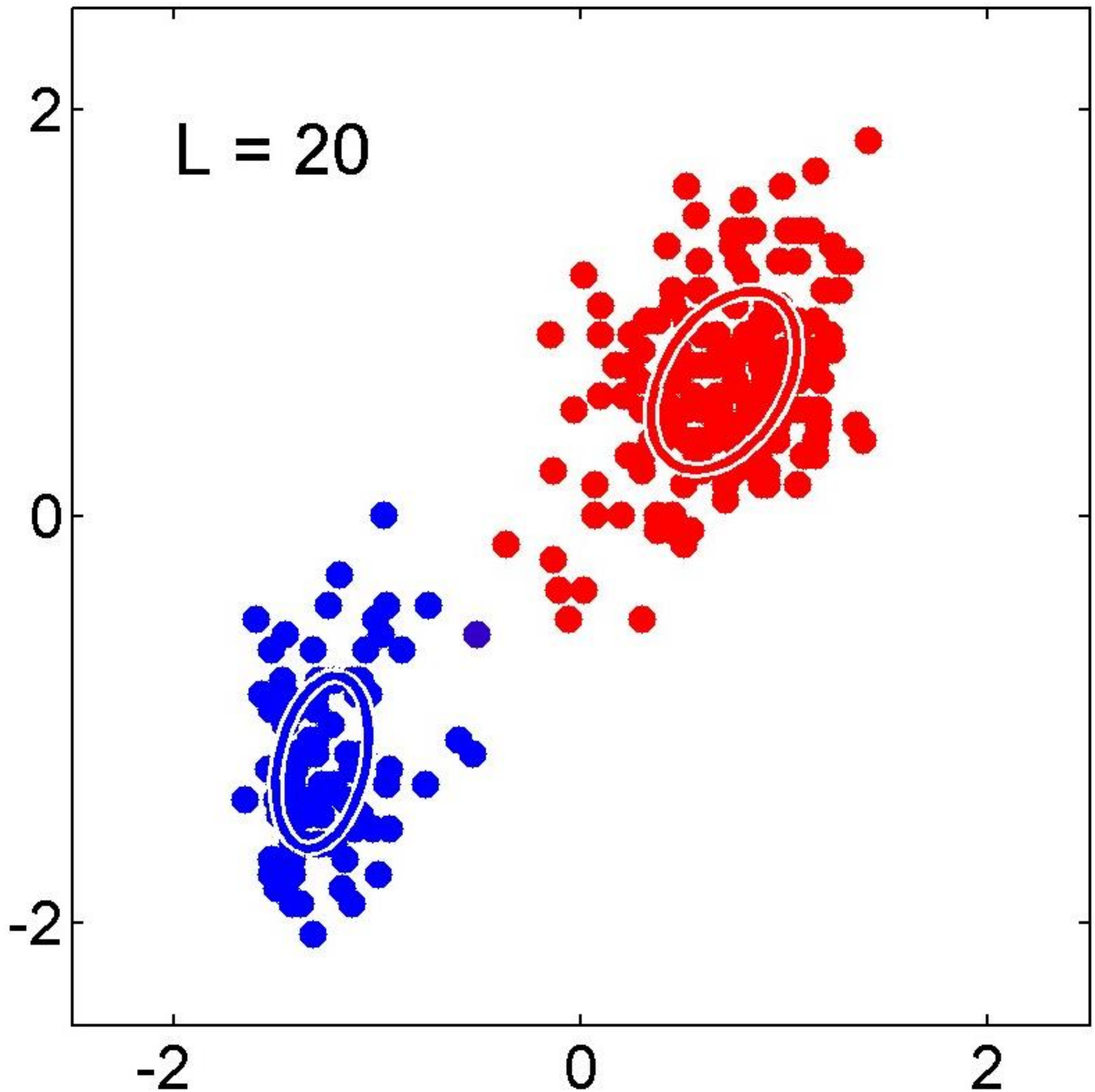


L = 1
2
0
-2
-2
0
2


L = 2
2
0
-2
-2
0
2



L = 5
2
0
-2
-2
0
2


L = 20
2
0
-2
-2
0
2

Challenges

- Can you try to obtain why K-means and EM algorithm on GMM have that form? Given the target function J:
  - K-means: minimize

$$J = \sum_{t=1}^{N} \sum_{l=1}^{K} r_{tl} \| \mathbf{x}_t - \boldsymbol{\mu}_l \|^2$$

  - EM: maximize

$$J = \prod_{t=1}^{N} [\sum_{l=1}^{K} \alpha_l G(\mathbf{x}_t \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)]$$

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