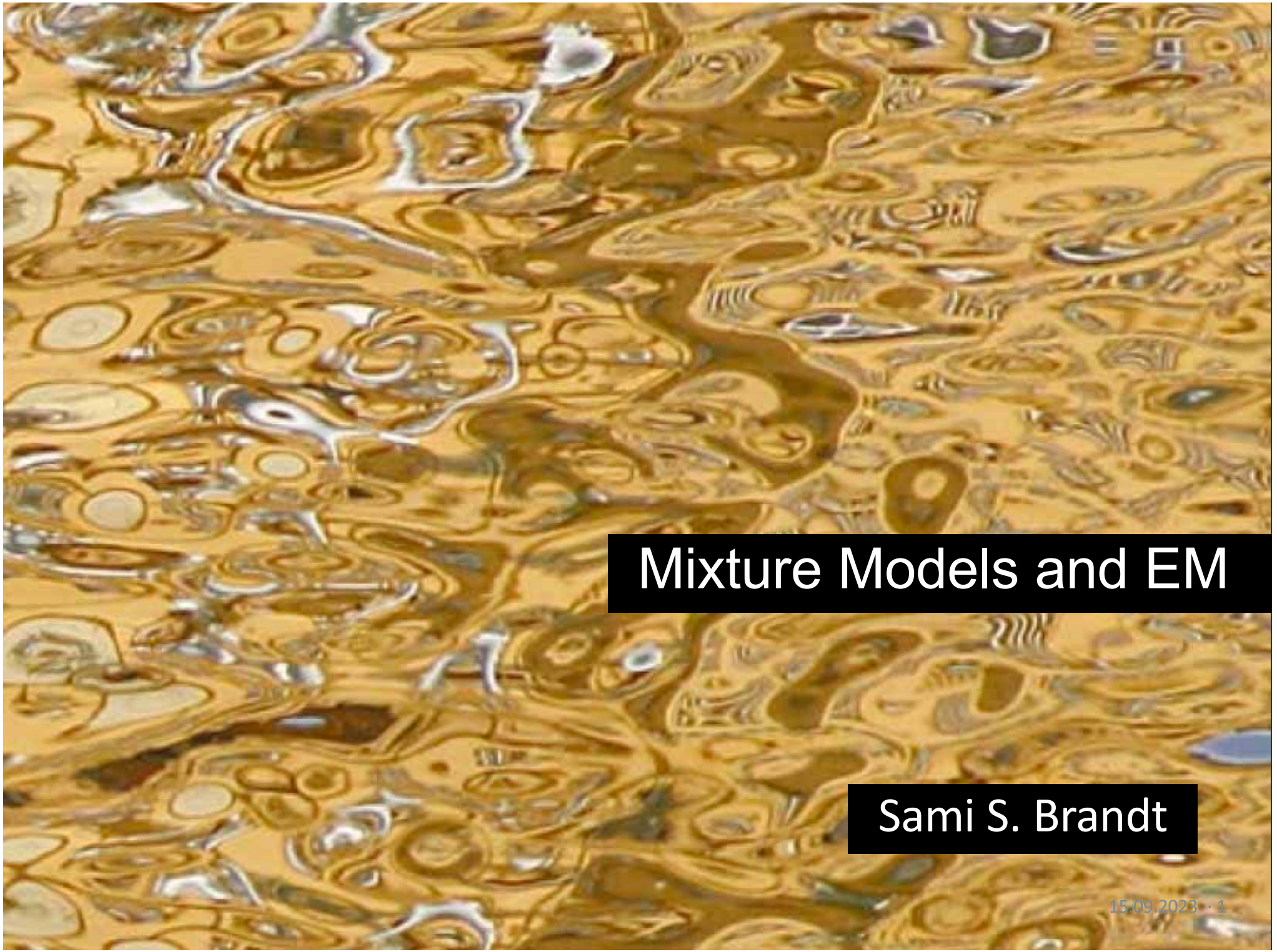


Mixture Models and EM

Sami S. Brandt

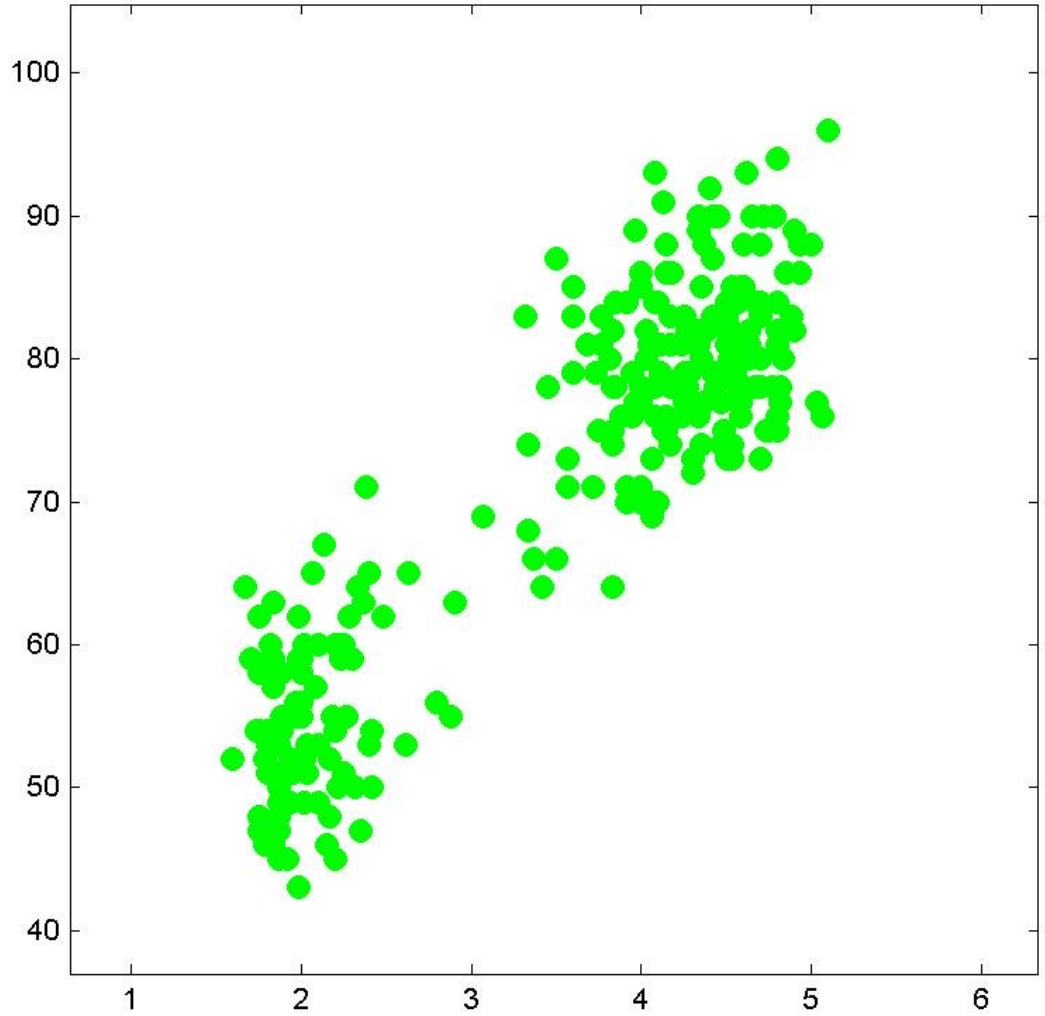
Learning Objectives for today

- Explain the difference between fuzzy and hard clustering
- Explain the principle of Expectation Maximization and relate the K-means algorithm to it
- Relate Maximum Likelihood principle and the EM algorithm
- Fit Gaussian Mixture Models

Old Faithful Geyser Data Set

The Data Set

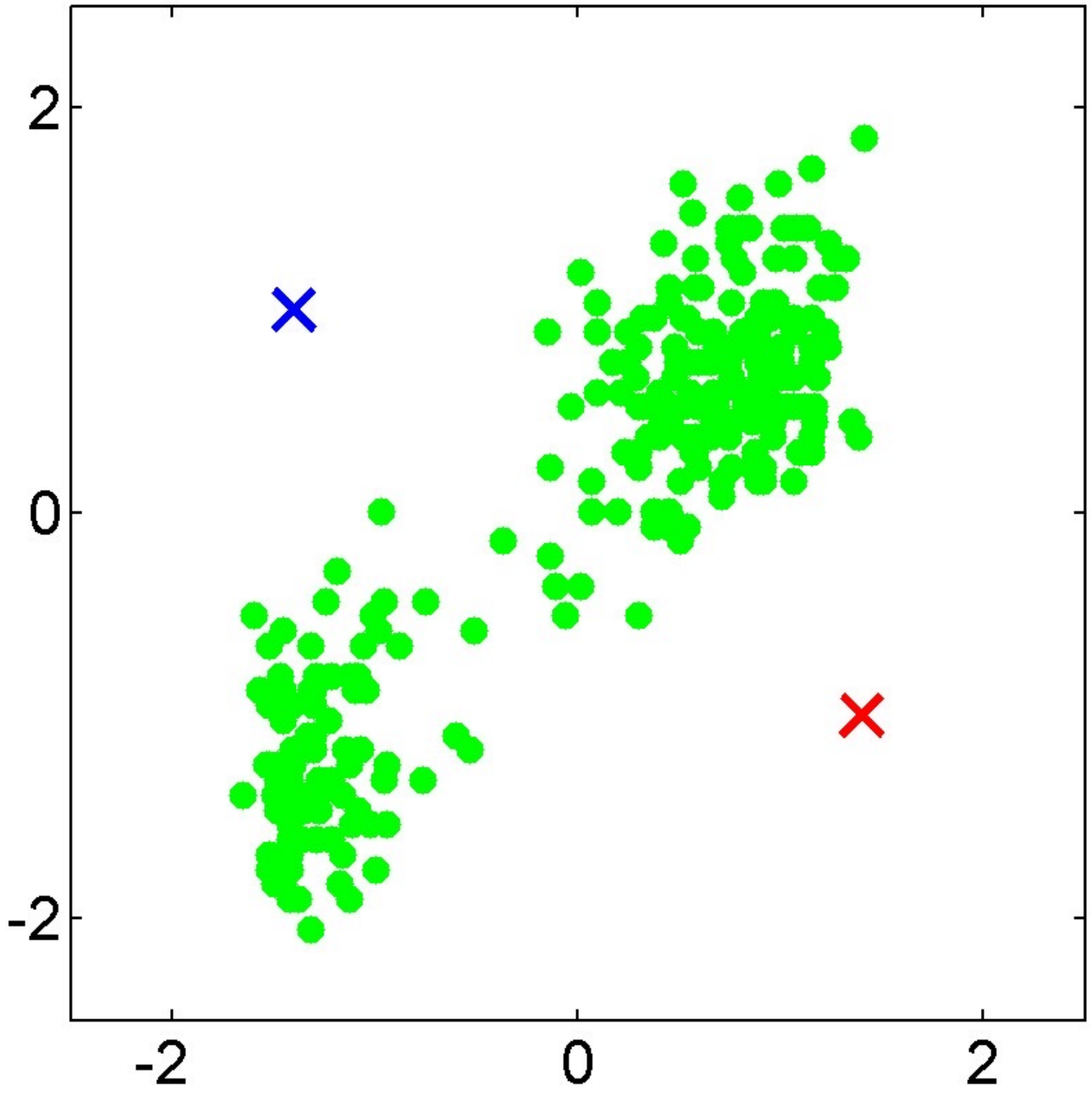
Time between eruptions (minutes)

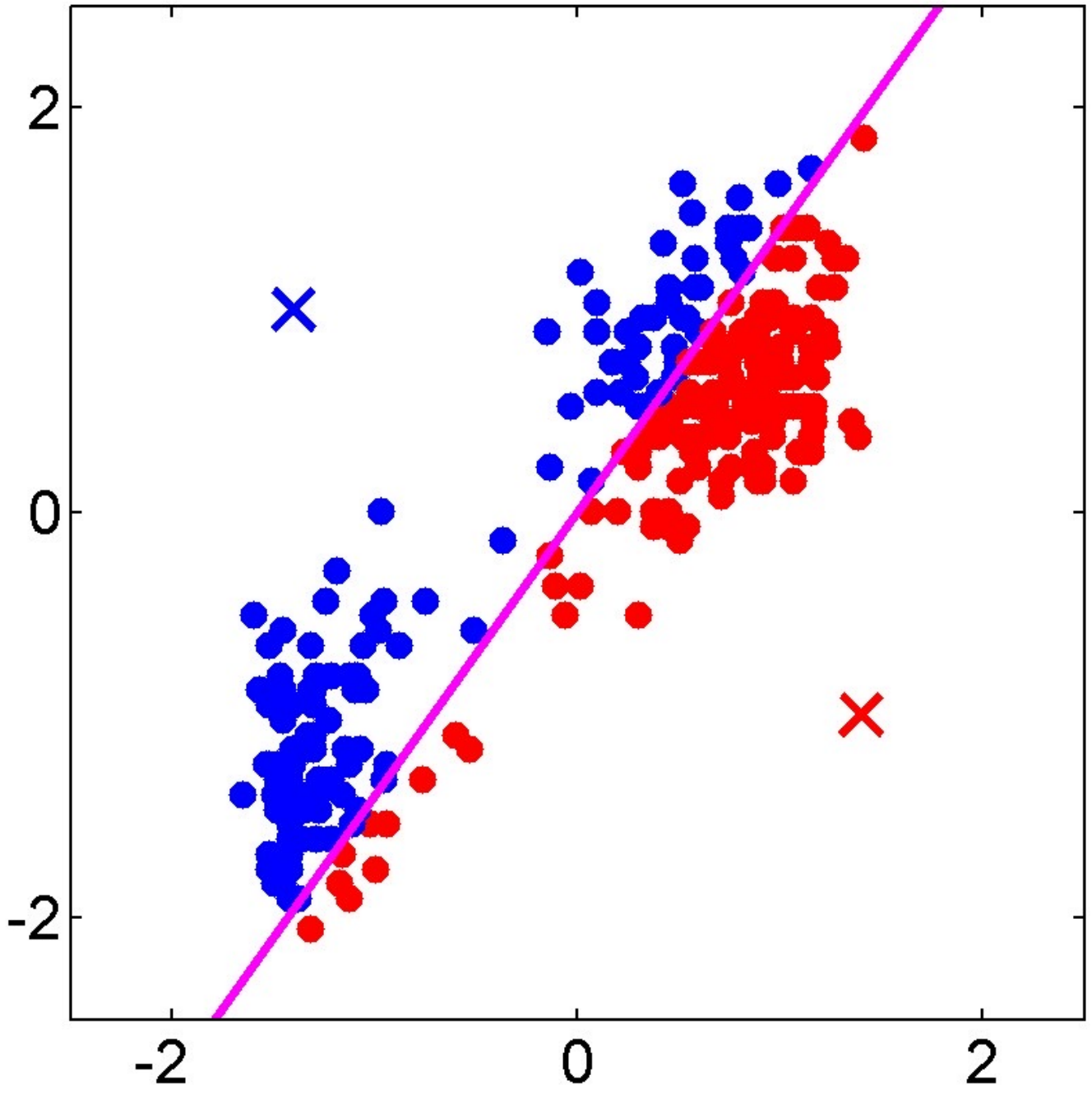


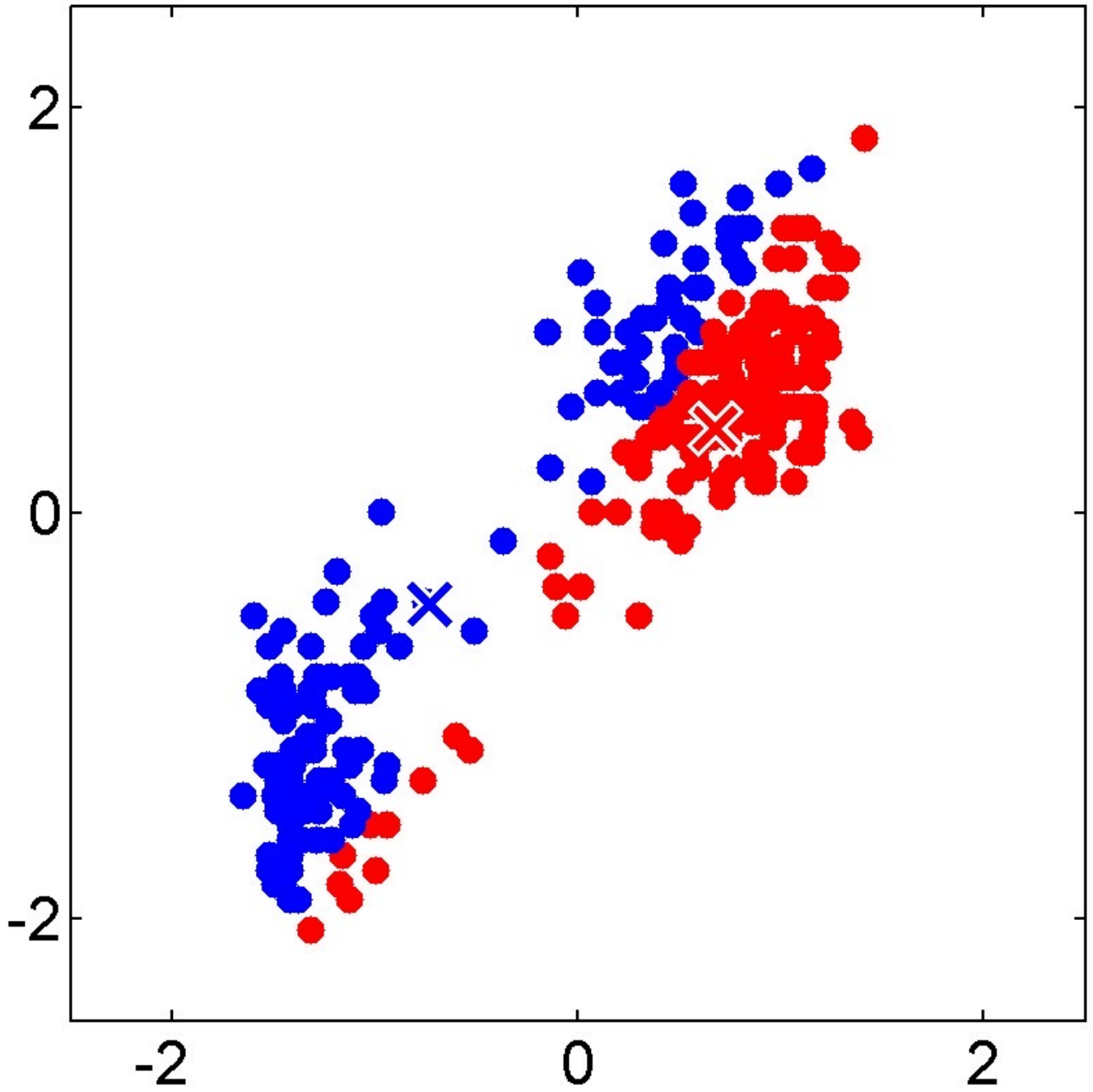
Duration of eruption (minutes)

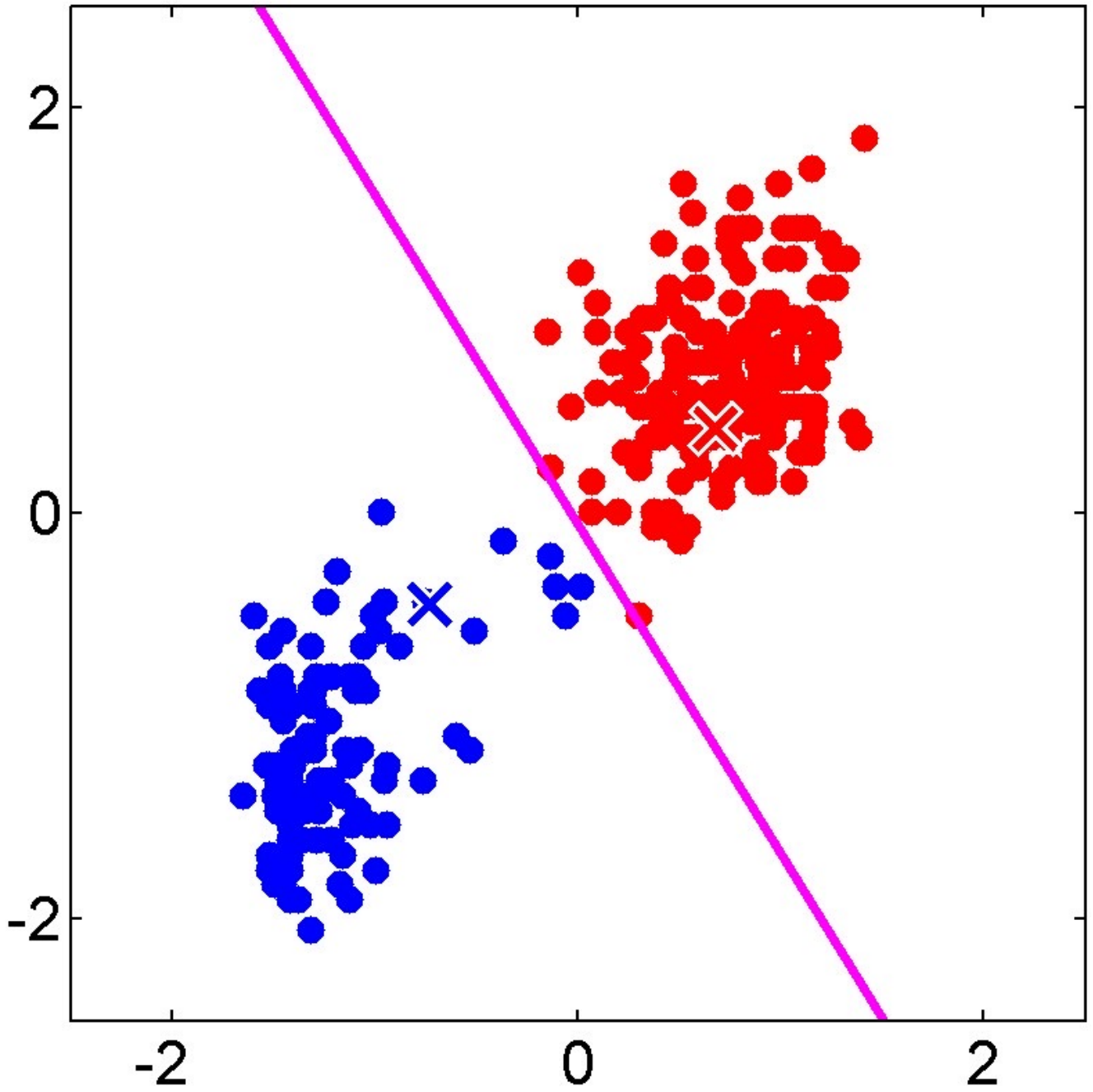
K-means Algorithm

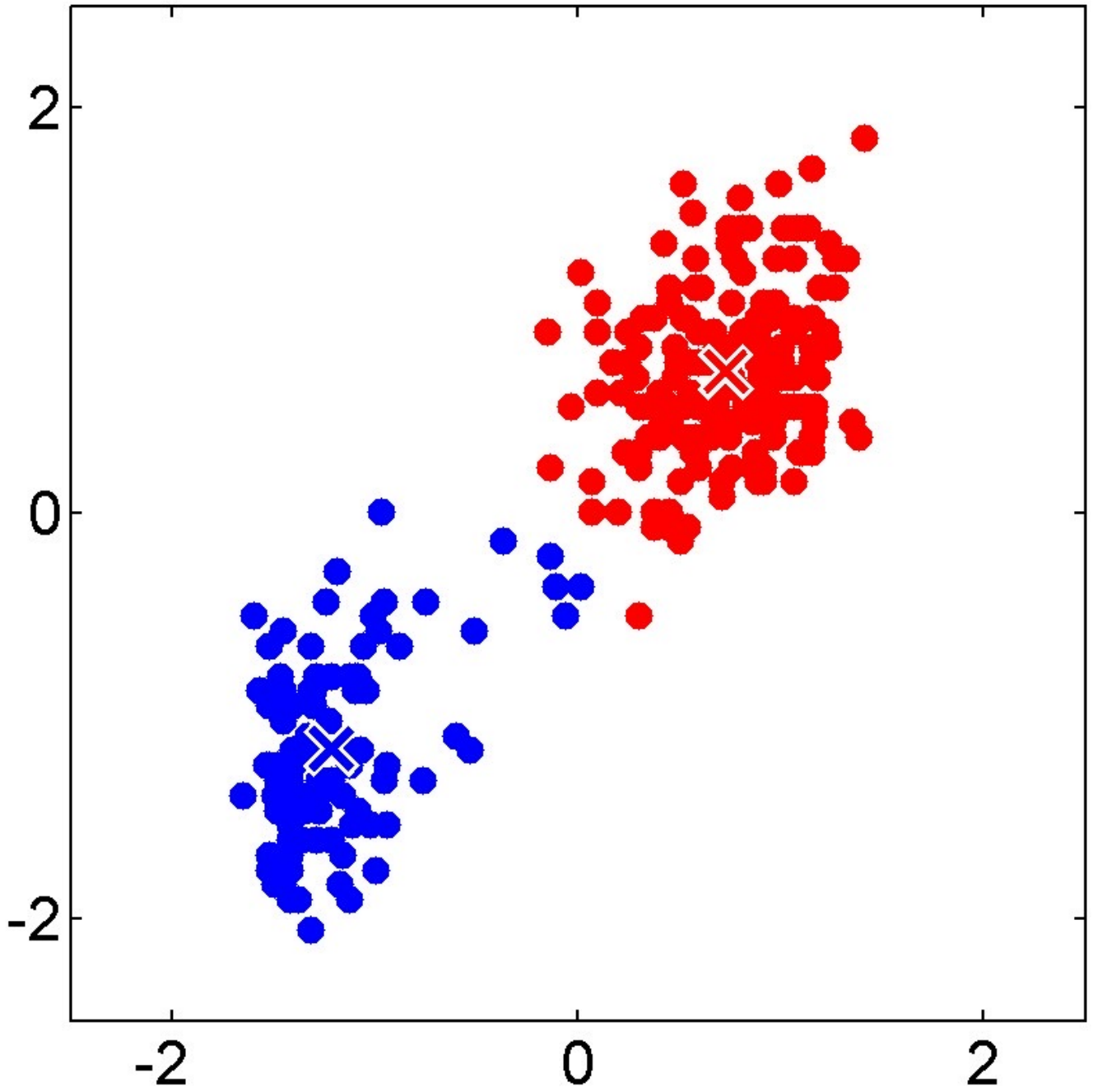
- Goal: represent a data set in terms of *K* clusters each of which is summarized by the prototype $\boldsymbol{\mu}_k$
- Initialize prototypes, then iterate between two phases:
  - **E-step**: assign each data point to nearest prototype
  - **M-step**: update prototypes to be the cluster means
- Simplest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

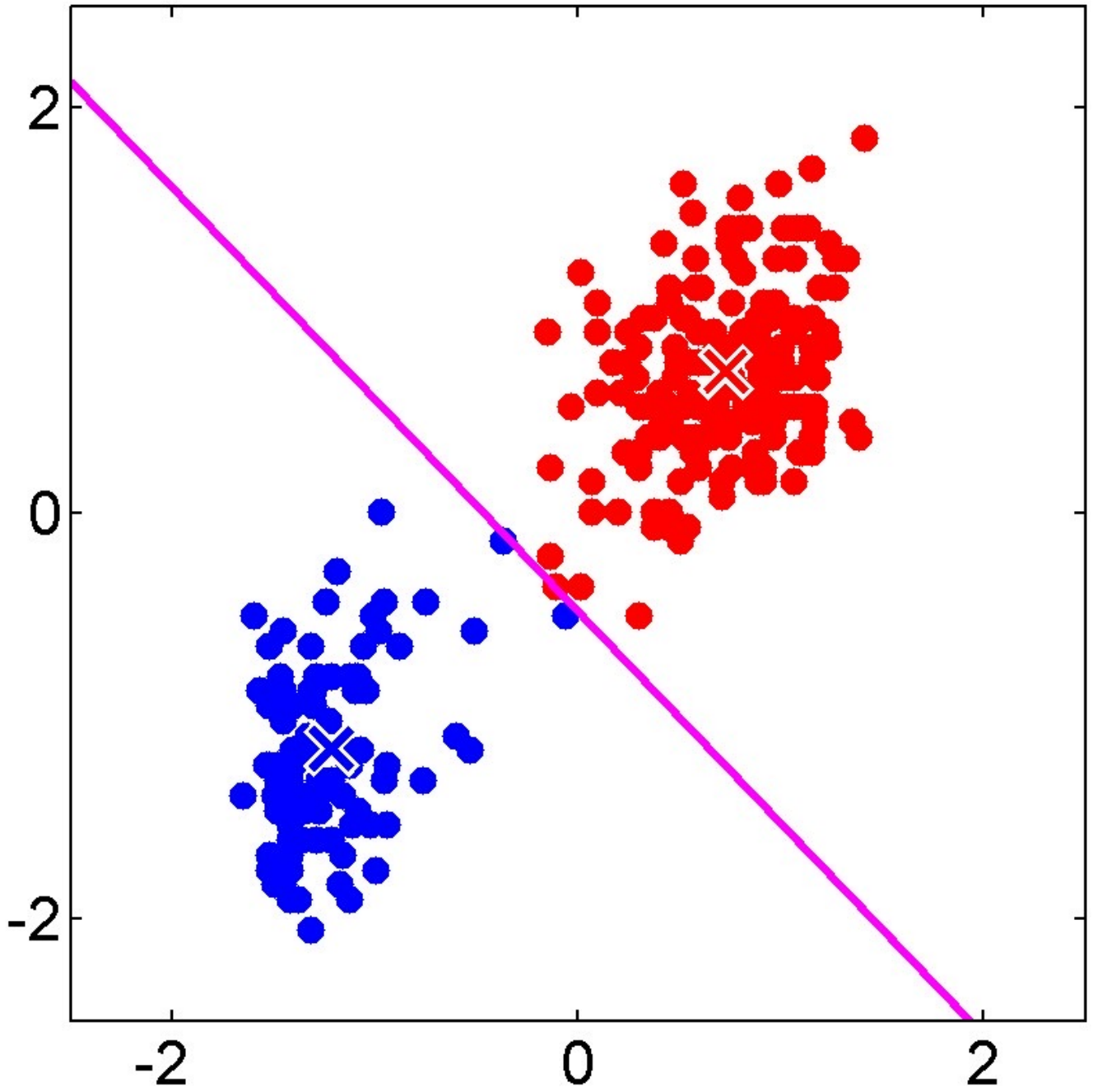

2
0
-2
-2
0
2

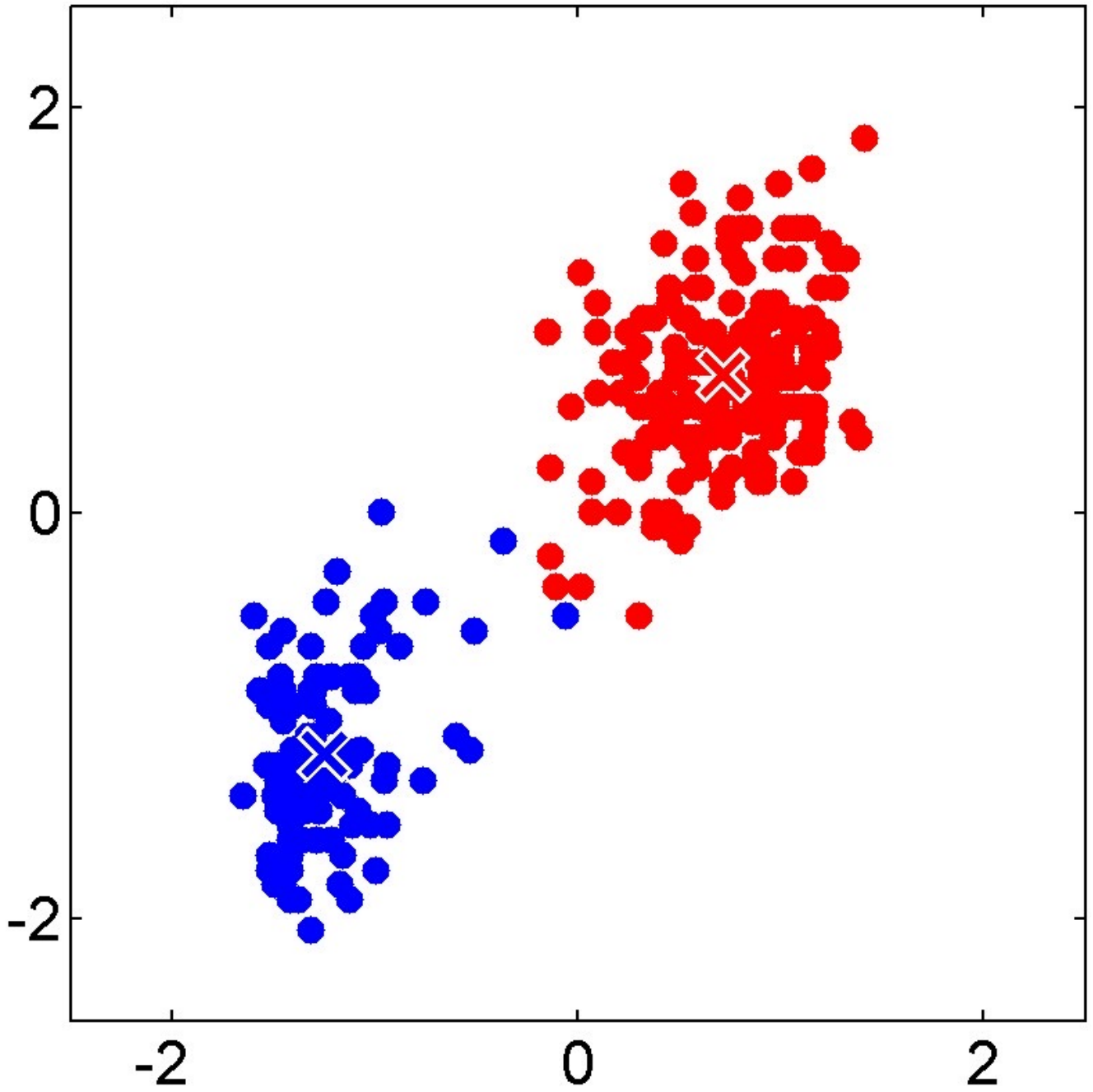

2
0
-2
-2
0
2

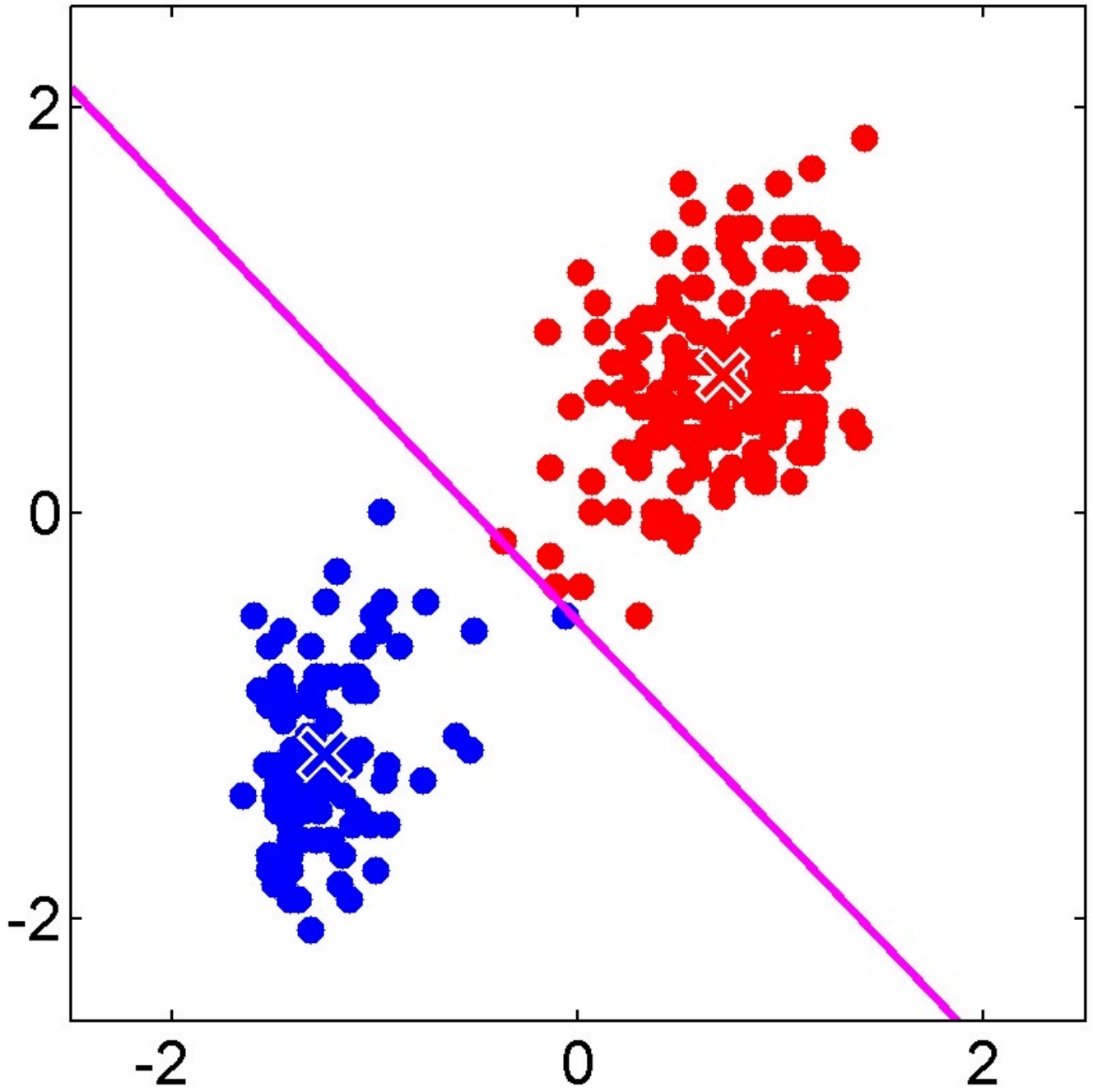

2
0
-2
-2
0
2

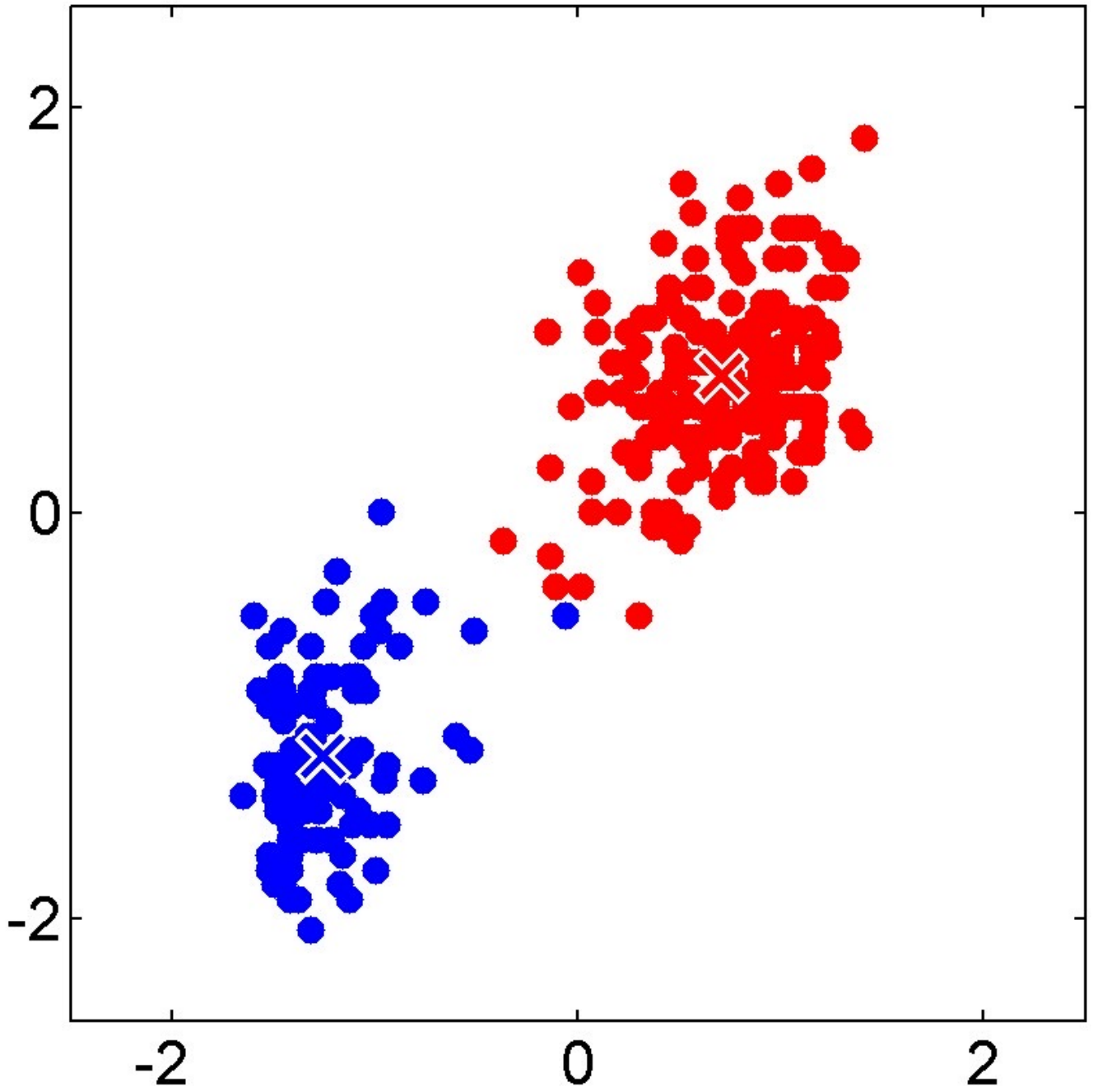

2
0
-2
-2
0
2


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

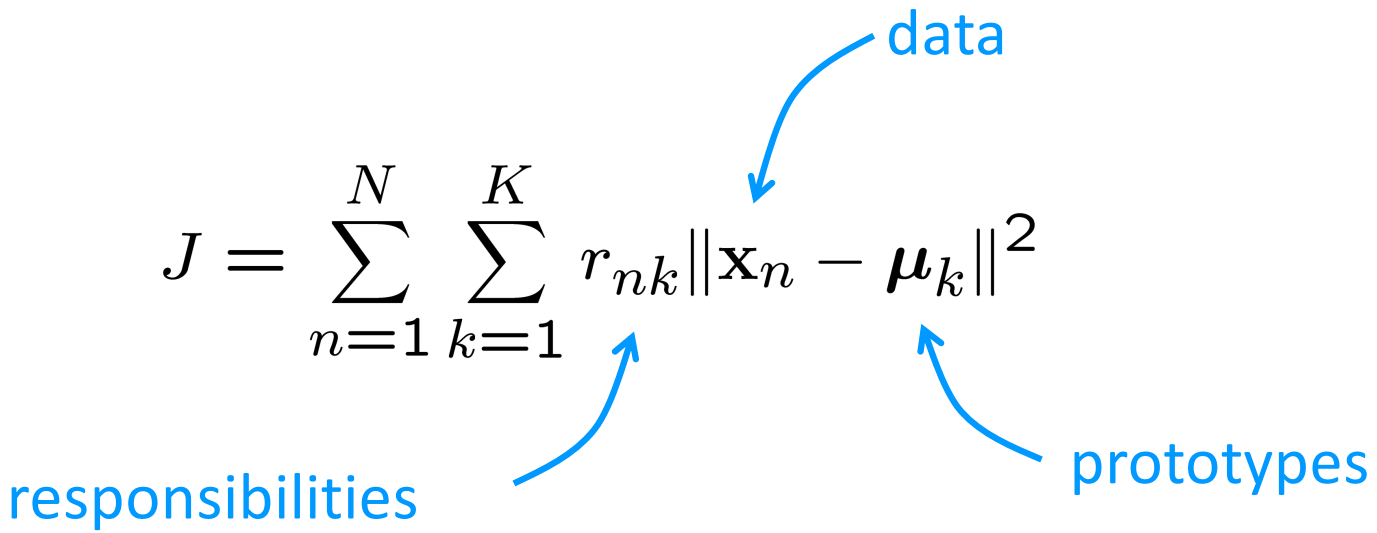
- *Responsibilities* $r_{nk} \in \{0, 1\}$
- assign data points to clusters such that

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

- 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



Minimizing the Cost Function

- **E-step:** minimize J w.r.t. r_{nk}

  assigns each data point to nearest prototype
- **M-step:** minimize J w.r.t $\boldsymbol{\mu}_k$

  yields

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

  each prototype set to the mean of points in that cluster
- Local convergence guaranteed since there is a finite number of possible settings for the responsibilities
- Globally? – No.

Convergence of K-means

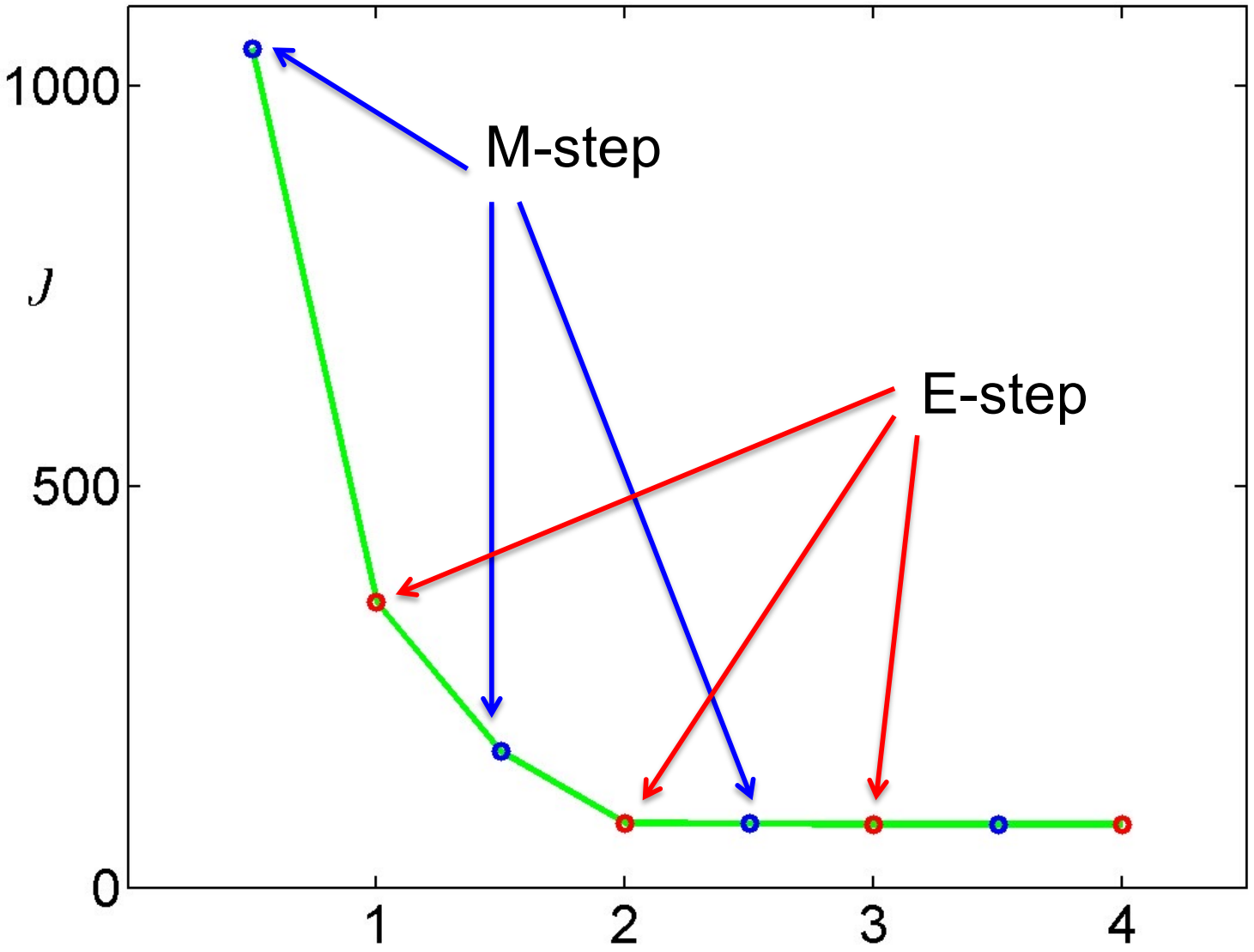
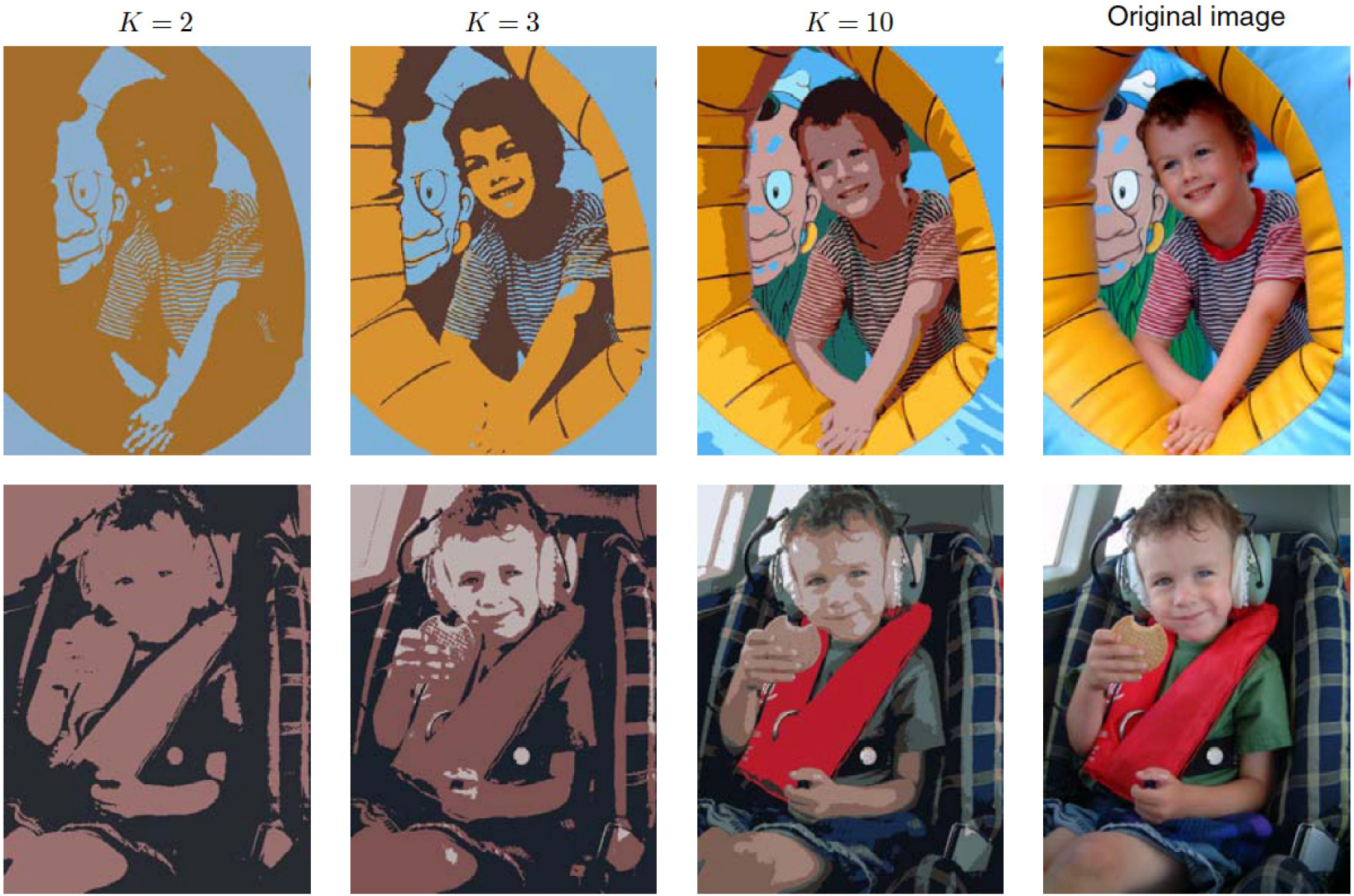


Image Segmentation by K-means



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*

Multivariate Gaussian Distribution (recap)

- Multivariate Gaussian pdf

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

mean

covariance

- Precision is the inverse of the covariance

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

- In one-dimensional space

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

Likelihood Function for a Gaussian (recap)

- Data set

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

- 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 the *likelihood function*

Maximum Likelihood Solution with a Gaussian(recap)

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

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

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

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

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 constraints imply

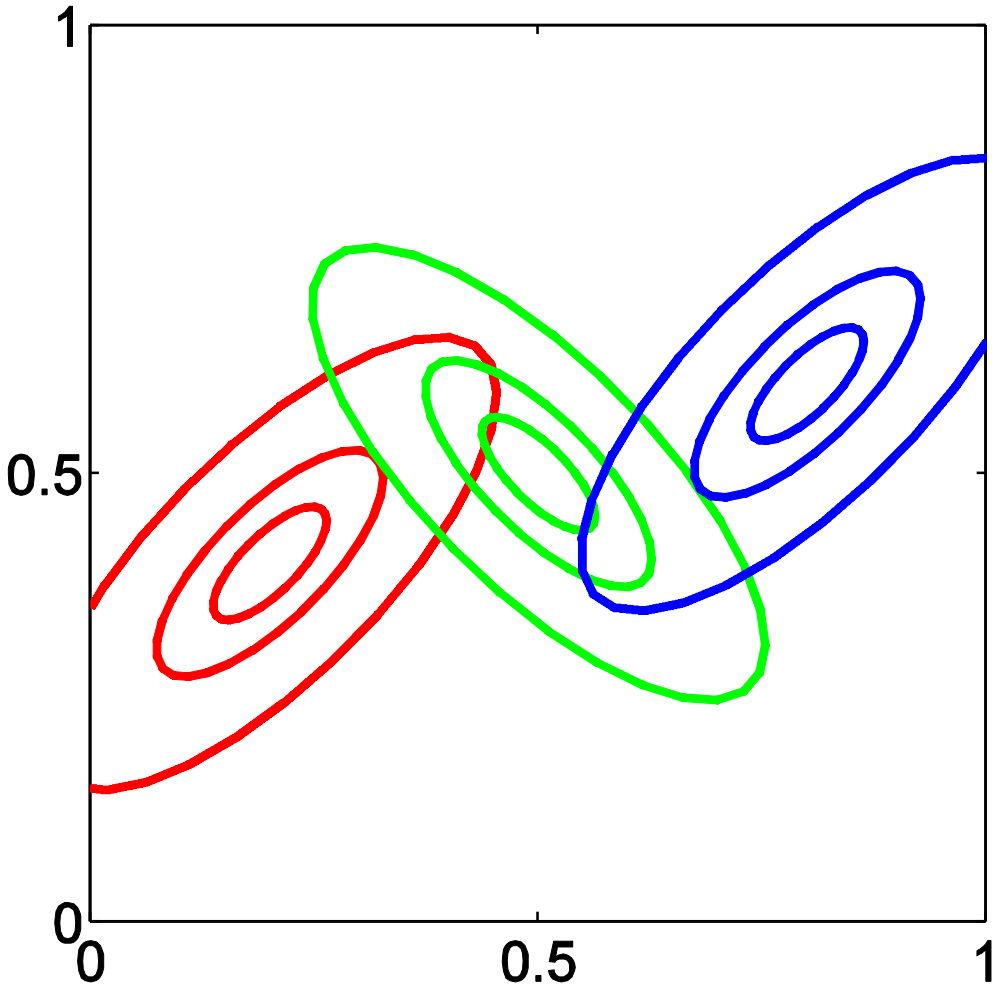
$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leqslant \pi_k \leqslant 1$$

- The mixing coefficients can be seen as prior probabilities

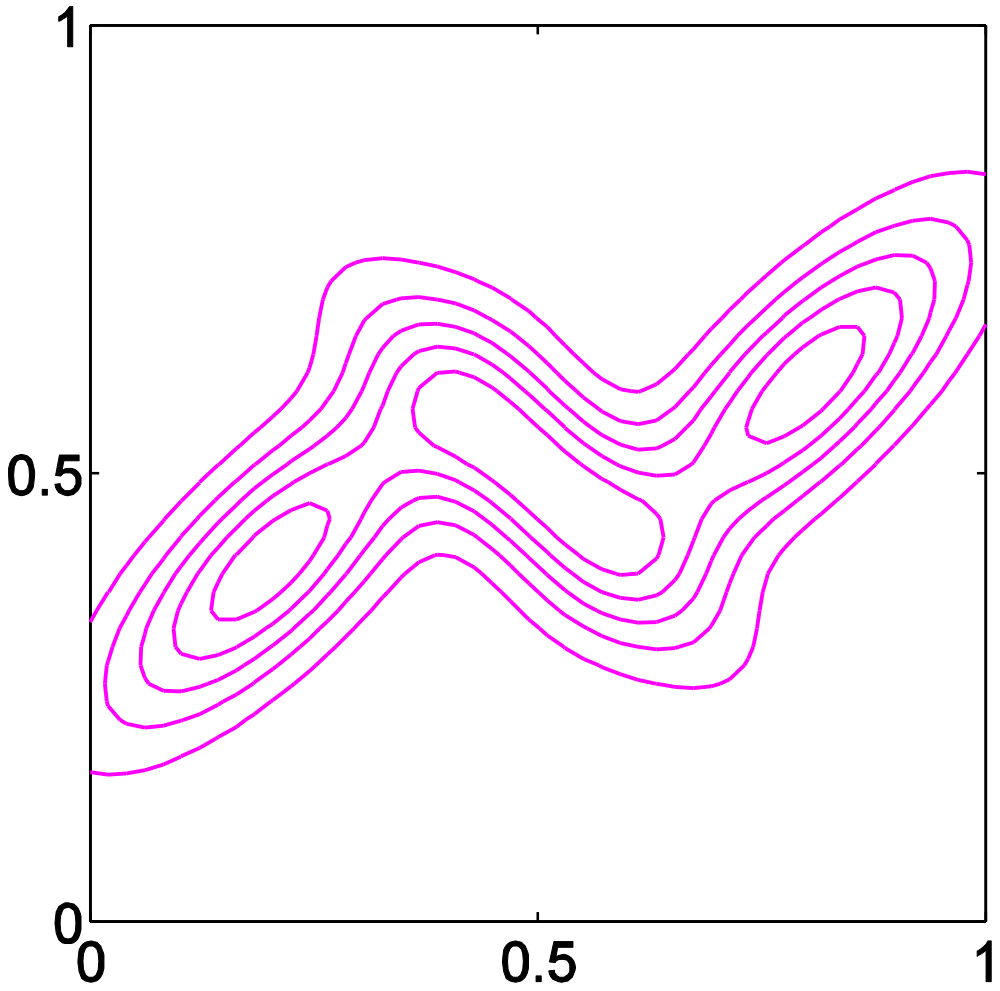
$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k) p(\mathbf{x}|k)$$

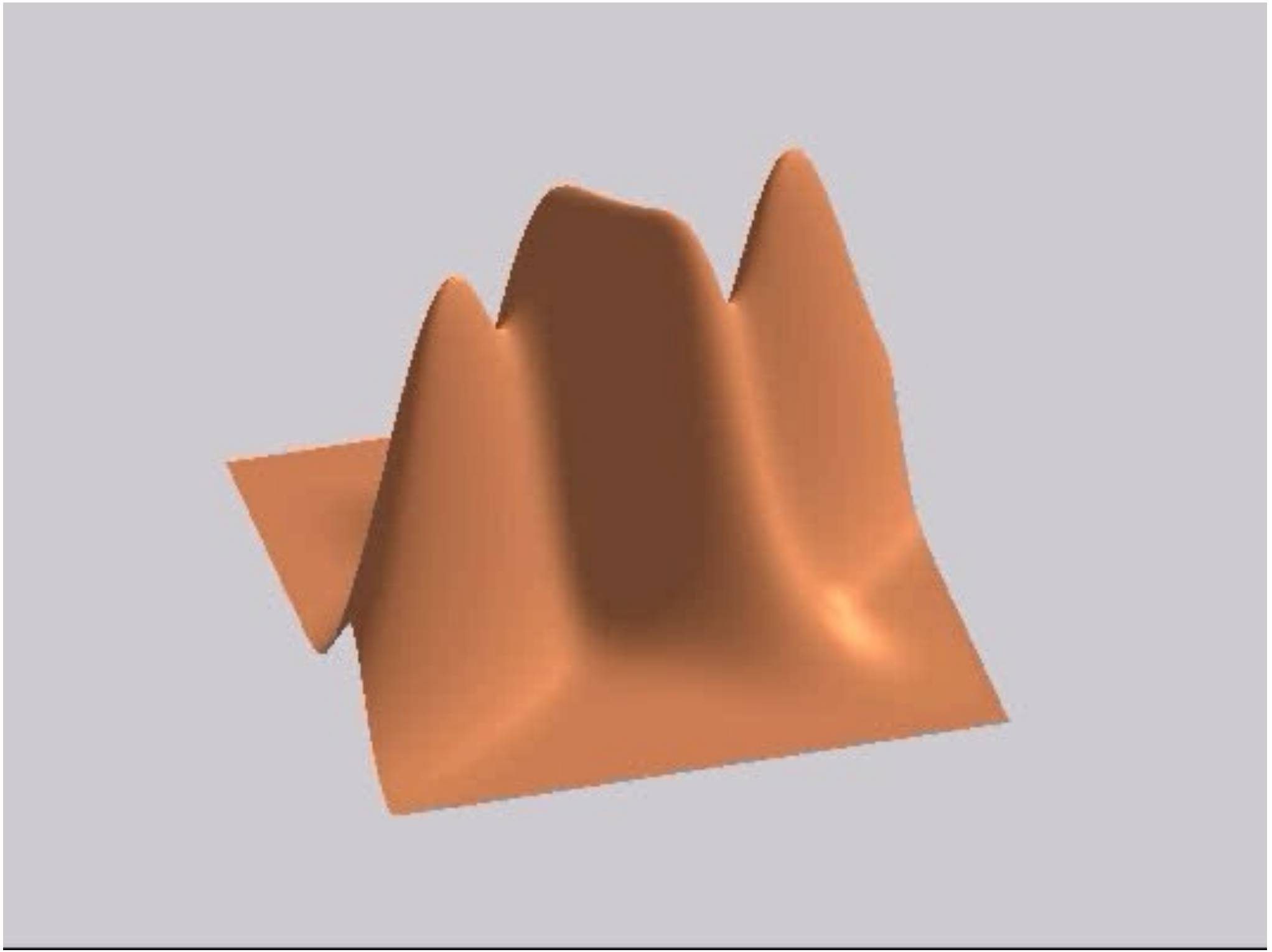
Example: Mixture of 3 Gaussians

Contours of the Gaussian Component PDFs



Contours of the Mixture PDF





Sampling from a Gaussian Mixture

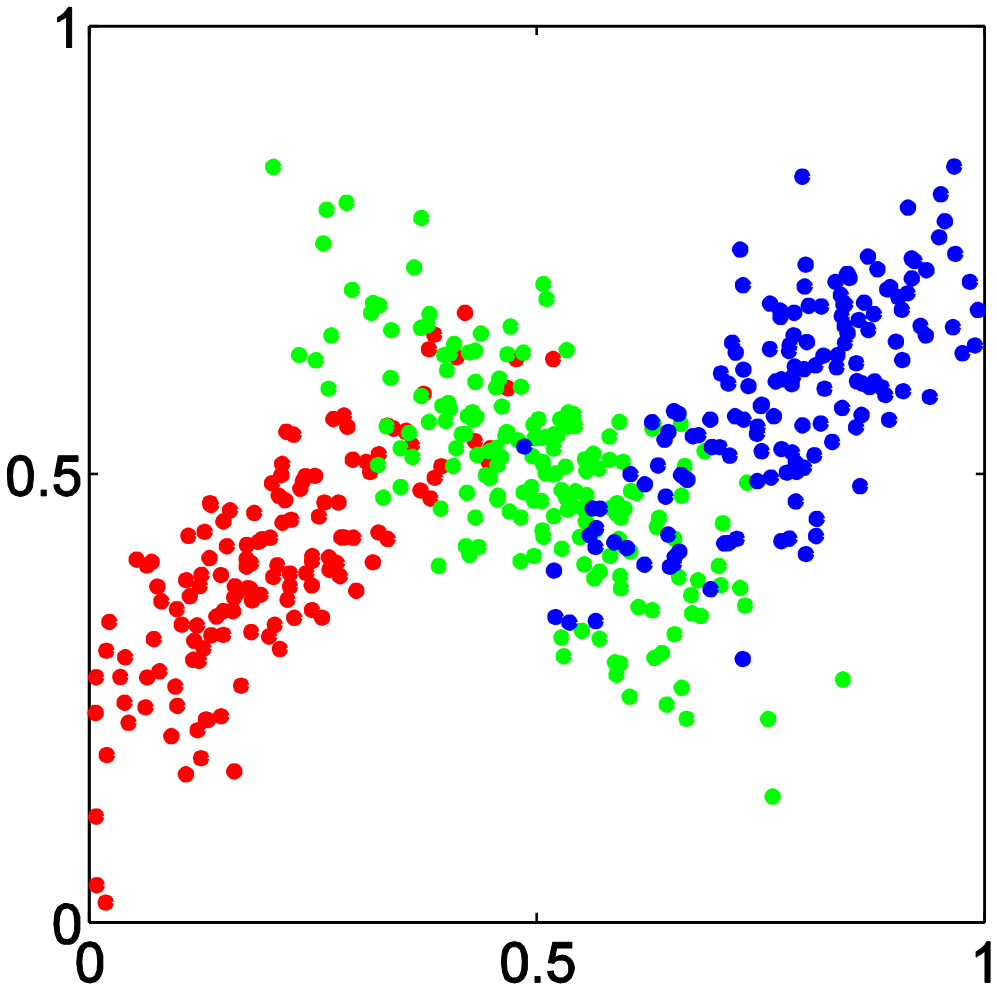
In simulation, the mixture parameters are known

- To generate a data point:

  First draw one of the components *k* with probabilies $p(k)$

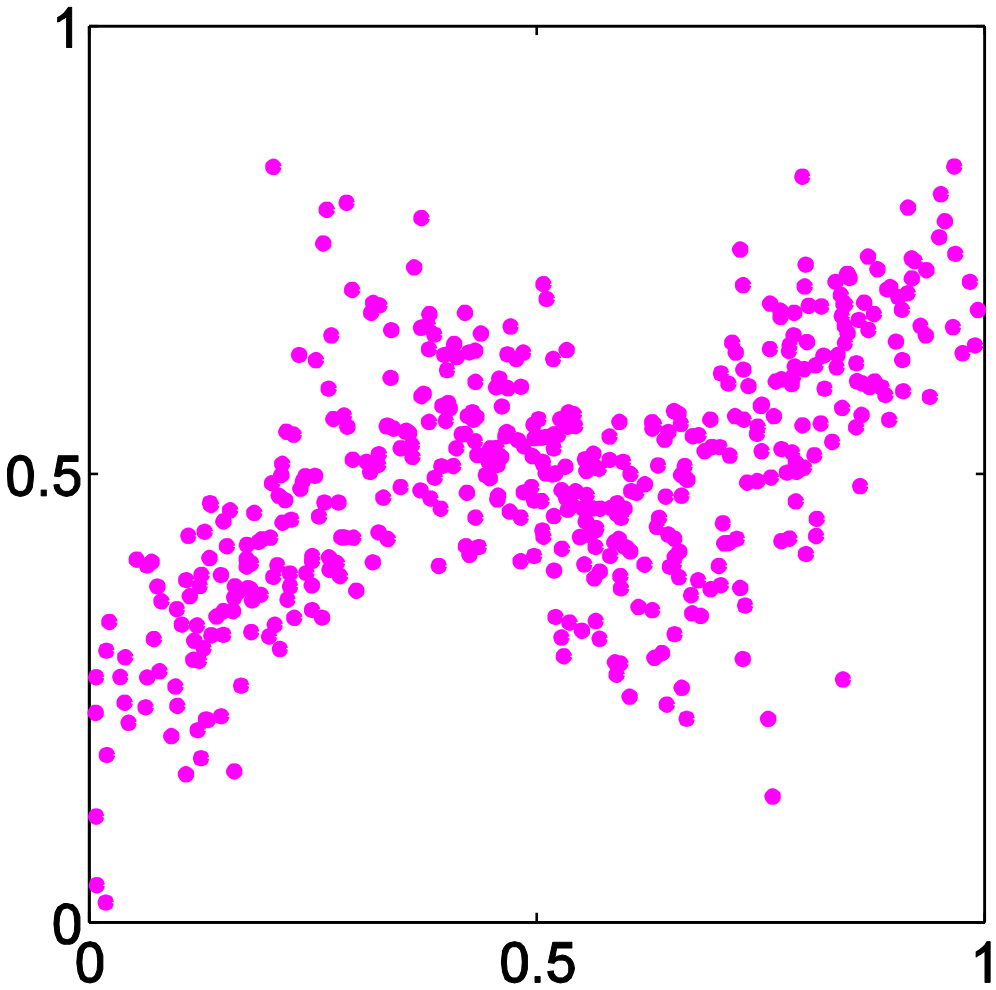
  then draw a sample $\mathbf{x}^{(n)}$ from the component $p(\mathbf{x}|k)$
- Repeat these two steps for each new data point

Synthetic Data Set



Synthetic Data Set Without Labels

Can you say from which cluster each data point comes from?



Fitting the Gaussian Mixture

- *Inverse* of the simulation process: given the data set, find the mixture parameters
  - mixing coefficients
  - means
  - covariances
- *If we knew* which component generated each data point, the maximum likelihood solution would be trivial
  - Separate Fitting Each Component to the corresponding cluster
- However, the data set is generally unlabelled
- The labels are referred to as the **latent** (= hidden) **variables**

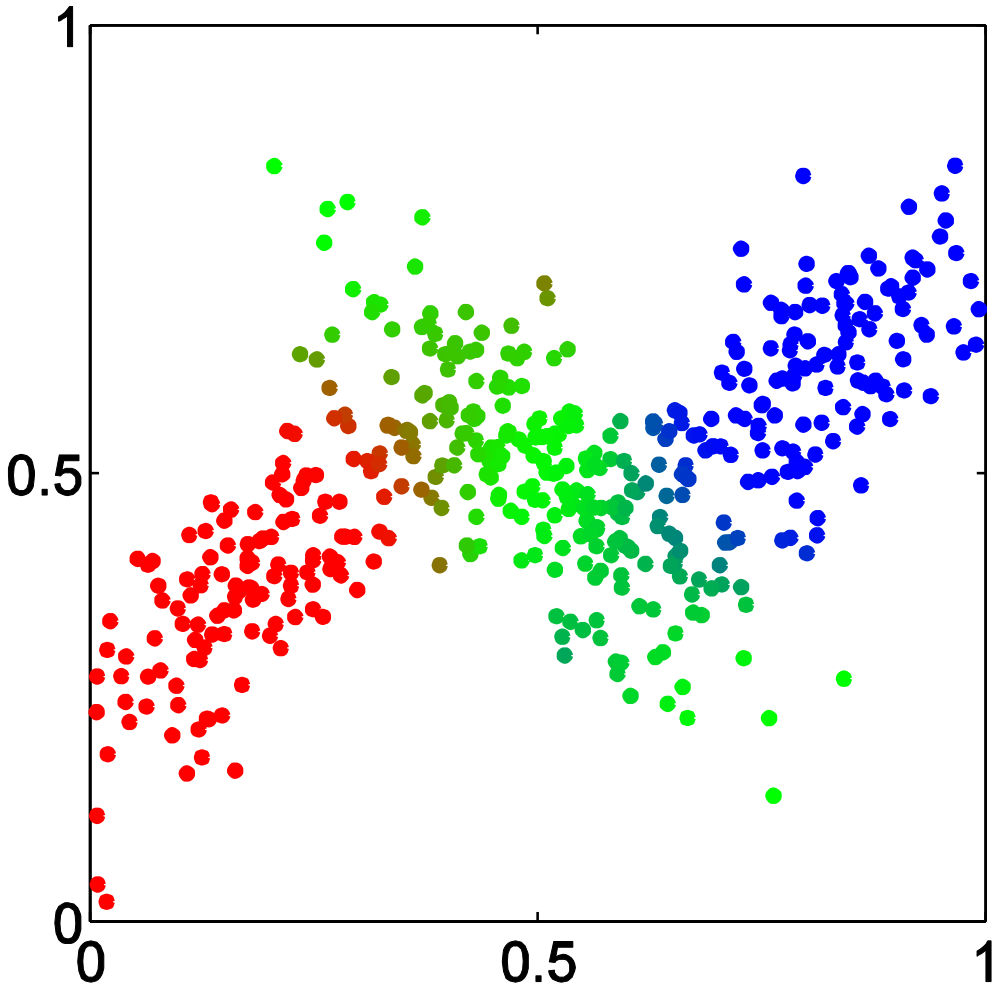
Posterior Probabilities

- 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 by Bayes' theorem as

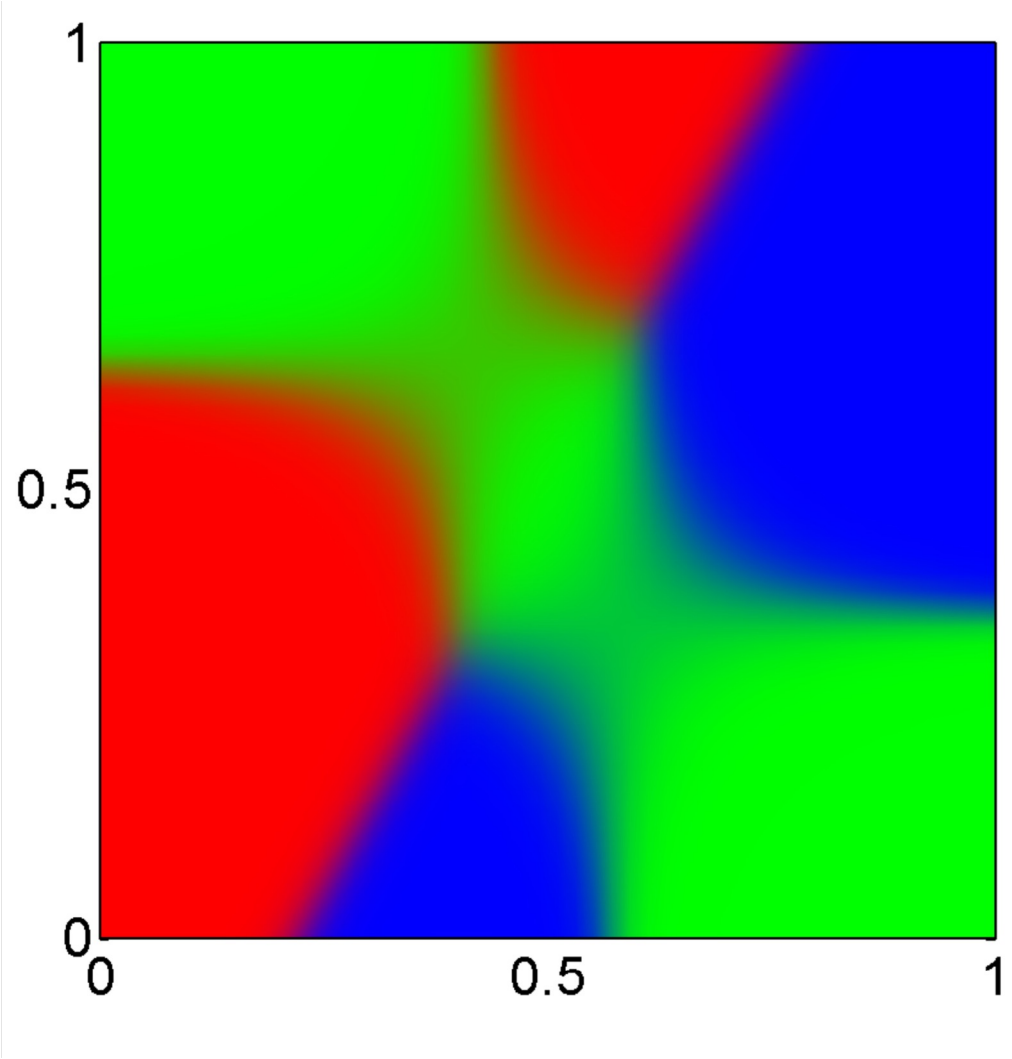
$$\gamma_k(\mathbf{x}) \equiv p(k|\mathbf{x}) = \frac{p(k)p(\mathbf{x}|k)}{p(\mathbf{x})}$$

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

Posterior Probabilities (colour coded)



Posterior Probability Map



Maximum Likelihood Estimation 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

Questions and Answers

Q: How to maximize the log likelihood

A: Expectation-Maximization (EM) algorithm

Q: How to avoid singularities in the likelihood function

A1: Good initialization helps

A2: Bayesian treatment (beyond the scope of this course)

Q: How to choose number of components *K*

A: Bayesian treatment (beyond the scope of this course)

EM Algorithm – Informal Derivation

- By differentiating the log likelihood, and setting the derivative with respect to $\boldsymbol{\mu}_j$ equal to zero yields

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

and further

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

which is the **weighted mean of the data**

EM Algorithm – Informal Derivation

- Similarly for the covariances

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

- For mixing coefficients use Lagrange multipliers yields

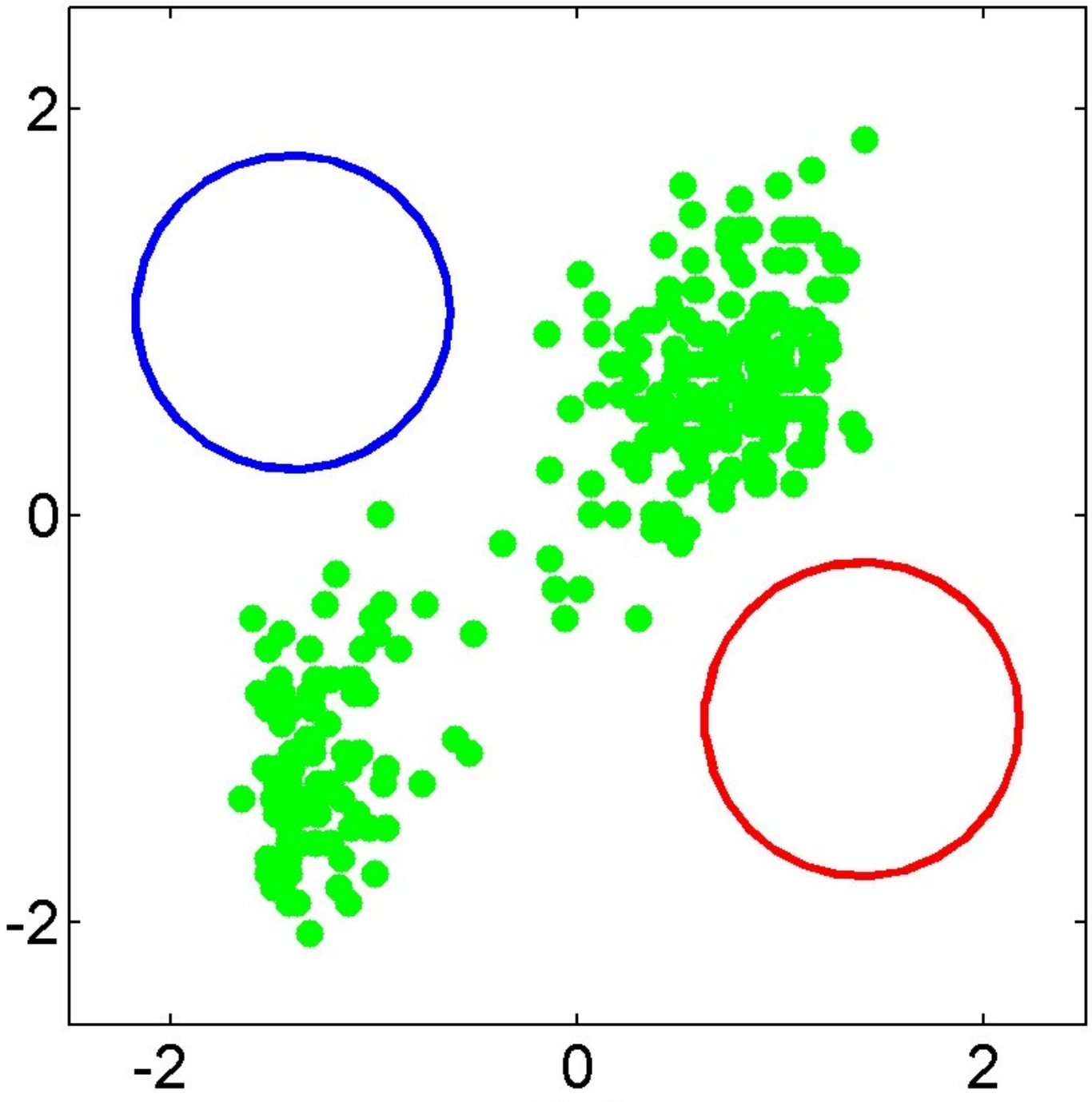
$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)$$

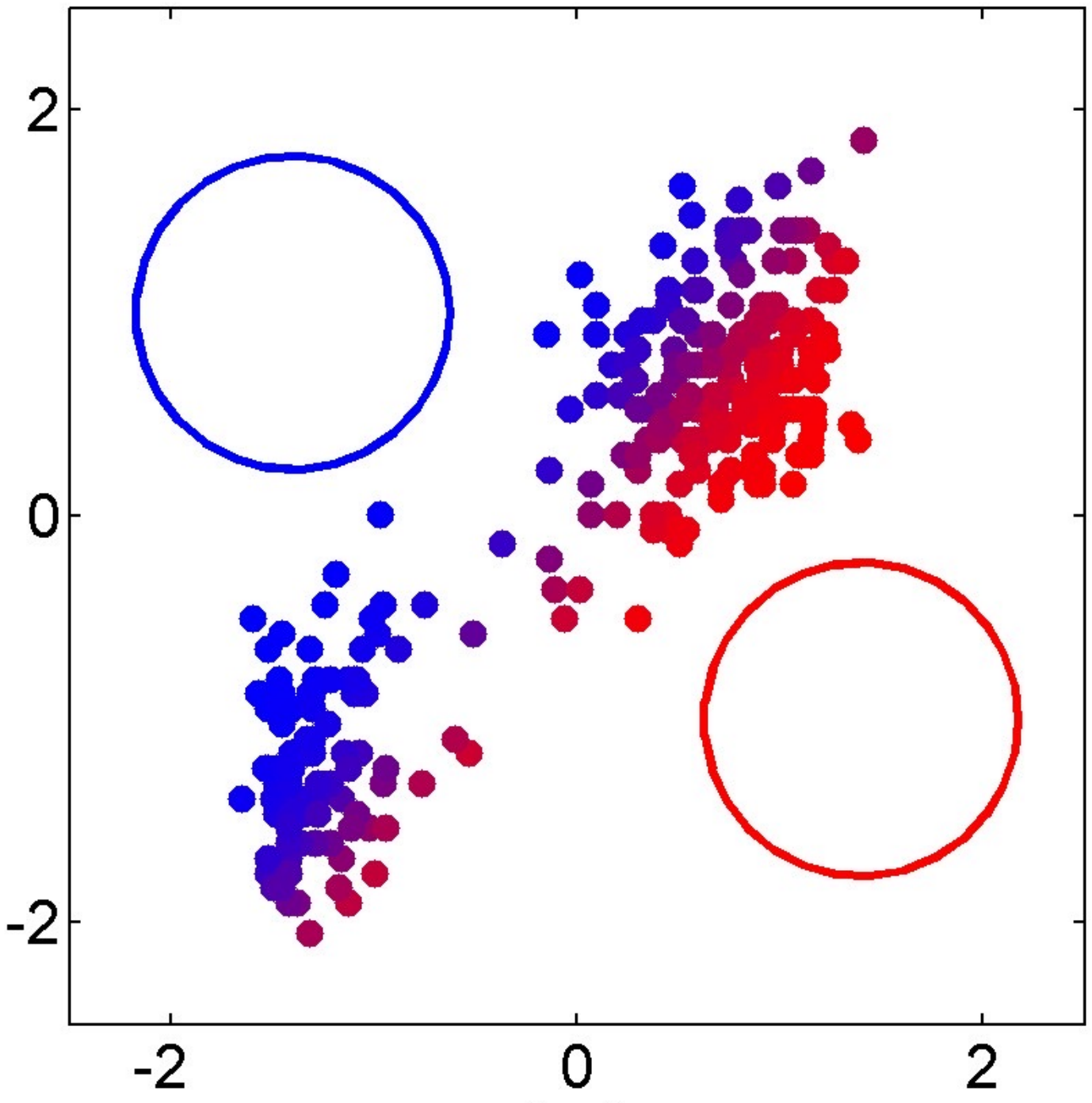
EM Algorithm – Informal Derivation

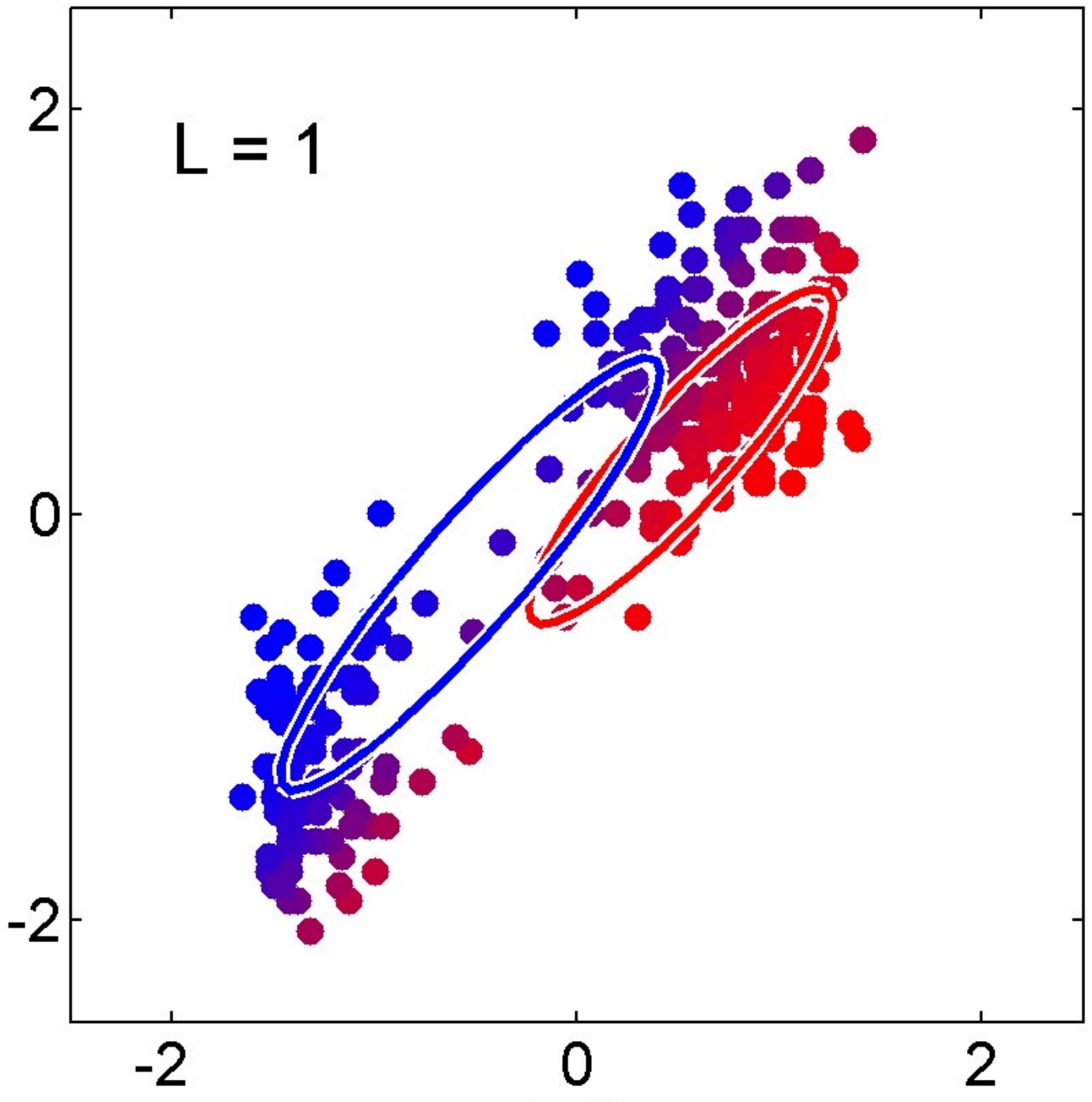
- The solutions are **not** closed form: they are **coupled**
- Suggests an iterative scheme:
  - Make initial guesses for the parameters
  - Alternate between

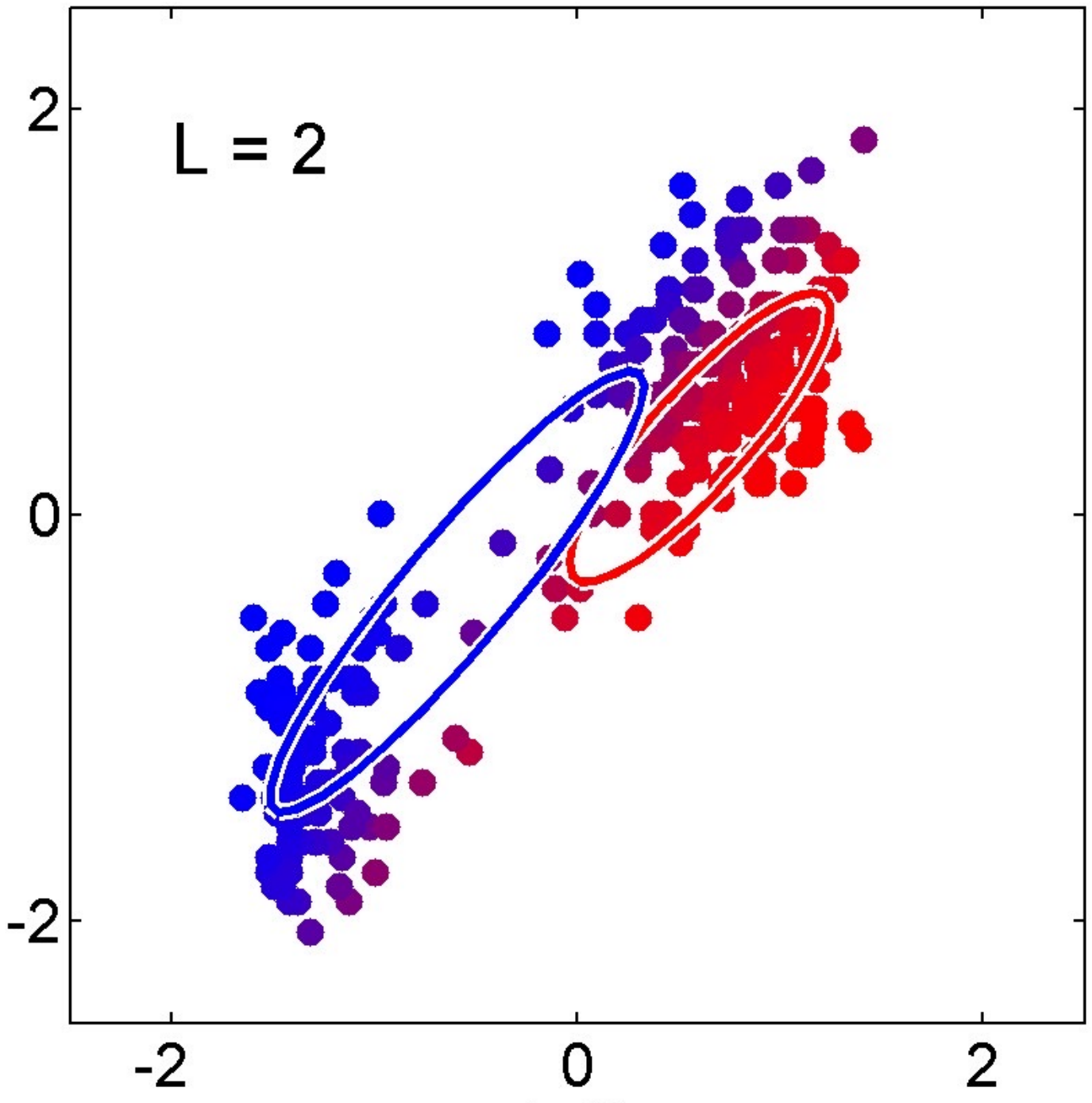
    E-step: evaluate responsibilities

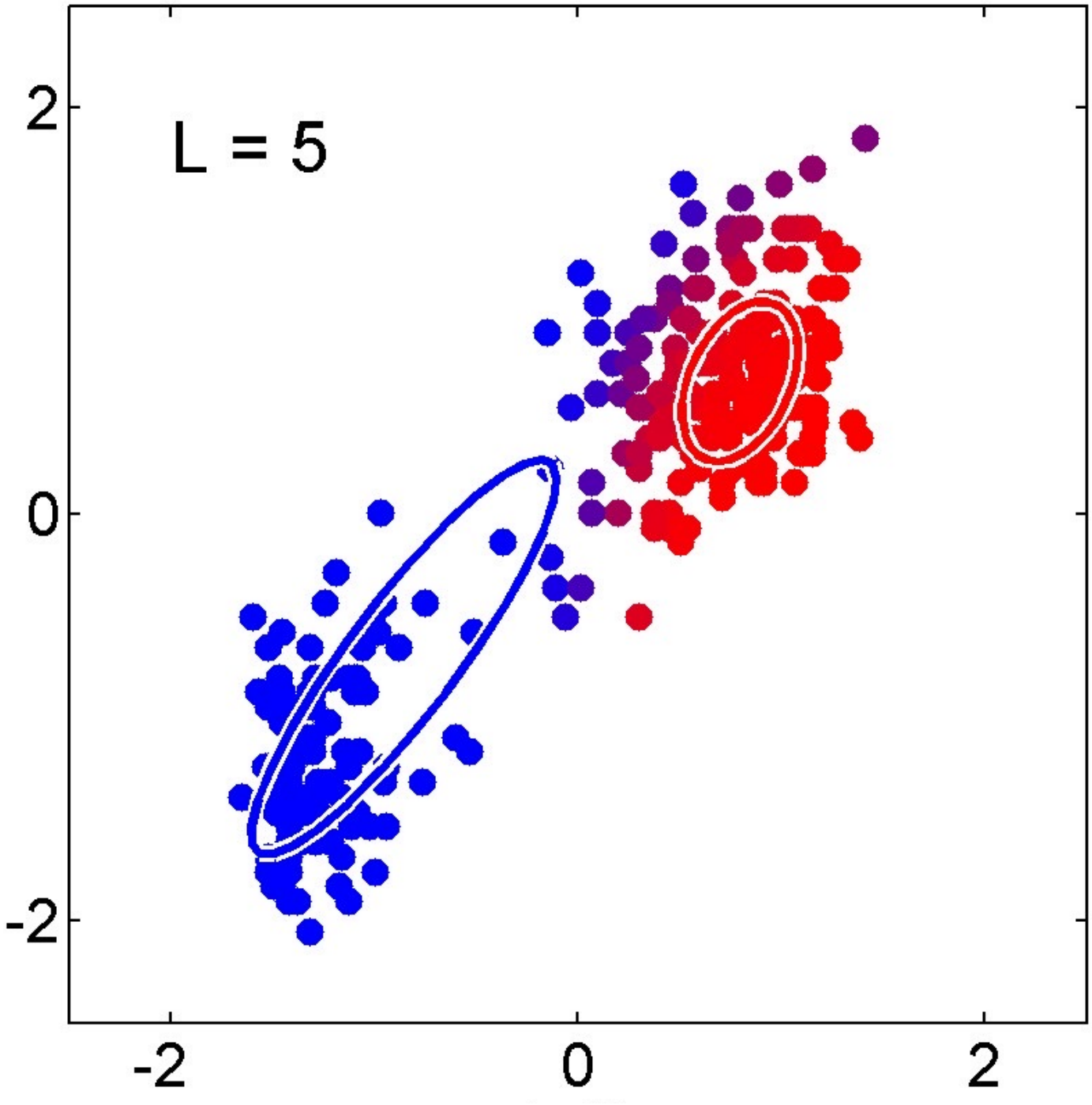
    M-step: update parameters using the ML results

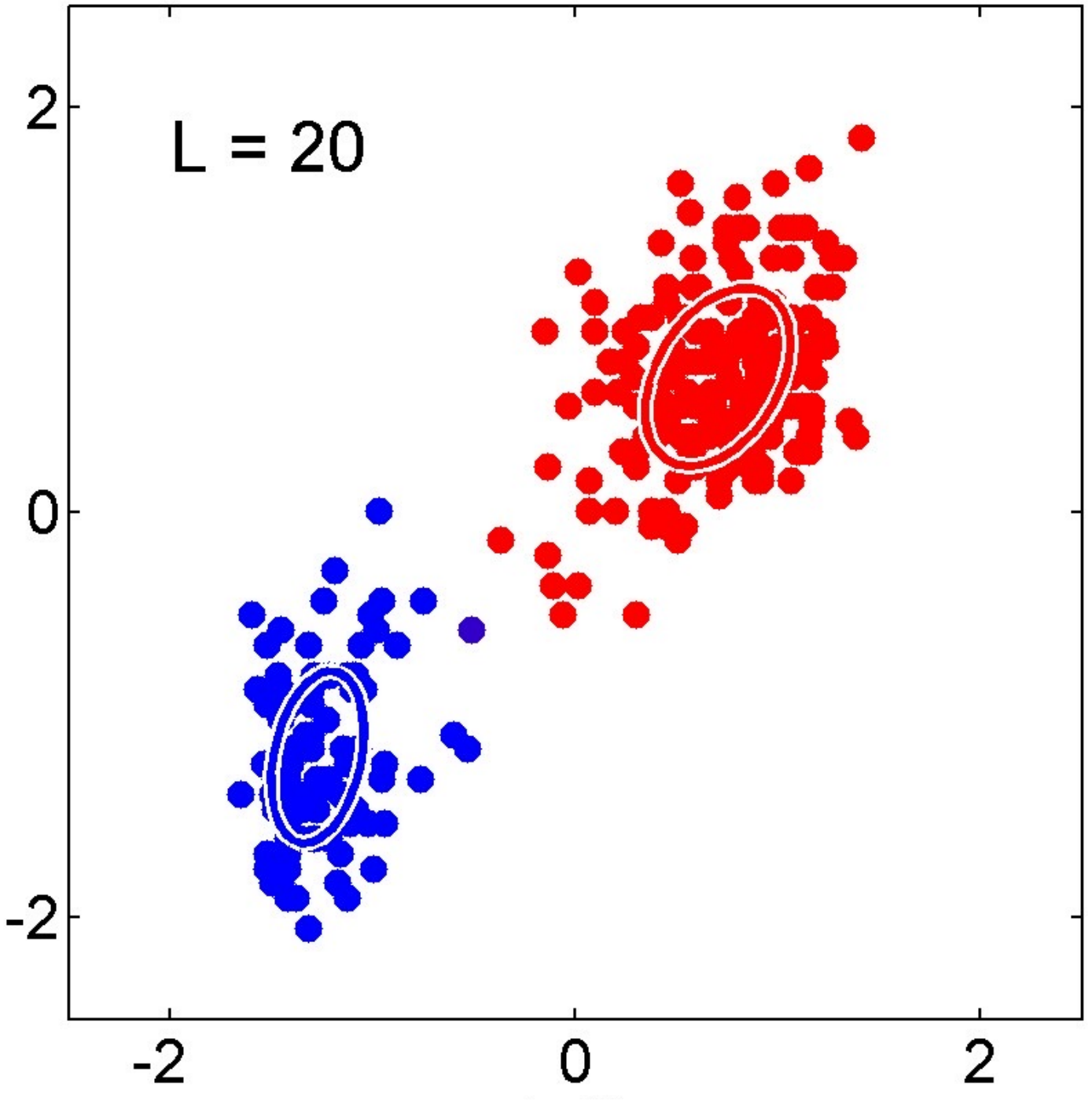

2
0
-2
-2
0
2


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

General Expectation Maximisation

- Let the likelihood of the **complete data** be

$$p\left(\mathbf{X},\mathbf{Z}\middle|\theta\right)$$

where $\mathbf{X}$ is the **observed data** and $\mathbf{Z}$ is **missing data**

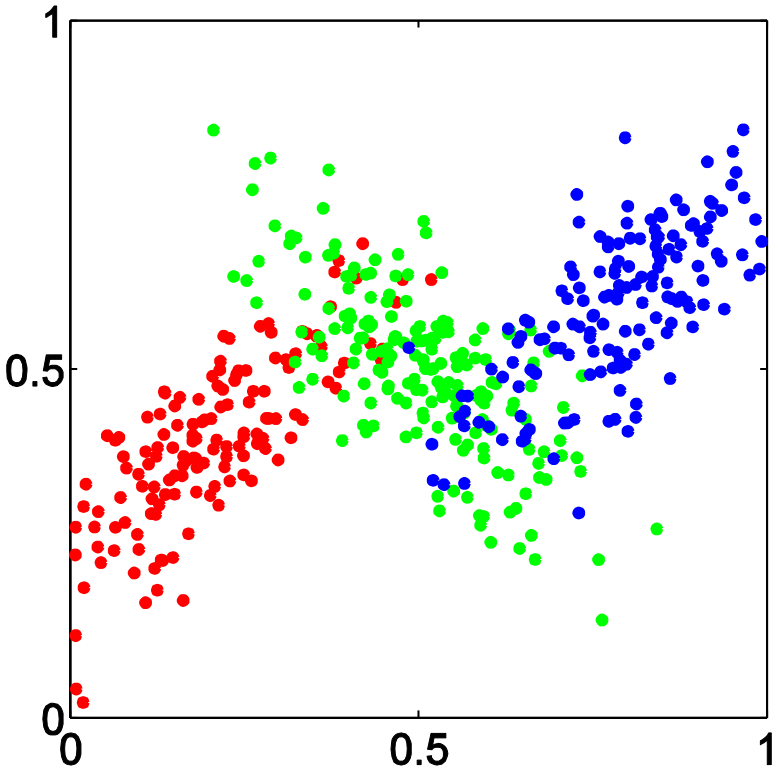
E-step: Construct

$$\chi\left(\theta,\hat{\theta}^{(i-1)}\right)=\mathrm{E}_{\mathbf{Z}\middle|\mathbf{X},\theta^{(i-1)}}\left\{\log p\left(\mathbf{X},\mathbf{Z}\middle|\theta\right)\middle|\mathbf{X},\hat{\theta}^{(i-1)}\right\}$$

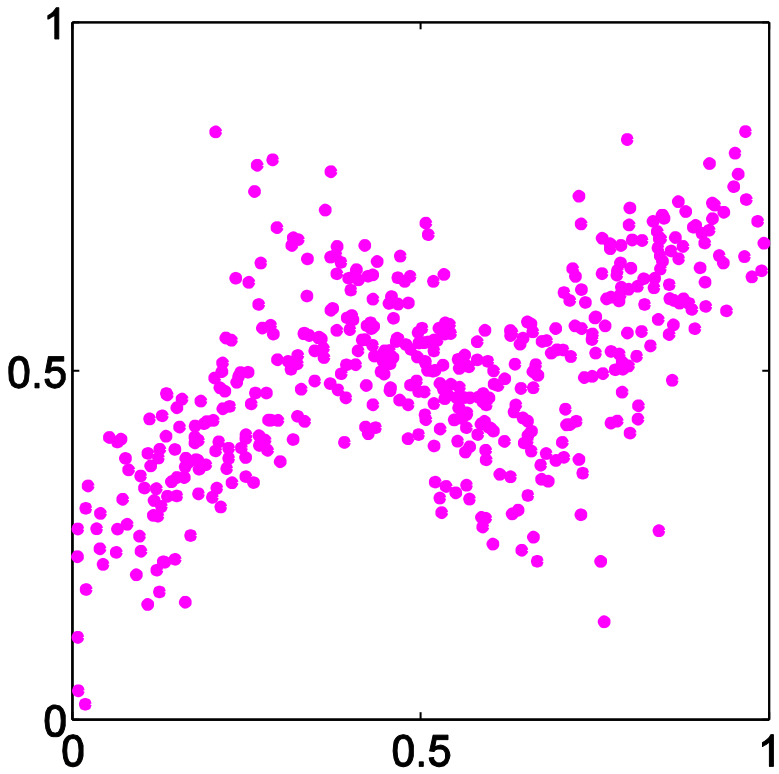
M-step: Maximize

$$\hat{\theta}^{(i)}=\arg\max_{\theta}\chi\left(\theta,\hat{\theta}^{(i-1)}\right)$$

Complete and Incomplete Data



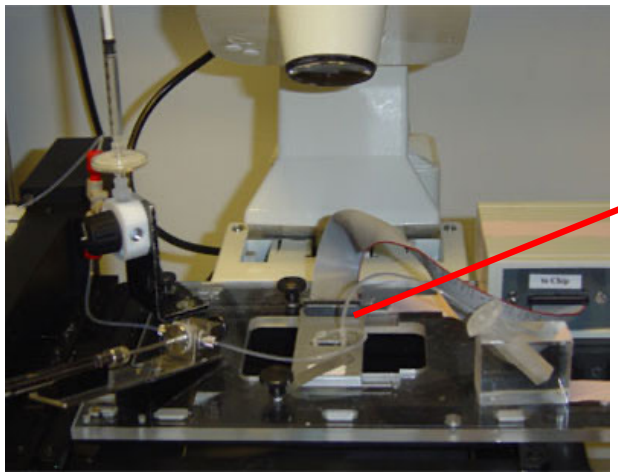
complete

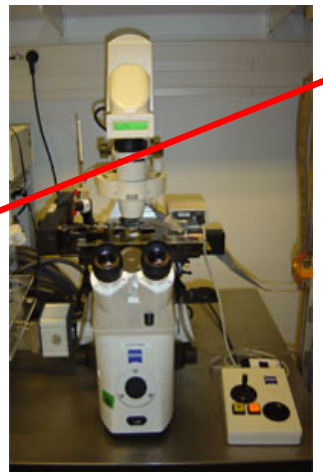


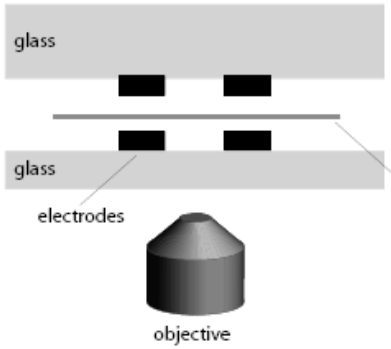
incomplete

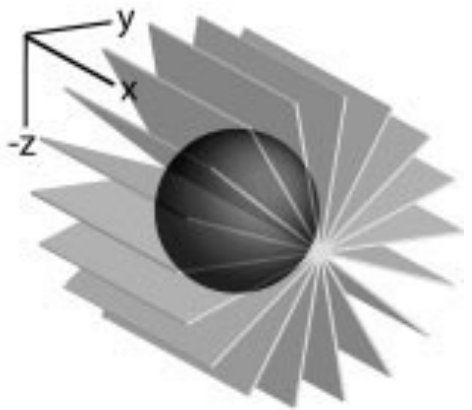
Application Example of EM

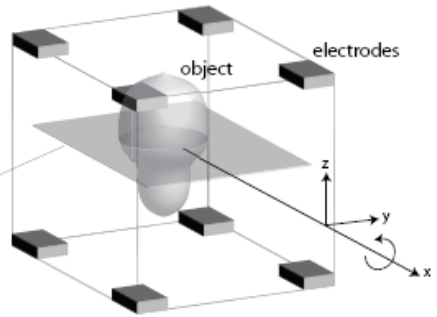
Micro-rotation fluorescence imaging for live cells





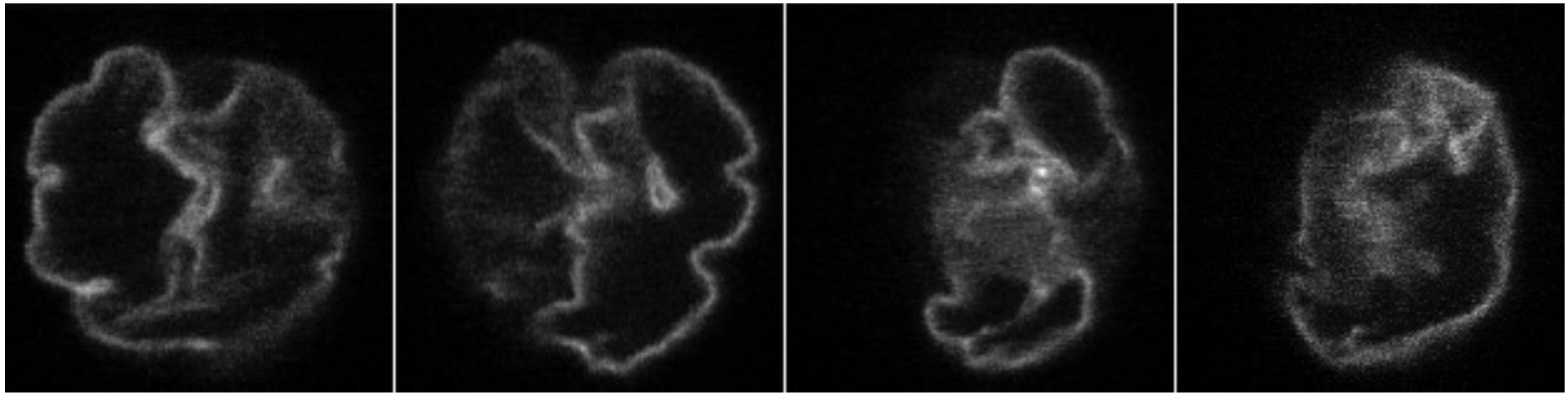






http://www.pfid.org/AUTOMATION/

Examples of micro-rotation images

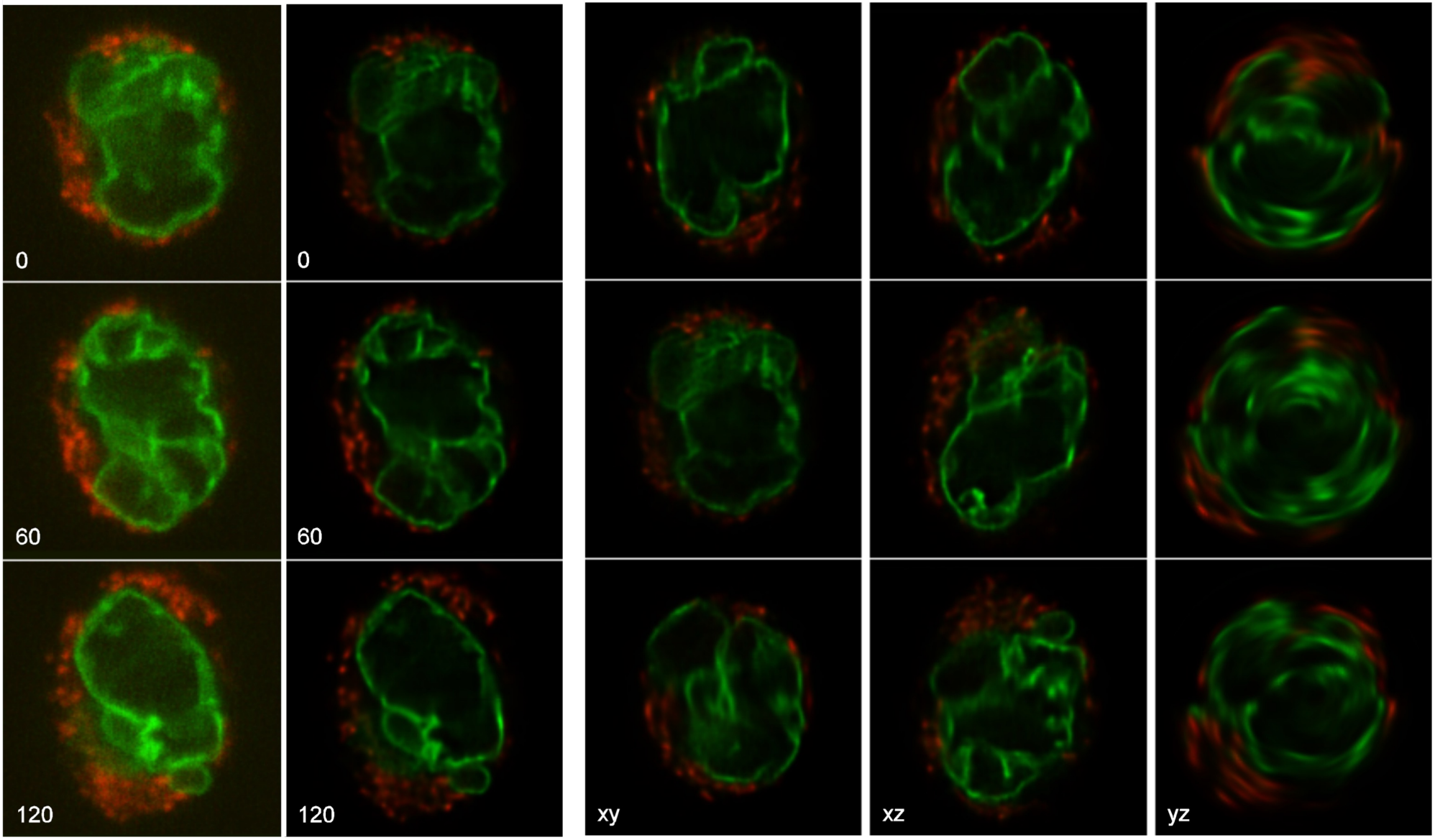


A human living cell, expressing fluorescence at nuclear envelope

Laksameethanasan et al. 2009

EMTV

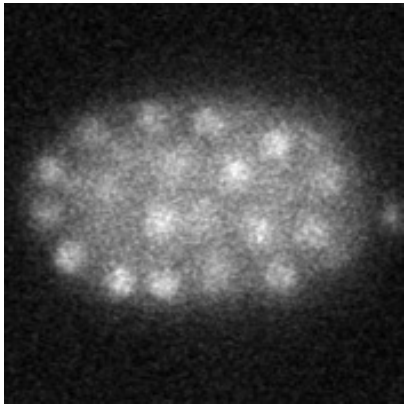
lamin-GFP + mito-OFP



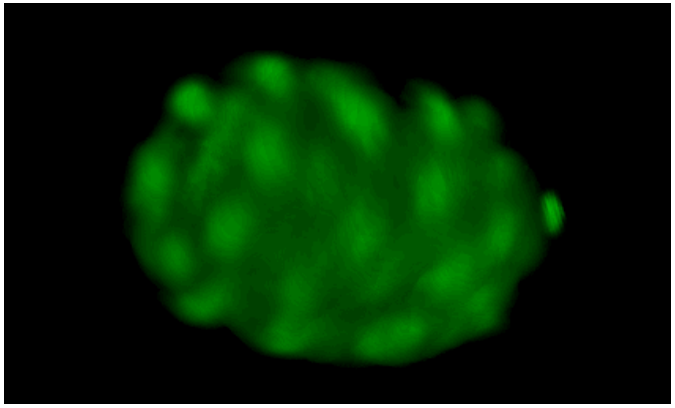
IT UNIVERSITY OF COPENHAGEN

Example of reconstructions

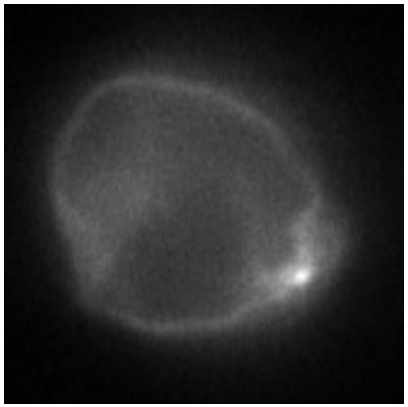
C.elegans embryo

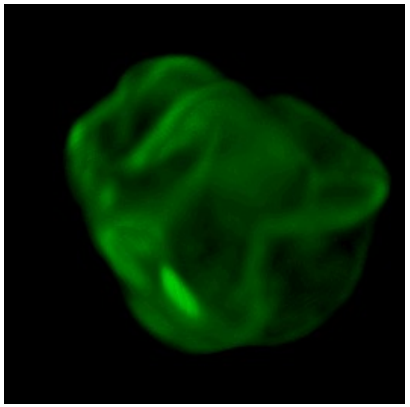


reconstruction

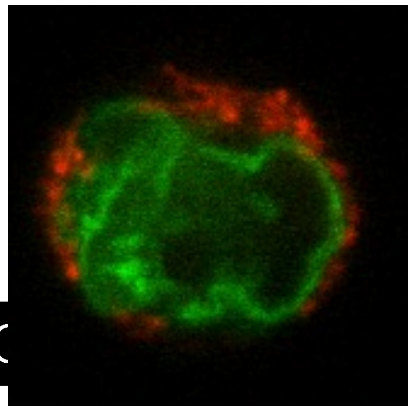


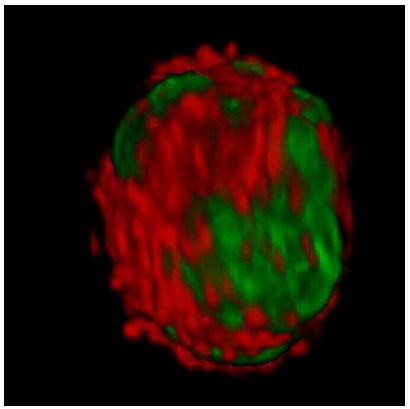
human living cell





human living cell





Summary

- K-means is a special case of fitting a Gaussian Mixture Model into data
- EM is the standard way of fitting a Gaussian Mixture Model
- Expectation Maximization is a general principle for solving maximum likelihood estimates with missing data