Current Topics in Computational Biology

IX: Clustering and mixture models

Christoph Lippert, PhD

Reference material

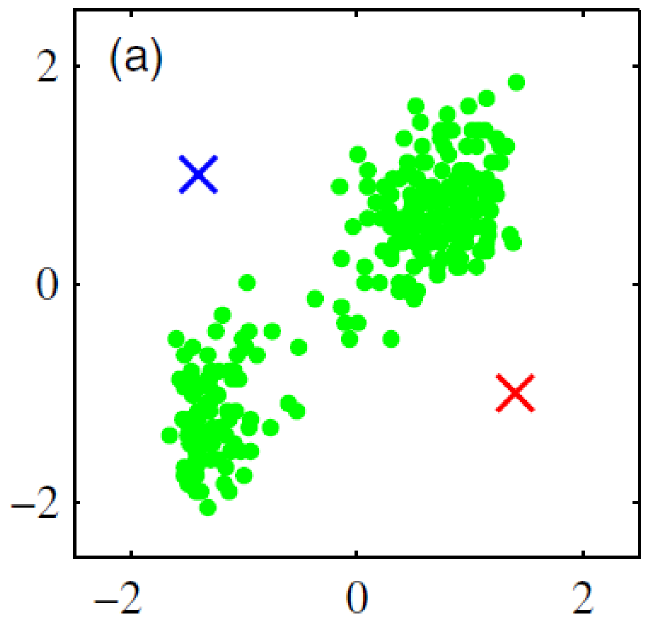
- C.M. Bishop: Pattern recognition and Machine Learning, Cambridge University Press, 2006
    - chapter 9
    - (chapter 2)

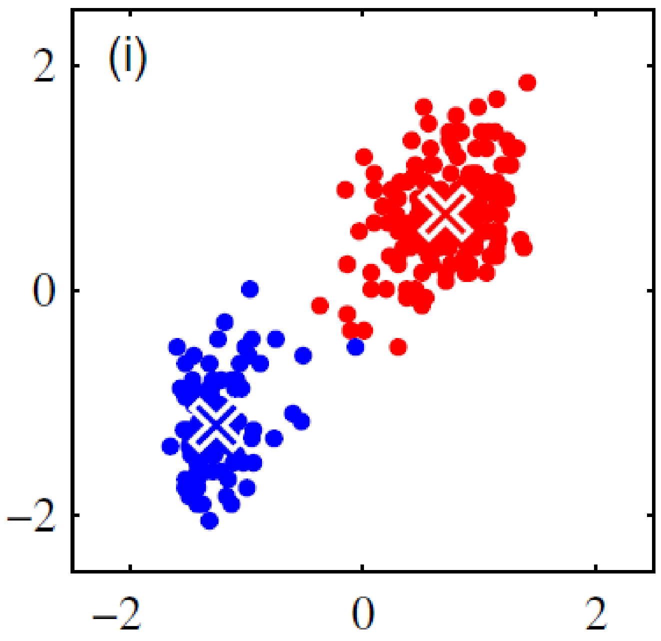
Clustering

- Class discovery
- Given a set of objects, group them into clusters (classes that are unknown beforehand)
- Unsupervised learning (no labels *y*)

Examples:

- Cluster images into categories
- Cluster patient data to find disease subtypes





What is clustering?

- **Supervised versus unsupervised learning**
- general inference problem: given x_i, predict y_i by learning a function
$$y = f(x)$$
- **training set:** set of examples $(x_i;\ y_i)$ where
$$y_i \ = \ f(x_i) + \epsilon_i$$
  (but f is still unknown!)
- **test set:** new set of data points x_i, where y_i is unknown
- **Supervised:**
  - use training data to infer your model, then apply this model to the test data
- **Unsupervised:**
  - no training data, learn model and apply it directly on the test data

K-means

Objective:

- Partition the dataset into *K* clusters such that the distance of each point to its cluster mean is minimized

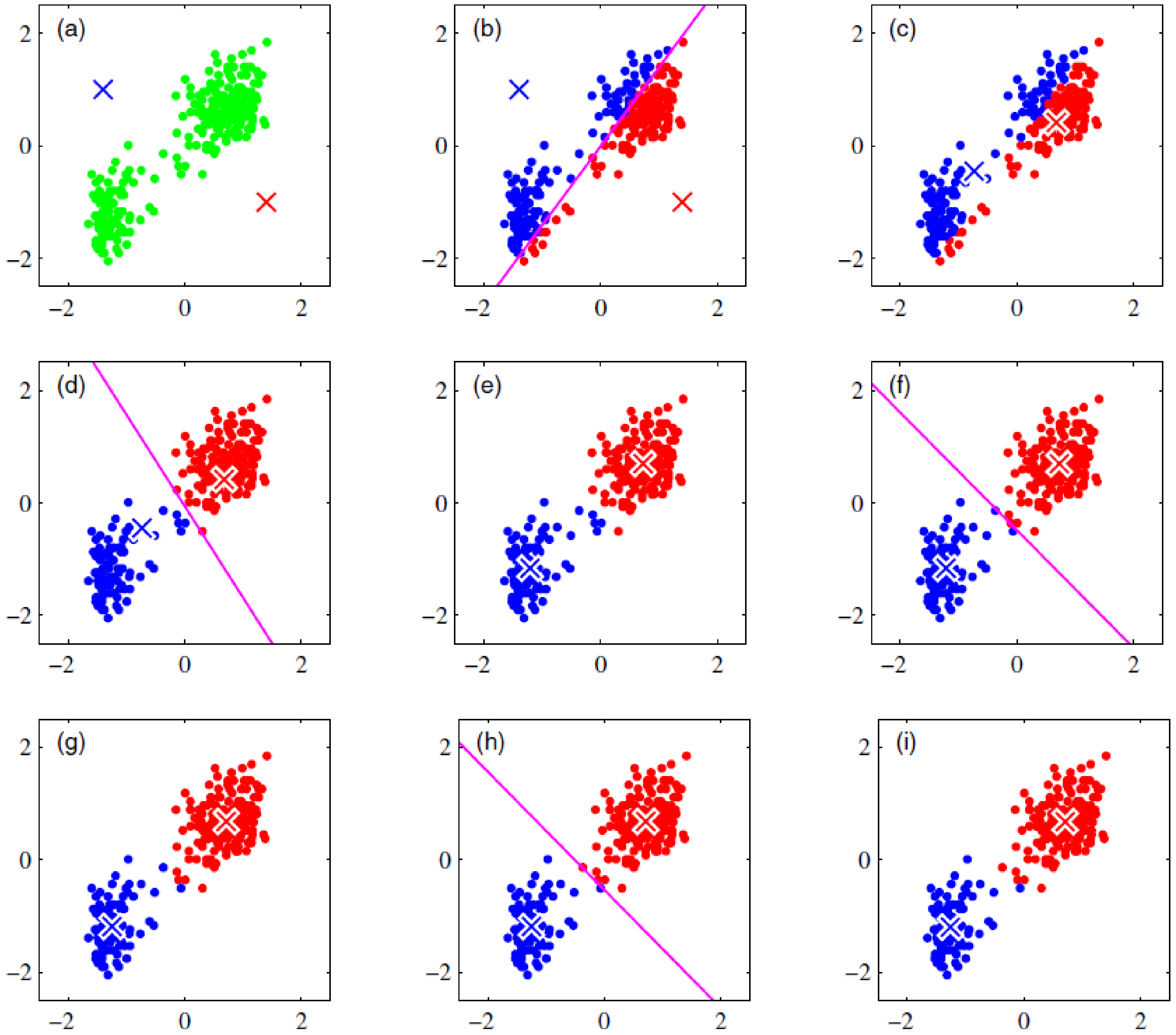
$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

where

- $r_{nk} \in \{0,1\}$ is a **cluster indicator**
- μ_k is the **cluster mean**

K-means

1. Initialize cluster means
2. Assign each point to the cluster whose mean is closest to the point
3. Re-compute the cluster means
4. If any point changed its cluster membership: Repeat from step 2



K-means

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

E-step:

Assign each point to the cluster whose mean is closest to the point

= minimize J given all μ_k w.r.t. all r_{nk}

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

M-step:

Re-compute the cluster means

= minimize J given all r_{nk} w.r.t. all μ_k

Set derivative of J w.r.t. μ to zero and solve:

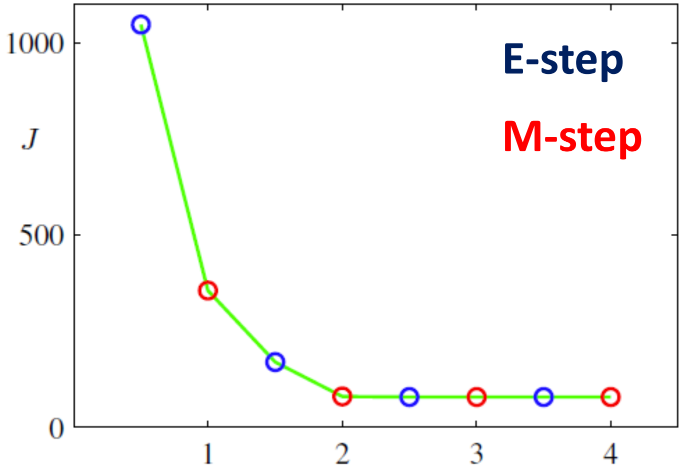
$$2 \sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0$$

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}.$$ (=cluster mean)

K-means

Things to note

- K-means is still the state-of-the-art method for most clustering tasks
- When proposing a new clustering method, one should always compare to K-means.
- The algorithm always converges
  - In each step the objective J is reduced or stays the same (=convergence)
- algorithm has several setbacks:
  - It is order-dependent.
  - Its results depends on the initialization of the clusters.
  - Its result may be a local optimum, not the global optimal solution.



K-means

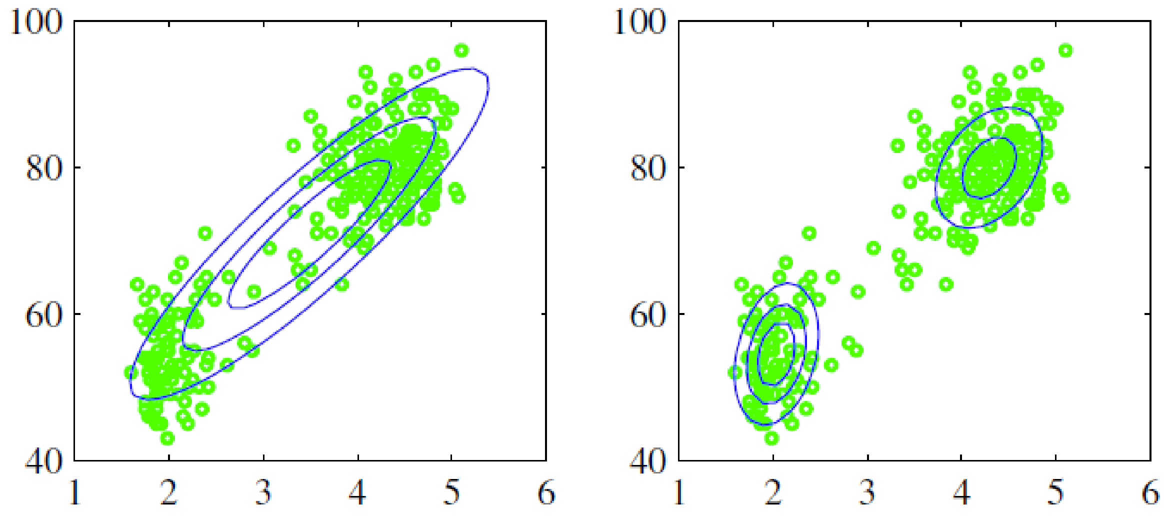
Image segmentation

- Represent each pixel as RGB
- Use *K*-means to cluster pixels
- Clusters represent homogenous segments of the image

- Drawbacks:
  - Ignores spatial information

Mixture density estimation

- Given data x_i
- estimate distribution $p(x)$
- e.g. estimate Gaussian using maximum likelihood (left)
  - Might be too simple
- More complex alternative: mixture of multiple Gaussians (right)
- Problem:
  - How to assign data points to individual Gaussians?
  - Similar to clustering

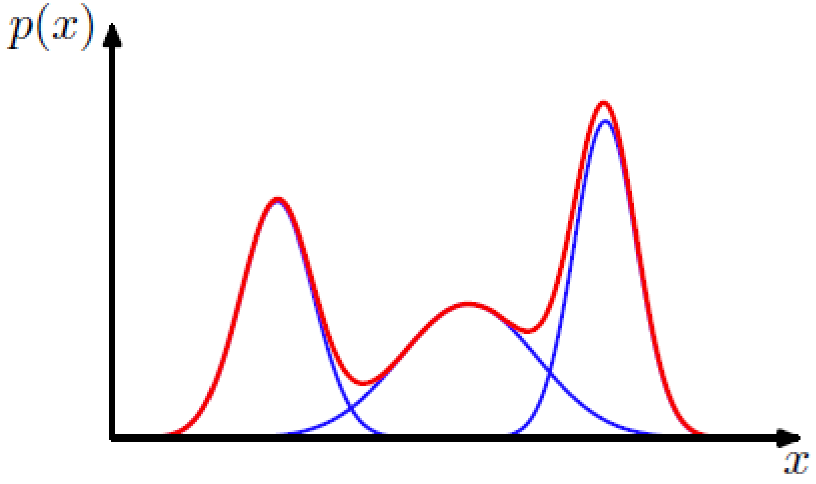


Mixture distributions

- Superposition of weighted base distributions p_k

$$p(x) = \sum_{k=1}^{K} \pi_k p_k(x)$$

  - Individual weighted distributions
  - Sum (mixture)
  - Mixing coefficients $\pi_k \geq 0$

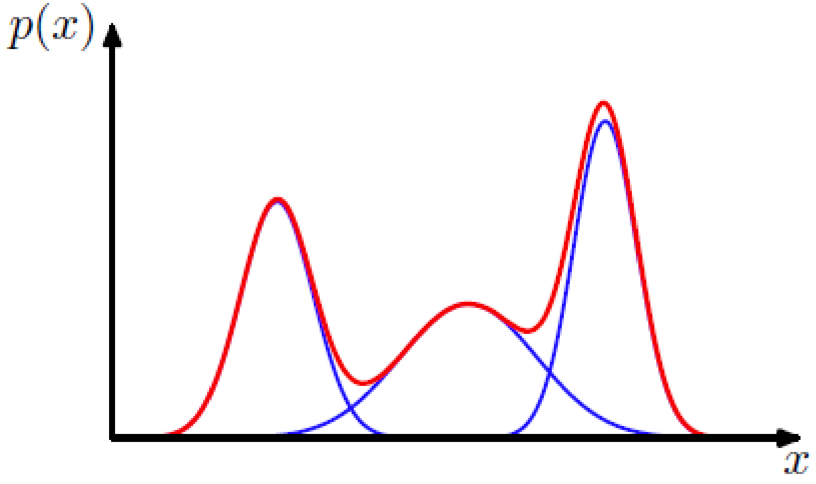


- Example: **mixture of Gaussians**

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

Mixture distributions

- Superposition of weighted distributions
  - Individual weighted distributions
  - Sum (mixture)
  - Mixing coefficients $\pi_k \geq 0$



- Mixture distribution is normalized

$$\int p(x) d\, x = 1$$

- As individual Gaussians are normalized:

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

- Example: **mixture of Gaussians**

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

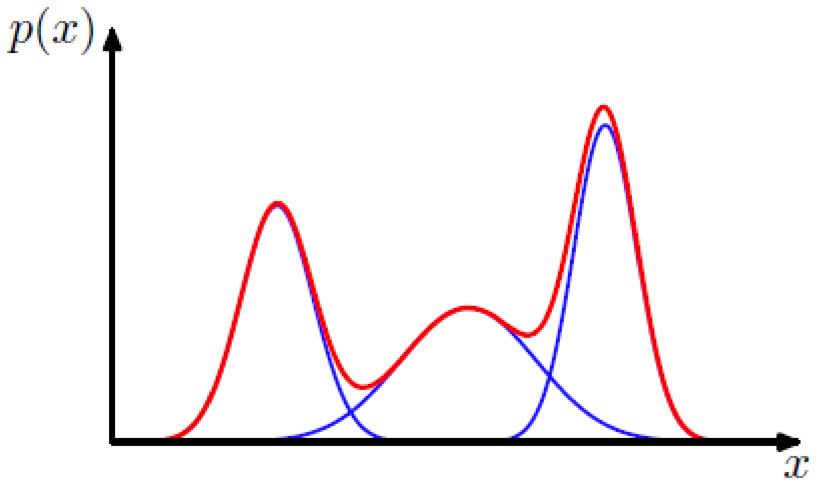
Mixture distributions are hierarchical models

- Marginal density *p(x)*

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

Prior of component **z**

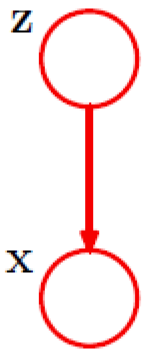
Conditional distribution of **x** given component **z**



- Example: **mixture of Gaussians**

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

- Hierarchical model
  - Sample **z** (cluster)
  - Sample **x** given **z**



Mixtures of Gaussians

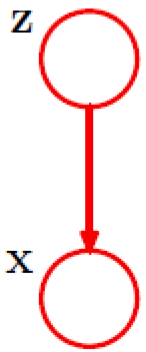
- Marginal density *p(x)*

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

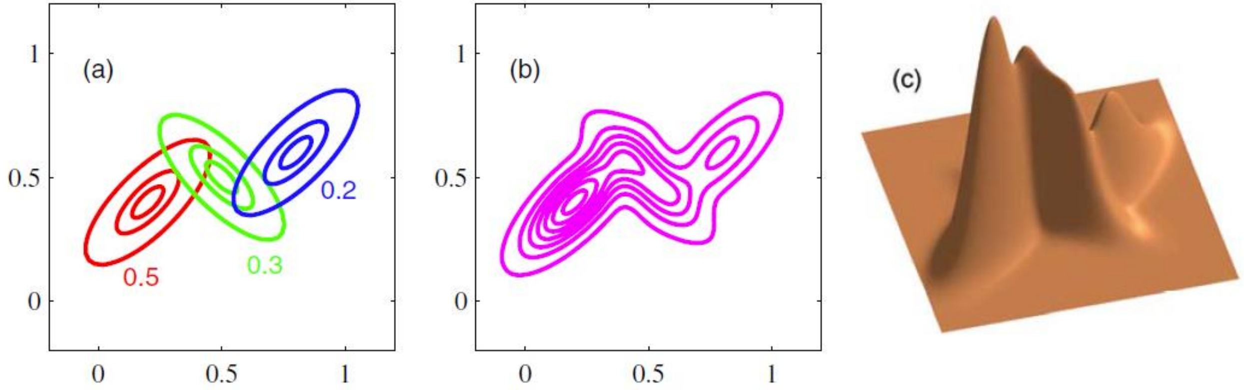
Prior of component **z**

Conditional distribution of **x** given component **z**

- Hierarchical model
  - Sample **z** (cluster)
  - Sample **x** given **z**



- Mixtures of Gaussians can approximate arbitrarily complex distributions



(a) 3 base components with priors
(b) Contours of marginal density
(c) Marginal density in 3D

Mixtures of Gaussians

Inference

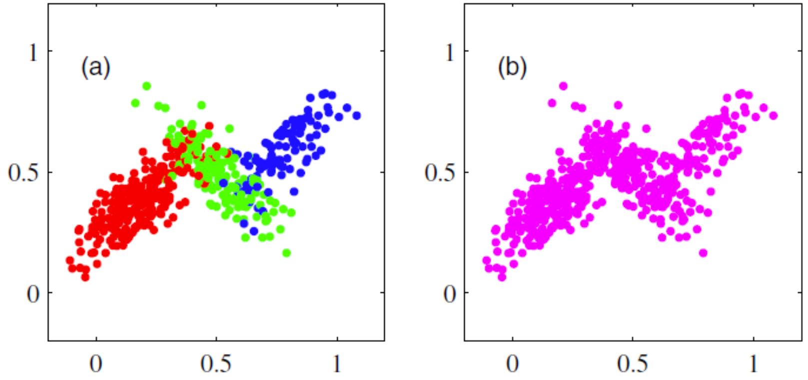
- Given data $\{x_1, \ldots, x_N\}$
- Can we infer *z* given *x*?
- Use Bayes Theorem!

$$p(z|x) = \frac{p(z)p(x|z)}{p(x)}$$

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

Prior of component **z**

Conditional distribution of **x** given component **z**



(a) True generating components
(b) Observed data (z unknown)

Mixtures of Gaussians

Inference

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

Prior of component **z**

Conditional distribution of **x** given component **z**

- Write *z* as **binary** vector of length *k*
  - Exactly 1 entry is one, others 0
  - The probability of each value being 1 equals π_k

$$p(z_k = 1) = \pi_k$$

  - In this form $p(z)$ can be written as

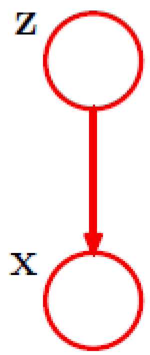
$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

  - The conditional of *x* given a particular value of *z* as

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  - The full conditional as

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



Mixtures of Gaussians

Inference

- Use Bayes Theorem!

$$p(z|x) = \frac{p(z)p(x|z)}{p(x)}$$

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

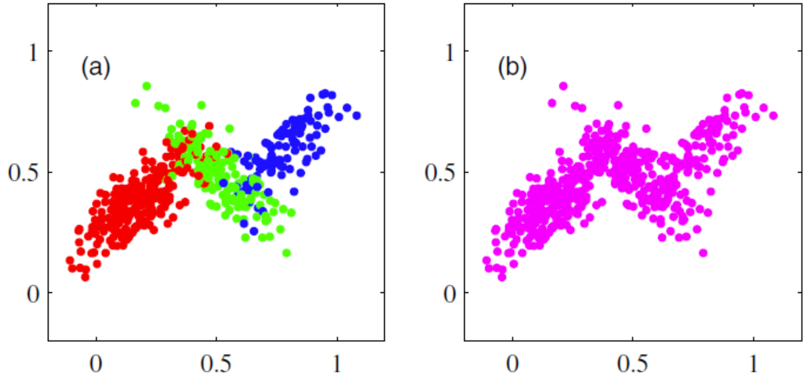
Prior of component **z**

Conditional distribution of **x** given component **z**

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)}$$

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

responsibilities



(a) True generating components
(b) Observed data (z unknown)

Mixtures of Gaussians

Inference

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

Prior of component **z**

Conditional distribution of **x** given component **z**

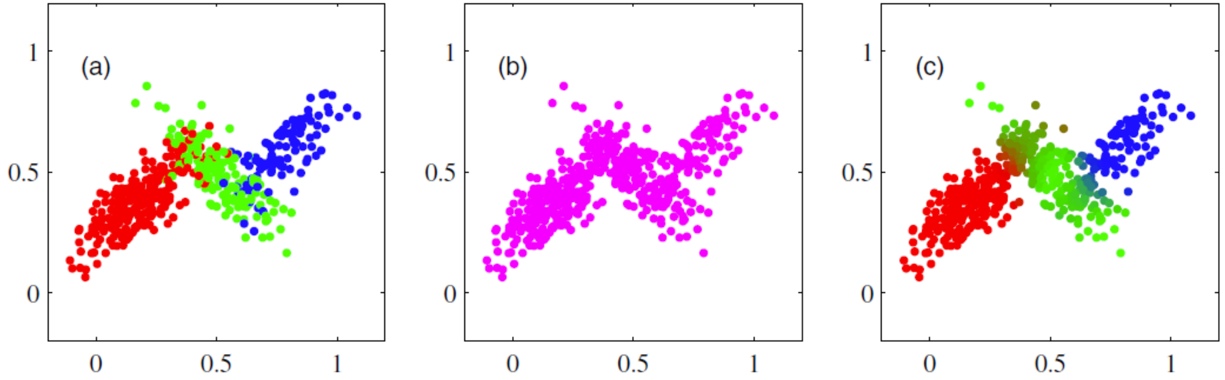
- Use Bayes Theorem!

$$p(z|x) = \frac{p(z)p(x|z)}{p(x)}$$

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)}$$

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

responsibilities



(a) True generating components
(b) Observed data (z unknown)
(c) Inferred responsibilities

Done?

Maximum likelihood estimation

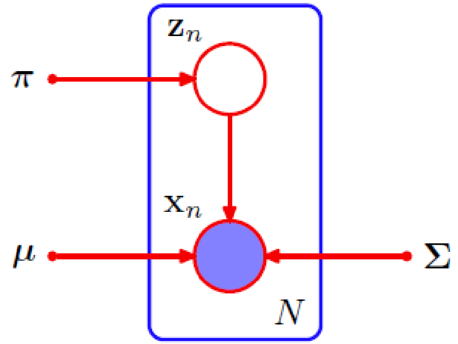
- Log likelihood

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

- Estimates for all parameters are required

$$\pi_k, \mu_k, \Sigma_k$$

- Could use gradient-based optimization to maximize the likelihood
- Alternative: the **EM algorithm**



Graphical model for N data points

Expectation-maximization algorithm

M-step: component mean estimation

- Log likelihood

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

- Derivative w.r.t. $\boldsymbol{\mu}_k$:

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

Multiplying both sides by $\boldsymbol{\Sigma}_k$ to cancel $\boldsymbol{\Sigma}_k^{-1}$

scalar (note that we hide the dependency on $\boldsymbol{\mu}_k$)

- Solving for $\boldsymbol{\mu}_k$

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

sample average weighted by responsibilities for k_{th} component

Expectation-maximization algorithm

M-step: component variance estimation

- Log likelihood

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

- Solving for $\boldsymbol{\Sigma}_k$

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

sample variance weighted by responsibilities for k_{th} component

Expectation-maximization algorithm

M-step: mixing coefficients estimation

- Log likelihood

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

- Use Lagrange multiplier λ to enforce constraint

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

- Derivative w.r.t. π_k

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

- Solve for π_k

$$\pi_k = \frac{N_k}{N} \qquad \text{(where } \lambda = -N\text{)}$$

Constraint on π_k:

$$\sum_{k=1}^{K} \pi_k = 1$$

Expectation-maximization algorithm
putting things together

1. E step

   Compute responsibilities

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

2. M step

   Maximize likelihood given the responsibilities

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

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

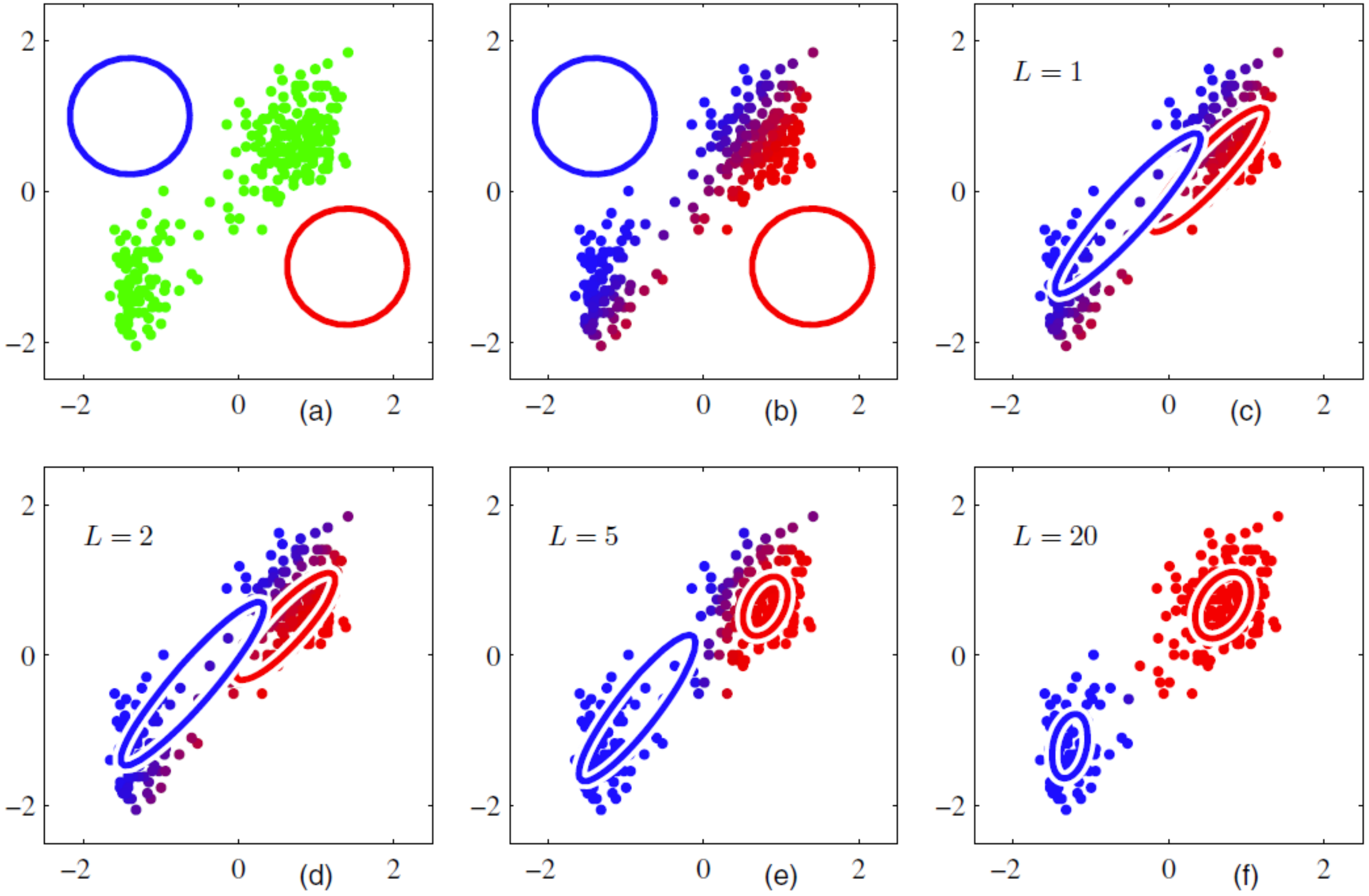
$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

3. Evaluate the log likelihood

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

4. Iterate if log likelihood has increased

Illustration of the EM algorithm



- Solution of K-means: