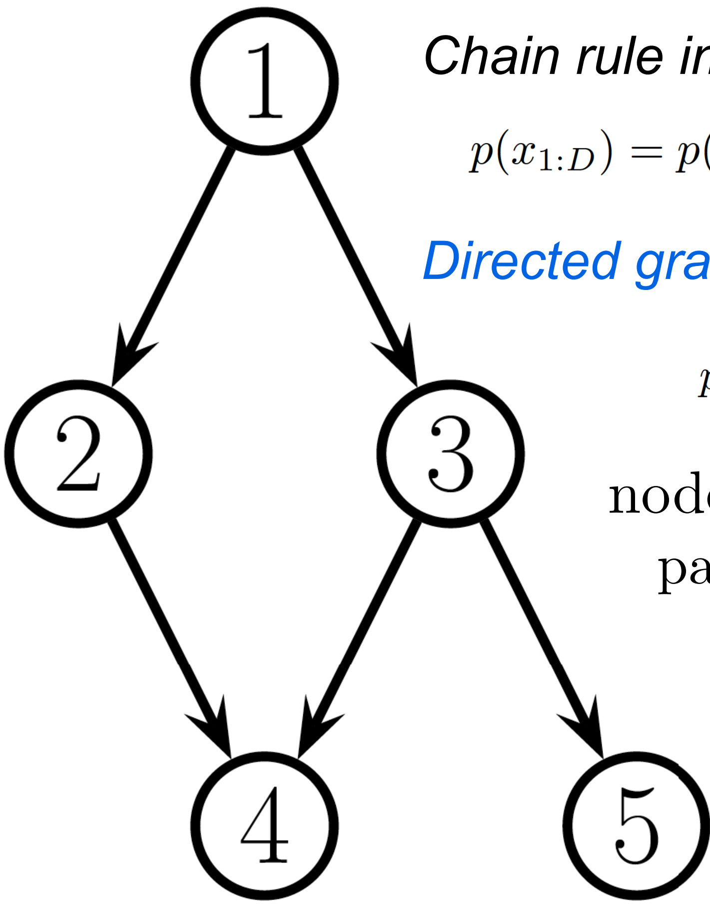
Introduction to Machine Learning

Brown University CSCI 1950-F, Spring 2012
Prof. Erik Sudderth

Lecture 19:
Directed Graphical Models
Expectation Maximization for Mixture Models

Many figures courtesy Kevin Murphy's textbook,
Machine Learning: A Probabilistic Perspective

Directed Graphical Models



Chain rule implies that any joint distribution equals:

$$p(x_{1:D}) = p(x_1)p(x_2|x_1)p(x_3|x_2,x_1)p(x_4|x_1,x_2,x_3)\dots p(x_D|x_{1:D-1})$$

Directed graphical model implies a restricted factorization:

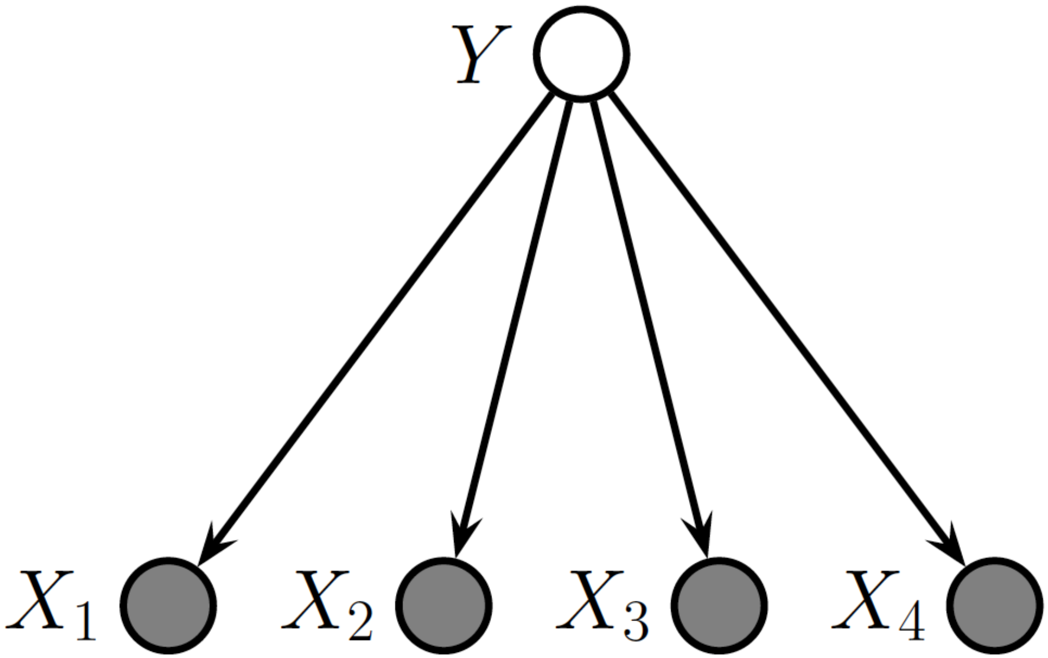
$$p(\mathbf{x}_{1:D}|G) = \prod_{t=1}^{D} p(x_t|\mathbf{x}_{\mathrm{pa}(t)})$$

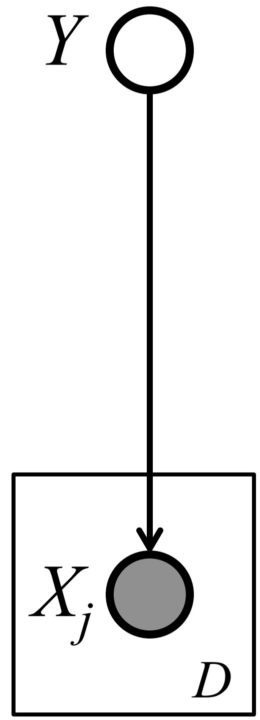
$$\begin{aligned} \text{nodes} &\rightarrow \text{random variables} \\ \mathrm{pa}(t) &\rightarrow \text{parents with edges pointing to node } t \end{aligned}$$

Valid for any directed acyclic graph (DAG): equivalent to dropping conditional dependencies in standard chain rule

$$\begin{aligned} p(\mathbf{x}_{1:5}) &= p(x_1)p(x_2|x_1)p(x_3|x_1,\cancel{x_2})p(x_4|\cancel{x_1},x_2,x_3)p(x_5|\cancel{x_1},\cancel{x_2},x_3,\cancel{x_4}) \\ &= p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2,x_3)p(x_5|x_3) \end{aligned}$$

Example: Shading & Plate Notation



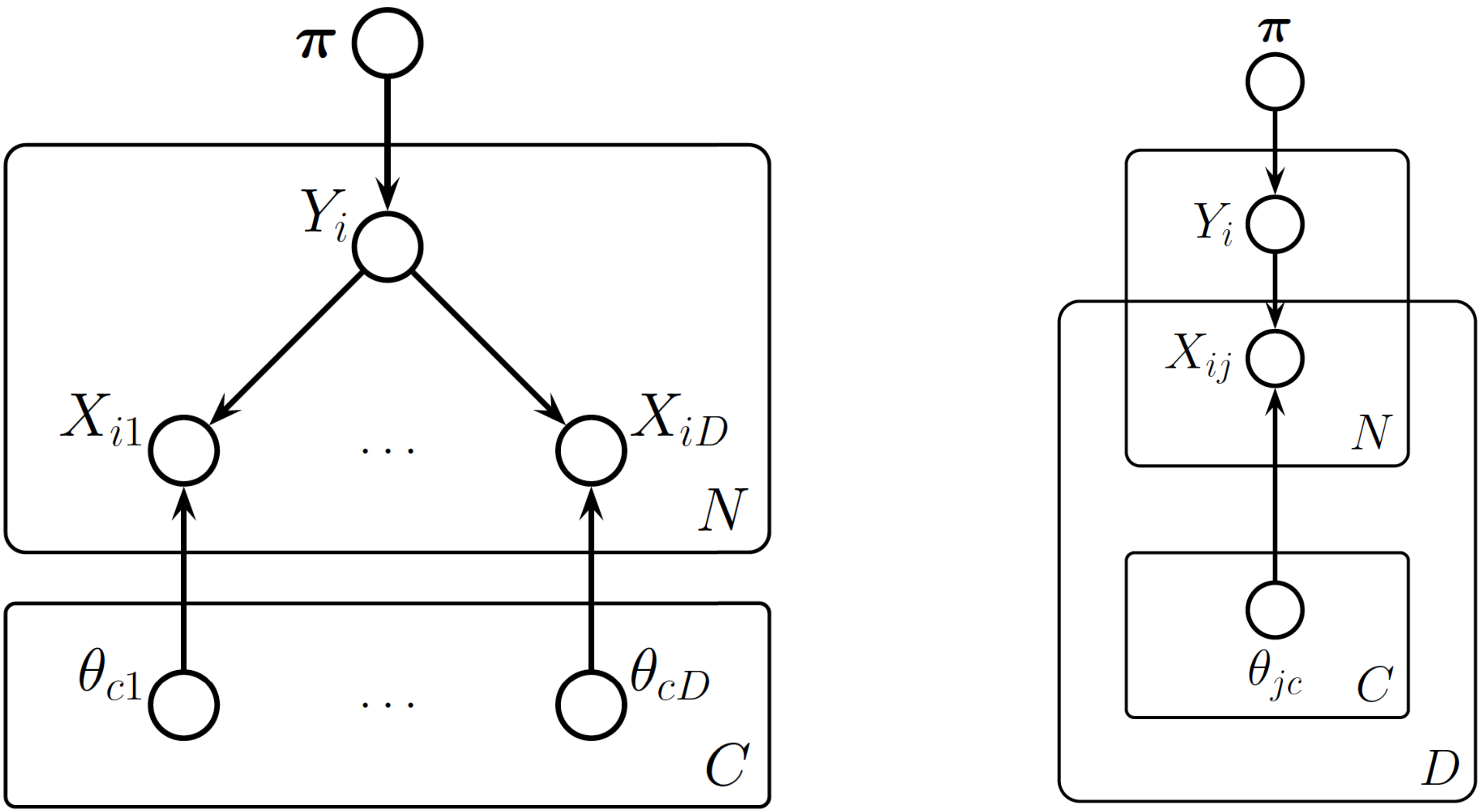


Plates denote replication of random variables

Naïve Bayes Inference: $$p(y, \mathbf{x}) = p(y) \prod_{j=1}^{D} p(x_j | y)$$

Convention: Shaded nodes are observed, open nodes are latent/hidden

Learning and Unknown Parameters

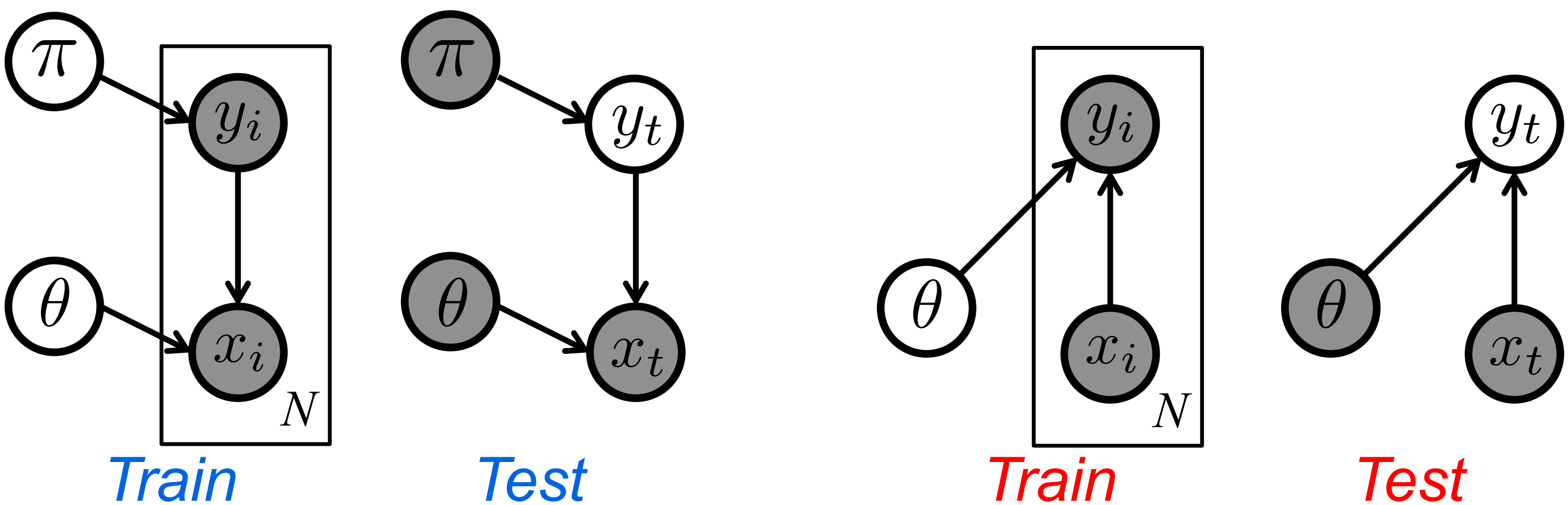


$$p(\pi)\left[\prod_{c=1}^{C}\prod_{j=1}^{D}p(\theta_{cj})\right]\prod_{i=1}^{N}\left[p(y_i \mid \pi)\prod_{j=1}^{D}p(x_{ij} \mid y_i, \theta_{j1}, \ldots, \theta_{jC})\right]$$

Supervised Learning

Generative ML or MAP Learning:

$$\max_{\pi,\theta} \; \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} [\log p(y_i \mid \pi) + \log p(x_i \mid y_i, \theta)]$$



Discriminative ML or MAP Learning:

$$\max_{\theta} \; \log p(\theta) + \sum_{i=1}^{N} \log p(y_i \mid x_i, \theta)$$

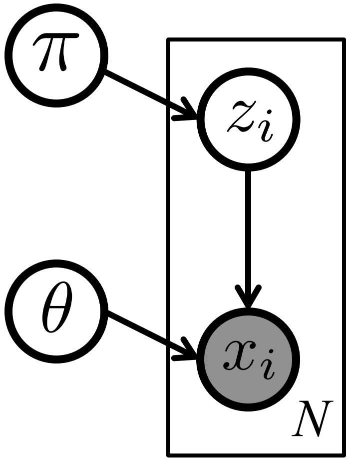
Unsupervised Learning

Clustering:

$$\max_{\pi,\theta} \; \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} \log \left[\sum_{z_i} p(z_i \mid \pi) p(x_i \mid z_i, \theta) \right]$$

Dimensionality Reduction:

$$\max_{\pi,\theta} \; \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} \log \left[\int_{z_i} p(z_i \mid \pi) p(x_i \mid z_i, \theta) \; dz_i \right]$$



- No notion of training and test data: labels are *never* observed
- As before, *maximize* posterior probability of model parameters
- For hidden variables associated with each observation, we *marginalize* over possible values rather than estimating
  - Fully accounts for uncertainty in these variables
  - There is one hidden variable per observation, so cannot perfectly estimate even with infinite data
- Must use generative model (discriminative degenerates)

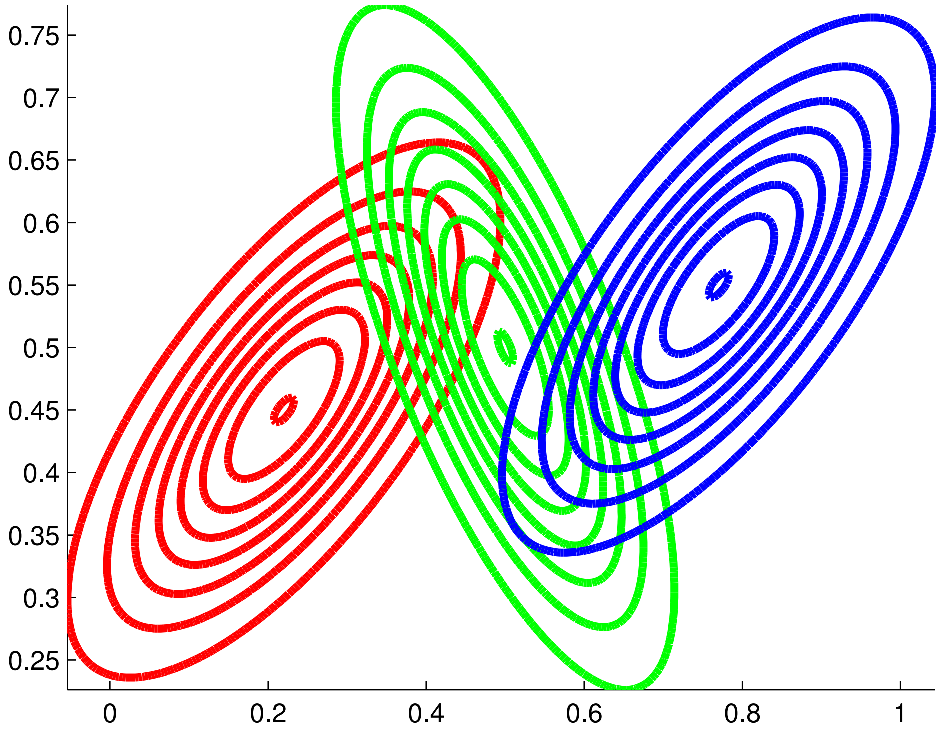
Gaussian Mixture Models

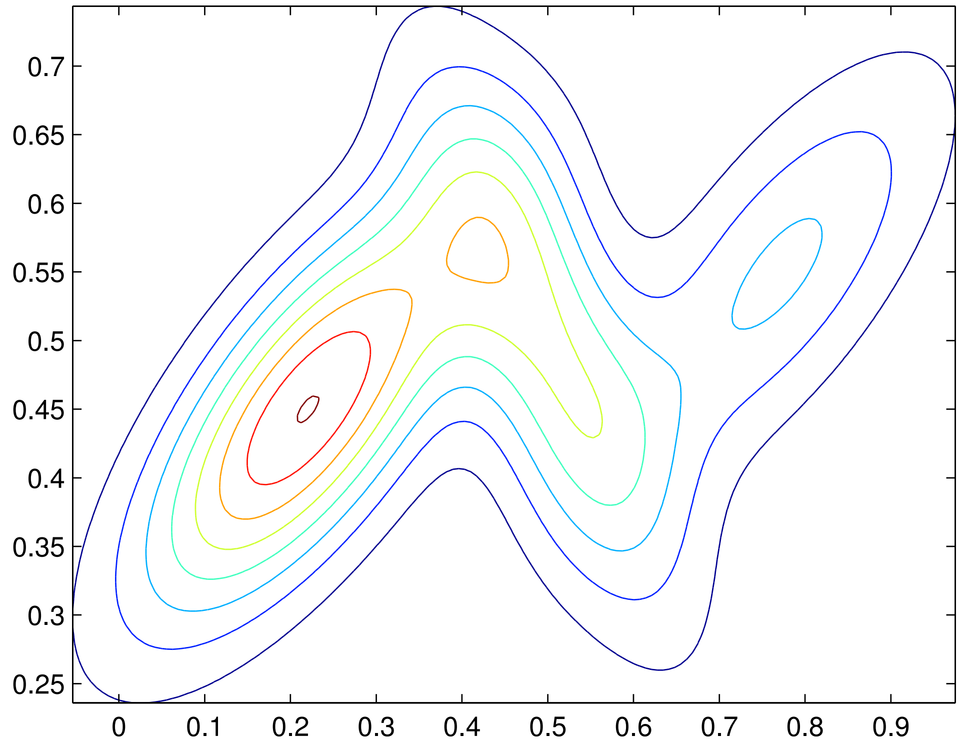
- Observed feature vectors: $x_i \in \mathbb{R}^d, \quad i = 1, 2, \ldots, N$
- Hidden cluster labels: $z_i \in \{1, 2, \ldots, K\}, \quad i = 1, 2, \ldots, N$
- Hidden mixture means: $\mu_k \in \mathbb{R}^d, \quad k = 1, 2, \ldots, K$
- Hidden mixture covariances: $\Sigma_k \in \mathbb{R}^{d \times d}, \quad k = 1, 2, \ldots, K$
- Hidden mixture probabilities: $\pi_k, \quad \sum_{k=1}^{K} \pi_k = 1$
- Gaussian mixture marginal likelihood:

$$p(x_i \mid \pi, \mu, \Sigma) = \sum_{z_i=1}^{K} \pi_{z_i} \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

$$p(x_i \mid z_i, \pi, \mu, \Sigma) = \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

Gaussian Mixture Models

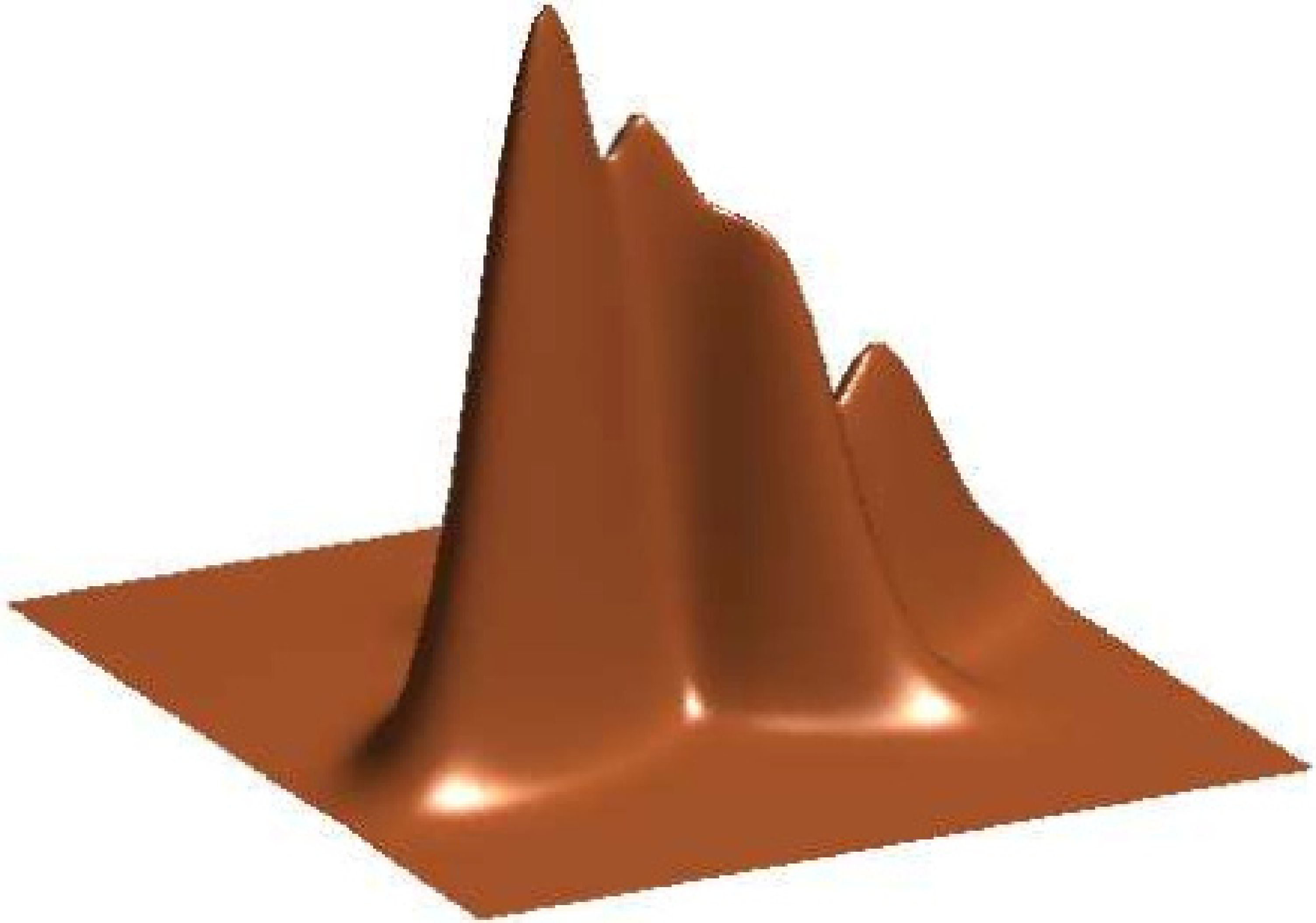




Mixture of 3 Gaussian Distributions in 2D

Contour Plot of Joint Density, Marginalizing Cluster Assignments

Gaussian Mixture Models



Surface Plot of Joint Density,
Marginalizing Cluster Assignments

Gaussian Discriminant Analysis

$y \longrightarrow$ class label in *{1,…,C}*, observed in training

$x \in \mathbb{R}^d \longrightarrow$ observed features to be used for classification

$$p(y, x \mid \pi, \theta) = p(y \mid \pi) p(x \mid y, \theta)$$

discriminant analysis is a generative classifier!

prior distribution

likelihood function

$$p(y \mid \pi) = \mathrm{Cat}(y \mid \pi)$$

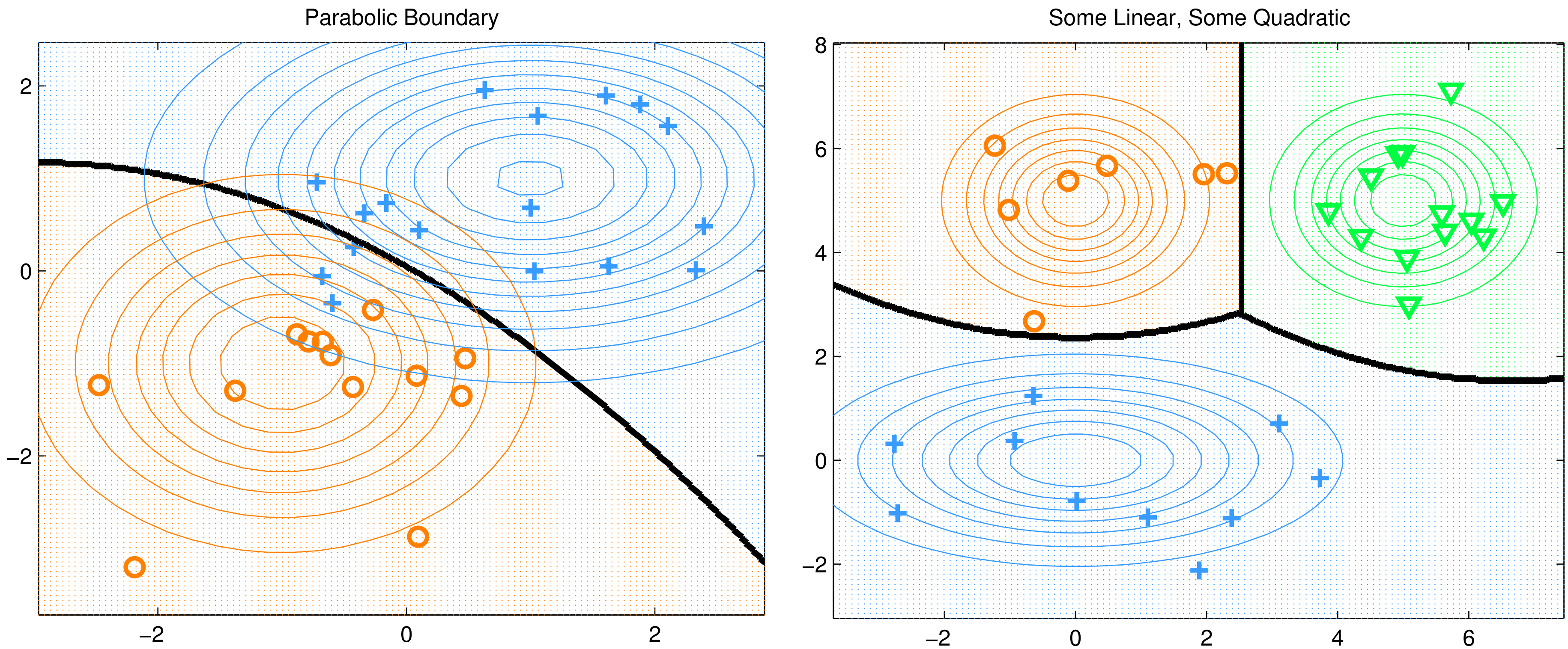
$$p(x \mid y = c, \theta) = \mathcal{N}(x \mid \mu_c, \Sigma_c) \qquad \theta_c = \{\mu_c, \Sigma_c\}$$

- Derive posterior distribution via Bayes' rule:

$$p(y = c \mid x, \theta, \pi) = \frac{p(y = c \mid \pi) p(x \mid y = c, \theta)}{\sum_{c'=1}^{C} p(y = c' \mid \pi) p(x \mid y = c', \theta)}$$

- Gaussian naïve Bayes model assumes diagonal covariances

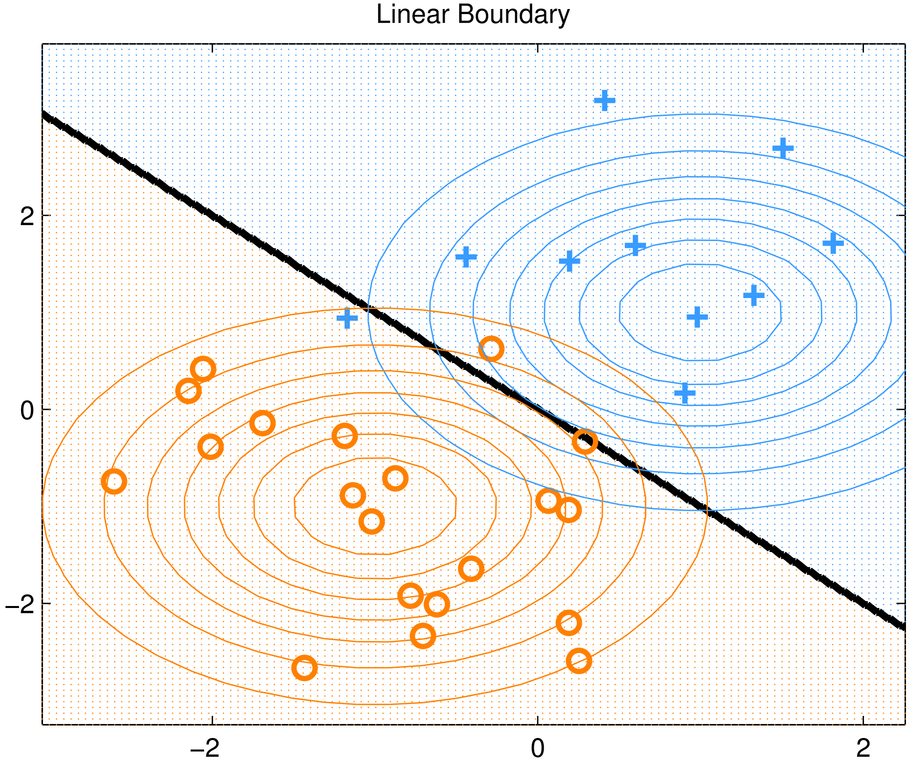
Quadratic Discriminant Analysis

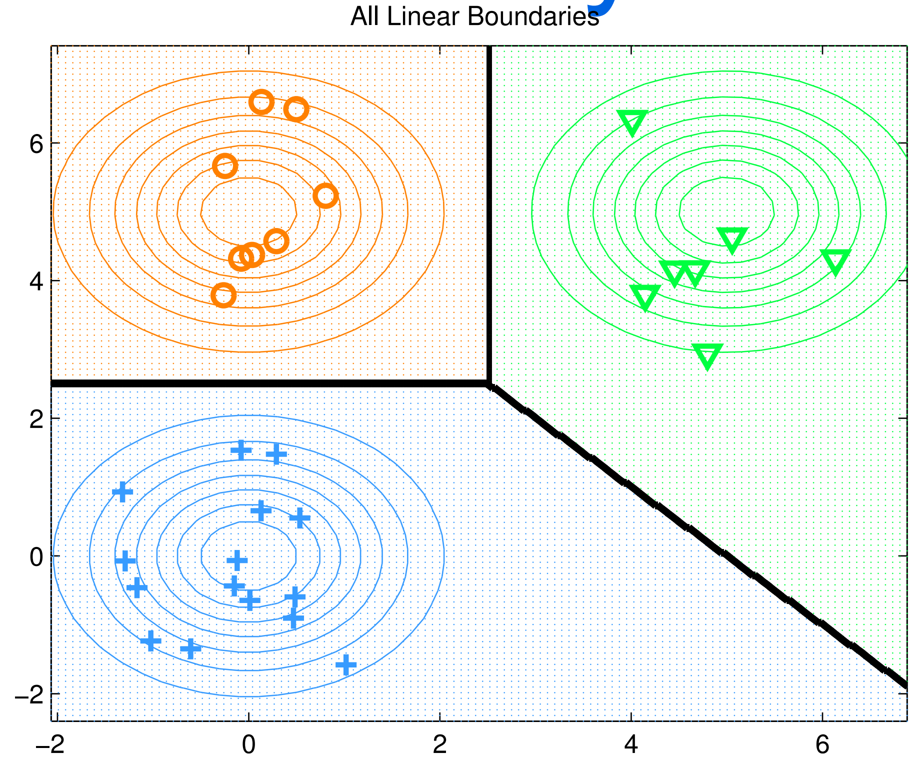


$$p(y=c|\mathbf{x},\boldsymbol{\theta}) = \frac{\pi_c |2\pi\boldsymbol{\Sigma}_c|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_c)^T\boldsymbol{\Sigma}_c^{-1}(\mathbf{x}-\boldsymbol{\mu}_c)\right]}{\sum_{c'} \pi_{c'} |2\pi\boldsymbol{\Sigma}_{c'}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{c'})^T\boldsymbol{\Sigma}_{c'}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{c'})\right]}$$

Optimal decision boundaries are quadratic functions

Linear Discriminant Analysis



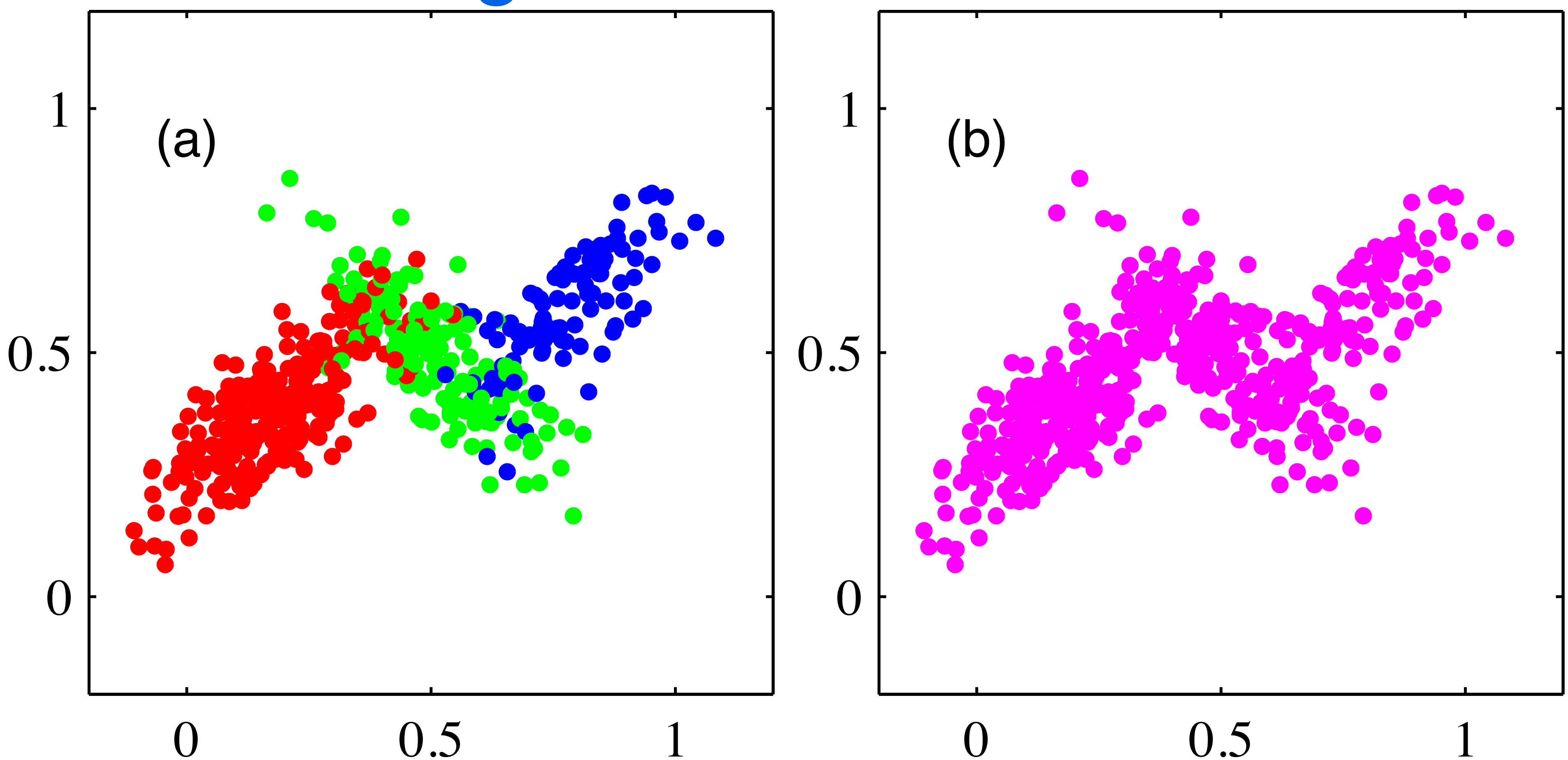


$$p(y=c|\mathbf{x},\boldsymbol{\theta}) \propto \pi_c \exp\left[\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c\right]$$

$$= \exp\left[\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log\pi_c\right]\exp[-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}]$$

Optimal decision boundaries are linear functions if $\Sigma_c = \Sigma$

Clustering with Gaussian Mixtures



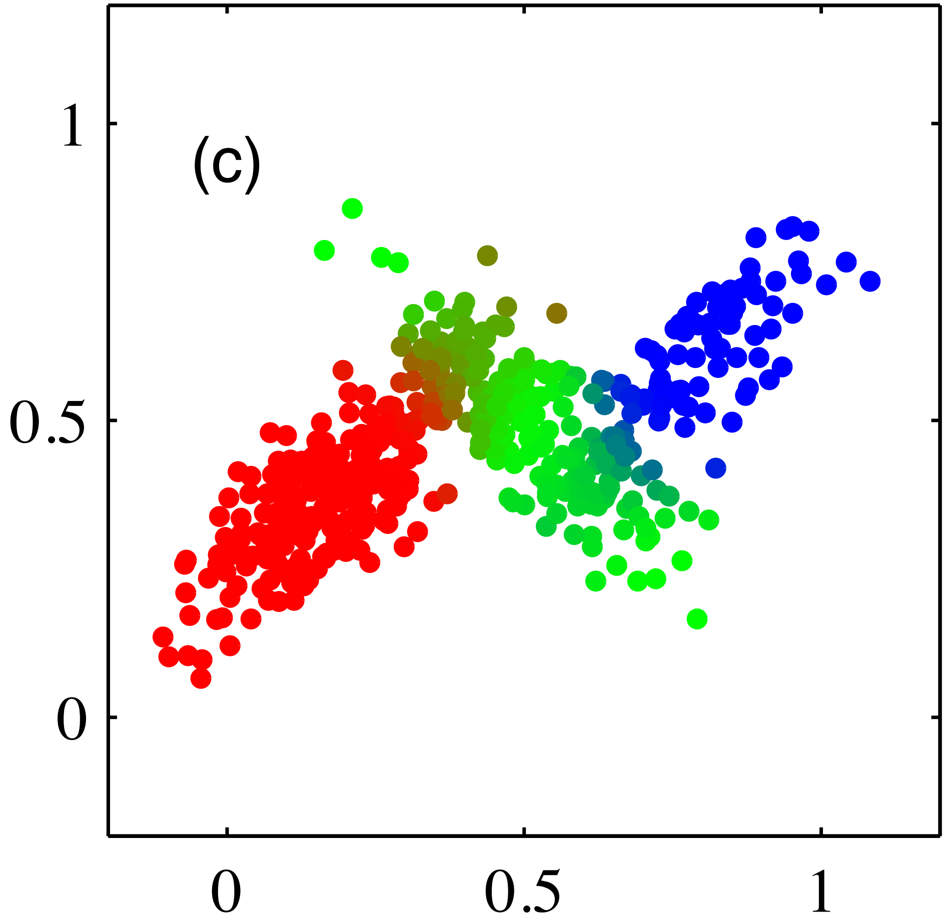
Complete Data Labeled by True Cluster Assignments

Incomplete Data: Points to be Clustered

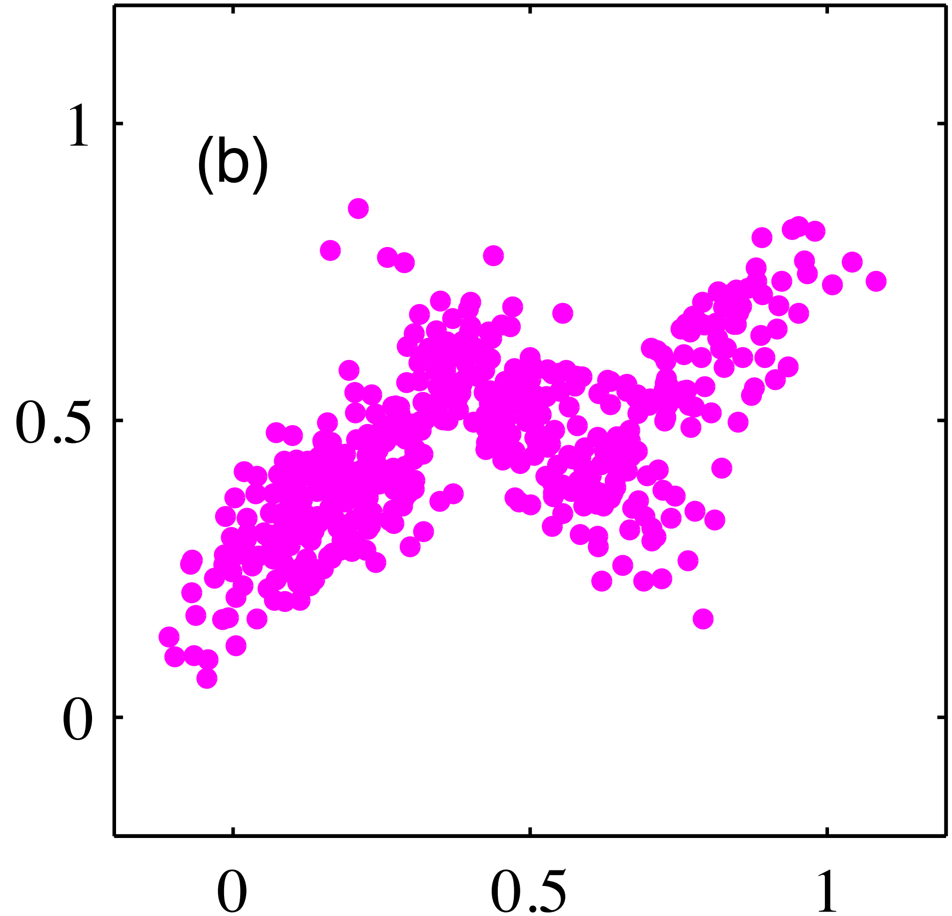
With complete data, learning is Gaussian discriminant analysis.

C. Bishop, Pattern Recognition & Machine Learning

Inference Given Cluster Parameters



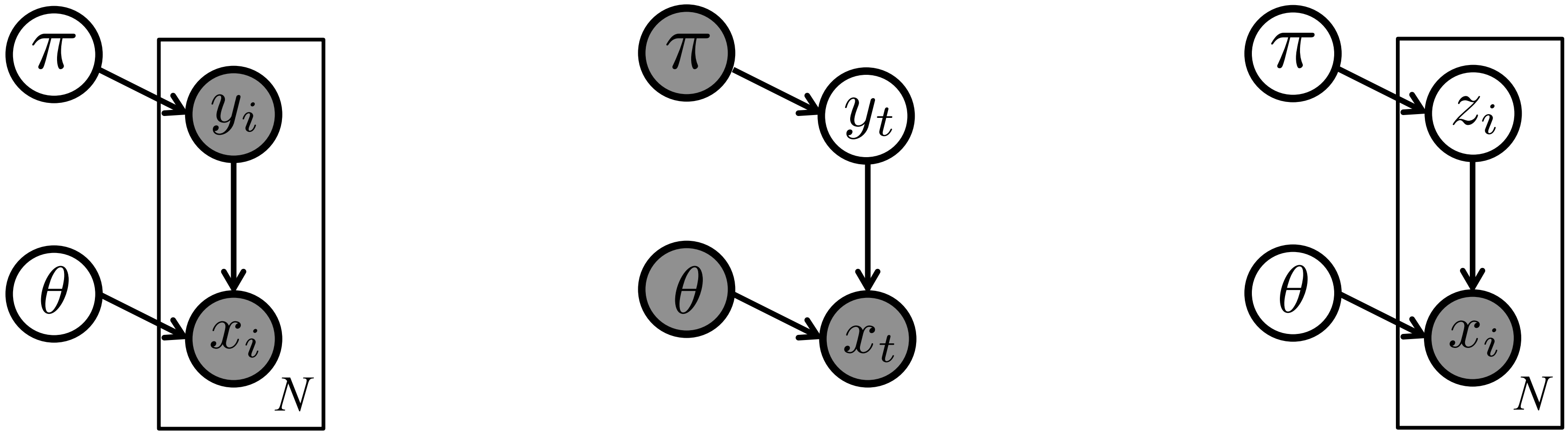
Posterior Probabilities of Assignment to Each Cluster



Incomplete Data: Points to be Clustered

$$r_{ik} = p(z_i = k \mid x_i, \pi, \theta) = \frac{\pi_k p(x_i \mid \theta_k)}{\sum_{\ell=1}^{K} \pi_\ell p(x_i \mid \theta_\ell)}$$

Unsupervised Learning Algorithms



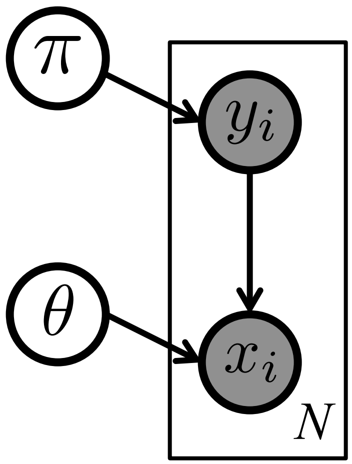
Supervised Training | *Supervised Testing* | *Unsupervised Learning*

π, θ → *parameters (define cluster location and shape)*

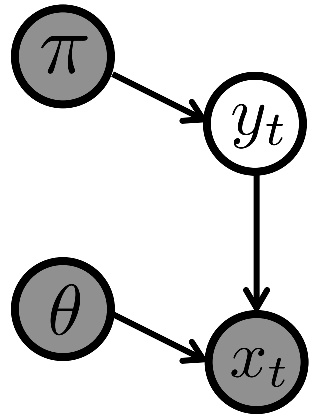
$z_1, \ldots, z_N$ → *hidden data (group observations into clusters)*

- **Initialization:** Randomly select starting parameters
- **Estimation:** Given parameters, find likely hidden data
  - Equivalent to *testing* phase of supervised learning
- **Learning:** Given hidden & observed data, find likely parameters
  - Equivalent to *training* phase of supervised learning
- **Iteration:** Alternate estimation & learning until convergence

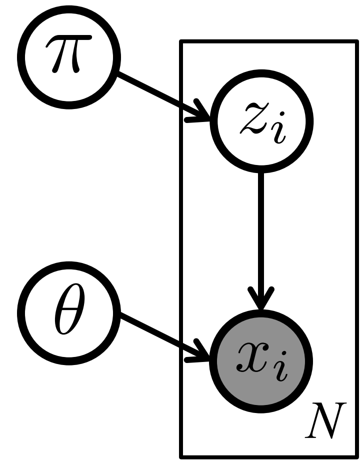
Expectation Maximization (EM)



Supervised Training



Supervised Testing



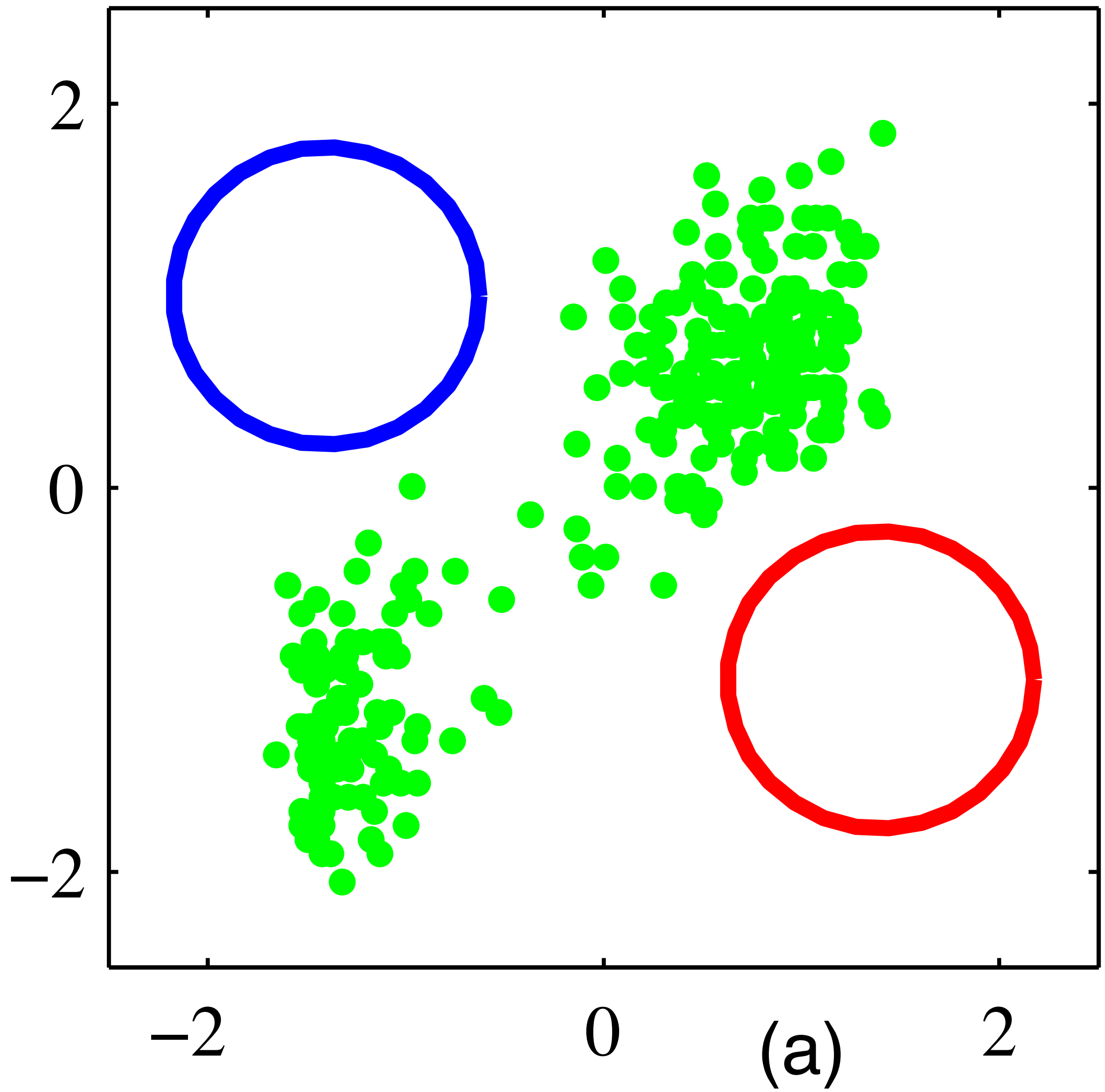
Unsupervised Learning

π, θ ⟶ *parameters (define cluster location and shape)*

$z_1, \ldots, z_N$ ⟶ *hidden data (group observations into clusters)*

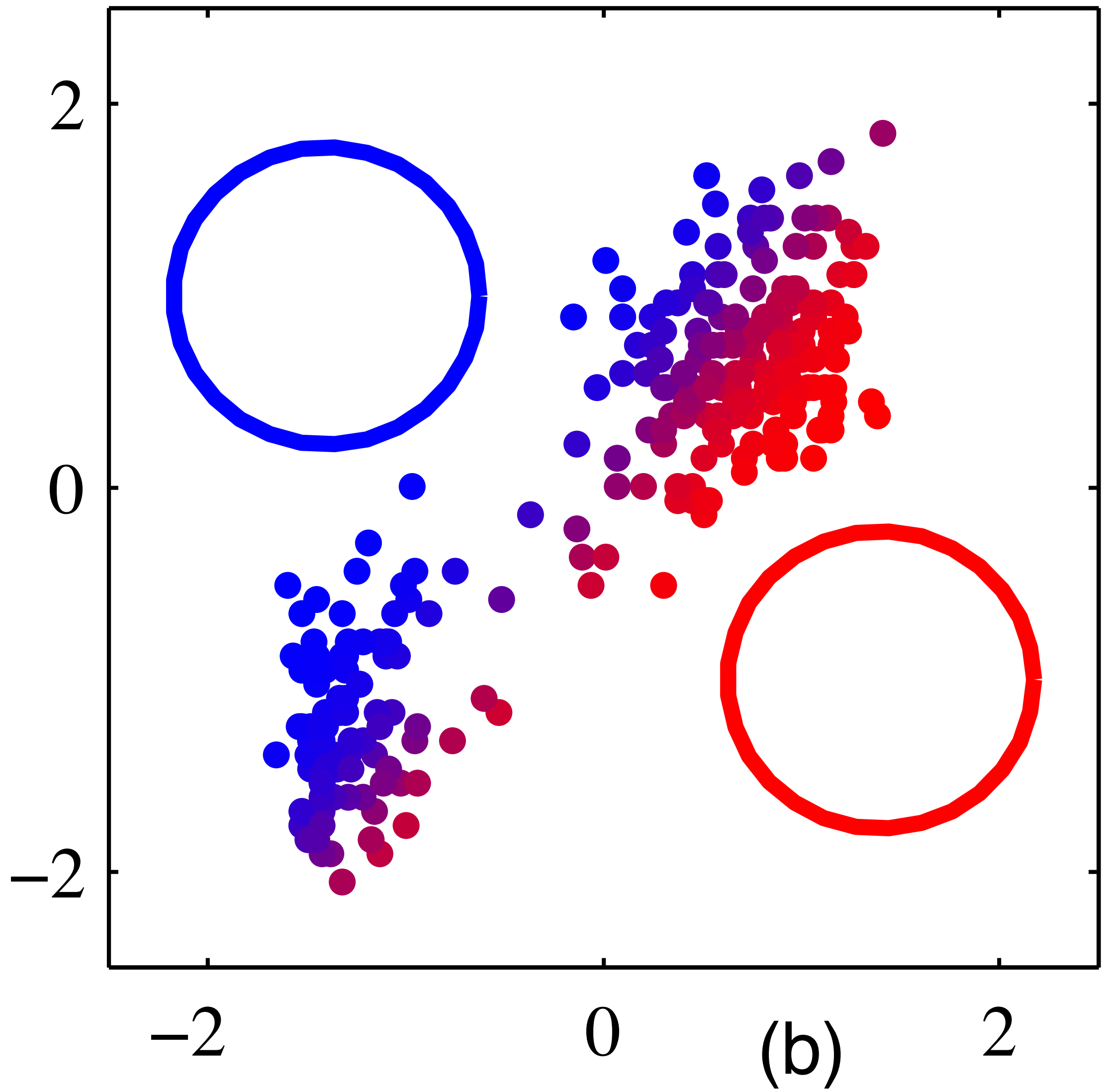
- **Initialization:** Randomly select starting parameters
- **E-Step:** Given parameters, find posterior of hidden data
  - Equivalent to test inference of full posterior distribution
- **M-Step:** Given posterior distributions, find likely parameters
  - Distinct from supervised ML/MAP, but often still tractable
- **Iteration:** Alternate E-step & M-step until convergence

EM Algorithm



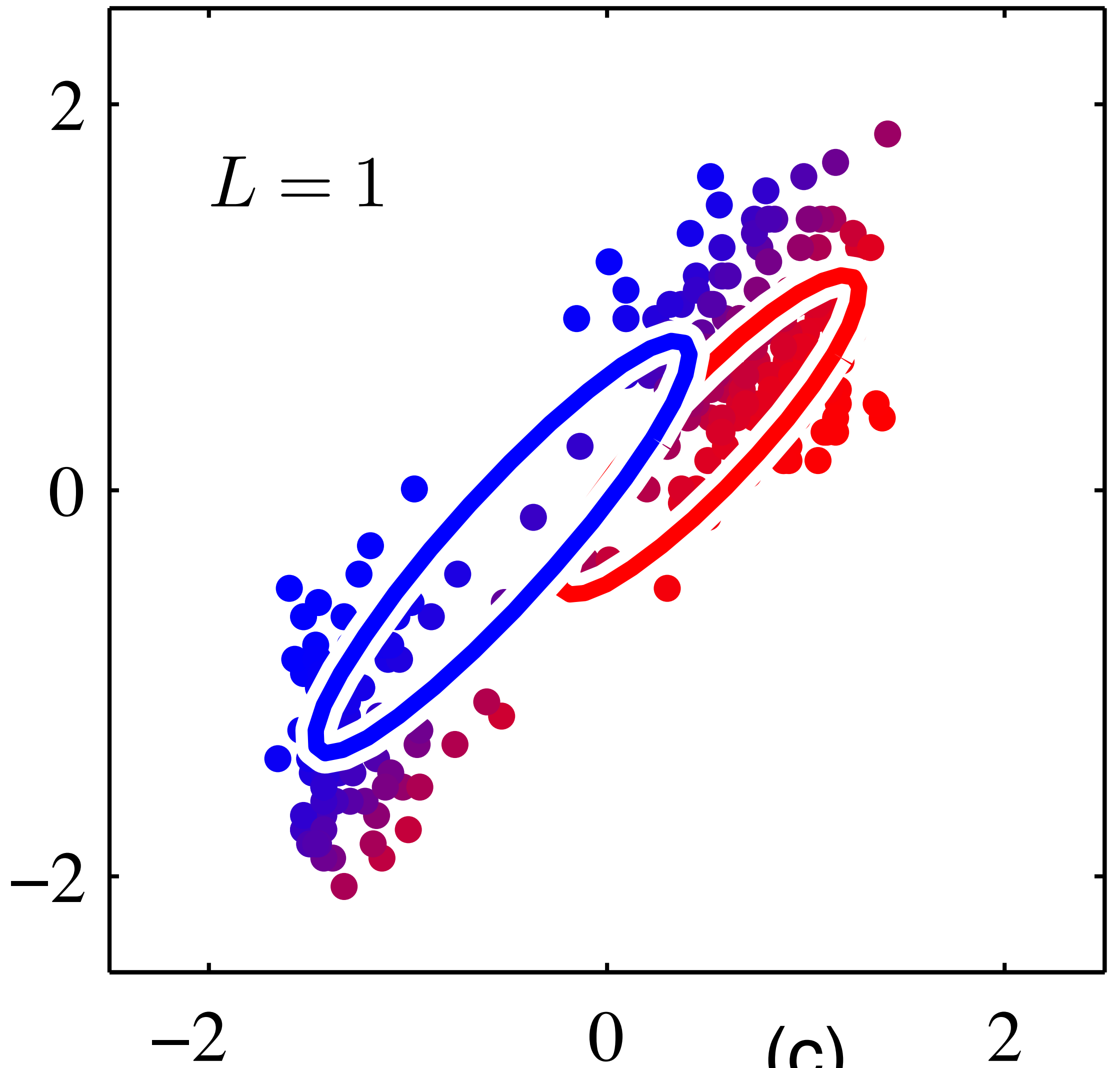
C. Bishop, Pattern Recognition & Machine Learning

EM Algorithm



C. Bishop, Pattern Recognition & Machine Learning

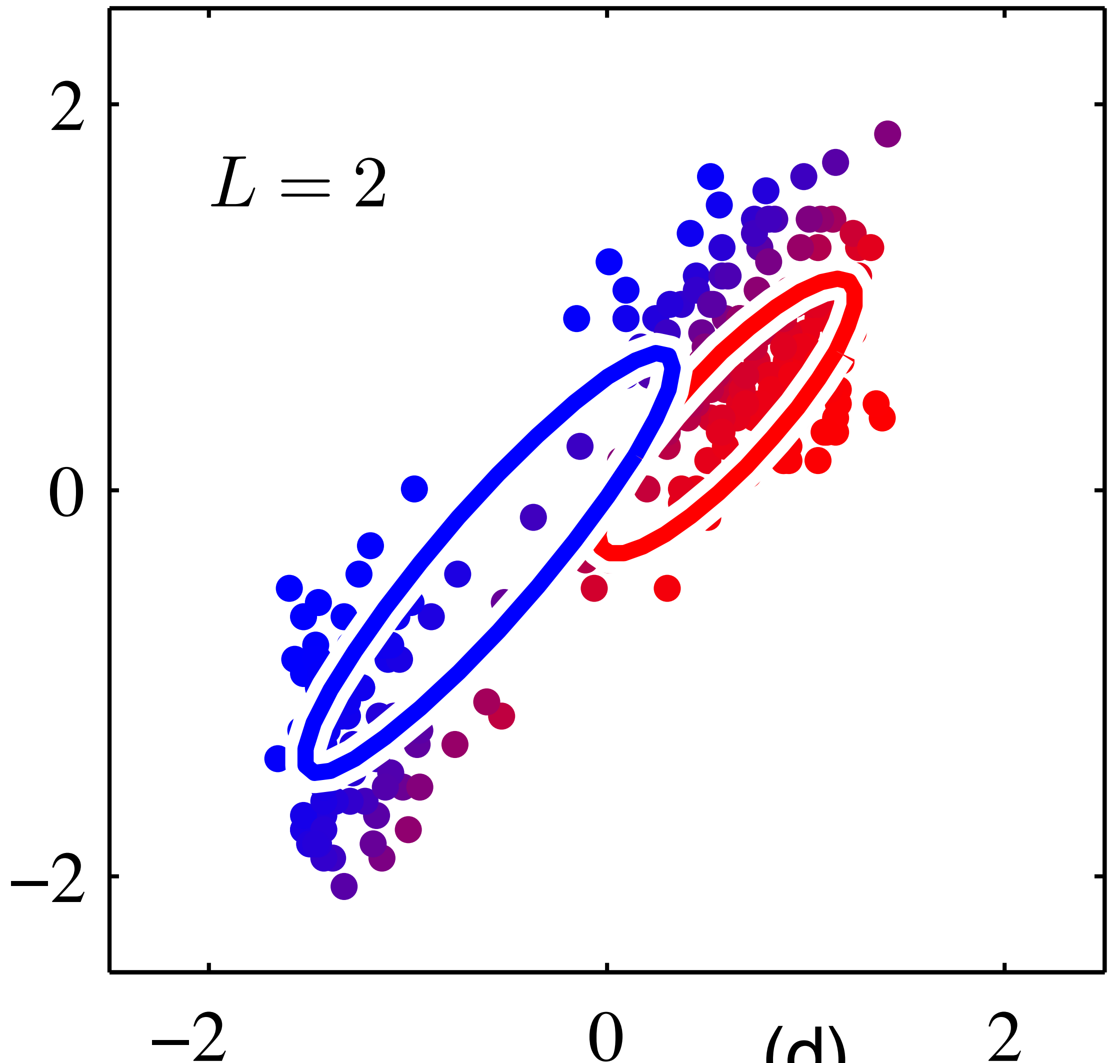
EM Algorithm



(c)

C. Bishop, Pattern Recognition & Machine Learning

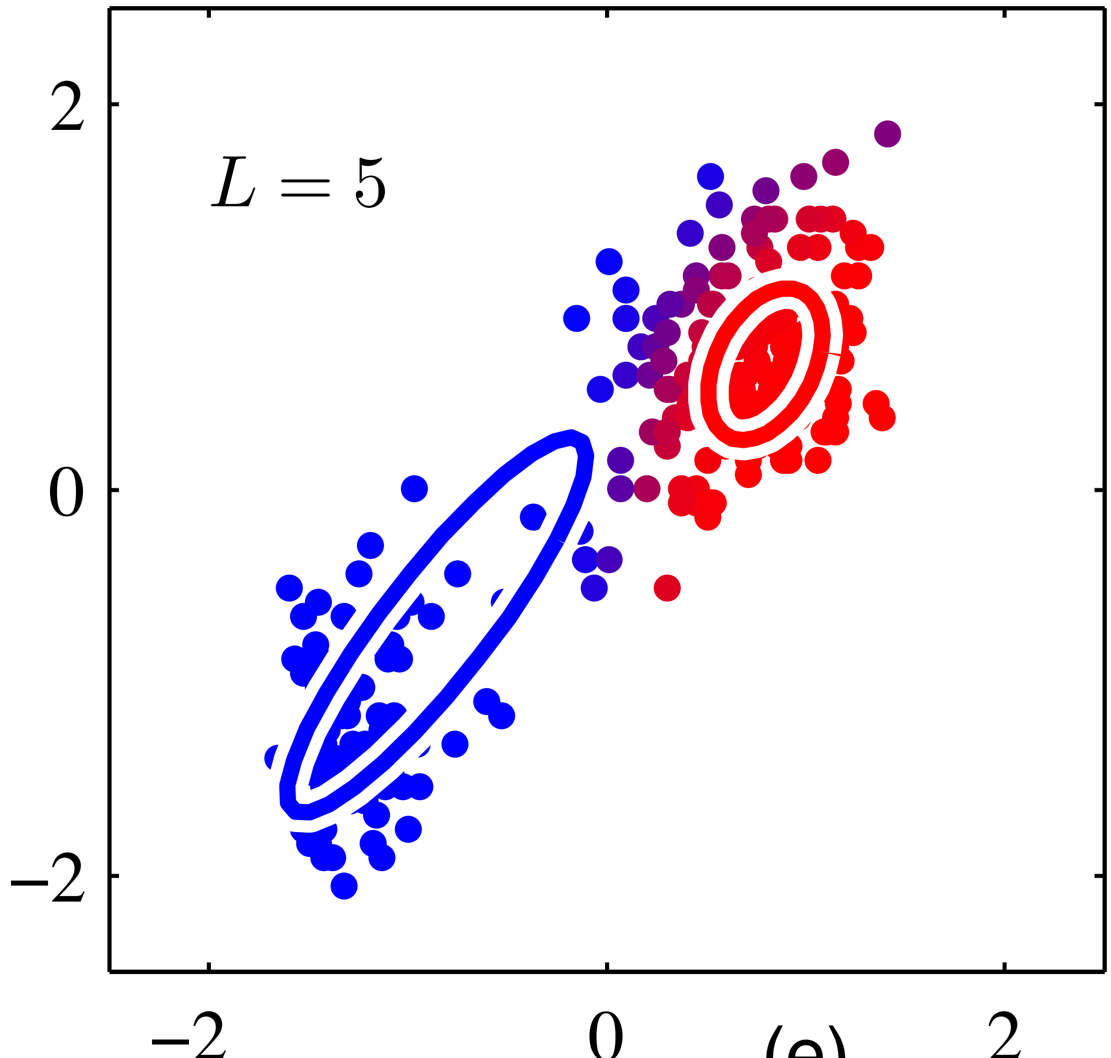
EM Algorithm



(d)

C. Bishop, Pattern Recognition & Machine Learning

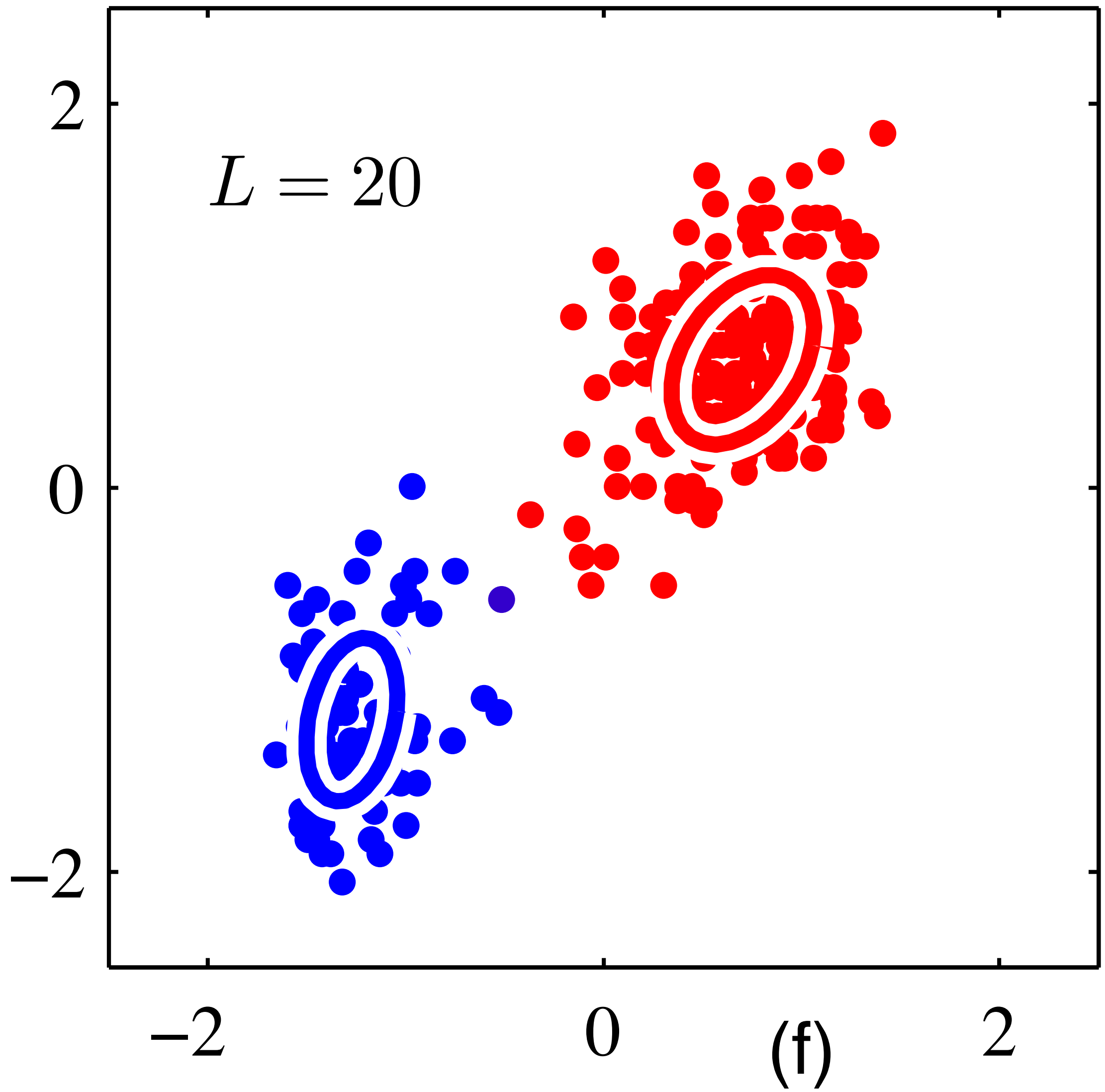
EM Algorithm



(e)

C. Bishop, Pattern Recognition & Machine Learning

EM Algorithm



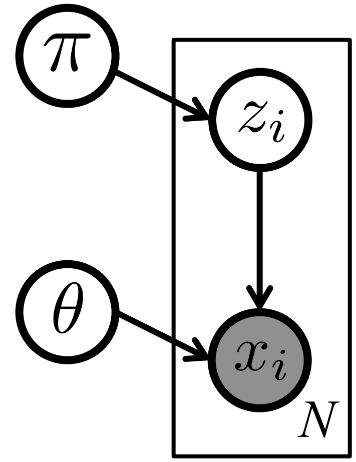
C. Bishop, Pattern Recognition & Machine Learning

EM for (Gaussian) Mixture Models

$$p(z_i \mid \pi) = \mathrm{Cat}(z_i \mid \pi)$$

$$p(x_i \mid z_i, \mu, \Sigma) = \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

$$p(x_i \mid \pi, \mu, \Sigma) = \sum_{z_i=1}^{K} \pi_{z_i} \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$



E-Step:

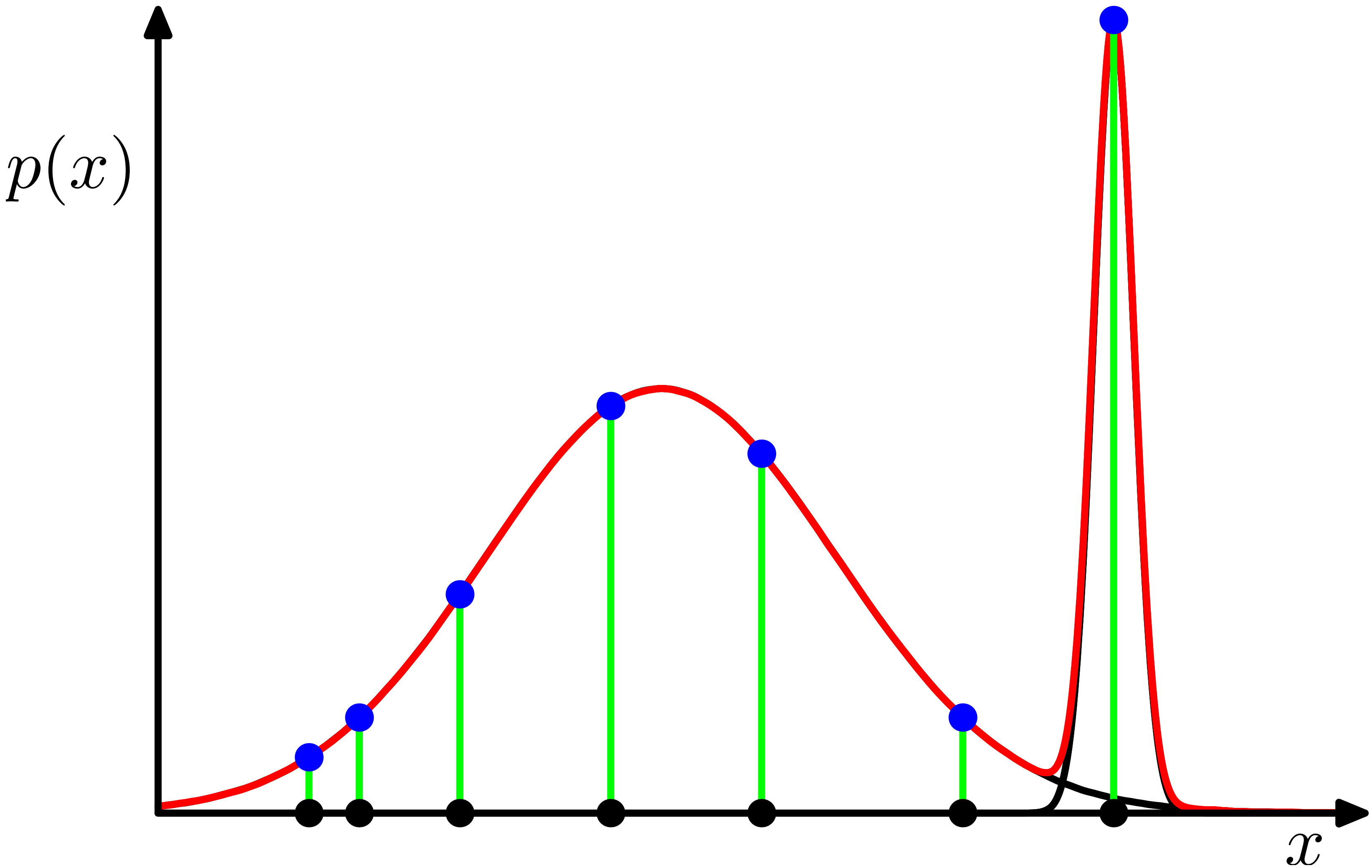
$$r_{ik} = p(z_i = k \mid x_i, \pi, \theta) = \frac{\pi_k p(x_i \mid \theta_k)}{\sum_{\ell=1}^{K} \pi_\ell p(x_i \mid \theta_\ell)}$$

M-Step:

$$\hat{\theta}_k = \arg\max_{\theta_k} \left[\log p(\theta_k) + \sum_{i=1}^{N} r_{ik} \log p(x_i \mid \theta_k) \right]$$

What happens when posteriors are perfectly confident: $r_{ik} \in \{0, 1\}$

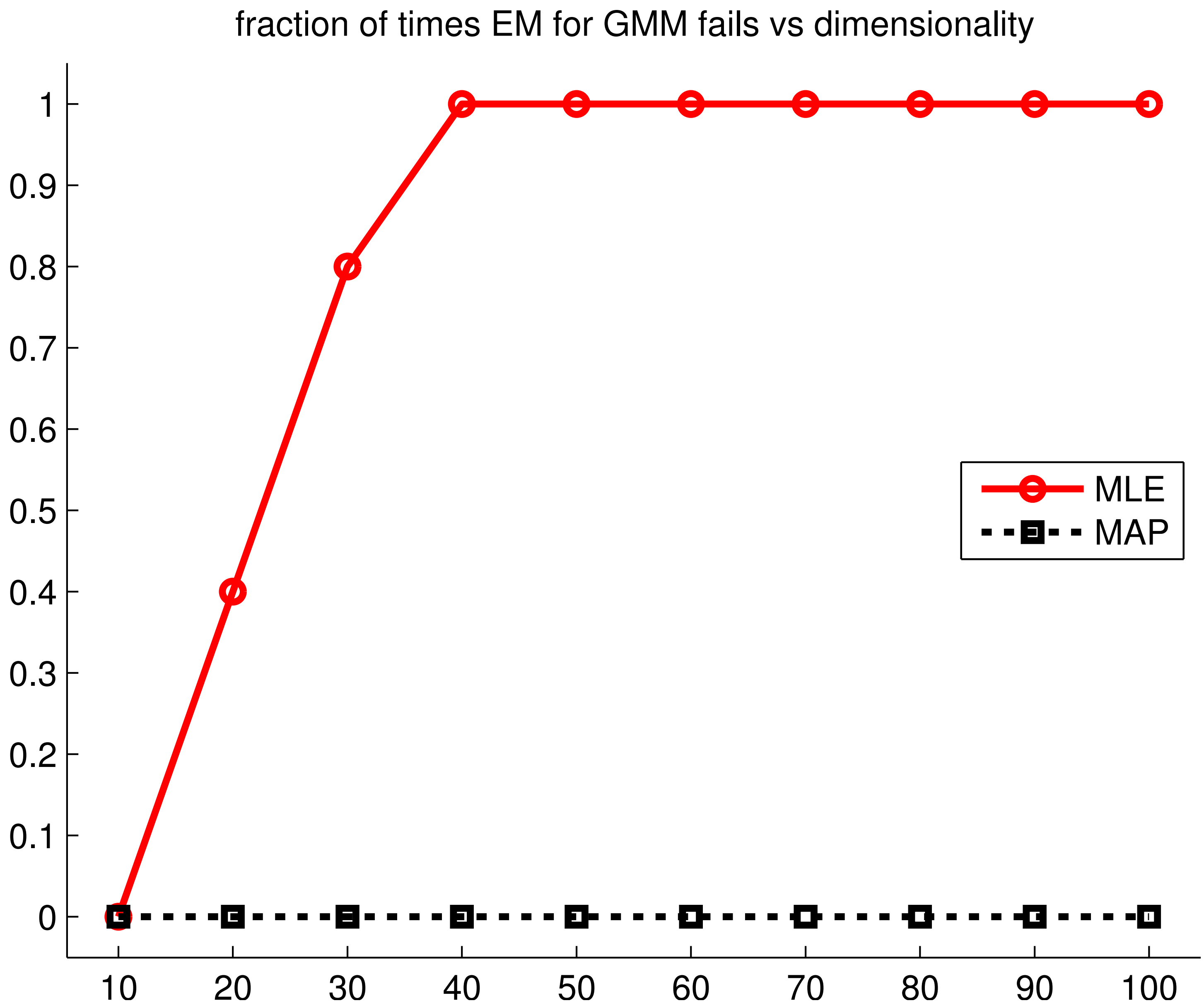
Singularities: ML for Gaussian Mixtures



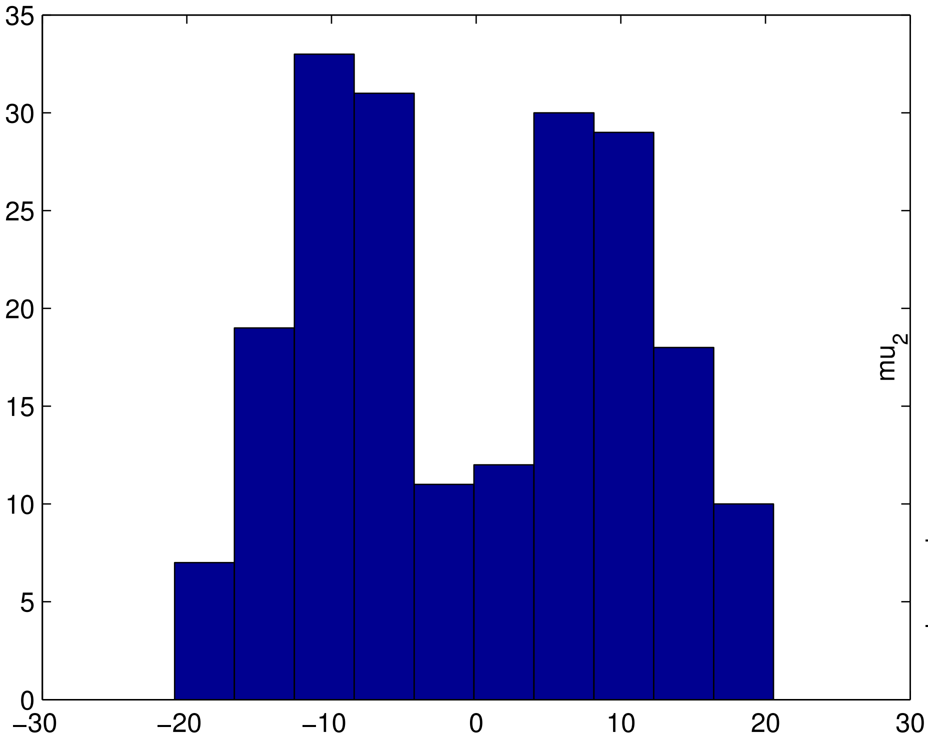
We are hoping EM will find a good local optimum…

C. Bishop, Pattern Recognition & Machine Learning

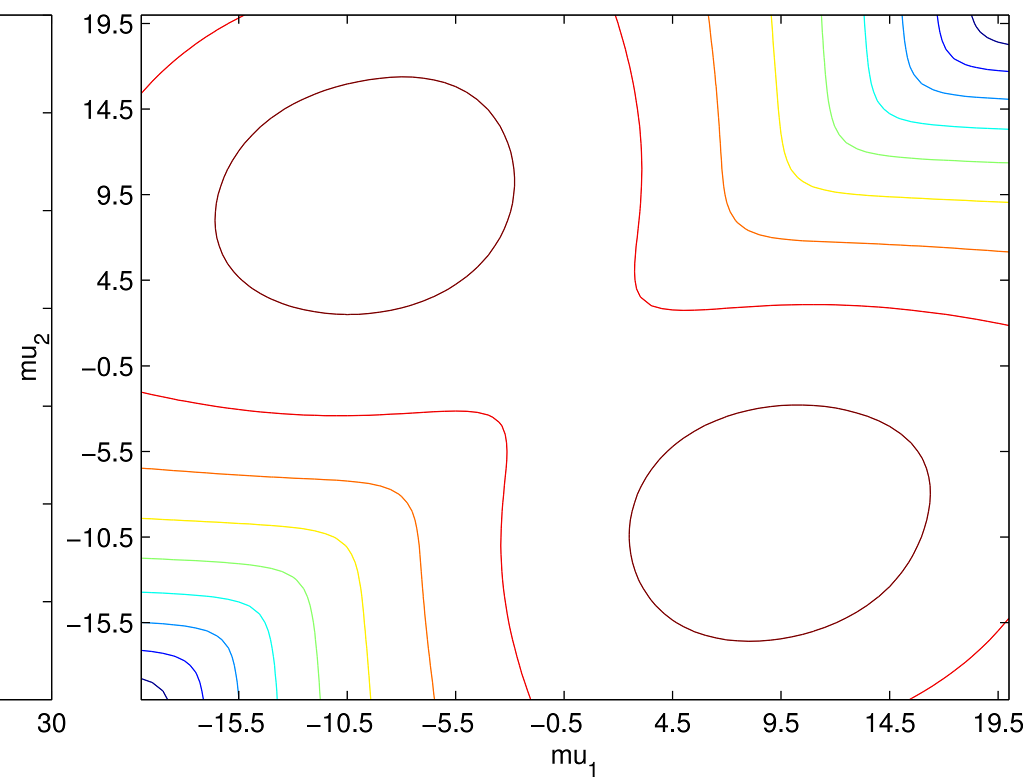
Numerical Instability: Gaussian Mixtures



Label Switching in Mixture Models



Histogram of 200 samples from a mixture of two 1D Gaussians



Two-component Gaussian mixture likelihood surface as function of means, for fixed variances