LECTURE 14: Mixture models and EM

- **Supervised vs. unsupervised learning**
- **Mixture models**
- **Expectation maximization (EM)**

Supervised vs. unsupervised learning (1)

- **The pattern recognition methods covered in class up to this point have focused on the issue of classification**
  - A pattern consisted of a pair of variables $\{x,\omega\}$ where
    - **x** was a collection of observations or features (feature vector)
    - ω was the concept behind the observation (label)

Supervised vs. unsupervised learning (1)

- Such pattern recognition problems are called supervised (training with a teacher)

  since the system is given BOTH the feature vector and the correct answer

- **In the next three lectures we investigate a number of methods that operate on unlabeled data**

Supervised vs. unsupervised learning (1)

- Given a collection of feature vectors $X=\{x^{(1}, x^{(2}, \ldots, x^{(N}\}$ without class labels ω_i, these methods attempt to build a model that captures the structure of the data
- These methods are called unsupervised (training without a teacher) since they are not provided the correct answer

Supervised vs. unsupervised learning (2)

- **Although unsupervised learning methods may appear to have limited capabilities, there are several reasons that make them extremely useful**
  - Labeling large data sets can be a costly procedure (i.e., speech recognition)
  - Class labels may not be known beforehand (i.e., data mining)

Supervised vs. unsupervised learning (2)

- Large datasets can be compressed by finding a small set of prototypes (kNN)

- **The supervised and unsupervised paradigms comprise the vast majority of pattern recognition problems**

Supervised vs. unsupervised learning (2)

- A third approach, known as reinforcement learning, uses a reward signal (real- valued or binary) to tell the learning system how well it is performing
  - In reinforcement learning, the goal of the learning system (or agent) is to learn a mapping from states onto actions (an action policy) that maximizes the total reward

Classification of unsupervised learning methods

- **Parametric (mixture models)**
  - These methods model the underlying class-conditional densities with a mixture of parametric densities, and the objective is to find the model parameters

$$P(x \mid \theta) = \sum_{i=1}^{C} P(x_i \mid \omega_i, \theta_i) P(\omega_i)$$

  - These methods are closely related to parameter estimation (Lecture 6)
    - Mixture models are the subject of this lecture

Classification of unsupervised learning methods

- **Non-parametric (clustering)**
  - No assumptions are made about the underlying densities, instead we seek a partition of the data into clusters
    - These methods are typically referred to as clustering, and will be the subject of the next two lectures:
    - Lecture 15 will focus on statistical clustering
    - Lecture 16 will deal with connectionist approaches

Classification of unsupervised learning methods

- **There are two reasons why we cover mixture models at this point**
  - The solution to the mixture problem (the EM algorithm) is also used for Hidden Markov Models, which will be introduced in just a few lectures
  - A particular form of the mixture model problem leads to the most widely used clustering method: the k-means algorithm (a.k.a. vector quantization)

Mixture models (1)

- **Consider the old problem of modeling a pdf given a dataset of examples**

  $X=\{x^{(1}, x^{(2}, \ldots, x^{(N}\}$

  - If the form of the underlying pdf was known (e.g. Gaussian), the problem could be solved using Maximum Likelihood (Lecture 6)

Mixture models (1)

- If the form of the pdf was unknown, the problem had to be solved with non-parametric density estimation methods such as Parzen windows (Lectures 7-8)

- **We will now consider an alternative density estimation method: modeling the pdf with a mixture of parametric densities**

Mixture models (2)

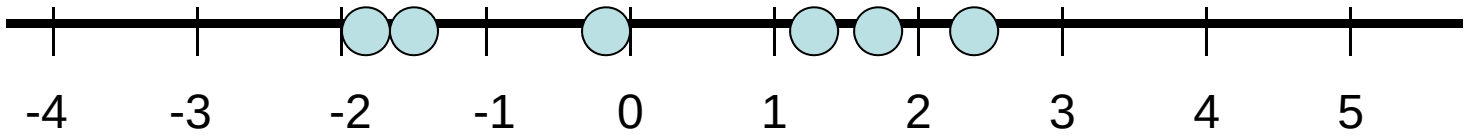
- **The mixture model problem can be posed in terms of the ML criterion**
  - Given a dataset of examples $X=\{x^{(1}, x^{(2}, \ldots, x^{(N}\}$, find the parameters of the model that maximize log likelihood of the data

$$\hat{\theta} = \text{argmax}\left[p(X \mid \theta)\right] = \text{argmax}\left[\sum_{n=1}^{N} \log p\left(x^{(n} \mid \theta\right)\right]$$

$$= \text{argmax}\left[\sum_{n=1}^{N} \log \sum_{c=1}^{C} p(x^{(n} \mid \theta_c)P(\omega_c)\right]$$

Motivating example

Data: $x = (x_1, x_2, \ldots, x_N)$



OBJECTIVE: Fit mixture of Gaussian model with C=2 components

$$\theta^* = argmax_\theta \prod_{n=1}^{N} p(x_i|\theta)$$

Model:

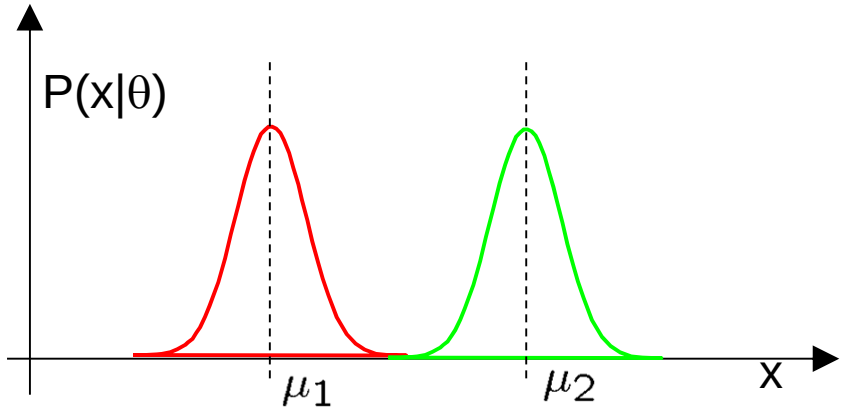
$$p(x_i|\theta) = \sum_c p(c|\theta)\, p(x_i|c,\theta)$$
$$= \sum_c w(c)\, N(x_i|\mu_c, \sigma_c)$$

where $\sum_{c=1}^{C} w(c) = 1$

Parameters: $\theta = \{w, \mu, \sigma\}$

keep w, σ fixed

i.e. only estimate μ

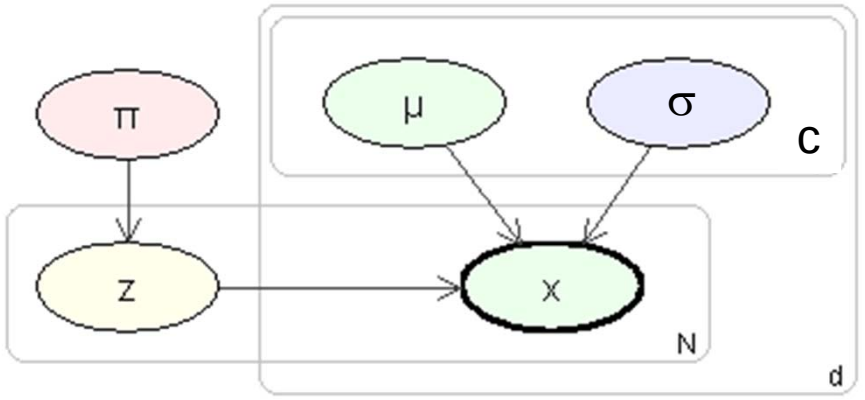


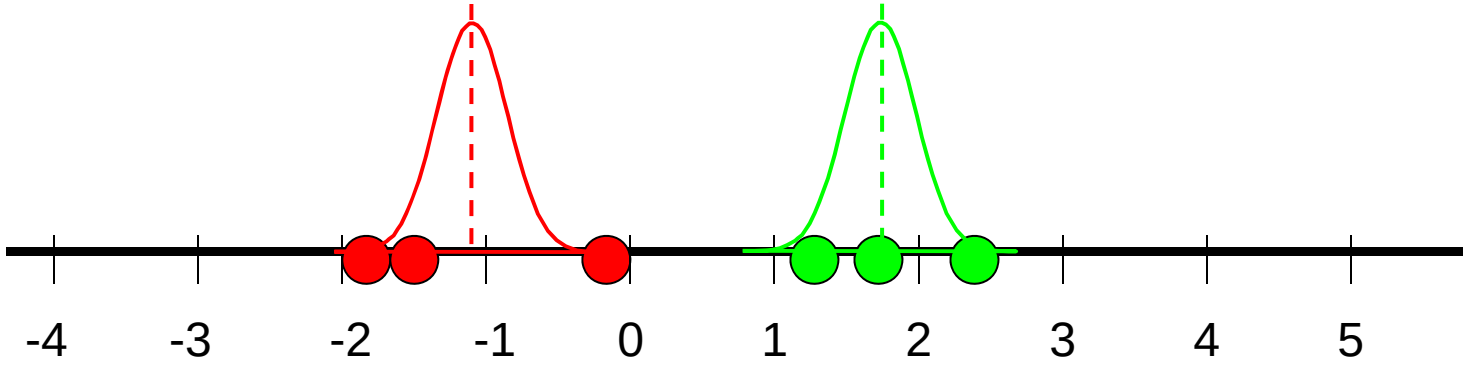
Probabilistic model

Imagine model generating data

Need to introduce label, z, for each data point

Label is called a *latent* variable
also called *hidden, unobserved, missing*





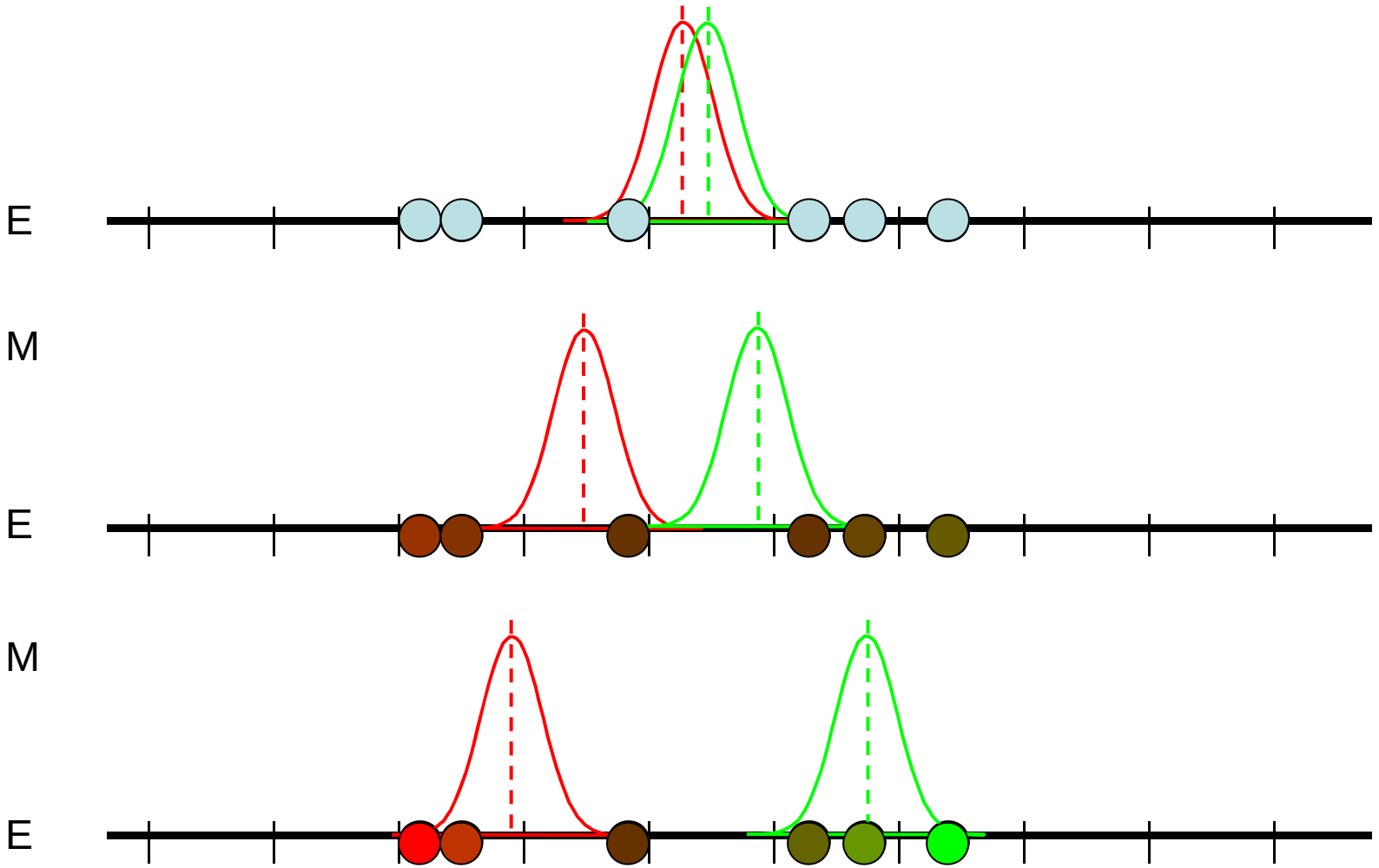
Simplifies the problem:

if we knew the labels, we can decouple the components as estimate parameters separately for each one

Intuition of EM

E-step: Compute a *distribution* on the labels of the points, using current parameters

M-step: Update parameters using current guess of label distribution.



The Expectation-Maximization algorithm (1)

- **The EM is a general method for finding the ML estimate of the parameters of a pdf when the data has missing values**
  - There are two main applications of the EM algorithm
    - When the data indeed has incomplete, missing or corrupted values as a result of a faulty observation process

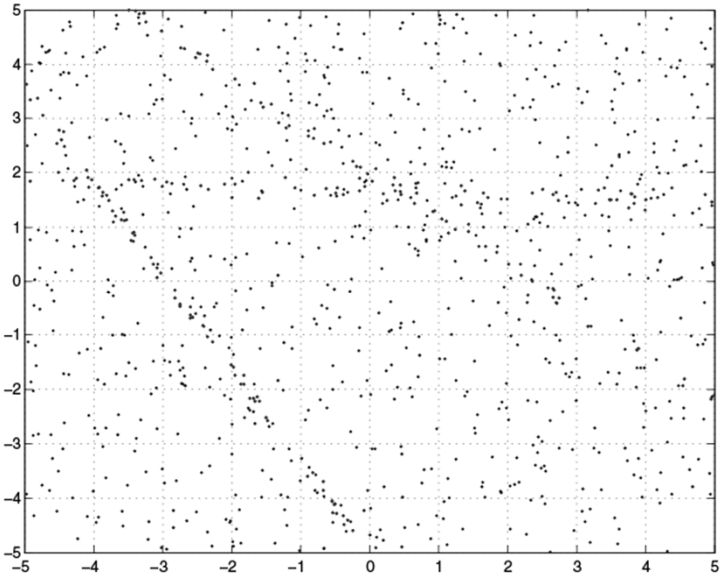
From [Bilmes, 1998]

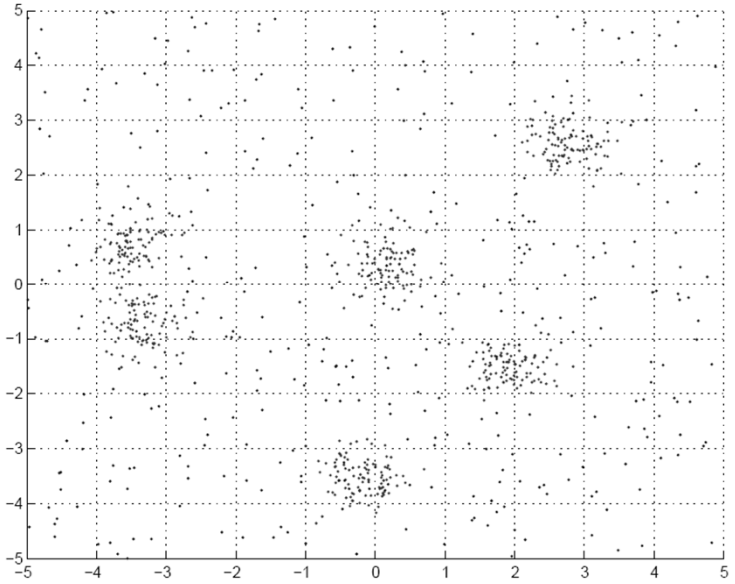
The Expectation-Maximization algorithm (1)

- When assuming the existence of missing or hidden parameters can simplify the likelihood function, which would otherwise lead to an analytically intractable optimization problem. This is the case that occupies our discussion
- **Assume a dataset containing two types of features**
  - A set of features X whose value is known. We call these the *incomplete* data
  - A set of features Z whose value is unknown. We call these the *missing* data

Applications of EM (1)

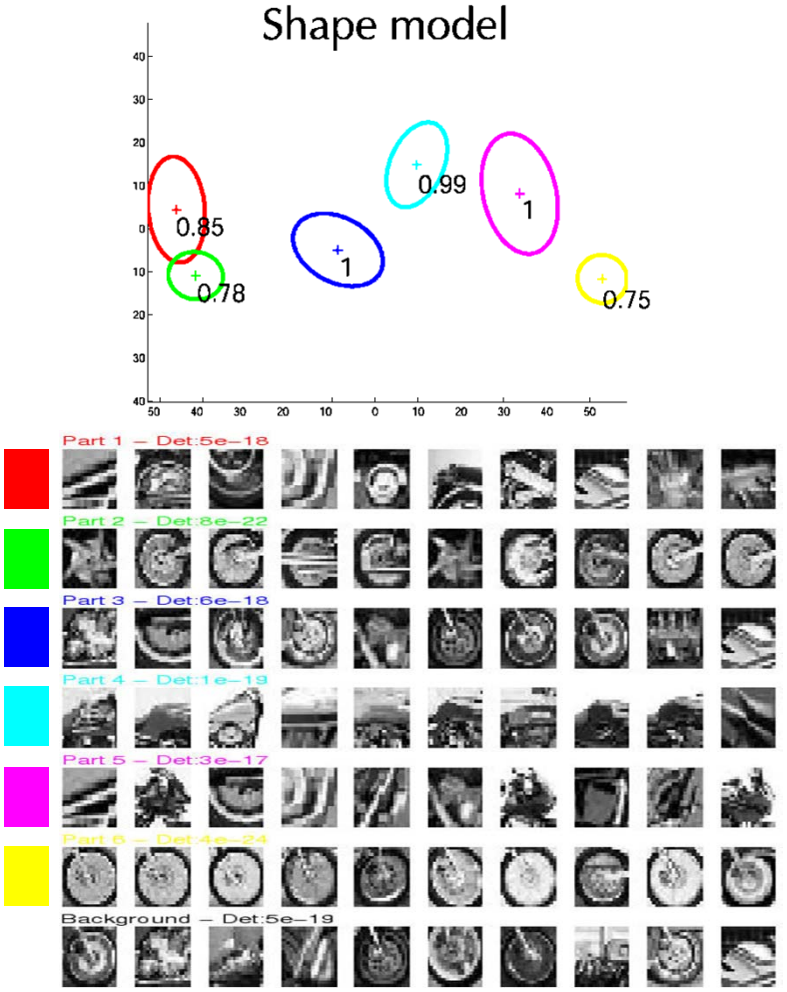
- Fitting mixture models





Applications of EM (2)

- Learning parts and structure models



Theory

Some definitions

Observed data $x = (x_1, x_2, \ldots, x_N)$ Continuous I.I.D

Latent variables $z = (z_1, z_2, \ldots, z_N)$ Discrete 1 ... C

Iteration index t

Log-likelihood [Incomplete log-likelihood (ILL)]

$$l(\theta; x) = log\, p(x|\theta) = log\, \prod_x p(x|\theta)$$
$$= \sum_x log\, \sum_z p(x, z|\theta)$$

Lower bound on log-likelihood

$$
\begin{aligned}
l(\theta; x) &= \log p(x \mid \theta) \\
&= \log \sum_z p(x, z \mid \theta) \\
&= \log \sum_z q(z \mid x) \frac{p(x, z \mid \theta)}{q(z \mid x)} \\
&\geq \sum_z q(z \mid x) \log \frac{p(x, z \mid \theta)}{q(z \mid x)} \\
&\triangleq \mathcal{L}(q, \theta), \quad \textbf{AUXILIARY FUNCTION}
\end{aligned}
$$

Use Jensen's inequality

Jensen's Inequality

Jensen's inequality: $f(\sum_j \lambda_j x_j) \geq \sum_j \lambda_j f(x_j)$

For a real continuous concave function $f()$ and $\lambda_j \geq 0,\ \sum_j \lambda_j = 1$

---

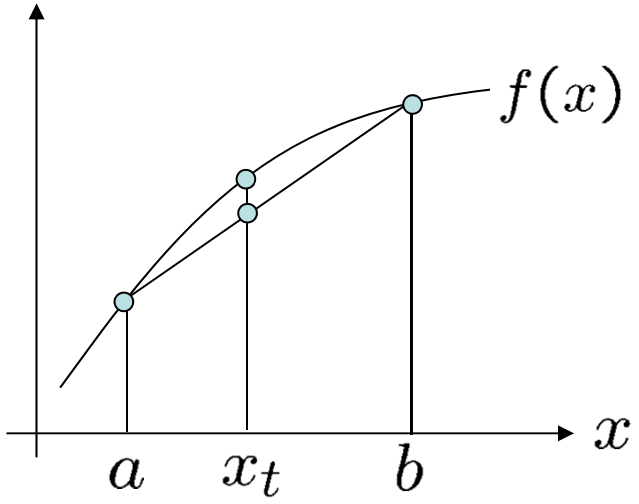
1. Definition of concavity. Consider $x_t = (1-t)a + tb$ where $0 \leq t \leq 1$

   then $f((1-t)a + tb) \geq (1-t)f(a) + tf(b)$

2. By induction:

   $f(\sum_{j=1}^{M} \lambda_j x_j) \geq \sum_{j=1}^{M} \lambda_j f(x_j)$

   for $M \geq 2$



Equality holds when all x are the same

EM is alternating ascent

Recall key result : Auxiliary function is LOWER BOUND on likelihood

$$\mathcal{L}(q, \theta) \leq l(\theta; x)$$

Alternately improve q then θ:

Turns out that $q(z|x,\theta) = p(z|x,\theta^t)$ is the best.

$$(\mathbf{E\ step}) \qquad q^{(t+1)} = \arg\max_q \mathcal{L}(q, \theta^{(t)})$$

$$(\mathbf{M\ step}) \qquad \theta^{(t+1)} = \arg\max_\theta \mathcal{L}(q^{(t+1)}, \theta).$$

Is guaranteed to improve likelihood itself….

E-step: What do we actually compute?

$$p(z|x,\theta^t)$$

nComponents x nPoints matrix (columns sum to 1):

	Point 1	Point 2				Point 6
Component 1						
Component 2						

Responsibility of component c for point i :

$$p(z_i = c|x_i,\theta^t) = \frac{p(x_i,z_i=c|\theta^t)}{\sum_c p(x_i,z_i=c|\theta^t)}$$

$$= \frac{\pi(z_i=c)\ N(x_i|\mu_c,\sigma_c)}{\sum_d \pi(z_i=d)\ N(x_i|\mu_d,\sigma_d)}$$

M-Step

Auxiliary function separates into ECLL and entropy term:

$$\begin{aligned}\mathcal{L}(q,\theta) &= \sum_z q(z \mid x) \log \frac{p(x,z \mid \theta)}{q(z \mid x)} \\ &= \sum_z q(z \mid x) \log p(x,z \mid \theta) - \sum_z q(z \mid x) \log q(z \mid x) \\ &= \langle l_c(\theta; x, z) \rangle_q - \sum_z q(z \mid x) \log q(z \mid x),\end{aligned}$$

ECLL Entropy term

Expected complete log-likelihood (ECLL)

$$\langle l_c(\theta; x, z) \rangle_q \triangleq \sum_{\boldsymbol{x}} \sum_z q(z \mid x, \theta) \log p(x, z \mid \theta)$$

$$= E_q[\log P(X, z|\theta)]$$

M-Step

Recall definition of ECLL:

$$\langle l_c(\theta; x, z) \rangle_q \triangleq \sum_x \sum_z q(z \mid x, \theta) \log p(x, z \mid \theta)$$

From previous slide: $q(z|x,\theta) = p(z|x,\theta^t)$

From E-step

$$\frac{\partial \mathcal{L}(q^{t+1},\theta)}{\partial \theta} = \frac{\partial \sum_x \sum_z p^{t+1}(z|x,\theta^t)\ log\ p(x,z|\theta)}{\partial \theta} = 0$$

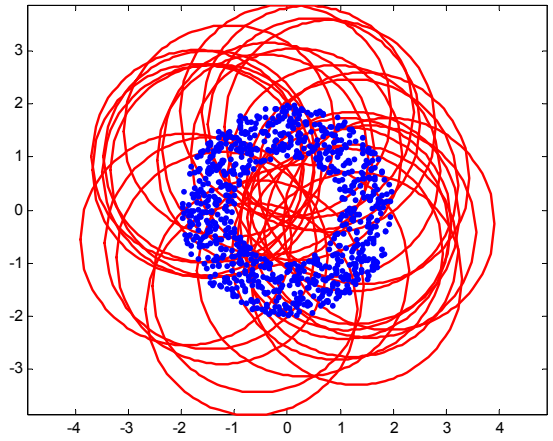
Let's see what happens for μ_c

Mixture models and the annulus problem (1)

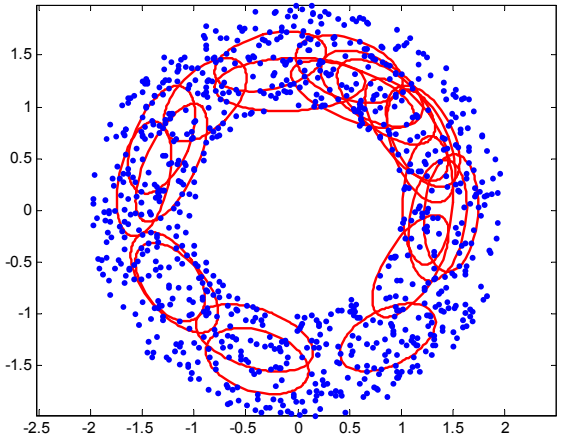
- A training set of N=900 examples was generated using a uniform pdf inside an annulus with inner and outer radii of 1 and 2 units, respectively
- A mixture model with C=30 Gaussian components was used to model the distribution of the training set

Mixture models and the annulus problem (2)

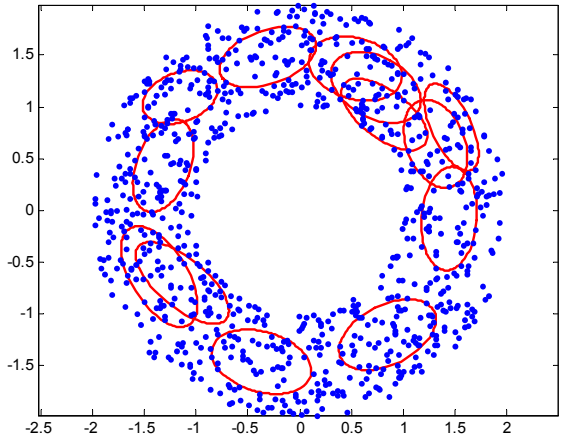
Iteration 0



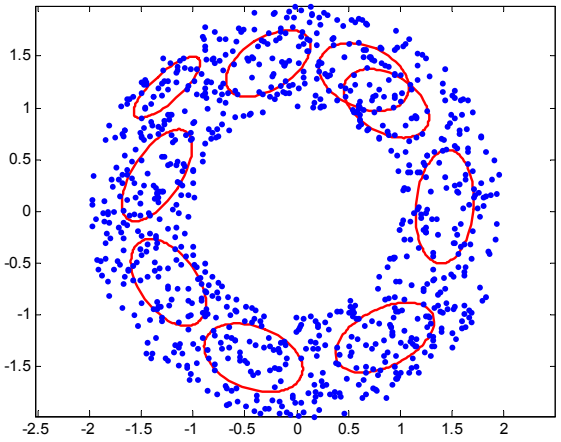
Iteration 25



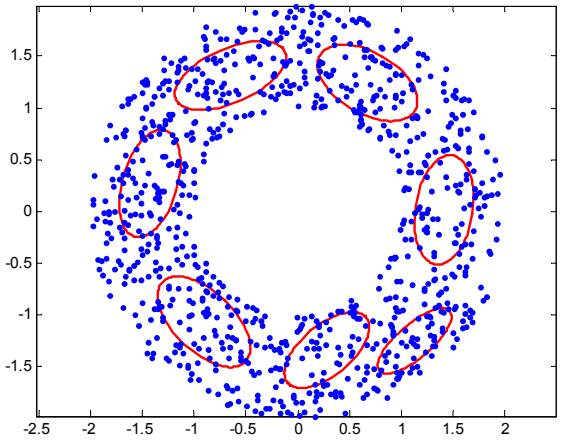
Iteration 50



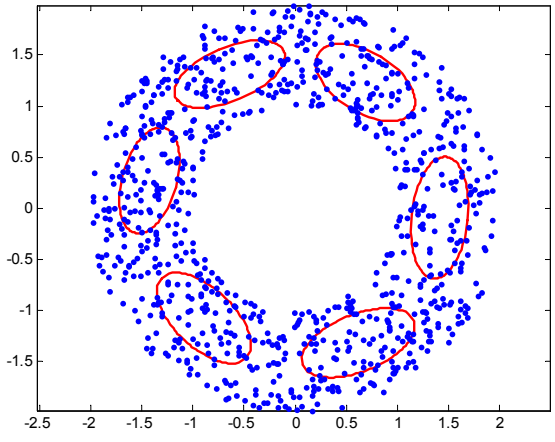
Iteration 75



Iteration 275



Iteration 300



Mixture models and the annulus problem (1)

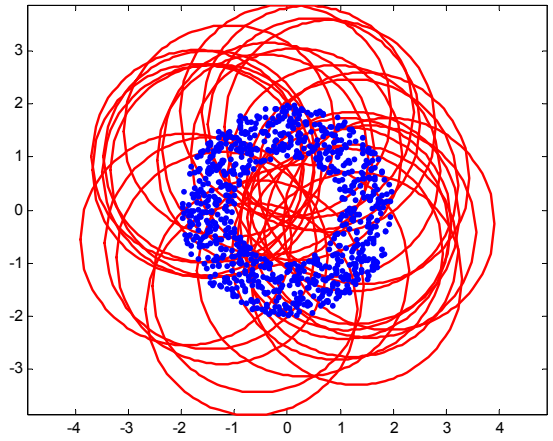
- **Training procedure**
  - The centers of the Gaussians were initialized by choosing 30 arbitrary points from the training set
  - The covariance matrices were initialized to be diagonal, with a large variance compared to that of the training data
    - To avoid singularities, at every iteration the covariance matrices computed with EM were regularized with a small multiple of the identity matrix

Mixture models and the annulus problem (1)

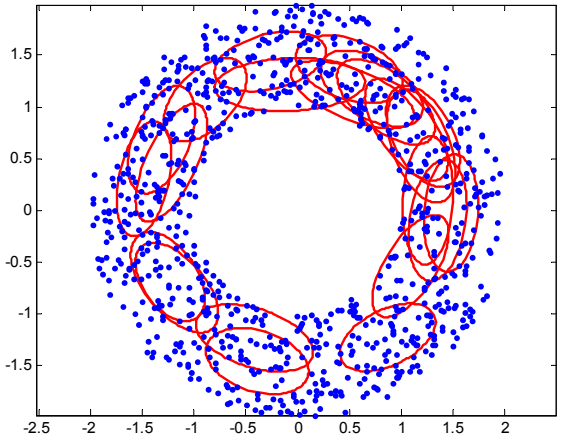
- Components whose mixing coefficients fell below a threshold were trimmed
  - This allowed the algorithm to produce a compact model with only a few of the initial C=30 Gaussian components
- **Illustrative results are provided in the next page for one particular run**

Mixture models and the annulus problem (2)

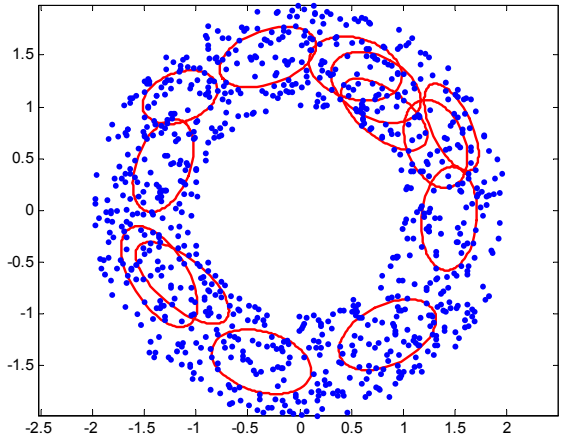
Iteration 0



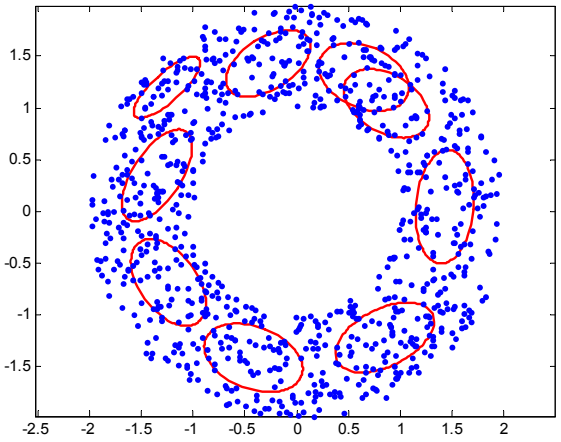
Iteration 25



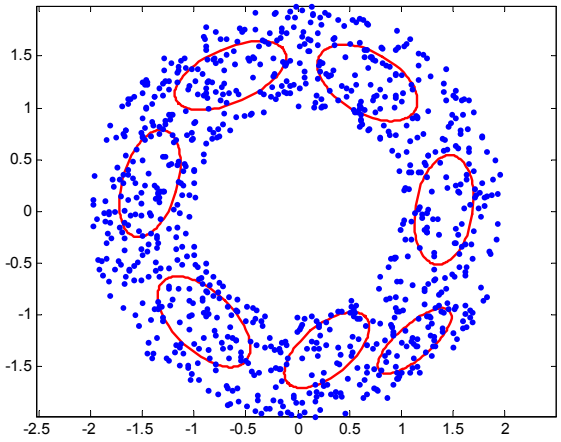
Iteration 50



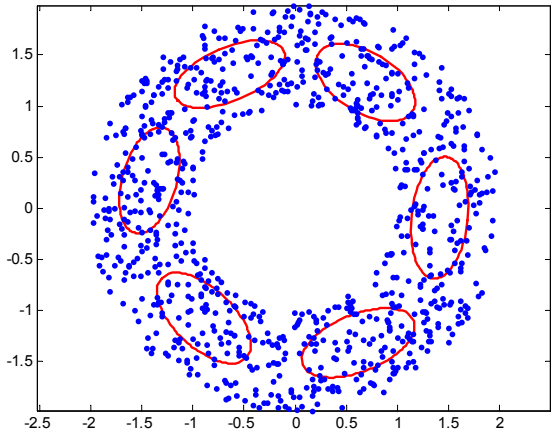
Iteration 75



Iteration 275



Iteration 300



Practical

Practical issues

Initialization

Mean of data + random offset
K-Means

Termination

Max # iterations
log-likelihood change
parameter change

Convergence

Local maxima
Annealed methods

Numerical issues

Regularize Covariance matrix to prevent blowup
Single point gives infinite likelihood

Local minima