

# Scientific Computing for Biologists

## Lecture 10: Mixture Models and Multi-dimensional Scaling

Instructor: Paul M. Magwene

06 November 2012

# Outline of Lecture

- Mixture model based clustering
- Multi-dimensional scaling (MDS)

# Clustering with Mixture Models

## Goal

Method for assigning observations to clusters and estimating parametric distributions that describe the clusters.

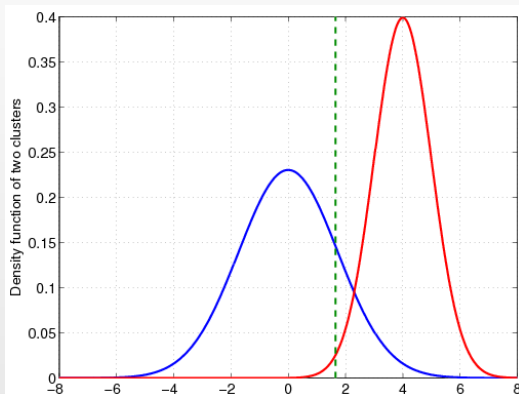
Assume that the data set represents observations drawn from a mixture of  $g$  sub-distributions (user specifies  $g$ ), and that the probability density function of the mixture is given by:

$$p_{\text{mix}} = \sum_{s=1}^g \pi_s p(\mathbf{x}; \theta_s)$$

Where the  $p(\mathbf{x}; \theta_s)$  represents the  $s$ -th ‘component density’ (sub-distributions) and the  $\theta_s$  are the component parameters. The  $\pi_s$  represent the weighting factor of the  $s$ -th component in the mixture.

## Advantages

- ▶ Well-studied statistical inference techniques available.
- ▶ Flexibility in choosing the component distributions.
- ▶ Obtain a density estimation for each cluster.
- ▶ A “soft” classification is available.



# Gaussian Mixture Models

A common starting point in mixture modeling is to assume that the components are Gaussian.

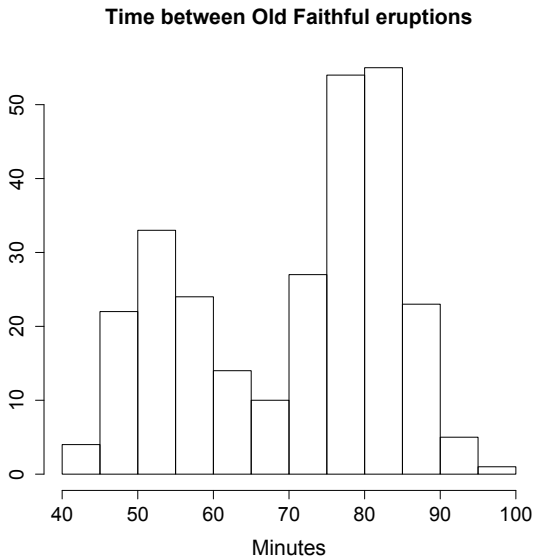
If the data are univariate, then the mixture model is given by:

$$p_{\text{mix}} = \sum_{s=1}^g \pi_s f(x|\mu_i, \sigma_i^2)$$

where the  $\mu_i$  and  $\sigma_i$  are the means and standard deviations of each component distribution and:

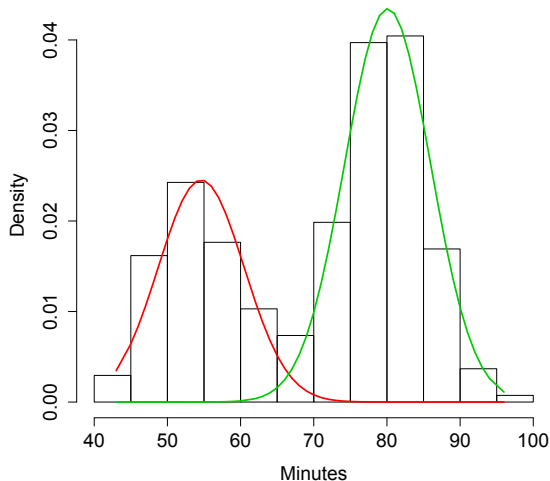
$$f(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

## Example: Waiting time between Old Faithful eruptions



## Example: Gaussian fit, Old Faithful waiting time

Time between Old Faithful eruptions



$$\pi = (0.36, 0.64)$$

$$\mu = (54.6, 80.1)$$

$$\sigma = (5.87, 5.87)$$

# Gaussian Mixture Models, Multivariate data

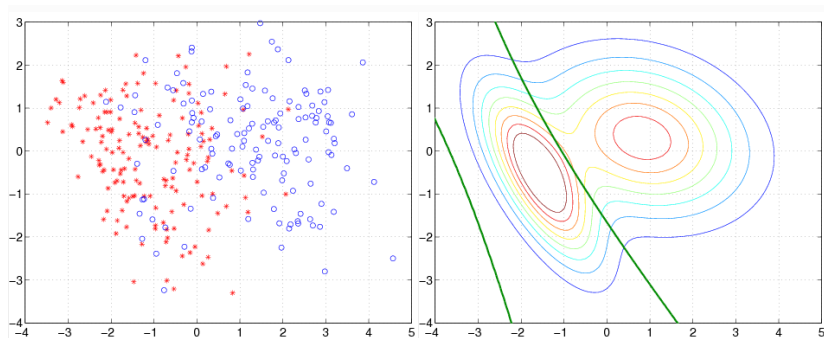
When the components are multivariate Gaussian distributions:

$$N(\mathbf{x}; \theta) \equiv (2\pi)^{-D/2} |\Sigma|^{-1/2} \exp \left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

each with a different mean vector,  $\boldsymbol{\mu}$  ( $\boldsymbol{\mu} \in \mathbb{R}^p$ ), and covariance matrix,  $\Sigma$  ( $p \times p$ ).



# Mixture Model Clustering, Example



Heart disease example: 297 samples (137 with heart disease). 13 quantitative variables (e.g. cholesterol, max heart rate, etc). Data centered and normalized. Data projected onto first two PCs. Two-component Gaussian mixture fit.

# How do we 'solve' the mixture model problem?

The mixture model problem involves optimization over multiple parameters.

The standard approach to estimating the parameters is called the "Expectation-Maximization" (EM) algorithm.

- Described by Dempster, Laird, and Rubin (1977)
- Provides a way to iterative compute a maximum likelihood estimation when the observed data are incomplete or there are 'latent' parameters.

# Overview of the EM Algorithm

- 1 Guess a set of starting parameters
- 2 Use these starting parameters to 'estimate' the complete data
- 3 Use the estimates of the complete data to update the parameters
- 4 Repeat steps 2 and 3 until convergence

## EM Algorithm for the Mixture of Gaussians

Parameters estimated at the  $p$ th iteration are marked by a superscript  $(p)$ .

1. Initialize parameters
2. E-step: Compute the posterior probabilities for all  $i = 1, \dots, n$ ,  $k = 1, \dots, K$ .

$$p_{i,k} = \frac{a_k^{(p)} \phi(x_i | \mu_k^{(p)}, \Sigma_k^{(p)})}{\sum_{k=1}^K a_k^{(p)} \phi(x_i | \mu_k^{(p)}, \Sigma_k^{(p)})}.$$

3. M-step:

$$a_k^{(p+1)} = \frac{\sum_{i=1}^n p_{i,k}}{n}, \quad \mu_k^{(p+1)} = \frac{\sum_{i=1}^n p_{i,k} x_i}{\sum_{i=1}^n p_{i,k}}$$

$$\Sigma_k^{(p+1)} = \frac{\sum_{i=1}^n p_{i,k} (x_i - \mu_k^{(p+1)})(x_i - \mu_k^{(p+1)})^t}{\sum_{i=1}^n p_{i,k}}$$

4. Repeat step 2 and 3 until converge.

# Multidimensional Scaling (MDS)

## Goal

Given dissimilarities between objects,  $d_{ij}$ , estimate a  $k$ -dimensional set of points,  $X$ , such that  $|x_i - x_j| \approx d_{ij}$ .

# Derivation of MDS

## Motivation

If we know the coordinates of  $n$  points in  $p$ -dimensional space, we can easily calculate the Euclidean distances between every pair of points. Can we reverse this process, starting with the distances and getting back the coordinates points?

Consider a data matrix  $X$  ( $n \times p$ ). Let  $Q = XX'$  be a  $n \times n$  matrix, where

$$q_{rs} = \sum_{j=1}^p x_{rj}x_{sj}$$

If  $d_{rs}^2$  is the squared Euclidean distance between points  $r$  and  $s$  then we can write this as:

$$\begin{aligned} d_{rs}^2 &= \sum_{j=1}^p (x_{rj} - x_{sj})^2 \\ &= q_{rr} + q_{ss} - 2q_{rs} \end{aligned}$$

## Derivation of MDS, cont.

With a little bit of simple algebra we can show that:

$$q_{rs} = -\frac{1}{2}(d_{rs}^2 - d_{r\cdot}^2 - d_{\cdot s}^2 + d_{\cdot\cdot}^2)$$

where a dot represent the average of values over the corresponding suffix:  $d_{r\cdot}^2$  is the average over the  $r$ th row of matrix  $D = (d_{ij}^2)$ ,  $d_{\cdot s}^2$  is the average over the  $s$ th column of  $D$ , and  $d_{\cdot\cdot}^2$  is the average of all elements of  $D$ . So, given  $D$ , the squared interpoint distances, we can regenerate  $Q$ .

Since  $Q$  is symmetric, we can use eigendecomposition to write  $Q = T\Lambda T'$  where  $\Lambda$  is a diagonal matrix of eigenvalues of  $Q$  and  $T$  is the matrix of eigenvectors. Furthermore we can write  $Q = T\Lambda T' = T\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}T' = XX'$  where  $X = T\Lambda^{\frac{1}{2}}$ .

Thus we've found how to get  $X$  from the squared distances.

See Krzanowski, W. J. (2000) Principles of multivariate analysis, for full details.

# Algorithm for MDS

Given an  $n \times n$  matrix of dissimilarities,  $D$ , with elements  $d_{ij}$ :

- 1 Form matrix,  $E$ , where  $e_{ij} = -\frac{1}{2}d_{ij}^2$
- 2 Subtract from each element of  $E$  the means of the row and column in which it is located and the mean of all elements of  $E$ ; call the resulting matrix  $F$
- 3 Calculate the eigenvalues ( $\lambda_i$ ) and eigenvectors  $v_i$  of  $F$ , sorted in decreasing order. Eigenvectors should be normalized (i.e.  $v_i \cdot v_i = 1$ ).
- 4 The coordinates of the  $n$  point on the  $j$ -th axis are given  $\sqrt{\lambda_j}v_j$



# Potential MDS Complications

If the  $d_{ij}$  are metric (i.e.  $d_{ij} \leq d_{ik} + d_{kj}$ ) then  $F$  is always positive semidefinite (psd; i.e. eigenvalues  $\geq 0$ ).

If  $F$  is not psd then how do you handle negative eigenvalues?

- Most common approach is only to consider positive eigenvalues
- This is OK if negative eigenvalues have small magnitude
- If negative eigenvalues are large then approximation tends to be poor

## Multidimensional Scaling: Keep in mind...

- The configuration produced by any MDS method is indeterminate with respect to translation, rotation, and reflection.

# Relationship between metric MDS and PCA

If the  $d_{ij}$  are Euclidean distances from a data matrix,  $X$ , then metric MDS of  $D$  yields the PC scores obtained by PCA of  $X$ .

## Interpretation

PCA and MDS are dual methods:

- One operates on variable space (PCA)
- The other operates on subject space (MDS)

## Other Metric MDS Approaches

- Classical MDS minimizes:

$$\sum_i \sum_j (\delta_{ij}^2 - d_{ij}^2)$$

where  $\delta_{ij}$  is the distance between observations  $i$  and  $j$  in the MDS approximation.

- Alternates approaches try to minimize other measures of discrepancy. For example, "Sammon MDS" minimizes:

$$\sum_i \sum_j (\delta_{ij} - d_{ij})^2$$

# Non-Metric MDS

Non-metric MDS approaches try to preserve only the rank order of the distances.

If

$$d_{i1,j1} < d_{i2,j2} < \dots < d_{im,jm}$$

then

$$\delta_{i1,j1} < \delta_{i2,j2} < \dots < \delta_{im,jm}$$

Shepard-Kruskal solution:

- Find  $\hat{d}_{ij}$  that minimizes:

$$\text{STRESS} = \sqrt{\left\{ \frac{\sum \sum_{i < j} (d_{ij} - \hat{d}_{ij})^2}{\sum \sum d_{ij}^2} \right\}}$$

# MDS Example: Road Distances

Input D: road distances between U.S. cities



Figure 1