Scientific Computing for Biologists

Lecture 10: Mixture Models and Multi-dimensional Scaling

Instructor: Paul M. Magwene

05 November 2013

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.

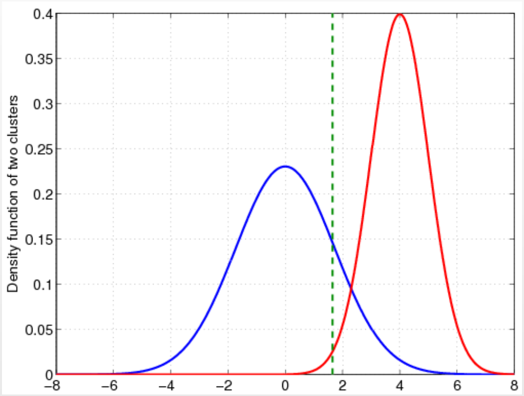
Assume that the data set represents observations drawn from a mixture of g sub-distributions (user specifies g), and that the probability density function of the mixture is given by:

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

Where the $p(\mathrm{x}; \theta_s)$ represents the s-th ‘component density’ (sub-distributions) and the θ_s are the component parameters. The π_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.

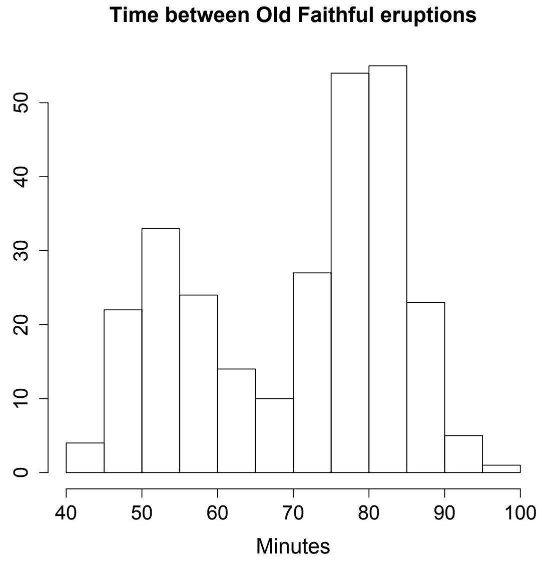
If the data are univariate, then the mixture model is given by:

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

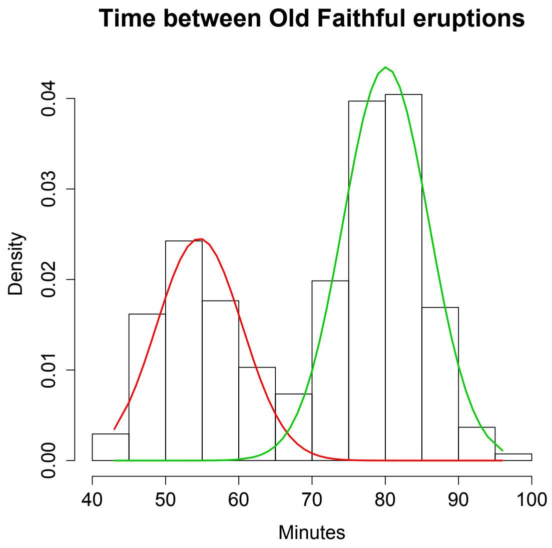
where the μ_i and σ_i are the means and standard deviations of each component distribution and:

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

Example: Waiting time between Old Faithful eruptions



Example: Gaussian fit, Old Faithful waiting time



$$\pi = (0.36, 0.64)$$
$$\mu = (54.6, 80.1)$$
$$\sigma = (5.87, 5.87)$$

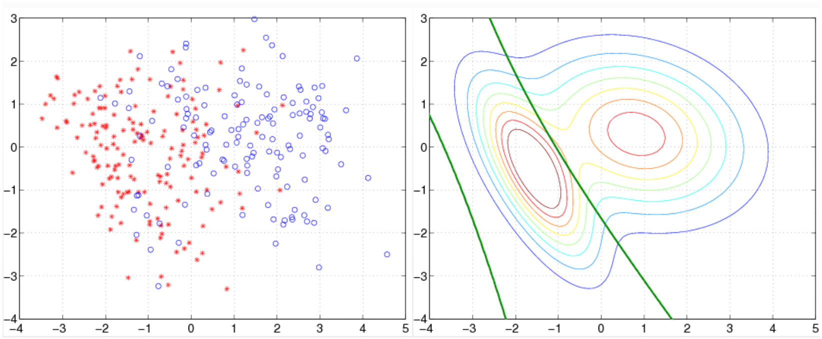
Gaussian Mixture Models, Multivariate data

When the components are multivariate Gaussian distributions:

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

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

Mixture Model Clustering, Example



Heart disease example: 297 samples (137 with heart disease). 13 quantitative varibles (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 pth iteration are marked by a superscript (p).

1. Initialize parameters
2. E-step: Compute the posterior probabilities for all $i = 1, ..., n$, $k = 1, ..., K$.
$$p_{i,k} = \frac{a_k^{(p)} \phi(x_i \mid \mu_k^{(p)}, \Sigma_k^{(p)})}{\sum_{k=1}^{K} a_k^{(p)} \phi(x_i \mid \mu_k^{(p)}, \Sigma_k^{(p)})} .$$
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

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 $\mathrm{Q} = \mathrm{XX}'$ 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:

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

where a dot represent the average of values over the corresponding suffix: $d_{r.}^2$ is the average over the rth row of matrix $\mathsf{D} = (d_{ij}^2)$, $d_{.s}^2$ is the average over the sth column of D, and $d_{..}^2$ 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 $\mathsf{Q} = \mathsf{T}\Lambda\mathsf{T}'$ where Λ is a diagonal matrix of eigenvalues of Q and T is the matrix of eigenvectors. Furthermore we can write $\mathsf{Q} = \mathsf{T}\Lambda\mathsf{T}' = \mathsf{T}\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}\mathsf{T}' = \mathsf{X}\mathsf{X}'$ where $\mathsf{X} = \mathsf{T}\Lambda^{\frac{1}{2}}$.

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 (λ_i) and eigenvectors v_i of F, sorted in decreasing order. Eigenvectors should be normalized (i.e. $\mathrm{v}_i \cdot \mathrm{v}_i = 1$).
4. The coordinates of the n point on the j-th axis are given $\sqrt{\lambda_j} \mathrm{v}_j$

Potential MDS Complications

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

If F is not psd than 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 than 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 δ_{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

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

then

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

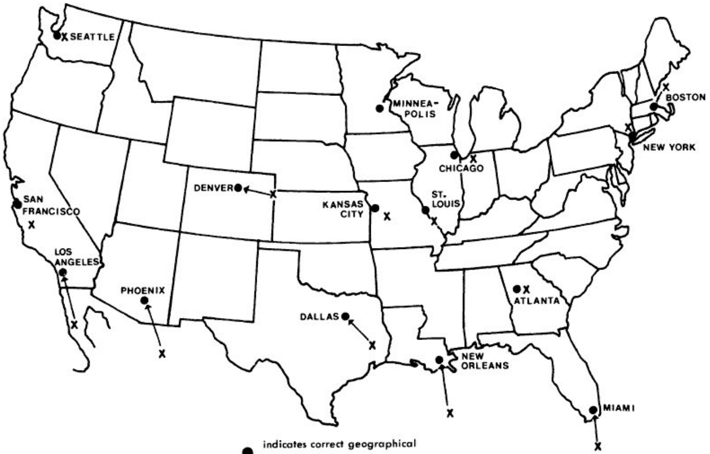
Shepard-Kruskal solution:

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

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

MDS Example: Road Distances

Input D: road distances between U.S. cities



● indicates correct geographical location of city

× indicates city position by two dimensional multidimensional scaling solution

Figure 1

Minimum Spanning Tree

Minimum Spanning Tree

Goal

Construct a tree that connects all points in the data set and whose total length is minimized.

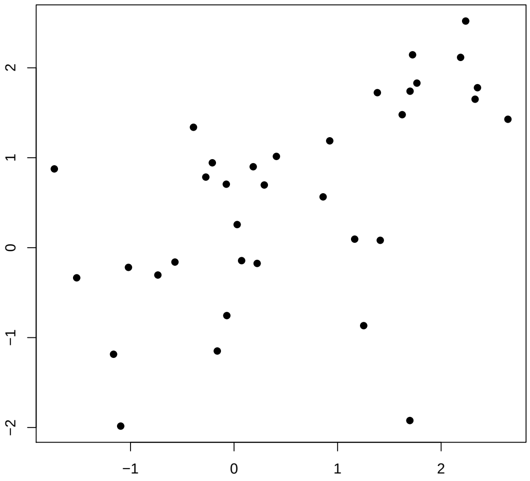
Statistical applications

- highlights close neighbors in a data set
- useful check for distortions produced by projection techniques
- tests of normality

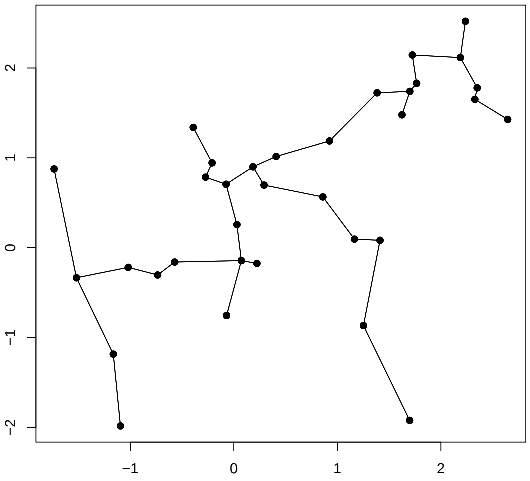
Other applications

- urban planning/engineering
- circuit design

Example Data Set



Minimum Spanning Tree: Example



Relationship between MST and Single Linkage Clustering

- Cut a single linkage dendrogram at height, $\delta \dashrightarrow$ clusters
- Remove all edges in the MST with length $\geq \delta \dashrightarrow$ subgraphs corresponding to the same clusters

A Generic MST Algorithm

Input: dissimilarity matrix, D, between each object (point) of interest

1. Create a graph, G, where $V = \{v_1, \ldots, v_n\}$ and $E = \{\}$ (E initially empty)
2. Find the smallest dissimilarity, d_{ij} where (i,j) is not in E.
3. Add (i,j) to E if (i,j) does not create a cycle
4. Repeat from step 2 until every vertex is included in at least one edge

Not particularly efficient algorithm, but simple. More efficient algorithms for finding MSTs include Kruskal's Algorithm and Prim's algorithm.

Applications of the MST

MST tends to highlight close neighbors; can be used to look for distortions associated with projections to lower dimensional spaces.

Using the MST to look for Projection Distortion

- Calculate the MST based on dissimilarity in a high-dimensional space
- Draw the MST edges among points in the projection space (e.g. MDS or PCA)
- MST edges that cross highlight geometric relationships among points that are not well represented by the projection