Scientific Computing for Biologists

Lecture 10: K-mean Clustering, Mixture Models and Multi-dimensional Scaling

Instructor: Paul M. Magwene

11 November 2014

Outline of Lecture

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

K-means Clustering

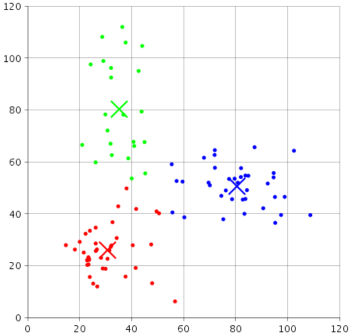
K-mean Clustering

General idea

Assign the n data points (or p variables) to one of K clusters to as to optimize some criterion of interest.

- The most common criterion to minimize is the sum-of-squares from the group centroids.

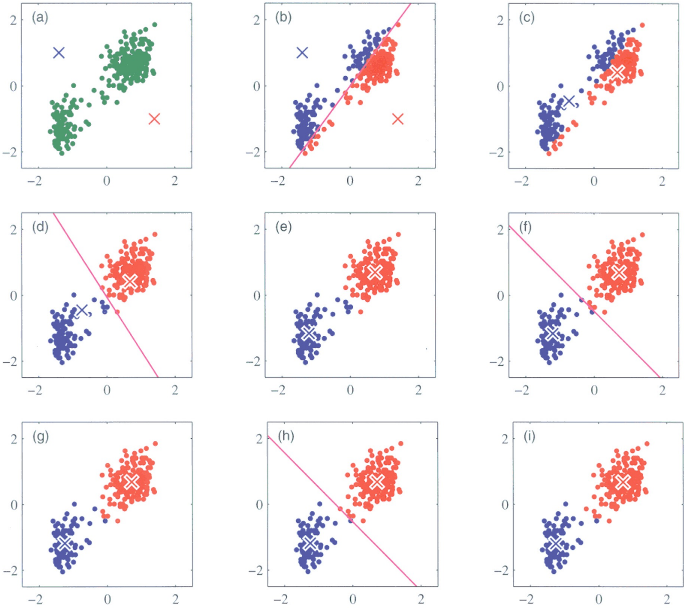
$$V = \sum_{i=1}^{k} \sum_{j \in g_i} |x_j - \mu_i|^2$$



Simple algorithm for K-means clustering

1. Decide on k, the number of groups
2. Randomly pick k of the objects to act as the initial centers
3. Assign each object to the group whose center it is closest to
4. Recalculate the k centers as the centroids of the objects assigned to them
5. Repeat from step 3 until centroids no longer move (convergence)

Illustration of K-means algorithm



Things to note re:K-means clustering

- The algorithm described above does not necessarily find the global optimum

- The algorithm is sensitive to choice of initial cluster center; k-means is often run multiple-time with different initial centers to insure inferred clusters are robust.

Mixture Modeling

Clustering with Mixture Models

Goal

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

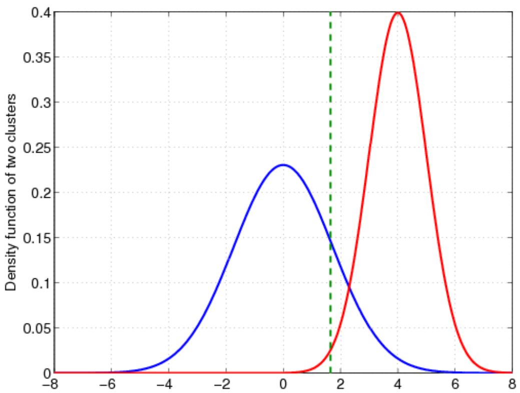
Assume that the data set represents observations drawn from a mixture of g sub-distributions (user specifies g), and that the probability density function of the mixture is given by:

$$p_{\text{mix}} = \sum_{s=1}^{g} \pi_s p(\boldsymbol{x}; \boldsymbol{\theta}_s)$$

Where the $p(\boldsymbol{x}; \boldsymbol{\theta}_s)$ represents the s-th ‘component density’ (sub-distributions) and the $\boldsymbol{\theta}_s$ 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.

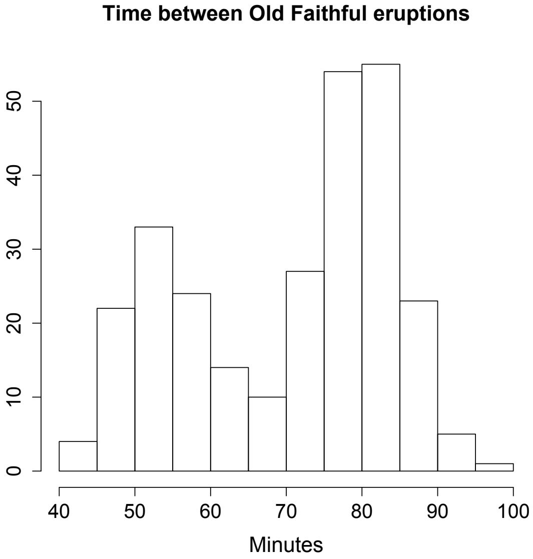
If the data are univariate, then the mixture model is given by:

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

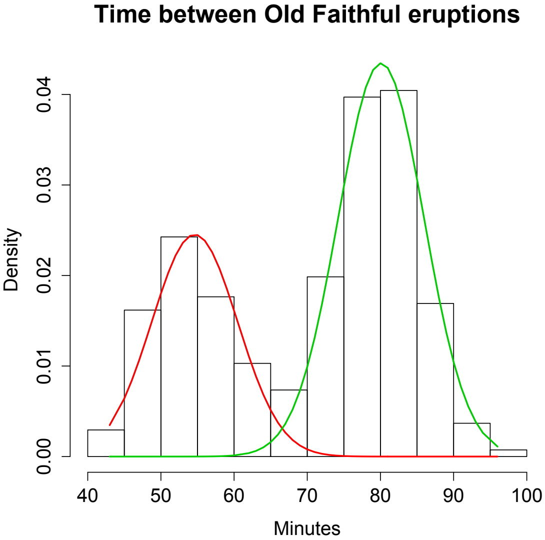
where the μ_i and σ_i are the means and standard deviations of each component distribution and:

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

Example: Waiting time between Old Faithful eruptions



Example: Gaussian fit, Old Faithful waiting time



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

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

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

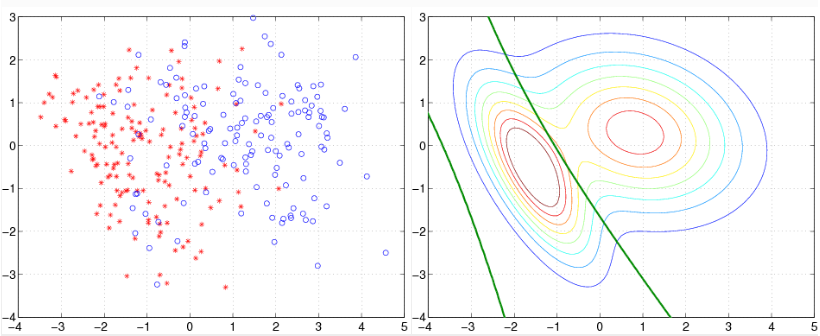
Gaussian Mixture Models, Multivariate data

When the components are multivariate Gaussian distributions:

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

each with a different mean vector, $\boldsymbol{\mu}$ ($\boldsymbol{\mu} \in \mathbb{R}^p$), 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

Multidimensional Scaling

Multidimensional Scaling (MDS)

Goal

Given dissimilarities between objects, d_{ij}, estimate a k-dimensional set of points, $\boldsymbol{X}$, 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 $\boldsymbol{X}$ $(n \times p)$. Let $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{X}'$ 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:

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

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

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

Thus we've found how to get $\boldsymbol{X}$ 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, $\boldsymbol{D}$, with elements d_{ij}:

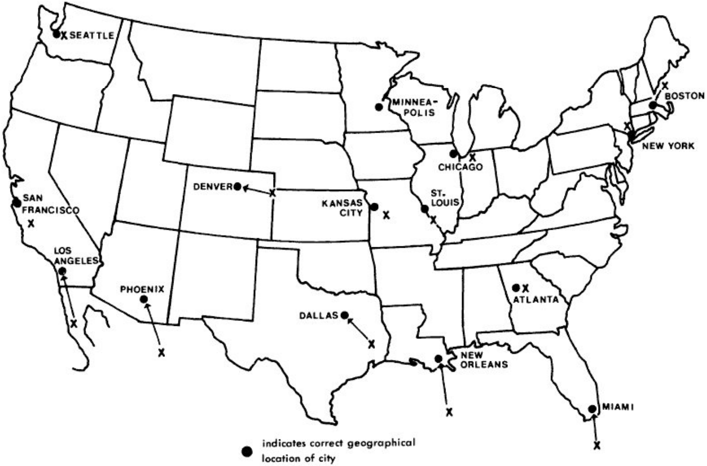
1. Form matrix, $\boldsymbol{E}$, where $e_{ij} = -\frac{1}{2} d_{ij}^2$
2. Subtract from each element of $\boldsymbol{E}$ the means of the row and column in which it is located and the mean of all elements of $\boldsymbol{E}$; call the resulting matrix $\boldsymbol{F}$
3. Calculate the eigenvalues (λ_i)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

MDS Example: Road Distances between U.S. Cities

```
     BOS  CHI   DC  DEN   LA  MIA   NY  SEA   SF
BOS    0  963  429 1949 2979 1504  206 2976 3095
CHI  963    0  671  996 2054 1329  802 2013 2142
DC   429  671    0 1616 2631 1075  233 2684 2799
DEN 1949  996 1616    0 1059 2037 1771 1307 1235
LA  2979 2054 2631 1059    0 2687 2786 1131  379
MIA 1504 1329 1075 2037 2687    0 1308 3273 3053
NY   206  802  233 1771 2786 1308    0 2815 2934
SEA 2976 2013 2684 1307 1131 3273 2815    0  808
SF  3095 2142 2799 1235  379 3053 2934  808    0
```

MDS Example: Road Distances

Input $\boldsymbol{D}$: road distances between U.S. cities



● indicates correct geographical location of city

× indicates city position by two dimensional multidimensional scaling solution

Figure 1

Multidimensional Scaling: Keep in mind...

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

Potential MDS Complications

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

If $\boldsymbol{F}$ 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

Relationship between metric MDS and PCA

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

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:

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