STAT 730 Chapter 13: Cluster Analysis

Timothy Hanson

Department of Statistics, University of South Carolina

Stat 730: Multivariate Data Analysis

Basic idea

Multidimensional scaling produces a (typically 2-dimensional) map that best preserves distances among n objects or variables originally in $\mathbb{R}^p$. The human mind naturally looks for groups or clusters of points; this is also a form of data reduction. We often assume objects are in some way exchangeable within a group, and then, after clustering, look for what makes groups "different."

There are many ways to cluster data, the most used being (1) k-means, (2) model-based methods, and (3) hierarchical clustering. We will discuss each in turn.

Start with data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$. Want to allocate data into k homogeneous groups or clusters. Also want to pick the "best" number of groups k.

End result is sets of indices $C_1, \ldots, C_k$ s.t. $\cup_{j=1}^{k} C_j = \{1, \ldots, n\}$.

k-means

k is picked ahead of time. Let $z_i = j$ if $\mathbf{x}_i$ has mean $\boldsymbol{\mu}_j$ and $\mathbf{z}' = (z_1, \ldots, z_n)$. Let $\boldsymbol{\mu}' = (\boldsymbol{\mu}'_1, \ldots, \boldsymbol{\mu}'_k)$, k-means minimizes

$$Q(\boldsymbol{\mu}, \mathbf{z}) = \sum_{i=1}^{n} ||\mathbf{x}_i - \boldsymbol{\mu}_{z_i}||^2,$$

according to the following algorithm. Initialize $\hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_k$, define $n_j = \sum_{i=1}^{n} I\{z_i = j\}$ to be the number of $\mathbf{x}_1, \ldots \mathbf{x}_n$ that come from mean $\boldsymbol{\mu}_j$. Note that $n_1 + \cdots + n_k = n$.

1. $z_i = \text{argmin}_{j=1,\ldots,k} ||\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j||^2$.
2. $\hat{\boldsymbol{\mu}}_j = \frac{1}{n_j} \sum_{i:z_i=j} \mathbf{x}_i$.

Repeat until convergence. The algorithm converges to a local minimum. This is a simple expectation-maximization (EM) algorithm (in disguise) for the model $\mathbf{x}_i \sim \sum_{j=1}^{k} w_j N_p(\boldsymbol{\mu}_j, \sigma^2 \mathcal{I})$ where $w_j = \frac{1}{k}$. Here, the augmented data are the z_i.

k-means in R

```
b=read.table("http://www.stat.sc.edu/~hansont/stat730/beverages.txt",
 header=T,row.names=1)
b
b=scale(b) # data from http://nutritiondata.self.com/

# k-means, adapted from http://www.statmethods.net/advstats/cluster.html
wss=(nrow(b)-1)*sum(apply(b,2,var))
for (i in 2:10) wss[i]=sum(kmeans(b,centers=i)$withinss)
plot(1:10,wss,type="b",xlab="Number of Clusters",
 ylab="Within groups sum of squares")
f=kmeans(b,4) # look for elbow as in scree plot: k=4 or k=5
f$cluster

# a plot
library(cluster)
?clusplot.default # uses PCA (data matrix) or MDS (D matrix)
clusplot(b,f$cluster,color=TRUE,shade=TRUE,labels=2,lines=0)
```

Model-based clustering

We can generalize the implied model under k-means to

$$\mathbf{x}_i \sim \sum_{j=1}^{k} w_j N_p(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j).$$

Your book considers $\boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$ and distinct $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k$. This is what is termed a "finite mixture model." Flexibility increases from common covariance $\sigma^2 \mathcal{I}_p$ (k-means) and common $w_1 = \cdots = w_k$, to common $\boldsymbol{\Sigma}$ but different $\{w_j\}$, to distinct $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k$ and different $\{w_j\}$.

The finite mixture provides a nonparametric model for the population density $f(\mathbf{x})$ of the $\mathbf{x}_1, \ldots, \mathbf{x}_n$, and so is useful outside of clustering as well.

E.M. algorithm

A refined EM algorithm is

1. E-step:
$$\hat{w}_{ij} = \frac{\hat{w}_j \phi_p(\mathbf{x}_i; \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)}{\sum_{s=1}^{k} \hat{w}_s \phi_p(\mathbf{x}_i; \hat{\boldsymbol{\mu}}_s, \hat{\boldsymbol{\Sigma}}_s)}.$$
2. For distinct $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k$ the M-step is
$$\hat{w}_j = \frac{1}{n} \sum_{i=1}^{n} \hat{w}_{ij}, \quad \hat{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{n} \hat{w}_{ij} \mathbf{x}_i}{\sum_{i=1}^{n} \hat{w}_{ij}}, \quad \hat{\boldsymbol{\Sigma}}_j = \frac{\sum_{i=1}^{n} \hat{w}_{ij} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j)'}{\sum_{i=1}^{n} \hat{w}_{ij}}.$$

As before, this is iterated until convergence. *There are multiple modes!* One needs to consider several dispersed starting values to be reasonably confident that the solution is a MLE. Note also that there are multiple MLEs without further constraints on the model.

The fitting of such models can be carried out using the `mclust` package in R. The choice of k is often made using either AIC or BIC; there are also refined versions of these especially for mixture models. Another graphical option is the use of silhouettes; see Marden section 12.1.2.

Finite mixture of normals in R

```
# model-based
library(mclust)
pca=prcomp(b,scale.=T)
f=Mclust(pca$x[,1:3],G=1:6)
plot(f,pca$x[,1:3]) # Mclust automatically picks best
# plot results, best is k=4, ellipsoidal, equal volume and shape
?mclustModelNames
summary(f,parameters=T)
f$classification

clusplot(b,f$classification,color=TRUE,shade=TRUE,labels=2,lines=0)
```

Bayesian approach

The Bayesian approach to clustering has been immensely successful over the last 20 years. The mixture model is written hierarchically

$$\mathbf{x}_i|\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma} \overset{ind.}{\sim} N_p(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}), \ i = 1, \ldots, n,$$

$$P(z_i = j|\mathbf{w}) = w_j, \ j = 1, \ldots, k,$$

$$\mathbf{w} \sim \text{Dirichlet}(\alpha \mathbf{1}_k),$$

$$(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \overset{iid}{\sim} N(\mathbf{m}, \mathbf{M}) \times \text{Wish}^{-1}(\mathbf{S}_0, d_0).$$

Updating proceeds through Gibbs sampling.

Gibbs sampling

Let $\mathbf{n}' = (n_1, \ldots, n_k)$ and $\bar{\mathbf{x}}_j = \frac{1}{n_j} \sum_{i:z_i=j} \mathbf{x}_i$.

$$P(z_i = j | \text{else}) \propto w_j \phi_p(\mathbf{x}_i | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \; j = 1, \ldots, k,$$

$$\boldsymbol{\mu}_j | \text{else} \sim N_p(\mathbf{V}_j[\mathbf{M}^{-1}\mathbf{m} + n_j \boldsymbol{\Sigma}^{-1} \bar{\mathbf{x}}_j], \mathbf{V}_j), \; \mathbf{V}_j = [\mathbf{M}^{-1} + n_j \boldsymbol{\Sigma}_j^{-1}]^{-1},$$

$$\boldsymbol{\Sigma}_j | \text{else} \sim \text{Wish}^{-1} \left(\left[\mathbf{S}_0^{-1} + \sum_{i:z_i=j} (\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)' \right]^{-1}, d_0 + n_j \right),$$

$$\mathbf{w} | \text{else} \sim \text{Dirichlet}(\alpha \mathbf{1}_k + \mathbf{n}).$$

The Gibbs sampler samples each full conditional distribution in turn. The iterates form a Monte Carlo approxmation to the posterior $[\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{w} | \mathbf{x}_1, \ldots, \mathbf{x}_n]$. An excellent package that implements this Gibbs sampler for censored, multivariate data is `mixAK`.

Nonparametric clustering

The Dirichlet process mixture considers an infinite number of clusters. The mixture models takes $k = \infty$ and places a different prior on the weights

$$v_j \overset{iid}{\sim} \text{beta}(1, \alpha),$$

$$w_j = v_j \prod_{s=1}^{j-1} (1 - v_s).$$

Called a "stick-breaking" prior; I'll show why on the board. You can show that $\sum_{j=1}^{\infty} w_j = 1$. This is the basis of hundreds of papers in Bayesian nonparametrics. Can fit such models in `DPpackage` for R.

Covariate-dependent mixtures

The Dirichlet process mixture model, as well as finite mixture models can accommodate covariates. Say coupled with each $\mathbf{x}_i$ is a vector of covariates $\mathbf{s}_i$. A particular mixture of experts model is written

$$\mathbf{x}_i | \mathbf{z}, \mathbf{B}, \mathbf{\Sigma} \overset{ind.}{\sim} N_p(\mathbf{B}_{z_i}\mathbf{s}_i, \mathbf{\Sigma}_{z_i}).$$

Called the linear dependent Dirichlet process. If additionally (or instead) the weights depend on covariates, say

$$v_{ij} = \Phi(\boldsymbol{\beta}_j'\mathbf{s}_i), \;\; w_{ij} = v_{ij}\prod_{s=1}^{j-1}(1 - v_{is}), \;\; P(z_i = j) = w_{ij},$$

then we have a probit stick-breaking process.

Finite-versions of these, $k < \infty$, easier to interpret. Many variations on this theme. See De Iorio, Müller, Dunson, Viele, Jordan, etc. Bayesian versions *much* easier to fit than frequentist. This is true for any latent-data model, e.g. generalized linear mixed models.

Hierarchical methods

- Hierarchical methods start with a dissimilarity matrix on n objects; every pair of objects has a distance d_{ij}.
- Number of ways to partition n objects into k groups is Stirling number of the 2nd kind,

$$\left\{ \begin{array}{c} n \\ k \end{array} \right\} = \frac{1}{k!} \sum_{j=0}^{k} (-1)^j \left(\begin{array}{c} k \\ j \end{array} \right) (k-j)^n \approx \frac{k^n}{k!},$$

  where the approximation is for fixed k. For example $\left\{ \begin{array}{c} 10 \\ 5 \end{array} \right\} = 42525.$
- The total number of partitions of n objects is the Bell number $\sum_{k=1}^{n} \left\{ \begin{array}{c} n \\ k \end{array} \right\}$, which is much bigger.
- Hierarchical methods take one pass through the $m = \frac{1}{2} n(n-1)$ distances trying to form a "reasonable" set of groups.

Single and complete linkage

Pick a threshhold $d_0 > 0$. Start with n clusters: $C_1, \ldots, C_n$ each has one index, $C_i = \{i\}$. At the kth iteration there are $n - k$ clusters $C_1, \ldots, C_{n-k}$ s.t. $C_1 \cup \cdots \cup C_{n-k} = \{1, \ldots, n\}$.

(a) Let $\mathbf{D}_k = [h_{ij}] \in \mathbb{R}^{(n-k)\times(n-k)}$ be the inter-cluster distance, defined on next slide.

(b) Let $h_{rs} = \min\{h_{ij}\}$. These is the distance between the two "closest" clusters. If $h_{rs} > d_0$ then stop.

(d) Merge C_r and C_s into a combined cluster $C_r \cup C_s$, leave the others alone, and renumber the clusters $C_1, \ldots, C_{n-k-1}$. Repeat.

Note that $d_0 = \max\{d_{rs}\}$ yields one cluster with all n objects. This approach is called agglomerative: starts with n clusters and ends with 1. Can stop the process at any point to yield desired number of clusters.

There are four commonly-used inter-cluster distances.

Inter-cluster distance measures

- $h_{ij} = \min\{d_{rs} : r \in C_i, s \in C_j\}$ produces "nearest neighbor" clustering. Only one pair needs to be less than d_0 to combine, hence this is also called "single linkage" clustering. Can produce meandering, chain-looking clusters.
- $h_{ij} = \max\{d_{rs} : r \in C_i, s \in C_j\}$ produces "farthest neighbor" clustering. All pairs among two clusters C_i and C_j must be less than d_0 to combine, so also termed "complete linkage." Produces compact clusters with no chaining effect. Clusters tend to have the same diameter, so can break up large clusters.
- Instead of the max or min, one can also use the average distance between two clusters; intermediate between single and complete linkage: $h_{ij} = \frac{1}{m_i m_j} \sum_{r \in C_i, s \in C_j} d_{rs}$.
- Ward's measure is $h_{ij} = \frac{m_i m_j}{m_i + m_j} || \frac{1}{m_i} \sum_{r \in C_i} \mathbf{x}_r - \frac{1}{m_j} \sum_{s \in C_j} \mathbf{x}_s ||^2$. Produces compact, spherical clusters; often a good default choice & used to initialize k-means.

Dendrogram

Tree diagram where object indices are along x-axis, and distances along y-axis. Shows the order (and distance) in which objects are joined into clusters.

A complete dendrogram extends the y-axis to $d_{r_m s_m}$, the largest distance. Varying numbers of clusters are obtained by simply slicing the dendrogram at any d_0 along the y-axis and reading off the separated clusters.

The dendrogram can be used to make a new distance matrix, see top p. 374 in MKB.

Hierarchical methods in R

```
# hierarchical methods
d=dist(b,method="euclidean") # distance matrix
par(mfrow=c(2,2))
f=hclust(d,method="single")
plot(f,sub="",xlab="Beverages",main="Single") # display dendogram
f=hclust(d,method="complete")
plot(f,sub="",xlab="Beverages",main="Complete")
f=hclust(d,method="average")
plot(f,sub="",xlab="Beverages",main="Average")
f=hclust(d,method="ward")
plot(f,sub="",xlab="Beverages",main="Ward")
groups=cutree(f,k=4) # cut tree into k clusters
groups
par(mfrow=c(1,1))
plot(f,sub="",xlab="Beverages",main="Ward")
rect.hclust(f,k=4,border="red")
```

Similarity/dissimilarity measures

- MKB Section 13.4 pp. 375–384 discusses various distance and similarity measures at length. Also see J & W Chapter 12.
- For continuous $\mathbf{x}_i$, Minkowski metric gives distances $d_{rs} = \left\{ \sum_{j=1}^{p} w_k |x_{rj} - x_{sj}|^{\lambda} \right\}^{1/\lambda}$ includes Euclidean $\lambda = 2$ and Manhattan (city-block) $\lambda = 1$. Here $w_j = 1$ for raw, $w_j = 1/s_j$ for standardized, and $w_j = 1/R_j$ standardized by range.
- For mixed data, Gower proposes the similarity $s_{rs} = 1 - \frac{1}{p} \sum_{j=1}^{p} w_j |x_{rj} - x_{sj}|$ where $w_j = 1$ if j is qualitative and $w_j = 1/R_j$ if j is quantitative. Podani (1999, *Taxon*) generalized to allow for ordinal.
- Also Canberra metric, Czekanowski coefficient, Mahalanobis distance for continuous; Mahalanobis distance for proportions; Jaccard for binary; many others. Levenshtein and Hamming distances for differences in text strings.
- de Leon and Carriére (2005, *JMA*) consider a Mahalanobis-distance for mixed continuous, ordinal, and nominal measurements.