Hierarchical clustering

Description
- Produces a set of nested clusters organized as a hierarchical tree.
- Can be visualized as a dendrogram: A tree-like diagram that records the sequences of merges or splits.

A clustering and its dendrogram.
Hierarchical clustering

**Strengths**
- Do not have to assume any particular number of clusters. Each horizontal cut of the tree yields a clustering.
- The tree may correspond to a meaningful taxonomy: (e.g., animal kingdom, phylogeny reconstruction, ...)
- Need only a similarity or distance matrix for implementation.

Hierarchical clustering

**Agglomerative**
- Start with the points as individual clusters.
- At each step, merge the closest pair of clusters until only one cluster (or some fixed number $k$ clusters) remain.

Hierarchical clustering

**Divisive**
- Start with one, all-inclusive cluster.
- At each step, split a cluster until each cluster contains a point (or there are $k$ clusters).

**Agglomerative Clustering Algorithm**

1. Compute the proximity matrix.
2. Let each data point be a cluster.
3. **While** there is more than one cluster:
   - Merge the two closest clusters.
   - Update the proximity matrix.

The major difference is the computation of proximity of two clusters.
Hierarchical clustering

Starting point for agglomerative clustering.

Intermediate point, with 5 clusters.

We will merge C2 and C5.

How do we update proximity matrix?
We need a notion of similarity between clusters.

Single linkage uses the minimum distance.

Complete linkage uses the maximum distance.

Group average linkage uses the average distance between groups.
Hierarchical clustering

Centroid uses the distance between the centroids of the clusters (presumes one can compute centroids...)

Proximity matrix and dendrogram of single linkage.

Distance matrix for nested clusterings

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.24</td>
<td>0.22</td>
<td>0.37</td>
<td>0.34</td>
<td>0.23</td>
</tr>
<tr>
<td>2</td>
<td>0.15</td>
<td>0.20</td>
<td>0.14</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.15</td>
<td>0.28</td>
<td>0.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.29</td>
<td>0.22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.39</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Nested cluster representation and dendrogram of single linkage.
Hierarchical clustering

**Single linkage**
- Can handle irregularly shaped regions fairly naturally.
- Sensitive to noise and outliers in the form of “chaining”.

The Iris data (single linkage)

Hierarchical clustering

Proximity matrix and dendrogram of complete linkage.
Hierarchical clustering

Nested cluster and dendrogram of complete linkage.

- Complete linkage
  - Less sensitive to noise and outliers than single linkage.
  - Regions are generally compact, but may violate “closeness”. That is, points may much closer to some points in neighbouring cluster than its own cluster.
  - This manifests itself as breaking large clusters.
  - Clusters are biased to be globular.

The Iris data (complete linkage)
Hierarchical clustering

Average linkage

- Given two elements of the partition $C_r, C_s$, we might consider

$$d_{GA}(C_r, C_s) = \frac{1}{|C_r||C_s|} \sum_{x \in C_r, y \in C_s} d(x, y)$$

- A compromise between single and complete linkage.
- Shares globular clusters of complete, less sensitive than single.
Hierarchical clustering

**Ward’s linkage**
- Similarity of two clusters is based on the increase in squared error when two clusters are merged.
- Similar to average if dissimilarity between points is distance squared. Hence, it shares many properties of average linkage.
- A hierarchical analogue of $K$-means.
- Sometimes used to initialize $K$-means.

The Iris data (Ward’s linkage)
Hierarchical clustering

Computational issues

- $O(n^2)$ space since it uses the proximity matrix.
- $O(n^3)$ time in many cases as there are $N$ steps, and at each step a matrix of size $N^2$ must be updated and/or searched.

Hierarchical clustering

Statistical issues

- Once a decision is made to combine two clusters, it cannot be undone.
- No objective function is directly minimized.
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers.
  - Difficulty handling different sized clusters and convex shapes.
  - Breaking large cluster.

Model-based clustering

Part II

Model-based clustering

General approach

- Choose a type of mixture model (e.g. multivariate Normal) and a maximum number of clusters, $K$
- Use a specialized hierarchical clustering technique.
- Uses some criterion to determine optimal model and number of clusters.
Model-based clustering

Choosing a mixture model

- General form of the mixture model
  \[ f(x) = \sum_{j=1}^{k} \pi_j f(x; \theta_j) \]
  
  - For multivariate normal, \( \theta_j = (\mu_j, \Sigma_j) \).
  
  - The EM algorithm we discussed before assumed \( \Sigma_j \) are all different in the different classes.
  
  - Other possibilities: \( \Sigma_j = \Sigma, \Sigma_j = \lambda_j \cdot I, \) etc.

Model-based agglomerative clustering

Ward’s criterion

- A hierarchical clustering algorithm that merges \( k \) clusters \( \{C_1^k, \ldots, C_k^k\} \) into \( k - 1 \) clusters based on
  \[ WSS = \sum_{j=1}^{k-1} WSS(C_j^{k-1}) \]
  where \( WSS \) is the within-cluster sum of squared distances.

- The procedure merges the two clusters \( C_i^k, C_l^k \) that produce the smallest increase in \( WSS \).

NCI data (Ward’s linkage)
Model-based agglomerative clustering

- If $\Sigma_j = \sigma^2 \cdot I$, then Ward’s criterion is equivalent to merging based on the criterion
  $$-2 \log L(\theta, l)$$

  where
  $$L(\theta, l) = \prod_{i=1}^{n} f(x_i; \theta_l)$$

  is called the classification likelihood.

- This idea can be used to make a hierarchical clustering algorithm for other types of multivariate normal models, i.e. equal shape, same size, etc.

Model selection & BIC

- Bayesian Information Criterion
  - Suppose we have several possible models $M = \{M_1, \ldots, M_T\}$ for a data set which we assume is given by a data matrix $X_{n \times p}$.
  - These models have parameters $\Theta = \{\theta_1, \ldots, \theta_T\}$.
  - Further, suppose that each one has a likelihood, $L_j$ and $\hat{\Theta} = \{\hat{\theta}_1, \ldots, \hat{\theta}_T\}$ are the maximum likelihood estimators.
  - We can compare
    $$-2 \log L_j(\hat{\theta}_j)$$

    but this ignores how much “fitting” each model does.
  - A common approach is to add a penalty that makes different models comparable.

- The BIC of a model is usually
  $$BIC(M_j) = -2 \log L_j(\hat{\theta}_j) + \log n \cdot \# \text{ parameters in } M_j.$$
Bayesian Information Criterion

- Typically, statisticians will try to prove choosing model with best BIC yields “correct model”.
- Some theoretical justification is needed for this, and this breaks down for mixture models. Nevertheless, it is still used.
- Another common criterion is AIC (Akaike Information Criterion)

\[
AIC(M_j) = -2 \log L_j(\hat{\theta}_j) + 2 \cdot \# \text{ parameters in } M_j.
\]

The Iris data “best” model: equal shape, 2 components

Summary

1. Choose a type of mixture model (e.g. multivariate Normal) and a maximum number of clusters, \( K \).
2. Use a specialized hierarchical clustering technique: model-based hierarchical agglomeration.
3. Use clusters from previous step to initialize EM for the mixture model.
4. Uses BIC to compare different mixture models and models with different numbers of clusters.
Part III

Outliers

What is an outlier?
The set of data points that are considerably different than the remainder of the data...

Applications
- Credit card fraud detection;
- Network intrusion detection;
- Misspecification of a model.

Concepts
- Given a data matrix $X$, find all the cases $x_i \in X$ with anomaly/outlier scores greater than some threshold $t$. Or, the top $n$ outlier scores.
- Given a data matrix $X$, containing mostly normal (but unlabeled) data points, and a test case $x_{new}^*$, compute an anomaly/outlier score of $x_{new}^*$ with respect to $X$. 

The Iris data
Outliers

**Issues**
- How many outliers are there in the data?
- Method is unsupervised, similar to clustering or finding clusters with only 1 point in them.
- Usual assumption: *There are considerably more “normal” observations than “abnormal” observations (outliers/anomalies) in the data.*

**General steps**
- Build a profile of the “normal” behavior. The profile generally consists of summary statistics of this “normal” population.
- Use these summary statistics to detect anomalies, i.e. points whose characteristics are very far from the normal profile.
- General types of schemes involve a statistical model of “normal”, and “far” is measured in terms of likelihood.
- Other schemes based on distances can be quasi-motivated by such statistical techniques . . .

**Statistical approach**
- Assume a parametric model describing the distribution of the data (e.g., normal distribution)
- Apply a statistical test that depends on:
  - Data distribution (e.g. normal)
  - Parameter of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit, \(\alpha\) or Type I error)

**Grubbs’ Test**
- Suppose we have a sample of \(n\) numbers \(Z = \{Z_1, \ldots, Z_n\}\), i.e. a \(n \times 1\) data matrix.
- Assuming data is from normal distribution, Grubbs’ tests uses distribution of
  \[
  \frac{\max_{1 \leq i \leq n} Z_i - \bar{Z}}{SD(Z)}
  \]
  to search for outlying large values.
Grubbs' Test

- Lower tail variant:
  \[ \min_{1 \leq i \leq n} \frac{Z_i - \bar{Z}}{SD(Z)} \]

- Two-sided variant:
  \[ \max_{1 \leq i \leq n} \frac{|Z_i - \bar{Z}|}{SD(Z)} \]

Having chosen a test-statistic, we must determine a threshold that sets our “threshold” rule.

- Often this is set via a hypothesis test to control Type I error.

  For large positive outlier, threshold is based on choosing some acceptable Type I error \( \alpha \) and finding \( c_\alpha \) so that

  \[ P_0 \left( \max_{1 \leq i \leq n} \frac{Z_i - \bar{Z}}{SD(Z)} \geq c_\alpha \right) \approx \alpha \]

  Above, \( P_0 \) denotes the distribution of \( Z \) under the assumption there are no outliers.

- If \( Z \) are IID \( N(\mu, \sigma^2) \) it is generally possible to compute a decent approximation of this probability using Bonferroni.

Two sided critical level has the form

\[ c_\alpha = \frac{n - 1}{\sqrt{n}} \sqrt{\frac{t^2_{\alpha/(2n), n-2}}{n - 2 + t^2_{\alpha/(2n), n-2}}} \]

where

\[ P(T_k \geq t_{1,k}) = \gamma \]

is the upper tail quantile of \( T_k \).

- In R, you can use the functions `pnorm`, `qnorm`, `pt`, `qt` for these quantities.

Model based: linear regression with outliers

Figure: Residuals from model can be fed into Grubbs’ test or Bonferroni (variant)
Outliers

Multivariate data
• If the non-outlying data is assumed to be multivariate Gaussian, what is the analogy of Grubbs' statistic

\[
\max_{1 \leq i \leq n} \left| \frac{Z_i - \bar{Z}}{\text{SD}(Z)} \right|
\]

• Answer: use Mahalanobis distance

\[
\max_{1 \leq i \leq n} \left( \frac{Z_i - \bar{Z}}{\Sigma^{-1}} \right)^T \Sigma^{-1} \frac{Z_i - \bar{Z}}{\Sigma^{-1}}
\]

• Above, each individual statistic has what looks like a Hotelling’s $T^2$ distribution.

Outliers

Likehood approach
• Assume data is a mixture

\[
F = (1 - \lambda)M + \lambda A.
\]

• Above, $M$ is the distribution of “most of the data.”

• The distribution $A$ is an “outlier” distribution, could be uniform on a bounding box for the data.

• This is a mixture model. If $M$ is parametric, then the EM algorithm fits naturally here.

• Any points assigned to $A$ are “outliers.”

Outliers

Likelihood approach
• Do we estimate $\lambda$ or fix it?

• The book starts describing an algorithm that tries to maximize the equivalent classification likelihood

\[
L(\theta_M, \theta_A; l) = \left( (1 - \lambda)^{\#l_M} \prod_{i \in l_M} f_M(x_i, \theta_M) \right) \times \left( \lambda^{\#l_A} \prod_{i \in l_A} f_A(x_i; \theta_A) \right)
\]

Outliers

Likelihood approach: Algorithm
• Algorithm tries to maximize this by forming iterative estimates $(M_t, A_t)$ of “normal” and “outlying” data points.

• At each stage, tries to place individual points of $M_t$ to $A_t$.

• Find $(\hat{\theta}_M, \hat{\theta}_A)$ based on partition new partition (if necessary).

• If increase in likelihood is large enough, call these new set $(M_{t+1}, A_{t+1})$.

• Repeat until no further changes.
Outliers

Nearest neighbour approach

- Many ways to define outliers.
- Example: data points for which there are fewer than \( k \) neighboring points within a distance \( \epsilon \).
- Example: the \( n \) points whose distance to \( k \)-th nearest neighbour is largest.
- The \( n \) points whose average distance to the first \( k \) nearest neighbours is largest.
- Each of these methods all depend on choice of some parameters: \( k, n, \epsilon \). Difficult to choose these in a systematic way.

Density approach

- For each point, \( x_i \), compute a density estimate \( f_{x_i,k} \) using its \( k \) nearest neighbours.
- Density estimate used is
  \[
  f_{x_i,k} = \left( \frac{\sum_{y \in N(x_i,k)} d(x_i,y)}{\#N(x_i,k)} \right)^{-1}
  \]
- Define
  \[
  LOF(x_i) = \frac{f_{x_i,k}}{(\sum_{y \in N(x_i,k)} f_{y,k})/\#N(x_i,k)}
  \]

Detection rate

- Set \( P(O) \) to be the proportion of outliers or anomalies.
- Set \( P(D|O) \) to be the probability of declaring an outlier if it truly is an outlier. This is the detection rate.
- Set \( P(D|O^c) \) to the probability of declaring an outlier if it is truly not an outlier.

Figure: Nearest neighbour vs. density based
Outliers

Bayesian detection rate

- Bayesian detection rate is

\[ P(O|D) = \frac{P(D|O)P(O)}{P(D|O)P(O) + P(D|O^c)P(O^c)}. \]

- The false alarm rate or false discovery rate is

\[ P(O^c|D) = \frac{P(D|O^c)P(O^c)}{P(D|O^c)P(O^c) + P(D|O)P(O)}. \]