Chapter VIII.3: Hierarchical Clustering

1. Basic idea
   1.1. Dendrograms
   1.2. Agglomerative and divisive

2. Cluster distances
   2.1. Single link
   2.2. Complete link
   2.3. Group average and Mean distance
   2.4. Ward’s method

3. Discussion

ZM Ch. 14; TSK Ch. 8
Basic idea

• Create clustering for each number of clusters $k = 1, 2, \ldots, n$

• The clusterings must be **hierarchical**
  – Every cluster of a $k$-clustering is a union of some clusters in an $l$-clustering for all $l < k$
  – I.e. for all $l$, and for all $k > l$, every cluster in an $l$-clustering is a subset of some cluster in $k$-clustering

• Example:
Dendrograms

The height of the subtree tree shows the distance between the two branches.
Dendrograms and clusters
Dendrograms

- Dendrograms show the hierarchy of the clustering
- The number of clusters can be deduced from dendrogram
  - Higher branches
- Outliers can be detected from dendrograms
  - Single points that are far from others
Agglomerative and divisive

• **Agglomerative**: bottom-up
  – Start with $n$ clusters
  – Combine two closest points into a cluster of two elements
    • Combine two closest clusters into one bigger cluster

• **Divisive**: top-down
  – Start with 1 cluster
  – Divide the cluster into two
    • Divide the largest (per diameter) cluster into two smaller
Cluster distances

• The distance between two points \( x \) and \( y \) is \( d(x,y) \)
• But what is the distance between two clusters?
• Many intuitive definitions – no universal truth
  – Different cluster distances yield different clusterings
  – The selection of cluster distance depends on application
• Some distances between clusters \( B \) and \( C \):
  – minimum distance \( d(B,C) = \min \{d(x,y) : x \in B \text{ and } y \in C\} \)
  – maximum distance \( d(B,C) = \max \{d(x,y) : x \in B \text{ and } y \in C\} \)
  – average distance \( d(B,C) = \text{avg} \{d(x,y) : x \in B \text{ and } y \in C\} \)
  – distance of centroids \( d(B,C) = d(\mu_B, \mu_C) \),
    where \( \mu_B \) is the centroid of \( B \) and \( \mu_C \) is the centroid of \( C \)
Single link

• The distance between two clusters is the distance between the closest points

\[-d(B,C) = \min \{d(x,y) : x \in B \text{ and } y \in C\}\]
Strengths of single-link

Original Points

Two Clusters

Can handle non-spherical clusters of unequal size
Weaknesses of single-link

- Sensitive to noise and outliers
- Produces elongated clusters
The distance between the clusters is the distance between the furthest points

\[-d(B,C) = \max\{d(x,y) : x \in B \text{ and } y \in C\}\]
Strengths of complete link

- Less susceptible to noise and outliers
Weaknesses of complete link

- Breaks largest clusters
- Biased towards spherical clusters
Group average and Mean distance

- **Group average** is the average of pairwise distances
  \[ d(B,C) = \text{avg}\{d(x,y) : x \in B \text{ and } y \in C\} \]
  \[ = \sum_{x \in B, y \in C} d(x, y)/(|B||C|) \]

- **Mean distance** is the distance of the cluster centroids
  \[ d(B,C) = d(\mu_B, \mu_C) \]
Properties of group average

• A compromise between single and complete link
• Less susceptible to noise and outliers
  – Similar to complete link
• Biased towards spherical clusters
  – Similar to complete link
Ward’s method

- **Ward’s distance** between clusters $A$ and $B$ is the increase in sum of squared errors (SSE) when the two clusters are merged

  - SSE for cluster $A$ is $\text{SSE}_A = \sum_{x \in A} ||x - \mu_A||^2$

  - Difference on merging clusters $A$ and $B$ to cluster $C$ is then $d(A, B) = \Delta \text{SSE}_C = \text{SSE}_C - \text{SSE}_A - \text{SSE}_B$

  - Equivalently, $d(A, B) = \frac{|A||B|}{|A|+|B|} ||\mu_A - \mu_B||^2$

  - Weighted mean distance
Discussion on Ward’s method

• Less susceptible to noise and outliers
• Biased towards spherical clusters
• Hierarchical analogue of $k$-means
  – Hence many shared pros and cons
  – Can be used to initialize $k$-means
Comparison

Single link

Complete link

Group average

Ward’s method

Nearest Neighbour

Furthest Neighbour

Single link

Complete link

Group average

Ward’s method

Nearest Neighbour

Furthest Neighbour

Single link

Complete link

Group average

Ward’s method

Nearest Neighbour

Furthest Neighbour

Single link

Complete link

Group average

Ward’s method

Nearest Neighbour

Furthest Neighbour
Lance–Williams formula

• After merging clusters $A$ and $B$ into cluster $C$, we need to compute $C$’s distance to other clusters $Z$
• Lance–Williams formula provides a general equation for this

$$d(C, Z) = \alpha_A d(A, Z) + \alpha_B d(B, Z) + \beta d(A, B) + \gamma |d(A, Z) - d(B, Z)|$$

<table>
<thead>
<tr>
<th>Method</th>
<th>$\alpha_A$</th>
<th>$\alpha_B$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single link</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>-1/2</td>
</tr>
<tr>
<td>Complete link</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>Group average</td>
<td>$</td>
<td>A</td>
<td>/(</td>
<td>A</td>
</tr>
<tr>
<td>Mean distance</td>
<td>$</td>
<td>A</td>
<td>/(</td>
<td>A</td>
</tr>
<tr>
<td>Ward’s method</td>
<td>$(</td>
<td>A</td>
<td>+</td>
<td>Z</td>
</tr>
</tbody>
</table>
Computational complexity

• Takes $O(n^3)$ time in most cases
  – $n$ steps
  – In each step, $n^2$ distance matrix must be updated and searched

• $O(n^2 \log(n))$ time for some approaches using appropriate data structures
  – Keep distances in a heap
  – Each step takes $O(n \log n)$ time

• $O(n^2)$ space complexity
  – Have to store the distance matrix
Chapter VIII.4: Density-Based Clustering

1. The Idea

2. The DBSCAN algorithm

ZM Ch. 15; TSK Ch. 8
The Idea

• Representation-based clustering can only find convex clusters
  – But data can have non-convex interesting clusters

• In **density-based clustering** a cluster contains a dense area of points
  – But how to define dense areas?
Some definitions

• An $\varepsilon$-neighbourhood of point $x$ of data $D$ is the set of points of $D$ that are within $\varepsilon$ distance from $x$
  
  $N_\varepsilon(x) = \{y \in D : d(x, y) \leq \varepsilon\}$

  $\varepsilon$ is a user supplied parameter

• Point $x \in D$ is a **core point** if $|N_\varepsilon(x)| \geq \text{minpts}$

  $\text{minpts}$ is a user supplied parameter

• Point $x \in D$ is a **border point** if it is not a core point but $x \in N_\varepsilon(z)$ for some core point $z$

• A point $x \in D$ that is neither a core nor a border point is called a **noise point**
Example

Core point

Border point

Noise point

\[ \text{minpts} = 5 \]
Density reachability

- Point \( x \in D \) is **directly density reachable** from point \( y \in D \) if
  - \( y \) is a core point
  - \( x \in N_\varepsilon(y) \)

- Point \( x \in D \) is **density reachable** from point \( y \in D \) if there is a chain of points \( x_0, x_1, \ldots, x_l \) s.t. \( x = x_0, y = x_l \), and \( x_{i-1} \) is directly density reachable from \( x_i \) for all \( i = 1, \ldots, l \)
  - Not a symmetric relationship

- Points \( x, y \in D \) are **density connected** if there exists a core point \( z \) s.t. both \( x \) and \( y \) are density reachable from \( z \)
Density-based clusters

- A **density-based cluster** is a maximal set of density connected points.
The DBSCAN algorithm

• for each unvisited point \( x \) in the data
  – compute \( N_\varepsilon(x) \)
  – if \( |N_\varepsilon(x)| \geq \text{minpts} \)
    • ExpandCluster(\( x \), ++clusterID)

• ExpandCluster(\( x \), ID)
  – assign \( x \) to cluster ID and set \( N \leftarrow N_\varepsilon(x) \)
  – for each \( y \in N \)
    • if \( y \) is not visited and \( |N_\varepsilon(y)| \geq \text{minpts} \)
      – \( N \leftarrow N \cup N_\varepsilon(y) \)
    • if \( y \) does not belong to any cluster
      – assign \( y \) to cluster ID
More on DBSCAN

• DBSCAN can return either overlapping or non-overlapping clusters
  – Ties are broken arbitrarily

• The main time complexity comes from computing the neighbourhoods
  – Total $O(n \log n)$ with spatial index structures
    • Won’t work with high dimensions, worst-case is $O(n^2)$

• With the neighbourhoods known, DBSCAN only needs a single pass over the data
The parameters

• DBSCAN requires two parameters, $\varepsilon$ and $\text{minpts}$

• $\text{minpts}$ controls the minimum size of a cluster
  – $\text{minpts} = 1$ allows singleton clusters
  – $\text{minpts} = 2$ makes DBSCAN essentially a single-link clustering
  – Higher values of $\text{minpts}$ avoids the long-and-narrow clusters of single link

• $\varepsilon$ controls the required density
  – Single $\varepsilon$ is not enough if the clusters are of highly different density
Chapter VIII.5: Co-clustering

1. Clustering written with matrices
2. Co-clustering definition
3. Algorithms
Clustering written with matrices

• Let $x_1, x_2, ..., x_n$ be the $m$-dimensional vectors (data points) we want to cluster

• Write these as an $n$-by-$m$ matrix $X$
  – Each data point is one row of $X$

• The exclusive representative clustering can be re-written using two matrices
  – Matrix $C$ (cluster assignment matrix) has $n$ rows and $k$ columns
  – Each row of $C$ has exactly one element 1 while others are 0
  – Matrix $M$ (mean matrix) has $k$ rows and $m$ columns
  – Each row of $M$ corresponds to a centroid of a cluster

• Loss function (SSE) is now $\|X - CM\|_2^2$
Example

$\begin{array}{c|c|c}
\mathbf{x} & 1 & 3 \\
\hline
\mathbf{x}_1 & 1 & 3 \\
\mathbf{x}_2 & 2 & 2 \\
\mathbf{x}_3 & 3 & 4 \\
\mathbf{x}_4 & 2 & 1 \\
\mathbf{x}_5 & 4 & 3 \\
\end{array}$

$C_1 = \{x_1, x_2, x_4\}$
$C_2 = \{x_3, x_5\}$

$\mu_1 = (1.66, 2)$
$\mu_2 = (3.5, 3.5)$

$X = \begin{pmatrix}
1 & 3 \\
2 & 2 \\
3 & 4 \\
2 & 1 \\
4 & 3
\end{pmatrix}$

$C = \begin{pmatrix}
1 & 0 \\
1 & 0 \\
0 & 1 \\
1 & 0 \\
0 & 1
\end{pmatrix}$

$M = \begin{pmatrix}
1.66 & 2 \\
3.5 & 3.5
\end{pmatrix}$

$X - CM = \begin{pmatrix}
1.666 & 2 & 1 \\
10.663 & 2 & 0 \\
3.65 & 3.5 & 5 \\
10.663 & 2 & -1 \\
30.55 & 3.5 & 0.5
\end{pmatrix}$
Co-clustering definition

• The same way we clustered $X$, we can also cluster $X^T$
  – This clusters the dimensions, not the data points

• An \((k,l)\)-co-clustering of $X$ is partitioning of rows of $X$
  into $k$ clusters and columns of $X$ into $l$ clusters
  – Row cluster $I$ and column cluster $J$ define a (combinatorial)
    sub-matrix $X_{IJ}$
    • Element $x_{ij}$ belongs to this sub-matrix if $i \in I$ and $j \in J$
  – Each sub-matrix $X_{IJ}$ is represented by single value $\mu_{ij}$

• Let $R$ be the \(n\)-by-\(k\) row cluster assignment matrix and $C$
  the \(m\)-by-\(l\) column cluster assignment matrix and
  $M = (\mu_{ij})$ the \(k\)-by-\(l\) mean matrix
  – The loss function is $\|X - RMC^T\|_2^2$
Example \((3,2)\)-co-clustering

\[
X = \begin{pmatrix}
1 & 3 & 2 \\
2 & 2 & 1 \\
0 & 1 & 0 \\
4 & 3 & 5
\end{pmatrix}
- \quad RMC^T = \begin{pmatrix}
1.5 & 2.5 & 1.5 \\
1.5 & 2.5 & 1.5 \\
0 & 1 & 0 \\
4.5 & 3 & 4.5
\end{pmatrix}
= \begin{pmatrix}
0.5 & 0.5 & 0.5 \\
0.5 & 0.5 & 0.5 \\
0 & 0 & 0 \\
0.5 & 0 & 0.5
\end{pmatrix}
\]

\[
R = \begin{pmatrix}
1 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix} \quad M = \begin{pmatrix}
1.5 & 2.5 \\
0 & 1 \\
4.5 & 3
\end{pmatrix} \quad C^T = \begin{pmatrix}
1 & 0 & 1 \\
0 & 1 & 0
\end{pmatrix}
\]
Algorithm

1. input data matrix $X$ and two integers $k$ and $l$
2. Cluster the rows of $X$ to $R$ (using e.g. $k$-means)
3. Cluster the columns of $X$ to $C$
4. Let $M = (\mu_{IJ})$, $\mu_{IJ} = (|I||J|)^{-1} \sum_{i \in I, j \in J} x_{ij}$
5. return $R$, $C$, and $M$
Chapter VIII.6: Discussion and clustering applications

1. Kleinberg’s impossibility theorem
   1.1. Kannan—Hopcroft possibility theorem

2. Example clustering applications
Kleinberg’s impossibility theorem

• A clustering function is a function $f$ that takes a distance matrix $D$ and returns a partition $\Gamma$
  – We expect nothing on the type of points
  – Distance is given using an implicit distance matrix
  – The number of clusters is defined somehow by the clustering function (build-in constant or something else)
  – For example, an algorithm returning a $k$-means clustering to $k=10$ clusters could be one clustering function

• Idea: list some properties any clustering function should satisfy and show that none can satisfy them all
Three properties

• *Scale-invariance*
  – Clustering does not change if we multiply the distances
  – $f(D) = f(\alpha D)$ for any $\alpha > 0$

• *Richness*
  – For any partition $\Gamma$, there is a distance matrix $D$ such that $f(D) = \Gamma$

• *Consistency*
  – The clustering does not change if we move points in the same cluster closer to each other and points in different clusters further away from each other
Impossibility result

**Theorem** [Kleinberg ’02]. There does not exist any clustering function $f$ that satisfies all three properties.

- Single-link hierarchical clustering that stops at $k < n$ clusters satisfies scale-invariance and consistency
- Single-link clustering that stops when the link length is some predefined fraction of maximum pairwise distance satisfies scale-invariance and richness
- Single-link that stops when the link length is longer than some predefined length satisfies richness and consistency
Kannan—Hopcroft possibility theorem

- Let’s assume we work on a finite Euclidean space
- Let’s replace Richness with *Richness II*:
  - For any set $C$ of $k$ points in the Euclidean space, there is an $n$ and a set $D$ of $n$ points such that the centers of the clustering $f(D)$ are exactly the $k$ points in $C$
    - Richness: all clusterings are achievable with proper metric
    - Richness II: all set of centers are achievable with proper set of points

**Theorem** [Kannan & Hopcroft ’13]. There is a clustering function $f$ that satisfies Scale invariance, Consistency, and Richness II.
Some clustering applications

• Biology
  – Creation of phylogenies (relations between organisms)
  – Inferring population structures from clusterings of DNA data
  – Analysis of genes and cellular processes (co-clustering)

• Business
  – Grouping of consumers into market segments

• Computer science
  – Pre-processing step to reduce computation (representative-based methods)
  – Automatic discovery of similar items
More clustering applications

Figure 408: Temperament cluster centers in the NFBC66 best model as starplots of normalized data. The male and female seem similar to each other despite having been learned independently. The subscales are as follows:

HAz1: anticipatory worry, HAz2: fear of uncertainty, HAz3: shyness, HAz4: fatigability;
NSz1: exploratory excitability, NSz2: impulsiveness, NSz3: extravagance, NSz4: disorderliness;
RDz1: sentimentality, RDz3: attachment, RDz4: dependence; P: persistence.

Stable, persistent, not very impulsive
High socio-economical status and education

Shy, pessimistic, prefer routines and privacy
Low socio-economic status, high levels of depression and schizophrenia

Wessman: Clustering methods in the analysis of complex diseases
Even more clustering applications
Summary

• Clustering is one of the most important and often-used data analysis methods

• Many different types of clustering
  – We covered representative-based, hierarchical, density-based, and co-clustering

• Analysis of the clustering methods is not always easy

• Always think what you’re doing if you use clustering
  – In fact, just always think what you’re doing