Stat 315c: Transposable Data Clustering

Art B. Owen

Stanford Statistics
Given $n$ objects with $d$ attributes, place them (the objects) into groups.
Clustering

Given $n$ objects with $d$ attributes, place them (the objects) into groups.

A form of unsupervised learning. Unsupervised because there is no response.
Clustering

- Given $n$ objects with $d$ attributes, place them (the objects) into groups.
- A form of unsupervised learning. Unsupervised because there is no response.
- Has a long history, at least as old as taxonomy.
_Clustering_

- Given $n$ objects with $d$ attributes, place them (the objects) into groups.
- A form of unsupervised learning. Unsupervised because there is no response.
- Has a long history, at least as old as taxonomy.
- Raises all the vexing issues of an exploratory method.
Clustering

- Given $n$ objects with $d$ attributes, place them (the objects) into groups.
- A form of unsupervised learning. Unsupervised because there is no response.
- Has a long history, at least as old as taxonomy.
- Raises all the vexing issues of an exploratory method.
- We’ll look at it as a precursor to 'bi-clustering' of objects and attributes.
Given \( n \) points in \( \mathbb{R}^d \), do they clump together into \( k \) clusters? If so, how to find the clusters, and the boundaries, and cluster identities?

Outcomes

In the best case, a clustering can reveal the presence of a new categorical variable, e.g. types of diabetes. Other times there are no clusters, just a 'smear'. Or we find clusters but not their meanings.

Key idea

Items within a cluster are more similar (less distant) to each other than items from different clusters.
Given \( n \) points in \( \mathbb{R}^d \):

- Do they clump together into \( k \) clusters?
- If so, how to find the clusters,
- and the boundaries,
- and cluster identities?

Outcomes

- In the best case, a clustering can reveal the presence of a new categorical variable, e.g. types of diabetes.
- Other times there are no clusters, just a ‘smear’.
- Or we find clusters but not their meanings.

Key idea

Items within a cluster are more similar (less distant) to each other than items from different clusters.
Clustering

Given $n$ points in $\mathbb{R}^d$
- Do they clump together into $k$ clusters?
- If so, how to find the clusters,
- and the boundaries,
- and cluster identities?

Outcomes

In the best case, a clustering can reveal the presence of a new categorical variable, e.g. types of diabetes.
Other times there are no clusters, just a ‘smear’
Or we find clusters but not their meanings.

Key idea
Items within a cluster are more similar (less distant) to each other than items from different clusters.
Given \( n \) points in \( \mathbb{R}^d \)

- Do they clump together into \( k \) clusters?
- If so, how to find the clusters, and the boundaries, and cluster identities?

**Outcomes**

- In the best case, a clustering can reveal the presence of a new categorical variable, e.g. types of diabetes.
- Other times there are no clusters, just a 'smear'
- Or we find clusters but not their meanings.
Clustering

Given $n$ points in $\mathbb{R}^d$

- Do they clump together into $k$ clusters?
- If so, how to find the clusters,
- and the boundaries,
- and cluster identities?

Outcomes

- In the best case, a clustering can reveal the presence of a new categorical variable, e.g. types of diabetes.
- Other times there are no clusters, just a 'smear'
- Or we find clusters but not their meanings.

Key idea

Items within a cluster are more similar (less distant) to each other than items from different clusters.
**k-means**

- **Algorithm**
  - Pick $k$ points $z_1, \ldots, z_k \in \mathbb{R}^d$
  - Repeat
    1. For $i = 1, \ldots, n$
       - Put $g(i) = \min_{1 \leq j \leq k} \|x_i - z_j\|_2$
    2. For $j = 1, \ldots, k$
       - Put $z_j = \text{avg}\{x_i | g(i) = j\}$

- **Issues**
  - Handle averaging over empty set
  - Pick stopping rule (it must converge, or at worst cycle)
  - Answer depends on starting points

- **Properties**
  - Usually rapid convergence
  - An iteration can be done in $O(nkd)$ time. No $n^2$ or $d^2$ or $k^2$.
    ($k \log(k)$ to sort negligible)
**$k$-means**

**Algorithm**

1. Pick $k$ points $z_1, \ldots, z_k \in \mathbb{R}^d$, then repeat
   - For $i = 1, \ldots, n$ put $g(i) = \min_{1 \leq j \leq k} \|x_i - z_j\|^2$
   - For $j = 1, \ldots, k$ put $z_j = \text{avg}\{x_i \mid g(i) = j\}$

**Issues**
- Handle averaging over empty set
- Pick stopping rule (it must converge, or at worst cycle)
- Answer depends on starting points
- Hard to pick $k$ (at least 30 methods proposed by 1985)

**Properties**
- Usually rapid convergence
- An iteration can be done in $O(nkd)$ time. No $n^2$ or $d^2$ or $k^2$.
  ($k \log(k)$ to sort negligible)
**k-means**

**Algorithm**

1. Pick $k$ points $z_1, \ldots, z_k \in \mathbb{R}^d$, then repeat
   - For $i = 1, \ldots, n$ put $g(i) = \min_{1 \leq j \leq k} \|x_i - z_j\|^2$
   - For $j = 1, \ldots, k$ put $z_j = \text{avg}\{x_i \mid g(i) = j\}$

**Issues**

- Handle averaging over empty set
- Pick stopping rule (it must converge, or at worst cycle)
- Answer depends on starting points
- Hard to pick $k$ (at least 30 methods proposed by 1985)
**Algorithm**

1. Pick \( k \) points \( z_1, \ldots, z_k \in \mathbb{R}^d \), then repeat
   - For \( i = 1, \ldots, n \) put \( g(i) = \min_{1 \leq j \leq k} \| x_i - z_j \|^2 \)
   - For \( j = 1, \ldots, k \) put \( z_j = \text{avg}\{ x_i \mid g(i) = j \} \)

**Issues**

- Handle averaging over empty set
- Pick stopping rule (it must converge, or at worst cycle)
- Answer depends on starting points
- Hard to pick \( k \) (at least 30 methods proposed by 1985)

**Properties**

- Usually rapid convergence
- An iteration can be done in \( O(nkd) \) time. No \( n^2 \) or \( d^2 \) or \( k^2 \). \((k \log(k)) \) to sort negligible)
$k$-means ctd

There’s a criterion

- Each step minimizes

$$\sum_{i=1}^{n} \| x_i - zg(i) \|^2$$

over its free variables

Hartigan and Wong (1979) get solutions where no $zg(i)$ change reduces ss

Exact min infeasible to get $k$-means

Pelleg and Moore: Efficient lookups to get $k$ into tens of thousands

Picks $k$ via AIC or BIC along the way

Defining true $k$ problematics (galaxies in clusters in super-clusters)
There’s a criterion

- Each step minimizes

\[ \sum_{i=1}^{n} \left\| x_i - z_{g(i)} \right\|^2 \]

over its free variables

- Hartigan and Wong (1979) get solutions st no \( g(i) \) change reduces ss
There's a criterion

- Each step minimizes
  \[ \sum_{i=1}^{n} \| x_i - z_g(i) \|^2 \]
  over its free variables
- Hartigan and Wong (1979) get solutions st no \( g(i) \) change reduces ss
- Exact min infeasible to get

**x-means**

- Pelleg and Moore
- Efficient lookups to get \( k \) into tens of thousands
- Picks \( k \) via AIC or BIC along the way
There's a criterion

- Each step minimizes
  \[ \sum_{i=1}^{n} \| x_i - z_{g(i)} \|^2 \]
  over its free variables
- Hartigan and Wong (1979) get solutions st no \( g(i) \) change reduces ss
- Exact min infeasible to get

*x*-means

- Pelleg and Moore
- Efficient lookups to get \( k \) into tens of thousands
- Picks \( k \) via AIC or BIC along the way

Defining true \( k \) problematic (galaxies in clusters in super-clusters)
Variations

Change dist

- Change the distance $L^2$ to $L^1$ to · · · $L^p$
- Changes mean to median to · · · arg min
Variations

Change dist
- Change the distance $L^2$ to $L^1$ to $\cdots$ $L^p$
- Changes mean to median to $\cdots$ arg min

Change $z$
- $k$ medoids
- Require $z_k = x_{i(k)}$ for some $i(k)$
- Avoids using $z$ with 2.3 kids, 30% pregnant, 10% male
- PAM “partitioning around medoids”
- Minimize $\sum_i \min_j D(x_i, z_j)$
- Slow. Uses only $D_{ij}$ values.
What is a cluster?

Defining issues

- Scale of variables matters
- Subset of variables matters too
  - Do whales go with penguins or with elephants?
  - Why does my breakfast cereal cluster with pure sugar?
- # Density bumps ≠ # mixture components

Scaling data

$$z_{ij} = \frac{x_{ij} - m_j}{s_j}$$

$m, s$ are mean & stdev (so $z_j \sim (0, 1)$)
or min & range (so $0 \leq z_{ij} \leq 1$)
or median & MAD (for robustness)

NB: transformation defined column-wise [vs row-wise or simultaneous]
What is a cluster?

Defining issues
- Scale of variables matters
- Subset of variables matters too
  - Do whales go with penguins or with elephants?
  - Why does my breakfast cereal cluster with pure sugar?
- # Density bumps ≠ # mixture components

Scaling data

\[ z_{ij} = \frac{x_{ij} - m_j}{s_j} \]

- \( m, s \) are mean & stdev (so \( z_j \sim (0, 1) \))
- or min & range (so \( 0 \leq z_{ij} \leq 1 \))
- or median & MAD (for robustness)
- NB: transformation defined column-wise [vs row-wise or simultaneous]
Weighting

Weight variables via $w_j \geq 0$

$$d_{ii'} = \sum_{j=1}^{d} w_j |x_{ij} - x_{i'j}| = \sum_{j=1}^{d} |\tilde{x}_{ij} - \tilde{x}_{i'j}|$$

with

$$\tilde{x}_{ij} = x_{ij} \times w_j$$

Weighting and scaling are equivalent (for $L^p$ distances)

Automatic scaling not always sensible. Variables in the same units should sometimes get the same scaling even if they have different variances.
Distances

\( n(n-1)/2 \) interpoint distances

- \( d_{ii'} = \text{dist}(x_i, x_{i'}) \)
- Usually
  1. \( d_{ii'} \geq 0 \)
  2. \( d_{ii} = 0 \)
  3. \( d_{ii'} = d_{i'i} \)

A metric distance has \( d_{ij} \leq d_{ik} + d_{jk} \) (Triangle inequality)

A 'Euclidean' distance \( d_{ij} = \|z_i - z_j\| \) for some points \( z_i = z(x_i) \)

An ultra-metric distance has \( d_{ij} \leq \max(d_{ik}, d_{jk}) \) (More later)
Distances

\( n(n - 1)/2 \) interpoint distances

- \( d_{ii'} = \text{dist}(x_i, x_{i'}) \)
- Usually
  1. \( d_{ii'} \geq 0 \)
  2. \( d_{ii} = 0 \)
  3. \( d_{ii'} = d_{i'i} \)

- A metric distance has \( d_{ij} \leq d_{ik} + d_{jk} \) (Triangle inequality)
- A 'Euclidean' distance \( d_{ij} = \|z_i - z_j\| \) for some points \( z_i = z(x_i) \)
- An ultra-metric distance has \( d_{ij} \leq \max(d_{ik}, d_{jk}) \) (More later)

Other distances

- Canberra distance \( \sum_{j=1}^{d} \frac{|x_{ij} - x_{ij'}|}{|x_{ij}| + |x_{ij'}|} \) (with 0/0 = 0)
- Angular or cosine distance (dogs||cats, wolves||tigers)
Similarities

Opposite of distance: $d \leftrightarrow S$

Eg $S_{ii'} = 1 - d_{ii'}$ or $1/d_{ii'}$ or $d_{ii'} = S - S_{ii'}$
**Similarities**

**Opposite of distance:** \(d \leftrightarrow S\)

\[ S_{ii'} = 1 - d_{ii'} \text{ or } 1/d_{ii'} \text{ or } d_{ii'} = S - S_{ii'} \]

**Correlation type similarities**

\[ S_{ii'} = \frac{\sum_{j=1}^{d} x_{ij} x_{i'j}}{\sqrt{\sum_{j=1}^{d} x_{ij}^2 \sum_{j=1}^{d} x_{i'j}^2}}, \quad \text{or,} \]
\[ = \left| \frac{\sum_{j=1}^{d} x_{ij} x_{i'j}}{\sqrt{\sum_{j=1}^{d} x_{ij}^2 \sum_{j=1}^{d} x_{i'j}^2}} \right|, \quad \text{or,} \]
\[ = \left| \frac{\sum_{j=1}^{d} (x_{ij} - \bar{x}_j)(x_{i'j} - \bar{x}_j)}{\sqrt{\sum_{j=1}^{d} (x_{ij} - \bar{x}_j)^2 \sum_{j=1}^{d} (x_{i'j} - \bar{x}_j)^2}} \right|, \quad \text{or,} \cdots \]
Some equalities are more equal than others

1. $i$ and $i'$ are both Nobel laureates (unusually strong similarity)
2. $i$ and $i'$ are both over 21 years old (mild similarity)
3. $i$ and $i'$ are both not Nobel laureates (barely similar at all)

We can handle 1 vs 2 by weighting the variables.
But 1 vs 3 is trickier (same variable).
Binary similarity measures  \[ d = 1 - S \]

**p features 2 × 2 table**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>0</td>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>

We want to count \( a \) more than \( d \)

**Generic measure: \( \alpha > 0, \delta \geq 0 \)**

\[
S_{ii'} = \frac{\alpha a + \delta d}{\alpha a + b + c + \delta d}
\]

\((\alpha, \delta)\) and \((\alpha', \delta')\) give the same ranking if \( \alpha \delta' = \alpha' \delta \)

**Specific measures**

- **Simple matching**
  \[
  S_{ii'} = \frac{a + d}{a + b + c + d}
  \]

- **Jaccard-Tanimoto**
  \[
  S_{ii'} = \frac{a}{a + b + c}
  \]
  \(= 1 \) when \( a + b + c = 0 \)

- **Russel-Rao**
  \[
  S_{ii'} = \frac{a}{a + b + c + d}
  \]

- **Sokal-Sneath I**
  \[
  S_{ii'} = \frac{2(a + d)}{2(a + d) + b + c}
  \]

- **Sokal-Sneath II**
  \[
  S_{ii'} = \frac{a}{a + 2(b + c)}
  \]

Janowitz recommends Jaccard or Russel-Rao
Agglomerative clustering

**General**

- Start with \( n \) clusters of one element each
- Repeat
  1. Find closest two clusters
  2. Merge them into a new cluster
- Until only one cluster remains
- We need point to cluster and cluster to cluster distances
Flavors of agglomerative clustering

**Single linkage**

\[ d(C_1, C_2) = \min_{i \in C_1} \min_{j \in C_2} d_{ij} \]

Get 'chaining'; friends of friends

**Complete linkage**

\[ d(C_1, C_2) = \max_{i \in C_1} \max_{j \in C_2} d_{ij} \]

Get dense nearly spherical clusters

**Average linkage**

\[ d(C_1, C_2) = \frac{1}{|C_1|} \frac{1}{|C_2|} \sum_{i \in C_1} \sum_{j \in C_2} d_{ij} \]

Compromise
Example

Bird data

```r
> dim(voeg)
[1] 395 34
> voeg[1:5,1:5]
Heron Mallard Sparrowhawk Buzzard Kestrel
1 0 0 0 0 1
2 1 0 0 0 0
3 0 0 1 0 0
4 0 0 1 0 0
5 1 1 0 0 0
```

1 means that place $i$ has bird $j$
Place names not present
Complete linkage dendrogram

Birds clustered by habitat

hclust(\texttt{\textasciitilde}, \texttt{\textasciitilde}complete\texttt{\textasciitilde})
Single linkage dendrogram

Birds clustered by habitat

hclust (*, 'single')
Average linkage dendrogram

Birds clustered by habitat

```
Rclus<-
```

Art B. Owen  (Stanford Statistics)  Clustering  19 / 26
Complete vs Average linkage dendrograms

Birds clustered by habitat

hclust(", complete")

Birds clustered by habitat

hclust(", average")
Heatmap
Dendrogram details

Ordering
- $n$ points $\implies n - 1$ splits $\implies 2^{n-1}$ orderings
- R puts 'tightest' cluster on the left
- heatmap lets you control ordering somewhat

Ultrametric
- $H_{ij}$ height at which clusters containing $i$ and $j$ merge
- It’s an ultrametric: for $i, j, k$ top two of $H_{ij}, H_{ik}, H_{jk}$ are equal
- Need $H_{ij} \leq \max(H_{ik}, H_{jk})$
- Non-ultrametric measures yield dendrograms with 'reversals'
Centroid merging

\[ d(C_1, C_2) = \| \bar{X}_{C_1} - \bar{X}_{C_2} \| \]

Requires original points, not just distances

Ward’s

\[ d(C_1, C_2) = \frac{2|C_1||C_2|}{|C_1| + |C_2|} \| \bar{X}_{C_1} - \bar{X}_{C_2} \| \]

Via within cluster SS (after \(-\) before)
Goal is to min

\[ \sum_j \sum_{i \in C_j} \| X_i - \bar{X}_{C_j} \|^2 \]
 Costs of hierarchical splits

- It takes $O(n^2 d)$ to get all pairwise distances
- We have to take $O(n)$ steps
- Naive implementation would be $O(n^3 d)$.

Lance-Williams family of methods

Dist of $i \cup j$ to $k$

$$\alpha_i d_{ki} + \alpha_j d_{kj} + \beta d_{ij} + \gamma |d_{ki} - d_{kj}|$$

eg: $\alpha = 1/2$, $\beta = 0$, $\gamma = -1/2$ for single linkage.
$\alpha_i$ can depend on $n_i$.
Updates lead to $O(n^2 d)$ cost.
Divisive clustering

Recipe

- Start with one cluster of \( n \) objects
- Repeat
  1. Select one cluster
  2. Split it into two
- Until there are \( n \) clusters of size 1

Choices

- Which cluster to split. E.g. largest 'diameter'
  \[
  \arg \max_j \max_{i,i' \in C_j} \| X_i - X_{i'} \|
  \]
- How to split it. E.g. remove far point, \( X_i \) goes with nearer of far point, \( \bar{X}_{C_j \text{—far}} \)

Apparently no good speedup
Optimization based clustering

E.g. Ward’s

Recipe

- Measure quality of a cluster
  - Scale est, like diameter, RMS width, median width
- Combine into quality of clustering
  - Typically the sum (max plausible)
- (Attempt to) optimize

Isolation measures

- Split $\min_{i \in C, j \not\in C} d_{ij}$
- Cut $\sum_{i \in C, j \not\in C} d_{ij}$