Cluster Analysis Utilities for Stata

Brendan Halpin, Dept of Sociology, University of Limerick

Stata User Group Meeting, Science Po, Paris, 6 July 2017
Extending Stata’s cluster capabilities

- Stata’s cluster/clustermat suite is a stable and extensive, but some gaps
- I propose a number of extensions
  - Comparison of cluster solutions: ari and permtab
  - Visualisations: silhouette plots and distance-matrix heatmaps
  - Cluster stopping rule utilities for distance matrices
  - Clustering based on medoids: PAM, fuzzy clustering

Slides: http://teaching.sociology.ul.ie/sugparis
Comparing cluster solutions: "unlabelled"

- Problem: comparing clusterings of the same data using different parameters or algorithms
- Cluster solutions are "unlabelled classifications"
  - Identity is only given by the cases they contain
- We compare solution sets in terms of the extent to which the partitioning of cases is similar
- Two implementations: ARI and PERMTAB
Adjusted Rand Index

- The adjusted Rand Index reports agreement based on all possible pairs of cases (Vinh et al., 2009)
- The index is higher where
  - if both elements of a pair are in the same cluster in one solution, they are also in the same cluster in the other solution
  - if both elements of a pair are in different clusters in one solution, they are also in different clusters in the other solution
- A perfect match yields a value of 1.0.
- Values below zero are possible but rare
use iris
gen id=_n
ccluster wards Sepal_Length Sepal_Width ///
    Petal_Length Petal_Width
ccluster gen g3 = groups(3)
ccluster kmedians Sepal_Length Sepal_Width ///
    Petal_Length Petal_Width, k(3) name(k3)
tab g3 k3
ari g3 k3
Stata Output

```
. tab g3 k3

<table>
<thead>
<tr>
<th>g3</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>61</td>
<td>3</td>
<td>0</td>
<td>64</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>36</td>
<td>0</td>
<td>36</td>
</tr>
<tr>
<td>Total</td>
<td>61</td>
<td>39</td>
<td>50</td>
<td>150</td>
</tr>
</tbody>
</table>

. ari g3 k3
Adjusted Rand Index: 0.9422
```
Permuting tables

- `permtab` has the same motivation but a different strategy.
- It tabulates the two cluster solutions, and permutes the column variable to maximise Cohen’s Kappa (Reilly et al., 2005).
- $\kappa_{\text{max}}$ will generally behave like ARI.
- The advantage of `permtab` is that you can view the best permutation, and save it as a new cluster variable.
. permtab g3 k3, gen(k3a)
Calculating permutations:
Kappa max: 0.9694
Permutation vector:
   1  2  3

   1  3  1  2

Permutated column variable generated from k3: k3a
. tab g3 k3a

<table>
<thead>
<tr>
<th></th>
<th>k3a</th>
<th></th>
<th></th>
<th></th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>g3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0</td>
<td>0</td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>61</td>
<td>3</td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>36</td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>50</td>
<td>61</td>
<td>39</td>
<td></td>
<td>150</td>
</tr>
</tbody>
</table>
permtab limits

- By default, permtab searches exhaustively through all permutations
- Uses Mata’s cvpermute permutation infrastructure
- For up to 8-10 clusters this is feasible, but time is $O(n!)$
  - If 8 clusters take 0.5s, 16 will take 8 years
- A heuristic solution provides very good results: hillclimb
Hill climb

Take the existing order

- Examine all pairwise swaps
- Implement the one with the biggest improvement in $\kappa$, if any
- Iterate until no improvement is found

Generates good results as long as there is some common pattern
. permtab z10 m10, algo(hc)
Calculating permutations:
Kappa max: 0.5255
Permutation vector:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>8</td>
<td>4</td>
<td>7</td>
<td>3</td>
<td>10</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>
Two visualisations are presented

- The silhouette plot
- The heatmap of the cluster-ordered distance matrix
Silhouette plots

- The silhouette statistic (Rousseeuw, 1987) indexes how well cases are located in clusters

\[ h_i = \frac{b_i - a_i}{\max(a_i, b_i)} \]  \hspace{1cm} (1)

where \( a_i \) is mean distance to members of the same cluster, \( b_i \) to the next nearest cluster

- Where clusters are properly distinct this will be closer to 1 than 0

- Cases can be "mis-assigned", being nearer the centre of another cluster than their own: negative silhouette width
Silhouette on Iris data

```
cluster wards Sepal_Length Sepal_Width ///
     Petal_Length Petal_Width
cluster gen g3 = groups(3)
matrix dissim di = Sepal_Length Sepal_Width ///
     Petal_Length Petal_Width, L2Squared
silhouette g3, dist(di) id(id) lwidth(0.8 0.8 0.8)
```
Silhouette plot
IMS lifecourse data: some problematic clusters
Visualising the distance matrix: DHM

- The distance matrix is at the heart of cluster analysis
- `dhm` allows us to visualise it as a heatmap
- Order is important: e.g., group by cluster solution, order within by dendrogram order or silhouette width
Cluster Analysis
Utilities for Stata
Brendan Halpin, Dept of Sociology, University of Limerick

Extending Stata Clustering
Comparing solutions: ari and permtab
Visualisations
Silhouette
Distance matrix heatmap
Cluster stopping rules
Calinski Duda-Hart
Partitioning around Medoids
Extracting medoids
PAM for distance matrices
PAM Step by Step clpam
Fuzzy clustering
Accessing

References

Towns in France: distance re monthly rainfall

http://math.agrocampus-ouest.fr/infoglueDeliverLive/digitalAssets/73503_pluie.csv
Cluster Analysis
Utilities for Stata
Brendan Halpin,
Dept of Sociology,
University of Limerick

Extending Stata
Clustering
Comparing solutions: ari and permtab
Visualisations
Silhouette
Distance matrix heatmap
Cluster stopping rules
Calinski
Duda-Hart
Partitioning around Medoids
Extracting medoids
PAM for distance matrices
PAM Step by Step clpam
Fuzzy clustering

Accessing
References

Towns in France: distance re monthly rainfall
IMS life-histories, dendrogram order
IMS life-histories, silhouette order
DHM syntax for previous 2 slides

- Distances are in matrix pwd; the grouping variable is g8
- g999 is a cluster group variable with a maximal number of clusters
- sw is a variable containing the silhouette width

cluster generate g999 = groups(9999), ties(fewer)
silhouette g8, dist(pwd) id(id) gen(sw)
dhm, mat(pwd) by(g8) order(g999) levels(100) box
dhm, mat(pwd) by(g8) order(sw) levels(100) box
Cluster stopping rules

How do we know how many clusters?
  - Theory?
  - Inspection of the data?
Two common indices: Caliński-Harabasz and Duda-Hart
Provided by Stata in cluster stop and cluster stop, duda
Do not work when clustering from distance matrices
Caliński-Harabasz index

- The CH logic is ANOVA-like: how much better is SS within clusters relative to overall SS (Caliński and Harabasz, 1974; Milligan and Cooper, 1985)
- Internally Stata calculates this by running ANOVAs, regressing each variable on the solution and cumulating a pseudo-F:

\[ pF = \frac{\sum MSS/(g - 1)}{\sum RSS/(N - g)} \] (2)
Equivalence

- However, there is an equivalence between squared deviations from the mean and squared pairwise distances:

\[ SS = \sum_{i=1}^{N} (x_i - \bar{x})^2 = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=i+1}^{N} (x_i - x_j)^2 \]  

(3)

- Thus we can also calculate the CH index from the pairwise distances:

\[ pF = \frac{(SS_t - \sum SS_g)/(g - 1)}{(\sum SS_g)/(N - g)} \]  

(4)

- See Halpin (2016) for more detail
cluster stop and calinski

cluster stop on variables

. cluster wards janvierp-decembrep
   cluster name: _clus_1
. cluster stop

<table>
<thead>
<tr>
<th>Number of clusters</th>
<th>Calinski/Harabasz pseudo-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>17.56</td>
</tr>
<tr>
<td>3</td>
<td>18.53</td>
</tr>
<tr>
<td>4</td>
<td>22.35</td>
</tr>
<tr>
<td>5</td>
<td>21.42</td>
</tr>
<tr>
<td>6</td>
<td>20.15</td>
</tr>
<tr>
<td>7</td>
<td>19.95</td>
</tr>
<tr>
<td>8</td>
<td>20.77</td>
</tr>
<tr>
<td>9</td>
<td>22.29</td>
</tr>
<tr>
<td>10</td>
<td>23.05</td>
</tr>
<tr>
<td>11</td>
<td>23.71</td>
</tr>
<tr>
<td>12</td>
<td>24.14</td>
</tr>
<tr>
<td>13</td>
<td>24.44</td>
</tr>
<tr>
<td>14</td>
<td>24.87</td>
</tr>
<tr>
<td>15</td>
<td>25.02</td>
</tr>
</tbody>
</table>

calinski on the distance matrix

. matrix dissim dd = janvierp-decembrep, L2squared
. calinski, dist(dd) id(id)

<table>
<thead>
<tr>
<th>Number of clusters</th>
<th>Calinski-Harabasz pseudo-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>17.56</td>
</tr>
<tr>
<td>3</td>
<td>18.53</td>
</tr>
<tr>
<td>4</td>
<td>22.35</td>
</tr>
<tr>
<td>5</td>
<td>21.42</td>
</tr>
<tr>
<td>6</td>
<td>20.15</td>
</tr>
<tr>
<td>7</td>
<td>19.95</td>
</tr>
<tr>
<td>8</td>
<td>20.77</td>
</tr>
<tr>
<td>9</td>
<td>22.29</td>
</tr>
<tr>
<td>10</td>
<td>23.05</td>
</tr>
<tr>
<td>11</td>
<td>23.71</td>
</tr>
<tr>
<td>12</td>
<td>24.14</td>
</tr>
<tr>
<td>13</td>
<td>24.44</td>
</tr>
<tr>
<td>14</td>
<td>24.87</td>
</tr>
<tr>
<td>15</td>
<td>25.02</td>
</tr>
</tbody>
</table>
Advantages

- calinski obviously allows estimating the CH index where the distances are available but not the original variables
- However, it also allows the calculation to be applied to other distances than L2Squared
- See also discrepancy measure (Studer et al., 2011) which applies similar reasoning to assessing partitions of distance matrices
Duda-Hart

- See also `dudahart` for the Duda-Hart index
- Similar calculation to CH, but focuses only on the cluster to be split
Extracting medoids

- Medoids are defined as the cases nearest the centres of clusters
- Can be used as base for clustering strategies, e.g. Partitioning around Medoids
- They can be used as group examplars
- They can be accessed when working from variables or distance matrices
  - `getmedoids` identifies medoids from a group variable and distance matrix
  - `getgroup` assigns cases to their nearest medoid
Medoids from Iris data

use iris, clear
gen id = _n
cluster wards Sepal_Length Sepal_Width ///
    Petal_Length Petal_Width
cluster gen g3 = groups(3)
matrix dissim dd = Sepal_Length Sepal_Width ///
    Petal_Length Petal_Width, L2Squared
getmedoids g3, dist(dd) id(id) gen(g3m)
Iris Medoids
getgroup

> See also getgroup: opposite direction
> Given a binary variable indicating medoids and a distance matrix, returns a group membership variable

```
. getmedoids g4, dist(dd) id(id) gen(g4m)
Translating cluster membership variable g4 into medoids index variable g4m

. getgroup g4m, dist(dd) id(id) gen(newgroup)
Creating newgroup variable as groups nearer to medoids in g4m

. permtab g4 newgroup
Calculating permutations:
Kappa max: 1.0000
Permutation vector:
  1 2 3 4
```

```
  1 3 4 2 1
```
Partitioning vs agglomerative clustering

- Numerous classes of clustering algorithm exist
- Agglomerative hierarchical methods such as Ward’s are popular
- But partitioning methods such as k-means, k-medians and Partitioning Around Medoids are also popular (and fast)
- Key idea:
  - Start with Nk cluster centres (perhaps at random)
  - Group cases around centres to form clusters
  - Find true centre of new clusters, iterate until stability
- How centres are defined differentiates the algorithms
  - k-means and k-medians uses cluster geometric centre
  - PAM uses the medoid, i.e., case closest to centre

Numerous classes of clustering algorithm exist. Agglomerative hierarchical methods such as Ward’s are popular. But partitioning methods such as k-means, k-medians and Partitioning Around Medoids are also popular (and fast). The key idea is to start with Nk cluster centres (perhaps at random), group cases around centres to form clusters, find the true centre of new clusters, and iterate until stability. How centres are defined differentiates the algorithms: k-means and k-medians use the cluster geometric centre, while PAM uses the medoid, i.e., the case closest to the centre.
Partition around medoids

- Stata provides k-means and k-medians for partition-clustering from variables
- When using pairwise distances, Partitioning Around Medoids (PAM) is possible:
  - select random cases (n=NK) as seeds, medoids
  - partition around medoids
  - define clusters wrt nearest medoid
  - for each cluster find a better medoid candidate
  - iterate until stable
- Described in Kaufman and Rousseeuw (2008)
Simulated data: 4 bivariate normal clusters
Analyst wishes to recover unknown clusters
Pick four cases at random as medoids
Create groups around initial medoids, iter 1
Find cases closer to each group centre, iter 1
2: New groups from revised medoids from iter 1
2: Revise medoids based on new groups
3: New groups from revised medoids from iter 2
3: Revise medoids based on new groups
4: New groups from revised medoids from iter 3
4: Revise medoids based on new groups
5: New groups from revised medoids from iter 5
Revised medoids are unchanged: PAM solution
Best possible partition
Cluster Analysis Utilities for Stata

Brendan Halpin, Dept of Sociology, University of Limerick

Extending Stata Clustering
Comparing solutions: ari and permtab
Visualisations
- Silhouette
- Distance matrix heatmap
Cluster stopping rules
- Calinski
- Duda-Hart
Partitioning around Medoids
- Extracting medoids
- PAM for distance matrices
- PAM Step by Step
- clpam
Fuzzy clustering
Accessing References

PAM

- Provided in clpam.ado

use iris, clear
gen id = _n
matrix dissim dd = Sepal_Length Sepal_Width ///
  Petal_Length Petal_Width, L2Squared
clpam k3, dist(dd) id(id) medoids(3) many
tab Species k3
### clpam output

```
. clpam k3, dist(dd) id(id) medoids(3) many
Random starting medoids (Nk=3)
(data already sorted by id)
Trying multiple starting points
. tab Species k3
```

<table>
<thead>
<tr>
<th>Species</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>setosa</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>versicolor</td>
<td>0</td>
<td>48</td>
<td>2</td>
<td>50</td>
</tr>
<tr>
<td>virginica</td>
<td>0</td>
<td>14</td>
<td>36</td>
<td>50</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>50</td>
<td>62</td>
<td>38</td>
<td>150</td>
</tr>
</tbody>
</table>
PAM options

- PAM results can depend strongly on the initial medoids
- Useful to initialise them, e.g., from a traditional cluster analysis
- Option `many` selects the best result from 100 random initialisations
- Option `ga` uses a genetic algorithm to search for a global optimum
Fuzzy clustering

- Fuzzy clustering allows objects to be members of multiple clusters, with varying strengths of attachment
- This gives the clustering algorithm extra degrees of freedom
- Can be more effective with noisy data
FCMdd algorithm

- clfuzz implements the fuzzy C-medoids clustering algorithm (FCMdd) (Bezdek, 1981; Krishnapuram et al., 1999)
- Minimises the sum of weighted distances to each cluster medoid, where the weight is based on the object’s attachment to the cluster
- Returns a variable holding the strongest cluster membership and an $N \times k$ matrix of object–cluster attachment strengths
- Note this is an experimental implementation!
Fuzzy clustering on simulated data

Cluster Analysis
Utilities for Stata
Brendan Halpin,
Dept of Sociology,
University of Limerick

Extending Stata Clustering
Comparing solutions: ari and permtab
Visualisations
Silhouette
Distance matrix heatmap
Cluster stopping rules
Calinski
Duda-Hart
Partitioning around Medoids
Extracting medoids
PAM for distance matrices
PAM Step by Step
clpam
Fuzzy clustering
Accessing
References
Fuzzy Irises

```
. clfuzz f3, dist(dd) id(id) k(3)
Iter 1:  1.021e+02
Iter 2:  1.235e+02
Iter 3:  2.097e+02
Iter 4:  37.8513782
Iter 5:  33.4751293
Iter 6:  30.8313277
Iter 7:  30.5336924
Medoids history
   1    2    3

   1  77  97  139
   2  65  75   79
   3  79  98   99
   4  24  92  98
   5   8  64  128
   6   8  64  148
   7   8  79  148
   8   8  79  148
```

```
. tab Species f3

<table>
<thead>
<tr>
<th>Species</th>
<th>f3</th>
<th></th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>setosa</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>versicolor</td>
<td>0</td>
<td>45</td>
<td>5</td>
</tr>
<tr>
<td>virginica</td>
<td>0</td>
<td>9</td>
<td>41</td>
</tr>
<tr>
<td>Total</td>
<td>50</td>
<td>54</td>
<td>46</td>
</tr>
</tbody>
</table>
```
Accessing slides and code

- **Slides:**
  
  http://teaching.sociology.ul.ie/sugparis

- **Code:**
  
  - ari & permtab are part of SADI:
    
    - ssc describe sadi or
    
    - net from http://teaching.sociology.ul.ie/sadi
    
    - net describe sadi

  - calinski, dudahart and discrepancy are on SSC

  - silhouette is on SSC

  - dhm, getmedoids, getgroup, clpam and clfuzz are part of package CLUTILS

    - net from
      
      http://teaching.sociology.ul.ie/statacode

    - net describe clutils

- **Contact:** brendan.halpin@ul.ie
References


