University of Limerick
Department of Sociology Working Paper Series

Working Paper WP2016-01
July 2016

Brendan Halpin

Department of Sociology, University of Limerick

Cluster Analysis Stopping Rules in Stata

Available at `http://ulsites.ul.ie/sociology/sites/default/files/wp2016-01.pdf`

Cluster Analysis Stopping Rules in Stata

Brendan Halpin (brendan.halpin@ul.ie)
Department of Sociology, University of Limerick

July 19, 2016*

Contents

* calinskiwp.org,v 1.2 2016/07/19 12:00:54 brendan Exp brendan

Abstract

Analysts doing cluster analysis sometimes want the data to tell them the optimum number of clusters. Common "stopping rules" use the Calinski-Harabasz pseudo-F statistic and Duda-Hart indices, which are based on squared Euclidean distances between cases. Cluster analysis operates on a pairwise matrix of distances between the objects clusters, which are usually created from the observed variables. However, approaches such as expert judgement or algorithmic pattern-recognition (as used for instance in sequence analysis) often output matrices of pairwise similarity or difference whose relationship to the observed variables is much less direct. Built-in Stata utilities allow calculation of the CH and DH indices when cluster analysis starts from variables, but not with cluster analysis that starts from a pairwise distance matrix (unless the distances are squared Euclidean distances defined on variables which are still available). In this note I present two small Stata utilities that will calculate the CH and DH statistics from the distance matrix, if the distances are squared Euclidean. If the distances have another metric, these utilities can be seen as calculating a pseudo-CH pseudo-F or pseudo-DH statistic, potentially extending their use to new applications.

1 Cluster analysis with variables and distance matrices

This paper discusses two utilities (downloadable from SSC) which extend Stata to calculate cluster stopping-rule indices from distance matrices, rather than from variables.

Cluster analysis is a descriptive and exploratory technique that attempts to group objects based on their similarity (Everitt et al., 2011). It operates on the matrix of pairwise distances (or dissimilarities) between the objects to be clustered. This is usually calculated on the fly by the cluster software, using observed variables to define distances between objects (depending on the cluster algorithm used, different distances are used, often but by no means always Euclidean or squared Euclidean). Stata's `cluster` command operates in this fashion. Sometimes analysts desire guidance in the number of clusters to create, and to this end there are a number of stopping-rule indices. Stata's `cluster stop` calculates some of these indices, in effect using the observed variables to make a pairwise matrix of squared Euclidean distances to do so.

However, cluster analysis does not always start from a set of observed variables. Sometimes we have pairwise similarity or dissimilarity data generated directly from other sources, such as expert judgement, machine learning or pattern recognition. A typical example in sociology is sequence analysis, which calculates dissimilarities between longitudinal series such as lifecourse histories (see e.g., Halpin, 2013, 2014a; Cornwell, 2015). When working with such approaches, we use Stata's `clustermat` command to carry out cluster analysis on the pairwise distance matrix, rather than on variables (for which we would use `cluster`).

However, Stata's built-in cluster commands do not allow calculation of stopping-rule indices when working directly from distance matrices. The existence of a `clustermat stop` command is confusing, and regularly confuses analysts. It takes a mandatory `variables()` option or subcommand, which it uses to identify a set of variables that it understands as being behind the distances, and then in effect calculates a pairwise matrix of squared Euclidean distances based on them. **That is, it *does not* use the original pairwise distances, if these come from a different source.**

2 Stopping rules

In hierarchical cluster analysis, the output of the analysis can be considered to be the dendrogram (the hierarchical tree created by initially grouping individual cases, and then groups, iteratively until you have only one group), but we often want to work with a particular cluster solution. That is, we want to cut the dendrogram at a particular level and come up with a single classification of our cases into a variable with a certain number of categories. We may do this pragmatically, selecting a grouping that "works" for our analysis, but we may sometimes want to select a "best" solution, one suggested by the data. "How many latent classes are there really?" is the question we might be thought to ask in that context.[1]

2.1 Calinski-Harabasz and Duda-Hart

There are a number of ways we can ask the data to tell us where to stop. These often start at one cluster, and ask if splitting it to make two will improve some measure of fit, some loss function, and continue, looking at each new solution in turn. The Calinski-Harabasz pseudo-F is one such measure. This involves looking at the sum of squared distances within the partitions, and comparing it to that in the unpartitioned data, taking account of the number of clusters and number of cases (Caliński and Harabasz, 1974). The Duda-Hart index does the same calculation, comparing the sum of squares in the next pair of clusters to be combined, before and after combining (Duda et al., 2000). The Duda-Hart index itself is simply the sum of the sum of squares in the two clusters, divided by the sum of squares in the combined cluster, but there is also a Duda-Hart T-squared statistic, which takes account of the number of cases (see Milligan and Cooper (1985) and Everitt et al. (2011) for discussion of cluster stopping rules). It is interesting to note that the CH pseudo-F for two clusters coincides with the DH T-squared for one, since they are making the same comparison, that is, one cluster (i.e., all the cases) versus two (the F distribution with df_1=1 and df_2=ν is equal to the square

[1] Sometimes, there in no clear structure of latent classes, which raises its own problems.

of the t-distribution with df=ν^2). For later comparisons, CH compares N clusters with one, using the whole data set, whereas DH compares 2 with 1 looking only at cases in the cluster that is to be split and its two subclusters.

2.2 How Stata calculates these measures

When it has access to the underlying variables, Stata calculates the CH index for each cluster solution (by default from 2 to 15 clusters) by regressing each variable on the cluster solution (i.e., carrying out an ANOVA) and cumulating the model sum of squares and residual sum of squares, to generate the pseudo-F statistic as follows:

$$pF = \frac{\sum MSS/(g-1)}{\sum RSS/(N-g)} \tag{1}$$

where N is the number of cases and g the number of groups.

The correspondence between ANOVA and summed squared distances within clusters arises because the sum of squared distances between the cases is directly proportional to the variable-wise sum of squares about the mean, and thus also to the sum of squared distances to the centre of the partition or cluster (see Appendix for a demonstration, and sec 4.1 of Studer et al. (2011) for fuller discussion).

Note that whatever linkage or algorithm is used to create the cluster solution, this involves assessing the fit in terms of squared Euclidean distances.

The DH statistic and T-squared are estimated in a similar fashion from the variables.

3 Calculating CH from the distance matrix

Clustering usually starts with variables, and creates a pairwise distance matrix from them. However, sometimes the distance matrix is generated or acquired independently, and in Stata we can do cluster analysis with the matrix directly, using the `clustermat` suite of commands.

While the suite includes a `clustermat stop` command, this will not work correctly for analysts working with an independently derived pairwise data matrix. However, it is straightforward to use the matrix to carry out a strictly equivalent operation, calculating the sum of squared distances within each partition and calculating a pseudo-F in the manner of the `discrepancy` measure (Studer et al., 2011):

$$pF = \frac{(SSt - \sum SSg)/(g-1)}{(\sum SSg)/(N-g)} \tag{2}$$

[2] See for instance, https://en.wikipedia.org/wiki/Student's_t-distribution#Relation_to_F-distribution.

where *SSt* is the summed squared distance within the whole matrix, and the *SSg*-s are summed squared distances within each partition.

The `calinski` and `dudahart` utilities described in this paper work directly on the distance matrix in this fashion. They are both available on SSC (see appendix B).

4 Demonstrating `calinski` and `dudahart`

Here I demonstrate that (i) `calinski` and `dudahart` produce the same results as Stata's built-in `cluster stop` when working on squared Euclidean distances calculated on variables, and (ii) that Stata's `clustermat stop` produces incorrect results when clustering is based on an external distance matrix.

4.1 Agreement between `cluster stop` and `calinski` and `dudahart`

I demonstrate the equivalence first on the NLSW88 data extract that comes with Stata. This carries out a simple cluster analysis based on four variables, and runs `cluster stop`, and then generates a squared-Euclidean matrix based on the same variables, and shows that `calinski` and `dudahart` return the same results.

```
set matsize 2500
sysuse nlsw88
keep age grade wage ttl_exp

// Keep complete cases only
foreach var of varlist age-ttl_exp {
  keep if !missing(‘var’)
}

// Generate and sort by an ID variable for calinski/dudahart commands
gen id = _n
sort id

// Carry out a conventional cluster analysis
cluster wards age-ttl_exp

// Create a matrix of squared Euclidean distances between the variables
matrix dissimilarity pwd2 = age-ttl_exp, L2squared
```

```
// compare builtin with add-on stopping rules
cluster stop
calinski, dist(pwd2) id(id)
cluster stop, rule(duda)
dudahart, dist(pwd2) id(id)
```

The Calinski-Harabasz results are shown in panel A of table 1, and show strict equivalence. The DH results are shown in panel B, and also match perfectly. Thus, the built-in Stata stopping rules for cluster analysis from variables match perfectly with the `calinski` and `dudahart` commands run on the matrix of pairwise squared-Euclidean distances between the variables, created independently by the `matrix dissim` command.

4.2 Disagreement with `clustermat stop`

Where the pairwise distance matrix comes from another source, for instance an algorithmic measurement or expert judgement of similarity or dissimilarity between objects, Stata's `clustermat stop` approach is, contrary to appearances, not valid. The `clustermat stop` command demands a `variables()` option, and Stata calculates the indices based on squared Euclidean distances defined by the variables that the option specifies, not the distance matrix used by `clustermat`. However, the original distance matrix does define the clusters used in the calculations. That is, `clustermat stop` uses one distance matrix to carry out the clustering and another to assess its fit. Many analysts are caught out by this; in fact, I don't think I have ever seen `clustermat stop` used correctly, as it would be very unusual to use `clustermat` instead of `cluster` when the relevant variables are available.

Here I use an example from sequence analysis, where distances between cases are determined algorithmically, treating a set of variables as representing sequences and using an edit distance to compare them. This edit distance (Optimal Matching distance) is often treated as squared Euclidean (Studer and Ritschard, 2015) but is not calculated as the squared Euclidean distance between the sequence state variables (it may be very different).[3] The `oma` command is provided by the SADI package, available on SSC and described in Halpin (2014a).

```
// First set up and run OM
// Substitution cost matrix
matrix sm1 = (0,1,1,2,1,3 \ ///
      1,0,1,2,1,3 \ ///
```

[3]The reference describes OM distances as non-Euclidean, but in private communication Studer describes them as "closer to squared Euclidean than Euclidean".

Table 1: Calinski-Harabasz and Duda-Hart results from `cluster stop` compared with `calinski` and `dudahart` using squared distances

(a) Calinski-Harabasz

```
. cluster stop
+----------------------------+
|              |  Calinski/  |
|  Number of   |  Harabasz   |
|  clusters    |  pseudo-F   |
|--------------+-------------|
|       2      |    734.76   |
|       3      |    955.45   |
|       4      |    883.67   |
|       5      |    844.84   |
|       6      |    841.31   |
|       7      |    777.69   |
|       8      |    730.60   |
|       9      |    697.26   |
|      10      |    674.10   |
|      11      |    652.76   |
|      12      |    625.06   |
|      13      |    603.08   |
|      14      |    581.12   |
|      15      |    563.83   |
+----------------------------+
```

```
. calinski, dist(pwd2) id(id)
+---------------------------------+
|  Number of   | Calinski-Harabasz |
|  clusters    |     pseudo-F      |
|--------------+-------------------|
|       2      |        734.76     |
|       3      |        955.45     |
|       4      |        883.67     |
|       5      |        844.84     |
|       6      |        841.31     |
|       7      |        777.69     |
|       8      |        730.60     |
|       9      |        697.26     |
|      10      |        674.10     |
|      11      |        652.76     |
|      12      |        625.06     |
|      13      |        603.08     |
|      14      |        581.12     |
|      15      |        563.83     |
+---------------------------------+
```

(b) Duda-Hart

```
. cluster stop, rule(duda)
+-------------------------------------+
|           |         Duda/Hart       |
| Number of |             |  pseudo   |
| clusters  | Je(2)/Je(1) |  T-squared |
|-----------+-------------+-----------|
|     1     |   0.7532    |   734.76  |
|     2     |   0.6183    |   795.86  |
|     3     |   0.7566    |   401.43  |
|     4     |   0.6936    |   420.06  |
|     5     |   0.7219    |   288.57  |
|     6     |   0.6682    |   246.82  |
|     7     |   0.7422    |   160.85  |
|     8     |   0.6852    |   223.26  |
|     9     |   0.8346    |   125.64  |
|    10     |   0.7957    |   117.62  |
|    11     |   0.7311    |   105.95  |
|    12     |   0.6190    |   112.62  |
|    13     |   0.7465    |   104.57  |
|    14     |   0.7484    |    37.99  |
|    15     |   0.7523    |    82.98  |
+-------------------------------------+
```

```
. dudahart, dist(pwd2) id(id)
+-------------------------------------+
|           |   Duda/Hart on distances |
| Number of |             |  pseudo    |
| clusters  | Je(2)/Je(1) |  T-squared |
|-----------+-------------+------------|
|     1     |     0.7532  |     734.76 |
|     2     |     0.6183  |     795.86 |
|     3     |     0.7566  |     401.43 |
|     4     |     0.6936  |     420.06 |
|     5     |     0.7219  |     288.57 |
|     6     |     0.6682  |     246.82 |
|     7     |     0.7422  |     160.85 |
|     8     |     0.6852  |     223.26 |
|     9     |     0.8346  |     125.64 |
|    10     |     0.7957  |     117.62 |
|    11     |     0.7311  |     105.95 |
|    12     |     0.6190  |     112.62 |
|    13     |     0.7465  |     104.57 |
|    14     |     0.7484  |      37.99 |
|    15     |     0.7523  |      82.98 |
+-------------------------------------+
```

```
      1,1,0,2,1,2 \ ///
      2,2,2,0,1,1 \ ///
      1,1,1,1,0,2 \ ///
      3,3,2,1,2,0 )
use mvad
sort id
oma state1-state72, subs(sm1) indel(1.5) pwd(omd) len(72)

// Generate clustering
clustermat wards omd, add

// Generate matrix of squared Euclidean distances based on variables
matrix dissim dd2 = state1-state72, L2squared

// Compare calinski.ado and clustermat stop
calinski, dist(omd) id(id)
clustermat stop, variables(state1-state72)
calinski, dist(dd2) id(id)

// Compare dudastop.ado and clustermat stop, rule(duda)
dudahart, dist(omd) id(id)
clustermat stop, variables(state1-state72) rule(duda)
dudahart, dist(dd2) id(id)
```

The indices are calculated first on the optimal matching distances, then using `clustermat stop`, then on the squared Euclidean distances calculated using `matrix dissim` from the variables. Table 2 shows the results using `calinski` on the correct distance matrix, then naively using `clustermat stop` and finally using `calinski` on the incorrect distance matrix, generated from the variables. The `clustermat stop` results coincide exactly with those using `calinski` in the incorrect distance matrix.

We see exactly the same pattern with the Duda-Hart indices (table 3).

From this is should be clear that when using a distance matrix generated otherwise than as squared Euclidean distances based on variables, `clustermat stop` should not be used.

Table 2: Calculating the CH index using (a) `calinski` on the correct distance matrix, (b) `clustermat stop` and (c)) `calinski` on the matrix of squared Euclidean distances between the variables

```
. calinski, dist(omd) id(id)
```

Number of clusters	Calinski-Harabasz pseudo-F
2	180.15
3	152.75
4	149.02
5	135.61
6	127.91
7	119.32
8	113.89
9	108.54
10	103.86
11	100.15
12	95.06
13	90.72
14	86.74
15	83.43

```
. clustermat stop,
  variables(state1-state72)
```

Number of clusters	Calinski/ Harabasz pseudo-F
2	210.03
3	141.89
4	105.67
5	94.42
6	83.93
7	75.24
8	86.85
9	95.71
10	87.37
11	79.38
12	79.40
13	73.44
14	68.70
15	64.38

```
. calinski, dist(dd2) id(id)
```

Number of clusters	Calinski-Harabasz pseudo-F
2	210.03
3	141.89
4	105.67
5	94.42
6	83.93
7	75.24
8	86.85
9	95.71
10	87.37
11	79.38
12	79.40
13	73.44
14	68.70
15	64.38

Table 3: Calculating the DH index using (a) `calinski` on the correct distance matrix, (b) `clustermat stop` and (c)) `calinski` on the matrix of squared Euclidean distances between the variables

```
. dudahart, dist(omd) id(id)
```

Number of clusters	Duda/Hart on distances Je(2)/Je(1)	pseudo T-squared
1	0.7976	180.15
2	0.7739	78.88
3	0.7619	136.88
4	0.7790	34.34
5	0.7519	67.96
6	0.7716	43.52
7	0.7505	76.48
8	0.8480	16.31
9	0.8091	33.96
10	0.7881	15.33
11	0.8366	26.76
12	0.8358	19.65
13	0.7454	12.64
14	0.8687	13.15
15	0.8383	6.37

```
. clustermat stop,
  variables(state1-state72) rule(duda)
```

Number of clusters	Duda/Hart Je(2)/Je(1)	pseudo T-squared
1	0.7717	210.03
2	0.8643	42.40
3	0.9326	31.65
4	0.8614	19.47
5	0.7882	55.36
6	0.7946	38.00
7	0.6492	124.27
8	0.7321	33.30
9	0.9077	14.65
10	0.9745	1.49
11	0.7982	34.65
12	0.9242	8.20
13	0.8129	8.52
14	0.9524	4.34
15	0.8456	6.02

```
. dudahart, dist(dd2) id(id)
```

Number of clusters	Duda/Hart on distances Je(2)/Je(1)	pseudo T-squared
1	0.7717	210.03
2	0.8643	42.40
3	0.9326	31.65
4	0.8614	19.47
5	0.7882	55.36
6	0.7946	38.00
7	0.6492	124.27
8	0.7321	33.30
9	0.9077	14.65
10	0.9745	1.49
11	0.7982	34.65
12	0.9242	8.20
13	0.8129	8.52
14	0.9524	4.34
15	0.8456	6.02

5 Do CH and DH make sense for other metrics?

The Calinski-Harabasz and Duda-Hart indices are conceived of in terms of ANOVA, so it makes most sense to use them where the cluster analysis is in terms of squared Euclidean distances. This directly suits clustering using Ward's linkage, because that is explicitly focused on minimising sums of squares (centroid and median linkage also default to squared Euclidean distances; do `help cluster linkage` in Stata for more detail). Other linkages and algorithms use other distances, for instance L1 or city-block distance. If you use Stata's `cluster stop` after such a clustering, it will use squared Euclidean distances to calculate the indices. It may make better sense to use `calinski` and `dudahart` on the appropriate distance matrix, since though their logic may formally be in ANOVA terms, in practice the sum of squared distances is related to the sum of distances to the centre of the cluster, and that makes a general sense. For instance, the CH index for g clusters is based on the sum of distances to the centres of the g clusters, compared to the sum of distances to the centre of the whole matrix, as laid out in equation 2:

$$pF = \frac{(SSt - \sum SSg)/(g-1)}{(\sum SSg)/(N-g)}$$

As mentioned above (and appendix A), if the distances are squared Euclidean, we can regard this as directly equivalent to ANOVA. However, for this to make intuitive sense, it is not necessary that Euclidean logic applies. Comparing the sum of distances in the whole data set to the cumulative sum of distances within partitions is arguably meaningful for any distance, even if not for ANOVA. Thus `calinski` and `dudahart` may be useful when conducting cluster analysis on variables, when using other distance measures.

To illustrate this, we go back to the NLSW88 example, and use a weighted average-linkage cluster analysis with L1 distances (i.e., absolute value).

```
set matsize 2500
sysuse nlsw88
keep age grade wage ttl_exp

// Keep complete cases only
foreach var of varlist age-ttl_exp {
  keep if !missing(‘var’)
}

// Generate an ID variable for calinski/dudahart commands
gen id = _n
sort id
```

Table 4: Calculating the CH index after a `waverage`-linkage clustering with L1-distances, using (a) `cluster stop` (b) `calinski` on the matrix of L1 distances between the variables and (c) `calinski` on the matrix of squared Euclidean distances between the variables

```
. cluster stop
+-------------------------+
|           |  Calinski/  |
| Number of |  Harabasz   |
| clusters  |  pseudo-F   |
|-----------+-------------|
|     2     |   795.74    |
|     3     |   682.84    |
|     4     |   481.65    |
|     5     |   364.95    |
|     6     |   367.32    |
|     7     |   396.63    |
|     8     |   356.46    |
|     9     |   443.79    |
|    10     |   396.50    |
|    11     |   363.45    |
|    12     |   359.62    |
|    13     |   332.63    |
|    14     |   371.66    |
|    15     |   364.99    |
+-------------------------+
```

```
. calinski, dist(pwd1) id(id)
+-------------------------------+
| Number of | Calinski-Harabasz |
| clusters  |     pseudo-F      |
|-----------+-------------------|
|     2     |        382.38     |
|     3     |        244.31     |
|     4     |        168.61     |
|     5     |        127.96     |
|     6     |        116.39     |
|     7     |        121.97     |
|     8     |        109.33     |
|     9     |        132.76     |
|    10     |        118.43     |
|    11     |        109.78     |
|    12     |        109.38     |
|    13     |        101.03     |
|    14     |        111.03     |
|    15     |        108.77     |
+-------------------------------+
```

```
. calinski, dist(pwd2) id(id)
+-------------------------------+
| Number of | Calinski-Harabasz |
| clusters  |     pseudo-F      |
|-----------+-------------------|
|     2     |        795.74     |
|     3     |        682.84     |
|     4     |        481.65     |
|     5     |        364.95     |
|     6     |        367.32     |
|     7     |        396.63     |
|     8     |        356.46     |
|     9     |        443.79     |
|    10     |        396.50     |
|    11     |        363.45     |
|    12     |        359.62     |
|    13     |        332.63     |
|    14     |        371.66     |
|    15     |        364.99     |
+-------------------------------+
```

```
cluster waverage age-ttl_exp, measure(L1)

// Create matrices of the variables, Euclidean and squared Euclidean
matrix dissimilarity pwd1 = age-ttl_exp, L1
matrix dissimilarity pwd2 = age-ttl_exp, L2squared

cluster stop
calinski, dist(pwd1) id(id)
calinski, dist(pwd2) id(id)
```

Table 4 shows in panel (a) the results from `cluster stop`, in panel (b) the results using `calinski` on the same distances (L1) that the clustering is carried out on, and in panel (c) the results from `calinski` using squared Euclidean distances between the variables. As can be seen, `cluster stop` evidently uses the squared Euclidean distances, while using the cluster groupings generated on L1 or absolute distances. The `calinski` results using the L1 distances are different. It does not seem intuitively reasonable to choose one metric to carry out a clustering and another to assess its quality, so it would seem preferable to use the `calinski` result on the correct distance matrix.

Note that the CH pseudo-F is exactly equal to the discrepancy pseudo-F of Studer et al. (2011). The arguments those authors make about using the discrepancy measure with dis-

tances other than squared Euclidean (in terms of association between the distance matrix and observed categorical variables) apply equally to using pseudo-F and related measures to examine the relationship between cluster solutions and the distance matrix (with the exception that the permutation-based p-values are not valid, given the cluster solution is derived from the distance matrix).

6 Should we fetishise stopping rules?

Stopping rules provide a way to justify a particular cluster solution on the basis of the data, providing guidance or an appearance of objectivity. However, depending on the problem at hand, they should be taken with a grain of salt. The notion of a clear set of latent classes might be inapplicable, or the stopping-rule-indicated cluster solution may be too detailed or not detailed enough for analytical purposes. That is, a set of pairwise distances may contain a great deal of information about relationships between objects without there being a clear pattern of groups: the distribution within the space implied by the distances may not be characterised by zones of density separated by zones of sparsity. In such cases, clustering provides potentially useful data reduction without identifying clear latent classes, but will be unstable. Indeed, stopping rules may be unhelpful, perhaps suggesting either 2 or an infeasibly large number of clusters. This is often the cases with sequence analysis of lifecourse data (Halpin, 2014b), where the space implied by the distances between sequences is highly structured but relatively evenly populated. The instability means that different settings or algorithms will produce different clusterings, but they will all to a greater-or-lesser degree reduce the extensive data in the pairwise distance matrix into an informative classification. By informative I mean that the partitions of the classification will contain objects that are mutually similar, and dissimilar to objects in other groups, in a way that is informatively associated with other variables. Where that is the case, pragmatic approaches should carry more weight than obedience to stopping rules.

7 Conclusion

The two utilities described in this paper, `calinski` and `dudahart`, replicate functionality provided by Stata's built-in `cluster stop` command, in a manner that also works for clustering based on distance matrices rather than variables. In particular, they provide functionality that is hinted at but not provided by the existing `clustermat stop` command. In addition, by allowing direct operation on the distance matrices, they allow the use of the Calinski-Harabasz and Duda-Hart rules on distances other than squared Euclidean, thus making the

rules applicable to a wider range of cluster linkages.

8 See also

A number of related utilities and packages are available from SSC:

- `silhouette`: calculates and graphs silhouette widths, illustrating the distribution of fit of cases within clusters
- `discrepancy`: Implements the discrepancy measure of Studer et al. (2011)
- SADI: a set of tools for sequence analysis in Stata described in Halpin (2014a). In particular its cluster-related utilities:
  - `permtab`: tabulate cluster solutions
  - `ari`: Adjusted Rand Index for comparing cluster solutions
  - `corrsqm`: Correlation between pairwise distance matrices

References

Caliński, T. and Harabasz, J. (1974) A dendrite method for cluster analysis, *Communications in Statistics*, **3**(1), 1–27.

Cornwell, B. (2015) *Social Sequence Analysis: Methods and Applications*. New York, Cambridge University Press.

Duda, R. O., Hart, P. E. and Stork, D. G. (2000) *Pattern Classification*, 2 edn. New York, John Wiley.

Everitt, B. S., Landau, S., Leese, M. and Stahl, D. (2011) *Cluster Analysis*, 5 edn. Chichester, Wiley.

Halpin, B. (2013) Sequence analysis, in J. Baxter (ed), *Oxford Bibliographies in Sociology*, Oxford University Press, New York.
URL: *http://www.oxfordbibliographies.com*

Halpin, B. (2014a) SADI: Sequence analysis tools for Stata, *Working Paper WP2014-03*, Dept of Sociology, University of Limerick, Ireland.

Halpin, B. (2014b) Three narratives of sequence analysis, in P. Blanchard, F. Bühlmann and J.-A. Gauthier (eds), *Advances in Sequence Analysis: Theory, Method, Applications*, Springer, Berlin.

Milligan, G. W. and Cooper, M. C. (1985) An examination of procedures for determining the number of clusters in a data set, *Psychometrika*, **50**(2), 159–179.

Studer, M. and Ritschard, G. (2015) What matters in differences between life trajectories: A comparative review of sequence dissimilarity measures, *Journal of the Royal Statistical Society: Series A (Statistics in Society)*. doi: 10.1111/rssa.12125.

Studer, M., Ritschard, G., Gabadinho, A. and Müller, N. S. (2011) Discrepancy analysis of state sequences, *Sociological Methods and Research*, **40**(3), 471–510.

A SS and distance to centre

There is a direct relationship between sum of squared deviations and the sum of distances between variables, as given by this equation:

$$SS = \sum_{i=1}^{N}(x_i - \bar{x})^2 = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=i+1}^{N}(x_i - x_j)^2 \tag{3}$$

That is, the sum of squared deviations is equal to $\frac{1}{N}$ times the sum of the squared distances in one triangle of the distance matrix (i.e., half of the symmetric matrix, containing each non-redundant distance once). This generalises to multiple variables or dimensions.

We can test this with Stata code as follows, creating a set of random variables, the case-wise distance matrix based on them (squared Euclidean), and calculating the conventional sum of squares around each variable, and showing the cumulative sum of squares is equal to $\frac{1}{N}$ times the sum of one triangle (`vech()` returns a vector containing one triangle of a square matrix) of the distance matrix.

```
set obs 100
gen x1 = rnormal(10,10)
gen x2 = rnormal(5,10)
gen x3 = rnormal(1,1)
matrix dissim dd2 = x1 x2 x3, L2squared
local rss = 0
reg x1
local rss = `rss' + e(rss)
reg x2
local rss = `rss' + e(rss)
reg x3
local rss = `rss' + e(rss)
mata: st_numscalar("ssd",sum((vech(st_matrix("dd2")))/100))
di "Sum of squared deviations:                " `rss'
di "1/N Sum of triangle of squared distances: " ssd
```

B Installation

Both `calinski` and `dudahart` are available at SSC. The following Stata commands will install them:

```
ssc install calinski
ssc install dudahart
```

C Help pages

Stata help pages for `calinski` and `dudahart` follow.

help calinski

Title

calinski —— Calinski-Harabasz cluster stopping index from distance matrix

Syntax

calinski , DISTmat(string) IDvar(varname) [NGroups(integer 15) NAME(clname) GRaph *]

options	Description
Required	
distmat(*matname***)**	names the distance matrix
idvar(*varname***)**	identifies the variable that links the sort-order of the distance matrix to the sort-order of the data
Optional	
ngroups	The number of cluster solutions to test (default 15)
name	Name of cluster analysis to use
graph	plot the index against cluster size
twoway_options	options allowed with **graph twoway**

Description

calinski calculates the Calinski-Harabasz pseudo-F for stopping rules in cluster analysis, from the pairwise distance matrix. This is widely used to determine the optimum number of clusters. Stata's default **cluster stop** does the same calculation on the basis of the original variables, but cannot operate on the distance matrix. **calinski** is thus useful when the original variables are not available, or when the distances are created other than as squared Euclidean distances between variables (as is the case for instance with sequence analysis).

NB: Stata's built-in **clustermat stop**, **variables(...)** does *not* estimate the CH pseudo-F on the distance matrix used by **clustermat**. Rather, it creates a new temporary distance matrix based on the variables listed in the **variables()** option.

calinski depends on discrepancy which can be installed from SSC:

. ssc install discrepancy

Returns:

r(calinski_#) Calinski-Harabasz pseudo-F for # groups

Remarks

While **cluster stop** and **clustermat stop** estimate the CH pseudo-F by cumulating the sum of squares from ANOVAs of the original variables on the cluster solution, and are therefore explicitly rooted in a squared-Euclidean distance point of view, **calisnki** takes the distances as they are found. If they are squared distances based on the original variables, the results will be identical to **cluster stop**. If they are squared Euclidean distances from another source, the interpretation will be the same. If they are other sorts of differences (e.g., non-Euclidean) the interpretation is not necessarily the same, but can be understood to be analogous, in the same way as the **discrepancy** partitioning of the distance matrix (described by Studer et al 2011) is analogous to ANOVA.

Because the order of the data and the order of the distance matrix must coincide, the dataset must be sorted by **idvar**. It is the user's responsibility that this variable defines the correct order.

References

Milligan, G. W., and M. C. Cooper. 1985. An examination of procedures for determining the number of clusters in a dataset. *Psychometrika* 50: 159–179.
M Studer, G Ritschard, A Gabadinho and NS Müller, Discrepancy analysis of state sequences, *Sociological Methods and Research*, 40(3):471–510

Author

Brendan Halpin, brendan.halpin@ul.ie

Examples

```
. calinski, dist(distances) id(id) graph
```

See Also

cluster stop
dudahart
discrepancy
SADI

help dudahart

Title

dudahart —— Duda-Hart cluster stopping index from distance matrix

Syntax

dudahart , DISTmat(string) IDvar(varname) [NGroups(integer 15) NAME(clname) GRaph *]

options	Description
Required	
distmat(*matname***)**	names the distance matrix
idvar(*varname***)**	identifies the variable that links the sort-order of the distance matrix to the sort-order of the data
Optional	
ngroups	The number of cluster solutions to test (default 15)
name	Name of cluster analysis to use
graph	If "both" plot the DH index and T-squared against cluster size, if "dh" the index only, if "dht" the T-squared only.
twoway_options	options allowed with **graph twoway**

Description

dudahart calculates the Duda-Hart index for stopping rules in cluster analysis, from the pairwise distance matrix. This is widely used to determine the optimum number of clusters. Stata's default **cluster stop** does the same calculation on the basis of the original variables, but cannot operate on the distance matrix. **dudahart** is thus useful when the original variables are not available, or when the distances are created other than as squared Euclidean distances between variables (as is the case for instance with sequence analysis).

NB: Stata's built-in{cmd:clustermat stop, variables(...) rule(duda)} does *not* estimate the DH index on the distance matrix used by **clustermat**. Rather, it creates a new temporary distance matrix based on the variables listed in the **variables()** option.

Returns:

r(duda_#) Duda-Hart Je(2)/Je(1) value for # groups
r(dudat2_#) Duda-Hart pseudo-T-squared value for # groups

Remarks

While **cluster stop**, **rule(duda)** and {cmd:clustermat stop, variables(...) rule(duda)} estimate the Duda-Hart index from the original variables of the cluster solution, and are therefore explicitly rooted in a squared-Euclidean distance point of view, **dudahart** takes the distances as they are found. If they are squared distances based on the original variables, the results will be identical to **cluster stop**. If they are squared Euclidean distances from another source, the interpretation will be the same. If there are other sorts of differences (e.g., non-Euclidean) the interpretation is not necessarily the same, but can be understood to be analogous, in the same way as the **discrepancy** partitioning of the distance matrix (described by Studer et al 2011) is analogous to ANOVA.

Because the order of the data and the order of the distance matrix must coincide, the dataset must be sorted by **idvar**. It is the user's responsibility that this variable defines the correct order.

References

Milligan, G. W., and M. C. Cooper. 1985. An examination of procedures for determining the number of clusters in a dataset. *Psychometrika* 50: 159-179.
M Studer, G Ritschard, A Gabadinho and NS Müller, Discrepancy analysis of state sequences, *Sociological Methods and Research*, 40(3):471-510

Author

Brendan Halpin, brendan.halpin@ul.ie

Examples

```
. dudahart, dist(distances) id(id) graph(dht)
```

See Also

cluster stop
silhouette
SADI