CCEA (Convergent Cluster & Ensemble Analysis)

Provo, Utah, USA
+1 801/477-4700
October, 2025

Convergent Cluster & Ensemble Analysis (CCEA) is Sawtooth Software's tool for discovering groups using continuous (metric) data. CCEA may be used to cluster survey respondents in market research applications (although the method is equally applicable for many other types of data). Therefore we describe the entities to be clustered as "respondents," though they need not be. For uniformity we use the term "variables" to describe the attributes on which respondents are measured. These variables should be continuous, such as rating scales. CCEA is not appropriate for clustering on categorical (nominal) data, for which other clustering methods such as latent class clustering would be more appropriate.

Cluster analysis consists of finding groups of cases (e.g. respondents) that tend to be similar *within* those groups on the basis variables (the variables used in clustering), but different on those same variables *between* the groups. Cluster ensemble analysis consists of leveraging a variety of cluster solutions (an *ensemble* of solutions) to find a single best *consensus* solution that has stronger characteristics than any one of the solutions within the ensemble.

CCEA may be considered the next generation to our previous CCA (Convergent Cluster Analysis) software system. In addition to the ensemble approach, CCEA includes the capabilities of CCA software for k-means cluster analysis. The literature argues that ensembles perform better than standard cluster analysis. Our work with CCEA v3 also supports that conclusion. Our cluster ensemble approach consistently obtains better solutions than the standard approach in CCA of selecting the highest-reproducibility solution for synthetic data sets with known cluster structure.

The quality of a cluster solution can depend importantly on the quality of the starting points. For this reason, CCEA dedicates a considerable portion of its resources generating "high quality" starting points (described later in this paper) and evaluating the reproducibility of solutions obtained from different sets of starting points. The run that is the most representative (reproducible) across multiple tries is returned as the final solution. This protects the analyst from stumbling into a poor solution based on an unlucky draw of starting points.

Because users may already have significant experience with CCA software, and because of its important historical precedent as a fine cluster approach, we include the standard CCA (Convergent Cluster Analysis) capabilities (with some algorithmic improvements over the previous CCA v2) within this software. That said, we are very enthusiastic about Ensemble Analysis, and our recent experience suggests users will obtain consistently better solutions. Users may decide to employ standard CCA cluster analysis within the package to investigate preliminary solutions, or especially to identify outliers (which is not a capability of Ensemble Analysis). After this preliminary work, we suggest applying what you have learned within CCEA's ensemble capabilities to obtain a final, even stronger, solution.

Uses of Cluster Analysis:

People seem to have a general interest in classifying things into groups. Given any large collection of things, we try to arrange them into categories. Marketers find it useful to think of their customers as "heavy users" and "light users." We think of ourselves as living in "big cities," "small towns," or "in the

country." We think of occupations as "blue collar" and "white collar," siblings as "brothers" and "sisters," and people as "children" or "adults."

In all these cases it is easier for us to think about a small number of categories instead of a large number of individuals. However, things seldom fall neatly into groups, and the simplification achieved by grouping almost always entails loss of information.

Cluster analysis is a way of using continuous (metric) basis variables for categorizing a collection of objects into groups (or "clusters"). Suppose we have a collection of objects, each of which has been described on a number of variables. The descriptions may be physical measurements or subjective ratings.

We want to organize the objects into groups so that those within each group are relatively similar and those in different groups are relatively dissimilar. Almost any division of objects into distinct subsets can accomplish this.

We also hope that the groups will be "natural," and "compelling," and will reveal structure actually present in the data. If we were to plot the clusters in a multidimensional space, we would like them to be relatively dense aggregations of points, with empty space between them. Ideally, the data should look like a collection of cantaloupes in space, not like a single watermelon. At the very least, the points toward the centers of the clusters should be more densely concentrated than the points between them.

As an example of the motivation for doing a cluster analysis, we might want to buy a new car, or advise others about what car to buy. In either case it might be useful to organize the cars into classes such as "large," "medium," "small," "luxury," "sporty," etc. One way to do this would be with cluster analysis, where the objects would be cars and the metric variables could be price, fuel economy, top speed, number of passengers, etc.

As another example, in marketing a new breakfast cereal we might suspect that potential users of our product would have a wide range of attitudes toward such benefits as convenience, nutrition, and economy. We might find it helpful to have a sample of them describe their attitudes by expressing agreement or disagreement with a number of statements on a metric scale. We might perform a cluster analysis of those data to see if the individuals fell naturally into groups of potential customers desiring different benefits, each of which would be approached differently.

Underlying any such use of cluster analysis, there are several issues that must be addressed:

1. What measure of "similarity" among objects will be used?

2. What method will be used to assign objects to groups?

3. How will we decide how many groups to consider?

4. How can we be sure that the grouping obtained is "natural," reproducible, and useful?

In designing the first CCA systems and this later CCEA System, several requirements affected our thinking about these issues:

1. We wanted CCEA to be usable in conjunction with other Sawtooth Software products, which find application in market research and the social sciences. In both fields the objects studied are often

people who have responded to surveys. Those projects usually involve large data sets, consisting of hundreds and perhaps thousands of respondents.

2. Although it is always desirable for researchers to be expert in the techniques they are using, that is not always true. Therefore, we wanted to choose techniques most likely to "work every time." We considered it essential to use methods most likely to yield reproducible solutions, and also to give the user an indication of the reproducibility of each solution.

3. We know from our own experience and from the literature that it is not possible to fully "automate" the process of deciding which of several alternative clusterings to accept. However, there are statistical indicators which, although imperfect, can be helpful in this regard. Consequently, we have provided tables of "norms" to aid in such decisions in the appendices of the CCEA software's manual. These are based on approximately 20,000 "Monte Carlo" clusterings.

The literature on cluster analysis is voluminous. In earlier CCA manuals, we recommended an excellent review article (Milligan, G. W. and Cooper, M. C. "Methodology Review: Clustering Methods" in Applied Psychological Measurement, Vol II, No. 4, Dec 87) that provides an exceptionally rich source of information on the general topic of "what works."

However, many new developments have occurred since the 1980s. We are grateful to Joe Retzer and Ming Shan of Maritz Research for calling our attention to cluster ensemble methods in their presentation at the 2007 Sawtooth Software Conference entitled, "Cluster Ensemble Analysis and Graphical Depiction of Cluster Partitions." Retzer and Shan refer to an article by Strehl and Ghosh that has provided ideas that have been especially helpful in developing our unique cluster ensemble approach (Strehl and Ghosh, 2002, "Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions," Journal of Machine Learning Research 3 (2002) 583-617. Available for download at www.strehl.com/download/strehl-jmlr02.pdf.)

Appropriate Data for CCEA

CCEA works best when the variables you are using for finding clusters are continuous. Examples of continuous variables include: age, income, rating scales, Likert scales, importance scores. The variables should all have similar variance. If they do not, you should standardize them (giving them equal means and variances), by selecting the standardization option within the software. As an example, if one of your variables is respondent's weight (in kilos) and the other is annual household income (typically in the tens of thousands), unless standardized, the income variable will have many times greater impact on the solution than weight. The process of standardization is quite simple for a variable: first, we compute the mean and the standard deviation. Next we subtract the mean from each value (thus centering the values), and then we divide each of those values by the standard deviation. This results in values that average zero and have a standard deviation of 1.0.

Categorical variables are generally not appropriate for use within CCEA. For example, a variable such as "preferred_color" where 1=blue, 2=red, 3=green, etc. would not be appropriate for use in CCEA. CCEA expects increasing values to indicate "more" of the variable and decreasing values to mean "less." Such is not the case with categorical variables.

Although one can cluster on individual-level utilities resulting from an HB analysis of Choice-Based Conjoint or MaxDiff (best-worst scaling), it is probably more appropriate to utilize Latent Class MNL procedures for these cases. Using CCEA would involve the two-stage procedure of first computing

utilities using HB, and then secondly using those data within CCEA (where any errors in the first stage would be accepted as "truth" in the second phase). Latent Class MNL provides a way to simultaneously estimate part-worth utilities and divide the sample into meaningful segments.

Note: CCEA's clustering algorithms cannot handle missing data. You can include variables in the data file that have missing values. But, variables selected for use in clustering must not have any missing values.

Standard K-Means Run

CCEA first returns a brief summary of the data file:

```
Data File: C:\Users\Bryan.SAWTOOTH\Documents\Sawtooth Software\CCEA Samples\car.csv
Variables are standardized.
Number of cases: 62
Number of variables per case: 9
The file contains these variables:
      1. Price
      2. Acceleration time 0-60
      3. 1/4 elapsed time
      4. Top speed
      5. Braking feet from 80 mph
      6. Slalom speed
      7. Skidpad g factor
      8. Interior noise db
      9. Fuel mpg

Variables included in this computation:
      2. Acceleration time 0-60
      3. 1/4 elapsed time
      4. Top speed
      5. Braking feet from 80 mph
      6. Slalom speed
      7. Skidpad g factor
      8. Interior noise db
      9. Fuel mpg
```

The procedure is as follows for CCEA's k-means cluster analysis:

Each solution proceeds automatically with these steps:

1. A set of "starting points" is determined. There are as many starting points as clusters desired.

2. Each respondent is classified into a group corresponding to the starting point to which he is most similar.

3. The averages for each variable are computed for the respondents in each group. These averages replace the starting points.

4. Steps 2 and 3 are repeated until no respondents are reclassified from the previous iteration.

The quality of the solution can depend importantly on the quality of the starting points. For this reason, CCEA dedicates a considerable portion of its resources generating "high quality" starting points (described in the section later in this article entitled *Starting Points*) and evaluating the reproducibility of solutions obtained from different sets of starting points.

The output is quite voluminous, so we'll break it down into sections and describe each part.

First, a summary of the settings for your run is displayed:

```
CCEA Computation (12/13/2007 12:50:41 PM)
=================================================

Minimum number of groups                      2
Maximum number of groups                      5
Number of replications                       30
Maximum number of iterations               1000
Minimum group size                            1
Including all cases (no outliers)
Random number seed                            1
Using a mix of starting point methods.
```

Next, a summary of the replications is displayed for the two-group solution, showing which starting point methods were used, how many iterations it took to converge in each case, and how many cases were in each of the two groups.

```
Solution for 2 Groups
Replication   Start          Iterations   Group sizes
-------------------------------------------------------------------------------
      1       Distance            2         6    56
      2       Hierarchical        2        56     6
      3       Density             6        30    32
      4       Distance            9        42    20
      5       Hierarchical        2         6    56
      6       Density            12        42    20
      7       Distance            3         6    56
      8       Hierarchical        2        55     7
      9       Density             6        40    22
     10       Distance            3         6    56
     11       Hierarchical        2        55     7
     12       Density             9        30    32
     13       Distance            6        30    32
     14       Hierarchical        7        30    32
     15       Density             9        30    32
     16       Distance            5        16    46
     17       Hierarchical        2        55     7
     18       Density             9        30    32
     19       Distance            5        16    46
     20       Hierarchical        5        22    40
     21       Density             6        33    29
     22       Distance            2        56     6
     23       Hierarchical        2        55     7
     24       Density             6        40    22
     25       Distance           13        20    42
     26       Hierarchical        2        53     9
     27       Density             9        30    32
     28       Distance            5        46    16
     29       Hierarchical        2         6    56
     30       Density             6        40    22
```

The first replication used the Distance-based starting point strategy. It took only 2 iterations for the solution to converge, and the two groups found contained 6 and 56 cases. It is evident from this table that there is quite a bit of variability among the 30 replicates of the 2-group solution. It would not appear that the data naturally cluster in a very stable way into 2 groups.

The next output table contains the reproducibilities for all replicates, displaying how consistently cases were assigned to groups when the procedure was repeated from different starting points. (We've only shown a portion of the table, as 30x30 replicates is too wide to display easily in this documentation.)

```
Pairwise Reproducibility of Replicates (2 Groups)
    |    1     2     3     4     5     6     7     8     9    10    11    12  ... 30
----+-------------------------------------------------------------------------  ----
  1 | ----- 100.0  22.6  54.8 100.0  54.8 100.0  96.8   9.7 100.0  96.8  22.6 ...  9.7
  2 | 100.0 -----  22.6  54.8 100.0  54.8 100.0  96.8   9.7 100.0  96.8  22.6 ...  9.7
  3 |  22.6  22.6 -----  67.7  22.6  67.7  22.6  25.8  67.7  22.6  25.8 100.0 ... 67.7
  4 |  54.8  54.8  67.7 -----  54.8 100.0  54.8  58.1  35.5  54.8  58.1  67.7 ... 35.5
  5 | 100.0 100.0  22.6  54.8 -----  54.8 100.0  96.8   9.7 100.0  96.8  22.6 ...  9.7
  6 |  54.8  54.8  67.7 100.0  54.8 -----  54.8  58.1  35.5  54.8  58.1  67.7 ... 35.5
  7 | 100.0 100.0  22.6  54.8 100.0  54.8 -----  96.8   9.7 100.0  96.8  22.6 ...  9.7
  8 |  96.8  96.8  25.8  58.1  96.8  58.1  96.8 -----   6.5  96.8 100.0  25.8 ...  6.5
  9 |   9.7   9.7  67.7  35.5   9.7  35.5   9.7   6.5 -----   9.7   6.5  67.7 ...100.0
 10 | 100.0 100.0  22.6  54.8 100.0  54.8 100.0  96.8   9.7 -----  96.8  22.6 ...  9.7
 11 |  96.8  96.8  25.8  58.1  96.8  58.1  96.8 100.0   6.5  96.8 -----  25.8 ...  6.5
 12 |  22.6  22.6 100.0  67.7  22.6  67.7  22.6  25.8  67.7  22.6  25.8 ----- ... 67.7
  .     .     .     .     .     .     .     .     .     .     .     .     ...  .
  .     .     .     .     .     .     .     .     .     .     .     .     ...  .
  .     .     .     .     .     .     .     .     .     .     .     .     ...  .
 30 |   9.7   9.7  67.7  35.5   9.7  35.5   9.7   6.5 100.0   9.7   6.5  67.7 ... -----

Avg |  57.1  57.1  55.7  62.6  57.1  62.6  57.1  58.0  38.6  57.1  58.0  55.7 ... 38.6

Replication 6 (density start) has the best reproducibility (62.6%)
The elapsed time for this solution was 0:00:00.
```

The average reproducibility for each replication is displayed and the replication with the best reproducibility is noted (ties are broken randomly). In this example, replicates 1, 2, 5, 7, and 10 were identical. But the cluster solution reflected by these replicates seems to be a less representative solution. Across all replicates, replication #6 seems to be the most reproducible, with an average adjusted reproducibility of 62.6%. It is taken as the "best" two-group solution, and the cluster membership for each case is saved to the **car_membership.csv** file.

Next, the cluster means and F ratios are computed and displayed. (We chose to standardize variables so the means are centered around zero):

```
Group Means and F Ratios
                                    1          2        --F--
-------------------------------------------------------------
   1. Acceleration time 0-      -0.48       1.01       58.52
   2. 1/4 elapsed time          -0.48       1.02       60.01
   3. Top speed                  0.45      -0.95       46.23
   4. Braking feet from 80      -0.15       0.32        3.16
   5. Slalom speed               0.17      -0.36        3.97
   6. Skidpad g factor           0.38      -0.81       27.48
   7. Interior noise db          0.12      -0.26        1.96
   8. Fuel mpg                  -0.40       0.85       31.96

Group Size                         42         20       23.25
```

The number of cases in each cluster is displayed (Cluster 1 has 42 cases, Cluster 2 has 20). Using the first line as an example, Acceleration time has a mean for Cluster 1 of -0.48, a mean for Cluster 2 of 1.01, and an F ratio of 58.52. The F ratios indicate the relative amount of difference among clusters for each of the variables. F ratios are obtained by dividing a "mean square between clusters" by a "mean square within clusters."

At the bottom of the last column, in the line designated "Group Size," is a "pooled" F ratio (23.25). The pooled F ratio is obtained by summing the numerators and denominators for the individual F ratios separately, and then dividing the sum of the numerators by the sum of the denominators. The pooled F ratio is an overall indicator of the amount of difference between clusters. Comparing these values for different cluster solutions may be helpful in deciding how many clusters to use.

These F ratios are provided as descriptive statistics, not as values that should be tested for statistical significance. Since the clusters were constructed to be as different from one another as possible on these variables, it would be inappropriate to test whether differences on these same variables are greater than would be expected "due to chance alone." These F ratios would almost certainly appear "highly significant" if tested, even if the data had consisted of nothing but random numbers. However, the F ratios are valuable as descriptive measures of the relative importance of each variable in the clustering. Those variables with the largest F ratios are those on which the clusters are most different. Variables with the smallest F ratios could probably have been omitted without much effect on the cluster analysis results.

Then, the means as deviations from grand means are computed and displayed:

Group Means as Deviations from Grand Means and F Ratios

	1	2	--F--
1. Acceleration time 0-	-0.48	1.01	58.52
2. 1/4 elapsed time	-0.48	1.02	60.01
3. Top speed	0.45	-0.95	46.23
4. Braking feet from 80	-0.15	0.32	3.16
5. Slalom speed	0.17	-0.36	3.97
6. Skidpad g factor	0.38	-0.81	27.48
7. Interior noise db	0.12	-0.26	1.96
8. Fuel mpg	-0.40	0.85	31.96
Group Size	42	20	23.25

(When variables are standardized, this screen is always identical to the previous screen.)

Then, for each cluster, the variables are sorted by the cluster's means expressed as deviations from grand means. This display lets you quickly see on which variables this cluster has a higher (more positive) or lower (more negative) value than other clusters. For each variable, the number of any other cluster with a more extreme deviation with the same sign is also noted.

For illustration, we'll jump ahead in the output and display the three-group solution result from this same run. The three-group solution found group sizes of 18, 38, and 6 cases:

Group Means as Deviations from Grand Means and F Ratios

	1	2	3	--F--
1. Acceleration time 0-	-0.91	0.10	2.14	69.43
2. 1/4 elapsed time	-0.98	0.16	1.96	62.49
3. Top speed	0.72	-0.08	-1.66	21.98
4. Braking feet from 80	-0.28	-0.11	1.56	11.05
5. Slalom speed	0.61	-0.11	-1.14	9.49
6. Skidpad g factor	0.96	-0.28	-1.10	22.95
7. Interior noise db	0.92	-0.40	-0.22	16.29
8. Fuel mpg	-0.41	-0.04	1.51	11.27
Group Size	18	38	6	21.73

```
Group 1 (18 cases) sorted by Deviations from Grand Means
Variable                        Mean        Dev    More Extreme (w/ same sign)
   6. Skidpad g factor          0.96       0.96
   7. Interior noise db         0.92       0.92
   3. Top speed                 0.72       0.72
   5. Slalom speed              0.61       0.61
   4. Braking feet from 80     -0.28      -0.28
   8. Fuel mpg                 -0.41      -0.41
   1. Acceleration time 0-     -0.91      -0.91
   2. 1/4 elapsed time         -0.98      -0.98

Group 2 (38 cases) sorted by Deviations from Grand Means
Variable                        Mean        Dev    More Extreme (w/ same sign)
   2. 1/4 elapsed time          0.16       0.16    3
   1. Acceleration time 0-      0.10       0.10    3
   8. Fuel mpg                 -0.04      -0.04    1
   3. Top speed                -0.08      -0.08    3
   5. Slalom speed             -0.11      -0.11    3
   4. Braking feet from 80     -0.11      -0.11    1
   6. Skidpad g factor         -0.28      -0.28    3
   7. Interior noise db        -0.40      -0.40

Group 3 (6 cases) sorted by Deviations from Grand Means
Variable                        Mean        Dev    More Extreme (w/ same sign)
   1. Acceleration time 0-      2.14       2.14
   2. 1/4 elapsed time          1.96       1.96
   4. Braking feet from 80      1.56       1.56
   8. Fuel mpg                  1.51       1.51
   7. Interior noise db        -0.22      -0.22    2
   6. Skidpad g factor         -1.10      -1.10
   5. Slalom speed             -1.14      -1.14
   3. Top speed                -1.66      -1.66
```

Groups 1 and 3 seem to be more extreme on nearly every variable than Group 2, which has a more central position. Note that Group 1 is characterized as having the highest positive deviation from the grand mean on Skidpad g factor (deviation +0.96), followed by Interior noise (deviation +0.92), Top speed (deviation +0.72), and Slalom speed (deviation +0.61). No other group shows a more extreme deviation from the mean in the same (positive) direction as group 1 on these four variables. Similarly, no other group is as extreme in the negative direction on the next four variables, on which Group 1 is positioned below the grand mean.

After the last cluster solution is printed (the 5-group solution in our case), each pair of solutions is automatically cross-tabulated. For example, the tabulation of adjacent pairs of solutions is as follows:

```
Tabulation of 2 group vs. 3 group solutions

          0     1     2     3   Total
       ----  ----  ----  ----    ----
  0 |     0     0     0     0 |     0
  1 |     0    18    24     0 |    42
  2 |     0     0    14     6 |    20
       ----  ----  ----  ----    ----
Total     0    18    38     6 |    62
```

Tabulation of 3 group vs. 4 group solutions

	0	1	2	3	4	Total
0	0	0	0	0	0	0
1	0	0	1	13	4	18
2	0	0	19	0	19	38
3	0	6	0	0	0	6
Total	0	6	20	13	23	62

Tabulation of 4 group vs. 5 group solutions

	0	1	2	3	4	5	Total
0	0	0	0	0	0	0	0
1	0	0	0	0	3	3	6
2	0	5	0	15	0	0	20
3	0	0	12	1	0	0	13
4	0	23	0	0	0	0	23
Total	0	28	12	16	3	3	62

In the tabulation of the two- vs. the three-group solution, the first row and column of the table reflects group 0 (any outliers). No cases were regarded as outliers since we didn't permit outliers--we had checked *Include all cases (no outliers)* within the software.

The 42 cases in group 1 of the 2-group solution were split between groups 1 and 2 of the 3-group solution. The 20 cases in group 2 of the 2-group solution were split into groups 2 and 3 of the 3-group solution.

This example was purely for illustrative purposes. Your data sets will typically have many more cases. In social sciences and marketing research, you usually wouldn't want to draw conclusions regarding groups containing just a few cases. Even so, the question arises whether there seems to be reproducible and natural cluster structure in this car data set.

Below, we've compared the adjusted reproducibility we achieved with this data run to that which could be observed using random data constructed with no cluster structure (as published in Table 1 in the section entitled *Reproducibility Norms* of the full CCEA Manual):

Adjusted Reproducibility

2-group solution:
Cars Data 63%
Random Data 58%

3-group solution:
Cars Data 79%
Random Data 46%

4-group solution:
Cars Data 82%
Random Data 43%

5-group solution:
Cars Data 68%
Random Data 47%

For each cluster solution, the adjusted reproducibility achieved with the cars data set exceeded that expected from a data set (using 10 basis variables) with no group structure. While this certainly is good news, it is a very low standard of achievement.

The 3- and 4-group solutions have in absolute terms the highest reproducibility, and they also display the largest gap between observed reproducibility and the benchmark based on no group structure. The 2-group solution seems to be a poor characterization of these data, barely exceeding the "no structure" threshold in terms of adjusted reproducibility.

Cluster Ensemble Analysis

Cluster Ensemble approaches (Strehl and Ghosh 2002, Retzer and Shan 2007, Orme and Johnson 2008) employ multiple cluster solutions as well, but rather than choose the *one* most representative solution, they develop a consensus solution based on a combination of the solutions available within the ensemble. The final solution is almost always different from all of the solutions in the ensemble. Ensemble Analysis benefits from a diverse set of cluster solutions, such as from different cluster methodologies (e.g. hierarchical, k-means, latent class clustering, DBSCAN, etc.), different basis variables, and different numbers of clusters This is made possible by the fact that Ensemble Analysis does not "look at" the original data, but rather examines only the assignments of individuals to clusters. The consensus solution combines information from those many partitionings to find one which is most representative of them all.

For nearly four decades, we have advocated using k-means clustering rather than hierarchical clustering methods. Our opinion has not changed, as we feel that k-means clustering (from multiple, intelligently-drawn starting points) generally is more robust under more conditions than hierarchical methods. However, when employing cluster ensembles, the literature suggests that ensembles benefit from including solutions representing a variety of methods that involve different inductive biases. Therefore, CCEA includes the capability of developing clusters within the ensemble via hierarchical (complete and average linkage) methods. Given our bias towards the k-means methodology, the default setting for our implementation of Cluster Ensemble Analysis provides more k-means solutions within the ensemble than hierarchical.

We should warn you that the hierarchical clustering methods require significant memory resources of the computer, since an n x n matrix of similarities must be constructed, where n is the number of cases. For its hierarchical solutions, we've found CCEA to have exceptional performance up to about 2,000 cases (runtimes typically about 2 minutes or less, for solutions exploring 2 to 6 groups). With even larger samples, runtimes become much slower, and the computer will eventually run out of memory. (Note: the k-means routines are extremely fast, even for very large datasets.)

We encourage researchers to append additional solutions obtained from other reliable sources within the ensemble file. We do not claim that our particular choice of k-means and hierarchical methods is optimal, and it is likely that we'll offer additional clustering methods within ensemble construction in future versions of the software.

A Direct Consensus Method Using "Clustering on Clusters"

In Strehl and Ghosh's 2002 article, the authors discuss multiple approaches for developing a consensus solution, given the availability of multiple segmentation solutions within an ensemble. Strehl and Ghosh use a method they call a *Meta-Clustering Algorithm*, based on the notion of "clustering clusters."

With the Meta-Clustering Algorithm, one develops multiple clustering solutions. These could vary in terms of:

- Method used (hierarchical, k-means under different starting points, etc.)
- Number of dimensions (for example, selected from 2 to 12 groups)
- Basis variables employed
- Pre-processing options (standardization, centering)

The group assignments for multiple cluster solutions (just three in this example) could look like the following when recorded in a data file:

Caseid	Solution#1	Solution#2	Solution#3
1001	1	4	2
1002	2	2	1
1003	2	3	1
1004	1	4	2

Solutions #1 and #3 are 2-group solutions, and across the first four cases they appear to be identical (except that the labels are switched). Solution #2 is a 4-group solution.

It is very easy to modify this file to have "indicator" (dummy) coding. Strehl and Ghosh code the information for a 2-group solution (such as Solution #1) using two columns, where the first column indicates whether the respondent belongs to the first group and the second column indicates membership in the second group.

Indicator Coding for Solution #1:

1001	1	0
1002	0	1
1003	0	1
1004	1	0

All three solutions in the example above could be coded in eight total indicator columns, or:

Indicator Coding for Solutions 1-3:

1001	1	0	0	0	0	1	0	1
1002	0	1	0	1	0	0	1	0
1003	0	1	0	0	1	0	1	0
1004	1	0	0	0	0	1	0	1

Strehl and Ghosh employ a method that involves repeatedly clustering (using a graph partitioning approach) and relabeling the clusterers, so that cluster #1 from the first solution corresponds to cluster #1 from the second solution, etc. This becomes a challenging optimization problem when many groups are included across many replicates, and with somewhat noisy datasets as would be found in practice.

We use Strehl and Gosh's first step, but have chosen to side-step the issue of relabeling altogether by simply clustering again on the indicator matrix (clustering on the cluster solutions, or "CC") without worrying about relabeling. In the example above, we simply use these eight columns as new basis

variables in a secondary cluster analysis, where we are looking for a final k-group solution (and the indicator variables could represent cluster solutions with either more or fewer clusters than the final k-group solution we seek). For our work, we leveraged CCA's standard approach of running multiple replicates (30) under k-means (using different, intelligently drawn starting points) and we selected the one solution that was most reproducible as a possible final solution and candidate stopping point. We have found it useful to include a large number of cluster solutions in the ensemble, representing a wide variety of numbers of clusters. There doesn't seem to be any harm (overfitting) in including a very large number of runs in the ensemble. We have had good results using sixty or seventy cluster solutions in the ensemble, ranging from 2-group solutions clear up to 30-group solutions. And, we find the final clustering result is more stable (when employing different starting point seeds) if using large, diverse ensembles. Our software implementation seems very fast, with an ensemble analysis as just described typically requiring only about 30 seconds for 1000 respondents.

If multiple solutions are obtained by "clustering on cluster solutions (CC)", one can compute reproducibility across those replicates (we employ 30 replicates) to ascertain how consistently one obtains the same result from different starting points. We might also consider the most reproducible of these as the best solution; however, it is not strictly necessary to introduce the notion of reproducibility. We can recode those replicates (now all on k-groups) using indicator coding and repeat the process (clustering on cluster solutions of cluster solutions (CCC)). This loop can continue indefinitely (CCC…C), but we find that the process converges very quickly, usually within 1 to 3 steps. When no respondents are reclassified in a subsequent step, we may take the previous candidate solution (the most reproducible one) as final. As far as we know, our approach is unique, though it owes a great deal to the notions set forth by Strehl and Ghosh.

The literature suggests that cluster ensembles which use diverse clusterers will be more robust to characteristics in the data that do not conform well to traditional k-means, such as elongated clusters. Even though we use k-means as our method to develop a consensus solution from the indicator coding matrix, the cluster solutions in our ensemble include hierarchical methods that add diversity and can yield more flexible final clusterings. However, our approach to ensemble construction and creating a consensus solution is based on the notion that clusters should be generally compact. For that reason, we have not employed single-linkage hierarchical clustering in the "clustering on clusters" consensus step. Therefore, our implementation should not be expected to work very well in recovering the sorts of artificial structures (spirals, rings, etc.) that other authors have used as a standard for prediction. But our approach should work well in detecting meaningful structure more commonly found in market and social research. And, if desired, one could use single-linkage hierarchical clustering to develop the consensus solution (rather than k-means), and this should do a creditable job of capturing data with very elongated or patterned structures.

How Well Does It Work?

We have compared the standard CCA (Convergent Cluster Analysis) approach of using k-means with 30 replicates (and choosing the most reproducible replicate) to our implementation of Ensemble Analysis across many data sets. The results are described in a white paper entitled, "Improving K-Means Cluster Analysis: Ensemble Analysis instead of Highest Reproducibility Replicates," available for downloading from our Technical Papers library at www.sawtoothsoftware.com. We drew comparisons based on over one-dozen comparisons on data sets between CCA and Ensemble Analysis. Here is one representative example of the performance edge for Ensemble Analysis as reported in that paper:

Comparative Test:

For this test, we developed an artificial dataset with true means and group sizes as follows:

```
                  True Group Means:
Group 1 (n=300):  6  4  4  1 10  4  6  1  7  1
Group 2 (n= 50):  4  5  8  5  5  8  7  3  5  2
Group 3 (n=100): 10  4  4  2  5 10  7  3  4  8
Group 4 (n=200):  5  2  2  8  8  5  2  4  3  1
Group 5 (n=150):  2  3  4  9  2  5  5 10  4 10
Group 6 (n=200):  2  5 10  6  7 10  9  9  3  4
```

We created five separate datasets for this test, disturbing the data by normal random error with standard deviation of 1, 2, 3, 4 or 5.

Hit rates (correct classification rates to known groups) by level of error disturbance were:

```
              CCA      Ensemble
Error = 1   100.0%      100.0%
Error = 2    99.0        99.1
Error = 3    89.1        90.0
Error = 4    73.4        76.1
Error = 5    62.3        70.0
```

Conclusions:

After examining over one-dozen data sets, our conclusions were as follows:

> "Our implementation of ensemble analysis generally performs better than CCA's approach of choosing the most reproducible replicate. The ensemble approach seems especially useful when the true sizes of the groups are quite different (which is often true in practice) and when groups have differing degrees of overlap with respect to each other on the basis variables (again more likely in practice). In those cases, it achieves significantly better hit rates, better fit to true group means, and better estimates of the true group sizes. With equal-sized groups that are completely unique with respect to their means on the basis variables, it seems to perform just as well as CCA's approach. Like CCA, our ensemble method provides a measure of reproducibility, which can be used to help determine how many groups provide a good characterization of the data structure. The reproducibility statistic for our ensemble method seems to perform just as well or better than the similar statistic in CCA for indicating the correct number of groups."
>
> "We haven't evaluated other methods of forming consensus solutions for ensembles, and thus cannot comment on the relative performance of our method versus others described in the literature. This remains an avenue for future research."

Usage Hints:

A common mistake is to use a large number of basis variables, believing that “more is better.” This is counterproductive with cluster analysis, as the more basis variables are used in the clustering procedure, the between-segment differences get smaller and smaller. This is often called “the curse of dimensionality.” For typical sample sizes used in marketing research surveys, it’s much better to carefully select around 20 or fewer variables to segment on than 50 or more.

If you are interested in seeing the results of the clustering runs included in the ensemble, CCEA saves the group memberships for each clustering within the ensemble to a .csv file. You can edit this file to include additional segmentation runs and submit the modified ensemble to CCEA to use in producing a new consensus solution.

Alternative Clustering Methods

We are concerned with methods of grouping objects based on their patterns of similarity to one another. There are some obvious ways of doing this that we shall dispose of briefly:

> Objects are frequently grouped on an a priori basis. We might ask survey respondents which cereal they eat most often, and then group them using their answers to just that question. This can be a very useful way to group objects, and is much simpler than the methods we are considering.
>
> We might combine information from two or more variables using graphical methods. Suppose we were investigating office copiers, and had already decided we were interested in only two variables: price and speed. We could plot each copier's position in a two-space on the basis of its price and speed. Then we might be able to see just by visual inspection where the points in the plot fell into "natural" clusters.

By contrast, the methods we are considering are appropriate for objects that differ on many metric variables simultaneously, and are useful in applications too large or complex to be resolved in the simple ways just described. Three main types of methods have received wide use in marketing and the social sciences.

- Hierarchical Cluster Analysis
- Q Analysis
- Partitioning Methods

We shall consider each method briefly.

Hierarchical Cluster Analysis:

Hierarchical methods start with similarity (or dissimilarity) values for all pairs of objects. If there were only a modest number of objects to be clustered, say 100, we could imagine a table with 100 rows and 100 columns, each entry indicating the similarity between the row object and the column object. At the outset we think of each object as defining its own cluster of size "one."

The algorithm is an extremely simple one:

> 1. Scan the entire (n x n) table to find the two most similar objects.
>
> 2. Combine those objects into a single group of size "two." Do this by deleting the row and column for one of the objects being combined, so the reduced table will have only n-1 rows and columns. Modify values in the row and column for the surviving object to indicate similarities of the other n-2 objects with the newly created group rather than with the former group (in this case, the former object). This may be done in several ways, depending on the choice of algorithm. The most common ways are:

- Each new value is the maximum of the two values it replaces. This produces clusters that are not very compact, since a point can be admitted to a cluster if sufficiently similar to one other member. This is sometimes called the "single linkage" method.

- Each new value is the minimum of the two values it replaces. This produces clusters that are quite compact, since a point can be admitted to a cluster only if sufficiently similar to every other member. This is sometimes called the "complete linkage method."

- Each new value is a weighted average of the two values it replaces.

3. Carry out steps 1 and 2 a total of n-1 times. At each stage two groups are combined and the number of groups remaining is reduced by 1. The groupings at each of the last few stages define the solutions for 2, 3, 4, ... etc. clusters.

The hierarchical methods are simple and elegant. They are widely used for clustering relatively small numbers of objects, and their popularity is deserved. However, they do present problems, particularly when dealing with large numbers of objects:

1. The table of similarities between objects can get very large, in some cases taxing the memory of today's PCs.

2. In our experience, and according to the literature, the hierarchical methods tend to be less reproducible than others. Relatively trivial-appearing decisions made early in the clustering can have large effects on the final outcome. This is less of a problem if the data are relatively error-free. However, when the data have a high level of error, such as with individual responses to questionnaire items, the hierarchical methods seem to have difficulty producing similar cluster structures when clustering new samples of objects from the same universe.

3. Once a hierarchical method groups two objects together, they will always remain that way. This means that two objects together in the three-cluster solution, for example, will also be together in the two-cluster solution. Each solution is obtained by combining two groups from the previous solution. There seems to be no obvious reason why the "best" solution with two clusters should be precisely the same as what can be obtained by combining two groups from the "best" solution with three clusters.

For these reasons, we have chosen not to base CCEA on hierarchical clustering methods. (However, hierarchical clustering is offered as an option for providing a "starting solution" to be refined by k-means.) We should also note that hierarchical methods are provided in CCEA's ensemble analysis, although the consensus solution is developed using k-means.

Q Analysis:

Factor analysis is a technique with a long history of usefulness for exploring relations among variables. The resulting groups are called "factors" rather than "clusters," but the basic similarity to cluster analysis is compelling. Factor analysis starts with a data table similar to that of cluster analysis: a table with objects on one border and variables on the other, with the numbers in the table containing measures of the objects on the variables. Factor analysis starts by computing similarities among variables (usually correlation coefficients) rather than among objects.

It should be no surprise that over the years several researchers have proposed "turning things around" by computing correlation coefficients between objects rather than variables, and then using factor analysis to obtain groups of objects. This technique has been used by psychologists for many years, and is most often known by the name "Q Analysis." As a way of clustering objects, Q Analysis has some strengths and some weaknesses:

1. One strength is that Q Analysis can easily handle a large number of objects. The data for only one object at a time have to be in computer memory for the main portion of the analysis. This means that Q Analysis can handle a virtually unlimited number of objects.

2. Q Analysis is also convenient. Computer programs for factor analysis are widely available.

3. Another strength is that of reproducibility. Our experience with Q Analysis has shown it to do a creditable job of producing similar solutions when used to analyze similar samples of objects from the same population.

4. On the other hand, many statisticians feel a fundamental discomfort with the method. The assumptions of factor analysis are harder for some to accept when the process is "turned on its side" and correlations are computed between objects and across variables.

5. A potentially most limiting problem is due to constraints relating the number of clusters that can be recovered to the number of variables analyzed. Q Analysis cannot produce more clusters than the number of variables. We can all imagine two dimensional spaces populated with points that fall into more than two clusters, but Q Analysis is not able to produce such a solution.

For both of these last two reasons, we have chosen not to base CCEA on Q Analysis. Fortunately, the method we have chosen shares the three above-mentioned strengths, without sharing the corresponding weaknesses.

Partitioning Methods:

The method we have chosen belongs to a class with many names:

- Partitioning Methods
- K Means Methods
- Iterative Reclassification Methods
- "Sift and Shift" Methods
- Convergent Methods

Like the hierarchical methods, the algorithm is simple and easy to visualize. However, unlike hierarchical methods, the solution for a particular number of clusters is obtained independently of solutions for other numbers of clusters. The two-cluster and three-cluster solutions need not have much in common.

We start with a table of object by variable values. We must also decide in advance how many clusters we want to have (indicated by the algebraic symbol k). The algorithm has these steps:

1. Using some means yet to be determined, select k "starting points." It is easiest to think about these points as being a random selection of k of the objects to be clustered, though they need not be. We shall have more to say later about choice of these points.

2. Compute the similarity of each object with a hypothetical object located at each of the k starting points. Classify each object as "belonging to" the group associated with the most similar starting point.

For example, suppose we had 1000 objects with scores on two variables. We could use that information to plot the objects as 1000 points on a two-dimensional scatter plot. Suppose we wanted to find three clusters. Then we could choose three points at random as "starting points." We could use a ruler to measure the distance of each of the 1000 object points to each of the three starting points. We would classify each object as (initially) "belonging to" the group associated with the closest starting point.

3. Calculate the means on the variables for each of the newly-formed groups of object points. The means define the "centers of gravity" of the new groups.

4. Replace the starting points with new points corresponding to these new means.

5. Repeat steps 2 - 4 until no object points are reclassified. When the procedure converges in this way, the points in each group define a cluster.

This procedure has an almost "magical" way of converging to reasonable-appearing solutions. Following are "pictures" of the way this procedure converges in only a few iterations, even with extremely "bad" starting points. (The reader will have to make allowances for imprecision in the location of text characters.)

Suppose the two swarms of "x" characters in Figure 1 were two clusters of points in a two-space awaiting discovery. If we want to find the two-cluster solution, we first pick two starting points. As a "bad case," suppose the starting points are at the "A" and "B" in the point swarm on the right.

Figure 1

```
     xxxxx                 xxxxx
   xxxxxxxxx             xxxxxxxxx
 xxxxxxxxxxxx          xxxxAxxxxxxx
xxxxxxxxxxxxxx       xxxxxxxxxxxxxx
 xxxxxxxxxxxx          xxxxxxxxxxBx
  xxxxxxxxxx            xxxxxxxxxx
    xxxxxx                xxxxxx
```

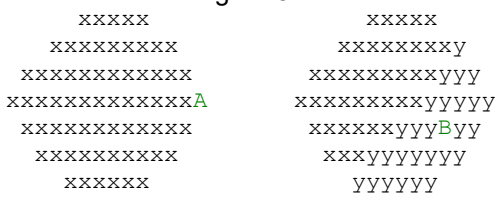
We measure the distance of each "x" to starting points "A" and "B," classifying each "x" into the group associated with the closer of those two. In Figure 2 each point is identified with an "x" or a "y," depending on whether it is closer to "A" or "B."

Figure 2

```
     xxxxx                 xxxxx
   xxxxxxxxx             xxxxxxxxy
 xxxxxxxxxxxx          xxxxAxxxxyyy
xxxxxxxxxxxxxx       xxxxxxxxxyyyyy
 xxxxxxxxxxxx          xxxxxyyyyBy
  xxxxxxxxxx            xxxyyyyyyy
    xxxxxx                yyyyyy
```

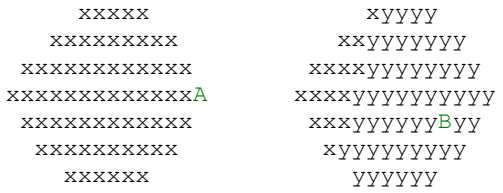
Notice that only the lower right side of the right-hand swarm in Figure 2 is closer to "B" than "A." Now we compute the averages, or "centers of gravity" of all the "x" points and all the "y" points. We indicate those by labels "A" and "B" in Figure 3.

Figure 3



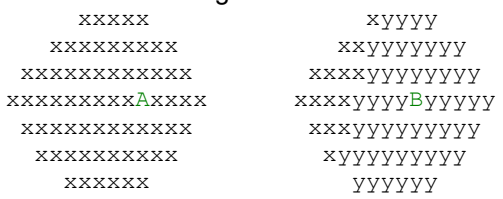
In Figure 4 we have reclassified each point according to whether it is closer to the new "A" or the "B."

Figure 4



Notice that only a minority of points in the right hand swarm are still closer to the "A" than the "B." Again, we compute the averages of the points now classified as "x" and those classified as "y," indicating those positions by "A" and "B" in Figure 5.

Figure 5



Finally, we would classify as "x" all the points closer to "A" and classify as "y" all points closer to "B."

Since all points on the left would now be identified as "x" and all on the right identified as "y," continuation of this process would result in no further reclassification of points.

This process would have converged even more quickly if our starting points had not been chosen so disadvantageously. For example, if one point had been in the swarm on the left and the other in the swarm on the right, convergence might have been immediate.

It's hard to see how any choice of starting points could lead to failure in this simple example. However, sensitivity to choice of starting point is the most serious problem with this method of cluster analysis. With more complex data structures it is usually found that the final solution depends upon choice of starting point. This means that it is worthwhile to try to find a good set of starting points. It also means that we must be especially careful to make sure that the solution we choose is so compelling that similar solutions would also be obtained when starting from other positions.

Milligan and Cooper, in the review cited above, remark:

> "In summary, the convergent k-means method tended to give the best recovery of cluster structure." (page 341)

They also report that choosing starting points at random is a relatively disadvantageous way to begin. In the next section we describe methods that are available in CCEA for improved choice of starting points.

Starting Points

Several types of starting points are available for K-Means clustering.

1. "Distance-based" starting points. These are points (respondents) chosen to be relatively far apart. A random subset* of the respondents is chosen for each replication for determining the starting points.

2. "Hierarchical-based" starting points. A random subset[1] of the respondents are chosen and a hierarchical ("complete linkage") cluster analysis is done. The centroid of each cluster is computed, and those centroids are taken as the starting points. Unless the data set is less than 50 people, the starting solution is likely to be different each time, and this method may be used advantageously to select starting points for several replications.

3. "Density-based" starting points. As with the previous method, a subset of respondents is chosen at random. An analysis is done to select respondents that are near the centers of relatively dense regions in the space. If there are more than 50 respondents in the data file, this method will also produce different starting solutions each time, and can therefore be used profitably for multiple replications.

4. **Mixed strategy.** This approach cycles among all of the starting point methods, including the user-defined strategy (if provided). We recommend this strategy for most purposes.

5. **User-Defined strategy**. If you have previously done work with this or a similar dataset such that you have an existing solution, this can be used as a starting point. If you previously have established group membership for each respondent, you can select a file containing that information as the starting point. Or, if you have group means on the variables, you can select a file that contains that information. The formats and procedures are described in the CCEA Manual. You would not use the user-defined strategy options for replicated clusterings, because they would all produce identical results.

[1] The random subset consists of the larger of 50 respondents or 10% of the entire data set, up to a maximum of 250 respondents; or all respondents if fewer than 50 are available.

References:

Orme, B. and R. Johnson (2008), "Improving K-Means Cluster Analysis: Ensemble Analysis instead of Highest Reproducibility Replicates," available at www.sawtoothsoftware.com/techpap.shtml.

Retzer, J. and M. Shan (2007), "Cluster Ensemble Analysis and Graphical Depiction of Cluster Partitions," Proceedings of the 2007 Sawtooth Software Conference, Sequim WA.

Strehl, A. and J. Ghosh (2002), "Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions," *Journal on Machine Learning Research (JMLR)*, 3:583-617, December 2002.