TWOSTEP CLUSTER

The TwoStep cluster method is a scalable cluster analysis algorithm designed to handle very large data sets. It can handle both continuous and categorical variables or attributes. It requires only one data pass. It has two steps 1) pre-cluster the cases (or records) into many small sub-clusters; 2) cluster the sub-clusters resulting from pre-cluster step into the desired number of clusters. It can also automatically select the number of clusters.

Note: This algorithm applies to SPSS 11.5 and later releases.

Notation

The following notation is used throughout this chapter unless otherwise stated:

K^A	Total number of continuous variables used in the procedure.
K^B	Total number of categorical variables used in the procedure.
L_k	Number of categories for the k-th categorical variable.
R_k	The range of the k-th continuous variable.
N	Number of data records in total.
N_k	Number of data records in cluster k.
$\hat{\mu}_k$	The estimated mean of the k-th continuous variable across the entire dataset.
$\hat{\sigma}_k^2$	The estimated variance of the k-th continuous variable across the entire dataset.
$\hat{\mu}_{jk}$	The estimated mean of the k-th continuous variable in cluster j.
$\hat{\sigma}_{jk}^2$	The estimated variance of the k-th continuous variable in cluster j.
N_{jkl}	Number of data records in cluster j whose k-th categorical variable takes the l-th category.
N_{kl}	Number of data records in the k-th categorical variable that take the l-th category.
d(j, s)	Distance between clusters j and s.
$< j, s >$	Index that represents the cluster formed by combining clusters j and s.

TwoStep Clustering Procedure

Pre-cluster

The pre-cluster step uses a sequential clustering approach. It scans the data records one by one and decides if the current record should be merged with the previously formed clusters or starts a new cluster based on the distance criterion (described below).

The procedure is implemented by constructing a modified cluster feature (CF) tree. The CF tree consists of levels of nodes, and each node contains a number of entries. A leaf entry (an entry in the leaf node) represents a final sub-cluster. The non-leaf nodes and their entries are used to guide a new record quickly into a correct leaf node. Each entry is characterized by its CF that consists of the entry's number of records, mean and variance of each range field, and counts for each category of each symbolic field. For each successive record, starting from the root node, it is recursively guided by the closest entry in the node to find the closest child node, and descends along the CF tree. Upon reaching a leaf node, it finds the closest leaf entry in the leaf node. If the record is within a threshold distance of the closest leaf entry, it is absorbed into the leaf entry and the CF of that leaf entry is updated. Otherwise it starts its own leaf entry in the leaf node. If there is no space in the leaf node to create a new leaf entry, the leaf node is split into two. The entries in the original leaf node are divided into two groups using the farthest pair as seeds, and redistributing the remaining entries based on the closeness criterion.

If the CF tree grows beyond allowed maximum size, the CF tree is rebuilt based on the existing CF tree by increasing the threshold distance criterion. The rebuilt CF tree is smaller and hence has space for new input records. This process continues until a complete data pass is finished. For details of CF tree construction, see the BIRCH algorithm (Zhang, Ramakrishnon, and Livny, 1996).

All records falling in the same entry can be collectively represented by the entry's CF. When a new record is added to an entry, the new CF can be computed from this new record and the old CF without knowing the individual records in the entry. These properties of CF make it possible to maintain only the entry CFs, rather than the sets of individual records. Hence the CF-tree is much smaller than the original data and can be stored in memory more efficiently.

Note that the structure of the constructed CF tree may depend on the input order of the cases or records. To minimize the order effect, randomly order the records before building the model.

Outlier Handling

An optional outlier-handling step is implemented in the algorithm in the process of building the CF tree. Outliers are considered as data records that do not fit well into any cluster. We consider data records in a leaf entry as outliers if the number of records in the entry is less than a certain fraction (25% by default) of the size of the largest leaf entry in the CF tree. Before rebuilding the CF tree, the procedure checks for potential outliers and sets them aside. After rebuilding the CF tree, the procedure checks to see if these outliers can fit in without increasing the tree size. At the end of CF tree building, small entries that cannot fit in are outliers.

Cluster

The cluster step takes sub-clusters (non-outlier sub-clusters if outlier handling is used) resulting from the pre-cluster step as input and then groups them into the desired number of clusters. Since the number of sub-clusters is much less than the number of original records, traditional clustering methods can be used effectively. TwoStep uses an agglomerative hierarchical clustering method, because it works well with the auto-cluster method (see the section on auto-clustering below).

Hierarchical clustering refers to a process by which clusters are recursively merged, until at the end of the process only one cluster remains containing all records. The process starts by defining a starting cluster for each of the sub-clusters produced in the pre-cluster step. (For more information, see "Pre-cluster" on p. 2 .) All clusters are then compared, and the pair of clusters with the smallest distance between them is selected and merged into a single cluster. After merging, the new set of clusters is compared, the closest pair is merged, and the process repeats until all clusters have been merged. (If you are familiar with the way a decision tree is built, this is a similar process, except in reverse.) Because the clusters are merged recursively in this way, it is easy to compare solutions with different numbers of clusters. To get a five-cluster solution, simply stop merging when there are five clusters left; to get a four-cluster solution, take the five-cluster solution and perform one more merge operation, and so on.

Accuracy

In general, the larger the number of sub-clusters produced by the pre-cluster step, the more accurate the final result is. However, too many sub-clusters will slow down the clustering during the second step. The maximum number of sub-clusters should be carefully chosen so that it is large enough to produce accurate results and small enough not to slow down the second step clustering.

Distance Measure

Log-Likelihood Distance

The log-likelihood distance measure can handle both continuous and categorical variables. It is a probability based distance. The distance between two clusters is related to the decrease in log-likelihood as they are combined into one cluster. In calculating log-likelihood, normal distributions for continuous variables and multinomial distributions for categorical variables are assumed. It is also assumed that the variables are independent of each other, and so are the cases. The distance between clusters j and s is defined as:

$$d\left(i,j\right)=\xi_i+\xi_j-\xi_{\langle i,j\rangle}$$

where

$$\xi_v=-N_v\left(\sum_{k=1}^{K^A}\frac{1}{2}\log\left(\hat{\sigma}_k^2+\hat{\sigma}_{vk}^2\right)+\sum_{k=1}^{K^B}\hat{E}_{vk}\right)$$

and

$$\hat{E}_{vk} = -\sum_{l=1}^{L_k} \frac{N_{vkl}}{N_v} \log \frac{N_{vkl}}{N_v}$$

If $\hat{\sigma}_k^2$ is ignored in the expression for ξ_{v}, the distance between clusters *i* and *j* would be exactly the decrease in log-likelihood when the two clusters are combined. The $\hat{\sigma}_k^2$ term is added to solve the problem caused by $\hat{\sigma}_{vk}^2 = 0$, which would result in the natural logarithm being undefined. (This would occur, for example, when a cluster has only one case.)

Euclidean Distance

This distance measure can only be applied if all variables are continuous. The Euclidean distance between two points is clearly defined. The distance between two clusters is here defined by the Euclidean distance between the two cluster centers. A cluster center is defined as the vector of cluster means of each variable.

Number of Clusters (auto-clustering)

TwoStep can use the hierarchical clustering method in the second step to assess multiple cluster solutions and automatically determine the optimal number of clusters for the input data. A characteristic of hierarchical clustering is that it produces a sequence of partitions in one run: 1, 2, 3, … clusters. In contrast, a *k*-means algorithm would need to run multiple times (one for each specified number of clusters) in order to generate the sequence. To determine the number of clusters automatically, TwoStep uses a two-stage procedure that works well with the hierarchical clustering method. In the first stage, the BIC for each number of clusters within a specified range is calculated and used to find the initial estimate for the number of clusters. The BIC is computed as

$$BIC(J) = -2\sum_{j=1}^{J} \xi_j + m_J \log(N)$$

where

$$m_J = J\left\{2K^A + \sum_{k=1}^{K^B} (L_K - 1)\right\}$$

and other terms defined as in “Distance Measure”. The ratio of change in BIC at each successive merging relative to the first merging determines the initial estimate. Let $dBIC(J)$ be the difference in BIC between the model with J clusters and that with (J + 1) clusters, $dBIC(J) = BIC(J) - BIC(J+1)$. Then the change ratio for model J is

$$R_1(J) = \frac{dBIC(J)}{dBIC(1)}$$

If $dBIC(1) < 0$, then the number of clusters is set to 1 (and the second stage is omitted). Otherwise, the initial estimate for number of clusters*k* is the smallest number for which $R_1(J) < 0.04$

In the second stage, the initial estimate is refined by finding the largest relative increase in distance between the two closest clusters in each hierarchical clustering stage. This is done as follows:

- Starting with the model C_k indicated by the BIC criterion, take the ratio of minimum inter-cluster distance for that model and the next larger model C_{k+1}, that is, the previous model in the hierarchical clustering procedure,

  $$R_2(k) = \frac{d_{\min}(C_k)}{d_{\min}(C_{k+1})}$$

  where C_k is the cluster model containing *k* clusters and $d_{\min}(C)$ is the minimum inter-cluster distance for cluster model *C*.

- Now from model C_{k-1}, compute the same ratio with the following model C_k, as above. Repeat for each subsequent model until you have the ratio $R_2(2)$.

- Compare the two largest R_2 ratios; if the largest is more that 1.15 times the second largest, then select the model with the largest R_2 ratio as the optimal number of clusters; otherwise, from those two models with the largest R_2 values, select the one with the larger number of clusters as the optimal model.

Variable Importance

The relative contribution of a variable to a cluster's creation can be computed for both continuous and categorical variables.

Continuous Variables

When the variable is continuous, the importance measure is based on:

$$t = \frac{\hat{\mu}_k - \hat{\mu}_{jk}}{\hat{\sigma}_{jk}/\sqrt{N_k}}$$

which, under the null hypothesis, is distributed as a *t* with $N_k - 1$ degrees of freedom. The significance level is two-tailed. Either the *t* statistic or its significance level can be reported as the importance measure.

Categorical Variables

When the variable is categorical, the importance measure is based on:

$$\chi^2 = \sum_{l=1}^{L_k} \left(\frac{N_{jkl} - N_{kl}}{N_{kl}}\right)^2$$

which, under the null hypothesis, is distributed as a χ^2 with L_k degrees of freedom. Either the χ^2 statistic or its significance level can be reported as the importance measure.

Cluster Membership Assignment

Without Outlier-Handling

Assign a record to the closest cluster according to the distance measure.

With Outlier-Handling

Log-Likelihood Distance

Assume outliers or noises follow a uniform distribution. Calculate both the log-likelihood resulting from assigning a record to a noise cluster and that resulting from assigning it to the closest non-noise cluster. The record is then assigned to the cluster which leads to the larger log-likelihood. This is equivalent to assigning a record to its closest non-noise cluster if the distance between them is smaller than a critical value $C = \log(V)$, where $V = \prod_m R_m \prod_m L_m$. Otherwise, designate it as an outlier.

Euclidean Distance

Assign a record to its closest non-noise cluster if the Euclidean distance between them is smaller than a critical value $C = 2\sqrt{\frac{1}{JK_A}\sum_{j=1}^{J}\sum_{k=1}^{K_A}\hat{\sigma}_{jk}^2}$. Otherwise, designate it as an outlier.

Missing Values

No missing values are allowed. Cases with missing values are deleted on a listwise basis.

References

Zhang, T., R. Ramakrishnon, and M. Livny. 1996. BIRCH: An efficient data clustering method for very large databases. In: *Proceedings of the ACM SIGMOD Conference on Management of Data,* Montreal, Canada: ACM, 103–114.

Chiu, T., D. Fang, J. Chen, Y. Wang, and C. Jeris. 2001. A Robust and Scalable Clustering Algorithm for Mixed Type Attributes in Large Database Environment. In: *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining,* SanFrancisco, CA: ACM, 268–263.