Elongated Intersected Clusters and Radial Coordinates Transformation Using SAS®
A. Suprun, CIBC, Toronto, ON

ABSTRACT
The paper compares a traditional approach to elongated intersected clustering using SAS® ACECLUS, CLUSTER, TREE procedures with a suggested approach that includes a preliminary transformation to radial coordinates. This transformation produces a much clearer picture of clusters and allows their easy separation.

The approach is good for elongated clusters intersecting at the same point. It does not matter if this point is the origin or not. If more than two clusters intersect at more than one point then this approach can be used in stages for every intersecting point separately.

Base SAS® 9.2 has been used to implement this approach, but older versions can be used too. The only limitation is the availability of the clustering procedures mentioned. Operating system is Windows XP.

INTRODUCTION
When dealing with intersected clusters in two dimensional settings, the following questions arise: How many clusters are there? How to separate them? It is also well known that clustering methods work well for round shaped clusters [1] while the elongated ones provide some challenges.

Using an example of two dimensional elongated intersected clusters, the paper compares a traditional approach to this problem with SAS® ACECLUS, CLUSTER, TREE procedures with a suggested approach that includes a preliminary transformation to radial coordinates.

This example (see Fig. 1) is based on the real life problem I had. Y is a target variable, while X is a predictor. It is desirable to separate three elongated clusters intersecting in the origin to perform subsequent regression Y=a_iX+b_i, where i = 1, 2, 3 is a cluster number.

We could solve this problem manually by drawing two straight lines coming from the origin, e.g. as shown on Fig. 1. However, there are many possible arrangements of these lines, and a question arises on the quality of clusters produced this way. It is necessary to have some criterion in order to select the best arrangement.
STRAIGHTFORWARD APPROACH

The methodology [1, 2] recommends converting to canonical variables in order to get round shaped, “cloud” clusters. The PROC ACECLUS performs this task, and then a clustering procedure, e.g. PROC CLUSTER, is used:

/* Making Copies of Y and X */;
   data y1;
      set y;
         Xst=X;  Yst=Y;
   run;
/* Making “Cloud” Clusters */;
   %let pr=0.2;
   proc aceclus data=y1 proportion=&pr maxiter=20 out=aceout;
      var Xst Yst;
   run;
/* Clustering */;
   proc cluster data=aceout method=centroid outtree=tree CCC pseudo noprint;
      var can:;
      copy X Y;
   run;
/* Selecting 3 clusters */;
   %let nclr=3;
   proc tree data=tree nclusters=&nclr out=treeout noprint;
      copy X Y can:;
   run;

It is a good idea to keep a copy of initial variables before doing any variable transformations because you can plot clusters on transformed or initial variables any time. Moreover, some procedures modify initial variables, e.g. PROC ACECLUS transforms initial variables into canonical ones. It is also recommended [1] to standardize variables before clustering, however in this example it is assumed that X and Y variables have the same scale for clarity.

PROC ACECLUS modifies initial variables to make clusters more round shaped, “cloud clusters”. This procedure requires the proportion parameter (the percentage of pairs to be included in estimation of within cluster covariance matrix), and the maxiter parameter (maximum number of iterations). I’ve used recommended values for these parameters.

PROC CLUSTER performs hierarchical clustering using the centroid method. The method arranges observations into clusters to make Euclidian distance between cluster means as big as possible: \( D_{ij}^2 = \sum (x_i - x_j)^2 \). Parameter CCC displays the cubic clustering criterion and produces approximate expected R^2, pseudo F and T^2 statistics. These results are intended to select appropriate number of clusters. They are stored in the tree dataset. Macro variable nclr is set to 3, the desired number of clusters.

PROC TREE creates the treeout dataset grouping all observations into three clusters. Original and canonical variables are reported for every observation. The procedure can also be used to plot clustering statistics.

Fig. 2 shows the result of clustering. It is clear that it is not what we intended to get, namely the blue cluster occupies some points that should belong to the green and orange clusters. It is worth noting that it is possible to separate good clusters by playing with different clustering methods and adjusting parameters. However, there is a more natural way to do this.
The configuration of clusters looks like three rays coming from the origin, so it is worth to try a radial coordinate transformation.

\[ r = \sqrt{x^2 + y^2}, \]
\[ \varphi = \arctan \left( \frac{y}{x} \right). \]

The result of this transformation is shown on Fig. 3. The clusters now are quite well separated on the \( \varphi \)-axis.
The same program is then applied to the transformed variables $r$ and $\phi$.

```sas
/* Radial Coordinates Transformation*/
data y2;
  set y;
  R=SQRT(X**2+Y**2);  Phi=ATAN(Y/X)/&pi*180;
  Rst=R;  Fst=Phi;
run;
/* Making "cloud" clusters */
proc aceclus data=y2 proportion=&pr maxiter=20 out=aceoutr;
  var Rst Fst;
run;
/* Cluster */
proc cluster data=aceoutr method=centroid outtree=treer CCC pseudo noprint;
  var can:;
  copy R Phi X Y;
run;
/* Selecting 3 Clusters */
proc tree data=treer nclusters=&nclr out=treeoutr noprint;
  copy R Phi X Y can:;
run;
```

The result of clustering in initial coordinates is shown on Fig. 4. Now clusters are well separated and suitable for subsequent regression. It should be noted that in the origin area the separation is questionable but outside this area separation is very good. Moreover, the clustering is done automatically without any arbitrary manual operations, e.g. drawing straight lines to separate clusters.
APPLICABILITY

This approach is good for elongated clusters intersecting at the same point, does not matter if this point is the origin or not. If they intersect outside the origin it is possible to shift coordinates to move the intersection point to the origin.

However, if clusters intersect in more than one point (see Fig. 5) then this approach can be used in stages, first for the data above the red dashed line and then for the data below it. For the data above the red line, it is necessary to shift coordinates to move the intersection point to the origin: $X_1=X-1.8$ and $Y_1=Y-2.8$. Then radial coordinates transformations is applied; and the above clustering program is executed.

The data below the red line have clusters intersecting at the origin so I only perform radial transformation and execute the clustering code. The last step combines resulting clusters from data above and below the red line. Fig. 6 illustrates the result of clustering for the two sets of data.
Fig. 5. Clusters intersecting in two points

Fig. 6. Resulting clusters for data above and below the red line
CONCLUSIONS
It has been shown that radial coordinates transformation is a natural approach to separate elongated intersected clusters. The applicability of this approach to clusters intersecting at more than one point has been demonstrated. The appropriate SAS® programs illustrate the approach.

REFERENCES
1. SAS® Course “Applied Clustering Techniques”.

ACKNOWLEDGMENTS
SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.