

## Introducing the MBC Procedure for Model-Based Clustering

Dave Kessler, SAS Institute Inc., Cary, NC

### ABSTRACT

Clustering has long been used to learn the structure of data and to classify individuals into distinct groups. For example, cluster analysis can be used for marketing segmentation that distinguishes potential buyers from nonbuyers. Classical clustering methods use heuristics or simple Euclidean distance to form clusters, but in some cases, the clustering solution is more parsimonious and more useful if it is based on the formal likelihood of a model. The model and its associated likelihood allow the clustering to accommodate more complex covariance structure that is exhibited in the data. The MBC procedure, available in SAS<sup>®</sup> Visual Statistics 8.3 in SAS<sup>®</sup> Viya<sup>®</sup>, enables you to fit mixtures of multivariate Gaussian (normal) distributions to your data to learn a cluster structure in an unsupervised manner. In addition to clustering the observations in a data set, you can use the fitted model to classify new observations. PROC MBC provides a weight of association for each new observation, enabling you to decide whether a new classification is a strong match for one cluster or needs closer expert examination to determine its cluster membership.

This paper describes the concepts behind model-based clustering and presents the basic mode of operation of PROC MBC. Several examples illustrate different use cases, including automated model selection through information criteria, the modeling of outliers, saving models, and applying saved models to new input data.

### INTRODUCTION

Clustering is the general term for machine learning techniques that group items according to a measure of likeness. In this usage, “learning” means developing a model for the data, often with the goal of making predictions about new data that are assumed to come from the same population that generated the data used for the model.

Clustering methods appeal to the idea that clustering is grouping like items with like items. These techniques measure this likeness by using a distance between the objects to be clustered. In many of these methods, such as  $k$ -means (MacQueen 1967), objects that are near to each other in Euclidean distance are clustered together.

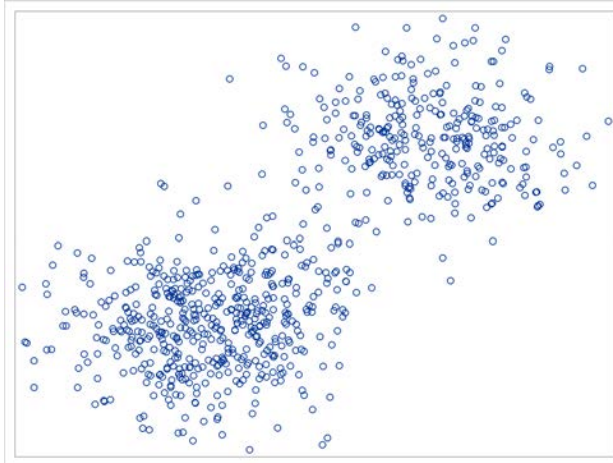
However, there are cases in which the clusters can be represented more parsimoniously if the clusters are not based solely on Euclidean distance. In addition, some scenarios include observations that do not fit well with any cluster and can be modeled better as noise. Also, the lack of a formal statistical model in some of these distance-based clustering methods means that assessment or selection of clustering solutions lacks principled statistical guidance.

Model-based clustering addresses many of these concerns by considering the clusters to be components in a finite mixture model. This allows the use of clusters that have various shapes as well as clusters whose role is to account for noise. Because the clustering is based on a statistical model, you can apply well-established statistical principles of model selection to decide how many clusters could sufficiently represent the data. For a comprehensive introduction to model-based clustering and finite mixture models, see McNicholas (2017) and McLachlan and Peel (2000).

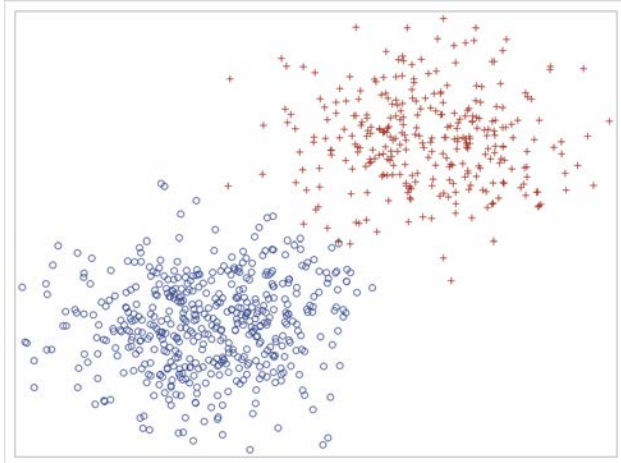
### CLUSTERING AND MODEL-BASED CLUSTERING

Figure 1 and Figure 2 show a simple two-dimensional data set that illustrates the basic problem in clustering. Figure 1 plots the data set but does not include the cluster labels. Simply put, the clustering problem is to recover the hidden cluster labels for the observations. Although Figure 1 shows clearly that there are two clusters, you need a general clustering method for routine applications to data sets. The  $k$ -means method is one such clustering method. Figure 2 shows the clustering that results when the  $k$ -means method is used to find two clusters. In this case, the distance-based clustering that  $k$ -means uses works well and identifies the same clusters that a visual inspection does.

**Figure 1** Unlabeled Data

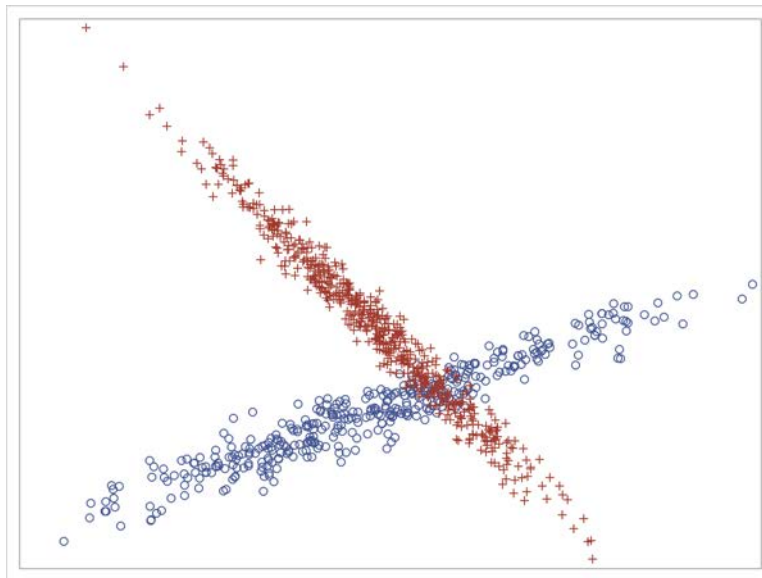


**Figure 2** K-Means Clustering Results



However, the  $k$ -means method and other clustering methods that are based on Euclidean distance disregard any correlation between the different features in the data set. **Figure 3** shows another data set that has two underlying groups. In this case, the two variables that make up the observations are correlated within each cluster. The groups are labeled with distinct symbols and colors.

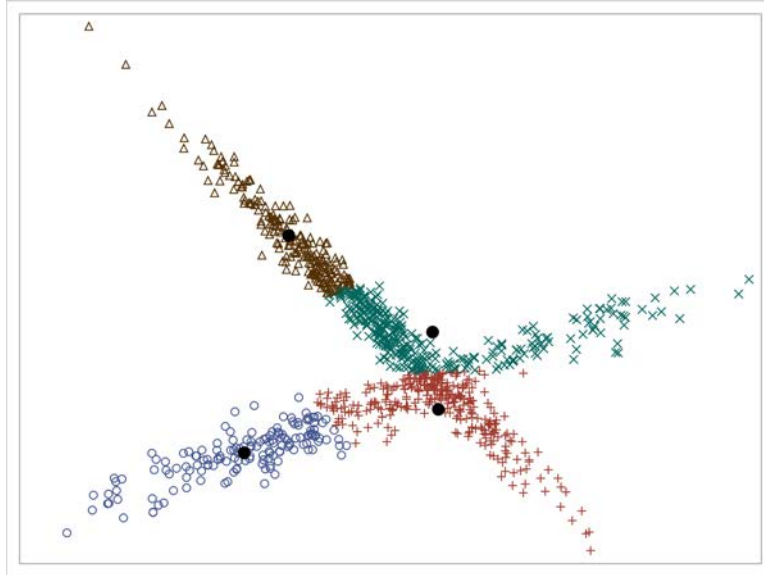
**Figure 3** Two Clusters with High Intracluster Correlation



A traditional distance-based clustering method like  $k$ -means will fail to model this type of cluster. If you apply the  $k$ -means method to these data, the total number of clusters might be larger because a larger number of circular, Euclidean distance-based clusters are needed to cover the data.

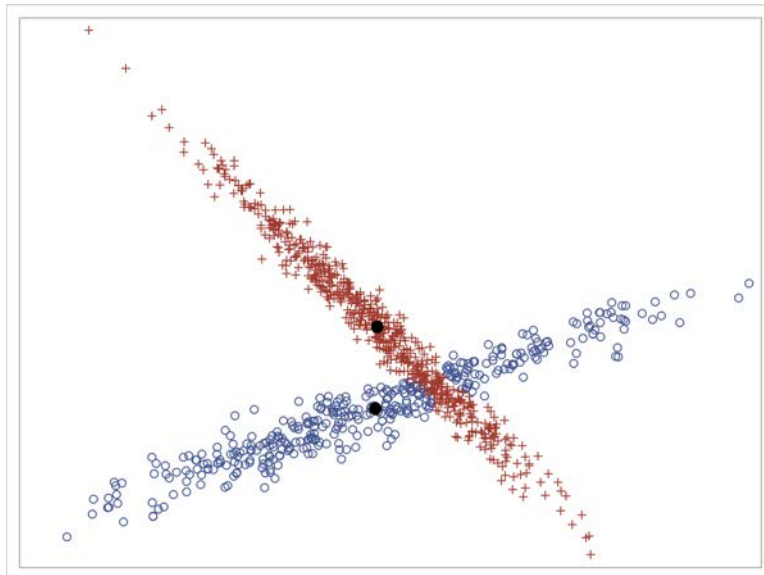
**Figure 4** shows the clustering results that the  $k$ -means method produces for these data. The final cluster assignments are indicated by different plot symbols and colors. In addition, the center of each cluster is indicated with a black dot. In this case, the  $k$ -means method finds four clusters, but the shape that is defined by the clustering assignments cannot accommodate the structure indicated by a visual inspection.

**Figure 4** K-Means Clustering Results



In contrast, model-based clustering incorporates correlations among the variables into its distance metric. As a result, it allows more flexible shapes for the clusters and can produce a more parsimonious clustering. [Figure 5](#) shows the clustering results from the model-based clustering approach, which selects a model that has two clusters.

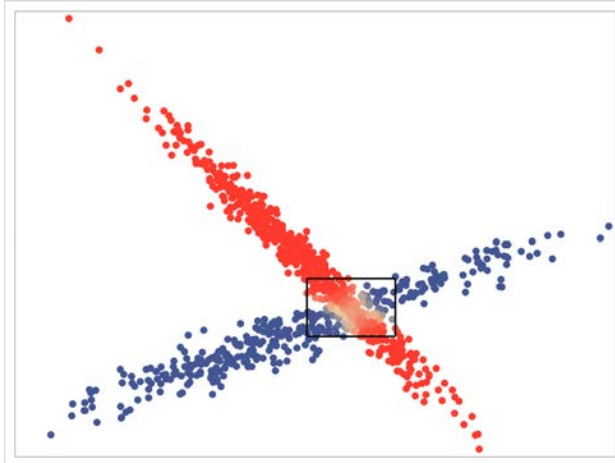
**Figure 5** Model-Based Clustering Results



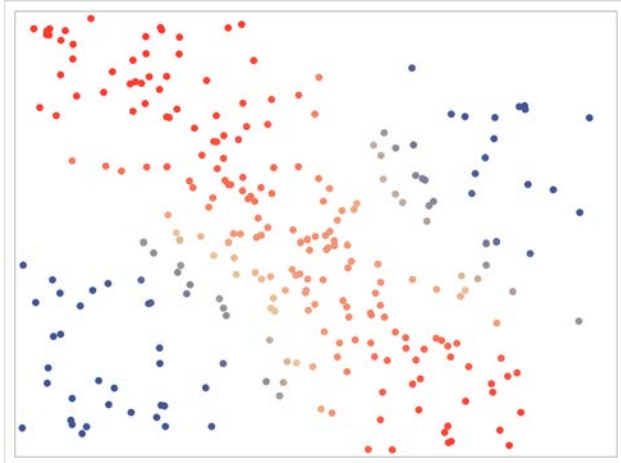
In addition to failing to consider the covariance structure of clusters, methods like *k*-means require each observation to be identified with exactly one cluster. But in practice, not all observations have clear cluster memberships. Methods like *k*-means do not account for the idea that observations might have non-negligible associations with more than one cluster. In contrast, model-based clustering can produce a weight of association—also called the posterior probability of each observation belonging to each cluster—that is based on a formal likelihood.

[Figure 6](#) plots the same result as in [Figure 5](#), but the color of the plot point indicates the strength of association with the two clusters. Points in the area of overlap between the two clusters show a strength of association that is between the two extremes. [Figure 7](#) expands the area outlined in [Figure 6](#), showing how the strength of the association, which is indicated by the darkness of the plot point, is lower in the area of overlap.

**Figure 6** Model-Based Clustering Weights



**Figure 7** Detail of Clustering Weights



In summary, model-based clustering offers several advantages over traditional clustering methods:

- It allows nonspherical cluster shapes through the modeling of flexible covariance structures.
- It permits the use of well-established statistical principles for model selection.
- It permits the use of a noise component to model noise data that are not of primary interest.
- It provides cluster membership weights instead of hard classification.

The following discussion and examples illustrate these ideas and apply them to several different situations.

### The Model

The “model” in model-based clustering is a finite mixture model that has the density function

$$f(y) = \sum_{k=1}^G \pi_k f_k(y|\theta_k)$$

where  $\pi_k$  are mixture weights,  $f_k$  indicates the density function for the  $k$ th component, and  $\theta_k$  are parameters that characterize the density function for the  $k$ th component. In PROC MBC, each  $f_k$  is either a multivariate normal distribution, where  $\theta_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ , or a uniform distribution, where  $\theta_k$  is a single parameter  $v$ . The uniform distribution is for modeling a noise component. There is at most one such noise component in the mixture model.

For components that have a multivariate Gaussian distribution,  $\boldsymbol{\Sigma}_k$  can be any nonsingular covariance matrix. PROC MBC characterizes the set of  $\boldsymbol{\Sigma}_k$  according to their similarities when the  $\boldsymbol{\Sigma}_k$  are expressed in an eigenvalue decomposition,

$$\boldsymbol{\Sigma}_k = \lambda_k \mathbf{D}_k \mathbf{A}_k \mathbf{D}_k'$$


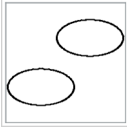

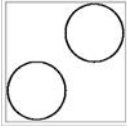
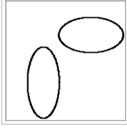

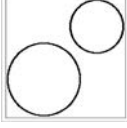


where  $\lambda_k$  is the volume parameter,  $\mathbf{D}_k$  is the orientation matrix, and  $\mathbf{A}_k$  is the shape matrix. The  $\mathbf{D}_k$  matrices and the  $\mathbf{A}_k$  matrices are defined to have determinants equal to 1.

The mixture is characterized by the constraints that are placed on the separate  $\boldsymbol{\Sigma}_k$ . These constraints are summarized in a three-letter code, in which the first letter indicates constraints on the volume parameter,  $\lambda_k$ ; the second letter indicates constraints on the shape matrix,  $\mathbf{A}_k$ ; and the third letter indicates constraints on the orientation matrix,  $\mathbf{D}_k$ . The letter “V” means that the corresponding parameter varies across the components, and the letter “E” means that the parameter is equal across the components. The letter “I” is used only for constraints on the orientation matrices or the shape matrices. When “I” is used for the orientation matrices, it indicates that the  $\boldsymbol{\Sigma}_k$  are not rotated relative to the coordinate axes and that each  $\mathbf{D}_k$  is equal to the identity matrix. When “I” is used for the shape matrices, it indicates that each  $\boldsymbol{\Sigma}_k$  is isotropic, and so each  $\mathbf{A}_k$  is equal to the identity matrix. In this case, the form of the  $\mathbf{D}_k$  does not

matter, and the letter “I” is always used for the orientation matrices. For example, a “VII” mixture is one in which the covariance matrices  $\Sigma_k$  have variable volume ( $\lambda_k$ ) parameters but are constrained to have isotropic (equal variance for each element and no correlation between pairs) covariance matrices. In contrast, an “EVI” mixture constrains the  $\lambda_k$  to all be equal to each other, but it permits the covariance matrices to have different shapes ( $\mathbf{A}_k$ ) and constrains the individual covariance matrices to have no rotation relative to the coordinate axes. In a “VVV” mixture, the separate  $\Sigma_k$  are not constrained in terms of volume, shape, or orientation. For a further discussion of the different constraints and this naming convention, see Fraley and Raftery (1998, 2007).

Table 1 displays the features of each covariance structure and includes graphical representations of each structure.

**Table 1** Covariance Structures

Name	Structure	Volume	Shape	Orientation	Profile
EEE	$\lambda \mathbf{DAD}'$	Equal	Equal	Equal	
EEI	$\lambda \mathbf{A}$	Equal	Equal	Coordinate axes	
EEV	$\lambda \mathbf{D}_k \mathbf{A} \mathbf{D}'_k$	Equal	Equal	Variable	
EII	$\lambda \mathbf{I}$	Equal	Spherical	Coordinate axes	
EVI	$\lambda \mathbf{A}_k$	Equal	Variable	Coordinate axes	
EVV	$\lambda \mathbf{D}_k \mathbf{A}_k \mathbf{D}'_k$	Equal	Variable	Variable	
VII	$\lambda_k \mathbf{I}$	Variable	Spherical	Coordinate axes	
VVI	$\lambda_k \mathbf{A}_k$	Variable	Variable	Coordinate axes	
VVV	$\lambda_k \mathbf{D}_k \mathbf{A}_k \mathbf{D}'_k$	Variable	Variable	Variable	

## Estimation

The MBC procedure uses the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) to obtain parameter estimates for the mixture models. PROC MBC treats the unknown individual component memberships as missing data. In the E-step, the procedure computes expected values for each observation's probability of assignment to each cluster, given the current values of the  $\pi_k$ ,  $\mu_k$ , and  $\Sigma_k$  parameters. These probabilities are the observation's partial memberships in each cluster. The procedure then uses these observation-specific probabilities to accumulate weighted sums of probabilities, observation vectors, and observation vector crossproducts for each cluster. In the M-step, these weighted sums provide the sufficient statistics that are needed for maximum likelihood estimates of the  $\pi_k$ ,  $\mu_k$ , and  $\Sigma_k$  parameters. The procedure repeats the E-step and the M-step, and it terminates when the ratio of the change in log likelihood between successive iterations to the current log likelihood is smaller than a configurable value.

When the mixture model includes a noise component, PROC MBC determines the parameter  $v$  from the volume that is occupied by the data and does not update this parameter during the EM iterations.

To implement the EM algorithm, you need some starting points for either the per-observation clustering weights or the parameter estimates. The MBC procedure has three methods for determining the starting point. The default method randomly assigns observations to the separate components and then uses these assignments to drive the first M-step; you can use the SEED= option to generate different sets of random assignments. You can also control the first M-step by using the INIT statement to specify variables that contain the initial weights for each observation. In this case, these initial weights are not randomly set by the procedure. Finally, you can use the INIT=KMEANS option. With this option, PROC MBC uses the  $k$ -means method to determine initial cluster centers, variances, and weights; these initial values drive the first E-step.

## Clustering Weights

Parameter estimation that uses the EM algorithm computes estimated weights for each observation, given the values of the mixture component parameters and the overall mixing weights. If you know the parameter values ( $\pi_k$ ,  $\theta_k$ ) and the observation vector ( $y_{i1}, \dots, y_{id}$ ), you can compute the observation-specific weights as follows:

$$\pi_{ik} = \frac{\pi_k f_k(\mathbf{y}_i; \theta_k)}{\sum_{h=1}^K \pi_h f_h(\mathbf{y}_i; \theta_h)}, k = 1, \dots, K$$

The weights sum to 1 for each observation  $y_i$ .

You can also use these weights as measures of the strength of association between each observation and the clusters in the model. In this setting, the weights are also referred to as “posterior weights.” The term “posterior” is used because the “prior” weights, which are represented by the  $\pi_k$ , are updated by the likelihood, which is represented by the  $f_k$ .

During the E-step, these weights are used as the cluster membership for each observation.

## Model Selection

One advantage of using a formal statistical model to define the clustering is that you can compare the fit of clustering models by using metrics that are based on well-established statistical principles. For example, you can use the log likelihood or information criteria like the Bayesian information criterion (BIC). Whereas the log likelihood provides a measure of absolute fit of the models, the information criteria balance the absolute fit and the complexity of the models by introducing a penalty for model complexity.

## SUMMARY OF PROC MBC FEATURES

PROC MBC fits mixtures of multivariate normal and uniform distributions to continuous data. The data must be in a SAS<sup>®</sup> Cloud Analytic Services (CAS) table. The procedure uses the expectation-maximization algorithm (Dempster, Laird, and Rubin 1977) to estimate the model parameters and address the unknown cluster membership.

In addition, PROC MBC provides the following features:

- nine different covariance structures for multivariate normal components
- a noise component for data outside multivariate normal clusters

- automated model selection that uses different measures of fit
- three methods of generating starting values
- use of distributed computing resources and multithreading for computation
- scoring of input data by using posterior mixture weights
- creation of binary model stores for scoring additional data sets

## EXAMPLE 1: DIAGNOSIS WITHOUT DIAGNOSIS

In this example, you see how a clustering procedure can discover much of the same structure as a traditional logistic regression, even without knowledge of the disease status of the observations in the training sample.

In traditional exploration of disease risk factors, you might use logistic regression to model the probability of developing a disease as a function of different predictors. In this setting, you know the disease outcome. But if your knowledge is limited to relevant risk factors and demographic information, you can still discover structure in the data that is indicative of the disease status.

The National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) has studied diabetes in the Akimel O’otham people in Arizona for over 30 years. Smith et al. (1988) donated data derived from these studies for use by the scientific community. These data are for female patients who are at least 21 years of age.

The following SAS statements create the data table **mycas.Diabetes**:

```
data mycas.Diabetes;
  input NPreg Glucose Pressure Triceps BMI Pedigree Age Diabetes Test@@;
  datalines;
6  148  72  35  33.6  0.627  50  1  1    1  85  66  29  26.6  0.351  31  0  1
1  89  66  23  28.1  0.167  21  0  0    3  78  50  32   31  0.248  26  1  0
... more lines ...

1  118  58  36  33.3  0.261  23  0  1    8 155  62  26   34  0.543  46  1  0
;
```

Table 2 describes the nine variables in the data table.

**Table 2** Variables in the **mycas.Diabetes** Data Table

Variable	Description
<b>NPreg</b>	Number of pregnancies
<b>Glucose</b>	Two-hour plasma glucose concentration in an oral glucose tolerance test
<b>Pressure</b>	Diastolic blood pressure (mmHg)
<b>Triceps</b>	Triceps skin fold thickness (mm)
<b>BMI</b>	Body mass index (weight in kg/(height in m) <sup>2</sup> )
<b>Pedigree</b>	Diabetes pedigree function
<b>Age</b>	Age (years)
<b>Diabetes</b>	0 if test negative for diabetes, 1 if test positive
<b>Test</b>	0 for training role, 1 for test

The variable **Test** identifies observations to be used in testing the fit. Observations whose **Test** value is equal to 0 are used in training or estimating the model. There are 330 observations in the training set and 202 observations in the test set. This example demonstrates how you can discover groups by clustering without using the disease status variable **Diabetes**. The disease status variable is used only in the assessment of the model fit and in comparison with other methods.

It is natural to restrict the analysis to mixtures that have two components, because the fundamental question concerns the presence or absence of disease in the study population. The following statements enable you to fit two-cluster models that use each of the covariance structures that are provided by the procedure. The NCLUSTERS=2 option

restricts the procedure to models that contain two Gaussian clusters. The COVSTRUCT option specifies a list of all nine supported covariance structures. The procedure uses the BIC to select the model that has the best fit. In this invocation of PROC MBC, the initial cluster memberships are determined by a random draw. The SEED option specifies the random seed that controls the initial assignment of observations to clusters, and it permits reproducible results under appropriate conditions.

```
proc mbc data=mycas.Diabetes (where=(test=0) )
  nclusters=2
  covstruct=(EEE EEI EEV EII EVI EVV VII VVI VVV)
  seed=1945684043;
  vars NPreg Glucose Pressure Triceps BMI Pedigree Age;
run;
```

Figure 8 shows information about the selected model.

**Figure 8** Selected Model Summary  
The MBC Procedure

Model Information	
Number of Gaussian Clusters	2
Covariance Structure	VVV
Noise Cluster Present	No
Expectation Technique	EM
Model Selection Criterion	BIC
Initialization Method	Random
EM Convergence Criterion	1e-05
Singularity Criterion	1e-08
Parameter Criterion	1e-08
Random Seed	1945684043

Figure 9 shows the estimated parameters for the two 7-dimensional multivariate Gaussian components. The mean values of each variable in the clusters might offer suggestions for how to interpret the meaning of the clusters. Cluster 1 appears to represent a group of women who are generally older and who have larger diabetic pedigree values than the women represented by cluster 2. In addition, the women represented by cluster 1 have had more pregnancies, higher glucose levels measured by the glucose tolerance test, higher BMI, higher triceps skinfold measurements, and higher blood pressure measurements. The combination of these observations with general knowledge about risk factors for diabetes suggests that cluster 1 represents the diabetic group within the training sample.



**Figure 9** Parameter Estimates for Selected Model

Cluster Parameter Estimates								
Cluster Variable	Mean	Covariance						
		NPreg	Glucose	Pressure	Triceps	BMI	Pedigree	Age
1 NPreg	5.70111	13.39700	-13.83197	5.37013	2.14457	-0.26701	-0.26385	14.02005
Glucose	134.71979		1088.68990	55.89173	62.20148	33.52675	0.97743	22.93639
Pressure	75.86760			130.03754	3.53867	12.22752	-0.38275	27.04862
Triceps	30.28484				116.35051	32.41058	0.11051	9.57830
BMI	33.13800					37.12695	0.23204	-4.23822
Pedigree	0.62760						0.17095	-0.82164
Age	40.60977							100.99226
2 NPreg	1.67273	2.31808	1.21264	-1.78359	-1.18070	-1.98990	-0.03669	1.79805
Glucose	112.91966		625.89150	68.14198	68.44316	58.17061	-0.13730	11.42867
Pressure	67.71455			147.89311	33.10401	29.38051	-0.07595	5.76021
Triceps	27.75946				120.67263	64.15389	0.48608	10.68884
BMI	32.26677					61.58041	0.35424	5.70984
Pedigree	0.39763						0.03981	0.01193
Age	24.67285							8.35643

Figure 10 shows the estimates of the mixing weights for the two components.

**Figure 10** Mixing Estimates for Selected Model

Cluster Mixing Probability Estimates	
Mixing Component	Mixing Probability
1	0.44683
2	0.55317

You can use the following statements to compare the estimate of the mixing weights with the proportion of observations that correspond to each value of the **Diabetes** variable in the training set:

```
proc freq data=mycas.Diabetes (where=(test=0)) ;
  tables diabetes;
run;
```

Figure 11 shows the proportions of diabetes that are known to exist in the training set. These values are not dramatically different from the mixing weights that PROC MBC estimates. This does not necessarily mean that the clusters identify disease status, but when this information is coupled with the observations about the comparison between the cluster mean values, it does indicate that the model has identified a cluster structure within the data that is related to disease status.

**Figure 11** Frequency of Diabetes in Training Set  
The FREQ Procedure

Diabetes	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	208	63.03	208	63.03
1	122	36.97	330	100.00

You can evaluate the performance of the clustering model by scoring the test observations that were held out of the analysis. In model-based clustering, “scoring” means using the fitted model to compute weights of association for each observation with each cluster, as described in the section “Clustering Weights” on page 6. You must store the fitted model so that it can be applied to the test observations in the original data set.

The following statements include the STORE statement, which saves the selected model in the **mycas.mbcStore** binary store:

```
proc mbc data=mycas.Diabetes (where=(test=0))
    nclusters=2 covstruct=(EEE EEI EEV EII EVI EVV VII VVI VVV)
    seed=1945684043;
    vars NPreg Glucose Pressure Triceps BMI Pedigree Age;
    store mycas.mbcStore;
run;
```

You can then score the test observations by using PROC CAS to invoke the **mbcScore** action directly. The following statements invoke the **mbcScore** action in the **mbc** action set and specify that it score only the test observations. The posterior weights are stored in the **WT1** and **WT2** variables, and the number of the cluster that has the highest posterior weight for each observation is stored in the **CLUSTER** variable. The COPYVARS statement copies all the other variables to the **mycas.mbcScore** data set.

```
proc cas;
    mbc.mbcScore /
        table={name='diabetes' where='(test=1)'}
        casOut={name='mbcscore'}
        restore={name='mbcstore'}
        copyVars='ALL'
        maxPost='CLUSTER'
        nextClus='WT';
run;
quit;
```

You can do a crosstabulation of the known diagnosis for each observation in the test set with the predicted cluster membership that is based on the clustering model. This provides an assessment of the clustering as a way to distinguish the two groups of patients. The following statements use the FREQ procedure to do this:

```
proc freq data=mycas.mbcScore;
    tables diabetes*cluster / nopct nocol;
run;
```

Figure 12 shows the summary table. There is a noticeable difference in the clustering for the different diagnoses.

**Figure 12** Crosstabulation of Disease Status with Predicted Cluster Membership

### The FREQ Procedure

Frequency Row Pct	Table of Diabetes by CLUSTER			
	Diabetes	CLUSTER(Cluster with largest weight using final parameters)		
		1	2	Total
0	54 36.73	93 63.27		147
1	44 80.00	11 20.00		55
<b>Total</b>	98	104		202

As a comparison, you can use the following statements to fit a logistic regression to the training data by using the LOGSELECT procedure in SAS Viya:

```
proc logselect data=mycas.Diabetes (where=(test=0)) ;
    model diabetes(event='1') = NPreg Glucose Pressure Triceps BMI Pedigree Age;
    store mycas.LogisticStore;
run;
```

Then, you can score the test data by using the **logisticScore** action in the **regression** action set. The following statements store the predicted value for each observation in the variable **Predict**:

```

proc cas;
  regression.logisticScore /
    table={name='diabetes' where='(test=1)'}
    restore={name='logisticstore'}
    casOut={name='logisticscore'}
    copyVars='ALL'
    pred='predict';
  run;
quit;

```

You can compare the clusterings by using the receiver operating characteristic (ROC) curves for each model. This requires a predicted weight for the outcome of interest. The mean values for the two clusters shown in [Figure 9](#) suggest that cluster 1 is more strongly associated with a diagnosis of diabetes. The crosstabulation in [Figure 12](#) shows that more subjects with diabetes are associated with cluster 1 than with cluster 2, and it confirms the impression that cluster 1 is more strongly associated with a diabetes diagnosis. On the basis of these observations, you use the **WT1** variable in the CAS output table **mycas.mbcScore** as the predicted value for the ROC curve. In the output data set **mycas.LogisticScore** that PROC LOGSELECT produces, the **PREDICT** variable contains the predicted value that you use for the ROC curve. The following DATA step combines the predictions from each model with the observed diagnosis in the test set:

```

data rocdata;
  merge mycas.LogisticScore
        mycas.mbcScore;
;

```

The following statements use PROC LOGISTIC to produce the ROC curves for the model-based clustering model and the logistic model:

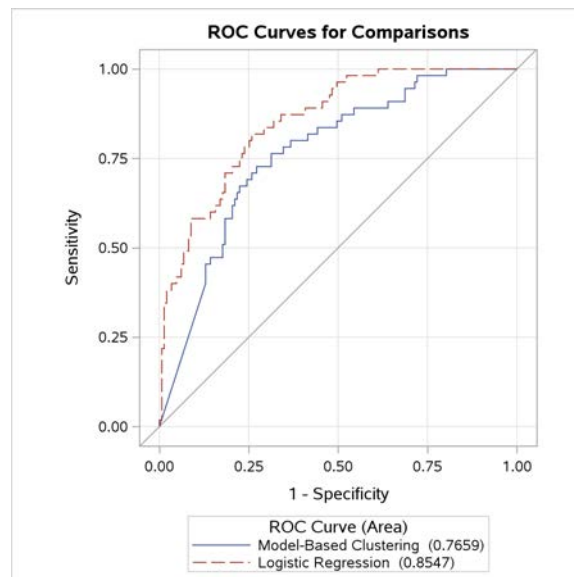
```

proc logistic data=rocdata;
  model diabetes(event='1') = predict wt1 / nofit;
  roc 'Model-Based Clustering' pred=wt1;
  roc 'Logistic Regression' pred=predict;
run;

```

[Figure 13](#) shows the ROC curves.

**Figure 13** ROC Curves for Logistic Regression and Model-Based Clustering



An ROC curve that is closer to the upper left corner indicates a model that has higher predictive ability. It is not surprising that the logistic regression performs better than the model-based clustering, because the logistic regression has more information in the form of the known diagnosis. However, the model-based clustering is able to find structure

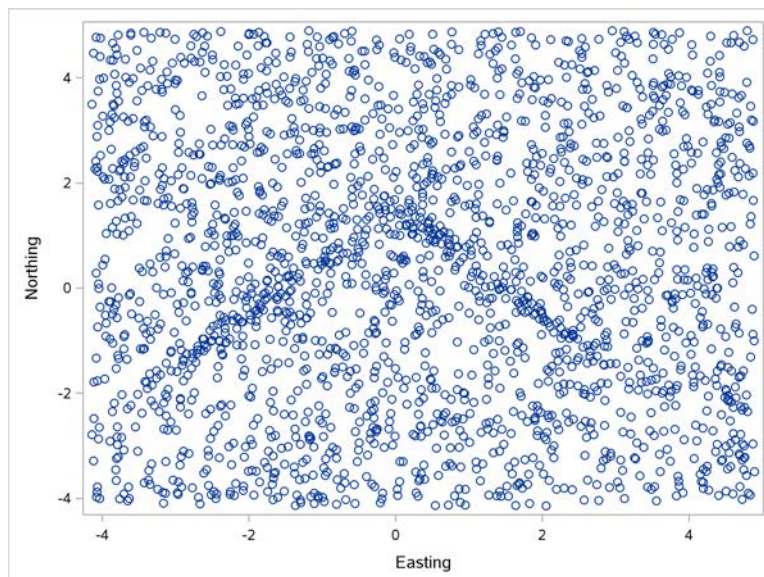
in the data without using this additional information. Because the explanatory variables in the data set are indeed risk factors for diabetes, perhaps it is not too surprising that you can discover structure even without the diagnosis information. Nonetheless, it is encouraging that the model-based clustering technique can at least find meaningful and interpretable clusters for observations. In this case, you can interpret the clusters as diabetic and nondiabetic groups. In a more general setting, model-based clustering might still enable you to develop a good model to classify observations into different groups, but your interpretation of the clusters would depend on additional knowledge and insight.

If you assign meaning to the clusters on the basis of your interpretation of the clustering variables and their relationship to an outcome of interest, such as disease status, the posterior weights can provide a sense of where further investigation might be warranted. In this example, if an individual does not have a convincingly high posterior weight for either the “disease” cluster or the “no disease” cluster, you might follow a different course of treatment than if the individual had a strong affinity for one of the clusters. That is, you might be more inclined to direct an intervention that regards an individual who has a borderline posterior weight as a potential diabetic, because the consequences of treating the person as a nondiabetic could be more serious.

## EXAMPLE 2: QUIETING THE NOISE

The examples in the introduction demonstrate how the shapes of the individual multivariate normal components allow the mixture to flexibly accommodate many clustering scenarios. However, if some of the observations do not conform to the assumption that the components are multivariate normal, your model might not fit well. Figure 14 shows a hypothetical data set that reveals the locations of mass concentrations over a specific level at the same depth under a plot of land. In this example, your goal is to identify regions that have a cohesive structure, indicating promising areas for further exploration.

Figure 14 Locations of Mass Concentrations



The following statements fit separate models that have one to ten multivariate normal components and each of the nine covariance structures. PROC MBC chooses the model that has the best (smallest) BIC value. You specify the INIT=KMEANS option to use the  $k$ -means method to determine initial cluster centers for each model. The OUTPUT statement saves the cluster weights and assignments in the CAS table **mycas.GoldScore**, and it adds the analysis variables **Easting** and **Northing** to the output CAS table. You use the MAXPOST option in the OUTPUT statement to include the number of the component for which each observation has the highest posterior probability.

```
proc mbc data=mycas.GoldMine
  init=kmeans
  seed=72145225
  covstruct=(EEE EEI EEV EII EVI EVV VII VVI VVV)
  nclusters=(1 to 10);
  var easting northing;
```

```

output out=mycas.GoldScore copyvars=(easting northing) maxpost;
run;

```

Figure 15 shows the trend in BIC with an increasing number of Gaussian clusters, which are grouped by covariance structure. In this case, adding more components under each covariance structure usually improves the fit as measured by the BIC.

**Figure 15** BIC and Number of Gaussian Clusters

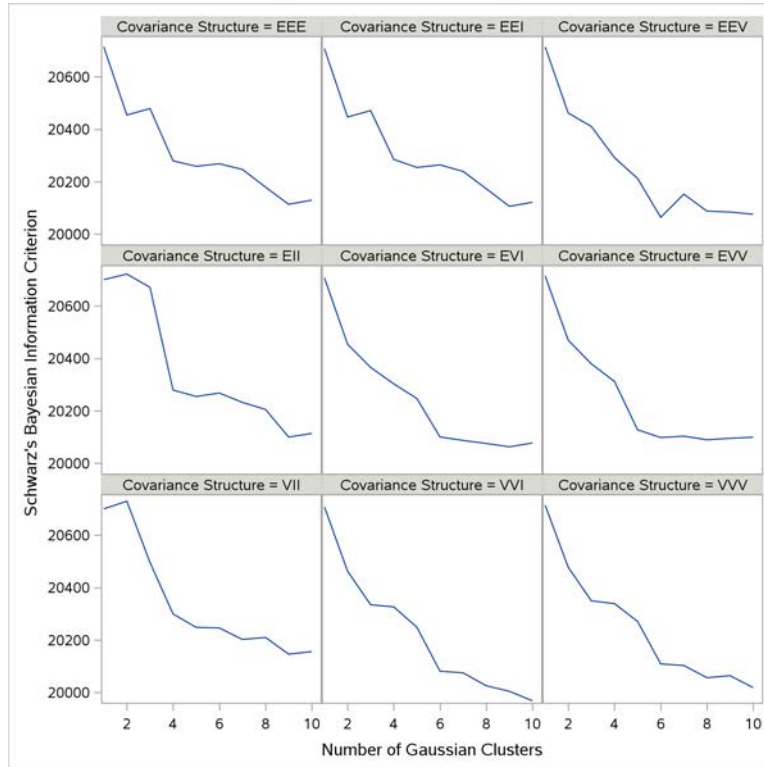


Figure 16 shows the 10 models that have the best (smallest) BIC values, in ascending order of BIC values. The model with 10 Gaussian clusters that uses the VVI covariance structure has the best BIC value, so it is chosen as the best clustering model in this analysis.

**Figure 16** Top 10 Models  
The MBC Procedure

Model Selection Summary							
Covariance Structure	Number of Clusters	Noise Component	Number of Parameters	-2 Log L	AIC	AICC	BIC
VVI	10	N	49	19591	19689	19691	19968
VVI	9	N	44	19666	19754	19756	20005
VVV	10	N	59	19564	19682	19686	20019
VVI	8	N	39	19726	19804	19805	20026
VVV	8	N	47	19695	19789	19791	20056
EVI	9	N	36	19786	19858	19859	20063
EEV	6	N	25	19872	19922	19922	20064
VVV	9	N	53	19656	19762	19765	20064
VVI	7	N	34	19813	19881	19882	20075
EVI	8	N	32	19829	19893	19894	20076

The mixing estimates, shown in Figure 17, indicate one cluster (cluster 7) that has a larger weight and nine other clusters that have smaller weights.

**Figure 17** Mixing Estimates for Selected Model

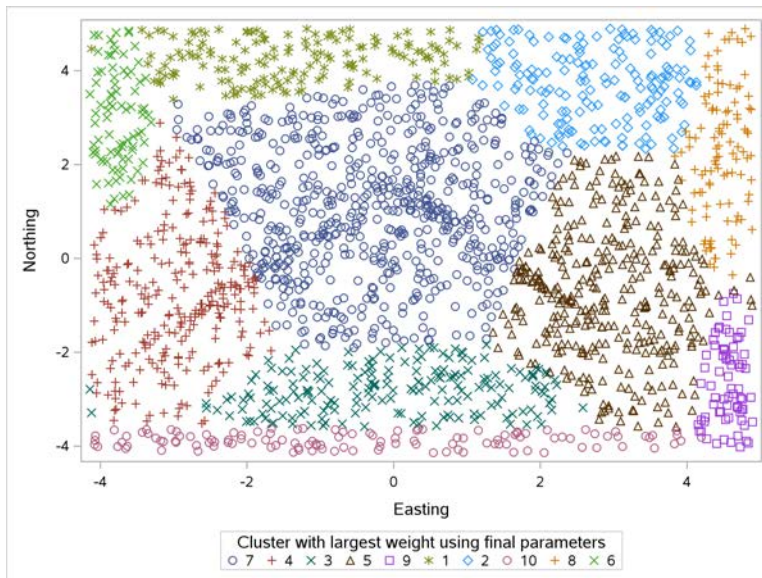
Cluster Mixing Probability Estimates	
Mixing Component	Mixing Probability
1	0.06816
2	0.07744
3	0.09614
4	0.12766
5	0.15339
6	0.03176
7	0.32809
8	0.04289
9	0.02929
10	0.04518

The following statements plot the clustering by using different plot symbols and colors to indicate the different clusters:

```
proc sgplot data=mycas.GoldScore;
  scatter x=easting y=northing / group=maxpost;
  xaxis label='Easting'; yaxis label='Northing';
run;
quit;
```

Figure 18 shows that the cluster assignments in the **MAXPOST** output variable do not identify any useful structure. There are regions of higher density that are split across clusters, and many of the points seem like background noise.

**Figure 18** Cluster Assignments



PROC MBC can accommodate this scenario by adding a uniform noise component. You can use the **NOISE=Y** option in the PROC MBC statement to include a noise component in each model that the procedure fits. In this case, you want to fit each model with and without a noise component, and the **NOISE=(Y N)** option does this. The **NEXTCLUS=WT** option in the OUTPUT statement records the clustering weights in variables labeled **WT $k$** , where  $k$  indicates the cluster. These clustering weights are produced using the final parameter estimates, so they are the “next” clustering. You can also use the **CURRCLUS=** option to include the clustering weights that produce the final parameter estimates. The following statements fit models that have one to ten Gaussian components and the noise component:

```

proc mbc data=mycas.GoldMine
  init=kmeans
  seed=72145225
  nclusters=(1 to 10)
  covstruct=(EEE EEI EEV EII EVI EVV VII VVI VVV)
  noise=(y n);
  vars easting northing;
  output out=mycas.GoldNoise copyvars=(easting northing) maxpost nextclus=wt;
run;

```

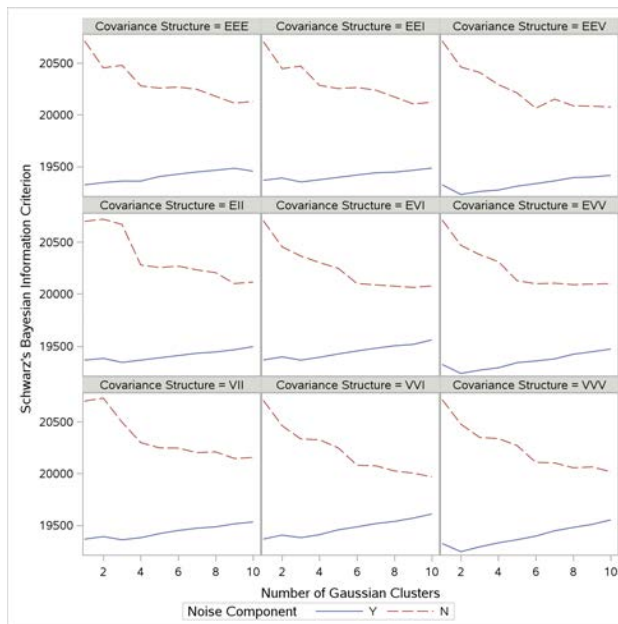
Figure 19 summarizes the top 10 models for this case. The best fit is a model that has a noise component and two Gaussian components that use the EEV covariance structure.

**Figure 19** Summary of Fit Statistics  
The MBC Procedure

Model Selection Summary							
Covariance Structure	Number of Clusters	Noise Component	Number of Parameters	-2 Log L	AIC	AICC	BIC
EEV	2	Y	11	19146	19168	19168	19231
EVV	2	Y	12	19146	19170	19170	19238
VVV	2	Y	13	19146	19172	19172	19246
EEV	3	Y	15	19144	19174	19174	19260
EVV	3	Y	17	19139	19173	19173	19269
EEV	4	Y	19	19127	19165	19166	19274
VVV	3	Y	19	19146	19184	19184	19292
EVV	4	Y	22	19123	19167	19167	19292
EEV	5	Y	23	19134	19180	19181	19311
EVV	1	Y	7	19270	19284	19284	19324

Figure 20 compares models with and without a noise component, which are grouped by covariance structure. Adding the noise component improves the fit for each covariance structure and each Gaussian cluster count. Figure 21 compares only models that have a noise component. In the models with a noise component where the covariance structure does not restrict the orientation of the clusters (structures EEV, EVV, and VVV), adding more than two Gaussian clusters does not improve the fit.

**Figure 20** BIC and Number of Gaussian Clusters



**Figure 21** Models with Noise Component

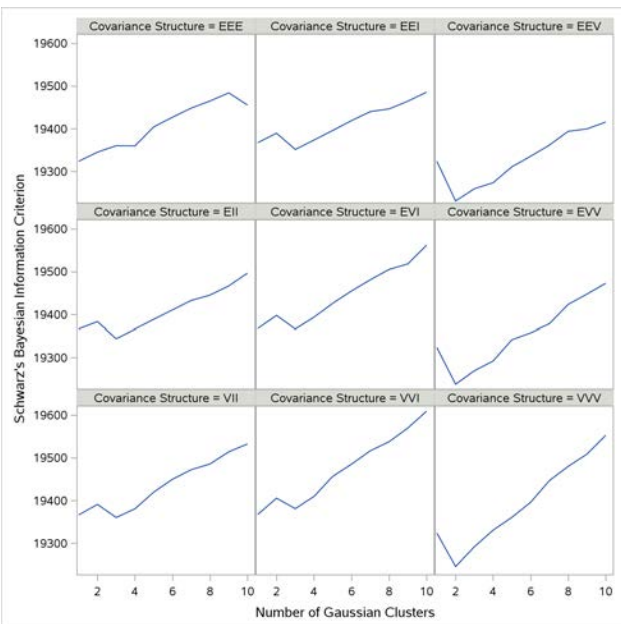


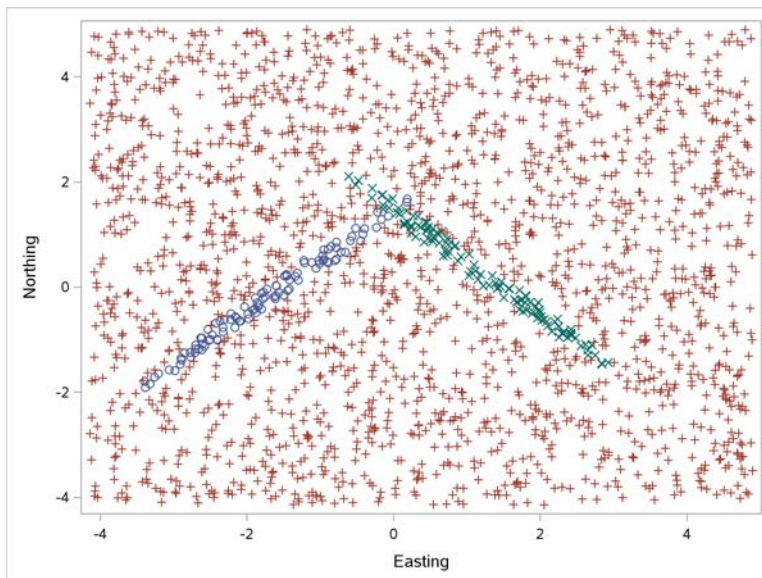
Figure 22 shows the “Mixing Estimates” table. By convention, the noise component is indexed as component 0. The two multivariate Gaussian components have approximately equal weights, and the noise component weight is much larger than the weight of the normal components.

**Figure 22** Mixing Estimates for Selected Model

Cluster Mixing Probability Estimates	
Mixing Component	Mixing Probability
0	0.90797
1	0.04318
2	0.04885

Figure 23 plots each observation with a distinct color and symbol according to the component for which it has the largest posterior probability; this is stored in the **MAXPOST** variable in the **mycas.GoldNoise** output data set. Adding the noise component allows the denser clusters to emerge. In this example, these denser areas of organization are the regions of interest.

**Figure 23** Cluster Assignments from Model Including Noise Component

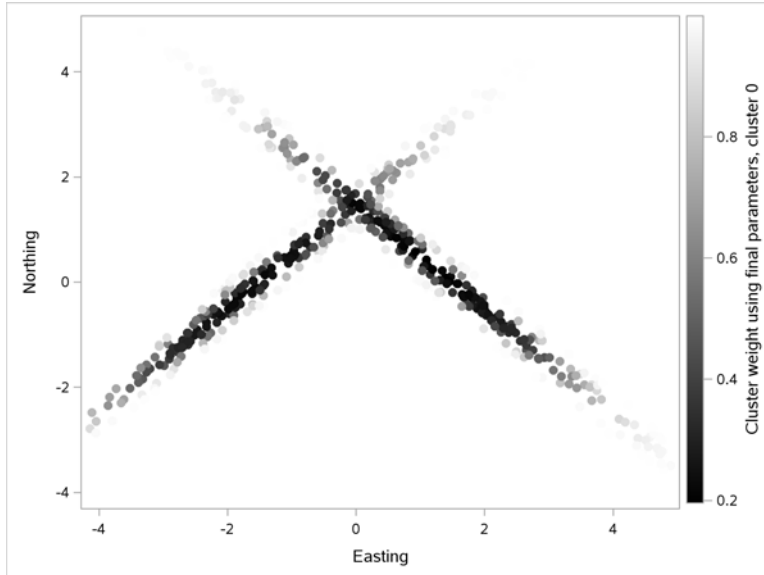


The output data set **mycas.GoldNoise** also contains the posterior weights for each observation and each component in the **WT0**, **WT1**, and **WT2** variables.

In Figure 24, the color of each plot point reflects the value of the **WT0** variable, which is that observation’s posterior weight for the noise cluster. By using a color ramp that colors the “noisiest” points—those with the highest value for **WT0**—to match the background color, you can see the regions that are assigned to the Gaussian components more clearly. The black and darker gray areas indicate regions that are not strongly associated with the noise component, which means that they are more strongly associated with the Gaussian components.



**Figure 24** Clustering Weights Using Color Ramp

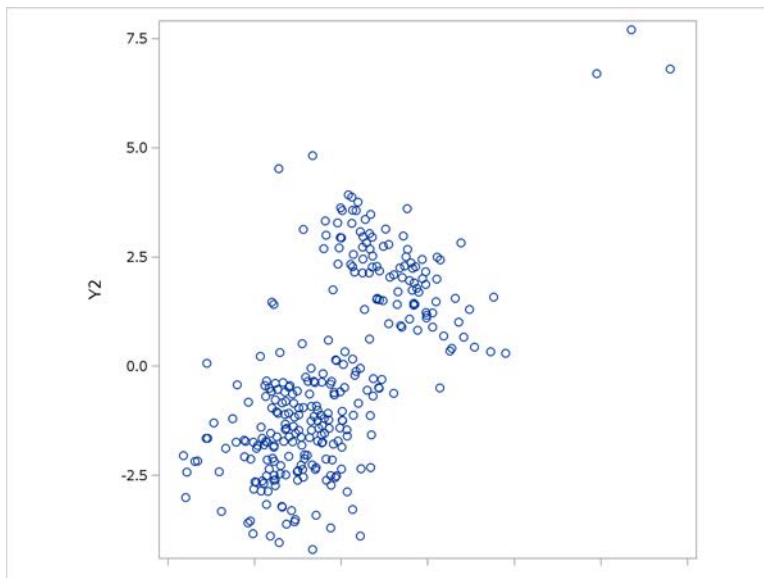


### EXAMPLE 3: OUTLIERS

In Example 2, you see how a noise component can improve clustering when many observations do not appear to have a centrally organized structure. The current example shows a complementary situation, where the model uses a noise component to address a small number of outliers. The noise component can reduce distortion in clusters by providing a component in the model for observations that do not fit well with the Gaussian clusters in the mixture.

Figure 25 plots a data set that has two apparent clusters and three outlying points that are not clearly associated with either cluster.

**Figure 25** Outlier Scenario



A first step is to model this scenario by using only Gaussian clusters in PROC MBC, as in the following statements. The plot suggests that three clusters are plausible, so you can specify NCLUSTERS=3. You can use PROC MBC to evaluate several different three-cluster models by specifying multiple values for the COVSTRUCT option. The INIT=KMEANS option specifies that the procedure use the result of the  $k$ -means method with three clusters to initialize each fit. The OUTPUT statement names the **mycas.c3out** output data set. The MAXPOST option identifies the

cluster that has the maximum clustering weight, and the COPYVARS=(ALL) option copies all the variables from the input data set to the output data set.

```
proc mbc data=mycas.sphc
  nclusters=3
  covstruct=(EEE EEI EEV EII EVI EVV VII VVI VVV)
  init=kmeans
  seed=12;
  vars y1 y2;
  output out=mycas.c3out maxpost copyvars=(all);
run;
```

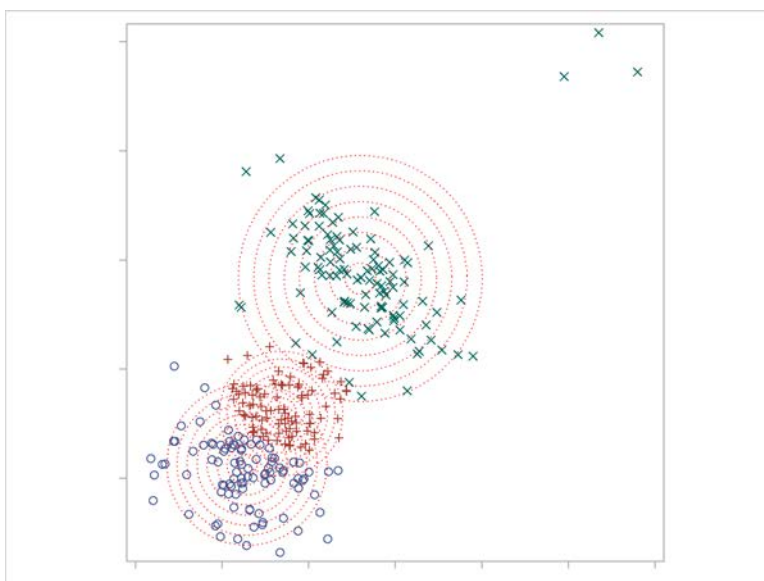
Figure 26 shows the 10 best models that the model selection process finds.

**Figure 26** Top 10 Models  
The MBC Procedure

Model Selection Summary							
Covariance Structure	Number of Clusters	Noise Component	Number of Parameters	-2 Log L	AIC	AICC	BIC
VII	3	N	11	2196.20047	2218.20047	2219.10768	2259.05153
VVI	3	N	14	2195.07446	2223.07446	2224.53280	2275.06672
VVV	3	N	17	2189.23454	2223.23454	2225.38191	2286.36800
EII	3	N	9	2247.50424	2265.50424	2266.11857	2298.92783
EEE	3	N	11	2236.35612	2258.35612	2259.26334	2299.20718
EVV	3	N	15	2218.26381	2248.26381	2249.93628	2303.96980
EEI	3	N	10	2247.20666	2267.20666	2267.96009	2304.34399
EEV	3	N	13	2235.33251	2261.33251	2262.59203	2309.61104
EVI	3	N	12	2246.05666	2270.05666	2271.13252	2314.62145

The selected model has three clusters and the VII covariance structure restriction. This means that the three clusters are isotropic (circular) and have different volumes. In the two-dimensional setting, it can be easier to visualize the result of the model selection by including features of the Gaussian clusters in the plot. In Figure 27, different symbols and colors identify the different cluster assignments. The plot also overlays contours of constant Mahalanobis distance for each cluster, centered at the respective cluster's estimated mean. The overlays illustrate the circular structure and the different volumes that correspond to the VII structure.

**Figure 27** Selected Three-Component Model with Contours



The three points that are more distant from the others are clustered with other points, rather than being assigned to their own cluster. The other points in that cluster suggest a shape that is not circular, indicating that the presence of the three outliers is distorting the clustering. In addition, the requirement for three clusters seems to have split a cohesive group of points in the lower left corner of the plot into two clusters. The interplay between the three more distant points and the model selection is complex, but the three points do not support a stand-alone cluster, and including them in one of the well-defined clusters distorts that cluster.

In this case, a noise component can accommodate these points and might yield Gaussian clusters that provide a better fit to the data. The following statements introduce a noise component and also consider models that have two or three Gaussian clusters:

```
proc mbc data=mycas.sphc
  nclusters=(2 3)
  covstruct=(EEE EEI EEV EII EVI EVV VII VVI VVV)
  noise=(n y)
  init=kmeans
  seed=12;
  vars y1 y2;
run;
```

Figure 28 shows summary information about the selected model. The “Cluster Mixing Probability Estimates” table indicates that the noise component (component 0) has a very low mixing probability, indicating that very few points have an association with this component. This is consistent with the small number of outlier points.

**Figure 28** Selected Model Information

**The MBC Procedure**

Model Information				
Number of Gaussian Clusters		2		
Covariance Structure		VVV		
Noise Cluster Present		Yes		
Expectation Technique		EM		
Model Selection Criterion		BIC		
Initialization Method		K-means		
EM Convergence Criterion		1e-05		
Singularity Criterion		1e-08		
Parameter Criterion		1e-08		
Random Seed		12		

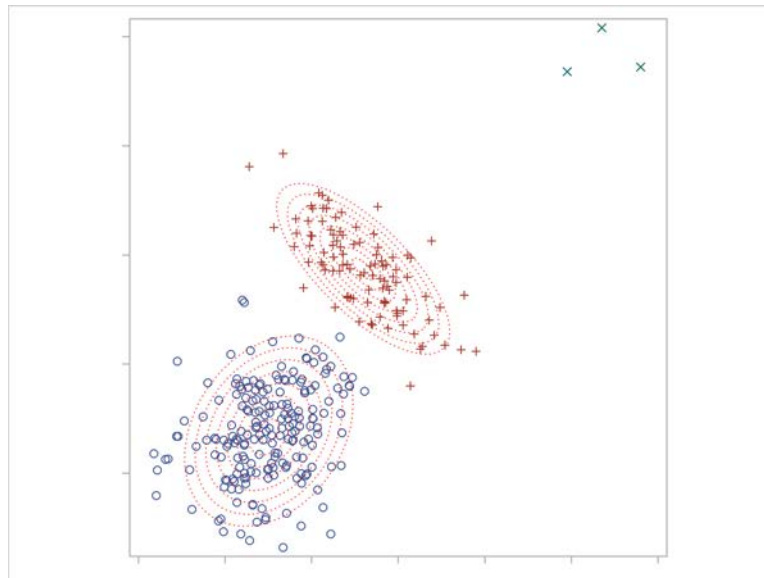
Cluster Parameter Estimates				
		Covariance		
Cluster	Variable	Mean	Y1	Y2
1	Y1	-0.99018	0.94408	0.30129
	Y2	-1.53153		1.18427
2	Y1	1.18948	0.99755	-0.71150
	Y2	2.18134		0.95314

Cluster Mixing Probability Estimates	
Mixing Component	Mixing Probability
0	0.01954
1	0.65516
2	0.32530

Figure 29 shows the posterior assignments for all observations and contours for the Gaussian clusters. In this model, the outlying points are assigned to the noise component. The Gaussian clusters provide a better representation of the remaining observations and are not distorted by the outliers. The covariance structure in the selected model is VVV, meaning that the Gaussian clusters are not constrained to share any characteristics, and the contours indicate that the clusters match the observed shapes well.

**Figure 29** Selected Model with Contours



In this two-dimensional scenario, it is not difficult to identify the observations that do not appear to be drawn from the same distribution as the others. In higher dimensions, it can be much more challenging to identify these outliers by inspection, and in that setting model-based clustering can be very useful in addressing the outlier issue.

## CONCLUSION

PROC MBC provides an unsupervised model-based clustering method that you can use to discover groups. It expands typical notions of clustering by using a finite mixture model whose components represent the clusters. Because model-based clustering is based on a formal statistical model, you can apply the statistical principles of likelihood and information criteria to compare and select mixture models for the data. The model likelihood also permits you to make model-based computations of cluster membership for each observation and each cluster in the model. In this paper, you see how to use PROC MBC to model different scenarios by using mixtures of multivariate Gaussian distributions and how to score additional data sets to find cluster memberships for new observations. In addition, PROC MBC can include a noise component in the mixture model to accommodate observations that are not well modeled by Gaussian components. This noise component can also help you to model outliers or to discover structure within noisy data.

## REFERENCES

- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society, Series B* 39:1–38.
- Fraley, C., and Raftery, A. E. (1998). *How Many Clusters? Which Clustering Method? Answers via Model-Based Cluster Analysis*. Technical report 329, Department of Statistics, University of Washington, Seattle.
- Fraley, C., and Raftery, A. E. (2007). "Bayesian Regularization for Normal Mixture Estimation and Model-Based Clustering." *Journal of Classification* 24:155–181.
- MacQueen, J. B. (1967). "Some Methods for Classification and Analysis of Multivariate Observations." *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* 1:281–297.

McLachlan, G. J., and Peel, D. (2000). *Finite Mixture Models*. New York: John Wiley & Sons.

McNicholas, P. D. (2017). *Mixture Model-Based Classification*. Boca Raton, FL: CRC Press.

Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., and Johannes, R. S. (1988). "Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus." In *Proceedings of the Symposium on Computer Applications and Medical Care*, 261–265. Los Alamitos, CA: IEEE Computer Society Press.

## **ACKNOWLEDGMENTS**

I am grateful to Bob Rodriguez, formerly of the Advanced Analytics Division at SAS, for his support in the development of the MBC procedure. I wish to thank Yiu-Fai Yung, Gerardo Hurtado, David Schlotzhauer, John Castelloe, Randy Tobias, Liz Edwards, Sumi Rasathurai, and Ed Huddleston for their helpful comments, all of which greatly improved the quality of this paper.

## **CONTACT INFORMATION**

Your comments and questions are valued and encouraged. Contact the author:

Dave Kessler  
SAS Institute Inc.  
SAS Campus Drive  
Cary, NC 27513  
dave.kessler@sas.com

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.

Other brand and product names are trademarks of their respective companies.