Paper 3409-2019

How to Evaluate Different Clustering Results

Ralph Abbey, SAS Institute Inc.

ABSTRACT

A wide variety of clustering algorithms are available, and there are numerous possibilities for evaluating clustering solutions against a gold standard. The choice of a suitable clustering algorithm and of a suitable measure for the evaluation depends on the data type; whether separate class label information exists (supervised clustering); and on the particular distribution of the observations, including characteristics such as the number of clusters, separability of the clusters, and the shape, size, and density of the clusters. This paper provides a survey of some of the most widely used clustering evaluation criteria. In addition, the paper describes recently developed criteria that are applicable for mixed interval-categorical data and for non-Euclidean distance metrics. Notable examples of the methods covered include residual sum-of-squares, purity, the silhouette measure, the Calinski-Harabasz measure, class-based precision and recall, the normalized mutual information, variation of information, and graph-sensitive indices.

INTRODUCTION

Clustering is the task of segmenting a data set into groups. The goal is to ensure that similar data are clustered together, while dissimilar data are in different clusters. Over the years, many methods for clustering have been proposed.

With many clustering methods generally available, a natural question is "How do I compare two different clustering results?" Many measures exist that compare clustering results, but these measures have different use cases, required assumptions, benefits, and downsides.

This paper gives you a broad overview of many popular clustering methods as well as many popular cluster evaluation measures. This paper presents the clustering methods and evaluation measures as a survey with citations so that you can further investigate the details if you desire. In addition, this paper provides a synthesis and offers some best-practice advice for cluster evaluation.

This paper has three main sections: Clustering Methods, Clustering Measures, and Clustering Evaluation. The Clustering Methods section describes popular clustering methods and the section contains background material for understanding how different cluster evaluation metrics apply to different methods. The Clustering Measures section describes many popular cluster evaluation metrics, including when these metrics are applicable. The Clustering Evaluation section synthesizes the information contained in the Clustering Methods and Clustering Metrics sections to provide general best practice advice.

CLUSTERING METHODS

This section contains an overview of many popular clustering approaches. This section serves as high-level background information, introducing many algorithms for clustering that you might encounter while learning about cluster analysis.

HIERARCHICAL CLUSTERING

Hierarchical clustering is a broad clustering method with multiple clustering strategies. Alternatively, you can think of hierarchical clustering as a class of clustering methods that all share a similar approach. For hierarchical clustering there are two main approaches: agglomerative and divisive.

Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering is a bottom-up approach in which each datum is initially individually grouped. Two groups are merged at a time in a recursive manner. The groups are merged until a stopping condition is met. In SAS® you can use agglomerative hierarchical clustering by using the CLUSTER procedure.

The process for merging two groups in agglomerative hierarchical clustering involves calculating the distance between the groups, and then choosing the two closest groups to merge. Table 1 contains some examples of how the distance between clusters is calculated. For a complete list of supported methods, you can see the PROC CLUSTER documentation.

Names	Description
Average Linkage	Average distance between all pairs of data between the two clusters.
Centroid Linkage	Distance between the centroids of the clusters.
Complete Linkage	Maximum distance between all pairs of data between the two clusters.
Single Linkage	Minimum distance between all pairs of data between the two clusters.

Table 1. Examples of Hierarchical Clustering Linkages and Descriptions

A common practice when using agglomerative hierarchical clustering is to visualize the clustering results as a dendrogram. Figure 1 is an example of a dendrogram that is generated by the CLUSTER procedure.

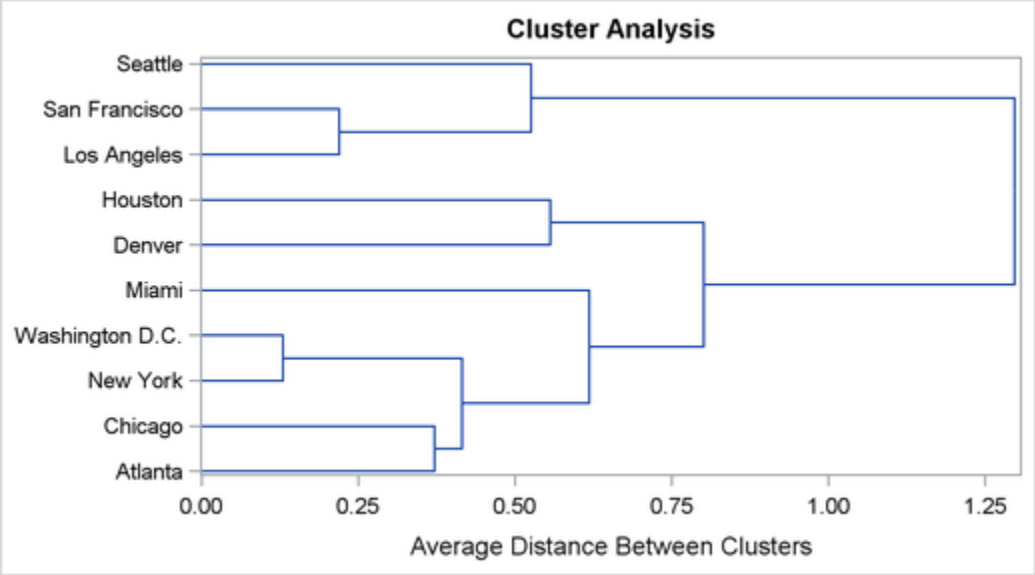

Figure 1. Example Dendrogram Output from PROC CLUSTER

To assign a new data point to an existing cluster, you can compute the distance from the new data to each existing cluster. You can compute this using the distance method originally used during cluster generation.

Divisive Hierarchical Clustering

Divisive hierarchical clustering is a top-down approach in which the entire data set is initially grouped. The data set is then split into subsets, which are each further split. This process occurs recursively until a stopping condition is met.

To assign a new data point to an existing cluster in divisive hierarchical clustering, you proceed through the divisive steps that were taken in the original clustering process. Assigning a new data point is similar to following a set of rules, though in this case the rules are the process by which the data was divided at each step of the clustering process.

Two well-known divisive hierarchical clustering methods are Bisecting K-means (Karypis and Kumar and Steinbach 2000) and Principal Direction Divisive Partitioning (Boley 1998). You can achieve both methods by using existing SAS procedures and the DATA step. Such an analysis, however, is outside of the scope of this paper.

CENTROID-BASED CLUSTERING

Centroid-based clustering is most well-known through the k-means algorithm (Forgy 1965 and MacQueen 1967). For centroid-based methods, the defining characteristic is that each cluster is defined by the "centroid", the average of all the data points in the cluster. In SAS you can use centroid-based clustering by using the FASTCLUS procedure, the HPCLUS procedure, or the KCLUS procedure in SAS® Viya®.

To assign a new data point to an existing cluster, you first compute the distance between the data point and each centroid. The centroid that has the minimum distance to the new data point indicates which cluster the data point belongs.

DISTRIBUTION-BASED CLUSTERING

Distribution-based clustering is an approach where the data are assumed to have come from multiple statistical distributions. Distribution-based clustering supports soft-clustering, where a data point has partial membership in multiple clusters, or a hard clustering, where a data point is assigned to only one cluster. In distribution-based clustering, each cluster represents a distribution. In hard clustering, the data is assigned to the cluster whose distribution is most likely the originator of the data. In SAS you can use distribution-based clustering by using the GMM procedure in SAS Viya. Also, the mbcFit and mbcScore actions in SAS Viya perform model based clustering using mixtures of multivariate Gaussians.

To assign a new data point to an existing cluster, you calculate how likely it is for the new data point to belong to each distribution. This can give either a soft clustering for the new data point, or the data point can be assigned to the cluster that has the most likely distribution from which the data point originated.

You can approach distribution-based clustering in another way using finite mixture models. In SAS you can use finite mixture models by using the FMM procedure. You can find more information about how finite mixture models relate to clustering in the FMM procedure documentation.

RULE-BASED METHODS

Rule-based clustering methods encompass a rule or set of rules that lead to a clustering of the data. A simple example is grouping data by geographic location such as city, county, or state. Often, rules of this form come from specific business use cases, as opposed to the optimization of an unsupervised learning algorithm.

You can generate more complicated rule-based clusters by using a decision tree with a target. The leaf nodes of the decision tree are then used as clusters of the data. When using

a decision tree to derive clusters, you should use shallow trees to avoid generating too many clusters.

A shallow decision tree with a target can give a powerful rule-based clustering. However, the optimization of this cluster is then based on the target as opposed to an unsupervised learning problem. In SAS you can generate decision trees by using the ARBOR procedure, the HPSPLIT procedure, or the TREESPLIT procedure in SAS Viya.

To assign a new data point to an existing cluster, you apply the rules in the rule-based model.

SPECTRAL-BASED METHODS

Spectral methods (von Luxburg 2007) rely on the spectrum (eigenvalues and eigenvectors) of a matrix. This matrix is often a similarity matrix, in which the entries are the similarities between the observations. The reciprocal of the values of a distance matrix is one example of a similarity matrix.

Once the set (or a partial set) of eigenvalues and eigenvectors is obtained, more traditional methods such as k-means are applied. Spectral clustering enjoys popularity because it blends density-based approaches by using the similarity matrix to centroid-based approaches.

A major difficulty with spectral-based methods is that it is not feasible to assign a new data point to a cluster. To do so, you need to compute the similarity of the new data point to the training data, and then do a projection into space of eigenvectors that was created during the clustering process.

Overall, while spectral-based methods are popular for data exploration purposes, these methods are not suitable for assigning new data points.

In SAS you can achieve some spectral-based clustering methods by using a mix of the DATA step, the PRINCOMP procedure, and one of the centroid-based clustering procedures in the Centroid-Based Clustering section. Such an analysis, however, is outside of the scope of this paper.

CLUSTERING MEASURES

This section contains an overview of many methods for cluster evaluation. This section serves as high-level background information, introducing you to many popular evaluation techniques that you might encounter in the literature for cluster analysis.

RESIDUAL SUM-OF-SQUARES

Also known as the sum of squared errors (SSE), the residual sum-of-squares measure is often applied to regression problems. In clustering contexts this refers to the sum of squared differences between each data point and the centroid of the cluster where the data point belongs. This can also be called the total within sum of squared errors.

SSE is calculated with the following equation:

$$SSE = \sum_{j=1}^{k} \sum_{x \in C_j} ||x - \overline{c}_j||^2$$

Equation 1 Sum of Squared Errors (SSE)

The SSE is a useful measure because for algorithms such as k-means, it is the same measure by which the clustering is optimized. Computing the SSE is also quick, as there are a total number of differences equal to the number of data points.

SSE as a cluster evaluation measure only applies to methods in which the cluster can be represented by the centroid. Using this measure with clusters derived from other methods can offer misleading insights into the efficacy of the clustering. Another often cited downside to SSE is that as the number of clusters increase, the SSE decreases. This means that SSE is not suitable for comparing clustering results with different numbers of clusters.

SILHOUETTE

The silhouette method provides a measure of how similar the data is to the assigned cluster as compared to other clusters. This is computed by calculating the silhouette value for each data point, and then averaging the result across the entire data set.

To compute the silhouette value for a single data point you need to compute the average distance between a data point and all clusters. The average distance between a data point and all other data points within a cluster is calculated with the following equation:

$$average\ distance = \frac{1}{n_c} \sum_{x \in C_j} ||x - d||$$

The silhouette value for a single data point is calculated with the following equation:

$$s = \frac{AverageOut - AverageIn}{\max(AverageOut, AverageIn)}$$

In this equation, *AverageOut* is the minimum average distance between the data point and data within other clusters, and *AverageIn* is the average distance between the data point and other data within the same cluster. The silhouette measure is the average of all the silhouette values computed for each data point.

The silhouette measure is restricted to the range of [-1, 1], with -1 meaning that no data are well suited to their assigned clusters, and 1 meaning that all data are well suited to their assigned clusters.

The silhouette measure is useful because it considers distance to other clusters in addition to the distance within the cluster. Because of this comparison, the silhouette measure is suitable for comparing clustering results that contain different numbers of clusters. If there are too many or too few clusters, the silhouette measure will be closer to zero than if an appropriate number of clusters is chosen.

Like SSE, the silhouette measure applies best to centroid-based clustering methods. This is because rule-based methods or hierarchical clustering methods don't seek to minimize distances in the same way that centroid-based cluster methods do. Because of this, methods that are not centroid-based are not appropriately described by the silhouette measure. An additional area of concern is that the silhouette measure takes much longer to compute than SSE. The total of differences is equal to the square of the number of total data points.

CALINSKI-HARABASZ INDEX

The Calinski-Harabasz index is similar to the F Statistic used in ANOVA. The F Statistic in the F Test is calculated with the following equation:

$$F = \frac{SS_{between}/(k-1)}{SS_{within}/(N-k)}$$

In the case of centroid-based clustering the sum of squares within is the same as the sum of squared errors from Equation 1. The sum of squares between clusters is calculated with the following equation:

$$SS_{between} = TSS - SS_{within}$$

TSS, the total sum of squares, is the total distance between each data point and the mean of the data:

$$TSS = \sum_{i=1}^{n} ||x_i - \overline{x}||$$

PURITY

To evaluate the purity of a clustering, you need a categorical target that was not used in the clustering process. This is called an external evaluation measure, as the categorical target is a variable that is external to the clustering process.

Purity, and the other following measures are measures of how closely two cluster results are to each other. When you use a categorical target to calculate these measures, you assume that there is a benchmark clustering in which each category of the target is in its own cluster.

The purity of a cluster is determined by assigning the cluster the category that is most frequent in the cluster, and then computing the accuracy of this assignment. The purity is calculated with the following equation:

$$purity = \frac{1}{N}\sum_{k} max_{t \in T}(c_k \cap t)$$

In this equation $t \in T$ indicates a target category, t, in the set of all target categories, T.

External evaluation measures are useful when comparing different types of clustering methods that might not both be well suited to distance-based clustering evaluation measures.

The purity measure doesn't penalize having too many clusters and is also poorly suited for imbalanced target category size.

CLASS-BASED PRECISION AND RECALL

Class-based precision and recall are, like purity, a form of external evaluation using a categorical target. When you use a categorical target to calculate this measure, you assume that there is a benchmark clustering in which each category of the target is in its own cluster. Similar to the purity measure, precision and recall require assigning each cluster a category of the class target that is most frequent in the cluster.

Precision and recall are often seen in binary classification problems. In the case of clustering, class-based precision and recall work on an underlying assumption that there is a benchmark classification. The clustering obtained is the compared to the benchmark.

The equation for precision is based on the number of True Positives (TP) and False Positives (FP):

$$P = \frac{TP}{TP + FP}$$

The equation for recall is based on the number of True Positives (TP) and False Negatives (FN):

$$R = \frac{TP}{TP + FN}$$

In the clustering context the number of true positives is the number of the same category that are clustered together. The number of false positives is the number of the different categories that are clustered together. The number of false negatives is the number of the same category that are clustered in different clusters.

NORMALIZED MUTUAL INFORMATION

Mutual information in probability theory describes the mutual dependence of two random variables. The mutual information is the amount of information that you can obtain for one random variable by observing the other random variable.

Mutual information relies on an external categorical target. When you use a categorical target to calculate this measure, you assume that there is a benchmark clustering in which each category of the target is in its own cluster.

For clustering evaluation, normalized mutual information also requires an external categorical target. The mutual information of a clustering is calculated with the following equation:

$$MI = \sum_{j=1}^{k} \sum_{t \in T} P(c_j \cap t) log \frac{P(c_j \cap t)}{P(c_j)P(t)}$$

In this equation $t \in T$ indicates a target category, *t*, in the set of all target categories, *T*. The probability $P(c_j \cap t)$ is number of category *t* in cluster j, divided by the total number of data points. The probability $P(c_j)$ is the total number of data points in cluster j divided by the total number of data points. The probability $P(t)$ is the total number of category *t* divided by the total number of data points.

The minimum mutual information is zero if the clustering is random with respect to the categorical target. Mutual information suffers from the same problem as purity in that the mutual information is maximized when each data point is in its own cluster. This is the motivation behind normalized mutual information. The normalization penalizes generating higher numbers of clusters.

The mutual information is normalized by using the entropy of the categorical targets and the entropy of the clusters. The two types of entropy are calculated with the following equations:

$$cluster\ entropy = -\sum_{j=1}^{k} P(c_j) \log(P(c_j))$$

$$categorical\ target\ entropy = -\sum_{t \in T} P(t) \log(P(t))$$

With the entropy of the clusters and the entropy of the categorical target the normalized mutual information is calculated with the following equation:

$$NMI = \frac{MI}{(cluster\ entropy + categorical\ target\ entropy)/2}$$

The normalized mutual information is always between zero and one. The value of zero has the same meaning as mutual information, that the clustering is completely random with respect to the categorical target. Because the normalization is based on cluster entropy, the normalized mutual information allows you to compare clustering results with different numbers of clusters. Similar to precision and recall, however, the normalized mutual information still favors clustering results that have one cluster per category of the target.

VARIATION OF INFORMATION

The variation of information is very similar to the mutual information, and it requires the same assumptions and existence of a categorical target.

The equation for variation of information is calculated with the following equation:

$$VI = -\sum_{j=1}^{k}\sum_{t\in T} P(c_j \cap t)[log\frac{P(c_j \cap t)}{P(c_j)} + log\frac{P(c_j \cap t)}{P(t)}]$$

In the case of variation of information, a lower number is better, as a lower value indicates a smaller distance between two clustering results. The benefit of the variation of information is that this measure obeys the triangle inequality, which has theoretical benefits.

GRAPH-SENSITIVE INDICES

Graph-sensitive indices (Hussain and Meilla 2014) are two newer approaches to comparing two clustering results. As with other external evaluation measures, you must use a categorical target as a benchmark to which you can compare your clustering results.

The two graph-sensitive indices are the random walk index (RWI) and the variation of information with neighbors (VIN). Both methods rely on first computing an adjacency matrix for the clustering. An adjacency matrix is a matrix of size n-by-n, in which the ij-th element is 1 if the data point i and data point j are clustered together.

The random walk index defines a random walk through the indices of the adjacency matrix and looks at the probability that the next index in the walk will have a specific cluster assignment and a specific target category, given the previous index's cluster assignment. The conditional probabilities that come from this random walk are similar to the variation of information, except that neighboring indices in the adjacency matrix also add to the information.

The variation of information with neighbors index also uses information about the cluster assignment of neighbors of a point. Instead of doing this by a random walk, the variation of information with neighbors measures the amount of information the cluster assignment of a neighborhood of points in the clustering gives about the target categories of the points.

For a full detail of how to calculate these graph-sensitive indices, please see the paper (Hussain and Meilla 2014).

The goal behind these two methods for cluster evaluation is to extend the notion of variation of information to include more information about the clusters. You can think of these as a more complete picture provided by the variation of information measure.

CLUSTERING EVALUATION

Table 2 summarizes the background given in the section Clustering Measures for quick viewing. This section contains some helpful best-practices for you to consider while performing clustering analysis.

Evaluation Measure	Compare Across Numbers of Clusters	Suitable for Comparing Across Clustering Methods	Requires Target Information
Residual Sum of Squares	No	No	No

Silhouette Measure	Yes	No	No
Calinski-Harabasz	Yes	No	No
Purity	No	Yes	Yes
Precision and Recall	No	Yes	Yes
Normalized Mutual Information	Yes	Yes	Yes
Variation of Information	Yes	Yes	Yes
Graph-Sensitive Indices	Yes	Yes	Yes

Table 2. Summary of Evaluation Measures

CLUSTER VISUALIZATION

One method you can use to evaluate clusters is visualizing the clustering. Often the data is multivariate, so a simple plot of all the variables is impossible. While you can look at a panel of variable-by-variable plots, it is likely unhelpful, especially when there are many variables. Instead, you might find it useful to consider one of the following methods for cluster visualization.

The CANDISC Procedure

The CANDISC procedure computes the canonical variables given a target class and creates output that you can plot using the SGPLOT procedure. If you have a data set named Cluster with variables X1, X2, X3, X4, and a cluster label _CLUSTER_ID_, then an example of using PROC CANDISC is shown in the following code:

```
proc candisc data=Cluster anova out=can;
  class _CLUSTER_ID_;
  var X1 X2 X3 X4;
  title2 'Canonical Discriminant Analysis of Iris Clusters';
run;

proc sgplot data=can;
   scatter y=Can2 x=Can1 / group=_CLUSTER_ID_;
   title3 'Plot of Canonical Variables Identified by Cluster';
run;
```

You can see example of the output from this example in Figure 2.

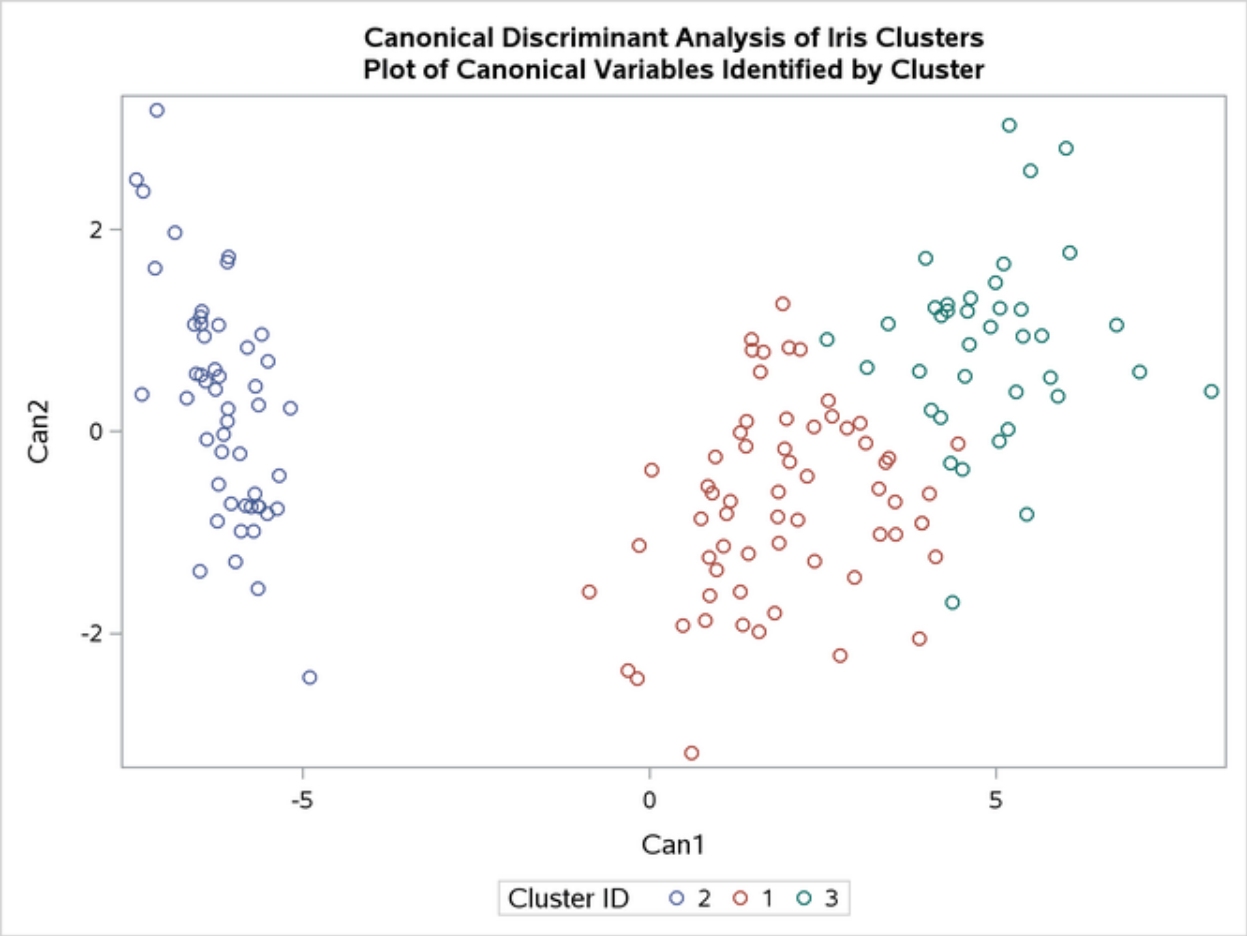

Figure 2. Example SGPLOT of Canonical Discriminant Analysis

The TSNE Procedure

The TSNE procedure in SAS Viya implements the t-distributed stochastic neighbor embedding (t-SNE) algorithm (Maaten and Hinton 2008). Unlike the canonical discriminant analysis, t-SNE computes an unsupervised embedding of the data into two or three dimensions. After performing the embedding, you can use the SGPLOT procedure with the cluster identifiers in a similar way to canonical discriminants.

One benefit for using t-SNE over canonical discriminants is that you can use the same embedding to compare multiple clustering results. By visualizing multiple clustering results in the same embedding, you gain insight that is not available through canonical discriminant analysis.

An example for running PROC TSNE is shown in the following code:

```
proc tsne data=mycas.cluster;
  input X1 X2 X3 X4;
  output out=sascas1.tsne_out copyvars=(_CLUSTER_ID_);
run;



proc sgplot data=sascas1.tsne_out;
```

```
    scatter y=_DIM_2_ x=_DIM_1_ / group=_CLUSTER_ID_;
    title1 'Scatter plot of iris embedding';
run;
```

You can see example output from this example code in Figure 3.

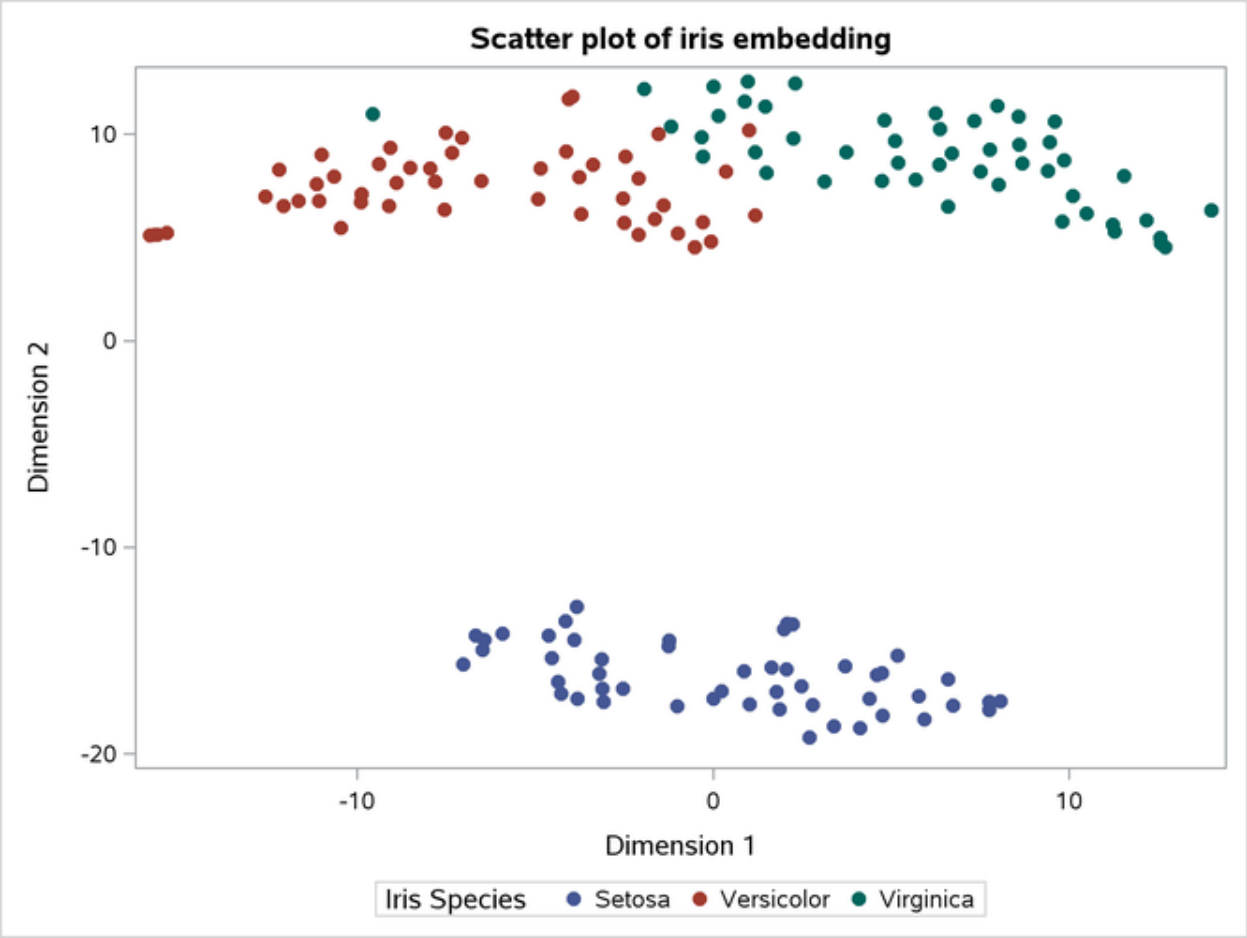

Figure 3 Example SGPLOT of the TSNE Procedure Output

For more information about the how the t-SNE algorithm works, please see the TSNE procedure documentation.

SELECTING THE APPROPRIATE CLUSTERING METHOD AND MEASURE

Before performing your cluster analysis, it is very important to consider your goal of clustering. Two broad use cases for clustering are for data exploration and for use in a predictive modeling flow, in which you want to develop a predictive model for each cluster. If you have other uses cases of clustering, you might still find the following tips helpful.

Clustering for Data Exploration

When using clustering as a data exploration tool, you might want to consider using multiple types of clustering, multiple numbers of clusters, multiple evaluation metrics, and cluster visualization. The methods in the Clustering Methods section all have different objectives that they optimize. By using multiple clustering methods, you explore your data in different ways.

For data exploration with centroid-based clustering methods, the Calinski-Harabasz measure is very useful as it allows you to explore different numbers of clusters, while still being quick to compute. The silhouette measure, on the other hand requires more computations, which can be prohibitive depending on the amount of data which you are trying to cluster.

While purity is not suitable as a measure to compare across different numbers of clusters, it is useful if you want to understand how your clustering interacts with categorical data that you might have. If you have multiple categorical variables, you should consider exploring the purity of the clustering for each categorical variable.

Clustering in a Predictive Modeling Flow

When clustering for a predictive modeling flow, the goal is to provide good clusters such that you can build accurate predictive models on the data within the clusters. When this is the case, you should use supervised learning approaches in addition to cluster evaluation measures.

In predictive modeling scenarios, the use of a holdout test set is important to ensure that you have not over-trained your models. In this case, you should generate your clusters without using your holdout test set. If the clusters are generated using the test set, then how your models are built on top of those clusters is biased.

In this scenario the entire process flow of feature engineering, clustering, and predictive modeling is compared to other process flows. Clustering is but one step of the process, and to evaluate it in a stand-alone way can lead you to optimize a local goal (for example, best clusters according to some measure), without optimizing the overall goal (for example, best predictive model flow according to a different measure).

For predictive modeling scenarios, you should choose the clustering method whose assumptions best fit the data or predictive models that you are using. For example, a shallow decision tree built with a continuous target will first generate a clustering based on a few rules, which segment the data based on ranges of the target. Then additional models, such as linear regression might produce different regressions for each of the clusters. In this way, even if the entire problem was not well-fit by a linear regression, it is possible that the individual clusters are better described by the smaller cluster-wise regressions.

CONCLUSION

This paper contains an overview of many popular clustering methods and how you can access these methods in SAS. The paper also contains a survey of commonly used cluster evaluation measures that you might encounter in the cluster analysis literature.

When clustering data, you should always keep your end goal in mind. Doing so will help you direct your focus on appropriate algorithmic and evaluation choices. Visualizations of your data and clusters can also help you determine the quality of your clustering results.

REFERENCES

Boley, Daniel. 1998. "Principal Direction Divisive Partitioning." *Data Mining and Knowledge Discovery* 2.4:325-344.

Forgy, Edward W. 1965. "Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classifications." *Biometrics* 21:768-769.

Hussain, Zaeem, and Meilla, Marina. 2014. "Graph Sensitive Indices for Comparing Clusterings." *arXiv preprint arXiv:1411.7582*.

Karypis, George, Kumar, Vapin, and Steinbach, Michael. 2000. "A Comparison of Document Clustering Techniques." *TextMining Workshop at KDD2000.* Boston, MA.

Maaten, Laurens van der, and Hinton, Geoffrey. 2008. "Visualizing Data using t-SNE." *Journal of Machine Learning Research* 9:2579-2605.

MacQueen, James. 1967. "Some Methods for Classification and Analysis of Multivariate Observations." *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. 1.14:281-297. Berkeley, CA.

SAS Institute Inc. 2018. "The CLUSTER Procedure." In *SAS/STAT 15.1 User's Guide.* Cary, NC: SAS Institute Inc. Available: https://go.documentation.sas.com/?docsetId=statug&docsetTarget=statug_cluster_examples01.htm&docsetVersion=15.1&locale=en.

SAS Institute Inc. 2018. "Getting Started: TSNE Procedure." In *SAS Visual Data Mining and Machine Learning 8.3: Procedures.* Cary, NC: SAS Institute Inc. Available: https://go.documentation.sas.com/?activeCdc=pgmsascdc&cdcId=amocdc&cdcVersion=8.0&docsetId=casml&docsetTarget=casml_tsne_gettingstarted.htm&locale=en.

Von Luxburg, Ulrike. 2007. "A Tutorial on Spectral Clustering." *Statistics and Computing* 17.4: 295-416.

CONTACT INFORMATION

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

Ralph Abbey
SAS Institute Inc.
Ralph.Abbey@sas.com