A Term Paper Report on

Understanding and Enhancement of Internal Clustering Validation Measures

Submitted in partial fulfillment of requirements to

For the TERM PAPER in

IV/IV B. Tech. in C.S.E (1st SEMESTER)

by

Batch No: 1

P.Bhavana (Y12CS812)

D.Sai Tarun (Y12CS829) G.Anvesh Babu(Y12CS848)

November- 2015

Department of Computer Science and Engineering

R.V.R. & J.C.COLEEGE OF ENGINEERING (AUTONOMOUS)

Chandramoulipuram :: Chowdavaram

GUNTUR-522019

R.V.R & J.C.COLLEGE OF ENGINEERING

(Autonomous)

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Term Paper project work titled **"Understanding and Enhancement of Internal Clustering Validation Measures"** is the bonafide work of **P.Bhavana (Y12CS812), D.Sai Tarun (Y12CS829), G.Anvesh Babu (Y12CS848)** who have carried out the work under my supervision, and submitted in partial fulfillment of the requirements of **CS-451, TERM PAPER** during the year **2015-2016**.

Mr.P.Venkateswara Rao	**Mr.A. Sri Nagesh**	**Dr.M.Sreelatha**
Guide,Term Paper	**Incharge,Term Paper**	(**Prof.,&HOD,CSE)**

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without a proper suggestion, guidance and environment. Combination of these three factors acts like a backbone to our term paper **"Understanding and Enhancement of Internal Clustering Validation Measures".**

We express our sincere thanks to **Mr. P. Venkateswara Rao,** Asst. Professor, Department of CSE, for his timely help, guidance and providing us with the most essential materials required for the completion of this work.

We are greatly indebted to **Dr**. **M. Sreelatha,** Prof, & HOD, Department of Computer Science and Engineering for valuable suggestion during the course period.

We regard our sincere thanks to our principal, **Dr. A. Sudhakar** for providing support and stimulating environment. We would like to express our gratitude to the management of R.V.R & J.C College of Engineering for providing us with a pleasant environment and excellent lab facility.

We would be thankful to all the teaching and non-teaching staff of department of Computer Science & Engineering for their cooperation, for the successful completion of the work.

P.Bhavana (Y12CS812)

D.Sai Tarun (Y12CS829)

G.Anvesh Babu (Y12CS848)

ABSTRACT

Clustering validation has long been recognized as one of the vital issues essential to the success of clustering applications. In general, clustering validation can be categorized into two classes, external clustering validation and internal clustering validation. In this paper, we focus on internal clustering validation and present a study of 11 widely used internal clustering validation measures for crisp clustering. The results of this study indicate that these existing measures have certain limitations in different application scenarios.

As an alternative choice, we are studying a new internal clustering validation measure, named clustering validation index based on nearest neighbors (CVNN), which is based on the notion of nearest neighbors. This measure can dynamically select multiple objects as representatives for different clusters in different situations. Experimental results show that CVNN outperforms the existing measures on both synthetic data and real-world data in different application scenarios.

CONTENTS

List of Tables

List of Figures

List of Abbreviations

CVNN	Clustering Validation index based on Nearest Neighbors
RMSSTD	Root Mean Square Standard Deviation
RS	R-squared
CH	Calinski-Harabasz
D	Dunn’s Index
S	Silhouette Index
DB	Davies-Bouldin
XB	Xie-Beni
kNN	k nearest neighbor
DBSCAN	Density Based Spatial Clustering of Applications with Noise

1. INTRODUCTION

Clustering, one of the most important unsupervised learning problems, is the task of dividing a set of objects into clusters such that objects within the same cluster are similar while objects in different clusters are distinct. Clustering is widely used in many fields, such as text mining, image analysis, and bioinformatics. As an unsupervised learning task, it is necessary to find a way to validate the goodness of partitions after clustering. Otherwise, it would be difficult to make use of different clustering results.

Clustering validation, which evaluates the goodness of clustering results, has long been recognized as one of the vital issues essential to the success of clustering applications. External clustering validation and internal clustering validation are two main categories of clustering validation. The main difference is whether external information is used for clustering validation. An example of external validation measure is entropy, which evaluates the purity of clusters based on the given class labels.

Unlike external validation measures, which use external information not present in the data, internal measures evaluate the goodness of a clustering structure without respect to external information. Since external validation measures know the true cluster number in advance, they are mainly used for choosing an optimal clustering algorithm on a specific data set. On the other hand, internal validation measures can be used to choose the best clustering algorithm as well as the optimal cluster number without any additional information. In practice, external information such as class labels is often not available in many application scenarios. Therefore, internal validation measures are the only option for cluster validation when there is no external information available.

In the literature, a number of internal clustering validation measures for crisp clustering have been proposed, such as the Calinski–Harabasz index (CH), the Davies–Bouldin index (DB), and standard deviation index (SD). We present a study of 11 widely used internal validation measures. We study the impact of monotonicity of the first three measures and study others in a very detailed way. We investigate their validation properties in different aspects, such as data with noise, different density, and arbitrary shapes.

We propose a new internal clustering validation measure, named clustering validation index based on nearest neighbors (CVNN), which is based on the notion of nearest neighbors. This

measure consists of two components which measures intercluster separation and intracluster compactness, respectively. We refer to the idea of k-nearest neighbor (*kNN*) consistency and propose an index by using dynamic multiple objects as representatives for different clusters in different situations when measuring the intercluster separation. The measurement of the intracluster compactness is similar to the existing measures.

1.1. Problem Statement

Clustering is one of the most important unsupervised learning problems. A number of internal clustering validation measures for crisp clustering have been proposed. In this paper, we present a study of 11 widely used internal clustering validation measures for crisp clustering.

However, current existing measures can be affected by various data characteristics. For example, noise in data can have a significant impact on the performance of an internal validation measure, if minimum or maximum pairwise distances are used in the measure. More fundamentally, existing measures are likely to perform well only in sphere-shaped clusters. The performance of existing measures in different situations remains unknown.

As an alternative choice, we propose a new internal clustering validation measure, named clustering validation index based on nearest neighbors (CVNN), which is based on the notion of nearest neighbors. We refer to the idea of k-nearest neighbor (*kNN*) consistency and propose an index by using dynamic multiple objects as representatives for different clusters in different situations when measuring the intercluster separation. The measurement of the intracluster compactness is similar to the existing measures.

1.2. Objectives of work

We conduct experiments on different data sets with different characteristics using existing internal clustering validation measures. The results of this study indicate that these existing measures have certain limitations in different application scenarios.

We take clustering validation index based on nearest neighbors (CVNN) and perform this measure on different data sets with different characteristics as above. Finally, we provide comparative experiments to evaluate the validation properties and performances of CVNN and also the existing measures. Experimental results show that CVNN outperforms the existing measures on both synthetic data and real-world data in different application scenarios. Therefore, we later conclude that CVNN can be used as a complementary measure to the existing measures, in particular, when data have arbitrary shapes.

1.3. Existing Work

In this section, we introduce some basic concepts of internal validation measures, as well as a suite of 11 widely used internal validation indices. As the goal of clustering is to make objects within the same cluster similar and objects in different clusters distinct, internal validation measures are often based on the following two criteria.

1. **Compactness**. It measures how closely related the objects in a cluster are. A group of measures evaluate cluster compactness based on variance. In addition, the cluster compactness can be based on distance, such as maximum or average pairwise distance and maximum or average center-based distance.
2. **Separation**. It measures how distinct or well separated a cluster is from other clusters. The pairwise distances between cluster centers or the pairwise minimum distances between objects in different clusters are widely used as measures of separation. Also, measures based on density are used in some indices.

The general procedure to determine the best partition and optimal cluster number of a set of objects by using internal validation measures is as follows.

Step 1: Initialize a list of clustering algorithms which will be applied to the dataset.

Step 2: For each clustering algorithm, use different combinations of parameters to get different clustering results.

Step 3: Compute the corresponding internal validation index of each partition obtained in Step2.

Step 4: Choose the best partition and the optimal cluster number according to the criteria.

Table 1.3 Existing Internal Validation Measures

	Measure	Notation	Definition	Optimal value
1	Root-mean-square std dev	$RMSSTD$	$\{\sum_i \sum_{x\in C_i} \parallel x - c_i \parallel^2 / [P\sum_i (n_i - 1)]\}^{\frac{1}{2}}$	Elbow
2	R-squared	RS	$(\sum_{x\in D} \parallel x - c \parallel^2 - \sum_i \sum_{x\in C_i} \parallel x - c_i \parallel^2) / \sum_{x\in D} \parallel x - c \parallel^2$	Elbow
3	Modified Hubert Γ statistic	Γ	$\frac{2}{n(n-1)} \sum_{x\in D} \sum_{y\in D} d(x,y) d_{x\in C_i, y\in C_j}(c_i, c_j)$	Elbow
4	Calinski-Harabasz index	CH	$\frac{\sum_i n_i d^2(c_i,c)/(NC-1)}{\sum_i \sum_{x\in C_i} d^2(x,c_i)/(n-NC)}$	Max
5	I index	I	$(\frac{1}{NC} \cdot \frac{\sum_{x\in D} d(x,c)}{\sum_i \sum_{x\in C_i} d(x,c_i)} \cdot \max_{i,j} d(c_i, c_j))^p$	Max
6	Dunn's indices	D	$\min_i \{\min_j (\frac{\min_{x\in C_i, y\in C_j} d(x,y)}{\max_k \{\max_{x,y\in C_k} d(x,y)\}})\}$	Max
7	Silhouette index	S	$\frac{1}{NC} \sum_i \{\frac{1}{n_i} \sum_{x\in C_i} \frac{b(x)-a(x)}{\max[b(x),a(x)]}\}$ $a(x) = \frac{1}{n_i - 1} \sum_{y\in C_i, y\neq x} d(x,y), b(x) = \min_{j,j\neq i} [\frac{1}{n_j} \sum_{y\in C_j} d(x,y)]$	Max
8	Davies-Bouldin index	DB	$\frac{1}{NC} \sum_i \max_{j,j\neq i} \{[\frac{1}{n_i} \sum_{x\in C_i} d(x,c_i) + \frac{1}{n_j} \sum_{x\in C_j} d(x,c_j)]/d(c_i,c_j)\}$	Min
9	Xie-Beni index	XB	$[\sum_i \sum_{x\in C_i} d^2(x,c_i)]/[n \cdot min_{i,j\neq i} d^2(c_i,c_j)]$	Min
10	SD validity index	SD	$Dis(NC_{max}) Scat(NC) + Dis(NC)$ $Scat(NC) = \frac{1}{NC} \sum_i \parallel \sigma(C_i) \parallel / \parallel \sigma(D) \parallel, Dis(NC) = \frac{max_{i,j} d(c_i,c_j)}{min_{i,j} d(c_i,c_j)} \sum_i (\sum_j d(c_i,c_j))^{-1}$	Min
11	S_Dbw validity index	S_Dbw	$Scat(NC) + Dens\_bw(NC)$ $Dens\_bw(NC) = \frac{1}{NC(NC-1)} \sum_i [\sum_{j,j\neq i} \frac{\sum_{x\in C_i \cup C_j} f(x,u_{ij})}{max\{\sum_{x\in C_i} f(x,c_i), \sum_{x\in C_j} f(x,c_j)\}}]$	Min

D: Data set

N : number of objects in D

P : attributes number of D

NC : number of clusters

c_i : the i^{th} cluster

n_i : number of objects in C_i.

c_i : center of C_i

$\sigma(C_i)$: variance center of C_i

d(x,y) : distance between x and y

$\parallel X_i \parallel$: $(X_i^T . X_i)^{1/2}$

Table 1.3 shows a suite of 11 widely used internal validation measures. To the best of our knowledge, these measures represent a good coverage of the validation measures available in different fields, such as data mining, information retrieval, and machine learning. The Definition column gives the computation forms of the measures. On the other hand, most indices consider

both of the evaluation criteria (compactness and separation) in the way of ratio or summation, such as the index DB, Xie–Beni index (XB), and S_Dbw; some only consider one aspect, such as root-mean-square standard deviation (RMSSTD), R-squared (RS), and G. Next, we briefly introduce these measures.

The RMSSTD is the square root of the pooled sample variance of all the attributes. It measures the homogeneity of the formed clusters.RS is the ratio of sum of squares between clusters to the total sum of squares of the whole data set. It measures the degree of difference between clusters. The modified Hubert G statistic (G) evaluates the difference between clusters by counting the disagreements of pairs of data objects in two partitions. The index CH evaluates the cluster validity based on the average between- and within-cluster sum of squares.

Index *I* (I) measures the separation based on the maximum distance between cluster centers and measures compactness based on the sum of distances between objects and their cluster center. The index by Dunn (D) uses the minimum pairwise distance between objects in different clusters as the intercluster separation and the maximum diameter among all clusters as the intracluster compactness. These three indices take a form of Index = (a · Separation)/(b · Compactness), where a and b are weights. The optimal cluster number is determined by maximizing the value of these indices. The Silhouette index (S) validates the clustering performance based on the pairwise difference of between- and within-cluster distances. In addition, the optimal cluster number is determined by maximizing the value of this index.

The DB is calculated as follows. For each cluster C, the similarities between C and all other clusters are computed, and the highest value is assigned to C as its cluster similarity. Then, the DB index can be obtained by averaging all the cluster similarities. The smaller the index is, the better the clustering result is. By minimizing this index, clusters are the most distinct from each other and, therefore, achieve the best partition. The index XB defines the intercluster separation as the minimum square distance between cluster centers and the intracluster compactness as the mean square distance between each data object and its cluster center. The optimal cluster number is reached when the minimum of XB is found.

In this paper, we will use these two improved measures. The idea of index SD is based on the concepts of the average scattering and the total separation of clusters. The first term evaluates compactness based on variances of cluster objects, and the second term evaluates separation

difference based on distances between cluster centers. The index SD is the summation of these two terms, and the optimal number of clusters can be obtained by minimizing the value of SD.

The index (S_Dbw) takes density into account to measure the intercluster separation. The basic idea is that, for each pair of cluster centers, at least one of their densities should be larger than the density of their midpoint. The intracluster compactness is the same as it is in SD. Similarly, the index is the summation of these two terms, and the minimum value of S_Dbw indicates the optimal cluster number.

1.4. Limitations

Here a study of the 11 internal validation measures and investigation on the validation properties of different internal validation measures in different aspects is given.

1. Impact of Monotonicity

The monotonicity of different internal validation indices can be evaluated by the following experiment. K-means algorithm is applied on the data set and gets the clustering results for different number of clusters. For example take a synthetic data set composed of 1000 data objects, which are well separated to five clusters.

The first three indices monotonically increase or decrease as the cluster number NC increases. On the other hand, the rest eight indices reach their maximum or minimum value as NC equals to the true cluster number.

There are certain reasons for the monotonicity of the first three indices. From the definition of G, only data objects in different clusters will be counted in the equation. As a result, if the data set is divided into two equal clusters, each cluster will have n/2 objects, and $n^2/4$ pairs of distances will be counted actually. If the data set is divided into three equal clusters, each cluster will have n/3 objects, and $n^2/3$ pairs of distances will be counted. Therefore, with the increasing of the cluster number NC, more pairs of distances are counted, which makes G increase.

2. Impact of noise

In order to evaluate the influence of noise on internal validation indices, a synthetic data set formulated by adding 5% noise to the data set used above. The experiment results show that D and CH choose the wrong cluster number. From our point of view, there are certain reasons that D and CH are significantly affected by noise. D uses the minimum pairwise distance between objects in different clusters as the intercluster separation and the maximum diameter among all clusters as the intracluster compactness. Moreover, the optimal number of clusters can be obtained by maximizing the value of D. When noise is introduced, the intercluster separation can decrease sharply since it only uses the minimum pairwise distance, rather than the average pairwise distance, between objects in different clusters. Thus, the value of D may change dramatically, and the corresponding optimal cluster number will be influenced by the noise. Therefore, for the same NC, CH will decrease by the influence of noise, which makes the value of CH instable. Finally, the optimal cluster number will be affected by noise.

3. Impact of Density

Data set with various densities is challenging for many clustering algorithms. Therefore, we are very interested in whether it also affects the performance of the internal validation measures. An experiment is done on a synthetic data set with different density, The results show that only I suggests the wrong optimal cluster number. Data set totally has 650 data objects. The reason why I does not give the right cluster number is not easy to tell. We can observe that I keep decreasing as cluster number NC increases. One possible reason by our guessing is the uniform effect of K-means algorithm, which tends to divide objects into relatively equal sizes. I measures compactness based on the sum of distances between objects and their cluster center. When NC is small, objects with high density are likely in the same cluster, which makes the sum of distances almost remain the same. Since most of the objects are in one cluster, the total sum will not change too much. Therefore, as NC increases, I will decrease since NC is in the denominator.

4. Impact of sub clusters

Subclusters are clusters that are closed to each other. Experiment is done on a synthetic data set which contains five clusters, and four of them are subclusters since they can form two pairs of clusters, respectively. The total number of data objects is 1000. The experiment results evaluate whether the internal validation measures can handle data set with subclusters. For this data set, D, S, DB, SD, and XB get the wrong optimal cluster numbers, while I, CH, and S_Dbw suggest the correct ones.

The reasons are as follows: S uses the average minimum distance between clusters as the intercluster separation. For data set with subclusters, the intercluster separation will achieve its maximum value when subclusters close to each other are considered as one big cluster. Therefore, the wrong optimal cluster number will be chosen due to subclusters.

5. Impact of Skew Distribution

It is common that clusters in a data set have unequal sizes. Take a synthetic data set with skewed distributions, which contains 1500 data objects. It consists of one large cluster and two small ones. Since K-means has the uniform effect which tends to divide objects into relatively equal sizes, it does not have a good performance when dealing with skewed distributed data sets. In order to demonstrate this statement, we employ four widely used algorithms from four different categories: K-means (prototype based), density based spatial clustering of applications

with noise (DBSCAN) (density based), Agglo based on average link (hierarchical), and Chameleon (graph based) . We apply each of them and divide the data set into three clusters, since three is the true cluster number. K-means performs the worst, while Chameleon is the best. We use Chameleon as the clustering algorithm. It shows that CH does not give correct cluster number.

6. Impact of Arbitrary Shapes

Data set with arbitrary shapes is always hard to handle. Experiment is done on a synthetic data set which consists of six irregular shapes of clusters. It is generated by removing 10% noise from the original data set which contains 8000 objects. Chameleon is taken as the best clustering algorithm.. Experiment results show that none of the existing measures can deal with data set with arbitrary structures. D uses the minimum pairwise distance between objects in different clusters to measure the intercluster separation. When dealing with arbitrary-shaped data sets, this can be misleading.

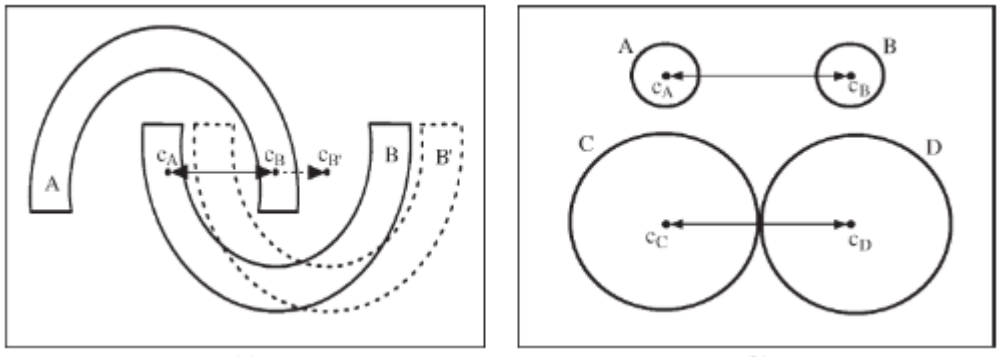

Figure 3.1.1 Intercluster Seperation between clusters a) Arcuate shape b) Sphere shape

For example, consider cluster A and cluster B shown in Fig. The minimum pairwise distance between these two clusters is almost zero while they are still separable. For CH, I, DB, SD, S_Dbw, and XB , these six indices use the cluster center of each cluster as the representative for that cluster when evaluating the intercluster separation. In addition, S uses the average minimum pairwise distance between objects in each cluster as the separation measurement, which can be viewed as equivalent as the minimum pairwise distance between cluster centers in a sense. Since it is meaningful to use the center to represent for the entire cluster only for the sphereshaped cluster, it implies that these indices can only work in the hypersphere condition. Fig. shows an illustration for this argument. In this figure, both clusters A and B have an arcuate

structure, and the cluster centers are not even in the clusters. If we move cluster B from the real-line place to the dash-line place B, A and B are getting closer, while the distance between their centers becomes larger. In this case, it is meaningless and incorrect to make cluster center representative for the entire cluster. Furthermore, sometimes, it may be also inappropriate to use cluster centers to compute the intercluster separation even for the sphere-shaped clusters. Here is an example. There are two small clusters (A and B) and two big clusters (C and D) in Fig. If we use the distance between cluster centers as the intercluster separation, then we have distances equal. However, it is clear that clusters A and B are better separated than clusters C and D.

2. Literature Review

At first, a process was described for partitioning an N-dimensional population into k sets on the basis of a sample. The process, which is called 'k-means,' appears to give partitions which are reasonably efficient in the sense of within-class variance.

the k-means procedure is easily programmed and is computationally economical, so that it is feasible to process very large samples on a digital computer. In addition to suggesting practical classification methods, the study of k-means has proved to be theoretically interesting. The k-means concept represents a generalization of the ordinary sample mean, and one is naturally led to study the pertinent asymptotic behavior, the object being to establish some sort of law of large numbers for the k-means.

A method for identifying clusters of points in a multi-dimensional Euclidean space is described and its application to taxonomy considered. It reconciles, in a sense, two different approaches to the investigation of the spatial relationships between the points, viz., the agglomerative and the divisive methods. A graph, is constructed on a nearest neighbor basis and then divided into clusters by applying the criterion of mini-mum within-cluster sum of squares. This procedure ensures an effective reduction of the number of possible splits. An informal indicator of the best number of clusters is suggested.

Later, a new measure is presented which indicates the similarity of clusters which are assumed to have a data density which is a decreasing function of distance from a vector characteristic of the cluster. The measure can be used to infer the appropriateness of data partitions and can therefore be used to compare relative appropriateness of various divisions of the data. The measure does not depend on either the number of clusters analyzed nor the method of partitioning of the data and can be used to guide a cluster seeking algorithm.

A new graphical display is proposed for partitioning techniques. Each cluster is represented by a so-called silhouette, which is based on the comparison of its tightness and separation. This silhouette shows which objects he well within their cluster, and which ones are merely somewhere in between clusters. The entire clustering is displayed by combining the silhouettes into a single plot, allowing an appreciation of the relative quality of the clusters and an overview of the data configuration. The average silhouette width provides an evaluation of clustering validity, and might be used to select an appropriate number of clusters.

A new clustering algorithm DBSCAN relying on a density-based notion of clusters which is designed to discover clusters of arbitrary shape. DBSCAN requires only one input parameter and supports the user in determining an appropriate value for it. DBSCAN is significantly more effective in discovering clusters of arbitrary shape than the well-known algorithm CLARANS, and that DBSCAN outperforms CLARANS by factor of more than 100 in terms of efficiency.

So far, existing algorithms use a static model of the clusters and do not use information about the nature of individual clusters as they are merged. Furthermore, one set of schemes ignores the information about the aggregate interconnectivity of items in two clusters. The other set of schemes ignores information about the closeness of two clusters as defined by the similarity of the closest items across two clusters.

Hierarchical clustering solutions, which are in the form of trees called dendrograms, are of great interest for a number of application domains. Hierarchical trees provide a view of the data at different levels of abstraction. The consistency of clustering solutions at different levels of granularity allows flat partitions of different granularity to be extracted during data analysis, making them ideal for interactive exploration and visualization. In addition, there are many times when clusters have subclusters, and the hierarchical structure are indeed a natural constraint on the underlying application domain. Later, cluster kNN consistency is proposed i.e. for any data point in a cluster, its k-nearest neighbors and mutual nearest neighbors should also be in the same cluster.

Clustering is mostly an unsupervised procedure and most of the clustering algorithms depend on assumptions and initial guesses in order to define the subgroups presented in a data set. As a consequence, in most applications the final clusters require some sort of evaluation. A scheme for finding the optimal partitioning of a data set during the clustering process regardless of the clustering algorithm used is proposed. More specifically, we present an approach for evaluation of clustering schemes (partitions) so as to find the best number of clusters, which occurs in a specific data set.

A symmetry-based cluster validity index, named Sym-index (Symmetry distance-based index), is proposed. It is able to correctly indicate the presence of clusters

of different sizes as long as they are internally symmetrical. A genetic-algorithm-based clustering technique that optimizes the Sym-index is used for image segmentation where the number of clusters is determined automatically.

A new measure of cluster stability to assess the validity of a cluster model is proposed. This stability measure quantizes the reproducibility of clustering solutions on a second sample, and it can be interpreted as a classification risk with regard to class labels produced by a clustering algorithm. The preferred number of clusters is determined by minimizing this classification risk as a function of the number of clusters.

An objective function called the Weighted Sum Validity Function (WSVF), which is a weighted sum of the several normalized cluster validity functions is proposed. Further, a Hybrid Niching Genetic Algorithm (HNGA), which can be used for the optimization of the WSVF to automatically evolve the proper number of clusters as well as appropriate partitioning of the data set is also proposed. Within the HNGA, a niching method is developed to preserve both the diversity of the population with respect to the number of clusters encoded in the individuals and the diversity of the subpopulation with the same number of clusters during the search. In addition, we hybridize the niching method with the k-means algorithm.

A novel evaluation measure for stream clustering called Cluster Mapping Measure (CMM) is proposed. CMM effectively indicates different types of errors by taking the important properties of evolving data streams into account. Through extensive experiments on real and synthetic data it is shown that CMM is a robust measure for stream clustering evaluation.

There are several impacts on the existing validation measures like monotonicity, noise, density, sub clusters, skew distribution, and arbitrary shapes. The problem with the existing measures is that they take cluster center as representative for the whole cluster. This is not valid in arbitrary shaped clusters. So here we propose a new measure CVNN that is based on notion of nearest neighbors.

3. Proposed Work

We propose a new internal validation measure based on the notion of nearest neighbors, as a complementary to the existing measures.

3.1. Intercluster Separation Based on Nearest Neighbors

An internal validation measure is generally based on intercluster separation and intracluster compactness. In the literature, some researchers believe that intercluster separation should play a more important role than intracluster compactness in clustering validation. A large number of research works emphasize the development of intercluster separation measures. In general, existing validation measures of intercluster separation can be categorized into six classes.

Let C_i and C_j be two clusters in a data set, c_i and c_j be the cluster centers of C_i and C_j, and n_i and n_j be the numbers of objects in C_i and C_j, respectively.

The following are the six classes.

1. $Sep(C_i,C_j) = \min_{x \in Ci, y \in Cj} d(x,y)$.
2. $Sep(C_i,C_j) = \max_{x \in Ci, y \in Cj} d(x,y)$.
3. $Sep(C_i,C_j) = (n_i \cdot n_j)^{-1} \sum_{x \in Ci, y \in Cj} d(x,y)$.
4. $Sep(C_i,C_j) = d(c_i,C_j)$.
5. $Sep(C_i,C_j) = \max\{\delta(C_i,C_j),\delta(C_j,C_i)\}$ where
   $\delta(C_i,C_j) = \max_{x \in Ci}\{\min_{y \in Cj} d(x,y)\}$ and
   $\delta(C_j,C_i) = \max_{y \in Cj}\{\min_{x \in C} (x,y)\}$.
6. $Sep(C_i,C_j) = Dens(C_{ij})/\max\{Dens(c_i),Dens(c_j)\}$ where
   C_{ij} is the midpoint of c_i and c_j and dens(c) is the density of c, usually computed by counting the number of objects within a certain distance from c.

Looking deeper into these six categories, we find that one single object is used to represent the entire cluster when calculating the intercluster separation in categories 1), 2), 4), 5), and 6). However, using single representative for the entire cluster is questionable since one single object cannot keep the geometrical information of the whole cluster. The measurement in category 3) is based on the average pairwise distance between objects in different clusters, which only considers the positions of the objects in clusters but not the object distributions which form the

geometrical information. Thus, there are certain limitations of the existing intercluster separation measures.

The criterion widely accepted to measure the intercluster separation present is to use cluster center as the representative for the entire cluster. However, as discussed earlier, as well as in Section III, this idea has its limitation since it loses the geometrical information and can only work on data sets with sphere-shaped clusters and is sometimes doubtable even in the spherical situation. Therefore, we propose a new intercluster separation measure as an alternative choice.

The basic idea of our measure is straightforward. We evaluate the intercluster separation only based on objects that carry the geometrical information of each cluster. Since one single object cannot reveal the geometrical information of the entire cluster, we use multiple objects as representatives; since, for different clusters, the object distributions and the relative positions to other clusters are different, we use different objects for the same cluster to present the geometrical information in different situations. In sum, we use dynamic multiple objects as representatives for different clusters in different situations when measuring the intercluster separation.

3.1.1 Cluster kNN Consistency*:*

For any data object in a cluster, its kNNs should also be in the same cluster. *kNN* was initially proposed as the foundation of the *kNN* classification algorithm 50 years ago. Later, it was realized that, since the goal of clustering is to divide objects into clusters, such that objects are more similar to objects within the cluster than to objects in other clusters, similar objects often tend to be close to each other. Thus, it is highly likely that an object and its nearest neighbors are all in the same cluster. In this sense, *kNN* consistency can be extended from supervised learning tasks (classification) to unsupervised learning tasks (clustering). Now, back to the question that which objects best represent for the entire clusters when evaluating the intercluster separation.

If an object is located in the center of a cluster and is surrounded by objects in the same cluster, it is well separated from other clusters and thus contributes little to the intercluster separation; if an object is located at the edge of a cluster and is surrounded mostly by objects in other clusters, it connects to other clusters tightly and thus contributes a lot to the intercluster separation. It shares the same idea of *kNN* consistency. Along this line, we propose a new measurement of the intercluster separation based on the notion of nearest neighbors, which is different from the existing measures.

3.1.2 Intercluster Separation Calculation (Sep):

$$Sep(NC,k) = \max_{i=1,2,...,NC}((1/n_i)\sum_{j=1,2,3,...ni}(q_j/k)) \text{ , where}$$

NC is the cluster number,

k is the number of nearest neighbors,

n_i is the number of objects in the ith cluster C_i ,

O_j is the jth object in C_i, and

q_j is the number of nearest neighbors of O_j which are not in cluster C_i.

We define the intercluster separation mainly in four steps.

1. First, for each object in each cluster, find out whether at least one of its *kNNs* is in other clusters.
2. Second, for objects with positive answers, assign a weight to each of them (q_j/k). These are the objects that best represent for the entire clusters; for objects with negative answers, the weight is zero.
3. Third, calculate the average weight of objects in the same cluster.
4. Finally, take the maximum average weight among all clusters as the intercluster separation. A lower value of Sep indicates a better intercluster separation.

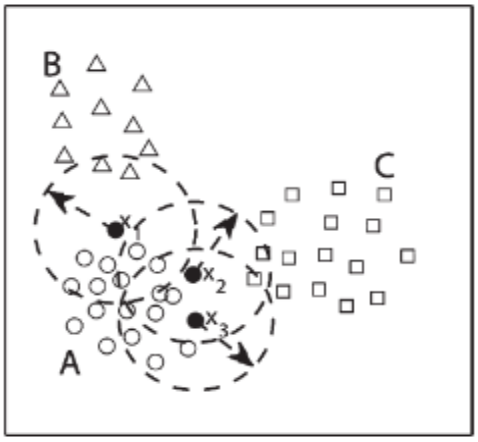

Figure 3.1.2 Intercluster separation calculation process.

Above figure shows an example of the intercluster separation calculation process. There are three clusters in this figure, which are represented by triangle, square, and circle, respectively. We set k =10 here. For objects in cluster A, we find out that only objects x_1, x_2, and x_3 have two, two, and one neighbor(s) in other clusters out of their ten-nearest neighbors separately. These three

objects are the representatives for cluster A when calculating the intercluster separation. Since there are 20 objects in cluster A,

Sep_A (3, 10) = (1/20) · (2/10 + 2/10 + 1/10) =0.025.

The same process can be applied to calculate Sep_B and Sep_B , and the maximum value is the intercluster separation index.

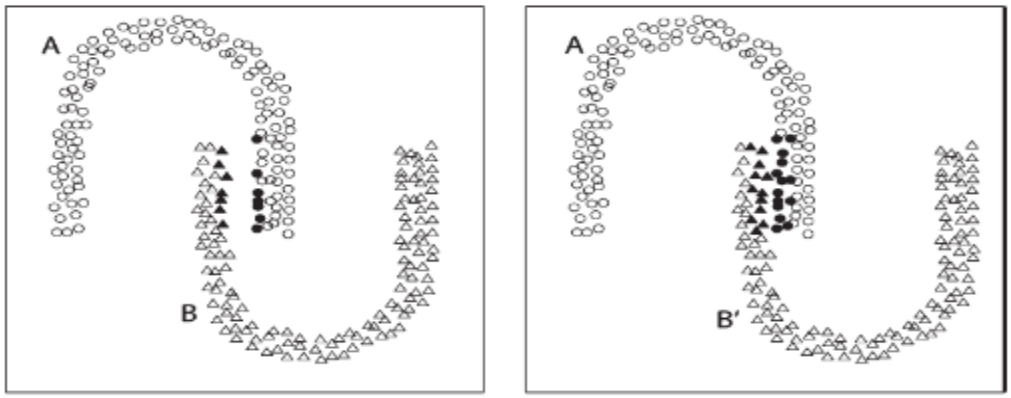

Figure 3.1.3 Illustration of dynamic effect of cluster representatives

a) Before movement b) After movement

Above Fig. shows the dynamic effect of how representatives for clusters evolve in different situations when measuring the intercluster separation. In this example, both clusters A and B have an arcuate structure, and solid objects are the representatives selected by our measure. Comparing Fig. (a) with Fig. (b), we can see that A and B are closer than A and B, which indicates that the intercluster separation is getting worse. Meanwhile, the numbers of representatives for both clusters re growing as well as our intercluster separation measure Sep, which agree with the indication that clusters are getting worse separated.

3.2 Intracluster Compactness

Intracluster compactness is the other indispensable part of internal validation measures. Usually, the compactness shows a monotonically decreasing tendency when cluster number approaches to the number of objects in the data set. Existing validation measures of intracluster compactness can be categorized into the following five classes.

Let C_i be one cluster in a data set, x and y be the two different objects in C_i, c_i is the cluster center of C_i, and n_i is the number of objects in C_i. The following are the five classes.

1. $Com(C_i) = (1/n_i)\sum_{x \in Ci} d^2(x,c_i)$.
2. $Com(C_i) = \max_{x,y \in Ci} d(x,y)$.
3. $Com(C_i) = (2/n_i .(n_i - 1)) \sum_{x,y \in Ci} d(x,y)$.
4. $Com(C_i) = (1/n_i)\sum_{x \in Ci} d(x,c_i)$.
5. $Com(C_i) = || \sigma(C_i) ||$.

In order for an internal validation measure to deal with arbitrary-shaped data set, we should avoid using the center point to represent for the entire cluster, since it only works in the hypersphere condition. As a result, we can eliminate the candidacies of categories 1, 2, and 5. In addition, the compactness should also not be determined by the distance between a single pair of objects in the cluster, which excludes category 2) as well. Therefore, the only option left as our measurement of intracluster compactness is category 3.

3.2.1. Intracluster Compactness (Com):

$Com(NC) = (2/n_i .(n_i - 1)) \sum_{x,y \in Ci} d(x,y)$, where

NC is the cluster number,

n_i is the number of objects in the ith cluster C_i,

and x and y are two different objects in C_i.

This measure is mainly based on the average pairwise distance between objects in the same cluster. Note that a lower value of Com indicates better intracluster compactness.

This measure does not take one object or the cluster center as the representative of the whole cluster. So this can be applied to the clusters with arbitrary shapes, skew distribution etc.

4. Implementation

To evaluate and compare the validation properties and performances of existing measures and the CVNN we first divide data sets into clusters using K-means, Chameleon, DBSCAN etc.

4.1 Algorithms

The algorithms used to divide data sets into clusters.

4.1.1 k-means

It is a centroid-based technique. It is one of the method in partitioning type clustering.

Input: k : the number of clusters.

D: a data set containing n objects.

Output: A set of clusters.

Method:

1) arbitrarily choose k objects from D as the initial cluster centers.
2) Repeat
3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
4) Update the cluster means that is, calculate the mean value of the objects for each cluster;
5) Until no change;

First, it randomly selects *k* of the objects in *D,* each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the Euclidean distance between the object and the cluster mean. The *k*-means algorithm then iteratively improves the within-cluster variation. For each cluster, it computes the new mean using the objects assigned to the cluster in the previous iteration. All the objects are then reassigned using the updated means as the new cluster centers. The iterations continue until the assignment is stable, that is, the clusters formed in the current round are the same as those formed in the previous round.

4.1.2 DBSCAN

It is a density based clustering method based on connected regions with high density.

Input: D: a dataset containing n objects.

ε: the radius parameter, and

MinPts: the neighborhood density threshold.

Output: A set of density based clusters.

Method:

```
1) mark all objects as unvisited;
2) do
3)         randomly select an unvisited object p;
4)         mark p as visited;
5)         if the ε-neighborhood of p has at least Minpts objects
6)               create a new cluster C, and add p to C;
7)               let N be the set of objects in the ε-neighborhood of p;
8)               for each point p' in N
9)                    if p' is unvisited
10)                       mark p' as visited;
11)                       if the ε-neighborhood of p' has at least Minpts points;
12)                       add those points to N;
13)                  if p' is not yet a member of any cluster, add p' to c;
14)              end for
15)              output C;
16)       else mark p as noise;
17) until no object is unvisited;
```

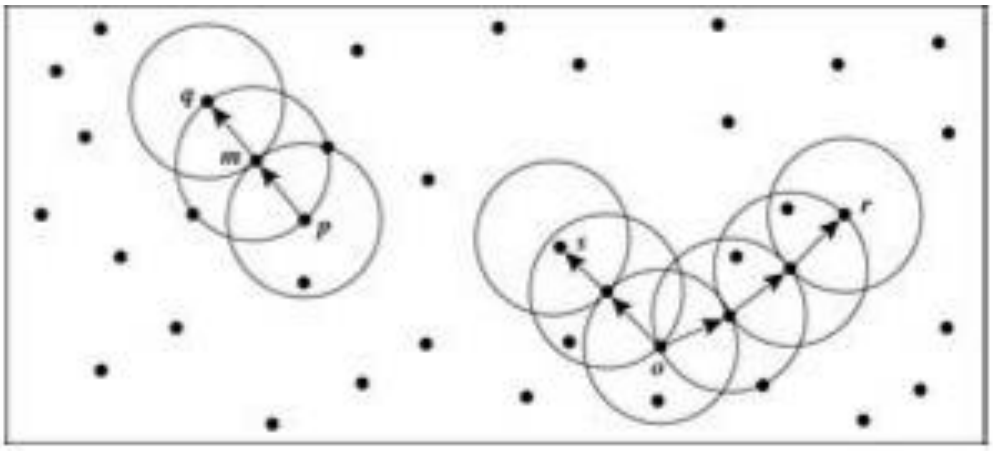

Figure 4.1.2 DBSCAN Example

Of the labeled points, m, p, o, r are core objects because each is in an ϵ-neighborhood containing at least three points. Object *q* is directly density- reachable from *m*. Object *m* is directly density-reachable from *p* and vice versa. Object *q* is (indirectly) density-reachable from *p* because *q* is directly density-reachable from *m* and *m* is directly density-reachable from *p*. However, *p* is not density-reachable from *q* because *q* is not a core object. Similarly, *r* and *s* are density-reachable from *o* and *o* is density-reachable from *r*. Thus, *o*, *r*, and *s* are all density-connected.

4.2 CVNN Index

Based on the intercluster separation and intracluster compactness,
we have the definition of our internal CVNN.

$$CVNN(NC,k) = Sep_{norm}(NC,k)+Com_{norm}(NC), \text{ where}$$

$$Sep_{norm}(NC,k) = Sep(NC,k)/(\max_{NCmin \le NC \le NCmax} Sep(NC,k)) \text{ and}$$

$$Com_{norm(NC)}=Com(NC)/ \max_{NCmin \le NC \le NCmax} Com(NC)).$$

This index takes a form of the summation of the intercluster separation and the intracluster compactness. Note that we normalize them to the same range before adding them up, since they should have the same order of magnitude. A lower value of CVNN indicates a better clustering result.

The computational complexity of CVNN is determined by the complexities of both Sep and Com. For Sep, the main computational cost is the search of the *kNNs* for each object in the data set. This is a one-time effort, and the result of the *kNN* search can be stored for future use. The brute force method gives a complexity of O(dN), where N is the total number of objects in the data set and d is the number of 2 dimensions. Some space-partition-based methods, such as k-d tree and R-tree, reduce the computational complexity down to O (dN log N). When computing Sep with the result of the *kNN* search, for each object O, we have to decide how many O *kNNs* share the same cluster with it. Considering the variation of NC, the total complexity of Sep should be

$$O(dN \log N)+O(N) \cdot k \cdot (NC_{max}- NC_{min}) = O(dN \log N).$$

For Com, since we have to compute the average pairwise distance of objects within the same cluster, the computational complexity is O(dN). Therefore, the complexity of CVNN is O(dN), which makes it affordable for large-scale and high-dimensional data sets.

5. Conclusion

In this paper, we have investigated the validation properties of a suite of 11 existing internal clustering validation measures in different aspects. As demonstrated by the experiment results, these measures are only applicable in certain situations. In particular, none of them performs well on data sets with arbitrary shapes. As a complementary measure to these existing measures, we proposed a new internal clustering validation measure, named CVNN, which exploits the notion of nearest neighbors and uses dynamic multiple objects as representatives for different clusters in different situations.

In future we conduct experimental results that show CVNN as capable of suggesting the correct number of clusters as well as the best partition on various synthetic and real-world data sets, including the data set with arbitrary cluster shapes.

6. References

1. Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in Proc. IEEE Int. Conf. Data Mining, 2010, pp. 911–916.
2. N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for nonnegative matrix factorization," IEEE Trans. Signal Process., vol. 60, no. 6, pp. 2882–2898, Jun. 2012.
3. T. Zhou, D. Tao, and X. Wu, "Manifold elastic net: A unified framework for sparse dimension reduction," Data Mining Knowl. Discov., vol. 20, no. 3, pp. 340–371, May 2011.
4. N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online non-negative matrix factorization with robust stochastic approximation," IEEE Trans. Neural Netw. Learn. Syst., vol. 23, no. 7, pp. 1087–1099, Jul. 2012.
5. U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 12, pp. 1650–1654, Dec. 2002.
6. A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Upper Saddle River, NJ: Prentice-Hall, 1988.
7. J. Wu, H. Xiong, and J. Chen, "Adapting the right measures for k-means clustering," in Proc. ACM SIGKDD, 2009, pp. 877–886.
8. P.-N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, 1st ed. Boston, MA: Addison-Wesley, 2005.
9. M.Brun, C.Sima, J.Hua, J.Lowey, B.Carroll, E.Suh, and E. R. Dougherty, "Model based evaluation of clustering validation measures," Pattern Recogn., vol. 40, no. 3, pp. 807–824, Mar. 2007.
10. M. J. Song and L. Zhang, "Comparison of cluster representations from partial second- to full fourth-order cross moments for data stream clustering," in Proc. IEEE ICDM, 2008, pp. 560–569.
11. H. Kremer, P. Kranen, T. Jansen, T. Seidl, A. Bifet, G. Holmes, and B. Pfahringer, "An effective evaluation measure for clustering on evolving data streams," in *Proc. ACM SIGKDD*, 2011, pp. 868–876.

12. W. Sheng, S. Swift, L. Zhang, and X. Liu, "A weighted sum validity function for clustering with a hybrid niching genetic algorithm," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 35, no. 6, pp. 1156–1167, Dec. 2005.
13. G. W. Milligan, "A Monte Carlo study of thirty internal criterion measures for cluster analysis," *Psychometrika*, vol. 46, no. 2, pp. 187–199, Jun. 1981.
14. Y. Zhao and G. Karypis, "Evaluation of hierarchical clustering algorithms for document datasets," in Proc. CIKM, 2002, pp. 515–524.
15. J. M. Kraus, C. Müssel, G. Palm, and H. A. Kestler, "Multi-objective selection for collecting cluster alternatives," Comput. Stat., vol. 26, no. 2, pp. 341–353, Jun. 2011.
16. S. Sharma, Applied Multivariate Techniques. New York: Wiley, 1996.
17. M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," J. Intell. Inf. Syst., vol. 17, no. 2/3, pp. 107–145, Dec. 2001.
18. L. Hubert and P. Arabie, "Comparing partitions," J. Classification,vol.2, no. 1, pp. 193–218, Dec. 1985.
19. T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," Commun. Stat., vol. 3, no. 1, pp. 1–27, 1974.
20. J. Dunn, "Well separated clusters and optimal fuzzy partitions," Cybern. Syst., vol. 4, no. 1, pp. 95–104, 1974.
21. P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," J. Comput. Appl. Math., vol. 20, no. 1, pp. 53–65, Nov. 1987.
22. D. Davies and D. Bouldin, "A cluster separation measure," IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.
23. X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," IEEE Trans. Pattern Anal. Mach. Intell., vol. 13, no. 8, pp. 841–847, Aug. 1991.
24. M. Kim and R. S. Ramakrishna, "New indices for cluster validity assessment," Pattern Recogn. Lett., vol. 26, no. 15, pp. 2353–2363, Nov. 2005.
25. M. Halkidi, M. Vazirgiannis, and Y. Batistakis, "Quality scheme assessment in the clustering process," in Proc. PKDD, 2000, pp. 265–276.
26. M. Halkidi and M. Vazirgiannis, "Clustering validity assessment: Finding the optimal partitioning of a data set," in Proc. IEEE Int. Conf. Data Mining, 2001, pp. 187–194.

27. S. Saha and S. Bandyopadhyay, “Application of a new symmetry-based cluster validity index for satellite image segmentation,” IEEE Geosci. Remote Sens. Lett., vol. 5, no. 2, pp. 166–170, Apr. 2002.
28. M. Halkidi and M. Vazirgiannis, “Clustering validity assessment using multi-representatives,” in Proc. SETN, 2002, pp. 237–248.
29. R. Tibshirani, G. Walther, and T. Hastie, “Estimating the number of clusters in a data set via the gap statistic,” J. R. Stat. Soc., Ser. B (Stat. Methodology), vol. 63, no. 2, pp. 411–423, 2001.
30. B. S. Y. Lam and H. Yan, “A new cluster validity index for data with merged clusters and different densities,” in Proc. IEEEICSMC, 2005, pp. 798–803.
31. J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in Proc. BSMSP, 1967, pp. 281–297.
32. G. Karypis, Cluto—Software for Clustering High-Dimentional Datasets. Minneapolis, MN: Karypis Lab, 2006, version 2.1.2.
33. H. Xiong, J. Wu, and J. Chen, “K-means clustering versus validation measures: A data distribution perspective,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 39, no. 2, pp. 318–331, Apr. 2009.
34. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in Proc. KDD, 1996, pp. 226–231.
35. G. Karypis, E.-H. S. Han, and V. Kumar, “Chameleon: Hierarchical clustering using dynamic modeling,” Computer, vol. 32, no. 8, pp. 68–75, Aug. 1999.
36. G. Karypis, Karypis Lab. [Online]. Available: http://glaros.dtc.umn.edu/ gkhome/
37. J. C. Bezdek and N. R. Pal, “Some new indexes of cluster validity,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 28, no. 3, pp. 301–315, Jun. 1998.
38. T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann, “Stability-based validation of clustering solutions,” Neural Comput., vol. 16, no. 6, pp. 1299–1323, Jun. 2004.
39. S. Yue, J.-S. Wang, T. Wu, and H. Wang, “A new separation measure for improving the effectiveness of validity indices,” Inf. Sci., vol. 180, no. 5, pp. 748–764, Mar. 2010.
40. C. Ding and X. He, “K-nearest-neighbor consistency in data clustering: Incorporating local information into global optimization,” in Proc. SAC, New York, 2004, pp. 584–589.

41. E. Fix and J. J. L. Hodges, “Discriminatory analysis, nonparametric discrimination: Consistency properties,” USAF Sch. Aviation Med., Randolph Field, TX, Tech. Rep. 4, 1951.
42. N. R. Pal and J. C. Bezdek, “On cluster validity for the fuzzy c-means model,” IEEE Trans. Fuzzy Syst., vol. 3, no. 3, pp. 370–379, Aug. 1995.
43. J. L. Bentley, “Multidimensional binary search trees used for associative searching,” Commun. ACM, vol. 18, no. 9, pp. 509–517, Sep. 1975.
44. A. Guttman, “R-trees: A dynamic index structure for spatial searching,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1984, pp. 47–57.
45. A. Frank and A. Asuncion, UCI Machine Learning Repository. Irvine, CA: Univ. California, Irvine, 2010. [Online]. Available: http://archive.ics.