Course Project ~ EE656A
Self Optimal Clustering Technique

By Shivam Shivanshu (190809)
Under supervision of Dr. Nishchal K. Verma

Table of Content

Paper Abstract

The paper presents a self-optimal clustering technique which is an enhanced version of existing Improved Mountain Clustering (IMC) technique. The SOC technique is also compared with some popular clustering algorithms such as K-means Clustering, Fuzzy Clustering and IMC over some benchmarking validation indices.

The optimizing factor in the threshold function is computed via interpolation and is found forming better clusters as per Global Silhouette Index, Partition Index, Separation Index and Dunn Index

Clustering And Existing Techniques

Clustering is a technique in unsupervised machine learning that involves grouping together similar data points based on their intrinsic properties or features. The goal of clustering is to partition a dataset into distinct groups, or clusters, such that the data points within each cluster are more similar to each other than they are to data points in other clusters. Clustering can be used for various purposes, such as data analysis, pattern recognition, image segmentation, and anomaly detection.

There are various types of clustering algorithms, such as k-means clustering, hierarchical clustering, density-based clustering, and spectral clustering, each with its own strengths and weaknesses. The choice of algorithm and parameters depends on the characteristics of the data and the specific application.

A general algorithm to perform clustering

The process of clustering typically involves the following steps:

1. Selecting a distance or similarity measure to quantify the similarity between data points
2. Choosing a clustering algorithm that determines how to group the data points based on their similarity
3. Defining the number of clusters or using a method to automatically determine the optimal number of clusters
4. Preprocessing the data by scaling or normalizing the features, if necessary
5. Applying the clustering algorithm to the data and obtaining the cluster assignments for each data point
6. Evaluating the quality of the clustering results using metrics such as the silhouette index or the partition index
7. Visualizing the clustering results using techniques such as scatter plots, heatmaps, or dendrograms.

Existing Clustering Algorithm: K - Means Clustering

K-means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into K clusters, where K is a **user-defined parameter** representing the number of clusters. The algorithm seeks to minimize the sum of the squared distances between each data point and its assigned cluster centroid.

One advantage of K-means clustering is its simplicity and efficiency, as it can handle large datasets and is computationally efficient. However, K-means has several limitations, such as **sensitivity** to the initial centroid positions, the assumption of **spherical clusters** of equal size, and the requirement of a **fixed number of clusters**.

Existing Clustering Algorithm: Fuzzy C-Means Clustering

Fuzzy C-means (FCM) clustering is a popular extension of the K-means algorithm that allows for soft clustering, where each data point can belong to **multiple clusters** with different degrees of membership. Unlike K-means, which assigns each data point to a single cluster, FCM assigns each data point a membership value that represents the degree of belongingness to each cluster.

The membership is calculated using the following formula :

$$\left(\frac{1}{\sum_{k=1}^{C} \left(\frac{x_i - c_j}{x_i - c_k} \right)} \right)^{\frac{2}{m-1}}$$

This represents the membership of data x_i in the c_j cluster
The sum of U_{ij} for fixed i over all j is 1, where U represents the belongingness matrix of dimension (N x C)

Existing Clustering Algorithm: IMC

- The Improved Mountain Clustering algorithm is a refinement of the Mountain Method of clustering introduced by Yager and Filev.
- The approach is based on density estimation in feature space with the highest peak extracted as a cluster center and a new density estimation created for extraction of the next cluster center. The process is repeated until a stopping condition is met.
- Comparisons have been made to other clustering algorithms such as the fuzzy c-means clustering algorithm, which show that the improved mountain method is a viable competitor, producing excellent partitions of large data sets.

$$\delta_m = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{\min\left(\mathbf{x}^j\right)}{\sum_{i=1}^{D} x_i^j} \right)$$

This is the threshold function used in the IMC technique. The threshold is constant and not updated heuristically

Issues in existing clustering algorithms

- All existing clustering algorithm in the literature cannot find all the clusters present in the data due to implicit assumption about cluster shapes (Spherical for K-means as example) or multi-cluster configurations based on similarity measures.
- Probabilistic clustering although gives non-overlapping clusters but it requires a large number of iterations to converge, hence has large time complexity
- Modified Mountain Clustering reduces the possible cluster point from set once one cluster point is determined, which reduces the possibility for other points to be treated as a cluster center

★ The SOC technique uses a mathematically optimized threshold function using interpolation (unlike IMC where a heuristically function is used) which provides better results on various performance indices.

Self Optimal Clustering Using Optimized Threshold Function

The proposed segmentation technique does not take account of any physical process unlike existing algorithm which directly segment data in space which works for homogenous data but defines hyperspace to represent data points which also aids in easy processing of data point in space.

The threshold function in SOC is optimized using Lagrange's form of interpolation polynomial (by Edward Waring and Leonhard Euler)

We'll look at algorithm and general structure of SOC technique in next slide :)

Self Optimal Clustering Technique Algorithm

Step 1: We normalize the data to fit it in a hypercube. Let x^j be j^{th} data point of dimension D and total number of data point is n.

Perform the transformation $x^j = \dfrac{x^j - (x)_{min}}{(x)_{max} - (x)_{min}}$ for j = [1, n]

$(x)_{min} = \{ \min x_1^j , \min x_2^j , \ldots \min x_D^j \}$

$(x)_{max} = \{ \max x_1^j , \max x_2^j , \ldots \max x_D^j \}$

Now all the data points are bounded.

Self Optimal Clustering Technique Algorithm

Step 2: We determine the threshold value $\boldsymbol{\delta}_m > 0$, which define the neighbourhood for the m^{th} cluster. $\boldsymbol{\delta}_m$ is a heuristic expression multiplied by an optimizing factor $\boldsymbol{\beta}_m$ for the mth cluster. $\boldsymbol{\beta}_m$ is calculated via the interpolation method. For the first iteration, $\boldsymbol{\beta}_m$ is taken as unity. The expression for threshold value :

$$\delta_m = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum_{i=1}^{D} x_i^j} \right) .(\beta_m).$$

Self Optimal Clustering Technique Algorithm

Step 3: We calculate potential value P(r, m) of each point for the m^{th} cluster using the mountain function stated below which is a function of the euclidean distance between x^r and all other points:

$$P_m^r = \sum_{j=1}^{n} \exp\left[-\left(\frac{d^2(\bar{\mathbf{x}}^r, \bar{\mathbf{x}}^j)}{\delta_m^2}\right)\right]$$

Step 4: Choose the data point 'r' such that the value of P(r, m) is maximized for r in [1..n] and label it as the m^{th} cluster centre

Step 5: Assign those data point to the m^{th} centre whose euclidean distance is less than $\boldsymbol{\delta}_m$ ie $d^2(\bar{\mathbf{x}}^r, \bar{c}_m) \leq \delta_m; \quad \forall r = 1, 2, \ldots, n.$

Step 6: Remove all the data point from the data set which are assigned to the m^{th} cluster

Self Optimal Clustering Technique Algorithm

Step 7: Repeat Steps 2 to 6 for the reduced data set to make successive clusters, equal to the optimum number of clusters M

Step 8: Distribute the rest of the data points among the formed clusters depending upon their Euclidean distances, i.e., nearness to the respective cluster centres (similar to K-means clustering)

Step 9: Calculate the global silhouette value via silhouette index using equations:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \qquad S_m = \frac{1}{N_m} \sum_{i=1}^{N_m} s(i). \qquad GSI = \frac{1}{M} \sum_{m=1}^{M} S_m$$

Self Optimal Clustering Technique Algorithm

A GSI value close to unity indicates better cluster formation which implies the silhouette values S_m approaches 1.

Step 10: Obtain the relationship between $\boldsymbol{\delta}_m$ and S_m for m = 1, 2..M using the Lagrange's interpolation formula :

For M pairs of ($\boldsymbol{\delta}_i$, S_i), the interpolation formula is :

$$S_t = \sum_{m=1}^{M} S_m . l_m(\delta_t)$$

$$l_m(\delta_t) = \prod_{k=1, k \neq m}^{M} \frac{(\delta_t - \delta_k)}{(\delta_m - \delta_k)}$$

This is a polynomial of degree M-1 with $\boldsymbol{\delta}_t$ as the variable. The max value of S_t is unity. We substitute S_t = 1 to find the roots $\boldsymbol{\delta}_t$ for threshold

Self Optimal Clustering Technique Algorithm

Step 11: Select the root of the polynomial in step 10 which gives maximum value of S_t. This selected root say η is the value of δ_t for maximum value of S_t

Step 12: We find the value β_m for m = 1,2,3..,M as:

$$\beta_m = \frac{\eta}{\delta_m}$$

Step 13: Using these new β_m values, repeat from Steps 2 to 13 until δ_m gets converged where the GSI value for the formed clusters is maximized. This gives the best possible cluster through this algorithm.

The number of iteration is important factor for time complexity and is fixed as **10** for the current analysis

Performance and Cluster Quality Indices: GSI

For given cluster X_m with m in [1, M]. Let s(i) be the quality measure of the i^{th} data point in X_m for i in [1, N_m] where N_m is size of the m^{th} cluster.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Here a(i) is average distance b/w i^{th} sample and all other samples in X_m and b(i) is the minimum of the average distance b/w i^{th} sample and all the samples in X_k for k ~= m. s(i) ϵ [-1,1] . A value close to 1 shows better clustering for i^{th} sample.

The silhouette value S_m for the m^{th} cluster is defined as $S_m = \frac{1}{N_m} \sum_{i=1}^{N_m} s(i).$

The GSI is defined as $GSI = \frac{1}{M} \sum_{m=1}^{M} S_m$

Any partition of data point in V clusters is considered optimum if GSI value is largest

Performance and Cluster Quality Indices: PI

PI is the ratio of the sum of compactness and separation of the clusters. It is a sum of individual cluster validity measures normalized through division by the fuzzy cardinality of each cluster.

The partition index measures the degree of similarity or dissimilarity among the data points within each cluster. A low partition index indicates that the data points within each cluster are highly similar to each other, while a high partition index indicates that the data points within each cluster are highly dissimilar to each other

$$PI = \sum_{m=1}^{M} \frac{\sum_{j=1}^{n} (\mu_{jm})^2 \|\overline{\mathbf{x}}^j - \overline{c}_m\|^2}{N_m \sum_{k=1}^{M} \|\overline{c}_k - \overline{c}_m\|^2}$$

where c_m is the mth cluster center, N_m is the fuzzy cardinality, i.e., sum (μ_{jm}), μ_{jm} is the membership of data point j in cluster m

Performance and Cluster Quality Indices: Dunn Index

The Dunn Index (DI) is a metric for evaluating clustering algorithms. It was introduced by J.C. Dunn in 1974. It is an internal evaluation scheme, where the result is based on the clustered data itself. The aim of the Dunn Index is to identify sets of clusters that are compact, with a small variance between members of the cluster, and well separated, where the means of different clusters are sufficiently far apart compared to the within-cluster variance.

The Dunn Index is calculated as the ratio of the smallest inter-cluster distance to the largest intra-cluster distance. The inter-cluster distance can be defined in several ways, such as the distance between the two closest clusters or the distance between cluster centroids. The intra-cluster distance can also be defined in several ways, such as the largest distance between any two points in a cluster or the average distance between all pairs of points in a cluster. Here we take distance as the squared Euclidean distance

$$DI = \min_{1 \le m \le M} \left\{ \min_{\substack{1 \le k \le M \\ k \ne m}} \left\{ \frac{d(X_m, X_k)}{\max_{1 \le m \le M} \{\Delta(X_m)\}} \right\} \right\}$$

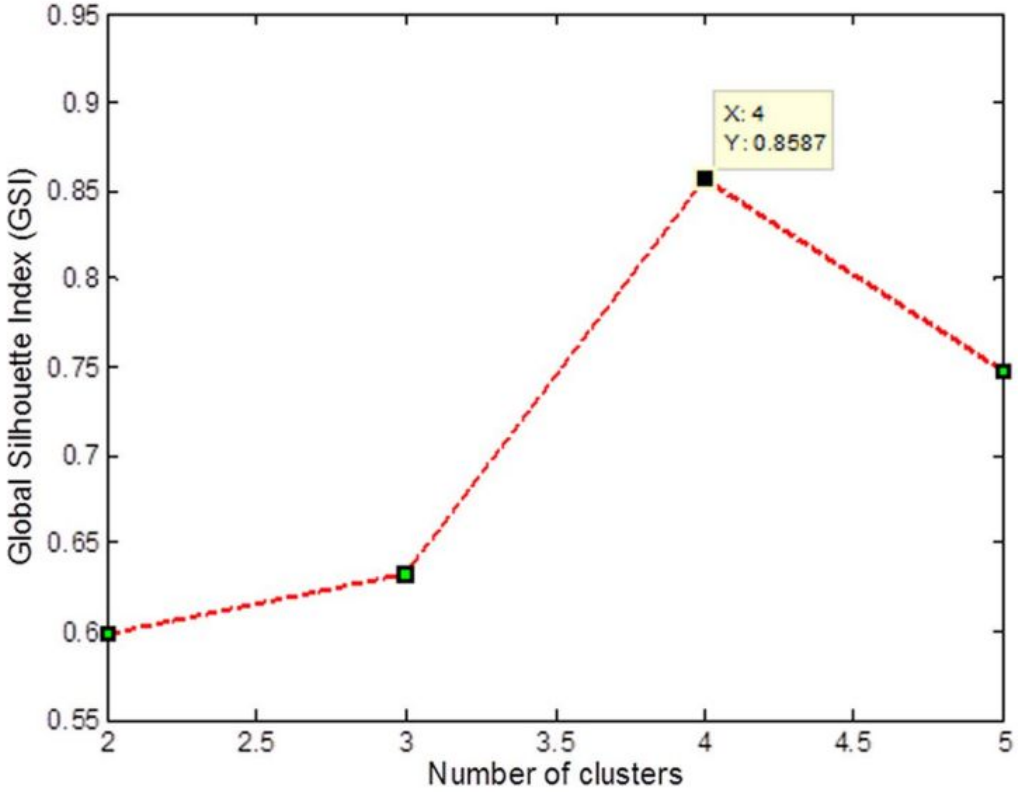
$d(X_m, X_k)$ is the average of the centroid linkage intercluster distance defining the distance between clusters X_m and X_k; $\Delta(X_m)$ represents the complete diameter intracluster distance of cluster X_m

Determining Optimum Number of Clusters

To determine the optimum number of clusters,GSI value is obtained via IMC technique for various clusters. With the obtained GSI values, that number of clusters is said to be optimum whose corresponding GSI value is found to be maximum.

However the paper uses GSI values as optimization objective, any other cluster quality indices can be used as an optimization objective for clustering problem such as Partition Index, Dunn Index or Calinski-Harabasz index, among others. Each index has its own strengths and weaknesses, and the choice of index depends on the specific clustering problem and the goals of the analysis.

Results And Performance



For the sample image above, visually it can be inferred that there are 4 clusters in image and the plot on left gives the same result with max GSI at K (n_cluster) = 4

Results And Performance

Considering the image taken below to perform different performance measures for different know clustering algorithms via well known validation indices

Clustering	GSI	PI	SI	DI
K - Means	0.6986	0.0541	0.0127	0.5255
FCM	0.6967	0.0515	0.0119	0.5354
K-Medoid	0.6941	0.0547	0.0129	0.4965
SOC	0.7056	0.0527	0.0119	0.6443

Results And Performance

- The results indicate that SOC is able to retrieve all the relevant clusters from sample images taken here. The cluster centers in the case of FCM are widely separated. It is able to retrieve all the basic clusters, giving less redundant clusters
- On various images analyzed in the comparison process, the global silhouette values of SOC are found to be well above that of the other clustering techniques in most of the cases and other validation indices as well
- K-medoid results are very much close to the K-means result, but it never shows the ability to outperform other clustering techniques, as FCM dominates over it in most of the cases
- Here, the silhouette index (GSI) is considered as the **ground truth** which is the well-known validation index and has been one of the best segmentation verifying techniques for long. Without much ambiguity, it could only be GSI evaluating the performance of cluster quality of various clustering techniques compared in this paper as GSI is shown to be the robust strategy for assessing the quality of clusters obtained

Project Simulation On Iris Dataset

The Iris dataset is a classic dataset in the field of machine learning and statistics. It contains 150 samples of iris flowers, with 50 samples from each of three different species: Iris setosa, Iris versicolor, and Iris virginica. Each sample has four features: sepal length, sepal width, petal length, and petal width, all measured in centimeters.

The SOC algorithm was deployed on this dataset to observe the change in GSI value over different values of K = number of clusters required

Number Of Clusters in SOC (nk)	GSI value
3	0.7008
4	0.5209
5	0.4273
6	0.4330

Conclusions And References

This paper proposes an advanced and optimized version of the IMC technique as SOC. The performances of a few well-known clustering techniques, e.g., K-means, FCM, EM, K-medoid, IMC-1, and IMC-2, are compared with that of the proposed SOC technique for the segmentation outcomes.

The performance of SOC is found to be the best in most of the cases followed by IMC-1, IMC-2, and FCM in terms of GSI values and several other validation indices as shown in the results section

References:

- https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6555929&tag=1
- Lecture Videos for Course EE656A by Dr. Nishchal K. Verma, Senior Member, IEEE
- https://numpy.org/doc/stable/
- Matlab Code ~ Dr. Nishchal K. Verma
- https://in.mathworks.com/help/stats/machine-learning-in-matlab.html

Thank You