Self-Optimal Clustering Technique Using Optimized Threshold Function

Nishchal K. Verma, *Senior Member, IEEE*, and Abhishek Roy

***Abstract*—This paper presents a self-optimal clustering (SOC) technique which is an advanced version of improved mountain clustering (IMC) technique. The proposed clustering technique is equipped with major changes and modifications in its previous versions of algorithm. SOC is compared with some of the widely used clustering techniques such as K-means, fuzzy C-means, Expectation and Maximization, and K-medoid. Also, the comparison of the proposed technique is shown with IMC and its last updated version. The quantitative and qualitative performances of all these well-known clustering techniques are presented and compared with the aid of case studies and examples on various benchmarked validation indices. SOC has been evaluated via cluster compactness within itself and separation with other clusters. The optimizing factor in the threshold function is computed via interpolation and found to be effective in forming better quality clusters as verified by visual assessment and various standard validation indices like the global silhouette index, partition index, separation index, and Dunn index.**

***Index Terms*—Expectation maximization algorithm, fuzzy cardinality, improved mountain clustering (IMC), interpolation polynomial.**

I. Introduction

CLUSTERS by nature are the collection of similar objects [1]. Each group or cluster is homogeneous, i.e., objects belonging to the same group are similar to each other. Also, each group or cluster should be different from other clusters, i.e., objects belonging to one cluster should be different from the objects of other clusters. Clustering is the process of grouping similar objects [2], and this could be hard or fuzzy. In hard clustering algorithm, each element is allocated to a single cluster during its operation; however, in fuzzy clustering method, a degree of membership is assigned to each element depending on its degree of association to several other clusters. It is possible to convert a fuzzy clustering to a hard clustering by associating each element to the cluster with the highest membership.

As per the literatures, among all the existing techniques, none of the clustering techniques can discover all the clusters present in the data [2] with equal facility because clustering algorithms often contain implicit assumptions about cluster shape or multiple-cluster configurations based on the similarity measures and grouping criteria used. This explains the reason behind the development of a large number of clustering techniques in the literature.

Fuzzy C-means (FCM) [3] is one of the well-known clustering techniques and gives good results in terms of cluster validity. Probabilistic clustering [4] is promising as it gives nonoverlapping clusters, but the large number of iterations required in the algorithm for the convergence increases its computational complexity to the highest. In the modified mountain clustering (MMC) [5], once a potential cluster point is determined, the potentials of other points are reduced. However, owing to this restriction, we tend to miss out certain points, which could as well be potential cluster centers. In the improved mountain clustering (IMC) version-1 (IMC-1) [6], [7] and IMC version-2 (IMC-2) [8], [39], after determining the first potential cluster center, a cluster is formed around this center and is removed from the rest of the data points, thereby maintaining the potential of the remaining data points. This technique gives better results in terms of cluster validity and time complexity [6]. We are able to get all-relevant clusters by reducing the number of redundant clusters. Most of the clusters are demarcated with good performance with this technique. The threshold function defined in IMC is heuristically estimated, always leaving scope for much better optimization of the threshold function and thus having further opportunity in obtaining better quality of clusters. Utilizing this opportunity, we have proposed a self-optimal clustering (SOC) technique with a mathematically optimized threshold function using an interpolation method and compared it with some of the well-known and widely used clustering techniques. It has been shown that the proposed technique is more effective at the optimum number of clusters with better visualized results and well supported by various validity indices as well.

This paper is organized into five sections. An overview of some of the existing clustering algorithms is given in Section II. The proposed technique SOC has been explained in detail in Section III. It also includes a brief discussion on various validity measures used for comparison purpose in this paper. In Section IV, the qualitative and quantitative results of the comparison of various clustering techniques have been discussed on the basis of various cluster validity measures and visual consideration. Finally, the conclusions are drawn in Section V.

II. Overview of Some Clustering-Based Techniques

Various techniques have been proposed so far to develop better and precise clustering algorithms using local variations

Manuscript received August 16, 2012; revised January 31, 2013; accepted February 10, 2013. Date of publication July 9, 2013; date of current version November 20, 2014. This work was supported by the Department of Science and Technology, Ministry of Science and Technology, Government of India, under Grant (SR/S3/EECE/0110/2010) at Indian Institute of Technology Kanpur Kanpur, India.

N. K. Verma is with the Indian Institute of Technology Kanpur, Kanpur 208016, India (e-mail: nishchal.iitk@gmail.com).

A. Roy was with National Institute of Technology Karnataka Surathkal, Mangalore 575025, India (e-mail: abhishekroyn@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSYST.2013.2261231

[9], normalized cuts [10], [11], robust analysis of feature spaces [12], saddle point detection [13], multiresolution [14], color-texture regions [15], and stochastic clustering [16]. These significant approaches highlight a wide scope in the field of clustering and segmentation. Among other developed techniques, Karger's contraction method [17], unsupervised segmentation [18], and a system comprising a threshold classifier [19] also contributed significantly in the development of the more precise clustering algorithms.

Many of the existing segmentation techniques [3], [20]–[22] are based on direct clustering in space and work well on homogeneous regions. The proposed segmentation technique does not take into account any physical processes. It mainly uses a set of defined dimensions in hyperspace to represent the corresponding values. Moreover, it facilitates easy processing of data points defined in hyperspace as well.

Some of the extensively used clustering techniques are FCM clustering [3], MMC [5], expectation–maximization clustering [21], mountain clustering [22], K-means clustering [23], and K-medoid [24], [25]. FCM clustering is developed by Dunn in 1973 and further improved by Bezdek *et al.* [26]. The mountain clustering algorithm is proposed by Yager and Filev [22] for estimating the number and location of cluster centers. This is a simple and easy-to-implement grid-based algorithm. Although this method seems simple, the computation grows exponentially with the dimension of hyperspace. To overcome the computational complexity of this clustering technique, Azeem *et al.* have presented the MMC technique which determines cluster centers by an iterative destruction of the mountain function.

The proposed SOC technique proposed here is an advanced version of the IMC technique. Its threshold function is optimized using Lagrange's form of interpolation polynomial [27]. This interpolation technique is named after Joseph Louis Lagrange and was first discovered by Edward Waring in 1779 and later rediscovered by Leonhard Euler in 1783.

III. SOC TECHNIQUE

A. Algorithm

The determination of the threshold function in SOC via interpolation method achieved a substantial improvement in cluster quality, for each successive cluster. SOC can be realized by the algorithm given hereinafter.

Step 1) Normalize the data for each dimension of hyperspace so that the data points are bounded by a unit hypercube. The j^{th} instance of the data in $\mathbf{x}$ hyperspace is defined as

$$\mathbf{x}^j = \left\{ x_1^j, x_2^j, \ldots, x_D^j \right\} \tag{1.a}$$

where D is the total number of dimensions of hyperspace.

Let $\bar{\mathbf{x}}^j$ be the normalized instance as

$$\bar{\mathbf{x}}^j = \frac{\mathbf{x}^j - (\mathbf{x})_{\min}}{(\mathbf{x})_{\max} - (\mathbf{x})_{\min}}; \quad \forall j = 1, 2, \ldots, n \tag{1.b}$$

where

$$(\mathbf{x})_{\min} = \left\{ \min_{j=1}^{n} x_1^j, \min_{j=1}^{n} x_2^j, \ldots, \min_{j=1}^{n} x_D^j \right\} \tag{2}$$

$$(\mathbf{x})_{\max} = \left\{ \max_{j=1}^{n} x_1^j, \max_{j=1}^{n} x_2^j, \ldots, \max_{j=1}^{n} x_D^j \right\} \tag{3}$$

and n is the total number of instances or data points in the data set.

Step 2) Determine the threshold value δ_m which is a positive value defining the neighborhood of the data point for the m^{th} cluster. δ_m is a heuristic expression multiplied by an optimizing factor β_m for the m^{th} cluster. β_m is calculated via the interpolation method which is described at a later part of the algorithm. Initially, β_m is assumed to have unity value while obtaining the m^{th} cluster for the first time. Compute the threshold function as

$$\delta_m = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j} \right) .(\beta_m). \tag{4}$$

Step 3) Calculate the potential value P_m^r of each point for the m^{th} cluster using the mountain function as expressed in (5), which is simply a function of distance $d^2(\bar{\mathbf{x}}^r, \bar{\mathbf{x}}^j) = (\bar{\mathbf{x}}^r - \bar{\mathbf{x}}^j) Q (\bar{\mathbf{x}}^r - \bar{\mathbf{x}}^j)'$ between $\bar{\mathbf{x}}^r$ and all other data points. Here, Q is a unity matrix

$$P_m^r = \sum_{j=1}^{n} \exp\left[-\left(\frac{d^2(\bar{\mathbf{x}}^r, \bar{\mathbf{x}}^j)}{\delta_m^2} \right) \right]. \tag{5}$$

Step 4) Select the data point corresponding to the highest value among $P_m^1, P_m^2, \ldots\ldots, P_m^n$ as the m^{th} cluster center $\bar{c}_m$. This could be represented as

$$\bar{c}_m = \bar{\mathbf{x}}^* \Leftarrow P_m^* = \max_{r=1}^{n} \left(P_m^r \right). \tag{6}$$

Here, the value of $*$ is that value of "r" at which the value of P_m^r is found to be the highest.

Step 5) Assign those data points in the data set to the m^{th} cluster whose Euclidean distance from the m^{th} cluster center is less than a threshold value δ_m, i.e.,

$$d^2(\bar{\mathbf{x}}^r, \bar{c}_m) \le \delta_m; \quad \forall r = 1, 2, \ldots, n. \tag{7}$$

Step 6) Eliminate all those data points from the data set which are assigned to the m^{th} cluster.

Step 7) Repeat Steps 2 to 6 for the reduced data set to make successive clusters, equal to the optimum number of clusters M (see Section III-D).

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 centers.

As stated earlier, SOC is an advanced version of the IMC method, and it is quite similar to the traditional IMC method until Step 8, but Step 9 and beyond contributes to major addition and modification in the advancement of the SOC method.

Step 9) Calculate the global silhouette value via the silhouette index (GSI) (see Section III-B) using (15)–(17) for the obtained clusters. A GSI value close to unity indicates better cluster formation, and this may be possible when silhouette values S_m for $m = 1, 2, 3, \ldots\ldots, M$ tend to attain unity value.

Let t be any cluster formed for which the threshold and silhouette values are δ_t and S_t, respectively.

Step 10) Obtain a relation between δ_t and S_t by interpolating δ_m for $m = 1, 2, 3, \ldots\ldots, M$ with their corresponding S_m values. Use Lagrange's interpolation formula as follows.

For a total of M clusters, there are M pairs of values as $(\delta_1, S_1), (\delta_2, S_2), \ldots\ldots, (\delta_M, S_M)$. The interpolation polynomial in Lagrange's form is a linear combination as shown in

$$S_t = \sum_{m=1}^{M} S_m.l_m(\delta_t) \tag{8}$$

where

$$\begin{aligned} l_m(\delta_t) &= \prod_{k=1,k\neq m}^{M} \frac{(\delta_t-\delta_k)}{(\delta_m-\delta_k)} \\ &= \frac{(\delta_t-\delta_1)}{(\delta_m-\delta_1)} \cdots\cdots \frac{\left(\delta_t-\delta_{(m-1)}\right)}{\left(\delta_m-\delta_{(m-1)}\right)} \frac{\left(\delta_t-\delta_{(m+1)}\right)}{\left(\delta_m-\delta_{(m+1)}\right)} \\ &\quad \cdots\cdots \frac{(\delta_t-\delta_M)}{(\delta_m-\delta_M)}. \end{aligned} \tag{9}$$

Using (8) and (9), we have

$$S_t = \sum_{m=1}^{M} S_m. \prod_{k=1,k\neq m}^{M} \frac{(\delta_t-\delta_k)}{(\delta_m-\delta_k)}. \tag{10}$$

On expanding (10), we get

$$\begin{aligned} S_t &= S_1. \prod_{k=2}^{M} \frac{(\delta_t-\delta_k)}{(\delta_1-\delta_k)} \\ &\quad + S_2. \prod_{k=1,k\neq 2}^{M} \frac{(\delta_t-\delta_k)}{(\delta_2-\delta_k)} + S_3. \prod_{k=1,k\neq 3}^{M} \frac{(\delta_t-\delta_k)}{(\delta_3-\delta_k)} \\ &\quad + \ldots\ldots\ldots\ldots + S_M. \prod_{k=1}^{(M-1)} \frac{(\delta_t-\delta_k)}{(\delta_M-\delta_k)}. \end{aligned} \tag{11}$$

On further expanding (11), the equation becomes

$$\begin{aligned} S_t &= S_1. \frac{(\delta_t-\delta_2)}{(\delta_1-\delta_2)} \frac{(\delta_t-\delta_3)}{(\delta_1-\delta_3)} \cdots \frac{(\delta_t-\delta_M)}{(\delta_1-\delta_M)} \\ &\quad + S_2. \frac{(\delta_t-\delta_1)}{(\delta_2-\delta_1)} \frac{(\delta_t-\delta_3)}{(\delta_2-\delta_3)} \cdots \frac{(\delta_t-\delta_M)}{(\delta_2-\delta_M)} \\ &\quad + \ldots + S_5. \frac{(\delta_t-\delta_1)}{(\delta_5-\delta_1)} \cdots \frac{(\delta_t-\delta_4)}{(\delta_5-\delta_4)} \frac{(\delta_t-\delta_6)}{(\delta_5-\delta_6)} \\ &\quad \cdots \frac{(\delta_t-\delta_M)}{(\delta_5-\delta_M)} + \ldots + S_M. \frac{(\delta_t-\delta_1)}{(\delta_M-\delta_1)} \frac{(\delta_t-\delta_2)}{(\delta_M-\delta_2)} \\ &\quad \cdots \frac{(\delta_t-\delta_{M-1})}{(\delta_M-\delta_{M-1})}. \end{aligned} \tag{12}$$

This is a polynomial equation with S_t expressed in terms of δ_t, and the maximum possible value of S_t is unity.

Step 11) Substitute S_t equal to unity in the obtained polynomial. This will give

$$\begin{aligned} 1 &= S_1. \frac{(\delta_t-\delta_2)}{(\delta_1-\delta_2)} \frac{(\delta_t-\delta_3)}{(\delta_1-\delta_3)} \cdots \frac{(\delta_t-\delta_M)}{(\delta_1-\delta_M)} \\ &\quad + S_2. \frac{(\delta_t-\delta_1)}{(\delta_2-\delta_1)} \frac{(\delta_t-\delta_3)}{(\delta_2-\delta_3)} \cdots \frac{(\delta_t-\delta_M)}{(\delta_2-\delta_M)} + \cdots \\ &\quad + S_5. \frac{(\delta_t-\delta_1)}{(\delta_5-\delta_1)} \cdots \frac{(\delta_t-\delta_4)}{(\delta_5-\delta_4)} \frac{(\delta_t-\delta_6)}{(\delta_5-\delta_6)} \cdots \frac{(\delta_t-\delta_M)}{(\delta_5-\delta_M)} \\ &\quad + \ldots + S_M. \frac{(\delta_t-\delta_1)}{(\delta_M-\delta_1)} \frac{(\delta_t-\delta_2)}{(\delta_M-\delta_2)} \cdots \frac{(\delta_t-\delta_{M-1})}{(\delta_M-\delta_{M-1})}. \end{aligned} \tag{13}$$

In (13), all the values except δ_t are known.

Step 12) Use (13) to find the roots of δ_t.

Step 13) Substitute the obtained roots back to the polynomial (13) one by one, and select that root for which the value of S_t is found to be closest to unity. The selected root η is the value of threshold function δ_t corresponding to the maximum value of S_t.

Step 14) Divide η by δ_m for $m = 1, 2, 3, \ldots\ldots, M$ to obtain the corresponding β_m values that will be substituted in (4) in the calculation of new δ_m while repeating the clustering process again, thus maximizing silhouette value S_m for the clusters formed

$$\beta_m = \frac{\eta}{\delta_m}; \quad \forall m = 1, 2, 3, \ldots\ldots, M. \tag{14}$$

Step 15) Using these new β_m values, repeat from Steps 2 to 14 until δ_m gets converged where the GSI value for the formed clusters is maximized. This gives the best possible cluster through this algorithm.

It is to be noted that, having the complexity of the algorithm as an important factor and ensuring fairness as per the detailed analysis of the pool of sample images, the number of iterations is fixed as 10.

Calculation of β: We have used some widely used and accepted validation indices to measure the quality of clusters. They are the global silhouette index (GSI) [28], [29], partition index (PI) [30], separation index (SI) [30], and Dunn index (DI) [31], [32]. All validation indices are based on the comparison of intercluster distances and intracluster distances. For better quality of clustering, intercluster distances should be as high as possible, and intracluster distances among data points forming clusters should be as low as possible [29]. Variation in δ_m brings changes in cluster quality and intercluster and intracluster distances and alters values of different validation

TABLE I
VARIATION OF δ_m, S_m, GSI, η AND β_m VALUES WITH VARYING ITERATION

Optimum no. of Clusters, M = 2	**No. of iterations**	δ_m		S_m		$GSI = \frac{\left(\sum_{m=1}^{M} S_m\right)}{M}$	η	$\beta_m = \frac{\eta}{\delta_m}$	
		δ_1	δ_2	S_1	S_2			β_1	β_2
	1	0.0326	0.0617	0.9093	0.7318	0.8206	0.0177	0.5429	0.2869
	2	0.0177	0.0166	0.9170	0.7285	0.8228	0.0181	1.0226	1.0904
	3	0.0334	0.0677	0.9093	0.7318	0.8206	0.0159	0.4760	0.2349
	4	0.0155	0.0135	0.9170	0.7285	0.8228	0.0164	1.0581	1.2148
	5	0.0344	0.0752	0.9093	0.7318	0.8206	0.0135	0.3924	0.1795
	6	0.0128	0.0101	0.8484	0.7455	0.7970	0.0168	1.3125	1.6634
	7	0.0427	0.1069	0.9154	0.7297	0.8226	0.0135	0.3162	0.1263
	8	0.0103	0.0069	0.8484	0.7455	0.7970	0.0152	**1.4758**	**2.2029**
	9	**0.0481**	**0.1412**	**0.9251**	**0.7268**	**0.8260**	0.0130	0.2703	0.0921
	10	0.0088	0.0049	0.8484	0.7455	0.7970	0.0144	1.6364	2.9388

indices too. Therefore, by altering δ_m, we can improve validation indices in order to optimize them for better cluster quality. For our experimentation, we have chosen one of the best cluster validation techniques [28], [29], the GSI, for improvement in δ_m. To find the relationship between δ_m and silhouette index values of different clusters S_m, we used interpolation method to formulate a polynomial showing the relationship between the two varying parameters. They are generalized henceforth in terms of δ_t and S_t. Among all interpolation methods, Lagrange's interpolation method is chosen because the distribution of δ_m in color segmentation may not be uniform in all cases. Lagrange's interpolation gives the $(M-1)^{th}$ degree polynomial for a total of M number of clusters formed. As the maximum possible value obtained for S_t could be unity, thus, the root η is calculated from the polynomial (13) where the value of threshold function δ_t corresponds to the maximum value of S_t. This η value is divided by existing δ_m to obtain corresponding β_m values. We can improve cluster quality by multiplying δ_m with optimizing factor β_m while calculating the values for the corresponding threshold function that, in turn, shifts its value closer to η thereby shifting its silhouette index value toward unity. Using these new β_m values, the algorithm steps are repeated until δ_m gets converged where the GSI value is maximized. On the basis of wide analysis and close observations of the results in all the cases, the number of repetitions of particular steps in the algorithm is chosen as ten to ensure the attainment of the maximum possible GSI value for the corresponding cases. At the end, select the β_m values corresponding to a particular repetition for which the GSI value is found to be the maximum, and repeat the algorithm at the optimum number of clusters M to get the best possible clusters.

In order to justify the algorithm mentioned earlier, a sample image comprising two different colors is analyzed here, and the variation in δ_m and S_m with each of the ten repetitions in the algorithm performed is tabulated in Table I. It shows the values of δ_m and S_m at different iterations. This variation shows the inherent tendency of shifting δ_m toward η and attaining possible values closest to it with every repetition of the algorithm until δ_m converges.

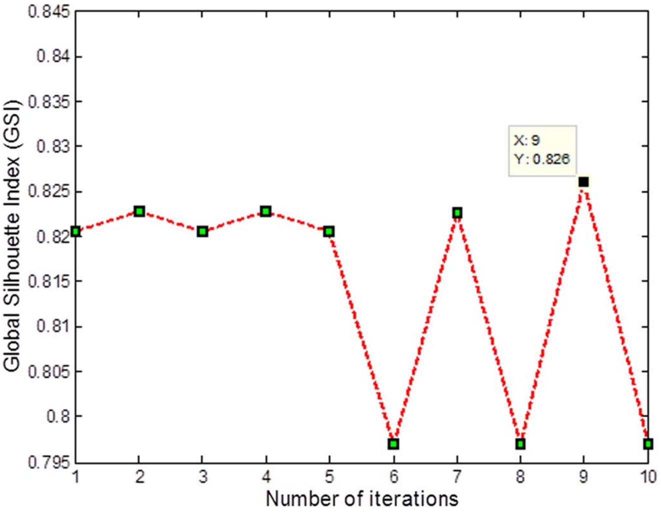


Fig. 1. Variation of GSI with number of iterations for the two-color sample image.

In the first iteration with the value of β_m assumed to be unity, the value of η corresponding to the maximum possible value of S_m, i.e., unity, is obtained as 0.0177. On dividing η with δ_m, the corresponding β_m values are obtained. Utilizing these computed β_m values in the calculation of δ_m for the next iteration, the δ_1 and δ_2 values are shifted toward η. This process continues in each of the iterations while the instances in the data set are clustered. After several iterations, we reach to a point at which the GSI value for the obtained clusters attains maximum. For the sample image shown in Table I, the maximum value of GSI is obtained after clustering the whole data set repeatedly for nine times, thus representing better quality of clusters formed with the corresponding δ_m values. In Table I, the maximum GSI value is highlighted along with the corresponding δ_m values at the ninth iteration, obtained with the aid of β_m values calculated after the eighth iteration while forming clusters via the SOC technique. Here, apart from GSI, values obtained for other validation indices, like PI and SI, also justify the selection of δ_m corresponding to the ninth iteration of the SOC algorithm. The plots of GSI, PI, and SI against the number of iterations are shown in Figs. 1–3, respectively. As the index value in each of the three cases varies between different ranges which are widely separated, thus, all three indices are plotted separately so that their graphical pattern and precise value could be easily analyzed distinctively. We can see that GSI becomes maximum at the ninth iteration in Fig. 1, whereas PI and SI

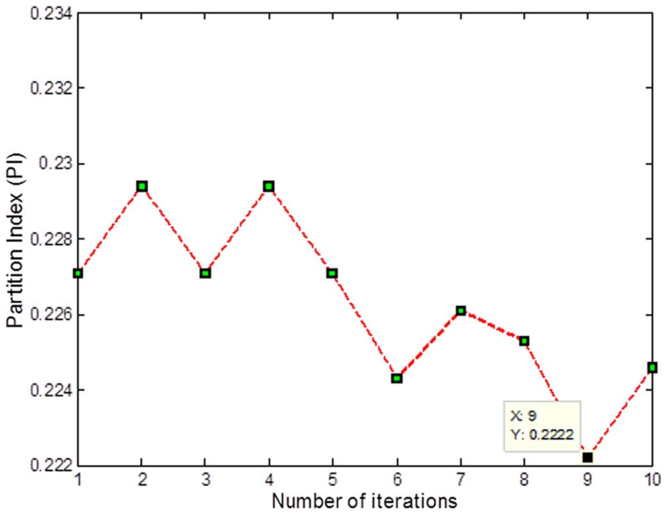


Fig. 2. Variation of PI with number of iterations for the two-color sample image.

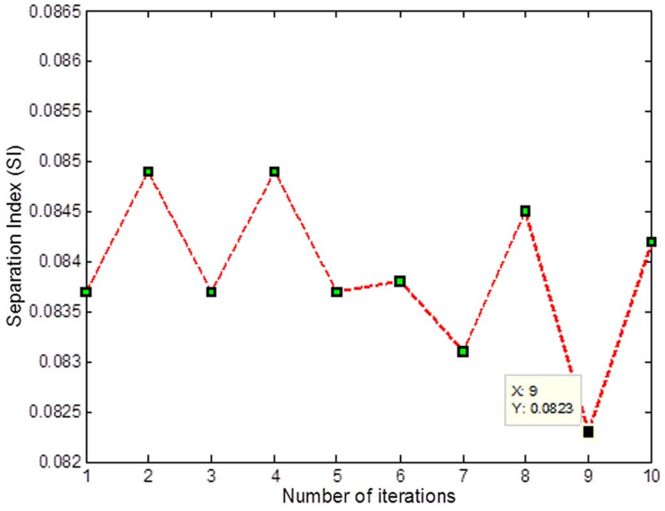


Fig. 3. Variation of SI with number of iterations for the two-color sample image.

values attain their respective minima at the same location in Fig. 2 and Fig. 3, respectively. GSI is shown to be the robust strategy for assessing the quality of clusters obtained [28], [29]. Thus, in most of the cases, choosing the maximum GSI value also makes the other index values better too, as the inherent cluster quality is improved.

B. Measures of Cluster Quality

We have employed four different validation indices as a measure of cluster quality. These validation indices are widely accepted and give results to a high degree of accuracy. Each one of them is clearly defined and described as follows.

GSI: For a given cluster X_m with $m = 1, 2, 3, \ldots\ldots, M$, GSI [28], [29] assigns to each sample of X_m a quality measure $s(i)$ with $i = 1, 2, 3, \ldots\ldots, N_m$ known as the silhouette width. Here, N_m is the number of samples in the m^{th} cluster. The silhouette width is a confidence indicator on the membership of the i^{th} sample in cluster X_m. It is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{15}$$

where $a(i)$ is the average distance between the i^{th} sample and all of the samples included in X_m; "max" is the maximum operator, and $b(i)$ is the minimum of the average distance between the i^{th} sample and all of the samples clustered in X_k $(k = 1, \ldots\ldots, M; k \neq m)$. From this formula, it follows that $-1 \leq s(i) \leq 1$.

When a $s(i)$ is close to 1, it indicates that the i^{th} sample has been well clustered, i.e., it was assigned to an appropriate cluster. When a $s(i)$ is close to zero, it suggests that the i^{th} sample could also be assigned to the closest neighboring cluster. The silhouette value S_m for the m^{th} cluster is defined as

$$S_m = \frac{1}{N_m}\sum_{i=1}^{N_m} s(i). \tag{16}$$

It has been shown that, for any partition $V \leftrightarrow X : X_1 \cup X_2 \cup \ldots\ldots.X_M$, a GSI value can be used as an effective validity index for V. It is computed as follows:

$$GSI = \frac{1}{M}\sum_{m=1}^{M} S_m. \tag{17}$$

Furthermore, it has been demonstrated that, in many cases, (17) can be applied to estimate the most appropriate number of clusters [8] for partition V. In those cases, the partition with the maximum GSI is taken as the optimal partition.

PI: PI [30] 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

$$PI = \sum_{m=1}^{M} \frac{\sum_{j=1}^{n}(\mu_{jm})^2 \|\overline{\mathbf{x}}^j - \overline{c}_m\|^2}{N_m \sum_{k=1}^{M} \|\overline{c}_k - \overline{c}_m\|^2} \tag{18}$$

where $\overline{c}_m$ is the m^{th} 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

$$\mu_{jm} \in \{0, 1\}, 1 \leq j \leq n, 1 \leq m \leq M$$
$$\sum_{m=1}^{M} \mu_{jm} = 1, 1 \leq j \leq n$$
$$0 \leq \sum_{j=1}^{n} \mu_{jm} \leq N, 1 \leq m \leq M.$$

A lower value of PI indicates a better partition.

SI: SI [30] uses a minimum-distance separation for partition validity. A lower value of SI indicates a better partition.

$$SI = \frac{\sum_{m=1}^{M}\sum_{j=1}^{n}(\mu_{jm})^2 \|\overline{\mathbf{x}}^j - \overline{c}_m\|^2}{n.\min_{k,m} \|\overline{c}_k - \overline{c}_m\|^2}. \tag{19}$$

DI: DI [31], [32] identifies sets of clusters that are compact and well separated. For any partition $V \leftrightarrow X : X_1 \cup X_2 \cup$

TABLE II
VALIDATION INDEX VALUES FOR IMC-1, IMC-2, AND SOC

Optimum no. of Clusters = 2	Clustering Method	GSI	PI	SI
	IMC-1	0.8206	0.2271	0.0837
	IMC-2	0.8217	0.2266	0.0836
	SOC	0.8260	0.2222	0.0823

$\ldots\ldots X_M$, where X_m represents the m^{th} cluster of such partition, the Dunn's validation index, DI, is defined as

$$DI = \min_{1\le m\le M}\left\{\min_{\substack{1\le k\le M\\ k\ne m}}\left\{\frac{d(X_m,X_k)}{\max\limits_{1\le m\le M}\{\Delta(X_m)\}}\right\}\right\} \tag{20}$$

where $d(X_m, X_k)$ 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. The main goal of this measure is to maximize intercluster distances while minimizing intracluster distances. Thus, a large value of DI corresponds to good clusters. Therefore, the number of clusters that maximizes DI could be taken as the optimal number of clusters M, and the one showing a higher value than others represents comparatively better cluster quality. The complete diameter intracluster distance is defined as

$$\Delta(X_m) = \max_{x,y\in X_m}\{d(x,y)\} \tag{21}$$

where X_m is a cluster from partition V; $d(x,y)$ defines the distance between any two samples x and y belonging to X_m. The centroid linkage intercluster distance is defined as

$$d(X_m,X_k) = \frac{1}{(|X_m|+|X_k|)}\left(\sum_{x\in X_m} d(x,v_t) + \sum_{y\in X_k} d(y,v_s)\right) \tag{22}$$

where

$$v_s = \frac{1}{|X_m|}\sum_{x\in X_m} x, \tag{23}$$

$$v_t = \frac{1}{|X_k|}\sum_{y\in X_k} y, \tag{24}$$

$|X_m|$ and $|X_k|$ provide the number of samples included in clusters X_m and X_k, respectively.

In this paper, we are focusing on proposing a clustering technique which ensures better performance in clustering and segmentation. It is ascertained to have better validity measure in cluster quality. From the term "better cluster quality," we understand "improved performance in terms of segmentation results." Cluster quality is measured in terms of cluster validity measures, for example, GSI. From (15)–(17), a high value of GSI is obtained when intracluster distances are less and intercluster distances are more, i.e., the segmented clusters have appropriate distribution of instances present in a sample under consideration for segmentation. This shows that instances from one such cluster characteristically differ from the instances of other neighboring clusters and are categorized in separate groups. Visual assessment confirms in the examples that, for a better cluster quality, each of the varying components present in a system should be grouped separately after segmentation. Thus, a higher value of GSI or DI, whereas a lower value of PI or SI, envisages better performance in terms of segmentation results.

C. Need for Optimum Threshold Function

There is a need to optimize the threshold function because of a good number of existing cases in which IMC-1 or IMC-2 remains unable to show better quality results in comparison to other clustering techniques. In few cases, IMC-2 gives results even inferior to IMC-1 as well. Also, with the threshold function of IMC-2 being modified on heuristic basis, there remains a lot of scope in the estimation of the heuristic factor in its threshold function for further improvisation. The threshold function in SOC is systematically optimized using interpolation method, in order to obtain the best possible clustering with this method. As an instance, we have taken a figure that describes the behavior of IMC-1, IMC-2, and SOC.

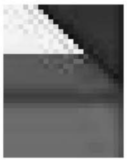
In Table II, the cluster quality is better when the GSI value is higher and the PI and SI values are lower.

From Table II, we can easily infer that, with the optimized threshold function, the SOC algorithm is giving much better results. Red and yellow colors are properly separated out from the analyzed image. The improvement in the quality of clustering with SOC is well supported with the relative increment in the GSI value and the significant decrement in the PI and SI values. Clearly, after this analysis, we can state that segmentation using SOC gives appreciably improved results as compared to IMC-1 in most of the cases and has become even better than IMC-2 in terms of various cluster quality validation measures.

D. Determining Optimum Number of Clusters

To determine the optimum number of clusters, we have calculated GSI 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 [28], [29]. To elucidate it clearly, we have taken a sample image comprising four different colors for illustration, and its various clusters are shown in Table III. Visually, we can see that this figure should have four optimum clusters. Now, this could be verified with the help of the GSI value. The graph in Fig. 4 shows the plot of GSI against the number of clusters for the image shown in Table III. From Fig. 4, clearly, the GSI value is maximum at the fourth cluster.

TABLE III
COLOR IMAGE HAVING FOUR DIFFERENT STRIPES

Sample Image	No. of clusters	1st luster	2nd Cluster	3rd Cluster	4th cluster	5th cluster
	2			NA	NA	NA
	3				NA	NA
	4					NA
	5					

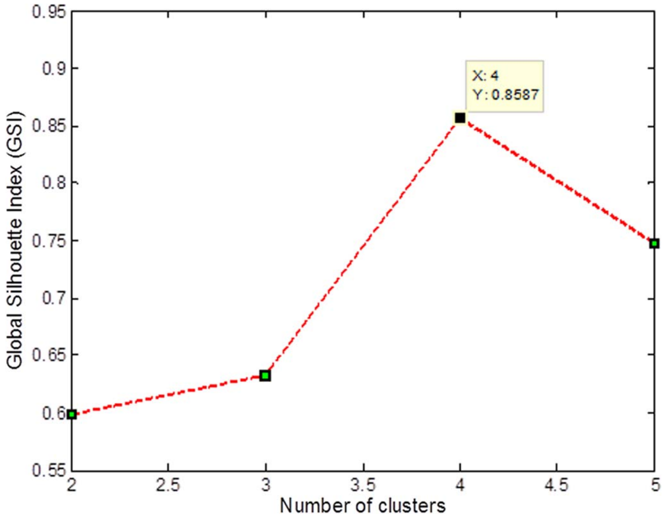


Fig. 4. Variation of GSI with number of clusters for the four-color sample image.

E. Simulation, Mathematical Proof, and Modeling of the SOC Algorithm and Its Results

This section basically aims at mathematically evaluating the convergence property in computing the initial threshold function and threshold function in the subsequent iterations while showing the optimizing nature of the SOC algorithm via simulations as well. In mathematics, an infinite series of numbers is said to converge absolutely if the sum of the absolute value of the summand is finite [33]. To be precise, a real or complex series $\sum_{n=0}^{\infty} a_n$ is said to converge absolutely if $\sum_{n=0}^{\infty} |a_n| = L$ for some real or complex number L. Similarly, an improper integral of a function, $\int_o^{\infty} f(x)dx$, is said to converge absolutely if the integral of the absolute value of the integrand is finite [33], which could be represented as $\int_o^{\infty} |f(x)|dx = L$. However, in mathematics, there are series or integrals which could converge, although they do not converge absolutely. Thus, if in a convergent series, any series which is not absolutely convergent is called conditionally convergent. This could be represented as a series $\sum_{n=0}^{\infty} a_n$ and said to converge conditionally if $\lim\limits_{m\to\infty} \sum_{n=0}^{m} a_n$ exists and is a finite number, i.e., does not evaluate to ∞ or $-\infty$, but $\sum_{n=0}^{\infty} |a_n| = \infty$. We thus applied these conditionally convergent properties on the functions used in our proposed algorithm to evaluate their convergence.

We have the initial threshold function represented as (4). Using (4) and (14), we have

$$\delta_m = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j} \right) \cdot \left(\frac{\eta}{\delta_m} \right). \tag{25}$$

It is to be noted that δ_m mentioned in the Right Hand Side of (25) represents the threshold function having its value calculated in the previous iteration of the calculation and helps in evaluating the value for δ_m shown at the Left Hand Side of (25). We removed the subscript m and introduced I as the subscript in (26) to represent a cluster formed in its I^{th} iteration.

Thus, we have

$$\delta_I = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j} \right)_I .(\beta_{I-1}). \tag{26}$$

Moreover, on substitution using (14)

$$\delta_I = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j} \right)_I \cdot \left(\frac{\eta_{I-1}}{\delta_{I-1}} \right). \tag{27}$$

On further expanding (27) using (4)

$$\delta_I = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j} \right)_I \cdot \left(\frac{\eta_{I-1}}{\left(\frac{1}{2n} \sum\limits_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j} \right)_{I-1} .(\beta_{I-2})} \right). \tag{28}$$

Eventually, we have (29) shown at the bottom of the next page. For any sample point $\mathbf{x}$, we have a range of possible values of $\mathbf{x}(\mathrm{R,G,B})$ as (0,0,0) to (255,255,255). Thus, the expression

$$\frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j}$$

has a minimum value of 0 and a maximum value of 0.3333. This is obtained by Matlab simulation which attempts to use the Nelder–Mead simplex algorithm [34] for returning a vector x that is a local minimizer of the mathematical function. Hence

$$\text{Range of } \frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j} = [0, 0.3333]$$

$$\text{Range of } \left(\frac{1}{2n}\sum_{j=1}^{n}\left(\frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j}\right)\right)_I = [0, 0.1666].$$

Moreover, β_0 is taken as unity in the first iteration in (4); thus

$$\delta_1 = \left(\frac{1}{2n}\sum_{j=1}^{n}\left(\frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j}\right)\right)_I .(\beta_0).$$

Hence

$$\begin{aligned}\text{range of } \delta_1 &= \text{Range of } \left(\frac{1}{2n}\sum_{j=1}^{n}\left(\frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j}\right)\right)_I \\ &= [0, 0.1666] \end{aligned} \tag{30}$$

which results in δ_1 as a constant having its value defined in a fixed range.

Thus, from (29) and (30), we have

$$\delta_I = (const.)_I.f(\eta_{I-1}). \tag{31}$$

Using (15)–(17), GSI is represented as

$$GSI = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{N_m}\sum_{i=1}^{N_m} s(i)$$

i.e.,

$$\begin{aligned} GSI = \frac{1}{M}\Bigg[&\left(\frac{1}{N_1}\left[s(1)+s(2)+\ldots+s(N_1)\right]\right) \\ &+\left(\frac{1}{N_2}\left[s(1)+s(2)+\ldots+s(N_2)\right]\right)+\ldots\ldots. \\ &+\left(\frac{1}{N_M}\left[s(1)+s(2)+\ldots+s(N_M)\right]\right)\Bigg]\end{aligned}$$

From (15), it follows that:

$$-1 \le s(i) \le 1 \tag{32}$$

i.e.,

$$-1 \le \frac{1}{N_M}\left[s(1)+s(2)+\ldots+s(N_M)\right] \le 1$$

i.e.,

$$-1 \le S_m \le 1. \tag{33}$$

Thus

$$\max[S_I] = 1. \tag{34}$$

Now, let t be any cluster formed for which the threshold and Silhouette values are δ_t and S_t, respectively.

From (12), we have

$$\begin{aligned} S_t = S_1.&\frac{(\delta_t-\delta_2)}{(\delta_1-\delta_2)}\frac{(\delta_t-\delta_3)}{(\delta_1-\delta_3)}\cdots\frac{(\delta_t-\delta_M)}{(\delta_1-\delta_M)} \\ +S_2.&\frac{(\delta_t-\delta_1)}{(\delta_2-\delta_1)}\frac{(\delta_t-\delta_3)}{(\delta_2-\delta_3)}\cdots\frac{(\delta_t-\delta_M)}{(\delta_2-\delta_M)}+\ldots \\ +S_5.&\frac{(\delta_t-\delta_1)}{(\delta_5-\delta_1)}\cdots\frac{(\delta_t-\delta_4)}{(\delta_5-\delta_4)}\frac{(\delta_t-\delta_6)}{(\delta_5-\delta_6)}\cdots\frac{(\delta_t-\delta_M)}{(\delta_5-\delta_M)} \\ +\ldots+S_M.&\frac{(\delta_t-\delta_1)}{(\delta_M-\delta_1)}\frac{(\delta_t-\delta_2)}{(\delta_M-\delta_2)}\cdots\frac{(\delta_t-\delta_{M-1})}{(\delta_M-\delta_{M-1})}.\end{aligned}$$

This is a polynomial equation with S_t expressed in terms of δ_t. Hence, from Lagrange's polynomial, we have

$$S_I = (const.)_I.f(\delta_I). \tag{35}$$

From (31), we already have

$$\delta_I = (const.)_I.f(\eta_{I-1}).$$

Furthermore, from (34), we have

$$\max[S_I] = 1.$$

From the proposed algorithm, selected root η in a particular iteration is the value of threshold function δ_t corresponding to the maximum possible value of S_t, i.e., unity. Therefore, clearly, the nature of our threshold function depends on the roots of the interpolation polynomial in Lagrange's form. Now, it becomes imperative to understand for which classes of

$$\delta_I = \left(\frac{1}{2n}\sum_{j=1}^{n}\frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j}\right)_I . \left[\frac{\eta_{I-1}}{\left(\frac{1}{2n}\sum\limits_{j=1}^{n}\frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j}\right)_{I-1} . \left[\eta_{I-2}\Big/\left((\ldots).\left((\ldots)\Big/(\ldots).\left(\eta_2\Big/\left(\frac{1}{2n}\sum\limits_{j=1}^{n}\frac{\min(\mathbf{x}^j)}{\sum\limits_{i=1}^{D} x_i^j}\right)_2 . \left(\frac{\eta_1}{\delta_1}\right)\right)\right)\right)\right]}\right] \tag{29}$$

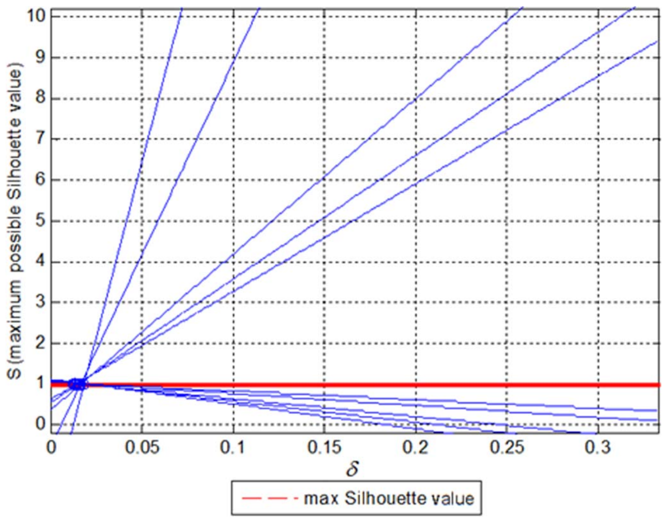


Fig. 5. Variation of S with threshold value δ over the range [0, 0.3333] in the Lagrange's function $S_I = (const.)_I.f(\delta_I)$ for the sample image in Table I.

functions and for which interpolation nodes the sequence of interpolating polynomials $S_I(\delta)$ converges to the interpolated function as the subscript I tends to infinity. For any function $f(x)$ that continuous on an interval $[a, b]$, there exists a table of nodes for which the sequence of interpolating polynomials $S_I(\delta)$ converges to $f(x)$ uniformly on $[a, b]$. This sequence of polynomials of the best approximation $S_I(\delta)$ converges to $f(x)$ uniformly due to the Weierstrass approximation theorem [35]. Now, we need to verify if each value of $S_I(\delta)$ may be obtained by means of interpolation on certain nodes. This is true due to a special property of polynomials of the best approximation known from the Chebyshev alternation theorem [36]. Choosing the points of intersection of interpolation nodes with $f(x)$, i.e., the maximum possible value of S, we obtain the required interpolating polynomial coinciding with the best approximation polynomial.

We obtained these interpolation nodes, i.e., the values of δ which is the same as η when the most optimized value of δ is chosen at the intersection points of the function $S_I = (const.)_I.f(\delta_I)$, by Matlab simulation. The incorporated algorithm in this case was originated by T. Dekker and uses a combination of bisection, secant, and inverse quadratic interpolation methods [37], [38] in order to find the roots of the continuous function of one variable. When this is applied on the given function within a defined range of δ values, which is between 0 and 0.1666 as calculated in (30) in our case, it determines the interpolation nodes for each of the iterations. These interpolation nodes are optimum values for each of the iterations.

On performing the required simulations using the sample image and corresponding data using Table I as reference, it was observed that the points of intersection with the function $S_I = (const.)_I.f(\delta_I)$ are obtained as the simulation results which are required interpolation nodes for each of the iterations as shown in Figs. 5 and 6. Further simulation results are captured in Fig. 7 in which, plotting the obtained interpolation nodes with the nodal values, η from Table I, the node points calculated in both the cases are found to be coinciding, and thus, the function $S_I = (const.)_I.f(\delta_I)$ is proved to be convergent at the obtained interpolation nodes. Fig. 5 plots function $S_I = (const.)_I.f(\delta_I)$ for ten times corresponding to ten different iterations having different $(const.)_I$ values in each of the iterations. The function $S_I = (const.)_I.f(\delta_I)$ appears linear in nature as it corresponds to the sample image shown in Table I having two clusters. In the cases with higher number of clusters, the degree of function $S_I = (const.)_I.f(\delta_I)$ also gets raised, and thereby, the nonlinear graph would be expected. With the aid of similar simulation on $S_I = (const.)_I.f(\delta_I)$, interpolation nodes are obtained as intersection points corresponding to each of the ten iterative functions in Fig. 5, where the former is represented by blue circles and the latter is represented by blue lines. Fig. 6 actually displays the same image as depicted in Fig. 5, but it is appropriately zoomed in at the intersection points to highlight the obtained interpolation nodes. The obtained nodes and corresponding values are plotted against each of the iterations in Fig. 7 and analyzed based on simulation results for the sample image in Table I. As per analysis of the results, interpolation nodes η using simulation results and η using the proposed algorithm are found to be the same, i.e., the Lagrange's polynomial function $S_I = (const.)_I.f(\delta_I)$ is found to be convergent for the obtained interpolation nodes with the best approximation, and as these nodes are the same as η calculated in Table I, thus, corresponding silhouette values are verified to give the best results in their corresponding iterations. It is further evident from simulation results in Fig. 7 that the silhouette value is highest for the ninth iteration, and hence, the corresponding δ value, represented with a green circle in the plot, is giving the most optimized result. Thus, we could summarize our conclusions once again with the help of simulation results in Fig. 7.

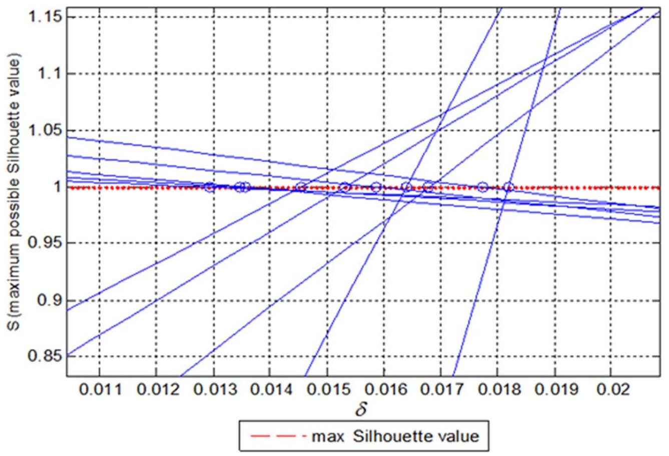


Fig. 6. Computation of interpolation nodes η using "fzero" simulation on the Lagrange's function $S_I = (const.)_I.f(\delta_I)$ for the sample image in Table I.

Interpolation nodes with the best approximations obtained for $S_I = (const.)_I.f(\delta_I)$ confirm the convergence of the function. Corresponding variable or input δ is thereby confirmed to converge to give the most optimized result or maximum silhouette value in the permissible range. Fig. 7 clearly shows the threshold function δ as conditionally convergent. A slight deviation in the graphical plot could be observed at times against the ideal condition or ideal converging nature, as, with each of the iterations, the possible combination of data points in clusters varies; thus, at times, S and δ values may be slightly

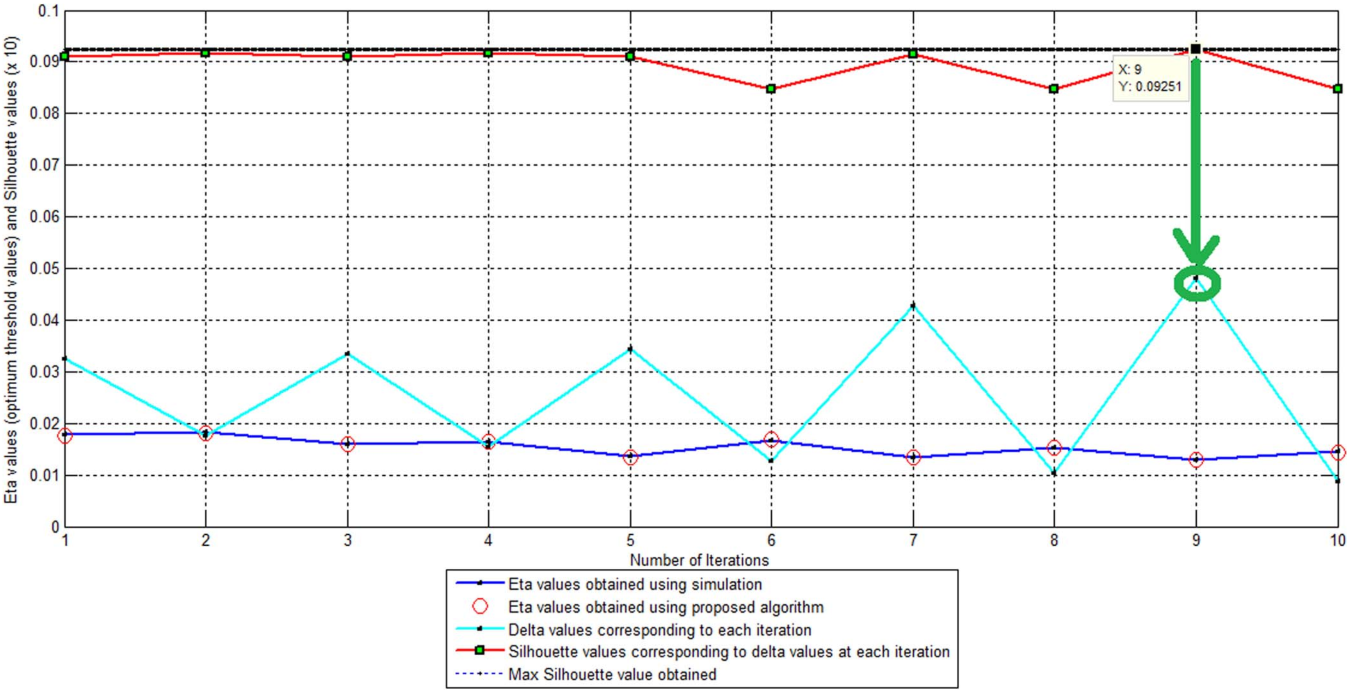


Fig. 7. Plot of simulation results with interpolation node η values obtained using simulation, η values obtained using proposed algorithm, δ values corresponding to each iteration values, corresponding S values, and maximum obtained S value against each of the iterations using the Lagrange's function $S_I = (const.)_I.f(\delta_I)$ for the sample image in Table I.

influenced by segmented data values and their combinations in subsequently formed clusters for a particular iteration. Also, the initial threshold function with threshold factor β having a constant value as "1" is observed to be nonconvergent, for it has a factor $1/n$ in its function. As $\lim_{N\to\infty} \sum_{n=1}^{N}(1/n)$ is a nonfinite series, hence, it impacts the convergence of the initial threshold function. This initial threshold function is made convergent with the multiplication factor β which, in turn, is obtained in each of the iterations from the Lagrange's polynomial $S_I = (const.)_I.f(\delta_I)$ which is already proved as a convergent polynomial at interpolation nodes with the best approximation.

Hence, the threshold function computed in each of the ten iterations is proved to be conditionally convergent, but due to listed minor limitations and variations in the cluster data formed with each of the iterations, the convergence process tends to get damped and diverged at specific nodes. Nevertheless, as per the conditional convergence nature of the threshold function, it is found to adjust itself automatically and thereby gives the most optimized threshold function and thereby better quality clusters. Keeping in view its high precision and accuracy over other existing techniques, the SOC technique is expected to find wide application and give promising results, particularly in medical imaging as it is found to display the most optimized results.

IV. Results and Discussion

A. Results of Comparison

The widely used clustering algorithms discussed earlier are tested and compared by taking various synthetic and natural images as case studies and examples here. Simulations are done on a desktop PC with 2.99-GB RAM, using a 3.00-GHz processor. Prior to the experiments, no preprocessing is done on these images. Red-Green-Blue features are used in clustering. The effectiveness of clusters is compared in terms of GSI, PI, SI, and DI.

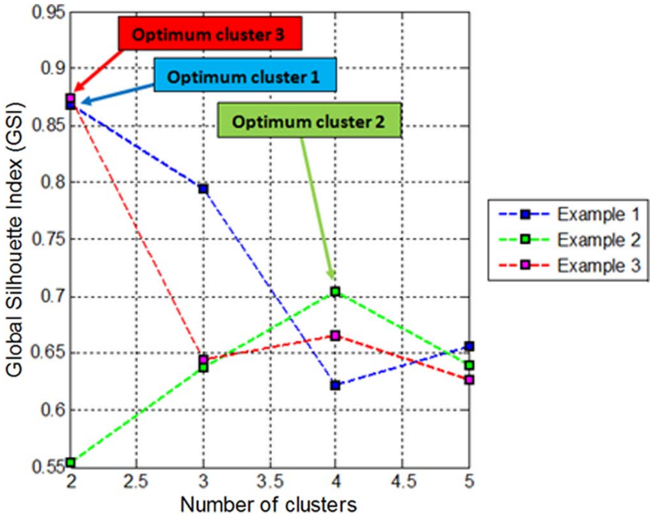


Fig. 8. Variation of GSI with number of clusters for the sample images taken in Examples 1, 2, and 3.

Example 1: Here, we have an image of a partially covered face. The image is clustered using K-means, FCM, EM, K-medoid, IMC-1, IMC-2, and SOC. From the plot in Fig. 8, we can say that optimum number of clusters is two for the sample image taken here. Table IV shows the comparison in terms of different cluster quality measurement indices like GSI, PI, SI, and DI values that show that SOC is much better than other clustering techniques used.

TABLE IV
COMPARISON OF VARIOUS CLUSTERING TECHNIQUES VIA VALIDATION INDICES

Optimum no. of Clusters = 2	Clustering Method	GSI	PI	SI	DI
	K-means	0.8528	0.1021	0.0322	0.9634
	FCM	0.8602	0.0911	0.0294	0.9876
	EM	0.6673	0.1952	0.0552	0.6185
	K-medoid	0.8644	0.0989	0.0316	1.0100
	IMC-1	0.8680	0.0801	0.0270	0.9983
	IMC-2	0.8709	0.0829	0.0283	1.0107
	SOC	0.8765	0.0802	0.0270	1.0453

TABLE V
COMPARISON OF VARIOUS CLUSTERING TECHNIQUES VIA VALIDATION INDICES

Optimum no. of Clusters = 4	Clustering Method	GSI	PI	SI	DI
	K-means	0.6986	0.0541	0.0127	0.5255
	FCM	0.6967	0.0515	0.0119	0.5354
	EM	0.3680	0.0950	0.0535	0.0399
	K-medoid	0.6941	0.0547	0.0129	0.4965
	IMC-1	0.7037	0.0551	0.0130	0.5650
	IMC-2	0.6984	0.0623	0.0152	0.5269
	SOC	0.7056	0.0527	0.0119	0.6443

TABLE VI
COMPARISON OF VARIOUS CLUSTERING TECHNIQUES VIA VALIDATION INDICES

Optimum no. of Clusters = 2	Clustering Method	GSI	PI	SI	DI
	K-means	0.8735	0.1006	0.0497	0.9921
	FCM	0.8741	0.0940	0.0465	0.9502
	EM	0.8716	0.1013	0.0499	0.9987
	K-medoid	0.8727	0.0972	0.0480	0.9913
	IMC-1	0.8746	0.0840	0.0416	0.9936
	IMC-2	0.8731	0.0864	0.0427	0.9917
	SOC	0.8747	0.0867	0.0431	0.9940

Example 2: Here, an image of a bottle with lighting at the background is taken for analysis purpose. The image is clustered using K-means, FCM, EM, K-medoid, IMC-1, IMC-2, and SOC. From the plot in Fig. 8, we can say that the optimum number of clusters is four here. Table V shows the comparison in terms of different cluster quality measurement indices. It is shown here that, although IMC-2 gives inferior results in this case compared to IMC-1, still, SOC extracts the best clusters among all compared techniques.

Example 3: Here, we are concerned with the image of a flower. Fig. 8 shows the variation of GSI values with the number of clusters and indicating two as the optimum number of clusters. Table VI shows the performance of K-means, FCM, EM, K-medoid, IMC-1, IMC-2, and SOC. Here, again, in spite of having inferior validation index values for IMC-2 as compared to IMC-1, SOC is still found dominating over others in terms of better results.

B. Discussion on Performance of Clustering Techniques

The analysis of clustering results on various images as a basis of comparison clearly shows that the quality of clustering in SOC is better than other clustering techniques mentioned here in most of the cases, although IMC-2 follows it closely in terms of cluster quality. 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 well-known validation indices used here for the analysis also support the superiority of SOC to a great extent. The few discrepancies that can be observed while interpreting validation index results could be due to minute differences in parameters on which computation in various validation indices is based on. 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. The disadvantage with FCM is that it is sensitive to the selection of initial partitions and may land up to a local minimum of the criterion function as we can see from the results of various images. The cluster validity values for some of the clusters retrieved by K-means are more than those of FCM, but those cases are very rare. The EM clustering results in unsatisfactory values in terms of validation indices. 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. IMC-2 results are quite comparable and better than IMC-1 in most of the cases, but its advanced-version SOC undoubtedly not only improvises the results of IMC-1 but also dominates over all other clustering techniques compared here including the IMC-2 results in terms of better cluster quality and favorable validation indices' values. Also, there have been few cases as in Examples 2 and 3 mentioned here where IMC-2 results in mediocre clustering as compared to IMC-1. Specifically, in those cases, the SOC

TABLE VII
LIST OF CLUSTERING ALGORITHMS (EXISTING/CONCEPTUALIZED/PROPOSED) HAVING INITIAL THRESHOLD FUNCTION WITH/WITHOUT MODIFICATION

Actual /Assumed name	Description of changes made	Modified threshold function
IMC-1	IMC technique with no changes in the threshold function	$\delta_m = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum_{i=1}^{D} x_i^j} \right)$
IMC-max	IMC technique with 'min' replaced by 'max' in the threshold function	$\delta_m = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{\max(\mathbf{x}^j)}{\sum_{i=1}^{D} x_i^j} \right)$
IMC-half	IMC technique with the factor of '1/2' removed from the threshold function	$\delta_m = \left(\frac{1}{n} \sum_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum_{i=1}^{D} x_i^j} \right)$
IMC-2	IMC technique with a factor 'no. of cluster / (no. of cluster + 1)' multiplied to the threshold function	$\delta_m = \left(\frac{1}{2n} \cdot \sum_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum_{i=1}^{D} x_i^j} \right) \cdot \left(\frac{nk}{nk+1} \right)$ where nk = total number of clusters
SOC	Proposed algorithm using IMC technique with no inherent changes in the INITIAL threshold function and a factor is multiplied in later threshold functions EXTERNALLY only	$\delta_m = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum_{i=1}^{D} x_i^j} \right) .(\beta_m),$ where β_m = optimizing factor multiplied externally

TABLE VIII
COMPARISON OF TABLE VII CLUSTERING ALGORITHMS VIA WELL-KNOWN VALIDATION INDEX—SILHOUETTE INDEX

Optimum no. of Clusters = 2	Clustering Method	GSI
	IMC-1	0.8680
	IMC-max	0.8659
	IMC-half	0.8641
	IMC-2	0.8709
	SOC	0.8765

TABLE IX
COMPARISON OF TABLE VII CLUSTERING ALGORITHMS VIA WELL-KNOWN VALIDATION INDEX—SILHOUETTE INDEX

Optimum no. of Clusters = 2	Clustering Method	GSI
	IMC-1	0.8079
	IMC-max	0.7993
	IMC-half	0.7899
	IMC-2	0.8075
	SOC	0.8079

TABLE X
COMPARISON OF TABLE VII CLUSTERING ALGORITHMS VIA WELL-KNOWN VALIDATION INDEX—SILHOUETTE INDEX

Optimum no. of Clusters = 4	Clustering Method	GSI
	IMC-1	0.7037
	IMC-max	NaN
	IMC-half	NaN
	IMC-2	0.6984
	SOC	0.7056

NAN DENOTES 'NOT A NUMBER'

technique is found to be highly acceptable with its encouraging results as shown. Undoubtedly, experimental results based on validation indices verify SOC as the superior clustering technique. 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 [28], [29], but including the evaluation results for the performance of cluster quality by the other three cluster validity techniques PI, SI, and DI also supports in large in drawing and verifying the conclusion of SOC as being more robust and reliable. Also, not only the extensive evaluation and performance analysis of clustering techniques in the discussed examples but also the simulations and the mathematical proof of the convergence property in computing the successive threshold function in each of the iterations provided in Section III-E strongly confirm SOC as the superior clustering technique with additional support. But then, there still remains a slight ambiguity whether we are obtaining the best results from SOC or there is need to further optimize the initial threshold function. To evaluate the same, the required heuristic modifications are made in the initial threshold function as shown in Table VII, and the clustering results are compared in Tables VIII–X. From the results of Tables VIII–X, it is clearly evident that hypothetical clustering algorithms IMC-max and IMC-half are underperformers, and thus, the modifications made in their threshold function could not be justified. In Table X, both of the hypothetical algorithms failed to create even optimum number of clusters. Thus, both of the algorithms with heuristically modified threshold functions are discarded. Again, as discussed already in this paper, IMC-2 gave a slightly better result than IMC-1 in Table VIII but failed to show any improvement in the analysis of the second and third sample images. Also, it is to be noted that, in IMC-2, modification is made to the threshold function by externally multiplying a factor and not by making any inherent changes in the threshold function. Therefore, first, this could not be called an inherent modification in the initial threshold function, and second, the SOC algorithm clearly outperforms the result of IMC-2 or of any other techniques discussed earlier, thereby proving the superiority of mathematically modified and iteratively optimized algorithm over any heuristic modifications which could be made in the initial threshold function with no mathematical justification. Hence, the same initial threshold function is taken in SOC as defined originally for producing comparatively better quality clusters. Overall, SOC gives the finest clustering results with most of the analyzed images, and the optimizing factor included in its algorithm helps it in attaining the best possible results with much improved quality for the obtained clusters.

V. Conclusion

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. During implementation, we have found that EM fails to yield some of the clusters and has not been as competitive as other clustering techniques. IMC-2 is expectedly giving fair results in most of the cases, but owing to the estimation of the additional factor in its threshold function in heuristic manner, the potential of the IMC-1 technique was not fully tapped. This problem has been eliminated in the SOC by optimizing its threshold function via interpolation to extract the best possible clustering results from it. 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

[1] R. Ali, U. Ghani, and A. Saeed, Data Clustering and Its Applications 1998. [Online]. Available: http://www.members.tripod.com/asim_saeed/paper.htm

[2] A. K. Jain, M. N. Murthy, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[3] M. Razaz, "A fuzzy c-means clustering placement algorithm," in *Proc. ISCAS*, 1993, pp. 2051–2054.

[4] I. Cadez and P. Smyth, "Probabilistic Clustering Using Hierarchical Models," Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, USA, Tech. Rep. 99-16, 1999.

[5] M. F. Azeem, M. Hanmandlu, and N. Ahmad, "Modified mountain clustering and dynamic fuzzy modeling," in *Proc. 2nd Int. Conf. Inf. Tech.*, Bhubaneswar, India, 1999, pp. 61–65.

[6] N. K. Verma and M. Hanmandlu, "Color segmentation via improved mountain clustering technique," *Int. J. Image Graph.*, vol. 7, no. 2, pp. 407–426, Apr. 2007.

[7] N. K. Verma, P. Gupta, P. Agarwal, M. Hanmandlu, S. Vasikarla, and Y. Cui, "Medical image segmentation using improved mountain clustering approach," in *Proc. 6th Int. Conf. ITNG*, Las Vegas, NV, USA, 2009, pp. 1307–1312.

[8] N. K. Verma, A. Roy, and S. Gupta, "Color segmentation using improved mountain clustering technique version-2," in *Proc. 2nd IEEE Int. Conf. Intell. Human Comput. Interact.*, Allahabad, India, 2011, pp. 536–542.

[9] P. F. Felzenszwalb and D. P. Huttenlocher, "Image segmentation using local variation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Santa Barbara, CA, USA, 1998, pp. 98–104.

[10] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[11] J. C. Gamio, S. J. Belongie, and S. Majumdar, "Normalized cuts in 3-D for spinal MRI segmentation," *IEEE Trans. Med. Imag.*, vol. 23, no. 1, pp. 36–44, Jan. 2004.

[12] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: Color image segmentation," in *Proc. Comput. Vision Pattern Recognit.*, 1997, pp. 750–755.

[13] D. Comaniciu, "Image segmentation using clustering with saddle point detection," in *Proc. IEEE Int. Conf. Image Process.*, Rochester, NY, USA, 2002, pp. III-297–III-300.

[14] J. Liu and Y. H. Yang, "Multiresolution colour image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 7, pp. 689–700, Jul. 1994.

[15] Y. Deng and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug. 2001.

[16] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 10, pp. 1053–1074, Oct. 2001.

[17] L. Lucchese and S. K. Mitra, "Color segmentation through independent anisotropic diffusion of complex chromaticity and lightness," in *Proc. Int. Conf. Image Process.*, 2001, pp. 746–749.

[18] L. Lucchese and S. K. Mitra, "Unsupervised low-frequency driven segmentation of color images," in *Proc. Int. Conf. Image Process.*, Kobe, Japan, 1999, pp. 240–244.

[19] J. Bruce, T. Balch, and M. Veloso, "Fast and inexpensive color image segmentation for interactive robots," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2000, pp. 2061–2066.

[20] J. Abonyi, R. Babuska, and F. Szeifert, "Modified Gath-Geva clustering for identification of Takagi-Sugeno fuzzy models," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 5, pp. 612–621, Oct. 2002.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[22] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 8, pp. 1279–1284, Aug. 1994.

[23] J. A. Hartigan and M. A. Wong, "A K-means clustering algorithm," *Appl. Stat.*, vol. 28, no. 1, pp. 100–108, 1979.

[24] L. Kaufman and P. J. Rousseeuw, "Clustering by means of medoids," in *Statistical Data Analysis Based on the L_1 Norm and Related Methods*, Y. Dodge, Ed. Amsterdam, The Netherlands: North-Holland, 1987, pp. 405–416.

[25] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*. New York, NY, USA: Wiley, 1990.

[26] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2/3, pp. 191–203, 1984.

[27] H. Jeffreys and B. S. Jeffreys, "Lagrange's interpolation formula," in *Methods of Mathematical Physics,* 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 1988, p. 260.

[28] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[29] N. Bolshakova and F. Azuaje, “Clustering validation techniques for genome expression data,” *Genomic Signal Process.*, vol. 83, no. 4, pp. 825–833, 2003.

[30] A. M. Bensaid, L. O. Hall, J. C. Bezdek, L. P. Clarke, M. L. Silbiger, J. A. Arrington, and R. F. Murtagh, “Validity-guided (Re) clustering with applications to image segmentation,” *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 2, pp. 112–123, May 1996.

[31] J. C. Dunn, “Well separated clusters and optimal fuzzy partitions,” *J. Cybern.*, vol. 4, no. 1, pp. 95–104, Jan. 1974.

[32] 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. 1988.

[33] W. Rudin, *Principles of Mathematical Analysis*. New York, NY, USA: McGraw-Hill, 1964.

[34] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, “Convergence properties of the Nelder-Mead simplex method in low dimensions,” *SIAM J. Optim.*, vol. 9, no. 1, pp. 112–147, 1998.

[35] E. Bishop, “A generalization of the Stone-Weierstrass theorem,” *Pacific J. Math.*, vol. 11, no. 3, pp. 777–783, 1961.

[36] G. W. Stewart, *Afternotes on Numerical Analysis*. Philadelphia, PA, USA: SIAM, 1996.

[37] R. Brent, *Algorithms for Minimization Without Derivatives*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1973.

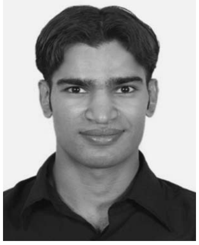
[38] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1976.

[39] N. K. Verma, A. Roy, and Y. Cui, “Improved mountain clustering algorithm for gene expression data analysis,” *J. Data Min. Knowl. Discov*, vol. 2, no. 1, pp. 30–35, 2011. [Online]. Available: http://www.bioinfo.in/uploadfiles/13155020612_1_2_JDMKD.pdf

Nishchal K. Verma (SM’13) received the Ph.D. degree in electrical engineering from the Indian Institute of Technology Delhi, Delhi, India.

He is currently an Assistant Professor with the Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur, India. His current research interests include soft computing, intelligent condition-based monitoring, machine learning, and computational intelligence and applications.

Dr. Verma is a fellow of the Institute of Electronics and Telecommunication Engineering, India.



Abhishek Roy received the B.Tech. degree in electrical and electronics engineering from the National Institute of Technology Karnataka, Surathkal, India, in 2010.

From 2010 to 2013, he has been with the Accenture Services Private Ltd., Gurgaon, India, as a Software Engineer, where he worked on various technological platforms including Salesforce Cloud Computing and Statistical Analysis Software Business Intelligence. He would be working toward the M.S. degree in computer science from 2013 to 2015 in Texas A&M University, College Station, USA. He is the author of various research papers in international conferences and reputed journals. His research interests include data mining, machine learning, clustering, and microarray data analysis.

Mr. Roy was a recipient of the O.P. Jindal Engineering and Management Scholar Award in 2007.