SELF-OPTIMAL CLUSTERING TECHNIQUE USING OPTIMIZED THRESHOLD FUNCTION

Aditya Nitin Patil(230074)
Anurag Sharma(230174)
Nonit Gupta (230712)
Sachin Kumhar (230891)

INTRODUCTION

- Clustering is the process of organizing similar data points into groups, called clusters, such that points within the same cluster are more similar to each other than to those in different clusters.
- Existing techniques like K-means, Fuzzy C-Means, and Expectation Maximization often fail to find optimal clusters in all datasets due to built-in assumptions.
- The proposed Self-Optimal Clustering (SOC) method improves on earlier techniques by optimizing the threshold function using interpolation, leading to better cluster quality.
- SOC is evaluated using standard validation indices (like Silhouette, Partition, Separation, and Dunn Index) and shows superior performance with more accurate and visually distinct clusters.

BACKGROUND

GENERAL OVERVIEW

- Newer techniques enhance clustering accuracy and efficiency:
  - Local data variation, normalized cuts, feature space analysis, saddle point detection, color-texture analysis, multiresolution segmentation, etc.
- Challenges with Traditional Segmentation:
  - Typically cluster based on 2D/3D spatial proximity, effective only for uniform, homogeneous data.
  - Struggle with complex, irregular data structures
- Hyperspace-Based Clustering:
  - Operates in hyperspace using multi-feature data (e.g., height, weight, color)
  - Ignores physical constraints, enabling greater flexibility for non-uniform data

PROBLEM STATEMENT

- Traditional clustering algorithms:
  - Need manual tuning (e.g., number of clusters)
  - Depend on fixed or heuristic threshold values
  - Often give inconsistent or overlapping clusters
- IMC-1 and IMC-2 are better but still use non-optimized thresholds
- Key Problem: Current methods can't automatically adjust themselves to always find the best clustering.

Proposed Solution – SOC

SOC = Self-Optimal Clustering

- Improves on IMC by optimizing the threshold function
- Uses Lagrange interpolation to compute best-fit threshold
- Iteratively updates clusters until quality is maximized
- Evaluates clusters using indices like:
  - Global Silhouette Index (GSI)
  - Partition Index (PI)
  - Separation Index (SI)
  - Dunn Index (DI)

MEASURE OF CLUSTER QUALITY

Silhouette Index (GSI) --

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

Dunn Index (DI) --

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

- Higher DI → Better clustering
  - Ratio of minimum inter-cluster distance to maximum intra-cluster distance.

MEASURE OF CLUSTER QUALITY

Partition Index (PI) --

- Ratio of intra-cluster compactness to inter-cluster separation.

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

Lower PI → Better clustering

- Clusters are more compact (points are closer to their center)
- Clusters are better separated from each other

Separation Index (SI)--

- The Separation Index checks how far apart the clusters are from each other, and also looks at how tightly packed the points are within each cluster.

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

SOC-TECHNIQUE

ALGORITHM STEPS

- Step 1)
  - To fit all the points in an hyperspace, we normalize the points:

    $xj = (xj - (x)min)/((x)max - (x)min)$ for $j = 1, 2, 3, \ldots, n$
- Step 2)
  - Threshold Function (δm) Calculation: $\delta_m = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{\min(\mathbf{x}^j)}{\sum_{i=1}^{D} x_i^j} \right) \cdot (\beta_m).$
- Step 3)
  - Potential Calculation: Each point's closeness to all other points

    $P_m^r = \sum_{j=1}^{n} \exp\left(-\frac{d^2(r,j)}{\delta_m^2} \right)$ where $d^2(r,j) = (r-j)Q(r-j)^T$
- Step 4)
  - Select Cluster Centre: Choosing point with highest potential

    $c_m = x^* \quad \text{where} \quad P_m^* = \max_r P_m^r$

- Step 5)
  - Assign all points to cluster if their distance from centre is less than δm

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

- Step 6)
  - Eliminate Assigned Points
- Step 7-8)
  - Repeating and distributing remaining points to nearest centers.
- Step 9)
  - Calculating GSI $s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad -1 < s(i) < 1 \qquad S_m = \frac{1}{N_m} \sum_{i=1}^{N_m} s(i). \qquad GSI = \frac{1}{M} \sum_{m=1}^{M} S_m.$

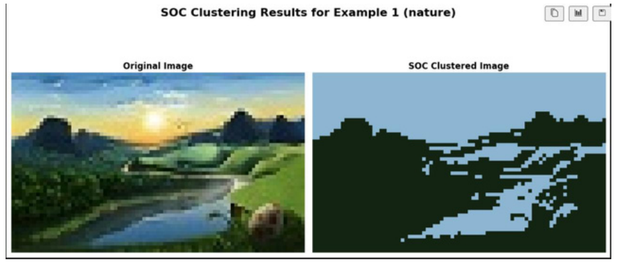
- Step 10-14)
  - Optimizing βm via Lagrange Interpolation

$$S_t = \sum_{m=1}^{M} S_m \cdot l_m(\delta_t) \qquad \beta_m = \frac{\eta}{\delta_m}, \quad m = 1, 2, \ldots, M$$

- Step 15)
  - Repeat steps 2-14 for 10 iterations or until δm converges

COMPARISON

.Nature Image (Optimal Clusters = 2)



Method	GSI	PI	SI	DI
IMC-1	0.8988	0.0006	4.3591	0.8324
IMC-2	0.8988	0.0006	4.3591	0.8324
IMC-max	0.8977	0.0006	4.4522	0.8587
IMC-half	0.8988	0.0006	4.3674	0.8324
SOC	0.8988	0.0006	4.4123	0.8324
K-means	0.8988	0.0007	4.5685	0.8324
FCM	0.8988	0.0007	4.5636	0.8324
EM	0.8972	0.0007	4.5639	0.8302
K-medoid	0.8988	0.0006	4.472	0.8324

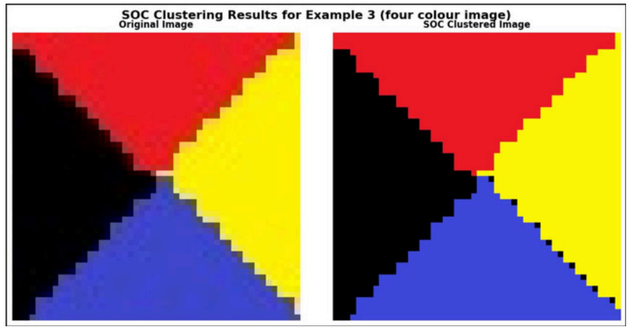
- IMC-max gave the best results with high GSI and DI and low PI, while SOC also performed well with balanced values, but EM gave the worst results with low GSI and very high SI.

Bottle with Back-lighting (Optimal Clusters = 4)

Method	GSI	PI	SI	DI
IMC-1	0.5938	0.0013	17.0855	0.5639
IMC-2	0.5896	0.0014	18.8696	0.4932
IMC-max	0.6485	0.0039	3.6483	0.8537
IMC-half	0.6332	0.0039	4.6574	0.6754
SOC	0.6043	0.0012	10.1565	0.7604
K-means	0.5926	0.0013	16.1668	0.4923
FCM	0.5913	0.0012	16.7719	0.5015
EM	0.2442	0.0018	111.3922	0.0921
K-medoid	0.5709	0.0015	19.9569	0.3486

- IMC-max had the highest GSI and DI but a higher PI than SOC, which gave balanced results with low PI and good SI, while EM again performed poorly with very high SI showing overlapping clusters.

Four-Color Image (Optimal Clusters = 4)



Method	GSI	PI	SI	DI
IMC-1	0.8988	0.0006	4.3591	0.8324
IMC-2	0.8988	0.0006	4.3591	0.8324
IMC-max	0.8977	0.0006	4.4522	0.8587
IMC-half	0.8988	0.0006	4.3674	0.8324
SOC	0.8988	0.0006	4.4123	0.8324
K-means	0.8988	0.0007	4.5685	0.8324
FCM	0.8988	0.0007	4.5636	0.8324
EM	0.8972	0.0007	4.5639	0.8302
K-medoid	0.8988	0.0006	4.472	0.8324

- All methods gave almost the same results in this case, with SOC, IMC versions, K-means, and FCM all reaching high GSI around 0.8988. The clustering was equally good across methods, with low PI and well-separated clusters.

RESULT AND DISCUSSION

- Clustering performance was evaluated using four standard indices:
  - GSI (Silhouette), PI (Partition), SI (Separation), and DI (Dunn Index).
- While some traditional methods (like K-means and FCM) showed high GSI, they had higher PI and SI, indicating less compact and more overlapping clusters.
- EM consistently performed the worst, with low GSI and very high SI, showing poor cluster separation
- SOC demonstrated balanced performance across all four indices in each case.
  - It maintained low PI and SI (indicating compact and well-separated clusters),
  - achieved competitive GSI and DI scores, even without any preprocessing.

CONCLUSION

This paper presents the Self-Optimal Clustering (SOC) technique as an optimized extension of the IMC method. While SOC does not achieve the best results in all cases, it consistently delivers competitive and reliable segmentation performance, especially in real-world image scenarios.

In some benchmarks, IMC-max outperforms SOC in terms of cluster quality indices, and IMC-2 also shows results very close to SOC. However, both rely on heuristic modifications, while SOC is based on a mathematically optimized threshold function.

Despite not always being the best, SOC demonstrates good clustering accuracy and can be effectively applied in practical image segmentation tasks, proving its real-world usability.

IMPLEMENTATION DEMO

THANK YOU