Enhanced Affinity for Spectral Clustering using Topological Node Features (TNFS)

Lalith Srikanth Chintalapati, Raghunatha Sarma Rachakonda

Abstract*: Data clustering is an active topic of research as it has applications in various fields such as biology, management, statistics, pattern recognition, etc. Spectral Clustering (SC) has gained popularity in recent times due to its ability to handle complex data and ease of implementation. A crucial step in spectral clustering is the construction of the affinity matrix, which is based on a pairwise similarity measure. The varied characteristics of datasets affect the performance of a spectral clustering technique. In this paper, we have proposed an affinity measure based on Topological Node Features (TNFs) viz., Clustering Coefficient (CC) and Summation index (SI) to define the notion of density and local structure. It has been shown that these features improve the performance of SC in clustering the data. The experiments were conducted on synthetic datasets, UCI datasets, and the MNIST handwritten datasets. The results show that the proposed affinity metric outperforms several recent spectral clustering methods in terms of accuracy.

Index Terms: Spectral clustering, Affinity matrix, Graph theory, Topological Node Features.

I. INTRODUCTION AND RELATED WORK

Many real-world pattern recognition applications such as social community detection [21], document clustering [17], health analytics [13], etc., require clustering technique as a critical step for their realization. Since the data space corresponding to a real-world application, in general, contains complex structures, traditional algorithms • such as k-means or single linkage fail to produce useful results. The reason for this is because these clustering algorithms are ideal only for discovering globular clusters or well-separated clusters. Spectral Clustering (SC) [19, 32] has become popular in recent times due to its advantages over the classical algorithms in handling complex data. The Fig. 1 illustrates the prowess of SC. We note that SC performs well even in the case of non-convex datasets such as Flame and two moon datasets.

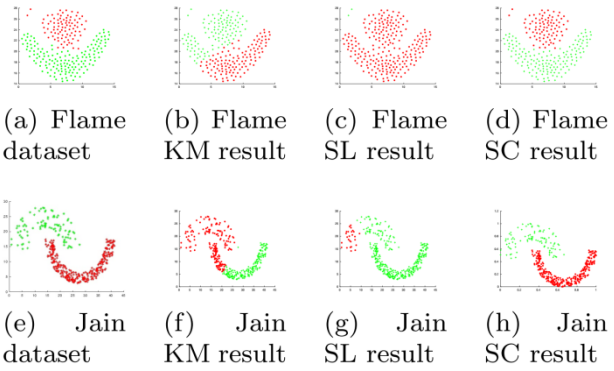


Fig. 1: First column represents the original datasets. The second and third columns represent the results of Kmeans (KM) clustering and Single Linkage (SL) methods respectively. The fourth column represents the results obtained using Spectral Clustering.

Instead of clustering in the original space, SC first maps the data points to a new space with reduced dimensionality, where the similarities become more apparent, and the clustering becomes easier. The mapping is achieved by projecting every data point onto an Eigenspace of the square kernel matrix, which essentially makes use of pairwise affinity among all the data points. SC is a less demanding yet powerful technique, as it requires only pairwise similarity or affinity among data points. It is entirely data-driven and easy to implement, thus making it suitable for a variety of applications [2].

Approaching the SC from a graph-cut point of view, the SC method removes inter-cluster edges which are weak and retains the strong intra cluster edges to form optimal clusters of the given data modelled as a graph. In order to obtain balanced and non-trivial clusters, Normalized cuts method [22] was proposed by Shi and Malik. The solution to the graph cut problem using Normalized cuts is NP-Hard. Spectral clustering is a solution to the relaxed versions of the graph cut problem [26]. The Eigen spectrum of the graph laplacians is utilized to arrive at the solution provided by the SC. Please refer to Luxburg tutorial [26] for a detailed explanation on the working of SC.

Fig. 2: Schematic diagram of spectral clustering.

Revised Manuscript Received on October 15, 2019
***** Correspondence Author
Lalith Srikanth Chintalapati, Department of Mathematics and Computer Science, Sri Sathya Sai Institute of Higher Learning, Puttaparthi, India. Email: lalithsrikanthc@sssihl.edu.in
Raghunatha Sarma Rachakonda, Department of Mathematics and Computer Science, Sri Sathya Sai Institute of Higher Learning, Puttaparthi, India. Email: rraghunathasarma@sssihl.edu.in

The schematic diagram of SC is displayed in Fig. 1. One of the crucial steps in SC is the construction of the affinity matrix by using an affinity metric. In SC, typically the affinity is defined by Gaussian kernel weighted Euclidean distance [19] as shown in (1).

$$A_{ij} = \exp\left(\frac{-||x_i - x_j||^2}{2\sigma^2}\right) \quad (1)$$

where $||x_i - x_j|| = \sqrt{\sum_{k=1}^{l} |x_{ik} - x_{jk}|^2}$ is the Euclidean distance between l dimensional data vectors, x_i, x_j and σ is the width of the Gaussian kernel. It is also known as standard deviation or Gaussian spread parameter [27].

In SC, the data is generally modelled as one of the three types of graphs viz. (i) ϵ-graph (ii) KNN graph (iii) Fully connected graph. In ϵ-graph, two data points are connected by an edge if the distance between them is less than ϵ. In the KNN graph, all the data points are connected to their K nearest neighbours. In a fully connected graph, all the points are connected to all other points in the dataset. The given data is modeled as one of these three types of graphs, and the affinity matrix of SC is evaluated from the corresponding graph. In recent years, many works have been proposed to enhance the affinity measure. They can be categorized into the following three groups:

A. *Scale Estimation Based Methods*

The methods under this category improvise on the definition of pairwise similarity metric by suitably estimating σ (scale) in (1). When the data is distributed in different scales, the computation of the pairwise affinity often leads to sub-optimal results. To overcome this issue, Zelnik and Perona [32] proposed Self-Tuning Spectral Clustering using local scaling parameter in the construction of the affinity matrix. Extending this work, Gu and Wang [9] proposed an affinity using local properties such as distance between a point and its k neighbours. V_i is the set of k nearest neighbours of x_i. The local scale σ_i is computed as the average distance from a point to its k-nearest neighbours. Using this local sigma, an affinity between x_i and x_j is defined as

$$A_{ij} = \exp\left(\frac{-||x_i - x_j||^2}{\sigma_i \sigma_j}\right) \quad (2)$$

B. *Density Based Estimation Methods*

The methods in this category incorporate the density information into the affinity definition. For practical purposes, the density is defined as the number of data points in n-dimensional unit space. The data with varying density can lead to inaccuracies in clustering results. In order to address this issue, a density-based similarity metric for the construction of the affinity matrix was proposed by Yang et al. [30]. The authors used KNN graph to model the data. If two nodes in this graph are connected by a path going through a high-density region, then they are considered to be similar. This method used the density of the regions between the corresponding points to define the similarity between them. In a different approach, Beauchemin [3] proposed a method to construct the affinity matrix employing a K-means based density estimator with the sub-bagging procedure. The method is based on the theoretical work of Wong [28], in which asymptotic properties of K-means were used for density estimation. The sub-bagging procedure is then employed to estimate the density more accurately. Using this robust density estimator, Beauchemin was able to propose a new density based similarity measure.

C. *Neighbourhood Based Methods*

In these methods, the neighbourhood-based information is incorporated into a pairwise affinity definition. Using one such neighborhood-based information called Common Nearest Neighbours (CNN) between two nodes, Zhang et al.[33] proposed an affinity measure in (4) given below. For two nodes x_i, x_j of a graph, $CNN(x_i, x_j)$ is defined to be the number of nodes in the intersection region of the ϵ-neighborhoods around x_i and x_j. ϵ-neighborhood (N_{x_i}) of x_i is defined as follows:

$$N_{x_i} = \{x_j | d(x_i, x_j) < \epsilon\} \quad (3)$$

$$A_{ij} = \begin{cases} \exp\left(\frac{-d(x_i,x_j)^2}{2\times\sigma^2\times(CNN(x_i,x_j)+1)}\right) & i \neq j \\ 0 & i = j \end{cases} \quad (4)$$

where x_i, x_j $\in$ S, the set of all data points. σ is the Gaussian scale parameter. CNN (x_i, x_j) is the number of common neighbours between the nodes x_i and x_j in the corresponding ϵ-graph generated from S. Thus, the CNN attribute brings in the notion of connectivity into the similarity measure.

Diao et al. [8] try to capture the spatial structure about the neighbourhoods of the correlative points by using Local Projection Distance Measure (LPDM) based similarity metric. They define Local-Projection-Neighbourhood (LPN) for any two points x_i, x_j as the overlapped region with specified Euclidean radius d(x_i, x_j) and consider the projections of all the m points in LPN onto the line joining x_i, x_j. An adjustable projection distance L is defined between these projected points $\{x_1, x_2, ..., x_m\}$.

$$L(x_i, x_j) = 10^{\rho d(x_i, x_j)} \quad (5)$$

where $d(x_i, x_j)$ is the Euclidean distance between the projected points x_i and x_j and ρ is the flexing factor. A novel similarity measure $S(x_i, x_j)$ is then defined, which is inversely proportional to the sum of all the projection distances as

$$S(x_i, x_j) = \frac{1}{\sum_{l=0}^{m} L(x_l, x_{l+1})} \quad (6)$$

where m is the number of projection points in LPN, $x_0 = x_i$ and $x_{m+1} = x_j$. The flexing parameter ρ and the parameter k used to construct K-nearest neighbour graph for reducing computational complexity play a crucial role in this method.

Affinity matrix enhancement based on "Neighbour propagation" was proposed by Li and Guo [16]. In this method, neighbourhood information is incorporated into the affinity matrix to propagate the similarity. Authors construct the distance matrix $B = (b_{ij})$ by computing the Euclidean distance between each pair of points, and compute the

similarity matrix $W = (w_{ij})$, containing pairwise similarity between each pair of points using Gaussian weighted kernel. They construct the neighbour relation matrix $N = (n_{ij})$, using distance threshold ϵ, that is, if $b_{ij} < \epsilon$, then n_{ij} =1 and n_{ji} =1. Then the matrices N and W are updated according to the neighbour propagation principle: If n_{ij} =1, n_{jk} =1 and n_{ik} =0, then set n_{ik} =1 and n_{ki} =1, simultaneously, update w_{ik} and w_{ki} as min(w_{ij}, w_{jk}).

Ye and Sakurai [31] proposed a method based on shared neighbours for enhancing affinity matrix. In this method, the influence of shared neighbours is employed to define two types of similarities between data points: 1) Number of shared neighbours and 2) Closeness of shared nearest neighbours. N_i is the set of neighborhood points of x_i. The shared nearest neighbours is defined as:

$$N_i \cap N_j = \begin{cases} N_i \cap N_j \cup \{x_{ij}'\}, & if\, x_1 \leftrightarrow x_2 \\ N_i \cap N_j, & otherwise \end{cases} \quad (7)$$

where x_{ij}' is a virtual point that represents x_i as a nearest neighbour of x_j and vice versa. The first type similarity defined by them is

$$A_{ij} = \frac{|N_i \cap N_j|}{k} \quad (8)$$

where $|N_i \cap N_j|$ is the number of shared nearest neighbours between X_i and X_j, and k is the maximum number of nearest neighbours shared between points x_i and x_j in the directed KNN graph. Second similarity they proposed is based on the closeness among the shared nearest neighbours. The shared nearest neighbours in $N_i \cap N_j$ are weighed according to their orders relative to the data points x_i, x_j. Let w_{ij} denote the weight of the shared nearest neighbours in $N_i \cap N_j$. If $x_r \in N_i \cap N_j$ and x_r is l_r^{th} neighbour of x_i and o_r^{th} nearest neighbour of x_j, then the weight is calculated as

$$w_{ij} = \sum_{x_r \in N_i \cap N_j} (k - l_r + 1)(k - o_r + 1) \quad (9)$$

using these weights, pairwise similarity SC-cSNN is calculated as

$$A_{ij} = \frac{w_{ij}}{w_{max}} \quad (10)$$

w_{max} is the maximum of all the weights w_{ij} between x_i and x_j. This work utilizes the nearest neighbours and their similarities, thereby accounting for the "closeness and local structure" of the data points. However, they do not incorporate density information effectively. Because of this, the method fails to cluster some of the datasets such as Aggre and Pathbased [15] accurately, as demonstrated in the results (section 4).

Arais et al.[1] proposed a spectral clustering algorithm based on features from local Principal Component Analysis (PCA). The algorithm uses local covariance matrices for enhancing the affinity measure. It is derived as follows: Let a set of points $S = \{x_1, \dots, x_n\}$ in R^l and the parameters: neighborhood radius $r > 0$, spatial scale $\epsilon > 0$, projection scale $\eta > 0$, intrinsic dimension d, number of clusters K be given. A set of n_0 centers $\{y_1, \dots, y_{n_0}\}$, are selected from the given data as explained below. Each time a y_j is picked randomly from the dataset such that, it is not contained in the neighborhoods $N_r(y_i)$ of previously chosen y_is.

For each i = 1, ..., n_0, the covariance matrix C_i of $N_r(y_i)$ is computed. Let Q_i denote the orthogonal projection of y_i onto the space spanned by the top d eigenvectors of C_i. Affinity between center pairs is defined as follows:

$$A_{ij} = \exp\left(\frac{||y_i - y_j||^2}{\epsilon^2}\right) \times \exp\left(\frac{||Q_i - Q_j||^2}{\eta^2}\right) \quad (11)$$

After applying spectral graph partitioning to A, the data points are clustered according to the closest center with respect to the Euclidean distance. Again, the main motive of this method is to capture the local structure into the construction of affinity matrix.

D. *Other Recent Methods*

In other recent methods, Yang and Wu [29] proposed a robust prototype clustering method. The authors propose the similarity based on modified Gaussian similarity function between the data points x_i, and the cluster centers z_j as:

$$S(x_i, z_j) = \left(\exp\left(\frac{-||x_i - z_j||^2}{\beta}\right)\right)^{\gamma} \quad (12)$$
$$i = 1, \dots, n, \quad j = 1, \dots, c,$$

where β is a normalized term used to control the size of the neighborhood and $\gamma (> 0)$ is the power parameter. Yang and Wu also proposed correlation comparison algorithm (CCA) for estimating γ. The intuition behind using the powered version is to minimize the sensitivity of β, so that it can be assigned a global value of the sample variance:

$$\beta = \frac{\sum_{i=1}^{n} ||x_i - \bar{x}||}{n}, \text{with} \quad \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \quad (13)$$

Inspired by this formulation of similarity, powered spectral clustering has been proposed by Nataliani et al. [18]. In this method, the similarity measure is given as:

$$A_{ij} = \exp\left(\frac{-||x_i - x_j||^2}{\beta}\right)^{\gamma} \text{ for i} \neq \text{j and 0 otherwise} \quad (14)$$

where β represents neighborhood size, and $\gamma > 0$ is the power parameter. In order to estimate optimal β the authors use the maximum value among all minimum distances between data points, viz:

$\beta = \max_i(\min_{j \neq i}||x_i - x_j||)$ for all x_i and x_j in the dataset.

Challa et al.[4] proposed spectral clustering using Power Ratio cut or PRcut. They aimed to come up with an efficient implementable algorithm which is faster than traditional spectral clustering and also preserves the accuracy. The data is pre-processed using Minimum Spanning Tree (MST) to reduce the size substantially and thereby saving on computing time. This faster alternative to the spectral clustering algorithm is obtained by considering the Γ-limit of spectral clustering methods. For this purpose, a discretization scheme was used for calculating the Γ-limit. The actual dividends of PRcut can be observed as the data size grows.

E. *Topological Node Features based Spectral Clustering (SC-TNF)*

All the techniques that are presented above from the

literature show that the local information plays a vital role in enhancing the affinity matrix to improve clustering results. To capture the local properties effectively, the Topological Node Features (TNFs) [7] have been utilized from the graph representation of the data. A TNF is mainly defined as topological information when viewed from any particular node of a graph. As an example, the degree of a node can be considered as a TNF.

TNFs are typically employed in solving graph isomorphism problems in the literature. The goal of graph isomorphism problem is to find a subgraph inside a larger graph. Cordella et al. [6] have used the degree of a vertex, for identifying a subgraph isomorphism. TNFs have been used in Sorlin and Solnon [23] to solve the subgraph isomorphism problem, as they capture the local structure in the data effectively. Dahm et al. [7] used novel TNFs for speeding up subgraph isomorphism detection. The work of Dahm et al. [7] has been used for building the affinity measure that effectively captures local information. The main TNFs used in the proposed affinity measure are degree, Clustering Coefficient (CC), and Summation Index (SI).

We have successfully strengthened the notion of affinity by combining Gaussian kernel function and TNFs. Most of the methods in literature use only one of the following features, viz. common neighbours, density, and local structure. In the case of the proposed algorithm, affinity measure is defined using multiple local features, thereby enhancing the affinity matrix construction. In the following sections, we show how these features have improved the clustering results.

F. *Contributions*

• Affinity measure has been redefined to include a broad spectrum of local characteristics such as local density, spatial nearness, and structural similarity.

• The local characteristics of the data are mapped to the TNFs of the corresponding graph appropriately. For example, the clustering coefficient TNF is used for estimating the local density measure.

If a method is based on only connectivity or density, it is limited by that feature's ability to distinguish the data points. Since the proposed method uses a combination of multiple features, the datasets with more than one characteristic can be clustered effectively. From the results obtained in section 4, it has been shown how different types of data are clustered with improved accuracy.

The outline of the paper is as follows: In section 2, the necessary background is briefly presented, wherein the traditional SC algorithm by Ng et al. [19] (referred as NJW) and the theory associated with SC and TNFs are described. In section 3, the proposed algorithm for affinity matrix creation using TNFs is presented. The results obtained in comparison with the state of the art techniques in SC are discussed in Section 4. In section 5, we conclude the paper and suggest possible future extensions.

II. BACKGROUND

In this section, we briefly present the required technical details of SC method and also the spectral clustering procedure proposed by Ng et al. [19].

A. *Spectral Clustering Approach*

Let $X = \{x_1, \ldots, x_n\}$ in R^l be the set of data points under consideration. The undirected ε-graph G = (V, E), where $V = \{v_1, \ldots, v_n\}$ the set of nodes and E is the set of edges, is constructed from X as follows. If the distance between any two data points x_i and x_j is less than a given parameter $\varepsilon > 0$, then the corresponding nodes v_i, v_j in the graph G are connected via an edge, denoted as (v_i, v_j) i.e. $E = \{(v_i, v_j) | d_{x_i x_j} < \varepsilon \quad \forall v_i, v_j \in V\}$, where $d_{x_i x_j}$ is the distance between the data points x_i and x_j. Let each edge (v_i, v_j) be weighted by $s_{ij} \geq 0$ which denotes the similarity specified by the user (1) between the data points x_i and x_j.

The critical idea in spectral clustering is to find the optimal embedding of the data in low dimensional space. The main ingredient for spectral clustering is the graph Laplacian matrix whose eigenvectors are used to find k-dimensional embedding of the data. In the work of Ng et al.[19], we note that the top k eigenvectors of graph Laplacian are used to find the embedding of the data. The process of evaluating the Laplacian matrix is as follows: The affinity matrix which captures the pairwise affinity between all the points is defined as $A = (s_{ij})$. From the affinity matrix A and Degree matrix D ($D_{ii} = \sum_j W_{ij}$), a normalized Laplacian matrix is constructed as: $L = D^{-1/2} A D^{-1/2}$. The eigenvectors obtained from the Laplacian L are used to find k-dimensional embedding of the data.

The spectral clustering procedure proposed by Ng et al. [19] is described in Algorithm 1. As mentioned in the algorithm the input parameters of k and σ are provided by the user along with the data points. In step 1, the affinity matrix is constructed using the Gaussian kernel. From the definition, it can be seen that affinity is inversely proportional to the distance between points. In steps 1 and 2, the pairwise similarity between points is captured, and the graph Laplacian matrix (L) is constructed. After performing eigenvalue decomposition of the Laplacian matrix (L), all the eigenvectors are sorted in descending order according to their corresponding eigenvalues. The first k eigenvectors are then arranged as columns to form the matrix X. Next, row-wise normalization is carried out on X to produce Y. Now, the rows of Y can be considered as k-dimensional tuples, which represent the data points. In steps 5 and 6, the k dimensional points are clustered using a simple clustering algorithm such as K-means.

Algorithm 1: Simple spectral clustering

Input: A set of points $S = \{x_1, \ldots, x_n\}$ in R^l, parameters k, σ

Output: k clusters of the given data

1. Form the affinity matrix $A \in R^{n \times n}$ defined by $A_{ij} = \exp(-||x_i - x_j||/2\sigma^2)$
2. Define D to be the diagonal matrix whose (i, i)-element is the sum of i-th row of A, and construct the matrix $L = D^{-1/2} A D^{-1/2}$.
3. Find $e_1, e_2, \ldots, e_k$,

the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $X = [e_1 \quad e_2 \quad \dots \quad e_k] \in R^{n \times k}$ by considering the eigenvectors as columns of the matrix.

4. Form the matrix Y from X by re-normalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2}$).
5. Treating each row of Y as a point in R^k, cluster them into k clusters via K-means or any other algorithm (that attempts to minimize distortion).
6. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

B. *Proposed Affinity Measure*

In this section, the TNF based framework is presented and the novel affinity metric is proposed. The following subsection describes the three TNFs used in this work.

C. *TNF Based Framework*

A TNF is essentially defined as topological information, when viewed from any particular node of the graph. Let G = (V, E) be the given graph where, V denotes the set of nodes: $V = \{x_1, x_2, \dots, x_N\}$ and E is the set of edges. The following are the TNFs used in the proposed algorithm:

1. The degree of a node x of G: This is given by $degree(x) = |\{y \in V | (x, y) \in E\}|$, where $|.|$ is the cardinality of the set.

2. Clustering Coefficient (ϕ_x) of node x: This is the number of edges that exist among the vertices in $N(x)$, i.e. $\phi_x = |\{(l, m) \in E | l, m \in N(x)\}|$, where $|.|$ is the cardinality of the set. We observe that ϕ_x gives an intuitive understanding of local density at x.

3. The Summation Index (SI): This is used to capture the structural information. SI is a way of summarizing TNF of a graph. Thus, it gives the power to encode neighboring structural characteristics into one value.

Dahm et al. [7] in their work define SI_i recursively as a sum of node degrees of neighboring nodes as:

$$SI_i(x) = \begin{cases} \text{degree}(x) & if\ i = 0 \\ \sum_{y \in N(x)} SI_{i-1}(y) & if\ i > 0 \end{cases} \quad (15)$$

Fig. 2 shows the computation of SI_1 from SI_0 in one iteration and computation of SI_2 from SI_1.

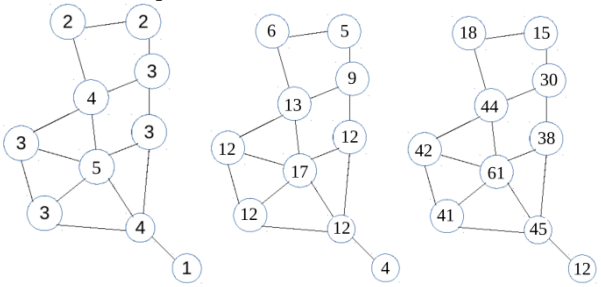


Fig. 3: (a) Initial TNF values SI_0 (b) SI_1 after Iteration 1 of (15). (c) SI_2 after iteration 2 of (15).

For every node x in G, let $\mathcal{V}_x$ denote the vector in $\mathcal{R}^3$ obtained by applying the (15) for $i = 0, 1$ and 2, i.e.

$$\mathcal{V}_x = (SI_0, SI_1, SI_2) \quad (16)$$

The vector $\mathcal{V}_x$ captures local structure, as additive information at various levels. For any x_i, x_j in G, let ζ_{ij} denotes the Euclidean distance between $\mathcal{V}_{x_i}$ and $\mathcal{V}_{x_j}$: $\zeta_{ij} = ||\mathcal{V}_{x_i} - \mathcal{V}_{x_j}||$, where $||.||$ is the Euclidean norm. Thus, ζ_{ij} quantifies how much x_i structurally differs from x_j. In addition to these TNFs, the proposed affinities utilize another feature (η_{ij}), the number of common points in the neighborhoods of x_i, x_j, that is $\eta_{ij} = |N(x_i) \cap N(x_j)|$. Intuitively, a higher value of η_{ij} indicates that x_i and x_j belong to the same cluster [33].

Based on the TNFs defined above, two algorithms, SC-TNF1 and SC-TNF2 are proposed to construct the affinity matrix. In the algorithm SC-TNF1, the affinity metric incorporates the local density and the number of common nearest neighbours. The algorithm SC-TNF2 improves over SC-TNF1 by considering structural information (SI) in addition to the local density and the number of common nearest neighbours. The output of SC-TNF1 or SC-TNF2 is used in step 1 of Algorithm 1 for performing clustering.

Algorithm 2: The proposed SC-TNF1

Input: A set of points $S = x_1, \dots, x_n$ in R^l, σ
Output: Affinity matrix A

1. Model the data points as a graph G (as shown in Sec 2.1).
2. At each node x, calculate the following TNFs:
   a. Degree of node ($degree(x)$)
   b. Clustering coefficient (ϕ_x)
3. Compute the similarity α_{ij} between any two nodes x_i, x_j as:

$$\alpha_{ij} = \exp\left(\frac{-||x_i - x_j||^2 \delta_{ij}}{2\sigma^2}\right)\eta_{ij} \quad (17)$$

   where $\delta_{ij} = |(\phi_i - \phi_j)|$, η_{ij} is the number of common points between $N(x_i), N(x_j)$, and σ is the scale parameter of the Gaussian function.
4. Output affinity matrix $A = (\alpha_{ij})$

Algorithm 3: The proposed SC-TNF2

Input: A set of points $S = x_1, \dots, x_n$ in R^l, σ
Output: Affinity matrix A

1. Model the data points as a graph G (as shown in Sec 2.1).
2. At each node x, calculate the following TNFs:
   a. Degree of node ($degree(x)$)
   b. Clustering coefficient (ϕ_x)
   c. SI vector $\mathcal{V}_x = (SI_0, SI_1, SI_2)$
3. Compute the similarity α_{ij} between any two nodes x_i, x_j

as:

$$\alpha_{ij} = \exp\left(\frac{-||x_i - x_j||^2 \delta_{ij}}{2\sigma^2}\right)\eta_{ij} \quad (18)$$

where $\delta_{ij} = |(\phi_i - \phi_j)|$, η_{ij} is the number of common points between $N(x_i), N(x_j)$, and σ is the scale parameter of the Gaussian function.

4. Define the similarity β_{ij} between any two nodes x_i, x_j as:

$$\beta_{ij} = \alpha_{ij}\left(1 + \frac{1}{1+\log(1+\zeta_{ij})}\right) \quad (19)$$

where ζ_{ij} is the Euclidean distance between local structural information of x_i and x_j.

5. Output affinity matrix $A = ((\beta_{ij})$ depending on the choice of user.

In step 3 of SC-TNF1, δ_{ij} is defined as the difference between clustering coefficients, ϕ_i and ϕ_j. Since clustering coefficient (ϕ) denotes the local density, if two points x_i, x_j are from neighborhoods of similar density then, $\delta_{ij} = |(\phi_i - \phi_j)|$ is less. Thus, for the points with similar local density, the computed affinity will be higher. SC-TNF1 also incorporates spatial nearness through η_{ij}. If η_{ij} is high, then it signifies that the ε neighborhoods of x_i and x_j are overlapping. The ε-neighborhood of x_i represents the set of neighbourhood points of x_i. This gives a measure of the nearness of the corresponding data points and further strengthens their affinity measure. From step 4 of SC-TNF2, we note that affinity is inversely proportional to ζ_{ij}. A smaller value of ζ_{ij} implies that x_i and x_j are structurally closer and thus will have higher affinity. Both the proposed affinity measures, SC-TNF1 and SC-TNF2, strengthen or penalize the affinity according to local topological graph properties. Because of this, SC-TNF1 or SC-TNF2 based spectral clustering algorithm is found to perform better across different types of datasets.

In the following subsection, the effectiveness of the proposed affinity measures in comparison with standard Gaussian kernel weighted similarity (1) on a synthetic dataset is demonstrated. A comprehensive set of results, comparing the proposed methods with the state of the art SC techniques, is presented in Section 4.

D. *TNF Based Affinity Measure versus Gaussian Kernel Distance*

In this section, the proposed TNF based methods are compared with Gaussian Kernel-based similarity, as described in NJW [19] (1). The Compound dataset [15] is considered for the experiment, a part of which is shown in Fig. 3. This part of the dataset has two clusters, an outer cluster, which is sparsely spread and a central cluster which is dense. Consider the points a, b, c, d and e from the figure. According to the ground truth, point a belongs to the sparse outer cluster. The NJW [19] wrongly assigns point a to center cluster, whereas the proposed techniques clusters it correctly (Fig. 2) by assigning it to the outer cluster. All the three affinities between a and surrounding points$b$,c, d, e are shown in Table 4. From the Table 4, it is evident that in the case of NJW, the affinity between a and b is higher than the affinities between a and its other neighbours, namely c, d and e. Whereas, in the case of both SC-TNF1 and SC-TNF2, the affinity between a and e is more than the affinities between a and its other neighbours, namely c, d and b. This led to the correct clustering of point a.

The two variations of affinity, namely SC-TNF1 and SC-TNF2, are compared in Table 4 using a synthetic dataset, Compound [15]. A comparison of SC-TNF1 and SC-TNF2 with the state of the art techniques is presented in Section 4. From the results, it has been observed that the SC-TNF2 is better than SC-TNF1 in all the datasets except Glass, E-coli, and BC. Since for the majority of cases, SC-TNF2 is better than SC-TNF1, the affinity considered for performing experiments is SC-TNF2. We believe that due to the incorporation of additional structural information in SC-TNF2, it is better than SC-TNF1.

Fig. 4: Dataset 1

Table 1: Affinities between various points in Fig. 4

Affinity	NJW	SC-TNF1	SC-TNF2
(a,b)	2.62e-73	4.85e-44	2.82e-42
(a,c)	4.15e-96	1.14e-44	3.92e-58
(a,d)	6.55e-90	5.92e-53	1.91e-51
(a,e)	3.86e-91	6.75e-39	3.36e-37

III. RESULTS AND ANALYSIS

This section presents the results of the proposed method on three different types of datasets, namely synthetic datasets, real datasets [15] and MNIST handwritten dataset [14]. The experimental results show that, in most of the cases, the proposed method performs better than the state of the art methods. The proposed techniques are compared with the following methods: spectral clustering algorithm (NJW) by Ng et al. [19] (2002), Neighbour Propagation (NP) (2012) proposed by Li and Guo [16], Shared Nearest Neighbours (SNN) (2016) proposed by Ye and Sakurai. [31], Powered Gaussian (PG) (2017) by Nataliani et al. [18], Spectral clustering using Local PCA (LPCA) [1] (2017), Powered Ratio Cut (PRCUT) [4] (2018). In the experiments conducted, three metrics were used for comparison: Adjusted Rand Index (ARI) [20], Normalized Mutual Information (NMI) [24] and Clustering Error (CE) [11]. A detailed explanation of these metrics is

given in the Appendix. The values of NMI and ARI approach unity when the result is closer to the ground truth. The metric CE tends to null as the clustering accuracy increases.

A comprehensive set of experiments with SC-TNF1 and SC-TNF2 on all the datasets have been conducted. The comparative results are presented in Table 3 to Table 11. The results show that SC-TNF2 performs better than all the methods, including SC-TNF1, in the majority of the cases. While SC-TNF1 is comparable in its performance with the other methods, in our opinion SC-TNF2 performs better than SC-TNF1 as it takes into account, the additional local structural information through Summation Index.

The computational complexity:

Let the number of data points be n. The asymptotic computational complexity for calculating the TNFs in step 2 of the Algorithm 3 is $O(n^3)$. For finding the affinity matrix in steps 3 and 4, the complexity is $O(n^2)$. Thus, the overall complexity of the proposed algorithm is $O(n^3)$. The complexity is arrived at by calculating the number of unit calculations done in the algorithm. The complexity of SC-TNF2 is similar to the computational complexities of the Powered Gaussian proposed by Nataliani et al., 2017 [3] and LPCA proposed by Castro et al., 2017 [4]. Besides, the proposed method uses sparse matrices for Laplacian matrix calculation, making it time and space efficient.

A. *Experimental Setup*

There are two main parameters in the proposed methods, namely ε and σ. The ε parameter is used for mapping given data points into the vicinity graph. TNFs are highly dependent on ε parameter. The σ parameter is the Gaussian kernel width used in the definition of affinity. To estimate the ε, the following approach has been taken:

The data is normalized (Appendix), and the pairwise Euclidean distances are calculated for all the data points. The variable $dist\_max$ is assigned the maximum of these pairwise distances. The ranges for ε and σ are fixed as [0, $dist\_max$] and [0, 1] respectively. The parameter space spanned by ranges of ε and σ is searched for the optimal combination using the following method. The clustering outputs of the proposed algorithm for all the values of ε and σ parameters in their respective ranges were obtained. The Silhouette Index metric [12] was then used to evaluate each of these clusterings. The optimal ε is selected based on the maximum value of the Silhouette Index metric (Appendix). In the experimental studies, it was found that, for all the datasets, the search between 0.01 and 0.60 (with step size .01) was sufficient to find the optimal σ value. The robustness of the SC-TNF2 with respect to the σ parameter has been analyzed, and the results are presented in Section 4.5.

B. *Synthetic Datasets*

In the 2D synthetic datasets, five datasets were considered for the experiments, namely Compound, Aggre, Flame, Jain, and Pathbased [15]. The datasets present challenges such as varying density, connectedness, etc. For example, the Flame dataset does not have clear cut boundaries for the clusters. The Compound dataset has convex and concave clusters with some connectivity amongst them. In the Aggre dataset, there are clusters within clusters and clusters of different densities. In the Pathbased dataset, there are clusters with two different types of properties: connectivity among the data points and the varied shapes of clusters. A clustering algorithm needs to account for different types of data characteristics to effectively cluster them.

In Fig. 2, the results are displayed for different methods applied to various datasets. The first column in the figure represents the original datasets. The second and third columns represent the results of SNN and PG methods on the datasets, respectively. The fourth column represents the results obtained based on the proposed SC-TNF2 algorithm. For the sake of brevity, plots for only two methods are displayed, namely SNN and PG. The comparative results of SC-TNF2 with the other recent methods are displayed in Table 2. The columns of the Table 2 are the different methods in the literature of SC. The rows are the different synthetic datasets considered for experiments. From the results, it can be seen that the SC-TNF2 is better than the other methods in the case of Compound, Pathbased datasets. It is the same as other methods in Jain dataset and is not the best result in the case of Flame and Aggre datasets. In Flame dataset, the result is the same as in PG and NP methods, which is the second best result. The result of Aggre is the third best after LPCA and PRCUT.

In Table 3, the comparative results between SC-TNF2 and the other methods using the NMI metric are displayed. SC-TNF2 is better than other methods for the case of Jain and Pathbased datasets. The NMI value is lesser than the other methods in the case of Flame, Compound, and Aggre datasets. It is noted that the SC-TNF2 method has produced second best result for these three datasets. The results of the comparison of SC-TNF2 with the other methods using CE metric are displayed in Table 4. It is observed that the results of SC-TNF2 are better than the other methods in the case of Compound, Jain, and Pathbased datasets. SC-TNF2 method is the second best when compared to other methods with regard to Flame and Aggre datasets.

Thus, from the results, it is concluded that SC-TNF2 gives best results in almost half the cases under consideration using the three metrics under consideration. It is noted that the value of ϵ plays a significant role in the construction of the vicinity graph, and hence, the clustering outcome. In all the experiments conducted, the ϵ value is determined using the Silhouette Index method. The optimal value of parameter ϵ corresponds to the highest possible ARI metric value for the dataset under consideration. We have employed Silhouette Index (SI) metric for arriving at the ϵ value, which may not give us the optimal ϵ.

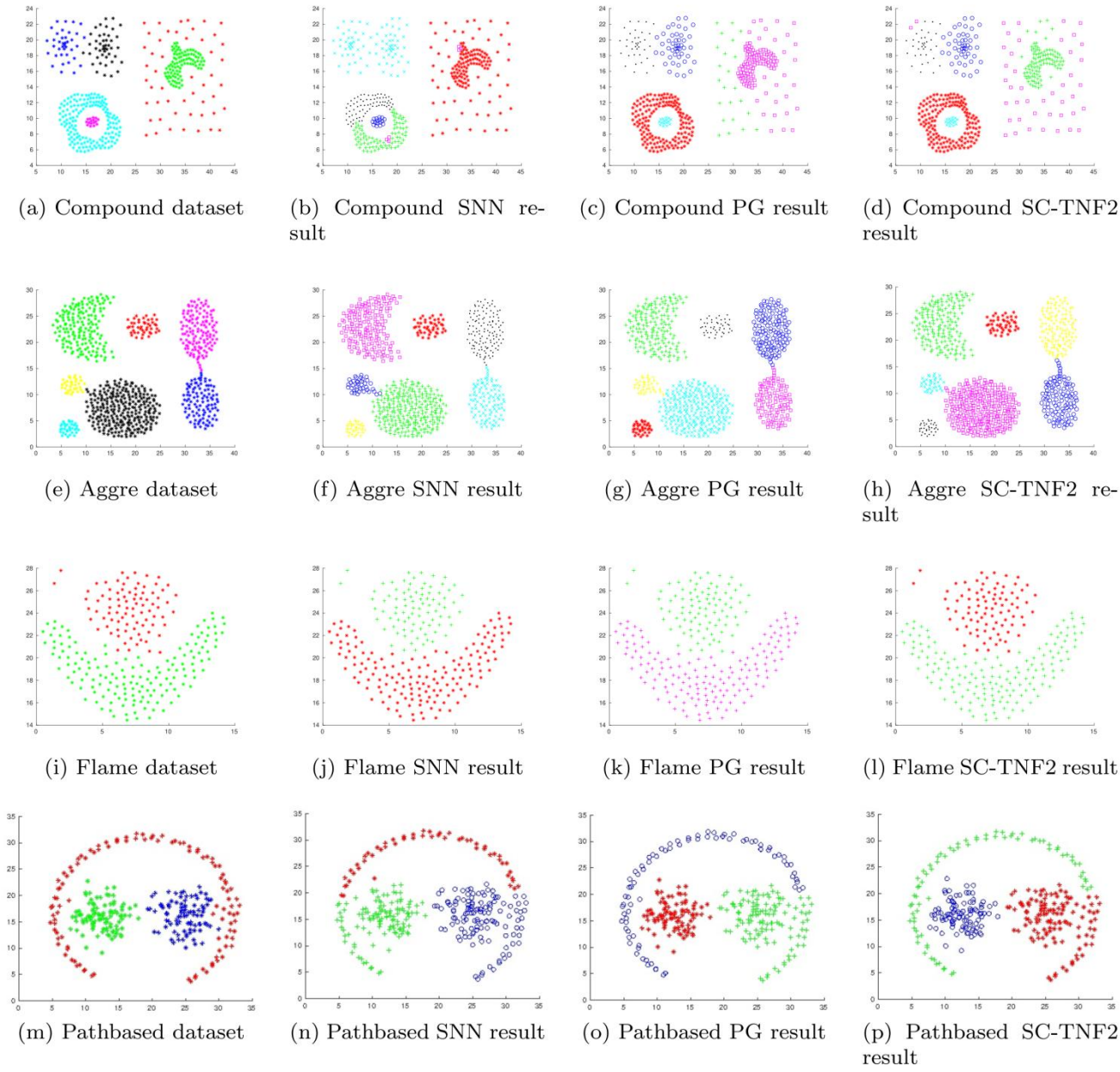
Further studies are required for improving the estimation of the optimal ϵ.

C. *Real World datasets*

The second type of datasets considered for the experimentation is the UCI real datasets [15]. These datasets are collected from real scenarios and have varied number of features and distributions. The number of clusters varies from 2 to 8. The number of data points varies from 13 to 699. The characteristics of five datasets considered are displayed in Table 5. Results of SC-TNF2 in comparison with other methods are given in Tables 6, 7, and 8. The comparison of SC-TNF2 with other methods

using the ARI metric is displayed in Table 6. From Table 6, it can be observed that SC-TNF2 is better than the other methods in the majority of the cases, except the five datasets of Glass, E-coli, Iris, Ion, and BC. SC-TNF1 gives the best result in the case of Glass dataset. However, for most of these datasets, SC-TNF2 is the second best result. The comparative results between SC-TNF2 and the other methods using NMI metric are presented in Table 7. In this table, SC-TNF2 obtained the best result in five datasets, namely Flea, Seeds, Soy, Wine, and Sonar. The results of the comparison of SC-TNF2 with other methods using CE metric are presented in Table 8. In this table, SC-TNF2 outperforms other methods in most cases except for the last five.

After compiling the results of three different metrics on UCI real datasets, it can be concluded that SC-TNF2 performs similar or better than other methods in the majority of the cases.



(a) Compound dataset (b) Compound SNN result (c) Compound PG result (d) Compound SC-TNF2 result

(e) Aggre dataset (f) Aggre SNN result (g) Aggre PG result (h) Aggre SC-TNF2 result

(i) Flame dataset (j) Flame SNN result (k) Flame PG result (l) Flame SC-TNF2 result

(m) Pathbased dataset (n) Pathbased SNN result (o) Pathbased PG result (p) Pathbased SC-TNF2 result

Fig. 5: First column represents the clusters in the original datasets. The second and third columns represent the clusters obtained through SNN and PG methods respectively and finally the last column shows the clusters obtained by proposed algorithm (SC-TNF2).

Table 2: Results of ARI metric on the synthetic datasets

				Methods				
Data sets	NJW	NP	SNN	PG	LPCA	PRCUT	SC-TNF1	SC-TNF2
Flame	**1.0000**	0.9666	0.9501	0.9666	0.9833	0.9666	0.5731	0.9666
Compound	0.9405	0.9450	0.5494	0.8955	0.8077	0.7810	0.7696	**0.9561**
Aggre	0.9869	0.8792	0.9840	0.9869	**0.9978**	0.9920	0.9869	0.9913
Jain	1.0000	0.6835	1.0000	1.0000	1.0000	1.0000	1.0000	**1.0000**
Pathbased	0.7143	0.6835	0.5434	0.6835	0.6346	0.6835	0.6954	**0.7275**

Table 3: Results of NMI metric on the synthetic datasets

				Methods				
Data sets	NJW	NP	SNN	PG	LPCA	PRCUT	SC-TNF1	SC-TNF2
Flame	**1.0000**	0.9355	0.8991	0.9269	0.8716	0.9355	0.5015	0.9355
Compound	0.9171	**0.9401**	0.7694	0.9119	0.8676	0.8109	0.7996	0.9395
Aggre	0.9824	0.9342	0.9799	0.9824	**0.9958**	0.9884	0.9824	0.9869
Jain	1.0000	0.7664	1.0000	1.0000	1.0000	1.0000	1.0000	**1.0000**
Pathbased	0.7825	0.7664	0.6082	0.7664	0.6059	0.7664	0.7726	**0.7895**

Table 4: Results of CE metric on the synthetic datasets

				Methods				
Data sets	NJW	NP	SNN	PG	LPCA	PRCUT	SC-TNF1	SC-TNF2
Flame	**0.0000**	0.0083	0.0125	0.0083	0.0167	0.0083	0.1208	0.0083
Compound	0.0526	0.1429	0.3308	0.0702	0.1153	0.2331	0.2331	**0.0276**
Aggre	0.0063	0.1320	0.0076	0.0063	**0.0013**	0.0038	0.0063	0.0051
Jain	0.0000	0.1300	0.0000	0.0000	0.0027	0.0027	0.0777	**0.0000**
Pathbased	0.1133	0.1300	0.1933	0.1300	0.1967	0.1300	0.1233	**0.1067**

Table 5: Attributes of real UCI datasets

Data set	Wine	Glass	Iris	Ion	Sonar
No of instances	178	214	150	351	208
No of attributes	13	9	4	34	60
No of clusters	3	6	3	2	2

Table 6: Results of ARI metric on the UCI real datasets

				Methods				
Data sets	NJW	NP	SNN	PG	LPCA	PRCUT	SC-TNF1	SC-TNF2
Flea	0.7448	0.8610	0.6531	0.7191	0.8615	0.6352	1.0000	**1.0000**
Seeds	0.7481	0.6645	0.7651	0.7031	0.7669	0.7145	0.7641	**0.8089**
Soy	0.8045	0.0528	0.9366	0.7477	0.7707	0.6610	1.0000	**1.0000**
Pima	0.0804	0.1338	0.0241	0.0472	0.0196	0.0320	0.1591	**0.1611**
Bupa	-0.0016	0.0122	0.0244	-0.0035	0.0045	0.0281	0.0008	**0.0391**
Wine	0.4127	0.9326	0.4058	0.3177	0.3454	0.1868	0.8804	**0.9471**
Sonar	0.0903	0.0111	0.0788	0.0085	0.0443	0.0088	0.0444	**0.1156**
Glass	0.2876	0.2630	0.2118	0.2262	0.2798	0.1804	**0.2978**	0.2671
E-coli	**0.7636**	0.7353	0.4156	0.7388	0.7188	0.3482	0.7474	0.7326
Iris	0.8161	**0.9222**	0.8512	0.7726	0.7880	0.8015	0.5638	0.8858
Ion	0.6647	0.7120	0.1636	0.1681	**0.7412**	0.0677	0.6438	0.6438
BC	0.8770	0.8716	**0.8934**	0.8824	0.8825	0.8283	0.8880	0.8878

Table 7: Results of NMI metric on the UCI real datasets

				Methods				
Data sets	NJW	NP	SNN	PG	LPCA	PRCUT	SC-TNF1	SC-TNF2
Flea	0.7172	0.8363	0.6736	0.6950	0.8360	0.6427	1.0000	**1.0000**
Seeds	0.7110	0.6409	0.7397	0.6812	0.7457	0.6893	0.7266	**0.7676**
Soy	0.8635	0.2677	0.9439	0.8479	0.8479	0.8301	1.0000	**1.0000**
Pima	0.0337	**0.1258**	0.0244	0.0182	0.0187	0.0528	0.1219	0.1219
Bupa	0.0140	0.0324	0.0324	0.0081	0.0324	**0.0412**	0.0307	0.0307
Wine	0.4554	0.9120	0.4462	0.4391	0.4598	0.3427	0.8694	**0.9276**
Sonar	0.1011	0.0749	0.1022	0.0105	0.0337	0.0679	0.1075	**0.1104**
Glass	**0.4670**	0.4320	0.3723	0.3993	0.4464	0.3769	0.4537	0.4347
E-coli	**0.7389**	0.6991	0.6175	0.6871	0.6954	0.4346	0.7197	0.6986
Iris	0.8058	**0.9011**	0.8449	0.7941	0.8224	0.7900	0.7355	0.8705
Ion	0.5463	0.6053	0.1286	0.1292	**0.6650**	0.1281	0.5573	0.5573
BC	0.7917	0.7830	**0.8246**	0.7984	0.8074	0.7256	0.8238	0.8074

Table 8: Results of CE metric on the UCI real datasets

				Methods				
Data sets	NJW	NP	SNN	PG	LPCA	PRCUT	SC-TNF1	SC-TNF2
Flea	0.0946	0.0541	0.1622	0.1081	0.0541	0.1622	0.0135	**0.0000**
Seeds	0.0905	0.1286	0.0857	0.1095	0.0857	0.1048	0.0857	**0.0667**
Soy	0.0851	0.5532	0.0213	0.1064	0.0851	0.1915	0.0000	**0.0000**
Pima	0.3385	0.3073	0.3385	0.3424	0.3411	0.3398	0.3464	**0.2982**
Bupa	0.4232	0.4116	0.4145	0.4290	0.4145	0.4029	0.4232	**0.3855**
Wine	0.2809	0.0225	0.2697	0.3820	0.3539	0.4157	0.0506	**0.0169**
Sonar	0.3462	0.4375	0.3558	0.4423	0.3894	0.4423	0.4615	**0.3269**
Glass	0.4533	0.4953	0.5327	0.4953	**0.4346**	0.5327	0.4953	0.4987
E-coli	0.1696	**0.1667**	0.4196	0.1964	0.2351	0.3899	0.1845	0.1747
Iris	0.0667	**0.0267**	0.0533	0.0867	0.0800	0.0733	0.3200	0.0400
Ion	0.0912	0.0769	0.2963	0.2934	**0.0684**	0.3191	0.0969	0.0969
BC	0.0315	0.0329	**0.0272**	0.0300	0.0300	0.0443	0.0286	0.0286

D. *Handwritten Data Sets*

The third type of dataset considered is the MNIST dataset. It is a handwritten digits database given by Lecun et al. [14]. It has a training set of 60,000 examples and a test set of 10,000 samples. For each of the ten digits, there is a test set of 1000 samples. All the samples are images of size 28x28. Since the proposed method is unsupervised, we have considered the images from test dataset of the MNIST dataset. For the experiments, 200 samples of each digit were considered. The proposed methods were tested on some of the challenging test cases such as {0, 8}, {3, 5, 8}, {1, 2, 3, 4}. Tables 9, 10, 11 show the comparison between SC-TNF2, NJW, NP, SNN, PG, LPCA and PRCUT methods. Tables 9, 10, 11 display the results of ARI, NMI and CE metrics on the test cases respectively. From the tables, we note that, for the case of {0, 8} dataset, almost all the methods perform well in clustering. However, in the case of {3, 5, 8} dataset, the SC-TNF2 performs considerably better than other methods. In the case of {1, 2, 3, 4} dataset, the accuracy values obtained by the SC-TNF2 are much higher than the other methods with respect to all the metrics. The following subsection discusses the robustness of the SC-TNF2 with respect to the σ parameter.

Table 9: Results of ARI metric on MNIST

				Methods				
Data sets	NJW	NP	SNN	PG	LPCA	PRCUT	SC-TNF1	SC-TNF2
{0,8}	**1.0000**	0.9601	0.9799	**1.0000**	0.9899	0.9799	**1.0000**	**1.0000**
{3,5,8}	0.6190	0.5664	0.5664	0.6498	0.5657	0.5621	0.5827	**0.7956**
{1,2,3,4}	0.3431	0.3349	0.3349	0.4220	0.3310	0.3512	0.5886	**0.5947**

Table 10: Results of NMI metric on MNIST

				Methods				
Data sets	NJW	NP	SNN	PG	LPCA	PRCUT	SC-TNF1	SC-TNF2
{0,8}	**1.0000**	0.9288	0.9594	**1.0000**	0.9772	0.9594	**1.0000**	**1.0000**
{3,5,8}	0.7502	0.7367	0.7367	0.7502	0.7502	0.7418	0.7502	**0.7811**
{1,2,3,4}	0.6282	0.5327	0.5234	0.5897	0.6325	0.6282	0.6454	**0.6635**

Table 11: Results of CE metric on MNIST

				Methods				
Data sets	NJW	NP	SNN	PG	LPCA	PRCUT	SC-TNF1	SC-TNF2
{0,8}	**0.0000**	0.0100	0.0050	**0.0000**	0.0025	0.0050	**0.0000**	**0.0000**
{3,5,8}	0.1633	0.3283	0.3283	0.1433	0.3367	0.3367	0.1767	**0.0747**
{1,2,3,4}	0.4313	0.4637	0.4637	0.3675	0.5012	0.4425	0.2737	**0.1825**

E. *Parameter Sensitivity*

In this section, the robustness of SC-TNF2 with respect to the change in σ parameter is evaluated. Three methods, NJW, NP and SC-TNF2, which have σ as a parameter, are compared in this section. The change in clustering accuracy is analyzed for the various methods, when the σ value is changed from .01 to 2 with a step size of 0.01. The metric for estimating the accuracy of the clustering is ARI. Sample plots of ARI versus sigma are shown in Fig. 6 to Fig. 11. From the Fig. 6 and 7, it is noted that SC-TNF2 is consistently better than other methods for the majority of sigma values. Fig. 8 shows that SC-TNF2 is better than other methods in the range of sigma values (.05-0.38). From the sensitivity plots for real datasets in Fig. 9, 10, 11 it can be observed that the proposed method gives better result than the other methods over the range of 0.01 to 0.70 but, falls in accuracy after a threshold. This reduction in accuracy could be attributed to the challenging nature of the real datasets. Additionally, Fig. 12 shows the average plot of ARI versus sigma with ARI values obtained from four synthetic datasets, namely Flame, Aggre, Compound, and Pathbased. From this plot, we can see that the TNF method is better and more robust than NP and NJW techniques. From the plots, it is evident that SC-TNF2 method is robust to change in sigma parameter.

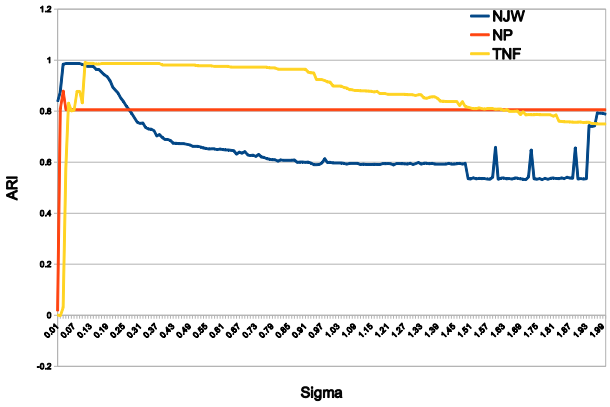


Fig. 6: ARI versus sigma plot for Aggre dataset

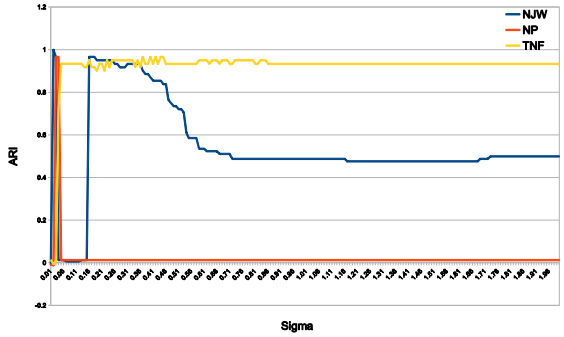


Fig. 7: ARI versus sigma plot for Flame dataset

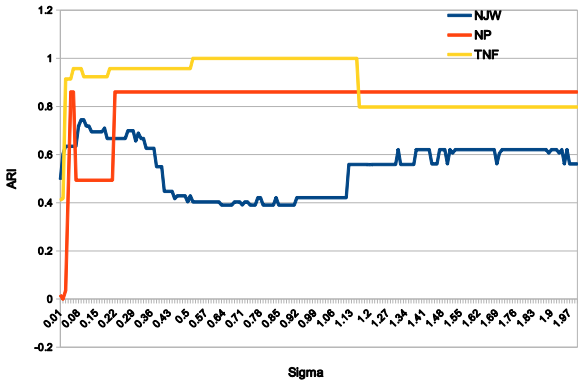


Fig. 8: ARI versus sigma plot for Compound dataset

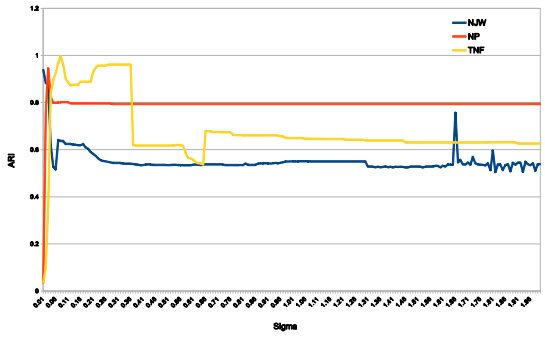


Fig. 9: ARI versus sigma plot for Flea dataset

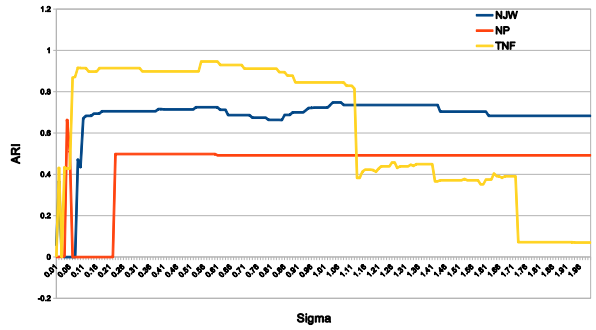


Fig. 10: ARI versus sigma plot for Seeds dataset

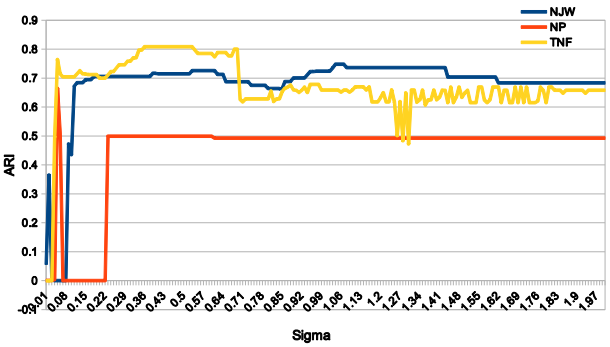


Fig. 11: ARI versus sigma plot for Wine dataset

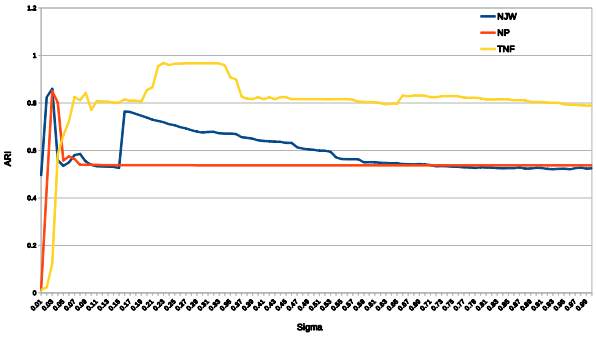


Fig. 12: ARI versus sigma plot on an average of four datasets

IV. CONCLUSION

In a spectral clustering algorithm, the pairwise similarity between data points plays a crucial role. In this work, a novel way to approach similarity measure for spectral clustering is proposed making use of Topological Node Features (TNF). As part of this work, two algorithms, SC-TNF1 and SC-TNF2, have been proposed. Characteristics such as local density and local structure were estimated using TNFs and were incorporated into the construction of pairwise affinity. Using topological graph properties, we were able to enhance or penalize the pairwise similarity. The experiments on synthetic, real and handwriting datasets show that proposed TNF based techniques improved the effectiveness of clustering. In our future work, we would like to adopt this metric for different applications such as image segmentation, motion segmentation, etc. The metric can also be strengthened by assimilating additional TNFs such as Listing Index and Tree Index [7].

APPENDIX

In this section, the various metrics which were used in this work, namely: ARI, NMI, CE, will be defined.

A. *Adjusted Rand Index (ARI)*

Hubert and Arabie [10] define ARI using the Contingency table. The Contingency table is constructed using the following steps:

Given a set D of p elements, and two groupings or partitions (e.g. clusterings) of these points, namely $I = \{I_1, I_2, \dots, I_r\}$ and $J = \{J_1, J_2, \dots, J_s\}$, the overlap between I and J can be summarized in a contingency table $[pt_{ij}]$ where each entry pt_{ij} denotes the number of objects in common between I_i and J_j: $pt_{ij} = |I_i \cap J_j|$

$I \backslash J$	J_1	J_2	...	J_s	*Sums*
I_1	pt_{11}	pt_{12}	...	pt_{1s}	t_1
I_2	pt_{21}	pt_{22}	...	pt_{2s}	t_2
⋮	⋮	⋮	⋱	⋮	⋮
I_r	pt_{r1}	pt_{r2}	...	pt_{rs}	t_r
Sums	l_1	l_2	...	l_s	

The adjusted form of the Rand Index, the Adjusted Rand Index, is

$$\text{AdjustedIndex} = \frac{\text{Index} - \text{ExpectedIndex}}{\text{MaxIndex} - \text{ExpectedIndex}} \quad (20)$$

, more specifically

$$ARI = \frac{\Sigma_{ij}\binom{pt_{ij}}{2} - [\Sigma_i \binom{t_i}{2} \Sigma_j \binom{l_j}{2}]/\binom{p}{2}}{\frac{1}{2}[\Sigma_i \binom{t_i}{2} + \Sigma_j \binom{l_j}{2}] - [\Sigma_i \binom{t_i}{2} \Sigma_j \binom{l_j}{2}]/\binom{p}{2}} \quad (21)$$

where pt_{ij}, t_i, l_j are values from the contingency table.

B. *Normalized Mutual Information (NMI)*

Normalized Mutual Information (NMI) is defined as in Strehl and Ghosh [24]:

$$NMI = \frac{\sum_{k=1}^{m}\sum_{l=1}^{q}(num_{k,l} \times log(\frac{n \times num_{k,l}}{num_k \times \hat{n}um_l}))}{\sqrt{(\sum_{k=1}^{m} num_k \times log(\frac{num_k}{n})) \times (\sum_{l=1}^{q} \hat{n}um_l \times log(\frac{\hat{n}um_l}{n}))}} \quad (22)$$

where n is the total number of points in the data, num_k denotes the number of datapoints contained in the cluster $C_k (1 \le k \le m)$, $\hat{n}_i$ is the number of data belonging to the l^{th} class $(1 \le l \le q)$ and $n_{i,j}$ denotes the number of data points that are in the intersection between C_i and the j^{th} class.

C. *Clustering error*

If $Conf$ is defined as the confusion matrix of two clusterings.

$$Conf(l_{true}, l) = |D_{l_{true}}^{true} \cap D_l| \quad (23)$$

Confusion ($Conf$) is the number of common points between clustering produced (l) and true clustering (l_{true}). According to Verma and Meila [25], clustering error (CE) is defined as:

$$CE(D, D^{true}) = \frac{\sum_{l_{true}} \sum_{l \neq l_{true}} Conf(l_{true}, l)}{n} \quad (24)$$

where n is the total number of points.

Due to the possibility of renumbering in the output clustering, cluster label 2 could be allocated cluster 1 by the algorithm and so on. Hence, CE is considered as a minimum value of all possible combinations of numberings.

This method has large number of computations. A maximum weighted bipartite matching problem based modelling is applied and its solution is found using linear programming, in order to reduce computation.

D. *Silhouette Index*

The definition of Silhouette index as given in Chen et al [5] and Kaufman and Rousseeuw (1987) [12] is given as follows: Silhouette index is a composite index reflecting the compactness and separation of the clusters, and can be applied to different distance metrics. For each point i, its silhouette index $si(i)$ is defined as

$$si(i) = \frac{b(i)-a(i)}{max\ (a(i),b(i))} \tag{25}$$

where $a(i)$ is the average distance of point i to all the other points in the same cluster, $b(i)$ is the average distance of point i to its nearest neighbours. The average of $s(i)$ across all points is calculated to reflect the overall quality of the clustering result. A larger averaged silhouette index indicates a better overall quality of the clustering result.

E. *Normalization*

Given the Data, the normalized data ($NData$) is calculated as follows:

Initialize the variables: a = 0, b = 1.

The variable $minData$ is assigned the minimum value of the data. The variable $maxData$ is assigned the maximum value of the data.

Define the following two values:

$$r = (a - b)/(minData - maxData) \tag{26}$$

$$s = (a - r) * minData. \tag{27}$$

Using the above variables, the normalized data $Ndata$ is obtained as:

$$NData = r * Data + s \tag{28}$$

ACKNOWLEDGMENT

We dedicate our work to Bhagawan Sri Sathya Sai Baba, the founder chancellor of Sri Sathya Sai Institute of Higher Learning.

We want to thank Prof. Sethuraman Panchanathan, Arizona State University, USA, Prof. N. B. Vineeth, IIT Hyderabad, India, Prof. V Chandrasekaran, SSSIHL, India, and Dr. Srinath M S, SSSIHL, India for their valuable advice.

REFERENCES

1. Arias-Castro, E., Lerman, G., Zhang, T.: Spectral clustering based on local pca. The Journal of Machine Learning Research 18(1), 253–309 (2017)
2. Bach, F.R., Jordan, M.I.: Learning spectral clustering, with application to speech separation. J MACH LEARN RES 7, 1963–2001 (2006)
3. Beauchemin, M.: A density-based similarity matrix construction for spectral clustering. Neurocomputing 151, 835–844 (2015)
4. Challa, A., Danda, S., Sagar, B.D., Najman, L.: Power spectral clustering. hal-01516649v3 pp. – (2018). URL https://hal.archives-ouvertes.fr/hal-01516649
5. Chen, G., Jaradat, S.A., Banerjee, N., Tanaka, T.S., Ko, M.S., Zhang, M.Q.: Evaluation and comparison of clustering algorithms in analyzing es cell gene expression data. Statistica Sinica pp. 241–262 (2002)
6. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: A (sub) graph isomorphism algorithm for matching large graphs. IEEE T PATTERN ANAL 26(10), 1367–1372 (2004)
7. Dahm, N., Bunke, H., Caelli, T., Gao, Y.: Efficient subgraph matching using topological node feature constraints. Pattern Recognit 48(2), 317–330 (2015)
8. Diao, C., Zhang, A.H., Wang, B.: Spectral clustering with local projection distance measurement. MATH PROBL ENG 2015 (2015)
9. Gu, R., Wang, J.: An improved spectral clustering algorithm based on neighbour adaptive scale. In: Business Intelligence and Financial Engineering, 2009. BIFE'09. International Conference on, pp. 233–236. IEEE (2009)
10. Hubert, L., Arabie, P.: Comparing partitions. Journal of classification 2(1), 193–218 (1985)
11. Jordan, F., Bach, F.: Learning spectral clustering. Adv. Neural Inf. Process. Syst 16, 305–312 (2004)
12. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis, vol. 344. John Wiley & Sons (2009)
13. Langone, R., Reynders, E., Mehrkanoon, S., Suykens, J.A.: Automated structural health monitoring based on adaptive kernel spectral clustering. Mechanical Systems and Signal Processing; 90(Supplement C), 64 – 78 (2017). DOI https://doi.org/10.1016/j.ymssp.2016.12.002 . URL http://www.sciencedirect.com/science/article/pii/S0888327016305131
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998). URL http://yann.lecun.com/exdb/mnist/
15. Lichman, M.: UCI machine learning repository (2013). URL http://archive.ics.uci.edu/ml
16. Li, X.Y., Guo, L.J.: Constructing affinity matrix in spectral clustering based on neighbor propagation. Neurocomputing 97, 125–130 (2012)
17. Mijangos, V., Sierra, G., Montes, A.: Sentence level matrix representation for document spectral clustering. PATTERN RECOGN LETT; 85(Supplement C), 29 – 34 (2017). DOI https://doi.org/10.1016/j.patrec.2016.11.008 . URL http://www.sciencedirect.com/science/article/pii/S0167865516303312
18. Nataliani, Y., Yang, M.S.: Powered gaussian kernel spectral clustering. Neural Comput Appl pp. 1–16 (2017)
19. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. ADV NEUR IN 2, 849–856 (2002)
20. Rand, W.M.: Objective criteria for the evaluation of clustering methods. J AM STAT ASSOC 66(336), 846–850 (1971)
21. Shen, G., Ye, D.: A distance-based spectral clustering approach with applications to network community detection. Journal of Industrial Information Integration; 6(Supplement C), 22 – 32 (2017)
22. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE T PATTERN ANAL 22(8), 888–905 (2000)
23. Sorlin, S., Solnon, C.: A parametric filtering algorithm for the graph isomorphism problem. Constraints 13(4), 518–537 (2008)
24. Strehl, A., Ghosh, J.: Cluster ensembles: a knowledge reuse framework for combining multiple partitions. J MACH LEARN RES 3, 583–617 (2003)
25. Verma, D., Meila, M.: A comparison of spectral clustering algorithms. University of Washington Tech Rep UWCSE030501 1, 1–18 (2003)
26. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing 17(4), 395–416 (2007)
27. Wang, W., Xu, Z., Lu, W., Zhang, X.: Determination of the spread parameter in the gaussian kernel for classification and regression. Neurocomputing 55(3), 643 – 663 (2003
28. Wong, M.A.: Asymptotic properties of k-means clustering algorithm as a density estimation procedure (1980)
29. Yang, M.S., Wu, K.L.: A similarity-based robust clustering method. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(4), 434–448 (2004)
30. Yang, P., Zhu, Q., Huang, B.: Spectral clustering with density sensitive similarity function. KNOWL-BASED SYST 24(5), 621–628 (2011)
31. Ye, X., Sakurai, T.: Robust similarity measure for spectral clustering based on shared neighbours. ETRI Journal 38(3), 540–550 (2016)
32. Zelnik Manor, L., Perona, P.: Self tuning spectral clustering. In: ADV NEUR IN, pp. 1601–1608 (2004)
33. Zhang, X., Li, J., Yu, H.: Local density adaptive similarity measurement for spectral clustering. PATTERN RECOGN LETT 32(2), 352–358 (2011)

AUTHORS PROFILE

Lalith Srikanth Chintalapati, completed his M.Sc. (Mathematics and Computer Science) and M.Tech (Computer Science) from Sri Sathya Sai Institute of Higher Learning (SSSIHL). He is currently pursuing PhD in the field of Spectral Clustering in Department of Mathematics and Computer Science, SSSIHL.

His research interest involves proposing novel pairwise affinity metrics using local neighborhoods properties of the data. These methods can be used to solve problems in which data set can be modeled as graph. He attended and presented some of his works in international conferences. Areas of Interest: Spectral Clustering, Image Processing, Machine Learning, Deep Learning, Computer Vision.

Rachakonda Raghunatha Sarma is Associate Professor in Department of Mathematics and Computer Science, Sri Sathya Sai Institute of Higher Learning (SSSIHL). He pursued his Ph.D. from SSSIHL in the field of corner detection. His areas of interest are Image Processing, Machine Learning, Deep Learning and Networks. He was principal investigator in many national level research projects from different recognized agencies. He has many publications in reputed conferences and journals to his credit. He was convener of International Workshop on Computer Vision and Machine Learning (IWCVML-2014) conducted at SSSIHL. Presently he is serving as Associate Department Head, Prashanthi Nilayam Campus, Department of Mathematics and Computer Science, SSSIHL.