



Cross-covariance based affinity for graphs

Rakesh Kumar Yadav¹ · Abhishek¹ · Shekhar Verma¹ · S Venkatesan¹

Accepted: 28 September 2020 / Published online: 20 November 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

The accuracy of graph based learning techniques relies on the underlying topological structure and affinity between data points, which are assumed to lie on a smooth Riemannian manifold. However, the assumption of local linearity in a neighborhood does not always hold true. Hence, the Euclidean distance based affinity that determines the graph edges may fail to represent the true connectivity strength between data points. Moreover, the affinity between data points is influenced by the distribution of the data around them and must be considered in the affinity measure. In this paper, we propose two techniques, $CCGA_L$ and $CCGA_N$ that use cross-covariance based graph affinity (CCGA) to represent the relation between data points in a local region. $CCGA_L$ also explores the additional connectivity between data points which share a common local neighborhood. $CCGA_N$ considers the influence of respective neighborhoods of the two immediately connected data points, which further enhance the affinity measure. Experimental results of manifold learning on synthetic datasets show that $CCGA$ is able to represent the affinity measure between data points more accurately. This results in better low dimensional representation. Manifold regularization experiments on standard image dataset further indicate that the proposed $CCGA$ based affinity is able to accurately identify and include the influence of the data points and its common neighborhood that increase the classification accuracy. The proposed method outperforms the existing state-of-the-art manifold regularization methods by a significant margin.

Keywords Graph · Affinity · Euclidean distance · Cross-Covariance · Neighborhoods · Manifold regularization

1 Introduction

Manifold learning and regularization methods have been widely used for data representation and processing respectively. A given data is represented as a weighted graph where the weights represent the similarity between data points. In an ideal case, similar data points should have

a higher affinity, which satisfies the manifold assumption. In manifold learning, clusters of data points having larger affinity are kept spatially close when projected to lower dimensional space. Similarly, in manifold regularization, the labels of data points in a large affinity neighborhood are assumed to be similar and, hence, the approximated function is penalized if it changes label in such a neighborhood. This shows that affinity is fundamental to both manifold learning and regularization methods, which also influence the accuracy of the underlying model.

Any manifold learning or regularization technique can be decomposed into three steps. (1) computation of the pairwise affinity, (2) creation of the graph, and (3) project the data into low dimensional space or approximate the generative function and penalize. When only data points are given, then an appropriate metric needs to be evolved to capture the similarity between data points. This is crucial as the efficiency of a learning technique hinges on the affinity metric. The affinity metric must be able to capture the dependence between data points and account for uneven sampling, if any, of the data points. Once the affinity is determined, geodesic

✉ Rakesh Kumar Yadav
pcl2014003@iiita.ac.in

Abhishek
rsi2016006@iiita.ac.in

Shekhar Verma
sverma@iiita.ac.in

S Venkatesan
venkat@iiita.ac.in

¹ Department of IT Deoghat, Indian Institute of Information Technology Allahabad, Jhalwa, Prayagraj, U.P. India, India

Table 1 Mathematical notations with their descriptions

Notation	Description
X	Dataset
D	Ambient Dimension
\mathbb{R}	Space
\mathcal{M}	Manifold
n	Total number of data instances
G	Graph
x_i	Data instance
W	Weight matrix
w_{ij}	Affinity between data point x_i and x_j
$A_m \cdot$	Row sum
$A \cdot m$	Column sum
$N(x_i)$	Neighborhood of x_i
$\kappa(x_i, x_j)$	Similarity between x_i and x_j
Q, Z	Random variables
\mathcal{F} and \mathcal{G}	Reproducing Kernel Hebert spaces
$HSIC(\cdot, \cdot)$	Hilbert-Schmidt Independence Criterion
$d(x_i, x_j)$	Distance between x_i and x_j
$d^{N(x_i)}(\cdot, \cdot)$	Distance between points in a neighborhood
$\kappa^{N(x_i)}(\cdot, \cdot)$	kernel defined in a neighborhood

using the heat kernel may be computed between data points assumed to be lying on the Riemannian manifold. The heat kernel enforces the smoothness assumption, which depends on the kernel bandwidth (σ).

The heat kernel affinity is governed by its bandwidth or scaling, which drastically reduces the affinity value for data points assumed to be distant or non-similar. The local scaling method further optimizes this bandwidth to fit the local neighborhoods of connected data points [1]. It often takes the distance between the point of interest and its 7th neighbor as affinity bandwidth, which suggests an increase in accuracy as compared to baseline methods. Instead of finding affinity from the given data points, [2] proposes to recreate optimal affinity assuming the graph partition has been provided. The removal of noisy edges that artificially increased or decreased the affinity between data points based on the maximum cliques further enhanced the manifold's modeling[3]. A slightly different approach introduced in the paper [4] that combined the similar information from various neighborhoods to enhance the robustness of k NN. An adaptive neighborhood assigning technique [5] proposed to learn the graph adjacency. It can extensively explore the local manifold structure but fails to decode the global structure of the manifold. A different approach introduced by [6] that used the information theoretic approach to decode the similarity information. These methods directly rely on the Euclidean distance to

construct a data graph that assumes the knowledge of the extent of the locally linear region and remains susceptible to noisy and highly correlated features.

In this work, we propose cross-covariance based graph affinity (**CCGA**) for the affinity approximation task. CCGA aims to preserve both linear and non-linear properties of the manifold accurately. The method initiates by mapping the data points into reproducing kernel Hilbert space (RKHS) using an appropriate kernel on the vertices of the underlying graph, which is followed by similarity measure between data points identified as neighbors in ambient space using Hilbert-Schmidt independence criterion (HSIC). The cross-covariance based affinity metric minimizes the effect of noise and error measurement by making it independent of its existing spatial locations. **CCGA** for the local region (CCGA_L) method aims to strengthen the affinity by discovering the connectivity between data points, which may or may not be immediately connected but share one or more common local neighborhoods that indicate an innate similarity. Additionally, CCGA for the neighborhood (CCGA_N) weighs the influence of respective neighbors of two connected data points for a correct measure of similarity.

The major contributions of the paper can be stated as, we propose Cross-Covariance based Graph Affinity between two data points that incorporates nonlinear dependence, the influence of neighboring data points, and the neighborhoods around data points. The first technique CCGA_L, try to include the influence of the data points in the common neighborhood in the affinity metric between two points. The second technique CCGA_N, takes into consideration the influence of the neighborhoods of data points that are adjacent to both the points. Table 1 describes the important mathematical symbols used in this paper.

The rest of the paper is organized as follows: Section 2 describes the existing literature that highlights state-of-the-art techniques and various affinities used in machine learning techniques. Section 3 illustrates the proposed technique and describes how measuring the dependence between the data points leads to a more accurate affinity. Sections 3.2 and 3.3 explains the affinity obtained based similarity estimation between two data in a local region and between two neighborhoods, respectively. Section 4 highlights the experiments performed on various real world and synthetic data to evaluate the performance of our techniques in terms of classification accuracy and dimensionality reduction. Section 5 concludes the findings and robustness of our techniques.

2 Related work

It is known that the affinity between data points depends not only on the location but also on the neighborhoods of

two points. The density of data has been considered by various metrics in supervised [7], unsupervised [8], semi-supervised [9] and deep learning [10]. The affinity between two data points with different neighborhood density is different from the affinity when both neighborhoods have the same density [11]. Many of the existing literature shows that they adjust local data structures to improve the characteristics of the affinity graph. However, these techniques are highly susceptible to the occurrence of noise and irrelevant features. They mainly focus on the parameter σ of the RBF heat kernel to determine the similarity between two data points. To solve such issues, the graph-theoretic approach is applied by [3] that consider rebuilding the tight neighborhood by picking maximal clique, which in other words, maximizes the average pairwise similarity. The use of contextual information is proposed in [12] to obtain affinity scores between the pairs of data points. The similarity information propagates as diffusion on the graph. The SSNMM-isomap framework [13] employs LLE to preserve the optimal local properties that minimizes the pairwise distances between intraclass data points that assumed to lie in the same manifold. The inter-class distances between data points that lie over different manifolds are simultaneously maximized. The LLE based neighborhood reconstruction error for preserving local topological structures leads to accurate low dimensional representation and, hence increases the model's accuracy. The work in the paper [14] carried out the diffusion on a locally constrained sparse graph to obtain a more accurate graph affinity. A similar kind of work [15] performed the diffusion process on a tensor product graph that captures the higher order information. However, in the aforementioned works, the diffusion process is highly susceptible to noise. This leads to the inaccurate affinity propagation in a graph.

DSCF-net [16] aims to learn an optimal representation matrix along with several sets of basis vectors. The deep factorized coefficient matrix accumulates the similarity information between samples and features. A technique [17] for 2D image is introduced to extract the neighborhood features by minimizing the Frobenius norm-based reconstruction error. This makes the distance metric more reliable and encodes the neighborhood reconstruction error precisely. The RS2ACF [18] framework optimizes the representation by constraining it in both projection and label space. The basis vector obtained from concept factorization is used to discover the latent semantic structure hidden in data proves to be useful for clustering. The label constrained representation matrix preserves the local geometrical information, which propagates accurate labels to unlabeled data. In [19], a parameter free affinity based clustering is introduced that estimate distance between data points, and a

threshold affinity is obtained. In paper [20], propose Consistent Affinity Graph Learning (CAGL) algorithm for multi-view data that selects the robust graph Laplacian matrices from each view. It models the hypergraph Laplacians as a point on Grassmann manifold and fuses all the views with CAGL algorithm. In [21], Axiomatic Fuzzy Set-based Spectral Clustering (AFSSC) is proposed that creates a highly robust affinity graph by exploiting and identifying the discriminative features.

Kernel dependency measures like HSIC capture both linear and nonlinear relationships by mapping the data into a high dimensional feature space. A kernel defined on the vertices of a graph gives a representation of the data and measure of similarity between data points in the RKHS. HSIC [22, 23] is calculated from the empirical estimate of the Hilbert-Schmidt norm on the cross-covariance operator. Empirical HSIC is used to measure statistical dependence between two random variables or two datasets assumed to be drawn from different probability density functions. HSIC has been optimized with greedy algorithms on data features [23, 24]. In the paper [25], the proposed feature space independent semi-supervised kernel matching method for domain adaptation. HSIC Lasso using dual augmented Lagrangian for global optimum, has been introduced in [26] for regression with feature reduction. ICA [27], with contrast function based on canonical correlation in the RKHS space, has been proposed for feature selection by finding the minimum correlated dimensions. The HSIC based supervised feature selection [24] measures the dependence between the given features with their respective labels and selects those features that give the maximum correlation. HSIC has also been used for dimensionality reduction [28, 29] such that maximum independent features are identified using the kernel method. The problem of hyperspectral image classification [30] has also utilized surrogate kernel based HSIC to further increase the classification accuracy.

3 Cross-covariance based graph affinity (CCGA)

CCGA based affinity measure has been proposed to capture the nonlinear dependence between data points of a point cloud lying on a Riemannian manifold in place of Euclidean distance. It is recognized that in addition to nonlinear dependence, pairwise affinities are also affected by the presence of common neighbors of data points. The first technique, CCGA_L, considers the dependence between neighboring points of a neighborhood to identify innate dependence and to incorporate them in the affinity measure between data points. The second technique,

CCGA_N assumes that common neighbors do not affect the affinity measure alone but with their own neighborhoods. This effect has been captured in the affinity measure for increased classification accuracy and dimensionality reduction performance.

A pairwise affinity metric needs to capture the mutual dependence (linear or nonlinear) between data points. Semi-supervised learning assumes that the conditional density of the points given the labels varies smoothly along the geodesics in the intrinsic geometry of the marginal distribution of the data points. If two points on the manifold have an edge i.e. high mutual dependence between them on which the marginal density is large, then the two points would generally share the same label. This entails that the affinity metric between two data points must also consider their neighborhoods. Nonlinear dependence can be computed using the kernel in an appropriate feature space. However, to consider the effect of neighborhood requires determination of the neighborhoods of the nodes. This becomes paradoxical as the affinity between data points define edges. Moreover, implicit mapping into the feature space does not preserve neighborhoods. This can be solved by defining local neighborhoods created by k NN or ϵ NN using Euclidean distance. This neighborhood is used to determine neighboring points and the effect of neighborhoods on affinity using kernel based dependency measure in the feature space. The resulting affinity is used to find the geodesic on the Riemannian manifold using a heat kernel.

Given n data points $(x_i)_{i=1}^n$ sampled on manifold \mathcal{M} , the local neighborhoods are designed using k -NN or ϵ -NN represented as $G = (X, W)$ where, X is collection of all data points and W contains the affinity between connected data points. Assume that around each data point x_i , the local neighbors are contained in $x_j \in N(x_i)$. The heat kernel based affinity between data points are obtained using:

$$w_{ij} = \exp - \frac{\|x_i - x_j\|_2^2}{2\sigma^2} \quad (1)$$

where, x_i is the point of interest, $x_j \in N(x_i)$ is its neighbor, and σ is the heat kernel bandwidth. In graph-based semi-supervised learning methods, affinity measures between labeled and unlabeled data points are used to determine the labels of unlabeled points. The choice of affinity measure is vital in determining the classes of the unlabeled points. A graph is created using k NN such that number of fixed neighbors in k NN may force data points beyond the linear region to be considered as a neighbor. In other situation, genuine linear neighborhood data points may be discarded. Thus, the neighborhood of a data point cannot be guaranteed to hold the local linearity assumption. This requires a non Euclidean measure that can capture the relation between x_i and $x_j \in N(x_i)$.

Kernel methods map the input data into a high-dimensional RKHS and define a linear method therein. The model results in a nonlinear relationship with respect to the input space. The mapping is implicit, and the nonlinear relations between data in the feature space are captured through the kernel function. A symmetric positive semi-definite matrix can be viewed as a proximity measure that has an inner product representation and satisfies the criteria required to be a distance metric. Consider for any proximity κ on a finite set X , if two data points x_i and x_j are connected then the distance between them is obtained using $d(x_i, x_j) = \frac{1}{2} (\kappa(x_i, x_i) + \kappa(x_j, x_j) - \kappa(x_i, x_j))$. This also indicates that the proximity measures can be combined to generate a distance metric that represents the pairwise affinity or dependence between data points.

In this paper, we follow the proximity approach to encode the similarity among data points through cross covariance based affinity. The method captures both linear and nonlinear properties of the underlying data manifold by means of statistical independence criterion. The cross-covariance operator assigns a value between 0 and 1 where, the former value is assigned for two completely different data points and the later value signifies that the two data points are exactly the same. HSIC of a data point to itself always results in a unity value. As the properties of underlying manifold \mathcal{M} is unknown and the sample data points cannot be guaranteed to have been drawn uniformly. Hence, the distribution parameters of resultant neighborhoods differ from each other. The similarity and differences of neighborhood distribution parameters can only be appropriately utilized when the data is mapped to its full feature space. However, due to the complexity of the mapping and increased dimensions, the kernel proximity automatically identifies the small linear subspace where the manifold properties can be exploited in a linear manner.

The proposed affinity framework relies on defining the local neighborhood on k NN method based on Euclidean distance. This is necessary as the neighborhood is not preserved in the feature space. The implicit kernel mapping redistributes the points. Hence, a Euclidean distance based neighborhood is fixed ab initio. Then the affinities between connected data points are determined through CCGA methods. The proposed techniques account for both linear as well as nonlinear relationships between each pair of connected data points by estimating the HSIC dependence.

3.1 Hilbert-schmidt independence criterion (HSIC)

HSIC is a kernel based similarity measure between two random variables. Given, Q and Z are two random variables sampled from their respective probability density functions. Assume, \mathcal{F} and \mathcal{G} are the two RKHSs for Q and Z

respectively. Let the $\phi(q)$ and $\psi(z)$ be the feature mapping functions such that $\phi(q) : Q \rightarrow \mathbb{R}$ and $\psi(z) : Z \rightarrow \mathbb{R}$. The kernel functions are given by $\kappa(q, q') = \langle \phi(q), \phi(q') \rangle_{\mathcal{F}}$ and $\hat{\kappa}(z, z') = \langle \psi(z), \psi(z') \rangle_{\mathcal{G}}$. The empirical estimation of HSIC using a finite number of data samples is defined as:

$$HSIC = \frac{1}{n} tr(\kappa C \hat{\kappa} C) \tag{2}$$

where, n is the total number of observation, κ , and $\hat{\kappa}$ are the two respective kernel matrix for Q , and Z respectively, and C is the centering matrix. The similarity value between two data vectors in the Hilbert space is given by (2).

3.2 CCGA for local region (CCGA_L)

Assume a data point x_i and its neighbors $x_j, \dots, x_k \in N(x_i)$. Based on smoothness assumption, if $N(x_i)$ is a local region with high affinity between neighborhood data points, then all the neighbors shall share similar label with high probability. This entails that two data points $x_j \in N(x_i)$ and $x_k \in N(x_i)$ that are part of the neighborhood $N(x_i)$ of x_i but not adjacent to each other i.e. $x_j \notin N(x_k)$ and $x_k \in N(x_j)$ may also generally share the same label. It also indicates that the affinity between such data points x_j and x_k should be large if they have multiple common neighborhood. However, in general for affinity calculation, the relation between such x_j and x_k is not considered unless either $x_j \in N(x_k)$ or $x_k \in N(x_j)$ is true. The proposed CCGA_L tries to identify such similarities among data points and accumulate them to define revamp affinity between two data points based on two factors, first, whether

they are neighbors and second, number of common local neighborhood shared by them.

Figure 1 shows the pictorial representation of CCGA_L, x_i as point of interest with data points in the k NN neighborhood $x_j \in N(x_i)$. Here, x_i is denoted by black filled circle and its neighbors with blue filled circles. The direct connectivity between x_i and $x_j \in N(x_i)$ has been illustrated using solid lines which is approximated similarly by both Euclidean distance and CCGA_L. This CCGA_L determines the statistical independence between the neighbors. This is represented by dashed line. If the data points x_j and x_k further share common local neighborhood apart from x_i , the next such distance should be added to previous value between them. The proximity between any two connected data points x_i and x_j can be defined as

$$\begin{aligned} d(x_i, x_j) &= \frac{1}{2} (\kappa(x_i, x_i) + \kappa(x_j, x_j) - 2\kappa(x_i, x_j)) \\ &= \frac{1}{2} (2 - 2\kappa(x_i, x_j)) \end{aligned}$$

Since, κ of a data point to itself is 1.

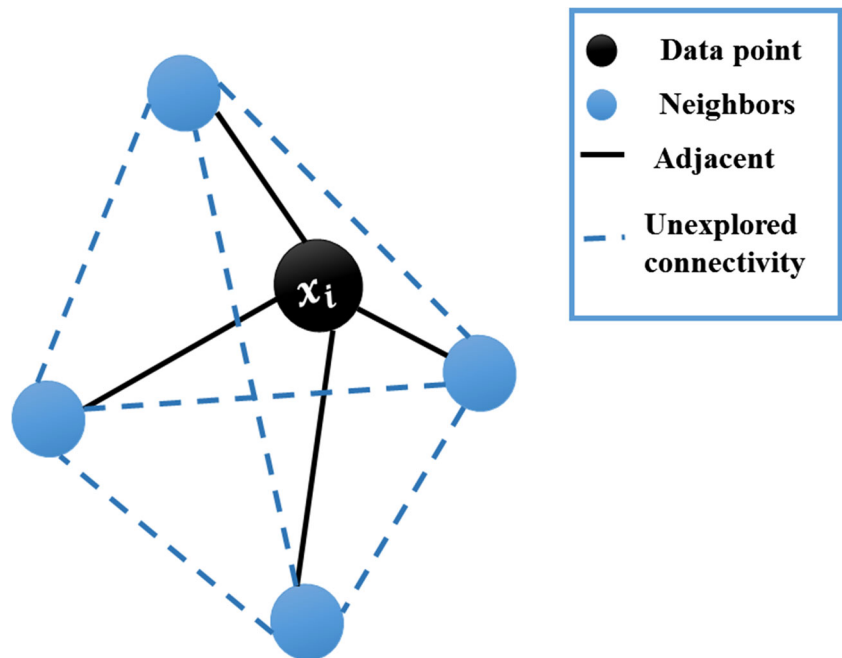
$$\therefore d(x_i, x_j) = 1 - \kappa(x_i, x_j) \tag{3}$$

Similarly, when two data points x_j , and x_k lie in the neighborhood $N(x_i)$, then their distance is calculated using

$$d^{(N(x_i))}(x_j, x_k) = 1 - \kappa^{(N(x_i))}(x_j, x_k)$$

where, $\kappa^{(N(x_i))}(\cdot, \cdot)$ signifies kernel proximity applied the neighborhood of $N(x_i)$, and $w^{(N(x_i))}$ represents the respective neighborhood distance. Further, if x_j , and x_k are immediately connected or are part neighborhoods of other data point ($N(\cdot)$), then the distance between them is

Fig. 1 CCGA for Local Region



obtained by averaging over all individual distances using [31]

$$d(x_j, x_k) = \frac{1}{n} \sum_{i=1}^n \left(1 - \kappa(x_j, x_k) + \kappa^{N(i)}(x_j, x_k) + \dots \right) \quad (4)$$

The final affinity w_{jk} between any two data points x_j and x_k where, $w(x_j, x_k) \neq 0$ can be obtained using the heat kernel as

$$w_{jk} = \exp - \frac{d(x_j, x_k)}{2\sigma^2} \quad (5)$$

The algorithm for CCGA for local region has been explained in Algo. 1.

The CCGA_L major steps are as follows,

1. Given a dataset X having n number of points, create a k NN neighborhood $N(x_i)$ around each data point x_i
2. Apply kernel $\kappa^{N(x_i)}$ and compute the $k \times k$ distance matrix.
3. Convert this distance matrix to an affinity matrix as in (5).
4. For each pair of data points x_j and x_k contained in the current $k \times k$ affinity matrix, add the current affinity values to the previously computed values.
5. Repeat the step 2 until all n data points are processed.

Algorithm 1 CCGA for Local Region

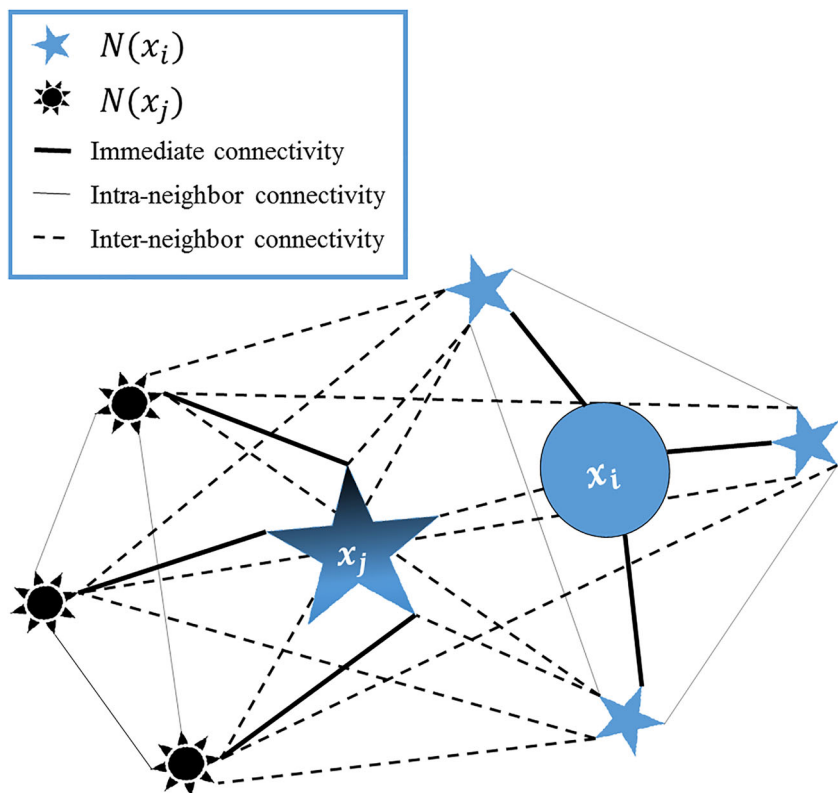
```

Input:  $X_{n \times D}$  and Number of neighbors  $k$ 
Output: Affinity matrix  $W$ 
1  $n = \text{size}(X, 1)$ ; // Number of data points
2  $W = \text{zeros}(n, n)$ ; // Initialize the Affinity matrix
3  $Idx = \text{createGraph}(X, k)$ 
4 for  $i = 1$  to  $n$  do
5    $Ii = [i, Idx(i, :)]$ 
6    $kt = |Ii|$ 
7    $Xi = X(Ii, :) - \text{repmat}(X(i, :), kt)$ 
8    $I = \text{identity}(kt)$ ; // Identity matrix
9    $C = I - \frac{1}{kt} * \mathbb{1}$ ; // Centering Matrix
10   $K = \kappa^{N(x_i)}(Xi)$ 
11   $d(Ii, Ii) = d(Ii, Ii) + (1 - C * K * C)$  (3)
12 end
13  $W(Ii, Ii) = \exp - \frac{d(Ii, Ii)}{2\sigma^2}$  (5)
14 Return  $W$ 
    
```

3.3 CCGA for neighborhood (CCGA_N)

CCGA_L is based on the assumption that common neighbors influence the mutual dependence between two data points.

Fig. 2 CCGA for Neighborhood



These individual influences may be further conditioned by their own neighborhoods. CCGA_N tries to assimilate this intangible effect of the neighborhoods of the neighbors themselves in the affinity metric. Again, to adhere to the manifold assumption, a neighborhood is identified using spatial proximity based on Euclidean distance. CCGA_N proceeds as follows, given data points in X , a local neighborhood $N(x_i)$ can be constructed using k NN or ϵ NN around every point x_i based on Euclidean distance between all data points. Once the graph is created, we want to compute the pairwise affinity, based on their nonlinear dependence, the effect of neighboring data points, and their respective neighborhoods between the point of interest x_i and each of its neighbor $x_j \in N(x_i)$. As shown in Fig. 2, the pairwise affinity between every such pair of x_i and x_j would further decide if they are likely to share similar labels. We have two neighborhoods $N(x_i)$ around the x_i and other $N(x_j)$ around the x_j such that $x_j \in N(x_i)$. To compute the nonlinear dependence, we apply RBF kernels: $\kappa^{N(x_i)}$ and $\kappa^{N(x_j)}$ for $N(x_i)$ and $N(x_j)$ respectively. Each k NN neighborhood contains $k + 1$ elements including point of interest. The kernels over $N(x_i)$ and $N(x_j)$ allows the algorithm to appropriately weigh the importance of neighbors. The HSIC based statistical independence between $N(x_i)$ and $N(x_j)$ is

$$w(N(x_i), N(x_j)) = \kappa^{N(x_i)} C \kappa^{N(x_j)} C \tag{6}$$

$w(N(x_i), N(x_j))$, is converted into distance as follows,

$$d(N(x_i), N(x_j)) = 1 - \left(\kappa^{N(x_i)} C \kappa^{N(x_j)} C \right) \tag{7}$$

The resultant $d(N(x_i), N(x_j))$ distance matrix consists of $(k + 1) \times (k + 1)$ where each $d_{i'j'}$ corresponds to distance between i^{th} member of $N(x_i)$, and j^{th} member of $N(x_j)$:

$$d(N(x_i), N(x_j)) = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1(k+1)} \\ d_{21} & \dots & \dots & d_{2(k+1)} \\ \vdots & \dots & \dots & \dots \\ \vdots & \dots & \dots & d_{(k+1)(k+1)} \end{bmatrix}$$

The matrix $d(N(x_i), N(x_j))$ now contains the distance calculated using HSIC between every pair of $N(x_i)$, and $N(x_j)$ represented as rows, and columns respectively as shown above. Finally, the distances between data points being member of such $N(x_i)$, and $N(x_j)$ are summed up together to strengthen such relations. $w(x_i, x_j) = w(N(x_i), N(x_j)) + w(N(x_i), N(x_k))_{x_j \in N(x_k)} + w(N(x_j), N(x_k))_{x_i \in N(x_k)} + w(N(x_k), N(x_l))_{x_i \in N(x_k), x_j \in N(x_l)} + \dots$

This distance is calculated between data point and each of its neighbors. Inside every neighborhood, $w(N(x_i), N(x_j))$ is computed k times. Over n data points, this algorithm is repeated $k \times n$ times to compute the distance matrix. The final affinity w_{ij} between any two data points can be

obtained using (5). The algorithm for CCGA for local region has been explained in Algorithm 2.

Algorithm 2 CCGA for Neighborhood

```

Input:  $X_{n \times D}$  and Number of neighbors  $k$ 
Output: Affinity matrix  $W$ 
1  $n = \text{size}(X, 1)$ ; // Number of data points
2  $W = \text{zeros}(n, n)$ ; // Initialize the Affinity matrix
3  $Idx = \text{createGraph}(X, k)$ 
4 for  $i = 1$  to  $n$  do
5      $Ii = [i, Idx(i, :)]$ 
6      $kt = |Ii|$ 
7      $Xi = X(Ii, :) - \text{repmat}(X(i, :), kt)$ 
8      $Ij = Idx(i, :)$ 
9     for  $i = 1$  to  $n$  do
10         $Ik = [Ij(j), Idx(Ij(j), :)]$ 
11         $kt = |Ik|$ 
12         $Xj = X(Ik, :) - \text{repmat}(X(Ij(j), :), kt)$ 
13         $I = \text{identity}(kt)$ ; // Identity matrix
14         $C = I - \frac{1}{kt} * \mathbb{1}$ ; // Centering matrix
15         $K = \kappa^{N(x_i)}(Xi)$ 
16         $H = \kappa^{N(x_j)}(Xj)$ 
17         $d(Ii, Ik) = d(Ii, Ik) + (1 - K * C * H * C)$ ;
           // (3)
18    end
19 end
20  $W(Ii, Ik) = \exp\left(-\frac{d(Ii, Ii)}{2\sigma^2}\right)$ ; // (5)
21 Return  $W$ 

```

The CCGA_N can be summarized in following steps.

1. Given data points in X , local neighborhood $N(x_i)$ is obtained with k NN algorithm around each data point.
2. After obtaining this graph, based on the Euclidean the distance, affinity between point of interest x_i and each of its neighbor $x_j \in N(x_i)$ is computed. In this way, we have two neighborhoods, $N(x_i)$ around x_i and $N(x_j)$ around x_j .
3. On each of this neighborhood, the RBF kernel $\kappa^{N(x_i)}$ and $\kappa^{N(x_j)}$ is applied.
4. The HSIC based statistical independence between two these neighborhoods is computed as (6).
5. The pairwise HSIC values between data points is converted into distance (7). This forms a distance matrix of size $(k + 1) \times (k + 1)$.
6. In every neighborhood $w(N(x_i), N(x_j))$ is computed k times which is repeated for n data points ($k \times n$) times.
7. The final affinity matrix is obtained by (3).

3.4 Complexity analysis

The techniques based on CCGA largely depends on the number of data points n and ambient dimension D . For

both, $CCGA_L$ and $CCGA_N$ RBF kernel ($\mathcal{O}(n)$) and k NN algorithm ($\mathcal{O}(D \times n \times k)$) has been used. The graph mainly depends on the k number of nearest neighbors. The time complexity for both techniques can be expressed as,

- $CCGA_L$, complexity is computed based on steps as defined in Algorithm 1.

1. Step 5, copying the data points $\mathcal{O}(1)$.
2. Step 6, number of element count $\mathcal{O}(k)$.
3. Step 7, matrix subtraction $\mathcal{O}(k \times d)$.
4. Step 8, identity matrix $\mathcal{O}(1)$.
5. Step 9, centering matrix $\mathcal{O}(k \times d)$.
6. Step 10, RBF kernel computation over k data points $\mathcal{O}(k)$.
7. Step 11, affinity approximation over k data point $\mathcal{O}(k \times k)$.

The over all time complexity for n data points using $CCGA_L$ technique is given as $\mathcal{O}(n \times k)(d + k)$.

- $CCGA_N$, complexity is computed based on steps as defined in Algorithm 2.

1. Step 5, copying the data points $\mathcal{O}(1)$.
2. Step 6 & 11, number of element count $\mathcal{O}(k)$.
3. Step 7 & 12, matrix subtraction $\mathcal{O}(k \times d)$.
4. Step 13, identity matrix $\mathcal{O}(1)$.
5. Step 14, centering matrix $\mathcal{O}(k \times d)$.
6. Step 15 & 16, RBF kernel computation over k data points $\mathcal{O}(k)$.
7. Step 17, distance estimation $\mathcal{O}(k)$.
8. Step 20, conversion into affinity $\mathcal{O}(k \times k)$.

The over all time complexity for n data points using $CCGA_N$ technique is given as $\mathcal{O}(k^3 \times n \times d)(d + k)$.

4 Experiment and Results

The performance of the proposed graph affinity based techniques $CCGA_L$ and $CCGA_N$ have been evaluated on both synthetic and real world datasets for manifold learning and graph Laplacian regularized classification, i.e. manifold regularization. The existing state-of-the-art graph affinity methods like heat, binary EMR [32] (both 24 and 72 graphs) used for graph construction are compared against our techniques in terms of classification error. In order to validate the robustness of our techniques, a wide range of real world datasets are used. The real world datasets mainly comprise of images, handwritten characters, scene recognition and brain computer interface. The binary classification error is estimated with two models LapSVM and LapRLSC for both test and unlabeled sets. The mean

classification error¹, along with standard deviation by varying the value of k are summarized in the tables.

4.1 Synthetic datasets

The synthetic datasets contain 3D manifold shapes whose low dimensional 2D representation is known. Non-linear dimensionality reduction based 2D projections using both $CCGA_L$ and $CCGA_N$ techniques are obtained. For comparison many existing state-of-the-art non-linear dimensionality reduction techniques namely LTSA (Local Tangent Space Alignment) [33], LLE (Local Linear Embedding) [34], Isomap (Isometric Mapping) [35], LE (Laplacian Eigenmap) [36], and HE (Hessian Eigenmap) [37] are employed along with linear dimensionality reduction technique PCA (Principal Component Analysis) [38]. The first row of figures in the Table 2 contains the original 3D manifolds and subsequent rows contain the 2D output from each of the employed techniques.

1. **Punctured Sphere:** The second column of the Table 2 shows the punctured sphere dataset. This artificially generated dataset contains 4000 data points in 3D. It is originally a 2D disc, which is elevated on the third dimension to make it a sphere. The results show that PCA, LLE, Isomap and HE are unable to determine the true 2D structure as they failed to preserve the intrinsic geometry of the punctured segment marked with red color. Both $CCGA_L$ and $CCGA_N$, along with LTSA and LE preserve the true intrinsic geometry and are able to unfold the true lower space structure. The HSIC based nonlinear dependence is able to find the true affinity among data points which leads to a better association among data points belonging to the same class.
2. **Sine on a hyperboloid:** The third column of the Table 2, shows 629 data points, sine wave wrapped around a hyperboloid in 3D space which is originally a smooth circle on a 2D plane. As evident, $CCGA_L$, $CCGA_N$, and Isomap give the best representation in 2D space while retaining the geometrical characteristics. All other techniques fail to either preserve the circle or its smoothness. These techniques only rely on the Euclidean distance for structural properties estimation and discard a large chunk of the nonlinear properties.
3. **Swiss roll with hole:** This dataset contains 3947 data points modeled in a Swiss roll shape with a small hole in between green and blue portion. When unrolled correctly, it should appear as a 2D flat strip with a hole. Isomap is able to give the closest output to its intrinsic

¹The least classification error for each dataset has been highlighted in bold in Tables 3 to 12

Table 2 CCGA_L and CCGA_N vs state-of-the-art techniques

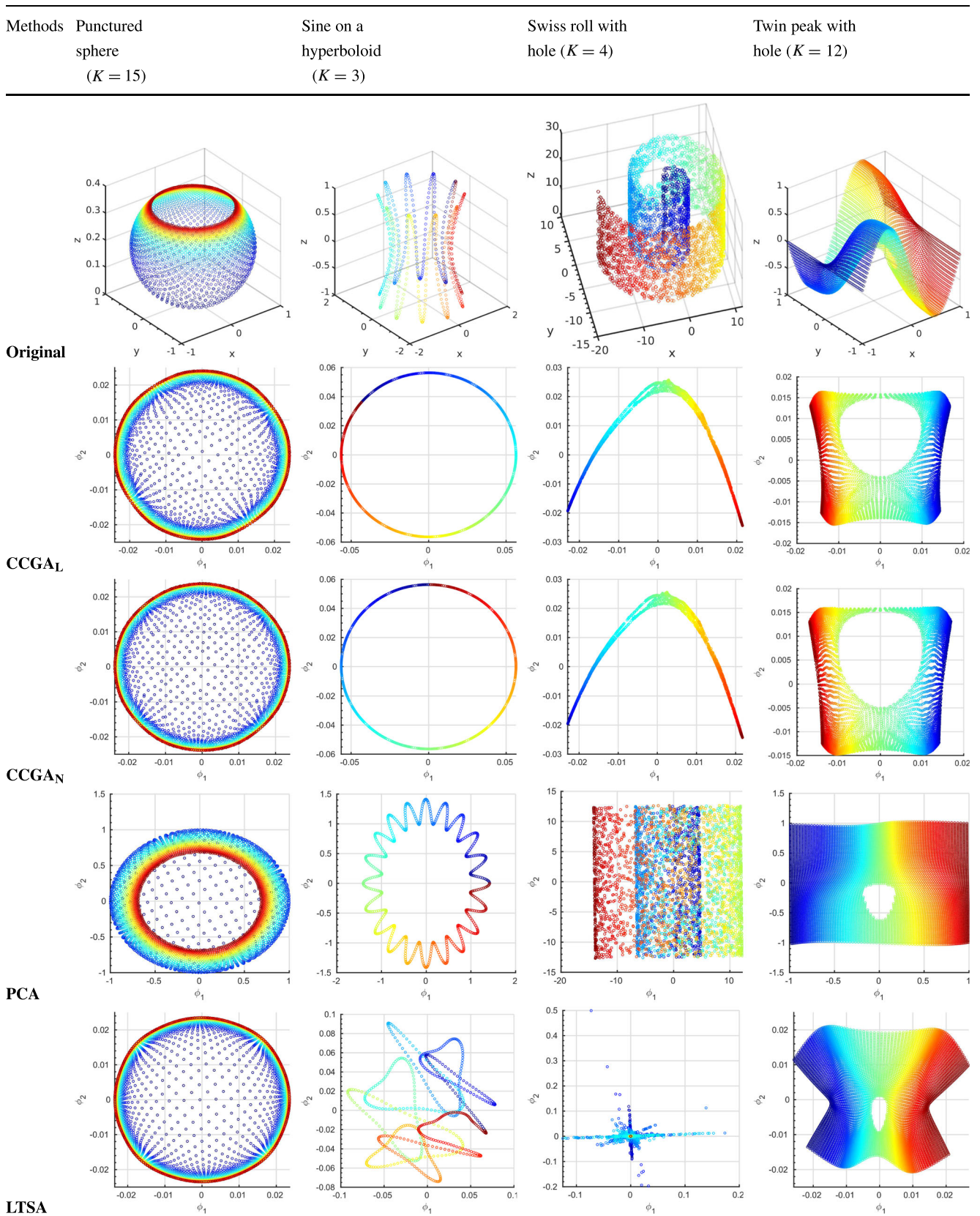
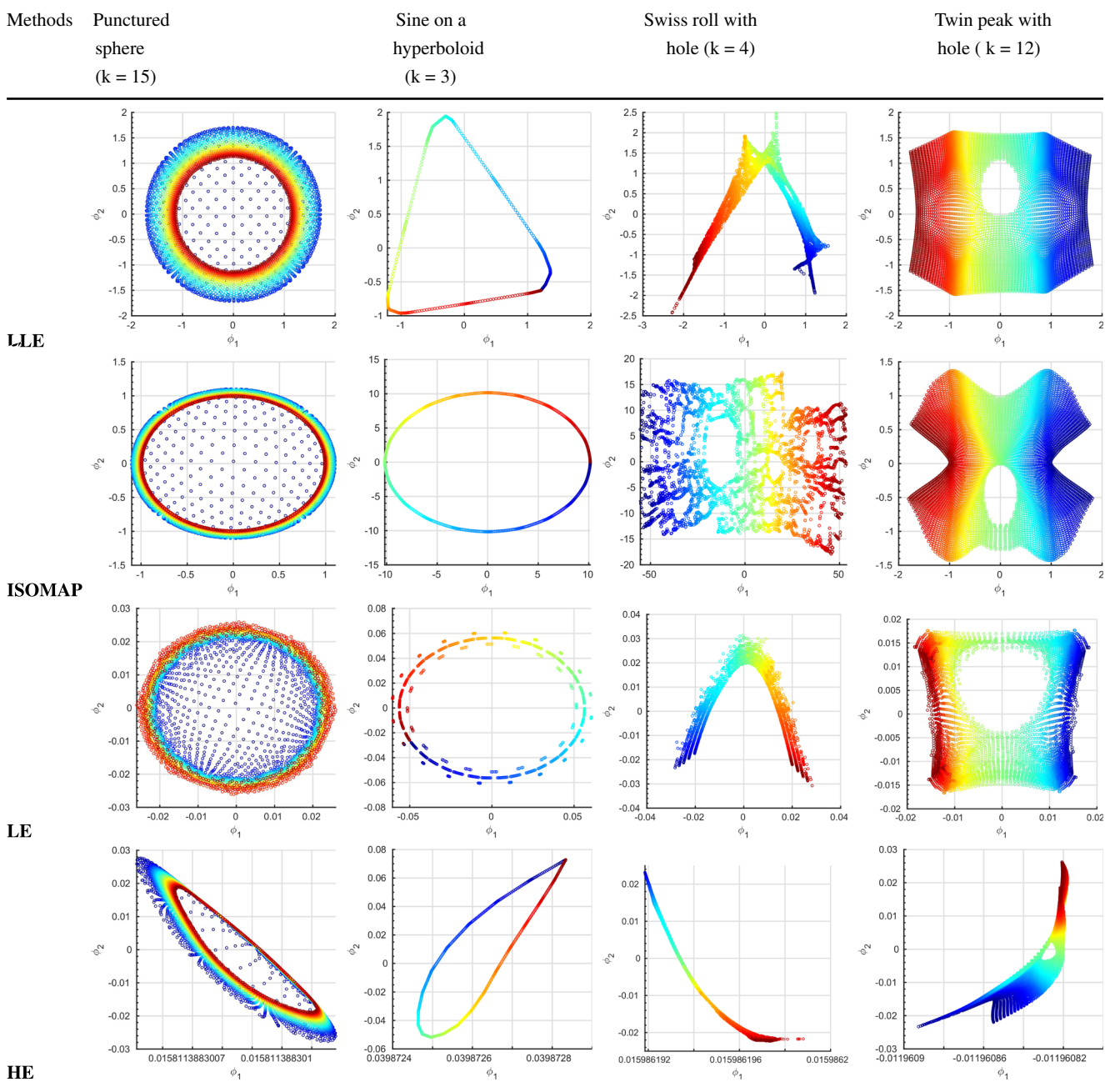


Table 2 (continued)



representation. LLE, $CCGA_L$, and $CCGA_N$ are able to preserve the shape of the hole along with the strip. Other methods failed to preserve the connectivity around the hole. $CCGA_L$, and $CCGA_N$ are not able to better unroll the structure. It seems that they discard some spurious dependencies between points in the hole perimeter.

4. **Twin peak with hole:** The last column of the Table 2, shows the artificially generated Twin peaks with a hole, which is originally a 2D flat strip with a hole containing 6990 data points. The results show that PCA gave

the most accurate representations, followed by $CCGA_L$ and $CCGA_N$. Other methods either failed to unroll the strip to its original structure or the boundary data points remained cluttered. All nonlinear dimensionality reduction techniques are able to retain the hole, but 2D shape distortion is maximum for Isomap as the shortest is not the correct geodesic distance.

Manifold learning results as shown in the Table 2 on the synthetic datasets clearly suggest that both the techniques are able to determine the optimal affinities that preserve the maxi-

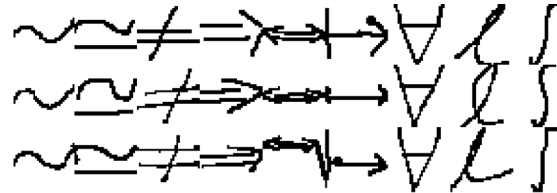
mum intrinsic geometrical properties. This leads to a better association among data points belonging to the same class. It is observed that for all the synthetic datasets, both, $CCGA_L$ and $CCGA_N$ are able to unroll the data and project them to 2D space while retaining the maximum structural characteristics.

4.2 Real world datasets

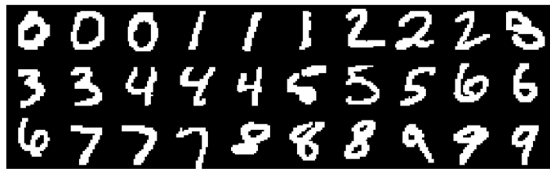
1. **USPS:** This dataset [39] contains scanned images (Fig. 3a) of handwritten digits on envelopes of the US postal services. The original scanned digits varying in



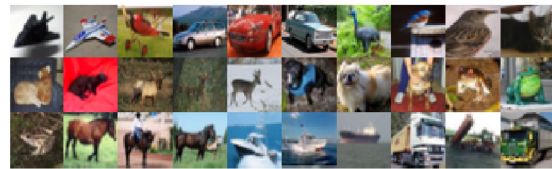
(a) USPS Dataset



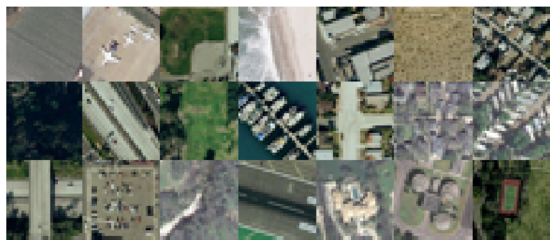
(b) Hasy_v2 Dataset



(c) MNIST Dataset



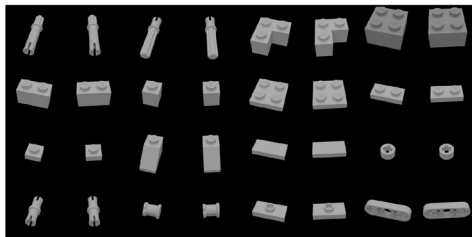
(d) CIFAR-10



(e) UC Merced Land use Dataset



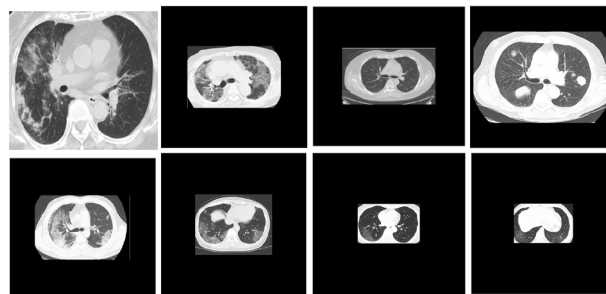
(f) CVPR'09 Dataset



(g) Lego Bricks Dataset



(h) Natural Images Dataset



(i) COVID-19 CT Scan Dataset; First row contains negative patient scans; Second row contains positive patient scans

Fig. 3 Real world Datasets Images

Table 3 Mean error(\pm Standard deviation) LapSVM (Test) $k = 6$

Method \rightarrow Dataset \downarrow	Heat	Binary	EMR ₂₄	EMR ₇₂	CCGA _L	CCGA _N
USPS	5.9853(\pm 2.0131)	6.0026(\pm 2.0067)	28.4868(\pm 1.2284)	28.4868(\pm 1.2284)	3.5657(\pm 1.4774)	2.4492(\pm1.152)
Hasy_v2	22.9591(\pm 0.65604)	3.3982(\pm 2.5612)	13.1766(\pm 0.60032)	13.1766(\pm 0.60032)	2.5943(\pm 0.35458)	2.4104(\pm0.33458)
MNIST	38.4751(\pm 0.54756)	10.4451(\pm 1.767)	35.9541(\pm 0.9027)	35.9541(\pm 0.9027)	7.5191(\pm1.5213)	8.654(\pm 1.9719)
BCI HaLT	20.6458(\pm 0.4408)	20.7146(\pm0.41966)	21.6187(\pm 0.55034)	21.6187(\pm 0.55034)	20.7254(\pm 0.41862)	20.7732(\pm 0.43116)
CIFAR-10	31.1644(\pm 1.0143)	31.3531(\pm 1.0624)	31.3738(\pm 1.068)	31.3738(\pm 1.068)	31.2954(\pm 1.0459)	30.5624(\pm0.93737)
Lego Bricks	13.9991(\pm 8.1623)	10.2096(\pm 9.077)	18.5477(\pm 5.5793)	17.2964(\pm 5.9808)	8.7628(\pm 9.0811)	8.5574(\pm9.3981)
UC Merced Land	27.049(\pm 3.5047)	25.1724(\pm 3.6374)	26.8367(\pm 3.5862)	26.8367(\pm 3.5862)	24.9548(\pm 3.4188)	23.8519(\pm 2.8291)
CVPR'09	28.7065(\pm 1.7688)	28.6331(\pm 1.9164)	29.5558(\pm 2.0614)	29.5558(\pm 2.0614)	24.6106(\pm1.7648)	25.6428(\pm 1.8901)
Natural Images	21.3264(\pm 2.336)	20.1441(\pm 2.4681)	20.8939(\pm 2.179)	20.7964(\pm 1.9059)	17.5111(\pm 2.6718)	16.1957(\pm 2.5134)
COVID-19 CT Dataset	11.4564(\pm 1.4242)	12.2342(\pm 1.7352)	14.2344(\pm 1.5345)	15.3133(\pm 1.4234)	10.3131(\pm 0.9312)	10.0834(\pm0.6859)

size and orientation is resized to 16×16 grayscale images with 7291 training and 2007 test instances.

- Hasy_v2**: This dataset [40] comprises of 168233 single handwritten latex symbols (Fig. 3b) across 369 different classes extracted from HWRT dataset. However, only 9 symbols have been used in the experiment as they contained ≥ 800 samples. The original data matrix is of size 12701×3072 .
- MNIST**: This dataset² contains the images (Fig. 3c) of handwritten digits (0 – 9). Thus, it has 10 classes, each corresponds to a different numeric digit. Each of this image is gray scale of size 28×28 . The dataset has total of 70,000 instances out of 60,000 has been used for the training and rest for testing.
- BCI HaLT**: BCI dataset [41] comprises of brain electroencephalographic (EEG) signals obtained from different imagery part in micro-volt unit. The recorded EEG signals correspond to brain activity of imagining the movement of left leg, right leg, left hand, right hand and tongue. Across multiple subjects, the experiment contained 3 HaLT reading for subject A only of size 2898×3570 .
- CIFAR-10**: This dataset [42] comprises of color images (Fig. 3d) of the objects like cars, cats, airplanes, deer, birds, ships, dogs, trucks, horses, and, frogs. It has 10 different classes with a total of 60,000
- colored images. Each of this image is of size 32×32 with 6000 images per class.
- Lego Bricks**: This dataset³ comprises of gray scale images (Fig. 3g) of the bricks manufactured by Lego. It has total of 16 different classes and each class has ≈ 400 images. Each image is of size 200×200 grayscale pixels.
- UC Merced Land**: This dataset [43] has outdoor images (Fig. 3e) of land divided into 21 classes. Each of this image has been obtained from the USGS National Map Urban Area Imagery collection for various urban areas around the country. Each of this class has 100 images and each image is of size 256×256 . Out of these 100 images, 50 images per class are randomly picked for the training set and rest 50 images per class are used for testing set.
- CVPR'09**: This dataset [44] comprises of 67 classes of image data (Fig. 3f). Each class has more than 100 images. Each image has been reduced to equal size of $32 \times 32 \times 3$. Finally, a matrix of size 15620×3072 which is further divided into training 70% and testing set
- Natural Images**: This dataset [45] contains images (Fig. 3h) of different objects like car, cat, airplane, motorbike, flower, person and fruit. It has total of 8

²<http://yann.lecun.com/exdb/mnist/>

³Mecabricks.com

Table 4 Mean error(\pm Standard deviation) LapSVM (Unlabeled) $k = 6$

Method \rightarrow Dataset \downarrow	Heat	Binary	EMR ₂₄	EMR ₇₂	CCGA _L	CCGA _N
USPS	5.9196(\pm 2.0438)	5.9352(\pm 2.059)	33.4718(\pm 1.1126)	33.4718(\pm 1.1126)	3.0609(\pm 1.6038)	1.6812(\pm1.1766)
Hasy_v2	10.7445(\pm 1.2973)	3.2973(\pm 2.6276)	13.3647(\pm 0.63509)	13.3647(\pm 0.63509)	2.4118(\pm 0.36646)	2.2367(\pm0.33978)
MNIST	8.2996(\pm 1.8522)	1.5(\pm 2.1197)	36.2816(\pm 0.93587)	36.2816(\pm 0.93587)	1.1498(\pm 1.4258)	1.049(\pm1.3817)
BCI HaLT	19.4467(\pm 0.44692)	19.509(\pm0.45214)	20.2633(\pm 0.52944)	20.2633(\pm 0.52944)	19.5453(\pm 0.45401)	19.6514(\pm 0.43513)
CIFAR-10	31.0024(\pm 1.0302)	31.1762(\pm 1.0589)	31.2028(\pm 1.0641)	31.2028(\pm 1.0641)	31.1123(\pm 1.0555)	30.3794(\pm0.95679)
Lego Bricks	12.2481(\pm 8.4792)	9.2706(\pm 9.1341)	19.0064(\pm 5.8736)	17.6866(\pm 6.0596)	7.771(\pm 9.0185)	6.9222(\pm8.9307)
UC Merced Land	18.6597(\pm 2.0702)	22.1499(\pm 1.8301)	18.8922(\pm2.311)	18.8922(\pm 2.311)	26.5602(\pm 2.622)	29.4454(\pm 2.6659)
CVPR'09	25.6905(\pm 1.6005)	25.6246(\pm 1.8902)	27.0735(\pm 1.7924)	27.0735(\pm 1.7924)	22.6165(\pm 1.7756)	22.584(\pm1.6859)
Natural Images	19.6906(\pm 1.2029)	17.8456(\pm 1.652)	19.441(\pm 1.7698)	19.4345(\pm 1.7707)	15.2854(\pm1.3185)	16.8485(\pm 1.3853)
COVID-19 CT Dataset	12.4564(\pm 1.9845)	12.4241(\pm 1.0995)	17.5433(\pm 2.6344)	18.2423(\pm 1.2342)	12.0345(\pm0.5435)	12.4566(\pm 0.9353)

classes and each this grayscale image has been resized to 28×28 . The complete dataset is of size 7599×784 .

10. **COVID-19 CT Scan Dataset** : This dataset [46] contains CT scan images (Fig. 3i) of COVID-19 positive and negative patients. It has a total of 451 CT

scan grayscale images out of which 275 belonged to positive patients and rest 195 were negative. During pre-processing, all images were changed to standard 512×512 pixels. Data was further split to 50% as training and 50% as testing set. The experiment was

Table 5 Mean error (\pm Standard deviation) LapRLSC (Test) with $k = 6$

Method \rightarrow Dataset \downarrow	Heat	Binary	EMR ₂₄	EMR ₇₂	CCGA _L	CCGA _N
USPS	6.0525(\pm 1.9381)	6.062(\pm 1.9493)	27.7172(\pm 1.7853)	27.7172(\pm 1.7853)	3.6006(\pm 1.4074)	2.5867(\pm1.1814)
HASY_v2	25.2406(\pm 0.96617)	2.1822(\pm0.32696)	12.2475(\pm 0.69306)	12.2475(\pm 0.69306)	2.4943(\pm 0.34876)	2.3303(\pm 0.34844)
MNIST	38.4103(\pm 0.57778)	10.1852(\pm1.4786)	35.5803(\pm 0.94473)	35.5803(\pm 0.94473)	13.4018(\pm 2.9515)	18.0368(\pm 3.0749)
BCI HaLT	19.5864(\pm 0.58403)	19.5775(\pm 0.56837)	20.2675(\pm 0.61025)	20.2675(\pm 0.61025)	18.0687(\pm0.58031)	18.73(\pm 0.52904)
CIFAR-10	28.6935(\pm 0.7116)	28.7648(\pm 0.71153)	28.7747(\pm 0.70966)	28.7747(\pm 0.70966)	28.7566(\pm 0.71979)	28.6896(\pm0.71141)
Lego Bricks	13.9378(\pm 8.4856)	10.7887(\pm 9.2475)	18.6991(\pm 5.8643)	17.2439(\pm 6.6636)	7.9037(\pm 9.1091)	6.5089(\pm9.4058)
UC Merced Land	27.0733(\pm 3.4778)	26.1643(\pm 3.7012)	26.4614(\pm 3.4997)	26.4581(\pm 3.4752)	23.5705(\pm 3.6623)	22.2962(\pm3.4904)
CVPR'09	28.7285(\pm 1.8327)	28.6894(\pm1.7715)	29.5571(\pm 2.065)	29.5571(\pm 2.065)	28.7167(\pm 1.9946)	28.7214(\pm 1.8789)
Natural Images	21.1371(\pm 1.6499)	20.0112(\pm 2.5415)	20.991(\pm 2.2732)	21.0072(\pm 2.2715)	18.911(\pm 2.7065)	17.8757(\pm2.3333)
COVID-19 CT Dataset	13.2383(\pm1.5345)	13.4234(\pm 1.4232)	16.4322(\pm 2.5545)	17.3133(\pm 2.9891)	14.3131(\pm 0.93423)	15.2342(\pm 0.9859)

Table 6 Mean error (\pm Standard deviation) LapRLSC (Unlabeled) $k = 6$

Method \rightarrow Dataset \downarrow	Heat	Binary	EMR ₂₄	EMR ₇₂	CCGA _L	CCGA _N
USPS	5.8922(\pm 2.0385)	5.8928(\pm 2.0405)	30.2328(\pm 1.5825)	30.2328(\pm 1.5825)	3.0793(\pm 1.4541)	1.7102(\pm0.9751)
Hasy_v2	10.5111(\pm 1.2068)	2.0817(\pm0.33731)	12.3712(\pm 0.71244)	12.3712(\pm 0.71244)	2.4037(\pm 0.35836)	2.2252(\pm 0.36416)
MNIST	8.0237(\pm 1.8893)	1.3303(\pm 2.0178)	35.6724(\pm 1.034)	35.6724(\pm 1.034)	1.1592(\pm 1.4827)	1.0551(\pm1.3974)
BCI HaLT	18.7148(\pm 0.58304)	18.6783(\pm0.59023)	19.1057(\pm 0.46685)	19.1057(\pm 0.46685)	18.6926(\pm 0.59944)	18.796(\pm 0.58017)
CIFAR-10	28.6764(\pm 0.67734)	28.7455(\pm 0.66441)	28.7507(\pm 0.66312)	28.7507(\pm 0.66312)	28.7434(\pm 0.66381)	27.5794(\pm0.68236)
Lego Bricks	11.5328(\pm 8.1146)	8.9066(\pm 8.7905)	18.6421(\pm 5.3925)	18.4514(\pm 5.9489)	7.198(\pm 8.9484)	6.7162(\pm8.6574)
UC Merced Land	18.6723(\pm 2.067)	28.2353(\pm 2.1595)	18.3936(\pm2.277)	18.3936(\pm 2.277)	24.0322(\pm 2.5972)	25.5714(\pm 2.8975)
CVPR'09	25.8091(\pm 1.6133)	25.7981(\pm 1.7799)	27.0735(\pm 1.7923)	27.0735(\pm 1.7923)	24.7966(\pm 1.8538)	23.7501(\pm1.7327)
Natural Images	19.7043(\pm 1.2225)	18.0463(\pm 2.4952)	19.419(\pm 1.777)	19.4072(\pm 1.7758)	18.7154(\pm 2.436)	16.632(\pm1.4767)
COVID-19 CT Dataset	15.3423(\pm 1.4564)	14.5435(\pm 1.0543)	12.4322(\pm1.3785)	14.6453(\pm 1.5354)	13.5433(\pm 1.5453)	14.2342(\pm 1.5859)

Table 7 Mean error (\pm Standard deviation) LapSVM (Test) $k = 10$

Method \rightarrow Dataset \downarrow	Heat	Binary	EMR ₂₄	EMR ₇₂	CCGA _L	CCGA _N
USPS	5.0859(\pm 2.1001)	5.0326(\pm 1.9367)	27.8768(\pm 1.1293)	26.4564(\pm 0.9583)	3.1552(\pm 1.2774)	1.2462(\pm1.003)
Hasy_v2	10.4718(\pm 0.69454)	3.1924(\pm 2.2765)	12.872(\pm 0.70001)	12.2332(\pm 0.50045)	1.9943(\pm0.3623)	2.0301(\pm 0.13543)
MNIST	36.7321(\pm 0.32343)	9.1432(\pm 1.773)	32.1237(\pm 0.6037)	33.7543(\pm 0.7047)	6.3123(\pm1.1314)	6.3654(\pm 2.001)
BCI HaLT	18.4658(\pm 0.3206)	17.2564(\pm0.4016)	19.4881(\pm 0.5932)	20.5057(\pm 0.51053)	18.1453(\pm 0.43827)	17.7732(\pm 0.43116)
CIFAR-10	30.0544(\pm 1.0093)	30.5231(\pm 1.0443)	30.2723(\pm 1.068)	31.5732(\pm 1.108)	30.054(\pm 1.0309)	29.4214(\pm0.8334)
Lego Bricks	12.3784(\pm 7.2413)	9.1209(\pm 8.134)	16.3517(\pm 4.4453)	17.1943(\pm 5.1908)	7.4651(\pm 7.0421)	6.3574(\pm4.5941)
UC Merced Land	25.105(\pm 3.4276)	24.0922(\pm 3.6374)	25.1233(\pm 2.1352)	25.1267(\pm 3.1532)	23.1344(\pm 3.4188)	22.4519(\pm 1.92191)
CVPR'09	26.5405(\pm 1.7556)	26.2122(\pm 1.8141)	27.3788(\pm 2.7789)	28.1258(\pm 2.0614)	26.1162(\pm1.4555)	27.1232(\pm 1.7821)
Natural Images	20.1624(\pm 2.1326)	19.1214(\pm 2.4323)	19.1282(\pm 2.0238)	19.8732(\pm 1.8009)	18.0051(\pm 2.6718)	17.9899(\pm1.9134)
COVID-19 CT Dataset	10.6564(\pm 1.3672)	12.3696(\pm 1.45784)	13.9689(\pm 2.0123)	17.2343(\pm 2.1533)	10.7565(\pm 0.8312)	9.645(\pm0.5645)

Table 8 Mean error(\pm Standard deviation) LapSVM (Unlabeled) $k = 10$

Method \rightarrow Dataset \downarrow	Heat	Binary	EMR ₂₄	EMR ₇₂	CCGA _L	CCGA _N
USPS	5.1132(\pm 2.1423)	5.0642(\pm 2.059)	31.3341(\pm 1.1214)	31.4123(\pm 1.1335)	2.0103(\pm 1.3448)	1.2834(\pm1.3442)
Hasy_v2	10.1125(\pm 1.2334)	2.9936(\pm 1.5244)	12.3347(\pm 0.5389)	12.3347(\pm 0.43509)	2.4118(\pm 0.36646)	1.9367(\pm0.33978)
MNIST	7.1294(\pm 1.3452)	1.4345(\pm 2.0497)	35.2312(\pm 0.73327)	35.2336(\pm 0.9232)	1.3434(\pm 1.4348)	1.0445(\pm1.3284)
BCI HaLT	18.3251(\pm 0.3445)	18.1649(\pm0.45214)	19.4343(\pm 0.52944)	19.3453(\pm 0.5434)	18.5443(\pm 0.45401)	19.6514(\pm 0.43513)
CIFAR-10	30.1334(\pm 1.1342)	30.1342(\pm 1.1549)	30.3434(\pm 1.0445)	30.1048(\pm 1.0441)	30.0120(\pm 1.04233)	29.3234(\pm0.89569)
Lego Bricks	11.348(\pm 8.1272)	8.3206(\pm 8.2311)	17.7764(\pm 5.5343)	17.5466(\pm 6.0496)	6.4231(\pm 8.0435)	6.0334(\pm8.9307)
UC Merced Land	17.23597(\pm 2.0212)	20.1323(\pm 1.2323)	17.3235(\pm2.7371)	17.5422(\pm 2.551)	24.4302(\pm 2.6656)	27.6654(\pm 2.6566)
CVPR'09	24.1105(\pm 1.2305)	24.2346(\pm 1.3202)	26.4555(\pm 1.7454)	26.0554(\pm 1.7543)	24.6543(\pm 1.5335)	23.5345(\pm1.3535)
Natural Images	18.6546(\pm 1.5353)	16.84536(\pm 1.4622)	18.6541(\pm 1.5548)	18.4665(\pm 1.6457)	16.2634(\pm 1.36455)	16.0065(\pm1.2856)
COVID-19 CT Dataset	11.764(\pm 1.6445)	13.46411(\pm 1.0645)	15.5644(\pm 2.7664)	17.6463(\pm 1.7552)	11.0646(\pm0.7465)	10.34434(\pm 0.7647)

Table 9 Mean error (\pm Standard deviation) LapRLSC (Test) with $k = 10$

Method \rightarrow Dataset \downarrow	Heat	Binary	EMR ₂₄	EMR ₇₂	CCGA _L	CCGA _N
USPS	5.03425(\pm 1.3243)	5.0432(\pm 1.3443)	26.6642(\pm 1.7424)	26.7432(\pm 1.4353)	2.1734(\pm 1.30755)	1.9867(\pm1.0445)
Hasy_v2	24.2456(\pm 0.953417)	2.1545(\pm0.3457)	11.25455(\pm 0.6454)	11.2454(\pm 0.7656)	2.8937(\pm 0.34876)	2.3324(\pm 0.34564)
MNIST	36.3134(\pm 0.44578)	9.4152(\pm1.4386)	34.4403(\pm 0.94473)	34.5423(\pm 0.8443)	12.4434(\pm 2.0955)	17.3448(\pm 2.34749)
BCI HaLT	19.5864(\pm 0.58403)	19.5775(\pm 0.56837)	20.2675(\pm 0.61025)	20.2675(\pm 0.61025)	19.5687(\pm0.58031)	19.63(\pm 0.52904)
CIFAR-10	26.6434(\pm 0.6766)	27.4348(\pm 0.6453)	26.7437(\pm 0.70966)	27.4237(\pm 0.70966)	27.7436(\pm 0.6146)	26.3496(\pm0.742341)
Lego Bricks	12.6487(\pm 6.46456)	9.64487(\pm 8.2665)	17.65641(\pm 4.8664)	16.6459(\pm 6.5666)	7.637(\pm 8.16446)	6.5645(\pm 7.4456)
UC Merced Land	26.5345(\pm 3.5345)	25.15656(\pm 2.4534)	25.5345(\pm 2.4557)	25.45381(\pm 2.4442)	23.5445(\pm 2.64523)	23.2423(\pm2.4544)
CVPR'09	27.7565(\pm 1.3236)	27.4564(\pm 1.7715)	28.6453(\pm 1.8565)	28.6772(\pm 1.7665)	24.6567(\pm 2.0046)	23.6214(\pm1.8789)
Natural Images	20.14471(\pm 1.6543)	19.052(\pm 1.5535)	19.4491(\pm 1.9732)	20.0552(\pm 2.4515)	17.9331(\pm 2.7065)	17.5357(\pm1.3543)
COVID-19 CT Dataset	11.3423(\pm1.7225)	12.2426(\pm 1.5345)	17.6444(\pm 2.6423)	18.4543(\pm 1.9644)	15.4531(\pm 0.9354)	14.5154(\pm 1.9565)

repeated 20 times, and in each round, 8 images from both classes were randomly labeled.

Extensive experiments are performed with **LapSVM** and **LapRLSC** classifiers. Root mean square error with standard deviation is reported in separate tables for unlabeled and test set Experiments on these datasets are performed by varying the k NN paramete as the affinity matrix used for constructing the graph depends on k , the number of neighbors. This parameter k is varied from 6 to 15. It is observed that the classification accuracy for all datasets increases, thus, with enriched neighborhood more accurate affinity is learned. On further increasing the value ($k = 16$) the accuracy suddenly dips and similar trend is followed on further increasing the neighborhood. The classification error for value of $k = 6$ and $k = 10$ for both LapSVM and LapRLSC are reported in the Tables 3, 4, 5, 6, 7, 8, 9, and 10.

On **USPS** dataset, $CCGA_L$ and $CCGA_N$ have highest classification accuracy as compared to the existing state-of-the-art techniques for both test and unlabeled set. On USPS test set, $CCGA_N$ achieves the classification accuracy of $> 98\%$ and $CCGA_L$ stands closest to it with accuracy $> 97\%$. On the other hand, classification accuracy with heat and binary affinities are $> 95\%$ and $> 94\%$ respectively. In contrast, EMR_{24} and EMR_{72} have $< 72\%$ classification accuracy, which is least among all. A similar trend in the classification accuracy is obtained for the

unlabeled set, where $CCGA_N$ has the best accuracy of $> 95.5\%$ and EMR_{72} has the least accuracy of $< 67\%$. This reveals that HSIC based dependence value obtained between data points is able to capture the nonlinear dependence between data points. This leads to better classification accuracy. On **HASY_v2** dataset for test as well as unlabeled set $CCGA_N$ has most accurate result $> 98\%$ followed by $CCGA_L$ ($> 98\%$) and binary ($> 97\%$). EMR_{24} and EMR_{72} have almost same accuracy ($\approx 87\%$). Heat affinity has the worst classification accuracy $\approx 78\%$ against others. Both LapRLSC and LapSVM have approximately similar result for Hasy_v2 dataset. This result trend suggests that the $CCGA_N$ better decodes the neighborhood connectivities to determine a more enriched affinity matrix. On **MNIST** dataset for test set, $CCGA_L$ provides the most accurate result for handwritten digits with classification accuracy $\approx 87\%$ whereas other proposed technique $CCGA_N$ has the classification accuracy $\approx 82\%$. Only accuracy obtained with binary affinity stand closest to our techniques ($> 90\%$). Other techniques have comparatively very poor classification accuracy ($< 65\%$). On other hand for unlabeled set, results are much better for all techniques like $CCGA_N$, $CCGA_L$ and binary weight have $\approx 98\%$ accuracy. Again EMR_{24} and EMR_{72} have the least accuracy. On **BCI HaLT** dataset, all techniques have almost same classification accuracy for both test and unlabeled set under LapSVM as well as LapRLSC. Binary weight and $CCGA_N$ have the highest accuracy

Table 10 Mean error (\pm Standard deviation) LapRLSC (Unlabeled) $k = 10$

Method → Dataset ↓	Heat	Binary	EMR ₂₄	EMR ₇₂	CCGA _L	CCGA _N
USPS	5.0491(± 2.1335)	5.8348(± 2.1465)	29.2432(± 1.3425)	28.2248(± 1.4245)	2.4553(± 1.0541)	1.2153(± 0.7756)
Hasy_v2	9.7811(± 1.2068)	1.9987(± 0.23361)	10.42342(± 0.64243)	11.4422(± 0.61424)	1.4237(± 0.35836)	2.2122(± 0.3123)
MNIST	7.31237(± 1.3893)	1.9903(± 2.9178)	33.6231(± 1.3384)	32.4244(± 1.3324)	1.4394(± 1.43447)	1.0001(± 0.24234)
BCI HaLT	17.42348(± 0.48423)	17.623483(± 0.4243)	18.4457(± 0.3645)	18.1237(± 0.5623)	17.4232(± 0.5234)	17.7423(± 0.48047)
CIFAR-10	27.6424(± 0.62374)	27.7425(± 0.5241)	27.75237(± 0.56312)	27.72307(± 0.36312)	26.7434(± 0.4481)	25.5794(± 0.6446)
Lego Bricks	10.5428(± 7.1426)	7.9446(± 8.79423)	17.6321(± 4.425)	17.444(± 5.94239)	7.23438(± 7.92344)	5.7424(± 7.64224)
UC Merced Land	17.6323(± 2.5547)	26.2443(± 2.1595)	17.3936(± 2.277)	17.536(± 2.277)	23.5622(± 2.5372)	24.5524(± 1.93975)
CVPR'09	24.5391(± 1.5553)	24.7531(± 1.7534)	26.0355(± 1.534523)	26.053(± 1.7923)	22.3426(± 1.8442)	21.4244(± 1.1424)
Natural Images	18.7423(± 1.22425)	17.2423(± 2.4422)	17.2424(± 1.4234)	18.4534(± 1.5335)	16.7534(± 2.5356)	16.053(± 1.3453)
COVID-19 CT Dataset	12.9423(± 1.5345)	12.4131(± 1.3445)	14.4272(± 1.9554)	15.3533(± 2.7232)	12.3645(± 0.65434)	14.2534(± 0.5485)

(> 83%). Whereas, others like heat affinity, $CCGA_L$, EMR_{24} and EMR_{72} have accuracy that lies between 78% to 80%. This shows that dataset intrinsic geometry has been well explored by the given techniques in order to determine the connectivities by accommodating both linear and nonlinear characteristics. On **CIFAR-10** dataset, there is a similar trend in the classification result. For both test and unlabeled sets classification accuracy has not much variation, the accuracy is $\approx 70\%$. $CCGA_N$ leads among all by 1% with classification accuracy 71%. On another object detection dataset, like **Lego Bricks**, $CCGA_L$ and $CCGA_N$ give best label prediction results. They achieve an accuracy of > 94% for both test and unlabeled sets. The next most accurate one is binary weight with accuracy > 91%. EMR_{24} and EMR_{72} are the worst performer with only > 86% accuracy. These result values indicate that the graphs obtained with EMR_{24} and EMR_{72} discard the major chunk of information present in the feature matrix of dataset. On other hand $CCGA_L$ and $CCGA_N$ better explore the feature matrix to build a high information contained graph. On **UC Merced** dataset, $CCGA_N$ has the highest accuracy (> 77%) under LapSVM and LapRLSC for test set. Whereas EMR_{24} (> 78%) under LapSVM for

unlabeled set and binary weight (72%) for test set under LapRLSC. There is a slight increase in the accuracy for all techniques under both LapSVM and LapRLSC for $k = 10$. For **CVPR'09** dataset, the proposed techniques $CCGA_L$ and $CCGA_N$ outperform others for both test and unlabeled sets. The average accuracy for all techniques is $\approx 72\%$ under both LapSVM and LapRLSC. On **Natural image** dataset, $CCGA_N$ outperform other techniques by a non negligible factor. Both $CCGA_L$ and $CCGA_N$ techniques have > 84% classification accuracy for both test and unlabeled set. Other techniques have accuracy $\approx 80\%$ under LapSVM and LapRLSC. For **COVID-19 CT** dataset, heat affinity has the maximum classification accuracy of $\approx 90\%$ and other techniques have accuracy > 92% for both unlabeled and test sets. The nonlinear dependencies estimated with HSIC determine the connectivities among data points which leads to better association among same class data points.

In order to further validate the efficacy of $CCGA_L$ and $CCGA_N$, random walk plots are constructed with state-of-the-art graph affinities on BCI the dataset are shown in the Fig. 4. An affinity matrix is considered to be better than other affinity matrices if its random walk plot has clear clusters. As evident, the random walk is shown in Fig. 4b

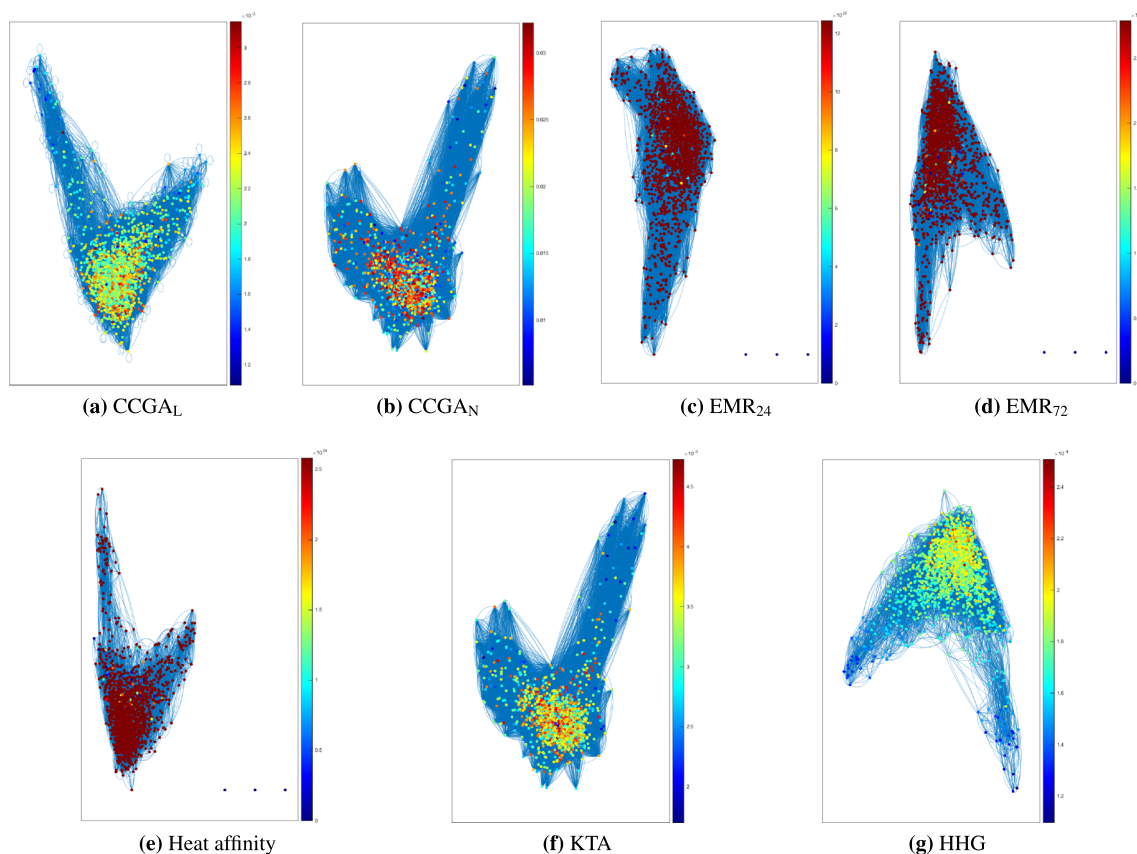


Fig. 4 Random walk on BCI 5F dataset

constructed using $CCGA_N$ contains cluster patches that define a strong intra-neighbor connectivity and sparse inter-neighbor edges. This supports the superior performance of $CCGA_N$ during classification as the affinity matrix helps in identifying clear clusters on the data points. The random walk using $CCGA_L$ (Fig. 4a) gives a comparable result wherein different classes of the dataset are discernible. The plots for EMR_{24} (Fig. 4c) and EMR_{72} (Fig. 4d) have less clear clusters. On the other hand, the random walk over non-parametric methods KTA^4 and HHG^4 do not have much recognizable clusters, while the former affinity do define some high strength groups, the later gave uniform affinity to all data points. Similarly, the heat affinity random walk also diluted the distinguishable clusters by assigning high affinity to every edge (Fig. 4e) as it captures linear dependencies only.

It is evident from the results of non-linear dimensionality reduction and the random walk plots that $CCGA_L$ and $CCGA_N$ are capable of preserving the smooth intrinsic geometrical properties better than other methods. Similarly, increased classification accuracy across the standard handwritten, BCI, object and scene detection dataset show that the graph affinity obtained gives information about the unexplored connectivities in the data which outperform the other affinity methods by a considerable margin.

4.3 Non parametric statistical tests

$CCGA_L$ and $CCGA_N$ are based on HSIC, which is a non-parametric statistical measure. While in the previous experiments, they have been compared with existing state-of-the-art parametric affinity methods, it is essential to compare them with state-of-the-art non-parametric statistical test measures. We have selected two such methods, namely kernel target alignment (KTA) [47] and Heller Heller Gorfine (HHG) [47]. As KTA works in feature space by aligning two kernel functions or a kernel and a target function which makes it very similar to HSIC based affinity method. On the other hand, HHG computes the statistical measure from the underlying Euclidean metric, which further allows the comparison to explore if statistical test based on Euclidean norm can appropriately enforce the smoothness constraint in the presence of nonlinear relation. Both KTA and HHG aim to compute the dependence between the samples of two random variables assuming they are not independent of each other.

KTA tries to learn data embedding in feature space such that the data points follow nice clustering properties i.e. data points belonging to the same class should be mapped spatially closer to each other as compared to

data points from different classes. Thus, the intra-class distance between data points in feature space remains close to zero, and inter-class distance, greater than zero. Over the data points in X , KTA employs two kernels $\kappa_1 = \sum_{i=j=1}^n \kappa_1(x_i, x_j)$ and $\kappa_2 = \sum_{i=j=1}^n \kappa_2(x_i, x_j)$. Further, the alignment between κ_1 and κ_2 is performed through

$$W = \frac{\langle \kappa_1, \kappa_2 \rangle_F}{\sqrt{\langle \kappa_1, \kappa_1 \rangle_F \langle \kappa_2, \kappa_2 \rangle_F}}$$

where, $\langle \kappa_1, \kappa_2 \rangle_F = \sum_{i=j=1}^n \kappa_1(x_i, x_j)\kappa_2(x_i, x_j)$. It is also interpreted as the cosine angle between two bi-dimensional vectors κ_1 and κ_2 . The measure of alignment between two different kernels on same pair of data points allows to identify true linear neighbors situated in feature space, this corresponds to small spatial distance or larger affinity. Similarly, the non-aligning kernels of data points would result in smaller affinity values representing absence of linear relationship as required by regularization to enforce function smoothness across the dense linear regions.

The other non-parametric statistical test HHG estimates the dependence between two random variables through norm distance. HHG assumes that if the two random variables X and X' are dependent then there exists a point (x_0, x'_0) and radii R_x and $R_{x'}$ such that the joint distribution of X and X' should be different from product of the marginal distribution of balls around (x_0, x'_0) . The independence test performed with HHG is given by,

$$HHG = \sum_{i=1}^n \sum_{j=1, j \neq i}^n = S(i, j)$$

where, $S(i, j)$ is computed by observing the two dichotomous random variables for n observation in both random variables,

$$S(i, j) = \frac{(n - 2)\{A_{12}(i, j)A_{21}(i, j) - A_{11}(i, j)A_{22}(i, j)\}^2}{A_1(i, j)A_2(i, j)A_{.1}(i, j)A_{.2}(i, j)}$$

Both KTA and HHG values have been used as an affinity metric in place of $CCGA_N$. Further, LapSVM and LapRLSC were executed for both methods on all benchmark datasets. The performance of all three non-parametric affinity methods have been compared on the mean classification error over both unlabeled and test set. The mean error for test and unlabeled set have been reported in Table 11 and Table 12. On **USPS** dataset, for both test and unlabeled set, $CCGA_N$ outperformed both KTA and HHG by achieving an accuracy of $> 98\%$. KTA also gave a close result and remained at second position whereas HHG gave poor accuracy for LapSVM ($> 71\%$). In case of **Hasy_v2** dataset, result of all three non-parametric affinity methods remained close and $CCGA_N$ managed to marginally outperform others. A similar trend can be observed in other datasets like MNIST, BCI HaLT, CIFAR-10 and Lego brick dataset.

⁴For explanation refer Section 4.3.

Table 11 Mean error (\pm Standard deviation) for Test set ($k = 6$)

Method \rightarrow	KTA		HHG		CCGA _N	
Dataset \downarrow	LapSVM	LapRLSC	LapSVM	LapRLSC	LapSVM	LapRLSC
USPS	3.3(\pm 1.5344)	3.8447(1.9252)	24.0332(\pm 1.6126)	23.4384(\pm 1.8533)	2.4104(\pm0.33458)	2.5867(\pm 1.1814)
Hasy_v2	5.0219(\pm 0.3899)	4.7491(0.39956)	3.5963(\pm 2.5182)	2.2731(\pm 0.3316)	2.4104(\pm 0.33458)	2.3303(\pm 0.34844)
MNIST	9.2015(\pm 2.0432)	18.511(3.0134)	10.4607(\pm 1.7727)	10.214(\pm 1.4854)	8.654(\pm1.9719)	18.0368(\pm 3.0749)
BCI	24.8703(\pm 0.39415)	29.7855(0.57972)	20.7107(\pm 0.42489)	19.5775(\pm 0.56837)	20.7732(\pm 0.43116)	18.73(\pm0.52904)
HaLT						
CIFAR-10	33.7413(\pm 1.3908)	35.212(\pm 1.454)	31.3532(\pm 1.0618)	30.7642(\pm 0.7105)	30.5624(\pm 0.93737)	28.6896(\pm0.71141)
Lego	13.321(\pm 9.323)	12.983(\pm 8.546)	10.1609(\pm 9.0374)	10.0636(\pm 9.0922)	8.5574(\pm 9.3981)	6.5089(\pm9.4058)
BricksS						
UC Merced	26.763(\pm 3.2439)	26.1543(\pm 3.6545)	25.0995(\pm 3.5483)	26.0824(\pm 3.7675)	23.8519(\pm 2.8291)	22.2962(\pm3.4904)
Land						
CVPR'09	32.765(\pm 1.652)	24.22(\pm1.232)	28.6259(\pm 1.8406)	26.0824(\pm 3.7675)	25.6428(\pm 1.8901)	28.7214(\pm 1.8789)
Natural	25.1902(\pm 3.978)	26.1624(3.6115)	19.9342(\pm 2.1742)	20.3912(\pm 2.7141)	16.1957(\pm2.5134)	17.8757(\pm 2.3333)
Images						

Though for all these datasets, the classification accuracy of CCGA_N remained only marginally better than KTA and HHG. In remaining datasets, out of the three non-parametric affinity methods, HHG gave the least accurate results whereas the accuracy results of KTA and CCGA_N remained very close.

The non parametric statistical evaluation carried out on various datasets in terms of classification accuracy suggests that HSIC based affinity gave more accurate intrinsic information. From Tables 11 and 12, it can be concluded that KTA and CCGA_N are performed similar in classification accuracy and same trend has been repeated

Table 12 Mean error (\pm Standard deviation) for Unlabeled set ($k = 6$)

Method \rightarrow	KTA		HHG		CCGA _N	
Dataset \downarrow	LapSVM	LapRLSC	LapSVM	LapRLSC	LapSVM	LapRLSC
USPS	2.7856(\pm 1.065)	2.8805(\pm 1.0663)	26.2032(\pm 1.3588)	25.6689(\pm 1.4892)	1.6812(\pm1.1766)	1.7102(\pm 0.9751)
Hasy_v2	3.8451(\pm 0.411)	4.8413(\pm 0.41825)	3.5037(\pm 2.5778)	2.176(\pm 0.35363)	2.2367(\pm 0.33978)	2.2252(\pm0.36416)
MNIST	1.106(\pm 4.3124)	4.1049(\pm 1.3093)	3.5013(\pm 2.1274)	2.3337(\pm 2.0545)	1.049(\pm1.3817)	1.0551(\pm 1.3974)
BCI	19.832(\pm 0.40552)	19.0049(\pm 0.56849)	19.5145(\pm 0.45644)	18.6783(\pm 0.59023)	19.6514(\pm 0.43513)	18.796(\pm 0.58017)
HaLT						
CIFAR-10	32.834(\pm 1.178)	30.512(\pm 1.768)	31.1762(\pm 1.0589)	28.7446(\pm 0.6641)	30.3794(\pm 0.95679)	27.5794(\pm0.68236)
Lego	10.418(\pm 7.432)	11.623(\pm 6.445)	8.9558(\pm 8.8238)	8.9379(\pm 8.6683)	6.9222(\pm 8.9307)	6.7162(\pm 8.6574)
Bricks						
UC Merced	25.765(\pm 1.232)	26.141(\pm 1.242)	21.5084(\pm1.8051)	27.3655(\pm 2.183)	29.4454(\pm 2.6659)	25.5714(\pm 2.8975)
Land						
CVPR'09	27.758(\pm 1.653)	28.637(\pm 1.23)	25.6181(\pm 1.8376)	25.8012(\pm 1.8334)	22.584(\pm1.6859)	23.7501(\pm 1.7327)
Natural	22.9309(\pm 3.1456)	23.5219(\pm 2.6424)	17.8556(\pm 1.6498)	17.8501(\pm 1.693)	16.8485(\pm 1.3853)	16.632(\pm1.4767)
Images						

for all datasets. However, HHG based error values remained very high in comparison to both KTA and $CCGA_N$. This reason for this poor performance of HHG is the Euclidean distance based metric. As this Euclidean distance, which proves to be inefficient in encoding the relations between the data points accurately.

5 Conclusion

In this paper, we proposed Cross-Covariance based Graph Affinity between two data points that incorporates nonlinear dependence, the influence of neighboring data points, and the neighborhoods around data points. The first technique $CCGA_L$, tried to include the influence of the data points in the common neighborhood in the affinity metric between two points. The second technique $CCGA_N$, took into consideration the influence of the neighborhoods of data points that are adjacent to both the points. Both these techniques were able to determine the linear and nonlinear affinity between each data point. The classification accuracy and dimensionality reduction on both real world and synthetic datasets clearly suggested that the influence of adjacent points must be considered in pairwise affinity. The classification accuracy on different categories of datasets like Object detection, handwritten, and scene detection was found to be consistently better in comparison to other methods by a margin of $\approx 1\%$ to 5% under both LapSVM and LapRLSC. $CCGA_N$, has better classification accuracy and dimensionality reduction results that suggest that nonlinear dependence and neighborhoods have significant influence on affinity between data points.

References

- Zelnik-Manor L, Perona P (2005) Self-tuning spectral clustering. In: Advances in neural information processing systems, pp 1601–1608
- Bach FR, Jordan MI (2006) Learning spectral clustering, with application to speech separation. *J Mach Learn Res* 7:1963–2001
- Pavan M, Pelillo M (2006) Dominant sets and pairwise clustering. *IEEE transactions on pattern analysis and machine intelligence* 29(1):167–172
- Premachandran V, Kakarala R (2013) Consensus of k-nns for robust neighborhood selection on graph-based manifolds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1594–1601
- Nie F, Wang X, Huang H (2014) Clustering and projected clustering with adaptive neighbors. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 977–986
- Zhu X, Change Loy C, Gong S (2014) Constructing robust affinity graphs for spectral clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1450–1457
- Aggarwal CC (2007) On density based transforms for uncertain data mining. In: 2007 IEEE 23rd International Conference on Data Engineering, pp 866–875. IEEE
- Soleimani BH, Matwin S, De Souza EN (2015) A density-penalized distance measure for clustering. In: Canadian conference on artificial intelligence, pp 238–249. Springer
- Azizyan M, Singh A, Wasserman L et al (2013) Density-sensitive semisupervised inference. *The Annals of Statistics* 41(2):751–771
- Nicolau M, McDermott J et al (2018) Learning neural representations for network anomaly detection. *IEEE transactions on cybernetics* 49(8):3074–3087
- Terziyan V (2017) Social distance metric: from coordinates to neighborhoods. *Int J Geogr Inf Sci* 31(12):2401–2426
- Bai X, Yang X, Latecki LJ, Liu W, Tu Z (2009) Learning context-sensitive shape similarity by graph transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(5):861–874
- Zhang Y, Zhang Z, Qin J, Zhang L, Li B, Li F (2018) Semi-supervised local multi-manifold isomap by linear embedding for feature extraction. *Pattern Recogn* 76:662–678
- Yang X, Koknar-Tezel S, Latecki LJ (2009) Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In: 2009 IEEE conference on computer vision and pattern recognition, pp 357–364. IEEE
- Yang X, Prasad L, Latecki LJ (2012) Affinity learning with diffusion on tensor product graph. *IEEE transactions on pattern analysis and machine intelligence* 35(1):28–38
- Zhang Y, Zhang Z, Zhang Z, Zhao M, Zhang L, Zha Z, Wang M (2020) Deep self-representative concept factorization network for representation learning. In: Proceedings of the 2020 SIAM international conference on data mining, pp 361–369. SIAM
- Zhang Z, Li F, Zhao M, Zhang L, Yan S (2017) Robust neighborhood preserving projection by nuclear/1, 1-norm regularization for image feature extraction. *IEEE Trans Image Process* 26(4):1607–1622
- Zhang Z, Zhang Y, Liu G, Tang J, Yan S, Wang M (2019) Joint label prediction based semi-supervised adaptive concept factorization for robust data representation. *IEEE Trans Knowl Data Eng* 32(5):952–970
- Mukhoty B, Gupta R, Lakshmanan K, Kumar M (2020) A parameter-free affinity based clustering. *Appl Intell*, pp 1–14
- Jing P, Su Y, Li Z, Nie L (2020) Learning robust affinity graph representation for multi-view clustering. *Inf Sci* 544:155–167
- Li Q, Ren Y, Li L, Liu W (2016) Fuzzy based affinity learning for spectral clustering. *Pattern Recogn* 60:531–542
- Aronszajn N (1950) Theory of reproducing kernels. *Transactions of the American mathematical society* 68(3):337–404
- Song L, Smola A, Gretton A, Bedo J, Borgwardt K (2012) Feature selection via dependence maximization. *The Journal of Machine Learning Research* 13(1):1393–1434
- Fukumizu K, Bach FR, Jordan MI (2004) Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *J Mach Learn Res* 5(Jan):73–99
- Xiao M, Guo Y (2014) Feature space independent semi-supervised domain adaptation via kernel matching. *IEEE transactions on pattern analysis and machine intelligence* 37(1):54–66
- Masaeli M, Fung G, Dy JG (2010) From transformation-based dimensionality reduction to feature selection
- Bach FR, Jordan MI (2002) Kernel independent component analysis. *Journal of machine learning research* 3:1–48
- Chen J, Stern M, Wainwright MJ, Jordan MI (2017) Kernel feature selection via conditional covariance minimization. In: Advances in Neural Information Processing Systems, pp 6946–6955

29. Zheng X, Ma Z, Li L (2019) Local tangent space alignment based on hilbert–schmidt independence criterion regularization. *Pattern Anal Applic*, pp 1–14
30. Damodaran BB, Courty N, Lefèvre S (2017) Sparse hilbert schmidt independence criterion and surrogate-kernel-based feature selection for hyperspectral image classification. *IEEE Trans Geosci Remote Sens* 55(4):2385–2398
31. Chebotarev PY, Shamis EV (2005) On a duality between metrics and σ -proximities. *arXiv preprint math*0508183
32. Geng B, Tao D, Xu C, Yang L, Hua X-S (2012) Ensemble manifold regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(6):1227–1233
33. Zhang T, Yang J, Zhao D, Ge X (2007) Linear local tangent space alignment and application to face recognition. *Neurocomputing* 70(7-9):1547–1553
34. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *science* 290(5500):2323–2326
35. Silva VD, Tenenbaum JB (2003) Global versus local methods in nonlinear dimensionality reduction. In: *Advances in neural information processing systems*, pp 721–728
36. Belkin M, Niyogi P (2002) Laplacian eigenmaps and spectral techniques for embedding and clustering. In: *Advances in neural information processing systems*, pp 585–591
37. Donoho DL, Grimes C (2003) Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences* 100(10):5591–5596
38. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. *Chemometrics and intelligent laboratory systems* 2(1-3):37–52
39. Hull JJ (1994) A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence* 16(5):550–554
40. Thoma M (2017) The hasyv2 dataset. *preprint arXiv*:1701.08380
41. Kaya M, Binli MK, Ozbay E, Yanar H, Mishchenko Y (2018) A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces. *Scientific data* 5:180211
42. Krizhevsky A, Hinton G et al (2009) Learning multiple layers of features from tiny images. *Citeseer*
43. Yang Y, Newsam S (2010) Bag-of-visual-words and spatial extensions for land-use classification. In: *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pp 270–279. *ACM*
44. Quattoni A, Torralba A (2009) Recognizing indoor scenes. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp 413–420. *IEEE*
45. Roy P, Ghosh S, Bhattacharya S, Pal U (2018) Effects of degradations on deep neural network architectures. *arXiv preprint arXiv*:1807.10108
46. Zhao J, Zhang Y, He X, Xie P (2020) Covid-ct-dataset: a ct scan dataset about covid-19. *arXiv preprint arXiv*:2003.13865
47. Cristianini N, Shawe-Taylor J, Elisseeff A, Kandola JS (2002) On kernel-target alignment. In: *Advances in neural information processing systems*, pp 367–373

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Rakesh Kumar Yadav received the B.Tech. degree in Information Technology from the Cochin University of Science and Technology, Kochi, India, and the M. Tech. degree in Cyber Law and Information Security from the Indian Institute of Information Technology, Allahabad, India, where he is currently pursuing the Ph.D. degree. His research interest areas are Machine Learning, Deep learning, and Wireless Sensor Networks.



Abhishek received the B.E. degree in computer science from the University of Pune, India, and the M.Tech. degree in wireless communication and computing from the Indian Institute of Information Technology, Allahabad, India, where he is currently pursuing the Ph.D. degree. His research interest areas are machine learning, computer networks, wireless sensor networks, and operating system.



Shekhar Verma received the Ph.D. degree in computer networks from the Indian Institute of Technology, Varanasi, India. He is currently Professor (IT) with the Indian Institute of Information Technology, Allahabad, India. His research interest areas are Machine Learning, Computer Networks, Wireless Sensor Networks, Information and Networks security, and Cryptography.



S. Venkatesan received the Ph.D. degree in Mobile Agent Security from the Anna University, Chennai, Tamilnadu, India. He is currently Assistant Professor (IT) with the Indian Institute of Information Technology, Allahabad, India. His research interest areas are mobile agent security, cryptography, block chain, cloud computing and social network privacy.