THEORETICAL ADVANCES

A rotation based regularization method for semi-supervised learning

Prashant Shukla[1] · **Abhishek[1]** · **Shekhar Verma[1]** · **Manish Kumar[1]**

Abstract

In manifold learning, the intrinsic geometry of the manifold is explored and preserved by identifying the optimal local neighborhood around each observation. It is well known that when a Riemannian manifold is unfolded correctly, the observations lying spatially near to the manifold, should remain near on the lower dimension as well. Due to the nonlinear properties of manifold around each observation, finding such optimal neighborhood on the manifold is a challenge. Thus, a sub-optimal neighborhood may lead to erroneous representation and incorrect inferences. In this paper, we propose a rotation-based affinity metric for accurate graph Laplacian approximation. It exploits the property of aligned tangent spaces of observations in an optimal neighborhood to approximate correct affinity between them. Extensive experiments on both synthetic and real world datasets have been performed. It is observed that proposed method outperforms existing nonlinear dimensionality reduction techniques in low-dimensional representation for synthetic datasets. The results on real world datasets like COVID-19 prove that our approach increases the accuracy of classification by enhancing Laplacian regularization.

1 Introduction

A semi-supervised method utilizes unlabeled data for training along with given labeled data to exploit the hidden intrinsic geometrical information. It implicitly assumes that the underlying data holds either of the three assumptions: smoothness, clustering or manifold [27]. Manifold learning methods exploit the manifold assumption. These methods attempt to preserve geometric properties such as distances, proximity, angles, or local patches [22].

The real-world data gathered from imaging devices, medical science, and business applications usually lie in high dimension, and this causes the curse of dimensionality. One of the main objectives, in the analysis of such high-dimensional datasets, is to learn their geometrical and topological structure. Generally, the data is parameterized as points in $\mathbb{R}^D$, the correlation between parameters often suggests the manifold assumption that the data points are distributed on a very low-dimensional space $\mathbb{R}^m$ embedded on a Riemannian manifold $\mathbb{R}^D$ and $m \ll D$ [2, 5, 6, 30, 33]. Manifold learning algorithms transform the high-dimensional data into a low-dimensional embedding space using existing dimensionality reduction methods. Principal component analysis (PCA) [9, 35, 40, 47], multidimensional scaling (MDS) [13–15, 19], linear discriminant analysis (LDA) [3, 7, 20], etc., are some popular linear dimensionality reduction algorithms. They provide true representation in the case of linear manifold, but fail to discover nonlinear or curved structures of the input data. In the case of handwritten characters, spoken letters and medical images, etc., manifolds do not follow linear properties and have nonlinear structure. The intrinsic geometry of the nonlinear manifold is explored by identifying the optimal local neighborhood around each observation. We assume that the data samples $x_i \in X$ are drawn from a smooth Riemannian manifold $\mathcal{M} \in \mathbb{R}^D$. If a smooth Riemannian manifold is unfolded correctly, the observations

✉ Prashant Shukla
 rsi2016502@iiita.ac.in

 Abhishek
 rsi2016006@iiita.ac.in

 Shekhar Verma
 sverma@iiita.ac.in

 Manish Kumar
 manish@iiita.ac.in

[1] Department of Information Technology, Indian Institute of Information Technology Allahabad, Deoghat, Jhalwa, Allahabad, U.P. 211012, India

lying spatially near on the manifold, should remain near on the lower dimension as well with their tangent spaces aligned.

Generally, due to the varying curvature of manifold around each observation, finding such optimal neighborhood on the manifold, is a challenge. Thus, the affinity calculated between these observations on manifold are erroneous as it may be affected from noise. On such a manifold, the tangent planes on observations are not aligned.

Manifold learning approaches are suitable for exploiting the nonlinear structures into a flat low-dimensional embedding space [22]. The aim of these approaches is to identify and exploit local linear space. The existing state-of-the-art algorithms like isometric feature mapping (ISOMAP) [33], local linear embedding (LLE) [29], Laplacian eigenmap (LE) [4], local tangent space alignment (LTSA) [36, 45], Hilbert–Schmidt independence criterion-regularized LTSA (HSIC–LTSA) [46], graph-regularized linear discriminant analysis (GRLDA) [16], jerk-based manifold regularization [39], robust Laplacian [1] identify and exploit such local structures. These methods have been applied on wide variety of applications; for instance, face recognition, facial expression transferring, handwriting identification, 3D body pose recovery, medical imaging and many more. An approach for face recognition is a two-dimensional neighborhood preserving projection (2DNPP) [42].

These state-of-the-art manifold learning methods can be categorized into distance-preserving methods, angle-preserving methods and proximity-preserving methods, which align local neighborhood for each data point into a global coordinate space. A method focuses on one perspective in order to preserve a single geometric property. For instance, Isomap is a distance preserving method; LE, LLE and LTSA are proximity-preserving methods, which assume that unfolding manifold results into aligned tangent planes of all the neighboring observations in the manifold [44]. LTSA assumes that the given data is uniformly distributed and data in local neighborhood of the manifold, follows linear properties, i.e., they lie in or close to a linear sub-space.

In LE, diffusion map (DM) [23] and vector diffusion map (VDM) [31], the data is represented as a weighted undirected graph. The vertices of graph correspond to the data observations, and the weights on edges quantify the affinity between them. In a local linear manifold, Euclidean distance is used as affinity metric, which can be described as a kernel function of the distance. If the data $\{x_i\}_{i=1}^n$ consists of n observations in $L_2(\mathbb{R}^3)$, then the distance between points x_i and x_j are calculated using Eq. (1)

$$d_E(x_i, x_j) = ||x_i - x_j||_{L_2(\mathbb{R}^3)}, \tag{1}$$

and affinity of edges is calculated by Eq. (2)

$$w_{ij} = e^{-d_E^2(x_i, x_j)/2}. \tag{2}$$

To call an embedding faithful, we check whether it preserves the local structure of neighborhoods on the manifold, i.e., handles distances, angles, and neighborhoods in a comprehensive way.

LTSA assumes the local linearity of the manifold. Thus, local linear approximations of a manifold are constructed as a collection of overlapping approximate tangent spaces at every observation. These are, then, globally aligned to construct the global coordinate system for the underlying nonlinear manifold [45]. Here, the local tangent space is used to provide a low-dimensional linear approximation of the local geometric structure of the nonlinear manifold. The proposed rotation-based regularization method is based on the observation that the Riemannian assumption of local linearity of the manifold may not hold in a kNN neighborhood. Hence, a dimensionality reduction method, which relies on this assumption may yield suboptimal performance. This entails that the local neighborhood must be flattened so that the Euclidean distance is an accurate measure of affinity. Diffusion map assumes that the Euclidean distance between observations is approximated by the diffusion distance in the original feature space between probability distributions centered at those observations [23]. It assumes a nonlinear geometry and measures the similarity between two points at a specific scale through a diffusion metric.

In this paper, we determine the accurate pairwise affinity by aligning the tangent spaces of all the local points with respect to the point of interest to exploit the property of aligned tangent spaces of observations in an optimal neighborhood. Rotation is used to align the neighbors which are deviated from the tangent plane of the point of interest, so that they are on the same Euclidean plane. If the points are already on the plane, they are unaffected by the rotation. This gives an enhanced affinity for Laplacian, which is useful when data is affected from noise and the manifold curvatures are variable.

The contributions of this work are as follows:

1. In proposed approach, the Riemannian manifold assumption of local linearity of the kNN graph neighborhoods around data points is ensured by flattening the manifold by rotating the tangent spaces of the neighbors with the tangent space of the data point of interest. The pairwise Euclidean distance between data points, then, becomes an accurate measure of the geodesic distance between vertices.

2. The updated affinities based on the pairwise Euclidean distances are used in the graph Laplacian-based manifold regularization. This yields higher classification accuracy as the modified graph Laplacian, the rotation-

based Laplacian, is able to give a better estimate of underlying marginal distribution.

The remainder of this paper is organized as follows: Sect. 2 defines the problem to be solved in this work. In Sect. 3, we propose our rotation-based Laplacian regularization approach for manifold learning and regularization. Section 4 contains the results obtained using our method and its comparison with state-of-the-art methods. Finally, Sect. 5 concludes our work by highlighting the salient features of rotation-based regularization.

2 Problem definition

On a Riemannian manifold, the local linear neighborhood assumption allows Euclidean distance as a measure of affinity between neighboring data points. Due to the unknown properties of the manifold, the identified neighbors may not lie in the locally linear patch around the point of interest. As the extent of the locally linear patch is unknown, kNN or ϵ neighborhood is chosen heuristically as the linear neighborhood. Euclidean distance is computed and used as the affinity measure between data points in the kNN or ϵ neighborhood, which is assumed to be linear. In such cases, the Euclidean distance between data points in the neighborhood fails to represent the affinity between them accurately. This requires either accurate determination of the linear region, which is difficult; or, linearization of the kNN or ϵ neighborhood.

3 Rotation-based regularization method

Manifold regularization uses the smoothness assumption that a function, f, should change slowly where the marginal probability density is high. This requires estimation of the marginal probability density. In semi-supervised learning, the unknown marginal distribution is estimated using the given data, especially the ample amount of unlabeled data. If the data points on the manifold are represented by a graph, the smoothness of the function, f, on the graph can be measured in terms of a quadratic form of the graph Laplacian. Specifically, Graph Laplacian can be used to estimate the marginal distribution. The data points are the vertices of the graph. However, a distance needs to be associated between the corresponding vertices. This entails an accurate estimation of the geodesic distance between vertices. On the Riemannian manifold, the Euclidean distance is an accurate measure of the geodesic distance in a locally linear region. Thus, the problem of determining the geodesic distance between adjacent vertices of the graph reduces to finding the locally linear region. If a small region around a

data point is flattened, the Euclidean distance is an accurate measure of geodesic distance between the vertices. This leads to accurate estimation of the graph Laplacian and, through it, the underlying marginal distribution. In the proposed rotation-based regularization method, the kNN is linearized by rotating the tangent planes of data points in the neighborhood followed by the semi-supervised learning classification using the updated affinities computed between tangent space aligned data points.

3.1 Neighborhood linearization through rotation

The proposed linearization method endeavors to flatten the local neighborhood around a data point chosen through kNN data points. This is achieved by rotating the tangent planes of neighboring data points with respect to the tangent plane of the point under consideration. A local linear graph of the dataset is created by fixing the neighborhood of all the data points using kNN. The tangent plane of a point is found using local PCA. The tangent planes of all the k neighboring points are also found. Once the tangent planes are determined, the tangent plane of the point of interest is fixed and other misaligned tangent planes are rotated to align them with the fixed tangent plane of the point under consideration. This flattens the chosen neighborhood and Euclidean distance can be used as a measure of affinity between data points.

Given n data samples with l labeled and $(n - l)$ unlabeled points where $(n - l) \gg l$ on a smooth Riemannian manifold $\mathcal{M}$, i.e., $\{x_i\}_{i=1}^n \in \mathbb{R}^D$ that actually lie on a much lower-dimensional space $\in \mathbb{R}^m$ i.e., $m \ll D$. It can be represented by

$$f : C \subset \mathbb{R}^m \to \mathbb{R}^D, \tag{3}$$

where C is a compact subset of $\mathbb{R}^m$ and f is the data generation function, i.e.,

$$x_i = f(\tau_i) + \eta_i, \tag{4}$$

where τ_i are original feature vectors or the lower-dimensional complement information and η_i is redundant data or noise. The noise may be introduced during various stages of data collection and preprocessing and may vary with the distance.

A manifold can be approximated with a graph by using a smooth function defined on the graph. This depends on the affinity matrix W as

$$L = D - W, \tag{5}$$

where elements of the affinity matrix W are calculated using heat kernel $w_{ij} = \frac{1}{C} \exp - \frac{d_{ij}}{\epsilon^2}$, where d_{ij} is the distance between points x_i and x_j, and diagonal matrix D has entries as $D_{ii} = \sum_{j=1}^n w_{ij}$. kNN is used to create the undirected graph

over given data points including both labeled and unlabeled ones.

Proposition 1 *According to manifold learning assumption, on a manifold $\mathcal{M}$, the tangent planes of the points x_i and x_k lying on a locally linear region are aligned.*

Given a datapoint x_i and its neighbor $x_k \in N(x_i)$, the tangent planes of x_i and each x_k should be aligned.

$$T_{x_i} \equiv T_{x_k}. \tag{6}$$

To find the tangent plane T_{x_i} of the point x_i, local PCA is performed on the set of k nearest neighbors $N_k(x_i)$ of point x_i.

$$T_{x_i} = N_k(x_i) \cdot V, \tag{7}$$

where V is a weight matrix and T_{x_i} contains the principal component scores. The m leading eigenvectors correspond to an orthogonal basis of T_{x_i}.

$$T_{x_i} = N_k(x_i) \cdot V_m. \tag{8}$$

It is known that on a manifold, geodesic distance is the shortest distance between any two data points which is assumed to be Euclidean if the data points are lying in a local linear region. However, since the extent of the linear region around a data point is not known, the geodesic distance may not be Euclidean in a chosen neighborhood.

To find the correct Euclidean distance between points x_i and its neighborhood x_k, we rotate T_{x_k} w.r.t. T_{x_i} and align them.

$$T_{x_i} \equiv \gamma T_{x_k}, \tag{9}$$

where γ is the orthogonal rotation matrix calculated using Procrustes analysis [28],

$$\xi(\gamma, \phi, \rho) = \sum_{x_k \in N_k(x_i)} \| T_{x_i} - \rho \gamma T_{x_k} - \phi \|, \tag{10}$$

where ϕ denotes translation, γ denotes rotation, and ρ denotes scaling. In the ideal case of local linear neighborhood, ϕ will be a zero matrix, and γ and ρ will be unit matrices. But due to nonlinear surface, we optimize the parameters using

$$\{\overline{\gamma}, \overline{\phi}, \overline{\rho}\} = \xi(\gamma, \phi, \rho). \tag{11}$$

This idea is depicted in Fig. 1. To give the optimal rotation $\overline{\gamma}$, eigenvalue decomposition of same centroid matrices is calculated.

$$\overline{T}_{x_i} = T_{x_i} - \frac{1}{k} \sum_{x_j \in N_k(x_i)} T_{x_j}$$

$$\overline{T}_{x_k} = T_{x_k} - \frac{1}{k} \sum_{x_j \in N_k(x_k)} T_{x_j}, \tag{12}$$
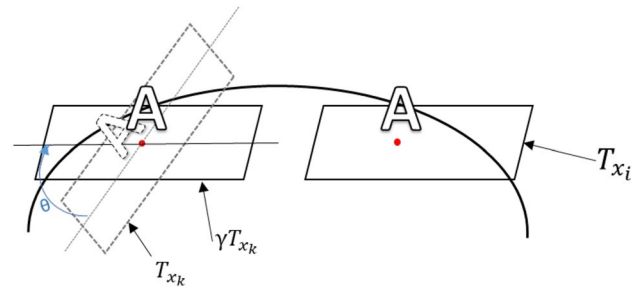

Fig. 1 Alignment of T_{x_k} w.r.t. T_{x_i} by rotation and finding the tangent plane γT_{x_k}

where k is the fixed number of neighbors. Putting these values in Eq. (10)

$$\begin{aligned}
\xi(\overline{\gamma}, \overline{\phi}, \overline{\rho}) &= \sum_{x_k \in N_k(x_i)} \| \overline{T}_{x_i} - \rho \gamma \overline{T}_{x_k} \|^2 \\
&= \sum_{x_k \in N_k(x_i)} \overline{T}_{x_i}^T \overline{T}_{x_i} + \overline{T}_{x_k}^T \overline{T}_{x_k} - 2\rho \gamma \overline{T}_{x_i}^T \overline{T}_{x_k}.
\end{aligned} \tag{13}$$

Let

$$P = \sum_{x_k \in N_k(x_i)} \overline{T}_{x_i}^T \overline{T}_{x_k}. \tag{14}$$

To minimize $\xi(\overline{\gamma}, \overline{\phi}, \overline{\rho})$, the term P is maximized; if its eigenvalue decomposition is given by $v\omega v^T$, the optimal rotation will be

$$\overline{\gamma} = \omega v^T. \tag{15}$$

Since finding the affinity between points x_i and x_k on tangent spaces is not same as affinity on manifold, we reconstruct the point x_k on original space using Eq. (16),

$$x_k' = T' \cdot V^{-1}, \tag{16}$$

where T' is the rotated tangent plane and V is the same weight matrix taken in Eq. (7).

$$T' = \overline{\gamma} T_{x_k}. \tag{17}$$

We calculate the Euclidean distance between data points x_i and x_k' as $d_{ik}' = \| x_i - x_k' \|_2^2$.

This distance d_{ik}' is used to calculate the revised affinity matrix using

$$w_{ik}' = \frac{1}{C} \exp{-\frac{d_{ik}'}{\epsilon^2}}. \tag{18}$$

The revised affinity enforces the function smoothening in semi-supervised learning by identifying affinity between neighboring data points accurately.

3.2 Laplacian regularized least squares classifier (LapRLSC)

Let l labeled data points are given as $\{x_i, y_i\}_{i=1}^l$ and $(n - l)$ unlabeled points are given as $\{x_u\}_{u=l+1}^n$. The prediction function is trained using the given labeled data points [6]

$$f^* = \arg\min \frac{1}{l} \sum_{i=1}^l \parallel y_i - f(x_i) \parallel^2 + \lambda_A \parallel f \parallel_A^2 + \lambda_I \parallel f \parallel_I^2, \tag{19}$$

where $\parallel f \parallel_A^2$ and $\parallel f \parallel_I^2$ are penalty terms in ambient space and intrinsic space, respectively. The unlabeled input data is used in prediction function by applying manifold assumption on the graph structure, considering the points in $\{x_u\}$ as nodes and the distances between them as weights. The intrinsic space regularization term $R(f)$ is calculated using [25]

$$R(f) = \frac{1}{2} \sum_{i=k=1}^n (f(x_i) - f(x_k))^2 w'_{ik}, \tag{20}$$

where w'_{ik} is calculated using Eq. (18). Expanding Eq. (20) using Eq. (5), we get

$$R(f) = \sum_{i=1}^n f(x_i)^2 \sum_{k=1}^n w'_{ik} - \sum_{i=k=1}^n w'_{ik} f(x_i) f(x_k) \tag{21}$$
$$= \mathbf{f}^T D \mathbf{f} - \mathbf{f}^T W \mathbf{f} = \mathbf{f}^T L \mathbf{f},$$

where $\mathbf{f}$ is $[f(x_1), f(x_2), \ldots, f(x_n)]^T$. After putting this value in Eq. (19), we get

$$f^* = \parallel \mathbf{Y}_l - \mathbf{f}_l \parallel^2 + \lambda_A \parallel f \parallel_A^2 + \lambda_I \mathbf{f}^T L \mathbf{f}, \tag{22}$$

where $\mathbf{Y}_l$ is vector of true labels of the labeled points. According to the classic Representer theorem [6],

$$f^* = \sum_{i=1}^n \alpha_i \mathcal{K}(x_i, x), \tag{23}$$

where α_i's are representation coefficients and $\mathcal{K}$ is a mercer kernel. According to this, Eq. (23) can be rewritten as

$$\mathbf{f} = \left[\sum_{i=1}^n \alpha_i \mathcal{K}(x_i, x_1), \sum_{i=1}^n \alpha_i \mathcal{K}(x_i, x_2), \ldots, \sum_{i=1}^n \alpha_i \mathcal{K}(x_i, x_n) \right]$$
$$= \mathbf{Ka}, \tag{24}$$

where $\mathbf{K}$ is kernel gram matrix and $\mathbf{a}$ is representation coefficient vector. According to kernel's property

$\mathcal{K}(x_i, x) = \langle \phi(x_i), \phi(x) \rangle = \phi(x_i)^T, \phi(x)$, where ϕ is kernel mapping, we can write the second term of Eq. (19) as

$$f = \left(\sum_{i=1}^n \alpha_i \phi(x_i)^T \right) = [\phi(x_1), \ldots, \phi(x_n)] \mathbf{a},$$

and

$$\parallel f \parallel_A^2 = \mathbf{a}^T \mathbf{Ka}. \tag{25}$$

Putting this values from Eqs. (24) and (25) in Eq. (22), we obtain

$$f(a)^* = \parallel \mathbf{Y}_l - \mathbf{K}_l \mathbf{a} \parallel^2 + \lambda_A \mathbf{a}^T \mathbf{Ka} + \lambda_I \mathbf{a}^T \mathbf{K} L \mathbf{Ka}^T. \tag{26}$$

The optimum solution, after setting the partial derivative $\frac{\partial f}{\partial \mathbf{a}} = 0$ is obtained as

$$a^* = \left(\mathbf{K}_l \mathbf{K}_l^T + \lambda_A \mathbf{K} + \lambda_I \mathbf{K} L \mathbf{K} \right)^{-1} \mathbf{K}_l \mathbf{Y}_l. \tag{27}$$

Algorithm 1: rotationLaplacian ($rLap$)

Input: X, k /* X is the data and k is number of neierest neighbors   */
Output: L_r /* $rLap$ Metric   */

1   Construct local linear graph using kNN
2   /* Rotate a point w.r.t. other points and calculate revised distances   */
3   **for** *each data point* x_i **do**
4      $[c1, s1] = \text{PCA}(N_k(x_i))$ /* c1 is coefficient and s1 is score   */
5   **for** *each* $x_k \in N_k(x_i)$ **do**
6   find nearest neighbors of x_k
7          $[c2, s2] = \text{PCA}(N_k(x_k))$
8          $tr = procrustes(s1, s2)$
9          $s2n = s2 * tr.T$ /* T is Orthogonal rotation and reflection component of transformation tr   */
10   recreate $s1, s2$
11   Calculate distance between x_i and x_k
12   **end**
13   **end**
14   Calculate revised weight matrix
15   Calculate revised Laplacian L_r

3.3 Complexity analysis

Complexity of *rLap* depends upon the number of data points n, their dimension d, number of nearest neighbors k, and procrustes analysis. We can calculate the complexity of *rLap* using following steps:

1. The kNN algorithm requires $O(nd + kn)$ time.[1]
2. As PCA takes $O(d^2n + d^3)$ time, and $d \ll n$, total time complexity would be $O(d^2n)$.
3. The upper bound for procrustes analysis is $O(d^3)$.

Thus, *rLap* is bounded by complexity $O(n(d^2n + k(d^2n + d^3)))$. As $n \gg k$ and d, we say that complexity of *rLap* is $O(n^2)$.

4 Experiments and results

In this section, the proposed rotation-based Laplacian regularization technique *rLap*[2] has been compared with existing state-of-the-art manifold learning and regularization methods on various real world and synthetic datasets. For data visualization, various 3D synthetic datasets have been projected on 2D. The performance is evaluated by comparing the intrinsic dimensional representations of all the methods. Further, real-world classification datasets have been used to train the RLSC model using all graph Laplacian variants. Their performance has been evaluated by calculating their root-mean-square error (RMSE) using Eq. (28)

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}, \qquad (28)$$

where $\hat{y}_i$ is predicted label.

4.1 Dimensionality reduction

In the following experiments, the proposed algorithm *rLap* has been compared with the existing state-of-the-art manifold learning approaches including Laplacian, DM, LTSA, entropy affinity *EA* [37], K_5 [32], K_7 [41] and min–max–mean (*MMM*) [43]. The algorithms have been applied on five synthetic datasets, namely swiss roll, swiss hole, punctured sphere, twin peaks, and elevated swiss roll, where the original 2D structure has been embedded in 3D space. The performance of proposed method on dimensionality reduction exhibits the extent to which the method is capable of preserving the local geometrical properties.

All datasets except twin peak dataset has 4000 data points. Twin peak data set consists of 7225 data points. Number of neighbors (k) is varied to find the best representation of data in lower-dimensional space. Table 1 shows the visualization results.

Swiss roll The swiss roll data is basically a 2D flat strip which is rotated as shown in the first column of Fig. 1 to make it a 3D structure. The dataset consists of 4000 points. Among all the methods, the results of *rLap* method gives the most accurate 2D representation. Laplacian gives a considerable representation as compared to other methods using $k = 6$. DM gives grossly inaccurate 2D representation exhibiting its incapability of preserving global connectivity and hence rotated strip could not be unfolded correctly.

Swiss hole The swiss hole dataset contains a hole in swiss roll dataset. The data set consists of 4000 points with a circular hole. Here, LTSA preserves the maximum intrinsic structure of the data. *rLap* method does not retain the shape of hole. However, *rLap* performs better than other methods, as they fail to preserve the shape of strip and give an inaccurate 2D representation with disconnected data points.

Elevated swiss roll The elevated swiss roll data is also a 2D flat strip similar to swiss roll with the varying third dimension. The 2D representation using *rLap* method gives a better result than all other methods. This proves that *rLap* holds properties similar to Laplacian, which can be used to exploit the intrinsic geometry of the data.

Punctured sphere The surface of a punctured sphere can be represented as a 2D flat surface. The best representation from *rLap* method comes at $k = 14$. *rLap* outperformed all the other methods except LTSA. The corners represented by *rLap* are smooth.

Twin peaks The twin peaks dataset containing 7225 points is originally a 2D flat surface with peaks at the two corners. *rLap* method gives comparable results to DM, which in turn remains the best performer.

It is evident from the results that LTSA and DM cannot preserve distances and angles due to their proximity-preserving nature. *EA* completely fails to unfold any dataset except punctured sphere. *rLap* method attempts to preserve distances and isometry. For punctured sphere dataset, flattening curved data into flat surface violates the distance preserving criterion, still our method gives comparable results.

4.2 Real-world datasets

The real-world classification datasets consist of different categories like image, sound, text and medical datasets. During experimental phase, for all the datasets, we used 'RBF' kernel of varying kernel width. The optimal results were found for kernel width $\sigma = 1$. The parameters defined for the manifold obtained, are λ_A and λ_I that correspond to the ambient geometry and intrinsic geometry. While regularization, value of λ_A was varied between 0.001 and 0.009 and λ_I was varied between 0.01 and 0.09. For the results calculated using other state-of-the-art methods, we fixed the value $\lambda_A = 0.005$ and $\lambda_I = 0.045$. For each class, the number of labeled examples is set to 2, selected randomly across 10

---

[1] http://www.csd.uwo.ca/courses/CS9840a/Lecture2_knn.pdf.

[2] Experimental codes are available at https://github.com/imprashantshukla/rLap.

Table 1 Dimensionality reduction: *rLap* versus other techniques

Dataset → Method ↓	Swiss Roll	Swiss Hole	Elevated Swiss Roll	Punctured Sphere	Twin Peak
kNN value	8	12	10	14	14
Original data Plot					
rLap					
Laplacian [6]					
Diffusion Map[22]					
LTSA[45]					
K_5[31]					
K_7[41]					
MMM[43]					
EA					

rounds of classification. *rLap* method is executed for RLSC and compared with Laplacian (*Lap*), p-Laplacian(*p-Lap*) [24], higher order Laplacian (L^m) [48], ensemble manifold regularization EMR-RLS-24G (*EMR*24) [11], *EA*, K_5, K_7 and *MMM* methods.

Isolet dataset Isolet dataset is a collection of English alphabet recorded in isolation with 150 people [10]. Each

letter of alphabet was spoken twice by each speaker. Speakers were divided into five groups having equal members, and each group outcome was termed as Isolet1 to Isolet5. Each Isolet set has 1560 samples. Each sample is represented by 617 features. Experiment has been repeated 10 times with every time a different set of labels are chosen. Isolet1 has been used for training purpose and Isolet5 for testing.
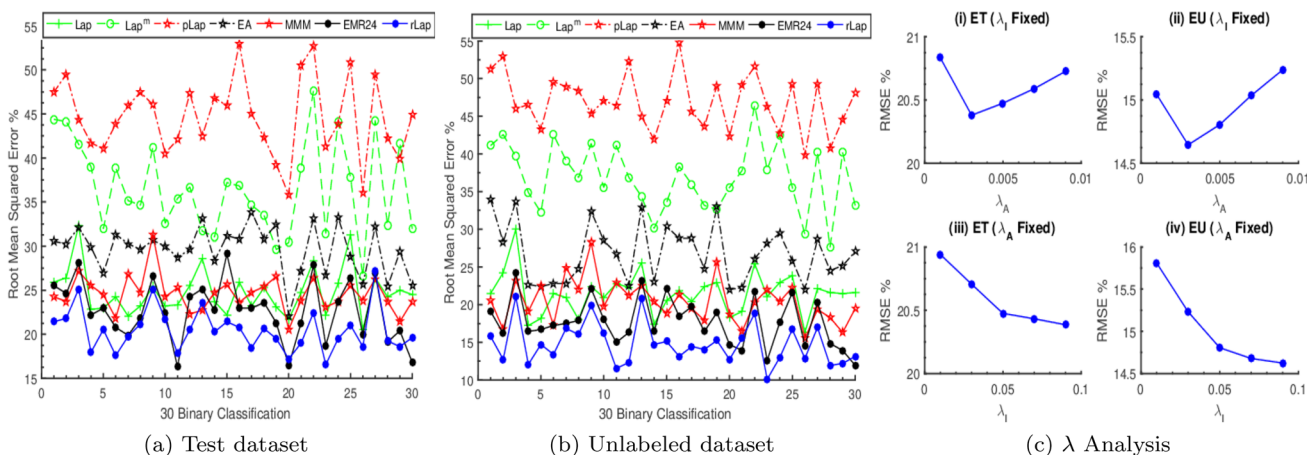
(a) Test dataset   (b) Unlabeled dataset   (c) λ Analysis

Fig. 2 ISOLET dataset

USPS handwritten digit dataset USPS dataset contains digits (0–9) digitized from handwritten zip codes [18]. Same digit has been sampled from different handwritings containing total 7291 samples. In the experiment, we have applied the concept of one-vs-all classification for predicting a digit. A total of 4000 images are used for training containing 400 instances per digit and remaining 3291 images for testing.

MNIST dataset MNIST dataset[3] consists of handwritten digits having a training set of 60,000 and test set of 10,000 examples. In this experiment, we have taken all the 70,000 images, out of which 4000 examples per digit have been used for training and rest 30,000 are used for testing.

HASYv2 dataset HASY dataset contains single symbols similar to MNIST. It contains 168,233 instances of 369 classes including Arabic numerals and Latin characters each of size 32×32. HASYv2 has much less samples per class [34]. The experiment consists of only nine symbols (classes) which have minimum 500 samples per class. Each symbol is pairwise classified from other one, hence having a 36 binary classification problem.

BCI dataset Brain–computer interface (BCI) is an electroencephalographic mental imagery dataset [17]. The dataset contains 60 hours of EEG BCI recordings spread across 75 experiments and 13 participants, featuring 60,000 mental imagery examples in four different BCI interaction paradigms with up to 6 EEG BCI interaction states. In our experiment we used HaLT interaction paradigm, which consists of the imageries of left and right hands movement, left and right leg movement and tongue movement, for a total of six mental states to be used for interaction with BCI. All readings of subjectA have been used for pairwise classification, hence having a 10 binary classification problem. A total of

1175 readings have been used for training the data and 1275 for testing.

COIL-20 dataset COIL-20 dataset consists of 1440 grayscale images of 20 different everyday objects of size 128×128 [26]. Forty-five instances per image, i.e., 900 images were taken for training. Remaining 27 instances per image, i.e., 540 images were taken for testing.

Cifar-10 dataset: Cifar-10 dataset contains 60,000 color images of 10 classes of size 32×32 pixels [21]. Every class consists of 6000 images. 50,000 images are given for training and 10,000 images for testing. In this experiment, only 6000 images are used for training, i.e., 600 images per class and total 4000 images are used for testing.

Fashion MNIST dataset Fashion MNIST [38] dataset consists of 70,000 images from 10 different fashion categories. All the images are in grayscale and of 28×28 pixel size. In the experiment, for each of the 10 rounds, total 42,469 data instances were selected randomly and rest were used for testing.

Lego brick dataset Lego bricks dataset contains total 16 categories of building bricks of different shapes, which are manufactured by Lego [12]. There are ≈ 400 images for each category. Images are in grayscale and size of each image is 200×200 pixels.

Figures 2, 3, 4, 5, 6, 7, 8, 9 and 10 show the mean square error (RMSE) results for test and unlabeled data of the datasets at $k = 6$ for RLSC.[4] The parametric analysis of λ_A and λ_I is also depicted in the figures.

While tuning λ_A for a fixed λ_I, it is observed that RMSE is increased gradually with increasing the value of λ_A for MNIST, BCI, Fashion MNIST, and Lego bricks datasets, where as RMSE is decreased for ISOLET and USPS

---

[4] ET and EU represent RMSE for test and unlabeled data respectively.

Table 2 Isolet test data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	24.91 ± 2.72	24.79 ± 2.65	24.82 ± 2.59	24.75 ± 2.59	24.75 ± 2.64
L^m	38.01 ± 5.43	37.18 ± 5.60	36.28 ± 5.34	36.09 ± 5.37	36.05 ± 5.44
p-Lap	45.55 ± 4.00	45.18 ± 4.26	44.35 ± 4.29	44.13 ± 4.60	44.02 ± 4.66
EA	30.07 ± 2.81	29.87 ± 2.56	29.66 ± 2.74	29.48 ± 2.62	29.42 ± 2.64
EMR24	22.75 ± 3.53	22.97 ± 3.41	22.69 ± 3.41	22.59 ± 3.46	22.53 ± 3.45
K_5	24.84 ± 1.72	24.71 ± 1.97	24.59 ± 1.89	24.52 ± 1.83	24.52 ± 1.83
K_7	24.73 ± 1.90	24.61 ± 1.96	24.56 ± 1.92	24.50 ± 1.92	24.47 ± 1.88
MMM	24.74 ± 2.10	24.61 ± 2.07	24.57 ± 2.04	24.51 ± 2.00	24.48 ± 1.97
rLap	$\mathbf{21.21 \pm 2.82}$	$\mathbf{20.58 \pm 2.83}$	$\mathbf{20.63 \pm 2.73}$	$\mathbf{19.96 \pm 2.60}$	$\mathbf{19.66 \pm 2.65}$

Minimum mean error has been highlighted in bold

Table 3 Isolet unlabeled data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	21.81 ± 2.71	21.52 ± 2.72	21.42 ± 2.78	21.34 ± 2.80	21.32 ± 2.83
L^m	38.67 ± 4.35	37.64 ± 4.50	36.50 ± 4.52	36.11 ± 4.55	35.84 ± 4.56
p-Lap	48.14 ± 3.26	47.52 ± 3.42	46.45 ± 3.75	45.94 ± 3.84	45.68 ± 3.94
EA	27.31 ± 3.94	27.02 ± 3.71	26.82 ± 3.83	26.53 ± 3.64	26.42 ± 3.49
EMR24	17.97 ± 3.37	17.76 ± 3.26	17.67 ± 3.17	17.63 ± 3.12	17.62 ± 3.12
K_5	21.01 ± 2.83	20.94 ± 2.89	20.52 ± 2.59	20.37 ± 2.36	20.01 ± 2.48
K_7	21.24 ± 2.61	20.99 ± 2.57	20.57 ± 2.52	20.32 ± 2.48	20.27 ± 2.48
MMM	20.79 ± 2.99	20.63 ± 2.82	20.46 ± 2.90	20.21 ± 2.71	19.94 ± 2.59
rLap	$\mathbf{16.62 \pm 2.76}$	$\mathbf{15.36 \pm 2.73}$	$\mathbf{14.38 \pm 2.73}$	$\mathbf{14.11 \pm 2.55}$	$\mathbf{14.01 \pm 2.77}$

Minimum mean error has been highlighted in bold
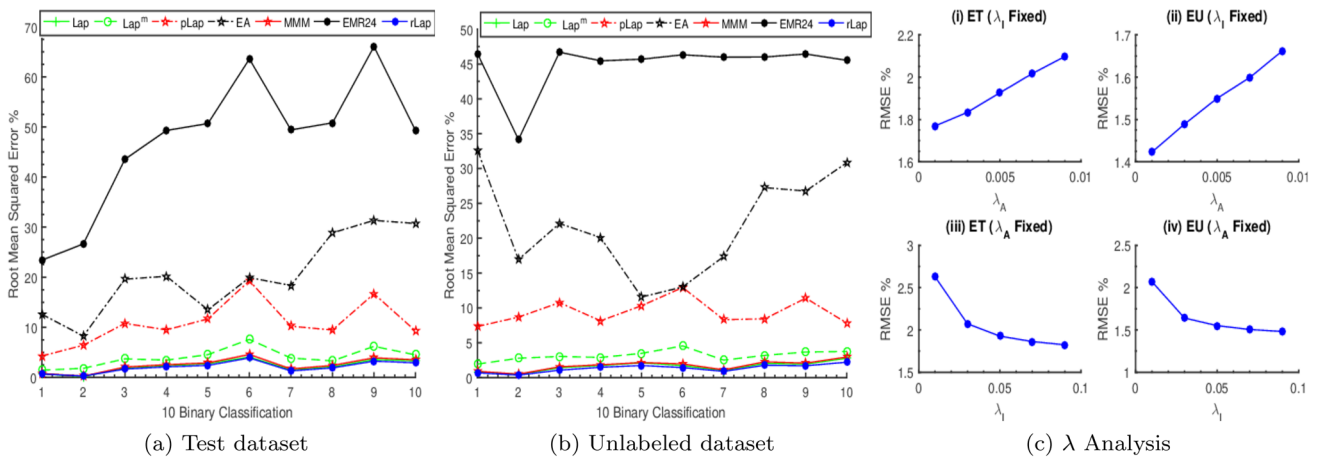
Fig. 3 USPS dataset

datasets. Change in λ_A does not affect RMSE for HASY dataset and for COIL20 dataset also RMSE does not change after $\lambda_A = 0.005$. While tuning λ_I, RMSE is increased for all the datasets except ISOLET and USPS datasets.

Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 and 19 contain the RMSE values of all the methods compared with *rLap* for $k = 6$ to $k = 20$. For the spoken letter dataset Isolet, *rLap* outperforms other methods for all the

NN values. It gives $\approx 80.34\%$ and $\approx 85.99\%$ accuracy for test and unlabeled data respectively.

For the handwritten datasets USPS, MNIST and HASYv2, test data accuracy is $\approx 97.98\%$, $\approx 88.58\%$ and $\approx 96.82\%$ respectively; unlabeled data accuracy is $\approx 98.63\%$, $\approx 98.07\%$ and $\approx 96.94\%$ respectively. In case of USPS dataset *rLap* performs better than the rest of the methods for $k = 6$ to $k = 12$, beyond that it is dominated by *MMM* and K_7

Table 4 USPS test data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	2.23 ± 1.24	2.26 ± 1.25	2.62 ± 1.46	2.83 ± 1.64	3.19 ± 1.81
L^m	4.03 ± 1.84	3.52 ± 1.82	3.01 ± 1.37	2.54 ± 1.38	2.41 ± 1.38
p-Lap	10.74 ± 4.41	9.63 ± 4.15	9.38 ± 4.06	8.92 ± 4.12	8.39 ± 4.04
EA	20.17 ± 7.68	20.32 ± 7.88	20.28 ± 7.91	20.48 ± 7.73	20.65 ± 7.97
EMR24	46.91 ± 13.92	47.26 ± 13.59	47.61 ± 14.04	47.92 ± 13.94	48.03 ± 13.79
K_5	2.98 ± 1.61	2.76 ± 1.53	2.69 ± 1.67	2.61 ± 1.58	2.51 ± 1.59
K_7	2.74 ± 1.48	2.46 ± 1.36	2.37 ± 1.30	2.28 ± 1.25	2.21 ± 1.29
MMM	2.71 ± 1.51	2.45 ± 1.34	2.30 ± 1.36	$\mathbf{2.21 \pm 1.30}$	$\mathbf{2.19 \pm 1.29}$
rLap	$\mathbf{2.13 \pm 1.13}$	$\mathbf{2.02 \pm 1.18}$	$\mathbf{2.17 \pm 1.41}$	2.30 ± 1.37	2.58 ± 1.52

Minimum mean error has been highlighted in bold

Table 5 USPS unlabeled data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	1.57 ± 0.72	1.61 ± 0.69	1.72 ± 0.78	1.96 ± 0.90	2.17 ± 1.03
L^m	3.18 ± 0.72	2.93 ± 0.74	2.79 ± 0.82	2.66 ± 0.82	2.53 ± 0.86
p-Lap	9.41 ± 1.80	8.68 ± 1.89	8.97 ± 2.04	9.13 ± 2.09	8.92 ± 2.16
EA	21.72 ± 7.49	21.86 ± 7.32	21.69 ± 7.17	21.80 ± 7.39	21.84 ± 7.38
EMR24	44.61 ± 3.52	44.87 ± 3.78	45.12 ± 3.83	45.04 ± 3.83	45.23 ± 3.96
K_5	1.93 ± 0.78	1.88 ± 0.76	1.79 ± 0.79	1.64 ± 0.79	1.50 ± 0.81
K_7	1.82 ± 0.71	1.71 ± 0.78	1.65 ± 0.78	1.58 ± 0.77	$\mathbf{1.42 \pm 0.73}$
MMM	1.83 ± 0.70	1.73 ± 0.72	1.67 ± 0.76	$\mathbf{1.56 \pm 0.79}$	1.49 ± 0.73
rLap	$\mathbf{1.27 \pm 0.58}$	$\mathbf{1.33 \pm 0.56}$	$\mathbf{1.58 \pm 0.64}$	1.64 ± 0.75	1.87 ± 0.87

Minimum mean error has been highlighted in bold

(a) Test dataset   (b) Unlabeled dataset   (c) λ Analysis
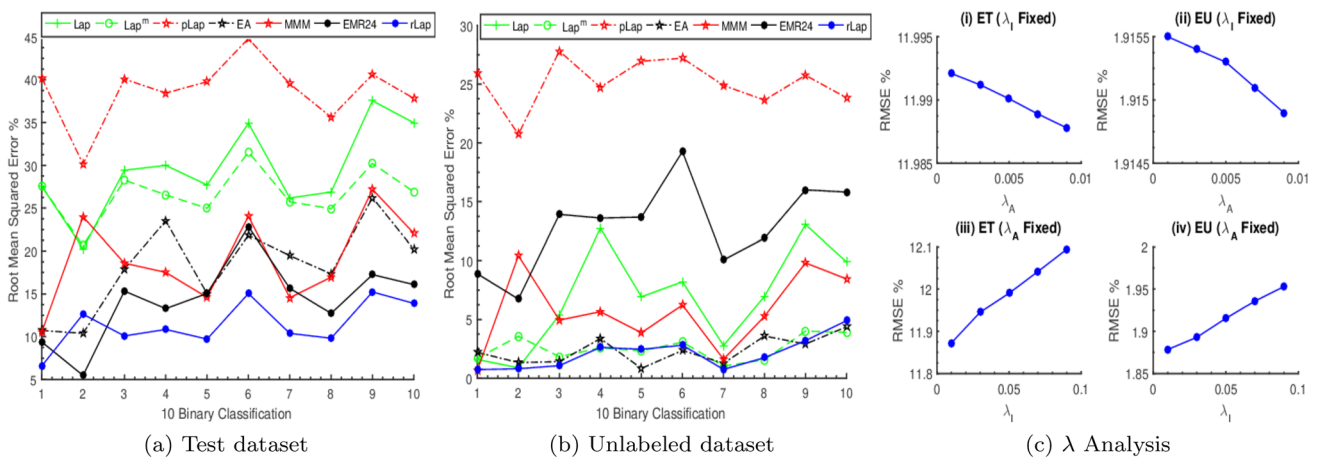
Fig. 4 MNIST dataset

methods. For MNIST dataset, *EA* outperforms *rLap* beyond NN = 12.

For BCI test data *rLap* performs better than the rest of the methods at $k = 6$ and $k = 12$. For unlabeled data *EMR*24 outperforms other methods. For both test and unlabeled datasets, the classification accuracy is ≈ 75%.

For object detection datasets Fashion MNIST and Lego brick, *rLap* out performs all the other methods. The achieved accuracy for test data is ≈ 93.21% and ≈ 85.52% respectively, whereas for unlabeled data it is ≈ 94.07% and ≈ 93.75%,, respectively. For the other two object detection datasets COIL 20 and Cifar, test data accuracies are ≈ 98.73% and

Table 6 MNIST test data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	28.47 ± 3.93	28.31 ± 4.10	27.94 ± 4.46	27.76 ± 4.68	27.78 ± 4.74
L^m	26.58 ± 3.08	26.23 ± 3.16	25.72 ± 3.39	25.43 ± 3.56	25.34 ± 3.58
p-Lap	38.50 ± 3.79	37.91 ± 3.89	37.15 ± 4.08	36.64 ± 4.26	36.40 ± 4.29
EA	18.26 ± 5.15	18.53 ± 5.28	18.17 ± 5.04	17.96 ± 5.23	17.60 ± 5.01
EMR24	14.31 ± 4.63	14.56 ± 4.72	14.74 ± 4.66	14.90 ± 4.73	15.03 ± 4.78
K_5	23.70 ± 5.81	23.41 ± 5.57	23.05 ± 5.82	22.93 ± 5.53	22.74 ± 5.18
K_7	20.59 ± 5.69	20.09 ± 5.78	19.83 ± 5.81	19.74 ± 5.46	19.58 ± 5.72
MMM	19.00 ± 5.21	18.71 ± 5.15	18.99 ± 5.19	18.61 ± 5.28	18.37 ± 5.03
rLap	$\mathbf{11.42 \pm 2.62}$	$\mathbf{11.51 \pm 2.66}$	$\mathbf{11.88 \pm 3.04}$	$\mathbf{12.68 \pm 3.34}$	$\mathbf{13.32 \pm 3.53}$

Minimum mean error has been highlighted in bold

Table 7 MNIST unlabeled data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	6.41 ± 4.26	6.90 ± 4.57	7.38 ± 4.81	7.71 ± 4.94	8.00 ± 5.05
L^m	2.37 ± 1.15	2.27 ± 1.15	2.16 ± 1.17	2.13 ± 1.22	2.12 ± 1.27
p-Lap	24.84 ± 2.23	23.68 ± 2.59	22.76 ± 3.07	22.32 ± 3.36	22.19 ± 3.36
EA	2.36 ± 1.17	2.57 ± 1.27	$\mathbf{2.30 \pm 1.23}$	$\mathbf{2.17 \pm 1.18}$	$\mathbf{2.06 \pm 1.09}$
EMR24	12.98 ± 3.71	13.14 ± 3.65	13.29 ± 3.78	13.64 ± 3.58	13.89 ± 3.67
K_5	10.57 ± 6.80	10.27 ± 6.28	10.03 ± 6.37	9.68 ± 6.49	9.31 ± 6.16
K_7	7.05 ± 4.67	6.89 ± 4.48	6.73 ± 4.59	6.38 ± 4.36	6.15 ± 4.10
MMM	5.96 ± 3.21	5.61 ± 3.42	5.88 ± 3.47	5.74 ± 3.53	5.46 ± 3.57
rLap	$\mathbf{1.93 \pm 1.44}$	$\mathbf{2.02 \pm 1.50}$	2.58 ± 1.87	2.93 ± 2.11	3.31 ± 2.36

Minimum mean error has been highlighted in bold

(a) Test dataset (b) Unlabeled dataset (c) λ Analysis
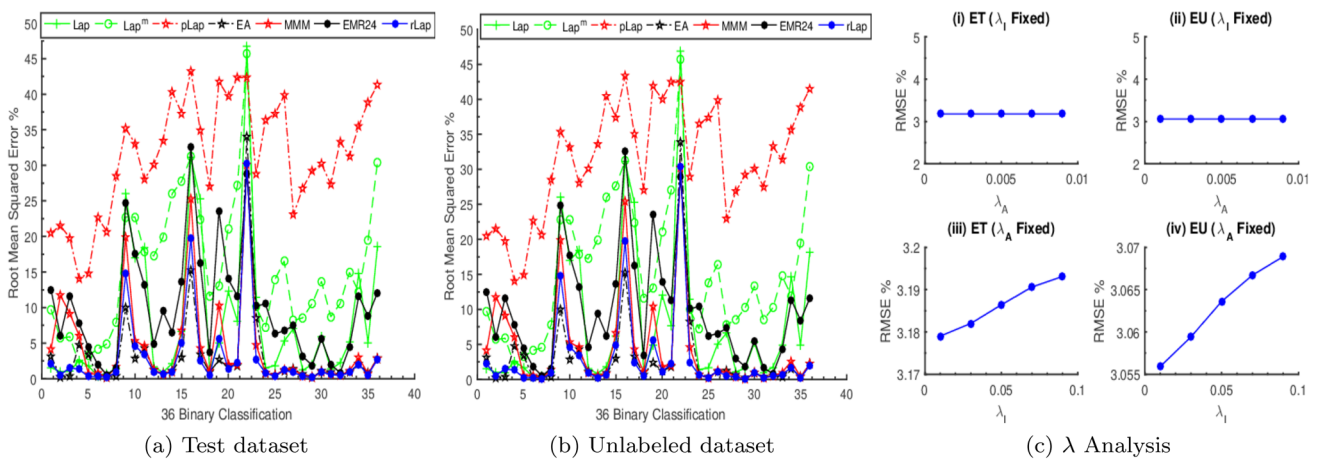
Fig. 5 HASY dataset

$\approx 75.43\%$ and unlabled data accuracies are $\approx 98.79\%$ and $\approx 75.71\%$. The results for COIL20 test data show that *rLap* has minimum RMSE for $k = 6$ to $k = 16$. For unlabeled data *Lap* performs better than *rLap* beyond NN = 12. In case of Cifar unlabeled data, *rLap* performs better, where as for test data *p-Lap* gives the best results from $k = 12$ onwards, still *rLap* method gave comparable results with *p-Lap*.

Table 8 HASY test data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	8.25 ± 10.47	8.28 ± 10.52	8.39 ± 10.52	8.40 ± 10.54	8.40 ± 10.54
L^m	15.44 ± 9.52	15.37 ± 9.54	15.27 ± 9.53	15.25 ± 9.52	15.24 ± 9.52
p-Lap	31.39 ± 8.13	31.34 ± 8.06	31.35 ± 8.05	31.14 ± 8.21	31.08 ± 8.37
EA	3.27 ± 6.10	3.33 ± 6.06	$\mathbf{3.41 \pm 6.20}$	$\mathbf{3.49 \pm 6.24}$	$\mathbf{3.56 \pm 6.28}$
EMR24	9.96 ± 7.78	10.27 ± 7.91	10.34 ± 7.83	10.43 ± 7.92	10.48 ± 7.98
K_5	7.41 ± 8.76	7.52 ± 8.19	7.68 ± 8.53	7.79 ± 8.81	7.95 ± 8.41
K_7	5.48 ± 7.53	5.63 ± 7.24	5.77 ± 7.34	5.85 ± 7.67	5.92 ± 7.81
MMM	4.67 ± 7.01	4.79 ± 7.27	4.85 ± 7.13	4.97 ± 7.52	5.03 ± 7.68
rLap	$\mathbf{3.18 \pm 6.09}$	$\mathbf{3.25 \pm 6.06}$	3.47 ± 6.29	3.68 ± 6.51	3.84 ± 6.70

Minimum mean error has been highlighted in bold

Table 9 HASY unlabeled data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	8.08 ± 10.53	8.11 ± 10.59	8.22 ± 10.59	8.23 ± 10.60	8.23 ± 10.60
L^m	15.34 ± 9.57	15.28 ± 9.60	15.18 ± 9.59	15.15 ± 9.59	15.14 ± 9.58
p-Lap	31.46 ± 8.18	31.40 ± 8.11	31.42 ± 8.10	31.21 ± 8.26	31.15 ± 8.41
EA	3.12 ± 6.13	3.27 ± 6.04	3.41 ± 6.18	3.58 ± 6.23	3.79 ± 6.31
EMR24	9.85 ± 7.83	9.96 ± 7.92	10.13 ± 7.84	10.39 ± 7.98	10.58 ± 8.06
K_5	7.27 ± 8.87	7.41 ± 8.80	7.56 ± 8.92	7.70 ± 8.95	7.86 ± 8.79
K_7	5.35 ± 7.62	5.46 ± 7.67	5.63 ± 7.74	5.81 ± 7.71	5.89 ± 7.82
MMM	4.53 ± 7.08	4.59 ± 7.01	4.74 ± 7.14	4.85 ± 7.17	4.90 ± 7.18
rLap	$\mathbf{3.06 \pm 6.15}$	$\mathbf{3.12 \pm 6.10}$	$\mathbf{3.33 \pm 6.34}$	$\mathbf{3.54 \pm 6.56}$	$\mathbf{3.70 \pm 6.75}$

Minimum mean error has been highlighted in bold

(a) Test dataset   (b) Unlabeled dataset   (c) λ Analysis
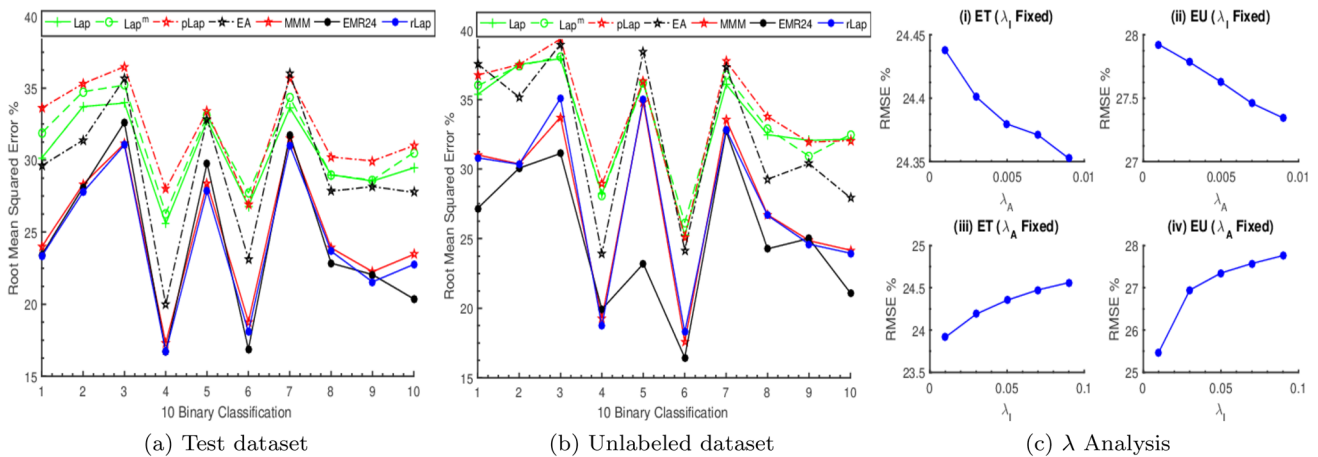
Fig. 6 BCI dataset

4.3 Medical image datasets

In case of medical image classification, we find limited number of labeled datasets. So it is important for a model to classify these medical images accurately using semi-supervised approach. In our experiment, we have considered five benchmark medical image datasets for binary classification.

Mammography images dataset This dataset is taken from Mammographic Image Analysis Society (Mini-MIAS)[5] to predict breast cancer. The dataset consisted of 322 mammography images having 1024×1024 dimensions each. The

---

[5] http://peipa.essex.ac.uk/info/mias.html.

Table 10 BCI test data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	30.40 ± 2.97	30.36 ± 3.02	30.35 ± 3.02	30.36 ± 3.02	30.36 ± 3.02
L^m	31.17 ± 3.19	31.12 ± 3.19	31.13 ± 3.17	31.13 ± 3.18	31.12 ± 3.19
p-Lap	32.08 ± 3.35	32.04 ± 3.33	32.04 ± 3.32	32.05 ± 3.32	32.04 ± 3.33
EA	29.43 ± 5.01	29.09 ± 5.11	29.23 ± 5.08	29.34 ± 5.12	29.41 ± 5.17
EMR24	24.41 ± 5.87	$\mathbf{24.45 \pm 5.82}$	24.51 ± 5.92	24.55 ± 5.89	24.78 ± 5.97
K_5	24.58 ± 4.91	24.57 ± 4.99	24.48 ± 4.97	24.44 ± 4.91	24.44 ± 4.91
K_7	24.57 ± 4.93	24.52 ± 4.87	24.48 ± 4.99	$\mathbf{24.43 \pm 4.95}$	$\mathbf{24.39 \pm 4.93}$
MMM	24.97 ± 4.89	24.93 ± 4.82	24.90 ± 4.85	24.89 ± 4.84	24.89 ± 4.84
rLap	$\mathbf{24.38 \pm 5.06}$	24.82 ± 4.98	$\mathbf{24.37 \pm 4.97}$	24.68 ± 4.91	24.82 ± 4.98

Minimum mean error has been highlighted in bold

Table 11 BCI unlabeled data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	33.35 ± 4.07	33.33 ± 4.14	33.32 ± 4.15	33.33 ± 4.15	33.33 ± 4.14
L^m	33.58 ± 4.02	33.47 ± 4.11	33.49 ± 4.10	33.47 ± 4.10	33.47 ± 4.11
p-Lap	34.14 ± 4.48	33.89 ± 4.45	33.94 ± 4.49	33.92 ± 4.46	33.89 ± 4.45
EA	33.07 ± 5.95	31.31 ± 5.67	32.31 ± 5.88	32.73 ± 5.71	33.05 ± 5.90
EMR24	$\mathbf{25.10 \pm 5.21}$	$\mathbf{25.38 \pm 5.27}$	$\mathbf{25.57 \pm 5.31}$	$\mathbf{25.78 \pm 5.34}$	26.15 ± 5.38
K_5	27.71 ± 6.30	27.70 ± 6.30	27.69 ± 6.28	27.67 ± 6.32	27.67 ± 6.32
K_7	27.68 ± 6.29	27.66 ± 6.27	27.64 ± 6.24	27.59 ± 6.22	27.59 ± 6.22
MMM	27.65 ± 6.02	27.62 ± 6.09	27.61 ± 6.09	27.56 ± 6.10	27.59 ± 6.11
rLap	31.38 ± 5.56	25.58 ± 6.47	27.62 ± 6.18	26.56 ± 6.32	$\mathbf{25.58 \pm 6.47}$

Minimum mean error has been highlighted in bold
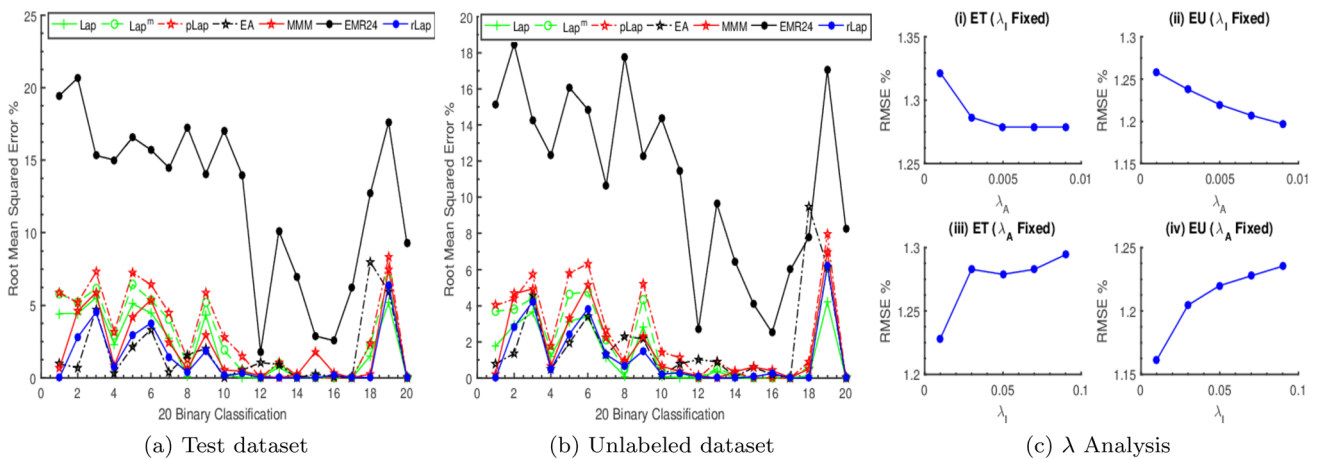
Fig. 7 Coil20 dataset

dataset was divided into 200 and 122 images for training and testing respectively.

Diabetic retinopathy images This dataset is taken from Indian Diabetic Retinopathy Image Dataset (IDRiD) Web site.[6] This abnormality of eyes affects the retina of patients by increasing the amount of insulin in their blood. In this experiment, a subset containing 516 images was used, where each image is of resolution 4288×2848 pixels. The training set consisted of 344 images and remaining 172 images were used for testing.

[6] https://idrid.grand-challenge.org/.

Table 12 Coil20 test data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	2.06 ± 2.14	2.06 ± 2.14	2.06 ± 2.16	2.04 ± 2.16	2.04 ± 2.16
L^m	2.72 ± 2.47	2.72 ± 2.47	2.72 ± 2.47	2.72 ± 2.47	2.72 ± 2.47
p-Lap	3.15 ± 2.92	3.15 ± 2.92	3.15 ± 2.92	3.14 ± 2.93	3.14 ± 2.93
EA	1.65 ± 2.21	1.71 ± 2.24	1.78 ± 2.19	1.85 ± 2.26	$\mathbf{1.94 \pm 2.23}$
EMR24	12.47 ± 5.74	12.68 ± 5.77	12.97 ± 5.81	13.08 ± 5.89	13.21 ± 5.94
K_5	2.55 ± 2.71	2.55 ± 2.71	2.55 ± 2.71	2.54 ± 2.70	2.54 ± 2.70
K_7	2.00 ± 2.36	2.00 ± 2.36	1.99 ± 2.34	1.99 ± 2.34	1.99 ± 2.34
MMM	1.94 ± 2.32	1.94 ± 2.32	1.92 ± 2.31	1.92 ± 2.31	1.92 ± 2.31
rLap	$\mathbf{1.27 \pm 1.85}$	$\mathbf{1.38 \pm 1.93}$	$\mathbf{1.62 \pm 2.18}$	$\mathbf{1.85 \pm 2.36}$	1.97 ± 2.51

Minimum mean error has been highlighted in bold

Table 13 Coil20 unlabeled data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	1.45 ± 1.51	1.45 ± 1.51	$\mathbf{1.45 \pm 1.51}$	$\mathbf{1.45 \pm 1.51}$	$\mathbf{1.45 \pm 1.51}$
L^m	1.89 ± 2.10	1.89 ± 2.10	1.89 ± 2.10	1.88 ± 2.09	1.88 ± 2.09
p-Lap	2.48 ± 2.57	2.48 ± 2.57	2.48 ± 2.57	2.47 ± 2.57	2.47 ± 2.57
EA	1.88 ± 2.41	1.92 ± 2.36	1.95 ± 2.33	1.99 ± 2.21	2.07 ± 2.32
EMR24	11.10 ± 5.00	11.34 ± 5.11	11.51 ± 4.96	11.89 ± 5.18	12.09 ± 5.23
K_5	1.72 ± 2.18	1.72 ± 2.18	1.72 ± 2.18	1.72 ± 2.18	1.71 ± 2.18
K_7	1.61 ± 2.12	1.61 ± 2.12	1.60 ± 2.10	1.60 ± 2.10	1.60 ± 2.10
MMM	1.71 ± 2.13	1.71 ± 2.13	1.71 ± 2.13	1.69 ± 2.11	1.69 ± 2.11
rLap	$\mathbf{1.21 \pm 1.78}$	$\mathbf{1.45 \pm 2.03}$	1.71 ± 2.59	2.01 ± 2.74	2.23 ± 2.98

Minimum mean error has been highlighted in bold

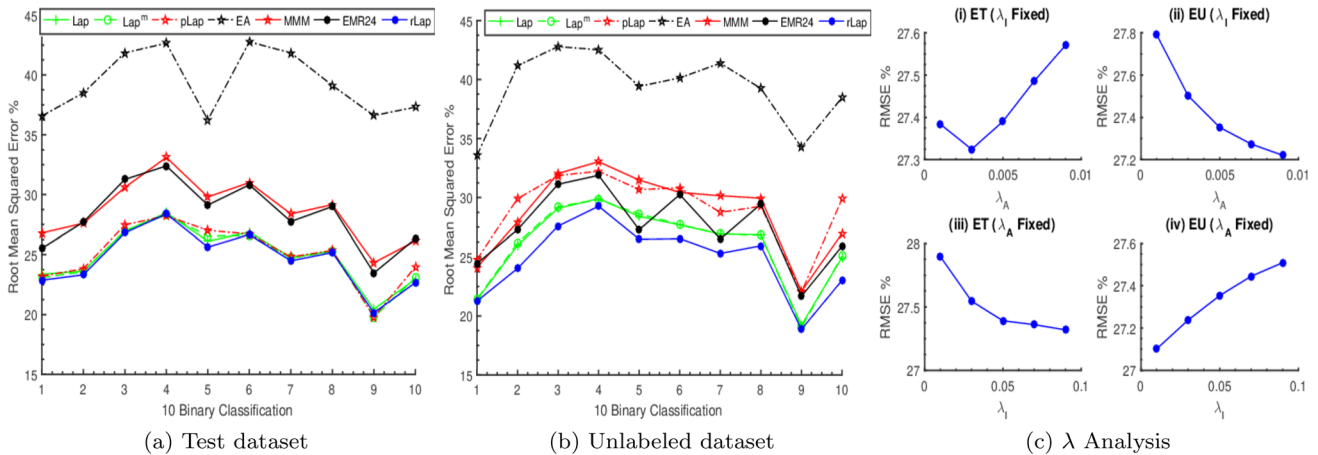(a) Test dataset (b) Unlabeled dataset (c) λ Analysis

Fig. 8 Cifar dataset

Chest X-ray images This dataset contains 338 chest X-ray images,[7] used to classify data for COVID-19. Though the dataset contains the images for SARS (Severe acute respiratory syndrome), ARDS (acute respiratory distress syndrome) and other classes [8], etc; we have applied the concept of one-vs-all classification for predicting COVID-19 only. Out of 422 X-ray and CT Scan images of 216 patients, we have taken only X-ray images of 194 patients, containing 272 COVID-19 positive images.

[7] https://github.com/ieee8023/covid-chestxray-dataset.

Table 14 Cifar test data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	24.89 ± 2.35	24.96 ± 2.33	25.03 ± 2.32	25.08 ± 2.28	25.13 ± 2.28
L^m	24.76 ± 2.52	24.74 ± 2.47	24.83 ± 2.48	24.86 ± 2.42	24.91 ± 2.41
p-Lap	25.00 ± 2.55	24.84 ± 2.56	$\mathbf{24.65 \pm 2.60}$	$\mathbf{24.58 \pm 2.64}$	$\mathbf{24.52 \pm 2.63}$
EA	39.34 ± 2.68	39.51 ± 2.59	39.64 ± 2.63	39.78 ± 2.74	39.91 ± 2.79
EMR24	28.33 ± 2.76	28.49 ± 2.50	28.58 ± 2.63	28.71 ± 2.72	28.80 ± 2.76
K_5	31.20 ± 3.05	31.48 ± 3.26	31.62 ± 3.01	31.84 ± 3.29	31.97 ± 3.37
K_7	30.87 ± 2.96	31.08 ± 3.02	31.21 ± 3.10	31.39 ± 2.99	31.48 ± 3.05
MMM	28.69 ± 2.58	28.73 ± 2.38	28.77 ± 2.40	28.80 ± 2.59	28.92 ± 2.64
rLap	$\mathbf{24.57 \pm 2.41}$	$\mathbf{24.63 \pm 2.36}$	24.73 ± 2.33	24.83 ± 2.32	24.90 ± 2.30

Minimum mean error has been highlighted in bold

Table 15 Cifar unlabeled data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	25.97 ± 3.33	25.81 ± 3.24	25.73 ± 3.24	25.73 ± 3.23	25.78 ± 3.19
L^m	26.16 ± 3.42	25.78 ± 3.38	25.53 ± 3.38	25.45 ± 3.39	25.50 ± 3.42
p-Lap	28.91 ± 3.20	28.08 ± 3.16	27.16 ± 3.20	26.50 ± 3.32	26.16 ± 3.39
EA	39.29 ± 3.16	39.21 ± 3.08	39.07 ± 3.29	38.93 ± 3.12	38.85 ± 3.25
EMR24	27.57 ± 3.17	27.49 ± 3.31	27.28 ± 3.27	27.09 ± 3.29	27.04 ± 3.29
K_5	29.68 ± 3.82	29.83 ± 3.60	29.97 ± 3.46	29.94 ± 3.51	29.96 ± 3.51
K_7	29.59 ± 3.79	29.68 ± 3.83	29.82 ± 3.39	29.89 ± 3.65	29.91 ± 3.69
MMM	28.81 ± 3.51	28.97 ± 3.58	28.76 ± 3.47	28.61 ± 3.63	28.61 ± 3.79
rLap	$\mathbf{24.74 \pm 3.13}$	$\mathbf{24.48 \pm 3.08}$	$\mathbf{24.33 \pm 3.04}$	$\mathbf{24.29 \pm 3.01}$	$\mathbf{24.31 \pm 2.98}$

Minimum mean error has been highlighted in bold

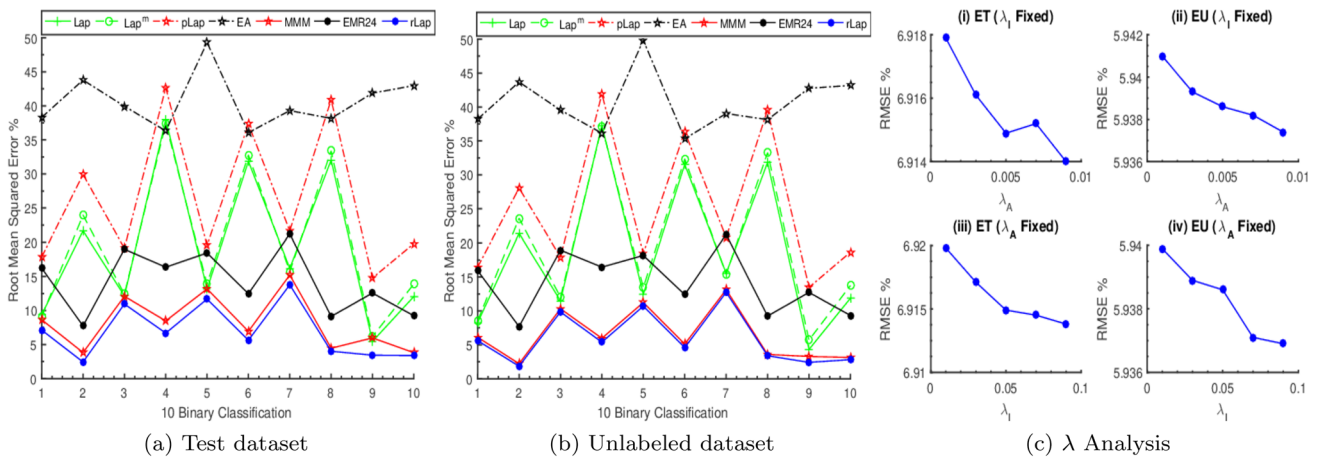(a) Test dataset (b) Unlabeled dataset (c) λ Analysis

Fig. 9 Fashion MNIST dataset

COVID CT Scan images This dataset contains COVID CT Scan images.[8] A total of 349 images of 216 patients are COVID-positive and 397 images are non-COVID images. All the images are of different resolutions, varying from minimum 153×124 pixels to maximum of 1853×1485 pixels; averaging 491×383 pixels. Total 400 images are considered in training and remaining 346 for testing.

---

8 https://github.com/UCSD-AI4H/COVID-CT.

Table 16 Fashion MNIST test data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	19.19 ± 11.14	19.47 ± 11.28	19.69 ± 11.37	19.78 ± 11.31	19.83 ± 11.68
L^m	19.98 ± 11.16	19.96 ± 11.26	20.36 ± 11.34	20.58 ± 11.53	20.82 ± 11.79
p-Lap	26.35 ± 10.44	26.94 ± 10.32	27.02 ± 10.38	27.40 ± 10.44	27.57 ± 10.49
EA	40.59 ± 4.01	40.73 ± 4.27	40.85 ± 4.18	40.92 ± 4.16	40.89 ± 4.19
EMR24	14.23 ± 4.67	14.52 ± 4.72	14.77 ± 4.84	14.91 ± 4.71	15.03 ± 4.96
K_5	9.21 ± 4.11	9.01 ± 4.26	9.33 ± 4.38	9.61 ± 4.53	9.84 ± 4.00
K_7	8.77 ± 4.06	8.68 ± 4.01	8.75 ± 4.15	8.92 ± 4.28	9.07 ± 4.31
MMM	8.26 ± 4.05	8.19 ± 4.13	8.35 ± 3.92	8.50 ± 4.41	8.97 ± 4.63
rLap	$\mathbf{6.91 \pm 3.95}$	$\mathbf{6.79 \pm 3.90}$	$\mathbf{6.84 \pm 3.86}$	$\mathbf{6.97 \pm 4.12}$	$\mathbf{7.12 \pm 4.04}$

Minimum mean error has been highlighted in bold

Table 17 Fashion MNIST unlabeled data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	18.60 ± 11.32	18.83 ± 11.51	18.99 ± 11.74	19.25 ± 11.68	19.37 ± 11.88
L^m	19.52 ± 11.23	19.73 ± 11.07	19.88 ± 11.57	20.14 ± 11.94	20.31 ± 11.84
p-Lap	25.16 ± 10.51	25.37 ± 10.56	25.42 ± 10.47	25.69 ± 10.60	25.83 ± 10.38
EA	40.58 ± 4.33	40.69 ± 4.35	40.84 ± 4.41	40.87 ± 4.47	40.97 ± 4.36
EMR24	14.21 ± 4.61	14.48 ± 4.56	14.73 ± 4.49	14.80 ± 4.68	14.94 ± 4.50
K_5	7.11 ± 3.81	7.02 ± 3.72	6.94 ± 3.84	7.14 ± 3.78	7.27 ± 3.93
K_7	6.75 ± 3.80	6.63 ± 3.84	6.70 ± 3.72	6.81 ± 3.90	6.96 ± 3.73
MMM	6.41 ± 3.82	6.46 ± 3.80	6.31 ± 3.86	6.57 ± 3.91	6.72 ± 3.73
rLap	$\mathbf{5.93 \pm 3.82}$	$\mathbf{5.98 \pm 3.82}$	$\mathbf{5.79 \pm 3.75}$	$\mathbf{6.02 \pm 3.93}$	$\mathbf{6.11 \pm 3.90}$

Minimum mean error has been highlighted in bold

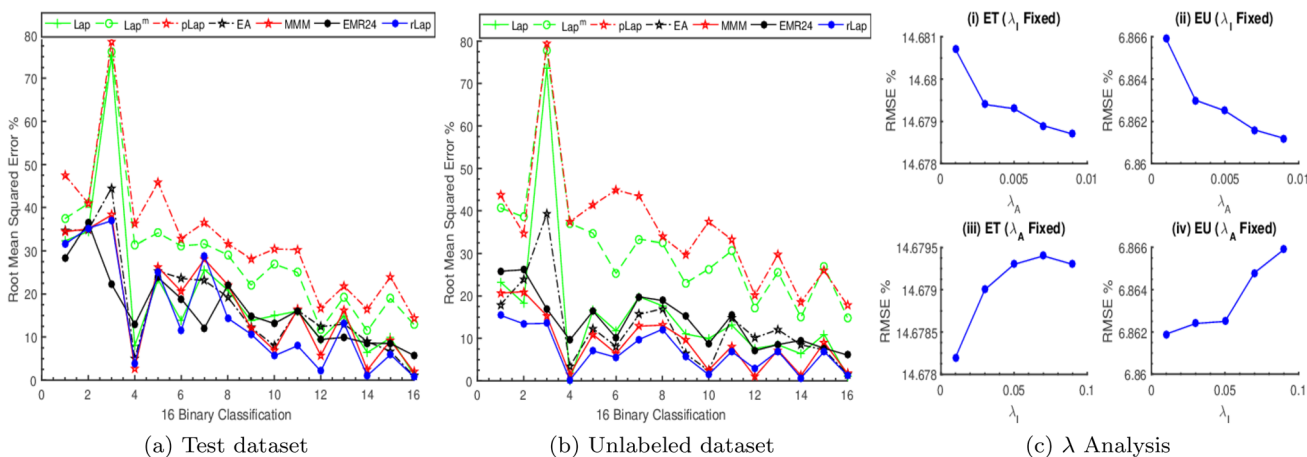(a) Test dataset   (b) Unlabeled dataset   (c) λ Analysis

Fig. 10 Lego bricks dataset

Alzheimer's Brain MRI This dataset consists of 6400 Alzheimer's Brain MRI images[9] of resolution 176×208 pixels each. The dataset contains 5121 images for training and 1279 images for testing. In Alzheimer's, cognitive impairment can be very mild, mild or moderate. So, the dataset has four classes of images namely non demented, very mild demented, mild demented and moderate demented. In the

---

[9] https://www.kaggle.com/tourist55/Alzheimers-dataset-4-class-of-images.

Table 18 Lego bricks test data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	20.00 ± 17.26	20.19 ± 17.29	20.38 ± 17.48	20.53 ± 17.73	20.65 ± 17.62
L^m	28.76 ± 15.50	28.92 ± 15.64	29.04 ± 15.38	29.30 ± 15.79	29.52 ± 15.47
p-Lap	33.26 ± 15.61	33.76 ± 15.70	33.93 ± 15.97	34.18 ± 16.25	34.33 ± 16.19
EA	18.04 ± 12.22	18.28 ± 12.54	18.17 ± 12.70	18.47 ± 13.06	18.81 ± 12.73
EMR24	16.44 ± 8.32	16.59 ± 8.29	16.36 ± 8.37	16.05 ± 8.58	16.14 ± 8.02
K_5	17.45 ± 14.16	17.51 ± 13.89	17.42 ± 14.06	17.64 ± 13.94	17.76 ± 14.67
K_7	16.52 ± 13.34	16.48 ± 13.26	16.42 ± 13.58	16.71 ± 13.60	16.93 ± 13.27
MMM	17.39 ± 12.33	17.52 ± 12.41	17.71 ± 12.73	17.84 ± 12.58	17.97 ± 12.86
rLap	$\mathbf{14.67 \pm 12.62}$	$\mathbf{14.48 \pm 12.84}$	$\mathbf{14.59 \pm 12.36}$	$\mathbf{14.68 \pm 12.52}$	$\mathbf{14.93 \pm 12.19}$

Minimum mean error has been highlighted in bold

Table 19 Lego bricks unlabeled data: mean error (± standard deviation)

Methods	$k = 6$	$k = 8$	$k = 12$	$k = 16$	$k = 20$
Lap	15.61 ± 16.72	15.69 ± 16.88	15.82 ± 16.73	16.13 ± 16.67	16.26 ± 16.78
L^m	31.21 ± 14.77	31.46 ± 14.80	31.58 ± 14.76	31.85 ± 14.82	31.94 ± 14.86
p-Lap	35.69 ± 14.55	35.81 ± 14.79	35.96 ± 14.62	36.13 ± 14.57	36.28 ± 14.69
EA	12.55 ± 9.39	12.68 ± 9.46	12.74 ± 9.57	12.92 ± 9.41	13.03 ± 9.58
EMR24	13.88 ± 6.45	13.79 ± 6.47	13.66 ± 6.53	13.53 ± 6.76	13.38 ± 6.83
K_5	8.73 ± 6.94	8.80 ± 6.78	8.86 ± 6.79	9.03 ± 6.82	9.17 ± 6.73
K_7	8.11 ± 5.96	8.46 ± 5.68	8.67 ± 5.86	8.87 ± 5.70	9.02 ± 5.94
MMM	8.89 ± 6.54	8.97 ± 6.59	9.02 ± 6.48	9.13 ± 6.41	9.24 ± 6.61
rLap	$\mathbf{6.86 \pm 4.89}$	$\mathbf{6.75 \pm 4.93}$	$\mathbf{6.82 \pm 4.94}$	$\mathbf{6.93 \pm 4.79}$	$\mathbf{7.02 \pm 4.85}$

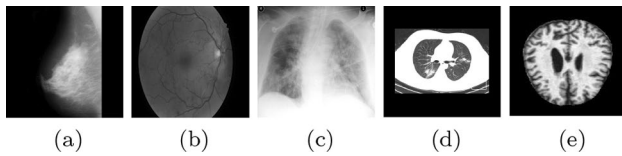Minimum mean error has been highlighted in bold

Fig. 11 Examples of various medical image datasets

experiment non demented and very mild demented images are considered as negative and rest as positive.

The images in Fig. 11 show examples of each medical image dataset. Results obtained for these datasets for $k = 6$ are summarized in Table 20 for test data and in Table 21 for unlabeled data. We can conclude that *rLap* method gives best results for all the medical image unlabeled datasets. For test datasets, *EMR* performs better than *rLap* for Mini-MIAS

Table 20 Medical images test datasets: mean error (± standard deviation)

Methods	Datasets				
	Mini-Mias (200, 122)	IDRiD (344, 172)	COVID-Xray (200, 138)	COVID-CT Scan (400, 346)	Alzheimer (5121, 1279)
Lap	30.13 ± 4.57	19.57 ± 6.83	26.13 ± 7.02	35.14 ± 9.45	24.98 ± 6.71
L^m	31.54 ± 5.17	18.93 ± 8.35	26.62 ± 7.47	30.03 ± 8.05	24.14 ± 7.38
p-Lap	37.46 ± 8.14	22.59 ± 6.57	33.45 ± 9.25	40.45 ± 7.26	31.70 ± 6.40
EA	30.00 ± 8.37	22.67 ± 9.12	26.48 ± 8.98	31.24 ± 11.62	25.51 ± 10.36
EMR24	$\mathbf{23.81 \pm 9.03}$	22.01 ± 9.48	23.17 ± 8.76	30.37 ± 13.50	28.98 ± 11.84
K_5	28.93 ± 8.59	21.31 ± 11.41	30.02 ± 7.23	34.76 ± 12.40	22.67 ± 11.27
K_7	27.87 ± 8.96	20.92 ± 11.53	29.89 ± 6.48	33.65 ± 11.16	22.21 ± 10.87
MMM	28.18 ± 8.04	20.71 ± 10.79	29.15 ± 7.94	33.24 ± 10.10	$\mathbf{21.07 \pm 9.36}$
rLap	24.68 ± 6.23	$\mathbf{17.83 \pm 8.46}$	$\mathbf{22.71 \pm 4.37}$	$\mathbf{29.13 \pm 9.21}$	21.17 ± 5.03

Minimum mean error has been highlighted in bold

Table 21 Medical images unlabeled datasets: mean error

Methods	Datasets				
	Mini-Mias (200, 122)	IDRiD (344, 172)	COVID-Xray (200, 138)	COVID-CT Scan (400, 346)	Alzheimer (5121, 1279)
Lap	30.27 ± 5.23	20.61 ± 7.18	25.12 ± 6.83	33.36 ± 11.30	22.46 ± 7.49
L^m	34.63 ± 6.81	20.19 ± 9.02	26.86 ± 6.10	29.90 ± 7.69	22.06 ± 7.92
p-Lap	35.81 ± 8.94	22.59 ± 7.12	36.57 ± 8.53	40.18 ± 7.58	29.53 ± 7.09
EA	31.41 ± 9.12	20.81 ± 8.46	26.07 ± 9.57	32.46 ± 11.18	26.94 ± 10.61
EMR24	25.50 ± 9.44	24.67 ± 10.00	25.38 ± 8.71	32.43 ± 11.05	26.61 ± 12.32
K_5	30.22 ± 8.34	21.53 ± 10.89	28.90 ± 6.49	34.67 ± 9.97	23.89 ± 9.73
K_7	29.84 ± 7.90	20.07 ± 10.43	28.08 ± 6.91	32.40 ± 9.58	22.54 ± 9.46
MMM	29.59 ± 7.26	19.14 ± 9.15	27.82 ± 7.24	32.48 ± 10.06	21.08 ± 8.62
rLap	$\mathbf{23.22 \pm 5.99}$	$\mathbf{18.93 \pm 8.37}$	$\mathbf{21.17 \pm 4.80}$	$\mathbf{29.27 \pm 8.77}$	$\mathbf{20.72 \pm 5.96}$

Minimum mean error has been highlighted in bold

dataset, whereas *MMM* gives the highest accuracy for Alzheimers' dataset. For COVID-19 and diabetic retinopathy *rLap* performs better than all state-of-the-art methods.

Remarks Based on the experimental results, it is evident that the proposed *rLap* method outperforms the existing manifold regularization methods. While the classification accuracy remained high for spoken alphabets (Isolet), image datasets (fashion MNIST, Lego bricks, and COVID-19 medical images) datasets, moderate enhancement in accuracy has been achieved for the unlabeled data of many handwritten datasets. Performance of the trained model on test data shows significant improvement COID20 dataset, inferring that the model does not overfit. For manifold learning as well, our method better unfolds the synthetic datasets in 2D. So it can be summarized that *rLap* performed better than existing state-of-the-art manifold learning and regularization methods for most of the datasets.

5 Conclusion

It is known that the extent of locally linear neighborhood on a Riemann manifold is difficult to ascertain. In this paper, we flatten a heuristically chosen neighborhood by aligning their tangent spaces to allow Euclidean distance as a measure of pairwise affinity. Extensive experiments on both synthetic and real world datasets prove that our proposed method performs well in both manifold learning and regularization. For dimensionality reduction, our algorithm gives better representation of synthetic datasets over the Laplacian, LTSA, DM, EA, K_5, K_7 and MMM approaches. The reduced classification error for RLSC proved that *rLap* based Euclidean distance between similar neighbors by aligning the tangent spaces of misaligned neighbors is a good representation of affinity. However, the choice of the neighborhood size is vital for the success of the alignment-based affinity measure.

References

1. Abhishek, Verma S (2019) Optimal manifold neighborhood and kernel width for robust non-linear dimensionality reduction. Knowl Based Syst 18:104953. https://doi.org/10.1016/j.knosys.2019.104953

2. Ando R, Zhang T (2006) Learning on graph with Laplacian regularization. Adv Neural Inf Proc Syst 19:25–32

3. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces versus fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720

4. Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput 15(6):1373–1396

5. Belkin M, Niyogi P, Sindhwani V (2005) On manifold regularization. In: AISTATS, p 1

6. Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7(Nov):2399–2434

7. Ching WK, Chu D, Liao LZ, Wang X (2012) Regularized orthogonal linear discriminant analysis. Pattern Recognit 45(7):2719–2732

8. Cohen JP, Morrison P, Dao L (2020) Covid-19 image data collection. arXiv: 2003.11597. https://github.com/ieee8023/covid-chestxray-dataset. Accessed April 2020

9. De la Torre F, Black MJ (2001) Robust principal component analysis for computer vision. In: Proceedings of eighth IEEE international conference on computer vision. ICCV 2001, vol 1. IEEE, pp 362–369

10. Fanty M, Cole R (1991) Spoken letter recognition. In: Lippmann RP, Moody J, Touretzky D (eds) Advances in neural information processing systems, vol 3. Morgan-Kaufmann, pp 220–226. https://proceedings.neurips.cc/paper/1990/file/49182f81e6a13cf5eaa496d51fea6406-Paper.pdf

11. Geng B, Tao D, Xu C, Yang L, Hua XS (2012) Ensemble manifold regularization. IEEE Trans Pattern Anal Mach Intell 34(6):1227–1233

12. Hazelzet J (2019) Images of lego bricks. https://www.kaggle.com/joosthazelzet/lego-brick-images. Accessed January 2020

13. Holm L, Sander C (1996) Mapping the protein universe. Science 273(5275):595–602

14. Hou J, Jun SR, Zhang C, Kim SH (2005) Global mapping of the protein structure space and application in structure-based inference of protein function. Proc Natl Acad Sci 102(10):3651–3656

15. Hou J, Sims GE, Zhang C, Kim SH (2003) A global representation of the protein fold space. Proc Natl Acad Sci 100(5):2386–2390

16. Huang S, Yang D, Zhou J, Zhang X (2015) Graph regularized linear discriminant analysis and its generalization. Pattern Anal Appl 18(3):639–650

17. Kaya M, Binli MK, Ozbay E, Yanar H, Mishchenko Y (2018) A large electroencephalographic motor imagery dataset for electro-encephalographic brain computer interfaces. Sci Data 5:180211

18. Kaynak C (1995) Methods of combining multiple classifiers and their applications to handwritten digit recognition. Unpublished master's thesis, Bogazici University

19. Kim J, Ahn Y, Lee K, Park SH, Kim S (2010) A classification approach for genotyping viral sequences based on multidimensional scaling and linear discriminant analysis. BMC Bioinform 11(1):434

20. Kosinov S, Pun T (2008) Distance-based discriminant analysis method and its applications. Pattern Anal Appl 11(3–4):227–246

21. Krizhevsky A et al (2009) Learning multiple layers of features from tiny images, vol 7. Citeseer

22. Lin T, Liu Y, Wang B, Wang L, Zha H (2016) Local orthogonality preserving alignment for nonlinear dimensionality reduction

23. Lin T, Zha H, Lee SU (2006) Riemannian manifold learning for nonlinear dimensionality reduction. In: European conference on computer vision. Springer, Berlin, pp 44–55

24. Liu W, Ma X, Zhou Y, Tao D, Cheng J (2018) *p*-Laplacian regularization for scene recognition. IEEE Trans Cybern 49(8):2927–2940

25. Liu X, Zhai D, Zhao D, Zhai G, Gao W (2014) Progressive image denoising through hybrid graph Laplacian regularization: a unified framework. IEEE Trans Image Process 23(4):1491–1503

26. Nene SA, Nayar SK, Murase H (1996) Columbia object image library (coil-20). Tech. rep

27. Olivier C, Bernhard S, Alexander Z (2006) Semi-supervised learning. IEEE Trans Neural Netw 20:542

28. Saeed N, Nam H (2016) Cluster based multidimensional scaling for irregular cognitive radio networks localization. IEEE Trans Signal Process 64(10):2649–2659

29. Saul LK, Roweis ST (2000) An introduction to locally linear embedding. Unpublished. http://www.cs.toronto.edu/~roweis/lle/publications.html. Accessed May 2019

30. Saul LK, Roweis ST (2003) Think globally, fit locally: unsupervised learning of low dimensional manifolds. J Mach Learn Res 4(June):119–155

31. Singer A, Wu HT (2012) Vector diffusion maps and the connection laplacian. Commun Pure Appl Math 65(8):1067–1144

32. Taşdemir K, Yalçin B, Yildirim I (2015) Approximate spectral clustering with utilized similarity information using geodesic based hybrid distance measures. Pattern Recognit 48(4):1465–1477. https://doi.org/10.1016/j.patcog.2014.10.023

33. Tenenbaum JB, De Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323

34. Thoma M (2017) The hasyv2 dataset. arXiv preprint arXiv:1701.08380

35. Turk MA, Pentland AP (1991) Face recognition using eigenfaces. In: Proceedings. 1991 IEEE Computer society conference on computer vision and pattern recognition. IEEE, pp 586–591

36. Vidya G, Omprakash S (2016) Survey on recent researches on high level image retrieval. Int J Comput Sci Eng 4(9):72–77

37. Vladymyrov M, Carreira-Perpinan M (2013) Entropic affinities: properties and efficient numerical computation. In: International conference on machine learning, pp 477–485

38. Xiao H, Rasul K, Vollgraf R (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747

39. Yadav RK, Verma S, Venkatesan S et al (2020) Regularization on a rapidly varying manifold. Int J Mach Learn Cybern 1–20

40. Yeung KY, Ruzzo WL (2001) Principal component analysis for clustering gene expression data. Bioinformatics 17(9):763–774

41. Zelnik-Manor L, Perona P (2004) Self-tuning spectral clustering. In: Advances in neural information processing systems 17, Neural Information Processing Systems, {NIPS} 2004, December 13–18, 2004,Vancouver, British Columbia, Canada, pp 1601–1608. http://papers.nips.cc/paper/2619-self-tuning-spectral-clustering

42. Zhang H, Wu QJ, Chow TW, Zhao M (2012) A two-dimensional neighborhood preserving projection for appearance-based face recognition. Pattern Recognit 45(5):1866–1876

43. Zhang L, Lin J, Karim R (2018) Adaptive kernel density-based anomaly detection for nonlinear systems. Knowl Based Syst 139:50–63

44. Zhang Z, Zha H (2003) Nonlinear dimension reduction via local tangent space alignment. In: International conference on intelligent data engineering and automated learning. Springer, Berlin, pp 477–481

45. Zhang Z, Zha H (2004) Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM J Sci Comput 26(1):313–338

46. Zheng X, Ma Z, Li L (2020) Local tangent space alignment based on Hilbert–Schmidt independence criterion regularization. Pattern Anal Appl 23(2):855–868. https://doi.org/10.1007/s10044-019-00810-6

47. Zheng-Bradley X, Rung J, Parkinson H, Brazma A (2010) Large scale comparison of global gene expression patterns in human and mouse. Genome Biol 11(12):R124

48. Zhou X, Belkin M (2011) Semi-supervised learning by higher order regularization. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp 892–900