Rezazadeh Hamedani *et al. J Big Data* (2021) 8:135
https://doi.org/10.1186/s40537-021-00527-6

Journal of Big Data

RESEARCH Open Access

Dissimilarity space reinforced with manifold learning and latent space modeling for improved pattern classification

Azadeh Rezazadeh Hamedani, Mohammad Hossein Moattar* and Yahya Forghani

*Correspondence:
moattar@mshdiau.ac.ir
Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

Abstract

Dissimilarity representation plays a very important role in pattern recognition due to its ability to capture structural and relational information between samples. Dissimilarity space embedding is an approach in which each sample is represented as a vector based on its dissimilarity to some other samples called prototypes. However, lack of neighborhood-preserving, fixed and usually considerable prototype set for all training samples cause low classification accuracy and high computational complexity. To address these challenges, our proposed method creates dissimilarity space considering the neighbors of each data point on the manifold. For this purpose, Locally Linear Embedding (LLE) is used as an unsupervised manifold learning algorithm. The only goal of this step is to learn the global structure and the neighborhood of data on the manifold and mapping or dimension reduction is not performed. In order to create the dissimilarity space, each sample is compared only with its prototype set including its k-nearest neighbors on the manifold using the geodesic distance metric. Geodesic distance metric is used for the structure preserving and is computed using the weighted LLE neighborhood graph. Finally, Latent Space Model (LSM), is applied to reduce the dimensions of the Euclidean latent space so that the second challenge is resolved. To evaluate the resulted representation ad so called dissimilarity space, two common classifiers namely K Nearest Neighbor (KNN) and Support Vector Machine (SVM) are applied. Experiments on different datasets which included both Euclidean and non-Euclidean spaces, demonstrate that using the proposed approach, classifiers outperform the other basic dissimilarity spaces in both accuracy and runtime.

Keywords: Dissimilarity space, Prototype selection, Manifold learning, Global structure learning, Locally Linear Embedding (LLE), Latent Space Model (LSM)

Introduction

In dissimilarity space embedding, each pattern is mapped into a vector space based on its differences to other patterns. The main characteristic of this method is that any dissimilarity measure, whether metric or non-metric, Euclidean or non-Euclidean, symmetric or asymmetric, can be used [1]. For two reasons, research in this area is important. Firstly, this method completes feature-based representation [2] and secondly, this method is a bridge between statistical and structural approaches in pattern recognition [3].

One of the most challenging points of this embedding is the prototype selection approach, which includes the selection strategy and the number of prototypes. It will have a very significant impact on reducing the computational complexity of measuring dissimilarity, the dimensions of the final embedding space and subsequently improving classifier's performance. Another significant point is the lack of attention to the data structure and preserving neighbors of samples while constructing and embedding to the dissimilarity space.

Our contribution in this paper is to provide a method for selecting prototypes considering the manifolds and data structure, then reducing dimensions and complexity of the proposed dissimilarity space. In the proposed method, prototype set P is not fixed for all samples and each sample has its own specific prototypes which are its neighbors on the manifold. So, one sample is not forced to measure its differences with all other members of prototype set. Also dimensions of the proposed dissimilarity space is reduced by Latent Space Model. In this way, set P is pruned and dimension of dissimilarity space is reduced so it is likely that the computational complexity will decrease while the performance of the classifiers in terms of accuracy and runtime can be improved.

The rest of the paper is organized as follows. “Related works” section describes the related works in this area. Then, in “Background knowledge” section, explanations are given about the LLE method, geodesic distance and the latent space model as the tools used in the proposed method. Then the proposed approach is fully explained step-by-step in “The proposed method” section. In “Experimental results and discussion” section, the evaluation datasets and the evaluation scenarios are discussed and the results are analyzed. Finally, “Conclusion” section summarizes the whole paper and proposes some guidance for the future works.

Related works

In 1985, Goldfrab proposed to compute the differences between structural data to represent them rather than a feature-based representation for the first time. Since it was not a useful approach, it could not be conspicuous enough for researchers. Pseudo-Euclidean space was also proposed by Goldfrab in the same year [3, 4]. After 1995, Duin and Pȩkalska [1] began their research in this area, and eventually in 2005 introduced a distance-based method called the dissimilarity representation as a substitute for the feature-based representation. Dissimilarity space was first presented in 2008 by Pȩkalska [5] as a way to map structural data to vector space, so that traditional learning algorithms can be applied. Also, in 2012, the dissimilarity space was described as a bridge between structural and statistical approaches [3].

In recent years, dissimilarity space has been applied in some major areas such as classification of images and spectrograms [6, 7], handwritten digits [2, 3, 8–13], radiological images processing to detect autoimmune diseases [14], classification of biological arrays such as microarray data [15], 3D streamlines of human brain [16], spectrum representation [17], time series classification [13, 18], time series classification [47] and graph representation and classification [19–21].

Nowadays deep learning models are pioneer in surveillance applications, but to achieve higher performance in reidentification persons, dissimilarity space is a better approach [22, 23]. In [22] dissimilarity representation is used to identify differences occurred in source and target captured videos. Also a dissimilarity-based loss function which is optimized by gradient descent is suggested to apply in identification system systems.

Prototype selection has an important role in reducing dimensions of dissimilarity space. In the prototype selection approaches, the most diverse samples have to be selected as prototypes and among similar samples only one will be selected [24]. In 2006, Pȩkalska et al. [25] proposed some methods for pruning the prototype set (P) such as Random, RandomC, Kcenters, and EditCon. Also Calaña et al. [26] used a genetic algorithm as a method for prototype selection. In 2015, a method for generating and selecting prototypes for structural data is presented [12]. Principal Component Analysis (PCA) is used as a method to reduce the dimension of the dissimilarity space [18] and Genetic Algorithm (GA) applied in the approximate search method in dissimilarity space [27].

In 2021, Silva et al. proposed a prototype generation method and estimates optimal number of prototypes [28]. This algorithm in first step with a Self-Organizing Maps neural network generates a reduced training samples then in next step uses a classifier based on the nearest informative neighbors.

Recent approaches of prototype selection applied to multi-label data classification [29], click fraud detection [30] and generating business process model [31] are proposed in the last two years. In [30], first of all divides imbalanced dataset into four groups and applies under-sampling technique to balance dataset. At the second step, only relevant samples are chosen as prototypes in order to reduce size of training set.

In [48] a novel prototype learning approach is proposed which is a modified version of Zero-shot learning which learns prototypes from unseen classes via training a classification model. This approach is a simple and efficient approach to learn prototypes from the class-level semantic information.

In all previously studied works, the prototype set P is constant for all samples. That is, any sample must necessarily be compared to all the members of prototype set P, and then the dissimilarity space is created. Therefore, in this paper the proposed set P is defined dynamically and unevenly for all samples. In other words, a sample is not required to be compared with all the prototypes in set P to create the dissimilarity space, but also considering the data structure and neighborhood of each sample on the manifold, it is just compared with a limited set of members. The main goals of this new framework are to reduce the complexity and dimensions of space and also to achieve higher classification performance and lower runtime.

Since similar data which belong to the same class have little differences, so they are closer to each other in the dissimilarity space, and vice versa. As a result, it’s likely that different class samples will be more distinct in this space, and so construction of classifiers will be more justified [25]. The most commonly used classifiers in the dissimilarity space are KNN and Linear SVM [3, 7, 14, 18]. In spite of the existence of higher performance classifiers, KNN classifier is one of the most widely used classifiers in this space, because of its simplicity and good behavior at metric-based spaces.

Therefore, in this paper the KNN and SVM classifiers are chosen to evaluate our proposed method.

Background knowledge

Locally linear embedding (LLE)

This learning algorithm recognizes the fundamental structure of the manifold and reduces dimensions. LLE is nonlinear, based on Eigen vectors and preserves the local neighborhood of samples [32]. It has three main steps:

Finding l neighbors for each x_i sample with D dimensions.

Reconstructing each sample based on the weighted linear combination of its l nearest neighbors and obtaining the optimal weights W_{ij} by minimizing the following equation. In other words, a weighted neighborhood graph is constructed in this step and W_{ij}=0 if x_j is not neighbor of x_i [33]:

$$\varepsilon(W) = \sum_i \left| \vec{x}_i - \sum_j^l W_{ij} \vec{x}_j \right|^2 \{x_j\} : l \text{ nearest neighbors of } x_i$$

$$\sum_j W_{ij} = 1 \tag{1}$$

With the optimal weights W_{ij} and minimizing the following equation, new embedded y_i vectors with d dimensions (d < D) can be obtained [32].

$$\Phi(\mathrm{y}) = \sum_i \left| \vec{y}_i - \sum_j^l W_{ij} \vec{y}_j \right|^2 \{y_j\} : l \text{ nearest neighbors of } \mathrm{y_i} \tag{2}$$

Some of its advantages are simplicity of implementation, escaping from local minima, mapping high dimensional data into a single coordinate system of lower dimensions and faster rate in calculating the Eigenvalues of the matrix $M = I - W^T(I - W)$ due to the fact that many weights are zero [32, 34].

Geodesic distance

In graph theory, the sequence of edges between two vertices is called path. The length of the path between two vertices in an unweighted graph is the number of edges and in a weighted graph is the sum of the weights of the edges between them. The distance between two vertices, which is equal to the length of the shortest path between them, is called geodesic distance [35].

In this step, an undirected weighted graph is created according to the LLE manifold. Each point on the LLE manifold is assumed as a vertex (y_i) and Euclidean distance of two adjacent vertices is the weight of their connected edge (Fig. 1).

In order to compute geodesic distance between two vertices $y_i$andy_j, if they are neighbors, the following equation is used:

$$gDist\left(y_i.y_j\right) \approx d\left(y_i.y_j\right) = \sqrt{\left(y_{i1} - y_{j1}\right)^2 + \left(y_{i2} - y_{j2}\right)^2 + \cdots + \left(y_{id} - y_{jd}\right)^2} \tag{3}$$

But if they are not neighbors, the length of shortest distance is approximately considered as geodesic distance [33].

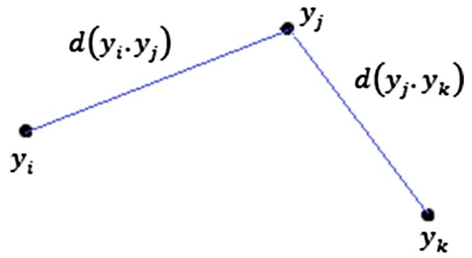


Fig. 1 The LLE neighborhood graph

$$gDist\left(y_i.y_k\right) = d\left(y_i.y_j\right) + d(y_j.y_k) \tag{4}$$

Latent Space Model (LSM)

Latent Space Model (LSM) was proposed in 2002 by Hoff et al. [36] to estimate similarity between nodes and their connectivity in the social networks. In this model, for each node x_i in the X_{N*N} input network, there is an embedded position z_i in the unseen latent spaceZ_{N*Q}. This latent space has three characteristics: (1) Euclidean property: many traditional classification algorithms can be performed in this space. (2) Its dimensions are lower than initial input network' dimensions ($Q \ll N$) so this model can be used as dimension reduction method. (3) The probability of the communication edge,Y_{ij}, between the two nodes x_i and x_j in the network, depends on the Euclidean distance between their positions in the latent space, z_i and z_j [36]. The process of the model is as follows:

- *Initialization* The latent space Z_{N*Q} is initialized randomly with a normal distribution with zero mean and unit variance.

$$z_{ij} = N(0.I)i.j = 1 \ldots N \tag{5}$$

- *Inference* In this step, Maximum a Posteriori (MAP) method is used to estimate hidden space Z under the model. Equation 6 estimates the latent variable Z by the MAP method where X presents input network and Z presents the latent space.

$$Z_{MAP} = arg_z \mathrm{maxlogp(X|Z)} \tag{6}$$

$$= arg_z \mathrm{max} \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{logp(x_i|Z)}$$

$$= arg_z \mathrm{min} \left(-\frac{1}{\mathrm{N}} \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{logp(x_i|Z)} \right)$$

In order to estimate the hidden variable Z by MAP, the gradient descent optimization method is used in a definite number of iterations until it converges to local optima [37].

$$\nabla_z \mathrm{logp}(\mathrm{X.Z}) \tag{7}$$

- *Estimate connected edges* This step is optional unless you are looking to analyze latent space and to estimate connectivity patterns. As the probability of being the edge Y_{ij} between the two nodes x_i and x_j in the network depends on the Euclidean distance between their positions in the latent space, z_i and z_j so $|z_i - z_j|$ is calculated. Then the probability of the edge, Y_{ij} is computed in terms of the Poisson probability distribution function.

$$Y_{ij} = Poisson(\frac{1}{|z_i - z_j|}) \tag{8}$$

The proposed method

The framework of the proposed method as illustrated in Fig. 2 consists of five steps. In the following, each step of the approach is described as an algorithm.

In Algorithm 1, we use the Locally Linear Embedding (LLE) algorithm, which is one of the neighborhood-preserving manifold learning algorithms to learn the data manifold. The input matrix X_{N*D} contains training samples. N is the number of samples in training set, and D presents the number of the dimensions of each sample. The output matrix Y_{N*d} where N is the number of samples and d shows the dimension of the embedded samples where d is lower than D ($d \ll D$). The LLE algorithm consists of three steps, which is described more in details in "Locally linear embedding (LLE)" section.

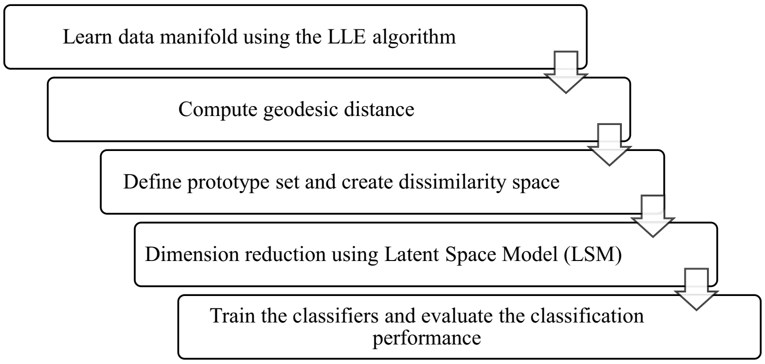

Learn data manifold using the LLE algorithm
Compute geodesic distance
Define prototype set and create dissimilarity space
Dimension reduction using Latent Space Model (LSM)
Train the classifiers and evaluate the classification performance

Fig. 2 The proposed data representation and classification scheme

Algorithm 1: Learn the data manifold using the LLE algorithm

```
    Input: $X_{N*D}$={$x_i$}$_1^N$ and $x_i$ is a D-dimensional vector
    Output: $y_{N*d}$={$y_i$}$_1^N$ and $y_i$ is a d-dimensional vector (d<D)

For all $x_i$ in training set
        $\{x_j\}^l$: $l$ −nearest neighbors of $x_i$ using Euclidean distance

        //find optimal $W_{ij}$
        $\varepsilon(W) = \arg min_W (\sum_i|\vec{x}_i - \sum_j^l W_{ij}\vec{x}_j|^2)$        $\sum_j W_{ij} = 1$

        // Find optimal $y_i$
        $\Phi(y) = \arg min_y \sum_i|\vec{y}_i - \sum_j^l W_{ij}\vec{y}_j|^2$
End
```

In Algorithm 2, first an undirected weighted graph is constructed according to the data manifold obtained by the LLE algorithm. Each new embedded point on the LLE manifold is assumed as a vertex (y_i) that is connected to the l— nearest neighbors (e_{ij}) and Euclidean distance between them is considered as the weight of their connected edge (w_{ij}). Now geodesic distance matrix is computed.

Algorithm2: Compute geodesic distance

```
    Input: y={$y_i$}$_1^N$
    Output: $\text{gDist}_{N*N}$

// Create Graph
For all nodes $y_i$ in graph G
        For ($y_j$:  $l$-nearest neighbors of $y_i$)
                Edges = $e_{ij}$
                $w_{ij} = d(i.j)$        //Euclidean distance
        End
End

// Compute geodesic distance matrix $\text{gDist}_{N*N}$
For all entries of $\text{gDist}_{N*N}$
        If exist( $e_{ij}$)
                $\text{gDist}_{ij} = w_{ij}$
        Else
                $\text{gDist}_{ij} =$ length of shortest path between $y_i$ and $y_j$
End
```

Algorithm 3 consists of two main steps: (1) define prototype set (2) construct dissimilarity space.

First, the geodesic distance matrix is sorted ascending and the k-nearest neighbors of each sample on the manifold are selected to define the prototype set P_i of each sample x_i (Eq. 9). It should also be noted that the farthest neighbor (α_i) of each sample is also determined.

$$P_i = \{p_{i1}.\dots.p_{ik}\} \tag{9}$$

In step 2, dissimilarity space is created as a matrix DS_{N*N} whose diagonal elements are zero.

As the default value for entries, the distance between each sample and its farthest neighbor (α_i) on the manifold is placed in the matrix DS (Eq. 11). Then, the Euclidean distance of each sample x_i with its k-nearest neighbors on the manifold $\{p_{i1}.\ldots.p_{ik}\}$ is computed and is placed in DS (Eq. 10). It should be noted that we prefer to use the initial samples (X_{N*D}) before any dimension reduction, so the dissimilarity space is not constructed using manifold embedding Y_{N*d}.

$$d(x_i.x_j) = d_{ij} = \sqrt{\left(x_{i1} - x_{j1}\right)^2 + \cdots + \left(x_{iD} - x_{jD}\right)^2} \tag{10}$$

$$DS_{ij} = \left\{ \begin{array}{ll} 0 & if\ i = j \\ d(x_i.x_j) & if\ j \in P_i \\ d(x_i.\alpha_i) & otherwise \end{array} \right\} \tag{11}$$

Algorithm 3: Define prototype set and create dissimilarity space

```
Step1: Define prototype set P

    Input: geodesic distance matrix gDist$_{N*N}$
    Output: prototype set P

    All neighbors=Sort (gDist, ascending)
    For all $x_i$ in training set
            // k-nearest neighbors' in
            $P_i$ = All_neighbors [1: k]
    End

Step2: Construct dissimilarity space

    Input: prototype set P
    Output: dissimilarity space $DS_{N*N}$

    For all $DS_{ij}$ in matrix $DS_{N*N}$
            If ($i = j$)
                    $DS_{ij} = 0$
            Else If ($x_j$ is in prototype set of $P_i$)
                    $DS_{ij} = d(x_i.x_j)$
            Else
                    // $\alpha_i$ = the farthest neighbor of $x_i$ on the manifold
                    $DS_{ij} = d(x_i.\alpha_i)$
    End
```

In algorithm 4, Latent Space Model (LSM) is used to simplify and to reduce the dimensions of dissimilarity space. First, the latent space with N*Q dimensions that N is the number of training samples and Q is lower than N, is initialized randomly with zero mean and unit variance. Then in a loop with the user-defined number of iterations, the Maximum a Posteriori (MAP) method estimates the optimal positions in the latent space Z_{N*Q}. To implement this step, Edward library [37], which is based on Python programming language, is used. Final augmented dissimilarity space is obtained by inner product of DS_{N*N} and Z_{N*Q} as in the following equation (Eq. 12).

$$DS_{Augmented} = DS_{N*N} \bullet Z_{N*Q} \tag{12}$$

Algorithm4: Dimension reduction using Latent Space Model (LSM)

```
Input: Dissimilarity space $DS_{N*N}$
Output: Latent space $Z_{N*Q}$ .($Q \ll N$) and $\mathrm{DS}_{\mathrm{Augmented}}$ (an N*Q transition matrix)
For all entries of $z_{ij}$ in latent space $Z_{N*Q}$
        $z_{ij} = N(0.I)$
End
While (estimation_iteration)
        $Z_{MAP} = arg_z \min(-\frac{1}{N}\sum_{i=1}^{N} \log p(x_i|Z))$
End
$DS_{Augmented} = DS_{N*N} \cdot Z_{N*Q}$
```

Finally, classifiers such as KNN and SVM are trained by the N-by-Q matrix of $DS_{Augmented}$ and classify new queries which are mapped into the augmented space. In order to evaluate the results, accuracy and runtime are considered as classification performance measurements.

Experimental results and discussion

Data sets

Since the proposed method is not limited to a particular problem, nine Euclidean and non-Euclidean data sets that are resulting in 10 dissimilarity spaces are used to evaluate the approach (more than one measure and different number of classes and samples are considered in these dissimilarity spaces). Datasets of classification problems in University of California Irvine (UCI) machine learning repository [38] such as iris, glass, wine, breast cancer, monk, ionosphere radar signals and 8×8 images of optical recognition of handwritten digits are used in this paper. USPS (United States Postal Services) [39] and MNIST (Mixed National Institute of Standards and Technology) [40] are two other standard data sets for the handwritten digit recognition problem and they are chosen to evaluate our approach. The images in USPS are 16×16 inches and they are shown as records with 256 features. Also, Euclidean and two-way tangent [41, 42] measures are chosen to construct the dissimilarity spaces of USPS data set. MNIST includes 70,000 handwritten digits which are centered in images with 28×28 pixels. The general information about these datasets is indicated in Table 1.

Performance measurement

In this study, accuracy measure, that is the most instinctive performance measure, is used to evaluate the classification performance of the proposed approach and also is reported with the measure of mean and standard deviation in K-fold cross validation as an evaluation method. This measure in Eq. 13 is a ratio of correctly classified samples (t) includes true positives plus true negatives to the total samples (n) [43].

$$Accuracy = \frac{t}{N} * 100 \tag{13}$$

Table 1 Descriptions of the datasets

Dataset	Measure Type	# Records	# Dimensions	# Classes
Iris	Euclidean	150	4	3
Glass	Euclidean	214	9	7
Ionosphere	Euclidean	351	34	2
Monk	Euclidean	122	6	2
Breast Cancer	Euclidean	569	30	2
Wine	Euclidean	178	13	3
USPS_nE	Two-way tangent	250	256	10
USPS_E	Euclidean	500	256	10
Digit5	Euclidean	200	64	5
MNIST	Euclidean	70,000	784	10

F1-score is another metric which used in this paper as a measure of test's accuracy and calculated based on precision and recall and is shown in Eq. 14. Tp,fp and fn are abbreviations of true positive, false positive and false negative identified results [44].

$$F_1 = 2 * \frac{precision * recall}{precision + recall} = 2 * \frac{tp}{tp + \frac{1}{2}(fp + fn)} \tag{14}$$

In addition, the efficiency of the proposed method is also evaluated in terms of runtime. The runtimes have been obtained on a laptop with an Intel® Core™ i5-3230 M CPU @ 2.8 GHz with 8 GB RAM.

Evaluation scenarios

Four scenarios are considered for the evaluation of the proposed approach:

1. *Evaluating the whole proposed framework* New dissimilarity space according to the proposed algorithm using predefined prototype number list for each dataset (Table 2) is constructed and it will be embedded into the lower dimension latent space. Then classifiers such as KNN with different K values (i.e., 1, 3, 5, 7, 11), linear SVM and polynomial SVM (degree = 2, 3) are used to evaluate the method in terms of accuracy and runtime.
2. *Comparing with basic dissimilarity spaces* The proposed method is compared in both accuracy and runtime with basic dissimilarity spaces which are defined as follows:
- DS_ALL: All samples in the training set are selected as the representatives. In other words, the representative set R is equal to the training set [3].
- DS_Random: K samples are randomly selected as the representatives [25].
- DS_RandomC: From each class C, k random samples are selected as the representatives (K = k × C) [25].

3. *Evaluating the LSM phase* In this case the classifiers are trained on the dissimilarity space DS_{N*N} without any dimension reduction and mapping into the latent space. Doing so, the impact of the LSM and dimension reduction on the performance of the classifiers is determined.
4. *Comparing with the recent similar works* The highest classification performance of the proposed method is compared with three similar recent publications in term

Table 2 Parameters of the proposed method

Dataset	LLE (k_LLE)	LSM (lsm_ dimension)	LSM (lsm_iteration)	K-fold cross validation	Prototype numbers (k_neighbor_manifold)
Iris	55	75	100	10	1,20,75,100,149
Glass	10	107	100	10	1,10,60,140,213
Ionosphere	35	200	100	10	1,20,50,100,350
Monk	10	120	100	10	1,20,60,90,121
Breast Cancer	35	500	100	10	1,10,20,50,100
Wine	15	140	100	10	1,10,20,50,80
USPS_nE	30	125	100	10	1,20,40,60,249
USPS_E	10	420	100	10	1,20,100,120,150
Digit5	20	160	100	10	1,10,20,100,150
MNIST	100	2000	100	10	1,1000,5000,10,000

of accuracy on the same datasets. More details about these papers are described in "Comparing with the recent approaches" section.

Results

Table 2 shows the values of the required parameters for each dataset including the number of neighbors needed to learn the LLE Manifold (k_LLE), the latent space dimensions (lsm_dimension), the number of iterations of estimation in LSM (lsm_iteration), the number of folds in cross validation (K-fold) evaluation scheme and the number of neighbors of each sample on the manifold that should be considered as prototypes in proposed method (k_neighbor_manifold). It should be noted that in this study the optimal k-nearest neighbors of LLE (k_LLE) is defined with an approach introduced in paper [45] and other parameters are tuned by a trial and error method.

Evaluating the whole proposed framework

As described in previous sections, the Locally Linear Embedding (LLE) algorithm, is learned the data manifold. Figure 3 shows the LLE embedded neighborhood graph with optimal number of neighbors.

The LLE embedded neighborhood graph of MNIST data due to its huge samples is ambiguous so its manifold based on first two features is shown in Fig. 4.

To compute geodesic distance, the undirected weighted graph is constructed based on the obtained LLE manifold. In this graph, as suggested in paper [45] each point is connected to the small fixed number of neighbors ($l = 10$).

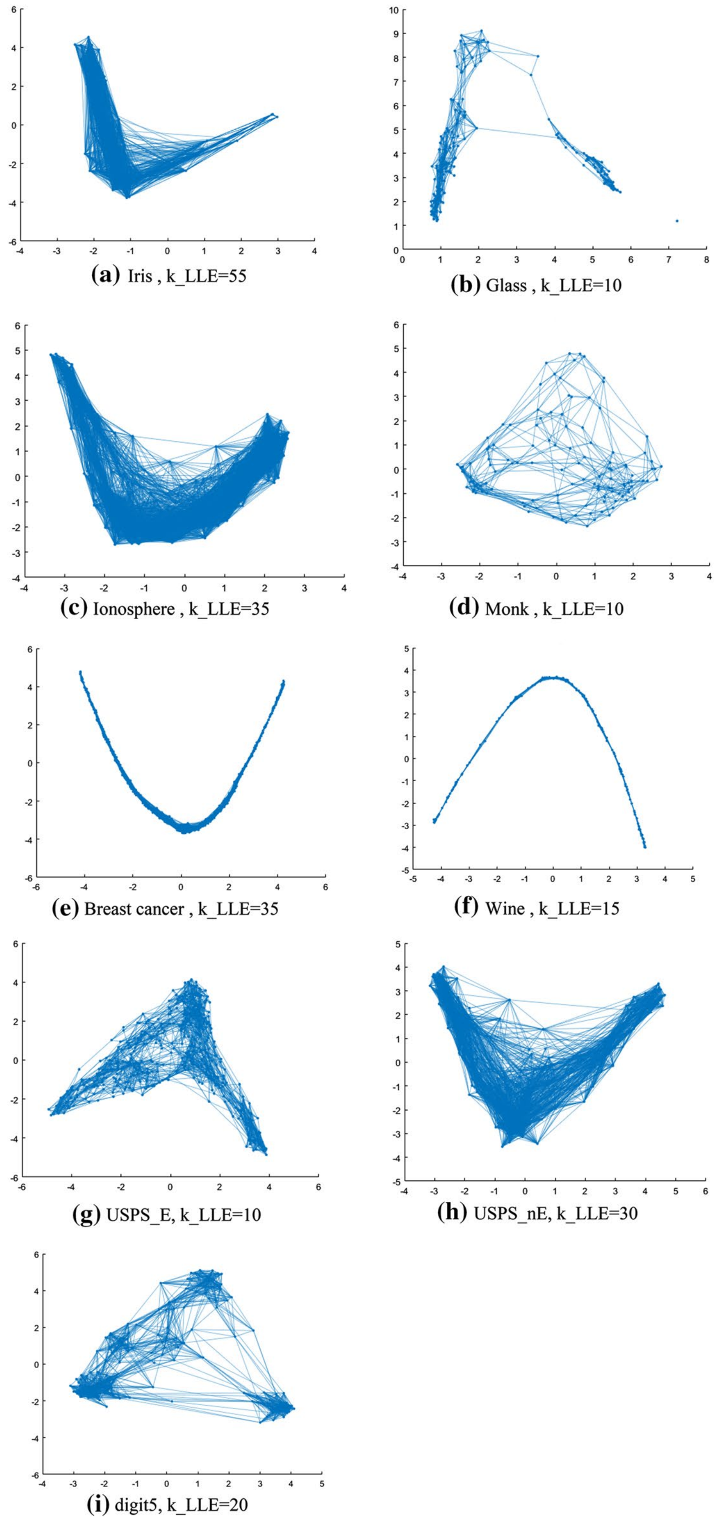
Tables 3 and 4 report the accuracy and runtime of the proposed method on Iris dataset. The numerical value in the columns shows the number of the nearest neighbors of each sample on the manifold which are determined as the prototypes for constructing dissimilarity space (k_neighber_manifold). The results of the Linear SVM shows that the maximum accuracy achieved by our proposed method with comparing each point to only its 20-nearest neighbors on the manifold, is 98% at 0.01s. This is while the accuracy of other classifiers in dissimilarity spaces considering with different number of prototypes are lower than 98.

Figure 5 demonstrates the classification error curve of the proposed method versus the number of prototypes of each sample on the manifold, on 10 evaluation data sets and optimal prototype numbers for each dataset are shown in this figure.

Comparing with the basic dissimilarity spaces

In Table 5 the best and the average accuracy of the proposed method on each dataset and in Table 6 the best and the average F1-score are compared with other basic dissimilarity spaces and the initial dataset. The numerical values in the parenthesis demonstrate the optimal number of prototypes and the best classifier on each dataset. It should be noted that the number of random prototypes in DS_Random and DS_RandomC are equals with optimal number of prototypes in proposed method.

Figures 6 and 7 demonstrate the comparison results of the proposed method and other dissimilarity spaces based on the best and the average accuracy, respectively.



(a) Iris , k_LLE=55

(b) Glass , k_LLE=10

(c) Ionosphere , k_LLE=35

(d) Monk , k_LLE=10

(e) Breast cancer , k_LLE=35

(f) Wine , k_LLE=15

(g) USPS_E, k_LLE=10

(h) USPS_nE, k_LLE=30

(i) digit5, k_LLE=20

Fig. 3 LLE embedded neighborhood graph

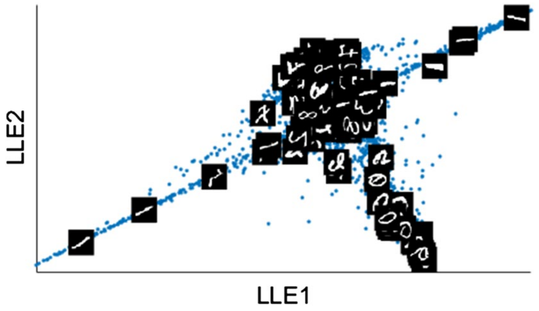


Fig. 4 MNIST LLE manifold

Table 3 Evaluation of proposed method on Iris dataset in terms of accuracy (%) with tenfold cross validation (best results denote in italics typeface)

Classifier	#Prototype				
	1	20	75	100	149
1-nn	96.00(± 1.2)	95.33(± 0.8)	96.00(± 0.2)	96.00(± 0.8)	96.00(± 0.7)
3-nn	94.00(± 2.1)	95.33(± 1.1)	94.00(± 1.2)	96.67(± 1.6)	96.00(± 1.4)
5-nn	95.33(± 2.3)	95.33(± 1.3)	93.33(± 2.4)	96.67(± 1.4)	95.33(± 1.2)
7-nn	91.33(± 3.3)	96.00(± 1.1)	92.67(± 3.2)	96.00(± 1.9)	97.33(± 0.5)
11-nn	94.00(± 2.5)	96.00(± 0.9)	94.67(± 2.1)	96.67(± 1.3)	96.67(± 1.2)
linSVM	94.67(± 1.1)	*98.00*(± 0.2)	95.33(± 2.3)	95.33(± 2.3)	95.33(± 2.3)
polySVM(2)	94.67(± 1.3)	97.33(± 0.8)	95.33(± 1.2)	94.67(± 3.1)	95.33(± 1.2)
polySVM(3)	94.67(± 2.3)	97.33(± 1.5)	95.33(± 1.4)	94.00(± 3.2)	94.67(± 1.3)
Average	94.33(± 1.38)	*96.33*(± 1.07)	94.58(± 1.15)	95.75(± 1.0)	95.83(± 0.85)

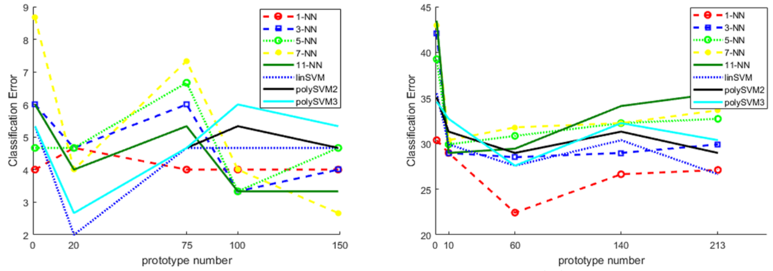
Table 4 Evaluation of proposed method on Iris dataset in terms of runtime (s)

Classifier	#prototype				
	1	20	75	100	149
1-nn	0.10	0.09	0.09	0.09	0.11
3-nn	0.10	0.09	0.11	0.09	0.11
5-nn	0.11	0.09	0.10	0.09	0.13
7-nn	0.10	0.09	0.09	0.09	0.09
11-nn	0.12	0.09	0.09	0.08	0.09
linSVM	0.01	*0.01*	0.00	0.01	0.01
polySVM(2)	0.01	0.00	0.00	0.01	0.01
polySVM(3)	0.01	0.00	0.00	0.01	0.01
Average	0.10	*0.09*	0.09	0.09	0.11

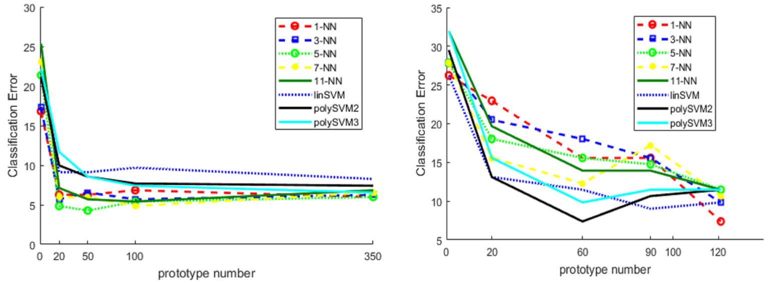
Evaluating the LSM phase

In this section, the proposed method, is evaluated in terms of accuracy and runtime with and without incorporating the LSM approach. This makes it possible to determine the impact and importance of using the LSM on the classifier's efficiency.

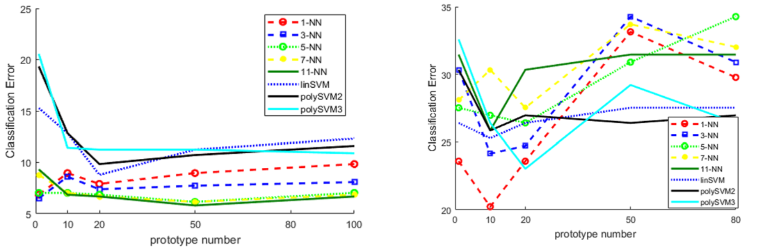
In Table 7, the best and average accuracy of the proposed method with and without LSM method are shown on all datasets and indicate the superiority of the proposed



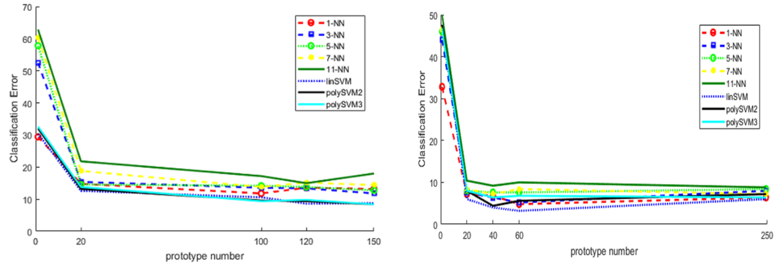
(a) Iris **(b)** Glass



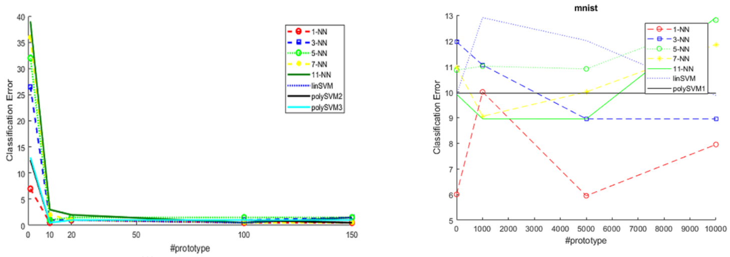
(c) Ionosphere **(d)** Monk



(e) Breast cancer **(f)** Wine



(g) USPS_E **(h)** USPS_nE



(i) digit5 **(j)** mnist

Fig. 5 Classification error of the proposed method versus the number of prototypes for the evaluation datasets

Table 5 Classification accuracy of the proposed method and other basic dissimilarity spaces (±standard deviation)

Dataset	Proposed Algorithm	DS_All	DS_Random	DS_RandomC	Dataset
Iris					
Best (20,linSVM)	*98.00(±0.2)*	96.00(±2.6)	96.67(±2.7)	96.67(±2.7)	97.33(±0.05)
Average	*96.33(±1.7)*	95.92(±2.7)	96.17(±2.8)	95.83(±3.2)	95.92(±0.04)
Glass					
Best(60,1-NN)	*77.57(±2.1)*	73.83(±6.2)	72.90(±6.1)	72.43(±5.3)	69.16(±5.8)
Average	71.61(±2.4)	70.09(±5.6)	69.68(±5.8)	70.15(±5.6)	67.00(±7.5)
Ionosphere					
Best(50,5-NN)	*95.73(±1.1)*	92.59(±2.8)	92.59(±2.2)	91.74(±2.6)	84.33(±3.6)
Average	*93.38(±1.5)*	93.36(±2.1)	93.02(±2.4)	92.45(±3.6)	82.59(±3.8)
Monk					
Best(60,polySVM2)	*92.62(±0.4)*	90.98(±1.2)	90.98(±1.2)	90.98(±1.3)	90.80(±1.3)
Average	*89.55(±0.9)*	89.14(±0.4)	87.86(±1.1)	89.14(±0.3)	82.07(±0.8)
BreastCancer					
Best(50,11-NN)	94.20(±0.8)	92.27(±3.2)	91.56(±3.9)	92.27(±2.8)	*96.49*(±0.5)
Average	*91.83(±0.7)*	91.55(±0.4)	87.81(±1.8)	89.26(±1.2)	89.98(±1.5)
Wine					
Best(10,linSVM)	74.72(±1.2)	80.34(±2.1)	73.03(±3.5)	74.72(±4.2)	*96.07*(±0.5)
Average	74.37(±1.8)	72.89(±3.2)	65.17(±6.1)	67.98(±4.5)	*96.35*(±0.4)
USPS_nE					
Best(60,linSVM)	*96.80(±0.2)*	94.40(±0.01)	88.00(±2.3)	87.60(±3.1)	87.60(±2.4)
Average	*93.55(±1.5)*	72.60(±3.2)	70.50(±3.5)	70.45(±3.1)	82.55(±2.4)
USPS_E					
Best(120,linSVM)	*91.40(±1.0)*	89.80(±0.8)	90.40(±0.4)	89.00(±0.7)	90.40(±0.7)
Average	88.03(±0.1)	*88.05*(±0.9)	87.05(±0.7)	86.10(±1.4)	85.80(±0.4)
Digit5					
Best(10,linSVM)	*99.50(±0.1)*	99.50(±0.07)	97.50(±0.7)	96.50(±0.8)	99.50(±0.00)
Average	99.19(±0.1)	*99.25*(±0.9)	96.50(±1.1)	96.75(±0.7)	98.69(±0.4)
MNIST					
Best(5000,1-NN)	94.05(±3.5)	*96.67*(±0.8)	95.67(±0.4)	83.33(±2.1)	97.30(±0.2)
Average	90.46(±4.6)	94.21(±0.9)	*94.25*(±0.5)	82.50(±2.3)	93.29(±1.01)

method over the method without LSM in most cases. It is due to the effect of the LSM on reducing the size of the dissimilarity space.

Figures 8 and 9 shows the comparison results of the proposed method with and without LSM based on the best and the average accuracy, respectively.

Also, in terms of runtime, as reported in Table 8, the proposed method requires less time for classification. In Figs. 10 and 11 comparison results between two cases are demonstrated more clearly. Since the classification runtime of MNIST dataset is not comparable to other datasets due its large scale, it is not shown in Figs. 10 and 11 and just is reported in Table 8.

Comparing with the recent approaches

In this section, the highest classification accuracy of the proposed method is compared with three latest papers on the same datasets. In 2015, Calvo et al. [12] proposed a two-stage algorithm to generate prototypes on structural data using dissimilarity space. First

Table 6 Classification F1-score of the proposed method and other basic dissimilarity spaces

Dataset	Proposed Algorithm	DS_All	DS_Random	DS_RandomC	Dataset
Iris					
Best (20,linSVM)	*0.98*(±0.03)	0.95(0.03)	0.96(0.02)	0.96(0.02)	0.93(±0.03)
Average	*0.95*(±0.03)	0.88(±0.04)	0.88(±0.03)	0.88(±0.03)	0.94(±0.02)
Glass					
Best(60,1-NN)	*0.56*(±0.32)	0.56(±0.3)	0.56(±0.3)	0.54(±0.3)	0.53(±0.2)
Average	*0.50*(±0.3)	0.49(±0.3)	0.46(±0.3)	0.46(±0.3)	0.45(±0.3)
Ionosphere					
Best(50,5-NN)	0.94(±0.03)	0.91(±0.03)	0.91(±0.03)	0.90(±0.04)	0.82(±0.1)
Average	0.92(±0.03)	0.92(±0.03)	0.91(±0.03)	0.90(±0.03)	0.83(±0.09)
Monk					
Best(60,polySVM2)	0.89(±0.00)	0.88(±0.00)	0.85(±0.01)	0.93(±0.00)	0.92(±0.00)
Average	0.87(±0.01)	0.88(±0.00)	0.88(±0.00)	0.89(±0.00)	0.81(±0.00)
BreastCancer					
Best(50,11-NN)	0.91(±0.03)	0.91(±0.03)	0.90(±0.03)	0.90(±0.03)	0.90(±0.03)
Average	0.8(±0.08)	0.82(±0.08)	0.85(±0.07)	0.83(±0.07)	0.86(±0.06)
Wine					
Best(10,linSVM)	0.69(±0.17)	0.80(±0.05)	0.70(±0.13)	0.69(±0.12)	0.96(±0.02)
Average	0.6(±0.2)	0.62(±0.17)	0.63(±0.20)	0.61(±0.20)	0.81(±0.02)
USPS_nE					
Best(60,linSVM)	0.89(±0.17)	0.85(±0.2)	0.75(±0.2)	0.78(±0.2)	0.79(±0.19)
Average	0.83(±0.2)	0.85(±0.19)	0.80(±0.24)	0.81(±0.19)	0.76(±0.18)
USPS_E					
Best(120,linSVM)	0.86(±0.09)	0.89(±0.06)	0.85(±0.09)	0.83(±0.09)	0.85(±0.09)
Average	0.78(±0.11)	0.73(±0.08)	0.81(±0.11)	0.80(±0.12)	0.83(±0.08)
Digit5					
Best(10,linSVM)	0.98(±0.01)	0.98(±0.01)	0.95(±0.01)	0.94(±0.05)	0.98(±0.01)
Average	0.98(±0.01)	0.92(±0.02)	0.95(±0.02)	0.95(±0.03)	0.98(±0.01)
MNIST					
Best(5000,1-NN)	0.85(±0.02)	0.90(±0.02)	0.89(±0.03)	0.92(±0.02)	0.98(±0.00)
Average	0.83(±0.02)	0.91(±0.01)	0.90(±0.02)	0.91(±0.01)	0.97(±0.01)

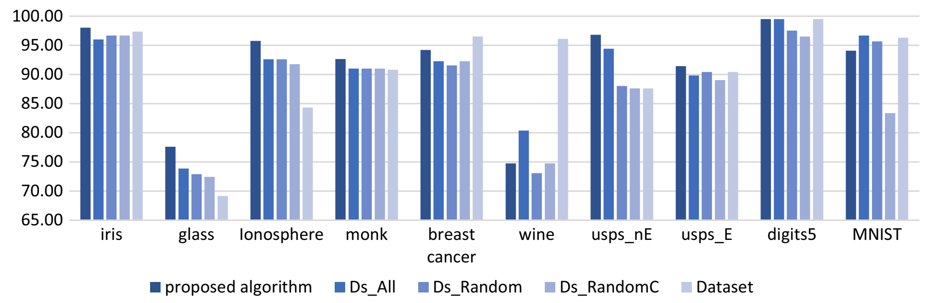


Fig. 6 Best Classification Accuracy of proposed method and other dissimilarity spaces on evaluation datasets

the structural data is embedded into the dissimilarity space and then the common prototype generation methods are applied. In 2017 a measure of ultrametricity is introduced for creating dissimilarity space [36]. Then the performance of KNN classifier on the

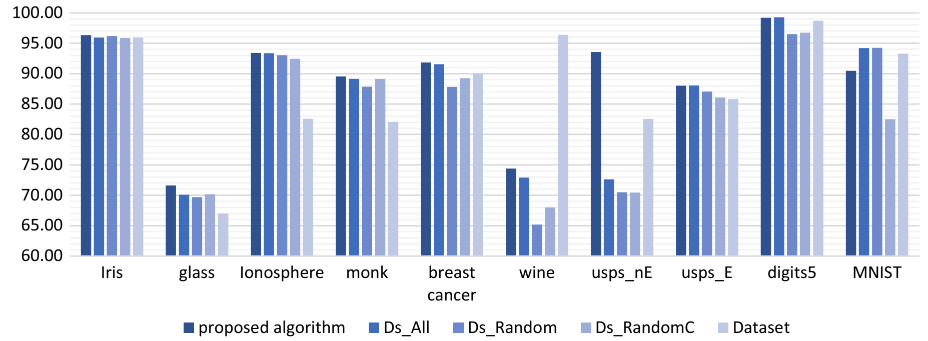


Fig. 7 Average classification accuracy of proposed method and other dissimilarity spaces on evaluation datasets

Table 7 Classification accuracy of the proposed method with and without LSM

Dataset	**Proposed Algorithm**		**Proposed Algorithm without LSM**	
	Accuracy(±std)	f_1	**Accuracy(±std)**	f_1
Iris				
Best (20,linSVM)	*98.00(±0.2)*	0.98(±0.03)	98.00(±0.0)	0.96(±0.03)
Average	96.33(±1.7)	0.95(±0.03)	96.67(±0.1)	0.96(±0.03)
Glass				
Best(60,1-NN)	*77.57(±2.1)*	0.56(±0.32)	76.64(±1.5)	0.50(±0.30)
Average	71.61(±2.4)	0.50(±0.3)	71.85(±2.5)	0.42(±0.60)
Ionosphere				
Best(50,5-NN)	*95.73(±1.1)*	0.94(±0.03)	91.45(±1.3)	0.94(±0.07)
Average	*93.38(±1.5)*	0.92(±0.03)	93.16(±0.2)	0.83(±0.07)
Monk				
Best(60,polySVM2)	*92.62(±0.4)*	0.89(±0.00)	90.98(±1.1)	0.87(±0.00)
Average	*89.55(±0.9)*	0.87(±0.01)	88.52(±0.7)	0.86(±0.01)
BreastCancer				
Best(50,11-NN)	*94.20(±0.8)*	0.91(±0.03)	94.20(±0.1)	0.91(±0.03)
Average	*91.83(±0.7)*	0.8(±0.08)	91.40(±0.6)	0.73(±0.09)
Wine				
Best(10,linSVM)	74.72(±1.2)	0.69(±0.17)	75.28(±0.5)	0.68(±0.18)
Average	*74.37(±1.8)*	0.6(±0.2)	73.53(±1.1)	0.5(±0.2)
USPS_nE				
Best(60,linSVM)	*96.80(±0.2)*	0.89(±0.17)	94.40(±0.9)	0.90(±0.11)
Average	*93.55(±1.5)*	0.83(±0.2)	74.10(±2.3)	0.85(±0.17)
USPS_E				
Best(120,linSVM)	*91.40(±1.0)*	0.86(±0.09)	90.60(±0.3)	0.80(±0.12)
Average	*88.03(±0.1)*	0.78(±0.11)	87.25(±0.2)	0.79(±0.14)
Digit5				
Best(10,linSVM)	*99.50(±0.1)*	0.98(±0.01)	99.50(±0.0)	0.97(±0.01)
Average	*99.19(±0.1)*	0.98(±0.01)	99.06(±0.1)	0.98(±0.01)
MNIST				
Best(50,linSVM)	94.05(±3.5)	0.85(±0.02)	88.88(±5.03)	0.75(±0.03)
Average	90.46(±4.6)	0.83(±0.02)	85.16(±4.7)	0.73(±0.02)

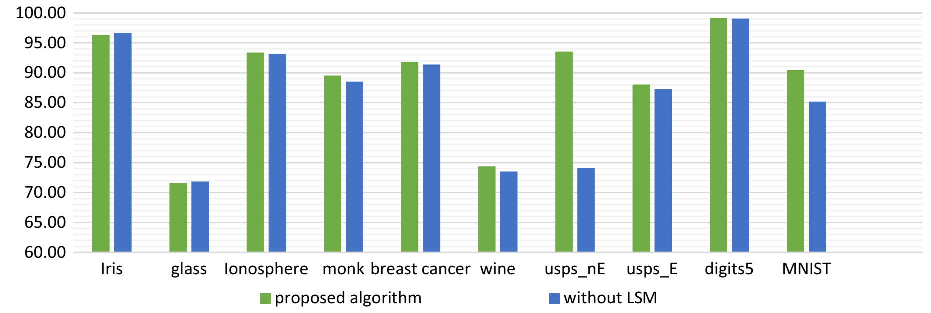


Fig. 8 Average classification accuracy of proposed method with and without LSM on evaluation datasets

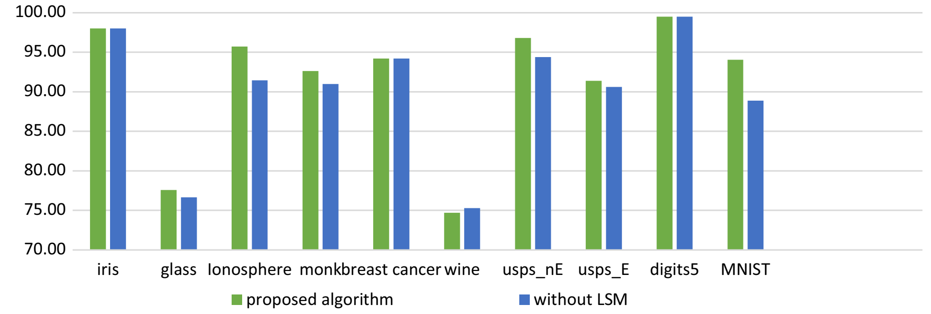


Fig. 9 The best classification accuracy of proposed method with and without LSM on evaluation datasets

transformed space was studied on Iris and Ionosphere datasets. Structured Sparse Dictionary Selection (SSDS) was proposed in 2017 by Wang et al. [24] as a novel prototype selection method. Table 9 shows that the proposed method outperforms the other three latest methods on the same datasets which besides the runtimes reported in Sect. 5.4.3 denotes the impact and effectiveness of the proposed dissimilarity space representation.

Conclusions

The dissimilarity space is a vector space which dimensions are determined by comparing differences between each sample in the training set and all members of the prototype set P. Therefore, prototype selection strategy and the size of the prototype set are so important and effective [1]. However, in previous works, the prototype set is identical for all samples [15, 25, 26], that is, all of them must necessarily be compared with all the members in the prototype set P. Also the neighboring of each sample and data structure on the manifold are not considered in prototype selection or mapping into dissimilarity space. Therefore, in this paper, a method is proposed to augment the dissimilarity space with manifold learning and increase the classification performance, considering the neighborhood of each sample on the manifold.

In the proposed method, first the data manifold is learned by the Linear Locally Embedding algorithm (LLE) [32, 34]. Then an undirected weighted graph is constructed according to the data manifold obtained by the LLE algorithm and the geodesic distance matrix is computed. The k-nearest neighbors of each sample on the manifold using the geodesic distance are selected as members of the prototype set P.

Table 8 Classification runtime of the proposed method with and without LSM

Dataset	Proposed Algorithm	Proposed Algorithm without LSM
Iris		
Best (20,linSVM)	0.01	0.05
Average	0.06	0.07
Glass		
Best(60,1-NN)	0.10	0.13
Average	0.10	0.13
Ionosphere		
Best(50,5-NN)	0.15	0.17
Average	0.13	0.17
Monk		
Best(60,polySVM2)	0.02	0.09
Average	0.08	0.08
BreastCancer		
Best(50,11-NN)	0.40	0.45
Average	0.47	0.53
Wine		
Best(10,linSVM)	0.10	0.15
Average	0.11	0.10
USPS_nE		
Best(60,linSVM)	0.08	0.15
Average	0.10	0.18
USPS_E		
Best(120,linSVM)	0.88	1.09
Average	0.51	0.60
Digit5		
Best(10,linSVM)	0.06	0.08
Average	0.09	0.10
MNIST		
Best(5000,1-NN)	10,000	12,000
Average	11,200	13,300

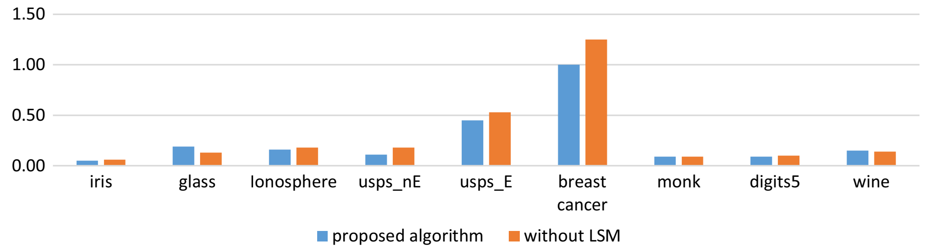


Fig. 10 Average classification runtime of proposed method with and without LSM on evaluation datasets

In the proposed process of constructing the dissimilarity space, it is enough to compare each sample with its k-nearest neighbors on the manifold, and not all the members of the prototype set P. This is an effective factor in reducing the computational complexity. In next step the Latent Space Model simplifies and reduces dimensions of

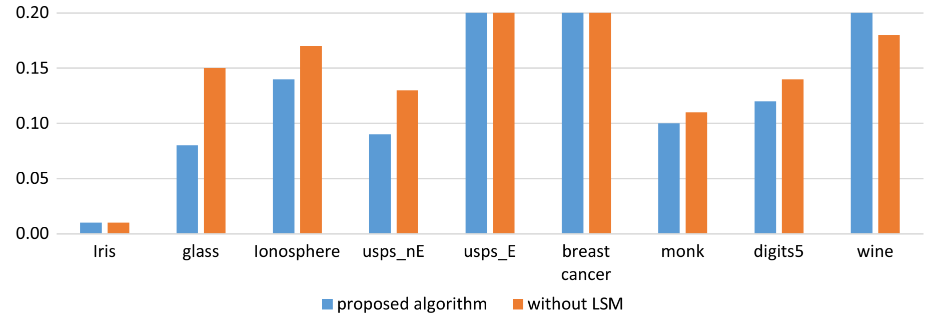


Fig. 11 The best classification runtime of proposed method with and without LSM on evaluation datasets

Table 9 Comparing the highest classification performance (accuracy in percent) of the proposed method with the recent similar approach (the values reported are the highest performance reported in each paper)

Methods	Iris	Ionosphere	USPS
Proposed method	98(± 0.2)(linSVM)	95.73(± 1.1)(5-NN)	96.80(± 1.0)(linSVM)
Calvo et al. [10]	–	–	86.70(1-NN)
Simovici et al. [46]	94.00(7-NN)	61.00(3-NN)	–
Wang et al. [24]	–	–	85.67(linSVM)

dissimilarity space more and embeds all samples to the Euclidean latent space which has a lower dimensionality than the constructed dissimilarity space [36]. Finally, the SVM and KNN classifiers are trained on the augmented dissimilarity space and the proposed method is evaluated in terms of classification accuracy and runtime. Advantages of the proposed method are as follows:

- Creating the prototype set P dynamically and unevenly for all samples, so that each sample is not required to be compared with all the prototypes in the set P, but only compared with its k-nearest neighbors on the manifold and then the dissimilarity space is created. As a result, this has a great effect on reducing the computational complexity.
- The proposed complete framework, in comparison with the method without LSM, requires less classification runtime, which indicates the positive effect of the LSM model on complexity reduction of the augmented dissimilarity space.
- The proposed framework outperforms the accuracy of the latest similar approaches.
- In most scenarios, the linear SVM classifier on the proposed method can achieve the highest accuracy.

The algorithm for estimating and embedding the dissimilarity space into the latent space is time-consuming. Therefore, using alternative algorithms as the future works can improve the runtime of the proposed method. Also, in the LSM method, determining the optimal number of latent space dimensions and the number of iterations for the estimation step will greatly influence the efficiency of the model and classifiers.

Also, other dimension reduction or mapping techniques to the latent space can be used for future works.

Abbreviations

LLE: Locally Linear Embedding; LSM: Latent Space Model; KNN: K Nearest Neighbor; SVM: Support Vector Machine; PCA: Principal Component Analysis; GA: Genetic Algorithm; MAP: Maximum a Posteriori; UCI: University of California Irvine; USPS: United States Postal Services; MNIST: Mixed National Institute of Standards and Technology; SSDS: Structured Sparse Dictionary Selection.

Acknowledgements

All the contributors are mentioned in the manuscript.

Authors' contributions

ARH (first author) performed the main implementations and evaluations. MHM and YH (second and third authors) were thesis supervisor and advisor respectively and the idea was originated by them. All authors read and approved the final manuscript.

Funding

The authors approve that there was no funding in preparing the manuscript.

Availability of data and materials

The datasets and codes will be available after acceptance.

Declarations

Ethics approval and consent to participate

This manuscript does not include studies involving human participants.

Consent for publication

This manuscript does not contain any individual person's data.

Competing interests

The authors approve that there is no conflict of interest in the manuscript.

Received: 29 June 2021 Accepted: 8 October 2021

References

1. Pękalska E, Duin RPW. The Dissimilarity Representation for Pattern Recognition: Foundations and Applications. Singapore: World Scientific; 2005. p. 607.
2. Xu W. Non-Euclidean Dissimilarity Data in Pattern Recognition. In: Department of Computer Science. 2013, University of York.
3. Duin RPW, Pękalska E. The dissimilarity space: Bridging structural and statistical pattern recognition. Pattern Recogn Lett. 2012;33(7):826–32.
4. Goldfrab L, Kanal L, Rosenfeld A. A new approach to pattern recognition. Prog Pattern Recognition. 1985;2:241–402.
5. Pękalska E, Duin RPW. Beyond Traditional Kernels: Classification in Two Dissimilarity-Based Representation Spaces. IEEE Trans Syst Man Cybern. 2008;38(6):729–44.
6. Nanni L, et al. Experiments of image classification using dissimilarity spaces built with siamese networks. Sensors. 2021;21(5):1573.
7. Nanni L, et al. Spectrogram Classification Using Dissimilarity Space. Appl Sci. 2020;10(12):4176.
8. Duin RPW, Pękalska E. Zero-error dissimilarity based classifiers. arXiv preprint arXiv:1601.04451. 2016.
9. Duin RPW, Pękalska E, Loog M, Non-Euclidean Dissimilarities: Causes, Embedding and Informativeness, in Similarity-Based Pattern Analysis and Recognition, M. Pelillo, editors. Springer. London: London; 2013. p. 13–44.
10. Eskander GS, Sabourin R, Granger E. On the dissimilarity representation and prototype selection for signature-based bio-cryptographic systems. In: Hancock E, Pelillo M, Ed. Similarity-Based Pattern Recognition: Second International Workshop, SIMBAD 2013, York, UK, July 3–5, 2013. 2013, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 265–280.
11. Bunke H, Riesen K. Towards the unification of structural and statistical pattern recognition. Pattern Recogn Lett. 2012;33(7):811–25.
12. Calvo-Zaragoza J, Valero-Mas JJ, Rico-Juan JR. Prototype generation on structural data using dissimilarity space representation: a case of study. Pattern Recogn Image Analysis. 2015;9117:75–82.
13. Iwana BK, et al. Efficient temporal pattern recognition by means of dissimilarity space embedding with discriminative prototypes. Pattern Recogn. 2017;64:268–76.
14. Theodorakopoulos I, et al. HEp-2 cells classification via sparse representation of textural features fused into dissimilarity space. Pattern Recogn. 2014;47(7):2367–78.
15. Garcia V, Sanchez JS. Mapping microarray gene expression data into dissimilarity spaces for tumor classification. Inf Sci. 2015;294:362–75.

16. Avesani P, et al. Tractography Mapping for Dissimilarity Space across Subjects. In: 2015 International Workshop on Pattern Recognition in NeuroImaging. 2015.
17. Paclík P, Duin RPW. Dissimilarity-based classification of spectra: computational issues. Real-Time Imaging. 2003; 9(4): 237–244.
18. Jain B, Spiegel S. Dimension Reduction in Dissimilarity Spaces for Time Series Classification. In: Douzal-Chouakria A, Vilar JA, Marteau P-F, editors. Advanced Analysis and Learning on Temporal Data: First ECML PKDD Workshop, AALTD 2015, Porto, Portugal, September 11, 2015, Revised Selected Papers. Cham: Springer International Publishing; 2016. p. 31–46.
19. Bunke H, Riesen K. Graph Classification Based on Dissimilarity Space Embedding, in Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, SSPR & SPR 2008, Orlando, USA, December 4–6, 2008. In: Proceedings, N. da Vitoria Lobo, et al., Editors. 2008, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 996–1007.
20. Livi L. Designing labeled graph classifiers by exploiting the R\'enyi entropy of the dissimilarity representation. arXiv preprint arXiv:1408.5286; 2014.
21. Livi L, Rizzi A, Sadeghian A. Optimized dissimilarity space embedding for labeled graphs. Inf Sci. 2014;266:47–64.
22. Mekhazni D, et al. Unsupervised Domain Adaptation in the Dissimilarity Space for Person Re-identification. Cham: Springer International Publishing; 2020.
23. Uddin MK, et al. Fusion in dissimilarity space for RGB-D person re-identification. In: Array. 2021. p. 100089.
24. Wang H, et al. Representative Selection with Structured Sparsity. Pattern Recogn. 2017;63(3):268–78.
25. Pękalska E, Duin RPW, Paclík P. Prototype selection for dissimilarity-based classifiers. Pattern Recogn. 2006;39(2):189–208.
26. Calana YP, et al. Prototype Selection for Dissimilarity Representation by a Genetic Algorithm. In: 2010 20th International Conference on Pattern Recognition. 2010.
27. Bernhauer D, Skopal T. Approximate search in dissimilarity spaces using GA. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. 2019, Association for Computing Machinery: Prague, Czech Republic. p. 279–280.
28. Silva LA, de Vasconcelos BP, Del-Moral-Hernandez E. A model to estimate the Self-Organizing Maps grid dimension for Prototype Generation. Intelligent Data Anal. 2021;25:321–38.
29. Devi VS, Kuruvilla SA, Aparna R. Prototype Selection and Dimensionality Reduction on Multi-Label Data. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD. 2020, Association for Computing Machinery Hemavati: Hyderabad, India. p. 195–199.
30. Sisodia D, Sisodia DS. Quad division prototype selection-based k-nearest neighbor classifier for click fraud detection from highly skewed user click dataset. Eng Sci Technol Int J. 2021;9:78.
31. Fani SM, Boltenhagen M, van der Aalst W. Prototype Selection Using Clustering and Conformance Metrics for Process Discovery. Cham: Springer International Publishing; 2020.
32. Saul LK, Roweis ST. An Introduction to Locally Linear Embedding. 2000.
33. Wilson RC. Similarities, Distances and Manifold Learning. 2012.
34. Ventura D. Manifold Learning Examples - PCA, LLE and ISOMAP. 2008.
35. Goddard W, Oellermann OR. Distance in Graphs. In: Dehmer M, editor. Structural Analysis of Complex Networks. Boston: Birkhäuser Boston; 2011. p. 49–72.
36. Hoff PD, Raftery AE, Handcock MS. Latent space approaches to social network analysis. J Am Stat Assoc. 2002;97(460):1090–8.
37. Dustin T, et al. Edward A library for probabilistic modeling, inference, and criticism. arXiv preprint arXiv:1610.09787; 2016.
38. Dheeru DAKT. UCI Machine Learning Repository, I. California: University of California, School of Information and Computer Sciences; 2017.
39. Hull J. A database for handwritten text recognition research. IEEE Trans Pattern Anal. 1994;16(5):550–4.
40. LeCun Y, Cortes C. MNIST handwritten digit database. 2010.
41. Keysers D, et al. Adaptation in statistical pattern recognition using tangent vectors. IEEE Trans Pattern Anal Mach Intell. 2004;26(2):269–74.
42. Haasdonk B. Distance matrices. 2005. https://lmb.informatik.uni-freiburg.de/people/haasdonk/datasets/distances.en.html.
43. Classification Accuracy. 2017. https://www.gepsoft.com/gepsoft/APS3KB/Chapter09/Section2/SS02.htm.
44. van Rijsbergen CJ. Information retrieval. 2nd ed. 1979.
45. Kurasova O, Dzemyda G. Selection of the number of neighbours of each data point for the Locally Linear Embedding Algorithm. Vol. 36; 2007.
46. Simovici DA, Vetro R, Hua K. Ultrametricity of Dissimilarity Spaces and Its Significance for Data Mining. In: Guillet F, Pinaud B, Venturini G, editors. Advances in Knowledge Discovery and Management, vol. 6. Cham: Springer International Publishing; 2017. p. 141–55.
47. Mauceri S, Sweeney J. Dissimilarity-based representations for one-class classification on time series. Pattern Recogn. 2020;100:107122.
48. Ji Z, Cui B, Yu Y, et al. Zero-shot classification with unseen prototype learning. Neural Comput Appl. 2021. https://doi.org/10.1007/s00521-021-05746-9.

Publisher's Note