NR-Isomap: An Incremental Approach with Gaussian Process Kernels for Denoising

Mahwish Yousaf
School of Computer Science and Technology University of Science and Technology of China
Hefei, China
mahwish@ustc.edu.cn

Muhammad Saaadat Shakoor Khan
Key Laboratory of Materials Physics Institute of Solid-State Physics, Chinese Academy of Sciences
Hefei, China
saadat@mail.ustc.edu.cn

Shamsher Ullah
Department Knowledge Unit of System and Technology University of Management and Technology Sialkot, Pakistan
shamsher.ullah@skt.umt.edu.pk

Shou Wang
School of Computer Science and Technology University of Science and Technology of China
Hefei, China
shouwang@ustc.edu.cn

Li Jing
School of Computer Science and Technology University of Science and Technology of China
Hefei, China
lj@ustc.edu.cn

2023 IEEE 6th International Conference on Big Data and Artificial Intelligence (BDAI) | 979-8-3503-0366-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/BDAI59165.2023.10256697

***Abstract*—The Isomap is a popular nonlinear dimensionality reduction method often used for low-dimensional embedding space. When the big data is noisy, Isomap often fails because of the graph structure' Short-Circuit Edges (SCE) problem. In this paper, an incremental Noise-Reduction Isomap (NR-Isomap) approach is to address the SCE problem in a graph structure. The GP kernels are combined with the manifold learning approach and provide significant results. NR-Isomap is based on the Hessian Locally Linear Embedding (HLLE) algorithm with Gaussian Process (GP) kernels. The HLLE algorithm provides the optimal solution to the SCE problem caused by an overly noisy dataset. NR-Isomap provides an answer to the Isomap SCE problem in noisy big data problems that cannot be handled using existing methods. Therefore, the GP kernel significantly improves and measures the performance of the NR-Isomap accuracy. Our NR-Isomap method outperforms experiments on SK-Learn datasets and shows efficient results compared to Isomap. Our NR Isomap is much more efficient and faster compared to Isomap. The efficiency and accuracy of the NR-Isomap method are demonstrated theoretically and compared with other manifold learning methods.**

Keywords*—*Noise reduction Isomap, Manifold, learning Hessian locally linear embedding algorithm, Gaussian process kernels, Big data

I. Introduction

Machine learning developed various models such as Gaussian Process (GP), Naïve Bayes (NB), Support Vector Machine (SVM), and Hidden Markov Model (HMM). Williams et al. [1] proposed GP models for machine learning to solve classification and nonlinear regression problems. GP models are expressive, avoid overfitting, can be interpret, and provide impressive performance in various empirical comparisons [2-4]. GP models can construct a direct prior over function instead of parameters based on multiple properties like smoothness, periodic, etc. It uses a covariance kernel and determines the similarities of the training data points in the random function. The choice of kernels profoundly affects the performance of the GP on a given input task [5].

Manifold learning offers various algorithms for linear and nonlinear dimensionality reduction for high dimensional and large datasets, such as Isomap [6], Laplacian Eigenmap [7], Locally Embedded Analysis (LEA) [8], Principle Component Analysis (PCA) [9], Locally Linear Embedding (LLE) [10], Local Tangent Space Alignment (LTSA) [11] and Locality Preserving Projection (LPP) [12]. Isomap has become a trendy topic in nonlinear manifold learning dimension reduction. Tenenbaum et al. [13] proposed the Isomap algorithm, which uses the MDS algorithm to calculate geodesic distance instead of Euclidean distance. It preserves the geometry structure and maps the data in low-dimensional space. If the dataset is too noisy and sparse, the k nearest neighbourhood data points on the manifolds will be far apart. If the knn distance between the folds of the manifolds is smaller than the knn distance, the Isomap algorithm often fails when the SCE problem occurs [14-16]. It is unsuitable for real datasets, which are usually infected with noise [17]. Some noisy dataset manifolds cannot find the low dimensional manifolds because the position of the data points is outside the manifolds.

In the noisy problem literature, Heeyoul et al. [18] deal with the SCE problem in the topological structure and propose Robust Kernel Isomap [6]. Choi et al. [19] proposed kernel Isomap for the lack of topological stability due to noise and outliers. Hong Chang et al. [20] proposed the Robust LLE for noisy datasets, using the Weighted-Robust PCA algorithm to eliminate the outliers and noise from the data. Bo Li et al. [21] proposed an extended Isomap to overcome the noise problem using a robust PCA and box statistical algorithm. Chen and Mao [22] suggested LP Center-based Isomap (LPC-Isomap) for high time complexity, sensitivity to outliers and short-circuit edge problems and used k-Nearest Center (k-NC) and Linear Patch (LP) algorithms. Chun Du et al. [23] proposed a Robust Isomap based on the Neighbor Ranking Metric (NRM) method for outliers that can eliminate the outliers of selected neighbourhood data points. Ashutosh et al. [15] suggested Locally Linear Isomaps (LL-Isomaps) for the problems with SCE and noisy datasets. Mahwish et al. [24] proposed the Noise Removal Isomap with a classification method (NRIC) for noise problems, that reduces noise and increases Isomap performance. Shengfeng et al. [25] presented the Robust Supervised Isomap method for noise problems in hyperspectral data, and it measures the similarities and calculates the triple geodesic distance of the Isomap. Alaor et al. [26] suggested ISOMAP-KL for the outliers and noise issues in data using a parametric patch-based algorithm. Guy et al. [16] presented high-time Topologically Constrained Isomap Embedding (TCIE) for removing and detecting inconsistent distance problems from Isomap. Yves et al. [14] proposed the Density-based Graph Denoising (DGD) method to detect the shortest and noisiest datasets. It can create a short circuit-free graph for noisy datasets. All of the previous techniques perform well for outliers, noise, and SCE problems but are not very efficient. Although these techniques focus only on the noise and SCE issues, they are neither accuracy nor time.

This paper aims to generate a neighbourhood graph on a noisy dataset and transform the data points into a low-dimensional embedded space. We propose an incremental Noise-Reduction Isomap (NR-Isomap) approach to overcome the noise due to the occurring SCE problem. NR-Isomap is based on the HLLE method which uses GP kernels to reduce the noise in the Isomap. Since the HLLE method is an optimal nonlinear manifold learning, it becomes significant to maintain the SCE structure after denoising the data points [27]. Moreover, the other mainstream of our NR-Isomap is to compute the accuracy through GP Kernels and presents accurate results for big and high-dimensional datasets while reducing noise. Because the existing techniques cannot be used the GP kernels for the accuracy performance of Isomap. The main contributions of this paper can be summarized in three aspects:

1. We propose an incremental NR-Isomap method for Isomap. This method can remove noise more efficiently than Isomap and produce significantly noise-free results.
2. We compute the accuracy of the NR-Isomap using GP kernels and efficiently simulate and measure the results.
3. We calculate the time speed of NR-Isomap, which takes less time than Isomap.

The rest of the paper is organized as follows. In section 2, we present the detail of Isomap, Dijkstra and MDS. Section 3 proposed the NR-Isomap method with the HLLE algorithm and detailed the Gaussian process. We describe the results and discussion on the real-world datasets in Section 4. Finally, in Section 5, the paper is concluded.

II. Related Work

Tenenbaum et al. [13] proposed the Isomap, which is widely used for nonlinear dimensionality reduction. It can preserve the global geometry structure of the data points and map the data into a low-dimensional embedding [28]. It computes a geodesic distance between all data points in the nearest neighbourhood [29]. The geodesic distances are calculated as the shortest path distance in the closest neighbourhood data points by the algorithm of Dijkstra and Floyd's [30], and it generated the distance matrix [31]. The steps of the Isomap algorithm [23, 29] are given below:

Algorithm 1: Isomap

Step-1: If k nearest neighbour data points x_i, x_j an estimate the euclidean distance d between two knn data points, $d = (x_i, x_j) = || x_i - x_j||^2$ via Ɛ-ball algorithm.

Step-2: Compute the geodesic distance between all pairs of knn data points and estimate the shortest path distance $d_G\,(x_i, x_j)$ between knn data points through Dijkstra and Floyd's algorithms.

Step-3: It transformed a low d-dimensional embedding using the MDS algorithm, which preserves the global geometry manifold structure.

A. Dijkstra Algorithm

Dijkstra's algorithm can be used for two graph theory problems: all-pairs and single-source shortest path. It can find the shortest path through the smallest path value between given nodes. It generated the resultant matrix for the shortest path values. If x_i and x_j are knn data points, the geodesic distance can be determined by the shortest path $min[d_g(x_i, x_j)]$ the distance between x_i and x_j by using Eq. (1) [1, 2].

$$D_g(x_i, x_j) = min[D_g(x_i, x_j), D_G(x_i, x_n) + D_g(x_n, x_j)] \quad (1)$$

B. Multidimensional Scaling (MDS)

Torgerson [3] has proposed an MDS algorithm that can be used for data dimensionality reduction, manifold learning, and feature extraction [4]. MDS is used in several research areas, such as Data Mining, Cryptography, Multi-objective Optimization, Psychology, and Marketing. The basic idea of MDS is to preserve the dissimilarity and similarity of the data points in the low-dimensional embedding space [5]. It can embed the data points over the Euclidean distance, which is appropriate for discovering the global geometry structure and optimal data representation [2]. Isomap is a variant of the classical MDS to model nonlinear data points using their geodesic distance. It evaluates the low-dimensional coordinates by distance matrix $d_g(x_i, x_j)$ and transforms them into low- dimensional embedded data. It possibly preserved the geometry structure of the manifold [1, 6].

III. From Isomap to NR-Isomap

This section proposes an incremental NR-Isomap method for noise sensitivity due to the SCE problem; then, the Isomap algorithm often fails for big data. Choose a data point $X = \{x_i, x_j, \dots, x_N,\ X_i \in R^l (i = 1,2, \dots, N)\}\ X_i \in R^l (i = 1,2, \dots, N\ )$for a test. The main goal of our proposed method is to find noise-free data points in *l* low-dimensional space. First, we preprocessed the high dimensional datasets through the KPCA algorithm. Second, we applied the HLLE algorithm to remove noise from datasets. Grimes et al. [7] proposed the HLLE method for the SCE problem since Isomap fails in such a case. Therefore, instead of -ball algorithms, we used the HLLE algorithm and determined the noise-free knn data points. The HLLE algorithm can handle much higher dimensional and big data analysis problems than Isomap. In addition, we calculate the accuracy of the noise-free data set by GP kernel and measure highly accurate results for the large and high-dimensional data.

Algorithm 2: NR-Isomap

Step-1: We preprocess the data points using the KPCA algorithm.

Step-2: We apply the HLLE algorithm to the preprocessing data points to remove the noise from knn data points.

Step-3: Then, we repeat Isomap steps 3 and 4. Also, we map the data into low dimensional embedded data after denoising data points.

Step-4: In the end, we calculate the accuracy via GP Kernels and compute the time.

A. Hessian Locally Linear Embedding (HLLE)

Donoho et al. [7] proposed a Hessian embedding method. The Hessian embedding is also called the Hessian LLE method. The Hessian embedding framework is considered a modification of LLE, and its theoretical framework is viewed as a Laplacian eigenmaps method. Both methods' frameworks are very similar. The main difference is that the Hessian operator is used instead of the Laplacian operator. HLLE is assured to improve the correct manifold if the manifold is locally isometric and connected to the subset of Rd [7]. Suppose X is the dataset consisting of the N data point $X = \{x_1, x_2, \ldots, x_N\}\ X_i \in R^D (i = 1,2, \ldots, N)$ in high dimensional space and lying on a smooth manifold $M \in R^D$ of the intrinsic structure of dimensionality reduction d<D. Therefore, the Hessian embedding method aims to find the orthogonal coordinates on a tangent space to map an HLLE $f = M \to \Re^D\ f \in C^2(M)$ function. The hessian f at m in tangent coordinate can be used as the ordinary hessian of g in Eq. (2).

$$H_f^{tan}(m_{i,j}) = \frac{\delta^2 g(x)}{\delta x_i \delta x_j} | x = 0 \tag{2}$$

Although different local coordinates can give different tangent hessian coordinates, all hessian can share the Frobenius norm $H_f^{tan}(m_F^2)$ to follow the quadratic Eq. (3).

$$H(f) = \int_M H_f^{tan}(m_F^2) d_m \tag{3}$$

Where dm is a strictly measured positive probability over the manifold M and m is an observation point of the manifold. H(f) is measured as the average curvature of the function f on the manifold M. $g(x) = f(x)$ is defined as a function $(g = U \to \Re^d)$, where $U \in R^d$ a neighbourhood of the origin, m is a neighbour of observation space m. The HLLE algorithm is given below:

Algorithm 3: Hessian Locally Linear Embedding Algorithm

Step-1: Find the k Nearest Neighborhood
k neighbourhood or $\boldsymbol{\varepsilon}$ neighbourhood method for finding the knn of every data point X_i. It calculates the Euclidean distance of every knn data point.
Step-2: Extract Tangent Coordinates
Perform the PCA and Singular Value Decomposition (SVD) methods for obtaining the tangent coordinates from X_i data points.
Step-3: Construction of Hessian Estimate Function
Define the infrastructure of the hessian function f by using Least Square Estimator (LSE) method. f is a smooth hessian function $f = M \to R$.
Step-4: Construction of Hessian Quadratic Function
Construct the symmetric Hessian matrix Hij, having an ij pair of coordinates.
Step-5: Compute the Embedding of Hessian Estimate Function
Perform an eigenanalysis of Hessian H and identify a d+1 eigenvector corresponding to the smallest d+1 eigenvalue of knn.

B. Description of GP

In the machine learning area, GP is based on two theories included, Bayesian and Statistical Learning (SL). GP is appropriate for handling complex regression problems like Small Sample Size (SSS), Strong Generalization (SG), High-Dimensional, Classification, and Nonlinear. GP is a collection of random variables, and it extracts the finite mean and Covariance Function (CF) from the random variables. The CF is often known as a kernel in Gaussian Process Regression (GPR). It plays a curial role in the determined shape of the prior and posterior of the GPR. Rasmussen et al. [8] have described the detail of the GP method and CF in Neural Network Algorithms (NNA) [9, 10]. GP commonly uses various kernel functions, including Matern Kernel, Radial Basis Function (RBF) Kernel, Dot Product Kernel, Rational Quadratic Kernel, and White Kernel [11]. The detail of Kernels is given below:

1) Matern Kernel

Matern Kernel is derived from the RBF kernel. It has additional parameter v, which controls the smoothness of the resulting functions. Matern Kernel is parametrized by the length-parameter l>0, a vector or a scalar with the exact input x dimensions. Matern Kernel is given by:

$$(x_i, x_j) = \sigma^2 \frac{1}{\Gamma(v)2^{v-1}} \left(\gamma\sqrt{2vd}\left(\frac{x_i}{l}, \frac{x_j}{l}\right)\right)^v Kv\left(\gamma\sqrt{2vd}\left(\frac{x_i}{l}, \frac{x_j}{l}\right)\right) \tag{4}$$

2) RBF Kernel

RBF Kernel is a Gaussian kernel and is often called a squared exponential kernel, used to find a regression line and nonlinear classification problems. RBF Kernel is parametrized by the length-parameter l>0, a vector or a scalar with the exact input x dimensions. RBF Kernel is given by:

$$k(x_i, x_j) = exp\left(-\frac{1}{2} d\left(\frac{x_i}{l}, \frac{x_j}{l}\right)^2\right) \tag{5}$$

The RBF Kernel uses the GP as the covariance function to mean square derivatives of all orders are very smooth.

3) Dot Product (DP) Kernel

The DP Kernel is a non-stationary kernel. It can be gained from linear regression by insertion N (0,1) on the priority coefficient $x_d (d = 1,2, \ldots, D)$ and inserting $N(0, \sigma_0^2)$ it on an a priori basis. The rotation of the dot product kernel is unchanged for the origin coordinates. DP Kernel parametrized by the σ_0^2=0, which is called the homogenous linear kernel. DP Kernel is given by:

$$k(x_i, x_j) = \sigma_0^2 + x_i . x_j \tag{6}$$

4) Rational Quadratic (RQ) Kernel

The RQ Kernel is the mixture of an infinite sum of RBF kernels with different characteristics of length scale. RQ kernel parametrized by the length-parameter l>0 and the infinite kernel mixing parameter α>0, which l is the only scalar isotropic parameter. RQ Kernel is given by:

$$ptk(x_i, x_j) = \left(1 + \frac{d(x_i, x_j)^2}{2\alpha}\right)^{-\alpha} \tag{7}$$

5) White Kernel

The primary use of white kernel is as part of sum kernel where it explains the noise part of the signal. White Kernel is given by:

$$k(x_i, x_j) = \left.\begin{matrix} \text{noise-level} \\ 0 \end{matrix}\right\} \quad \begin{matrix} \text{if } x_i = x_j \\ \text{else} \end{matrix}$$

$$(x_i, x_j) = noise\text{–}level \text{ if } x_i = x_j\ else = 0 \tag{8}$$

Where the *noise-level* parameter is equal to the *noise of the variance.*

IV. Results and Discussions

In this section, we compared the seven SK-Learn datasets such as make_classification [12], make_hastie_10_2 [13], make_moon, make_circle and make_blob [14, 15], S_curve and Swiss roll [16, 17] datasets with noise NR Isomap results with Isomap. We calculated the accuracy of NR-Isomap and Isomap by cross-validating with GP kernels. We calculated the accuracy of only five datasets, including make_circle, make_classification, make_moon, make_hastie_10_2, and make_blob. We used 5K cross-validation to determine accuracy. In addition, GP kernels are used both for the algorithm and for measuring its performance.

A. Performance of Noise Reduction-Isomap in the Presence of Noise for Data Size (1000 and 1500)

TABLE I. Experimental Properties of Datasets

Number of Items	Description
Number of Sample Data Points	1000
Number of Sample Data Points	1500
Number of K Nearest Neighbours	10
Number of Dimensions	2
Standard Deviation Noise	0.15

As mentioned above, Isomap performance is not well-behaved in the presence of noise. We generated a make_circle, make_classification, make_moon, make_hastie_10_2 and make_blob, Swiss Roll and S_Curve with 1000 and 1500 data points for the NR-Isomap and Isomap. Table I shows the experimental properties of data sets. We first applied the PCA algorithm, added Gaussian noise in the data points (standard deviation noise = 0.15) and then used the HLLE algorithm with Dijkstra and MDS algorithms with fixed nearest neighbour k = 10. We also have the performance of NR-Isomap and Isomap on non-noisy and noisy make_circle, make_classification, make_moon, make_hastie_10_2 and make_blob, Swiss Roll and S_Curve datasets in Figs. 1 to 14.

1) make_circle Dataset

We tested the performance of NR-Isomap against Isomap for the make_circle dataset. Table I shows the brief detail of the make_circle dataset properties. We generated the circle with fixed Gaussian noise (standard deviation noise = 0.15), fixed k = 10, and data points = 1000 and 1500 in Fig. 1 and Fig. 2. We have presented the comparative results of the make_circle dataset manifold learning techniques and NR-Isomap. Figs 1 and 2 (A) Original circle data (1000 and 1500), (B) Isomap, (C) NR Isomap, (D) PCA, (E) Standard Locally Linear Embedding (SLLE) and (F) Spectral Embedding (SE). We have shown that the NR-Isomap method performs better than Isomap and manifold learning techniques in Figs. 1 and 2. The results of our method effectively reduced the noise of the circle data set instead of Isomap and manifold learning techniques.

2) make_blob Dataset

For the make_blob dataset, we tested the performance of NR-Isomap against Isomap and different learning techniques. Table I specified the short properties of the make_blob data and cannot support the noise value. For this reason, we used the make_blob dataset without Gaussian noise (standard deviation noise value). In Figs 3 and 4, we generated the blob with no noise value, with fixed k=10 and data points=1000 and 1500. Figs 3 and 4 (A) original blob data (1000 and 1500), (B) Isomap, (C) NR-Isomap, (D) PCA, (E) Standard Locally Linear Embedding (SLLE), and (F) Spectral Embedding (SE). Figs 3 and 4 show that the NR-Isomap method outperforms Isomap and manifold learning techniques and effectively reduces the noise of the blob dataset.

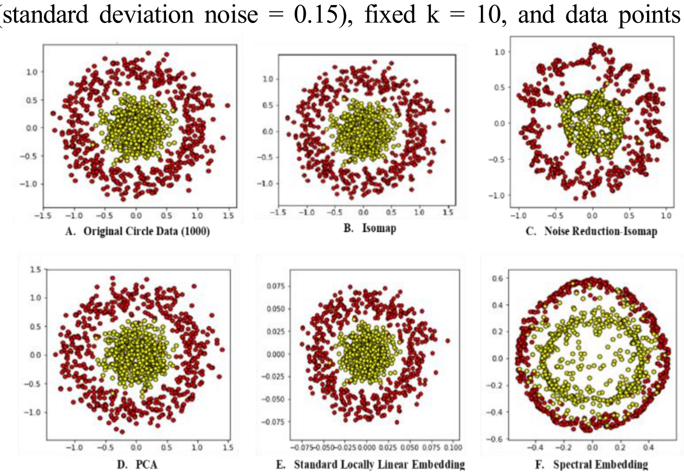

Fig. 1. Embedding Results of Circle Dataset with 1000 by NR-Isomap and Isomap

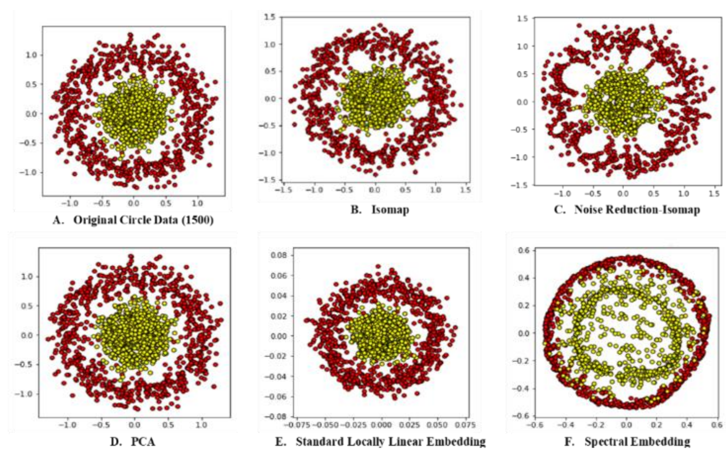

Fig. 2. Embedding Results of Circle Dataset with 1500 data points by NR-Isomap and Isomap

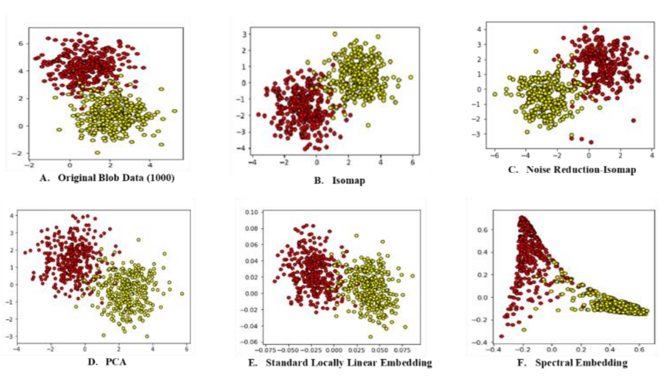

Fig. 3. Embedding Results of Blob Dataset with 1000 data points by NR-Isomap and Isomap

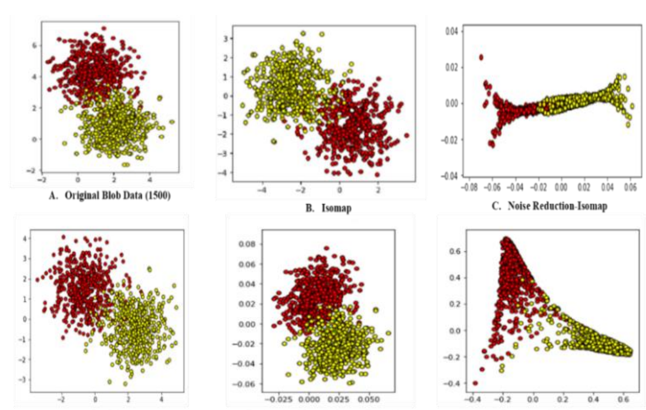

Fig. 4. Embedding Results of Blob Dataset with and 1500 data points by NR-Isomap and Isomap

3) make_classification Dataset

For the make_classification dataset, we have shown the performance of NR-Isomap compared to Isomap and manifold learning techniques. We followed the same properties of the make_blob record. Also, we used the make_classification dataset without Gaussian noise (standard deviation noise value). Figures 5 and 6 present the classification data with no noise value, fixed k=10 and data points=1000 and 1500. Fig. 5 and 6(A) Original classification data (1000 and 1500), (B) Isomap, (C) NR-Isomap, (D) PCA, (E) Standard Locally Linear Embedding (SLLE) and (F) Spectral Embedding (SE) Figs 5 and 6 show that the NR-Isomap method offers better performance than Isomap and manifold learning techniques, and efficiently reduces noise from the classification dataset.

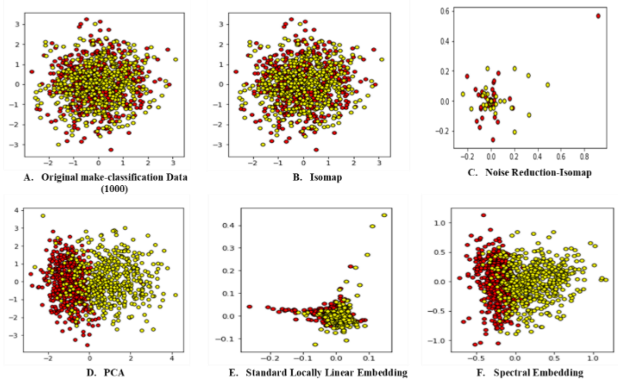

Fig. 5. Embedding Results of Make_classification Dataset with and 1000 data points by NR-Isomap and Isomap

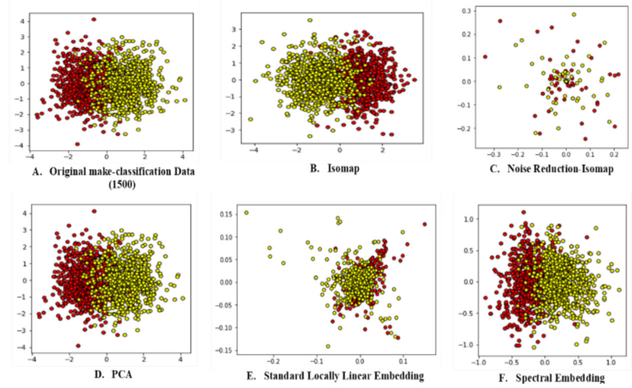

Fig. 6. Embedding Results of Make_classification Dataset with and 1500 data points by NR-Isomap and Isomap

4) make_moon Dataset

For the make_moon dataset, we demonstrated the performance of NR-Isomap versus Isomap and manifold learning techniques. We followed the same properties of the make_blob data. Also, we used the make_moon dataset without Gaussian noise (standard deviation noise value). In Figs 7 and 8, we have presented the moon data with no noise value, fixed k=10 and data points=1000 and 1500. Figs. 7 and 8(A) Original moon data (1000 and 1500), (B) Isomap, (C) NR-Isomap, (D) PCA, (E) Standard Locally Linear Embedding (SLLE) and (F) Spectral Embedding (SE). Figs 7 and 8 show that the NR-Isomap method offers better performance than Isomap and manifold learning techniques and efficiently reduces noise from the moon dataset.

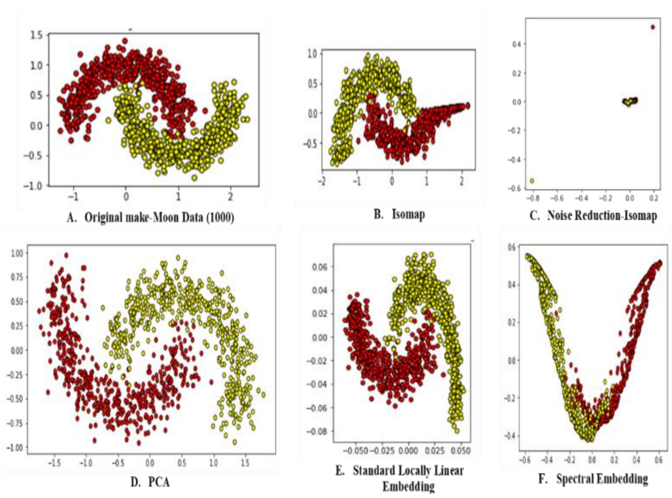

Fig. 7. Embedding Results of Make_moon Dataset with and 1000 data points by NR-Isomap and Isomap

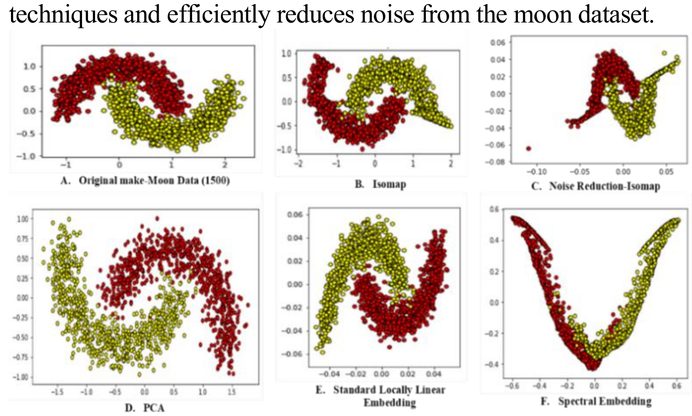

Fig. 8. Embedding Results of Make_moon Dataset with and 1500 data points by NR-Isomap and Isomap

5) make_hastie_10_2 Dataset

For the make_hastie_10_2 dataset, we presented the comparative performance of NR-Isomap versus Isomap and manifold learning techniques. We followed the same properties of the make_blob record. Also, it cannot support Gaussian noise (noise value of standard deviation). In Figs 9 and 10, we generated the hastie_10_2 data with no noise value, fixed k=10, and data points=1000 and 1500. According to Figs. 9 and 10(A) Original classification data (1000 and 1500), (B) Isomap, (C) NR-Isomap, (D) PCA, (E) Standard Locally Linear Embedding (SLLE) and (F) Spectral Embedding (SE). Figs 9 and 10 demonstrate the significant performance of NR-Isomap versus Isomap and manifold learning techniques, efficiently reducing noise from the hastie_10_2 dataset.

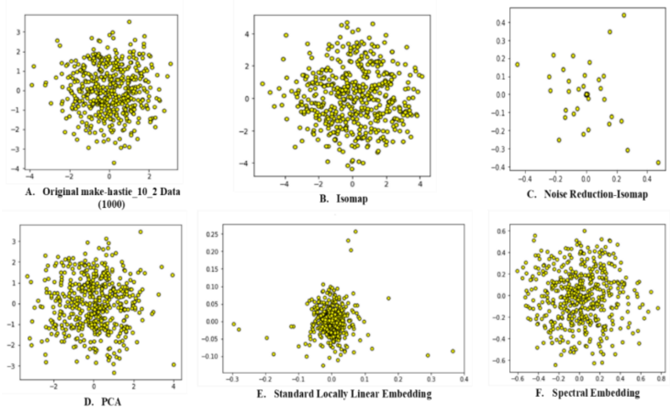

Fig. 9. Embedding Results of Make_hastie_10_2 Dataset with and 1000 data points by NR-Isomap and Isomap

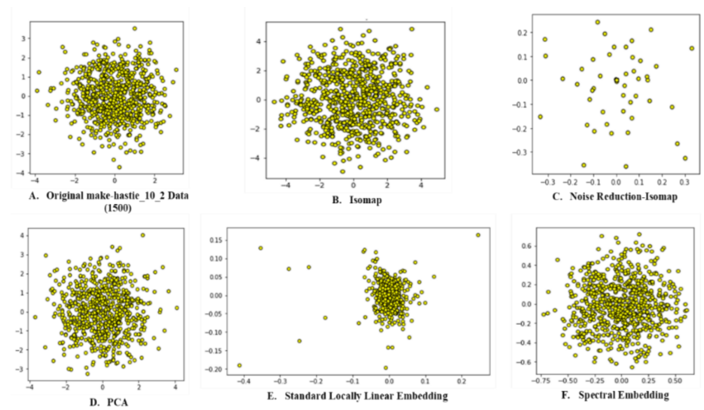

Fig. 10. Embedding Results of Make_hastie_10_2 Dataset with and 1500 data points by NR-Isomap and Isomap

6) Swiss Roll and S_curve Datasets

For Swiss Roll and S_Curve datasets, we tested the performance of INR-Isomap versus Isomap and several learning techniques. In Figs 11 to 14, we generated the Swiss Roll, and S_Curve fixed Gaussian noise (standard deviation noise = 0.15), fixed k = 10 and data points = 1000 and 1500 according to Table I. 11 to 14 presented the comparative performance of (A) original Swiss-Roll data (1000 and 1500), (B) Isomap, (C) INR-Isomap, (D) PCA, (E) Standard Locally Linear Embedding (SLLE), and (F) Spectral Embedding (SE) for Swiss Roll and S_Curve dataset. Figs 11 to 14 show that the INR-Isomap method outperforms Isomap and manifold learning techniques for 1000 and 1500 data points. The results of INR-Isomap significantly reduced the noise of the Swiss roll and the S_Curve dataset compared to Isomap and manifold learning techniques.

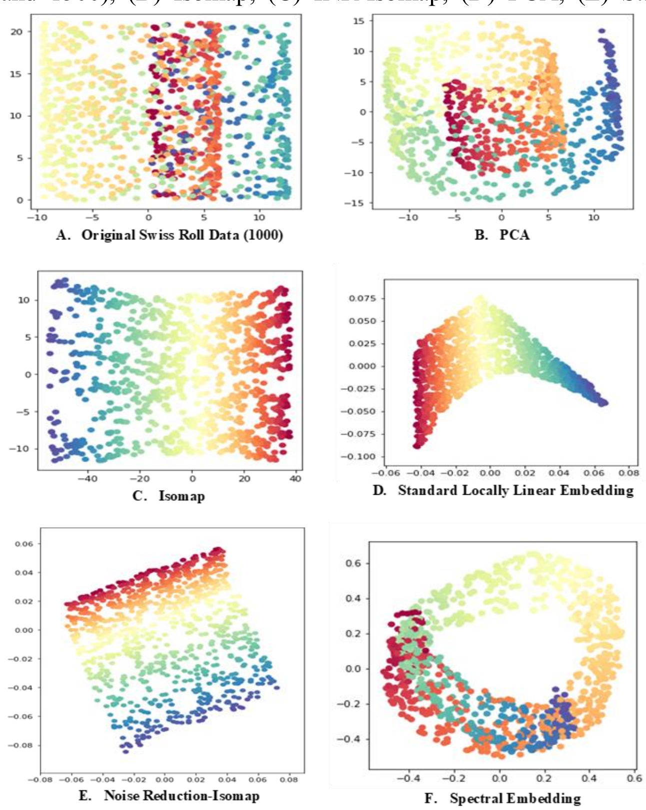

Fig. 11. Embedding Results of Swiss Roll Dataset with 1000 data points by NR-Isomap and Isomap

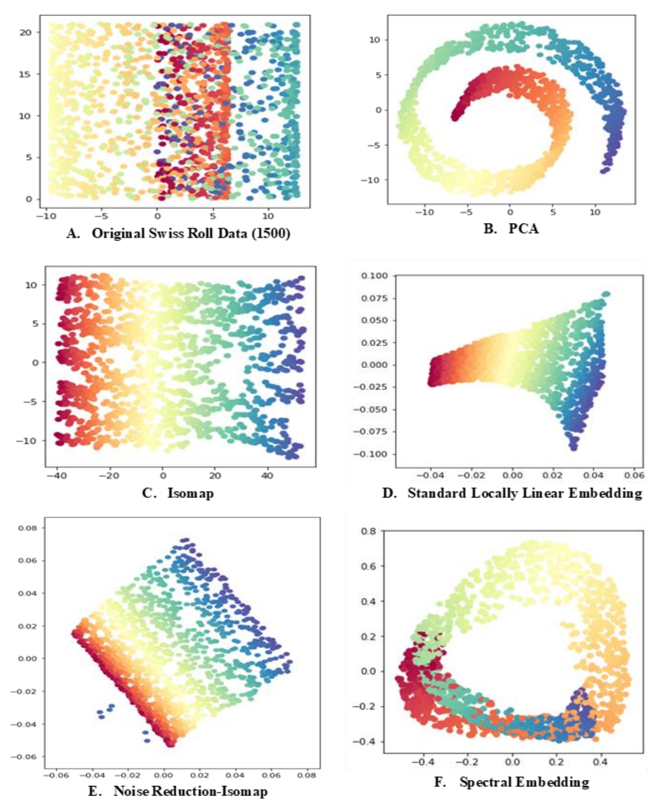

Fig. 12. Embedding Results of Swiss Roll Dataset with 1500 data points by NR-Isomap and Isomap

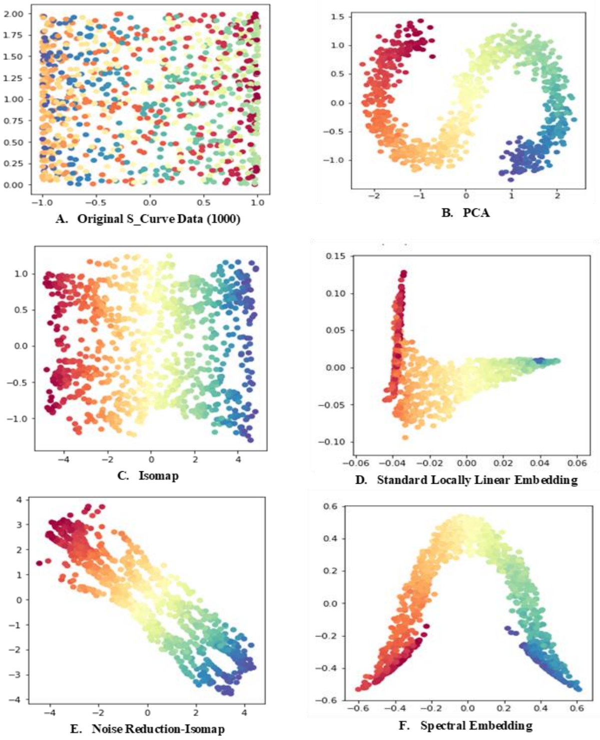

Fig. 13. Embedding Results of S_Curve Dataset with 1000 data points by NR-Isomap and Isomap

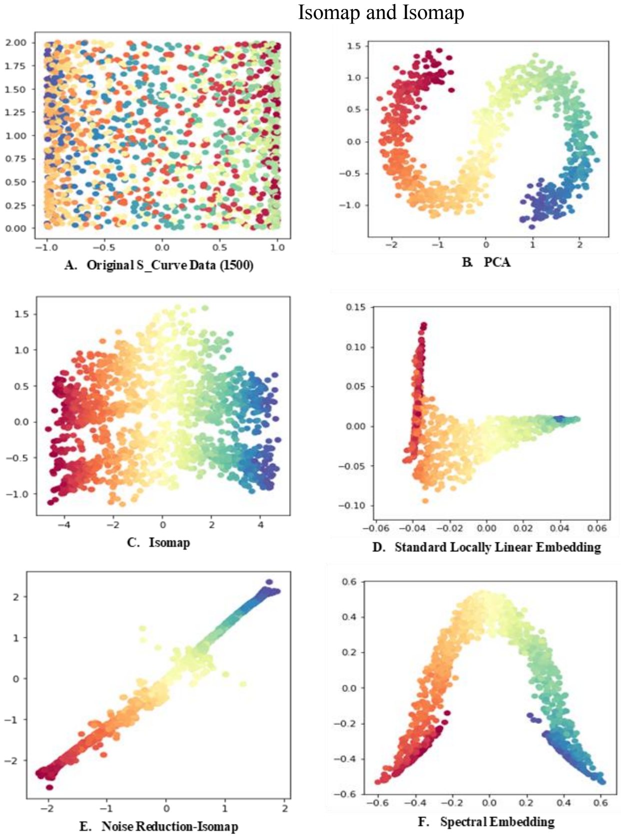

Fig. 14. Embedding Results of S_Curve Dataset with 1500 data points by NR-Isomap and Isomap

B. Performance of Accuracy and Time of NR-Isomap and Isomap

In this section, we calculated the accuracy and time required for the make_circle, make_blob, make_moon, make_classification, and make_hastie_10_2 datasets. We also compared NR-Isomap with Isomap. Table II shows the accuracy comparison of the five Gaussian kernels, including Matern, RBF, dot product (DP), rational quadratic (RQ), and white for 1000 data points. For the make_circle data, Matern Kernel achieved 0.98% accuracy, RBF 0.97%, DP 0.96%, RQ 0.95%, and White Kernel 0.94% accuracy. Although Isomap (Matern and RBF) was performed with an accuracy of 0.89% to 0.85%. For the make_moon data, Matern and RBF achieved 0.93% accuracy, DP and RQ 0.91%, and White kernel achieved 0.90% accuracy. While Isomap (Matern and RBF) achieved 0.91%, DP 0.90%, RQ and White have an accuracy of 0.89%. For the make_blob data, Matern and RBF have achieved 0.95% to 0.94% accuracy, DP 0.93%, and RQ and white core have achieved 0.92% accuracy. While Isomap (Matern and RBF) achieved an accuracy of 0.89% to 0.87%, DP and RQ achieved 0.86%, and White Kernel has achieved 0.85% accuracy. For make_hastie_10_2 data, Matern and RBF have 0.78%, DP and RQ 0.76% to 0.75%, and white kernel has achieved 0.74% accuracy. Although Isomap (Matern and RBF) achieved an accuracy of 0.66% to 0.65% with 0.70% to 0.69%, DP 0.68%, RQ and white kernels. For make_classification data, Matern and RBF are 0.85%, DP and RQ are 0.82% to 0.80%, and the white kernel has achieved 0.79% accuracy. In comparison, Isomap (Matern and RBF) achieved 0.78% to 0.77% accuracy with 0.82%, DP 0.80%, RQ, and white kernels. In addition, Fig. 15 shows the accuracy comparison performance of data size 1000 between NR-Isomap and Isomap with the five datasets. In Fig. 15 it can be seen that NR-Isomap can achieve higher accuracy performance over Isomap. NR-Isomap performs better than Isomap with Gaussian kernels.

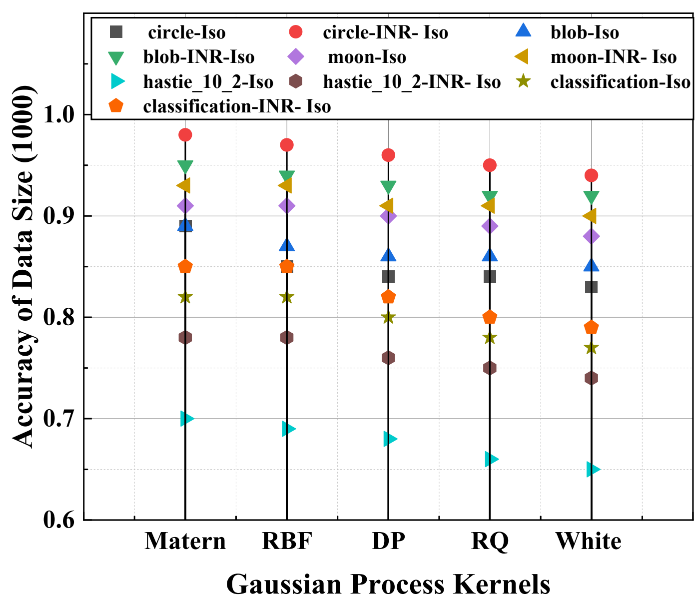

Fig. 15. Accuracy of Data Size (1000)

TABLE II. ACCURACY OF DATA-SIZE (1000)

Kernels	make_circle-Iso	make_circle-NR-Iso	make_blob-Iso	make_blob-NR-Iso	make_moon-Iso	make_moon-NR-Iso	make_hastie_10_2-Iso	make_hastie_10_2-NR-Iso	make_classification-Iso	make_classification-NR-Iso
Matern	0.89	0.98	0.89	0.95	0.91	0.93	0.70	0.78	0.82	0.85
RBF	0.85	0.97	0.87	0.94	0.91	0.93	0.69	0.78	0.82	0.85
DP	0.84	0.96	0.86	0.93	0.90	0.91	0.68	0.76	0.80	0.82
RQ	0.84	0.95	0.86	0.92	0.89	0.91	0.66	0.75	0.78	0.80
White	0.83	0.94	0.85	0.92	0.89	0.90	0.65	0.74	0.77	0.79

TABLE III. ACCURACY OF DATA SIZE (1500)

Kernels	make_circle-Iso	make_circle-NR-Iso	make_blob-Iso	make_blob-NR-Iso	make_moon-Iso	make_moon-NR-Iso	make_hastie_10_2-Iso	make_hastie_10_2-NR-Iso	make_classification-Iso	make_classification-NR-Iso
Matern	0.92	0.99	0.89	0.93	0.90	0.92	0.69	0.75	0.84	0.91
RBF	0.92	0.98	0.89	0.93	0.90	0.91	0.69	0.75	0.83	0.90
DP	0.87	0.98	0.88	0.92	0.89	0.91	0.67	0.74	0.82	0.90
RQ	0.87	0.95	0.86	0.90	0.88	0.90	0.65	0.72	0.82	0.89
White	0.85	0.94	0.86	0.90	0.86	0.90	0.60	0.70	0.80	0.87

According to Table III, we calculated the accuracy comparison of the five datasets with Gaussian kernels such as Matern, RBF, DP, RQ, and White for 1500 data points. For the make_circle data, Matern, RBF, and DP kernels achieved an accuracy of 0.99% to 0.98%, and the White kernel achieved an accuracy of 0.94%. However, Isomap achieved an accuracy of 0.92% for both Matern and RBF kernels. For make_blob data, NR-Isomap achieved 0.93% accuracy (Matern and RBF kernels), and DP kernel has achieved 0.92% accuracy against Isomap. For make_moon data, NR-Isomap achieved an accuracy of 0.92% to 0.91% (Matern, RBF, and DP kernels), and the RQ kernel achieved 0.90% accuracy against Isomap of 0 .90% to 0.86%. For the make_hastie_10_2 data, Matern and RBF kernels have achieved 0.75% accuracy, and the white kernel has achieved 0.70% accuracy. However, Isomap has achieved an accuracy of 0.69% to 0.60% for matrices, RBF, and white kernels. For make_classification data, NR-Isomap achieved an accuracy of 0.91% to 0.87% compared to Isomap accuracy (0.84% to 0.80%). From Fig. 16, it can be seen that NR-Isomap can achieve higher accuracy than Isomap when the data size is 1500. As the number of data points increases, the accuracy of Isomap decreases. NR-Isomap outperforms Isomap with Gaussian kernels for both data points (1500, 1000).

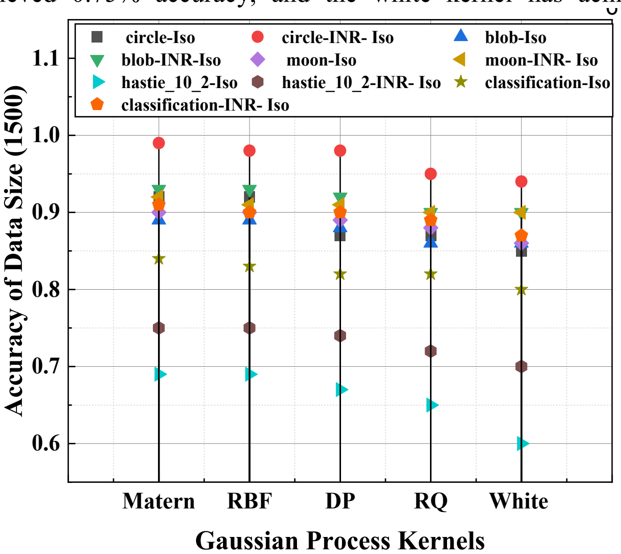

Fig. 16. Accuracy of Data Size (1500)

TABLE IV. TIME COMPARISONS BETWEEN NR-ISOMAP AND ISOMAP

DATASETS	ISOMAP-(1500)	NR-ISOMAP-(1500)	ISOMAP-(1000)	NR-ISOMAP-(1000)
MAKE_CIRCLE	5.5	0.2	4.5	2.2
MAKE_BLOB	10.5	7.5	3.5	2.6
SWISS ROLL	7.1	6.3	3.2	2.2
S_CURVE	5.2	4.7	3.5	2.5
MAKE_HASTIE_10_2	2.9	1.15	0.11	0.9
MAKE_CLASSIFICATION	10.5	0.24	0.81	0.2
MAKE_MOON	11.6	0.5	0.75	1.8

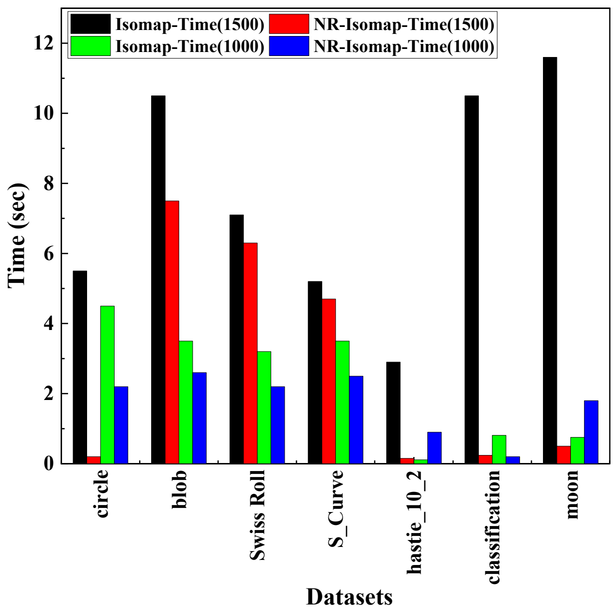

Fig. 17. Time Comparisons between NR-Isomap and Isomap

In Table IV, we have presented the time comparison between NR-Isomap and Isomap with seven datasets. According to 1500 data points, NR-Isomap time speed is faster than Isomap for the datasets make_circle (0.2), S_Curve (0.4), and make_moon (0.5). NR-Isomap generated the circle within 0.2 compared to Isomap. The time of the make_circle dataset tends to be the lowest compared to the other datasets. For 1000 data points, NR-Isomap time rate is significantly slower than Isomap. NR-Isomap generated the circle and Swiss roll within 2.2 seconds compared to Isomap. make_classification and make_moon times (0.2 and 1.8) are faster than other datasets. However, Isomap time speed is faster at 1000 data points than at 1500 data points. Fig. 17 shows the comparative time performance between NR-Isomap and Isomap with seven-datasets. We can see that the NR-Isomap time is quite good than Isomap. Moreover, NR-Isomap can reduce the time speed and provide accurate results compared with Isomap. Our NR-Isomap is faster and saves time, especially for large and high-dimensional datasets.

V. Conclusion

The major limitation of Isomap is noise sensitivity due to the SCE problem. We introduced an incremental NR Isomap based on the HLLE algorithm with GP kernels. The HLLE algorithm produces optimal results for the noisy big data sets and reduces the SCE problem more efficiently than Isomap. In the experimental results, we calculate the accuracy performance of the NR-Isomap method based on the five GP kernels. We compared five GP kernels with NR-Isomap and Isomap. Our proposed approach works better with GP kernel, k and Gaussian noise values than Isomap. Moreover, GP kernels are used both for the algorithm and for measuring its performance. Also, we estimate the time comparison between NR-Isomap and Isomap. The time of NR Isomap is several times lower than that of Isomap. In future work, we will focus on deep learning methods.

References

[1] C. K. Williams and C. E. Rasmussen, Gaussian processes for machine learning (no. 3). MIT press Cambridge, MA, 2006.

[2] M. Kuss, C. E. Rasmussen, and R. Herbrich, "Assessing Approximate Inference for Binary Gaussian Process Classification," Journal of machine learning research, vol. 6, no. 10, 2005.

[3] C. E. Rasmussen, "Evaluation of Gaussian processes and other methods for non-linear regression," University of Toronto Toronto, Canada, 1997.

[4] A. G. Wilson, D. A. Knowles, and Z. Ghahramani, "Gaussian process regression networks," arXiv preprint arXiv:1110.4411, 2011.

[5] A. Wilson and R. Adams, "Gaussian process kernels for pattern discovery and extrapolation," in International conference on machine learning, 2013, pp. 1067-1075: PMLR.

[6] M. Balasubramanian and E. L. Schwartz, "The isomap algorithm and topological stability," Science, vol. 295, no. 5552, pp. 7-7, 2002.

[7] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," presented at the Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, Vancouver, British Columbia, Canada, 2001.

[8] Y. Fu and T. S. Huang, "Locally linear embedded eigenspace analysis," IFP-TR, Univ. of Illinois at Urbana-Champaign, 2005.

[9] I. T. Jolliffe, "Principal components in regression analysis," in Principal component analysis: Springer, 1986, pp. 129-155.

[10] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," science, vol. 290, no. 5500, pp. 2323-2326, 2000.

[11] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," SIAM journal on scientific computing, vol. 26, no. 1, pp. 313-338, 2004.

[12] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 3, pp. 328-340, 2005.

[13] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," science, vol. 290, no. 5500, pp. 2319-2323, 2000.

[14] Y. Michels, E. Baudrier, L. Mazo, and M. Tajine, "Density based graph denoising for manifold learning," 2019.

[15] A. Saxena, A. Gupta, and A. Mukerjee, "Non-linear dimensionality reduction by locally linear isomaps," in International Conference on Neural Information Processing, 2004, pp. 1038-1043: Springer.

[16] G. Rosman, A. Bronstein, M. Bronstein, and R. Kimmel, "Manifold analysis by topologically constrained isometric embedding," International Journal of Applied Mathematics and Computer Sciences, vol. 1, no. 3, pp. 117-123, 2004.

[17] H. Chang and D.-Y. Yeung, "Robust locally linear embedding," Pattern recognition, vol. 39, no. 6, pp. 1053-1065, 2006.

[18] H. Choi and S. Choi, "Robust kernel isomap," Pattern Recognition, vol. 40, no. 3, pp. 853-862, 2007.

[19] H. Choi and S. Choi, "Kernel isomap," Electronics letters, vol. 40, no. 25, pp. 1612-1613, 2004.

[20] H. Chen, G. Jiang, and K. Yoshihira, "Robust nonlinear dimensionality reduction for manifold learning," in 18th International Conference on Pattern Recognition (ICPR'06), 2006, vol. 2, pp. 447-450: IEEE.

[21] B. Li, D.-S. Huang, and C. Wang, "Improving the robustness of isomap by de-noising," in 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 266-270: IEEE.

[22] W. Chen and W. Mao, "An Improved Isomap Algorithm Based on LP-Centers," in 2015 International Conference on Artificial Intelligence and Industrial Engineering, 2015: Atlantis Press.

[23] C. Du, S. Zhou, J. Sun, and J. Zhao, "Robust ISOMAP Based on Neighbor Ranking Metric," in International Conference on Intelligent Computing, 2012, pp. 221-229: Springer.

[24] M. Yousaf, M. S. S. Khan, T. U. Rehman, S. Ullah, and L. Jing, "NRIC: A Noise Removal Approach for Nonlinear Isomap Method," Neural Processing Letters, 2021/03/08 2021.

[25] S. Ding, C. A. Keal, L. Zhao, and D. Yu, "Dimensionality reduction and classification for hyperspectral image based on robust supervised ISOMAP," Journal of Industrial and Production Engineering, vol. 39, no. 1, pp. 19-29, 2022.

[26] A. C. Neto and A. L. Levada, "ISOMAP-KL: a parametric approach for unsupervised metric learning," in 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2020, pp. 287-294: IEEE.

[27] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," Proceedings of the National Academy of Sciences, vol. 100, no. 10, pp. 5591-5596, 2003.

[28] F. J. Orts Gomez, G. Ortega Lopez, E. Filatovas, O. Kurasova, and G. E. M. Garzón, "Hyperspectral image classification using Isomap with SMACOF," Informatica, vol. 30, no. 2, pp. 349-365, 2019.

[29] Y.-K. Lei, Y. Xu, S.-W. Zhang, S.-L. Wang, and Z.-G. Ding, "Fast ISOMAP based on minimum set coverage," in International Conference on Intelligent Computing, 2010, pp. 173-179: Springer.

[30] S. Hougardy, "The Floyd–Warshall algorithm on graphs with negative cycles," Information Processing Letters, vol. 110, no. 8-9, pp. 279-281, 2010.
[31] T. Qu and Z. Cai, "An improved Isomap method for manifold learning," International Journal of Intelligent Computing and Cybernetics, vol. 10, no. 1, pp. 30-40, 2017.
[32] W. S. Torgerson, "Multidimensional scaling: I. Theory and method," Psychometrika, vol. 17, no. 4, pp. 401-419, 1952.
[33] B. Ghojogh et al., "Feature selection and feature extraction in pattern analysis: A literature review," arXiv preprint arXiv:1905.02845, 2019.
[34] R. Beals, D. H. Krantz, and A. Tversky, "Foundations of multidimensional scaling," Psychological review, vol. 75, no. 2, p. 127, 1968.
[35] M. L. Stein, Interpolation of spatial data: some theory for kriging. Springer Science & Business Media, 1999.
[36] K. V. Subrahmanyam et al., "Prediction of heavy rainfall days over a peninsular Indian station using the machine learning algorithms," Journal of Earth System Science, vol. 130, no. 4, pp. 1-9, 2021.
[37] X. Liang, "Supervised and Unsupervised Learning Models," in Social Computing with Artificial Intelligence: Springer, 2020, pp. 27-82.
[38] I. Guyon, "Design of experiments of the NIPS 2003 variable selection benchmark," in NIPS 2003 workshop on feature extraction and feature selection, 2003, vol. 253, p. 40.
[39] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction. Springer, 2009.
[40] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," the Journal of machine Learning research, vol. 12, pp. 2825-2830, 2011.
[41] O. Kramer, "Cascade support vector machines with dimensionality reduction," Applied Computational Intelligence and Soft Computing, vol. 2015, 2015.
[42] J. B. Tenenbaum, "Mapping a manifold of perceptual observations," in Advances in neural information processing systems, 1998, pp. 682-688.
[43] T. Qu and Z. Cai, "A Fast Isomap Algorithm Based on Fibonacci Heap," in International Conference in Swarm Intelligence, 2015, pp. 225-231: Springer.