ISOMAP: A Dimensionality Reduction Technique

Bushra Haque[1], Yichen Ji[1], and Luis Sierra Muntané[1]

[1]Department of Statistical Sciences, University of Toronto

April 4, 2025

Contents

0.1 Contributions

- Yichen Ji: introduction, manifold learning and MDS
- Bushra Haque: simulations, discussion and algorithm design
- Luis Sierra: optimal embedding, analysis, results, discussion and simulations

1 Introduction

Real-world data is often nonlinear in nature: the underlying data generation processes — such as physical systems, biological mechanisms, or human behavior — tend to involve complex, nonlinear interactions among latent variables. The manifold hypothesis posits that many high-dimensional datasets intrinsically lie along low-dimensional manifolds embedded within the higher-dimensional space; that is, although observations exist in high-dimensional ambient spaces, the intrinsic degrees of freedom governing the data generation process are often much fewer (Tenenbaum et al. (2000)). For example, facial image data may contain thousands of pixels in the original input space, but the variation across images is largely governed by a small set of latent factors such as identity, pose, lighting, and expression (Yang (2002)).

Under the manifold hypothesis, it is common practice to assume that high-dimensional data lies on or near a low-dimensional manifold. This assumption helps mitigate the curse of dimensionality, which otherwise results in data sparsity and poor generalization in high-dimensional spaces (Wagaman and Levina (2009); Choi and Choi (2007)). By conducting dimensionality reduction techniques or constraining the model hypothesis space to a lower-dimensional structure, we can achieve more interpretable representations, better generalization, and more tractable optimization, which is crucial for algorithms such as PCA, t-distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP) (Lee and Verleysen (2007); McInnes et al. (2020)).

In fact, not all dimensionality reduction techniques approach the manifold structure in the same way. On the one hand, classical linear approaches like Principal Component Analysis (PCA) and Multi-Dimensional Scaling (MDS) demonstrate the limitations of linear assumptions when applied to nonlinear data: PCA projects input data onto a linear manifold (e.g. hyperplane) that maximizes variance based on the eigenstructure of the data covariance matrix, while MDS preserves pairwise distances, which, under Euclidean metrics, is equivalent to PCA, but fails to account for nonlinear manifold curvature. In contrast, nonlinear techniques such as t-SNE and UMAP focus on preserving local neighborhood structure. These methods are effective at uncovering local clusters and improving data visualization, but often

at the cost of distorting global structures or lacking formal guarantees of global optimality. In addition, kernel PCA extends traditional PCA by applying the kernel trick to capture nonlinear structures in data and projecting it into a high-dimensional feature space before performing linear PCA. But still, kernel PCA depends significantly on the chosen kernel function to represent nonlinearity and does not explicitly preserve manifold geometry (Williams (2000); Anowar et al. (2021)).

The objective of this project is to investigate Isometric Feature Mapping (ISOMAP) as a nonlinear method for obtaining low-dimensional embeddings that preserve the global geometric relationships of data points lying on a general manifold. ISOMAP distinguishes itself by aiming for a faithful preservation of the global geometry. By accounting for all pairs of points via geodesic distances, it attempts to reflect the true low-dimensional global geometry of the data manifold, which complements the local focus of t-SNE and UMAP. ISOMAP naturally extends MDS by generalizing the notion of distance to be the shortest path distance instead of the direct Euclidean distance used in the latter method. As a consequence, ISOMAP inherits the desirable properties of MDS, including global optimality, asymptotic convergence, and polynomial time algorithms.

This paper is organized as follows. Section 2 begins with an overview of manifold learning, followed by a discussion of optimal embedding, the role of multidimensional scaling, and the design of the ISOMAP algorithm. Section 3 presents experimental studies on simulated data and real-world datasets including Swiss Roll, Stanford Faces and MNIST, along with a detailed analysis of the results. Section 4 offers a discussion of the findings and highlights key limitations. Additional details and results for experiments are provided in Appendix 1.

2 Methodology

2.1 Manifold Learning

Let $\mathcal{M}$ be a smooth compact Riemannian sub-manifold embedded in $\mathbb{R}^N$, i.e., $\mathcal{M} \hookrightarrow \mathbb{R}^N$ with manifold dimension $\dim(\mathcal{M}) = d \ll N$. The Riemannian metric g defined on $\mathcal{M}$ induces a geodesic distance between any two points $x_i, x_j \in \mathcal{M}$, given by the length of the shortest path lying entirely on the manifold, denoted as $d_{\mathcal{M}}(x_i, x_j)$. This distance reflects the true geometry of $\mathcal{M}$ and, in general, differs from the Euclidean distance $\|x_i - x_j\|$ computed in $\mathbb{R}^N$, especially when $\mathcal{M}$ is curved and data points are far away from each other.

Manifold learning aims to uncover the topological properties of the original data by constructing a mapping $\Phi : \mathcal{M} \to \mathbb{R}^d$ that approximately preserves geodesic distances:

$$\|\Phi(x_i) - \Phi(x_j)\| \approx d_{\mathcal{M}}(x_i, x_j),$$

for all — or at least nearby - pairs of data points $x_i, x_j \in \mathcal{M}$. This approach enables faithful low-dimensional representations that retain the manifold's essential structure.

In practice, we do not observe the continuous manifold $\mathcal{M}$ directly, but rather a finite set of sample points $\{x_i\}_{i=1}^n \subset \mathcal{M} \subset \mathbb{R}^N$. As such, we must approximate the intrinsic geodesic distance $d_{\mathcal{M}}(x_i, x_j)$ between points using only the observed data. A possible approach is to construct a weighted graph $G = (V, E)$, where each vertex corresponds to a data point and edges connect neighboring points with weights based on local Euclidean distances. The geodesic distance is then approximated by the shortest path distance $d_G(x_i, x_j)$ over the graph:

$$d_{\mathcal{M}}(x_i, x_j) \approx d_G(x_i, x_j),$$

and as the graph gets denser, the hope is that $d_G(x_i, x_j)$ will converge to the geodesic distance over the manifold.

2.2 Multidimensional Scaling

Early formulations of Multidimensional Scaling (MDS), as introduced in works such as Torgerson (1952); Shepard (1980); Ramsay (1982), provide a framework for constructing a configuration of points in a low-dimensional Euclidean space such that their pairwise distances approximate a given dissimilarity matrix.

Let $D \in \mathbb{R}^{n \times n}$ be a symmetric matrix of pairwise dissimilarities between n samples, where D_{ij} denotes the distance between data points x_i and x_j. Classical MDS assumes these dissimilarities arise from squared Euclidean distances and aims to find embedded points

$y_1, \ldots, y_n \in \mathbb{R}^d$ such that $\|y_i - y_j\|^2 \approx D_{ij}^2$. In Euclidean space, we have a direct relationship between squared distances and inner products, and we need the inner products to recover the embedding:

$$\|y_i - y_j\|^2 = \langle y_i, y_i \rangle + \langle y_j, y_j \rangle - 2 \langle y_i, y_j \rangle .$$

For centered data, squaring the distances allows us to work backward from distances to inner products via double centering. To achieve this goal, MDS applies a double-centering transformation to the element-wise squared dissimilarity matrix $D^{(2)} = \left[D_{ij}^2\right]_{i,j=1}^n$. Let $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ denote the double-centering matrix, where $\mathbf{1}$ is the column vector of ones. The Gram matrix is given by:

$$B = -\frac{1}{2} H D^{(2)} H. \tag{2.1}$$

This matrix operation centers the data in the low-dimensional space and expresses inner products between the embedded points. Since $B = YY^\top$ for some $Y \in \mathbb{R}^n$, we recover the embedding by conducting eigen-decomposition $B = V\Lambda V^\top$ and constructing the low-dimensional embedding $Y = V_d \Lambda_d^{\frac{1}{2}}$ where Λ_d contains the top d eigenvalues and V_d the corresponding eigenvectors, and each row of Y gives the coordinates of a data point in the d-dimensional Euclidean space. Therefore, the resulting low-dimensional coordinates preserve the global manifold structure encoded by the original dissimilarity measures.

2.3 Optimal Embedding

A discrete set of points cannot uniquely determine an arbitrary manifold without strong additional assumptions on said manifold's structure. However, a natural option is to find the manifold with the lowest curvature that fits the available datapoints $\{x_i\}_{i=1}^n \subset \mathcal{M}$. One can use a graph with the points as vertices to construct a discrete approximation of said manifold so that the metric from the original manifold is approximated by a distance on the resulting graph in terms of the length of the path connecting the two points. In this section we will provide some intuition for why these ideas work to give us an optimal embedding of our data.

When dealing with connected manifolds, the approximating graph should also be connected, or else the corresponding path distance on the graph would be infinite. This can pose certain conditions on the hyperparameters of the algorithm, as these determine how the graph is constructed.

Firstly, we will explore why minimizing the Laplacian operator over the manifold produces a desirable embedding.

Theorem 2.1. *For a smooth compact, second countable d-dimensional (Riemannian) man-*

ifold $\mathcal{M}$, its optimal embedding $f : \mathcal{M} \to \mathbb{R}^d$ is given locally by the solution to the problem

$$\arg\max_{f \in \mathcal{F}} \int_{\mathcal{M}} \Delta_{\mathcal{M}}(f) f,$$

where we assume some metric $g = d_{\mathcal{M}}$ over the manifold that allows us to define the volume element for integration.

The operator $\Delta_{\mathcal{M}}$ is often referred to as the Laplace-Beltrami operator, named as such to distinguish it from the regular Laplacian (the divergence of the gradient), for whom it is its analogue for a general Riemannian manifold $\mathcal{M}$. This means that when the metric g is the standard Euclidean metric, $\Delta_{\mathcal{M}}$ reduces to the standard Laplacian. For a proof sketch, consider two points $x, z \in \mathcal{M}$ that are sufficiently close together. Let their geodesic distance on the manifold be $l = \text{dist}_{\mathcal{M}}(x, z)$. We wish to find an embedding f that optimally preserves locality under some fixed constraints, i.e. such that points that are close together in the manifold are also close together under the image by f. Define a geodesic path $c : I \subset \mathbb{R} \longrightarrow \mathcal{M}$ with an arc-length parametrization such that $c(0) = x$ and $c(l) = z$. We can express their distance under the embedding f as

$$|f(z) - f(x)| = \int_0^l \langle \nabla f(c(t)), c'(t) \rangle \, \mathrm{d}t \leq l \|\nabla f(c(t))\|,$$

where the inequality comes from Cauchy-Schwarz. This means that distances in the image of f depend up to first order on the gradient of the embedding. In fact we can write

$$f(z) = f(x) + l\|\nabla f(c(t))\| + o(l).$$

From this, by the smoothness assumption we can seek to find an embedding $f^\star$ with minimal norm of its gradient, so that

$$f^\star = \arg\min \int_{\mathcal{M}} \|\nabla f\|^2. \tag{2.2}$$

Now recall Stokes' Theorem where for a vector field X, for vanishing boundary conditions we have that $\int \langle X, \nabla f \rangle = -\int \text{div}(X) f$ and so by setting $X = \nabla f$ we can write

$$\int_{\mathcal{M}} \|\nabla f\|^2 = -\int_{\mathcal{M}} \Delta(f) f,$$

from which we obtain the alternative form of equation (2.2) as

$$f^\star = \arg\max \int_{\mathcal{M}} \Delta(f) f, \tag{2.3}$$

where the maximization can be taken over some prescribed class of functions $\mathcal{F}$. Such a class in machine learning articles is often an L^p or $\mathcal{C}^\infty$ class over the manifold.

In order to translate this functional analysis problem into a discrete, data-friendly domain, we require a sufficiently dense set of points $\{x_i\}_{i=1}^n \subset \mathcal{M}$ in the sense that as $n \to \infty$ and $\varepsilon \to 0$, we have that

$$n\varepsilon^{d/2} \longrightarrow \infty,$$

ensuring an appropriate asymptotic density of points in a ball of radius $\sqrt{\varepsilon}$. Without loss of generality, consider the kernel function and corresponding matrix

$$k_\varepsilon(x, y) = \exp\left(-\frac{\|x - y\|^2}{\varepsilon}\right), \qquad W_{ij} = k_\varepsilon(x_i, x_j),$$

so that the degree in the complete graph with vertices $V = \{x_i\}_{i=1}^n$ and edge weights W_{ij} in the limit behaves as

$$d_i = \sum_{j=1}^n W_{ij}, \qquad d(x) \to \int_{\mathcal{M}} k_\varepsilon(x, y)p(y)\mathrm{d}V(y),$$

for $p(y)$ the sampling distribution over $\mathcal{M}$. If we expand the local averaging by Taylor up to second order, we see that

$$\int_{\mathcal{M}} k_\varepsilon(x, y)f(y)p(y)\mathrm{d}V(y) = f(x)m_0(x) + \varepsilon\frac{m_2(x)}{2}\Delta_{\mathcal{M}}f + \mathcal{O}(\varepsilon^{3/2}),$$

for m_0 and m_2 the zero and second order moments. Given the symmetry of the kernel, the odd moments are equal to zero. This allows us to conjecture a result of the kind of

$$\lim_{\varepsilon\to 0}\lim_{n\to\infty}\frac{1}{\varepsilon}\left(\frac{\sum_{j=1}^n k_\varepsilon(x_i, x_j)(f(x_i) - f(x_j))}{\sum_{j=1}^n k_\varepsilon(x_i, x_j)}\right) = \Delta_{\mathcal{M}}f(x_i).$$

In order to solve the problem explicitly, we define the Laplacian L of a graph $\mathcal{G}$, and will be analogous to the Laplacian operator on manifolds.

Definition 2.2. *Given a graph $\mathcal{G} = (V, E)$, its Laplacian L is given by*

$$L = D - A,$$

which for a weighted graph with weights $w : E \longrightarrow [-1, 1]$ *reads as*

$$L_{u,v} = \begin{cases} 1 - \frac{w(v,v)}{d_v}, & \text{if } u = v \text{ and } d_u \neq 0 \\ \frac{-w(u,v)}{\sqrt{d_u d_v}}, & \text{when } u \sim v \\ 0, & \text{otherwise.} \end{cases}$$

The Laplacian can be naturally generalized from a matrix to an operator in the space of functions $g : V(\mathcal{G}) \to \mathbb{R}$ as

$$\mathcal{L}\left[g(u)\right] = \frac{1}{\sqrt{d_u}} \sum_{\substack{v \\ v \sim u}} \left(\frac{g(u)}{d_u} - \frac{g(v)}{d_v} \right),$$

which lets us consider the vertices as the images of some elements by a function f, as is the case with our embedding. Again, without loss of generality, we can select specific weights for the edges yielding a form of the Laplacian that makes the proof of convergence easier. This involves the monotone transformation of the Euclidean distance between points defined by the exponential function. The Laplacian with such weights used in Belkin and Niyogi (2008) is of the form

$$L_n^t f(x_i) = f(x_i) \sum_{j=1}^{n} e^{-\frac{\|x_i - x_j\|^2}{4t}} - \sum_{j=1}^{n} f(x_j) e^{-\frac{\|x_i - x_j\|^2}{4t}}. \tag{2.4}$$

In fact, the authors prove a result showing uniform convergence in probability of the graph Laplacian to the Laplace-Beltrami operator.

Theorem 2.3. *(Belkin and Niyogi, 2008, Theorem 2) Let* $\{x_i\}_{i=1}^n \subset \mathcal{M}$ *be sampled over a uniform distribution on a d-dimensional compact manifold* $\mathcal{M} \subset \mathbb{R}^n$*. Let* $\mathcal{F}$ *be the function space* $\mathcal{C}^\infty(\mathcal{M})$*, such that* Δf *is Lipschitz. Then there exists a sequence of real numbers* $t_n \to 0$ *and* $C > 0$ *such that*

$$\lim_{n \to \infty} \sup_{f \in \mathcal{F}} \left\| C \frac{(4\pi t_n)^{-\frac{d+2}{2}}}{n} L^{t_n} f - \Delta_{\mathcal{M}} f \right\| = 0 \tag{2.5}$$

in probability, where L_n^t *is defined as in equation* (2.4)*.*

With this result, we justify the asymptotic relationship

$$-\Delta_{\mathcal{M}} \longleftrightarrow L(\mathcal{G}),$$

even though we made the strong assumption that the points were sampled from $\mathcal{M}$ uniformly, without any account of the possible N-dimensional noise that often appears in practice. We believe that this formal analysis is beyond the scope of this project.

2.4 Algorithm Design

The algorithm has two main variants corresponding to the method used to construct the neighbourhood graph, namely k-ISOMAP and ε-ISOMAP. Although the original paper suggests that either the standard Euclidean metric or a domain-specific metric can be used to compute the pairwise dissimilarities between samples, the following algorithms and experiments in section 3 are based on the Euclidean distance (Tenenbaum et al. (2000)).

Algorithm 1 k-ISOMAP

Input: data $X \in \mathbb{R}^{n \times N}$, neighbours k, dimension d
Output: embedding $Y \in \mathbb{R}^{n \times d}$
Step 1: Construct Neighbourhood Graph
Initialize $\mathbf{G} = \mathbf{0} \in \mathbb{R}^{n \times n}$
Set $G_{ij} = \|x_i - x_j\|_2$ $\forall$ pairs of points (x_i, x_j) if x_j is a k-nearest neighbour of x_i
Step 2: Construct Shortest Path Distance Graph
Initialize $\mathbf{D}$: $D_{ij} = G_{ij}$ if x_i and x_j are neighbours, otherwise $D_{ij} = \infty$
For each value of $k \in \{1, ..., n\}$, set $D_{ij} = \min(D_{ij}, D_{ik} + D_{kj})$
Step 3: Construct Low-Dimensional Embedding
Compute $\mathbf{B} = -\frac{1}{2}\mathbf{H}\mathbf{D}^{(\mathbf{2})}\mathbf{H}$ where $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ and $D_{ij}^{(2)} = [D_{ij}]_{i,j=1}^2$ (see (2.1))
Let $\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ be the resulting Eigendecomposition
Return $\mathbf{Y} = \mathbf{V}_{(d)}\mathbf{\Lambda}_{(d)}^{\frac{1}{2}}$ corresponding to the largest d eigenvalues

Algorithm 2 ε-ISOMAP

Input: data $X \in \mathbb{R}^{n \times N}$, ball radius ε, dimension d
Output: embedding $Y \in \mathbb{R}^{n \times d}$
Step 1: Construct Neighbourhood Graph
Initialize $\mathbf{G} = \mathbf{0} \in \mathbb{R}^{n \times n}$
Set $G_{ij} = \|x_i - x_j\|_2$ $\forall$ pairs of points (x_i, x_j) if x_j is within ε distance of x_i
Step 2: Construct Shortest Path Distance Graph
Initialize $\mathbf{D}$: $D_{ij} = G_{ij}$ if x_i and x_j are within ε distance, otherwise $D_{ij} = \infty$
For each value of $k \in \{1, ..., n\}$, set $D_{ij} = \min(D_{ij}, D_{ik} + D_{kj})$
Step 3: Construct Low-Dimensional Embedding
Compute $\mathbf{B} = -\frac{1}{2}\mathbf{H}\mathbf{D}^{(\mathbf{2})}\mathbf{H}$ where $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ and $D_{ij}^{(2)} = [D_{ij}]_{i,j=1}^2$ (see (2.1))
Let $\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ be the resulting Eigendecomposition
Return $\mathbf{Y} = \mathbf{V}_{(d)}\mathbf{\Lambda}_{(d)}^{\frac{1}{2}}$ corresponding to the largest d eigenvalues

Step 2 in the algorithm involves constructing a complete graph where for each edge we have a weight corresponding to the minimum path distance over the graph. This is represented in the matrix D and is constructed using the Floyd-Warshall algorithm, which runs in $\Theta(n^3)$ time (Cormen et al., 2009). Another possibility involves using Dijkstra's algorithm for every

single point, but given that the output graph can be relatively dense, we opted for the former option.

In Step 3, MDS is performed as described in section 2.2, which involves calculating an eigendecomposition, giving us a cubic time complexity. Note that selecting the d largest eigenvalues is a constant time operation with respect to the other parameters, and since $d \ll N$, it is not unreasonable to treat this step as being performed in constant time. This is relevant since it means that retaining more complexity (features) in our final embedding does not appreciably increase the computational cost of the procedure.

Our full implementation in both the `R` programming language and Python can be found in this repository.

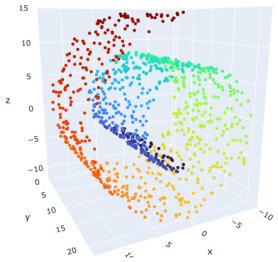
3 Experiments

To evaluate the performance of ISOMAP, we conducted experiments on both simulated and real-world datasets in order to illustrate the strengths and weaknesses of the algorithm.

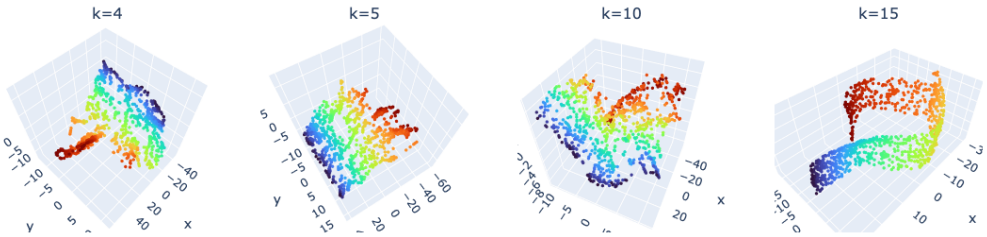
3.1 Swiss Roll

We first applied ISOMAP to a simulated Swiss Roll dataset ($n = 1000$) which is a classic manifold that lies in 3D as illustrated in Figure 1a but is intrinsically 2D. Notice that points closer to the origin of the roll (i.e., blue) are more densely sampled than the points furthest away from the origin (i.e., red). Figures 1b and 1c illustrate that for small values of k and ε respectively, ISOMAP performs poorly in sparse regions of the manifold. This is expected since ISOMAP relies on a well-connected local neighbourhood graph and the insufficient connectivity within these regions results in distortions. This can also be seen in Figure 2a which corresponds to the 2D embeddings for small k and ε - ideally these embeddings should resemble an unrolled Swiss Roll.

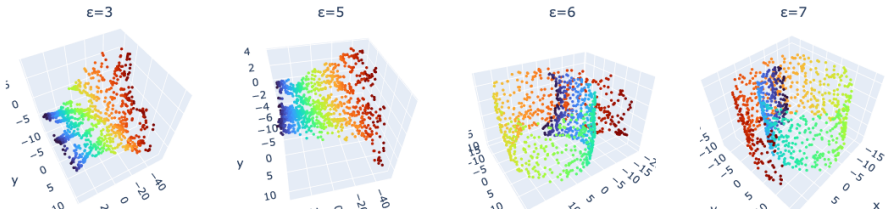
On the other hand, Figures 1b and 1c show that ISOMAP is able to sufficiently capture the connectivity of the sparse regions of the manifold for large values of k and ε respectively, However, the corresponding 2D embeddings shown in Figure 2b resemble a cross-section of the 3D embedding as opposed to an unrolled Swiss Roll.



(a) Swiss Roll in 3D

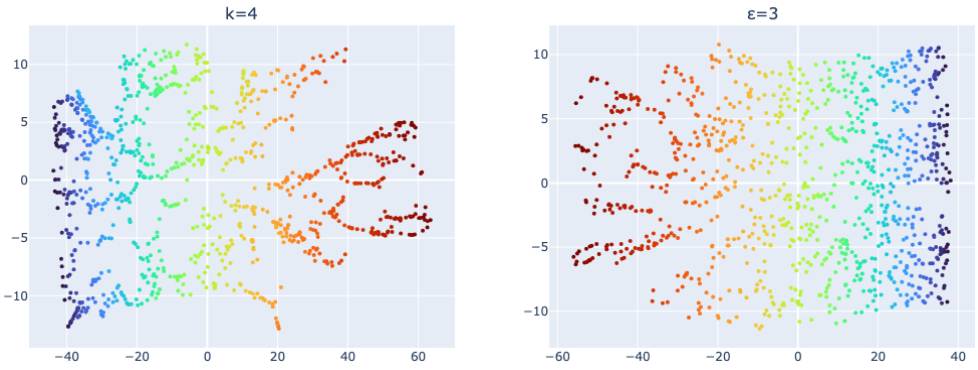


(b) k-ISOMAP Results for $k = 4, 5, 10, 15$

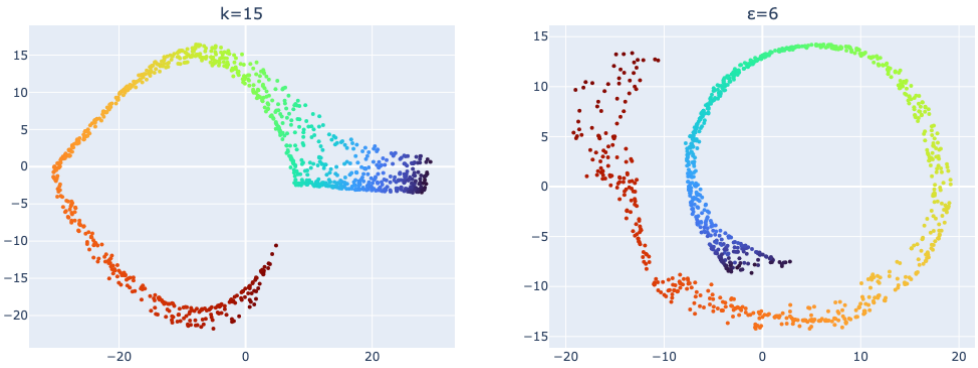


(c) ε-ISOMAP Results for $\varepsilon = 3, 5, 6, 7$

Figure 1: Performing ISOMAP on a 3D Swiss Roll to retrieve 3D Embedding



(a) Results for k-ISOMAP with $k = 4$ and ε-ISOMAP with $\varepsilon = 4$



(b) Results for k-ISOMAP with $k = 15$ and ε-ISOMAP with $\varepsilon = 6$

Figure 2: Performing ISOMAP on a 3D Swiss Roll to retrieve 2D Embedding

3.2 Stanford Faces

Next we applied ISOMAP to the Stanford Faces dataset, sourced from this repository. This dataset is a collection of 698 images of faces with image size 64×64, thus it lies in a 4096-dimensional space. The hypothesis that the data lies in an intrinsically lower dimensional manifold stems from the fact that the images differ solely in the rotation of the face from left-to-right and the tilt of the face from bottom-to-top, but the same face is always used. Figures 3 and 4 illustrates select 2D embeddings returned by ISOMAP (see 1.3 for further results). For k-ISOMAP with $k = 6$, the horizontal component of the embedding in Figure 3 appears to depict the rotation of the neck from left (negative values) to right (positive values) while the vertical component encodes the tilt of the neck from bottom (negative values) to top (positive values).

For ε-ISOMAP with $\varepsilon = 20$, the horizontal component of the embedding in Figure 4 appears to show a similar rotation of the neck from left-to-right, although there are some samples which don't follow this general trend. Furthermore, there is no clear interpretation of the vertical component for this embedding, suggesting that that this embedding may not faithfully represent the intrinsic dimensionality of the data. Note that the number of samples is lower than the number of variables, so ε-ISOMAP, which is especially sensitive to producing sparse graphs, produced an embedding that was both less able to uniformly separate the data and less interpretable when projected onto two dimensions.

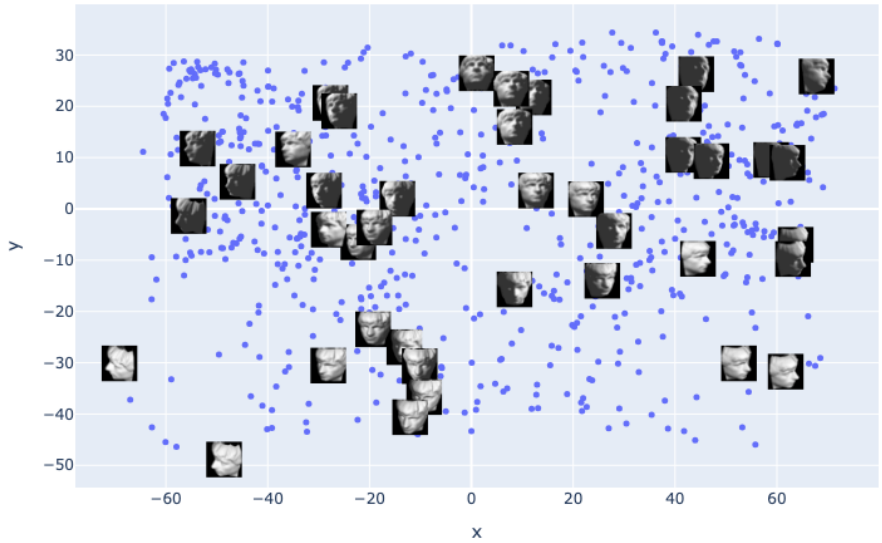


Figure 3: Results for k-ISOMAP with $k = 6$

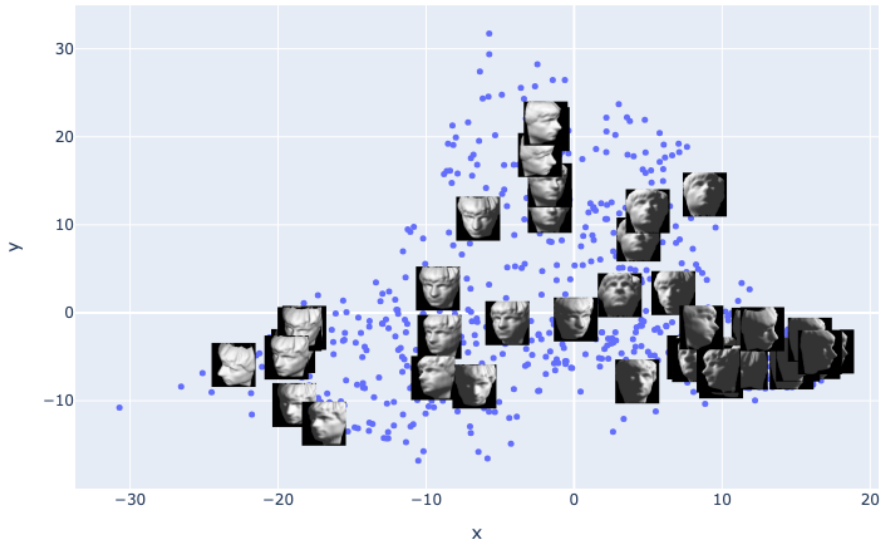


Figure 4: Results for ε-ISOMAP with $\varepsilon = 20$

3.3 MNIST

The Modified National Institute of Standards and Technology, or MNIST for short, is a dataset consisting on 60,000 images of 28×28 grayscale pixels. The images show handwritten digits and it has been extensively used as a benchmark for image processing systems and machine learning models. It is no longer accessible from its original web address (as of April 2025) but it is so well-known that many copies can be easily found online.

We applied ISOMAP to sub-samples of this dataset in a couple of ways. Given the size of the dataset, we sampled uniformly across the images to build our true dataset, including 400 instances for each label to maintain class balance. We also made embeddings for datapoints sharing a single label in order to get a better grasp of what ISOMAP was doing.

In figure 6 we show the embedding onto two components for a random subset of images sharing the label "8". In it we can see how the first component seems to encode some notion of tilt in the images, as the images with a large negative value in the first component are leaning with a positive slope, whereas those with a large positive value in the first component lean towards a negative slope. Moreover, the second component seems to encode the roundness of the images, where a negative value for this component seems to align with rounded circles and sometimes even eights made with two stokes, whereas a more positive value of the same component corresponds to eights that have less round circles and seem to be drawn in a single stroke, more akin to a vertical infinity symbol.

With the embeddings computed from the subset containing 400 images from each label, we performed a clustering using K-means. This choice of algorithm was made due to its simplicity and ease of analysis, but other more sophisticated techniques would have yielded better results. Nevertheless, even with a simple procedure, the class separations were notable, as encapsulated in figure 5. In it, we can see how the general trend is strong, with a majority of images being correctly classified, in fact, the overall accuracy was 56.11%. Table 1 shows a breakdown of the accuracy for the classification of each label, where we can see notable discrepancies in the individual accuracies. In particular, the embedding struggled to separate the class for the label "5", and there was a large overlap between the labels "4", "7" and "9". A possible conjecture for this phenomenon may be that these numbers have certain similarities in how they are handwritten by humans, leading to them having similar embeddings. Even in the worst case, the lowest accuracies were still above the value that would correspond to the null model, namely 10%.

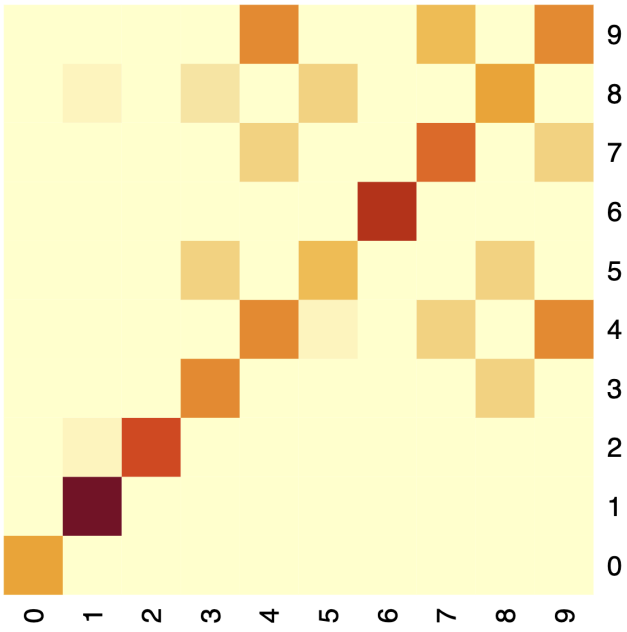


Figure 5: Confusion matrix after K-means of MNIST embedding with $k = 20$ and $d = 30$, where a darker color corresponds to a higher value.

Table 1: K-means accuracy for each of the labels from the MNIST embeddings $k = 20$, $d = 30$.

0	1	2	3	4	5	6	7	8	9
0.828	0.933	0.672	0.517	0.352	0.298	0.782	0.451	0.441	0.338

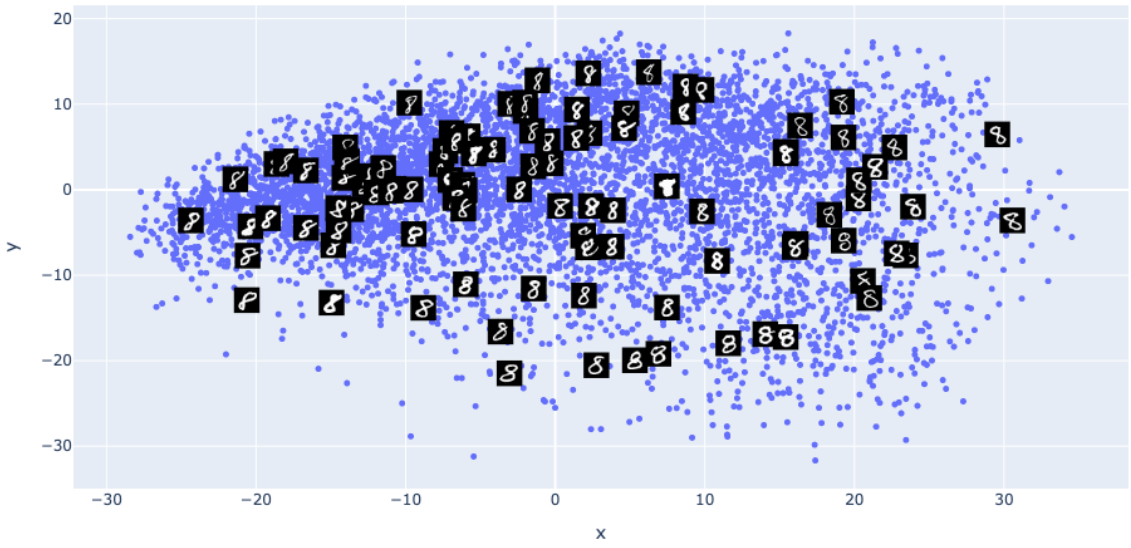
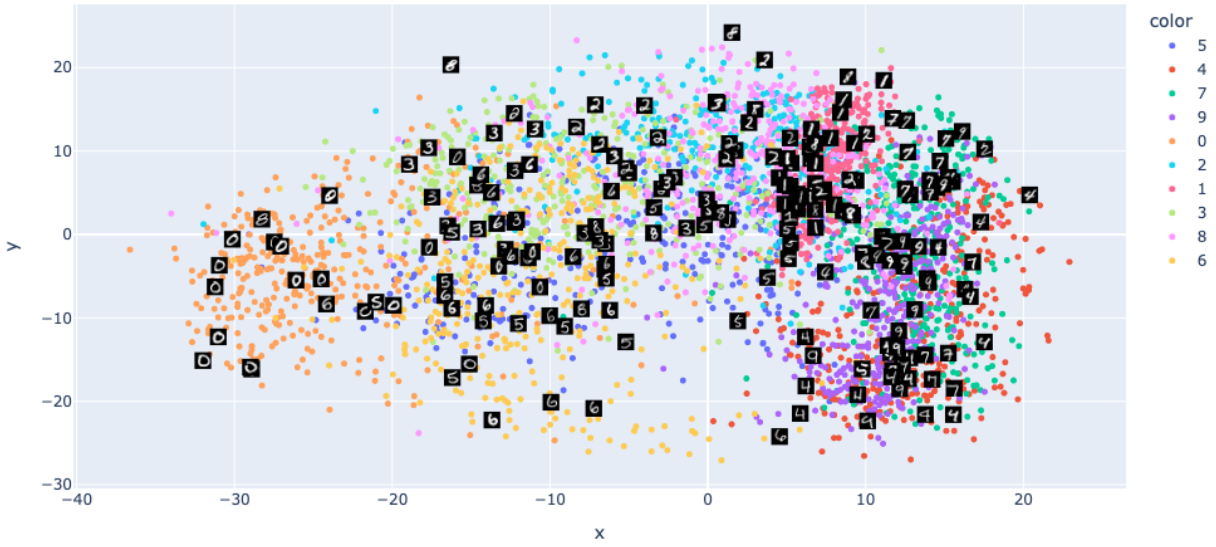
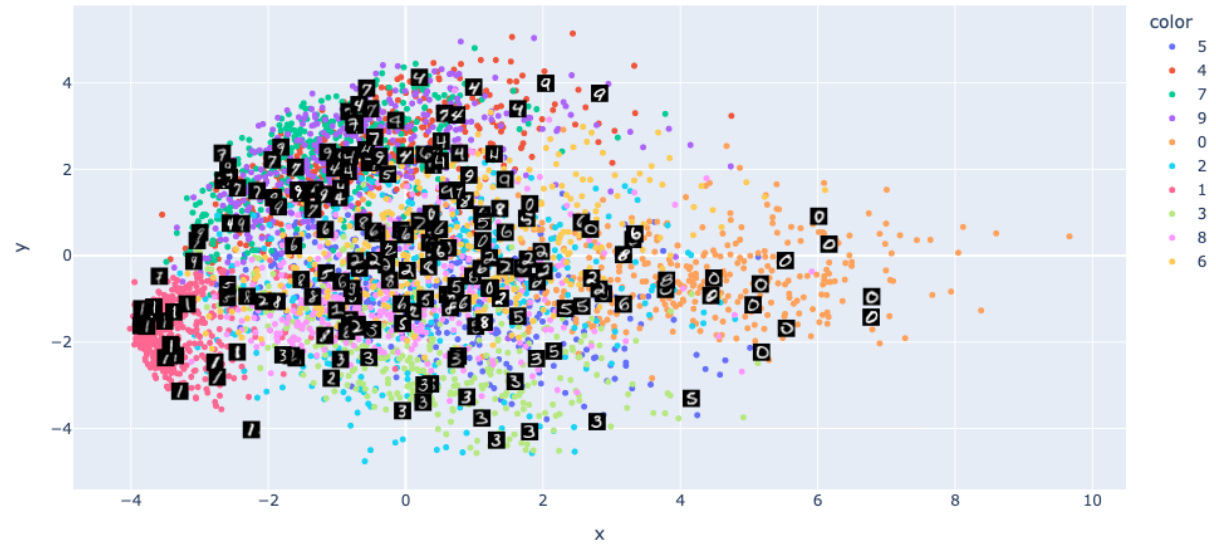


Figure 6: Results for k-ISOMAP with $k = 6$, only for the label "8".



(a) Results for k-ISOMAP with $k = 6$.



(b) Results for ε-ISOMAP with $\varepsilon = 15$.

Figure 7: Performing ISOMAP on MNIST Dataset, with 400 Points Sampled per Class

4 Conclusion

From the previous simulations we can see that despite ISOMAP's simplicity, it is able to capture the global structure of the manifolds, especially when the sample size chose is large. The accuracy of the clustering of the embedded points for the MNIST dataset is remarkable given the high speed of the computation, especially as compared to other methods such as t-SNE or UMAP, which are much more costly in terms of computational complexity.

We have seen that, unlike linear methods such as PCA, ISOMAP can uncover non-linear structures in the data and preserves more complex geometric relationships. It goes beyond simply finding directions of maximum variance, enabling it to handle data that lie on curved manifolds, as seen in the Swiss roll example.

The algorithmic steps are relatively straightforward and build upon classical MDS, making it highly interpretable.

Since ISOMAP is fundamentally a distance-based method operating in Euclidean space, its performance is heavily influenced by the choice of neighborhood tuning parameters k or ε. When working with the ε-ISOMAP variant, if the value of ε is made too small, the graph may become very sparse or disconnected which at best will lead to a numerically unstable matrix inversion and, at worst, to a singular matrix. If on the other hand, the parameters are too large, local neighborhoods may span multiple manifold regions and, therefore, distort the geodesic structure. ISOMAP is also sensitive to several practical issues common in high-dimensional data analysis. The presence of noise can distort local distances and create spurious connections in the neighborhood graph, which in turn undermines the local linearity assumption. Furthermore, as dimensionality increases, phenomena such as distance concentration and data sparsity make it harder to distinguish between near and far neighbors, weakening the reliability of local distance measures.

A fundamental limitation of ISOMAP lies in its reliance on approximating the true geodesic distance $d_{\mathcal{M}}(x_i, x_j)$ using a graph-based shortest-path estimate $d_G(x_i, x_j)$. This approximation can be viewed as a two-step process: first, approximating the manifold geodesic $d_{\mathcal{M}}$ with an idealized sample-based geodesic d_S, and second, approximating d_S with the shortest-path graph distance d_G. The accuracy of $d_{\mathcal{M}} \approx d_S$ depends critically on the density and uniformity of the sampled data. In regions of low sample density or sparse sampling, particularly in high dimensions, the intermediate estimate d_S becomes unreliable due to gaps in the coverage of the manifold. Moreover, the approximation $d_S \approx d_G$ assumes that Euclidean distances provide a faithful representation of local manifold geometry, which holds only when the neighborhood size is small enough that curvature effects are negligible in the tangent space.

References

Anowar, F., Sadaoui, S., and Selim, B. (2021). Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). *Computer Science Review*, 40:100378.

Belkin, M. and Niyogi, P. (2008). Towards a theoretical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308. Learning Theory 2005.

Choi, H. and Choi, S. (2007). Robust kernel isomap. *Pattern recognition*, 40(3):853–862.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. The MIT Press, 3rd edition.

Lee, J. A. and Verleysen, M. (2007). *Nonlinear dimensionality reduction*. Springer Science & Business Media.

McInnes, L., Healy, J., and Melville, J. (2020). Umap: Uniform manifold approximation and projection for dimension reduction.

Ramsay, J. O. (1982). Some statistical approaches to multidimensional scaling data. *Journal of the Royal Statistical Society: Series A (General)*, 145(3):285–303.

Shepard, R. N. (1980). Multidimensional scaling, tree-fitting, and clustering. *Science*, 210(4468):390–398.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419.

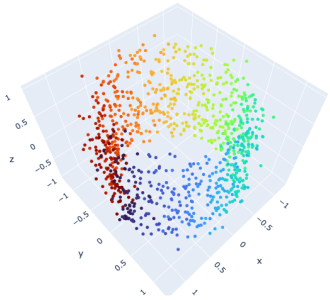
Wagaman, A. and Levina, E. (2009). Discovering sparse covariance structures with the isomap. *Journal of Computational and Graphical Statistics*, 18(3):551–572.

Williams, C. (2000). On a connection between kernel pca and metric multidimensional scaling. *Advances in neural information processing systems*, 13.

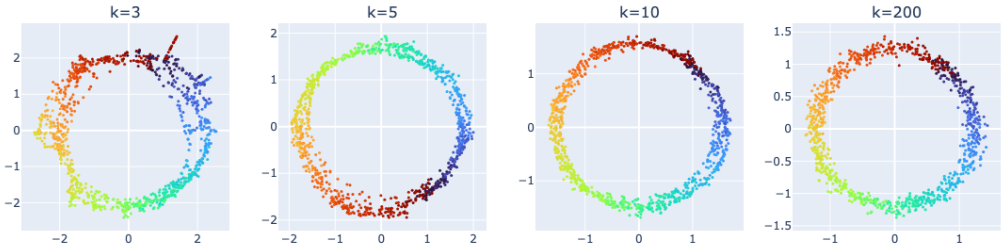
Yang, M.-H. (2002). Face recognition using extended isomap. In *Proceedings. International Conference on Image Processing*, volume 2, pages II–II. IEEE.

Appendix 1 More Simulation Results

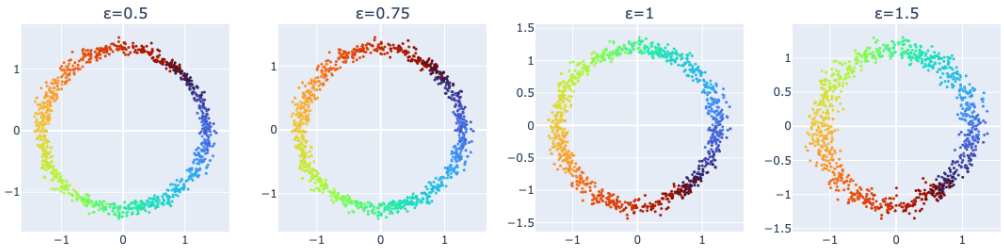
1.1 Cylinder



(a) Original Cylinder in 3D.



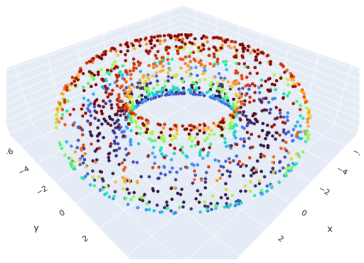
(b) k-ISOMAP Results for $k = 2, 5, 10, 200$.



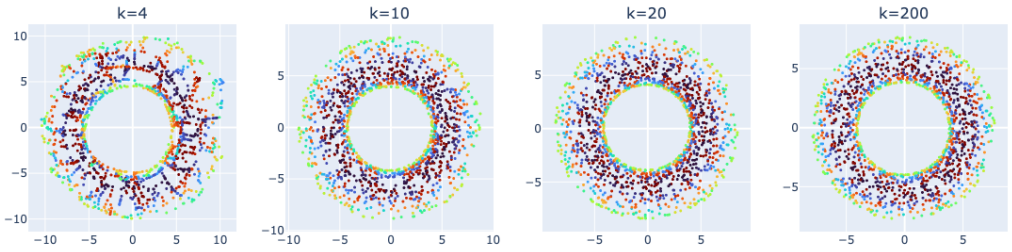
(c) ε-ISOMAP Results for $\varepsilon = 0.5, 0.75, 1, 1.5$.

Figure 8: Performing ISOMAP on a 3D Cylinder to retrieve 2D Embedding.

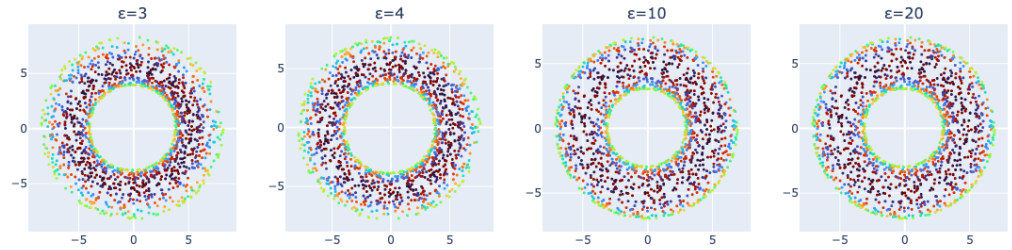
1.2 Torus



(a) Original Torus in 3D.



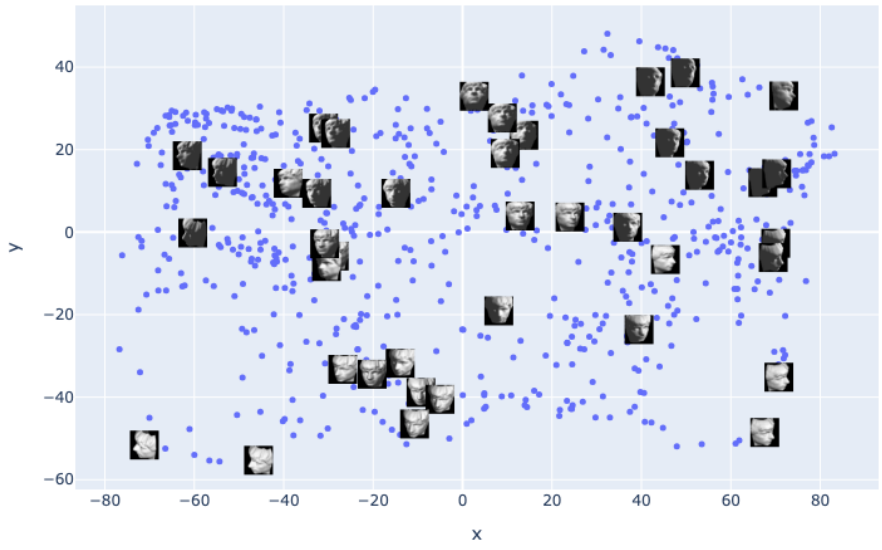
(b) k-ISOMAP Results for $k = 4, 10, 20, 200$.



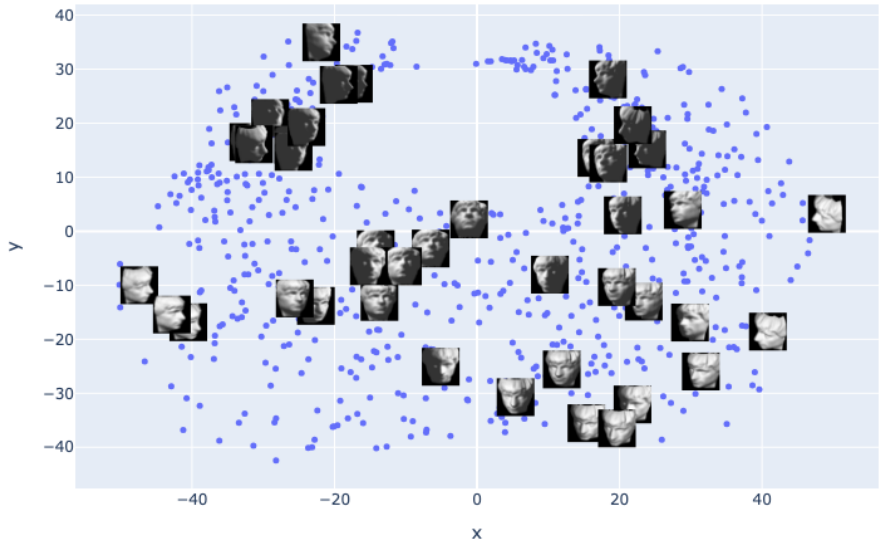
(c) ε-ISOMAP Results for $\varepsilon = 3, 4, 10, 20$.

Figure 9: Performing ISOMAP on a 3D Torus to retrieve 2D Embedding.

1.3 More on Stanford Faces

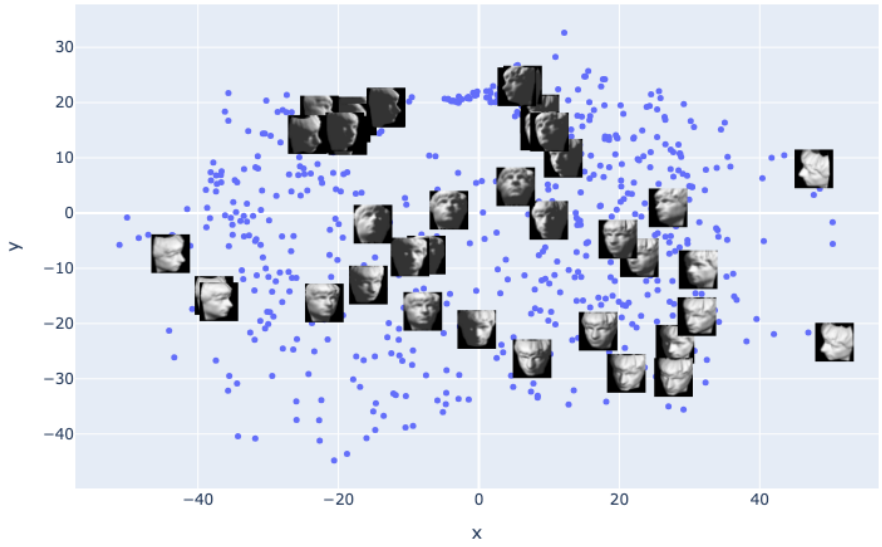


(a) Results for k-ISOMAP with $k = 4$.

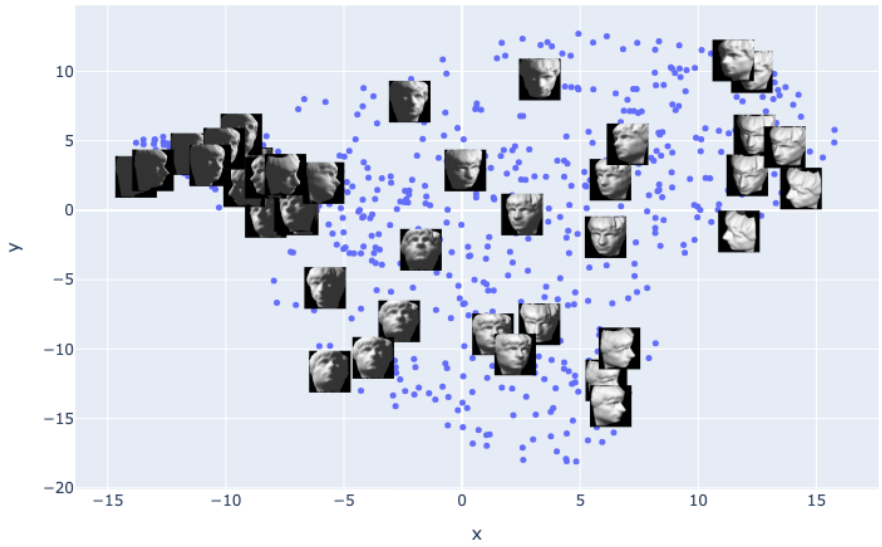


(b) Results for k-ISOMAP with $k = 10$.

Figure 10: Performing k-ISOMAP on Stanford Faces Dataset.



(a) Results for ε-ISOMAP with $\varepsilon = 11$.



(b) Results for ε-ISOMAP with $\varepsilon = 45$.

Figure 11: Performing ε-ISOMAP on Stanford Faces Dataset.