Homework 3 40pt (Math 466, Spring 2025)

Due Date: Apr. 3, Thur, 10 pm

Instructions

- Submissions must be made in PDF format. If there are coding problems, please provide source code that directly generates the reported results.
- In written problems, if no derivation is asked for, just give the answer. Otherwise, try to give the derivation in a clear and mathematically rigorous way.
- For coding problems, please include a mini-report summarizing your experiment, along with any relevant figures or tables.

  When creating plots or tables, ensure they are clear and easy to interpret by using different colors or markers, and by adding axis labels, titles, captions, etc. to enhance readability.

  You are also required to submit the code used to generate the results presented in your mini-report. This can be provided as either a .ipynb or .py file.
- Collaboration and AI tools are allowed, but you must write your solutions independently and acknowledge collaborators/AI tools.

Questions

1. **(18pt) This problem is to compare dimension reduction methods we learned in class.**

   In class, we covered several dimension reduction methods, including PCA, MDS, Isomap, Laplacian eigenmaps, and t-SNE. To implement these methods in this problem, you can use the following `sklearn` packages: `PCA`, `MDS`, `Isomap`, `SpectralEmbedding` and `TSNE`

   Use the provided code to generate Swiss-roll data. Then, apply the five methods above to reduce the data dimensionality to 2D and compare their performance.

   If you are unsure how to compare the methods, consider the following aspects:

- the 2D representation of the data
- The computation time for obtaining the 2D embedding (excluding time spent on data generation and plotting)
- The embedding results with varying numbers of training samples
- For methods without a closed-form solution (e.g., MDS, t-SNE), the effect of different initializations or varying iteration numbers on the embedding results
- For graph-based methods (e.g., Isomap, Laplacian eigenmaps), the impact of different choices of k in the kNN graph
- For graph-based methods (e.g., Isomap, Laplacian eigenmaps), the effect of different scaling values in the RBF kernel graph.

2. **(10pt) This problem is about PCA.**

   Define the data matrix $X = \begin{bmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{bmatrix}_{n \times D}$ $(n > D)$, and the centralized data matrix $X_c := \begin{bmatrix} (x_1 - \mu)^\top \\ \vdots \\ (x_n - \mu)^\top \end{bmatrix}$, where $\mu = \frac{1}{n} \sum_{i=1}^n x_i$ is the empirical mean.

   Let the full SVD of X_c be $X_c = U \Sigma V^\top$, where $U = [u_1, \cdots, u_n]$, $V = [v_1, \cdots, v_D]$, $\operatorname{diag}(\Sigma) = [\sigma_1, \sigma_2, \cdots, \sigma_D]$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_D \geq 0$.

   Denote the first d principal components as $w_1, \cdots, w_d$ and the d-dimensional PCA embedding of $x_1, \cdots, x_n$ as $y_1, \cdots, y_n$.

   (a) (2pt) Write the dimension of w_j and y_i, and use σ's, u's and v's to represent the principal components w_j and representation y_i.

   (b) (6pt) Consider a set of grayscale images of resolution 8×8. We can view each data $x_i, i = 1, \cdots, n$ as a vector in $\mathbb{R}^D$ with $D = 64$.

   - Use the provided code to load, add noise and visualize the clean image data `data` and the noisy image data `data_noise`. Apply PCA to the clean image data and noisy image data.
   - Plot all the singular values of clean data and noisy data respectively in descending order, and compare them on the same plot
   - Plot the histogram of the singular values of clean data and noisy data respectively in 40 bins and compare them on the same plot.
   - Visualize the first 10 principal components of the clean data and noisy data respectively, and compare the two sets of principal components.

   Summarize your observations and provide explanations.

(c) (2pt) Based on your observations, in which applications do you think PCA can be used? What potential issues might arise, and what are the corresponding solutions? Please explain your answer.

3. **(12 pt + optional 6 pt) This problem is about manifold learning.**

In this problem, we demonstrate how diffusion maps compute an embedding that reflects only the manifold structure invariant to the distribution.

Recap: Given $x_1, \cdots, x_n$, we construct a weighted undirected graph G with n nodes.

- Let W be the graph affinity matrix, $W_{ij} \geq 0$ and $W = W^T$,
- Let D be the degree matrix of W, with diagonal entries $d_i := \sum_{k=1}^n W_{ik}$,
- Assume $d_k > 0$ and define the transition matrix $P = D^{-1}W$.

As shown in class, P is a right stochastic matrix. The entry P_{ik} represents the probability of transitioning from x_i to x_k in one step.

Diffusion map (step I): Diffusion maps are built on the spectral information of the transition matrix P. We first examine the existence of the eigen-decomposition of P and the properties of its eigenvalues.

(a) (4pt) Prove that there exists an invertible matrix Ψ and a real diagonal matrix Λ such that $P = \Psi\Lambda\Psi^{-1}$. Additionally, prove that $\Psi^{-1} = \Psi^\top D$.

(b) (4pt) Prove that any eigenvalue of P lies in $[-1, 1]$.

Diffusion map (step II): In class, we discussed that that, for any i, k, $(P^t)_{ik}$ represents the probability of jumping from x_i to x_k in t steps. In addition,

- For any i, $(P^t)_{i,\cdot}$ is a probability distribution associated with x_i.
- For any k, d_k can be interpreted as the volume of x_k on the graph.

The diffusion distance between x_i, x_j at scale t is defined as

$$D_t^2(x_i, x_j) := \sum_{k=1}^n \frac{1}{d_k}((P^t)_{ik} - (P^t)_{jk})^2.$$

This can be understood as a weighted norm between two probability distributions, reflecting the connectivity of the data at a given scale.

Denote the eigenvalues of P as $\lambda_0 \geq \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n-1}$ with corresponding eigenvectors as $\psi_0, \psi_1, \cdots, \psi_{n-1}$. A diffusion map $\Psi_t : \{x_1, x_2, \cdots, x_n\} \to \mathbb{R}^s$ computes the s-dimensional representation of the data points by

$$\Psi_t(x_i) := [\lambda_1^t\psi_1(i), \lambda_2^t\psi_2(i), \cdots, \lambda_s^t\psi_s(i)]^\top,$$

where $\psi_k(i)$ is the i-th entry of the vector ψ_k.

(c) (optional, 6pt) Prove that if $s = n - 1$, then

$$D_t^2(x_i, x_j) = \|\Psi_t(x_i) - \Psi_t(x_j)\|_2^2,$$

i.e., the diffusion map embedding preserves the diffusion distance.

Diffusion map (anisotropic): Use the provided code to generate and visualize the 1D toroidal helix manifold embedded in 3D Euclidean space. Notice that certain regions contain a higher concentration of points. This can be interpreted as sampling from a non-uniform distribution on the manifold (though not in a rigorous sense).

If we define the affinity matrix as

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\epsilon}\right),$$

this affinity is isotropic, meaning it does not account for variations in point density. As a result, the embedding may be influenced by the density of points on the manifold.

In many applications, however, one aims to recover the intrinsic manifold structure independent of the data distribution. This can be achieved using the following procedure with some α:

- Construct the isotropic affinity matrix $W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\epsilon}\right)$ and compute the corresponding degree matrix D.
- Construct the anisotropic affinity matrix $W^{(\alpha)}$ by $W_{ij}^{(\alpha)} := \frac{W_{ij}}{d_i^\alpha d_j^\alpha}$ and compute the corresponding degree matrix $D^{(\alpha)}$
- Compute the anisotropic diffusion map defined by the anisotropic transition matrix $P^{(\alpha)} := (D^{(\alpha)})^{-1} W^{(\alpha)}$.

(d) (4pt) Choose parameters $t = 1, s = 2$ and $\epsilon = 0.1$, compute the diffusion map embedding of the data `X,y` given in the code. Here `X[i,:]` corresponds to x_i and `y[i]` is a label of x_i.

- Compute the diffusion map embedding for $\alpha = 0$ and $\alpha = 1$.
- Plot the diffusion map embedding for $\alpha = 0$ and $\alpha = 1$. In the plot, color each point $\Psi(x_i)$ with `y[i]`.
- Compare the embeddings for $\alpha = 0$ and $\alpha = 1$. How do the results differ? How do you understand the results and the differences?