Name: Pavitr Chandwani
Roll No.: 23110241

CS 303 Coding Assignment Report

Problem 1:

Choice of Distance Metric:

The choice of distance metric is especially crucial in dimensionality reduction especially. You have to make sure to choose the correct metric depending on the type of dimension of the data.

Suppose you have 2D data, then using Euclidean distance would make more sense as it captures all the helpful quality of the data, but it fails when the data is non-linear and doesn't capture everything in manifolds.

Geodesic and diffusion distances are commonly used for these non-linear manifold distances. While geodesic is commonly used, it is very computationally extensive. On the other hand, diffusion distance measures the strength of the number of paths connecting two points in a non-linear setting. It captures the whole global geometric distance while not compromising with anything.

Why DIffusion Maps outperform PCA/t-SNE:

- **Global structure is preserved:** As discussed above, Diffusion Maps uses the diffusion distance method, which helps preserve the quality of higher dimensional data and doesn't assume much, unlike others, as PCA assumes data is linear and t-SNE distorts global qualities sometimes.
- **Computational Efficiency on Large Datasets:** Diffusion maps use matrix techniques, which provide a good amount of leverage on computational efficiency and accuracy. In contrast, other techniques are computationally expensive and fail for more complex structures/data.
- **Better Complex Manifold Learning:** Diffusion maps are ideal for non-linear and high-dimensional datasets as they can capture the geometry of datasets, while PCA fails for non-linear datasets and t-SNE fails on a global level as it focuses on local neighborhoods.

Problem 2:

Trade-offs in each Method:

1. **Nelder-Mead Optimization:**
    a. Pros:
        i. Simple to implement and doesn't require much information like a gradient.
        ii. Easy to use and effective on small-dimensional data
    b. Cons:
        i. It can fail for higher-dimensional data or non-smooth functions.
        ii. Doesn’t work that well with noisy functions.
2. **Simulated Annealing:**
    a. Pros:
        i. Works well with high-dimension data and noisy functions.
        ii. Suitable for dealing with local minima due to their probabilistic nature.
    b. Cons:
        i. Computationally expensive due to its nature.
        ii. Slower to converge compared to other optimization methods.
3. **CMA-ES Method:**
    a. Pros:
        i. Very good with high-dimension and non-convex optimization problems.
        ii. Preferred due to self-adaptation.
        iii. Handles noisy functions well.
    b. Cons:
        i. Computationally expensive due to frequent updation of the covariance matrix.
        ii. High convergence time. (It takes a lot of time to solve large-dimensional problems.)