Coding Assignment (10 marks)

Course number: CS-303

March 6, 2025

Problem 1. (Data Vizualization). This assignment explores **Diffusion Maps**, a manifold learning technique, to **cluster time-series data**. You will work with the **UCI Human Activity Recognition (HAR)** dataset and apply Diffusion Maps for dimensionality reduction before clustering. The goal is to analyze how well this approach separates human motion activities.

Dataset Description. You will use the **UCI HAR dataset**, which consists of sensor readings from accelerometers and gyroscopes attached to smartphones. The dataset contains labeled activities:

- Walking
- Walking upstairs
- Walking downstairs
- Sitting
- Standing
- Lying down

Dataset Link: UCI HAR Dataset

Task 1: Preprocessing the Time-Series Data (1 mark)

1. Load the HAR dataset and segment the time series into overlapping windows. 2. Compute **pairwise distances** between time-series segments using:

- **Dynamic Time Warping (DTW) distance**
- **Euclidean distance**

3. Construct a similarity matrix W, where W_{ij} measures similarity between time-series x_i and x_j.

Task 2: Apply Diffusion Maps for Dimensionality Reduction (1.5 mark)

1. Construct the **Diffusion Kernel**:

$$K_{ij} = \exp\left(-\frac{d(x_i, x_j)^2}{\epsilon}\right) \tag{1}$$

where $d(x_i, x_j)$ is the **DTW distance**. 2. Compute the **normalized graph Laplacian** and its leading eigenvectors. 3. Use the top **2 or 3 diffusion coordinates** to embed the time series into a **low-dimensional space**.

Task 3: Clustering in the Diffusion Space (1 mark)

1. Apply **K-Means** or **DBSCAN** to the diffusion map embeddings. 2. Evaluate clustering performance using:

- **Adjusted Rand Index (ARI)**
- **Silhouette Score**

Task 4: Visualization and Interpretation (1.5 mark)

1. **Scatter Plot** the diffusion embeddings, color-coded by activity labels. 2. Compare clustering quality using:

- Raw feature space
- PCA
- t-SNE
- Diffusion Maps

3. Discuss **why Diffusion Maps work well** for time-series clustering.

Task 5: Explorative

- Implement **Multiscale Diffusion Maps** to capture hierarchical structures in time-series data.
- Use **Spectral Clustering** instead of K-Means for improved grouping.

Expected Deliverables

- **Python Notebook** implementing Diffusion Maps for clustering.
- Include **plots** showing low-dimensional embeddings of time-series segments in the above notebook
- **Short Report** explaining the choice of distance metric and why Diffusion Maps outperform PCA/t-SNE.

Problem 2. (Derivative-Free Optimization Methods). This assignment explores **Nelder-Mead**, **Simulated Annealing**, and **Covariance Matrix Adaptation Evolution Strategy (CMA-ES)**. You will implement these optimization techniques and compare their performance on different objective functions. Implement or use available libraries to apply the following optimization techniques:

- **Nelder-Mead** (Simplex Method)
- **Simulated Annealing**
- **CMA-ES (Covariance Matrix Adaptation Evolution Strategy)**

Ensure that each method is applied with appropriate hyperparameters.

Task 1: Benchmarking on Test Functions (1 mark)

1. Optimize the following benchmark functions:

- **Rosenbrock function**:

$$f(x,y) = (1-x)^2 + 100(y-x^2)^2 \tag{2}$$

- **Rastrigin function**:

$$f(\mathbf{x}) = 10d + \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i)\right] \tag{3}$$

- **Ackley function**:

$$f(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + e \tag{4}$$

2. Compare convergence speed and accuracy of each method on these functions.

Task 2: Hyperparameter Tuning in Machine Learning (2 marks)

1. Apply the three optimization methods to tune hyperparameters for a machine learning model:
   - Consider a Support Vector Machine (SVM) for classification on the MNIST dataset.
   - Optimize hyperparameters such as kernel type, regularization parameter (C), and gamma.
   - Compare the test accuracy obtained using each optimization technique.
2. Analyze and compare the efficiency of each method in terms of:
   - Number of function evaluations required to reach optimal hyperparameters.
   - Performance stability over multiple runs.
   - Final classification accuracy.

Task 3: Performance Analysis and Visualization (2 marks)

1. Plot the optimization trajectories for each method in 2D and 3D where applicable.
2. Compare the number of function evaluations required to reach the minimum.
3. Analyze robustness to different initial conditions.
4. Visualize the hyperparameter search landscape and final model performance.

Expected Deliverables

- **Python Notebook** implementing and comparing all three optimization techniques.
- Include **Plots and analysis** of function convergence in the above notebook
- **Short Report** discussing the trade-offs of each method.