Assignment 4

Objectives

The objectives of this assignment are as follows:

- Implementing dimensionality reduction through PCA and Autoencoders
- Implementing clustering algorithms: K-means and Gaussian Mixture Models
- Evaluating clustering performance using multiple metrics
- Analyzing the impact of different dimensionality reduction techniques on clustering
- Statistical validation and comparison of approaches

Problem Statement

Given the Breast Cancer Wisconsin (Diagnostic) dataset, the objective is to perform comprehensive unsupervised clustering analysis to identify inherent patterns or groupings within the dataset. This dataset contains features computed from digitized images of fine needle aspirates (FNAs) of breast masses, aiming to distinguish between malignant and benign tumors.

Assignment Details

Part 1: Implementation from Scratch (55%)

Implement the following **using only NumPy** (no scikit-learn):

1. **PCA (Principal Component Analysis)** - 12%
    - Implement full eigenvalue decomposition approach
    - Include explained variance ratio calculation
    - Implement reconstruction error computation
    - Support inverse transform (reconstruction)
2. **Autoencoder** - 18%
    - Implement a fully connected autoencoder with:
        - At least 3 hidden layers in encoder and decoder
        - Customizable bottleneck dimension
        - Multiple activation functions (ReLU, sigmoid, tanh)
    - Implement backpropagation from scratch
    - Include training with mini-batch gradient descent
    - Implement learning rate scheduling
    - Add regularization (L2)
3. **K-Means Clustering** - 10%
    - Implement K-Means++ initialization
    - Include random initialization for comparison

Prof. Dr. Marwan Torki
Eng. Youssef El-Ebiary
Eng. Youssef El-Kady
Eng. Ziad Fahmy
Eng. Sarah Mahmoud

- Implement convergence criteria (tolerance-based and max iterations)
- Track and report inertia history

4. **Gaussian Mixture Models (GMM)** - 15%
    - Implement full EM algorithm from scratch
    - Support all covariance types: full, tied, diagonal, spherical
    - Include log-likelihood computation
    - Implement convergence monitoring
    - Add numerical stability handling (avoid singular matrices)

Part 2: Comprehensive Experiments (30%)

Conduct **six experiments** comparing dimensionality reduction and clustering combinations:

Experiment 1: K-Means on original data

- Find optimal k using: elbow method, silhouette analysis, and gap statistic
- Compare K-Means++ vs random initialization
- Report convergence speed

Experiment 2: GMM on original data

- Find optimal components using BIC and AIC
- Compare all four covariance types
- Analyze log-likelihood convergence

Experiment 3: K-Means after PCA

- Test with different numbers of principal components (2, 5, 10, 15, 20)
- Analyze trade-off between dimensionality and clustering quality
- Compare reconstruction error vs clustering performance

Experiment 4: GMM after PCA

- Use same principal component variations as Experiment 3
- Compare all covariance types for each dimension
- Analyze how dimensionality affects optimal covariance type

Experiment 5: K-Means after Autoencoder

- Train autoencoders with different bottleneck sizes (2, 5, 10, 15, 20)
- Compare with PCA results from Experiment 3
- Analyze reconstruction loss vs clustering performance

Experiment 6: GMM after Autoencoder

- Use same bottleneck dimensions as Experiment 5

- Compare with PCA-GMM results from Experiment 4
- Determine which dimensionality reduction technique works best with GMM

Part 3: Evaluation & Analysis (15%)

For **each experiment**, compute and report:

1. **Internal Validation Metrics:**
   - Silhouette Score (implementation required)
   - Davies-Bouldin Index (implementation required)
   - Calinski-Harabasz Index (implementation required)
   - Within-cluster sum of squares (WCSS)
   - For GMM: BIC, AIC, and log-likelihood
2. **External Validation Metrics** (use labels only for evaluation, not training):
   - Adjusted Rand Index (implementation required)
   - Normalized Mutual Information (implementation required)
   - Purity (implementation required)
   - Confusion matrix analysis
3. **Dimensionality Reduction Quality:**
   - Reconstruction error (MSE)
   - For PCA: explained variance ratio
   - For Autoencoder: training loss curves
4. **Statistical Analysis:**
   - Create comprehensive comparison tables across all 6 experiments
   - Perform paired statistical tests between methods
   - Analyze computational complexity (time and space)
5. **Required Visualizations:**
   - 2D projections of all dimensionality reduction results (using first 2 components/dimensions)
   - Cluster assignments overlaid on 2D projections
   - Elbow curves with marked optimal k
   - BIC/AIC curves for GMM
   - Training curves for autoencoder (loss vs epochs)
   - Heatmap comparing all methods across all metrics
   - Confusion matrices for best-performing methods

Technical Requirements

- **Prohibited:** scikit-learn for implementations (can use only for validation)
- **Required:** NumPy for all implementations
- **Allowed:** Pandas (data loading only), Matplotlib/Seaborn (visualization only)

Extra Notes

- Dataset: Breast Cancer Wisconsin Dataset
- Normalize/standardize your data appropriately
- Handle missing values if any
- Document all hyperparameters and random seeds
- Work in groups of three: submit as id1_id2_id3.zip

Grading Scheme

Implementations (55%):

- PCA: 12%
- Autoencoder: 18%
- K-Means: 10%
- GMM: 15%

Experiments (30%):

- Six experiments properly conducted: 20%
- Parameter tuning and optimization: 10%

Evaluation & Analysis (15%):

- Metrics implementation and computation: 8%
- Statistical analysis and comparison: 4%
- Visualizations: 3%

Penalties:

- Late submission: -10% per day (maximum 3 days, then 0)
- Code doesn't run: -30%
- Missing implementations: -50%
- Incomplete report: -20%

Final Notes

1. You should work in groups of maximum three, not finding a team is NOT an excuse.
2. You should deliver with a naming scheme id _assigment.zip.
3. Delivery will be ignored if you didn't follow the naming scheme provided in 2, any one of the team ids can be used.
4. Any form of academic dishonesty, including but not limited to using AI tools (such as

Prof. Dr. Marwan Torki

Eng. Youssef El-Ebiary Eng. Youssef El-Kady
Eng. Ziad Fahmy Eng. Sarah Mahmoud

ChatGPT or other code generation platforms), copying open-source code without proper attribution, or engaging in 'vibe coding' without genuine understanding, will be considered a serious violation and will be heavily penalized.

Prof. Dr. Marwan Torki

Eng. Youssef El-Ebiary
Eng. Youssef El-Kady
Eng. Ziad Fahmy
Eng. Sarah Mahmoud