MNIST Classification using MLE, PCA, FDA, and Discriminant Analysis

Anand Kumar

February 17, 2025

1 Introduction

This report implements a complete classification pipeline for handwritten digit images using only classes 0, 1, and 2 from the MNIST dataset. The steps include:

- **Data Loading and Preprocessing:** Downloading the MNIST dataset (in IDX format) using the `kagglehub` module, loading the raw image and label files, normalizing the data, and selecting 100 random samples per class for both training and testing.
- **Maximum Likelihood Estimation (MLE):** Estimating the mean vector and covariance matrix for each class.
- **Principal Component Analysis (PCA):** Reducing the dimensionality of the image space using three settings:
  - Retaining 95% of the total variance.
  - Retaining 90% of the total variance.
  - Keeping only the first two principal components.
- **Fisher's Discriminant Analysis (FDA):** Computing the FDA projection using the PCA (95%) features to further reduce the dimensionality to 2 (which is the maximum possible for three classes).
- **Discriminant Analysis:** Implementing classification using both Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) on the various feature representations.
- **Visualization:** Plotting the 2D projections of the data obtained via FDA and PCA (2 components) to visualize class separability.

2 Dataset

The MNIST dataset is downloaded using `kagglehub.dataset_download("hojjatk/mnist-dataset")`. The dataset contains the standard IDX files:

- `train-images.idx3-ubyte`
- `train-labels.idx1-ubyte`
- `t10k-images.idx3-ubyte`
- `t10k-labels.idx1-ubyte`

Custom functions are provided to load the IDX format files into NumPy arrays. The images are then normalized to the range [0, 1]. Only images corresponding to classes 0, 1, and 2 are retained, and for each class, 100 samples are randomly selected for both training and testing.

3 Methodology

3.1 Maximum Likelihood Estimation (MLE)

For each class, the mean vector and covariance matrix are computed from the training data:

- **Mean (μ_c):** Calculated as the average of all feature vectors for a class.
- **Covariance (Σ_c):** Computed explicitly using:

$$\Sigma_c = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu_c)(x_i - \mu_c)^T$$

These MLE estimates provide a basis for the Gaussian class-conditional densities assumed in QDA.

3.2 Principal Component Analysis (PCA)

The PCA function is implemented from scratch:

- The training data are centered by subtracting the mean.
- The covariance matrix is computed explicitly.
- Eigen-decomposition of the covariance matrix is performed.
- The eigenvalues are sorted in descending order, and the number of principal components k is chosen such that at least 95% (or 90%) of the total variance is retained.
- For the 2-component PCA experiment, after obtaining all components (via a high variance threshold), the first two eigenvectors are manually selected.

3.3 Fisher's Discriminant Analysis (FDA)

FDA is implemented by computing:

- **Within-class scatter matrix (S_W):** The sum of the covariance matrices (using centered data) for each class.
- **Between-class scatter matrix (S_B):** Computed using the differences between each class mean and the overall mean, scaled by the number of samples in each class.

The generalized eigenvalue problem $\text{inv}(S_W)S_B$ is solved to find the optimal projection matrix W. For three classes, the maximum projection dimension is 2.

3.4 Discriminant Analysis

Both LDA and QDA classifiers are implemented from scratch:

- **LDA:** Assumes a common covariance matrix for all classes. The discriminant function is given by:
$$g(x) = x^T\Sigma^{-1}\mu_c - \frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c + \log(\pi_c)$$
where π_c is the class prior.
- **QDA:** Uses class-specific covariance matrices. Its discriminant function is:
$$g(x) = -\frac{1}{2}\log|\Sigma_c| - \frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c) + \log(\pi_c)$$

The classifiers are evaluated on features obtained from:

- PCA with 95% variance.
- PCA with 90% variance.
- PCA with only 2 components.
- FDA projection (using PCA 95% features) for 2D space.

3.5 Experimental Results

Two plots are generated:

- **FDA Projection Plot:** A scatter plot of the training data projected onto the 2 FDA dimensions shows clusters corresponding to the three classes.
- **PCA (2 Components) Plot:** A scatter plot of the training data using the first two principal components provides an overall view of the data's variance.

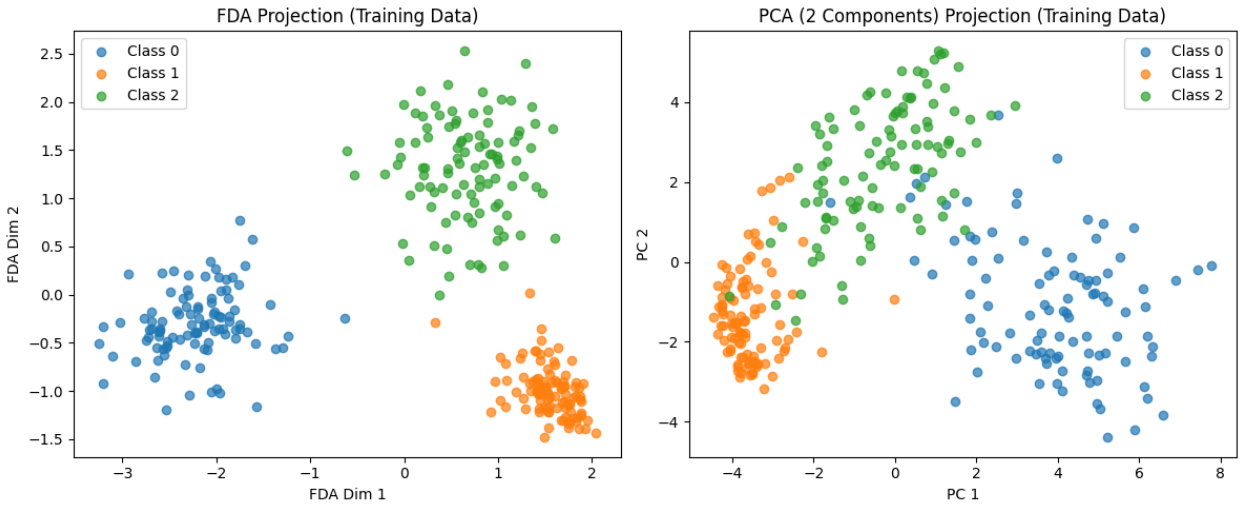

Figure 1: FDA and PCA(2 component Projection

```
 Classification on PCA (95% variance) features
LDA:  Train Accuracy = 100.00%, Test Accuracy = 96.00%
QDA:  Train Accuracy = 100.00%, Test Accuracy = 86.67%

Classification on FDA-projected (2D) features (LDA)
LDA on FDA: Train Accuracy = 100.00%, Test Accuracy = 96.00%

 Classification on PCA (90% variance) features (LDA)
LDA: Train Accuracy = 99.00%, Test Accuracy = 95.33%

Classification on PCA (2 components) features (LDA)
LDA: Train Accuracy = 91.67%, Test Accuracy = 93.67%
```

4 Conclusion

This report demonstrates a full classification pipeline for MNIST digits (0, 1, and 2) using only NumPy and Matplotlib. Key contributions include:

- Data loading from IDX files and preprocessing.
- Implementation of MLE for class-wise parameter estimation.
- Dimensionality reduction using PCA and FDA.
- Classification using LDA and QDA from scratch.
- Visualization of transformed feature spaces.

The experimental results indicate that while PCA is effective for dimensionality reduction, FDA further enhances class separation. The choice of classifier and the number of retained principal components significantly impact performance. Future work could involve further parameter tuning and evaluating on larger subsets of the MNIST dataset.