Assignment 2 - Classification and Clustering

***Zhao Jinqiu*[1]**

Southern University of Science and Technology, Department of Biomedical Engineering

Guangdong, People's Republic of China

Abstract

In this assignment, I employed the widely recognized MNIST handwritten digit database, which holds a significant position in the fields of data mining and machine learning. The core task involved the application of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) techniques for dimensionality reduction of the dataset. Additionally, to conduct a comprehensive clustering analysis, both K-means and hierarchical clustering methods were utilized. Upon completing the dimensionality reduction process, Support Vector Machine (SVM) and a basic feedforward neural network were implemented for binary classification tasks. The efficacy of these models was evaluated using cross-validation methods, and to visually represent the analysis results, Receiver Operating Characteristic (ROC) curves were plotted.

Introduction

The assignment for CS303B is focused on classification and clustering using the MNIST handwritten digit database. MNIST dataset is from National Institute of Standards and Technology (NIST), which is an entry-level computer vision data set. The data set is composed of a large number of images of handwritten digits from 250 different people. So it can be used to do handwritten number recognition. This assignment is based on its sub-set: images of digits '3', '6' and '9'. Fig. 1 shows several images in this sub-set.

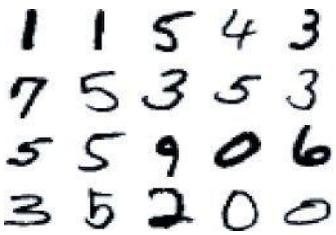

Figure 1: The images of digits in MNIST dataset.

In Part I, I first used PCA to reduce the dimensions of each image to 2. Then I clustered the 2-dimension data points into 3 clusters by K-Means and Hierarchical Clustering. I rewrote the codes of K-Means method instead of using the built-in MATLAB function. Comparing the clustering results of the two different methods, the accuracy is highest 93.76% for K-Means with appropriate start points;93.49% for Hierarchical Method using average linkage. I repeated the procedure in Part I using LDA. The accuracy of k-means clustering is 93.76%The accuracy of Hierarchical clustering is 98.91%

In Part II, I separated the images of '3' from the rest and classified the sub-set with 3 classifiers: SVM with Linear Kernel, SVM with RBF Kernel and Neural Network with one hidden layer. The average accuracy of 5-fold cross validation is 95.44%, 95.68% and 95.83% respectively, and the average AUCs are 0.9656, 0.9895 and 0.9849. In addition, I tuned the RBF kernel parameter γ to improve the classifier performance.The appropriate range of γ is approximately from 0.1 to no more than 15.

Methods and Materials

Materials

MATLAB 2023A ,MNIST dataset.

Methods

1. Dimension Reduction

Dimension reduction is a fundamental technique in data analysis and machine learning, aimed at simplifying complex data sets by reducing the number of variables or features under consideration. This process helps in alleviating issues related to high-dimensionality, such as overfitting and computational inefficiency, commonly known as the "curse of dimensionality".Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two prominentmethods of dimension reduction.

1.1 PCA(Principal Component Analysis)

Principal Component Analysis (PCA) is an advanced unsupervised statistical method that compresses high-dimensional data into a smaller set of 'summary indices', simplifying data analysis and visualization. Its primary aim is to discover orthogonal directions in the data subspace that maximize the variance of the projected data. This technique is crucial for revealing key patterns in complex datasets and reducing data dimensionality, while maintaining the fundamental structure and diversity of the original data.[]

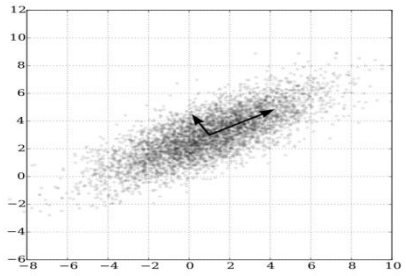

Figure 2: PCA schematic diagram.

1.2 LDA(Linear Discriminant Analysis)

Contrasting with PCA's objective of maximizing the variance across the reduced-dimensional space, Linear Discriminant Analysis (LDA) adopts a different approach.LDA,asupervised dimensionality reductionalgorithm, aims to project the data into a lower-dimensional space where intra-class compactness and inter-class separability are optimized. This is achieved by maximizing the distance between classes while minimizing the variance within each class, as depicted in Fig.3.In essence, LDA focuses on enhancing class discriminability, unlike the typically unsupervised nature of PCA.

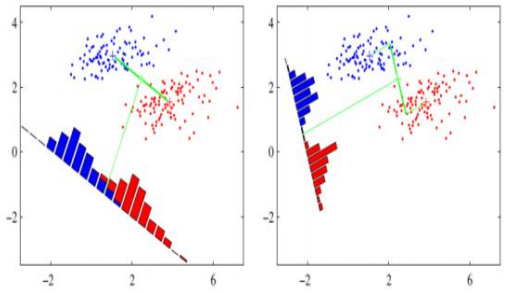

Fig.3 LDA map the data points to a line with maximum between-class distance and minimum within-class distance.

When we need to reduce the dimensionality of three types of data, it is difficult to determine the inter class variance.Let me first introduce a mathematical concept called a scatter matrix.For a dataset containing n samples in m dimensions, it can be represented using an m×n matrix X. That is:

$$\mathbf{X} = [\mathbf{X_1}, \mathbf{X_2}, \ldots \mathbf{X_n}]$$

The overall sample mean can be represented as

$$\overline{\mathbf{X}} = \frac{\mathbf{1}}{\mathbf{n}} \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{Xi}$$

Therefore, the scatter matrix of the data X is an m×m positive semi-definite matrix as follows:

$$\mathbf{S} = \frac{\mathbf{1}}{\mathbf{n}} \sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{Xi} - \overline{\mathbf{X}})(\mathbf{Xi} - \overline{\mathbf{X}})^{\mathbf{T}}$$

The overall walk matrix for all data is

$$S_t = \sum_{j=1}^{n} (x_j - \mu)(x_j - \mu)^T$$

xj traverses the entire dataset, while μi is the mean vector of all data. And the intra-class dispersion matrix Swi for each category of data Ci

$$S_{w_i} = \sum_{x_i \in C_i} (x_i - \mu_i)(x_i - \mu_i)^T$$

And the intra class scatter matrix of t he entire data is

$$S_w = \sum_{j=1}^{N} S_{w_i}$$

Add up the intra class scatter of all N-cla ss data. There exists a relationship betwee n the overall scatter matrix St, the intra-cl ass scatter matrix Sw, and the inter class s catter matrix Sb:

$$S_t = S_w + S_b$$

Finally, our calculation is based on:

$$\lambda w = S_w^{-1} S_b w$$

To maintain the integrity of the Linear Discriminant Analysis, it is crucial to balance the sample sizes across groups. Unequal sample sizes can introduce biases and lead to discrepancies in the results generated by my custom LDA algorithm.

2.Clustering

Clustering method is the data analysis tech -nique used tog roup data points into mult iple sets or clusters. The data points in th ese setshave higher similarity in certain fe atures a-nd lower similarity with points in other sets.

2.1 K-Means Clustering Algorithm

K-means is a popular and widely used clustering algorithm within the realm of unsupervised learning. Its primary aim is to partition data points into K clusters, ensuring that points within each cluster are as similar as possible, while points in different clusters are dissimilar. The steps of the K-means clustering algorithm are as follows:

1.Initialization: Randomly select K data points as the initial cluster centers.

2.Assignment: Assign each data point to the nearest cluster center, forming K clusters.

3.Update: Calculate the mean of each cluster and set it as the new cluster center.

4.Iteration: Repeat the assignment and update steps until a stopping criterion is met, such as no significant change in cluster centers, or a predefined number of iterations is reached.

The K-means algorithm is simple and efficient, particularly suitable for large datasets. However, it has limitations, such as the need to pre-determine the value of K, sensitivity to outliers, potential convergence to local optima, and the assumption of spherical clusters. Therefore, appropriately choosing the value of K and addressing the characteristics of the data is crucial in different application contexts.

2.2 Hierarchical Clustering Algorithm

Hierarchical Clustering is a widely-used method of cluster analysis that seeks to organize data by creating a multi-level, nested cluster structure. Unlike partitioning clustering methods like K-means, which require the number of clusters to be specified in advance, hierarchical clustering does not necessitate this. This method can be categorized into two types:

2.2.1*Agglomerative Hierarchical Clustering*:

1.Bottom-up approach: Initially, each

data point is considered as an individual cluster.

2.The algorithm progressively merges the most similar clusters together.

3.This process is repeated until all data points are aggregated into a single cluster or a certain stopping criterion is met.

2.2.1 *Divisive Hierarchical Clustering*:

1.Top-down approach: Initially, all data points are considered as a single large cluster.

2.This large cluster is then split into smaller clusters.

3.This splitting process continues until each data point is its own cluster or a predefined condition is achieved.

The results of hierarchical clustering are typically represented by a dendrogram, a tree-like diagram showing the relationship of similarity between data points and their clustering structure. Hierarchical clustering is particularly suitable for scenarios where the dataset is not large, and the number of clusters is not predefined. Its main advantages include not having to specify the number of clusters in advance and providing rich hierarchical structural information. However, it has a higher computational complexity and may not be suitable for large-scale datasets.

3.Classification

The most basic approach of machine learning is to use algorithms to parse data, learn from it, and then make decisions or predictions about something in the world.In this Assignment ,SVM classification learner and a simple neural network classifier were used for the classification task.

3.1 SVM(Support Vector Machine)

SVM (Support Vector Machine) is a binary classification model whose basic model is a linear classifier defined with the largest interval in the feature space of the data. Actually, according to different kernel used, SVM models can be generally divided into two types: linear and nonlinear

3.1.1 *Linear Support Vector Machine:*

Linear SVM is a machine learning algorithm used for classification and regression problems, particularly suitable for binary classification problems. Its main goal is to find the optimal hyperplane (linear decision boundary) to separate data points of different categories and maintain the maximum margin on both sides of the hyperplane to improve the model's generalization ability.

The following are the key concepts and working principles of linear SVM:

1.Hyperplane: In two-dimensional space, a hyperplane is a straight line, while in higher dimensional space, it is a linear decision boundary. The equation for a hyperplane can be expressed as: $w^T * x + b = 0$, where w is the normal vector (direction of the hyperplane), x is the eigenvector of the data point, and b is the offset (intercept).

2.Support vectors: Support vectors are the data points closest to the hyperplane, which determine the position and direction of the hyperplane. During the training process, the goal of SVM is to find a hyperplane that maximizes the distance (interval) between the support vector and the hyperplane.

3.Margin: The interval is the distance from the support vector to the hyperplane, and the goal of SVM is to maximize this interval to improve the robustness and generalization ability of the model.

4.Objective Function: The objective of linear SVM is to minimize the norm (L2 norm) of the weight vector w while satisfying the classification correctness constraints for all training samples. This can be expressed as an optimization problem. Figure .4 shows the Schematic diagram of linear support vector machine

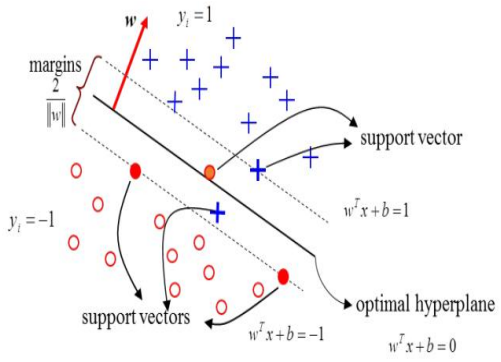

Figure .4 Schematic diagram of linear support vector machine

3.1.2 *Nonlinear Support Vector Machine:*

If the data is not linearly separable, linear support vector machine can map the data to a higher dimensional space by using kernel functions.Radial Basis Function Support Vector Machine (RBF SVM) is a variant of Support Vector Machine (SVM) that uses the Radial Basis Function (RBF) kernel to perform classification or regression tasks.The RBF kernel, also known as the Gaussian kernel, is a widely used kernel function in SVMs. It allows SVMs to transform the input data into a higher-dimensional space where the data points are more likely to be linearly separable. The RBF kernel's formula is K(x, x') = exp(-γ * ||x - x'||^2), where γ (gamma) is a hyperparameter that controls the shape of the kernel, and ||x - x'||^2 represents the squared Euclidean distance between data points x and x'.RBF SVM is a powerful tool for a wide range of machine learning tasks, especially when the data is not linearly separable. However, it requires careful hyperparameter tuning to achieve good performance, and it can be computationally intensive for large datasets. Additionally, overfitting can occur if the hyperparameters are not chosen appropriately. Cross-validation and grid search are common techniques for hyperparameter tuning with RBF SVM.

3.2 Neural Network

Neural Networks (NN) are computational models that mimic the structure and function of biological neural networks. The models do computations using a large number of artificial neurons, and they can change the internal parameters on the basis of external information.

1.Neuron: The basic unit of a neural network that simulates the working principle of biological neurons. Each neuron receives multiple inputs, performs weighted summation, and performs non-linear transformation on the results through an activation function to generate an output. Activation functions are commonly used to introduce nonlinear properties to enable neural networks to learn complex functional relationships.

2.Layer: Neural networks typically consist of multiple layers, divided into input layer, hidden layer, and output layer. The input layer receives the raw data, the output layer produces the model's prediction results, and the hidden layer performs complex feature transformations and pattern learning within it. Deep neural networks refer to neural networks that contain multiple hidden layers.

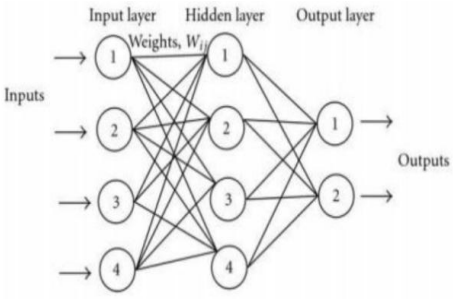

Figure.5 Schematic Plot of Neural Network.

3.Weights and Biases: Each connection (or edge) has a related weight used to adjust the importance of the input signal. Each neuron also has a bias term to adjust the threshold of the activation function. These weights and biases are parameters of the neural network, which are learned and optimized through the training process to enable effective model fitting.

4.Forward Propagation: Forward propagation is the process used by neural networks to calculate prediction results. The input signal is transmitted from the input layer to the hidden layer and output layer, and is subjected to weighted and nonlinear transformation through the weights and biases of neurons, ultimately generating prediction results.

5.Backpropagation: Backpropagation is a crucial step in neural network training. It uses a loss function to measure the prediction error of the model, and then calculates the gradient of each parameter through a chain rule (chain differentiation). These gradients are used to update weights and biases to minimize the loss function and make the network's predictions closer to the actual labels.

6.Loss Function: The loss function is used to measure the difference between the model's predictions and the actual labels. The goal is to minimize the loss function to obtain the optimal model parameters.

Cross entropy is a commonly used method in machine learning to measure the difference between the model output and actual labels, especially in classification problems. It is a concept from information theory that measures the discrepancy between two probability distributions.In machine learning, cross entropy is often used as a loss function, particularly in classification problems. For binary classification, the cross entropy loss function can be expressed as:

$$H(y,p) = -\sum(y\log(p) + (1-y)\log(1-p))$$

Here, y represents the actual labels, and p represents the probabilities predicted by the model. For multi-class classification, the cross entropy loss function can be expressed as:

$$H(y,p) = -\sum y\log(p)$$

In this formula, y is typically a one-hot encoded vector representing the actual class, and p is the probability distribution output by the model.One of the main advantages of the cross entropy loss function is its sensitivity to small changes in the model's output. The loss increases rapidly when the predicted probabilities deviate from the true labels. This allows the model to learn and adjust its parameters quickly, especially during the initial stages of training.In practical applications, such as in neural networks, cross entropy is often combined with the softmax function as the final layer. The softmax function converts the neural network's output into a probability distribution, and the cross entropy measures the difference between this predicted distribution and the actual distribution.

4.Cross Validation

In practice, utilizing the entire dataset for model training is often not feasible, as it would leave no data for model validation, crucial for assessing its performance. However, partitioning a portion of the data as a fixed test set, separate from the training data, can lead to an inefficient utilization of available data.To address this issue and maximize data utilization, a widely employed technique is cross-validation. I will introduce a specific cross-validation method utilized in this assignment, known as K-fold cross-validation. The procedure involves partitioning the available dataset into K equally sized subsets, or "folds." The model is then trained and validated K times, each time using a different fold as the validation set and the remaining K-1 folds for training. This process ensures that every data point is used for validation exactly once over the K iterations.

After completing K iterations, the performance metrics (e.g., accuracy, error, or any relevant evaluation metric) are averaged to obtain a more reliable estimate of the model's performance. K-fold cross-validation

helps mitigate the variance associated with a single train-test split, providing a more comprehensive assessment of how well the model generalizes to unseen data.

Common choices for K include 5-fold and 10-fold cross-validation, but the selection depends on the dataset size and the specific problem at hand. K-fold cross-validation is invaluable for model selection, hyperparameter tuning, and detecting overfitting or underfitting, making it an essential tool in the evaluation of machine learning models.

5. ROC curves

Receiver Operating Characteristic (ROC) curves, often referred to as ROC curves, are graphical representations used in machine learning and statistics to assess the performance of binary classification models. These curves illustrate the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity) across different thresholds for a given classification model. ROC curves are particularly useful for evaluating and comparing the performance of various classifiers or models.The main characteristics of the ROC curve are as follows:

1.The perfect classifier positions the curve in the upper left corner (0,1) and can completely distinguish between positive and negative examples.

2.The curve close to the upper left corner indicates good model performance, high sensitivity, and low false positive rate.

3.The curve that tends towards the diagonal (45 degree angle) represents random guessing or poorly performing models, making it difficult to effectively distinguish between positive and negative examples.

4.The area under the ROC curve (AUC ROC) is typically used to quantitatively evaluate classifier performance. A AUC value close to 1 indicates good performance, while a value close to 0.5 indicates poor performance.

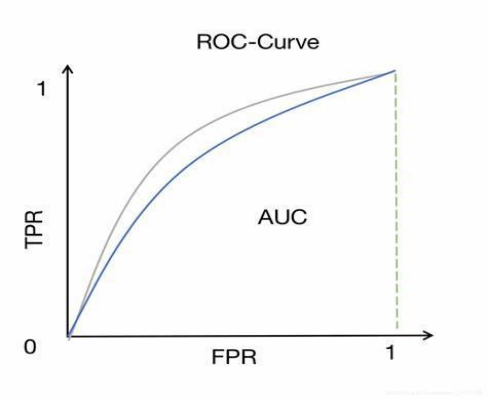

Figure.6 Schematic Plot of ROC curve.

Results and Conclusion

Part I: Dimension Reduction and Clustering.

In this part, I load the minst data and use the matlab function zscores() to do the preprocessing and then use the function pca() to do the Dimension Reduction.After that I performed K-mean clustering and hie-rarchical clustering on the processed data, and the results are shown below.

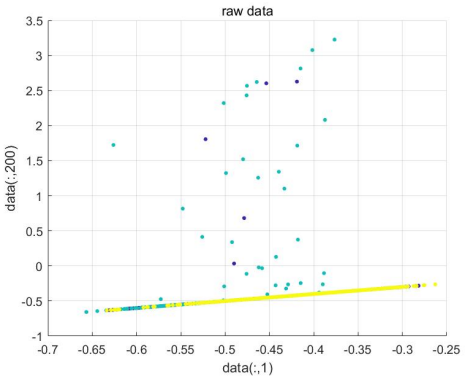

Figure.7the raw data.

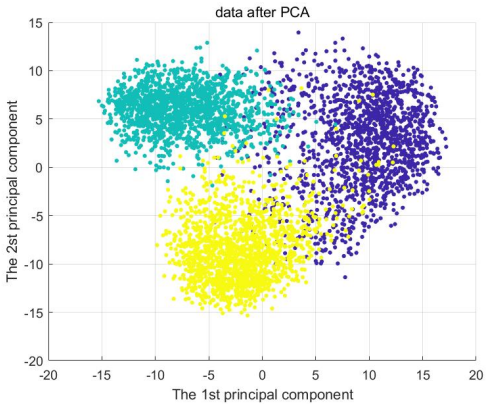

Figure.8.the data after pca.

Because the clustering results cannot reflect the distribution of real labels, I have adopted **a label mapping method** based on data indexing to determine which labels correspond to different clusters. By combining the contour coefficients of clustering, this method is used to analyze the clustering effect. Figure 9 shows the pca data after the k-means clustering with the true labels. And the contour coefficients of this clustering is 0.7784.The accuracy of kmean clustering is 93.76%. Figure.10 and Figure.11, shows the dendrogram Results and the pca data after the Hierarchical Clustering with the true labels. And the contour coefficients of this clustering is 0.7484.The accuracy of Hierarchical clustering is 93.49%

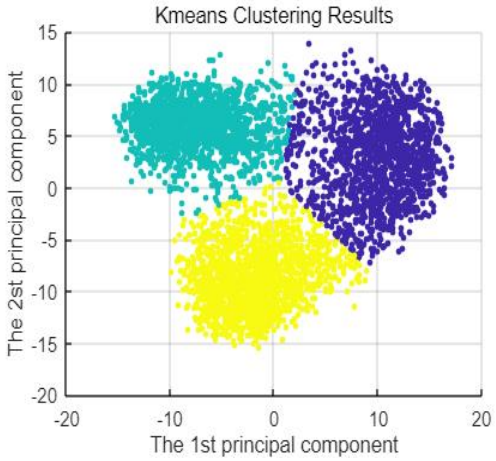

Figure.9 Kmeans Clustering Results

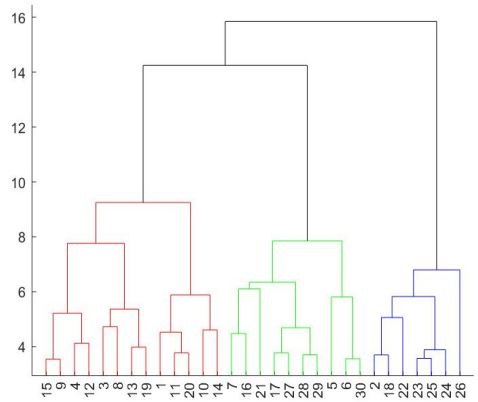

Figure.10.Dendrogram Results

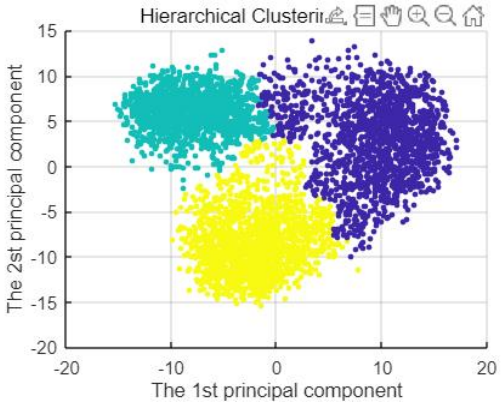

Figure.11 Hierarchical Clustering Results

Now.I use LDA instead of PCA for dimensionality reduction and repeat the steps.In this section, I manually implemented the lda algorithm myself.The code can be found in the appendix.

The figure 12 shows the data after LDA method. The figure 13 ,figure 14 and figure15 shows the results of different clustering methods

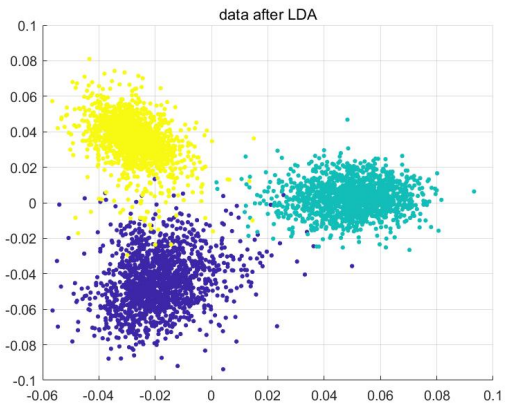

Figure.12 the data after LDA.

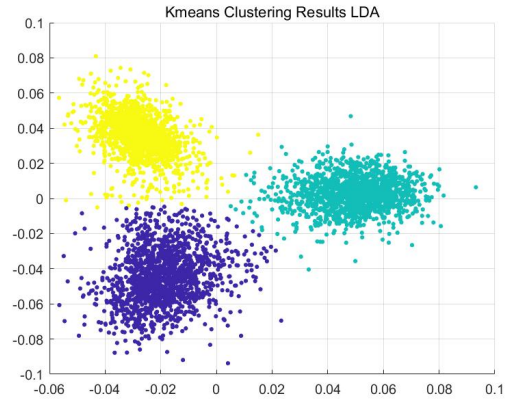

Figure.13 Kmeans Clustering Results

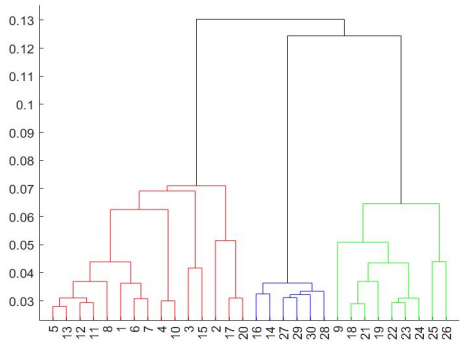

Figure.14 Dendrogram Results

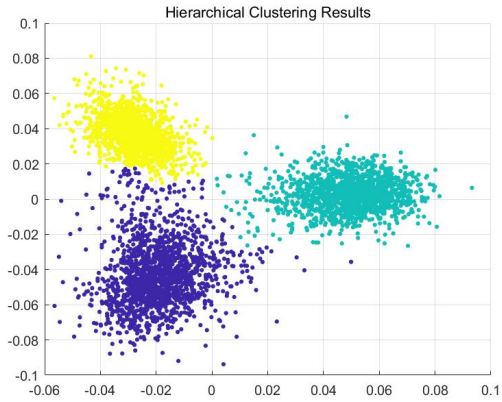

Figure.15 Hierarchical Clustering Results

The contour coefficients of k-means clustering is 0.8984.The accuracy of k-means clustering is 93.76%;the contour coefficients of Hierarchical clustering is 0.9078.The accuracy of Hierarchical clustering is 98.91%.

Part II: Binary Classification

In this part, I divide the dataset by digit3 against others firstly and then use the data after pca to do the binary classification. I also use 5-fold cross-validation in this part to get a authentic accuracy to evaluate the model and try to plot the ROC curves and calculate the area under the curve (AUC) by the matlab function perfcurve(). The first kind model is SVM,Figure16 shows the results of SVM with RBF kernal. I fixed the hyper-parameters C equal to 1 and use the fitsvm to get the best gamma(KernelScale).The following figures shows the results.

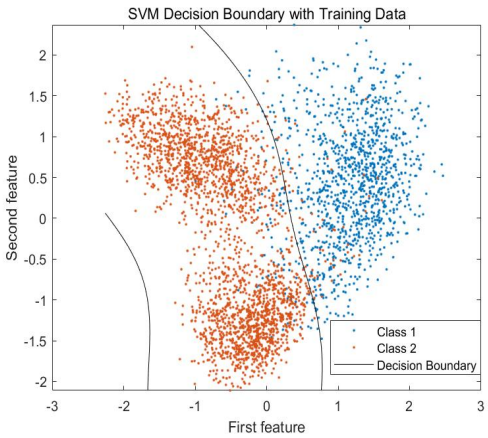

Figure.16.RBF-SVM Decision Boundary with Training Data

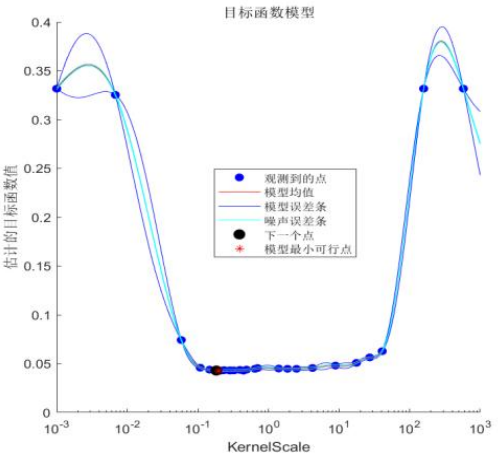

Figure.17 visualization of parameter optimization process.

Figure 16.shows the RBF-SVM Decision Boundary with Training Data.Figure 17 shows the impact of kernel scale parameters on errors.The horizontal axis (X-axis) represents "KernelScale", which is the scale parameter of the kernel function. It is on a logarithmic scale, covering the rangefrom 0.001 to 1000.I found that the KernelScale is not much difference between 0.1 and 10.and the best Gamma(KernelScale) is 0.2021.The accuary on test set is 95.44% and here is the ROC curve,with the AUC is 0.9609.

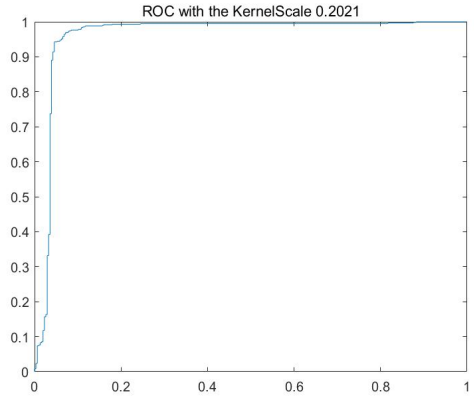

Figure.18 ROC curve with RBF SVM

Figure19 shows the scatter plot after classification. Figure20 shows the ROC curves of the linear SVM. the accuary on test set is 95.80%

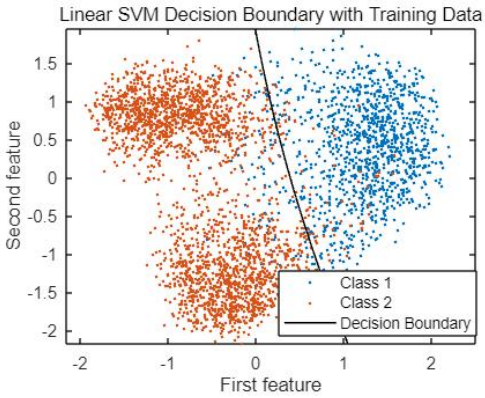

Figure 19 the scatter plot of the linear SVM.

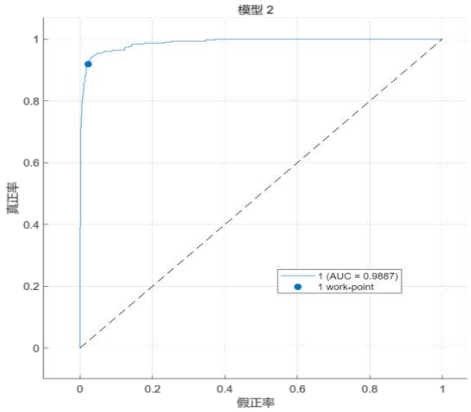

Figure.20 The ROC curves of the linear SVM.

The remaining requirement for this section is to use a neural network with a hidden layer for model training. Figure22shows my model structure.I used 5-fold cross validation and train 10000 epochs.The accuary on this model is 95.68%.Figure 21 shows the ROC curve and the AUC is 0.9866.

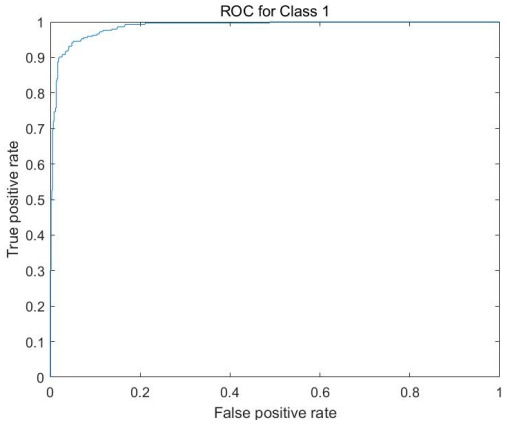

Figure.21 the structure of my NN

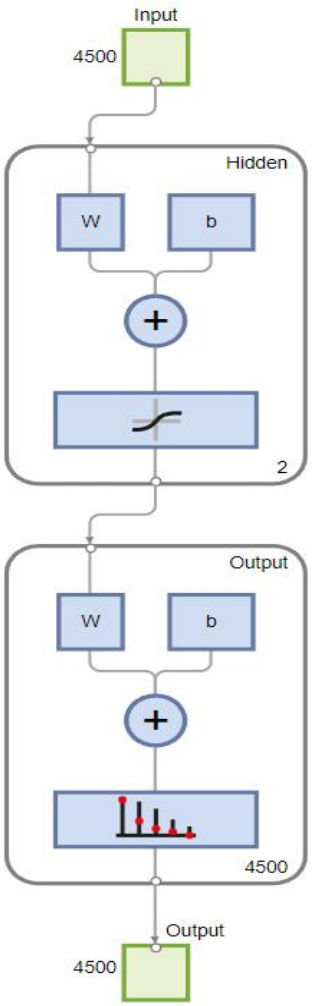

Figure.22 the structure of my NN

Repeat

Then I used the subset data1, data5, and data7 to repeat the assignment. The results showed that except for the significantly different results in feature dimensionality

reduction especially PCA, the classification performance was roughly the same. All three classification models had high accuracy (95%+) and large AUC values (0.94+). These results indicate that the model is not only applicable to specific datasets, but can also effectively handle data from different sources or types.

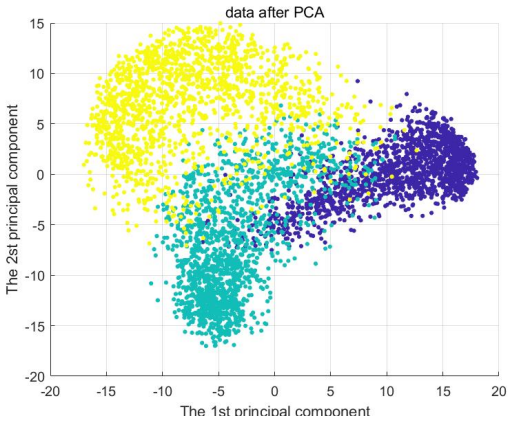

Figure23.PCA on digit1,digit5 and digit7.

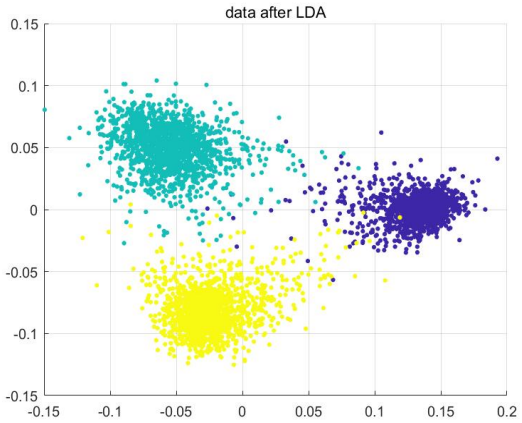

Figure24.LDA on digit1,digit5 and digit7.

Attached code

LDA:

```
X = data';
% Set the number of samples and features
ns = 4500;
nf = 784;
% Set the target dimension for LDA
targetdim = 2;
% Regularization parameter to avoid singularity
issues
lambda = 1e-3;
% Calculate the total scatter matrix
X_ave = mean(X);
st = (X - X_ave)' * (X - X_ave);
% Separate the dataset into three classes
C1 = X(1:1500, :);
C2 = X(1501:3000, :);
C3 = X(3001:end, :);
% Calculate means for each class
xC1 = mean(C1);
xC2 = mean(C2);
xC3 = mean(C3);
% Calculate within-class scatter matrices
sC1 = (C1 - xC1)' * (C1 - xC1);
sC2 = (C2 - xC2)' * (C2 - xC2);
sC3 = (C3 - xC3)' * (C3 - xC3);
% Sum up within-class scatter matrices and add
regularization term
sw = sC1 + sC2 + sC3 + lambda * eye(nf);
% Calculate between-class scatter matrix
sb = st - sw;
% Solve the generalized eigenvalue problem
[V, D] = eig(sw \ sb);
% Sort eigenvalues and select the top eigenvectors
[~, sortedindices] = sort(diag(D), 'descend');
eigvec = V(:, sortedindices(1:targetdim));
% Project the data onto the new lower-dimensional
space
Xldaself = X * eigvec;
```