MVE441 Statistical learning for big data - Take-home examination-Q2

Sky Sunsaksawat

June 2024

1 Introduction

Dataset2

For this question you will use a variant of the MNIST digits data that I have created/manipulated in various ways. The data set is quite large with 50000 28*28 raster scan images. The first column of the data matrix is the label.

Q-2a

The sample of MNIST dataset before dimension reduction are shown in the figure 1. GridSerachCV was used to select the number of component.The classifier that was used to test is Support Vector Classifier (SVC). The optimal number of components are 50 for PCA, and 60 for NMF and kernelPCA. The set of dimension reduction techniques which are utilized in this subtask are the following:

- Linear dimension reduction

1. Principle Component Analysis (PCA)
2. Non-negative matrix factorization (NMF)

- Non-linear dimension reduction

1. kernel Principle Component Analysis (kernelPCA)

Please note that color range in visulaization are the same for the original dataset, the PCA dimension reduction dataset and the NMF dimension reduction dataset. For the kernelPCA dimension reduction dataset is different in order to provide better visualization.

• **The difference between method**

PCA is a linear transformation that identifies the directions of greatest variance in the data.These directions are called principal components and they are orthogonal, which means they are independent of each other.The result after utilized this technique are shown in figure2.

In terms of NMF, it is also a linear transformation, but it has the added constraint that the resulting factors must be non-negative. This can be useful for data that naturally has non-negative values, such as images or text data. NMF decomposes the data into two matrices, W and H, which represent the parts of the data.Figure 3 is shown the result after perform NMF on original data with 60 components.

Lastly, Kernel PCA is a non-linear transformation technique. This is heavily resource consumption. Directly perform kernelPCA are not the great approach.It uses a kernel trick to map the data into a higher-dimensional space where it is possible to find linear relationships between the data points. Once the data is mapped into this higher-dimensional space, PCA can be applied to find the principal components.

To apply kernelPCA to our large dataset, iterative resampling technique is introduced to prevent over resource. This non-linear technique samples and trains 20% of observations per iteration. After that, the trained model will then transform the entire dataset. Lastly, the results are averaged.

• **Class separation**

PCA identifies regions that contribute the most to overall variance. As we can see in figure2, some regions are more intense compared to other. Then, considering on NMF technique, as shown in figure 3, in my opinion, NMF perform

best in capture the characteristics of number and identifies specific regions that are critical for class differentiation. While kernelPCA,as shown in figure 4, are not clear in capturing the characteristic of the number, but importantly provide some regions that are useful in class separation.

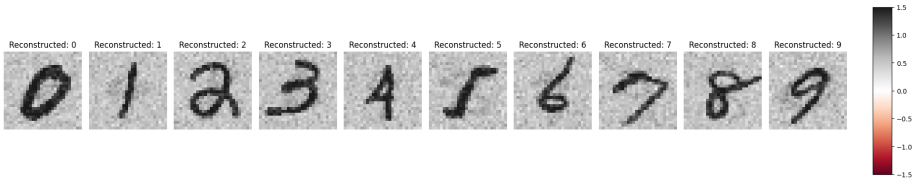

Figure 1: The original dataset

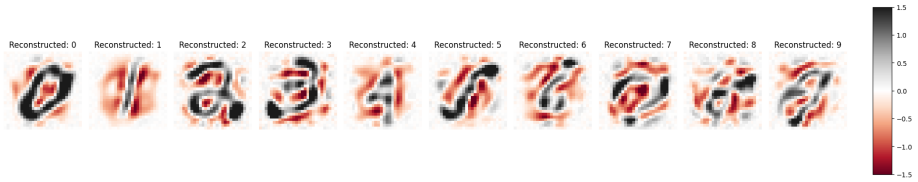

Figure 2: The dataset that are reduced dimensional by PCA

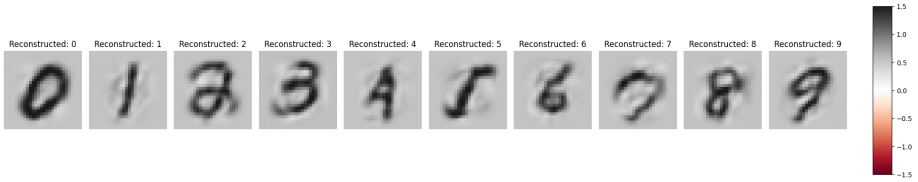

Figure 3: The dataset that are reduced dimensional by NMF

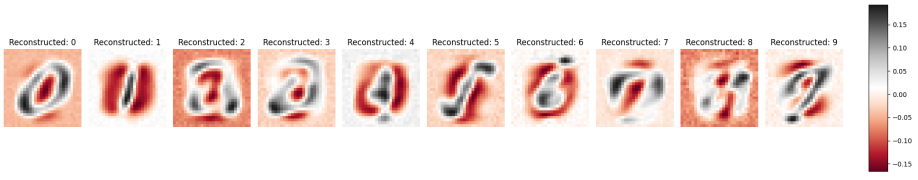

Figure 4: The dataset that are reduced dimensional by kernelPCA

Q-2b

To complete this task, the data that used in trained a model are from the kernelPCA reduction technique, which have dimension (50000,60). The set of classifier are performed here consist of

1. SGDClassifier
2. MLPClassifier
3. Support Vector Classifier(SVC)
4. KNearestNeighborsClassifier(KNN)
5. RandomForestClassifier

Each classifier have different characteristic. SGDClassifier is suitable for large-scale dataset where speed is critical factor, but it has limitation to handle complex patterns. MLPClassifier works well with a lot of data. It can utilize the neural network for complex tasks. For SVC and KNN, they are both accurate and work well with different amounts of data. SVC is especially good for complex patterns.The last classifier is RandomForestClassifier, which is robust and reliable when as more data becomes available.

• **The dimensional dataset issues**

As we can see from figure 5, the label distribution has minor imbalances. Due to the large dataset, resampling process to re-balance data are not considered. To train model, sampling and batching data strategy are applied to avoid excessive computational consumption.

• RandomForestClassifier, the sub-sampling once technique are utilized as to reduce time consuming in training data. The sampling size used in process is 10000. However, this method is difficult to ensure that subsampling is representative.

• SGDClassifier and MLPClassifier are support batching training. So, we can easily train the model with batching. The batch size used in training is 128.

• SVC and KNN do not have any batching or sampling method support. Luckily, the computational issue due to data size are not the main problem for these classifier. It can be simply directly train the model with all train set.

• **Performance**

Figure 6 depicts accuracy over the training size for each classifier. When the training size is minimal, SGDClassifier and MLPClassifier's performance are substantial low. SGDClassifier typically require larger datasets to generalize well and the perceptron loss in this project cannot capture complex patterns in the data. So, there is minor improvement as the training size increasing. MLPClassifier might unperformed due to insufficient data. Performance of this classifier improves significantly with larger sample sizes as we can see in the plot. SVC outperform in all training size due to the non-linear RBF kernel that used in this problem. KNN shows consistency high accuracy across sample size suggesting that the distance-based approach is very effective for this

dataset. RandomForestClassifier are robust and capable of handling a wide range of datasets. It tends to improve with additional data, but may still not show significant improvements.

• **Challenges**

During classifying dataset, the model keep predicting the wrong label even the accuracy is high as we can see in the confusion matrix of SVC, which have the highest performance, in the figure 7. So, to ensure that there are mislabeling observations or noise in the dataset, which make the accuracy of each model lower, the confidence prediction are utilized to see the result when model predicts with very high confidence. Figure 8 can confirms that the mislabeling data points play vital role in our dataset. The first third columns can be obviously seen the ground truth number. Nevertheless, the true labels do not match the reality. Furthermore, the two leftmost columns indicate that noises also disturb model's accuracy. The bottom row of this figure is the images after perform kernelPCA. We can see that the model predicts the wrong number from the ground truth. However, these pictures are also mislabeling.

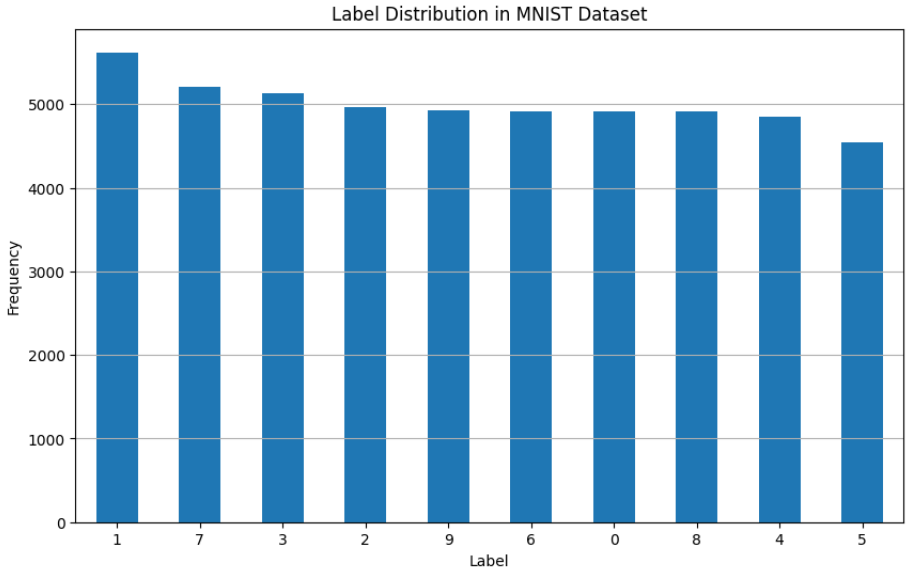

Figure 5: The label distribution of MNIST dataset

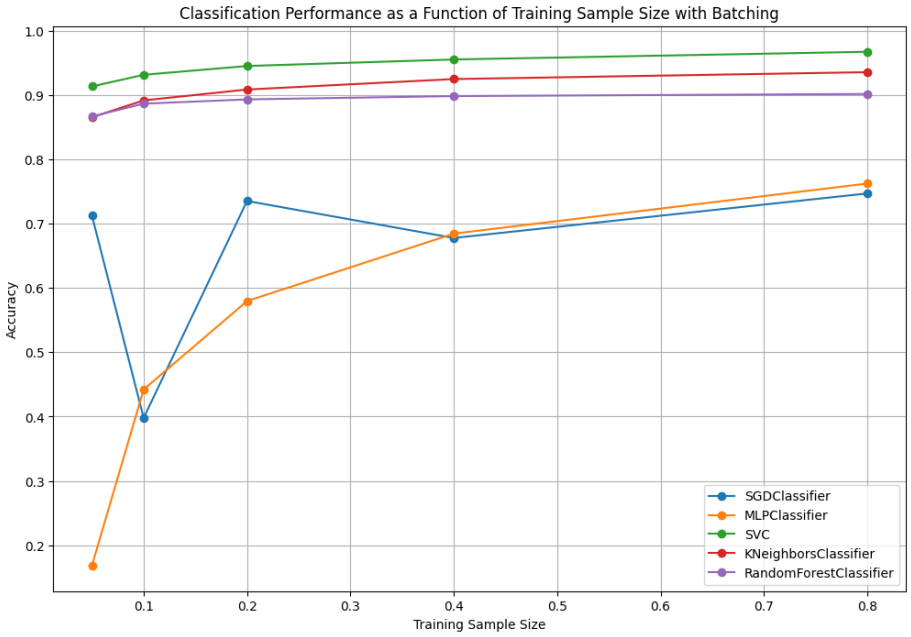

Figure 6: Classification accuracy as a function of training size

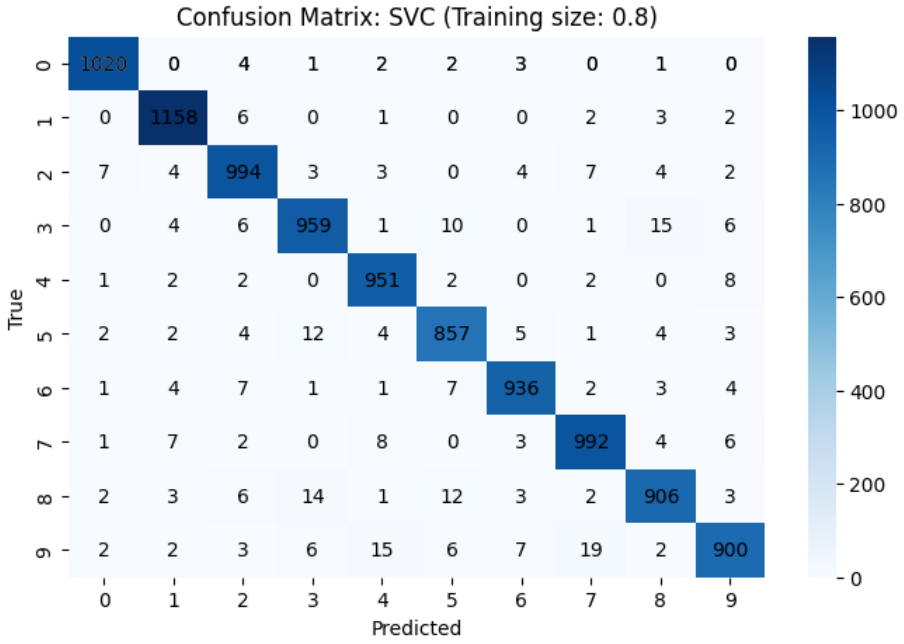

Figure 7: Confusion matrix of SVC classifier, training size = 0.8

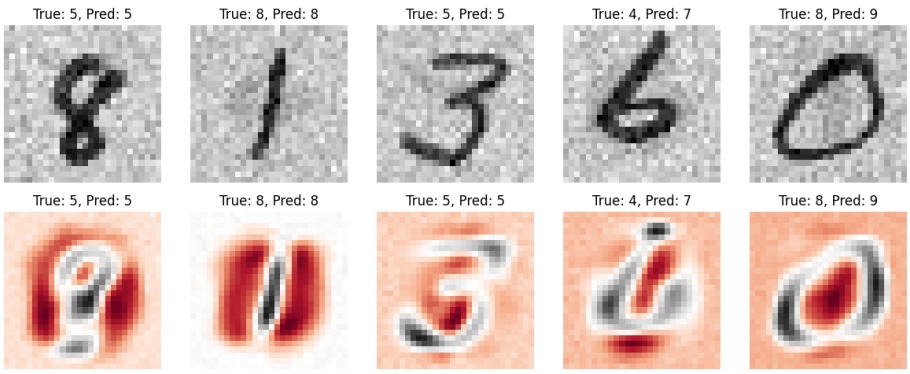

Figure 8: Sample of observations that incorrect prediction with high confidence and its dimension reduction with kernelPCA images

Q-2c

In this task, the clustering algorithms used here consist of 3 different methods.

1. Kmeans clustering
2. Agglomerative Clustering
3. Gaussian Mixture Clustering

Kmeans works well for spherical and well-separated clusters but it can get confused by clusters that overlap or have strange shapes.Agglomerative Clustering is better for these odd-shaped clusters, but it can be easily tricked by errors or unusual data points. Gaussian Mixture Clustering is more flexible and can handle clusters that overlap or not perfectly round. However, it may struggle with clusters that do not conform to a bell-shaped curve.

- **The number of cluster selection (k)**

In order to select the number of cluster, 3 different methods including, Elbow heauristic method, Silhoulette score and Davies-Bouldin Index are used as shown in figure 9 . Starting from Elbow's method, this plot shows the within-cluster sum of squares (WCSS) decreasing as the number of clusters increases.The elbow point is where the rate of decrease sharply slows down. From my perspective, it is difficult to decide the best k from this plot. Therefore, the Silhouette score plot and Davies-Boudlin Index are also introduced to help made decision. Silhouette score shows the average silhouette score for each value of k. A higher silhouette score indicates well-separated clusters.As we can see, the suitable k from Silhouete graph is quite high and not practical. For Davies-Boudlin index, a lower value refer to better clustering. It starts stable around k = 10 to 15. Therefore, from the Davies-Boudlin index showing stability and the Silhouette Score gradually increasing without the requirement for an excessive number of clusters, k = 10 is the optimal number to cluster the dataset.

- **Data utilization**

The data set used in clustering is reduced dimension by non-matrix factorization (NMF) from original dimension becoming (50000,60). To perform clustering on the large dataset, batching of data are applied. The strategy using here involves a two-step process: local clustering followed by global clustering. Begining with the local clustering, the data is divided into smaller batches.The smaller groups are then clustered using a technique such as K-means. The centroids of each group are then individually gathered. Next, all of the centroids collected from the smaller groups are combined before the 3 clustering algorithms consist of Kmeans, Agglomerative and Gaussian Mixture technique are utilized to cluster these centroids. Finally, the global cluster labels are allocated to the original data points based on their nearest local centroid. The clustering results are shown in figure10.To visualization, the reduced dimension data are applied by t-SNE plot.The original data is just for comparison.

- **Retrievable data**

To answer this topic, "classe" in this answer are refereed to the original data cluster in figure 10. Class "1" has high potential for retrieval because it is well-separated, dense, and has regular shapes. It is well clustered using the Gaussian Mixture algorithm. The secondary potentials are "6" and "9" on the right side of the graph. They are also tightly cluster by all three algorithms, but they cannot be separated. The possible reason is because they may share some characteristics between classes. "5","8" also has a chance to be retrievable because it is well clustered in the slightly upper middle of the graph. However, they might share properties in common, similar to "6" and "9". On the other hand, "0","2","3","4" and "7" might be unable to retrieve because their clusters typically overlapping and are irregular shapes in all algorithms.

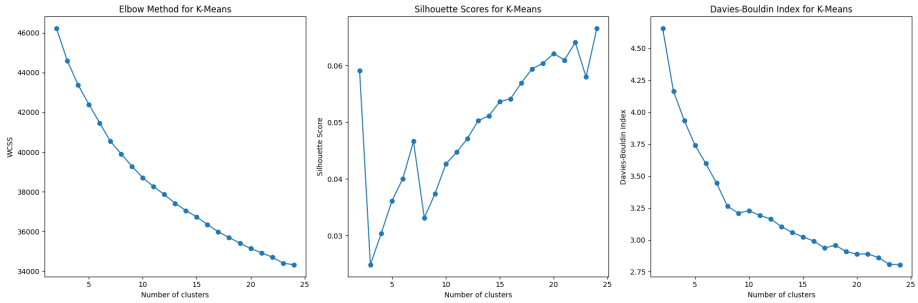

Figure 9: This figure shows the Elbow Method, Silhouette Score, and Davies-Bouldin Index for the number of cluster selection

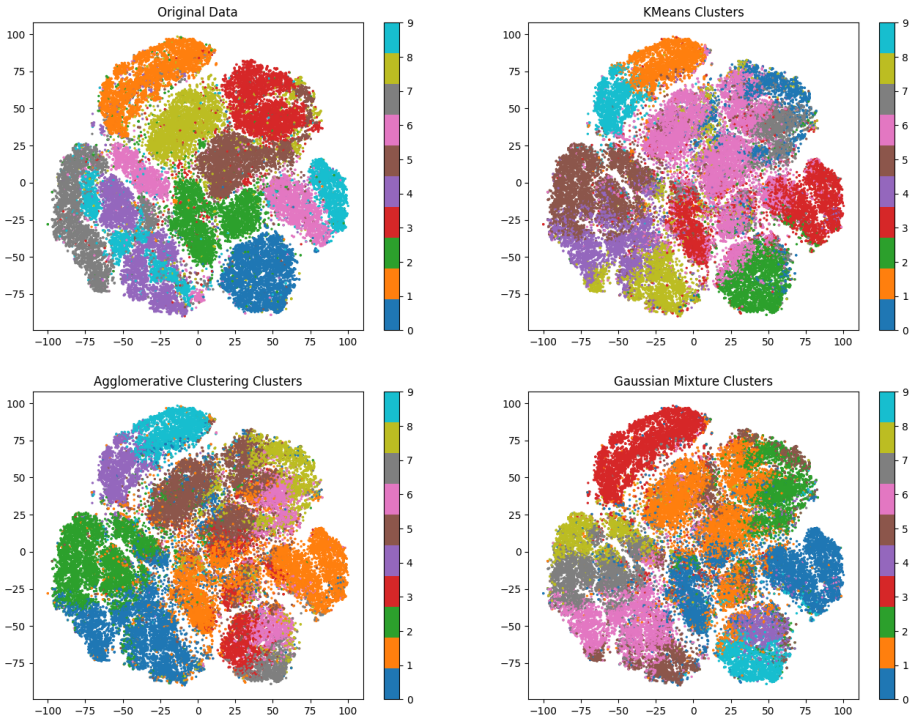

Figure 10: This figure shows the original dataset and 3 different clustering algorithms: Kmeans, Agglomerative and Gaussian Mixture. Visualisation by using t-SNE