Rodolfo Cota Borbon

301288986

Assignment 1: Dimensionality Reduction using PCA

**Analysis Report**

**Part A**

1. **Retrieve and load the**[**mnist\_784**](https://www.openml.org/d/554)**dataset of 70,000 instances.**
2. **Display each digit.**
3. **Use PCA to retrieve the first and second principal component and output their *explained variance ratio*.**

After applying PCA to the normalized pixel features of the MNIST dataset, the results show that the first principal component captures the most variance, accounting for 9.75% of the total variance. The second principal component captures

1. **Plot the projections of the first and second principal component onto a 1D hyperplane.**

**A comparison of a graph

Description automatically generated**

According to the image above, the first principal component captures the most variance in the data, which is why its distribution is more spread out, ranging from approximately -1 to 10. In comparison, the second principal component’s distribution ranges from -5 to 5. A more spread-out distribution indicates higher variance, while a more concentrated distribution indicates lower variance.

1. **Use Incremental PCA to reduce the dimensionality of the MNIST dataset down to 154 dimensions.**
2. **Display the original and compressed digits from (5).**

**A black and white logo

Description automatically generated**

According to the image above, after reducing the original dataset from 784 (28x28 pixels) to 154 components using Incremental PCA, we observed that this approach effectively preserved the essential features that define the digits, making them recognizable. Despite some loss of detail, the overall structure and shape of the digits remain intact. This indicates that Incremental PCA successfully captured the most important features of the data that captures the majority of the variance.

1. **Create a video discussing the code and result for each question. Discuss challenges you confronted and solutions to overcoming them, if applicable.**

**Part B**

1. **Generate Swiss roll dataset.**
2. **Plot the resulting generated Swiss roll dataset.**

A graph of a graph with colored dots

Description automatically generated with medium confidence

1. **Use Kernel PCA (kPCA) with linear kernel, a RBF kernel, and a sigmoid kernel.**
2. **Plot the kPCA results of applying the linear kernel, a RBF kernel, and a sigmoid kernel from (3). Explain and compare the results.**

A diagram of a number of dots

Description automatically generated with medium confidence

The image above shows the results of applying Kernel PCA to the Swiss roll dataset using three different kernels: linear, RBF, and sigmoid.

* **Linear Kernel**: The linear kernel does not effectively unroll the Swiss roll. This is because it is designed for datasets with linear relationships and cannot capture the complex non-linear structure of the Swiss roll.
* **RBF Kernel**: The RBF kernel successfully unrolls the Swiss roll into a 2D representation. This kernel is well-suited for non-linear datasets, revealing patterns and clusters within the data. Although some points are sparse, the overall structure is more interpretable.
* **Sigmoid Kernel**: The sigmoid kernel performs similarly to the linear kernel, failing to unroll the Swiss roll. It retains a multi-dimensional representation, indicating it is not as effective as the RBF kernel for this type of data.

In summary, the RBF kernel is the most effective for unrolling the Swiss roll into a 2D space, making it easier to analyze and interpret complex data.

1. **Using kPCA and a kernel of your choice, apply Logistic Regression for classification. Use *GridSearchCV* to find the best kernel and*gamma* value for kPCA in order to get the best classification accuracy at the end of the pipeline. Print out best parameters found by *GridSearchCV*.**
2. **Plot the results from using *GridSearchCV* in (5).**

A graph with lines and numbers

Description automatically generated

The graph shows the results of a GridSearchCV process used to find the best parameters for a classification model through cross-validation. By examining the lines, we can compare how each kernel performs across different gamma values.

For instance, one of the peaks of the RBF kernel line is higher than the others at a certain gamma value, achieving about 0.528 in mean test accuracy. This indicates that the RBF kernel generally performs better for this dataset. Additionally, as observed in step 4, where we plotted the application of PCA with different kernels, the RBF kernel effectively unfolds this dataset, making it suitable for this type of non-linear, complex data. This effectiveness is also reflected during the regularization step, confirming that the RBF kernel is still the best choice for this complex, non-linear dataset.

Therefore, we should focus on fine-tuning the gamma parameter to further improve the mean test accuracy, rather than spending time on other kernels. This approach will help speed up the regularization step.

1. **Create a video discussing the code and result for each question. Discuss challenges you confronted and solutions to overcoming them, if applicable.**