PRINCIPAL COMPONENT ANALYSIS

MILAN KRESOVIĆ – S266915 ERASMUS STUDENT

INTRODUCTION

Principal component analysis (PCA) is an unsupervised method used for dimensionality reduction in data. It transforms a large set of variables into a smaller one that still contains most of the information from the large set called **principal components**.

This homework was done on the dataset of 400 images of four different classes. Each image is of the size 227x227 and in the RGB format, so the total number of features in each image is 227x227x3. The data was loaded and PCA was applied on it. The result of PCA are principal components in order of significance, which were used to discuss how well is the image reconstructed using different number of principal components. Dataset was also projected onto the graph for different pairs of PC and further discussed for which PC is the variance the biggest.

Naïve Bayes Classifier was used to classify the dataset (that has been divided into the training and testing set) for different pairs of PC to see how does the accuracy changes with changing the combination of the pairs of principal components. Later, classified models from the classifier were used to visualize decision boundaries for the first two principal components.

DATA PREPARATION

All images were loaded and converted into the 154587-dimensional vector (227x227x3). Every vector was stacked into the matrix **Data** which was of size 400x154587 i.e. 400 rows representing each image and the columns representing all the features. While loading Data matrix each image was given a label which represents what category it belongs. This was saved in the **Class** array.

Just before applying the PCA, Data matrix had to be standardized. That is, each feature has a zero (0) mean and unit (1) variance like the standard normal distribution. Because PCA is a variance maximizing exercise having data standardized helps improving the results.

EXTRACTION OF THE PRINCIPAL COMPONENTS

PCA was done for extracting the principal components of the whole dataset and then re-projected for only the first 60 PC/ 6 PC/ 2 PC and the last 6 PC.

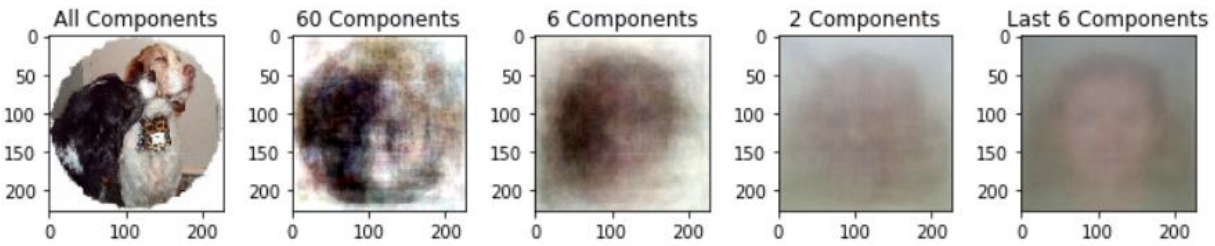

Figure 1. Reconstruction of the image using different number of principal components

What can be seen on the figure 1 is that using all the components, algorithm is able to reconstruct well the whole image, whereas by using the first 60 components we get a slightly worse reconstruction, but that is still recognizable. That way we can assume that it is possible to use less than all the components to reconstruct the original image well. The interesting question is how many principal components is needed to acquire fairly good reconstruction of the image.

This can be answer by comparing cumulative sum of the explained variance ratio for all the components.

CHOOSING RIGHT NUMBER OF THE COMPONENTS

To get fairly good estimation of how many principal components are needed to well describe the data, we should perform cumulative explained variance ratio as the function of the number of components.

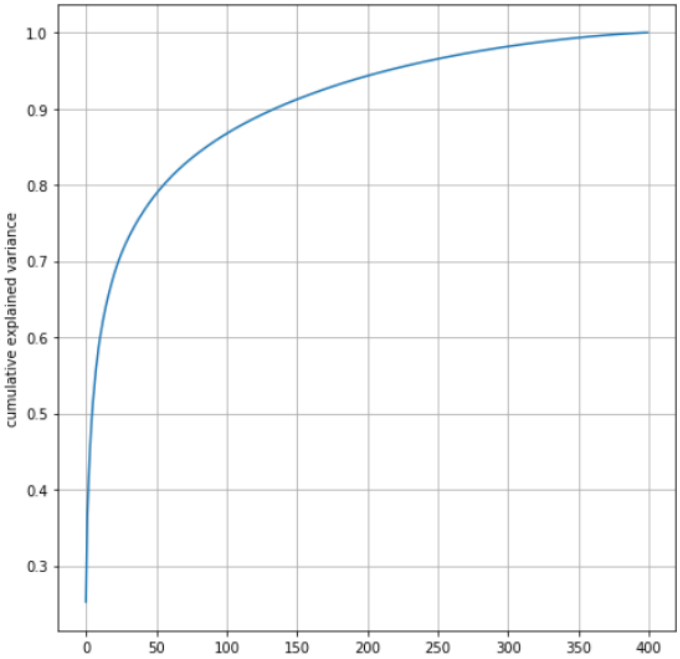

Figure 2. Cumulative explained variance ratio as the function of the number of components

Curve in the figure 2 quantifies how much of the total variance is contained in the first N components. Choosing only the first two principal components can't get us anywhere given that it comprises only around 30% of the total variance, but choosing all the components to get 100% doesn't totally serve to the purpose of the PCA which is dimensionality reduction.

What can be seen from this graph is that choosing more than the first 130 principal components can get us having around 90% of the variance. So deciding how many components is needed for the good reconstruction of the original image can be reformulated as the question how many cumulative explained variance ratio is good enough for us.

VISUALIZING DATA USING SCATTER-PLOT

Scatter-plot was used to plot principal component values with different colors for different classes. First graph is for first two PC, second graph is for third and fourth PC and last one is for tenth and eleventh PC.

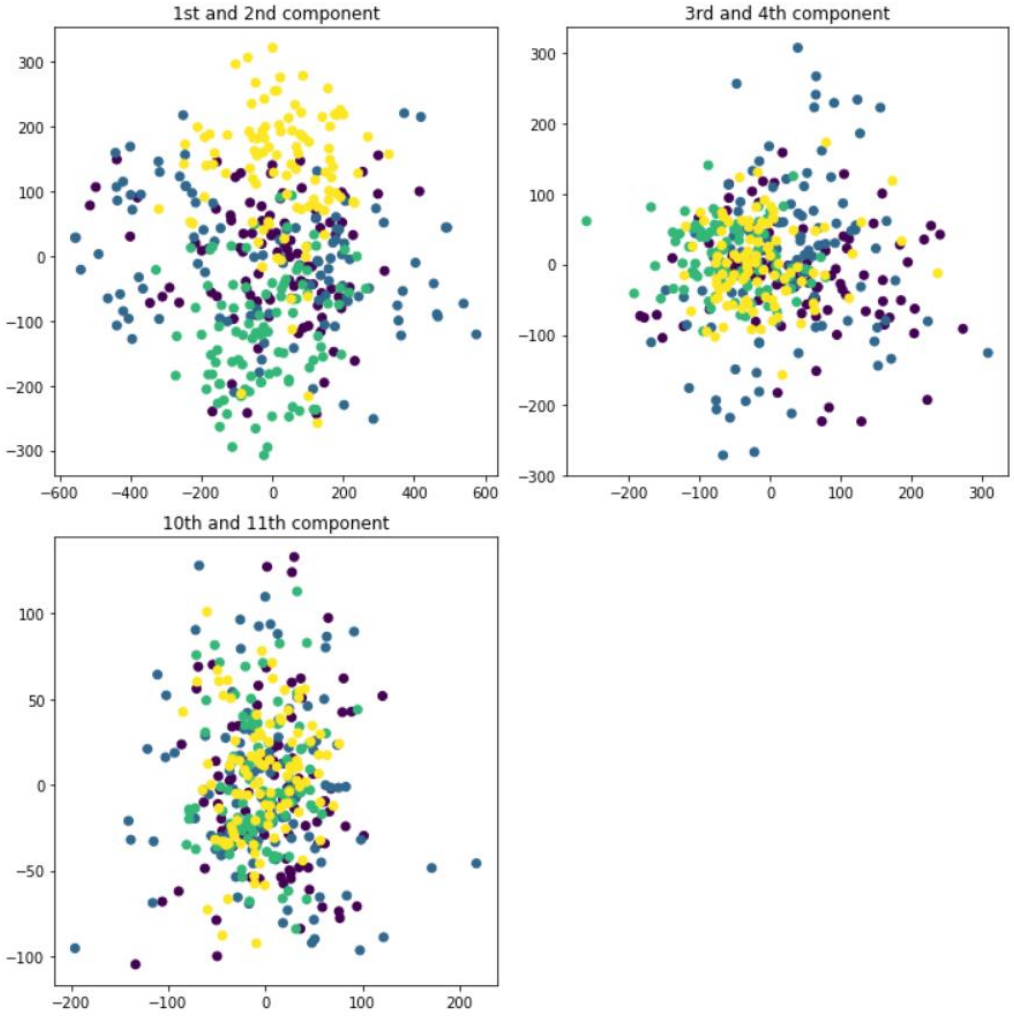

Figure 3. Plotted graphs for 1st and 2nd PC, 3rd and 4th PC and, 10th and 11th PC, respectively

In figure 3 we can see the first graph that has plotted values for the first two components. As the first two components represent projection of the data points along the direction with the largest variance, we've actually found the optimal stretch and rotation that allows us to best see clustering of classes in two dimensions. These two components have maintained the highest variance and thus saved the most information so that differentiation of the classes is the most visible.

3rd and 4th PC still retain some distinction between features of different classes, but less than the first two. In the tenth and eleventh PC this distinction is really hard to make.

NAÏVE BAYES CLASSIFIER

Naïve Bayes classifier is a statistical classification technique based on Bayes Theorem. It is a supervised learning algorithm. This classifier assumes that the effect of a particular feature in a class is independent of other features.

Matrix **Data** and labels array **Class** were split in training and testing set. The first set was used to train the classifier, while the second one was used to test how well is it doing. Testing set was comprised of 20% of the original dataset.

The classification was done on the data projected onto the first two principal components, and then for the data projected onto the third and fourth principal components.

What can be obtained from the data is that if we choose to do a classification on all the PC we get the best accuracy **(Around ~ 75%)**. But what is interesting is that using only first two PC we get accuracy worse by only **15%.** As for the 3rd and 4th PC, the results are worse than for the first two. This was expected given the fact that, even when we used scatter-plot, clustering was more profound for the 1st and 2nd PC.

VISUALIZING DECISION BOUNDARIES OF THE CLASSIFIER

A classifier partitions the feature space into the decision regions. Every decision region has features from the same class. Therefore, decision boundaries are there to separate different decision regions. Applying decision boundaries onto first two principal components we get a 2D plot with four different decision regions.

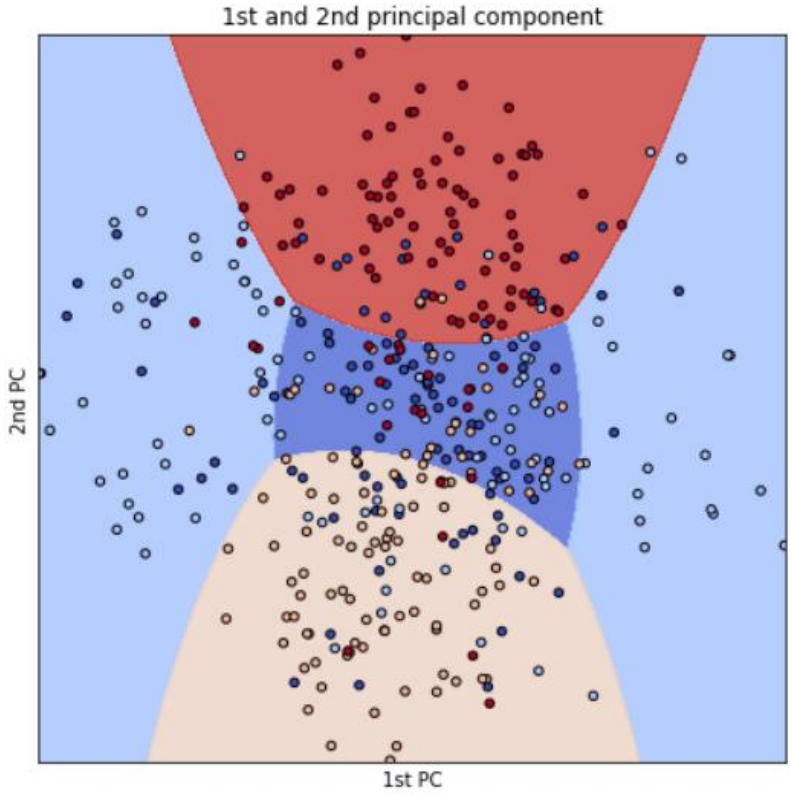

Figure 4. Plot of the decision boundaries of the classifier for the first two principal components

In figure 4 we can see that applying Naïve Bayes classifier onto the first two principal components cannot give us sharply partitioned regions in 2D. This is due to the fact that as we have seen in the scatter-plot and in the accuracy of the Naïve Bayes classifier for the first two components, overlapping of the values of different classes is too high. That is, not enough of the variance is expressed by only 1st and 2nd PC so that classes can be visibly separated.