Artificial Intelligence & Machine Learning
Homework 1 - Report

Antonio Tavera
243869

Abstract

In this homework is requested to apply Principal Component Analysis on a set of images. It is shown what happen if different principal components are chosen as basis for image representation and classification.
This report, just like the source code, is divided into different sections: it examines first the data preparation, then the representation of the re-projected images on the computed PCs and last it analyzes the results of the performed Naive Bayes classifier.

---

1. Data Preparation

In the first part of the homework is asked for loading data from the specified folder. This is done selecting the first choice from the displayed menu or automatically, if not done yet, for all the other choices. The called *load_dataset()* function sort the directories and visit them recursively. Every time a new folder is found its name is saved and used as a label for all the images that are loaded from it.
The images are then converted to a 154587-dimensional vector through the *ravel()* function and are appended to a *X_train* array; the corresponding *y_train* labels array is filled too.

2. Principal Component Visualization

2.1. PCA

I observed that re-projecting images with a low number or PCs after the analysis is done for all the dataset, it is possible to note shadow of people even if we are considering another class (Figure 2). This is due to a different proportion among the number of images for each classes, especially for the people one, and this is the reason why I decide to do the exercise in two ways, first computing the PCs class by class and then considering the whole dataset.

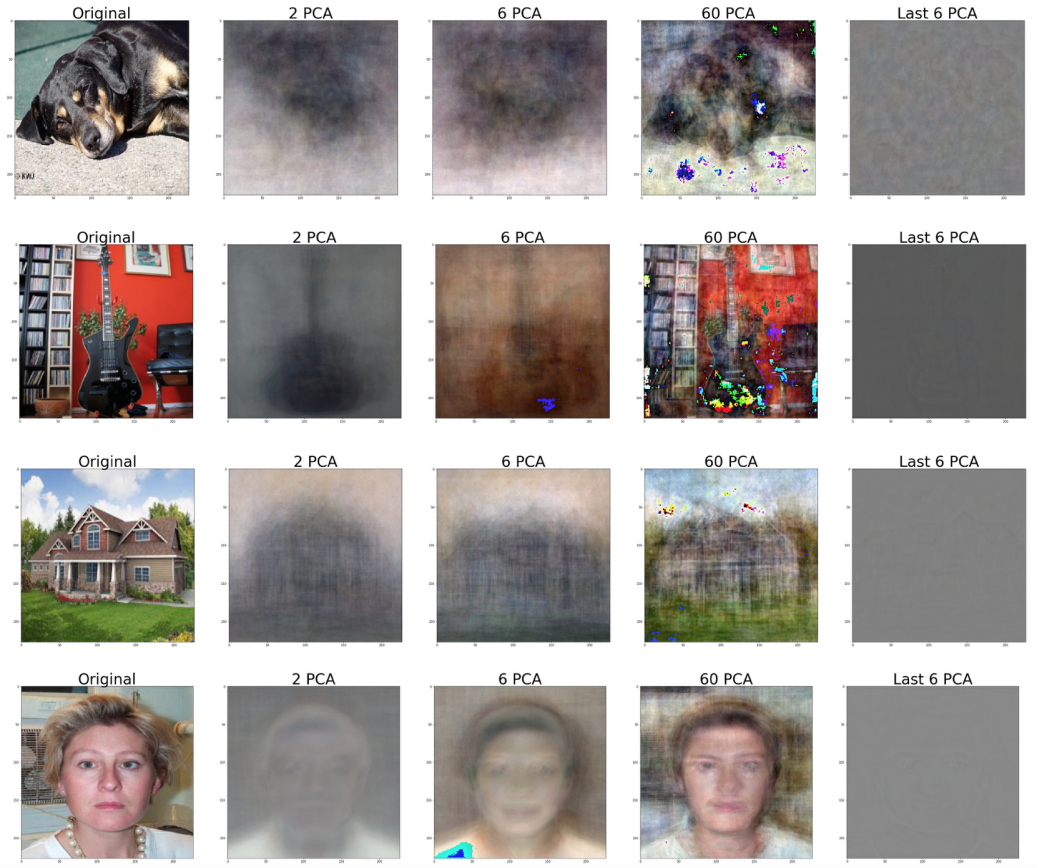

Figure 1: Outcomes using the class by class Principal Component Analysis. Starting from the top it is possible to see the four different analyzed classes: dogs, guitars, houses and people. The analysis is done on the first 2, the first 6, the first 60 and the last 6 Principal Components. For each class, the first image displayed is the original one.

2.1.1. PCA class by class

Selecting the second option from the menu it is possible to perform the Principal Component Analysis considering the classes one by one. For each class, the images are first normalized, subtracting the mean and dividing by the standard deviation (all these values are saved for the unnormalization process). Then, through the *fit()* function the PCs are extracted from the training set and saved internally in the *pca.component_* variable (if I need only the first principal components, sorted by variance, it is possible to do it simply setting the *n_components* variable within the PCA instantiation, otherwise are computed all the components and then selected only the right ones).
The next step is to transform the training set into the new space; an element-wise multiplication is done between the normalized data and the computed PCs. Because the shape of *PCA.components_* is *(n_components, n_features)* while the shape of data to transform is *(n_samples, n_features)*, it is needed to transpose *PCA.components_* to perform the dot product.
To obtain again images that can be visualized is performed an inverse transformation, a dot product between the transformed data and the computed principal components. Images are than unnormalized multiplying and adding respectively the saved standard deviation and mean.
Still considering the classes separately, a random image is chosen from all the performed Principal Component Analysis and compared to the original one, you can see it in Figure 1. Analyzing this figure is possible to see that with the first 2 or 6 principal components, due to the fact that are sorted by variance, we are able to see without much trouble, the silhouette of the belonging class. With an high number of components, like the first 60, it is possible to see, with a little noise, the corresponding original image. For the same reason, if we select the last 6 components it is normal to obtain a grey image and to not be able to recognize what class it is.

2.1.2. PCA whole dataset

Selecting the third option from the menu it is possible to repeat the experiment considering the whole dataset. The passages are the same as those described in the section above. What is expected, due to the fact that exists a different proportion in the number of each class, is that there is a prevalence of people features in the computed principal components; is possible to see this in Figure 2. In fact, if we analyze the images re-projected using as basis the first two or six components, we obtain images with the

silhouette of a person, but this could mislead the user because what we are really examining is a dog class. Results change if we consider the first 60 PCs, it is possible to see, with difficulty, the corresponding class but not the original image, as we are able to do if we consider the class one by one as done in Figure 1.

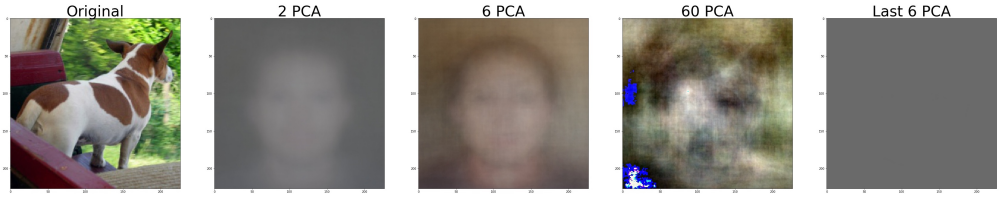

Figure 2: Outcomes using the whole dataset to perform the Principal Component Analysis. The selected class is a dog. The analysis is done on the first 2, 6, 60 and the last 6 Principal Components. The first image displayed is the original one.

2.2. Scatter Plot

A scatter plot is a two dimensional data visualization that uses dots to represent the values obtained for two different variables.
Selecting the fourth option from the menu, it is possible to perform a scatter plot on the transformed training set, re-projected with only the the firsts two, the third and the fourth, the tenth and eleventh Principal Components. Results can be seen in Figure 3.

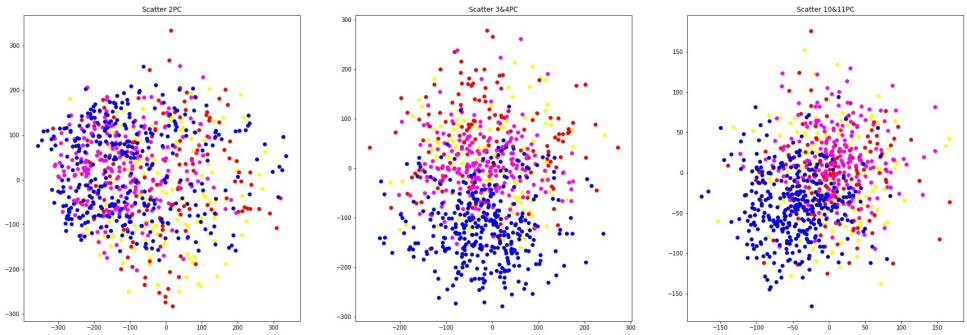

Figure 3: Scatter plot of the transformed training set. The first refers (x-axis, y-axis respectively) to the first two PCs, the second refers to the third and fourth PCs while the last refers to the tenth and eleventh PCs. In yellow are represented the transformed set of the dog class, in red the guitar, in magenta the house and in blue the people one.

What we can see from the results is that the points that refers to the transformed images using as basis the firsts two principal components are far away one from the other and well stretched along the direction with the largest variance; this is due to the fact that we are analyzing the first PCs. Indeed, if we move to the next two illustrations we can see that points start to squeeze towards the origin. This demonstrate that the first PCs are the one with the higher variance, while moving towards the last we find principal components with a lower and lower variance that are essentially noise.

3. Classification

The last choice offered by the menu is to perform a classification of the dataset using a Naive Bayes Classifier with Gaussian class-conditional distribution. First the dataset is split between training set and test set. To do this the *split_dataset()* function is called; it first shuffles data, then takes the last 20% as test set while keeping the remaining as training set. At the end of the process data are sorted again. Then a Gaussian Naive Bayes model is created and trained with the training data.
After all, the trained model is tested using the test set and accuracy is displayed to the user. This test is then repeated on the firsts two and the third and fourth PCs re-projected images. These are the accuracy obtained from the different tests:

- Accuracy Original Dataset = 80,6%
- Accuracy Dataset re-projected using first 2 PCs = 48,8%
- Accuracy Dataset re-projected using first 3 and 4 PCs = 46,5%

Analyzing these results it is possible to see that the classifier has good ability to distinguish class if considering the original dataset, while the probability to choose the right class became fifty-fifty if the classifier considers the dataset re-projected with the first two PCs. The accuracy continue decreasing as we choose PCs with less variance.
Probably, if we consider as dataset the re-projected images after class by class PCA, the accuracy increase.

4. Conclusions

The question is how many principal components are we going to choose for our feature subspace, to preserve data without much distortion. To answer this question a useful measure we can do is the so-called explained variance, that tells us how much information (variance) can be attributed to each of the principal components.
If we plot the explained variance ratio as a function of the number of components (Figure 4) it is possible to see that the firsts 100 components contains approximately 80% of the variance, while we need around 800 components to describe close the 100% of the variance.

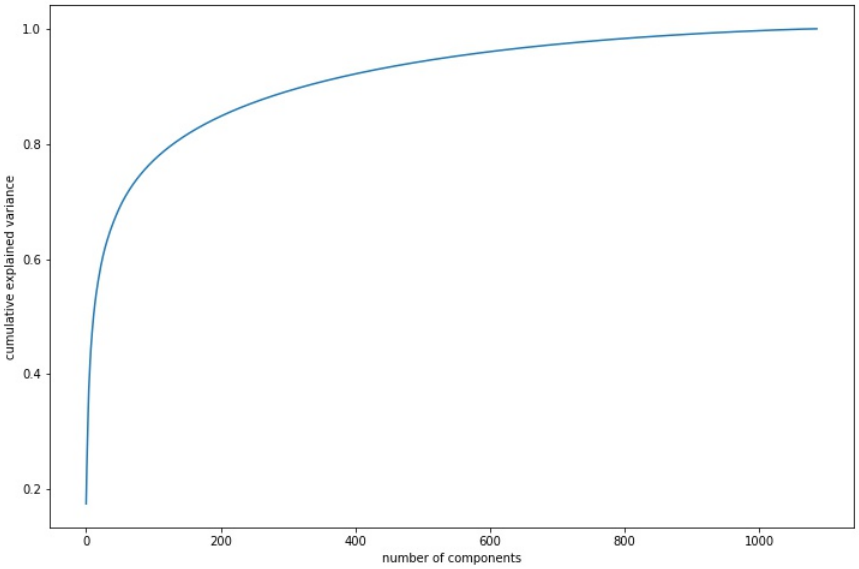

Figure 4: This curve quantifies how much of the total variance is contained within the first N components.