Machine Learning and Artificial Intelligence
Homework 1

Debora Caldarola s263626

Prof. Barbara Caputo, A.Y. 2018/2019

1 Introduction

Principal Components Analysis (PCA) is an unsupervised method that aims to find the smallest subspace such that as much information about the original data as possible is preserved.
Given a fixed number of points, PCA has as its goal to find an orthogonal set of linear basis vectors such that the average reconstruction error is minimized. In the found optimal low-dimensional encoding of the data the mean squared distance between the data point and their projection is minimized and the variance of projected data maximized (the higher the value of the variance, the more information is stored).
Each Principal Component is orthogonal to the previous one and points in the direction of the largest variance.
PCA finds its main applications in data visualization, data compression and noise reduction.
The purpose of this experience is to apply such method in order to show what happens if different principal components (PC) are chosen as basis for images representation and classification. Moreover, a classifier has to be picked and applied to classify the images under different PC re-projections.
In the following paper, the results will be expressed and analyzed.

1.1 Homework Sub-tasks

1. Setup the programming environment; download and load the provided subset of PACS dataset.
2. Show what happens when an image is re-projected with only the first 60 PC, the first 6 PC, the first 2 PC and the last 6 PC.
3. Using scatter-plot, visualize the dataset projected on the first 2 PC, the 3th and the 4th ones, the 10th and the 11th ones.
4. Classify the dataset using a Naïve Bayes Classifier in the following cases: unmodified images, images projected into the first 2 PC and on the 3th and 4th ones.
5. Visualize decision boundaries of the classier.

2 Data Preparation

The provided datased consists of a subset of PACS dataset made of four visual object categories: `{'dog', 'guitar', 'house', 'person'}`. The total amount of images is 1087 with a 3x227x227 sample size.

2.1 Programming Environment Setup

The exercise will be solved using Python 3.7 programming language. Moreover, the following libraries and packages will be referenced:

- `Image from PIL`, the necessary module to handle images;
- `numpy`, the fundamental package for scientific computing with Python;
- `glob`, the module that finds all the pathnames matching a specified pattern according to the rules used by the Unix shell;
- `matplotlib.pyplot`: `matplotlib` is a Python 2D plotting library; `pyplot` provides a MATLAB-like interface;
- `scikit-learn`, a free software machine learning library for Python. Specific imported packages: `sklearn.decomposition`, `sklearn.model_selection`, `sklearn.naive_bayes`, `sklearn.metrics`.

2.2 Data Storage

The images are loaded and stored in numpy 3D-arrays, which are immediately converted into contiguous flattened arrays in order to obtain a vectorial representation. The dimension of the resulting vector $\mathbf{x}$ is equal to 154587, which comes from the sample size (3x227x227). By operating such a transformation, we lose information about the resolution and the channels of the image: we can do so aware of that fact that PCA and training will not need such details.
All the computed vectors are then saved in a $Nx154587$ matrix $\mathbf{X}$, where N is equal to 1087 and stands for the total amount of read samples. The rows of $\mathbf{X}$ will be referred as *examples*, the columns as *features*.
An N-dimensional vector $\mathbf{y}$ is built to hold the ordinal labels of the images, according to the original folder they belonged to (PACS_homework/dog, PACS_homework/guitar, PACS_homework/house, PACS_homework/person).

3 Principal Component Visualization

In order to apply the PCA algorithm, tha matrix $\mathbf{X}$ is standardized (each feature is made zero mean and unit variance) so that all the variables are scaled and treated as equally important (PCA will tend to treat variables with greater variance as more relevant).

3.1 Application of PCA

In order to visualize the effect of the PCA algorithm, a random image is taken from the dataset (Figure 3.1). Our aim is to show what happens when that image is re-projected considering only the first two, or six, or sixty principal components, or the last six ones. We expect to obtain an approximate and inaccurate result, because of the few principal components we are taking into account.

The main executed steps are the following:

1. Fit PCA on the training set, which is the standardized matrix $\mathbf{X}$ in our case.
2. Extract the required principal components from the fitted model.
3. Project the 154587-dimensional data onto those components.
4. Re-project the images on the original space.
5. Plot the reduced images.

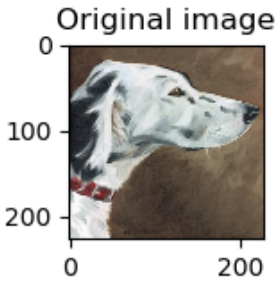
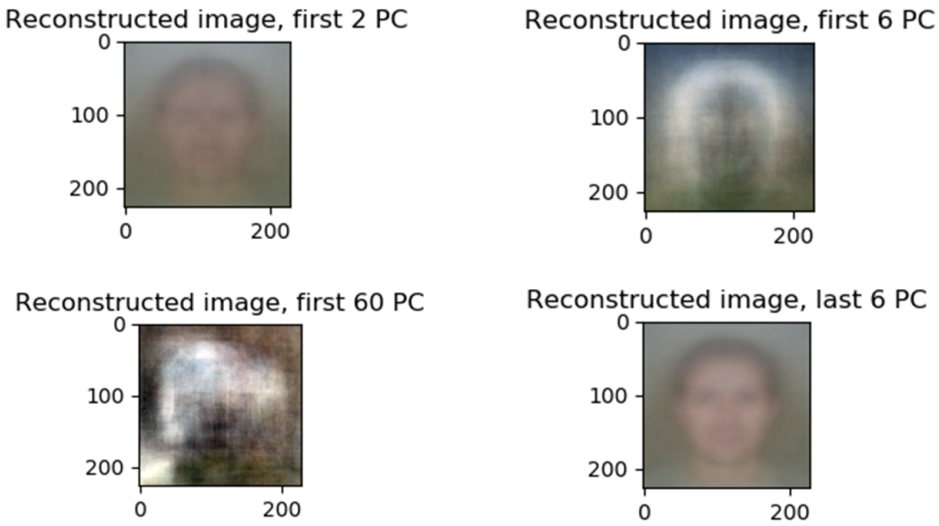


Figure 1: Original image from the dataset

The result of the algorithm is showed in the following images:



As we can see, the more components are involved, the more precise the results get, because there is access to a wider set of information. At the same time, in the PCA algorithm, once the new variables are computed, the one with the largest variance is projected on the first axes. That means the first principal components will contain more information than the last ones, which is why the image reconstructed using only the last six components is far less precise than the one computed having as a reference the first six PC (or even the first two ones). In fact, if we take a look at the variance contained in the first and last six PC, there will a substantial difference: as for the first one, the computed value is 0.2, larger than what is obtained for the second one, $1.6x10^{-5}$.
In addition, in the dataset there are more images picturing a person than the ones representing guitars, houses or dogs. That division is the reason why we recognize the shape of a human face in the samples reconstructed with a low number of principal components: the largest variance is found where images belonging to the `'person'` class lay. In fact, the shape of the dog (Figure 3.1) begins to appear when the first 60 PC are used.

3.2 Visualization of Dataset Projection

Using scatter plots, it is now possibile to see the projection of the dataset. The exercise is repeated using the first 2 PC, the third and the fourth ones, and lastly the tenth and eleventh ones.
The scatter plots representing the projected images are shown in Figure 2, Figure 3 and Figure 4: different colours stand for different classes.

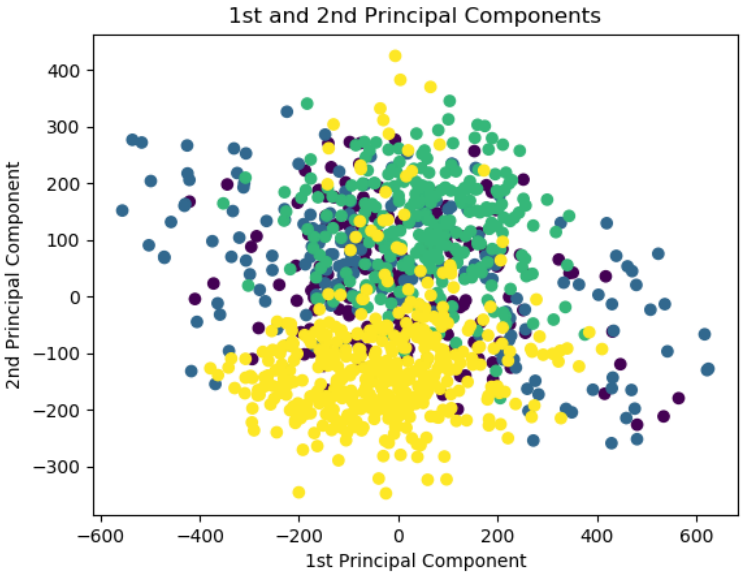


Figure 2: First and second PC

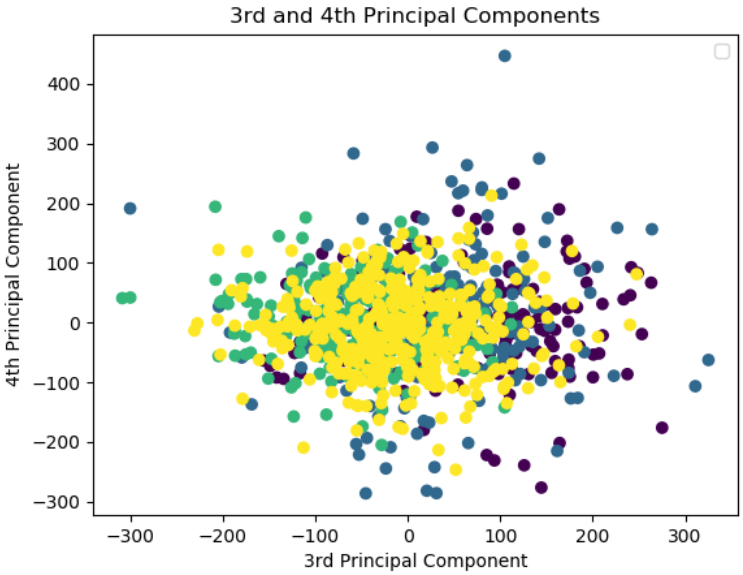


Figure 3: Third and fourth PC

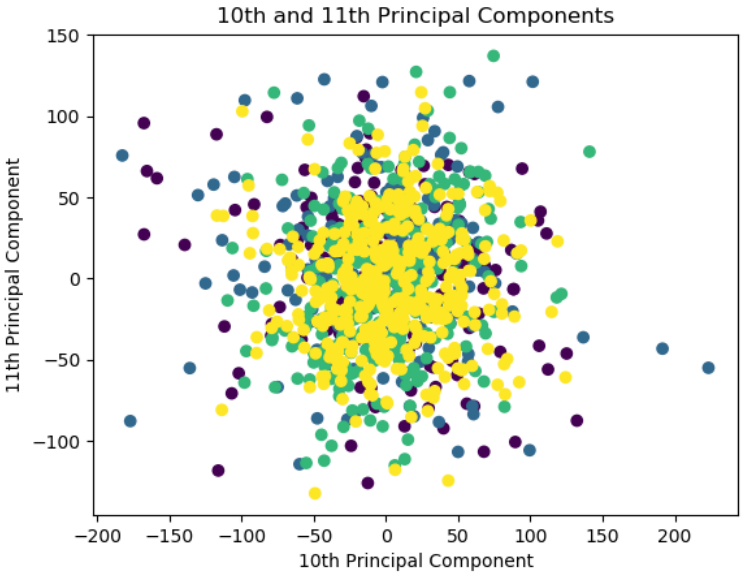


Figure 4: Tenth and eleventh PC

As shown in the scatter plots, the algorithm is able to classify the input samples more clearly when the first two components are used: in fact, we can see how in that case at least two out of four classes are projected distinctly. The worst result is obtained using the 10th and 11th PC, where the boundaries among the classes cannot be distinguished.
In order to explain that experience we can tap into the theorical background: it is known that the most important variables in the original feature space are the ones with the largest variance and those are the ones that contribute most to the most relevant PC. That implies the first principal components contain more information about the dataset than the others and so the farer we get from the first component the less precise result will be computed from the PCA algorithm.

3.3 Variance Analysis

The main use of PCA is to reduce the size of the feature space while retaining as much of the information as possible. A way too see how much information we retain is to look at the cumulative explained variance ratio of the principal components as a function of the number of components.
As shown in Figure 5 and Figure 6, the first 5 components contain approximately 50% of the variance, while almost 1000 components are needed to describe the dataset close to 100% of the variance.

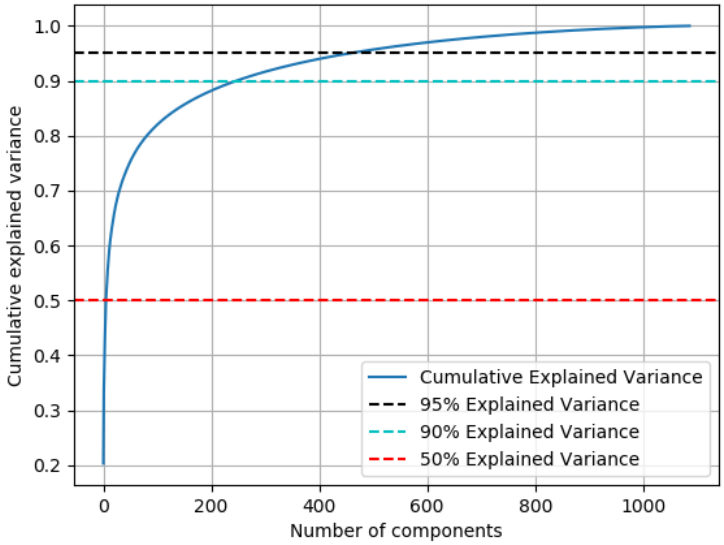


Figure 5: Cumulative Explained Variance Ratio

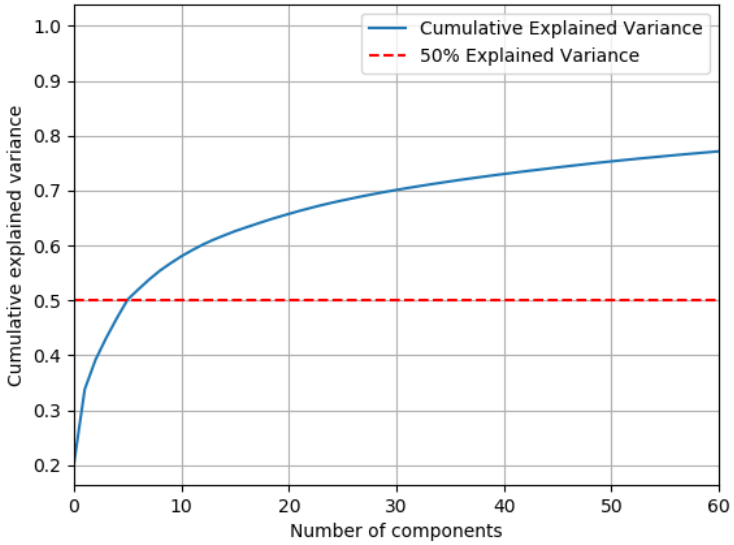


Figure 6: Cumulative Explained Variance Ratio - Zoom on the first PC

4 Classification

4.1 Naive Bayes Classifier

Naive Bayes classifiers are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. As a consequence of that hypotesis, each distribution can be independently estimated as a one dimensional distribution and problems arising from the curse of dimensionality may be alleviated. Given k classes, d samples, the class variable $\boldsymbol{y}$, the dependant features vector $\boldsymbol{x}$=$\{x_i, \ldots, x_d\}$, according to the formulation of the Naive Bayes classifier, the predicted label $\hat{y}$ is:

$$\hat{y} = \underset{y \in \{1,\ldots,k\}}{\operatorname{argmax}} P(y|\mathbf{x}_1, ..., \mathbf{x}_d) = \underset{y \in \{1,\ldots,k\}}{\operatorname{argmax}} P(y) \prod_{i=1}^{d} P(\mathbf{x}_i|y),$$

where $P(y)$ represents the relative frequence of class y in the training set.
The Maximum A Posteriori (MAP) can be used to estimate $P(y)$ and $P(x_i|y)$. The main difference among the Naive Bayes classifiers is the assumptions they make regarding the distribution of $P(x_i|y)$: we consider the likelihood of the features to be a **Gaussian**. Moreover, we assume the distribution of labels to be uniform.
So considering that in our case k=4, the previous formulation can be rewritten as:

$$\hat{y} = \underset{y \in \{1,2,3,4\}}{\operatorname{argmax}} \frac{1}{4} \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right),$$

where σ_y and μ_y are respectively the variance and the mean of y and are estimated using maximum likelihood.

4.2 Dataset Classification

The input samples and the given labels, stored respectively in the matrix $\mathbf{X}$ and in the vector $\mathbf{y}$, are split into training and testing sets. In particular, 20% of the dataset is included in the test split.
The dataset is classified using a Naive Bayes classifier, which is trained and tested with the Gaussian class-conditional distribution so that the vector of predicted labels $\hat{y}$ is computed, according to the formulation discussed in Section 4.1.
Three different situations are introduced: classification of unmodified images, images projected onto the first two components and on the third and the fourth ones.
In order to determine the quality of the prediction, the different models are compared basing on the obtained accuracy:

- Accuracy on **unmodified images**: 74.3%;
- Accuracy on images projected onto the **first two components**: 66.1%;
- Accuracy on images projected on the **third and fourth components**: 44.9%.

Reducing the number of features the model is trained on, the resulting accuracy decreases. It is reasonable to obtain the lowest value when the third and fourht PC are considered, because of the percentage of variance and information they detain.

4.3 Decision Boundaries

The region of a problem space in which the output label of a classifier is ambiguous takes the name of *decision boundary*.
In order to plot the decision boundaries of the Naive Bayes classifier, the decision surfaces are computed

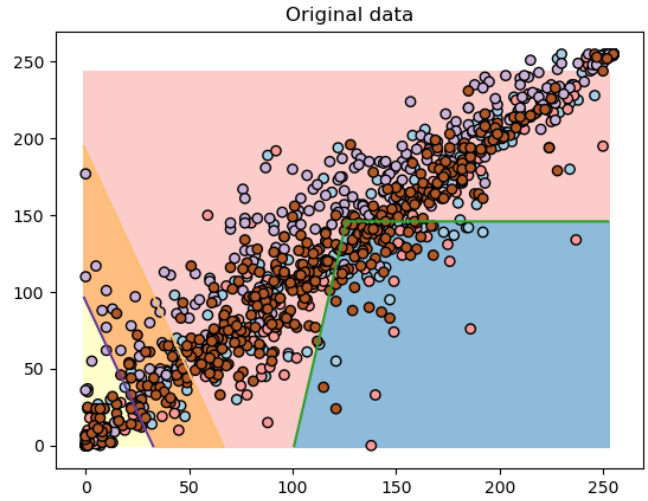


Figure 7: Decision boundaries on unmodified images

and plotted using the `matplotlib` package.
In Figure 7, 8, 9, the captured plots show both the decision boundaries and surfaces for the four labels that classify the dataset. The three considered cases are the classifier trained for unmodified images, images projected onto the first two components and on the 3rd and 4th ones.

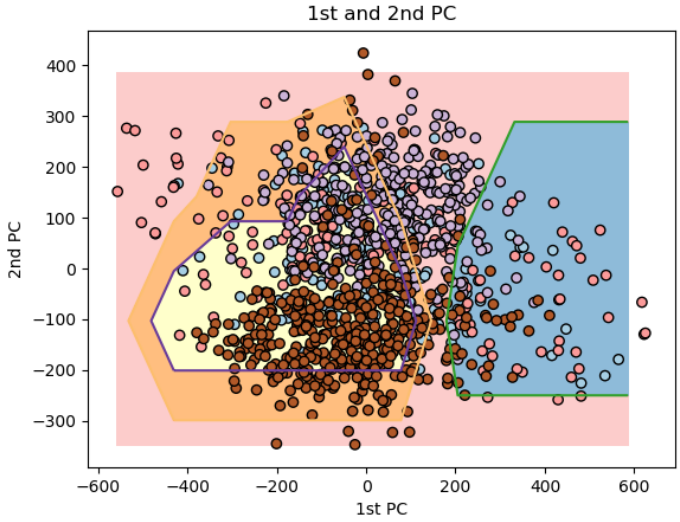


Figure 8: Decision boundaries on images projected onto the first 2 PC

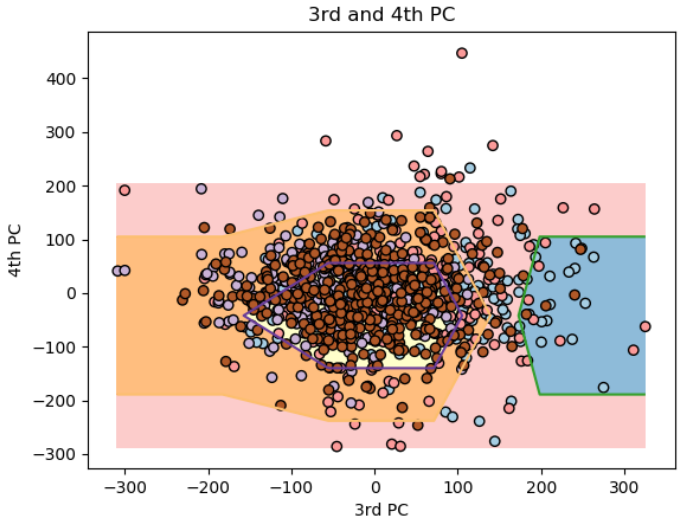


Figure 9: Decision boundaries on images projected on the 3rd and 4th PC

5 Conclusions

The report showed a possible application of PCA (Principal Component Analysis): image reduction. In particular, we saw how the algorithm relates on the variance to decide which variables can be considered more important than others and what happens to the visualized images when distinct components are chosen.
The second main task was to classify the dataset using the Naive Bayes classifier. The classifier was trained on both unmodified input samples and projected images, analyzing the main differences depending on the accuracy. The decision boundaries were showed too.
In conclusion, PCA is a very powerful algorithm to be used whenever dimensionality reduction can speed up the algorithm, without loosing too much information. In order to success in that, the cumulative explained variance ratio should be considered.