Principal Component Analysis

Giuseppe L'Erario

Introduction

The *Principal Component Analysis* (PCA) is an unsupervised linear transformation technique used to reduce dimensionality of the data.

In this report the aim is to preprocess a set of images with PCA and then to train and test a *Naïve Bayes Classifier*.

1 PCA

With PCA one can find the direction of maximum variance in high-dimensional data and then project these data onto a subspace with less dimension of the original data.

The principal components are the directions of the maximum variance, and they *must* be orthogonal to each other (in other words, they must be uncorrelated).

The principal components are very sensitive to the scale of the data. It is useful indeed to *normalize* the data (rescaling the features to a range of [0,1]) in order to analyze features of different scale, and *standardize* them (center the mean to zero and the standard deviation to 1).

The pixels of every image are converted to matrix of values. This matrix is then reshaped into a vector.

The commands used to read and preprocess data are:

```
#creation of images file list
for num in n_img:
    img_list += glob.glob(img_folder+str(num)+'_*')
#conversion of the images
for filename in img_list:
    im = np.asarray(Image.open(filename).convert("RGB"))
    im_raveled = np.ravel(im)
    img_data_raveled.append(im_raveled)
#...and reshaping in 149152-dim vector
X = np.array(img_data_raveled).reshape((len(img_list)), -1)
#normalize and scaling
X_std = preprocessing.scale(X)
```

PCA algorithm:

- builds the covariance matrix;
- decomposes the covariance matrix in its eigenvectors and eigenvalues;

- return a matrix wich has as columns the eigenvectors associated to a eigenvalues (ordered from bigger to smaller).

The bigger is the eigenvalue, the more is the variance associated to a feature.

The aim is to train a classifier using less features, without loosing too much information. The results are:

- less computation resources usage
- less importance to "lightweight" features

PCA algorithm is performed easily with the *sklearn* library:

```
pca = PCA(value)
X_PCA = pca.fit_transform(X_std)
```

where *value* is the number of the principal components.

Plots

The results are shown by the plots. There are also the decision boundaries found by the classifier (explained later).

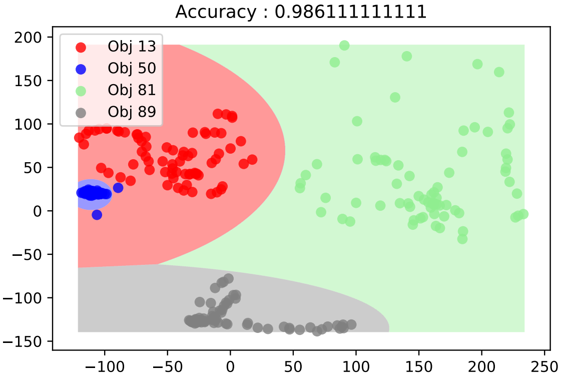

Figure 1: First ancd second Principal Components

It is possible to see in fig.1, as expected, a good separation: the first two principal components bring with them the biggest part of the information.

In fig.2 and fig.3 the classes are not well separated. These components are not so important to characterize the objects. In other words these features are similar among objects.

We can see the "quantity" of information of every component described by the variance, in fig.4. The first two components got about 50% of the information (*cumulative explaindes variance*).

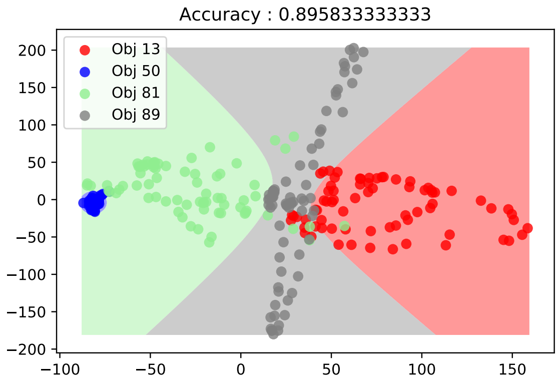

Figure 2: Third and fourth PCs

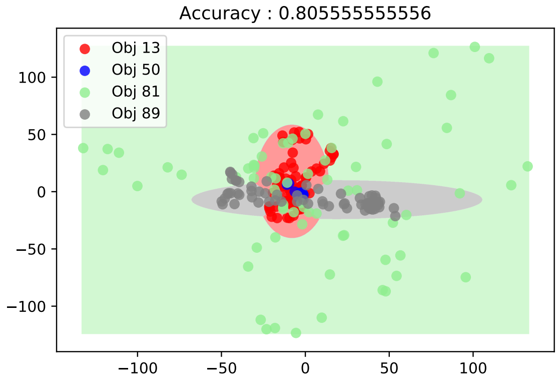

Figure 3: Tenth and eleventh PCs

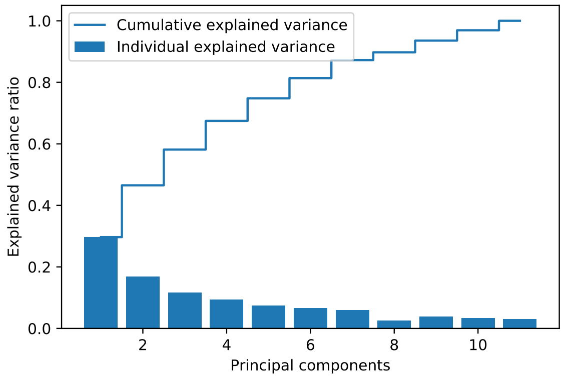

Figure 4: Variance of each Principal Component

2 Naïve Bayes Classifier

The *Naïve Bayes Classifier* are a class of probabilistic classifier based on Naïve Bayes rule.

With this classifier is possible to determine the probability of an element to belong to a class of objects.

We can use a classifier based on Gaussian distribution from *sklearn* library:

```
clf = GaussianNB()
clf.fit(X_train, y_train)
```

where *X_train* and *y_train* are obtained through:

```
X_train, X_test, y_train, y_test =
        train_test_split(X_PCA, y, test_size=0.5)
```

With this command the dataset is divided in 2 separate datasets, one used to train the classifier and the second one used to test the accuracy of the trained classifier.

It can be seen in fig.1 that by using the first two principal components the classifier can achieve more than 90% of accuracy. This result underlines what said before: the first two principal components have enough information to classify an object.

On the other side, using 3rd-4th and 10th-11th principal components the accuracy is considerably decreased: these features are useless for classification tasks.