Homework 1
class in “Machine Learning”, Fall 2016/17

Marco Favorito
Master of Science in Engineering in Computer Science
Department of Computer, Control, and Management Engineering
University of Rome “La Sapienza’
favorito.1609890@studenti.uniroma1.it

December 3, 2016

The numbers in the following section are the answers to relative questions at same-named sections on document `pca.pdf` in course's Dropbox shared folder.

1 Principal Component Visualization

In the following, I will refer to $\boldsymbol{X}$ as the dataset matrix $(4 \times 72 = 288) \times (128 \cdot 128 \cdot 3 = 49152)$ about only first 4 object (in each view, # 72 images per object).

2 Classification

$\boldsymbol{X}_s$ is $\boldsymbol{X}$ standardized, and $\boldsymbol{X}_t$ is the PCA matrix with dimensions (#examples = 288) $\times$ (n), where n is the number of the (best) principal component to be considered. $\boldsymbol{X}_t$ can be obtained by:

```
X_t = PCA(n).fit transform (X)
```

In the following I reported the plot of the 1st and 2nd principal component, the 3rd and 4th one, the 10th and 11th one.

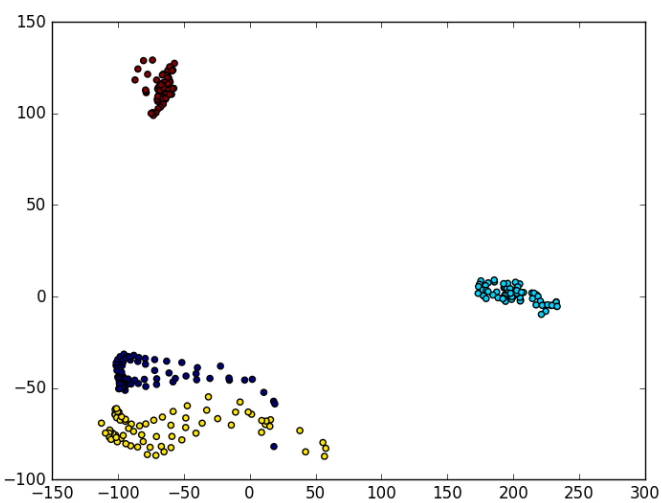

Figure 1: Plot of the 1st and the 2nd principal component

We can notice that in Figure 1 the plotted samples are distributed in some regular pattern, a little less in Figure 2 and definitely not in Figure 3. From a

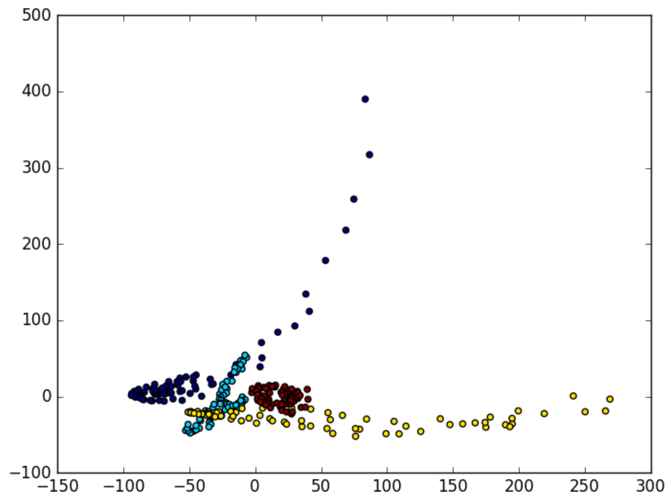

Figure 2: Plot of the 3rd and the 4th principal component

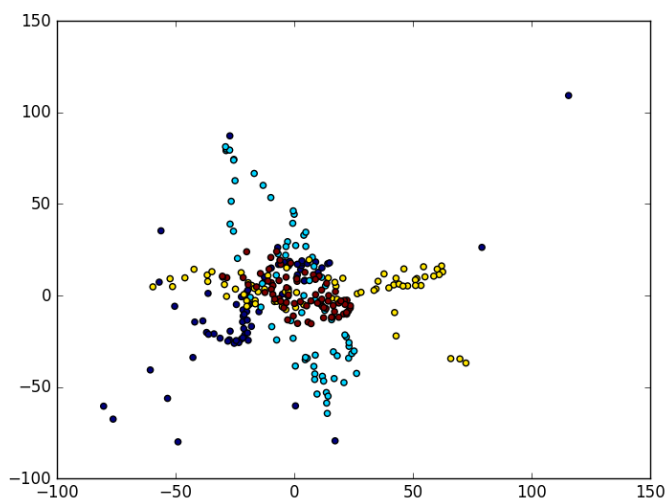

Figure 3: Plot of the 10th and the 11th principal component

theoretical perspective, we can explain it as follows: PCA finds an alternative basis vector which “explain better” the data, i.e. redirect basis vectors such that the variance of data is maximized. Indeed we find much less dispersion of data in first two components plot than others. It is easy to see that data are plotted into well-distinguishable groups only with the two first principal component, and how things get worse in the latter two.
In order to decide how much principal components to include, just look at their eigenvalue, which scaled by the sum of all eigenvalues, represent the precentage of importance in representing data.

3 Classification

Results are quite different. My program yields this output:

```
Using test size = 0.33:
1st and 2nd pcs
Success percentage: 0.96875 on 96 test images
3rd and 4th pcs
Success percentage: 0.75 on 96 test images
```

As we can see, performances seems better in the former case, with almost 97% of success in classifying correctly test items, instead of 75% in the latter one. This is what we expected, since first two components represent better the data, as we said before. In the following figure I will show decision boundaries of both classifiers:

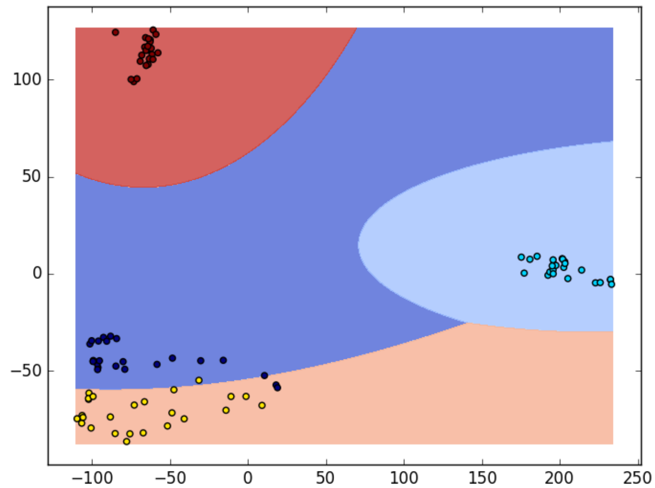

Figure 4: Decision boundaries for the classifier trained with 1st and 2nd components

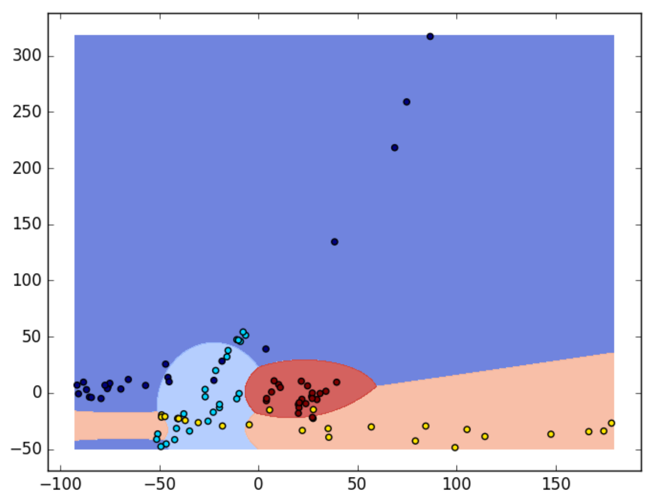

Figure 5: Decision boundaries for the classifier trained with 3rd and 4th components