Principal Component Analysis

Edoardo Fantolino

Abstract

This document contains my idea about the homework of Computational Linear Algebra regarding the topic of Principal Component Analysis (PCA). This document is addressed to Professor Stefano Berrone and Francesco Della Santa.

Keywords: large scale problems, linear algebra, PCA.

Introduction

The main topic of this homework is the so called RedShift phenomenon. It means that when we see an object that is moving away from us, there will be an increase in the true wavelength of the object. Basically, it is a Doppler effect but for light. I drawn a picture to better explain the phenomenon. In Fig.1 you can see a star that is moving from right to left. James will see the light of the star shifted toward the blue while Robert will see the same star but the light will be shifted toward the red because of the increase in the wavelength or a decrease in frequency.

Thanks to this phenomenon we can understand if an object is moving toward us or away from us. This is very important, because the scientists and astrophysicists discovered that most of the galaxies are shifted toward the red, and that means that the universe is in an expansion phase. Now, they have to discover if this behaviour will continue or if after this expansion phase there will be a contraction that will lead to a Big Crunch.

This homework will make us understand what are the most important information that allow us to estimate the Redshift of a given galaxy.

In this assignment, we explore the COMBO17 dataset (Classifying Objects by Medium-Band Observations). In this database we find information about some galaxies of our universe.

The homework is divided into 4 main parts:

1. importation, cleaning, division (train, test) of the dataset.
2. application of the PCA for computing the Principal Components (PCs) and analysis of the results.
3. graphical representation of the PCs.
4. prediction over the test set using k-Nearest Neighbours.

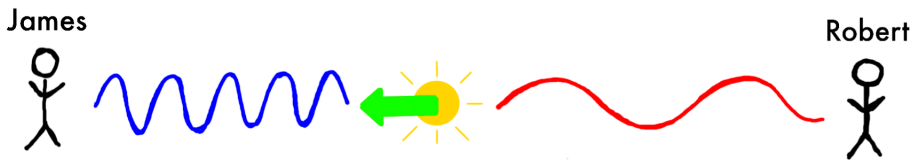


FIGURE 1: James will experience the BlueShift while Robert will experience the RedShift.

1 Managing COMBO17

In the first requirement we need to prepare the dataset. I used the pandas package to locally store the COMBO17.csv file. After a brief check of the dataset we can observe that there are some missing and invalid information. We need to clear our data because otherwise we will find some wrong results.

First, we can observe that there are some missing values (NaN). In this case, the approach was to erase the rows because they represented approximately only the 0.01% of our dataset. We delete few information that probably were in some way misleading. An other issue arise when we analyse the column called 'ApDRmag'. A negative value of this parameter is not physically meaningful. The number of negative 'ApDRmag' is 2266. This value represents a huge chunk of our dataset. Deleting all this rows is out of discussion. So I decided to set the negative value to zero because this value correspond to a single point source that is the minimum allowed.

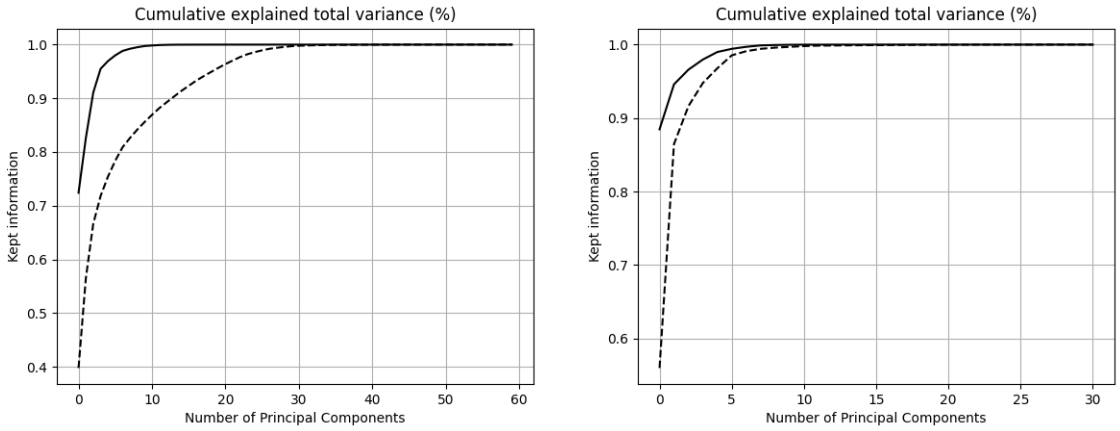
In the next step, I split the data into two main groups, train and test. For doing that, I imported a function from 'sklearn.model_selection' that is called 'train_test_split' that allow us to split our DataFrame in two parts. A little calculation allow us two precisely divide the sets as required. The first set has 2500 elements (train set) and all the other rows are in the second set (test set). Then, I saved the data in a directory. I made some output to get a quick feedback just to understand if everything worked properly.

2 Principal Components Analysis

In this section we began the second requirement. It is asked to use the PCA class of scikit-learn to compute the PCs of COMBO17. As required I removed the columns related to the redshift (Mcz, e.Mcz, MCzml, chi2red). Moreover, I deleted also another column, the 'Nr' column. I deleted it because it is the ID number of the galaxy and this column don't represent a useful or natural characteristic of the galaxies.

After a while, I started thinking about the column about the errors. They do not represent a direct characteristic of the galaxy but maybe they hide some useful information. So, here we have to possibilities:

- keep the error columns
- delete all the error columns



(A) With error columns (B) Without error columns

FIGURE 2: The dashed line represent the cumulative sum of the explained variance ratio with the normalized data while the full line is the cumulative sum of the explained variance ratio for the not normalized data.

I developed two parallel cases: one that kept the errors and the other was free of all the columns related with the errors. As you will see, at the end I compere the two prevision in order to establish which one of the two strategies was the best.

Another choice that we can make is to normalize or not our data (to use or not the StandarScaler). To understand which approach is the best, it can be useful to plot a graph. In this graph we will see the cumulative sum of the explained variance ration that represent how much information we keep if we use a certain number of principal components. In Fig.2(A) you can see the result for the case with error columns. We observe that if we use the not normalized data we can keep more information with fewer PCs. For example with the first principal component we explain 72.40% of the variance. With the second PC we add a 10.2% and with the third PC we add another 8.4%. If we add all this three PCs together we obtain a cumulative sum higher than 90%. Instead, if we normalize our data, with the first three PCs we arrive only to a cumulative sum of 66%.

If we observe Fig.2(B), we can infer that we reach higher values of kept information with fewer PCs. I explain this fact because we have reduced the number of features ($60 \rightarrow 31$), so it is easier to represent the starting information with fewer features.

In Tab.1 are represented the different combination of strategies with the relative number

	keep 95% of information	
	with error columns	**without error columns**
not normalized	4 PCs	3 PCs
normalized	20 PCs	5 PCs

TABLE 1: Table that shows how many PCs we need if we want to reach the threshold of 95% with different combination of strategies.

of PCs to use in order to keep 95% of the original information. For the results obtained and shown in Tab.1, I decided to not normalize the data, but as said before, I continue to keep two different script in order to evaluate if the errors hide some not obvious information.

Then I represented in the bar graph the PCs of the case without error columns and I tried to interpret them in order to find a name.

For many different reason I put the images in the last page of the document (Fig.6, Fig.7, Fig.8). We can easily spot that there is some correlation between some of these features. For example in all the three bar graphs we see that UjMAG, BjMAG, VjMAG,...,VbMAG are linked toghether. They follow the same path with respect to the PC that we consider.

I decided to call the first PC "Galaxy luminosity" because in the assignment there is written that this feature give the absolute magnitude in different bands and so they are linked with the overall luminosity of the galaxies. I called the second PC "Galaxy kinematic" because it remembered me the variables of the kinetic formula. The most relevant features are the Rmag and mumax. We now that there is a relationship between mumax and ApDRmag, and this relationship is an indicator of the galaxy size. So I used this name because we have the Rmag that is strictly linked with the speed of the galaxy and then we have a measure of the dimension of the galaxy (mass, size). For the third PC I used a a combination of the names of the most relevant features. The name is "VB-iS280U". Vmag and Bmag are directly correlated, and they are positive (VB). Then we have S280MAG that is the most significant feature that is directly correlated with UMAG (so S280U). I put an 'i' between them because this two groups are indirectly correlated (while one is positive, the other one is negative).

3 PCA graphical representation

In this section I try to explain and interpret the resulta of the PCA. The majority of the graph that I will show are scatter plot in 2D. They will represent the dataset transfrormed with a PCA that keep 95% of the original information. Each point of the scatter plot has a particular color that represent the value of the Mcz. You can understand the relationship between Mcz value and colors thanks to the graph in Fig.3. We will call the transformed data Q. I think that we can get an idea of the "goodness" of the first Q dimension thanks to the plot shown in Fig.4. Basically, I plotted the first dimension of Q in one dimension, but the points were one above the others. So, in order to better visualize the distribution of the points, I added as second dimension a fictitious noise. Moreover, to make it easier to

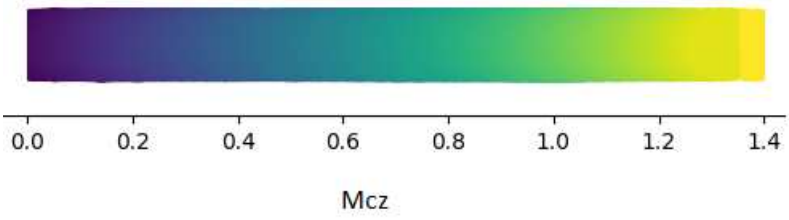


FIGURE 3: The relationship between colors and Mcz values

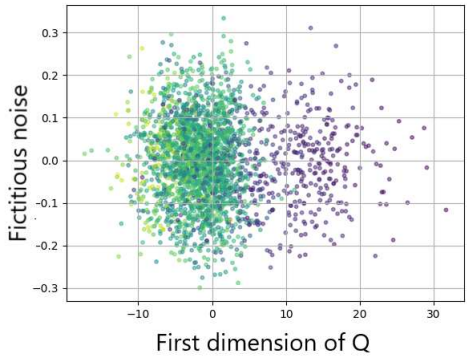
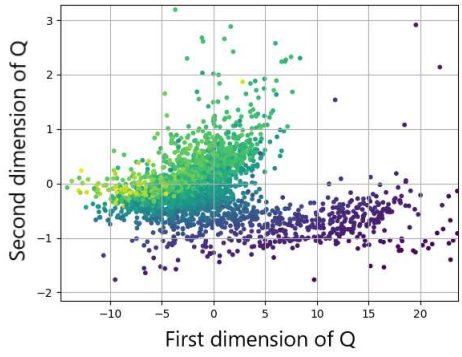
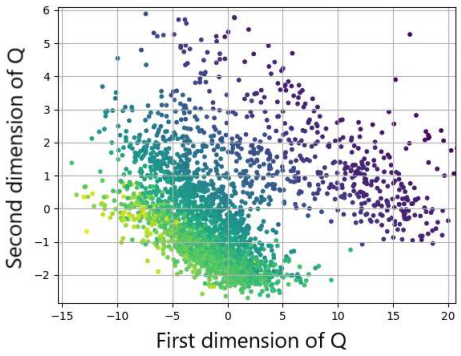


FIGURE 4: 1D first principal component with fictitious noise.



(A) Case with error columns



(B) Case without error columns

FIGURE 5: Plots of first and second PCs

see the points density I made them a little bit transparent (changing the alpha parameter in the scatter method from the default 1 to 0.5). We can see that there are two main clusters, one on the right that has more purple points, one on the left were the majority of the points tend to the green shades. It is not a clear division, in the sense that some purple points and some green points are overlapped or far away from the main cluster, but the majority of the galaxies with similar Mcz seem already to group together. In any case, with this graph is justified and evident that the first principal component explain a great portion of our samples.

While the first dimension of Q in the two cases is the same, if we plot combinations on the first, second and third dimensions we will see some significant changes. Here I show you some projections, while in the script you can find the code to plot a 3D dynamic plot.

In Fig.5 I show you the first dimension of Q and the second dimension in both cases: where I deleted all the error columns and when I kept them. Here, It is clear that galaxies with similar Mcz are grouped together and galaxies with different Mcz are separated. In fact with both the cases represented in Fig.5 we kept more than 80% of the original information. To have a better idea is essential to see the 3D plot that you can find in the python code. In the 3D plot the points of different color are even better separated and we make easier and

more precise the future work of K-Nearest-Neighbour algorithm.

4 PCA and K-Nearest Neighbour

In the last requirement we have to use the K-Nearest Neighbour to predict the Mcz of some galaxies. Basically, I imported from sklearn.neighbors the KNeighborsRegressor. Then I set the number of neighbours equal to 5. Next, I trained the regressor with the previous data (the 2500 galaxies of the train set that we talked about before). After this, I used the predict method in order to find a prevision of the Mcz values of the galaxies of the test set. I computed the Mean Absolute Error (MAE) and the Mean Relative Error (MRE) as follow:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |\widehat{Mcz_i} - Mcz_i|$$

$$MRE = \frac{1}{N}\sum_{i=1}^{N} \frac{|\widehat{Mcz_i} - Mcz_i|}{|Mcz_i|}$$

As we can see from table.2 we obtain better result when we delete the error columns. In fact, they do not represent natural information about the galaxies and they do not have some meaningful hidden information. Probably, this columns generate some noise in the algorithm and they are not so useful in the prediction pipeline. Anyway, the gap between the two final result doesn't seem so much relevant.

I thought about some ways to improve even better the precision:

- tune the parameters of K-Nearest-Neighbour
- increase the information kept

if we change the number of nearest neighbour we can see that the result don't change in a significant way. In order to obtain a more precise result I suggest to increase the number of PCs, that is translated in 'keep more columns'. Obviously we can get a more precise prediction, but in case of a larger database it could slow down our algorithm and it could be a problem. It would be a balance between precision and computational cost.

	keep 95% of information	
	with error columns	**without error columns**
MAE	3.92×10^{-2}	3.87×10^{-2}
MRE	5.76×10^{-5}	5.68×10^{-5}

TABLE 2

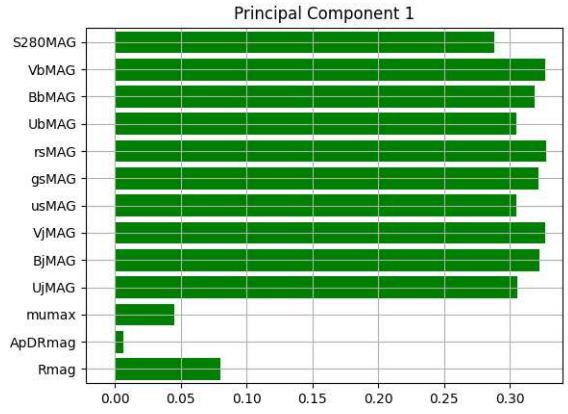


FIGURE 6: Galaxy luminosity. PC1. Section 2

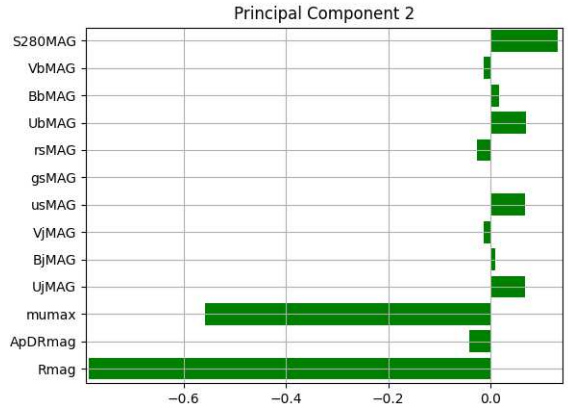


FIGURE 7: Galaxy kinematic. PC2. Section 2

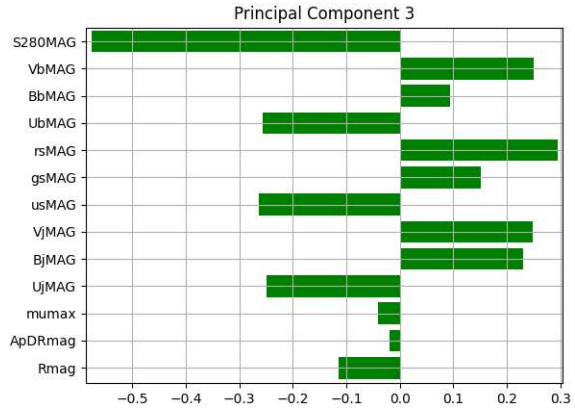


FIGURE 8: VB-iS280U. PC3. Section 2