Homework of Principal Components Analysis

Computational Linear Algebra for Large Scale Problems
Politecnico di Torino

Hajali Bayramov - S288874
Hafez Ghaemi - S289963

May 24, 2021

Abstract

In this homework we applied the PCA to a dataset of galaxy characteristics and gave interpretations to the PCs (principal components) and other results. A file named "INSTRUCTIONS.txt" containing the explanations how to run the codes and reproduce the results presented in this report, is attached.

1 Introduction

A dataset with the characterization of galaxy observations used to build a model for estimation of *redshift* is used in this homework. 65 attributes with 3462 rows describe the dataset.

2 Problem Overview

As can be seen from the dataset, it is quite large to be used in the building a regression the model. Thus, implementing principal component analysis (PCA) before building a model in order to decrease the size of the data used and time for building the model. Accuracy, on the other hand may or may not differ much. PCA analysis is not specifically applied for the accuracy boost.

Dataset is divided into train and test and *Mcz* is used as target variable in our model.

3 Methodology

3.1 Preprocessing

For the preprocessing of the dataset we have few of steps. First, check if there exist rows with null value(s) in the columns. After dropping those rows, we need to make sure that useless columns like IDs and the columns related to *redshift* are dropped as well. After that we can make sure that dataset is properly preprocessed.

If we take a look at the correlation values among the columns, we can see that there are many highly correlated attributes inside the dataset. This gives us a hint that if PCA is used we can reach good accuracy with fewer number of components.

3.2 Preparation of the dataset

In order to analyze the results and errors our potential regression model(s) will achieve, we need first to split our dataset into train and test sets. For the training dataset, 2500 random rows are selected from the whole dataset.

3.3 PCA

To analyze principal components we have two options:
1. without normalizing the input data, 2. transfering the data into standard scores. After doing PCA using *sklearn* library of *python*, we now can analyze the explained variances and their ratios graphically. Looking at the figure 1, it can be noticed that doing PCA without normalizing the input gives us 90% explained variance ratio with much fewer components than using normalized data. Although, after 30 components, cumulative sums of explained variance ratios look the same, and reach maximum.

Also, figure 2 shows us that as few as 2 components can give us good explained variance ratio if we do not implement z transform, i.e. 0.75.

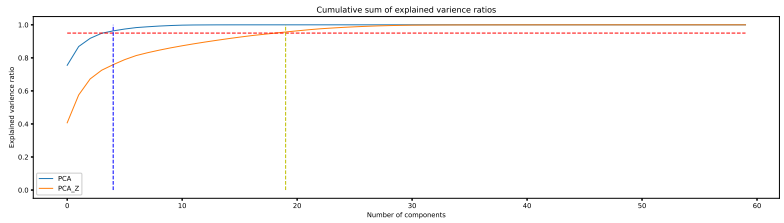


Figure 1: Cumulative sum of explained variance ratios over the number of components (Auxilary dashed lines are to help the viewers notice important points, such as 90% explained variance ratio)

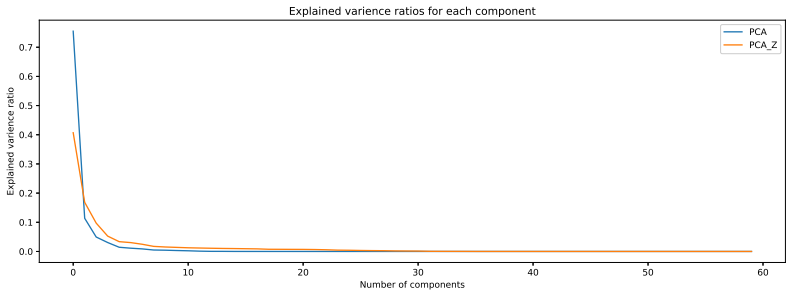
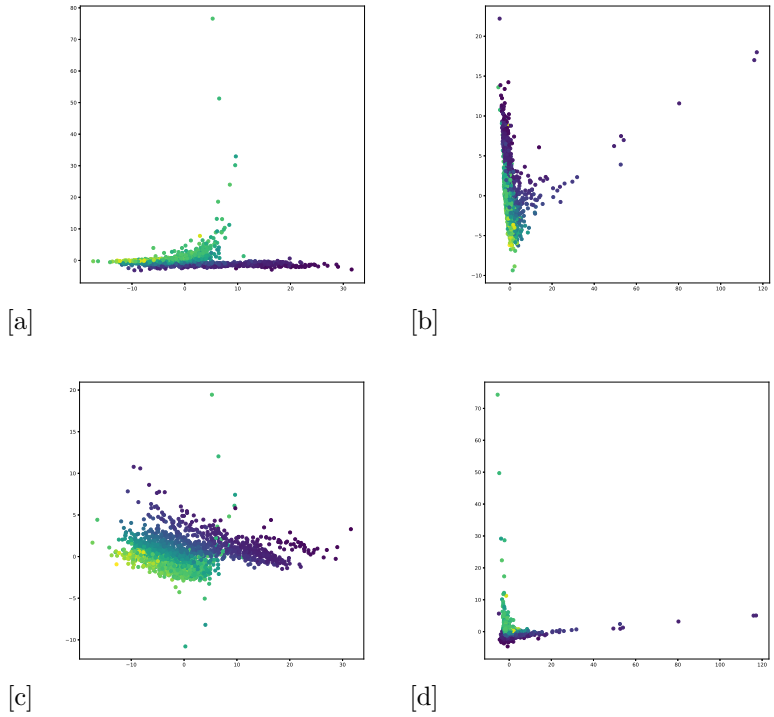


Figure 2: Explained variance ratios for components

If we look at the cumulative sum of explained variance ratio after every

component for both PCA and PCA with z-scores, we see that the former reaches 99% with 8 components while that number is 26 for the latter.

We can also have a look at some example 2D graphs taken from representation of data in the m-dim space of PCs with figure 3. It can be seen that after applying PCA, points can easily be represented by a color spectrum which represent Mcz values. This, per se, can give us a hint that our model is going to fit well to the data with reduced dimensionality. Moreover, this visualization shows that PCA has performed well in the separation of the components.



[a] [b]

[c] [d]

Figure 3: (a) Qm w/o z, 1st and 2nd (b) Qm w/ z, 1st and 2nd (c) Qm w/o z, 1st and 3rd (d) Qm w/ z, 1st and 3rd

3.4 Methodology and Results

As it is required in the task, we build multiple k-NN regressor models to obtain the R^2 scores and error terms among those with PCA, PCA with normalized data, and without PCA. It is expected from the model to be fast and memory-efficient, because it uses less data by keeping the explained variance ratio high. So, basically the data with PCA can explain a "high portion" of the original data with less data.

After completing the evaluations, table 1 is built. We can see that our expectations were correct: even if the accuracy and error do not differ much while using the PCA, execution time surely does, many folds indeed.

For the evaluation, R^2 score is used which is common practice for regression models. In the equation below SS_{RES} is the residual sum of squared errors of

our model, SS_{TOT} is the total sum of squared errors of our model, y_i is the true value, $\hat{y}_i$ is the predicted value, while $\bar{y}_i$ is the mean value.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}$$

For the error, two types are used: MAE(Mean Absolute Error) and MRE(Mean Relative Error). N is the number of elements, Mcz_i is the true value, $\widehat{Mcz_i}$ is the predicted value.

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

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

Table 1: Comparison table with accuracy and error of the models

knn models	R^2	MAE	MRE	time	component
with z-score w/o PCA	0.896	0.067	0.123	0.337	-
with z-score with **PCA**	0.958	0.043	0.073	0.05	8
w/o z-score w/o PCA	0.981	0.03	0.048	0.141	-
w/o z-score with **PCA**	0.981	0.031	0.049	0.029	8

4 Conclusion

To sum up, this work proved the usefulness of PCA on a large dataset in terms of time and memory. The regression model built after applying PCA has generated the results faster with fewer number of attributes while maintaining the high R^2-low error. Exploitation of PCA is especially useful when there are too many features and there is need for protyping with large number of different models are being built. There might be cases, eg. NN, Deep Learning, in which building take too much time. Machine learning practitioners can make use of PCA to gain time and memory efficiency.