

Table of content

Table of content.....	1
Rationale.....	2
Background.....	2
Introduction of data set	2
Goal	2
Challenging problem	3
Solutions - with linear algebra theories and techniques	3
Step1. Find Sample covariance matrix	3
Step2. Normalized $x_{k,i}$ and find correlation matrix	3
Step3. Find λ_1, λ_2 of matrix ρ	4
Experiment result of PCA steps.....	4
Analysis of PCA experiment results.....	5
Further improvement	6
1. Improved by missing value strategy	6
2. Improved by removing redundant features	7
3. Comparison	8
Discussions	8
Problems	8
Future work and improvement.....	9
Summary	9
References.....	10
PCA Theorem	10
Coding	10
Code and experiment results	10

Rationale

During the semester, I took a machine learning course which focuses on learning of training theories and algorithm. As for the data preprocessing of a data set, ex. feature selection, we did not get into many details. Thus, it always bothers me when encountering feature selection. I did it by manual selection with my own feelings without mathematical algorithm, or by mathematical algorithm which I am not familiar with the theorem behind.

However, after I learned Principal components analysis from Linear Algebra, I finally realized that I didn't have a deep understanding of the data itself. Therefore, after learning PCA, I want to have a deeper understanding of the relationship between various features in a data set.

Background

Introduction of data set

A telephone company wants to predict whether the customers would stop using its services and why the customers stop using its services. Therefore, the telephone company collected 4225 data each with 44 features. The features included are as follow. The result of prediction can be divided into six categories: No Churn, Competitor, Dissatisfaction, Attitude, Price and Other.

Count	State	Tenure in Months	Online_Security	Contract	
Gender	City	Offer	Online_Backup	Paperless_Billing	
Age	Zip Code	Phone Service	Device_Protection	Payment_Method	
Under_30	Lat Long	Avg Monthly Long-	_Plan	Monthly_Charge	
Senior Citizen	Latitude	Distance Charges	Premium_Tech_	Total_Charges	
Married	Longitude	Multiple_Lines	Support	Total_Refunds	
Dependents	Satisfaction Score	Internet_Service	Streaming_TV	Total_Extra_Data_Charges	
Number of	Quarter	Internet_Type	Streaming_Movies	Total_Long_Distance_Charges	
Dependents	Referred a Friend	Avg_Monthly_GB	Streaming_Music	Total_Revenue	
Country	Number of Referrals	_Download	Unlimited_Data	Total_Long_Distance_Charges	

Goal

The target for me now is to ***analyze relationship between different features*** by Principal Components analysis I've learned in class. Thus, I will not discuss about the training algorithm as well as the training results.

Challenging problem

1. Missing values

Since the data I get exists some missing values that the user did not want to fill in, I deal with the problem by filling the missing data with a constant value. (Encode as a new class)

2. Encoding strategy

Since some of the features are filled in with "string values", I encode them with float variable to facilitate our analysis.

Solutions - with linear algebra theories and techniques

Step1. Find Sample covariance matrix

First, I reorganized the data set to Matrix A which each element of a column should minus the mean of the feature.

$$A = \begin{bmatrix} x_{1,1} - \bar{x}_1 & \cdots & x_{1,n} - \bar{x}_n \\ \vdots & \vdots & \vdots \\ x_{m,1} - \bar{x}_1 & \cdots & x_{m,n} - \bar{x}_n \end{bmatrix}$$

Where m represents the number of data and n represents the number of features

Then, by $\Sigma = \frac{1}{m-1} A^T A$ I can get the sample covariance matrix Σ .

Step2. Normalized $x_{k,i}$ and find correlation matrix

Now, I normalize matrix A to B which each element of a column should be divided by $\sqrt{\text{variance}}$ of the feature.

$$B = \begin{bmatrix} y_{1,1} & \cdots & y_{1,n} \\ \vdots & \vdots & \vdots \\ y_{m,1} & \cdots & y_{m,n} \end{bmatrix}$$

Where m represents the number of data and n represents the number of features

Then, by $\rho = \frac{1}{m-1} B^T B$ I can get the sample covariance matrix ρ .

Since I want to verify whether I calculated the correlation matrix correctly, there is a written function which I can get the correlation matrix easily.

```
correlation_matrix = df.corr() #df represent our dataset by DataFrame
```

Thus, by double checking $\rho = \text{correlation_matrix}$, I can ensure both step1 and 2 works well.

Step3. Find λ_1, λ_2 of matrix ρ

For correlation matrix ρ , I can find all of the eigenvalues and corresponding eigenvectors by `linalg` package.

I choose the first and second eigenvectors as weight e_1 and e_2 .

By finding the weighted index z , we call $e_1^T y = e_{1,1}y_{k,1} + e_{1,2}y_{k,2} + \dots + e_{1,44}y_{k,44}$ as **First Principal Component** and $e_2^T y = e_{2,1}y_{k,1} + e_{2,2}y_{k,2} + \dots + e_{2,44}y_{k,44}$ as **Second Principal Component**.

Application / Example

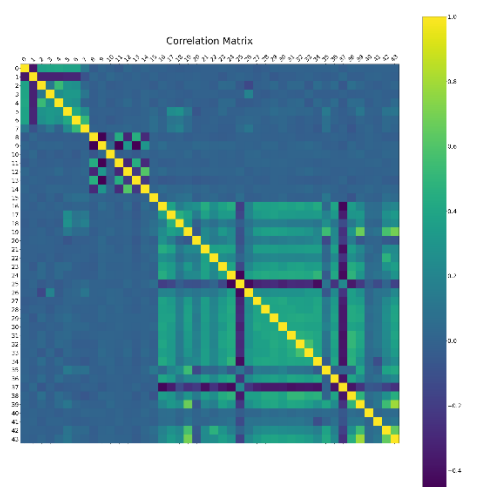
Experiment result of PCA steps

Step1. Find Sample covariance matrix

- The result matrix can be found in https://github.com/joanne40226/2021_Linear_algebra_FALL_finalproject/tree/master/experiment_results - covariance_Matrix.xlsx/
since the size of the matrix is too big that it is hard to be visualized in the report.
- Since the dataset hasn't been normalized, the elements in the covariance matrix differ. Thus, it is hard to show the differences by visualizing by colors.

Step2. Normalized $x_{k,j}$ and find correlation matrix

- The result matrix can be found in https://github.com/joanne40226/2021_Linear_algebra_FALL_finalproject/tree/master/experiment_results - correlation_Matirx.xlsx
since the size of the matrix is too big that it is hard to be visualized in the report. However, it can be visualized by colors and showed clearly since the dataset has been normalized, the data are scaled equally.



As the figure showed, we can see that the correlation matrix is a symmetric matrix, and the highly positively related features are colored by a brighter green.

Step3. Find λ_1, λ_2 of matrix ρ

After calculations, I get the results as follow.

$\lambda_1 = 8.680715488475105$

$V1 = \begin{bmatrix} 2.57752041e-03, & -1.17623073e-02, & 1.16538289e-02, & 1.98018459e-04, \\ 1.32544336e-02, & 3.23293107e-02, & 1.10386123e-02, & 5.82587084e-03, \\ -1.03163395e-02, & 1.38348110e-02, & 9.33652949e-03, & -1.84310720e-03, \\ 7.61571614e-03, & -1.44574311e-02, & 1.25339934e-02, & 5.36214308e-03, \\ 2.06704420e-01, & 2.00882500e-01, & 1.06082468e-01, & 2.03947073e-01, \\ 8.18466646e-02, & 1.93231850e-01, & 1.36842788e-01, & 2.07406040e-01, \\ 2.26248837e-01, & -1.43302988e-01, & 1.52143733e-01, & 2.08125556e-01, \\ 2.17702771e-01, & 2.25251863e-01, & 2.16948551e-01, & 2.27245048e-01, \\ 2.28503410e-01, & 2.19370094e-01, & 2.14793714e-01, & 7.90798870e-02, \\ 1.97634410e-01, & -2.03339068e-01, & 2.38448344e-01, & 2.22992468e-01, \\ 3.11424451e-02, & 4.79051282e-02, & 1.68241390e-01, & 2.22435934e-01 \end{bmatrix}$

$\lambda_2 = 3.177065092391582$

$V1 = \begin{bmatrix} 0.36794131, & -0.31677521, & 0.30312665, & 0.29124306, & 0.32480929, \\ 0.38191737, & 0.37667234, & 0.20800247, & -0.10345095, & 0.08391263, \\ -0.012909, & -0.09689543, & 0.08530804, & -0.08522418, & 0.08574828, \\ 0.04178878, & -0.0432891, & 0.05921473, & 0.10939283, & 0.09753221, \\ -0.08560133, & -0.0463961, & -0.01771655, & -0.0076357, & -0.06899802, \\ 0.05888823, & -0.03821653, & -0.03515761, & -0.0333409, & -0.00237678, \\ -0.01952375, & -0.02524696, & -0.03665582, & -0.04716375, & -0.06504295, \\ 0.13331984, & -0.04554855, & 0.02525987, & -0.03709976, & 0.07629343, \\ -0.00479749, & -0.01124446, & 0.08512717, & 0.08640113 \end{bmatrix}$

Analysis of PCA experiment results

After getting all the eigenvalues and eigenvectors from the correlation matrix ρ , I can now calculate the ratio of variance for each eigenvalue by

$$\frac{\lambda_j}{\sum_{j=1}^d \lambda_j}$$

The result is as follow, I then visualized it to the figure below to make it even clearer.

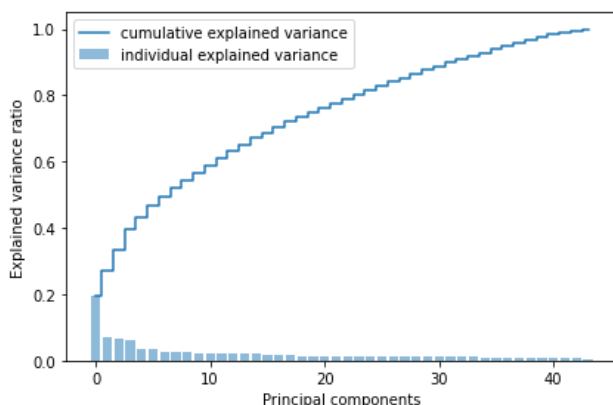


Figure1. individual and cumulative explained variance

[0.1972423038054866, 0.0721893868774706, 0.06463219260853068, 0.060676685764343166, 0.03657377619378347, 0.03409933872531678, 0.026905871000797436, 0.026468139293151453, 0.02373162488605451, 0.02278052115308888, 0.022599464663530095, 0.02211844980562858, 0.02115461375290483, 0.02002527367832385, 0.019451792724728828, 0.017475482801226624, 0.01624383674909187, 0.015483361953386605, 0.014513475929109855, 0.014228198467951987, 0.014155816555998482, 0.013727546804152115, 0.013417752144106176, 0.013170159595120029, 0.012857449938318652, 0.012428518036048329, 0.012119913710320824, 0.012005726069228619, 0.011744090922018125, 0.011604362673625024, 0.011151764257148181, 0.011095487929013636, 0.010492637796927598, 0.010370751685454262, 0.010006931026582578, 0.009556976168275846, 0.009449873318313777, 0.00914896692082696, 0.00902699427067582, 0.008532651404858146, 0.008103399625432686, 0.00661254397324625, 0.006245212535651495, 0.004381129994473522]

Figure2. Variance ratio of each eigenvalue

It can be seen from the figure that the first principal component accounts for 20% of the variance, the second principal component accounts for about 7%, and the sum of both for 27%. We can say that the sum of the first two principal component did not take up a large proportion of the variance. We can see that the dataset may not have strong correlation that only a quarter of the variance are included by the first two principal component.

Moreover, the second principal component accounts for about 7% and the 44th principal component accounts for about 0.4%. The difference is small. Thus, we can say that the second principal component might not help us a lot since it does not

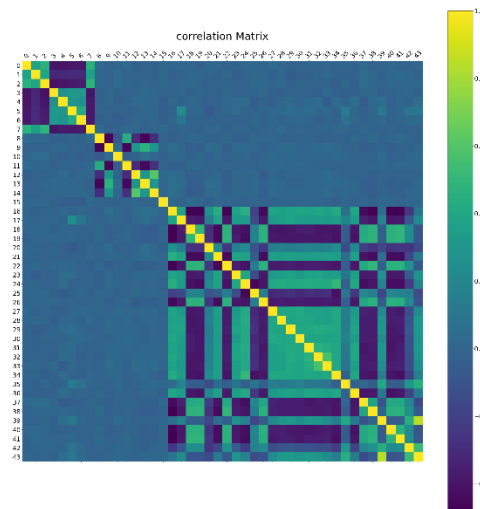
take up a significant and identifiable portion of the variance.

Further improvement

1. Improved by missing value strategy

Since the inaccuracy may be caused by inappropriate data preprocessing, I tried another strategy for missing value – fill in by mean of the feature. The results are as follow.

The figure below is the visualized correlation matrix. In comparison with the correlation matrix, we found by previous strategy. It is obvious that more correlations are negatively related.



The first two eigenvalues and eigenvectors are found as follow.

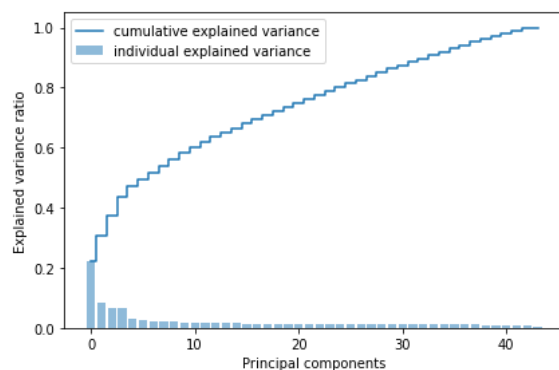
$\lambda_1 = 9.827835227567043$

$v_1 = [4.00646483e-04, 1.01211354e-02, 7.26119103e-03, 3.51375437e-03, -7.06114275e-03, -1.16983379e-02, -4.56597366e-03, 2.15470654e-04, 1.00036750e-02, -1.46021863e-02, -5.88142727e-03, 4.11210578e-03, -1.02830609e-02, -1.52013394e-02, -1.40137454e-02, 2.37011457e-03, -2.19068842e-01, -1.87563680e-01, 2.18074791e-01, 2.20287912e-01, -1.11067686e-01, -2.00687692e-01, 2.13848618e-01, -1.91555828e-01, -2.20025404e-01, 1.24356456e-01, 2.12053086e-01, -1.95756817e-01, -1.99356219e-01, -2.03960366e-01, -2.00199435e-01, -2.06491883e-01, -2.08845122e-01, -2.02736925e-01, -2.10881463e-01, -3.59486342e-02, -1.97121229e-01, 2.04382670e-01, 2.13099502e-01, -8.37198453e-02, 2.15582494e-01, 2.14785123e-01, 8.68751353e-02, -9.75629809e-02]$

$\lambda_2 = 3.635584368105596$

$v_1 = [3.76380241e-01, 3.19077366e-01, 3.74184943e-01, -3.24872955e-01, -3.18560633e-01, -3.46223101e-01, -3.31653289e-01, 3.71691477e-01, 5.70834103e-02, -4.24579025e-02, 1.31337187e-02, 4.93095170e-02, -3.76138205e-02, -4.26115298e-02, -3.97359853e-02, 2.17572844e-02, 1.29431374e-02, -3.62901587e-02, -1.40354820e-02, -2.51340279e-02, 4.30706069e-02, 2.01616663e-02, -2.84871445e-02, -1.34676757e-02, 1.13530374e-02, 3.79748624e-03, -1.42664601e-02, 5.17249428e-03, 9.15277443e-03, -2.55810913e-02, -7.91924687e-03, -1.24804093e-02, -3.85613784e-03, -3.20786671e-04, 9.85038145e-03, -6.18427738e-02, 1.25589149e-03, 9.52629399e-03, -2.34187605e-02, -7.86368732e-02, -9.89374710e-03, -1.90803362e-02, -6.75335767e-02, -7.64282944e-02]$

The individual and cumulative explained variance results are visualized as follow.



[0.2233070377988575, 0.08260736541779322, 0.06760182941364659, 0.0646390146061657, 0.03240368425473734, 0.025133228421839873, 0.022981471842085103, 0.02235779164760385, 0.02183070477325126, 0.019111392795284496, 0.018781280023146506, 0.01797122599675786, 0.016672578106425428, 0.015626171774856482, 0.014789941862829531, 0.014559444188456812, 0.014422143295417352, 0.014204238515741372, 0.013986917203382701, 0.013567985545147874, 0.01328369509856143, 0.013117971403647436, 0.013005043535312741, 0.012887930845559201, 0.012695135210031832, 0.012484551979490702, 0.012356383151209785, 0.012256201420460603, 0.01202264764092449, 0.01184692456138759, 0.011693750713061946, 0.011512047754287815, 0.011367298597067877, 0.011238523949400562, 0.011109498799090008, 0.010866613642465668, 0.010823966825319027, 0.010608675165159606, 0.009731966725249449, 0.009459665218799705, 0.00909092629264455, 0.008545413387684157, 0.008376162584937712, 0.0030635576035241484]

From the statistic above, we can see that the first principal component accounts

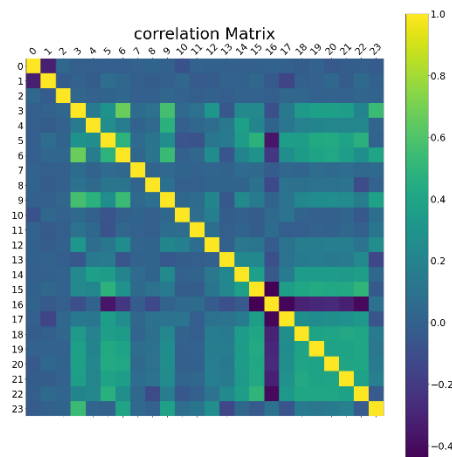
for 22% of the variance, the second principal component accounts for about 8%, and the sum of both for 30%, which is slightly better than the previous result. Moreover, the second principal component accounts for about 8% and the 44th principal component accounts for about 0.3%. The difference is small however slightly better than the previous result.

Thus, we can say that after changing strategy of missing values by filling in with mean values, the performance does not go well however better than the strategy to fill in with constant values.

2. Improved by removing redundant features

Since the poor performance may also cause by redundant features, I try to remove features I considered to be useless (ex. Latitude, longitude, lat-longitude, zip code, Under30 ...etc.) and remain only 24 features.

The correlation matrix is as follow.



The first two eigenvalues and eigenvectors are found as follow.

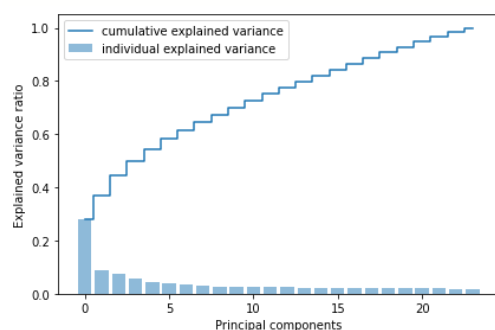
$$\lambda_1 = 6.72355414$$

$v_1 = [-1.11777572e-02 \ -8.85418359e-03 \ -5.41833123e-01 \ 1.37600668e-02 \ 6.44731896e-02 \ -1.65560736e-02 \ 1.22551730e-01 \ -2.44149311e-01 \ 1.73837613e-03 \ 4.19032057e-02 \ 7.47930100e-01 \ -1.28390518e-02 \ 6.37056678e-02 \ 1.66977028e-01 \ -2.07261509e-02 \ -9.18930689e-02 \ -3.93790237e-02 \ 8.52479561e-02 \ 2.12054227e-02 \ 9.56953906e-03 \ 5.26972142e-02 \ 1.10092524e-01 \ -2.38436161e-02 \ 1.89968837e-03]$

$$\lambda_2 = 2.09391863$$

$v_2 = [-7.13335097e-03 \ 2.46667256e-02 \ -5.98742915e-01 \ -2.69310812e-03 \ 1.63885888e-02 \ -1.52306095e-02 \ -5.32488412e-02 \ 8.02976732e-02 \ -7.17840684e-03 \ -6.37669954e-02 \ -2.58831492e-01 \ -3.17048263e-02 \ 2.80580330e-04 \ -8.62896802e-02 \ 5.23468961e-02 \ 1.19651220e-01 \ 3.00342854e-01 \ -1.60103305e-01 \ -1.29178967e-01 \ 7.12241536e-02 \ -5.01690223e-01 \ -1.83442787e-01 \ 1.46326024e-01 \ 3.00137807e-01]$

The individual and cumulative explained variance results are visualized as follow.

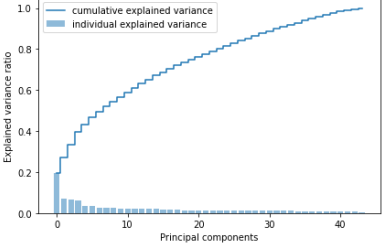
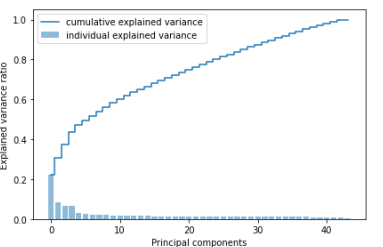
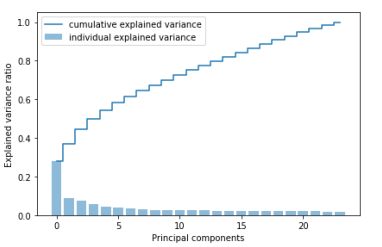


[0.2800817976234536, 0.0872259642232883, 0.07590853924908929, 0.05710921849065819, 0.0427526147915797, 0.04081054974244648, 0.03261922101842224, 0.030794290161644255, 0.027240699703755436, 0.025879710375094806, 0.02559590677179664, 0.02480192004150746, 0.02390846345259219, 0.02354617511507383, 0.022950093375203377, 0.022138058206862554, 0.021667853639753067, 0.021226935417024986, 0.02090737649409911, 0.020653408064907862, 0.020131586995333815, 0.019863172187459022, 0.017306268230621008, 0.01488017662833298]

From the statistic above, we can see that the first principal component accounts

for 28% of the variance, the second principal component accounts for about 8.7%, and the sum of both for 36.7%, which is slightly better than both previous results. However, the second principal component accounts for about 8.7% and the 44th principal component accounts for about 1.4%. The difference is smaller than both previous results since for this strategy, there are only 24 eigenvalues remain, causing less difference between smaller eigenvalues.

3. Comparison

Visualized Result			
Missing value strategy	filling in with another constant value	filling in with feature mean	filling in with feature mean
Additional Strategy	X	X	removing redundant features.
Sum ratio of variance by first two principal components	27%	30%	36.7%
Performance			BEST

Discussions

Problems

From the experiment results from PCA steps, we can verify that the calculation taught by class works and it can explain the data clearly.

Ex1. The correlation matrix is symmetric and the diagonals of the matrix equal to 1.

Ex2. The largest eigenvalue of correlation matrix maximizes $e_1^T B^T B e_1 = \lambda_1$ which can also prove the Rayleigh's principal.

However, as for the analysis afterwards, we can see that the result doesn't go well. As mentioned in the previous discussion, I thought that the second principal component might not help a lot since it does not take up a significant and identifiable portion of the variance.

This problem may result from the data preprocessing which has not been done well. Since I deal with the missing value only by a different but meaningless float variable or by mean values and encode the “string” values by meaningless float variables, it might cause a huge amount of loss and inaccuracies.

Moreover, there are a lot of features that I considered to be useless. For example, I have the information of the country, city, state, zip code, latitude, longitude and also the lat-longitude, however I might only need one of it and the rest of the information will be known. Thus, redundant information may cause errors from customers when filling in the form or from clerks when recording data. It may also lead to loss and inaccuracies.

From the experiment of the improved strategies, we can verify that the data preprocess influences the upcoming result indeed. However, the performance can still be enhanced by better preprocess strategies.

Future work and improvement

Since I considered the result to be strongly influenced by the imperfect data preprocessing, the outcome can be improved by better preprocess. Thus, for the future work, I hope to find better strategies for missing value processing and data encoding. Moreover, I can try to clean up the raw data first by removing redundant features manually which might be able to reduce noises and avoid loss.

By improvement above, I hope the sum of the first two principal component to take up at least 50% of the variance.

Summary

From this experiment and discussion on principal component analysis, we can verify the theorem of PCA taught by class and explain the dataset clearly. However, there is still space for improvement by further data preprocessing. Thus, it is important for us to have tidy data that can be analyzed easily so that better results can be obtained. There is still a lot for me to do in order to get a better performance and logical result.

References

PCA Theorem

- [1] Jin, Jiashun, and Wanjie Wang. "Influential features PCA for high dimensional clustering." *The Annals of Statistics* 44.6 (2016): 2323-2359.
- [2] Suykens, Johan AK, et al. "A support vector machine formulation to PCA analysis and its kernel version." *IEEE Transactions on neural networks* 14.2 (2003): 447-450.

Coding

- [3] <https://arbu00.blogspot.com/2017/02/6-principal-component-analysispca.html>

Code and experiment results

The complete code and experiment results including raw data can be found on https://github.com/joanne40226/2021_Linear_algebra_FALL_finalproject