Assignment 2:
Principal Component Analysis and Multidimensional Scaling

Noah Steidle

April 10, 2022

Abstract

Principal Component Analysis (PCA) and *Multidimensional Scaling* (MDS) are well-known tools in data analysis which at their core realise a dimensionality reduction of given input data, but in detail describe two different approaches on how to reach that goal. In this context, the two methods will be introduced both mathematically and algorithmically, before the second part considers a problem in stock market analysis which will be addressed via PCA.

1 Introduction

The amount of available data and also the rate of data collection are growing day by day, leading to an urgent necessity for powerful and easy-realisable data analysis tools. As big data collections are confusing to understand and often contain repetitive and disturbed sections, it is a primary goal to reduce the given input to a more understandable set of data without giving up too much information. In mathematical terms this process can be interpreted as a dimensionality reduction, and *Principal Component Analysis* (PCA), as well as, *Multidimensional Scaling* (MDS) describe two approaches to reach this goal.

Hereby, PCA utilises applied linear algebra, in specific a vector space transformation, to map (i.e. project) the data to a lower-dimensional space. This approach can be used in many different contexts and is common to use as a first step when big data collections are faced. Besides, it offers other applications like de-noising signals, blind source separation and data compression, and therefore often is called "one of the most important results from applied linear algebra".[8, 9]

On the other hand, a different approach is to analyse the proximity of data and find a mapping into a lower-dimensional space which preserves this proximity in a certain sense; this technique is the general idea behind MDS. According to [1] MDS mainly serves four purposes: visualization, testing and analysing differences between objects, discovering dimensions that underlie measures of similarity, and providing a psychological model based on a specifically chosen distance function.[1, 7]

In the proceeding of this report both methods PCA and MDS are introduced with a focus on the mathematical background and by providing algorithmically ideas to realise the techniques. Furthermore, a problem in finance, namely index replication, is considered and solved with the help of PCA. Hereby, the methodology is explained step by step and all necessary code frames are allocated in the appendix.

2 Methods

2.1 Principal component analysis (PCA)

This short introduction to PCA is based on [8] and [9].

Data in many dimensions can be confusing and hard to analyse. A simple and non-parametric method to reduce common data to a lower dimension while preserving as much information as possible is called *Principal Component Analysis* which aims to determine principal components of a set and using them (or a subset) to perform a change of basis. Hereby, principal components of a collection of points describe an ordered set of directions that maximise the variance in each dimension. In other words, the complete data is reduced to a lower-dimensional set (based on the principal components) which contains the most "important" information and reduces redundancy.

Mathematically spoken, we are given data $X \in \mathbb{R}^{m \times n}$ and would like to find a projection $P \in \mathbb{R}^{m \times m}$ such that

$$Y = PX$$

for $Y \in \mathbb{R}^{m \times n}$. PCA defines the change of basis to be "best" when the variance of the data (signal) is maximised and at the same time the covariance is minimised. This implies to consider the covariance matrix

$$C_Y = \frac{1}{n-1} YY^T$$

and to find a diagonalization of C_Y. In fact, by assuming P is orthonormal, one can use the tools of linear algebra to find

$$C_Y = \frac{1}{n-1}YY^T = \frac{1}{n-1}PXX^TP = \frac{1}{n-1}PAP^T$$

with $A := XX^T \in \mathbb{R}^{m\times m}$ symmetric. This means that A is diagonalisable with $E \in \mathbb{R}^{m\times m}$ orthonormal and $D \in \mathbb{R}^{m\times m}$ diagonal,

$$A = EDE^T.$$

Finally, one chooses the rows of P to be the eigenvectors of A which implies $P = E^T$ and therefore

$$C_Y = \frac{1}{n-1}PAP^T = \frac{1}{n-1}P(EDE^T)P^T = \frac{1}{n-1}(PP^T)D(PP^T) = \frac{1}{n-1}D.$$

As a result, the covariance matrix of the transformed data has been diagonalised. The rows of P are named principal components and as they are eigenvectors of the covariance matrix they contain information about the "importance"; when structured in descending order they offer valuable insight to choose an order for the dimensionality reduction.

In general, four assumptions/constraints have to be considered before applying PCA

1. The projection is linear (change of basis);
2. The projection is orthonormal;
3. The data is Gaussian;
4. Large variances have important dynamics.

Even though all assumptions and constraints can be weakened by employing further-developed approaches, PCA offers a fast and reliable method and is widely used in a variety of contexts.

2.2 Multidimensional scaling

The information for this introduction is mainly taken from Chapter 3 of [7].

A set of non-linear techniques to analyse data based on proximity is called *Multidimensional Scaling* (MDS). Also, MDS is a method to execute dimensionality reduction, but opposed to PCA, focuses on obtaining the same proximity of data points in the high-dimensional data as in the reduced-dimensional data. Since the development of MDS, many modifications have been presented, mainly differing in the choice of a function f which maps the distances of the original data into the lower dimensional space. This relation can be presented as

$$d_{rs} \approx f(\delta_{rs}),$$

where d_{rs} are the distances in low dimension and δ_{rs} are the dissimilarities in original dimension. For f continuous and monotonic, the process of MDA is called *Metric MDS* and stands in opposition to *Nonmetric MDS*, where only rank orders of dissimilarities have to be preserved. In the following only *Metric MDS* will be considered.

2.2.1 Classical MDS

The *Classical Multidimensional Scaling* describes a variant of MDS where the proximity measure is Euclidean,

$$\delta_{rs} = \left(\sum_i (x_{ri} - x_{si})^2 \right)^{\frac{1}{2}},$$

for $X = \{x_{ri}\}$ with $r = 1, \ldots, n$ and $s = 1, \ldots, p$.
By this choice, firstly, a closed form solution exists and, secondly, the function f is the identity function. In general, with this setup the classical scaling method relates to PCA, introduced in Section 2.1, and, in fact, often PCA and Classical MDS are used synonymously nowadays.

Following [3] the algorithm for Classical MDS looks like,

1. Form matrix $A = -\frac{1}{2}\delta_{rs}^2$;
2. Form matrix $B = HAH$ where H is the centering Matrix

   $$H = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T;$$

3. Find the spectral decomposition $B = V\Lambda V^T$;
4. The coordinates in the lower dimensional space are given by the rows of

   $$X = V_d\Lambda_d^{\frac{1}{2}},$$

   where V_d contains the eigenvectors corresponding to the d largest eigenvalues, and Λ_d contains the d largest eigenvalues along the diagonal.

Following the same steps as in PCA, the value of d can be chosen according to the context. In terms of visualization in MDS often $d = 2$ is used.

3 Problem

The second section of this report examines a usage of PCA in finance; more concrete the goal is to replicate a stock market index with a portfolio selected with the help of PCA. This approach was mainly inspired by [10].

A famous framework for pricing assets (e.g. a stock) is called the *Capital Asset Pricing Model* (CAPM) whose fundamental equation is given as

$$\mathbb{E}[R_i] = R_f + \beta_i \left(\mathbb{E}[R_m] - R_f\right),$$

where $\mathbb{E}[R_i]$ is the expected return on the capital asset, R_f is the risk-free rate of interest and $\beta_i \left(\mathbb{E}[R_m] - R_f\right)$ is the market factor return. More information on CAPM can be found, for instance, here [2].
As the return of an asset (a stock) should be at least equal to the return of the risk-free asset, the market factor $\mathbb{E}[R_m] - R_f$ is the primary driver of the overall return.

In the following, we would like to exemplary show that the first principal component of daily stock returns of a stock market index (here: GDAXI) can be used to approximate the market factor.

4 Methodology

The PCA is executed on data which describes the daily returns of the Dax Performance-Index (GDAXI) from the 1st of January 2022 to the 31st of March 2022. This period was chosen, because no components joined or left the index in this time. In general, the stock market index DAX30 lists 30 major German companies trading on the Frankfurt Stock Exchange.[1]

The necessary data for this time frame was accessed from Yahoo Finance [4] with the help of the *Yahoo finance and Quandl data downloader* created by Artem Lenskiy [5], which enables data scraping from Yahoo Finance in Matlab.

In order to analyse the provided data and visualize the information in a lower dimension, PCA, which was introduced in Section 2.1, was used. MDS was not employed in this composition, and interested readers are for example referred to [6].

[1] In 2021 the index was expanded from 30 to 40 components. Nonetheless, all further calculations were executed considering only 30 companies.

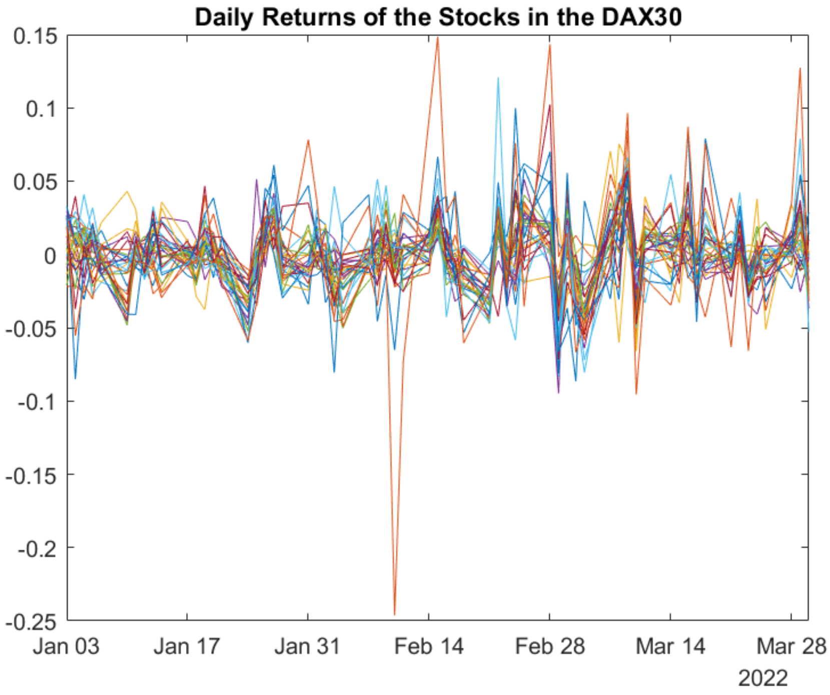

Figure 1: Daily returns of DAX30 from 1st of January 2022 to 31st of March 2022.

The code can be found in Appendix A. It may be noted that the code in the appendix makes usage of PCA as provided by Matlab and an own implementation of PCA.

5 Results

When visualizing the daily returns of the stocks in DAX30 from the 1st of January 2022 to the 31st of March 2022, the raw data is quite extensive and might not reveal important information; this can be seen here Figure 1.
By processing the data with PCA and deriving the first principal component, we reduce the amount of data points and keep only the most "important" information. The first principal component is visualized in Figure 2.

In fact, formulating a new portfolio and using a linear combination of the first principal component to determine the weights, results in an approximate for the whole index. This is explained by the derivation of PCA via variance and the overall market factor being the main driver for stock returns. Figure 3 shows the result in this context and confirms the possible approximation of the index via the first principal component.

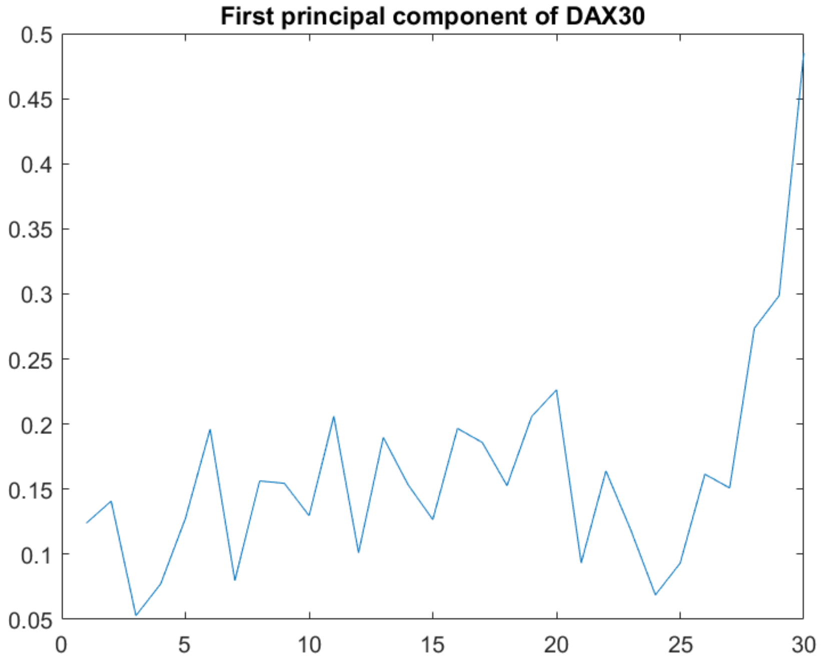

Figure 2: First principal component of DAX30 from 1st of January 2022 to 31st of March 2022.

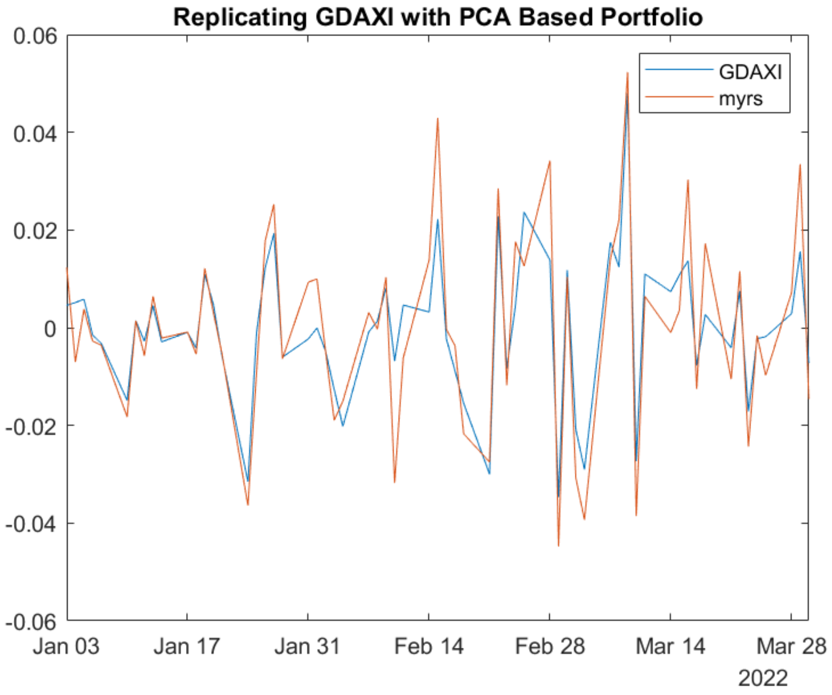

Figure 3: Comparison of DAX30 index and PCA replication of DAX30.

6 Conclusions

One may conclude with the successful application of PCA on the task of replicating a market index. Even though the replication was not perfect, it still is an easy way to achieve an approximation and also confirms the assumption in that sense that economical theory and numerical dimensionality reduction by variance conclude with the same result.

Overall, PCA and also MDS offer powerful techniques to obtain hidden information in the data, and are an useful addition to every data analysis because the implementation is easy and necessary assumptions are not too strong. In general, PCA is good when a lot of data is explicitly represented in $\mathbb{R}^d$, while MDS is efficient for few data points in very high dimension. MDS also introduced the research to further developed tools like Isomap.

References

[1] Inger Borg and Patrick J.F. Groenen. *Modern Multidimensional Scaling.* New York: Springer Science+Business Media, 2005.

[2] Richard A. Brealey, Stewart C. Myers, and Franklin Allen. *Principles of Corporate Finance.* 13. New-York: McGraw Hill, 2019.

[3] Chun-houh Chen, Wolfgang Härdle, and Antony Unwin. *Handbook of Data Visualization.* Berlin/Heidelberg: Springer-Verlag, 2008.

[4] Yahoo Finance. *MDAX PERFORMANCE-INDEX (MDAXI).* 2022. URL: `https://de.finance.yahoo.com/quote/%5EMDAXI` (visited on 05/04/2022).

[5] Artem Lenskiy. *Yahoo finance and Quandl data downloader.* 2021. URL: `https://github.com/Lenskiy/Yahoo-Quandl-Market-Data-Donwloader`.

[6] J. Tenreiro Machado, Fernando B. Duarte, and Gonçalo Monteiro Duarte. "Analysis of stock market indices through multidimensional scaling". In: *Communications in Nonlinear Science and Numerical Simulation* 16.12 (2011), pp. 4610–4618. ISSN: 1007-5704. DOI: `https://doi.org/10.1016/j.cnsns.2011.04.027`. URL: `https://www.sciencedirect.com/science/article/pii/S1007570411002218`.

[7] Wendy L. Martinez, Angel R. Martinez, and Jeffrey Solka. *Exploratory Data Analysis with MATLAB.* London: Chapman and Hall/CRC, 2005.

[8] Mark Richardson. "Principal Component Analysis". In: (2009). URL: `http://www.dsc.ufcg.edu.br/~hmg/disciplinas/posgraduacao/rn-copin-2014.3/material/SignalProcPCA.pdf` (visited on 04/01/2022).

[9] Jon Shlens. "A TUTORIAL ON PRINCIPAL COMPONENT ANALYSIS. Derivation, Discussion and Singular Value Decomposition". In: (2003). URL: `https://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pdf` (visited on 04/01/2022).

[10] Yao Lei Xu. *Stock Market Analytics with PCA.* 2020. URL: `https://towardsdatascience.com/stock-market-analytics-with-pca-d1c2318e3f0e` (visited on 04/04/2022).

A Code

```
%% Get Data

% ^GDAXI

gdaxi = ... getMarketDataViaYahoo('^GDAXI','1-Jan-2022','31-Mar-2022','1d');

% Components of ^GDAXI

data = zeros(63,30);
components_dax = ... {'SHL.DE','MRK.DE','SY1.DE','FME.DE','LIN.DE', ...
    'VOW3.DE','FRE.DE','1COV.DE','BMW.DE','DTG.DE','AIR.DE', ... ...
    'ALV.DE','PUM.DE','BAS.DE','RWE.DE','SIE.DE','ADS.DE', ...
    'HEI.DE','IFX.DE','CON.DE','DB1.DE','MTX.DE','HNR1.DE', ...
    'BAYN.DE','DTE.DE', ... 'DPW.DE','DBK.DE','ZAL.DE','HFG.DE','DHER.DE'};

cnt = 1;
for comp = components_dax

    data_comp = getMarketDataViaYahoo(char(comp), ...
        '1-Jan-2022','31-Mar-2022','1d');
    data(:,cnt) = ... (data_comp.Close-data_comp.Open)./(data_comp.Open);
    cnt = cnt+1;

end

% De-mean the data
data = data - mean(data);
```

```
%% Visualize Data

figure();
plot(gdaxi.Date,data(:,1));
hold on
for c = 2:30
    plot(gdaxi.Date,data(:,c));
end
hold off
xlim([datetime("2022-01-03") datetime("2022-03-30")])
title('Daily Returns of the Stocks in the DAX30');
```

```
%% Apply PCA by Matlab

[PC1,L1,latent1] = pca(data);

pcc11 = PC1(:,1);
weights1 = abs(pcc11)/sum(abs(pcc11));
myrs1 = (weights1.')*(data.');

% Visualize first principal component
figure()
plot(pcc11)
title('First principal component of DAX30');
```

```
%% Apply own PCA

% Calculate eigenvalue and eigenvectors of the Covariance ...
    matrix of the data
covMatrix = cov(data);
[PC2,L2] = eig(covMatrix);

% Sort matrix of eigenvectors in desc. order in terms of ...
    component variance
[¬,ind] = sort(diag(L2),'descend');
PC2 = PC2(:,ind);

% Save eigenvalues / principal components variances
latent2 = diag(L2);
latent2 = latent2(ind);

% Save Projection of original data on the princ comp vector ...
    space
L2 = data*PC2;

pcc12 = PC2(:,1);

weights2 = abs(pcc12)/sum(abs(pcc12));
myrs2 = (weights2.')*(data.');
```

```
%% Visualize ^GDAXI and Replication based on PCA

% PCA by Matlab
figure()
plot(gdaxi.Date,(gdaxi.Close-gdaxi.Open)./(gdaxi.Open))
hold on
plot(gdaxi.Date,myrs1);
hold off
xlim([datetime("2022-01-03") datetime("2022-03-30")])
legend('GDAXI','myrs','myrs2');
title('Replicating GDAXI with PCA Based Portfolio');

% PCA by Me
figure()
plot(gdaxi.Date,(gdaxi.Close-gdaxi.Open)./(gdaxi.Open))
hold on
plot(gdaxi.Date,myrs2);
hold off
xlim([datetime("2022-01-03") datetime("2022-03-30")])
legend('GDAXI','myrss');
title('Replicating GDAXI with PCA Based Portfolio');
```

The complete Matlab-files can be found here.