ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Bachelor Thesis Econometrie en Operationel Research

# The application of Penalized Matrix Decomposition to construct minimal variance Portfolios

Name student: Joël van der Werf

Student ID number: 483640

Supervisor: P.C. Schoonees

Second assessor: F. Frasincar

Date final version: 02-07-2023



**Abstract**

In this thesis we present an application of Penalized Matrix Decomposition (PMD) to construct latent factor models and built minimum variance portfolios from these factor models. We introduce the PMD algorithms in accordance to Witten et al. (2009). We will apply this to a simulation example to test the method. We will then built factor models and portfolios in accordance with Conlon et al. (2021). We will compare these portfolios with an equal weight portfolio for stocks from the S&P 500. We find that the new portfolio construction methods do not improve on the equal weights portfolio.

# Contents

# 1 Introduction

Dimension reduction techniques are more important than ever. Data with large dimensionality gives us opportunities, but also gives issues, for example the curse of dimensionality described by Donoho et al. (2000). Another paper that addresses issues with high dimensional data is Johnstone and Titterington (2009).

One of the ways to address high dimensionality is dimension reduction. A well known dimension reduction techniques is Principal Component Analysis (PCA). This method extracts the eigenvectors from the original data matrix and tries to explain the largest part of the variance with the least amount of vectors. Another approach that closely relates to PCA is Singular Value Decomposition(SVD). This method decomposes a matrix in three other matrices, with eigenvectors and eigenvalues. Both techniques will be discussed in more detail in the section theoretical background.

A method that uses SVD is Penalized Matrix Decomposition(PMD) from Witten et al. (2009). This technique gives us a flexible set of algorithms to derive either a single-factor or a multi-factor model. By adjusting the penalty functions on the vectors from the Singular Value Decomposition we can adjust the algorithms to give different results. One useful application is sparse PCA. This gives us Principal Components with sparsity. This means we have a large amount of zero elements in the principal components, which results in better interpretation and lower computation times. This method has been applied in different fields, but is quite unknown in the financial world. This is why we will apply PMD to estimate latent factor models and construct minimum variance portfolios. The main research question is: Does PMD give minimum variance portfolios that perform well compared to other portfolios?

The main research question is inspired by Conlon et al. (2021), who created portfolios from latent factor models using PCA and Sparse PCA. Besides this they use different machine learning methods, which are not used in this research.

We will contribute to these papers by applying PMD to reduce the dimension of out stock data from the S&P 500. This will construct new latent factor models which we will use to construct minimum variance portfolios. After that we will compare the constructed portfolios with a simple equal weights portfolio using the Sharpe-Ratio and Value-at-Risk.

The paper is structured in the following way. In section 2 we review the relevant literature for the research. In section 3 we describe methods needed to get the results. In section 4 we describe the data and in section 5 we specify the methods to our specific applications. In section 6 we discuss the results and in section 7 we provide a conclusion and give options for further research.

# 2 Literature Review

This section is a review of the literature of several subject relevant for the research. First we will discuss the literature on Penalize Matrix Decomposition, then we will look at Sparse PCA and we will finish with Sparse PCA and portfolio construction.

## 2.1 Penalized Matrix Decomposition

The paper that relates most with my proposed research is Witten et al. (2009). The paper constructs a new way to build rank-K approximations for matrices. Their goal is threefold. Firstly they aim to create interpretable factors. Secondly they show that the algorithms can be used to construct sparse Principle Components similar to the SCoT-LASS method. Thirdly they show that PMD on a cross-product matrix gives a method for penalized Canonical Correlation Analysis. For these goals they use both a simulated data set and a data set containing gene expression. The results show that all three goals are achieved and therefore PMD seems to be a useful method to create a sparse decomposition of a matrix.

One of the other papers using PMD is Zheng et al. (2011). PMD is applied here to extract metasamples from genome data. Compared to other traditional clustering methods, the paper shows that PMD can identify the features of the samples well. This shows that PMD works especially well for the DNA datasets.

## 2.2 Sparse PCA

As this thesis focuses on the PCA side of PMD, it is logical to look at recent papers about Sparse PCA. One of these papers is Zou and Xue (2018). This paper gives an overview of the different methods to construct Sparse PCA and their applications. The first two methods discussed in Zou and Xue (2018) can also be seen in Witten et al. (2009), namely SCoTLASS and SPCA. The downside of SCoTLASS is that it is hard to compute, which makes it inconvenient in practice. They also introduce iterative thresholding methods, which are similar to PMD. Different methods they mention are DSPCA and a generalized power method. Both these methods have shown promise through research. After the overview of methods the paper discusses the benefits and the downsides of sparse PCA. One important downside of normal PCA that is is not consistent in high dimension data sets, but literature shows that sparse PCA is consistent, which is a great improvement. One of the important downsides of Sparse PCA is that there is no clear procedure to set the sparse parameters in Sparse PCA. Another big discussion in the paper is the trade-off between sparsity and the variance explained.

## 2.3 Sparse PCA and portfolio construction

One of the area's where PMD hasn't been applied yet is the financial sector. However there are papers on sparse PCA in this field. Wang et al. (2013) looks at sparse PCA for high dimensional multivariate vector autoregresive time series. The paper itself doesn't give a clear conclusion of the use of sparse PCA for the Vector-Auto-Regressive (VAR) model in practice, but it shows that sparse PCA is of interest in finance.

However, we choose to follow more closely the work of Conlon et al. (2021). In this paper they use machine learning methods to build Factor-Based portfolios and to optimize them. They create a latent factor model, and they use different methods to estimate the weights for the factor model. Among the methods are PCA and Sparse PCA. We will extent on these methods and apply PMD to try and improve the factor model and resulting portfolios.

Also interesting is a paper on portfolio construction using normal PCA. Chen (2014) constructs portfolios by combining PCA and Markowitz portfolio theory. This paper however creates the portfolios differently then we do, hence we cannot apply these methods in combination with our chosen framework.

# 3 Theoretical Background

In this section we introduce important theoretical concepts for my research. In the first three subsections we explain Principal Component Analysis, Penalized Matrix Decomposition and the basis of latent factor models and their connection to portfolio theory respectively.

## 3.1 Principal Component Analysis

Principal Component Analysis (PCA) is one of the most used dimension reduction techniques given inHotelling (1933). The goal is to explain a large part of the variance of the original data in a linear combination of the original variables. Besides reducing the dimension of the original data it also helps with interpretation. A good representation of PCA is given in Johnson and Wichern (2014). For PCA we need to know the estimated covariance matrix and the original data $\mathbf{X}$, which is a $n$ x $p$ matrix. From there we can determine the eigenvalues $\lambda_i$ and their corresponding eigenvectors $\boldsymbol{e}_i$. Next we order them by eigenvalue starting with the highest eigenvalue. Now we can construct the ith Principal Component by:

$$\boldsymbol{y}_i = \mathbf{X}\boldsymbol{e}_i \tag{1}$$

In this equation $\boldsymbol{e}_i$ is a vector of length $p$ and the resulting principal component $\boldsymbol{y}_i$ is a vector of length $n$.To choose the number of principal components, we use the variance explained by the $k$ first principal components. A common choice is to retain around 80% of the total variance, as chosen by Johnson and Wichern (2014). We can calculate the percentage of the variance explained by the first $k$ first principal components by:

$$P_k = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \tag{2}$$

where $p$ is the total number of eigenvalues.

For the method in the next subsection we need the Singular Value Decomposition(SVD). This method can also be used to reduce the dimensionality of a matrix. The method uses the following decomposition:

$$\mathbf{X} = \mathbf{UDV}^T \tag{3}$$

In this equation $\mathbf{U}$ is an $n$ x $n$ matrix, $\mathbf{V}$ is a $p$ x $p$ matrix and $\mathbf{D}$ is a $n$ x $p$ diagonal matrix, where the elements are in descending order. Also the following holds for the SVD: $\mathbf{U}^T\mathbf{U} = I_n$ and $\mathbf{U}^T\mathbf{U} = I_p$. The Penalized Matrix Decomposition, which is discussed in the next subsection, will be a generalisation of the SVD by implementing additional constraints on the matrices $\mathbf{U}$ and $\mathbf{V}$.

The relation between the SVD and PCA is in the use of eigenvectors. The $\mathbf{V}$ in the SVD is matrix containing the same eigenvectors as are used in PCA.

## 3.2 Penalized Matrix Decomposition

Penalized Matrix Decomposition (PMD) is a method proposed by Witten et al. (2009). PMD is an adaptation of the Singular Value Decomposition. This gives the following optimization problem for PMD:

$$\min_{d,\boldsymbol{u},\boldsymbol{v}} \frac{1}{2}\|\mathbf{X} - d\boldsymbol{u}\boldsymbol{v}^T\|_F^2 \text{ subject to } \|\boldsymbol{u}\|_2^2 = 1, \|\boldsymbol{v}\|_2^2 = 1, P_1(\boldsymbol{u}) \leq c_1, P_2(\boldsymbol{v}) \leq c_2, d \geq 0, \tag{4}$$

In this equation we use $\|\mathbf{X}\|_F^2$, which is the squared Frobenius norm and $\|\boldsymbol{u}\|_2^2$,which is the squared $L_2$ norm. The squared Frobenius norm is the sum of squared elements of a matrix and the squared $L_2$ norm is the equivalent of the Frobenius norm but for vectors. In equation 4 $d$, $\boldsymbol{u}$ and $\boldsymbol{v}$ comprise the results of the SVD and $P_1$ and $P_2$ are convex penalty functions, which can be chosen in accordance with the data or application. The Penalty functions we will implement are lasso and fused lasso. Next we want to rewrite equation 4 to a more convenient problem. This can be done using theorem 2.1 from Witten et al. (2009). This theorem helps remove the norm in the objective. However, because of the equality constraints, the criterion is not convex, when we fix either $\boldsymbol{u}$ or $\boldsymbol{v}$ . We can solve this by relaxing the equality constraints and iteratively fix $\boldsymbol{u}$ and $\boldsymbol{v}$ as done in Witten et al. (2009). This gives the following problem:

$$\max_{\boldsymbol{u},\boldsymbol{v}} \boldsymbol{u}^T \mathbf{X} \boldsymbol{v} \text{ subject to } \|\boldsymbol{u}\|_2^2 \leq 1, \|\boldsymbol{v}\|_2^2 \leq 1, P_1(\boldsymbol{u}) \leq c_1, P_2(\boldsymbol{v}) \leq c_2. \tag{5}$$

From equation 5 an algorithm can be build for a single factor PMD model, similar to Algorithm 1 in Witten et al. (2009) :

**Algorithm 1** Computation of single-factor PMD model

1. initialize $\boldsymbol{v}$ to have $L_2$-norm 1.
2. iterate until convergence:
   (a) $\boldsymbol{u} \leftarrow \arg\max_{\boldsymbol{u}} \boldsymbol{u}^T \mathbf{X} \boldsymbol{v}$ subject to $P_1(\boldsymbol{u}) \leq c_1$ and $\|\boldsymbol{u}\|_2^2 \leq 1$.
   (b) $\boldsymbol{v} \leftarrow \arg\max_{\boldsymbol{v}} \boldsymbol{u}^T \mathbf{X} \boldsymbol{v}$ subject to $P_2(\boldsymbol{v}) \leq c_2$ and $\|\boldsymbol{v}\|_2^2 \leq 1$.
3. d $\leftarrow \boldsymbol{u}^T \mathbf{X} \boldsymbol{v}$.

We can adjust the PMD algorithm by choosing the penalty functions. Examples of pairs of penalty function are $(L_1, L_1)$, $(L_1, FL)$ or $(-, L_1)$. $L_1$ represents a penalty of the form $\Sigma_{i=1}^n |u_i|$, where $u_i$ are the elements of the vector $\boldsymbol{u}$. The $FL$ represent a fused lasso penalty of the form $\Sigma_{i=1}^n |u_i| + \lambda \Sigma_{i=2}^n |u_i - u_{i-1}|$, where $\lambda > 0$. The last set of penalty functions applies no additional restrictions on $\boldsymbol{u}$ and gives a method for sparse PCA. Besides the single factor algorithm, Witten et al. (2009) also constructs algorithm 2, a PMD algorithm for multiple factors:

**Algorithm 2** Computation of K-factor PMD model

1. Let $\mathbf{X}^1 \leftarrow X$.
2. for $k \in 1, ..., K$:
   (a) Find $\boldsymbol{u}_k, \boldsymbol{v}_k$ and $d_k$ by applying algorithm 1 (Single factor PMD) to data $\mathbf{X}^k$.
   (b) $\mathbf{X}^{k+1} \leftarrow \mathbf{X}^k - d_k \boldsymbol{u}_k \boldsymbol{v}_k^T$.

First we focus on algorithm 1 and compare this method to other sparse PCA methods on both simulated data and on gene expression data, as done in Witten et al. (2009). To extend on this paper we will apply PMD Algorithm 2 to data from the stock market. For this data we will use the stocks in the S&P500, which we will need to adjust for dividend and splits, to give the daily returns.

## 3.3 Factor-Based portfolio optimization

We apply PMD on the financial data, using methods proposed by Conlon et al. (2021). First we need to build the latent factor model. To do this we can use the general form defined in Conlon et al. (2021). First we choose $i = 1, ..., N$ assets with $t = 1, ..., T$ observations and $k = 1, ..., K$ and let $p$ be the number of predictors. The general form for latent factor models is as follows:

$$r_{i,t} = \alpha_i + \beta_i(\mathbf{X}_t\mathbf{W}) + u_{i,t} = \alpha_i + \beta_i\mathbf{F}_t + u_{i,t} \tag{6}$$

In this equation $r_{i,t}$ are the returns of asset $i$ at time $t$, the $\alpha_i$ is the intercept for asset $i$, $\beta_i$ is the estimated factor loading for asset $i$ and $u_{i,t}$ are the error terms for asset we at time t. The matrix $\mathbf{X}_t$ is a set of predictors of size $T$ x $p$ and $\mathbf{W}$ is the matrix containing the weights of size $p$ x $K$, which we will construct using PCA and Sparse PCA. The multiplication gives an estimate of the Factors $\mathbf{F}_t$, a matrix of size $T$ x $K$. After constructing the factors, we apply Ordinary Least Squares (OLS) to estimate $\alpha_i$ and $\beta_i$. From this estimated model we can calculate the error terms $u_{i,t}$.

The goal for the portfolios we construct is minimum variance, which means we want to minimize the risk. This is also the approach from Conlon et al. (2021). We first need a definition for the variance of the portfolio, which can simply be derived from equation 6:

$$\mathbf{\Sigma_r} = \mathbf{B}^T\mathbf{\Sigma_f}\mathbf{B} + \mathbf{\Sigma_u} \tag{7}$$

In this equation the sigmas represent the the covariance matrix to the corresponding indices and the $\mathbf{B}$ is a $K$ x $N$ matrix where each column contains the factor loadings $\beta_i$. To estimate the weights for a minimum-variance portfolio we need to solve the following optimization:

$$\arg\min_{\boldsymbol{\omega}} \boldsymbol{\omega}^T\hat{\mathbf{\Sigma}}_r\boldsymbol{\omega}, \text{ s,t } \boldsymbol{\omega}1_N = 1, \omega_i \geq 0, \text{for } i = 1, ...N \tag{8}$$

In this equation the $\boldsymbol{\omega}$ is the portfolio weight factor and $1_N$ is a unit vector of size N. The last constraint states that the portfolio weights can not be negative. This is used in theory by for example Conlon et al. (2021). However, in practice we can choose to remove this criterion and allow shortselling of assets.

# 4 Data

For the research we use several datasets. First we use a simulated data set with the following Data Generating Process: we create a dataset $\mathbf{X}$, which is 12x1000. This represents 12 patients with each 1000 genes. For the values in $\mathbf{X}$ we use:

1. For i $\in 1, ..., 5$ and j $\in 100, ...500, X_{i.j} \sim N(1, 1)$
2. Otherwise, $X_{i,j} \sim N(0, 1)$.

This is the same as described in appendix 3 from Witten et al. (2009). This means that patients 1 to 5 have higher values between positions 100 and 500.

For the Risk Model we use a data set comprised of the stocks in the S&P500. Not all stocks are available for the full sample period, so some stocks have been removed from the sample. This leaves us with 441 stocks with 1759 observations for each stock. For these stocks we calculate the daily log returns. First the stock prices have been adjusted for dividend and then they are transformed in daily log returns by the taking the difference of the log price on the day minus the log price on the previous day. For the sample we choose the time period 2012 to 2018 and split them in a training sample from 2012 through 2016 and a test sample from 2017 through 2018. This sample is chosen to avoid the effect of COVID on daily returns, which are highly volatile.

# 5 Methodology

We split the research into two parts. We first study the medical data, as used in Witten et al. (2009) and replicate their results. After that we focus on using the PMD to construct portfolios and compare it to other possible portfolios.

## 5.1 Simulation Example

In this part we follow Witten et al. (2009) and their simulation example. First we introduce a soft threshold function:

$$S(a, c) = sgn(a)(|a| - c)_+ \tag{9}$$

In this equation $c$ is a constant larger than 0 and the $(...)+$ indicates that the part between the brackets equals zero if it is negative and the normal value if it is positive. The sgn(a) function returns 1 if a is positive or zero and returns $-1$ if a is negative.

Next we change algorithm 1 to fit two different penalty functions for PMD: PMD(L1,L1), shown in algorithm 3, and PMD(L1,FL), shown in algorithm 4. These algorithms are identical to Algorithms 3 and 4 in Witten et al. (2009). To get this we only have to change steps 2(a) and 2(b).

**Algorithm 3** Computation of single-factor PMD model for L1,L1

1. initialize $\boldsymbol{v}$ to have $L_2$-norm 1.
2. iterate until convergence:
   (a) $\boldsymbol{u} \leftarrow \frac{S(\mathbf{X}\boldsymbol{v},\Delta_1)}{\|S(\mathbf{X}\boldsymbol{v},\Delta_1)\|_2}$, where $\Delta_1 = 0$, if this results in $\|\boldsymbol{u}\|_1 \leq c_1$, otherwise, $\Delta_1$ is chosen such that $\|\boldsymbol{u}\|_1 = c_1$, where $\Delta_1$ is a positive constant.
   (b) $\boldsymbol{v} \leftarrow \frac{S(\mathbf{X}^T\boldsymbol{u},\Delta_2)}{\|S(\mathbf{X}^T\boldsymbol{u},\Delta_2)\|_2}$, where $\Delta_2 = 0$, if this results in $\|\boldsymbol{v}\|_1 \leq c_1$, otherwise, $\Delta_1$ is chosen such that $\|\boldsymbol{v}\|_1 = c_1$, where $\Delta_2$ is a positive constant.
3. d $\leftarrow \boldsymbol{u}^T\mathbf{X}\boldsymbol{v}$.

**Algorithm 4** Computation of single-factor PMD model for L1,FL

1. initialize $\boldsymbol{v}$ to have $L_2$-norm 1.
2. iterate until convergence:
   (a) $\boldsymbol{u} \leftarrow \frac{S(\mathbf{X}\boldsymbol{v},\Delta_1)}{\|S(\mathbf{X}\boldsymbol{v},\Delta_1)\|_2}$, where $\Delta_1 = 0$, if this results in $\|\boldsymbol{u}\|_1 \leq c_1$, otherwise, $\Delta_1$ is chosen such that $\|\boldsymbol{u}\|_1 = c_1$, where $\Delta_1$ is a positive constant.
   (b) $\boldsymbol{v} \leftarrow \arg\min_v\{\frac{1}{2}\|\mathbf{X}^T\boldsymbol{u} - \boldsymbol{v}\|^2 + \lambda_1\sum_j |v_j| + \lambda_2\sum_j |v_j - v_{j-1}|\}$
3. d $\leftarrow \boldsymbol{u}^T\mathbf{X}\boldsymbol{v}$.

The selection of the tuning parameters $c_1$ and $c_2$ is done in accordance with algorithm 5 in Witten et al. (2009). As this part is not a big part of the research wechoose to not implement any additional methods for choosing these parameters.

## 5.2 Factor-Based portfolio optimization

Next we use the model in equation 6 and apply different methods to estimate the factors. The methods we apply are normal PCA, PMD(L1,L1) and PMD(L1,FL). After using these methods we construct the portfolio weights using 8. After this we analyse three different aspects: the factors and their loadings and the portfolios created. For the factors and the loadings we try to find an interpretation. From the portfolio we asses if the risk is reduced and if the returns of the portfolios are good out of sample compared to other portfolios. We can compare them to a simple 1/N portfolio and the weighted index of the S&P 500. A good measurement for risk is the Sharpe ratio. This can be found by:

$$\frac{\bar{r}_p - \bar{r}_f}{SD} \tag{10}$$

In this equation $\bar{r}_p$ is the average of the value of the out of sample portfolio returns and the $\bar{r}_f$ is the average risk-free rate. SD represents the standard deviation of the portfolios returns. Another strong measurement of risk is historical Value-at-risk(VaR). The historical VaR at 1-$\alpha$ level is:

$$-z_a\hat{\sigma}_p - \hat{r}_p \tag{11}$$

In this equation $z_a$ is the a% quantile of a standard normal distribution and its Cumulative Density Function(CDF) and $\hat{\sigma}_p$ is the average standard deviation of the portfolio.

# 6 Results

In this section we will examine the results from the two parts described in the methodology. First we will look at the simulation example and then the application of the PMD on the stock market data described in the Data section.

## 6.1 Simulation Example

We apply the algorithms 3 and 4 on the simulated data described in the data section. This creates two sets of ($\boldsymbol{u}$,$\boldsymbol{v}$) pairs. These results are shown in 1. We can compare these to the generative model and to the results of Witten et al. (2009).

In figure 1 the PMD(L1,FL) is a clear improvement over PMD(L1,L1). The $\boldsymbol{u}$ closely follows the one of the generative model closely, while $\boldsymbol{v}$ is more smooth than for PMD(L1,L1). However our results are different than those from Witten et al. (2009). These differences can be found in the fact that the simulation datasets will not be identical and that we could have used different tuning parameters, although the difference remains curious.

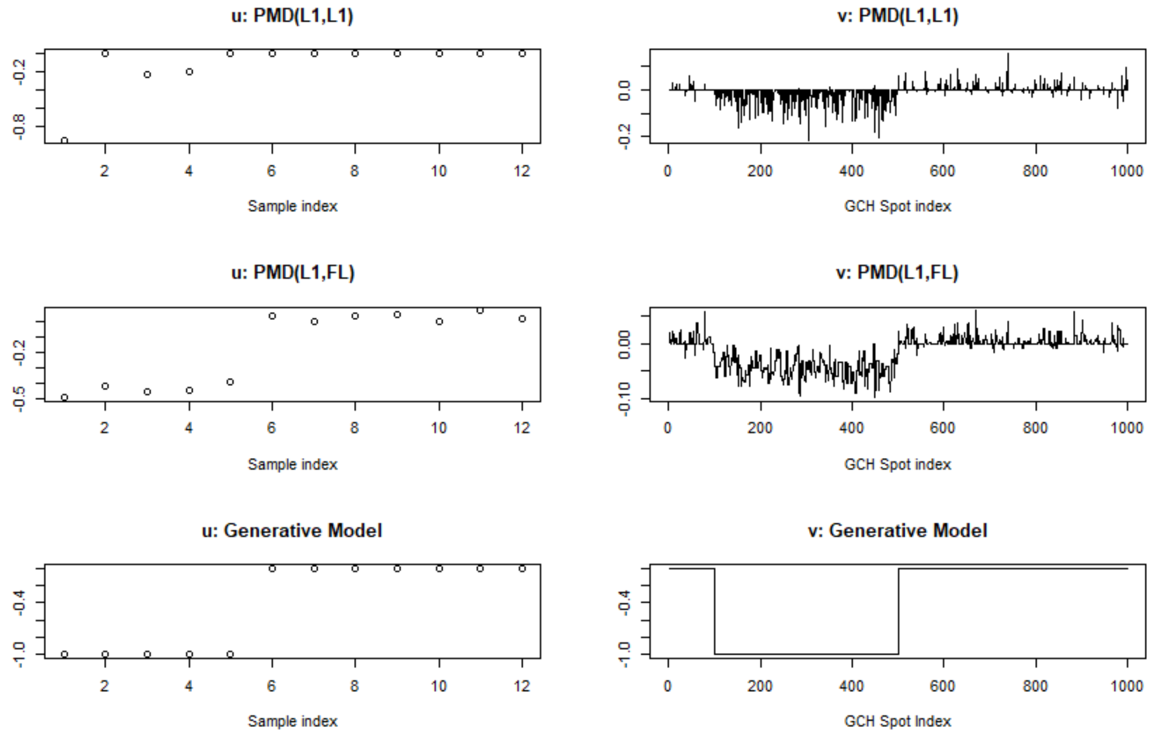


Figure 1: Reproduction of Figure 2 from Witten et al. (2009). See section 4 for the data generating process. In this figure the simulation example is given for PMD(L1,L1), PMD(L1,FL) and the original generative process.

## 6.2 Minimum Variance Portfolios

In this section we will look at the minimum variance portfolios using the statistical latent factor models. We will look at the results of the methods, the factors and the portfolios and their performance out of sample.

The first thing we look at in this section is the first Principal components. These are plotted in figure 2.

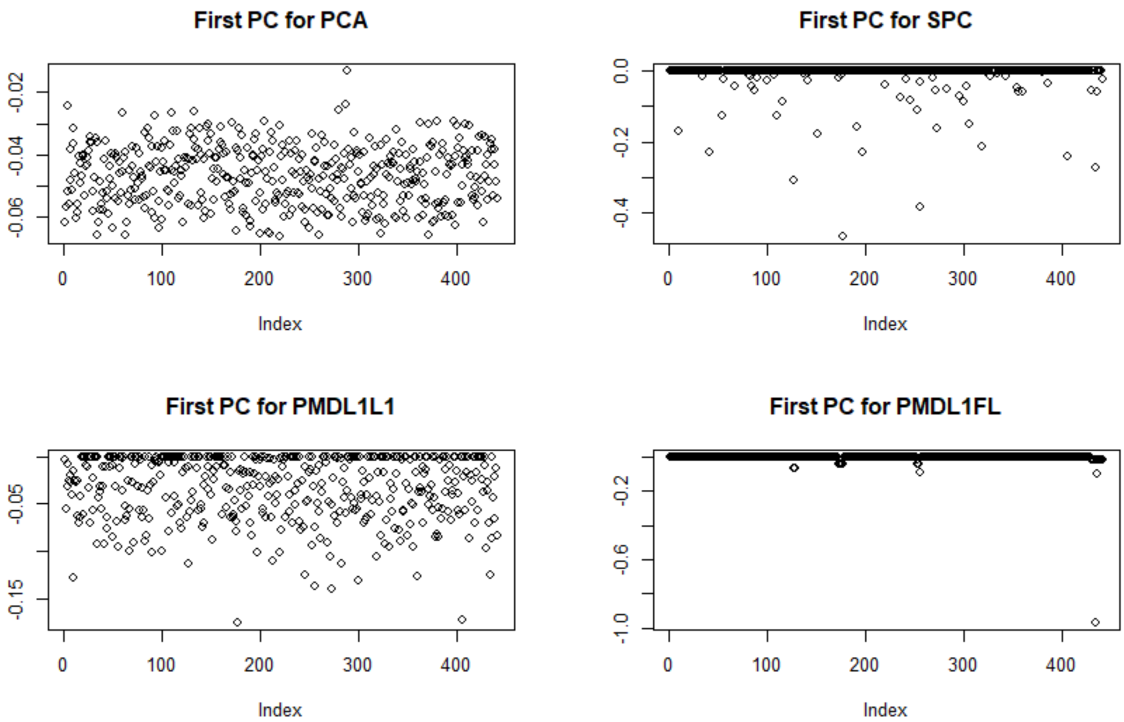


Figure 2: Plot of the first Principal Component for the different methods.

It is clear to see that there are large differences in sparsity between the different methods. This makes sense for the normal PCA, and we can attribute the difference of sparsity between the other methods to the fact that we didn't choose any restrictions on the level of sparsity besides letting the Algorithm 5 form Witten et al. (2009) decide the tuning parameter. Also the most remarkable result from this figure is the PMDL1FL graph. We can see that most values are close to zero, with the exception of one. This is the stock of The Williams Company. The results can be explained by the large fluctuations in the price of this stock. In our training sample the stock has a highest price close to 60 and a lowest price of around 15.

Next we will look at the first factors and see if we can find an interpretation. The first factors are shown in figure 3.

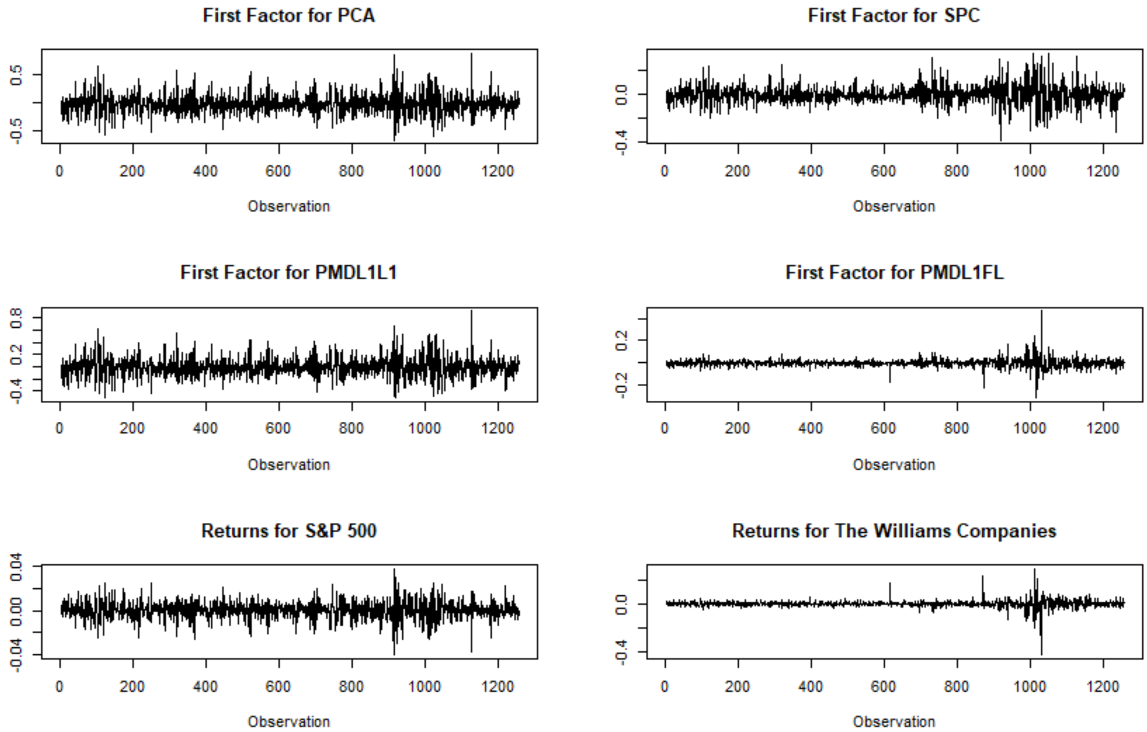


Figure 3: Plot of the first Factor for the different methods. The last plot shows the S&P 500 returns as comparison.

We can see that the different factors have their differences, but tend to have similar shapes. If we compare the factors to the returns from the S&P 500 we can see that there is a clear relation between the first factors and the returns from the S&P 500. The first factor tends to follow the inverse of the returns of the S&P 500. This can be clearly seen at the large outlier around observation 1100 in figure 3 for the PCA and PMDL1L1. For the returns we see a relatively large negative return from the S&P 500 returns, whereas most of the factors show a large peak. Also the rest of the trends follow this correlation. This is similar to the market factor from the classic Fama-French model as described in Fama and Frech (1993). This shows that the factor models created follow the literature.

Again the PMDL1FL shows a difference. But we can see that if we compare this factor to the returns for The Williams Companies we can see that the factor for the PMDL1FL follows the inverse of these returns. This is logical if we look at the first PC in figure 2.

Next we will look at the portfolio weight for the different methods. Wechose to use the number of factors that would equal a variance explained of 50%. We would like this to be higher, but this would result in a very large number of factors. This results in 128 factors. This is quite a large number of factors needed. The portfolio weights are shown in figure 4.

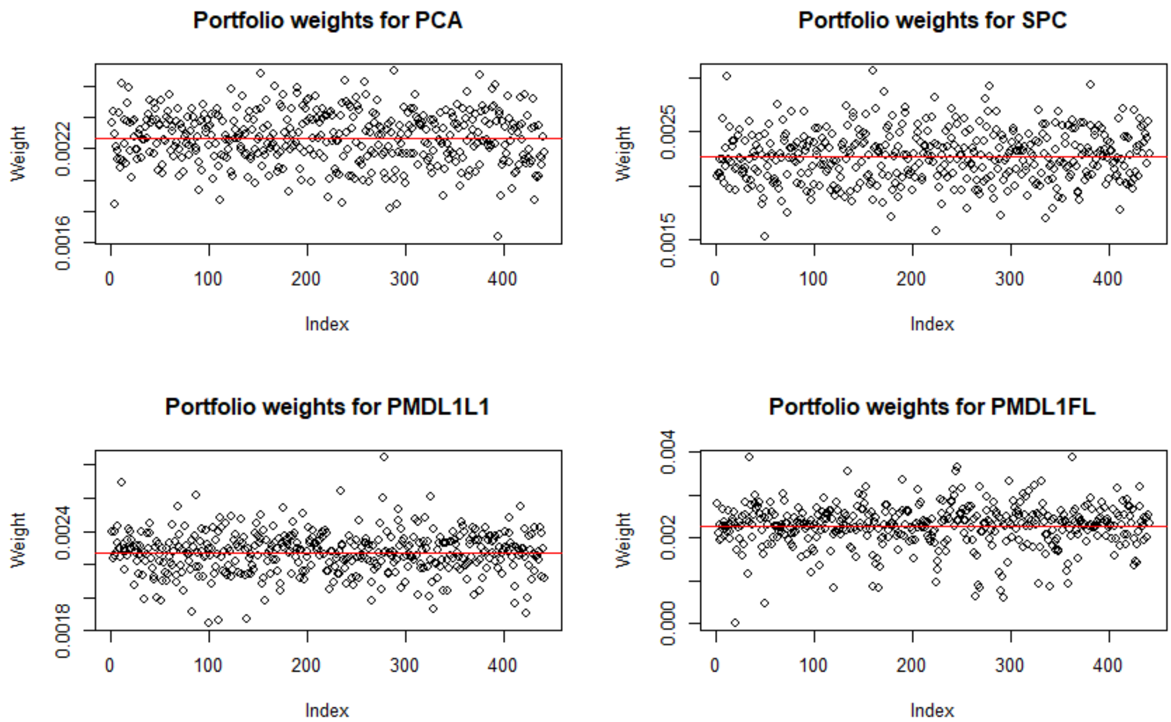


Figure 4: Plot for the portfolio weights for the different methods. The red line are the 1/N portfolio weights.

From these graphs it becomes clear that the different portfolio weights all are close to the equal weights portfolio. Interesting to note is that the weights for the portfolios using sparse methods do not result in sparse portfolio weights. This means that we can't compensate lesser returns with less transaction costs. Furthermore, there are no large outliers in the portfolio weights.

Next we examine the performance of the different portfolios. The Sparpe-Ratio and the Value-at-Risk are shown in 1 below. The results in this table give a strong indication that the methods do not improve existing portfolio construction methods. This is best shown in the Sharpe-Ratios. The fact that all the portfolios get outperformed by the equal weight portfolio(1/N) shows that the methods do not perform well, at least for the current specifications. Among the different methods PMDL1L1 performs best and is close to the equal weights portfolio. The Sharpe-ratios are all below 1. This means that the portfolios carry more risk than return according to this metric. If we look at the VaR we

Table 1: Performance metrics for the portfolios

| Methods | Sharpe-Ratio | Value-at-Risk |
|---|---|---|
| PCA | 0.866 | -0.0131 |
| SPC | 0.749 | -0.0131 |
| PMDL1L1 | 0.93 | -0.0131 |
| PMDL1FL | 0.804 | -0.0133 |
| 1/N | 0.967 | -0.0131 |

can see that almost all the values are the same. The only difference is the result from the PMDL1FL. The VaR values are all negative and this is a positive result. This means that with 95% probability the portfolios will give at least a 1% return.

# 7 Conclusion

We have applied the PMD to both simulated gene data and used it in latent factor models to construct minimum variance portfolios. For the PMD we have followed the algorithm derived in Witten et al. (2009) and to construct the portfolios we have used the factor model and minimum variance problem from Conlon et al. (2021). The combination of these papers allowed us to look at different minimum variance portfolios and the underlying factors.

As an answer to our research question we find that the portfolios constructed using PMD didn't improve on existing techniques. All the methods involving some sort of dimension reduction performed worse than the traditional equal weights portfolio in Sharpe-Ratio and were close to identical for the VaR. In further research the tuning parameters and the number of factors could be a point of interest. Also other penalty functions can be considered for the PMD.

# 8 Appendix

## 8.1 Code description

For the code description we have two parts. The first part is for the simulation results in section 6.1 and the second part is for the results from the portfolio construction in section 6.2.

For the simulation results, we import the PMA package from Witten et al. (2009). Then we setup the simulation data as explained in the data section. Then we apply the PMD and find the optimal tuning parameters. Then we plot the results and the representation of the data generating process. This gives figure 1.

For the portfolio construction we use the same package again. We first import the stocks in the S&P 500 and then remove the ones we cant use for our sample. Next we import the prices for the stocks and then transform them to daily log returns. We then adjust the sample and split it in a training sample and a test sample. After that we calculate the first components for the different methods and plot them. This gives figure 2. Then we decide what number of factors results in 50% variance explained. Then we apply the methods to extract the number of factor loadings and the calculate the factor. We can then plot figure 3. After that we estimate the factor model using OLS and calculate the residuals. Then we construct the covariance structure for the factors and the returns according to equation 7. Then we can solve the optimization problem given by equation 8. After that we can plot the portfolio weights for figure 4. The last thing we do is calculate the Sharpe-Ration and VaR.

# References

H. Chen. *Portfolio Construction Using Principle Component Analysis.* PhD thesis, Worcester Polytechnic Institute, 2014.

T. Conlon, J. Cotter, and I. Kynigakis. Machine learning and factor-based portfolio optimization. *Michael J. Brennan Irish Finance Working Paper Series Research Paper*, 21(6), 2021.

D. L. Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000.

E. F. Fama and K. R. Frech. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33:3–56, 1993.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis*. Pearson Education Limited, 2014.

I. M. Johnstone and D. M. Titterington. Statistical challenges of high-dimensional data. *Philosophical Rransactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(6):4237–4253, 2009.

Z. Wang, F. Han, and H. Liu. Sparse principal component analysis for high dimensional multivariate time series. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pages 48–56. PMLR, 2013.

D. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

C.-H. Zheng, L. Zhang, T.-Y. Ng, C. K. Shiu, and D.-S. Huang. Molecular pattern discovery based on penalized matrix decomposition. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(6):1592–1603, 2011.

H. Zou and L. Xue. A selective overview of sparse principal component analysis. *Proceedings of the IEEE*, 106(8):1311–1320, 2018.