Basic methods for data analysis

Philipp Renz

January 5, 2022

Contents

Chapter 1

Introduction

Chapter 2

Linear algebra: Basics

In data analysis we will often represent our data as matrices. Consequently many algorithms are described using the concept of matrices and their properties which could be broadly called the study of linear algebra. Here I don't want to give an axiomatic introduction but rather provide some basic results with the prerequisite that the reader already knows some basics about vectors and matrices. We will denote vectors as bold lowercase letters

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}, \tag{2.1}$$

where a_i is the ith entry of the vector $\boldsymbol{a}$.
Matrices are denoted by bold uppercase letters

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}. \tag{2.2}$$

Depending on the context it is useful to view a matrix as a concetation of either row or column vectors.

$$\boldsymbol{A} = \begin{pmatrix} — & \boldsymbol{a}_1^r & — \\ & \vdots & \\ — & \boldsymbol{a}_m^r & — \end{pmatrix} = \begin{pmatrix} | & & | \\ \boldsymbol{a}_1^c & \dots & \boldsymbol{a}_n^c \\ | & & | \end{pmatrix}. \tag{2.3}$$

We made an explicit distinction between row- and column vectors using superscripts here but these will often be left out for better readability.

2.1 Matrix multiplication

Given two matrices

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \in R^{m \times n}, \qquad \boldsymbol{B} = \begin{pmatrix} b_{11} & \dots & b_{1p} \\ \vdots & \ddots & \vdots \\ b_{n1} & \dots & b_{np} \end{pmatrix} \in R^{n \times p} \tag{2.4}$$

their product $\boldsymbol{C} = \boldsymbol{AB}$ is of shape $(m \times p)$ and it's entries are

$$c_{ik} = \sum_{j=1}^{n} a_{ij} b_{jk} \tag{2.5}$$

The resulting matrix $\boldsymbol{C}$ has a shape of $(m \times p) \leftarrow (m \times n)(n \times p)$. The inner dimension n is "eliminated" by summing over it. One can also see that the sum moves horizontally/rowwise over the entries of A and down the columns of B

Often we are interested in a special case in which one of the matrices reduces to a vector, that means it has only one column or one row. We define a column vector as a shorthand notation for a matrix $\boldsymbol{B} \in R^{n\times 1}$ with only one column and drop the column index:

$$\boldsymbol{B} = \begin{pmatrix} b_{11} \\ \vdots \\ b_{n1} \end{pmatrix} \widehat{=} \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \boldsymbol{b} \tag{2.6}$$

Implicitly when we talk of a vector b with n elements we mean a matrix $\boldsymbol{B} \in R^{n\times 1}$. When we want to talk about a matrix with only one row we write this a transposed vector

$$\boldsymbol{b^T} = (b_1 \dots b_n) \widehat{=} (b_{11} \dots b_{1n}) \tag{2.7}$$

2.1.1 Vector-vector products

Two vectors can be multiplied with each other in two different ways. The scalar or inner product is defined as

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^T \boldsymbol{b} = \sum_{i=1}^{n} a_i b_i \tag{2.8}$$

This can be seen as a multiplication of matrices with shapes $(1 \times m)(m \times 1) \to (1 \times 1)$ and results in a scalar value.
The outer product between two vectors is

$$\boldsymbol{C} = \boldsymbol{a}\boldsymbol{b}^T = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} \begin{pmatrix} b_1 & \dots & b_n \end{pmatrix} = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \dots & a_1 b_n \\ a_2 b_1 & a_2 b_2 & \dots & a_2 b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_m b_1 & a_m b_2 & \dots & a_m b_n \end{pmatrix} \tag{2.9}$$

This means that every column is a multiple of $\boldsymbol{a}$ and every row a multiple of $\boldsymbol{b}$. We see that the entry $c_{ij} = a_i b_j$.

2.1.2 Matrix-vector multiplication

The product of a matrix and vector $\boldsymbol{c} = \boldsymbol{A}\boldsymbol{b}$ again is a vector. If $\boldsymbol{A} \in R^{m\times n}$ and $\boldsymbol{b} \in R^n$ their product will be of shape $(m \times 1)$, or $\boldsymbol{c} \in R^m$.

Linear combination of columns

One can view matrix-vector product, $\boldsymbol{c} = \boldsymbol{A}\boldsymbol{b}$, as calculating a linear combination of the columns of $\boldsymbol{A}$ weighted by the entries of $\boldsymbol{b}$:

$$\boldsymbol{A}\boldsymbol{b} = \begin{pmatrix} \sum_{j=1}^{n} a_{1j} b_j \\ \sum_{j=1}^{n} a_{2j} b_j \\ \vdots \\ \sum_{j=1}^{n} a_{mj} b_j \end{pmatrix} = \begin{pmatrix} a_{11}b_1 + a_{12}b_2 + \dots + a_{1n}b_n \\ a_{21}b_1 + a_{22}b_2 + \dots + a_{2n}b_n \\ \vdots \quad\quad \vdots \quad\quad \ddots \quad\quad \vdots \\ a_{m1}b_1 + a_{m2}b_2 + \dots + a_{mn}b_n \end{pmatrix} = \begin{pmatrix} | & | & & | \\ \boldsymbol{a}_1 b_1 + \boldsymbol{a}_2 b_2 + \dots + \boldsymbol{a}_n b_n \\ | & | & & | \end{pmatrix} \tag{2.10}$$

$$= \boldsymbol{a}_1 b_1 + \boldsymbol{a}_2 b_2 + \dots + \boldsymbol{a}_n b_n = \sum_{i=1}^{n} \boldsymbol{a}_i b_i \tag{2.11}$$

One prominent example where this view is useful is that if the columns of $\boldsymbol{A}$ form a basis. Then a vector $\boldsymbol{c}$ can be written as $\boldsymbol{c} = \boldsymbol{A}\boldsymbol{c}_A$, where $\boldsymbol{c}_A$ are the coordinates of $\boldsymbol{c}$ with respect to the basis $\boldsymbol{A}$. Equivalently if the columns of $\boldsymbol{B}$ form another basis then

$$\boldsymbol{c} = \boldsymbol{A}\boldsymbol{c}_A = \boldsymbol{B}\boldsymbol{c}_B. \tag{2.12}$$

From this equation one can easily get the prescription of calculating the coordinates wrt. to $\boldsymbol{A}$ given those wrt. $\boldsymbol{B}$,

$$\boldsymbol{c}_A = \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{c}_B. \tag{2.13}$$

Dot product

Alternatively we can view a matrix vector product $\boldsymbol{Ab}$ as calculating the dot products of $\boldsymbol{b}$ with each row of $\boldsymbol{A}$,

$$\boldsymbol{Ab} = \begin{pmatrix} — & \boldsymbol{a}_1 & — \\ & \vdots & \\ — & \boldsymbol{a}_m & — \end{pmatrix} \begin{pmatrix} | \\ \boldsymbol{b} \\ | \end{pmatrix} = \begin{pmatrix} \boldsymbol{a}_1 \cdot \boldsymbol{b} \\ \vdots \\ \boldsymbol{a}_m \cdot \boldsymbol{b} \end{pmatrix} \tag{2.14}$$

2.1.3 Matrix-matrix multiplication

Now that we've looked at some special cases of matrix multiplication we can go back to the general case $\boldsymbol{C} = \boldsymbol{AB}$.
Recalling the definition of matrix multiplication

$$c_{ik} = \sum_{j=1}^{n} a_{ij} b_{jk} \tag{2.15}$$

we can look at the index k and see that the k-th column of $\boldsymbol{C}$ is formed by the matrix vector product of A and the k-th column vector of $\boldsymbol{B}$,

$$\boldsymbol{C} = \boldsymbol{AB} = \boldsymbol{A} \begin{pmatrix} | & & | \\ \boldsymbol{b}_1 & \dots & \boldsymbol{b}_p \\ | & & | \end{pmatrix} = \begin{pmatrix} | & & | \\ \boldsymbol{Ab}_1 & \dots & \boldsymbol{Ab}_p \\ | & & | \end{pmatrix} = \begin{pmatrix} | & & | \\ \boldsymbol{c}_1 & \dots & \boldsymbol{c}_p \\ | & & | \end{pmatrix}. \tag{2.16}$$

Breaking matrix-matrix multiplication down to multiple matrix vector operations allows us to view it as calculating multiple linear combinations of the columns of $\boldsymbol{A}$ or calculating the all inner products between the rows of $\boldsymbol{A}$ and the columns of $\boldsymbol{B}$,

$$\boldsymbol{C} = \begin{pmatrix} — & \boldsymbol{a}_1 & — \\ & \vdots & \\ — & \boldsymbol{a}_m & — \end{pmatrix} \begin{pmatrix} | & & | \\ \boldsymbol{b}_1 & \dots & \boldsymbol{b}_p \\ | & & | \end{pmatrix} = \begin{pmatrix} \boldsymbol{a}_1 \cdot \boldsymbol{b}_1 & \boldsymbol{a}_1 \cdot \boldsymbol{b}_2 & \dots & \boldsymbol{a}_1 \cdot \boldsymbol{b}_p \\ \boldsymbol{a}_2 \cdot \boldsymbol{b}_1 & \boldsymbol{a}_2 \cdot \boldsymbol{b}_2 & \dots & \boldsymbol{a}_2 \cdot \boldsymbol{b}_p \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_m \cdot \boldsymbol{b}_1 & \boldsymbol{a}_m \cdot \boldsymbol{b}_2 & \dots & \boldsymbol{a}_m \cdot \boldsymbol{b}_p \end{pmatrix}. \tag{2.17}$$

From this we can read that $c_{ik} = \boldsymbol{a}_i \boldsymbol{b}_j$.

Sum of outer products

Another view that is often useful is to write a matrix-matrix multiplication as a sum of outer products. Looking again at our basic definition

$$c_{ik} = \sum_{j=1}^{n} a_{ij} b_{jk} = a_{i1} b_{1k} + a_{i2} b_{2k} + a_{i3} b_{3k} \tag{2.18}$$

we can see that for example the term $a_{i2} b_{2k}$ is the ik-th entry of the dot-product between the second column vector of $\boldsymbol{A}$ and the second row vector of $\boldsymbol{B}$. Thus we can deduce that the matrix $\boldsymbol{C}$ is a sum of outer-products,

$$\boldsymbol{C} = \boldsymbol{AB} = \begin{pmatrix} | & & | \\ \boldsymbol{a}_1 & \dots & \boldsymbol{a}_n \\ | & & | \end{pmatrix} \begin{pmatrix} — & \boldsymbol{b}_1 & — \\ & \vdots & \\ — & \boldsymbol{b}_n & — \end{pmatrix} = \sum_{i=1}^{n} \boldsymbol{a}_i \boldsymbol{b}_i^T. \tag{2.19}$$

Chapter 3

Principal component analysis

Motivating example There is a park and we are given the locations of the trees in it given two numbers. It appears that they are approximately planted in a line with slight deviations. However we would like to find a single direction which tells us the most about the location of tree. In the figure it is evident that if we can get a very good estimate of where trees are, by knowing how far to go along the red line. The (blue) distance orthogonal to it is of less importance. If we encoded each point by it's red and blue distance we would find that the red ones vary a lot more than the blue ones. Thus we will define the most important direction as the one along which the data has the highest variance.

3.1 Variance maximization

Given a vector $\boldsymbol{v}$ with unit length $||\boldsymbol{v}|| = 1$ we can calculate the coordinate wrt. to it via a dot product,

$$y_i = \boldsymbol{x}_i \cdot \boldsymbol{v}, \tag{3.1}$$

yielding a scalar for each data-point. The variance of the values $\{y_1, \ldots, y_n\}$ is

$$\frac{1}{n}\sum_{i=1}^{n} y_i^2 - \bar{y}^2 \tag{3.2}$$

$$= \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x}_i\boldsymbol{v})^2 - \left(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i\boldsymbol{v}\right)^2 = \qquad \text{def. of } y_i \tag{3.3}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m} x_{ij}v_j x_{ik}v_k - \boldsymbol{v}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{v} = \qquad \text{write dot product as sum} \tag{3.4}$$

$$= \frac{1}{n}\sum_{j=1}^{m}\sum_{k=1}^{m} v_j \sum_{i=1}^{n} x_{ji}^T x_{ik} v_k - \boldsymbol{v}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{v} = \qquad \text{rearrange} \tag{3.5}$$

$$= \frac{1}{n}\sum_{j=1}^{m}\sum_{k=1}^{m} v_j [\boldsymbol{X}^T\boldsymbol{X}]_{ik} v_k - \boldsymbol{v}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{v} = \qquad \text{replace sum by m.m.} \tag{3.6}$$

$$= \frac{1}{n}\sum_{j=1}^{m}\sum_{k=1}^{m} v_j [\boldsymbol{X}^T\boldsymbol{X}]_{ik} v_k - \boldsymbol{v}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{v} = \qquad \text{replace sum by m.m.} \tag{3.7}$$

$$= \frac{1}{n}\boldsymbol{v}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{v} - \boldsymbol{v}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{v} = \qquad \text{replace sums by m.m.} \tag{3.8}$$

$$= \boldsymbol{v}^T\left(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{\mu}^T\right)\boldsymbol{v} = \qquad \text{rearrange} \tag{3.9}$$

$$= \boldsymbol{v}^T\boldsymbol{C}\boldsymbol{v} \qquad \text{def. of covariance} \tag{3.10}$$

Since $\boldsymbol{C}$ is real and symmetric we can express it using it's eigenvalues and eigenvectors,

$$\boldsymbol{C} = \sum_{j=1}^{m} \lambda_j \boldsymbol{w}_j \boldsymbol{w}_j^T. \tag{3.11}$$

Since $(w_1, \dots, w_m)$ form an orthonormal basis we can also write $\boldsymbol{v}$ as linear combination of them,

$$\boldsymbol{v} = \sum_{i=1}^{m} \alpha_i \boldsymbol{w}_i. \tag{3.12}$$

Thus we shift problem of finding the entries of $\boldsymbol{v}$ to finding the α_i that maximize the variance. We can write our objective as

$$\begin{aligned}
\boldsymbol{v}^T \boldsymbol{C} \boldsymbol{v} &= \sum_{i=1}^{m} \alpha_i \boldsymbol{w}_i^T \sum_{j=1}^{m} \boldsymbol{w}_j \boldsymbol{w}_j^T \lambda_j \sum_{k=1}^{m} \alpha_k \boldsymbol{w}_k^T && (3.13)\\
&= \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} \alpha_i \alpha_k \lambda_j \boldsymbol{w}_i^T \boldsymbol{w}_j \boldsymbol{w}_j^T \boldsymbol{w}_k && (3.14)\\
&= \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} \alpha_i \alpha_k \lambda_j \delta_{ij} \delta_{jk} && (3.15)\\
&= \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j \lambda_j \delta_{ij} \qquad \text{sum over } k && (3.16)\\
&= \sum_{i=1}^{m} \alpha_i \alpha_i \lambda_i \qquad \text{sum over } j && (3.17)\\
&= \sum_{i=1}^{m} \alpha_i^2 \lambda_i. && (3.18)
\end{aligned}$$

Since $||\boldsymbol{v}|| = 1$ we know that $\sum_{i=1}^{m} \alpha_i^2 = 1$. The above result gives us a weighted average which is maximal when $\alpha_1 = 1$ since λ_1 is (one of) the largest eigenvalue(s). Thus the direction in which the variance of our data is maximal is just the direction of the eigenvector with the largest eigenvalue.
We now want to find the second most important direction. For this to make sense we need some additional requirement for the new direction, otherwise we could get the same one again. Since we now want to find a second vector we rename $\boldsymbol{v}$ to $\boldsymbol{v_1}$ and are looking for a vector $\boldsymbol{v_2}$ that maximizes $\boldsymbol{v}_2^T \boldsymbol{C} \boldsymbol{v}_2$, with the constraint that it has to be orthognal to $\boldsymbol{v}_1$, that is $\boldsymbol{v}_1 \cdot \boldsymbol{v}_2 = 0$. If we write the second vector in terms of the eigenvectors of $\boldsymbol{C}$,

$$\boldsymbol{v}_2 = \sum_{i=1}^{m} \beta_i \boldsymbol{w}_i, \tag{3.19}$$

then

$$\boldsymbol{v}_1^T \boldsymbol{v}_2 = \sum_{i=1}^{m} \beta_i \boldsymbol{w}_i^T \boldsymbol{w}_1 = \sum_{i=1}^{m} \beta_i \delta_{1i} = \beta_1 = 0 \tag{3.20}$$

We also have that

$$\boldsymbol{v}_2^T \boldsymbol{C} \boldsymbol{v}_2 = \sum_{i=1}^{m} \lambda_i \beta_i^2, \tag{3.21}$$

and similar to the reasoning above to maximize this under the orthogonality constraint we have to set $\beta_2 = 1$ and thus $\boldsymbol{v}_2 = \boldsymbol{w}_2$. Problem: Why do we want the directions to be orthogonal. Somehow clear but I don't like the reasoning. Is this easier to do with lossy compression?

3.2 PCA as lossy compression

Above we tried to find directions with maximal variance. Next we want to look at PCA as lossy compression. In general we could try to find a simple rule that allows us to reconstruct our data. For example if we have bivariate data with the relation $y_i = x_i^2$ we only really need to store the x_i-values for each datapoint as we can reconstruct the y-values from it. But if we allow to general rules we run into the problem of overfitting. Thus we will allow our compression method to use a linear transformations.

We now look at what happens when we leave some of the vectors out of the picture and only include l components to encode our data.

$$\boldsymbol{x_i} \approx \boldsymbol{\mu} + \sum_{j=1}^{l} \alpha_{ij} \boldsymbol{v}_j \tag{3.22}$$

We can calculate the mean squared error of this approximation over the whole dataset.

$$\begin{aligned}
\text{MSE} =& \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{x}_i - \boldsymbol{\mu} + \sum_{j=1}^{l} \alpha_{ij} \boldsymbol{v_j} \right)^2 && (3.23)\\
=& \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{\mu} + \sum_{j=1}^{m} \alpha_{ij} \boldsymbol{v_j} - \boldsymbol{\mu} - \sum_{j=1}^{l} \alpha_{ij} \boldsymbol{v_j} \right)^2 && (3.24)\\
=& \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=l+1}^{m} \alpha_{ij} \boldsymbol{v_j} \right)^2 && (3.25)\\
=& \frac{1}{n} \sum_{i=1}^{n} \sum_{j=l+1}^{m} \alpha_{ij}^2 && (3.26)\\
=& \frac{1}{n} \sum_{i=1}^{n} \sum_{j=l+1}^{m} \left((\boldsymbol{x}_i - \boldsymbol{\mu})^T \boldsymbol{v}_j \right)^2 && (3.27)\\
=& \frac{1}{n} \sum_{i=1}^{n} \sum_{j=l+1}^{m} \left((\boldsymbol{x}_i - \boldsymbol{\mu})^T \boldsymbol{v}_j \right) \left((\boldsymbol{x}_i - \boldsymbol{\mu})^T \boldsymbol{v}_j \right) && (3.28)\\
=& \frac{1}{n} \sum_{i=1}^{n} \sum_{j=l+1}^{m} \sum_{a=1}^{m} (x_{ia} - \mu_a) v_{ja} \sum_{b=1}^{m} (x_{ib} - \mu_b) v_{jb} && (3.29)\\
=& \sum_{j=l+1}^{m} \sum_{a=1}^{m} \sum_{b=1}^{m} v_{ja} \left(\frac{1}{n} \sum_{i=1}^{n} (x_{ia} - \mu_a)(x_{ib} - \mu_b) \right) v_{jb} && (3.30)\\
=& \sum_{j=l+1}^{m} \sum_{a=1}^{m} \sum_{b=1}^{m} v_{ja} C_{ab} v_{jb} && (3.31)\\
=& \sum_{j=l+1}^{m} \boldsymbol{v}_j^T \boldsymbol{C} \boldsymbol{v}_j && (3.32)\\
=& \sum_{j=l+1}^{m} \boldsymbol{v}_j^T \lambda_j \boldsymbol{v}_j && (3.33)\\
=& \sum_{j=l+1}^{m} \boldsymbol{v}_j^T \boldsymbol{v}_j \lambda_j && (3.34)\\
=& \sum_{j=l+1}^{m} \lambda_j && (3.35)\\
& && (3.36)
\end{aligned}$$

This means that the mean squared error is given by the sum of the eigenvalues of eigenvectors not used in the approximation. If we want to have low reconstruction error using only a few components we should use the ones with high eigenvalues.

Chapter 4

Linear algebra: Singular value decomposition

The singular value decomposition is a useful way to factorize a matrix. Any matrix $\boldsymbol{M} \in R^{m\times n}$ can be written as a product of three matrices

$$\underset{(m\times n)}{\boldsymbol{M}} = \underset{(m\times m)}{\boldsymbol{U}}\underset{(m\times n)}{\boldsymbol{\Sigma}}\underset{(n\times n)}{\boldsymbol{V}^T} \tag{4.1}$$

$$, \tag{4.2}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal

$$\boldsymbol{U}\boldsymbol{U}^T = \mathbf{1}_{m\times m} \qquad\qquad \boldsymbol{V}\boldsymbol{V}^T = \mathbf{1}_{n\times n}, \tag{4.3}$$

and $\boldsymbol{\Sigma}$ is diagonal.
Without loss of generality we can assume that $m \geq n$. This would only result in empty columns instead of empty rows of $\boldsymbol{\Sigma}$. The shapes look approximately like this (5×3) in the example.

$$\begin{bmatrix} \\ \boldsymbol{M} \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \boldsymbol{U} \\ \\ \end{bmatrix}_{m\times m} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & & \end{bmatrix}_{m\times n} \begin{bmatrix} \\ \boldsymbol{V}^T \\ \\ \end{bmatrix}_{n\times n} \tag{4.4}$$

Proof of existence We will prove that such a representation always exists. We know that $\boldsymbol{M}^T\boldsymbol{M}$ has an orthogonal eigenbasis as it is symmetric, and non-negative eigenvalues as it is positive-definite,

$$\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{v}_i = \alpha_i\boldsymbol{v}_i, \qquad\qquad \alpha_i \geq 0, \qquad\qquad \boldsymbol{v}_i \cdot \boldsymbol{v}_j = \delta_{ij}. \tag{4.5}$$

Next we define vectors

$$\boldsymbol{u}_i = \frac{\boldsymbol{M}\boldsymbol{v}_i}{\alpha_i} \tag{4.6}$$

and observe that they are also pairwise orthogonal

$$\boldsymbol{u}_i\boldsymbol{u}_j = \frac{\boldsymbol{v}_i^T\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{v}_i}{\sqrt{\alpha_i\alpha_j}} = \frac{\boldsymbol{v}_i^T\boldsymbol{v}_i\alpha_i}{\sqrt{\alpha_i\alpha_j}} = \frac{\delta_{ij}\alpha_i}{\sqrt{\alpha_i\alpha_j}} = \frac{\delta_{ij}\alpha_i}{\sqrt{\alpha_i\alpha_i}} = \delta_{ij} \tag{4.7}$$

We also get that the $\boldsymbol{u}_i$ are eigenvectors of $\boldsymbol{M}\boldsymbol{M}^T$,

$$(\boldsymbol{M}\boldsymbol{M})^T\boldsymbol{u}_i = (\boldsymbol{M}\boldsymbol{M})^T\boldsymbol{M}\boldsymbol{v}_i\frac{1}{\sqrt{\alpha_i}} = \boldsymbol{M}(\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{v}_i\frac{1}{\sqrt{\alpha_i}} = \alpha_i\boldsymbol{M}\boldsymbol{v}_i\frac{1}{\sqrt{\alpha_i}} = \alpha_i\boldsymbol{u}_i \tag{4.8}$$

Next we assume that$\boldsymbol{M}^T\boldsymbol{M}$ has l non-zero eigenvalues. We take the respective eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_l$ and compute the corresponding $\boldsymbol{u}_i$ simultaneously, while using $\sigma_i = \alpha_i^{-\frac{1}{2}}$,

$$\begin{pmatrix} | & & | \\ \boldsymbol{u}_1 & \ldots & \boldsymbol{u}_l \\ | & & | \end{pmatrix} = \boldsymbol{M} \begin{pmatrix} | & & | \\ \boldsymbol{v}_1 & \ldots & \boldsymbol{v}_l \\ | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_l \end{pmatrix} \tag{4.9}$$

$$\begin{pmatrix} | & & | \\ \boldsymbol{u}_1 & \ldots & \boldsymbol{u}_l \\ | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_l \end{pmatrix} = \boldsymbol{M} \begin{pmatrix} | & & | \\ \boldsymbol{v}_1 & \ldots & \boldsymbol{v}_l \\ | & & | \end{pmatrix} \tag{4.10}$$

$$\begin{pmatrix} | & & | \\ \boldsymbol{u}_1 & \ldots & \boldsymbol{u}_l \\ | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_l \end{pmatrix} \begin{pmatrix} — & \boldsymbol{v}_1 & — \\ & \vdots & \\ — & \boldsymbol{v}_l & — \end{pmatrix} = \boldsymbol{M} \begin{pmatrix} | & & | \\ \boldsymbol{v}_1 & \ldots & \boldsymbol{v}_l \\ | & & | \end{pmatrix} \begin{pmatrix} — & \boldsymbol{v}_1 & — \\ & \vdots & \\ — & \boldsymbol{v}_l & — \end{pmatrix} \tag{4.11}$$

To get to our desired factorization and isolate $\boldsymbol{M}$ we would like to get somehow get rid of the last to matrices on the right. However while $\boldsymbol{V}^T\boldsymbol{V} = \mathbf{1}_{l\times l}$, $\boldsymbol{V}^T\boldsymbol{V} = \mathbf{1}_{n\times n}$ does not hold in general, and we can't easily cancel it out. We know however, that

$$\boldsymbol{v}_i^T\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{v_i} = \boldsymbol{v}_i^T\boldsymbol{v}_i\alpha_i \tag{4.12}$$

$$\alpha_i = 0 \;\Rightarrow\; \boldsymbol{M}\boldsymbol{v}_i = 0 \tag{4.13}$$

This allows us to do the following, where the first equality holds because we use the full set of eigen-

vectors n eigenvectors,

$$\boldsymbol{M} = \boldsymbol{M}\begin{pmatrix} | & & | & | & & | \\ \boldsymbol{v}_1 & \dots & \boldsymbol{v}_l & \boldsymbol{v}_{l+1} & \dots & \boldsymbol{v}_n \\ | & & | & | & & | \end{pmatrix}\begin{pmatrix} — & \boldsymbol{v}_1 & — \\ & \vdots & \\ — & \boldsymbol{v}_l & — \\ — & \boldsymbol{v}_l + 1 & — \\ & \vdots & \\ — & \boldsymbol{v}_n & — \end{pmatrix} \tag{4.14}$$

$$= \begin{pmatrix} | & & | & | & & | \\ \boldsymbol{M}\boldsymbol{v}_1 & \dots & \boldsymbol{M}\boldsymbol{v}_l & \boldsymbol{M}\boldsymbol{v}_{l+1} & \dots & \boldsymbol{M}\boldsymbol{v}_n \\ | & & | & | & & | \end{pmatrix}\begin{pmatrix} — & \boldsymbol{v}_1 & — \\ & \vdots & \\ — & \boldsymbol{v}_l & — \\ — & \boldsymbol{v}_l + 1 & — \\ & \vdots & \\ — & \boldsymbol{v}_n & — \end{pmatrix} \tag{4.15}$$

$$= \begin{pmatrix} | & & | & | & & | \\ \boldsymbol{M}\boldsymbol{v}_1 & \dots & \boldsymbol{M}\boldsymbol{v}_l & \boldsymbol{0} & \dots & \boldsymbol{0} \\ | & & | & | & & | \end{pmatrix}\begin{pmatrix} — & \boldsymbol{v}_1 & — \\ & \vdots & \\ — & \boldsymbol{v}_l & — \\ — & \boldsymbol{v}_l + 1 & — \\ & \vdots & \\ — & \boldsymbol{v}_n & — \end{pmatrix} \tag{4.16}$$

$$= \boldsymbol{M}\begin{pmatrix} | & & | & | & & | \\ \boldsymbol{v}_1 & \dots & \boldsymbol{v}_l & \boldsymbol{0} & \dots & \boldsymbol{0} \\ | & & | & | & & | \end{pmatrix}\begin{pmatrix} — & \boldsymbol{v}_1 & — \\ & \vdots & \\ — & \boldsymbol{v}_l & — \\ — & \boldsymbol{v}_l + 1 & — \\ & \vdots & \\ — & \boldsymbol{v}_n & — \end{pmatrix} \tag{4.17}$$

$$= \boldsymbol{M}\sum_{i=1}^{l} \boldsymbol{v}_i\boldsymbol{v}_i^T + \sum_{i=l+1}^{n} \boldsymbol{0}\ \boldsymbol{v}_i^T = \boldsymbol{M}\sum_{i=1}^{l} \boldsymbol{v}_i\boldsymbol{v}_i^T \tag{4.18}$$

$$= \boldsymbol{M}\begin{pmatrix} | & & | \\ \boldsymbol{v}_1 & \dots & \boldsymbol{v}_l \\ | & & | \end{pmatrix}\begin{pmatrix} — & \boldsymbol{v}_1 & — \\ & \vdots & \\ — & \boldsymbol{v}_l & — \end{pmatrix}. \tag{4.19}$$

This is just the rhs. of (**??**) and we get that

$$\boldsymbol{M} = \begin{pmatrix} | & & | \\ \boldsymbol{u}_1 & \dots & \boldsymbol{u}_l \\ | & & | \end{pmatrix}\begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_l \end{pmatrix}\begin{pmatrix} — & \boldsymbol{v}_1 & — \\ & \vdots & \\ — & \boldsymbol{v}_l & — \end{pmatrix} \tag{4.20}$$

This is the reduced singular value decomposition of matrix.