1

Mathematical Principles Related to Modern System Analysis

Summary

In the mathematical field of numerical analysis, model order reduction is the key to processing measured data. This also enables us to interpolate and extrapolate measured data. The philosophy of model order reduction is outlined in this chapter along with the concepts of total least squares and singular value decomposition.

1.1 Introduction

In mathematical physics many problems are characterized by a second order partial differential equation for a function as

$$Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + F = f(x, y), \tag{1.1}$$

and $u(x, y)$ is the function to be solved for a given excitation $f(x, y)$, where

$$\begin{aligned} u_{xx} &= \partial^2 u/\partial x^2; \\ u_{xy} &= \partial^2 u/(\partial x\, \partial y); \\ u_{yy} &= \partial^2 u/\partial y^2; \\ u_x &= \partial u/\partial x; \\ u_y &= \partial u/\partial y \end{aligned} \tag{1.2}$$

When $B^2 - AC < 0$ and assuming $u_{xy} = u_{yx}$, then (1.1) is called an elliptic partial differential equation. These classes of problems arise in the solution of boundary value problems. In this case, the solution $u(x, y)$ is known only over a boundary {or equivalently a contour $B(x, y)$} and the goal is to continue the given solution $u(x, y)$ from the boundary to the entire region of the real plane $\mathfrak{R}(x, y)$.

Modern Characterization of Electromagnetic Systems and Its Associated Metrology, First Edition.
Tapan K. Sarkar, Magdalena Salazar-Palma, Ming Da Zhu, and Heng Chen.

When $B^2 - AC = 0$ we obtain a parabolic partial differential equation for (1.1), which arises in the solution of the diffusion equation or an acoustic propagation in the ocean. Such applications are characterized by the term initial value problems. The solution is given for the initial condition $u(x, y = 0)$ and the goal is to find the solution $u(x, y)$ for all values of x and y.

Finally when $B^2 - AC > 0$, we obtain a hyperbolic partial differential equation. This type of equation arises from the solution of the wave equation. The characteristic of the wave equation is that if a disturbance is made in the initial data, then not every point of space feels the disturbance at once. The disturbance has a finite propagation speed. This feature makes it distinct from the elliptic and parabolic partial differential equations when a disturbance of the initial data is felt at once by all points in the domain. Even though these equations have significantly different mathematical properties, the solution methodology, just like for every numerical method in solution of an operator equation, is essentially the same, by exploiting the principle of analytic continuation.

The solution u of these equations is made in a straight forward fashion by assuming: it to be of the form

$$u(x, y) = \sum_i \alpha_i \phi_i(x, y), \tag{1.3}$$

where $\phi_i(x, y)$ are some known basis functions; and the final solution is to be composed of these functions multiplied by some constants α_i which are the unknowns to be determined using the specific given boundary conditions. The solution procedure then translates the solution of a functional equation to the solution of a matrix equation, the solution of these unknown constants is much easier to address. The methodology starts by substituting (1.3) into (1.1) and then solving for the unknown coefficients α_i from the boundary conditions for the problem if the equations are in the differential form or by integrating if it is an integral equation. Then once the unknown coefficients α_i are determined, the general solution for the problem can be obtained using (1.3).

A question that is now raised is: what is the optimum way to choose the known basis functions ϕ_i as the quality of the final solution depends on the choice of ϕ_i? It is well known in the numerical community that the best choices of the basis functions are the eigenfunctions of the operator that characterizes the system. Since in most examples one is dealing with a real life system, then the operators, in general, are linear time invariant (LTI) and have a bounded input and bounded output (BIBO) response resulting in a second-order differential equation, which is the case for Maxwell's equations. In the general case, the eigenfunctions of these operators are the complex exponentials, and in the transformed domain, they form a ratio of two rational polynomials. Therefore, our goal is to fit the given data for a LTI system either by a sum of complex

exponentials or in the transformed domain approximate it by a ratio of polynomials. Next, it is illustrated how the eigenfunctions are used through a bias-variance tradeoff in reduced rank modelling [1, 2].

1.2 Reduced-Rank Modelling: Bias Versus Variance Tradeoff

An important problem in statistical processing of waveforms is that of feature selection, which refers to a transformation whereby a data space is transformed into a feature space that, in theory, has exactly the same dimensions as that of the original space [2]. However, in practical problems, it may be desirable and often necessary to design a transformation in such a way that the data vector can be represented by a reduced number of "effective" features and yet retain most of the intrinsic information content of the input data. In other words, the data vector undergoes a dimensionality reduction [1, 2]. Here, the same principle is applied by attempting to fit an infinite-dimensional space given by (1.3) to a finite-dimensional space of dimension p.

An important problem in this estimation of the proper rank is very important. First if the rank is underestimated then a unique solution is not possible. If on the other hand the estimated rank is too large the system equations involved in the parameter estimation problem can become very ill-conditioned, leading to inaccurate or completely erroneous results if straight forward LU-decomposition is used to solve for the parameters. Since, it is rarely a "crisp" number that evolves from the solution procedure determining the proper rank requires some analysis of the data and its effective noise level. An approach that uses eigenvalue analysis and singular value decomposition for estimating the effective rank of given data is outlined here.

As an example, consider an M-dimensional data vector $u(n)$ representing a particular realization of a wide-sense stationary process. (Stationarity refers to time invariance of some, or, all of the statistics of a random process, such as mean, autocorrelation, nth-order distribution. A random process $X(t)$ [or $X(n)$] is said to be strict sense stationary (SSS) if all its finite order distributions are time invariant, i.e., the joint cumulative density functions (cdfs), or probability density functions (pdfs), of $X(t_1), X(t_2), \ldots, X(t_k)$ and $X(t_1+\tau), X(t_2+\tau), \ldots, X(t_k+\tau)$ are the same for all k, all $t_1, t_2, \ldots, t_k$, and all time shifts τ. So for a SSS process, the first-order distribution is independent of t, and the second-order distribution — the distribution of any two samples $X(t_1)$ and $X(t_2)$ — depends only on $\tau = t_2 - t_1$ To see this, note that from the definition of stationarity, for any t, the joint distribution of $X(t_1)$ and $X(t_2)$ is the same as the joint distribution of $X\{t_+ + (t - t_1)\} = X(t)$ and $X\{t_2 + (t - t_1)\} = X\{t + (t_2 - t)\}$. An independent and identically distributed (IID) random processes are SSS. A random walk and

Poisson processes are not SSS. The Gauss-Markov process (as we defined it) is not SSS. However, if we set X_1 to the steady state distribution of X_n, it becomes SSS. A random process $X(t)$ is said to be wide-sense stationary (WSS) if its mean, i.e., $\varepsilon\,(X(t)) = \mu$, is independent of t, and its autocorrelation functions $RX(t_1, t_2)$ is a function only of the time difference $t_2 - t_1$ and are time invariant. Also $\varepsilon\,[X(t)^2] < \infty$ (technical condition) is necessary, where ε represents the expected value in a statistical sense. Since $RX(t_1, t_2) = RX(t_2, t_1)$, for any wide sense stationary process $X(t)$, $RX(t_1, t_2)$ is a function only of $|t_2 - t_1|$. Clearly a SSS implies a WSS. The converse is not necessarily true. The necessary and sufficient conditions for a function to be an autocorrelation function for a WSS process is that it be real, even, and nonnegative definite By nonnegative definite we mean that for any n, any $t_1, t_2, \ldots, t_n$ and any real vector $a = (a_1, \ldots, a_n)$, and $X(n)$; $a_i a_j R(t_i - t_j) \geq 0$. The power spectral density (psd) $SX(f)$ of a WSS random process $X(t)$ is the Fourier transform of $RX(\tau)$, i.e., $SX(f) = \Im\{RX(\tau)\} = \int_{-\infty}^{\infty} RX(\tau)\ \exp\ (-j\,2\,\pi\,\tau)\ d\tau$. For a discrete time process X_n, the power spectral density is the discrete-time Fourier transform (DTFT) of the sequence $RX(n)$: $SX(f) = \sum_{n=-\infty}^{\infty} RX(n)\ \exp\ (-j\,2\,\pi\,nf)$. Therefore $RX(\tau)$ (or $RX(n)$) can be recovered from $SX(f)$ by taking the inverse Fourier transform or inverse DTFT.

In summary, WSS is a less restrictive stationary process and uses a somewhat weaker type of stationarity. It is based on requiring the mean to be a constant in time and the covariance sequence to depend only on the separation in time between the two samples. The final goal in model order reduction of a WSS is to transform the M-dimensional vector to a p-dimensional vector, where $p < M$. This transformation is carried out using the Karhunen-Loeve expansion [2]. The data vector is expanded in terms of q_i, the eigenvectors of the correlation matrix $[R]$, defined by

$$[R] = \varepsilon\left[u(n)u^H(n)\right] \tag{1.4}$$

and the superscript H represents the conjugate transpose of $u(n)$. Therefore, one obtains

$$u(n) = \sum_{i=1}^{M} c_i(n)q_i \tag{1.5}$$

so that

$$[R]\,q_i = \lambda_i q_i, \tag{1.6}$$

where $\{\lambda_i\}$ are the eigenvalues of the correlation matrix, $\{q_i\}$ represent the eigenvectors of the matrix R, and $\{c_i(n)\}$ are the coefficients defined by

$$c_i(n) = q_i^H u(n)\ \text{ for }\ i = 1, 2, \ldots, M. \tag{1.7}$$

To obtain a reduced rank approximation $\hat{u}(n)$ of $u(n)$, one needs to write

$$\hat{u}(n) = \sum_{i=1}^{p} c_i(n) q_i \tag{1.8}$$

where $p < M$. The reconstruction error Ξ is then defined as

$$\Xi(n) = u(n) - \hat{u}(n) = \sum_{i=p+1}^{M} \lambda_i. \tag{1.9}$$

Hence the approximation will be good if the remaining eigenvalues $\lambda_{p+1}, \ldots \lambda_M$ are all very small.

Now to illustrate the implications of a low rank model [2], consider that the data vector $u(n)$ is corrupted by the noise $v(n)$. Then the data $y(n)$ is represented by

$$y(n) = u(n) + v(n). \tag{1.10}$$

Since the data and the noise are uncorrelated,

$$\varepsilon\left[u(n)v^H(n)\right] = [0] \text{ and } \varepsilon\left[v(n)v^H(n)\right] = \sigma^2[\mathrm{I}], \tag{1.11}$$

where [0] and [I] are the null and identity matrices, respectively, and the variance of the noise at each element is σ^2. The mean squared error now in a noisy environment is

$$\Xi_o = \varepsilon\left[\left\|y(n) - u(n)\right\|^2\right] = \varepsilon\left[\left\|v(n)v^H(n)\right\|\right] = \sum_{i=1}^{M} \left|v(n)\right|^2 = M\sigma^2. \tag{1.12}$$

Now to make a low-rank approximation in a noisy environment, define the approximated data vector by

$$r(n) = \hat{u}(n) + \hat{v}(n) = \sum_{i=1}^{p} c_i(n) q_i + v_i(n). \tag{1.13}$$

In this case, the reconstruction error for the reduced-rank model is given by

$$\Xi_{rr} = \varepsilon\left[\left\|r(n) - \hat{u}(n)\right\|^2\right] = \sum_{i=p+1}^{M} \lambda_i + p\,\sigma^2. \tag{1.14}$$

This equation implies that the mean squared error Ξ_{rr} in the low-rank approximation is smaller than the mean squared error Ξ_o to the original data vector without any approximation, if the first term in the summation is small. So low-rank modelling provides some advantages provided

$$\sum_{i=p+1}^{M} \lambda_i < (M-p)\sigma^2, \tag{1.15}$$

which illustrates the result of a *bias-variance* trade off. In particular, it illustrates that using a low-rank model for representing the data vector $u(n)$ incurs a bias through the p terms of the basis vector. Interestingly enough, introducing this bias is done knowingly in return for a reduction in variance, namely the part of the mean squared error due to the additive noise vector $v(n)$. *This illustrates that the motivation for using a simpler model that may not exactly match the underlying physics responsible for generating the data vector u(n), hence the bias, but the model is less susceptible to noise, hence a reduction in variance* [1, 2].

We now use this principle in the interpolation/extrapolation of various system responses. Since the data are from a linear time invariant (LTI) system that has a bounded input and a bounded output and satisfy a second-order partial differential equation, the associated time-domain eigenvectors are sums of complex exponentials and in the transformed frequency domain are ratios of two polynomials. As discussed, these eigenvectors form the optimal basis in representing the given data and hence can also be used for interpolation/extrapolation of a given data set. Consequently, we will use either of these two models to fit the data as seems appropriate. To this effect, we present the Matrix Pencil Method (MP) which approximates the data by a sum of complex exponentials and in the transformed domain by the Cauchy Method (CM) which fits the data by a ratio of two rational polynomials. In applying these two techniques it is necessary to be familiar two other topics which are the singular value decomposition and the total least squares which are discussed next.

1.3 An Introduction to Singular Value Decomposition (SVD) and the Theory of Total Least Squares (TLS)

1.3.1 Singular Value Decomposition

As has been described in [https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf] "*Singular value decomposition (SVD) can be looked at from three mutually compatible points of view. On the one hand, we can see it as a method for transforming correlated variables into a set of uncorrelated ones that better expose the various relationships among the original data items. At the same time, SVD is a method for identifying and ordering the dimensions along which data points exhibit the most variation. This ties in to the third way of viewing SVD, which is that once we have identified where the most variation is, it's possible to find the best approximation of the original data points using fewer dimensions. Hence, SVD can be seen as a method for data reduction.* We shall illustrate this last point with an example later on.

First, we will introduce a critical component in solving a Total Least Squares problem called the Singular Value Decomposition (SVD). The singular value decomposition is one of the most important concepts in linear algebra [2].

To start, we first need to understand what eigenvectors and eigenvalues are as related to a dynamic system. If we multiply a vector $\boldsymbol{x}$ by a matrix $[\boldsymbol{A}]$, we will get a new vector, $\boldsymbol{A}\mathbf{x}$. The next equation shows the simple equation relating a matrix $[\boldsymbol{A}]$ and an eigenvector $\boldsymbol{x}$ to an eigenvalue λ (just a number) and the original $\boldsymbol{x}$.

$$A\boldsymbol{x} = \lambda\boldsymbol{x} \tag{1.16}$$

$[\boldsymbol{A}]$ is assumed to be a square matrix, $\boldsymbol{x}$ is the eigenvector, and λ is a value called the eigenvalue. Normally when any vector $\boldsymbol{x}$ is multiplied by any matrix $[\boldsymbol{A}]$ a new vector results with components pointing in different directions than the original $\boldsymbol{x}$. However, eigenvectors are special vectors that come out in the same direction even after they are multiplied by the matrix $[\boldsymbol{A}]$. From (1.16) we can see that when one multiplies an eigenvector by $[\boldsymbol{A}]$, the new vector $A\boldsymbol{x}$ is just the eigenvalue λ times the original $\boldsymbol{x}$. This eigenvalue determines whether the vector $\boldsymbol{x}$ is shrunk, stretched, reversed, or unchanged. Eigenvectors and eigenvalues play crucial roles in linear algebra ranging from simplifying matrix algebra such as taking the 500^{th} power of $[\boldsymbol{A}]$ to solving differential equations. To take the 500^{th} power of $[\boldsymbol{A}]$, one only needs to find the eigenvalues and eigenvectors of $[\boldsymbol{A}]$ and take the 500^{th} power of the eigenvalues. The eigenvectors will not change direction and the multiplication of the 500^{th} power of the eigenvalues and the eigenvectors will result in $[\boldsymbol{A}]^{500}$. As we will see in later sections, the eigenvalues can also provide important parameters of a system transfer function such as the poles.

One way to characterize and extract the eigenvalues of a matrix $[\boldsymbol{A}]$ is to diagonalize it. Diagonalizing a matrix not only provides a quick way to extract eigenvalues but important parameters such as the rank and dimension of a matrix can be found easily once a matrix is diagonalized. To diagonalize matrix $[\boldsymbol{A}]$, the eigenvalues of $[\boldsymbol{A}]$ must first be placed in a diagonal matrix, $[\boldsymbol{\Lambda}]$. This is completed by forming an eigenvector matrix $[\boldsymbol{S}]$ with the eigenvectors of $[\boldsymbol{A}]$ put into the columns of $[\boldsymbol{S}]$ and multiplying as such

$$[\boldsymbol{S}]^{-1}[\boldsymbol{A}][\boldsymbol{S}] = [\Lambda] = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \tag{1.17}$$

(1.17) can now be rearranged and $[\boldsymbol{A}]$ can also be written as

$$[\boldsymbol{A}] = [\boldsymbol{S}][\Lambda][S]^{-1} \tag{1.18}$$

We start to encounter problems when matrices are not only square but also rectangular. Previously we assumed that $[\boldsymbol{A}]$ was an n by n square matrix. Now we will assume $[\boldsymbol{A}]$ is any m by n rectangular matrix. We would still like to simplify the matrix or "diagonalize" it but using $[\boldsymbol{S}]^{-1}[\boldsymbol{A}][\boldsymbol{S}]$ is no longer ideal for a few reasons; the eigenvectors of $[\boldsymbol{S}]$ are not always orthogonal, there are sometimes not enough eigenvectors, and using $\boldsymbol{A}\,\boldsymbol{x} = \lambda\,\boldsymbol{x}$ requires $[\boldsymbol{A}]$ to be a square matrix.

However, this problem can be solved with the singular value decomposition but of course at a cost. The SVD of [$\boldsymbol{A}$] results in the following

$$[\boldsymbol{A}]_{(m\times n)} = [\boldsymbol{U}][\Sigma][\boldsymbol{V}]^T = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_r \end{bmatrix}_{(m\times r)} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}_{(r\times r)} \begin{bmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_r \end{bmatrix}_{(n\times r)} \tag{1.19}$$

where m is the number of rows of [$\boldsymbol{A}$], n is the number of columns of [$\boldsymbol{A}$], and r is the rank of [$\boldsymbol{A}$]. The SVD of [$\boldsymbol{A}$], which can now be rectangular or square, will have two sets of singular vectors, $\boldsymbol{u}$'s and $\boldsymbol{v}$'s. The $\boldsymbol{u}$'s are the eigenvectors of [$\boldsymbol{A}$] [$\boldsymbol{A}$]$*$ and the $\boldsymbol{v}$'s are the eigenvectors of [$\boldsymbol{A}$]$^{\mathrm{T}}$[$\boldsymbol{A}$]. [$\boldsymbol{U}$] and [$\boldsymbol{V}$] are also unitary matrices which means that [$\boldsymbol{U}$]$*$[$\boldsymbol{U}$] =[$\boldsymbol{I}$] and [$\boldsymbol{V}$]$*$[$\boldsymbol{V}$] = [$\boldsymbol{I}$]. In other words, they are orthogonal where $*$ denoted complex conjugate transpose. The σ's are the singular values which also so happen to be the square roots of the eigenvalues of both [$\boldsymbol{A}$][$\boldsymbol{A}$]* and [$\boldsymbol{A}$]*[$\boldsymbol{A}$]. We are not totally finished however because [$\boldsymbol{U}$] and [$\boldsymbol{V}$] are not square matrices. While (1.19) is the diagonalization of [$\boldsymbol{A}$], the matrix equation is technically not valid since we cannot multiply these rectangular matrices of different sizes. To make them square we will need $n - r$ more $\boldsymbol{v}$'s and $m - r$ more $\boldsymbol{u}$'s. We can get these required $\boldsymbol{u}$'s and $\boldsymbol{v}$'s from the nullspace $\boldsymbol{N}(\boldsymbol{A})$ and the left nullspace $\boldsymbol{N}(\boldsymbol{A}*)$. Once the new $\boldsymbol{u}$'s and $\boldsymbol{v}$'s are added, the matrices are square and [$\boldsymbol{A}$] will still equal [$\boldsymbol{U}$][$\boldsymbol{\Sigma}$][$\boldsymbol{V}$]$^{\mathrm{T}}$. The true SVD of [$\boldsymbol{A}$] will now be

$$[\boldsymbol{A}]_{(m\times n)} = [\boldsymbol{U}][\Sigma][\boldsymbol{V}]^T = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_r & \cdots & \boldsymbol{u}_m \end{bmatrix}_{(m\times m)} \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_r & \\ & & & 0 \end{bmatrix}_{(m\times n)} \begin{bmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_r & \cdots & \boldsymbol{v}_n \end{bmatrix}_{(n\times n)} \tag{1.20}$$

The new singular value matrix [Σ] is the same matrix as the old $r \times r$ matrix but with $m - r$ new rows of zero and $n - r$ columns of new zero added. The theory of total least squares (TLS) heavily utilizes the SVD as will be seen in the next section.

As an example consider the letter **X**. We now discretize the image on a 20×20 grid as seen in Figure 1.1. We place a 1 where there is no portion of the letter **X** and a zero where there is some portion of the letter. This results in the following matrix (1.21) representing the letter **X** digitally on a 20×20 grid of the matrix $\boldsymbol{A}$. Now if we ask the question "What is the information content in the matrix **X** consisting of 1's and 0's**?** Clearly one would not require 20×20 = 400 pieces of information to represent **X** in (1.21) as there is a lot of redundancy in the system. This is where the SVD comes in and addresses this issue in quite a satisfactory way. If one performs a SVD of the matrix A given by (1.21), one will find that diagonal $\sum$ matrix in (1.20) has a lot of zero entries. In fact only seven of the singular values are not close to zero as presented in Table 1.1. This implies that

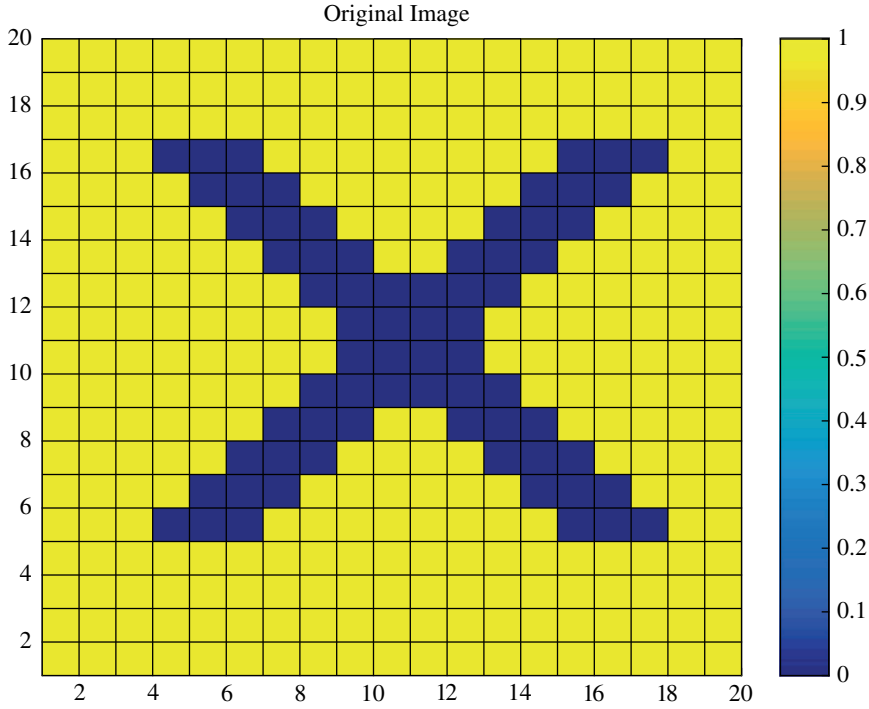

Figure 1.1 Discretization of the letter X on a 20×20 grid.

Table 1.1 List of all the singular values of the matrix $\boldsymbol{A}$.

16.9798798959208
4.57981833410903
3.55675452579199
2.11591593148044
1.74248432449203
1.43326538670857
0.700598651862344
$8.40316310453450\times10^{-16}$
$2.67120960711993\times10^{-16}$
$1.98439889201509\times10^{-16}$
$1.14953653060279\times10^{-16}$
$4.74795444425376\times10^{-17}$
$1.98894713470634\times10^{-17}$
$1.64625309682086\times10^{-18}$
$5.03269559457945\times10^{-32}$
$1.17624286081108\times10^{-32}$
$4.98861204114981e\times10^{-33}$
$4.16133005156854\times10^{-49}$
$1.35279056788548\times10^{-80}$
$1.24082758075064\times10^{-112}$

the information content of the picture in Figure 1.1 is small. The SVD can also be used to reconstruct the original data matrix with reduced information without sacrificing any accuracy.

$$\mathbf{A} = \begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&0&0&0&1&1&1&1&1&1&1&1&0&0&0&1&1&1\\
1&1&1&1&0&0&0&1&1&1&1&1&1&0&0&0&1&1&1&1\\
1&1&1&1&1&0&0&0&1&1&1&1&0&0&0&1&1&1&1&1\\
1&1&1&1&1&1&0&0&0&1&1&0&0&0&1&1&1&1&1&1\\
1&1&1&1&1&1&1&0&0&0&0&0&0&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&0&0&0&0&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&0&0&0&0&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&0&0&0&0&0&0&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&0&0&0&1&1&0&0&0&1&1&1&1&1\\
1&1&1&1&1&1&0&0&0&1&1&1&1&0&0&0&1&1&1&1\\
1&1&1&1&1&0&0&0&1&1&1&1&1&1&0&0&0&1&1&1\\
1&1&1&0&0&0&1&1&1&1&1&1&1&1&0&0&0&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1
\end{bmatrix} \tag{1.21}$$

To illustrate the point, we now perform a rank-1 approximation for the matrix and we will require

$$\boldsymbol{A} \approx \boldsymbol{U}_1 \sigma_1 \boldsymbol{V}_1^* \tag{1.22}$$

where $\boldsymbol{U}_1$ is a column matrix of size 20×1 and so is $\boldsymbol{V}_1$. The rank one approximation of $\boldsymbol{A}$ is seen in Figure 1.2.

The advantage of the SVD is that an error in the reconstruction of the image can be predicted without actually knowing the actual solution. This is accomplished by looking at the second largest singular value. The result is not good and we did not expect it to be. So now if we perform a Rank-2 reconstruction for the image, it will be given by

$$\boldsymbol{A} \approx \sigma_1 \boldsymbol{U}_1 \boldsymbol{V}_1^* + \sigma_2 \boldsymbol{U}_2 \boldsymbol{V}_2^* \tag{1.23}$$

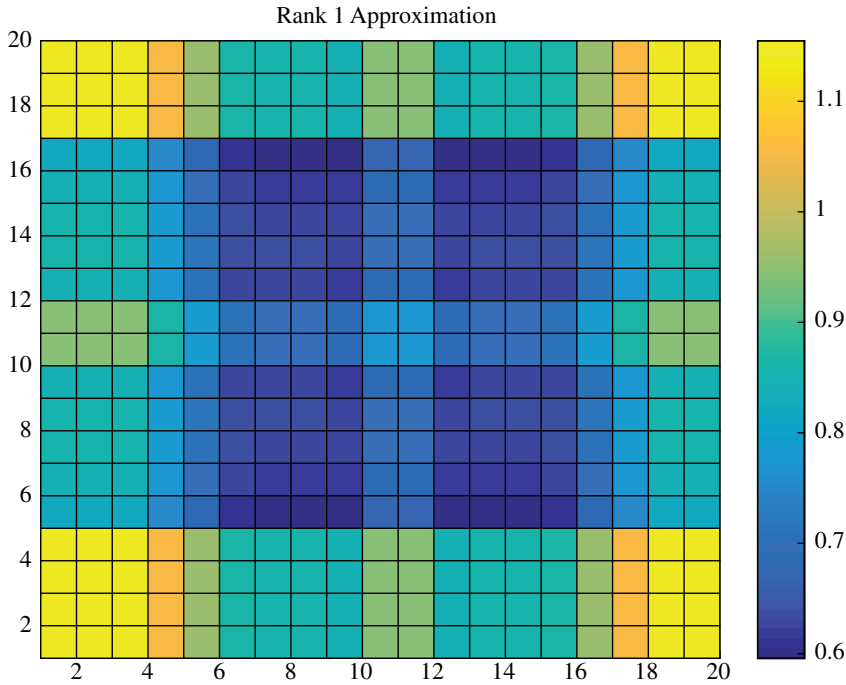

Figure 1.2 Rank-1 approximation of the image X.

where now two of the right and the left singular values are required. In this case the reank-2 approximation will be given by Figure 1.3. In this case we have captured the largest two singular values as seen from Table 1.1.

Again as we study how the picture evolves as we take more and more on the singular values and the vectors the accuracy in the reconstruction increases. For example, Figure 1.4 provides the rank-3 reconstruction, Figure 1.5 provides the rank-4 reconstruction, Figure 1.6 provides the rank-5 reconstruction, Figure 1.7 provides the rank-6 reconstruction. It is seen with each higher rank approximation the reconstructed picture resembles the actual one. The error in the approximation decreases as the magnitudes of the neglected singular values become small. This is the greatest advantage of the Singular Value Decomposition over say for example, the Tikhonov regularization as the SVD provides an estimate about the error in the reconstruction even though the actual solution is unknown.

Finally, it is seen that there are seven large singular values and after that they become mostly zeros. Therefore, we should achieve a perfect reconstruction with Rank-7 which is seen in Figure 1.8 and going to rank-8 which is presented in Figure 1.9, will not make any difference in the quality of the image. This is

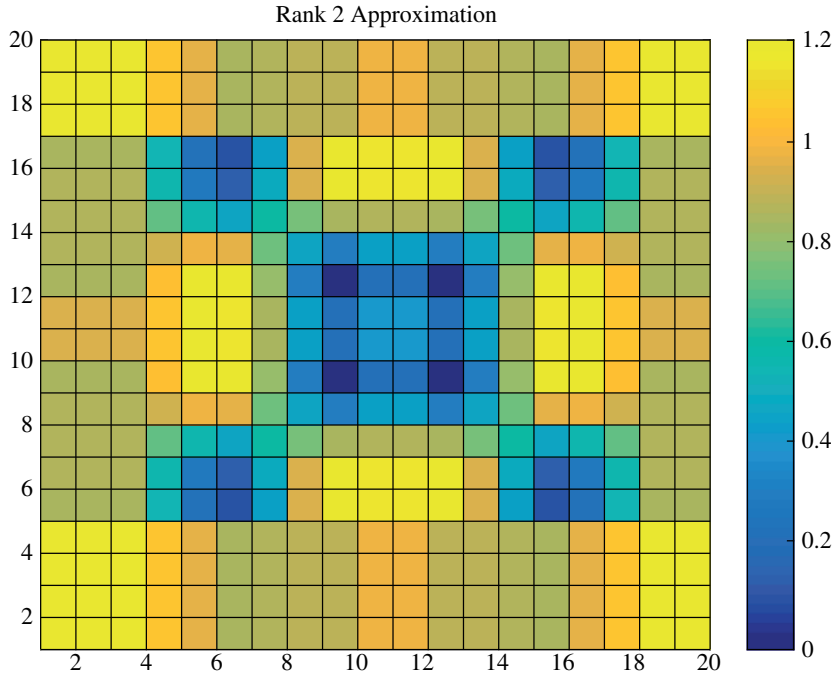

Figure 1.3 Rank-2 approximation of the image X.

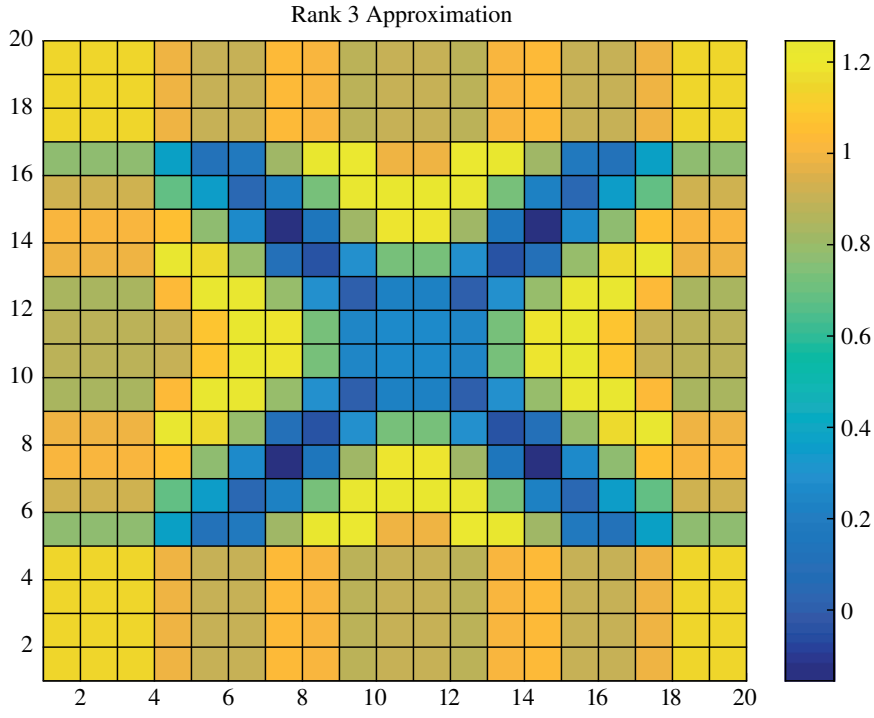

Figure 1.4 Rank-3 approximation of the image X.

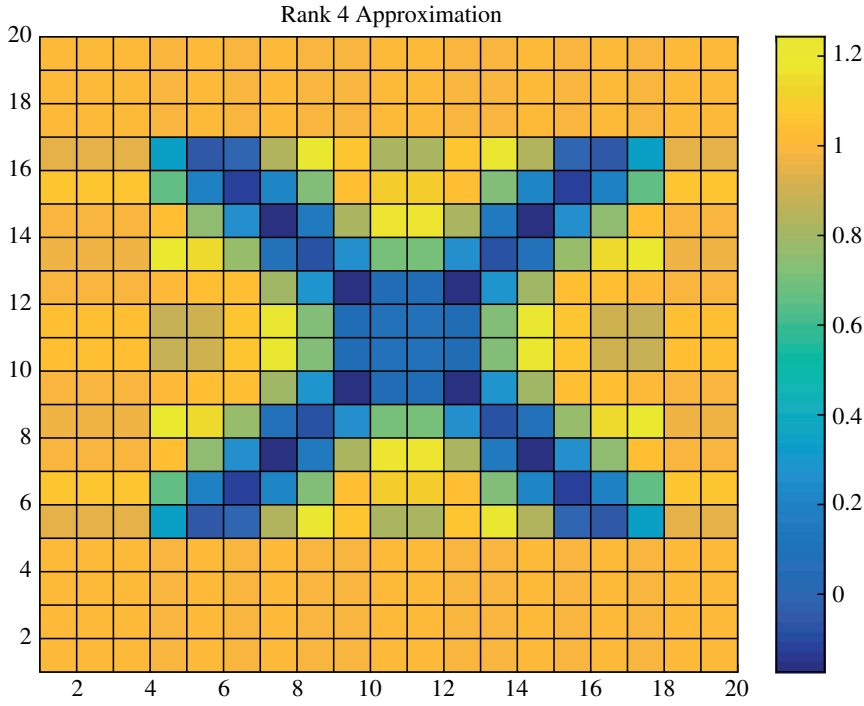

Figure 1.5 Rank-4 approximation of the image X.

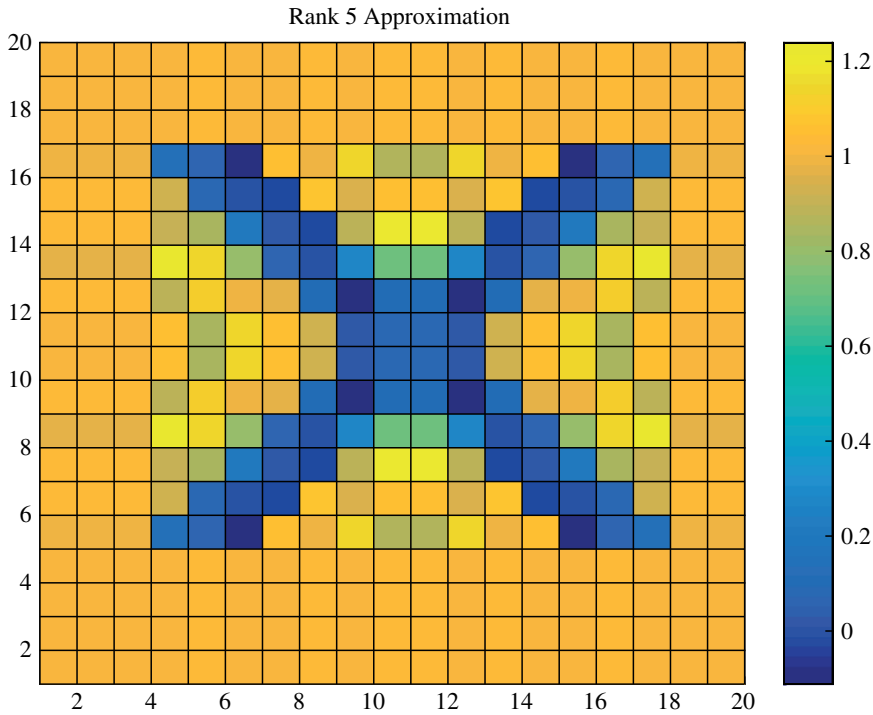

Figure 1.6 Rank-5 approximation of the image X.

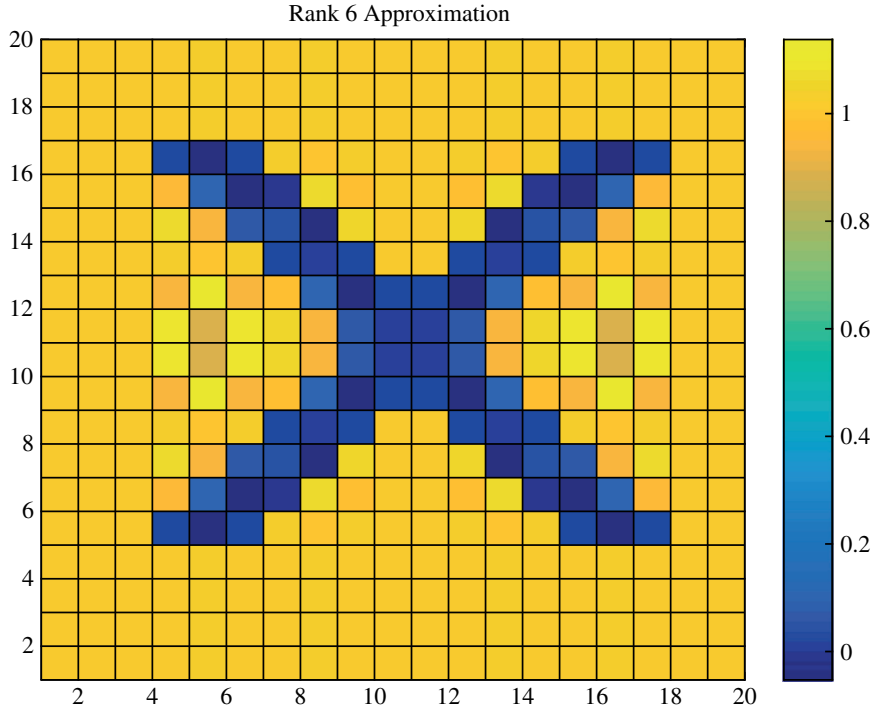

Figure 1.7 Rank-6 approximation of the image X.

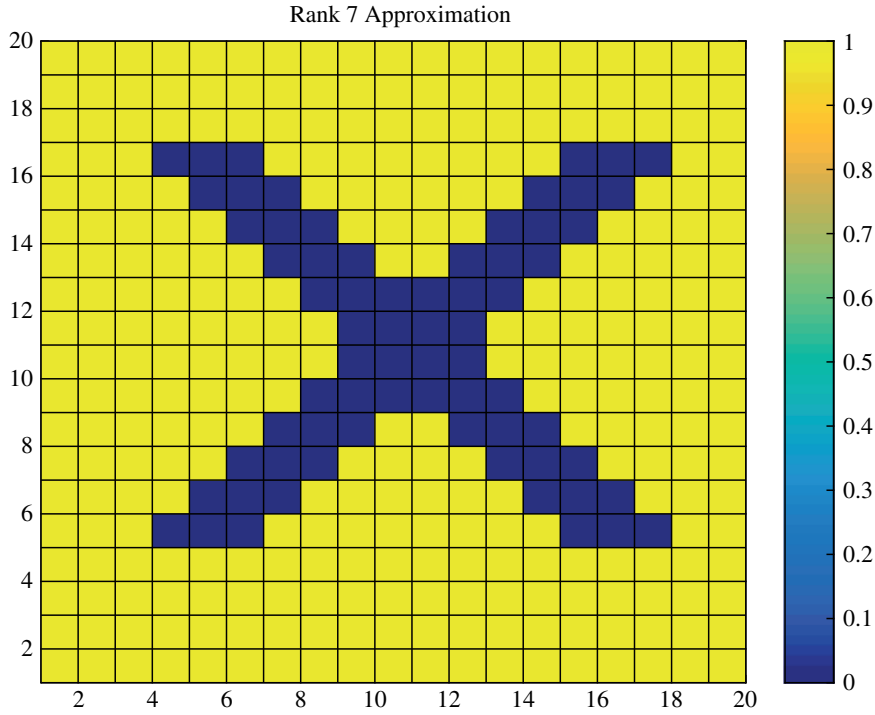

Figure 1.8 Rank-7 approximation of the image X.

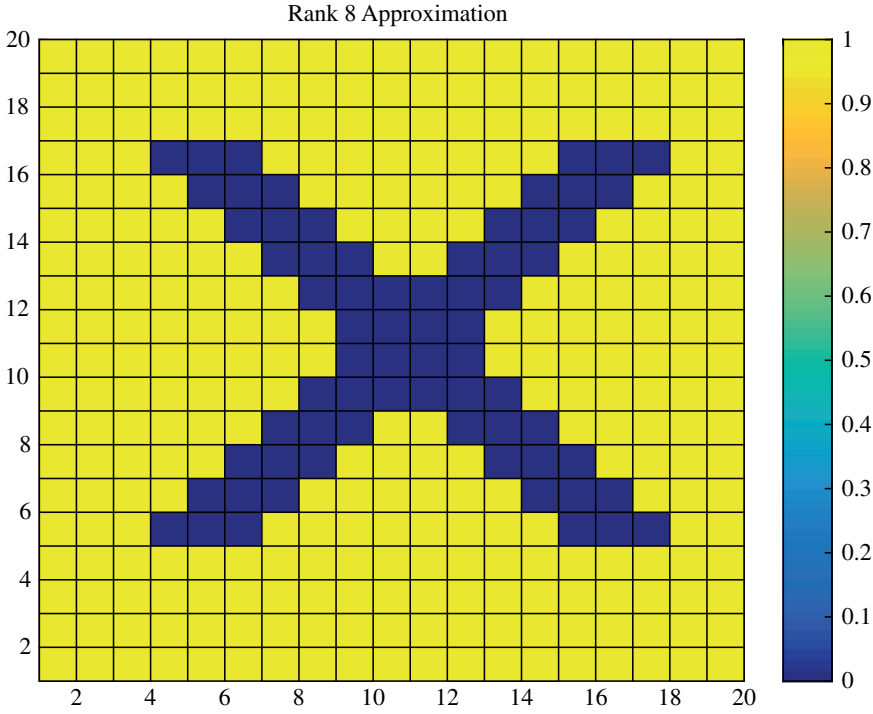

Figure 1.9 Rank-8 approximation of the image X.

now illustrated next through the mean squared error between the true solution and the approximate ones.

In Figure 1.10, the mean squared error between the actual picture and the approximate ones is presented. It is seen, as expected the error is given by this data

$$(0.32,\ 0.22,\ 0.16,\ 0.09,\ \ 0.06,\ 0.02,\ 0.00,\ 0.00, \ldots\ldots\ldots). \tag{1.24}$$

This is a very desirable property for the SVD. Next, the principles of total least squares is presented.

1.3.2 The Theory of Total Least Squares

The method of total least squares (TLS) is a linear parameter estimation technique and is used in wide variety of disciplines such as signal processing, general engineering, statistics, physics, and the like. We start out with a set of m measured data points $\{(x_1,y_1),\ldots,(x_m,y_m)\}$, and a set of n linear coefficients $(a_1,\ldots,a_n)$ that describe a model, $\hat{y}(x;a)$ where $m > n$ [3, 4]. The objective of total least

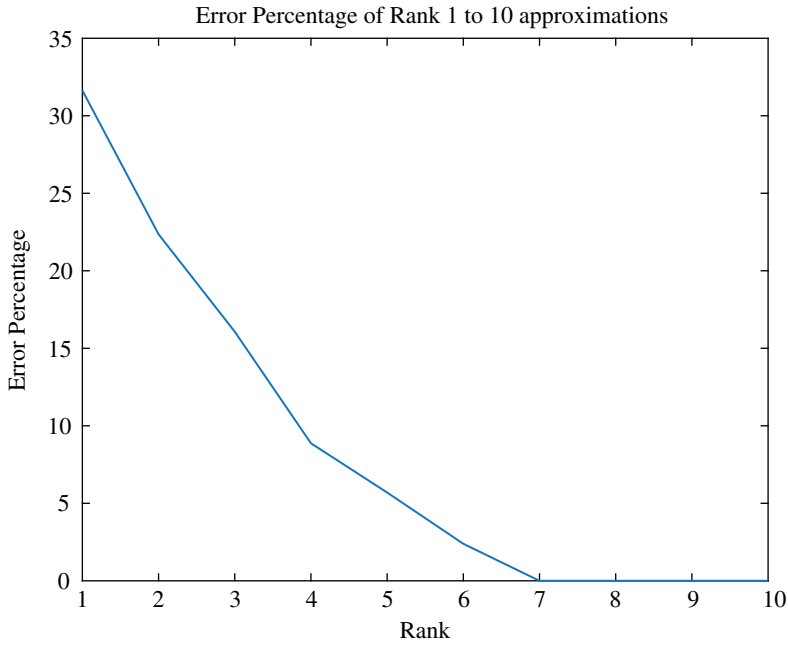

Figure 1.10 Mean squared error of the approximation.

squares is to find the linear coefficients that best approximate the model in the scenario that there is missing data or errors in the measurements. We can describe the approximation by a simple linear expression

$$y \approx Xa \tag{1.25}$$

Since $m > n$, there are more equations than unknowns and therefore (1.25) has an overdetermined set of equations. Typically, an overdetermined system of equation is best solved by the ordinary least squares where the unknown is given by

$$a = (X^*X)^{-1}X^*y \tag{1.26}$$

where X^* represents the complex conjugate transpose of the matrix X. The least squares can take into account if there are some uncertainties like noise in y as it is a least squares fit to it. However, if there is uncertainty in the elements of the matrix X then the ordinary least squares cannot address it. This is where the total Least squares come in. In the total least squares the matrix equation (1.25) is cast into a different form where uncertainty in the elements of both the matrix X and y can be taken into account.

$$\left[X \vdots y\right]\begin{bmatrix} a \\ -1 \end{bmatrix} = \left[X \vdots y\right][b] = [0] \tag{1.27}$$

In this form one is solving for the solution to the composite matrix by searching for the eigenvector/singular vector corresponding to the zero eigen/singular value. If the matrix X is rectangular then the eigenvalue concept does not apply and one needs to deal with the singular vectors and the singular values.

The best approximation according to total least squares is that minimizes the norm of the difference between the approximated data and the model $\hat{y}(x;a)$ as well as the independent variables X. Considering the errors of the measured data vector, y, and the independent variables, X, (1.25) can be re-written as

$$y + \tilde{y} = \left[X + \tilde{X}\right]a \tag{1.28}$$

where $\tilde{y}$ and $\tilde{X}$ are the errors in both the dependent variable measurements and independent variable measurements, respectively. We then want to approximate in a way that minimizes these errors in the dependent and independent variables. This can be expressed by,

$$min \left\| \left[\tilde{X}\,\tilde{y}\right] \right\|_F^2 \tag{1.29}$$

where $\left[\tilde{X}\,\tilde{y}\right]$ is an augmented matrix with the columns of error matrix $\tilde{X}$ concatenated with the error vector $\tilde{y}$. The operator $\|\bullet\|_{\mathbf{F}}$ represents the Frobenius norm of the augmented matrix. The Frobenius norm is defined as the square root of the sum of the absolute squares of all of the elements in a matrix. This can be expressed in equation form as the following, where A is any matrix,

$$\|A\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2 = trace\left(A^T A\right) = \sum_{i=1}^{n} \sigma_i^2 \tag{1.30}$$

and where σ_i is the i-th singular value of matrix A.

We will now bring the right-hand side of (1.28) over to the left side of the equation and equate it to zero as such

$$\left[X + \tilde{X}; y + \tilde{y}\right]\begin{bmatrix} a \\ -1 \end{bmatrix} = 0 \tag{1.31}$$

If the concatenated matrix $[X\ y]$ has a rank of $n + 1$, the $n + 1$ columns of the matrix are linearly independent and the $n + 1$, m-dimensional columns of the matrix span the same n-dimensional space as X. In order to have a unique solution for the coefficients, a, the matrix $[X + \tilde{X}; y + \tilde{y}]$ must have n linearly independent columns. However, this matrix has $n + 1$ columns in total and therefore is rank is deficient by 1. We then must find the smallest matrix $[\tilde{X}\,\tilde{y}]$ that

changes matrix $[X\ y]$ with a rank of $n + 1$, to a matrix $\{[X\ y] + [\widetilde{X}\ \widetilde{y}]\}$ with a rank n. According to the Eckart-Young-Mirsky theorem we can achieve this by defining $\{[X\ y] + [\widetilde{X}\ \widetilde{y}]\}$ as the best rank-n approximation to $[X\ y]$ and by eliminating the last singular value of $[X\ y]$ which contains the least amount of system information and provides a unique solution. The Eckart–Young–Mirsky theorem (https://en.wikipedia.org/wiki/Low-rank_approximation) states a low-rank approximation is a minimization problem, in which the cost function measures the fit between a given matrix (the data) and an approximating matrix (the optimization variable), subject to a constraint that the approximating matrix has reduced rank. To illustrate how this is accomplished, we take the SVD of $[X\ y]$ as follows

$$[X \quad y] = \begin{bmatrix} U_x & u_y \end{bmatrix} \begin{bmatrix} \Sigma_x & \\ & \sigma_y \end{bmatrix} \begin{bmatrix} V_{xx} & v_{xy} \\ v_{yx} & v_{yy} \end{bmatrix}^T \tag{1.32}$$

where U_x has n columns, u_y is a column vector, Σ_x contains the n largest singular values diagonally, σ_y is the smallest singular value, V_{xx} is a $n \times n$ matrix, and v_{yy} is scalar. Let us multiple both sides by matrix V.

$$[X \quad y] \begin{bmatrix} V_{xx} & v_{xy} \\ v_{yx} & v_{yy} \end{bmatrix} = \begin{bmatrix} U_x & u_y \end{bmatrix} \begin{bmatrix} \Sigma_x & \\ & \sigma_y \end{bmatrix} \tag{1.33}$$

Next, we will equate just the last columns of the matrix multiplication occurring in (1.32).

$$[X \quad y] \begin{bmatrix} v_{xy} \\ v_{yy} \end{bmatrix} = u_y \sigma_y \tag{1.34}$$

From the Eckart-Young theorem, we know that $\{[X\ y] + [\widetilde{X}\ \widetilde{y}]\}$ is the closest rank-n approximation to $[X\ y]$. Matrix $\{[X\ y] + [\widetilde{X}\ \widetilde{y}]\}$ has the same singular vectors contained in Σ_x above with σ_y equal to zero. We can then write the SVD of $\{[X\ y] + [\widetilde{X}\ \widetilde{y}]\}$ as such

$$\begin{bmatrix} X + \widetilde{X}; & y + \widetilde{y} \end{bmatrix} = \begin{bmatrix} U_x & u_y \end{bmatrix} \begin{bmatrix} \Sigma_x & \\ & 0 \end{bmatrix} \begin{bmatrix} V_{xx} & v_{xy} \\ v_{yx} & v_{yy} \end{bmatrix}^T \tag{1.35}$$

To obtain $[\widetilde{X}\ \widetilde{y}]$ we must solve the following

$$\begin{bmatrix} \widetilde{X} & \widetilde{y} \end{bmatrix} = [X \quad y] + \begin{bmatrix} \widetilde{X} & \widetilde{y} \end{bmatrix} - [X \quad y] \tag{1.36}$$

(1.36) can be solved by first using (1.32) and (1.35) which results in

$$\begin{aligned}\begin{bmatrix}\widetilde{X} & \widetilde{y}\end{bmatrix} &= -\begin{bmatrix}U_x & u_y\end{bmatrix}\begin{bmatrix}0 & \\ & \sigma_y\end{bmatrix}\begin{bmatrix}V_{xx} & v_{xy}\\ v_{yx} & v_{yy}\end{bmatrix}^T \\ &= -\begin{bmatrix}0 & u_y\sigma_y\end{bmatrix}\begin{bmatrix}V_{xx} & v_{xy}\\ v_{yx} & v_{yy}\end{bmatrix}^T = -u_y\sigma_y\begin{bmatrix}v_{xy}\\ v_{yy}\end{bmatrix}^T\end{aligned} \tag{1.37}$$

Then, from (1.34) we can rewrite (1.37) as

$$\begin{bmatrix}\widetilde{X} & \widetilde{y}\end{bmatrix} = -\begin{bmatrix}X & y\end{bmatrix}\begin{bmatrix}v_{xy}\\ v_{yy}\end{bmatrix}\begin{bmatrix}v_{xy}\\ v_{yy}\end{bmatrix}^T \tag{1.38}$$

Finally, $\{[X\ y] + [\widetilde{X}\ \widetilde{y}]\}$ can be defined as

$$\begin{bmatrix}X + \widetilde{X}; & y + \widetilde{y}\end{bmatrix} = \begin{bmatrix}X & y\end{bmatrix} - \begin{bmatrix}X & y\end{bmatrix}\begin{bmatrix}v_{xy}\\ v_{yy}\end{bmatrix}\begin{bmatrix}v_{xy}\\ v_{yy}\end{bmatrix}^T \tag{1.39}$$

After multiplying each term in (1.39) by $\begin{bmatrix}v_{xy}\\ v_{yy}\end{bmatrix}$ we get the following

$$\begin{bmatrix}X + \widetilde{X}; & y + \widetilde{y}\end{bmatrix}\begin{bmatrix}v_{xy}\\ v_{yy}\end{bmatrix} = \begin{bmatrix}X & y\end{bmatrix}\begin{bmatrix}v_{xy}\\ v_{yy}\end{bmatrix} - \begin{bmatrix}X & y\end{bmatrix}\begin{bmatrix}v_{xy}\\ v_{yy}\end{bmatrix} \tag{1.40}$$

The right-hand side cancels and we are left with

$$\begin{bmatrix}X + \widetilde{X}; & y + \widetilde{y}\end{bmatrix}\begin{bmatrix}v_{xy}\\ v_{yy}\end{bmatrix} = 0 \tag{1.41}$$

From (1.31) and (1.41) we can solve for the model coefficient a as

$$a = -v_{xy}v_{yy}^{-1} \tag{1.42}$$

The vector v_{xy} is the first n elements of the $n+1$-th columns of the right singular matrix V, of $[X\ y]$ and v_{yy} is the $n+1$-th element of the $n+1$ columns of V. The best approximation of the model is then given by

$$\hat{y} = \begin{bmatrix}X + \widetilde{X}\end{bmatrix}a \tag{1.43}$$

This completes the total least squares solution.

1.4 Conclusion

This first chapter provides the mathematical fundamentals that will be utilized later. The principles presented are the basis of low-rank modelling, singular value decomposition and the method of total least squares.

References

1 L. I. Scharf and D. Tufts, "Rank Reduction for Modeling Stationary Signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, pp. 350–355, 1987.
2 S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, NJ, Third Edition, 1996.
3 G. Strang, *Introduction to Linear Algebra*, Cambridge Press, Wellesley, MA, Fifth Edition, 2016.
4 G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, The Johns Hopkins University Press, Baltimore, MD, 2013.