SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Image Deblurring with Regularized Least Squares and Beyond

av

Lars Lidvall

2022 - No K22

Image Deblurring with Regularized Least Squares and Beyond

Lars Lidvall

Självständigt arbete i matematik 15 högskolepoäng, grundnivå

Handledare: Yishao Zhou

2022

Abstract

In this report we study the optimization problem *least squares* $\min_x \|b - Ax\|_2^2$ with applications to image deblurring. Least squares can be solved directly in four ways: by the normal equation, the pseudoinverse, QR-decomposition and regularization. The theory for these methods are covered in detail. Least squares can be solved iteratively using a gradient descent method. Specifically gradient descent and Polyak heavy ball are covered, with focus on the theory of convergence for these methods. Finally, a direct and an iterative method are compared on a specific example of deblurring, where other ways to deblur are also discussed.

Acknowledgements

First and foremost I would like to thank my supervisor professor Yishao Zhou for being especially good with responding by e-mail and always dedicating considerable time to give me tips and sources to read, which has been greatly appreciated.

I would also like to extend a particular thanks to a previous teacher Alexander Westerström for bringing back my passion for mathematics and learning, by having been encouraging and always interesting to talk to and discuss various things with.

Moreover I would like to thank my parents Rolf and Lena for continuously supporting me while writing this report and through other endeavours.

Lastly I would like to thank more people close to me. Especially my partner Julia, brothers Kalle and Hugo, and friends Didrik, Jones, Binel, Nikola, Max, and Micke, for caring about me and my work throughout the process of working on it.

Contents

Introduction

In mathematics as a whole, solving equations is very important. Practically speaking, in Linear Algebra we are often talking about solving systems of linear equations, where a system of m equations with n variables can be neatly written as $Ax = b$, where A is an $m \times n$ matrix, x is a vector with n unknown variables and b is a vector of m constants. This is a quite general form of problem which perhaps not surprisingly can model problems in "the real world". For real world applications, it may therefore be important to be able to solve $Ax = b$. However, this system may be unsolvable for many reasons. Approximate solutions are hence necessary.

A quite straightforward idea is that we look at the error of an approximate solution $\hat{x}$, being $b - A\hat{x}$, where this is a good approximation if this error is in some way close to the zero vector $\mathbf{0}$ with m zeros. In this report we will measure the closeness to the zero vector by the squared *Euclidean norm* $(\|b - A\hat{x}\|_2)^2$, more simply denoted by $\|b - A\hat{x}\|_2^2$. The main idea is that we can look at this as a function $f : \mathbb{R}^n \to \mathbb{R}$, defined by $f(x) = \|b - Ax\|_2^2$, which we can try to minimize, resulting in the optimization problem

$$\min_x \|b - Ax\|_2^2.$$

This problem is called *least squares*, but it still has some flaws. To solve this problem we find the stationary points of f, which turn out to be the points $\hat{x}$ satisfying $A^T A\hat{x} = A^T b$. This is called the *normal equation*. Again we arrive at a system of linear equations which may be unsolvable. In this report we will cover two ways to *regularize* the problem of least squares such that we may arrive at a solution.

The first way covered is by changing the problem to a new regularized least squares problem $\min_x \|b - Ax\|_2^2 + \delta R(x)$ and solving that, giving an approximate solution to least squares.

The second way covered is by using an iterative method until we arrive at some point $\hat{x}$ which does satisfy the normal equation and is therefore a solution to least squares. Specifically *gradient descent* methods are covered in this report.

For results of these methods on a particular example, see section 5.3 Results from page 70 and a few pages onward.

Note that the content covered here is part of a very broad and deep subject, where the goal of this report is to explain these concepts to students that have taken basic courses in Linear Algebra and Analysis. The layout of the report is according to my own understanding of the material.

1 Preliminaries

In this chapter we will cover some concepts that are used extensively in this report. Knowing these concepts are needed to be able to have a deeper understanding of the material.

1.1 Basics

In this report we are almost exclusively working within the field $\mathbb{R}$, so some definitions are different to if we were working within the field $\mathbb{C}$. Specifically we will use the transpose T instead of the Hermitian conjugate H in the definition of orthogonal and Euclidean norm.

1.1.1 Special rectangular matrices

First we introduce the notation for the entries of a matrix, because this will sometimes be the easiest way to conceptualize different matrices generally.

Definition 1.1.1. The *entry* of a matrix $A \in \mathbb{R}^{m\times n}$ in row $1 \leq i \leq m$ from the top, and column $1 \leq j \leq n$ from the left, will be denoted by $(A)_{i,j}$.

Definition 1.1.2. The k:th *entry* of a vector $v \in \mathbb{R}^n$, where $1 \leq k \leq m$ will be denoted by v_k or $(v)_k$.

Definition 1.1.3. $D \in \mathbb{R}^{m\times n}$ will be called *diagonal* if and only if

$$(D)_{i,j} = 0, \quad \text{when } i \neq j.$$

This means that the entries of D not on its main diagonal are all equal to zero.

Definition 1.1.4. $Q \in \mathbb{R}^{m\times n}$ will be called *orthogonal* if and only if

$$Q^T Q = I_n,$$

where $I_n \in \mathbb{R}^{n\times n}$ is the $n \times n$ identity matrix.

Definition 1.1.5. $R \in \mathbb{R}^{m\times n}$ will be called *upper-triangular* if and only if

$$(R)_{i,j} = 0, \quad \text{when } i > j.$$

This means that the entries of R below its main diagonal are all equal to zero.

Definition 1.1.6. $M \in \mathbb{R}^{m\times n}$ will be called a *block matrix* if M is defined by the matrices $M_1 \in \mathbb{R}^{m_1\times n_1}$, $M_2 \in \mathbb{R}^{m_1\times(n-n_1)}$, $M_3 \in \mathbb{R}^{(m-m_1)\times n_1}$ and $M_4 \in \mathbb{R}^{(m-m_1)\times(n-n_1)}$ positioned like

$$M = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

M_1, M_2, M_3 and M_4 will be called the *blocks* of M. Note that M is still a standard matrix, just with the entries of M_1, M_2, M_3 and M_4.

Remark. If "0" is a block then this represents the matrix of only zeros with the dimensions of that block.

1.1.2 Important subspaces

Given a matrix $A \in \mathbb{R}^{m\times n}$, there are two important subspaces $\mathbf{N}(A) \subseteq \mathbb{R}^n$ and $\mathbf{C}(A) \subseteq \mathbb{R}^m$ associated with it. These subspaces are related by the *rank-nullity* theorem. If the matrix is square ($m = n$), then this will tell us if A is invertible.

Definition 1.1.7. The *null space* of a matrix $A \in \mathbb{R}^{m\times n}$ will be denoted by $\mathbf{N}(A)$ and be defined by

$$\mathbf{N}(A) = \{x \in \mathbb{R}^n \mid Ax = \mathbf{0} \in \mathbb{R}^m\}.$$

This is the set of vectors $x \in \mathbb{R}^n$ which become mapped to the zero vector $\mathbf{0} \in \mathbb{R}^m$ by left multiplication with A.

Remark. The null space $\mathbf{N}(A)$ will be called *trivial* if and only if $\mathbf{N}(A) = \{\mathbf{0}\}$, meaning that the only solution to $Ax = \mathbf{0} \in \mathbb{R}^m$ is given by $x = \mathbf{0} \in \mathbb{R}^n$.

Definition 1.1.8. The *column space* of a matrix $A \in \mathbb{R}^{m\times n}$ will be denoted by $\mathbf{C}(A)$ and be defined by

$$\mathbf{C}(A) = \{y \in \mathbb{R}^m \mid y = Ax,\ x \in \mathbb{R}^n\}. \tag{1.1.1}$$

This is the set of vectors $y \in \mathbb{R}^m$ of the form $y = Ax$.

Definition 1.1.9. The *rank* of a matrix $A \in \mathbb{R}^{m\times n}$ will be denoted by $\mathbf{rank}(A)$ and be defined by

$$\mathbf{rank}(A) = \dim(\mathbf{C}(A)).$$

This is the *dimension* of the column space of A.

Remark. $A \in \mathbb{R}^{m\times n}$ will have *full rank* if and only if $\mathbf{rank}(A) = n$.

Now we will state some very well known theorems of Linear Algebra. The proofs will not be given because they are out of the scope of this report.

Theorem 1.1.10 (*rank-nullity*). *Given a matrix $A \in \mathbb{R}^{m\times n}$*

$$\mathbf{rank}(A) + \dim(\mathbf{N}(A)) = n. \tag{1.1.2}$$

Remark. In particular, $\mathbf{N}(A)$ is trivial if and only if A has full rank.

Proof. See page 79 of [4]. o.ε.δ.

Theorem 1.1.11. *A square matrix $M \in \mathbb{R}^{n\times n}$ is invertible if and only if M has full rank.*

Proof. See page 62 of [4]. o.ε.δ.

Theorem 1.1.12 (*spectral*). *A square matrix $M \in \mathbb{R}^{n\times n}$ is (orthogonally) diagonalizable if M is symmetric: $M^T = M$.*

Proof. See page 384 of [2]. o.ε.δ.

1.1.3 Vector norms

The norm of a vector is a measurement of the "length" of a vector. Given a vector $v \in \mathbb{R}^n$, perhaps the most natural norm is the *Euclidean norm*, which gives the length based upon the generalized Pythagorean theorem.

Definition 1.1.13. The *Euclidean norm*, or L_2 *norm*, of a vector $v \in \mathbb{R}^n$ will be denoted by $\|v\|_2$ and be defined by

$$\|v\|_2 = \sqrt{v_1^2 + \ldots + v_n^2} = \sqrt{v^T v}.$$

Remark. In this report, most often the square of the Euclidean norm will be used. This will hence be $\|v\|_2^2 = v^T v$.

Two other standard norms are the L_1 and L_∞ norm.

Definition 1.1.14. The L_1 *norm* of a vector $v \in \mathbb{R}^n$ will be denoted by $\|v\|_1$ and be defined by

$$\|v\|_1 = |v_1| + \ldots + |v_n|.$$

Definition 1.1.15. The L_∞ *norm* of a vector $v \in \mathbb{R}^n$ will be denoted by $\|v\|_\infty$ and be defined by

$$\|v\|_\infty = \max\{|v_1|, \ldots, |v_n|\}.$$

These norms fulfill the definition of being a norm, which is the following.

Definition 1.1.16. $\|\cdot\|$ is a *norm* on $\mathbb{R}^n$ if and only if

$$\|u+v\| \leq \|u\| + \|u\|$$

$$\|\alpha u\| = |\alpha|\|u\|$$

$$\|u\| = 0 \implies u = \mathbf{0}$$

holds for any $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$.

Lemma 1.1.17. *$\|\cdot\|_p$ is a norm for $p \in \{2, 1, \infty\}$.*

Proof. See page 181 and 182 of [4]. ο.ε.δ.

A useful theorem about the Euclidean norm is a special case of the *Cauchy-Schwartz* inequality.

Theorem 1.1.18 (*Cauchy-Schwartz*). *$|u^T v| \leq \|u\|_2\|v\|_2$, where equality holds if and only if u is a scalar multiple of v.*

Proof. See page 168 of [4]. ο.ε.δ.

1.2 Singular value decomposition (SVD)

This section covers the *singular value decomposition* of a matrix $A \in \mathbb{R}^{m\times n}$, given by $A = U\Sigma V^T$. This is one of the most important decompositions in all of Linear Algebra.

1.2.1 Singular values and the SVD

Given a matrix $A \in \mathbb{R}^{m\times n}$, if $m \neq n$, then concept of eigenvalues (and eigenvectors) become meaningless because $Ax \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$, meaning that $Ax \neq \lambda x$, no matter what. This means that if $m \neq n$, we do not have the concept of diagonalization $A = PDP^{-1}$, because there are no eigenvalues. We do however always have the singular value decomposition $A = U\Sigma V^T$, which is very similar in nature. U and V will both be square orthogonal matrices, and Σ will be a diagonal matrix with the *singular values* of A.

Definition 1.2.1. The *singular values* of a matrix $A \in \mathbb{R}^{m\times n}$ will be denoted by $\sigma_1, \ldots, \sigma_{\min\{m,n\}}$. If $\min\{m, n\} = m$, then the singular values of A are the eigenvalues of $AA^T \in \mathbb{R}^{m\times m}$, but *square rooted*. If $\min\{m, n\} = n$, then the singular values of A are the eigenvalues of $A^TA \in \mathbb{R}^{n\times n}$, but *square rooted*.

Now we will show that the non-zero eigenvalues of AA^T and A^TA are the same. This will mean that it does not matter from which matrix we calculate $\sigma_1, \ldots, \sigma_{\min\{m,n\}}$, since, if the non-zero eigenvalues are the same, then the rest of the eigenvalues are zero for both of them. This gives us every eigenvalue needed, and therefore every singular value of A.

Lemma 1.2.2. *Given $\lambda \neq 0$, then λ is an eigenvalue of AA^T if and only if λ is an eigenvalue of A^TA.*

Proof. (if, $\Longleftarrow$). Let $\lambda \neq 0$ be an eigenvalue of A^TA. By definition of eigenvalue, there exists some vector $x \in \mathbb{R}^n$ with $x \neq \mathbf{0} \in \mathbb{R}^n$ such that

$$A^TAx = \lambda x \implies AA^TAx = \lambda Ax \implies AA^T(Ax) = \lambda(Ax). \tag{1.2.1}$$

Note now that $Ax \neq \mathbf{0}$, because if $Ax = \mathbf{0}$ then $A^TAx = \lambda x$ gives $A^T\mathbf{0} = \mathbf{0} = \lambda x \implies x = \mathbf{0}$ or $\lambda = 0$. Neither of these can be the case, so $Ax \neq \mathbf{0}$. Because also $AA^T(Ax) = \lambda(Ax)$, this means that Ax is an eigenvector to AA^T with the same eigenvalue $\lambda \neq 0$.

(only if, $\Longrightarrow$). Let $\lambda \neq 0$ be an eigenvalue of AA^T. By definition of eigenvalue, there exists some vector $y \in \mathbb{R}^m$ with $y \neq \mathbf{0} \in \mathbb{R}^m$ such that

$$AA^Ty = \lambda y \implies A^TAA^Ty = \lambda A^Ty \implies A^TA(A^Ty) = \lambda(A^Ty). \tag{1.2.2}$$

Note now that $A^Ty \neq \mathbf{0}$, because if $A^Ty = \mathbf{0}$ then $AA^Tx = \lambda x$ gives $A\mathbf{0} = \mathbf{0} = \lambda y \implies y = \mathbf{0}$ or $\lambda = 0$. Neither of these can be the case, so $A^Ty \neq \mathbf{0}$. Because also $A^TA(A^Ty) = \lambda(A^Ty)$, this means that A^Ty is an eigenvector to A^TA with the same eigenvalue $\lambda \neq 0$. ο.ε.δ.

Lemma 1.2.3. *The singular values of any matrix are non-negative real numbers, that can therefore be ordered: $\sigma_1 \geq \ldots \geq \sigma_{\min\{m,n\}} \geq 0$.*

Proof. By [4] (p233) the eigenvalues of A^TA are non-negative real numbers. Therefore, the non-zero singular values of A are positive, because the square root of a positive number is a positive number. The singular values of A which are not non-zero, are all equal to zero. We can therefore order them like

$$\sigma_1 \geq \ldots \geq \sigma_{\min\{m,n\}} \geq 0. \tag{1.2.3}$$

ο.ε.δ.

We can now define the pieces of the *singular value decomposition* for an arbitrary matrix $A \in \mathbb{R}^{m \times n}$.

Definition 1.2.4. Given $A \in \mathbb{R}^{m\times n}$, let the singular values of A be $\sigma_1 \geq \ldots \geq \sigma_{\min\{m,n\}} \geq 0$. The matrix $\Sigma \in \mathbb{R}^{m\times n}$ will be defined as the diagonal matrix with the singular values of A on the diagonal, in (not strictly) decreasing order. That is, for $1 \leq k \leq \min\{m,n\}$

$$(\Sigma)_{k,k} = \sigma_k,$$

with all other entries being zero.

Definition 1.2.5. Given $A \in \mathbb{R}^{m\times n}$, by definition 1.2.1 and lemma 1.2.2 we have that $\sigma_1^2 \geq \ldots \geq \sigma^2_{\min\{m,n\}} \geq 0$ are eigenvalues of AA^T, where the (if $\min\{m,n\} = n < m$) rest of the eigenvalues are zero. Let $u_1, \ldots, u_m$ be corresponding eigenvectors to every eigenvalue of AA^T, which by [4] (p214) can be chosen to be mutually orthogonal and of Euclidean length 1. Now we define

$$U = \begin{bmatrix} u_1 & \cdots u_{\min\{m,n\}} & \cdots & u_m \end{bmatrix} \in \mathbb{R}^{m\times m}. \tag{1.2.4}$$

Definition 1.2.6. Given $A \in \mathbb{R}^{m\times n}$, by definition 1.2.1 and lemma 1.2.2 we have that $\sigma_1^2 \geq \ldots \geq \sigma^2_{\min\{m,n\}} \geq 0$ are eigenvalues of A^TA, where the (if $\min\{m,n\} = m < n$) rest of the eigenvalues are zero. Let $v_1, \ldots, v_n$ be corresponding eigenvectors to every eigenvalue of A^TA, which by [4] (p214) can be chosen to be mutually orthogonal and of Euclidean length 1. Now we define

$$V = \begin{bmatrix} v_1 & \cdots v_{\min\{m,n\}} & \cdots & v_n \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{1.2.5}$$

Lemma 1.2.7. *A matrix $M \in \mathbb{R}^{m\times n}$ with mutually orthogonal vectors of Euclidean length 1 is an orthogonal matrix: $M^TM = I_n$.*

Proof. Let $w_1, \ldots, w_n$ be mutually orthogonal vectors in $\mathbb{R}^m$ of Euclidean length 1. Let

$$M = \begin{bmatrix} w_1 & \cdots & w_n \end{bmatrix} \in \mathbb{R}^{m\times n}. \tag{1.2.6}$$

By definition of matrix multiplication, we have that the entry in row $1 \leq i \leq n$ and column $1 \leq j \leq n$ of $M^TM \in \mathbb{R}^{n\times n}$ is

$$(M^TM)_{i,j} = w_i^T w_j. \tag{1.2.7}$$

By definition of mutually orthogonal $w_i^T w_j = 0$ if $i \neq j$. By definition of Euclidean length 1 we have $w_k^T w_k = \|w_k\|_2^2 = 1^2 = 1$. Therefore, for $1 \leq k \leq n$

$$(M^TM)_{k,k} = 1, \tag{1.2.8}$$

and the rest of the entries are zero. This is the definition of I_n. Therefore

$$M^TM = I_n, \tag{1.2.9}$$

meaning that M is orthogonal. ο.ε.δ.

Corollary 1.2.8. *$U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are square orthogonal matrices.*

Proof. This immediately follows from definitions 1.2.5 and 1.2.6 with lemma 1.2.7. o.ε.δ.

Definition 1.2.9. Let A be any matrix in $\mathbb{R}^{m\times n}$. By [4] (p214), the *singular value decomposition* (SVD) of A will always exist and be given by

$$A = U\Sigma V^T.$$

1.2.2 The compact SVD

The compact SVD is all about simplifying the SVD by only considering the positive singular values. It turns out that the number of positive singular values is exactly the rank of the matrix.

Lemma 1.2.10. *Given a matrix $A \in \mathbb{R}^{m\times n}$, then $\mathbf{rank}(A) = r$ if and only if*

$$\sigma_1 \geq \ldots \geq \sigma_r > 0 = \sigma_{r+1} = \ldots = \sigma_{\min\{m,n\}}.$$

Proof. Let the SVD of A be $A = U\Sigma V^T$ and let $\mathbf{rank}(A) = r$. Because U and V^T are invertible, they are full rank by theorem 1.1.11. Therefore

$$r = \mathbf{rank}(A) = \mathbf{rank}(U\Sigma V^T) = \mathbf{rank}(\Sigma). \tag{1.2.10}$$

The rank of Σ is the number of independent columns of Σ. Because it is a diagonal matrix, this is the number of non-zero columns of Σ, which is exactly the number of non-zero singular values of A. By lemma 1.2.3 this means that

$$\sigma_1 \geq \ldots \geq \sigma_r > 0 = \sigma_{r+1} = \ldots = \sigma_{\min\{m,n\}}. \tag{1.2.11}$$

o.ε.δ.

Now we can define the pieces of the *compact singular value decomposition* for an arbitrary matrix $A \in \mathbb{R}^{m\times n}$ of rank r.

Definition 1.2.11. Let Σ be as in definition 1.2.4. We now define $\Sigma_r \in \mathbb{R}^{r\times r}$ to be the matrix consisting of the first r rows and columns of Σ.

Remark. Because Σ is diagonal this means that Σ_r is also diagonal.

Remark. If $\mathbf{rank}(A) = r$ this means by lemma 1.2.10 that Σ_r is a block in Σ seen as a block matrix, given by

$$\Sigma = \begin{bmatrix} \Sigma_r & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

Definition 1.2.12. Let U be as in definition 1.2.5. We now define $U_r \in \mathbb{R}^{m\times r}$, where $1 \leq r \leq m$, to be the matrix consisting of the first r columns of U. That is

$$U_r = \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix} \in \mathbb{R}^{m\times r}.$$

Definition 1.2.13. Let V be as in definition 1.2.6. We now define $V_r \in \mathbb{R}^{n\times r}$, where $1 \leq r \leq n$, to be the matrix consisting of the first r columns of V. That is

$$V_r = \begin{bmatrix} v_1 & \cdots & v_r \end{bmatrix} \in \mathbb{R}^{n\times r}.$$

Corollary 1.2.14. *$U_r \in \mathbb{R}^{m\times r}$ and $V_r \in \mathbb{R}^{n\times r}$ are orthogonal matrices.*

Proof. This immediately follows from definition 1.2.12 and definition 1.2.13 with lemma 1.2.7. ο.ε.δ.

We will now see that the compact SVD given by $U_r\Sigma_r V_r^T$ is in fact equal to the standard SVD given by $U\Sigma V^T$.

Theorem 1.2.15. *Given $A \in \mathbb{R}^{m\times n}$ with $\mathbf{rank}(A) = r$, then $A = U\Sigma V^T = U_r\Sigma_r V_r^T$.*

Proof. Let $A \in \mathbb{R}^{m\times n}$ and $\mathbf{rank}(A) = r$. Let us express U, Σ and V as block matrices with U_r, Σ_r and V_r as blocks used within them.

$$U = \begin{bmatrix} U_r & U_{\text{other}} \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_r & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix}, \quad V = \begin{bmatrix} V_r & V_{\text{other}} \end{bmatrix}, \tag{1.2.12}$$

where $U_{\text{other}} \in \mathbb{R}^{m\times(m-r)}$ and $V_{\text{other}} \in \mathbb{R}^{n\times(n-r)}$. By direct computation

$$\begin{aligned} A = U\Sigma V^T &= \begin{bmatrix} U_r & U_{\text{other}} \end{bmatrix} \begin{bmatrix} \Sigma_r & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix} \begin{bmatrix} V_r & V_{\text{other}} \end{bmatrix}^T \\ &= \begin{bmatrix} U_r & U_{\text{other}} \end{bmatrix} \begin{bmatrix} \Sigma_r & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{\text{other}}^T \end{bmatrix} \\ &= \begin{bmatrix} U_r & U_{\text{other}} \end{bmatrix} \begin{bmatrix} \Sigma_r V_r^T + 0V_{\text{other}}^T \\ 0V_r^T + 0V_{\text{other}}^T \end{bmatrix} \\ &= \begin{bmatrix} U_r & U_{\text{other}} \end{bmatrix} \begin{bmatrix} \Sigma_r V_r^T \\ 0_{(m-r)\times n} \end{bmatrix} \\ &= U_r\Sigma_r V_r^T + U_{\text{other}} 0_{(m-r)\times n} \\ &= U_r\Sigma_r V_r^T. \end{aligned} \tag{1.2.13}$$

ο.ε.δ.

Definition 1.2.16. Let the SVD of $A \in \mathbb{R}^{m\times n}$ be $A = U\Sigma V^T$. Let $\mathbf{rank}(A) = r$. The *compact singular value decomposition* (compact SVD) of A will then be

$$A = U_r\Sigma_r V_r^T.$$

2 Least squares problems

This chapter is based upon section II.2 in Linear Algebra and Learning from Data by Gilbert Strang [12]. In this chapter we will let $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$.

2.1 Least squares

A recurring problem in linear algebra is to solve equations of the form $Ax = b$ with respect to x. If and only if a unique solution exists it takes the form $x = A^{-1}b$. Note that it is a requirement that A is square, meaning $m = n$, in order for A^{-1} to exist. Let us now define a *well-posed problem*.

Definition 2.1.1. The problem of solving an equation of the form $Ax = b$ with respect to x will be called a *well-posed problem* if there exists a unique solution.

In the complementary case we call it an *ill-posed problem*.

Definition 2.1.2. The problem of solving an equation of the form $Ax = b$ with respect to x will be called an *ill-posed problem* if there are no solutions or an infinite number of solutions.

Ill-posed problems of this form might still be very important to solve, at least approximately. For these cases we will use some version of the method of *least squares*. The idea is to find an $\hat{x}$ that minimizes the expression $\|b - A\hat{x}\|_2^2$. That is

$$\min_x \|b - Ax\|_2^2 = \|b - A\hat{x}\|_2^2. \tag{2.1.1}$$

This will be the closest approximate solution in terms of the squared Euclidean norm. There are many ways to find such an $\hat{x}$. In this report we will cover four ways, which are by the normal equation, pseudoinverse, QR-decomposition and regularization. This is covered in the following sections and the next chapter.

2.2 Normal equation

The normal equation is $A^T A\hat{x} = A^T b$. In this section we will show that when this has a unique solution it is the unique minimizer of $\|b - A\hat{x}\|_2^2$.

2.2.1 Deriving the normal equation

To find an $\hat{x}$ that minimizes $\|b - A\hat{x}\|_2^2$ we first define an appropriate function $f : \mathbb{R}^n \to \mathbb{R}$ by

$$\begin{aligned} f(x) &= \|b - Ax\|_2^2 \\ &= (b - Ax)^T(b - Ax) \\ &= (b^T - x^T A^T)(b - Ax) \\ &= b^T b - b^T Ax - x^T A^T b + x^T A^T Ax. \end{aligned} \tag{2.2.1}$$

It is then clear that the minimizer of this function is an $\hat{x}$ that minimizes the expression $\|b - A\hat{x}\|_2^2$. Let us find local extrema of f this by solving $\nabla f(\hat{x}) = 0$. By definition of matrix-multiplication we have

$$b^T b = \sum_{i=1}^{m} (b_i)^2, \tag{2.2.2}$$

$$b^T Ax = \sum_{i=1}^{m} \sum_{j=1}^{n} b_i (A)_{i,j} x_j, \tag{2.2.3}$$

$$x^T A^T b = \sum_{j=1}^{n} \sum_{i=1}^{m} x_j (A^T)_{j,i} b_i, \tag{2.2.4}$$

$$x^T A^T Ax = \sum_{j=1}^{n} \sum_{i=1}^{m} \sum_{k=1}^{n} x_j (A^T)_{j,i} (A)_{i,k} x_k. \tag{2.2.5}$$

We can now use standard derivative rules to compute the gradient $\nabla f(x)$. This is a vector with entry t being $(\nabla f(x))_t = \frac{\partial f}{\partial x_t}(x)$. By the linearity of the derivative operator

$$\frac{\partial f}{\partial x_t}(x) = \frac{\partial}{\partial x_t} b^T b - \frac{\partial}{\partial x_t} b^T Ax - \frac{\partial}{\partial x_t} x^T A^T b + \frac{\partial}{\partial x_t} x^T A^T Ax. \tag{2.2.6}$$

Using the equations above, these terms can then be computed by

$$\frac{\partial}{\partial x_t} b^T b = 0, \tag{2.2.7}$$

$$\frac{\partial}{\partial x_t} b^T Ax = \sum_{i=1}^{m} b_i (A)_{i,t} = \sum_{i=1}^{m} b_i (A^T)_{t,i} = \sum_{i=1}^{m} (A^T)_{t,i} b_i = (A^T b)_t, \tag{2.2.8}$$

$$\frac{\partial}{\partial x_t} x^T A^T b = \sum_{i=1}^{m} (A^T)_{t,i} b_i = (A^T b)_t. \tag{2.2.9}$$

For $\frac{\partial}{\partial x_t} x^T A^T A x$ we break it down into cases. The following indices will refer to the indices used in equation (2.2.5). Terms with $j \neq t$ and $k \neq t$ contribute 0. Terms with $j = t$ and $k \neq t$ contribute

$$\sum_{i=1}^{m} \sum_{k=1}^{n} (A^T)_{t,i} (A)_{i,k} x_k - \sum_{i=1}^{m} (A^T)_{t,i} (A)_{i,t} x_t = (A^T A x)_t - \sum_{i=1}^{m} (A^T)_{t,i} (A)_{i,t} x_t. \quad (2.2.10)$$

Terms with $j \neq t$ and $k = t$ contribute

$$\begin{aligned}
&\sum_{j=1}^{n} \sum_{i=1}^{m} x_j (A^T)_{j,i} (A)_{i,t} - \sum_{i=1}^{m} x_t (A^T)_{t,i} (A)_{i,t} \\
=&\sum_{j=1}^{n} \sum_{i=1}^{m} (A^T)_{j,i} (A)_{i,t} x_j - \sum_{i=1}^{m} (A^T)_{t,i} (A)_{i,t} x_t \\
=&\sum_{j=1}^{n} \sum_{i=1}^{m} (A)_{i,j} (A^T)_{t,i} x_j - \sum_{i=1}^{m} (A^T)_{t,i} (A)_{i,t} x_t \\
=&\sum_{j=1}^{n} \sum_{i=1}^{m} (A^T)_{t,i} (A)_{i,j} x_j - \sum_{i=1}^{m} (A^T)_{t,i} (A)_{i,t} x_t \\
=&(A^T A x)_t - \sum_{i=1}^{m} (A^T)_{t,i} (A)_{i,t} x_t.
\end{aligned} \quad (2.2.11)$$

Note that this expression equals the expression for the prior case. Terms with $j = t$ and $k = t$ contribute

$$2 \sum_{i=1}^{m} (A^T)_{t,i} (A)_{i,t} x_t.$$

These are all the cases, therefore

$$\begin{aligned}
\frac{\partial}{\partial x_t} x^T A^T A x &= 2(A^T A x)_t - 2 \sum_{i=1}^{m} (A^T)_{t,i} (A)_{i,t} x_t + 2 \sum_{i=1}^{m} (A^T)_{t,i} (A)_{i,t} x_t \\
&= 2(A^T A x)_t.
\end{aligned} \quad (2.2.12)$$

Using equation (2.2.6) we arrive at

$$\begin{aligned}
(\nabla f(x))_t = \frac{\partial f}{\partial x_t}(x) &= 0 - (A^T b)_t - (A^T b)_t + 2(A^T A x)_t \\
&= 2(A^T A x)_t - 2(A^T b)_t.
\end{aligned} \quad (2.2.13)$$

This means that

$$\nabla f(x) = 2A^T A x - 2A^T b. \quad (2.2.14)$$

We can now find local extremizers $\hat{x}$ by setting this equal to $\mathbf{0}$ and solving.

$$\begin{aligned} \nabla f(\hat{x}) = 2A^TA\hat{x} - 2A^Tb = \mathbf{0} \\ \iff A^TA\hat{x} = A^Tb. \end{aligned} \tag{2.2.15}$$

This is the *normal equation*.

2.2.2 Analyzing the normal equation

To solve the normal equation $A^TA\hat{x} = A^Tb$ is a well-posed problem if and only if A^TA is invertible. Note that $A^TA \in \mathbb{R}^{n\times n}$, so the normal equation is a square system (of equations). This is necessary, but not sufficient for A^TA to be invertible. However, if the null space of A^TA is trivial it is sufficient.

Lemma 2.2.1. $\mathbf{N}(A^TA) = \mathbf{N}(A)$.

Proof. Let $x \in \mathbf{N}(A^TA)$. By definition of the null space and the Euclidean norm

$$\begin{aligned} A^TAx = \mathbf{0} &\implies x^TA^TAx = 0 \\ &\iff (Ax)^T(Ax) = 0 \\ &\iff \|Ax\|_2^2 = 0 \\ &\iff \|Ax\|_2 = 0 \\ &\implies Ax = \mathbf{0} \\ &\iff x \in \mathbf{N}(A) \\ &\implies \mathbf{N}(A^TA) \subseteq \mathbf{N}(A). \end{aligned} \tag{2.2.16}$$

Let $x \in \mathbf{N}(A)$. By definition of the null space

$$\begin{aligned} Ax = \mathbf{0} &\implies A^TAx = \mathbf{0} \\ &\iff x \in \mathbf{N}(A^TA) \\ &\implies \mathbf{N}(A) \subseteq \mathbf{N}(A^TA) \end{aligned} \tag{2.2.17}$$

By equation (2.2.16) and (2.2.17) we have

$$\mathbf{N}(A^TA) \subseteq \mathbf{N}(A) \subseteq \mathbf{N}(A^TA) \iff \mathbf{N}(A^TA) = \mathbf{N}(A). \tag{2.2.18}$$

ο.ε.δ.

Theorem 2.2.2. *A^TA is invertible if and only if $\mathbf{N}(A)$ is trivial.*

Proof. Let $\mathbf{N}(A)$ be trivial. By lemma 2.2.1 this means that $\mathbf{N}(A^TA)$ is trivial. By the rank-nullity theorem A^TA has full rank. Since A^TA also is square, it is then invertible.

Let A^TA be invertible. This means that $\mathbf{N}(A^TA)$ is trivial. By lemma 2.2.1 $\mathbf{N}(A)$ is then also trivial. o.ε.δ.

Corollary 2.2.3. *The normal equation $A^TA\hat{x} = A^Tb$ has a unique solution if and only if $\mathbf{N}(A)$ is trivial.*

Proof. This immediately follows from theorem 2.2.2. o.ε.δ.

Theorem 2.2.4. *If $\hat{x} = (A^TA)^{-1}A^Tb$ exists, it is the unique minimizer of $\|b - A\hat{x}\|_2^2$.*

Proof. Let $\hat{x} = (A^TA)^{-1}A^Tb$ exist. This is a minimizer if $\|b - A\hat{x}\|_2^2 \leq \|b - Ax\|_2^2$ for each $x \in \mathbb{R}^n$. Let $r(x) = b - Ax$. Therefore equivalently we wish to show that $\|r(\hat{x})\|_2^2 \leq \|r(x)\|_2^2$ for each $x \in \mathbb{R}^n$. Note that

$$\begin{aligned} r(x) &= b - Ax \\ &= b - A\hat{x} + A\hat{x} - Ax \\ &= r(\hat{x}) + A(\hat{x} - x). \end{aligned} \tag{2.2.19}$$

By the definition of the Euclidean norm

$$\begin{aligned} &\|r(x)\|_2^2 \\ &= r(x)^T r(x) \\ &= (r(\hat{x}) + A(\hat{x} - x))^T (r(\hat{x}) + A(\hat{x} - x)) \\ &= (r(\hat{x})^T + (\hat{x} - x)^T A^T)(r(\hat{x}) + A(\hat{x} - x)) \\ &= r(\hat{x})^T r(\hat{x}) + r(\hat{x})^T A(\hat{x} - x) + (\hat{x} - x)^T A^T r(\hat{x}) + (\hat{x} - x)^T A^T A(\hat{x} - x) \\ &= \|r(\hat{x})\|_2^2 + r(\hat{x})^T A(\hat{x} - x) + (\hat{x} - x)^T A^T r(\hat{x}) + \|A(\hat{x} - x)\|_2^2. \end{aligned} \tag{2.2.20}$$

Note now that

$$\begin{aligned} A^T r(\hat{x}) &= A^T(b - A\hat{x}) \\ &= A^Tb - A^TA(\hat{x}) \\ &= A^Tb - A^TA(A^TA)^{-1}A^Tb \\ &= A^Tb - A^Tb \\ &= \mathbf{0}, \end{aligned} \tag{2.2.21}$$

and therefore

$$r(\hat{x})^T A = (A^T r(\hat{x}))^T = \mathbf{0}^T. \tag{2.2.22}$$

By equation (2.2.20)

$$\begin{aligned}\|r(x)\|_2^2 &= \|r(\hat{x})\|_2^2 + r(\hat{x})^T A(\hat{x}-x) + (\hat{x}-x)^T A^T r(\hat{x}) + \|A(\hat{x}-x)\|_2^2 \\ &= \|r(\hat{x})\|_2^2 + \mathbf{0}^T(\hat{x}-x) + (\hat{x}-x)^T\,\mathbf{0} + \|A(\hat{x}-x)\|_2^2 \\ &= \|r(\hat{x})\|_2^2 + \|A(\hat{x}-x)\|_2^2 \\ &\geq \|r(\hat{x})\|_2^2.\end{aligned} \tag{2.2.23}$$

Hence, $\hat{x} = (A^TA)^{-1}A^Tb$ is a minimizer of $\|b - A\hat{x}\|_2^2$ when it exists. Since it is the unique solution to the normal equation (2.2.15) it is the unique stationary point of $f(x) = \|b - Ax\|_2^2$. It is therefore the unique minimizer of $\|b - A\hat{x}\|_2^2$. ο.ε.δ.

2.3 Pseudoinverse

In this section we will develop some theory for the pseudoinverse of a matrix A, which will be denoted by A^+. This *always exists* for every matrix A and we will show that when a unique solution to the normal equation exists, then $\hat{x} = A^+b$ is equal to that solution.

2.3.1 Definition of the pseudoinverse

The definition of the pseudoinverse builds upon the SVD explored in section 1.2. We have that every matrix $A \in \mathbb{R}^{m\times n}$ can be expressed as

$$A = U\Sigma V^T, \tag{2.3.1}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are both orthogonal matrices and $\Sigma \in \mathbb{R}^{m\times n}$ is diagonal. Let the rank of A be $r \leq \min\{m, n\}$, meaning that the compact SVD of A is

$$A = U_r\Sigma_r V_r^T, \tag{2.3.2}$$

where $U_r \in \mathbb{R}^{m\times r}$ and $V_r \in \mathbb{R}^{n\times r}$ are the first r column of U and V, and $\Sigma_r \in \mathbb{R}^{r\times r}$ are the first r columns and rows of Σ. We first define what the pseudoinverse of the diagonal matrix Σ is. For $1 \leq k \leq r$ it is the case that $(\Sigma)_{k,k} = \sigma_k > 0$, and all other entries of Σ are 0.

Definition 2.3.1. The *pseudoinverse* of Σ will be denoted by $\Sigma^+ \in \mathbb{R}^{n\times m}$ which for $1 \leq k \leq r$ will be defined by

$$(\Sigma^+)_{k,k} = \frac{1}{\sigma_k},$$

with all other entries being 0.

Example 2.3.2. Let $A \in \mathbb{R}^{4\times 3}$, with $\mathbf{rank}(A) = 2$ where $A = U\Sigma V^T$ is the SVD of A. Then

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad \Sigma^+ = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Definition 2.3.3. The *pseudoinverse* of any matrix $A \in \mathbb{R}^{m\times n}$ will be

$$A^+ = V\Sigma^+ U^T \in \mathbb{R}^{n\times m}.$$

2.3.2 Properties of the pseudoinverse

Using the definitions above we now wish to derive interesting and important statements about the pseudoinverse.

Lemma 2.3.4. *Σ^+ is the block matrix*

$$\Sigma^+ = \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times m}.$$

Proof. Because $(\Sigma)_{k,k} = \sigma_k > 0$ for $1 \leq k \leq r$ with all other entries being 0 it is the case that $(\Sigma_r)_{k,k} = \sigma_k > 0$ with all other entries being 0. Because Σ_r then is diagonal its inverse $\Sigma_r^{-1} \in \mathbb{R}^{r\times r}$ is

$$(\Sigma_r^{-1})_{k,k} = \frac{1}{\sigma_k} > 0, \tag{2.3.3}$$

for $1 \leq k \leq r$, with all other entries being 0. If we pad Σ_r^{-1} by adding columns and rows of zeros until it has m rows and n columns, then this new matrix is the block matrix

$$\begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times m}.$$

Note that it has entries $\frac{1}{\sigma_k} > 0$ with $1 \leq k \leq r$ for the r first entries of its main diagonal, with all other entries being 0. By definition 2.3.1 this is the pseudoinverse Σ^+. ο.ε.δ.

Corollary 2.3.5. *$\Sigma^+\Sigma$ and $\Sigma\Sigma^+$ are the matrices*

$$\Sigma^+\Sigma = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times n}, \tag{2.3.4}$$

$$\Sigma\Sigma^+ = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{m\times m}, \tag{2.3.5}$$

where I_r is the $r \times r$ identity matrix.

Proof. This immediately follows from lemma 2.3.4. ο.ε.δ.

Theorem 2.3.6. $A^+ = V_r \Sigma_r^{-1} U_r^T$. *(This is the compact pseudoinverse.)*

Proof. By definition 2.3.3 we have $A^+ = U\Sigma^+ V^T$. By lemma 2.3.4

$$\begin{aligned} A^+ &= U\Sigma^+ V^T \\ &= U \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} V^T \\ &= U \begin{bmatrix} \Sigma_r^{-1} V_r^T \\ 0 \end{bmatrix} \\ &= U_r \Sigma_r^{-1} V_r^T. \end{aligned} \tag{2.3.6}$$

ο.ε.δ.

Let us use this to determine what A^+A and AA^+ is.

Corollary 2.3.7. $A^+A = V_r V_r^T$.

Proof. Using theorem 2.3.6 we have

$$\begin{aligned} A^+A &= V_r \Sigma_r^{-1} U_r^T U_r \Sigma_r V_r^T \\ &= V_r \Sigma_r^{-1} \Sigma_r V_r^T \\ &= V_r V_r^T. \end{aligned} \tag{2.3.7}$$

ο.ε.δ.

Corollary 2.3.8. $AA^+ = U_r U_r^T$.

Proof. Using theorem 2.3.6 we have

$$\begin{aligned} AA^+ &= U_r \Sigma_r V_r^T V_r \Sigma_r^{-1} U_r^T \\ &= U_r \Sigma_r^{-1} \Sigma_r U_r^T \\ &= U_r U_r^T. \end{aligned} \tag{2.3.8}$$

ο.ε.δ.

Theorem 2.3.9. *If A has rank n then $A^+A = I_n$. (This means that A^+ is a left inverse of A.)*

Proof. Let A have rank $r = n$. Therefore $V_r = V_n = V$. From corollary 2.3.7 it immediately follows that $A^+A = VV^T = I_n$. ο.ε.δ.

Theorem 2.3.10. *If A has rank m then $AA^+ = I_m$. (This means that A^+ is a right inverse of A.)*

Proof. Let A have rank $r = m$. Therefore $U_r = U_m = U$. From corollary 2.3.8 it immediately follows that $AA^+ = UU^T = I_m$. o.ε.δ.

Theorem 2.3.11. *$A^+Ax = x$ if and only if x is in the row space $\mathbf{C}(A^T)$.*

Proof. (if, $\Longleftarrow$). Let $x \in \mathbf{C}(A^T)$ and A have rank r. By the compact SVD we have that $A^T = (U_r\Sigma_r V_r^T)^T = V_r\Sigma_r^T U_r^T = V_r\Sigma_r U_r^T$. By the definition of the column space $x = A^T y$ for some $y \in \mathbb{R}^m$, so by corollary 2.3.7

$$\begin{aligned} A^+Ax &= A^+AA^Ty \\ &= V_rV_r^TV_r\Sigma_rU_r^Ty \\ &= V_r\Sigma_rU_r^Ty \\ &= A^Ty \\ &= x. \end{aligned} \tag{2.3.9}$$

(only if, $\Longrightarrow$). Let $A^+Ax = x$. Therefore

$$\begin{aligned} x &= A^+Ax \\ &= V_rV_r^Tx \\ &= V_r(\Sigma_rU_r^TU_r\Sigma_r^{-1})V_r^Tx \\ &= (V_r\Sigma_rU_r^T)(U_r\Sigma_r^{-1}V_r^Tx) \\ &= A^Ty \end{aligned} \tag{2.3.10}$$

with $y = U_r\Sigma_r^{-1}V_r^Tx \in \mathbb{R}^m$, meaning that $x \in \mathbf{C}(A^T)$. o.ε.δ.

Theorem 2.3.12. *$AA^+y = y$ if and only if y is in the column space $\mathbf{C}(A)$.*

Proof. (if, $\Longleftarrow$). Let $y \in \mathbf{C}(A)$ and A have rank r. By definition of the column space $y = Ax$ for some $x \in \mathbb{R}^n$, so by corollary 2.3.8

$$\begin{aligned} AA^+y &= AA^+Ax \\ &= U_rU_r^TU_r\Sigma_rV_r^Tx \\ &= U_r\Sigma_rV_r^Tx \\ &= Ax \\ &= y. \end{aligned} \tag{2.3.11}$$

(only if, $\implies$). Let $AA^+y = y$. Therefore

$$\begin{aligned} y &= AA^+y \\ &= U_rU_r^Ty \\ &= U_r(\Sigma_rV_r^TV_r\Sigma_r^{-1})U_r^Ty \\ &= (U_r\Sigma_rV_r^T)(V_r\Sigma_r^{-1}U_r^Ty) \\ &= Ax \end{aligned} \tag{2.3.12}$$

with $x = V_r\Sigma_r^{-1}U_r^Ty \in \mathbb{R}^n$, meaning that $y \in \mathbf{C}(A)$. o.ε.δ.

2.3.3 The pseudoinverses relation to the normal equation

When minimizing $\|b - A\hat{x}\|_2^2$ we arrive at the normal equation $A^TA\hat{x} = A^Tb$. From theorem 2.2.2 we know that this is a well-posed problem exactly when $\mathbf{N}(A)$ is trivial, with solution $\hat{x} = (A^TA)^{-1}A^Tb$. We will show that if this is the case, then $\hat{x} = A^+b$ is also the solution.

Theorem 2.3.13. *If $\mathbf{N}(A)$ is trivial then $A^+ = (A^TA)^{-1}A^T$.*

Proof. Let $\mathbf{N}(A)$ be trivial. By the rank-nullity theorem $\mathbf{rank}(A) = n$. Let $A = U_n\Sigma_nV_n^T = U_n\Sigma_nV^T$ be the compact SVD of A. Then

$$\begin{aligned} (A^TA)^{-1}A^T &= ((U_n\Sigma_nV^T)^T(U_n\Sigma_nV^T))^{-1}(U_n\Sigma_nV^T)^T \\ &= (V\Sigma_n^TU_n^TU_n\Sigma_nV^T)^{-1}V\Sigma_n^TU_n^T \\ &= (V\Sigma_n\Sigma_nV^T)^{-1}V\Sigma_nU_n^T \\ &= (V\Sigma_n^{-2}V^T)V\Sigma_nU_n^T \\ &= V\Sigma_n^{-2}\Sigma_nU_n^T \\ &= V\Sigma_n^{-1}U_n^T \\ &= V\Sigma^+U^T, \end{aligned} \tag{2.3.13}$$

by padding Σ_n^{-1} with $m-n$ rows and columns of zeros, meaning that the last $m-n$ rows of U^T do not contribute to the product. o.ε.δ.

2.4 QR-decomposition

This section is about ways to attain the QR-decomposition of a matrix $A = QR$ in a numerically stable way, with $Q \in \mathbb{R}^{m\times m}$ being an orthogonal matrix and $R \in \mathbb{R}^{m\times n}$ being upper-triangular. We may use this in order to compute the solution to the normal equation when a unique solution exists. In that case we have that $\hat{x} = R_1^{-1}Q_1^Tb$. (See definition 2.4.1.)

2.4.1 QR-decompositions relation to the normal equation

First we wish to use the QR-decomposition of the matrix $A \in \mathbb{R}^{m\times n}$ to solve $Ax = b$. If we QR-decompose A as $A = QR$, then

$$Ax = b \iff QRx = b \iff Rx = Q^T b. \tag{2.4.1}$$

Note that R is upper-triangular. This is then the same thing as using Gauss-elimination to reach row echelon form, where back substitution can be used to solve the system. However, this only has a unique solution when A is invertible, meaning that A must be square and $N(A)$ must be trivial.

In the case when $A \in \mathbb{R}^{m\times n}$ we instead consider the least squares problem $\min_x \|b - Ax\|_2^2$, where by definition of the normal equation, minimizers $\hat{x}$ must satisfy $A^T A\hat{x} = A^T b$. From theorem 2.2.2 we know that solving the normal equation $A^T A\hat{x} = A^T b$ is a well-posed problem exactly when $\mathbf{N}(A)$ is trivial, with solution $\hat{x} = (A^T A)^{-1}A^T b$. Note that if $\mathbf{N}(A)$ is trivial, by the rank-nullity theorem A has full rank $\mathbf{rank}(A) = n$. This means that $m \geq n$.

Now, in this case when $m \geq n$, we will relate the QR-decomposition of A to the normal equation $A^T A\hat{x} = A^T b$.

Definition 2.4.1. Let $A = QR$ be a QR-decomposition of $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, where $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $R \in \mathbb{R}^{m\times n}$ is upper-triangular. We now define $R_1 \in \mathbb{R}^{n\times n}$ to be the first n *rows* of R, and $Q_1 \in \mathbb{R}^{m\times n}$ to be the first n *columns* of Q and $Q_2 \in \mathbb{R}^{m\times(m-n)}$ to be the last $m - n$ columns of Q. That is

$$Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}, \quad \text{and} \quad R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}.$$

Lemma 2.4.2. *If $A = QR$ is a QR-decomposition of $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, then $A = Q_1 R_1$.*

Proof. By definition 2.4.1 we have that

$$A = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0_{(m-n)\times n} \end{bmatrix} = Q_1 R_1 + Q_2 0_{(m-n)\times n} = Q_1 R_1. \tag{2.4.2}$$

ο.ε.δ.

We will show that when the normal equation has a unique solution, which we know is given by $\hat{x} = (A^T A)^{-1}A^T b$, then $\hat{x} = R_1^{-1}Q_1^T b$ is also the solution. Therefore we wish to show in that case that $R_1^{-1}Q_1^T = (A^T A)^{-1}A^T$. By theorem 2.2.2 we know that this case is exactly when $\mathbf{N}(A)$ is trivial.

Theorem 2.4.3. *If* $\mathbf{N}(A)$ *is trivial then* $R_1^{-1}Q_1^T = (A^TA)^{-1}A^T$*, where* $A = QR$ *is a QR-decomposition of* A*.*

Proof. Let $A \in \mathbb{R}^{m\times n}$ and $\mathbf{N}(A)$ be trivial. By the rank-nullity theorem we have that $\mathbf{rank}(A) = n$. Because the matrix $Q \in \mathbb{R}^{m\times m}$ is orthogonal and square, it is also invertible. This means that Q has full rank. Because $A = QR$ we have that $\mathbf{rank}(R) = \mathbf{rank}(A) = n$.

Since all rows in R are only zeros, except those in R_1, this means that the n rows of $R_1 \in \mathbb{R}^{n\times n}$ must be linearly independent for $\mathbf{rank}(R) = n$. Therefore $\mathbf{rank}(R_1) = n$. This means that R_1 is invertible, and so $R_1^T \in \mathbb{R}^{n\times n}$ is invertible because it then must have n linearly independent columns. Using $A = QR$ and lemma 2.4.2 we have

$$\begin{aligned}
(A^TA)^{-1}A^T &= ((QR)^T(QR))^{-1}(QR)^T \\
&= (R^TQ^TQR)^{-1}(QR)^T \\
&= (R^TR)^{-1}(QR)^T \\
&= (R^TR)^{-1}(Q_1R_1)^T \\
&= \left(\begin{bmatrix} R_1 \\ 0 \end{bmatrix}^T \begin{bmatrix} R_1 \\ 0 \end{bmatrix}\right)^{-1} R_1^TQ_1^T \\
&= \left(\begin{bmatrix} R_1^T & 0^T \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}\right)^{-1} R_1^TQ_1^T \\
&= (R_1^TR_1 + 0^T0)^{-1}R_1^TQ_1^T \\
&= (R_1^TR_1)^{-1}R_1^TQ_1^T \\
&= R_1^{-1}(R_1^T)^{-1}(R_1^T)Q_1^T \\
&= R_1^{-1}I_nQ_1^T \\
&= R_1^{-1}Q_1^T.
\end{aligned} \tag{2.4.3}$$

ο.ε.δ.

2.4.2 Householder reflections

The idea with Householder reflections is to create t matrices $H_1, \ldots, H_t$ such that when multiplying a matrix $A' \in \mathbb{R}^{m\times n}$ with H_k from the left, then this results in a new matrix that has all entries as zero below its main diagonal in the k:th column.

We will have $H_t \ldots H_1A = R \in \mathbb{R}^{m\times n}$, where $t = \min\{m-1, n\}$, forced to be an upper-triangular matrix. This will give rise to a QR-decomposition of A. We first develop some theory about *Householder reflection matrices*, which will be used in order to construct the matrices $H_1, \ldots, H_t$.

Definition 2.4.4. A $m \times m$ *Householder reflection matrix* will be a matrix of the form

$$H_v = I_m - 2\frac{vv^T}{\|v\|_2^2} = I_m - 2uu^T,$$

where $v \in \mathbb{R}^m$, or $u \in \mathbb{R}^m$ with $\|u\|_2 = 1$.

Lemma 2.4.5. *Householder reflection matrices H are symmetric: $H^T = H$.*

Proof. Let $H = I_m - 2uu^T$. We have that

$$H^T = (I_m - 2uu^T)^T = I_m^T - 2(u^T)^T u^T = I_m - 2uu^T = H. \tag{2.4.4}$$

o.ε.δ.

Lemma 2.4.6. *Householder reflection matrices H are orthogonal: $H^T H = I_m$.*

Proof. Let $H = I_m - 2uu^T$. Note that $\|u\|_2 = 1$. We have that

$$\begin{aligned} H^T H = H^2 &= (I_m - 2uu^T)(I_m - 2uu^T) \\ &= I_m^2 - 4uu^T + 4uu^T uu^T \\ &= I_m - 4uu^T + 4u(\|u\|_2^2)u^T \\ &= I_m - 4uu^T + 4u(1^2)u^T \\ &= I_m - 4uu^T + 4uu^T \\ &= I_m. \end{aligned} \tag{2.4.5}$$

o.ε.δ.

Corollary 2.4.7. *Householder reflection matrices are involutory: $H^{-1} = H$.*

Proof. By lemma 2.4.6 and lemma 2.4.5 we have that

$$I_m = H^T H = HH \implies H^{-1} = H. \tag{2.4.6}$$

o.ε.δ.

Lemma 2.4.8. *$H_v y$ is the reflection of y in the hyperplane through $\mathbf{0}$ with normal vector v. (This is where the name Householder* reflection *matrix comes from.)*

Proof. Omitted. See section A.2 of appendix A on page 81. o.ε.δ.

What we now wish to be able to do is, given a vector a, find the reflection such that a is reflected to another vector r. Because this is a reflection, note that they have to have the same length $\|a\|_2 = \|r\|_2$. It turns out that this reflection is given by the matrix H_v where $v = a - r$.

Theorem 2.4.9. *If $v = a - r$ and $\|a\|_2 = \|r\|_2$ then $H_v a = r$.*

Proof. Let $v = a - r$ and $\|a\|_2 = \|r\|_2$. Note first that

$$\|a\|_2 = \|r\|_2 \iff \|a\|_2^2 = \|r\|_2^2 \iff a^T a = r^T r. \tag{2.4.7}$$

Note further that

$$a^T r = r^T a \tag{2.4.8}$$

holds for every pair of vectors $a \in \mathbb{R}^m$, $r \in \mathbb{R}^m$. We have that

$$\begin{aligned}
H_v a &= \left(I_m - 2\frac{vv^T}{\|v\|_2^2} \right) a \\
&= I_m a - 2\frac{(a-r)(a-r)^T a}{\|a-r\|_2^2} \\
&= a - 2\frac{(a-r)(a-r)^T a}{(a-r)^T(a-r)} \\
&= a - 2\frac{(a-r)(a^T - r^T)a}{(a^T - r^T)(a-r)} \\
&= a - 2\frac{(aa^T - ar^T - ra^T + rr^T)a}{a^T a - a^T r - r^T a + r^T r} \\
&= a - 2\frac{aa^T a - ar^T a - ra^T a + rr^T a}{a^T a - r^T a - r^T a + a^T a} \\
&= a - 2\frac{a(a^T a) - a(r^T a) - r(a^T a) + r(r^T a)}{2(a^T a) - 2(r^T a)} \\
&= a - \frac{a(a^T a - r^T a) - r(a^T a + r^T a)}{a^T a - r^T a} \\
&= a - (a - r) \\
&= r.
\end{aligned} \tag{2.4.9}$$

ο.ε.δ.

Definition 2.4.10. If

$$a = \begin{bmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_j \\ \vdots \\ a_m \end{bmatrix} \in \mathbb{R}^m,$$

then let the vector from row i to row j be

$$a_{i\to j} = \begin{bmatrix} a_i \\ \vdots \\ a_j \end{bmatrix} \in \mathbb{R}^{j-i+1}.$$

Definition 2.4.11. Let $e_i \in \mathbb{R}^j$ be the vector in $\mathbb{R}^j$ with all zeros as entries, except a one at entry i.

We now wish to use Householder reflection matrices in order to transform vectors

$$a_{k\to m} = \begin{bmatrix} a_k \\ a_{k+1} \\ \vdots \\ a_m \end{bmatrix}, \quad \text{to vectors} \quad \beta e_1 = \begin{bmatrix} \beta \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{m-k+1}.$$

Note that because this is a reflection it is required that

$$\|\beta e_1\|_2 = \|a_{k\to m}\|_2 \implies \beta = \pm\|a_{k\to m}\|_2. \tag{2.4.10}$$

By theorem 2.4.9 we can now construct the Householder reflection matrix that transforms $a_{k\to m}$ to βe_1^{m-k+1}.

Definition 2.4.12. Given a matrix $A' \in \mathbb{R}^{m\times n}$ with column vectors $a^1, \ldots, a^n$, let H_k, where $1 \le k \le \min\{m, n-1\}$, be the block matrix

$$H_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix} \in \mathbb{R}^{m\times m},$$

where

$$v = (a^k)_{k\to m} - \beta e_1 \in \mathbb{R}^{m-k+1},$$

with $\beta = \pm\|(a^k)_{k\to m}\|_2$, and $H_v = I_{m-k+1} - 2\frac{vv^T}{\|v\|_2^2}$.

Remark. For $k = 1$, it is meant that $H_1 = H_v$.

Remark. In numerical implementations one takes $\operatorname{sign}(\beta) = \operatorname{sign}(a_k^k)$.

Example 2.4.13. Let

$$A = \begin{bmatrix} -2 & 3 & 5 \\ 5 & 6 & -1 \\ -8 & 3 & 3 \\ 4 & -6 & 5 \end{bmatrix}, \tag{2.4.11}$$

and we will calculate H_2A. We have that

$$(a^2)_{2\to 4} = \begin{bmatrix} 3 \\ 6 \\ 3 \\ -6 \end{bmatrix}_{2\to 4} = \begin{bmatrix} 6 \\ 3 \\ -6 \end{bmatrix}. \tag{2.4.12}$$

We arbitrarily decide that β will be positive.

$$\beta = +\sqrt{(6)(6) + (3)(3) + (-6)(-6)} = \sqrt{81} = 9. \tag{2.4.13}$$

Therefore

$$v = \begin{bmatrix} 6 \\ 3 \\ -6 \end{bmatrix} - \begin{bmatrix} 9 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -3 \\ 3 \\ -6 \end{bmatrix}, \tag{2.4.14}$$

so

$$\|v\|_2^2 = (-3)(-3) + (3)(3) + (-6)(-6) = 54, \tag{2.4.15}$$

and

$$vv^T = \begin{bmatrix} -3 \\ 3 \\ -6 \end{bmatrix} \begin{bmatrix} -3 & 3 & -6 \end{bmatrix} = \begin{bmatrix} 9 & -9 & 18 \\ -9 & 9 & -18 \\ 18 & -18 & 36 \end{bmatrix}. \tag{2.4.16}$$

Hence

$$\begin{aligned} H_v &= I_{4-2+1} - 2\frac{vv^T}{\|v\|_2^2} = I_3 - \frac{2}{54}vv^T = I_3 - \frac{1}{27}vv^T \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{27}\begin{bmatrix} 9 & -9 & 18 \\ -9 & 9 & -18 \\ 18 & -18 & 36 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{3}\begin{bmatrix} 1 & -1 & 2 \\ -1 & 1 & -2 \\ 2 & -2 & 4 \end{bmatrix} \\ &= \frac{1}{3}\begin{bmatrix} 3-1 & 1 & -2 \\ 1 & 3-1 & 2 \\ -2 & 2 & 3-4 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 & 1 & -2 \\ 1 & 2 & 2 \\ -2 & 2 & -1 \end{bmatrix}. \end{aligned} \tag{2.4.17}$$

This means that

$$H_2 = \begin{bmatrix} I_{2-1} & 0 \\ 0 & H_v \end{bmatrix} = \begin{bmatrix} I_1 & 0 \\ 0 & H_v \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} & -\frac{2}{3} \\ 0 & \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ 0 & -\frac{2}{3} & \frac{2}{3} & -\frac{1}{3} \end{bmatrix}. \tag{2.4.18}$$

We can now calculate

$$H_2A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} & -\frac{2}{3} \\ 0 & \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ 0 & -\frac{2}{3} & \frac{2}{3} & -\frac{1}{3} \end{bmatrix} \begin{bmatrix} -2 & 3 & 5 \\ 5 & 6 & -1 \\ -8 & 3 & 3 \\ 4 & -6 & 5 \end{bmatrix} = \begin{bmatrix} -2 & 3 & 5 \\ -2 & 9 & -3 \\ -1 & 0 & 5 \\ -10 & 0 & 1 \end{bmatrix}. \tag{2.4.19}$$

Note that the entries below the main diagonal in column $k = 2$ now are only zeros. Note further that the row(s) above row $k = 2$ are not altered from A. These are key insights that we will use to create an upper-triangular matrix from any matrix A.

Lemma 2.4.14. *Given $A' \in \mathbb{R}^{m\times n}$, the first $k-1$ rows of H_kA' are the same as the first $k-1$ rows of A'. (Note that H_k is as in definition 2.4.12.)*

Proof. Let the rows of $A' \in \mathbb{R}^{m\times n}$ be the row vectors $a^{1*}, \dots, a^{m*} \in \mathbb{R}^{1\times n}$. We have that

$$H_kA' = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix} \begin{bmatrix} a^{1*} \\ \vdots \\ a^{(k-1)*} \\ a^{k*} \\ \vdots \\ a^{m*} \end{bmatrix} = \begin{bmatrix} I_{k-1} \begin{bmatrix} a^{1*} \\ \vdots \\ a^{(k-1)*} \end{bmatrix} \\ H_v \begin{bmatrix} a^{k*} \\ \vdots \\ a^{m*} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a^{1*} \\ \vdots \\ a^{(k-1)*} \\ H_v \begin{bmatrix} a^{k*} \\ \vdots \\ a^{m*} \end{bmatrix} \end{bmatrix}, \tag{2.4.20}$$

which has the same first $k-1$ rows as A'. ο.ε.δ.

Lemma 2.4.15. *Given $A' \in \mathbb{R}^{m\times n}$, the last $m-k$ rows of the k:th column of H_kA' are all zeros. (Note that H_k is as in definition 2.4.12.)*

Proof. Let the columns of $A' \in \mathbb{R}^{m\times n}$ be the vectors $a^1, \dots, a^n \in \mathbb{R}^m$. Let the rows of A' be the row vectors $a^{1*}, \dots, a^{m*} \in \mathbb{R}^{1\times n}$. Let $v = (a^k)_{k\to m} - \beta e_1 \in \mathbb{R}^{m-k+1}$, with $\beta = \pm\|(a^k)_{k\to m}\|_2$. By equation (2.4.20)

$$\begin{aligned} H_kA' = \begin{bmatrix} a^{1*} \\ \vdots \\ a^{(k-1)*} \\ H_v \begin{bmatrix} a^{k*} \\ \vdots \\ a^{m*} \end{bmatrix} \end{bmatrix} &= \begin{bmatrix} (a^1)_{1\to k-1} \cdots (a^k)_{1\to k-1} \cdots (a^n)_{1\to k-1} \\ H_v \left[(a^1)_{k\to m} \cdots (a^k)_{k\to m} \cdots (a^n)_{k\to m} \right] \end{bmatrix} \\ &= \begin{bmatrix} (a^1)_{1\to k-1} \cdots (a^k)_{1\to k-1} \cdots (a^n)_{1\to k-1} \\ H_v(a^1)_{k\to m} \cdots H_v(a^k)_{k\to m} \cdots H_v(a^n)_{k\to m} \end{bmatrix}. \end{aligned} \tag{2.4.21}$$

By theorem 2.4.9 we have that

$$H_v(a^k)_{k\to m} = \beta e_1 = \pm\|(a^k)_{k\to m}\|_2 e_1 \in \mathbb{R}^{m-k+1}. \tag{2.4.22}$$

Column k of $H_k A'$ is therefore

$$\begin{bmatrix} (a^k)_{1\to k-1} \\ \pm\|(a^k)_{k\to m}\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^m. \tag{2.4.23}$$

Because $(a^k)_{1\to k-1} \in \mathbb{R}^{k-1}$ and $\pm\|(a^k)_{k\to m}\|_2 \in \mathbb{R}^1$, the last $m-k$ rows of the k:th column of $H_k A'$ are zero. o.ε.δ.

Lemma 2.4.16. *Given $A \in \mathbb{R}^{m\times n}$, then $H_t \dots H_1 A$, where $t = \min\{m-1, n\}$, is an upper-triangular matrix. (Note that H_k, where $1 \le k \le t$, is as in definition 2.4.12.)*

Proof (by induction). Let

$$P(c) = \text{"The first } c \text{ columns of } H_c \dots H_1 A \text{ form an upper-triangular matrix"}, \tag{2.4.24}$$

and let $t = \min\{m-1, n\}$. We will begin by proving by induction that $P(t)$ holds, and then argue why this means that $H_t \dots H_1 A$ is upper-triangular.

Base case. By lemma 2.4.15 we have that the last $m-1$ rows of the first column of $H_1 A$ are zeros. These are the rows below the main diagonal. Therefore the first column of $H_1 A$ forms an $m \times 1$ upper-triangular matrix, meaning that $P(1)$ holds.

Inductive hypothesis. Assume that $P(k-1)$ holds for some $2 \le k \le t$. This means that the first $k-1$ columns of $A' = H_{k-1} \dots H_1 A$ form an $m\times(k-1)$ upper-triangular matrix.

Induction step. Let the columns of A' be the vectors $a^1, \dots, a^n \in \mathbb{R}^m$. Let $v = (a^k)_{k\to m} - \beta e_1 \in \mathbb{R}^{m-k+1}$, with $\beta = \pm\|(a^k)_{k\to m}\|_2$. Because $P(k-1)$ holds, the last $m-(k-1) = m-k+1$ rows of the first $k-1$ columns of A' are zero. Therefore, with $\mathbf{0} \in \mathbb{R}^{m-k+1}$

$$A' = \begin{bmatrix} (a^1)_{1\to k-1} & \cdots & (a^{k-1})_{1\to k-1} & (a^k)_{1\to k-1} & \cdots & (a^n)_{1\to k-1} \\ \mathbf{0} & \cdots & \mathbf{0} & (a^k)_{k\to m} & \cdots & (a^n)_{k\to m} \end{bmatrix}. \tag{2.4.25}$$

Where then changing A' to H_kA' by left multiplying by H_k results in

$$H_kA' = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix} A'. \tag{2.4.26}$$

By equation (2.4.20), this means that the first $k-1$ rows will be unchanged from A', and the last $m-k$ rows, as a matrix, will be left multiplied by H_v. Note by equation (2.4.25) that the last $m-k$ rows of the first $k-1$ columns of A' are only zeros. Since $H_v\,\mathbf{0} = \mathbf{0}$, this means that *all of (the rows)* of the first $k-1$ columns of H_kA' are the same as the first $k-1$ columns of A'. That is

$$H_kA' = \begin{bmatrix} (a^1)_{1\to k-1} & \cdots & (a^{k-1})_{1\to k-1} & (a^k)_{1\to k-1} & \cdots & (a^n)_{1\to k-1} \\ \mathbf{0} & \cdots & \mathbf{0} & H_v(a^k)_{k\to m} & \cdots & H_v(a^n)_{k\to m} \end{bmatrix}. \tag{2.4.27}$$

By the inductive hypothesis $P(k-1)$ holds. This means that the first $k-1$ columns of H_kA' are *also* upper-triangular. By lemma 2.4.15 the last $m-k$ rows of the *k:th column* of H_kA' are all zeros. These are the rows below the main diagonal. Therefore, the first k columns of $H_kA' = H_k \dots H_1A$ form an $m \times k$ upper-triangular matrix, meaning that $P(k)$ holds.

Inductive conclusion. By induction, $P(t)$ holds, where $t = \min\{m-1, n\}$. This means that the first t columns of $A' = H_t \dots H_1A$ form an $m \times t$ upper-triangular matrix. We will now split it into two cases: $m > n$, and $m \leq n$.

Case $m > n$. This means that $t = \min\{m-1, n\} = n$ and A is "tall and thin". Since $P(t)$ holds, the first n columns of $H_t \dots H_1A \in \mathbb{R}^{m\times n}$ form an $m \times n$ upper-triangular matrix. Since $H_t \dots H_1A \in \mathbb{R}^{m\times n}$, this means that $H_t \dots H_1A$ is upper-triangular.

Case $m \leq n$. This means that $t = \min\{m-1, n\} = m-1$ and A is square, or A is "short and thick". Since $P(t)$ holds, the first $m-1$ columns of $H_t \dots H_1A \in \mathbb{R}^{m\times n}$ form an $m \times (m-1)$ upper-triangular matrix. Note that the columns of $H_t \dots H_1A$ are in $\mathbb{R}^m$, which means that the m:th column of $H_t \dots H_1A$ is in $\mathbb{R}^m$. No matter what entries the m:th column of $H_t \dots H_1A$ has, the first m columns of $H_t \dots H_1A$ will therefore be a square $m \times m$ upper-triangular matrix. Because $n \geq m$ this trivially means that (all of) $H_t \dots H_1A$ is upper-triangular. ο.ε.δ.

Lemma 2.4.17. *Given $A \in \mathbb{R}^{m\times n}$, then H_k is a symmetric matrix: $H_k^T = H_k$. (Note that H_k is as in definition 2.4.12.)*

Proof. Omitted. See section A.2 of appendix A on page 81. o.ε.δ.

Lemma 2.4.18. *Given $A \in \mathbb{R}^{m\times n}$, then H_k is an orthogonal matrix: $H_k^T H_k = I_m$. (Note that H_k is as in definition 2.4.12.)*

Proof. Omitted. See section A.2 of appendix A on page 81. o.ε.δ.

Lemma 2.4.19. *Given $A \in \mathbb{R}^{m\times n}$, then $H_t \dots H_1$, where $t = \min\{m-1, n\}$, is an orthogonal matrix. (Note that H_k, where $1 \leq k \leq t$, is as in definition 2.4.12.)*

Proof. Omitted. See section A.2 of appendix A on page 81. o.ε.δ.

Theorem 2.4.20. *Any matrix $A \in \mathbb{R}^{m\times n}$ can be factored as $A = QR$ where $Q = H_1 \dots H_t$ is an orthogonal matrix and $R = H_t \dots H_1 A$ is an upper-triangular matrix. (Note that H_k, where $1 \leq k \leq t$, is as in definition 2.4.12.)*

Proof. By lemma 2.4.19, we have that $Q = H_1 \dots H_t$ is orthogonal. By lemma 2.4.16, we have that $R = H_t \dots H_1 A$ is upper-triangular. By lemma 2.4.17

$$R = H_t \dots H_1 A = (H_1 \dots H_t)^T A = Q^T A. \tag{2.4.28}$$

Since Q is square and orthogonal

$$Q^T Q = I_m \implies Q^T = Q^{-1} \implies QQ^T = I_m. \tag{2.4.29}$$

Therefore

$$QR = QQ^T A = A. \tag{2.4.30}$$

o.ε.δ.

2.4.3 Least squares numerical implementation

We will do this using Householder reflections, because they are numerically stable compared to standard Gram-Schmidt, which is highly prone to round-off errors [12] (p131), [2] (p397). First we consider the block matrix

$$\left[\begin{array}{c|c} A & b \end{array}\right] \in \mathbb{R}^{m\times(n+1)},$$

as in elimination. We will use definition 2.4.12, to be able to calculate the matrices $H_1, \dots, H_t$. By lemma 2.4.19 we have that $Q^T = H_t \dots H_1$ is orthogonal, and by

theorem 2.4.20 $R = Q^T A$ gives rise to the QR-decomposition $A = QR$. Explicitly we will do the calculation

$$H_t(H_{t-1} \dots (H_2(H_1 \left[\begin{array}{c|c} A & b \end{array}\right])) \dots) = Q^T \left[\begin{array}{c|c} A & b \end{array}\right] = \left[\begin{array}{c|c} R & Q^T b \end{array}\right], \tag{2.4.31}$$

where $t = \min\{m-1, n\}$. Now we have the matrix R and vector $Q^T b$, so because R is upper-triangular, the equation

$$Rx = Q^T b \iff Ax = b \tag{2.4.32}$$

is immediately solved by back substitution if $m \leq n$, meaning A is square, or A is "short and thick". If $m > n$ then we instead only consider the first $k \leq n$ rows which are non-zero in R. This results in the new system $\left[\begin{array}{c|c} R_\sim & Q_\sim^T b \end{array}\right]$, where $R_\sim$ is square, or $R_\sim$ is "short and thick". This system can therefore be immediately solved by back substitution.

If $k = n$ then by definition 2.4.1, we are considering the square system $\left[\begin{array}{c|c} R_1 & Q_1^T b \end{array}\right]$. Solving this by back substitution then results in the least squares solution $\hat{x} = R_1^{-1} Q_1^T b$.

3 Regularizing least squares problems

This chapter is based upon a project description in the course MM5016 Numerical Analysis HT21 (Fall 2021) by professor Yishao Zhou, and page 132 in Linear Algebra and Learning from Data [12].

3.1 Regularization

Regularization is a method of adjusting minimization problems that are ill-posed or are prone to ”complicated” solutions. A regularized minimization problem is the same minimization problem as prior but a *penalty term* is added, which will be based on the variable(s). The idea is that now it will be a well-posed problem and ”complicated” solutions will be discouraged because they will be given a higher penalty from the penalty term.

Remark. ”Complicated” could mean that the minimization problem is near ill-posed, leading to numerical instability and high error in the solution. It could also mean solutions which have high ”entropy”: containing large values, or has many parts, etcetera.

Definition 3.1.1. A minimization problem $\min_\chi f(\chi)$ will be *regularized* if one instead considers

$$\min_\chi f(\chi) + \delta R(\chi),$$

where $\delta > 0$ is the *regularization parameter* and $R(\chi) \geq 0$ is the *entropy measure*.

3.2 Tikhonov regularization (L_2)

3.2.1 Definition of Tikhonov regularization

Tikhonov regularization is a way of regularizing least squares problems. It is named after the Russian mathematician Andrey Nikolayevich Tikhonov.

Definition 3.2.1. The *Tikhonov regularization* of $\min_x \|b - Ax\|_2^2$ is the minimization problem $\min_x \|b - Ax\|_2^2 + \|\Gamma x\|_2^2$, where in this report we will consider the case $\Gamma = \sqrt{\delta} I_n$, leading to the minimization problem

$$\min_x \|b - Ax\|_2^2 + \delta \|x\|_2^2.$$

Remark. This special case is also called L_2 regularization and it is called ridge regression when solving this in regression [12] (p132).

3.2.2 Solution to Tikhonov

In order to solve Tikhonov regularized least squares problems $\min_x \|b - Ax\|_2^2 + \delta\|x\|_2^2$, we first note that this can be restated as a standard least squares problem.

Theorem 3.2.2. $\|b - Ax\|_2^2 + \delta\|x\|_2^2 = \left\| \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} x \right\|_2^2$, *where* $\mathbf{0}$ *is the zero vector in* $\mathbb{R}^n$.

Proof. By definition of the Euclidean norm

$$\begin{aligned} \|b - Ax\|_2^2 + \delta\|x\|_2^2 &= (b - Ax)^T(b - Ax) + \delta x^T x \\ &= (b^T - x^T A^T)(b - Ax) + \delta x^T x \\ &= b^T b - b^T Ax - x^T A^T b + x^T A^T Ax + \delta x^T x. \end{aligned} \tag{3.2.1}$$

At the same time

$$\begin{aligned} \left\| \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} x \right\|_2^2 &= \left(\begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} x \right)^T \left(\begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} x \right) \\ &= \left(\begin{bmatrix} b \\ \mathbf{0} \end{bmatrix}^T - x^T \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix}^T \right) \left(\begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} x \right) \\ &= \left(\begin{bmatrix} b^T & \mathbf{0}^T \end{bmatrix} - x^T \begin{bmatrix} A^T & \sqrt{\delta} I_n^T \end{bmatrix} \right) \left(\begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} \right) \\ &= \left(\begin{bmatrix} b^T & \mathbf{0}^T \end{bmatrix} - \begin{bmatrix} x^T A^T & \sqrt{\delta} x^T \end{bmatrix} \right) \left(\begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} Ax \\ \sqrt{\delta} x \end{bmatrix} \right) \\ &= \begin{bmatrix} b^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} b^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} Ax \\ \sqrt{\delta} x \end{bmatrix} \\ &\quad - \begin{bmatrix} x^T A^T & \sqrt{\delta} x^T \end{bmatrix} \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} x^T A^T & \sqrt{\delta} x^T \end{bmatrix} \begin{bmatrix} Ax \\ \sqrt{\delta} x \end{bmatrix} \\ &= b^T b + \mathbf{0}^T \mathbf{0} - b^T Ax - \sqrt{\delta}\, \mathbf{0}^T x - x^T A^T b - \sqrt{\delta} x^T \mathbf{0} \\ &\quad + x^T A^T Ax + \sqrt{\delta}\sqrt{\delta} x^T x \\ &= b^T b - b^T Ax - x^T A^T b + x^T A^T Ax + \delta x^T x. \end{aligned} \tag{3.2.2}$$

This is the same expression as in equation (3.2.1). Hence $\|b - Ax\|_2^2 + \delta\|x\|_2^2$ and $\left\| \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} x \right\|_2^2$ are equal. ο.ε.δ.

We can now use our theory developed in chapter 2 in order to solve this standard least squares problem

$$\min_x \left\| \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} x \right\|_2^2.$$

But first we develop some theory in order to obtain an even stronger result than we achieved in chapter 2.

Lemma 3.2.3. *If the eigenvalues of $A^T A$ are λ_j, where $1 \leq j \leq n$, then the eigenvalues of $(A^T A + \delta I_n)$ are $\lambda_j + \delta$.*

Proof. Let the eigenvalues of $A^T A$ be λ_j, where $1 \leq j \leq n$. By definition of eigenvalues

$$0 = \det(A^T A - \lambda_j I_n). \tag{3.2.3}$$

Let η be an eigenvalue of $(A^T A + \delta I_n)$. By definition

$$\begin{aligned} 0 &= \det(A^T A + \delta I_n - \eta I_n) \\ &= \det(A^T A - (\eta - \delta) I_n). \end{aligned} \tag{3.2.4}$$

Therefore, by equation (3.2.3) for each $1 \leq j \leq n$, we have that $\eta = \lambda_j + \delta$ is an eigenvalue of $(A^T A + \delta I_n)$, because

$$\det(A^T A - (\eta - \delta) I_n) = \det(A^T A - (\lambda_j + \delta - \delta) I_n) = \det(A^T A - \lambda_j I_n) = 0. \tag{3.2.5}$$

Note that $(A^T A + \delta I_n) \in \mathbb{R}^{n \times n}$ has n eigenvalues. Therefore $\eta = \lambda_j + \delta$ for $1 \leq j \leq n$ are all the eigenvalues of $(A^T A + \delta I_n)$. o.ε.δ.

Corollary 3.2.4. *If the singular values of A are σ_i, where $1 \leq i \leq \min\{m, n\}$, then the singular values of $\begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix}$ are $\sqrt{\sigma_i^2 + \delta}$.*

Proof. Let $R = \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix}$. Further, let the singular values of A be $\sigma_i(A)$, where $1 \leq i \leq \min\{m, n\}$, and let the singular values of R be $\sigma_i(R)$. Furthermore let the eigenvalues of $A^T A$ be $\lambda_j(A^T A)$, where $1 \leq j \leq n$, and let the singular values of $R^T R$ be $\lambda_j(R^T R)$. By definition of singular values $\sigma_j(A) = \sqrt{\lambda_j(A^T A)}$, and similarly $\sigma_j(R) = \sqrt{\lambda_j(R^T R)}$, where $\sigma_i(A) = \sigma_i(R) = 0$ if $i > n$. Note first that

$$R^T R = \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix}^T \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} = \begin{bmatrix} A^T & \sqrt{\delta} I_n^T \end{bmatrix} \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} = A^T A + \delta I_n. \tag{3.2.6}$$

By lemma 3.2.3 $\lambda_j(R^TR) = \lambda_j(A^TA) + \delta$. This means that

$$\sigma_i(R) = \sqrt{\lambda_j(R^TR)} = \sqrt{\lambda_j(A^TA) + \delta} = \sqrt{(\sigma_i(A))^2 + \delta}. \tag{3.2.7}$$

o.ε.δ.

Lemma 3.2.5. *If every eigenvalue of $M \in \mathbb{R}^{n\times n}$ is not equal to 0, then M is invertible.*

Proof. Let $M \in \mathbb{R}^{n\times n}$ have no eigenvalues equal to 0. Note that M is square, and is therefore invertible if and only if $\mathbf{N}(M)$ is trivial. By definition of the null-space

$$x \in \mathbf{N}(M) \iff Mx = 0 \iff Mx = 0x. \tag{3.2.8}$$

Therefore, we have that $\mathbf{N}(M)$ is trivial if and only if the only solution to $Mx = 0x$ is $x = \mathbf{0}$. By the definition of eigenvector, $x \neq \mathbf{0}$ satisfying $Mx = 0x$ is an eigenvector of M with eigenvalue 0. But, since M has no eigenvalues equal to 0, the only solution to $Mx = 0x$ is $x = \mathbf{0}$. Therefore $\mathbf{N}(M)$ is trivial and M is invertible. o.ε.δ.

Lemma 3.2.6. *$(A^TA + \delta I_n)$ is invertible.*

Proof. Note that $\delta > 0$ by definition of regularization parameter. Let the eigenvalues of A^TA be λ_j, where $1 \leq j \leq n$. Note that A^TA can only be assumed to be positive *semi*-definite because

$$x^TA^TAx = \|Ax\|_2^2 \geq 0, \tag{3.2.9}$$

where this is zero when $x \in \mathbf{N}(A)$. A^TA being positive semi-definite is equivalent with the eigenvalues λ_j of A^TA being real non-negative [4] (p233). That is, $\lambda_j \geq 0$. By lemma 3.2.3 the eigenvalues of $(A^TA + \delta I_n)$ are $(\lambda_j + \delta)$. Since $\delta > 0$

$$\lambda_j \geq 0 \implies (\lambda_j + \delta) > 0. \tag{3.2.10}$$

This means that all n eigenvalues $(\lambda_j + \delta)$ of $(A^TA + \delta I_n)$ are positive, and therefore non-zero. By lemma 3.2.5 this means that $(A^TA + \delta I_n)$ is invertible. o.ε.δ.

Theorem 3.2.7. *$\hat{x}_\delta = (A^TA + \delta I_n)^{-1}A^Tb$ always exists and is the unique minimizer of $\|b - A\hat{x}_\delta\|_2^2 + \delta\|\hat{x}_\delta\|_2^2$.*

Proof. By lemma 3.2.6 $(A^TA + \delta I_n)^{-1}$ exists, therefore $(A^TA + \delta I_n)^{-1}A^Tb$ exists. By theorem 3.2.2 we have that

$$f(x) = \|b - Ax\|_2^2 + \delta\|x\|_2^2 = \left\| \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} x \right\|_2^2. \tag{3.2.11}$$

By theorem 2.2.4, then the unique minimizer of $f(\hat{x}_\delta)$ is

$$\begin{aligned}
\hat{x}_\delta &= \left(\begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix}^T \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} \right)^{-1} \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix}^T \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} \\
&= \left(\begin{bmatrix} A^T & \sqrt{\delta} I_n^T \end{bmatrix} \begin{bmatrix} A \\ \sqrt{\delta} I_n \end{bmatrix} \right)^{-1} \begin{bmatrix} A^T & \sqrt{\delta} I_n^T \end{bmatrix} \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} \\
&= (A^T A + \sqrt{\delta}\sqrt{\delta} I_n^T I_n)^{-1} (A^T b + \sqrt{\delta} I_n^T \mathbf{0}) \\
&= (A^T A + \delta I_n)^{-1} A^T b.
\end{aligned} \tag{3.2.12}$$

o.ε.δ.

We now know how to solve Tikhonov regularized least squares problems of the form $\min_x \|b - Ax\|_2^2 + \delta \|x\|_2^2$, which are theoretically *always* solvable.

3.2.3 Tikhonov solutions relation to the pseudoinverse

It turns out that the solution to the Tikhonov regularized least squares problem $\hat{x}_\delta = (A^T A + \delta I_n)^{-1} A^T b$ is intimately related to the solution to the pseudoinverse solution to the standard least squares problem $\hat{x} = A^+ b$. (Note that these are solutions that always exist, compared to the solution $(A^T A)^{-1} A^T b$, which, by theorem 2.2.2, requires that $\mathbf{N}(A)$ is trivial.) In fact, we will prove the following theorem.

Theorem 3.2.8. $\lim_{\delta \to 0} (A^T A + \delta I_n)^{-1} A^T = A^+$.

Proof. Let the SVD of A be $A = U \Sigma V^T$. Note that $U^T = U^{-1} \in \mathbb{R}^{m \times m}$ and $V^T = V^{-1} \in \mathbb{R}^{n \times n}$. This means that

$$\begin{aligned}
A^T A + \delta I_n &= (U \Sigma V^T)^T (U \Sigma V^T) + \delta I_n \\
&= V \Sigma^T U^T U \Sigma V^T + \delta I_n \\
&= V \Sigma^T \Sigma V^T + \delta I_n \\
&= V \Sigma^T \Sigma V^T + \delta V V^T \\
&= V (\Sigma^T \Sigma + \delta I_n) V^T.
\end{aligned} \tag{3.2.13}$$

By lemma 3.2.6 $(\Sigma^T \Sigma + \delta I_n)$ is invertible. Therefore

$$\begin{aligned}
(A^T A + \delta I_n)^{-1} &= (V (\Sigma^T \Sigma + \delta I_n) V^T)^{-1} \\
&= (V^T)^{-1} (\Sigma^T \Sigma + \delta I_n)^{-1} (V)^{-1} \\
&= V (\Sigma^T \Sigma + \delta I_n)^{-1} V^T.
\end{aligned} \tag{3.2.14}$$

Now we can express $(A^TA + \delta I_n)^{-1}A^T$ as

$$\begin{aligned}(A^TA + \delta I_n)^{-1}A^T &= (V(\Sigma^T\Sigma + \delta I_n)^{-1}V^T)(U\Sigma V^T)^T \\ &= V(\Sigma^T\Sigma + \delta I_n)^{-1}V^TV\Sigma^TU^T \\ &= V(\Sigma^T\Sigma + \delta I_n)^{-1}\Sigma^TU^T \\ &= VD_\delta U^T,\end{aligned} \tag{3.2.15}$$

where $D_\delta = (\Sigma^T\Sigma + \delta I_n)^{-1}\Sigma^T$. Let us now figure out what the entries of $D_\delta = (\Sigma^T\Sigma + \delta I_n)^{-1}\Sigma^T$ are. Let all of the singular values of A be $\sigma_1 \geq \ldots \geq \sigma_r > 0 = \sigma_{r+1} = \ldots = \sigma_{\min\{m,n\}}$, where $r = \mathbf{rank}(A)$. Note now that $\Sigma \in \mathbb{R}^{m\times n}$, $\Sigma^T \in \mathbb{R}^{n\times m}$ and $I_n \in \mathbb{R}^{n\times n}$ are diagonal matrices. By definition, for $1 \leq k \leq r$ we have that $(\Sigma)_{k,k} = (\Sigma^T)_{k,k} = \sigma_k > 0$ and the rest of their entries are zeros. Therefore, for $1 \leq k \leq r$

$$(\Sigma^T\Sigma)_{k,k} = \sigma_k^2 > 0, \tag{3.2.16}$$

with all other entries being zero, including the last $n - r$ entries on the main diagonal. And therefore, since $\delta > 0$, for $1 \leq k \leq n$

$$(\Sigma^T\Sigma + \delta I_n)_{k,k} = \sigma_k^2 + \delta > 0, \tag{3.2.17}$$

where all other entries are zero. This means that $(\Sigma^T\Sigma + \delta I_n) \in \mathbb{R}^{n\times n}$ is diagonal, which means that for $1 \leq k \leq n$

$$((\Sigma^T\Sigma + \delta I_n)^{-1})_{k,k} = \frac{1}{\sigma_k^2 + \delta} > 0, \tag{3.2.18}$$

and the rest of the entries are zero. Therefore $(\Sigma^T\Sigma + \delta I_n)^{-1}$ is diagonal, and so for $1 \leq k \leq \min\{m, n\}$

$$(D_\delta)_{k,k} = ((\Sigma^T\Sigma + \delta I_n)^{-1}\Sigma^T)_{k,k} = \frac{1}{\sigma_k^2 + \delta}(\sigma_k) = \frac{\sigma_k}{\sigma_k^2 + \delta} \geq 0, \tag{3.2.19}$$

with all other entries being 0, where $\frac{\sigma_k}{\sigma_k^2+\delta} = 0$ if and only if $\sigma_k = 0$, which is true if and only if $k > r$. Therefore $D_\delta \in \mathbb{R}^{n\times m}$ is diagonal, with the first r entries on the main diagonal being non-zero, and the last $\min\{m, n\} - r$ entries on the main diagonal being zeros. Because D_δ is diagonal we can write

$$(A^TA + \delta I_n)^{-1}A^T = VD_\delta U^T = \sum_{k=1}^{\min\{m,n\}} ((D_\delta)_{k,k})\, v_k u_k^T, \tag{3.2.20}$$

where v_k is the k:th column of V, and u_k^T is the k:th row of U^T. Therefore, because this is a finite sum

$$\begin{aligned}\lim_{\delta\to 0}(A^TA+\delta I_n)^{-1}A^T &= \lim_{\delta\to 0}\sum_{k=1}^{\min\{m,n\}}\left((D_\delta)_{k,k}\right)v_ku_k^T \\ &= \sum_{k=1}^{\min\{m,n\}}\lim_{\delta\to 0}\left((D_\delta)_{k,k}\right)v_ku_k^T \\ &= \sum_{k=1}^{\min\{m,n\}}\left(\lim_{\delta\to 0}(D_\delta)_{k,k}\right)v_ku_k^T,\end{aligned} \tag{3.2.21}$$

since $v_ku_k^T$ is a constant matrix with respect to δ. Now, for $1 \leq k \leq r$

$$\lim_{\delta\to 0}(D_\delta)_{k,k} = \lim_{\delta\to 0}\frac{\sigma_k}{\sigma_k^2+\delta} = \frac{\sigma_k}{\sigma_k^2+0} = \frac{1}{\sigma_k}, \tag{3.2.22}$$

and for $r < k \leq \min\{m,n\}$

$$\lim_{\delta\to 0}(D_\delta)_{k,k} = \lim_{\delta\to 0}\frac{\sigma_k}{\sigma_k^2+\delta} = \lim_{\delta\to 0}\frac{0}{0^2+\delta} = \lim_{\delta\to 0}0 = 0. \tag{3.2.23}$$

By definition 2.3.1, this means that

$$\lim_{\delta\to 0}(A^TA+\delta I_n)^{-1}A^T = \lim_{\delta\to 0}VD_\delta U^T = V\Sigma^+U^T = A^+. \tag{3.2.24}$$

o.ε.δ.

3.3 Other types of regularization (L_1, L_∞, "L_0")

In this section we will define more types of regularization of least squares problems other than Tikhonov regularization. There will be figures to see and compare what kind of solutions these regularizations encourage or discourage for $x \in \mathbb{R}^2$.

It turns out there are no general closed form solutions to these regularized problems, and specific optimization theory and algorithms would be needed to solve these problems numerically. However, this is beyond the scope of this report and will not be covered.

3.3.1 Definitions of the regularizations

LASSO regularization is a way of regularizing least squares problems. LASSO is an acronym meaning "least absolute shrinkage and selection operator" [13] (p267).

Definition 3.3.1. The *LASSO regularization*, or alternatively L_1 *regularization*, of $\min_x \|b - Ax\|_2^2$ is the minimization problem

$$\min_x \|b - Ax\|_2^2 + \delta\|x\|_1.$$

It turns out that because $\|b - Ax\|_2^2$ is a convex function of x, then solutions $\hat{x}_\delta$ to the LASSO regularized minimization problem will contain (many) entries that are zero. Solutions with many zeros are called sparse solutions, and in figure 2.2 on page 11 of [3] we see why the LASSO regularized minimization will produce such solutions compared to the solutions to the Tikhonov (L_2) regularized least squares problems.

Definition 3.3.2. The L_∞ *regularization* of $\min_x \|b - Ax\|_2^2$ is the minimization problem

$$\min_x \|b - Ax\|_2^2 + \delta\|x\|_\infty.$$

The solutions $\hat{x}_\delta$ to the L_∞ regularized minimization problem will, by definition of the L_∞ norm, not contain any entry that is too large, given that δ is large enough.

Definition 3.3.3. The *"L_0 norm"* of a vector $v \in \mathbb{R}^n$ will be denoted by $\|v\|_0$ and be defined by

$$\|v\|_0 = |\{v_i \mid v_i \neq 0\}|.$$

This is the number of entries of v that are not equal to 0.

Remark. This may also be denoted by $\mathrm{Card}(v)$ [12] (p100).

Remark. This does not fulfill the definition of vector norm. ($\|av\|_0 \neq |a|\|v\|_0$.)

Definition 3.3.4. The L_0 *regularization* of $\min_x \|b - Ax\|_2^2$ is the minimization problem

$$\min_x \|b - Ax\|_2^2 + \delta\|x\|_0.$$

By definition of L_0 regularization, if the regularization parameter δ is large enough, then solutions $\hat{x}_\delta$ to the L_0 regularized minimization problem will contain (many) entries that are equal to 0. This is a more direct approach to sparse solutions than L_1 regularisation. See figure 2.6 on page 22 of [3], and compare with the previous figure cited, being figure 2.2 on page 11.

Another way to regularize the problem of least squares is to combine two different types of regularizations. This brings us to the elastic net, which is a combination of the LASSO and Tikhonov regularizations.

Definition 3.3.5. The *elastic net regularization* of $\min_x \|b - Ax\|_2^2$ is the minimization problem

$$\min_x \|b - Ax\|_2^2 + \delta\|x\|_1 + (1 - \delta)\|x\|_2^2.$$

3.3.2 How to attain the solutions

In the case when $A \in \mathbb{R}^{m \times n}$ is an orthogonal matrix, the L_1 regularized least squares problem does have a closed form solution (see page 269 of [13]). In the general case, more advanced concepts such as *second order cone programming* (SOCP) can be used to restate the problem [1] (p310).

One way of solving the L_1, L_∞ and elastic net regularized least squares problems numerically is by using a proximal gradient method [12] (p357, p191). Another way is by the homotopy method *least angle regression* (LARS) [3] (p17).

To solve L_0 regularized least squares is especially hard, because it is no longer a convex optimization problem. Algorithms to solve this problem numerically are even in the class *non-deterministic polynomial-time hard* (NP-hard) [11] (p2). A very efficient way to solve the *convex relaxation* of this problem is by using a *forward backward* algorithm [11] (p2).

4 Gradient descent methods

Gradient descent methods are methods based on the iterative method *gradient descent*, which are used to find local minima of functions $f : \mathbb{R}^n \to \mathbb{R}$ based on the gradient of the function $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$. Therefore, in this chapter we will assume that f has a defined gradient for every element in $\mathbb{R}^n$.

The following methods are all iterative, meaning that we consider some discrete time $k \geq 0$, where for each time step, one iteration of the methods will be performed. Here we are interested in what happens as $k \to \infty$ where we wish for our methods to as rapidly as possible converge to a local minimum.

4.1 Gradient descent

In this section we will cover the definition of gradient descent and interesting properties of it.

4.1.1 Definition of gradient descent

We will consider $f : \mathbb{R}^n \to \mathbb{R}$ to be a function that we wish to minimize.

Definition 4.1.1. Let $f : \mathbb{R}^n \to \mathbb{R}$. Let $x^{(0)} \in \mathbb{R}^n$ be arbitrary. For $k \geq 0$, one iteration of *gradient descent* is

$$x^{(k+1)} = x^{(k)} - s^{(k)} \nabla f(x^{(k)}),$$

where $s^{(k)} > 0$ is the *step size* at discrete time k.

4.1.2 Properties of gradient descent

Theorem 4.1.2. *The direction of $-\nabla f(x) \neq \mathbf{0}$ in $\mathbb{R}^n$ is locally the direction of steepest descent with respect to f from the point $x \in \mathbb{R}^n$.*

Proof. Recall that the directional derivative in the direction of a vector $u \in \mathbb{R}^n$ with $\|u\|_2 = 1$ of a function $f : \mathbb{R}^n \to \mathbb{R}$ is

$$D_v f(x) = \nabla f(x) \cdot u = (\nabla f(x))^T v \in \mathbb{R}. \tag{4.1.1}$$

How locally steep the function is in the direction along the vector $w \in \mathbb{R}^n$ will be defined to be $D_u f(x)$, where $u = \frac{w}{\|w\|}$. Therefore, a direction of steepest *descent* for f from $x \in \mathbb{R}$ is a vector v satisfying

$$\min_u D_u f(x) = D_v f(x) \tag{4.1.2}$$

subject to

$$\|u\|_2 = 1. \tag{4.1.3}$$

The Cauchy-Schwartz inequality says that

$$|a \cdot b| \leq \|a\|_2 \|b\|_2, \tag{4.1.4}$$

for vectors $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$, which means that

$$a \cdot b \geq -\|a\|_2 \|b\|_2. \tag{4.1.5}$$

Note now that the lower bound is reached if and only if $b = -\gamma a$, where $\gamma \geq 0$. We will now use this in our problem to find a lower bound of $D_u f(x)$. We have that

$$D_u f(x) = \nabla f(x) \cdot u \geq -\|\nabla f(x)\|_2 \|u\|_2, \tag{4.1.6}$$

and that this lower bound is reached if and only if $u = -\gamma \nabla f(x)$. Constrained by $\|u\|_2 = 1$ we have that $\gamma = \frac{1}{\|\nabla f(x)\|_2}$ and

$$u = v = -\frac{\nabla f(x)}{\|\nabla f(x)\|_2} \tag{4.1.7}$$

is the *unique* solution to $\min_u D_u f(x)$ subject to $\|u\|_2 = 1$. Finally we note that this is in the direction of $-\nabla f(x)$. Therefore, locally, the direction of steepest descent from a point $x \in \mathbb{R}^n$ with respect to a function $f : \mathbb{R}^n \to \mathbb{R}$, is in the direction of $-\nabla f(x) \in \mathbb{R}^n$. ο.ε.δ.

Lemma 4.1.3. *If ∇f is L-Lipschitz continuous then*

$$f(b) \leq f(a) + (\nabla f(a))^T (b - a) + \frac{L\|b - a\|_2^2}{2}.$$

Proof. Let ∇f be L-Lipschitz continuous. ∇f being L-Lipschitz continuous means that for every $c \in \mathbb{R}^n$ and $d \in \mathbb{R}^n$

$$\|\nabla f(c) - \nabla f(d)\|_2 \leq L\|c - d\|_2. \tag{4.1.8}$$

With $u = (\nabla f(c) - \nabla f(d))$ and $v = (c - d)$, the Cauchy-Schwartz inequality says that

$$\begin{aligned} u^T v \leq \|u\|_2 \|v\|_2 \iff (\nabla f(c) - \nabla f(d))^T (c - d) &\leq \|\nabla f(c) - \nabla f(d)\|_2 \|c - d\|_2 \\ &\leq L\|c - d\|_2^2. \end{aligned} \tag{4.1.9}$$

For convenience, we will now define $g : [0,1] \to \mathbb{R}$ by

$$g(t) = f(a + t(b-a)). \tag{4.1.10}$$

Moreover, by the chain rule

$$g'(t) = (\nabla f(a + t(b-a)))^T (b-a), \tag{4.1.11}$$

meaning that by inequality (4.1.9)

$$\begin{aligned} t(g'(t) - g'(0)) &= t(((\nabla f(a + t(b-a)))^T (b-a)) - ((\nabla f(a))^T (b-a))) \\ &= t(\nabla f(a + t(b-a) - \nabla f(a)))^T (b-a) \\ &= (\nabla f(a + t(b-a) - \nabla f(a)))^T (t(b-a)) \\ &= (\nabla f(a + t(b-a) - \nabla f(a)))^T ((a + t(b-a)) - a) \\ &\leq L\|(a + t(b-a)) - a\|_2^2 \\ &= L\|t(b-a)\|_2^2 \\ &= t^2 L\|b-a\|_2^2. \end{aligned} \tag{4.1.12}$$

If we from now on restrict $t \neq 0$, meaning $t \in]0,1]$, then this means

$$\begin{aligned} g'(t) - g'(0) &\leq tL\|b-a\|_2^2 \\ \iff g'(t) &\leq g'(0) + tL\|b-a\|_2^2. \end{aligned} \tag{4.1.13}$$

Integrating both sides of this inequality over $]0,1[$ with respect to t results in

$$\begin{aligned} &\int_0^1 g'(t)dt = g(1) - g(0) = f(b) - f(a) \\ \leq &\int_0^1 g'(0)dt + \int_0^1 tL\|b-a\|_2^2 dt = g'(0) + \frac{L\|b-a\|_2^2}{2} \\ = &(\nabla f(a))^T (b-a) + \frac{L\|b-a\|_2^2}{2}. \end{aligned} \tag{4.1.14}$$

Therefore

$$f(b) \leq f(a) + (\nabla f(a))^T (b-a) + \frac{L\|b-a\|_2^2}{2}, \tag{4.1.15}$$

where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$ are arbitrary. ο.ε.δ.

Theorem 4.1.4. *If $x^{(0)}, x^{(1)}, \ldots$ are generated from gradient descent with $f : \mathbb{R}^n \to \mathbb{R}$ and $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ is L-Lipschitz continuous, then $f(x^{(0)}) > f(x^{(1)}) > \ldots$ as long as $\nabla f(x^{(k)}) \neq \mathbf{0}$ and $s^{(k)} < \frac{2}{L}$.*

Proof. Let $x^{(k+1)}$ be generated from gradient descent with f. That is

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - s^{(k)}\nabla f(x^{(k)}) \\ \implies x^{(k+1)} - x^{(k)} &= -s^{(k)}\nabla f(x^{(k)}). \end{aligned} \tag{4.1.16}$$

Let ∇f be L-Lipschitz continuous. By lemma 4.1.3 we have that

$$\begin{aligned} f(x^{(k+1)}) &\leq f(x^{(k)}) + (\nabla f(x^{(k)}))^T(x^{(k+1)} - x^{(k)}) + \frac{L\|x^{(k+1)} - x^{(k)}\|_2^2}{2} \\ &= f(x^{(k)}) - s^{(k)}(\nabla f(x^{(k)}))^T(\nabla f(x^{(k)})) + \frac{L\| - s^{(k)}\nabla f(x^{(k)})\|_2^2}{2} \\ &= f(x^{(k)}) - s^{(k)}\|\nabla f(x^{(k)})\|_2^2 + (s^{(k)})^2\frac{L\|\nabla f(x^{(k)})\|_2^2}{2} \\ &= f(x^{(k)}) - s^{(k)}\left(1 - \frac{L}{2}s^{(k)}\right)\|\nabla f(x^{(k)})\|_2^2. \end{aligned} \tag{4.1.17}$$

If $s^{(k)}\left(1 - \frac{L}{2}s^{(k)}\right)\|\nabla f(x^{(k)})\|_2^2 > 0$ we are done. Because $\nabla f(x^{(k)}) \neq \mathbf{0}$ we have that $\|\nabla f(x^{(k)})\|_2^2 > 0$, and $s^{(k)} > 0$, so we are done if $\left(1 - \frac{L}{2}s^{(k)}\right) > 0$. Therefore we are done if $1 > \frac{L}{2}s^{(k)}$, that is, if $s^{(k)} < \frac{2}{L}$. Since we have assumed in the statement of the theorem that $s^{(k)} < \frac{2}{L}$, this means that

$$f(x^{(k+1)}) \leq f(x^{(k)}) - s^{(k)}\left(1 - \frac{L}{2}s^{(k)}\right)\|\nabla f(x^{(k)})\|_2^2 < f(x^{(k)}), \tag{4.1.18}$$

and so $f(x^{(k+1)}) < f(x^{(k)})$ for any $k \geq 0$, resulting in

$$f(x^{(0)}) > f(x^{(1)}) > \dots \tag{4.1.19}$$

ο.ε.δ.

Theorem 4.1.5. *If ∇f is L-Lipschitz continuous, $s^{(k)} = c < \frac{2}{L}$ is fixed, and a global minimum of f exists, then gradient descent converges to a local minimum of f.*

Proof. Let ∇f be L-Lipschitz continuous and $s^{(k)} = c < \frac{2}{L}$. By inequality (4.1.18) we have that

$$\begin{aligned} f(x^{(k)}) - c\left(1 - \frac{L}{2}c\right)\|\nabla f(x^{(k)})\|_2^2 &\geq f(x^{(k+1)}) \\ \iff f(x^{(k)}) - f(x^{(k+1)}) &\geq c\left(1 - \frac{L}{2}c\right)\|\nabla f(x^{(k)})\|_2^2. \end{aligned} \tag{4.1.20}$$

For convenience, let $t = c\left(1 - \frac{L}{2}c\right)$. Therefore

$$f(x^{(k)}) - f(x^{(k+1)}) \geq t\|\nabla f(x^{(k)})\|_2^2. \tag{4.1.21}$$

Because $k \geq 0$ is arbitrary, all at the same time

$$\begin{aligned} f(x^{(0)}) - f(x^{(1)}) &\geq t\|\nabla f(x^{(0)})\|_2^2 \\ f(x^{(1)}) - f(x^{(2)}) &\geq t\|\nabla f(x^{(1)})\|_2^2 \\ &\vdots \\ f(x^{(N-1)}) - f(x^{(N)}) &\geq t\|\nabla f(x^{(N-1)})\|_2^2 \\ f(x^{(N)}) - f(x^{(N+1)}) &\geq t\|\nabla f(x^{(N)})\|_2^2, \end{aligned} \tag{4.1.22}$$

where $N > 0$. Note that adding all these N inequalities produces a telescoping sum. More precisely

$$\begin{aligned} \sum_{k=0}^{N} t\|\nabla f(x^{(k)})\|_2^2 &\leq \sum_{k=0}^{N} f(x^{(k)}) - f(x^{(k+1)}) \\ &= \sum_{k=0}^{N} f(x^{(k)}) - \sum_{k=0}^{N} f(x^{(k+1)}) \\ &= f(x^{(0)}) + \sum_{k=1}^{N} f(x^{(k)}) - \sum_{k=0}^{N-1} f(x^{(k+1)}) - f(x^{(N+1)}) \\ &= f(x^{(0)}) + \sum_{k=1}^{N} f(x^{(k)}) - \sum_{k=1}^{N} f(x^{(k)}) - f(x^{(N+1)}) \\ &= f(x^{(0)}) - f(x^{(N+1)}). \end{aligned} \tag{4.1.23}$$

Let the global minimum value of f be $f^* \in \mathbb{R}$. Therefore

$$\sum_{k=0}^{N} t\|\nabla f(x^{(k)})\|_2^2 \leq f(x^{(0)}) - f(x^{(N+1)}) \leq f(x^{(0)}) - f^*. \tag{4.1.24}$$

Because f^* is the global minimum value of f we have that

$$f^* \leq f(x^{(0)}) \implies 0 \leq f(x^{(0)}) - f^*. \tag{4.1.25}$$

This is a non-negative constant that we can call $E = f(x^{(0)}) - f^*$. Because $N > 0$ is arbitrary

$$\begin{aligned} &\sum_{k=0}^{N} t\|\nabla f(x^{(k)})\|_2^2 \leq E \\ \implies &\lim_{N\to\infty} \sum_{k=0}^{N} t\|\nabla f(x^{(k)})\|_2^2 \leq E. \end{aligned} \tag{4.1.26}$$

Remember that $t = c\left(1 - \frac{L}{2}c\right)$ and by definition of step size $0 < c < \frac{2}{L}$. By definition of L-Lipschitz continuous seen in inequality (4.1.8), it only possible that $0 \leq L$. (If $L = 0$ take "$\frac{2}{L}$" to mean $+\infty$). This means that $\frac{L}{2}c < 1$, meaning that $t > 0$. Therefore

$$0 \leq \lim_{N\to\infty} \sum_{k=0}^{N} t\|\nabla f(x^{(k)})\|_2^2 \leq E, \tag{4.1.27}$$

and so $\lim_{N\to\infty} \sum_{k=0}^{N} t\|\nabla f(x^{(k)})\|_2^2$ is a convergent series, meaning that the terms must go to zero. Hence

$$\begin{aligned} &\lim_{k\to\infty} t\|\nabla f(x^{(k)})\|_2^2 = 0 \\ \implies &\lim_{k\to\infty} \|\nabla f(x^{(k)})\|_2^2 = 0 \\ \implies &\lim_{k\to\infty} \nabla f(x^{(k)}) = \mathbf{0}\,. \end{aligned} \tag{4.1.28}$$

This means that gradient descent reaches a stationary point of f in the limit, meaning because of theorem 4.1.4 that gradient descent converges to a local minimum of f. ο.ε.δ.

4.2 Gradient descent to solve least squares problems

In this section we will now try to solve the standard least squares problem $\min_x \|b - Ax\|_2^2$ by using gradient descent. Note first that minimizing $\|b - Ax\|_2^2$ with respect to x is the same as minimizing $\frac{1}{2}\|b - Ax\|_2^2$ with respect to x. Let $f(x) = \frac{1}{2}\|b - Ax\|_2^2$. By equation (2.2.14) we have that

$$\begin{aligned} \nabla f(x) &= \nabla \frac{1}{2}\|b - Ax\|_2^2 \\ &= \frac{1}{2}\nabla\|b - Ax\|_2^2 \\ &= \frac{1}{2}(2A^TAx - 2A^Tb) \\ &= A^TAx - A^Tb \\ &= A^T(Ax - b). \end{aligned} \tag{4.2.1}$$

Remark. The matrix A^T is factored out because it is more computationally efficient to compute Ax followed by $A^T(Ax)$ compared to computing A^TA followed by $(A^TA)x$.

Using gradient descent with $f(x) = \frac{1}{2}\|b - Ax\|_2^2$ is called the *Landweber iteration*.

4.2.1 Definition of the Landweber iteration

We consider $f(x) = \frac{1}{2}\|b - Ax\|_2^2$ to be a function that we wish to minimize. To find a minimizer of this function we perform the Landweber iteration.

Definition 4.2.1. Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Let $x^{(0)} \in \mathbb{R}^n$ be arbitrary. For $k \geq 0$, one iteration of the *Landweber iteration* is

$$x^{(k+1)} = x^{(k)} - s^{(k)} A^T (Ax^{(k)} - b),$$

where $s^{(k)} > 0$ is the step size at discrete time k.

4.2.2 Analysing the Landweber iteration

First we will analyze the step size of the Landweber iteration. After that we will consider a fixed step size and analyze the convergence of the method.

Theorem 4.2.2. *The optimal step size for the $(k+1)$:st step in the Landweber iteration is*

$$s^{(k)} = \frac{\|\nabla f(x^{(k)})\|_2^2}{\|A\nabla f(x^{(k)})\|_2^2},$$

where optimal step size means

$$\min_s f(x^{(k+1)}; s) = f(x^{(k+1)}; s^{(k)}),$$

and $f(x) = \frac{1}{2}\|b - Ax\|_2^2$ and $\nabla f(x) = A^T(Ax - b)$.

Proof. Let $f(x) = \frac{1}{2}\|b - Ax\|_2^2$ and let $x^{(k+1)}$ be as in the Landweber iteration, but with $s^{(k)} = s$. For simplicity we let $p^{(k)} = -\nabla f(x^{(k)}) = -A^T(Ax^{(k)} - b)$. We then have

$$x^{(k+1)} = x^{(k)} - sA^T(Ax^{(k)} - b) = x^{(k)} + sp^{(k)}. \tag{4.2.2}$$

In order to find the optimal step size, let us define a function $g(s)$ dependent on the step size s that we can try to optimize. Let $g : \mathbb{R} \to \mathbb{R}$ be defined by

$$\begin{aligned}
2g(s) &= 2f(x^{(k+1)}; s) \\
&= 2f(x^{(k)} + sp^{(k)}) \\
&= \|b - A(x^{(k)} + sp^{(k)})\|_2^2 \\
&= \|b - Ax^{(k)} - sAp^{(k)}\|_2^2 \\
&= \|(b - Ax^{(k)}) - sAp^{(k)}\|_2^2 \\
&= ((b - Ax^{(k)}) - sAp^{(k)})^T((b - Ax^{(k)}) - sAp^{(k)}) \\
&= ((b - Ax^{(k)})^T - s(Ap^{(k)})^T)((b - Ax^{(k)}) - sAp^{(k)}) \\
&= (b - Ax^{(k)})^T(b - Ax^{(k)}) - s(b - Ax^{(k)})^T Ap^{(k)} \\
&\quad - s(Ap^{(k)})^T(b - Ax^{(k)}) + s^2(Ap^{(k)})^T Ap^{(k)}.
\end{aligned} \tag{4.2.3}$$

Note now that

$$\begin{aligned}(b - Ax^{(k)})^T Ap^{(k)} &= (A^T(b - Ax^{(k)}))^T p^{(k)} \\ &= (p^{(k)})^T p^{(k)} \\ &= \|p^{(k)}\|_2^2,\end{aligned} \tag{4.2.4}$$

and similarly

$$\begin{aligned}(Ap^{(k)})^T(b - Ax^{(k)}) &= (p^{(k)})^T A^T(b - Ax^{(k)}) \\ &= (p^{(k)})^T p^{(k)} \\ &= \|p^{(k)}\|_2^2.\end{aligned} \tag{4.2.5}$$

Therefore

$$\begin{aligned}2g(s) &= \|b - Ax^{(k)}\|_2^2 - 2s\|p^{(k)}\|_2^2 + s^2\|Ap^{(k)}\|_2^2 \\ &= 2\left(\frac{1}{2}\|b - Ax^{(k)}\|_2^2 - s\|p^{(k)}\|_2^2 + \frac{s^2}{2}\|Ap^{(k)}\|_2^2\right).\end{aligned} \tag{4.2.6}$$

This is a standard second degree polynomial in s with positive coefficient to the s^2 term. Therefore $s^{(k)}$ satisfying $g'(s^{(k)}) = 0$ is the unique global minimizer of g. By standard derivative rules

$$g'(s) = -\|p^{(k)}\|_2^2 + s\|Ap^{(k)}\|_2^2. \tag{4.2.7}$$

Therefore

$$\begin{aligned}g'(s^{(k)}) = 0 &\iff -\|p^{(k)}\|_2^2 + s^{(k)}\|Ap^{(k)}\|_2^2 = 0 \\ &\iff s^{(k)}\|Ap^{(k)}\|_2^2 = \|p^{(k)}\|_2^2 \\ &\implies s^{(k)} = \frac{\|p^{(k)}\|_2^2}{\|Ap^{(k)}\|_2^2}.\end{aligned} \tag{4.2.8}$$

By definition of $p^{(k)}$

$$\begin{aligned}s^{(k)} &= \frac{\|p^{(k)}\|_2^2}{\|Ap^{(k)}\|_2^2} = \frac{\|-\nabla f(x^{(k)})\|_2^2}{\|-A\nabla f(x^{(k)})\|_2^2} = \frac{\|\nabla f(x^{(k)})\|_2^2}{\|A\nabla f(x^{(k)})\|_2^2} \\ &= \frac{\|A^T(Ax^{(k)} - b)\|_2^2}{\|A(A^T(Ax^{(k)} - b))\|_2^2}.\end{aligned} \tag{4.2.9}$$

ο.ε.δ.

Before we study the convergence of the Landweber iteration we cover some useful concepts and theory.

Definition 4.2.3. The *spectral radius* $\rho(M)$ of a square matrix $M \in \mathbb{R}^{n\times n}$ is the absolute value of an eigenvalue of M with the largest absolute value.

Theorem 4.2.4. *For $M \in \mathbb{R}^{n\times n}$*

$$\lim_{k\to\infty} M^k = 0_{n\times n}$$

if and only if $\rho(M) < 1$. (This means that M is a convergent matrix.)

Proof. (if, $\Longleftarrow$). Let $\rho(M) < 1$. By [4] (p119), every square matrix is *not* diagonalizable, but every square matrix has a more general *Jordan decomposition*. This means that we can decompose our matrix M as

$$M = SJS^{-1}, \tag{4.2.10}$$

where $S \in \mathbb{R}^{n\times n}$ is an invertible matrix and $J \in \mathbb{R}^{n\times n}$ is the *Jordan normal form* of $M \in \mathbb{R}^{n\times n}$ defined by being the block matrix

$$J = \begin{bmatrix} J_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & J_2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 0 & J_{l-1} & 0 \\ 0 & 0 & \cdots & 0 & 0 & J_l \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{4.2.11}$$

with l being the number of unique eigenvalues of M, and moreover, with the values $\gamma_1, \ldots, \gamma_l$ being all the different values of eigenvalues of M we define

$$J_j = \begin{bmatrix} \gamma_j & 1 & 0 & \cdots & 0 & 0 \\ 0 & \gamma_j & 1 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 & 0 \\ 0 & 0 & \cdots & 0 & \gamma_j & 1 \\ 0 & 0 & \cdots & 0 & 0 & \gamma_j \end{bmatrix} \in \mathbb{R}^{n_j\times n_j}, \tag{4.2.12}$$

where n_j is the (algebraic) multiplicity of the eigenvalue γ_j of M. Note now that

$$\begin{aligned} & M = SJS^{-1} \\ \Longrightarrow\ & M^2 = SJS^{-1}SJS^{-1} = SJ^2S^{-1} \\ \Longrightarrow\ & M^k = SJ^kS^{-1} \\ \Longrightarrow\ & \lim_{k\to\infty} M^k = S\left(\lim_{k\to\infty} J^k\right)S^{-1}. \end{aligned} \tag{4.2.13}$$

Note also that because every block J_j is a square matrix

$$J = \begin{bmatrix} J_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & J_2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 0 & J_{l-1} & 0 \\ 0 & 0 & \cdots & 0 & 0 & J_l \end{bmatrix} \implies J^k = \begin{bmatrix} J_1^k & 0 & 0 & \cdots & 0 & 0 \\ 0 & J_2^k & 0 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 0 & J_{l-1}^k & 0 \\ 0 & 0 & \cdots & 0 & 0 & J_l^k \end{bmatrix}. \tag{4.2.14}$$

Therefore, if we can show that $\lim_{k\to\infty} J_j^k = 0_{n_j \times n_j}$ for every block $1 \leq j \leq l$, then we will be done. Note now that we can express each block J_j as a sum of two simple matrices

$$J_j = \begin{bmatrix} \gamma_j & 1 & 0 & \cdots & 0 & 0 \\ 0 & \gamma_j & 1 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 & 0 \\ 0 & 0 & \cdots & 0 & \gamma_j & 1 \\ 0 & 0 & \cdots & 0 & 0 & \gamma_j \end{bmatrix} = \begin{bmatrix} \gamma_j & 0 & 0 & \cdots & 0 & 0 \\ 0 & \gamma_j & 0 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 0 & \gamma_j & 0 \\ 0 & 0 & \cdots & 0 & 0 & \gamma_j \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}$$

$$= \gamma_j \, I_{n_j} + \, N. \tag{4.2.15}$$

Note further that N is an upper-triangular matrix, where also every entry on the main diagonal is zero. By [4] (p133), this means that

$$N^{n_j} = 0_{n_j \times n_j}. \tag{4.2.16}$$

Because $\gamma_j I_{n_j}$ is a multiple of an identity matrix, it will commute in matrix multiplication with any other $n_j \times n_j$ matrix. Specifically

$$(\gamma_j I_{n_j})N = \gamma_j N = N(\gamma_j I_{n_j}). \tag{4.2.17}$$

With this knowledge we can compute J_j^k with the use of the binomial theorem, since $\gamma_j I_{n_j}$ and N commuting means that they act like regular numbers in multiplication. That is

$$J_j^k = (\gamma_j I_{n_j} + N)^k = \sum_{i=0}^{k} \binom{k}{i} (\gamma_j I_{n_j})^{k-i} N^i = \sum_{i=0}^{k} \binom{k}{i} \gamma_j^{k-i} N^i. \tag{4.2.18}$$

We may now assume that $k \geq n_j - 1$, because we are only interested in what happens as $k \to \infty$. Since $N^i = 0_{n_j \times n_j}$ for $i \geq n_j$, we have that

$$J_j^k = \sum_{i=0}^{n_j-1} \binom{k}{i} \gamma_j^{k-i} N^i \tag{4.2.19}$$

is a finite sum. We may therefore move our limit inside the sum, resulting in

$$\lim_{k\to\infty} J_j^k = \lim_{k\to\infty} \sum_{i=0}^{n_j-1} \binom{k}{i} \gamma_j^{k-i} N^i = \sum_{i=0}^{n_j-1} \left(\lim_{k\to\infty} \binom{k}{i} \gamma_j^{k-i} \right) N^i. \tag{4.2.20}$$

We are done if this is the zero matrix, which is true if and only if $\lim_{k\to\infty} \binom{k}{i} \gamma_j^{k-i} = 0$ for every term $1 \leq i \leq n_j - 1$. Keep in mind that we have assumed that $\rho(M) < 1$ meaning that $|\gamma_j| < 1$ for every $1 \leq j \leq l$. We have

$$\begin{aligned} 0 &\leq \left| \binom{k}{i} \gamma_j^{k-i} \right| = \left| \frac{k!}{i!(k-i)!} \gamma_j^{k-i} \right| = \frac{k!}{i!(k-i)!} \left| \gamma_j^{k-i} \right| = \frac{k!}{i!(k-i)!} |\gamma_j|^{k-i} \\ &\leq \frac{k!}{(k-i)!} |\gamma_j|^{k-i} = k \dots (k-i+1) |\gamma_j|^{k-i} \leq k^i \left| \gamma_j \right|^{k-i}. \end{aligned} \tag{4.2.21}$$

Note that k^i grows like a polynomial, while $|\gamma_j|^{k-i} < 1$ decreases exponentially. This is a standard limit [8] (p160) with value

$$\lim_{k\to\infty} k^i \left| \gamma_j \right|^{k-i} = 0. \tag{4.2.22}$$

This means that

$$\begin{aligned} 0 \leq \left| \binom{k}{i} \gamma_j^{k-i} \right| &\leq k^i \left| \gamma_j \right|^{k-i} \\ \implies \lim_{k\to\infty} 0 \leq \lim_{k\to\infty} \left| \binom{k}{i} \gamma_j^{k-i} \right| &\leq \lim_{k\to\infty} k^i \left| \gamma_j \right|^{k-i} \\ \implies 0 \leq \lim_{k\to\infty} \left| \binom{k}{i} \gamma_j^{k-i} \right| &\leq 0 \\ \implies \lim_{k\to\infty} \left| \binom{k}{i} \gamma_j^{k-i} \right| &= 0 \\ \implies \lim_{k\to\infty} \binom{k}{i} \gamma_j^{k-i} &= 0. \end{aligned} \tag{4.2.23}$$

Therefore we have

$$\lim_{k\to\infty} J_j^k = \sum_{i=0}^{n_j-1} \left(\lim_{k\to\infty} \binom{k}{i} \gamma_j^{k-i} \right) N^i = \sum_{i=0}^{n_j-1} 0 N^i = 0_{n_j \times n_j}. \tag{4.2.24}$$

Hence $\lim_{k\to\infty} J_j^k = 0_{n_j\times n_j}$ for every block $1 \leq j \leq l$ of the Jordan normal form J. This means by equation (4.2.14) that

$$\lim_{k\to\infty} J^k = 0_{n\times n}. \tag{4.2.25}$$

From equation (4.2.13) we conclude that

$$\lim_{k\to\infty} M^k = S\left(\lim_{k\to\infty} J^k\right) S^{-1} = S0_{n\times n}S^{-1} = 0_{n\times n}. \tag{4.2.26}$$

(only if, $\implies$). Let $\lim_{k\to\infty} M^k = 0_{n\times n}$. Let an eigenvalue of M be γ with $u \neq \mathbf{0}$ as a corresponding eigenvector. This means that

$$\begin{aligned} & Mu = \gamma u \\ \implies & M^2u = \gamma Mu = \gamma^2 u \\ \implies & M^k u = \gamma^k u. \end{aligned} \tag{4.2.27}$$

Hence

$$\begin{aligned} & \lim_{k\to\infty} M^k = 0_{n\times n} \\ \implies & \lim_{k\to\infty} M^k u = 0_{n\times n}u = \mathbf{0} \\ \implies & \lim_{k\to\infty} \gamma^k u = \mathbf{0} \\ \implies & \lim_{k\to\infty} \gamma^k = 0 \\ \implies & |\gamma| < 1. \end{aligned} \tag{4.2.28}$$

Since this holds for any eigenvalue of M, this means that

$$\max_{\gamma \text{ eigenvalue of } M} |\gamma| < 1 \iff \rho(M) < 1. \tag{4.2.29}$$

ο.ε.δ.

Now wish to study the convergence of the Landweber iteration more specifically than we did with the general gradient descent. We will consider a *fixed step size* $s^{(k)} = s$ and wish to find the step size that gives the best conservative *convergence rate* of the method.

Lemma 4.2.5. *If the Landweber iteration converges to* $\lim_{k\to\infty} x^{(k)} = x^*$*, then* x^* *fulfills the normal equation:* $A^TAx^* = A^Tb$*.*

Proof. Let $\lim_{k\to\infty} x^{(k)} = x^*$. By the definition of the Landweber iteration

$$\begin{aligned} & x^{(k+1)} = x^{(k)} - s^{(k)}A^T(Ax^{(k)} - b) \\ \implies & \lim_{k\to\infty} x^{(k+1)} = \lim_{k\to\infty} x^{(k)} - s^{(k)}A^T(Ax^{(k)} - b) \\ \implies & x^* = x^* - sA^T(Ax^* - b) \end{aligned} \tag{4.2.30}$$

and at the same time

$$\begin{aligned} & x^* = x^* - sA^T(Ax^* - b) \\ \iff \quad & x^* = x^* - sA^TAx^* + sA^Tb \\ \iff \quad & \mathbf{0} = -sA^TAx^* + sA^Tb \\ \iff \quad & sA^TAx^* = sA^Tb \\ \iff \quad & A^TAx^* = A^Tb. \end{aligned} \tag{4.2.31}$$

o.ε.δ.

Lemma 4.2.6. *In the Landweber iteration* $(x^{(k+1)} - x^*) = (I_n - sA^TA)(x^{(k)} - x^*)$*.*

Proof. Let $x^* \in \mathbb{R}^n$ be a point that the Landweber iteration *could* converge to, using the step size $s^{(k)} = s$. By lemma 4.2.5, this means that x^* fulfills the normal equation. From the definition of the Landweber iteration we have that

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - sA^T(Ax^{(k)} - b) \\ &= x^{(k)} - sA^TAx^{(k)} + sA^Tb \\ &= x^{(k)} - sA^TAx^{(k)} + sA^TAx^* \\ &= (I_n - sA^TA)x^{(k)} + s(A^TAx^*). \end{aligned} \tag{4.2.32}$$

Therefore

$$\begin{aligned} (x^{(k+1)} - x^*) &= (I_n - sA^TA)x^{(k)} + s(A^TAx^*) - x^* \\ &= (I_n - sA^TA)x^{(k)} + (sA^TA - I_n)x^* \\ &= (I_n - sA^TA)x^{(k)} - (I_n - sA^TA)x^* \\ &= (I_n - sA^TA)(x^{(k)} - x^*). \end{aligned} \tag{4.2.33}$$

o.ε.δ.

Theorem 4.2.7. *For a fixed step-size s, the Landweber iteration converges if and only if $\rho(I_n - sA^TA) < 1$.*

Proof. Note first that the Landweber iteration converges to a point x^* if and only if $\lim_{k\to\infty} x^{(k)} = x^*$. Now let the error vector at step k of the Landweber iteration be $e^{(k)} = x^{(k)} - x^*$. We have that

$$\begin{aligned} &\lim_{k\to\infty} x^{(k)} = x^* \\ \iff &\lim_{k\to\infty} (x^{(k)} - x^*) = \mathbf{0} \\ \iff &\lim_{k\to\infty} e^{(k)} = \mathbf{0}\,. \end{aligned} \tag{4.2.34}$$

By lemma 4.2.6, in the Landweber iteration

$$\begin{aligned} &e^{(k+1)} = (I_n - sA^TA)e^{(k)} \\ \iff &e^{(k+1)} = (I_n - sA^TA)^{k+1}e^{(0)}. \end{aligned} \tag{4.2.35}$$

We now assume that $e^{(0)} \neq \mathbf{0}$ because otherwise our initial guess $x^{(0)}$ equals the solution x^*, which is not interesting because then the Landweber iteration would not be needed. This means that the Landweber iteration converges if and only if

$$\begin{aligned} &\lim_{k\to\infty} e^{(k+1)} = \mathbf{0} \\ \iff &\mathbf{0} = \lim_{k\to\infty} (I_n - sA^TA)^{k+1}e^{(0)} \\ \iff &\lim_{k\to\infty} (I_n - sA^TA)^{k+1} = 0_{n\times n}. \end{aligned} \tag{4.2.36}$$

By theorem 4.2.4 this is equivalent to

$$\rho(I_n - sA^TA) < 1. \tag{4.2.37}$$

ο.ε.δ.

Now we wish to find the fixed step size that produces the best conservative convergence rate for the Landweber iteration. This means that we wish to find a step size s that *minimizes* $\rho(I_n - sA^TA)$.

Theorem 4.2.8. *$s = \frac{2}{\lambda_1+\lambda_n}$ is the unique minimizer of $\rho(I_n - sA^TA)$, with minimum value $\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}$, where $\lambda_1 > 0$ is a largest eigenvalue of A^TA, and λ_n is a smallest eigenvalue of A^TA.*

Remark. We assume $\lambda_1 > 0$, because otherwise A^TA is the zero matrix, which is not interesting. If $\lambda_n = 0$ take "$\frac{1}{\lambda_n}$" to mean $+\infty$.

Proof. By [4] (p233), the eigenvalues of A^TA are real and non-negative, and can therefore be ordered: $\lambda_1 \geq \ldots \geq \lambda_n \geq 0$. By lemma 3.2.3, the eigenvalues of $(I_n - sA^TA)$ are $1 - s\lambda_n \geq \ldots \geq 1 - s\lambda_1$, since multiplying a matrix by a scalar multiplies its eigenvalues by the same scalar. We have that

$$\begin{aligned}\rho(I_n - sA^TA) &= \max\{|1 - s\lambda_n|, \ldots, |1 - s\lambda_1|\} \\ &= \max\{|1 - s\lambda_n|, |1 - s\lambda_1|\},\end{aligned} \tag{4.2.38}$$

because $1 - s\lambda_n$ is the largest eigenvalue, and $1 - s\lambda_1$ is the smallest eigenvalue, which is possibly the largest in the absolute value. Therefore

$$\min_s \rho(I_n - sA^TA) = \min_s \max\{|1 - s\lambda_n|, |1 - s\lambda_1|\}. \tag{4.2.39}$$

To solve this problem we will make a table of the signs of the derivative and values of the function

$$g(s) = \max\{|1 - s\lambda_n|, |1 - s\lambda_1|\}. \tag{4.2.40}$$

First we find out the points of intersection t between $|1 - s\lambda_n|$ and $|1 - s\lambda_1|$. Either $1 - s\lambda_n$ and $1 - s\lambda_1$ have the same sign or the opposite sign, leading to two cases.

$$\begin{aligned}1 - t_1\lambda_n &= 1 - t_1\lambda_1 \\ \implies t_1\lambda_n &= t_1\lambda_1 \\ \implies t_1(\lambda_n - \lambda_1) &= 0 \\ \implies t_1 &= 0\end{aligned} \tag{4.2.41}$$

and

$$\begin{aligned}-(1 - t_2\lambda_n) &= 1 - t_2\lambda_1 \\ \implies -1 + t_2\lambda_n &= 1 - t_2\lambda_1 \\ \implies t_2\lambda_n + t_2\lambda_1 &= 2 \\ \implies t_2 &= \frac{2}{\lambda_1 + \lambda_n}.\end{aligned} \tag{4.2.42}$$

For $s \leq 0 = t_1$, because $\lambda_1 \geq \lambda_n \geq 0$, we have that

$$\begin{aligned}-s\lambda_1 \geq -s\lambda_n \geq 0 &\implies 1 - s\lambda_1 \geq 1 - s\lambda_n \geq 1 \geq 0 \\ &\implies |1 - s\lambda_1| \geq |1 - s\lambda_n| \\ &\implies g(s) = |1 - s\lambda_1| = 1 - s\lambda_1.\end{aligned} \tag{4.2.43}$$

For $s \geq t_2 = \frac{2}{\lambda_1+\lambda_n}$, we first note that in general, if $\lambda_j > 0$, then $|1-s\lambda_j| = 0$ at $s = \frac{1}{\lambda_j}$ and $|1-s\lambda_j|$ has slope $-\lambda_j < 0$ if $s < \frac{1}{\lambda_j}$ and slope $\lambda_j > 0$ if $s > \frac{1}{\lambda_j}$. Therefore, when $t_2 \leq s \leq \frac{1}{\lambda_n}$, it is the case that $|1-s\lambda_n|$ is decreasing and $|1-s\lambda_1|$ is increasing since

$$\lambda_1 \geq \lambda_n \geq 0 \implies \lambda_1 \geq \frac{\lambda_1+\lambda_n}{2} \implies \frac{1}{\lambda_1} \leq \frac{2}{\lambda_1+\lambda_n} = t_2, \tag{4.2.44}$$

and $s \geq t_2$. Because $|1-s\lambda_n|$ and $|1-s\lambda_1|$ intersect at $s = t_2$, this means that for $t_2 \leq s \leq \frac{1}{\lambda_n}$ we have

$$|1-s\lambda_1| \geq |1-s\lambda_n| \implies g(s) = |1-s\lambda_1|. \tag{4.2.45}$$

Note further that there are only two intersection points: $t_1 = 0$ and $t_2 = \frac{2}{\lambda_1+\lambda_n} \geq 0 = t_1$, and that $|1-s\lambda_1|$ and $|1-s\lambda_n|$ are continuous functions of s. Therefore it must still be the case that $|1-s\lambda_1| \geq |1-s\lambda_n|$ for $s \geq \frac{1}{\lambda_n}$, since they do not intersect for any larger values of s than $s = t_2$. Hence, for $s \geq t_2$

$$g(s) = |1-s\lambda_1| = s\lambda_1 - 1, \tag{4.2.46}$$

because $t_2 \geq \frac{1}{\lambda_1}$. For the case $t_1 < s < t_2$ it suffices to check which of $|1-s\lambda_1|$ and $|1-s\lambda_n|$ are larger for any value of $s \in]t_1, t_2[$, since these are the only intersection points of the continuous functions $|1-s\lambda_1|$ and $|1-s\lambda_n|$ of s. We may take $s = \frac{1}{2\lambda_1}$, since $0 < \frac{1}{2\lambda_1} < \frac{1}{\lambda_1} \leq t_2$. We have

$$|1-s\lambda_1| = \left|1 - \frac{1}{2\lambda_1}\lambda_1\right| = \left|1-\frac{1}{2}\right| = \frac{1}{2}, \tag{4.2.47}$$

and, while noting that $\lambda_1 \geq \lambda_n \geq 0 \implies 0 \leq \frac{\lambda_n}{\lambda_1} \leq 1$, and $s = \frac{1}{2\lambda_1} < \frac{1}{\lambda_1} \leq \frac{1}{\lambda_n}$, we attain

$$|1-s\lambda_n| = 1-s\lambda_n = 1 - \frac{1}{2\lambda_1}\lambda_n = 1 - \frac{1}{2}\frac{\lambda_n}{\lambda_1} \geq 1 - \frac{1}{2} = \frac{1}{2} = |1-s\lambda_1|. \tag{4.2.48}$$

Therefore, when $t_1 \leq s \leq t_2$, it is the case that

$$g(s) = |1-s\lambda_n| = 1-s\lambda_n. \tag{4.2.49}$$

Now let us calculate $g(t_1)$ and $g(t_2)$. For $s \leq t_1$ we have $g(s) = 1-s\lambda_1$, so

$$g(t_1) = 1-(0)\lambda_1 = 1, \tag{4.2.50}$$

and for $s \geq t_2$ we have $g(s) = s\lambda_1 - 1$, so

$$g(t_2) = \left(\frac{2}{\lambda_1+\lambda_n}\right)\lambda_1 - 1 = \frac{2\lambda_1-\lambda_1-\lambda_n}{\lambda_1+\lambda_n} = \frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}. \tag{4.2.51}$$

We can now make the following table (and the graph seen in figure 4.1)

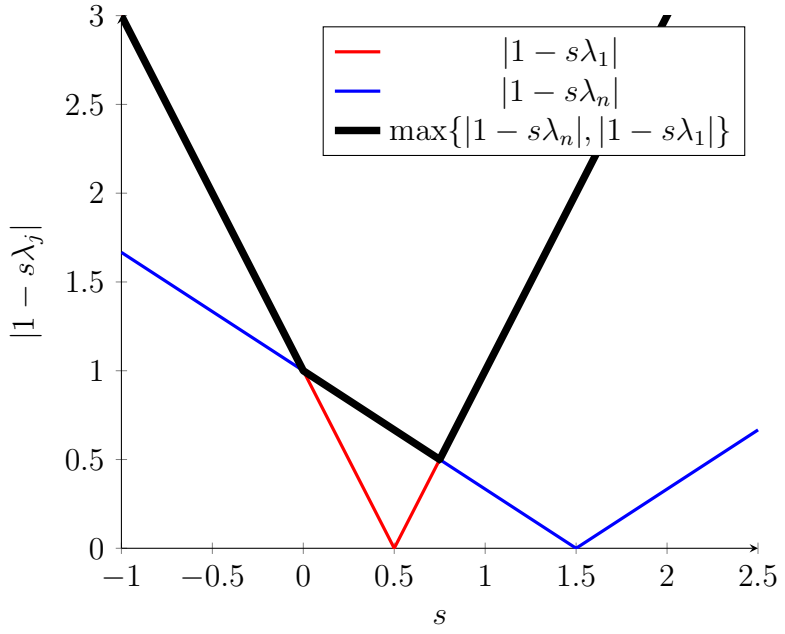

Figure 4.1: (For this render $\lambda_1 = 2 \geq \lambda_n = \frac{2}{3} \geq 0$.)

s			0		$\frac{2}{\lambda_1+\lambda_n}$	
$g'(s)$	$-\lambda_1$	$\wr$	$-\lambda_n$	$\wr$	λ_1	
$g(s)$	$1 - s\lambda_1$	1	$1 - s\lambda_n$	$\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}$	$s\lambda_1 - 1$	

and the simpler variant

s		t_1		t_2		
$g'(s)$	$-$	$\wr$	$-$	$\wr$	$+$	.
$g(s)$	$\searrow$	1	$\searrow$	$\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}$	$\nearrow$	

This means that $s = t_2 = \frac{2}{\lambda_1+\lambda_n}$ is the unique minimizer of the function $g(s) = \max\{|1 - s\lambda_n|, |1 - s\lambda_1|\} = \rho(I_n - sA^TA)$ with minimum value $\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}$. That is

$$\min_s \rho(I_n - sA^TA) = \frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}, \tag{4.2.52}$$

uniquely, for $s = \frac{2}{\lambda_1+\lambda_n}$.

ο.ε.δ.

Remark. In terms of the singular values of A, this means that the optimal fixed step size for the Landweber iteration is $s = \frac{2}{\sigma_1^2+\sigma_n^2}$ with best conservative convergence rate being $\frac{\sigma_1^2-\sigma_n^2}{\sigma_1^2+\sigma_n^2}$.

Corollary 4.2.9. *The Landweber iteration converges if $s^{(k)} = s = \frac{2}{\lambda_1+\lambda_n}$, where $\lambda_1 > 0$ is a largest eigenvalue of A^TA, and $\lambda_n > 0$ (strictly larger than 0) is a smallest eigenvalue of A^TA.*

Proof. By theorem 4.2.8 we have that if $s = \frac{2}{\lambda_1+\lambda_n}$ then $\rho(I_n - sA^TA) = \frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}$, where $\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n} < 1$ since $\lambda_1 \geq \lambda_n > 0$. By theorem 4.2.7 we have that this means that the Landweber iteration converges. ο.ε.δ.

4.3 Polyak heavy ball

Before we introduce the *Polyak heavy ball* method, we shall first go though an example showing a weakness of gradient descent. That is, that gradient descent sometimes experiences a "zig-zag phenomenon", which the Polyak heavy ball method tries to counteract.

Example 4.3.1. Take $f(x;b) = \frac{1}{2}(x_1^2 + bx_2^2)$, where $0 < b \leq 1$ is a "small" number. (The graph of f will be like an oblong bowl over the x_1-x_2 plane.) This is clearly minimized at $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ with minimum value 0. To find this minimizer using gradient descent we do for $k \geq 0$ the iteration

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - s^{(k)}\nabla f(x^{(k)}) = x^{(k)} - s^{(k)} \begin{bmatrix} x_1^{(k)} \\ bx_2^{(k)} \end{bmatrix} \\ &= x^{(k)} - s^{(k)} \begin{bmatrix} 1 & 0 \\ 0 & b \end{bmatrix} x^{(k)} = \begin{bmatrix} 1 - s^{(k)} & 0 \\ 0 & 1 - s^{(k)}b \end{bmatrix} x^{(k)}. \end{aligned} \tag{4.3.1}$$

"Exact line search" gives the optimal step size $s^{(k)} = s = \frac{2}{1+b}$ [12] (p348). Therefore, if we start at the point $x^{(0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and use this step size, then

$$x^{(1)} = \begin{bmatrix} 1 - \frac{2}{1+b} & 0 \\ 0 & 1 - \frac{2}{1+b}b \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{b-1}{b+1} & 0 \\ 0 & \frac{1-b}{1+b} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{4.3.2}$$

And so

$$x^{(n)} = \begin{bmatrix} \frac{b-1}{b+1} & 0 \\ 0 & \frac{1-b}{1+b} \end{bmatrix}^n \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \left(\frac{b-1}{b+1}\right)^n & 0 \\ 0 & \left(\frac{1-b}{1+b}\right)^n \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \left(\frac{b-1}{b+1}\right)^n \\ \left(\frac{1-b}{1+b}\right)^n \end{bmatrix}. \tag{4.3.3}$$

Because $0 < b \leq 1$, if $b = 1$ we immediately arrive at the minimizer $(x_1, x_2) = (0, 0)$ in $x^{(1)}$. However, if b is "small", then $\left(\frac{1-b}{1+b}\right)$ is a positive number close to 1, meaning

that x_1 will converge to 0 at a somewhat slow rate. At the same time $\left(\frac{b-1}{b+1}\right)$ will be a negative number close to -1. Therefore, the value of x_2 will alternate between a negative and positive value in each iteration, but also converging at a somewhat slow rate to 0. The path of (x_1, x_2) over each iteration of gradient descent will therefore be in a "zig-zag" shape down to the minimizer $(x_1, x_2) = (0, 0)$.

Note that in the oblong bowl shape of the graph of f, it is a straight path down to the minimizer $(x_1, x_2) = (0, 0)$. However, as we have seen, in gradient descent, the path taken to the minimizer is *not* straight, but instead is a zig-zag path. This is clearly not optimal. (See Figure VI.9 on page 349 of [12].)

The *Polyak heavy ball* method counteracts this problem by introducing one more term than in gradient descent. This term can be called the "*momentum term*" given by $\beta^{(k)}(x^{(k)} - x^{(k-1)})$, with *momentum* $\beta^{(k)}$. Note that

$$(x^{(k)} - x^{(k-1)}) = (x^{(k-1)} - s^{(k-1)}\nabla f(x^{(k-1)})) - x^{(k-1)} = -s^{(k-1)}\nabla f(x^{(k-1)}). \quad (4.3.4)$$

Therefore the momentum term is precisely $\beta^{(k)}$ times the change made in the last step. The idea is that this will make the result of the next step (k) not be too radically different from the last step $(k-1)$. This will make zig-zag less prominent.

The results generated from Polyak heavy ball can therefore be thought of as moving in "the path of a heavy ball", where, the higher momentum we are considering, the heavier the ball. This is because a heavy ball does not deviate much from a straight path down, compared to a light ball, which might zig-zag.

4.3.1 Definition of Polyak heavy ball

Polyak heavy ball is a method for finding a local minimum of some function of the type $f : \mathbb{R}^n \to \mathbb{R}$ that we wish to minimize. The method is named after Russian mathematician Boris Teodorovich Polyak [9] (p65).

Definition 4.3.2. Let $f : \mathbb{R}^n \to \mathbb{R}$. Let $x^{(0)} \in \mathbb{R}^n$ be arbitrary, and let $x^{(1)} = x^{(0)}$. For $k \geq 1$, one iteration of Polyak heavy ball is

$$x^{(k+1)} = x^{(k)} + \beta^{(k)}(x^{(k)} - x^{(k-1)}) - s^{(k)}\nabla f(x^{(k)}),$$

where $s^{(k)} > 0$ is the *step size* and $\beta^{(k)} > 0$ is the *momentum* at discrete time k.

4.4 Heavy ball to solve least squares problems

Now we wish to use the Polyak heavy ball method in order to try to make the Landweber iteration converge faster. That is, we will consider the function $f(x) = \frac{1}{2}\|b - Ax\|_2^2$ in Polyak heavy ball in order to solve the least squares problem given by $\min_x \|b - Ax\|_2^2$.

4.4.1 Definition of the Landweber iteration with momentum

Definition 4.4.1. Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Let $x^{(0)} \in \mathbb{R}^n$ be arbitrary, and let $x^{(1)} = x^{(0)}$. For $k \geq 1$, one iteration of the *Landweber iteration with momentum* is

$$x^{(k+1)} = x^{(k)} + \beta^{(k)}(x^{(k)} - x^{(k-1)}) - s^{(k)}A^T(Ax^{(k)} - b),$$

where $s^{(k)} > 0$ is the *step size* and $\beta^{(k)} > 0$ is the *momentum* at discrete time k.

4.4.2 Analysing the Landweber iteration with momentum

First we cover some similar concepts as with the Landweber iteration, where we again will be considering a fixed step size $s^{(k)} = s$, but also a fixed momentum $\beta^{(k)} = \beta$.

Lemma 4.4.2. *If the Landweber iteration with momentum converges to* $\lim_{k\to\infty} x^{(k)} = x^*$, *then* x^* *fulfills the normal equation:* $A^TAx^* = A^Tb$.

Proof. Let $\lim_{k\to\infty} x^{(k)} = x^*$. By the definition of the Landweber iteration with momentum

$$\begin{aligned} & x^{(k+1)} = x^{(k)} + \beta(x^{(k)} - x^{(k-1)}) - sA^T(Ax^{(k)} - b) \\ \implies & \lim_{k\to\infty} x^{(k+1)} = \lim_{k\to\infty} x^{(k)} + \beta(x^{(k)} - x^{(k-1)}) - sA^T(Ax^{(k)} - b) \\ \implies & x^* = x^*\beta(x^* - x^*) - sA^T(Ax^* - b) \\ \implies & x^* = x^* - sA^T(Ax^* - b). \end{aligned} \tag{4.4.1}$$

By equation (4.2.31) this means that x^* fulfills the normal equation. ο.ε.δ.

Because the Landweber iteration with momentum uses two steps at a time, it becomes quite different to analyze from now on. (It is different in a similar way to how a second order differential equation is different to a first order differential equation [12] (p352).)

Lemma 4.4.3. *In the Landweber iteration with momentum*

$$\begin{bmatrix} x^{(k+1)} - x^* \\ x^{(k)} - x^* \end{bmatrix} = \begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix} \begin{bmatrix} x^{(k)} - x^* \\ x^{(k-1)} - x^* \end{bmatrix}.$$

Proof. Let $x^* \in \mathbb{R}^n$ be a point of that the Landweber iteration with momentum *could* converge to, using the step size $s^{(k)} = s$ and momentum $\beta^{(k)} = \beta$. By lemma 4.4.2,

this means that x^* fulfills the normal equation $A^TAx^* = A^Tb$. By the definition of the Landweber iteration with momentum

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \beta(x^{(k)} - x^{(k-1)}) - sA^T(Ax^{(k)} - b) \\ &= x^{(k)} + \beta x^{(k)} - \beta x^{(k-1)} - sA^TAx^{(k)} + sA^Tb \\ &= x^{(k)} + \beta x^{(k)} - \beta x^{(k-1)} - sA^TAx^{(k)} + sA^TAx^*. \end{aligned} \tag{4.4.2}$$

This means that

$$\begin{aligned} (x^{(k+1)} - x^*) &= x^{(k)} - x^* + \beta x^{(k)} - \beta x^{(k-1)} - sA^TAx^{(k)} + sA^TAx^* \\ &= (x^{(k)} - x^*) + \beta x^{(k)} - \beta x^{(k-1)} - sA^TA(x^{(k)} - x^*) \\ &= (x^{(k)} - x^*) + \beta x^{(k)} - \beta x^{(k-1)} + (-\beta x^* + \beta x^*) - sA^TA(x^{(k)} - x^*) \\ &= (x^{(k)} - x^*) + \beta(x^{(k)} - x^*) - \beta(x^{(k-1)} - x^*) - sA^TA(x^{(k)} - x^*) \\ &= -\beta(x^{(k-1)} - x^*) + (I_n + \beta I_n - sA^TA)(x^{(k)} - x^*) \\ &= ((1+\beta)I_n - sA^TA)(x^{(k)} - x^*) - \beta I_n(x^{(k-1)} - x^*). \end{aligned} \tag{4.4.3}$$

Note also that quite trivially

$$(x^{(k)} - x^*) = I_n(x^{(k)} - x^*) + 0_{n\times n}(x^{(k-1)} - x^*). \tag{4.4.4}$$

Putting equations (4.4.3) and (4.4.4) in matrix form yields the block matrix equation

$$\begin{bmatrix} x^{(k+1)} - x^* \\ x^{(k)} - x^* \end{bmatrix} = \begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix} \begin{bmatrix} x^{(k)} - x^* \\ x^{(k-1)} - x^* \end{bmatrix}. \tag{4.4.5}$$

ο.ε.δ.

Theorem 4.4.4. *The Landweber iteration with momentum converges if and only if*

$$\rho\left(\begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix}\right) < 1.$$

Proof. Note first that the Landweber iteration with momentum converges to a point x^* if and only if $\lim_{k\to\infty} x^{(k)} = x^*$. Now let the error vector at step k of the Landweber iteration with momentum be $e^{(k)} = x^{(k)} - x^* \in \mathbb{R}^n$. We have that

$$\begin{aligned} &\lim_{k\to\infty} x^{(k)} = x^* \\ \iff &\lim_{k\to\infty} (x^{(k)} - x^*) = \mathbf{0} \\ \iff &\lim_{k\to\infty} e^{(k)} = \mathbf{0}\,. \end{aligned} \tag{4.4.6}$$

Now let $\hat{e}^{(k)} \in \mathbb{R}^{2n}$ for $k \geq 1$ be the vector

$$\hat{e}^{(k)} = \begin{bmatrix} e^{(k)} \\ e^{(k-1)} \end{bmatrix}. \tag{4.4.7}$$

Note that

$$\begin{aligned} & \lim_{k\to\infty} e^{(k)} = \mathbf{0} \\ \iff & \lim_{k\to\infty} e^{(k)} = \mathbf{0} \quad \text{and} \quad \lim_{k\to\infty} e^{(k-1)} = \mathbf{0} \\ \iff & \lim_{k\to\infty} \hat{e}^{(k)} = \mathbf{0} \in \mathbb{R}^{2n}. \end{aligned} \tag{4.4.8}$$

Therefore the Landweber iteration with momentum converges if and only if we have that $\lim_{k\to\infty} \hat{e}^{(k)} = \mathbf{0}$. By lemma 4.4.3, in the Landweber iteration with momentum

$$\begin{aligned} \hat{e}^{(k+1)} &= \begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix} \hat{e}^{(k)} \\ \iff \hat{e}^{(k+1)} &= \begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix}^k \hat{e}^{(1)}. \end{aligned} \tag{4.4.9}$$

We now assume that $e^{(0)} \neq \mathbf{0} \in \mathbb{R}^n$, $e^{(1)} \neq \mathbf{0} \in \mathbb{R}^n$ because otherwise our initial guess $x^{(0)} = x^{(1)}$ (where this equality holds by definition) equals the solution x^*, which is not interesting because then the Landweber iteration with momentum would not be needed. This means that $\hat{e}^{(1)} \neq \mathbf{0} \in \mathbb{R}^{2n}$ and the Landweber iteration converges if and only if

$$\begin{aligned} & \lim_{k\to\infty} \hat{e}^{(k+1)} = \mathbf{0} \\ \iff & \mathbf{0} = \lim_{k\to\infty} \begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix}^k e^{(1)} \\ \iff & \lim_{k\to\infty} \begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix}^k = 0_{(2n)\times(2n)}. \end{aligned} \tag{4.4.10}$$

By theorem 4.2.4 this is equivalent to

$$\rho\left(\begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix}\right) < 1. \tag{4.4.11}$$

ο.ε.δ.

Lemma 4.4.5. *Given that $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A^TA,*

$$\rho\left(\begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix}\right) = \max_{1\leq i\leq n} \rho\left(\begin{bmatrix} 1+\beta-s\lambda_i & -\beta \\ 1 & 0 \end{bmatrix}\right).$$

Proof. Note that $A^TA \in \mathbb{R}^{n\times n}$ is symmetric because $(A^TA)^T = (A)^T(A^T)^T = A^TA$. By the spectral theorem, this means that A^TA is orthogonally diagonalizable as

$$A^TA = Q\Lambda Q^T, \tag{4.4.12}$$

where $Q \in \mathbb{R}^{n\times n}$ is an orthogonal matrix and $\Lambda \in \mathbb{R}^{n\times n}$ is a diagonal matrix with the eigenvalues $\lambda_1, \ldots, \lambda_n$ of A^TA on its main diagonal. Because Q is square and orthogonal $Q^TQ = I_n \implies Q^T = Q^{-1} \implies QQ^T = I_n$. Hence

$$\begin{aligned}
M &= \begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix} \\
&= \begin{bmatrix} (1+\beta)I_n - Q\Lambda Q^T & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix} \\
&= \begin{bmatrix} (1+\beta)QQ^T - Q\Lambda Q^T & -\beta QQ^T \\ QQ^T & 0_{n\times n} \end{bmatrix} \\
&= \begin{bmatrix} Q((1+\beta)I_n - \Lambda)Q^T & Q(-\beta I_n)Q^T \\ Q(I_n)Q^T & Q(0_{n\times n})Q^T \end{bmatrix} \\
&= \begin{bmatrix} Q((1+\beta)I_n - \Lambda) & Q(-\beta I_n) \\ Q(I_n) & Q(0_{n\times n}) \end{bmatrix} \begin{bmatrix} Q^T & 0 \\ 0 & Q^T \end{bmatrix} \\
&= \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} (1+\beta)I_n - s\Lambda & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix}^T \\
&= \hat{Q}L\hat{Q}^T,
\end{aligned} \tag{4.4.13}$$

where

$$\hat{Q} = \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \in \mathbb{R}^{(2n)\times(2n)}, \quad L = \begin{bmatrix} (1+\beta)I_n - s\Lambda & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix} \in \mathbb{R}^{(2n)\times(2n)}. \tag{4.4.14}$$

(Keep in mind that the zero blocks in $\hat{Q}$ really are $0_{n\times n}$.) Note now that $\hat{Q} \in \mathbb{R}^{(2n)\times(2n)}$ is a square orthogonal matrix, and therefore invertible. This means that M and L are similar matrices, and so

$$\rho(M) = \rho(\hat{Q}L\hat{Q}^T) = \rho(L). \tag{4.4.15}$$

Note now that permutation matrices are invertible and that they are their own inverse. This means that L will be similar to $P_{2n}\ldots P_1LP_1\ldots P_{2n}$, where each P_j is a permutation matrix, permuting the rows of L from the left and the columns of L from the right. Specifically we may permute L to the similar matrix

$$B = \begin{bmatrix} B_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & B_n \end{bmatrix} \in \mathbb{R}^{(2n)\times(2n)}, \tag{4.4.16}$$

where for $1 \leq i \leq n$

$$B_i = \begin{bmatrix} 1+\beta-s\lambda_i & -\beta \\ 1 & 0 \end{bmatrix} \in \mathbb{R}^{2\times 2}. \tag{4.4.17}$$

Moreover, by [2] (p318), because the blocks of B are square and lie on the main diagonal, this means that the $2n$ eigenvalues of B is the union of each of the 2 eigenvalues of the n blocks $B_1 \dots B_n$. Therefore, a largest eigenvalue in absolute value of B is a largest eigenvalue in absolute value of one of the blocks, meaning

$$\rho(L) = \rho(B) = \max_{1\leq i\leq n} \rho(B_i). \tag{4.4.18}$$

Finally, by equation (4.4.15) this means that

$$\begin{aligned} &\rho(M) = \rho(L) = \max_{1\leq i\leq n} \rho(B_i) \\ \iff &\rho\left(\begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix}\right) = \max_{1\leq i\leq n} \rho\left(\begin{bmatrix} 1+\beta-s\lambda_i & -\beta \\ 1 & 0 \end{bmatrix}\right). \end{aligned} \tag{4.4.19}$$

ο.ε.δ.

Lemma 4.4.6. *The eigenvalues of*

$$B_i = \begin{bmatrix} 1+\beta-s\lambda_i & -\beta \\ 1 & 0 \end{bmatrix}$$

are

$$\begin{aligned} \mu_1 &= \frac{1}{2}\left((1+\beta-s\lambda_i) + \sqrt{(1+\beta-s\lambda_i)^2 - 4\beta}\right), \\ \mu_2 &= \frac{1}{2}\left((1+\beta-s\lambda_i) - \sqrt{(1+\beta-s\lambda_i)^2 - 4\beta}\right). \end{aligned}$$

Proof. By the definition of eigenvalues, μ is an eigenvalue of B_i if and only if

$$\begin{aligned} \det(B_i - \mu I_2) = 0 &\iff \det\left(\begin{bmatrix} 1+\beta-s\lambda_i-\mu & -\beta \\ 1 & -\mu \end{bmatrix}\right) = 0 \\ &\iff -(1+\beta-s\lambda_i-\mu)\mu + \beta = 0 \\ &\iff -(-\mu + (1+\beta-s\lambda_i))\mu + \beta = 0 \\ &\iff \mu^2 - \mu(1+\beta-s\lambda_i) + \beta = 0. \end{aligned} \tag{4.4.20}$$

The desired result is attained by use of the quadratic formula. ο.ε.δ.

Lemma 4.4.7. *The eigenvalues of*

$$B_i = \begin{bmatrix} 1+\beta-s\lambda_i & -\beta \\ 1 & 0 \end{bmatrix}$$

are not real numbers if and only if s fulfills

$$(1-\sqrt{\beta})^2 < s\lambda_i < (1+\sqrt{\beta})^2.$$

Proof. By lemma 4.4.6, the eigenvalues of B_i are not real numbers if and only if

$$\begin{aligned} & (1+\beta-s\lambda_i)^2 - 4\beta < 0 \\ \iff & (1+\beta-s\lambda_i)^2 - (2\sqrt{\beta})^2 < 0 \\ \iff & (1+\beta-s\lambda_i-2\sqrt{\beta})(1+\beta-s\lambda_i+2\sqrt{\beta}) < 0 \\ \iff & ((1-2\sqrt{\beta}+\beta)-s\lambda_i)((1+2\sqrt{\beta}+\beta)-s\lambda_i) < 0 \\ \iff & ((1-\sqrt{\beta})^2-s\lambda_i)((1+\sqrt{\beta})^2-s\lambda_i) < 0. \end{aligned} \tag{4.4.21}$$

Note that $(1-\sqrt{\beta})^2 < (1+\sqrt{\beta})^2$. Therefore we wish that

$$(1-\sqrt{\beta})^2 - s\lambda_i < 0 \quad \text{and} \quad (1+\sqrt{\beta})^2 - s\lambda_i > 0 \tag{4.4.22}$$

for $((1-\sqrt{\beta})^2 - s\lambda_i)((1+\sqrt{\beta})^2 - s\lambda_i)$ to be negative. Hence

$$\begin{aligned} & (1-\sqrt{\beta})^2 < s\lambda_i \quad \text{and} \quad (1+\sqrt{\beta})^2 > s\lambda_i \\ \iff & (1-\sqrt{\beta})^2 \ < \ s\lambda_i \ < \ (1+\sqrt{\beta})^2. \end{aligned} \tag{4.4.23}$$

ο.ε.δ.

Lemma 4.4.8. *If s fulfills $(1-\sqrt{\beta})^2 \leq s\lambda_i \leq (1+\sqrt{\beta})^2$ and μ_1 and μ_2 are the eigenvalues of*

$$B_i = \begin{bmatrix} 1+\beta-s\lambda_i & -\beta \\ 1 & 0 \end{bmatrix}$$

then

$$|\mu_1| = |\mu_2| = \sqrt{\beta}.$$

Remark. This means that $\rho(B_i)$ is independent of λ_i.

Proof. Let s fulfill $(1-\sqrt{\beta})^2 \leq s\lambda_i \leq (1+\sqrt{\beta})^2$. By lemma 4.4.7 and lemma 4.4.6 this means that

$$\Delta = (1+\beta-s\lambda_i)^2 - 4\beta \leq 0 \tag{4.4.24}$$

and

$$\mu_1 = \frac{1}{2}\left((1+\beta-s\lambda_i) + \sqrt{\Delta}\right) = \frac{1}{2}\left((1+\beta-s\lambda_i) + i\sqrt{-\Delta}\right), \tag{4.4.25}$$

$$\mu_2 = \frac{1}{2}\left((1+\beta-s\lambda_i) - \sqrt{\Delta}\right) = \frac{1}{2}\left((1+\beta-s\lambda_i) - i\sqrt{-\Delta}\right). \tag{4.4.26}$$

By the definition of absolute value for complex numbers

$$\begin{aligned} |\mu_2|^2 &= \frac{1}{4}\left((1+\beta-s\lambda_i)^2 + (-\sqrt{-\Delta})^2\right) \\ &= \frac{1}{4}\left((1+\beta-s\lambda_i)^2 + (\sqrt{-\Delta})^2\right) \\ &= |\mu_1|^2. \end{aligned} \tag{4.4.27}$$

Therefore

$$\begin{aligned} |\mu_1|^2 = |\mu_2|^2 &= \frac{1}{4}\left((1+\beta-s\lambda_i)^2 + (\sqrt{-\Delta})^2\right) \\ &= \frac{1}{4}\left((1+\beta-s\lambda_i)^2 - \Delta\right) \\ &= \frac{1}{4}\left((1+\beta-s\lambda_i)^2 - \left((1+\beta-s\lambda_i)^2 - 4\beta\right)\right) \\ &= \frac{1}{4}\left((1+\beta-s\lambda_i)^2 - (1+\beta-s\lambda_i)^2 + 4\beta\right) \\ &= \frac{1}{4}(4\beta) \\ &= \beta. \end{aligned} \tag{4.4.28}$$

Hence

$$|\mu_1| = |\mu_2| = \sqrt{\beta}. \tag{4.4.29}$$

o.ε.δ.

Corollary 4.4.9. *Given that $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of $A^T A$, if the step size s fulfills $(1-\sqrt{\beta})^2 \leq s\lambda_i \leq (1+\sqrt{\beta})^2$ then*

$$\rho\left(\begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix}\right) = \sqrt{\beta}.$$

Proof. By lemma 4.4.5 and lemma 4.4.8

$$\begin{aligned} \rho\left(\begin{bmatrix} (1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n} \end{bmatrix}\right) &= \max_{1\leq i\leq n} \rho\left(\begin{bmatrix} 1+\beta-s\lambda_i & -\beta \\ 1 & 0 \end{bmatrix}\right) \\ &= \max_{1\leq i\leq n}\{|\mu_1|, |\mu_2|\} \\ &= \max_{1\leq i\leq n}\{\sqrt{\beta}, \sqrt{\beta}\} \\ &= \sqrt{\beta}. \end{aligned} \tag{4.4.30}$$

o.ε.δ.

Theorem 4.4.10. $(s,\beta) = \left(\left(\frac{2}{\sqrt{\lambda_1}+\sqrt{\lambda_n}}\right)^2, \left(\frac{\sqrt{\lambda_1}-\sqrt{\lambda_n}}{\sqrt{\lambda_1}+\sqrt{\lambda_n}}\right)^2\right)$ *is the unique minimizer of*

$$\rho\left(\begin{bmatrix}(1+\beta)I_n - sA^TA & -\beta I_n \\ I_n & 0_{n\times n}\end{bmatrix}\right)$$

with respect to (s,β) *with minimum value* $\frac{\sqrt{\lambda_1}-\sqrt{\lambda_n}}{\sqrt{\lambda_1}+\sqrt{\lambda_n}} = \sqrt{\beta}$*, where* $\lambda_1 > 0$ *is a largest eigenvalue of* A^TA*, and* $\lambda_n > 0$ *is a smallest eigenvalue of* A^TA*.*

Proof. See [14]. ο.ε.δ.

Remark. Compare this convergence rate with the best conservative convergence rate for the Landweber iteration, being $\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}$ as seen in theorem 4.2.8.

Remark. In terms of the singular values of A, note that $\frac{\sqrt{\lambda_1}-\sqrt{\lambda_n}}{\sqrt{\lambda_1}+\sqrt{\lambda_n}} = \frac{\sigma_1-\sigma_n}{\sigma_1+\sigma_n}$ and $\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n} = \frac{\sigma_1^2-\sigma_n^2}{\sigma_1^2+\sigma_n^2}$.

4.5 Generalizations and other methods

For us to have gradient descent produce monotonically decreasing distance to an optimum x^* of f with each iteration we only needed a condition on ∇f (as seen in theorem 4.1.4), meaning that we only need certain smoothness of f. We did *not* need f to be convex.

However, to obtain an expression for the convergence rate we even need a *stronger* version of convexity (see chapter 6 discussion). But we do have that $f(x) = \|b - Ax\|_2$ fulfills this, which is why we could obtain our results for convergence rate.

We could instead of considering $f(x) = \|b - Ax\|_2$ have considered a more general function which also fulfills this stronger version of convexity. If instead of considering $f(x) = \|b - Ax\|_2$ we would have considered $f(x) = x^TQx + q^Tx + c$, where $Q \in \mathbb{R}^{n\times n}$ is symmetric positive definite, $q \in \mathbb{R}^n$ and $c \in \mathbb{R}$, then all the same results as in the Landweber iteration and Landweber iteration with momentum are obtained except with Q substituted for A^TA [14].

Another gradient descent method not covered in this report is the *Nesterov accelerated gradient* (NAG) method, by Russian mathematician Yurii Evgen'evich Nesterov [7] (p543). It is a generalization of the Polyak heavy ball method.

A specific case of the Polyak heavy ball method is the *conjugate gradient* method, where the optimal step size $s^{(k)}$ and momentum $\beta^{(k)}$ is used at each step (see page 68 of [9]).

5 Applications to image deblurring

This chapter is based upon a project description in the course MM5016 Numerical Analysis HT21 (Fall 2021), by professor Yishao Zhou.

5.1 Modelling a deblurring problem

In this section we cover how to mathematically model the problem of "deblurring" a digital image. We assume that the image is blurry in some way (e.g. out of focus, motion blur), and that this blurry image can be attained by applying a blur to some "not blurry" image.

The unknown *not* blurry image will be encoded as a matrix X. The blur will almost be a linear transformation, meaning that we can encode it as a matrix A. The blurry image that we do have will be the matrix B. The not blurry image can then be attained by solving $AX = B$.

5.1.1 Encode digital image as a very tall matrix

Given a digital image of size $h \times w$ px, it can be flattened to one very tall image of size $(hw) \times 1$ px, meaning that it will be one pixel wide. This digital image can then be broken up into three color channels: red, green and blue, where we then have a red image, blue image, and green image, all of size $(hw) \times 1$ px.

For a pixel in a digital image, the *color brightness intensity* of a color channel is an integer between 0 and 255. The brightness intensity in each color channel, for each pixel, is a complete description of any digital image. Therefore we can encode our digital image of flattened size $(hw) \times 1$ px as a $(hw) \times 3$ matrix

$$M = \begin{bmatrix} v_{\text{red}} & v_{\text{green}} & v_{\text{blue}} \end{bmatrix} \in \mathbb{R}^{(hw)\times 3}, \tag{5.1.1}$$

where $v_c \in \mathbb{R}^{hw}$ ($c \in \{\text{red}, \text{green}, \text{blue}\}$) is a vector of the brightness intensity in color channel c for each pixel in the flattened $(hw) \times 1$ px digital image.

5.1.2 Blur as a linear transformation with rounding

Given a matrix of brightness intensities $X \in \mathbb{R}^{(hw)\times 3}$, we can think of the blurring of the corresponding image as a linear transformation of X, given by AX, where $A \in \mathbb{R}^{(hw)\times(hw)}$ is a square matrix. However, we need to make sure that the brightness intensities are integers between 0 and 255. Therefore we will introduce an

appropriate rounding function $\mathrm{rd} : \mathbb{R}^{(hw)\times 3} \to \mathbb{R}^{(hw)\times 3}$, where

$$(\mathrm{rd}(M))_{i,j} = \begin{cases} 0, & \text{if } (M)_{i,j} < 0 \\ 255, & \text{if } (M)_{i,j} > 255 \\ \mathrm{round}((M)_{i,j}), & \text{otherwise,} \end{cases} \tag{5.1.2}$$

and

$$\mathrm{round}((M)_{i,j}) = \left\lfloor (M)_{i,j} + \frac{1}{2} \right\rfloor. \tag{5.1.3}$$

The blurred image can therefore be written as $\mathrm{rd}(AX)$.

5.1.3 Setting up least squares

We can now state what problem we are trying to solve. We wish to find a matrix $X \in \mathbb{R}^{(hw)\times 3}$ satisfying

$$\mathrm{rd}(AX) = B, \tag{5.1.4}$$

where $B \in \mathbb{R}^{(hw)\times 3}$ is an encoded blurry digital image and $A \in \mathbb{R}^{(hw)\times(hw)}$ is a matrix corresponding with a linear transformation which performs the type of blurring that the image corresponding with B has.

However, this problem is clearly not solvable, because rd does not have an inverse. We can instead consider the problem

$$AX = B, \tag{5.1.5}$$

which will give an approximate solution if A is invertible. But, since A represents blurring, it will almost guaranteed not be invertible, since information is lost when blurring. If we break down the matrix $X \in \mathbb{R}^{(hw)\times 3}$ by its color channels like

$$X = \begin{bmatrix} x_{\text{red}} & x_{\text{blue}} & x_{\text{green}} \end{bmatrix}, \tag{5.1.6}$$

and also break down B by its color channels like

$$B = \begin{bmatrix} b_{\text{red}} & b_{\text{green}} & b_{\text{blue}} \end{bmatrix} \tag{5.1.7}$$

we instead have the problem of solving

$$Ax_c = b_c, \tag{5.1.8}$$

for $c \in \{\text{red}, \text{green}, \text{blue}\}$. Because A is almost guaranteed not to be invertible, $\mathbf{N}(A)$ will not be trivial. We can therefore not use the solution $\hat{x}_c = (A^TA)^{-1}A^Tb_c$ to the least squares problem

$$\min_{x_c} \|b_c - Ax_c\|_2^2,$$

since $(A^TA)^{-1}$ will not exist by theorem 2.2.2. Either an iterative method could be used to attain some approximate solution, or we could consider a regularized least squares problem

$$\min_{x_c} \|b_c - Ax_c\|_2^2 + \delta R(x_c),$$

which might produce better solutions. Now we could choose R such that this has a closed form solution we could calculate directly, or we could use an iterative method still. If we do solve this for each color channel, we obtain the solution

$$\hat{X}_\delta = \begin{bmatrix} \hat{x}_{\text{red}_\delta} & \hat{x}_{\text{green}_\delta} & \hat{x}_{\text{blue}_\delta} \end{bmatrix}, \tag{5.1.9}$$

which will correspond to an image that will be the image corresponding with B, but which has been approximately deblurred.

5.2 Solving a deblurring problem

In this section we will go through how to practically solve the least squares problem for deblurring. We will use the programming language MATLAB to implement these solutions.

5.2.1 Direct solution

If we consider the Tikhonov regularized least squares problem

$$\min_{x_c} \|b_c - Ax_c\|_2^2 + \delta\|x_c\|_2^2,$$

we can immediately obtain the solution, which by theorem 3.2.7 says that it always exists and is equal to

$$\hat{x}_{c_\delta} = (A^TA + \delta I_{hw})^{-1}A^Tb_c. \tag{5.2.1}$$

This can be done using the MATLAB command `lsqr` [5], where by theorem 3.2.2, we know that we should input the matrix

$$\begin{bmatrix} A \\ \sqrt{\delta} I_{hw} \end{bmatrix} \in \mathbb{R}^{(2hw)\times(hw)}$$

for the first argument and

$$\begin{bmatrix} b_c \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(2hw)\times 3}$$

for the second argument.

5.2.2 Iterative solution

We can solve the standard least squares problem

$$\min_{x_c} \|b_c - Ax_c\|_2^2$$

by using an iterative method like the Landweber iteration with momentum, until we converge $\hat{x}_{c_\delta}$, which by lemma 4.4.2 is a point fulfilling the normal equation $A^T A\hat{x}_{c_\delta} = A^T b$. We may now realize that doing the Landweber iteration with momentum for each of the color channels $c \in \{\text{red}, \text{green}, \text{blue}\}$ like

$$x_c^{(k+1)} = x_c^{(k)} + \beta^{(k)}(x_c^{(k)} - x_c^{(k-1)}) - s^{(k)}A^T(Ax_c^{(k)} - b_c) \tag{5.2.2}$$

is the same thing as doing

$$X^{(k+1)} = X^{(k)} + \beta^{(k)}(X^{(k)} - X^{(k-1)}) - s^{(k)}A^T(AX^{(k)} - B), \tag{5.2.3}$$

because subtractions and additions are element-wise, and

$$\begin{bmatrix} Ax_{\text{red}}^{(k)} & Ax_{\text{green}}^{(k)} & Ax_{\text{blue}}^{(k)} \end{bmatrix} = A \begin{bmatrix} x_{\text{red}}^{(k)} & x_{\text{green}}^{(k)} & x_{\text{blue}}^{(k)} \end{bmatrix} = AX^{(k)}. \tag{5.2.4}$$

Therefore, (5.2.3) is the iteration we will do in MATLAB. However, practically, a fixed point $\hat{X} = \begin{bmatrix} \hat{x}_{\text{red}} & \hat{x}_{\text{green}} & \hat{x}_{\text{blue}} \end{bmatrix}$ satisfying $A^T A\hat{X} = A^T B$ might not be reached in a finite amount of steps. Therefore we will set how many iterations we will do and hope for a good result. This will be done with a for-loop in MATLAB.

5.3 Results

The blurred digital image we encode with the matrix B can be seen in figure 5.1. Trying to solve $AX = B$ naively, *without* least squares or regularization as in equation (5.1.5), yields the image in figure 5.2. (This was done with the MATLAB command \ [6].) Time taken was about 30 seconds on my machine.

Using the direct method as described in subsection 5.2.1 yields the result in figure 5.3. Time taken was about 40 seconds on my machine. Using the iterative method as described in subsection 5.2.2 yields the result in figure 5.4. Time taken was about 40 seconds on my machine.

The MATLAB code used can be seen in section A.1 of appendix A on page 78.

Figure 5.1: The image to be deblurred. Dimensions are 539x373 px, encoded as a 201047×3 matrix B.

(*The reader is encouraged to guess what the image is supposed to depict.*)

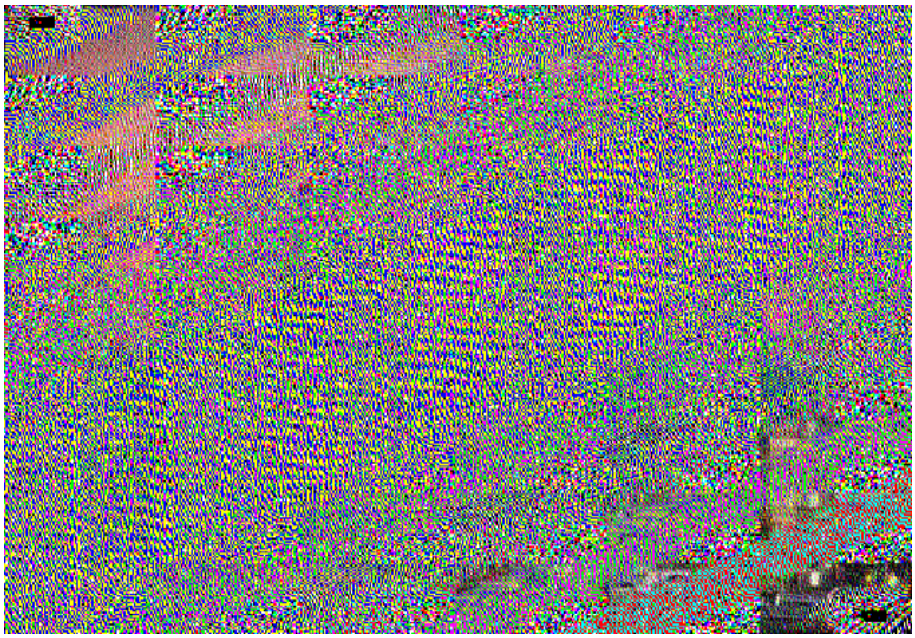

Figure 5.2: Solving $AX = B$ *without* least squares or regularization.

Figure 5.3: Solving using Tikhonov regularized least squares.

Figure 5.4: Solving using the Landweber iteration with momentum.

6 Discussion

In this chapter we will discuss the theory of gradient descent and Polyak heavy ball, as well as the practical element of this report, being deblurring.

6.1 Theory of gradient descent and Polyak heavy ball

6.1.1 Convergence in the general case

We have seen in the case in the case when $f(x) = \|b - Ax\|_2^2 = x^T(A^TA)x + (-2b^TA)x + (b^Tb)$, and in fact when $f(x) = x^TQx + q^Tx + c$, that gradient descent and Polyak heavy ball are guaranteed to converge for a fixed step size (and momentum) if and only if $\rho(M) < 1$ where M is some matrix containing A^TA or Q respectively. These are indeed special cases of the more general case when $f : \mathbb{R}^n \to \mathbb{R}$ is a twice differentiable function.

For a function $f : \mathbb{R}^n \to \mathbb{R}$ to be twice differentiable means that every entry of the *Hessian* $\nabla^2 f(x) \in \mathbb{R}^{n\times n}$ is defined for every $x \in \mathbb{R}^n$, where this is simply a matrix of every second partial derivative of f, being

$$(\nabla^2 f(x))_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}(x). \tag{6.1.1}$$

In fact, note that $\nabla^2(\|b-Ax\|_2^2) = A^TA$ and $\nabla^2(x^TQx+q^Tx+c) = Q$. Now we wish to be more general, where we first define

$$B \preccurlyeq M \iff 0 \leq x^T(M - B)x, \tag{6.1.2}$$

meaning that M is positive semi-definite is equivalent to $0_{n\times n} \preccurlyeq M$, which is more simply denoted by $0 \preccurlyeq M$. Note that in the proof of convergence of Polyak heavy ball we need the following

$$\mu I_n \preccurlyeq \nabla^2 f(x^{(k)}) \preccurlyeq LI_n, \tag{6.1.3}$$

which is true if f is μ-strongly convex ($\mu > 0$) and L-Lipschitz smooth ($L > 0$) in the case when f is a general non-linear function. Specifically this means that

$$0 < \mu \leq \lambda_n \leq \lambda_1 \leq L, \tag{6.1.4}$$

where λ_n is a smallest eigenvalue and λ_1 is a largest eigenvalue of *any* Hessian $\nabla^2 f(x^{(k)})$ in the iteration. If at any step $\lambda_1^{(k)} > L$, we do not have guaranteed convergence.

In the more specific case, we had that the best conservative convergence rate was $\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}$ for gradient descent and $\frac{\sqrt{\lambda_1}-\sqrt{\lambda_n}}{\sqrt{\lambda_1}+\sqrt{\lambda_n}}$ for Polyak heavy ball, where λ_n was a smallest and λ_1 was a largest eigenvalue of the *constant* Hessian (A^TA or Q). Note that we had guaranteed convergence when A^TA (or Q) were positive definite. In this case we will study how many iterations are needed to attain an ε-accurate solution.

6.1.2 Comparing number of iterations for an ε-accurate solution

Let us consider the case when we have a fixed step size (and momentum) in gradient descent and Polyak heavy ball used on a function of the form $f(x) = x^TQx + q^Tx + b$ (which includes $f(x) = \|b - Ax\|_2^2$), where Q (or A^TA) is symmetric positive-definite. This means that $\rho(M) = \frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n} < 1$, where

$$e^{(k)} = M^k e^{(0)}, \tag{6.1.5}$$

with $e^{(k)}$ being the error of gradient descent at step k, and that $\rho(\hat{M}) = \frac{\sqrt{\lambda_1}-\sqrt{\lambda_n}}{\sqrt{\lambda_1}+\sqrt{\lambda_n}} < 1$ where

$$\hat{e}^{(k)} = \hat{M}^k \hat{e}^{(0)}, \tag{6.1.6}$$

with $\hat{e}^{(k)}$ being the error of Polyak heavy ball at step k. Now we will say that we have reached an ε-accurate solution by step k with gradient descent if $\rho(M)^k < \varepsilon$ (and similarly if $\rho(\hat{M})^k < \varepsilon$ for Polyak heavy ball). That is, if

$$\begin{aligned} \rho(M)^k < \epsilon &\implies k\log(\rho(M)) < \log(\varepsilon) \implies -k\log(\rho(M)) > -\log(\varepsilon) \\ &\implies k\log\left(\frac{1}{\rho(M)}\right) > \log\left(\frac{1}{\varepsilon}\right). \end{aligned} \tag{6.1.7}$$

Since $\rho(M) < 1$ this means that $\frac{1}{\rho(M)} > 1$, so $\log\left(\frac{1}{\rho(M)}\right) > 0$. Hence

$$k > \frac{1}{\log\left(\frac{1}{\rho(M)}\right)} \log\left(\frac{1}{\varepsilon}\right). \tag{6.1.8}$$

Now to make the expressions simpler, we will introduce the following number

$$\kappa = \frac{\lambda_1}{\lambda_n} > 1 \tag{6.1.9}$$

(assuming $\lambda_1 > \lambda_n$.) Therefore

$$\rho(M) = \frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} = \frac{\kappa - 1}{\kappa + 1} \implies \frac{1}{\rho(M)} = \frac{\kappa + 1}{\kappa - 1} = \frac{\kappa - 1 + 2}{\kappa - 1} = 1 + \frac{2}{\kappa - 1} \tag{6.1.10}$$

and similarly

$$\rho(\hat{M}) = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_n}}{\sqrt{\lambda_1} + \sqrt{\lambda_n}} = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \implies \frac{1}{\rho(\hat{M})} = \frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1} = 1 + \frac{2}{\sqrt{\kappa} - 1}. \tag{6.1.11}$$

Using the first degree Taylor polynomial of the (natural) logarithm results in

$$\frac{1}{\log\left(\frac{1}{\rho(M)}\right)} = \frac{1}{\log\left(1 + \frac{2}{\kappa - 1}\right)} \approx \frac{1}{\frac{2}{\kappa - 1}} = \frac{\kappa - 1}{2}, \tag{6.1.12}$$

and similarly

$$\frac{1}{\log\left(\frac{1}{\rho(\hat{M})}\right)} \approx \frac{\sqrt{\kappa} - 1}{2}. \tag{6.1.13}$$

Hence we reach an ε-accurate solution using gradient descent and Polyak heavy ball using

$$k \in \mathcal{O}\left(\kappa \log\left(\frac{1}{\varepsilon}\right)\right) \qquad \text{and} \qquad k \in \mathcal{O}\left(\sqrt{\kappa} \log\left(\frac{1}{\varepsilon}\right)\right) \tag{6.1.14}$$

number of iterations, respectively.

6.2 Deblurring in practice

6.2.1 Direct versus iterative solution in practice

As we have seen in the results section of this report, given a blurry image of dimensions 539x373 px, a deblurred image can be attained in a reasonable time frame. However, given an image of much larger dimensions, it might take considerably more time. A similar problem is if there are many smaller images to deblur, for example in a blurry video.

In the practical case where the time taken is more limited, one could think that simply seeing what the image depicts is enough, where a high level of sharpness is not necessary. For example, in figures 5.3 and 5.4 we could tell in both cases that the image depicts Stockholm City Hall. In this case it took around 40 seconds to produce the images on my machine.

However, note that we can stop an iterative solution (Landweber with momentum) whenever we want, since we supply the number of iterations taken. This is different to the direct solution (Tikhonov regularization), which finishes only after a specified tolerance is reached [5]. (Note that if you implement the solution to

the Tikhonov regularized least squares problem yourself, then you can supply the number of iterations taken if you use an iterative method to calculate the inverse $(A^TA + \delta I_n)^{-1}$. This is actually done in the MATLAB command `lsqr` by solving $(A^TA + \delta I_n)\hat{x} = A^Tb$ using the conjugate gradient method [5], which is a specific case of the Polyak heavy ball method [9] (p68).)

In both the iterative and direct case we could tweak these parameters such that we arrive at a deblurred image (which is less sharp, but hopefully sharp enough to see what the image depicts) in a desired amount of time.

6.2.2 Deep neural network

A different approach to deblurring is using a *deep neural network*. This is a vector valued mathematical function f which takes an input vector $x^{(0)}$ and repeatedly alternates between applying an affine transformation and a entry-wise non-linear function. That is for $1 \leq k < L$

$$\begin{aligned} x^{(k)} &\mapsto z^{(k+1)} = W^{(k+1)}x^{(k)} + b^{(k+1)} \\ z^{(k+1)} &\mapsto x^{(k+1)} = \sigma(z^{(k+1)}), \end{aligned} \tag{6.2.1}$$

where L is the amount of *layers* of the deep neural network, with $f(x^{(0)}) = x^{(L)}$ being the output, $W^{(k+1)}$ is a matrix of *weights*, $b^{(k+1)}$ is a vector of *biases*, and σ is a non-linear function applying on the entries of the inputted vector $z^{(k+1)}$. A specific non-linear function is the *rectified linear unit* $\mathrm{ReLU}(z_i) = \max\{0, z_i\}$.

Given training data $D = \{(d^{(1)}, y^{(1)}), \ldots, (d^{(N)}, y^{(N)})\}$ of blurry image and corresponding sharp image pairs, both encoded as vectors, then one can define a *cost* function of the form

$$\mathrm{cost}(p; D) = \frac{1}{N}\sum_{i=1}^{N} C(p; d^{(i)}, y^{(i)}), \tag{6.2.2}$$

where p is a vector of every weight and bias. That is, a vector of every entry of every matrix $W^{(1)}, \ldots, W^{(L)}$ and vector $b^{(1)}, \ldots, b^{(L)}$. This means that we can value how good our choice of weights and biases are based on each training data point individually. (This is needed for *stochastic gradient descent*, a gradient descent method.)

Note that, if the number of layers is set, and the dimensions of the weight matrices and bias vectors are set, and the non-linear function is set, then f is completely determined by the entries of p (since p decides every weight and bias). We may therefore denote our deep neural network by f_p. A specific cost function is the

mean squared error (MSE), which has

$$C(p; d^{(i)}, y^{(i)}) = \frac{1}{M}\|y^{(i)} - f_p(d^{(i)})\|_2^2 = \frac{1}{M}\sum_{j=1}^{M}\left((y^{(i)})_j - (f_p(d^{(i)}))_j\right)^2. \tag{6.2.3}$$

This is the sum of the squares of the "pixel differences" (depending on how we encode our images as vectors) of the actual sharp image and the result of the deep neural network, where M is the dimension of the vectors we encode our images as. Hence, if we use MSE

$$\text{cost}(p; D) = \frac{1}{NM}\sum_{i=1}^{N}\|y^{(i)} - f_p(d^{(i)})\|_2^2. \tag{6.2.4}$$

To choose the weighs and biases p defining the deep neural network, we "train" the network, by simply minimizing $\text{cost}(p; D)$. This is typically done by a *gradient descent method*, where the gradient for each p in the iterative method is calculated using *backpropagation*.

If the process of "training" the deep neural network f_p is successful, then we arrive at a function f_{p^*}, where p^* makes the cost low, meaning that f_{p^*} will be able to take a blurry image d as input, and output an image y which is not very different to how a corresponding sharp image probably is. (The theory of deep neural networks is indeed deep and details are heavily out of the scope of this report.)

A more advanced neural network is a *convolutional neural network*, which is a similar type of function to a standard deep neural network. (See page 387 of [12].) For a guide to image deblurring using a a convolutional neural network in the programming language Python, see [10].

A Appendix

A.1 MATLAB code

The version of MATLAB used was MATLAB R2021a.

Code for running the program

```
% Load blurred image and blurring matrix
[B,h,w,A] = setup();

% Hyperparameter
sqrt_thelta = 10^-2;

% Paremeters for iterative method
iters = 200;
s = 2;
beta = 0.1;

% Deblur image naively, with Tikhonov, and with Landweber
tic; im1 = naive(B, A);                              t1 = toc;
tic; im2 = tikhonov(B, A, sqrt_thelta);              t2 = toc;
tic; im3 = landwebermomentum(B, A, iters, s, beta); t3 = toc;

% Create the deblurred images
figure(1); create_image(im1,h,w);
figure(2); create_image(im2,h,w);
figure(3); create_image(im3,h,w);

% Print the results
print -f1 -dpng naive.png
print -f2 -dpng tikhonov.png
print -f3 -dpng landwebermomentum.png

% Display time taken
fprintf('Naive:                   %g s\n', t1);
fprintf('Tikhonov:                %g s\n', t2);
fprintf('Landweber with momentum: %g s\n', t3);
```

Code for blurring matrix A and blurred image B

```
%   B       - height*width-by-3 matrix.  The columns corresponds to
%             the RGB components of the pixel colors.
%   height  - height of the image
%   width   - width of the image
%   A       - blurring matrix (B = A*true + noise).
%
function [B, height, width, A] = setup()
  B       = imread('blurry.png');
  height  = size(B,1);
  width   = size(B,2);
  A       = formA(fspecial('motion', 100, 25), height, width);
  B       = reshape(double(B),height*width,3);

% Form a matrix corresponding to blurring with the kernel
% specified by h.
%
function A = formA(h, height, width)
  % Construct blurring matrix
  [i,j,hij] = find(h);
  i = i-(size(h,1)+1)/2;
  j = j-(size(h,2)+1)/2;

  % Image dimensions
  N = height*width;

  % Array of pixel coordinates
  pixi = (1:height)'*ones(1,width);
  pixj = ones(height,1)*(1:width);
  pixk = reshape(1:(height*width), height, width);

  % Construct blurring matrix
  A = sparse(N,N);
  for l = 1:length(hij)
    hpixi = pixi+i(l);
    hpixj = pixj+j(l);
    hpixk = (hpixj-1)*height+hpixi;
    Iact = find(hpixi > 0 & hpixi < = height & hpixj > 0 & hpixj ...
        < = width);
    AA = sparse(pixk(Iact), hpixk(Iact), ...
        hij(l)*ones(length(Iact),1), N, N);
    A = A+AA;
  end
```

Code for naive solution

```
function X=naive(B, A)
X=A\B;
end
```

Code for Tikhonov solution

```
function X=tikhonov(B, A, sqrt_thelta)
n = length(A);

A = [A; sqrt_thelta .* speye(n)];
B = [B; zeros(n,3)];

for j=1:3  % solve for each color channel
    X(:,j) = lsqr(A, B(:,j), 10^-4, 1000);
end
end
```

Code for Landweber with momentum solution

```
function X=landwebermomentum(B, A, n, s, beta)

At = A';                          %A transpose

X = B;
last = B;

for J = 1:n
this = X;
next = this + beta.*(this - last) - s.*(At*(A*this - B));
last = this;
X = next;
end
```

Code for creating an image from a very tall matrix

```
% Display the image represented by the matrix X.  X has ...
    height*width rows
% (one per pixel) and three columns (for RGB) containing floating ...
    point
% values between 0 and 255 representing color intensities.  Any ...
    values that
% fall out of the range 0 to 255 will be bumped back into range.

function create_image(X, height, width)
  image(reshape(min(max(X,0),255),height,width,3)/255);
  axis off;
```

A.2 Omitted proofs

Lemma A.2.1. *$H_v y$ is the reflection of y in the hyperplane through $\mathbf{0}$ with normal vector v. (This is where the name Householder* reflection *matrix comes from.)*

Proof. The orthogonal projection of y onto the hyperplane through $\mathbf{0}$ with normal vector v is

$$y - \left(\frac{v^T y}{\|v\|_2}\right)\frac{v}{\|v\|_2},$$

so the reflection of y in the hyperplane with normal vector v is

$$\begin{aligned} y - 2\left(\frac{v^T y}{\|v\|_2}\right)\frac{v}{\|v\|_2} &= y - 2\frac{v}{\|v\|_2}\left(\frac{v^T y}{\|v\|_2}\right) \\ &= y - 2\frac{vv^T}{\|v\|_2^2}y \\ &= \left(I_m - 2\frac{vv^T}{\|v\|_2^2}\right)y \\ &= H_v y. \end{aligned} \tag{A.2.1}$$

ο.ε.δ.

Lemma A.2.2. *Given $A \in \mathbb{R}^{m\times n}$, then H_k is a symmetric matrix: $H_k^T = H_k$. (Note that H_k is as in definition 2.4.12.)*

Proof. By lemma 2.4.5 we have that

$$H_k^T = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix}^T = \begin{bmatrix} I_{k-1}^T & 0 \\ 0 & H_v^T \end{bmatrix} = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix} = H_k. \tag{A.2.2}$$

o.ε.δ.

Lemma A.2.3. *Given $A \in \mathbb{R}^{m\times n}$, then H_k is an orthogonal matrix: $H_k^T H_k = I_m$. (Note that H_k is as in definition 2.4.12.)*

Proof. By lemma 2.4.6 we have that

$$\begin{aligned} H_k^T H_k &= \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix}^T \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix} = \begin{bmatrix} I_{k-1}^T & 0 \\ 0 & H_v^T \end{bmatrix} \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix} \\ &= \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v^T \end{bmatrix} \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix} = \begin{bmatrix} I_{k-1}I_{k-1} & 0 \\ 0 & H_v^T H_v \end{bmatrix} \\ &= \begin{bmatrix} I_{k-1} & 0 \\ 0 & I_{m-k+1} \end{bmatrix} \\ &= I_m. \end{aligned} \tag{A.2.3}$$

o.ε.δ.

Corollary A.2.4. *Given $A \in \mathbb{R}^{m\times n}$, then H_k is an involutory matrix: $H_k^{-1} = H_k$.*

Proof. By lemma 2.4.18 and lemma 2.4.17 we have that

$$I_m = H_k^T H_k = HH \implies H_k^{-1} = H_k. \tag{A.2.4}$$

o.ε.δ.

Lemma A.2.5. *Given $A \in \mathbb{R}^{m\times n}$, then $H_t \dots H_1$, where $t = \min\{m-1, n\}$, is an orthogonal matrix. (Note that H_k, where $1 \le k \le t$, is as in definition 2.4.12.)*

Proof. Let $Q^T = H_t \dots H_1$. By definition, this is an orthogonal matrix if $Q^T Q = I_m$. By lemma 2.4.18, for $1 \le k \le t$

$$H_k^T H_k = I_m \implies H_k^T = H_k^{-1} \implies H_k H_k^T = I_m. \tag{A.2.5}$$

Therefore

$$\begin{aligned} Q^T Q &= (H_t \dots H_1)(H_t \dots H_1)^T \\ &= (H_t \dots H_1)(H_1^T \dots H_t^T) \\ &= H_t \dots I_m \dots H_t^T \\ &= H_t H_t^T \\ &= I_m. \end{aligned} \tag{A.2.6}$$

o.ε.δ.

Bibliography

[1] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[2] Stephen Friedberg, Arnold Insel, and Lawrence Spence. *Linear Algebra*. Pearson new international edition. Pearson Education Limited, 2013.

[3] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015.

[4] Anders Holst and Victor Ufnarovski. *Matrix Theory*. Studentlitteratur AB, 2014.

[5] MathWorks. *Solve system of linear equations — least-squares method*. URL: https://se.mathworks.com/help/matlab/ref/lsqr.html (visited on 05/18/2022).

[6] MathWorks. *Solve system of linear equations $Ax = B$ for x*. URL: https://se.mathworks.com/help/matlab/ref/mldivide.html (visited on 05/18/2022).

[7] Yurii Evgen'evich Nesterov. "A method of solving a convex programming problem with convergence rate $O\left(\frac{1}{k^2}\right)$". Russian. In: *Doklady Akademii Nauk*. Vol. 269. 3. Russian Academy of Sciences. 1983, pp. 543–547.

[8] Arne Persson and Lars-Christer Böiers. *Analys i en variabel*. Swedish. Studentlitteratur AB, 2010.

[9] Boris Teodorovich Polyak. *Introduction to Optimization*. New York: Optimization Software, 1987.

[10] Sovit Ranjan Rath. *Image Deblurring using convolutional neural networks and Deep Learning*. May 2020. URL: https://debuggercafe.com/image-deblurring-using-convolutional-neural-networks-and-deep-learning/ (visited on 05/18/2022).

[11] Emmanuel Soubies, Laure Blanc-Féraud, and Gilles Aubert. "A Continuous Exact l0 penalty (CEL0) for least squares regularized problem". In: *SIAM Journal on Imaging Sciences* 8.3 (July 2015), pp. 1607–1639. URL: https://hal.inria.fr/hal-01102492.

[12] Gilbert Strang. *Linear Algebra and Learning from Data*. Wellesley-Cambridge Press, 2019.

[13] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society (Series B)* 58 (1996), pp. 267–288.

[14] Trung Vu. *Convergence of Heavy-Ball Method and Nesterov's Accelerated Gradient on Quadratic Optimization*. Sept. 2018. URL: https://trungvietvu.github.io/notes/2018/Momentum (visited on 05/18/2022).