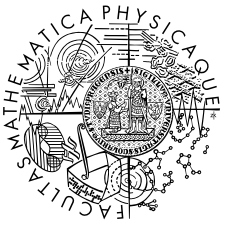

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Martina Tichá

Applications of least squares

Department of Algebra

Supervisor of the bachelor thesis:	doc. RNDr. Jiří Tůma, DrSc.
Study programme:	Mathematics
Study branch:	Mathematical Methods of Information Security

Prague 2017

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ..... Martina Tichá

Title: Applications of least squares

Author: Martina Tichá

Department: Department of Algebra

Supervisor: doc. RNDr. Jiří Tůma, DrSc., Department of Algebra

Abstract: In this work we analyze the method of least squares. We explain the basic mathematical theory that is crucial for understanding the method of least squares and show how the least squares problem can be solved via the system of normal equations using the QR-decomposition, Cholesky-decomposition or SVD of some matrix. We also show how the method of least squares can be used to solve the problem of data fitting and data classification. We have experimentally verified the theory covered in this thesis by implementing algorithm for recognizing handwritten digits. Apart from the handwritten digits recognition problem we show two more practical examples of the application of least squares. The first one relates to finding a solution to the least norm problem. In the second example we use the method of least squares to estimate the parameters of a linear measurement model.

Keywords: Least squares, QR-decomposition, SVD, computing with orthogonal matrices, classification and other problems

I must express my sincere gratitude to doc. RNDr. Jiří Tůma, DrSc. for his continuous support, patience, valuable advice and time spent on corrections.

Contents

Introduction

The method of least squares (LSM) was first published by Legendre in his work *Nouvelles methods pour la deretmination des orbiter des cometes* in Paris, 1805. Since then, many mathematicians, including Laplace, Gauss and plenty of others, studied this method and its possible applications [5]. Until now, the method of least squares finds its usage in many fields of mathematics, physics and especially statistics, where LMS is one of the most common techniques of data estimation.

In this thesis, we will analyze the method of least squares with focus on its application. For ease of understanding, the theoretical insights covered in this text will be illustrated on a practical example - handwritten digits recognition problem.

Handwritten digits recognition

The handwritten digits recognition problem is one of the variety of problems which can be solved using least squares. More specifically, it can be solved using the least squares classification which we will cover in details later in the thesis.

Problem definition Given a database of handwritten digits, containing training-set of 60 000 gray-scale images of size 28×28 represented as n-vectors with $n = 28 \times 28 = 784$, we want to find an algorithm which classifies an image to one of 10 clusters, each corresponding to one digit.

This database is available at http://yann.lecun.com/exdb/mnist/. The number of examples per digit varies between 5421 (for digit five) and 6742 (for digit one). Additionally, this database contains a test-set of 10 000 handwritten digits that can be used to assess how well the algorithm that we will find recognizes previously unseen digits.

The handwritten digits recognition problem together with the results of implemented algorithms is also presented in [1]. For the purpose of this thesis we implemented all the algorithms for the digits recognition described in this text in

Python using the Scikit-learn library which is an open source Python library for machine learning. Our results were very similar to those which can be found in [1].

In the first chapter we describe the least squares problem, show that there is a solution to this problem and explain when this solution is unique. In the second chapter we show three methods which are commonly used when finding a solution to the least squares problem. In Chapter 3 we will explain the least squares data fitting method. In Chapter 4 we describe the least squares classification. Chapter 5 covers the regularized least squares fitting and regularized least squares classification. Finally, in Chapter 6 we show two more practical examples of the application of least squares.

1 The method of least squares

In this chapter we will first describe the least squares problem. Next, we will show that there exists a solution to the least squares problem and explain when this solution is unique. We will then show that the least squares problem can be characterized by a system of so called *normal equations*. In the end of this chapter we will also derive a solution which does not operate with the system of normal equations.

1.1 Least squares problem

Suppose we have a system of linear equations

$$A\mathbf{x} = \mathbf{b}, \tag{1}$$

where A is an $m \times n$ matrix, $m > n$ and $\mathbf{b}$ is an m-vector. This system is called *over-determined* as there are more equations (m) then variables (n). Any system of equations has a solution if and only if the vector $\mathbf{b}$ is a linear combination of the columns of the matrix A.

In some cases, however, this assumption is not fulfilled, which means, we are not able to find a solution to the given system. Still, we can find a "best approximate solution", mathematically said we can find an n-vector $\hat{\mathbf{x}}$ that minimizes the *residual* r, where the residual r for the system of equations $A\mathbf{x} = \mathbf{b}$ is defined as $r = \|A\mathbf{x} - \mathbf{b}\|$, where $\|\cdot\|$ is an arbitrary norm.

We denote $A\hat{\mathbf{x}} \approx \mathbf{b}$ by which we express the fact that $\hat{\mathbf{x}}$ is not necessarily a solution to the system of equations $A\mathbf{x} = \mathbf{b}$, however the difference $A\hat{\mathbf{x}} - \mathbf{b}$ is "small" in some sense where the smallness is measured by a norm $\|\cdot\|$. The most frequently used norm in this context is the Euclidean norm $\|\cdot\|_2$.

Definition 1.1 Let $A \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $n, m \in \mathbb{N}$. Then:

1. The *least squares problem* is a problem of finding vector $\hat{\mathbf{x}} \in \mathbb{R}^n$ so that the norm $\|A\hat{\mathbf{x}} - \mathbf{b}\|_2$ is minimal.

2. Any vector $\hat{\mathbf{x}}$ which minimizes this norm is called a *least squares solution* to the system of equations $A\mathbf{x} = \mathbf{b}$.

Notes:

1. Consider the vector $\hat{\mathbf{x}}$ from the previous definition. Then, for every $\mathbf{x} \in \mathbb{R}^n$: $\| A\hat{\mathbf{x}} - \mathbf{b} \|_2 \leq \| A\mathbf{x} - \mathbf{b}\|_2$.

2. If the system of equations $A\mathbf{x} = \mathbf{b}$ has a solution then any vector $\mathbf{x}$ which is a solution to this system is also a least squares solution to this system.

1.2 Existence and uniqueness of the least squares solution

Recall that the system of linear equations $A\mathbf{x} = \mathbf{b}$ has a solution if and only if $\mathbf{b} \in R(A) = \langle \mathbf{a}_1, \ldots, \mathbf{a}_n \rangle$, where $\mathbf{a}_1, \ldots, \mathbf{a}_n$ are columns of the matrix A. In case $\mathbf{b} \notin R(A)$, there is no solution to this system. Instead of the exact solution, we will search for $\hat{\mathbf{x}}$ which minimizes the norm of the residue $\|A\mathbf{x} - \mathbf{b}\|_2$. We will see that for such a vector $\hat{\mathbf{x}}$, $A\hat{\mathbf{x}}$ is the orthogonal projection of the vector $\mathbf{b}$ onto a vector space generated by the columns of the matrix A.

First of all we will formulate some useful definitions and theorems. All the definitions and theorems were adopted from [4, Chapter 7].

Definition 1.2 Let $V = \mathbb{R}^n, n \in \mathbb{N}$ be a vector space. The *dot product* of two vectors $\mathbf{a}, \mathbf{b} \in V$ is defined as

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + \cdots + a_n b_n \tag{2}$$

Definition 1.3 Let $V = \mathbb{R}^n, n \in \mathbb{N}$ be a vector space, $\mathbf{a}, \mathbf{b} \in V$. We say that the vectors $\mathbf{a}$ and $\mathbf{b}$ are *perpendicular* and denote $\mathbf{a} \perp \mathbf{b}$ if and only if $\mathbf{a} \cdot \mathbf{b} = 0$.

Definition 1.4 Let $V = \mathbb{R}^n$, $n \in \mathbb{N}$ be a vector space. Then the set of vectors $M^\perp$ defined as

$$M^\perp = \{\mathbf{v} \in V : \mathbf{v} \perp M\} \tag{3}$$

is called the *orthogonal complement* of the set M.

Observation 1.5 The set $M^{\perp}$ from previous definition is a subspace of the vector space V.

Theorem 1.6 Let $V = \mathbb{R}^m, m \in \mathbb{N}$, be a vector space and W its subspace of a dimension $n \leq m$. Then

$$V = W \oplus W^{\perp}. \tag{4}$$

Corollary 1.7 Under the assumptions from previous theorem, every vector $\mathbf{v} \in V$ can be uniquely expressed as the sum of a vector $\mathbf{v}_W \in W$ and a vector $\mathbf{v}_{W^{\perp}}$ that is perpendicular to W.

Definition 1.8 Let $V = \mathbb{R}^n, n \in \mathbb{N}$ be a vector space, W its subspace and let $\mathbf{v} \in V$ be a vector. The vector $\mathbf{v}_W$ is called *orthogonal projection* of the vector $\mathbf{v}$ onto the vector space W.

Theorem 1.9 (Pythagoras theorem) Let $V = \mathbb{R}^n, n \in \mathbb{N}$ be a vector space. If the vectors $\mathbf{u}, \mathbf{v} \in V$ are perpendicular, then

$$\|\mathbf{u} + \mathbf{v}\|_2^2 = \|\mathbf{u}\|_2^2 + \|\mathbf{v}\|_2^2 . \tag{5}$$

Theorem 1.10 (Least squares solution) Vector $\hat{\mathbf{x}}$ is a least squares solution of $A\mathbf{x} = \mathbf{b}$ if and only if $A\hat{\mathbf{x}} = \mathbf{b}_{R(A)}$.
This condition is fulfilled if and only if $\hat{\mathbf{x}}$ is a solution to the system of equations

$$A^T A\mathbf{x} = A^T \mathbf{b}. \tag{6}$$

Definition 1.11 (System of normal equations) The system of equations (6) from the previous theorem is called the *system of normal equations* of $A\mathbf{x} = \mathbf{b}$.

If the columns of the matrix A are linearly independent, then the matrix $A^T A$ is regular and the system has a unique solution, which can be expressed as

$$\hat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}. \tag{7}$$

To prove Theorem 1.10, we first need the following lemma.

Lemma 1.12 Let $V = \mathbb{R}^n$, $n \in \mathbb{N}$ be a vector space, W its subspace, $\mathbf{v} \in V$. Then, the vector $\mathbf{v}_W - \mathbf{v}$ is the unique vector with the smallest norm among all the vectors $\mathbf{w} - \mathbf{v}, \mathbf{w} \in W$.

Proof. From Corollary 1.7 we know that every vector $\mathbf{v} \in V$ can be uniquely expressed as a sum $\mathbf{v} = \mathbf{v}_W + \mathbf{v}_{W^\perp}$, where $\mathbf{v}_W \in W$ and $\mathbf{v}_{W^\perp} \in W^\perp$, from which it follows that $\mathbf{v}_W - \mathbf{v} = -\mathbf{v}_{W^\perp}$.

Let $\mathbf{w} \in W$ be an arbitrary vector in the vector space W. We will write the vector $\mathbf{w} - \mathbf{v}$ as follows:

$$\mathbf{w} - \mathbf{v} = (\mathbf{w} - \mathbf{v}_W) + (\mathbf{v}_W - \mathbf{v}) = (\mathbf{w} - \mathbf{v}_W) + (-\mathbf{v}_{W^\perp}).$$

The vectors $(-\mathbf{v}_{W^\perp})$ and $(\mathbf{w} - \mathbf{v}_W)$ are perpendicular as $(\mathbf{w} - \mathbf{v}_W) \in W$ and $(-\mathbf{v}_{W^\perp}) \in W^\perp$. We can therefore apply Theorem 1.9:

$$\|\mathbf{w} - \mathbf{v}\|_2^2 = \|\mathbf{w} - \mathbf{v}_W\|_2^2 + \|-\mathbf{v}_{\mathbf{W}^\perp}\|_2^2,$$

from which it is obvious that for a fixed $\mathbf{v} \in V$ and variable $\mathbf{w} \in W$ the norm $\|\mathbf{w} - \mathbf{v}\|_2$ is minimal if and only if $\mathbf{w} = \mathbf{v}_W$. □

Proof of Theorem 1.10: Recall that the least squares solution to the system of equations $A\mathbf{x} = \mathbf{b}$ is defined (definition 1.1) as a vector $\hat{\mathbf{x}}$ which minimizes the norm $\|A\mathbf{x} - \mathbf{b}\|_2$, $\mathbf{x} \in \mathbb{R}^n$. We can reformulate the problem of minimizing $\|A\mathbf{x} - \mathbf{b}\|_2$ for $\mathbf{x} \in \mathbb{R}^n$ as a problem of minimizing $\|\mathbf{w} - \mathbf{b}\|_2$ for $\mathbf{w} \in R(A)$. We can then use Lemma 1.12 with $W = R(A)$ and $\mathbf{v} = \mathbf{b}$ to get

$$\hat{\mathbf{x}}\, minimizes \qquad \|A\mathbf{x} - \mathbf{b}\|_2 \qquad if\ and\ only\ if \qquad A\hat{\mathbf{x}} = \mathbf{b}_{R(A)}.$$

We will now prove the second part of the theorem. Vector $A\hat{\mathbf{x}}$ is the orthogonal projection of the vector $\mathbf{b}$ onto the subspace $R(A)$. Corollary 1.7 applied on

$W = R(A)$ and $\mathbf{v}_W = A\hat{\mathbf{x}}$ says that we can express the vector $\mathbf{b}$ as $\mathbf{b} = A\hat{\mathbf{x}} + A\hat{\mathbf{x}}^{\perp}$. Hence

$$\mathbf{b} - A\hat{\mathbf{x}} = A\hat{\mathbf{x}}^{\perp} \in R(A)^{\perp} = Ker(A^T)$$

which implies

$$A\hat{\mathbf{x}} - \mathbf{b} \in R(A)^{\perp} = Ker(A^T)$$

and so

$$A^T(A\hat{\mathbf{x}} - \mathbf{b}) = 0,$$

which is equivalent to

$$A^T A\hat{\mathbf{x}} = A^T\mathbf{b}.$$

□

If there is no solution to the system of equations $A\mathbf{x} = \mathbf{b}$, when calculating a least squares solution to this system, it is convenient to distinguish two cases, depending on whether or not the columns of the matrix A are linearly dependent.

Providing that the columns of the matrix A from the system of equations $A\mathbf{x} = \mathbf{b}$ are linearly independent, the projection $\mathbf{b}_{R(A)}$ of the vector $\mathbf{b}$ can be uniquely expressed as a linear combination of the columns of the matrix A, i.e. there exists exactly one solution to the least squares problem.

If the columns of the matrix A are linearly dependent, there are infinitely many possibilities how to express the projection $\mathbf{b}_{R(A)}$ as a linear combination of the columns of A as for a solution $\hat{\mathbf{x}}$ to the system of normal equations $A^T A\mathbf{x} = A^T\mathbf{b}$, the vector $\hat{\mathbf{x}} + \mathbf{z}$, where $\mathbf{z} \in Ker(A)$ is also a solution to this system as $A^T A(\hat{\mathbf{x}} + \mathbf{z}) = A^T A\hat{\mathbf{x}} + A^T A\mathbf{z} = A^T A\hat{\mathbf{x}} + A^T(0, \ldots, 0)^T = A^T A\hat{\mathbf{x}}$. In this case the solution to the least squares problem is not unique. In order to make the solution unique we have to add some more constraints. One of the possibilities is to find such a solution $\hat{\mathbf{x}}$ to the normal equations that the norm $\|\hat{\mathbf{x}}\|_2$ is minimal. We will return to this problem in Chapter 6.

1.3 Solution to the least squares problem using calculus

We will now show another approach how to derive that a least squares problem can be solved via the system of normal equations. This approach was presented in [1, Chapter 12.2].

Instead of searching for $\hat{\mathbf{x}}$ which minimizes the norm $\|A\mathbf{x} - \mathbf{b}\|_2$, we will be searching for $\hat{\mathbf{x}}$ that minimizes the squared norm $\|A\mathbf{x} - \mathbf{b}\|_2^2 = \|\mathbf{r}\|_2^2 = r_1^2 + \cdots + r_m^2$, as for this expression, there exists the derivative with respect to $\mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^n$ and the minimizer is obviously the same for both expressions. From calculus we know that any minimizer $\hat{\mathbf{x}}$ of the function $f(\mathbf{x}) = \|A\mathbf{x} - \mathbf{b}\|_2^2$ must satisfy

$$\frac{\partial f}{\partial x_i}(\hat{\mathbf{x}}) = 0, i = 1, \ldots, n,$$

which can simply be expressed as the vector equation

$$\nabla f(\hat{\mathbf{x}}) = 0,$$

where $\nabla f(\hat{\mathbf{x}})$ is the gradient of f evaluated at $\hat{\mathbf{x}}$. Our goal is to show that this gradient can be expressed in matrix form as

$$\nabla f(\mathbf{x}) = 2A^T(A\mathbf{x} - \mathbf{b}).$$

This formula can be derived from chain rule (see [1, Chapter 10.2]) and following observation:

First of all we will write out the least squares objective as a sum. We get

$$\|A\mathbf{x} - \mathbf{b}\|_2^2 = \sum_{i=1}^{m}(\sum_{j=1}^{n} A_{ij}x_j - b_i)^2.$$

In order to get $\nabla f(\mathbf{x})_k$ we take the partial derivative of f with respect to x_k. Differentiating the sum term by term, we get

$$\begin{aligned}\nabla f(\mathbf{x})_k &= \frac{\partial f}{\partial x_k}(\mathbf{x}) \\ &= \sum_{i=1}^{m} 2(\sum_{j=1}^{n} A_{ij}x_j - b_i)(A_{ik}) \\ &= \sum_{i=1}^{m} 2(A^T)_{ki}(A\mathbf{x} - \mathbf{b})_i \\ &= (2A^T(A\mathbf{x} - \mathbf{b}))_k,\end{aligned}$$

from where it follows

$$\nabla f(\hat{\mathbf{x}}) = 0 \quad \Leftrightarrow \quad \forall k = 1, \ldots, n : (2A^T(A\mathbf{x}-\mathbf{b}))_k = 0 \quad \Leftrightarrow \quad 2A^T(A\mathbf{x}-\mathbf{b})) = 0.$$

So far we know that any minimizer $\hat{\mathbf{x}}$ of $\|A\mathbf{x} - \mathbf{b}\|_2^2$ must satisfy

$$\nabla f(\hat{\mathbf{x}}) = 2A^T(A\hat{\mathbf{x}} - \mathbf{b}) = 0,$$

which we can rewrite as

$$A^T A\hat{\mathbf{x}} = A^T\mathbf{b},$$

which gives us the same solution to the least squares problem like the one we derived in previous section.

1.4 Finding the orthogonal projection of a vector onto a subspace

In the following, we will explain how to find the unique orthogonal projection of the vector $\mathbf{b}$ onto the vector space $R(A)$. This will give us another possibility how to find a solution to the least squares problem. For this, we will use the following theorem.

Theorem 1.13 [4, Theorem 7.34] Let $V = \mathbb{R}^n, n \in \mathbb{N}$ be a vector space, $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k, \mathbf{v} \in V$, $W = \langle \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k \rangle$. The orthogonal projection of

the vector $\mathbf{v}$ onto the vector space W is given by the vector

$$\mathbf{v}_W = a_1\mathbf{w}_1 + a_2\mathbf{w}_2 + \cdots + a_k\mathbf{w}_k, \tag{8}$$

where $\mathbf{a} = (a_1, a_2, \ldots, a_k)^T$ is an arbitrary solution to the system of equations give by the augmented matrix

$$\left(\begin{array}{cccc|c} \mathbf{w}_1 \cdot \mathbf{w}_1 & \mathbf{w}_1 \cdot \mathbf{w}_2 & \cdots & \mathbf{w}_1 \cdot \mathbf{w}_k & \mathbf{w}_1 \cdot \mathbf{v} \\ \mathbf{w}_2 \cdot \mathbf{w}_1 & \mathbf{w}_2 \cdot \mathbf{w}_2 & \cdots & \mathbf{w}_2 \cdot \mathbf{w}_k & \mathbf{w}_2 \cdot \mathbf{v} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{w}_k \cdot \mathbf{w}_1 & \mathbf{w}_k \cdot \mathbf{w}_2 & \cdots & \mathbf{w}_k \cdot \mathbf{w}_k & \mathbf{w}_k \cdot \mathbf{v} \end{array}\right) \tag{9}$$

Proof. The vector $\mathbf{v}_W$ lies in the vector space W, hence it can be written as a linear combination of its generators

$$\mathbf{v}_W = a_1\mathbf{w}_1 + a_2\mathbf{w_2} + \cdots + a_k\mathbf{w}_k.$$

The vector $\mathbf{v}_W$ is the orthogonal projection of $\mathbf{v}$ onto W, i.e. $\mathbf{v} - \mathbf{v}_W$ is perpendicular to the vector space W which holds if and only if it is perpendicular to all its generators $\mathbf{w}_1, \mathbf{w_2}, \ldots \mathbf{w}_k$. It follows

$$\begin{aligned} 0 &= \mathbf{w}_i \cdot (\mathbf{v} - \mathbf{v}_W) \\ &= \mathbf{w}_i \cdot (\mathbf{v} - a_1\mathbf{w}_1 - a_2\mathbf{w}_2 - \ldots - a_k\mathbf{w}_k) \\ &= \mathbf{w}_i \cdot \mathbf{v} - a_1(\mathbf{w}_i \cdot \mathbf{w}_1) - a_2(\mathbf{w}_i \cdot \mathbf{w}_2) - \ldots a_k(\mathbf{w}_i \cdot \mathbf{w}_k), \end{aligned}$$

from where we get for every $i = 1, 2, \ldots, k$

$$a_1(\mathbf{w}_i \cdot \mathbf{w}_1) + a_2(\mathbf{w}_i \cdot \mathbf{w_2}) + \ldots + a_k(\mathbf{w}_i \cdot \mathbf{w}_k) = \ \mathbf{w}_i \cdot \mathbf{v}.$$

Therefore the vector of the coefficients $(a_1, a_2, \ldots, a_k)^T$ is the solution to the system (9). □

The matrix

$$\begin{pmatrix} \mathbf{w}_1 \cdot \mathbf{w}_1 & \mathbf{w}_1 \cdot \mathbf{w}_2 & \cdots & \mathbf{w}_1 \cdot \mathbf{w}_k \\ \mathbf{w}_2 \cdot \mathbf{w}_1 & \mathbf{w}_2 \cdot \mathbf{w}_2 & \cdots & \mathbf{w}_2 \cdot \mathbf{w}_k \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_k \cdot \mathbf{w}_1 & \mathbf{w}_k \cdot \mathbf{w}_2 & \cdots & \mathbf{w}_k \cdot \mathbf{w}_k \end{pmatrix} \tag{10}$$

which is the left side of the matrix 9 from the previous theorem is called the *Gram matrix* of the sequence of vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k$. For linearly independent vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k$, $(a_1, a_2, \ldots, a_k)^T$ are the coefficients of the vector $\mathbf{v}_W \in W$ with respect to the basis $\mathbf{w}_1, \ldots, \mathbf{w}_k$ of W. These are unique for all the vectors $\mathbf{w} \in W$ and so the Gram matrix is regular. Conversely, for linearly dependent vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k$, the Gram matrix is singular.

We can now apply Theorem 1.13 with the vector $\mathbf{v} = \mathbf{b}$ and the subspace $W = R(A)$, and so we get the orthogonal projection $\mathbf{b}_{R(A)}$ of $\mathbf{b}$ onto $R(A)$. Applying Theorem 1.10, we can compute the least squares solution of $A\mathbf{x} = \mathbf{b}$ as a solution to the system of equations $A\mathbf{x} = \mathbf{b}_{R(A)}$.

2 Solving least squares problems

In the previous chapter, we showed that the least squares problem can be solved via the system of normal equations. We could now simply use the Gaussian elimination to find the solution to the system of normal equations. This approach, however, is not quite good because of its numerical instability.

In this chapter we will show three methods that are commonly used when finding a solution to the least squares problem. First two methods (solution using QR-decomposition and solution via system of normal equations) are only applicable in case the matrix A from the system of normal equations has a full column rank. In the end of this chapter, we will see a method which can also be used in case the columns of the matrix A are linearly dependent. This method is based on so called singular value decomposition (SVD).

2.1 Solving LS problem via QR-decomposition

First of all, we will look at the QR-decomposition itself. We will start with the explanation of the Gram-Schmidt orthogonalization which will help us to understand what the QR-decomposition is, when does it exist and how to find it. Later on, we will show how it can be used in finding the solution to the least squares problem.

Gram-Schmidt orthogonalization

Let $V \leq \mathbb{R}^m$, $m \in \mathbb{N}$ be a subspace of the dimension n and $(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$ its basis. The *Gram-Schmidt orthogonalization* is an algorithm which starts with the basis vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ and constructs new basis vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n$ such that $\mathbf{w}_i \cdot \mathbf{w}_j = 0$ for $i \neq j$ and $\langle \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k \rangle = \langle \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k \rangle$ for every $k = 1, \ldots, n$.

The Gram-Schmidt orthogonalization works as follows. Let $\mathbf{w}_1 = \mathbf{v}_1$. We want to find a vector $\mathbf{w}_2$ that is perpendicular to $\mathbf{w}_1$ and $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle = \langle \mathbf{w}_1, \mathbf{w}_2 \rangle$, which is

equivalent to finding a number $a \in \mathbb{R}$ so that

$$\mathbf{w}_2 = a\mathbf{w}_1 + \mathbf{v}_2, \quad \mathbf{w}_1 \cdot \mathbf{w}_2 = 0.$$

From this two equations we get

$$0 = \mathbf{w}_1 \cdot \mathbf{w}_2 = \mathbf{w}_1 \cdot (a\mathbf{w}_1 + \mathbf{v}_2) = a \left\|\mathbf{w}_1\right\|^2 + \mathbf{w}_1 \cdot \mathbf{v}_2$$

which gives us the expression of a as

$$a = -\frac{\mathbf{w}_1 \cdot \mathbf{v}_2}{\left\|\mathbf{w}_1\right\|^2}$$

and so

$$\mathbf{w}_2 = \mathbf{v}_2 - \frac{\mathbf{w}_1 \cdot \mathbf{v}_2}{\left\|\mathbf{w}_1\right\|^2}\mathbf{w}_1.$$

To prove that $\langle \mathbf{w}_1, \mathbf{w}_2 \rangle = \langle \mathbf{v}_1, \mathbf{v}_2 \rangle$, we will show that $\langle \mathbf{w}_1, \mathbf{w}_2 \rangle \subseteq \langle \mathbf{v}_1, \mathbf{v}_2 \rangle$ and $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle \subseteq \langle \mathbf{w}_1, \mathbf{w}_2 \rangle$.
$\mathbf{w}_1 = \mathbf{v}_1 => \mathbf{w}_1 \in \langle \mathbf{v}_1, \mathbf{v}_2 \rangle$,
$\mathbf{w}_2 = \mathbf{v}_2 - \frac{\mathbf{w}_1 \cdot \mathbf{v}_2}{\|\mathbf{w}_1\|^2}\mathbf{w}_1 = \mathbf{v}_2 - \frac{\mathbf{w}_1 \cdot \mathbf{v}_2}{\|\mathbf{w}_1\|^2}\mathbf{v}_1 => \mathbf{w}_2 \in \langle \mathbf{v}_1, \mathbf{v}_2 \rangle$,
and so $\langle \mathbf{w}_1, \mathbf{w}_2 \rangle \subseteq \langle \mathbf{v}_1, \mathbf{v}_2 \rangle$.

$\mathbf{v}_1 = \mathbf{w}_1 => \mathbf{v}_1 \in \langle \mathbf{w}_1, \mathbf{w}_2 \rangle$,
$\mathbf{v}_2 = \mathbf{w}_2 + \frac{\mathbf{w}_1 \cdot \mathbf{v}_2}{\|\mathbf{w}_1\|^2}\mathbf{w}_1 => \mathbf{v}_2 \in \langle \mathbf{w}_1, \mathbf{w}_2 \rangle$,
and so $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle \subseteq \langle \mathbf{w}_1, \mathbf{w}_2 \rangle$.

To compute the vector $\mathbf{w}_3$, we need to find numbers $a_1,\ a_2 \in \mathbb{R}$ so that

$$\mathbf{w}_3 = a_1\mathbf{w}_1 + a_2\mathbf{w}_2 + \mathbf{v}_3, \quad \mathbf{w}_1 \cdot \mathbf{w}_3 = \mathbf{w}_2 \cdot \mathbf{w}_3 = 0.$$

We get two equations:

$$\begin{aligned} 0 = \mathbf{w}_1 \cdot (a_1\mathbf{w}_1 + a_2\mathbf{w}_2 + \mathbf{v}_3) &= a_1 \left\|\mathbf{w}_1\right\|^2 + a_2(\mathbf{w}_1 \cdot \mathbf{w}_2) + \mathbf{w}_1 \cdot \mathbf{v}_3 \\ &= a_1 \left\|\mathbf{w}_1\right\|^2 + \mathbf{w}_1 \cdot \mathbf{v}_3 \end{aligned}$$

$$0 = \mathbf{w}_2 \cdot (a_1\mathbf{w}_1 + a_2\mathbf{w}_2 + \mathbf{v}_3) = a_1(\mathbf{w}_1 \cdot \mathbf{w}_2) + a_2 \left\|\mathbf{w}_2\right\|^2 + \mathbf{w}_2 \cdot \mathbf{v}_3$$
$$= a_2 \left\|\mathbf{w}_2\right\|^2 + \mathbf{w}_2 \cdot \mathbf{v}_3,$$

that give us the expressions of a_1 and a_2 as

$$a_1 = -\frac{\mathbf{w}_1 \cdot \mathbf{v}_3}{\left\|\mathbf{w}_1\right\|^2}$$

$$a_2 = -\frac{\mathbf{w}_2 \cdot \mathbf{v}_3}{\left\|\mathbf{w}_2\right\|^2}.$$

Hence, we can compute the vector $\mathbf{w}_3$ as

$$\mathbf{w}_3 = \mathbf{v}_3 - \frac{\mathbf{w}_1 \cdot \mathbf{v}_3}{\left\|\mathbf{w}_1\right\|^2}\mathbf{w}_1 - \frac{\mathbf{w}_2 \cdot \mathbf{v}_3}{\left\|\mathbf{w}_2\right\|^2}\mathbf{w}_2. \tag{11}$$

To prove that $\langle \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3 \rangle = \langle \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \rangle$, we will again show that $\langle \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3 \rangle \subseteq \langle \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \rangle$ and $\langle \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \rangle \subseteq \langle \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3 \rangle$.

We already know that $\mathbf{w}_1, \mathbf{w}_2 \in \langle \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \rangle$, which implies that the vectors $\mathbf{w}_1, \mathbf{w}_2$ can be expressed as a linear combination of the vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ which together with the equation (11) implies that also the vector $\mathbf{w}_3$ can be expressed as a linear combination of the vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$. Hence, $\mathbf{w}_3 \in \langle \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \rangle$ and so $\langle \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3 \rangle \subseteq \langle \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \rangle$.

We also showed in previous that the vectors $\mathbf{v}_1, \mathbf{v}_2$ lie in the vector space $\langle \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3 \rangle$. From the equation (11) we can see that the vector $\mathbf{v}_3$ can be expressed as a linear combination of the vectors $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$, which implies that $\mathbf{v}_3 \in \langle \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3 \rangle$. It follows that $\langle \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \rangle \subseteq \langle \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3 \rangle$.

In this manner we can continue to get all the basis vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n$.

Theorem 2.1 (about Gram-Schmidt orthogonalization)
Let $V = \mathbb{R}^n$, $n \in \mathbb{N}$ be a vector space with the basis $(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$. Let

$$\begin{aligned}
\mathbf{w}_1 &= \mathbf{v}_1, \\
\mathbf{w}_2 &= \mathbf{v}_2 - \frac{\mathbf{w}_1 \cdot \mathbf{v}_2}{\|\mathbf{w}_1\|^2}\mathbf{w}_1 \\
&\vdots \\
\mathbf{w}_n &= \mathbf{v}_n - \frac{\mathbf{w}_1 \cdot \mathbf{v}_n}{\|\mathbf{w}_1\|^2}\mathbf{w}_1 - \cdots - \frac{\mathbf{w}_{n-1} \cdot \mathbf{v}_n}{\|\mathbf{w}_{n-1}\|^2}\mathbf{w}_{n-1}.
\end{aligned} \tag{12}$$

Then $(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$ is an orthogonal basis for V such that $\langle \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k \rangle = \langle \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k \rangle$, $k \leq n$.

QR-decomposition

Suppose we have a matrix $A = (\mathbf{v}_1 | \mathbf{v}_2 | \ldots | \mathbf{v}_n)$ of type $m \times n$ whose columns are linearly independent, which means they are basis vectors of the vector space $R(A)$. From Theorem 2.1 we know that there exists an orthogonal basis $(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$ of the vector space $R(A)$ such that $\langle \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k \rangle = \langle \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k \rangle$, $k \leq n$. The Gram-Schmidt orthogonalization can also be expressed in the form of so called *QR-decomposition* of the matrix $A = (\mathbf{v}_1 | \mathbf{v}_2 | \ldots | \mathbf{v}_n)$.

Theorem 2.2 (about QR-decomposition) [4, Theorem 7.44] Let A be a real $m \times n$ matrix, whose columns are linearly independent. Then, there exists a decomposition

$$A = QR, \tag{13}$$

where Q is an $m \times n$ matrix with orthonormal columns and R is an upper triangular matrix of the order n with positive elements on its main diagonal.

Proof. Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ be the column vectors of the matrix A. We will apply on these vectors the Gram-Schmidt orthogonalization where we will normalize every resulting vector $\mathbf{w}'_i, i = 1, \ldots, n$. For every $i = 1, \ldots, n$ we get

$$\mathbf{w}'_i = \mathbf{v}_i - (\mathbf{w}_1 \cdot \mathbf{v}_i)\mathbf{w}_1 - (\mathbf{w}_2 \cdot \mathbf{v}_i)\mathbf{w}_2 - \cdots - (\mathbf{w}_{i-1} \cdot \mathbf{v}_i)\mathbf{w}_{i-1}, \quad \mathbf{w}_i = \frac{\mathbf{w}'_i}{\|\mathbf{w}'_i\|},$$

from which we can express the vector $\mathbf{v}_i$ as

$$\mathbf{v}_i = (\mathbf{w}_1 \cdot \mathbf{v}_i)\mathbf{w}_1 + (\mathbf{w}_2 \cdot \mathbf{v}_i)\mathbf{w}_2 + \cdots + (\mathbf{w}_{i-1} \cdot \mathbf{v}_i)\mathbf{w}_{i-1} + \|\mathbf{w}'_i\|\,\mathbf{w}_i.$$

This can be written in a matrix form as

$$(\mathbf{v}_1|\mathbf{v}_2|\ldots|\mathbf{v}_n) = (\mathbf{w}_1|\mathbf{w}_2|\ldots|\mathbf{w}_n)\begin{pmatrix} \|\mathbf{w}'_1\| & \mathbf{w}_1 \cdot \mathbf{v}_2 & \ldots & \mathbf{w}_1 \cdot \mathbf{v}_n \\ 0 & \|\mathbf{w}'_2\| & \ldots & \mathbf{w}_2 \cdot \mathbf{v}_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & \|\mathbf{w}'_n\| \end{pmatrix}$$

Thus we can set $Q = (\mathbf{w}_1|\ldots|\mathbf{w}_n)$ and $R = \begin{pmatrix} \|\mathbf{w}'_1\| & \mathbf{w}_1 \cdot \mathbf{v}_2 & \ldots & \mathbf{w}_1 \cdot \mathbf{v}_n \\ 0 & \|\mathbf{w}'_2\| & \ldots & \mathbf{w}_2 \cdot \mathbf{v}_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & \|\mathbf{w}'_n\| \end{pmatrix}$.

□

The Gram-Schmidt orthogonalization is an easy in its implementation way to find the QR-decomposition. Still, there are other, more numerically stable methods to find the QR-decomposition of a matrix. More about this subject can be found in [3, Chapter 3.5].

Solution to least squares problem using QR-decomposition

Theorem 1.10 says that a solution to the least squares problem is equal to a solution to the system of normal equations $A^TA\mathbf{x} = A^T\mathbf{b}$. We will now use the QR-decomposition of the matrix A to solve this system.

The columns of the matrix Q are orthonormal, which implies $Q^TQ = I_n$. Apply-

ing the QR-decomposition (13) of the matrix A, we get:

$$\begin{aligned} A^T A\mathbf{x} &= A^T\mathbf{b} \\ (QR)^T QR\mathbf{x} &= (QR)^T\mathbf{b} \\ R^T Q^T QR\mathbf{x} &= R^T Q^T\mathbf{b} \\ R^T R\mathbf{x} &= R^T Q^T\mathbf{b} \\ R\mathbf{x} &= Q^T\mathbf{b} \end{aligned}$$

The last system of equations is a system with an upper triangular matrix. Hence, we can get $\mathbf{x}$ using back-substitution.

2.2 Solving LS problem via system of normal equations (Cholesky decomposition)

The following method which assumes the columns of the matrix A being linearly independent is suitable if the number m of the rows of the matrix A is significantly larger then the number of its columns and the number n of its columns is small. In this case, the $n \times n$ matrix $A^T A$ from the system of normal equations is small and we can use general methods to solve this system.

Since the columns of the matrix A are linearly independent, the matrix $A^T A$ is regular, symmetric and positive definite. For such a matrix, there exists the *Cholesky decomposition.*

Theorem 2.3 (about Cholesky decomposition) [3, Theorem 4.8] Let M be a symmetric, positive definite matrix. Then there exists a unique decomposition

$$M = LL^T, \tag{14}$$

where L is a lower triangular matrix with positive elements on the diagonal.

There are various methods for calculating the Cholesky decomposition. One of them is to be found in [3, Chapter 4.4 - A8].

Once we have the Cholesky decomposition (14) of our matrix $A^T A$, we can easily find the solution $\hat{\mathbf{x}}$ to the system of normal equations:

$$A^T A\hat{\mathbf{x}} = A^T\mathbf{b}$$
$$L^T L\hat{\mathbf{x}} = A^T\mathbf{b}$$
$$L^T\mathbf{w} = A^T\mathbf{b},$$

where $\mathbf{w} = L\hat{\mathbf{x}}$. The matrix L is a 5 lower triangular matrix, hence L^T is an upper triangular matrix and we can compute $\mathbf{w}$ using backward substitution. Once we have computed the vector $\mathbf{w}$, the result $\hat{\mathbf{x}}$ can be computed using the forward substitution.

2.3 Solving LS problem via Singular Value Decomposition (SVD)

The last method of solving the least squares problem that we will talk about is a method which uses SVD-decomposition of the matrix A from the definition of the LS problem (1.1). This approach is the most universal way to solve the least squares problem and can also be used in case the columns of the matrix A are linearly dependent. First of all, we need to formulate the following theorem.

Theorem 2.4 (about Singular Value Decomposition) [3, Theorem 5.6] Let A be a real $m \times n$ matrix, $r = rank(A)$. Then, there exist orthogonal matrices U of order m and V of order n and a $m \times n$ matrix Σ with zero values everywhere except for the main diagonal, where the (i,i) entry is σ_i for $i = 1, \ldots, r$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ so that

$$A = U\Sigma V^T. \tag{15}$$

The decomposition of the matrix A from the previous theorem is called the *Singular Value Decomposition* of A. The numbers $\sigma_1, \ldots, \sigma_r$ are called *singular values* of the matrix A.

Since the last $n-r$ columns of the matrix Σ are 0-vectors, the last $n-r$ rows of the matrix V^T as well as the last $m-r$ columns of the matrix U are irrelevant for the SVD of A as they are always multiplied by a 0-vector. Hence, we can express the SVD of A as

$$A = U_r \Sigma_r V_r^T, \tag{16}$$

where $U_r = (\mathbf{u}_1 | \dots | \mathbf{u}_r)$, $\Sigma_r = diag_{r \times r}(\sigma_1, \dots, \sigma_r)$ and $V_r^T = (\mathbf{v}_1 | \dots | \mathbf{v}_r)^T$. This decomposition of the matrix A is called the *reduced Singular Value Decomposition.*

We will now use the reduced SVD decomposition of the matrix A to solve the system of normal equations $A^T A\mathbf{x} = A^T\mathbf{b}$. We already know that if the columns of the matrix A are linearly independent, the vector $\mathbf{x}$ from this system can be expressed as $\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b}$. The matrix $A^T A$ from the right side of this equations can be rewritten as:

$$\begin{aligned} A^T A &= (U_r \Sigma_r V_r^T)^T U_r \Sigma_r V_r^T \\ &= V_r \Sigma_r^T U_r^T U_r \Sigma_r V_r^T \\ &= V_r \Sigma_r^T \Sigma_r V_r^T \\ &= V_r \Sigma_r^2 V_r^T. \end{aligned}$$

In the second step we use the fact that $U_r^T U_r = I_{r \times r}$ as the columns of the matrix U are orthonormal. The matrix Σ_r is a diagonal matrix which implies $\Sigma_r^T = \Sigma_r$ and so $\Sigma_r^T \Sigma_r = \Sigma_r^2$, which is used in the last step.

We can now express the vector $\mathbf{x}$ using the SVD of the matrix A as:

$$\begin{aligned} \mathbf{x} &= (A^T A)^{-1} A^T \mathbf{b} \\ &= (V_r \Sigma_r^2 V_r^T)^{-1} (U_r \Sigma_r V_r^T)^T \mathbf{b} \\ &= V_r \Sigma_r^{-2} V_r^T V_r \Sigma_r^T U_r^T \mathbf{b} \\ &= V_r \Sigma_r^{-2} \Sigma_r^T U_r^T \mathbf{b} \\ &= V_r \Sigma_r^{-1} U_r^T \mathbf{b}. \end{aligned}$$

In the third step we use that $V^TV = I_{r\times r}$ as the columns of the matrix V are orthonormal. In the last step, we again use the diagonality of the matrix Σ_r which gives us $\Sigma_r^{-2}\Sigma_r^T = \Sigma_r^{-1}$.

We will now show that this solution holds even if the columns of the matrix A are not linearly independent. Let us define $\hat{\mathbf{x}} = V_r\Sigma_r^{-1}U_r^T\mathbf{b}$. We will show that $\hat{\mathbf{x}}$ is the solution to the equation $A^TA\mathbf{x} = A^T\mathbf{b}$. Above we saw that the matrix A^TA can be rewritten using the reduced SVD of the matrix A as $A^TA = V_r\Sigma_r^2V_r^T$. Using this, we get

$$\begin{aligned} A^TA\hat{\mathbf{x}} &= V_r\Sigma_r^2V_r^TV_r\Sigma_r^{-1}U_r^T\mathbf{b} \\ &= V_r\Sigma_r^2\Sigma_r^{-1}U_r^T\mathbf{b} \\ &= V_r\Sigma_rU_r^T\mathbf{b} \\ &= V_r\Sigma_r^TU_r^T\mathbf{b} \\ &= (U_r\Sigma_rV_r^T)^T\mathbf{b} \\ &= A^T\mathbf{b} \end{aligned}$$

In the second step, we use that $V_r^TV_r = I_{r\times r}$ as the columns of the matrix V are orthonormal. In the third and forth step we again use the diagonality of the matrix Σ_r.

3 Least squares data fitting

In this chapter we will explain the *least squares data fitting method*. This method will not be directly used in our digit recognition problem. However, it is the corner stone for least squares classifier (covered in following chapter) which is a powerful method for the digit recognizing and cannot be understood without understanding the least squares data fitting method. The text of the chapter is based on [1, Chapter 13]

3.1 The data fitting problem

Suppose that we have an n-vector $\mathbf{x}$ and a scalar y, and we believe that they are related by some unknown function $f : \mathbb{R}^n \to \mathbb{R}$:

$$y = f(\mathbf{x}). \tag{17}$$

The vector $\mathbf{x}$ usually represents a set of n feature values, and is called the *feature vector*. The scalar y represents the outcome. For example, the vector $\mathbf{x}$ can represent age, weight, cholesterol level and systolic blood pressure of a patient and the outcome y the probability of heart attack in the patient in next five years.

Typically, we don't know f but we might have an idea about what approximately it should look like. But we do know some data $\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_N$, $y_1, y_2, \ldots, y_N$ where the n-vectors $\mathbf{x}_i, i = 1, \ldots, N$ are the feature vectors and the scalars y_i the corresponding outcomes.

3.2 Data fitting model

As we said, we suppose that the variable $\mathbf{x}$ is related with the scalar y so that $y = f(\mathbf{x})$, where $f : \mathbb{R}^n \to \mathbb{R}$ is a function which we do not know. Our goal is to find a function $\hat{f}$ which approximates the unknown function f, i.e. for every input vector $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \ldots, N$, $f(\mathbf{x}_i) \approx \hat{f}(\mathbf{x}_i)$.

The general model of data fitting has the following form:

$$\hat{f}(\mathbf{x}) = \theta_1 f_1(\mathbf{x}) + \cdots + \theta_p f_p(\mathbf{x}). \tag{18}$$

The functions $f_i : \mathbb{R}^n \to \mathbb{R}$ are basis functions we have to choose according to what we think the function f looks like and θ_i are the model parameters we will determine having chosen the basis functions and having available the set of training data. This form of model is called *linear in the parameters* as for every $\mathbf{x}$, $\hat{f}(\mathbf{x})$ is linear in the parameters θ_i. The function $\hat{f}(\mathbf{x})$ is called the *data fitting function.*

Let us define $\hat{y} = \hat{f}(\mathbf{x})$. The scalar $\hat{y}$ is then the prediction of the outcome y based on the vector of independent values $\mathbf{x}$.

3.3 Prediction error

Our goal is to find a function $\hat{f}$ which fits our given data best, which means for all the pairs of the given data $\mathbf{x}_i, y_i$, $i = 1, \ldots, N$, $\hat{f}(\mathbf{x}_i) \approx y_i$. In other words, the prediction errors which are the *residuals* for the data points defined as

$$r_i = \hat{y}_i - y_i, \;\; i = 1, \ldots, N \tag{19}$$

are small.

Let $\mathbf{y}$, $\hat{\mathbf{y}}$ and $\mathbf{r}$ denote the N-vectors with the entries y_i, $\hat{y}_i$ and r_i respectively. Hence, we can express the vector of residuals as $\mathbf{r} = \hat{\mathbf{y}} - \mathbf{y}$. We are searching for a function $\hat{f}$ so that the vector $\mathbf{r}$ is "minimal". One of the possible measure is given by the **rms**() value:

$$\mathbf{rms}(\mathbf{r}) = \sqrt{\frac{r_1^2 + \cdots + r_n^2}{N}} = \frac{\|\mathbf{r}\|_2}{\sqrt{N}}. \tag{20}$$

This RMS (root-mean-squared) value [1, Chapter 3.1] is usually used for comparing norms of vectors with different dimensions. In our case, the number of the pairs of the given data N is constant. We can therefore use as the measure the Euclidean norm of the residual vector $\mathbf{r}$.

3.4 Least squares model fitting

We will now show a common method of choosing the model parameters based on least squares. As we already said, we want to minimize the Euclidean norm of the residual vector $\|\mathbf{r}\|_2$. Expressing $\hat{y}_i = \hat{f}(\mathbf{x}_i)$ in the general model of data fitting (18), we get

$$\hat{y}_i = \theta_1 f_1(\mathbf{x}_i) + \cdots + \theta_p f_p(\mathbf{x}_i). \tag{21}$$

Let us define the $N \times p$ matrix A as

$$A_{ij} = \hat{f}_j(\mathbf{x}_i),\ i = 1, \ldots, N,\ j = 1, \ldots, p \tag{22}$$

and $\theta = (\theta_1, \ldots, \theta_p)^T$. The j-th column of A is the j-th basis function evaluated at each of the data points $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and its i-th row consists of the values of all the p basis functions on the ith data point $\mathbf{x}_i$. The whole vector $\hat{\mathbf{y}}$ can be then written as

$$\hat{\mathbf{y}} = A\theta. \tag{23}$$

Hence, the norm of the residual vector can be expressed as

$$\|\mathbf{r}\|_2 = \|A\theta - \mathbf{y}\|_2 . \tag{24}$$

Finding the parameter vector $\hat{\theta}$ which minimizes the norm $\|\mathbf{r}\|_2$ is a least squares problem with $\hat{\mathbf{x}} = \hat{\theta}$ and $\mathbf{b} = \mathbf{y}$. For a matrix A with full column rank, we can express the least squares approximation of the parameter vector θ as

$$\hat{\theta} = (A^T A)^{-1} A^T \mathbf{y}. \tag{25}$$

We say that the model parameter values $\hat{\theta}$ are obtained by *least squares fitting* on the data set. The N-vector $\hat{\mathbf{y}} = A\hat{\theta}$ is the vector of outcomes predicted by our model with the vector of parameters $\hat{\theta}$.

4 Least squares classification

We will now use a method based on least squares in the classification problem, which is a problem of identifying which of a finite number of groups a data belongs to. First, we will explain this method on a Boolean classifier. Then, we will construct a multi-class classifier from a set of Boolean classifiers. This chapter is based on [3, Chapter 14].

4.1 Boolean classifier based on least squares

In a *Boolean classification problem*, the outcome takes only two possible values (labels). The goal is to find an algorithm which decides what group the input $\mathbf{x}$ belongs to based on the values of its components (features).

First of all we carry out the ordinary least squares fitting, i.e. we choose the basis functions $f_1, \dots f_p$ and find the parameters $\theta_1, \dots \theta_p$ that minimize the sum

$$(\hat{f}(\mathbf{x}_1) - y_1)^2 + \cdots + (\hat{f}(\mathbf{x}_N) - y_N)^2, \tag{26}$$

where $\hat{f}(\mathbf{x}) = \theta_1 f_1(\mathbf{x}) + \cdots + \theta_p f_p(\mathbf{x})$ and $y_i = 1$ for $\mathbf{x}_i$ being an element of the group with the label 1 and $y_i = -1$ otherwise.

Once we have the data fitting function $\hat{f}(\mathbf{x})$, we can define the least squares classifier as:

$$\tilde{f}(\mathbf{x}) = \mathbf{sign}(\hat{f}(\mathbf{x})), \tag{27}$$

where $\mathbf{sign}(a) = +1$ for $a \geq 0$ and $\mathbf{sign}(a) = -1$ for $a < 0$.

The function $\tilde{f}$ defined above is called the *Boolean least squares classifier*.

The intuition behind this definition is simple. The value $\hat{f}(\mathbf{x}_i)$ should be high for $y_i = +1$ and low for $y_i = -1$, which practically means that basis functions applied on two inputs from the same group should give us similar outputs and basis functions applied on inputs from different groups should give us, on the

contrast, different outputs. The higher the absolute value of $\tilde{f}(\mathbf{x})$ is, the more confident about our guess we are.

Application on handwritten digits recognition problem

For each digit we can define a Boolean classifier which distinguishes the digit from the other nine digits. In our experiments we will demonstrate a classifier which distinguishes the digit zero.

Simple least squares solution. First of all, we will only care about the $28 \times 28 = 784$ pixel intensities of each image. We will apply the least squares classification with the basis functions $f_1, \dots f_{784}$; $f_i : \mathbb{R}^n \to [0, 1]$, defined as $f_i(\mathbf{x}) = \mathbf{e}_i\mathbf{x}$, where $\mathbf{e}_i$ is an element of a standard basis. These functions map the input vectors into scalars of pixel intensities of the gray-scale images represented by these vectors at corresponding positions. Additionally, we add one more basis function f_{785}, defined as $f_{785}(\mathbf{x}) = 1,\ \mathbf{x} \in \mathbb{R}^{784}$, to the set of basis functions. We now have $p = 785$ basis functions in the least squares classifier.

The results of this classification algorithm applied on the MNIST data set are following: from 10 000 digits in the test set where 909 of them are zero digits, 43 nonzero digits were wrongly classified as 0 (the false positive rate $\approx 4.7\%$) and 114 zero digits were not classified as zero (the true positive rate $\approx 88.8\%$). The overall error rate is approx. 1.57%.

Least squares solution using clustering. To improve our classifier, we now add 20 more basis functions based on the result of so called *k-means algorithm.*

K-means algorithm

The k-means algorithm is an algorithm which aims to partition a set of N vectors $\mathbf{x}_1, \dots, \mathbf{x}_N$ into k clusters so that every vector $\mathbf{x}_i$ belongs to the cluster with the nearest mean measured as the Euclidean distance.

`Algorithm:` [1, Algorithm 4.1]
given: a set of N vectors $\mathbf{x}_1, \dots, \mathbf{x}_N$ and an initial set of k cluster representative

vectors $\mathbf{z}_1, \ldots, \mathbf{z}_k$ (k different vectors that are randomly chosen from the set on N vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$)

repeat until the cluster representative vectors no longer move:

1. *Partition the vectors into k groups.* For each $\mathbf{x}_i, i = 1, \ldots, N$ assign $\mathbf{x}_i$ to the group associated with the nearest cluster representative.

2. *Update cluster representatives.* For each cluster $j = 1, \ldots, k$, set $\mathbf{z}_j$ to be the mean of the vectors in the group j.

Running this algorithm with the number of required clusters $k = 20$, we will get 20 representatives $\mathbf{z}_1, \ldots \mathbf{z}_{20}$ of these clusters. We define the new basis functions $exp(-\left\|\mathbf{x} - \mathbf{z}_i\right\|^2 / \sigma),\ i = 1, \ldots 20,\ \sigma = 10$.

Adding these 20 basis functions to our classifier, we can see that the number of digits that are wrongly classified as zero decreases from 43 to 26 (the false positive rate decreases from approx. 4.7% to approx. 2.86%) and the number of zero digits that were not classified as zero decreases from 114 to 90 (the true positive rate increases from approx. 88.8% to approx. 90.8%). The overall error rate decreases from approx. 1.57% to approx. 1.16%.

4.2 Multi-class classifier based on least squares

Unlike in Boolean classification problem, where the number of labels equals two, in *multi-class classification problem* we have $|L| > 2$, where L is the set of all possible label values. A multi-class classifier is then a function $\tilde{f} : \mathbb{R}^n \to L$, which gives us a prediction of a label $\tilde{f}(\mathbf{x})$ for every $\mathbf{x} \in \mathbb{R}^n$.

To create a multi-class classifier, we will extend the Boolean classifier based on least squares which we have explained in the previous part. Suppose we have a data set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N),\ N \in \mathbb{N}$, where $\mathbf{x}_i$ are the feature vectors and y_i corresponding labels. From this data set we will create $|L|$ new data sets. For every label value $l \in L$ we create a data set $(\mathbf{x}_1^l, y_1^l), \ldots, (\mathbf{x}_N^l, y_N^l)$, where $\mathbf{x}_i^l = \mathbf{x}_i,\ i = 1, \ldots, N$ and $y_i^l = 1$ if $y_i = l$ and $y_i^l = -1$ if $y_i \neq l$. Each of this new data sets will be used to create the corresponding Boolean classifier (for each

label value $l \in L$, $(\mathbf{x}_1^l, y_1^l), \ldots, (\mathbf{x}_N^l, y_N^l)$ is the data set for the Boolean classifier for the label l against the others).

From this $|L|$ Boolean classifiers we will create a classifier which chooses one of the $|L|$ possible labels by selecting the label for which the level of confidence of the Boolean classifier is the highest, i.e. the corresponding data fitting function applied on the given input vector has the highest value. Our classifier is then

$$\tilde{f}(\mathbf{x}) = \underset{l \in L}{\operatorname{argmax}} \hat{f}^l(\mathbf{x}), \tag{28}$$

where $\hat{f}^l$ is the data fitting function (18) for the label l against the others.

Application on handwritten digits recognition problem

We will now illustrate this method by applying it on the MNIST data set. For every digit $k = 0, \ldots, 9$, we compute the least squares Boolean classifier

$$\tilde{f}^{(k)}(\mathbf{x}) = sign(\hat{\theta}_1^{(k)} f_1(\mathbf{x}) + \cdots + \hat{\theta}_p^{(k)} f_p(\mathbf{x})). \tag{29}$$

Note: Here $\hat{\theta}^{(k)}$ is the vector of weights which minimizes the sum squared error (26) computed during the process of finding the Boolean classifier for the digit k.

From those ten data fits, we create a multi-class classifier

$$\tilde{f}(\mathbf{x}) = \underset{k \in K}{\operatorname{argmax}} \hat{f}^{(k)}(\mathbf{x}), \tag{30}$$

where $K = \{0, \ldots, 9\}$.

Applying the multi-class classifier on the MNIST data set with the simple 784 basis functions based on the pixel intensities and one additional basis function with the constant value of one, the error rate on the test set is approx. 14%. Adding the 20 basis functions based on the k-means algorithm, the error rate decreases to approx. 10%.

5 Regularized least squares

5.1 Regularized least squares fitting

When searching for an appropriate data fitting model, we should be aware of the issue of data over-fitting, which can be described as a situation when the model achieves a good fit on the given data set but works poorly on previously unseen data. This may happen when the model is too specific and does not generalize well.

The easiest way to avoid this is to keep the model simple, i.e. not to use too many basis functions. Another possibility is to use so called *regularization* [1, Chapter 15.4.1].

To motivate regularization, consider the general data fitting model

$$\hat{f}(\mathbf{x}) = \theta_1 f_1(\mathbf{x}) + \cdots + \theta_p f_p(\mathbf{x}). \tag{31}$$

In this model, θ_i shows how much our prediction depends on $f_i(\mathbf{x})$. Therefore if θ_i is large, our prediction will be very sensitive to changes in the value of $f_i(\mathbf{x})$, such as those we might expect in new, unseen data. This suggests that we should prefer θ_i to be small not to make our model too sensitive. This leads to a least squares problem with two criteria, the primary objective $\|A\theta - \mathbf{y}\|_2^2$ and the secondary objective $\|\theta\|_2^2$. This so called *bi-criterion* problem can be solved by minimizing the sum

$$\|A\theta - \ \mathbf{y}\|_2^2 + \lambda \|\theta\|_2^2, \tag{32}$$

where $\lambda > 0$ is called the *regularization parameter*. This method is called the *Tikhonov regularization* [3, Chapter 6.5 - (6.22)].

Regularization path For every choice of λ, we get a different vector of parameters θ. The way the parameter vector θ changes with λ is called the *regularization path*. To choose an appropriate value of the parameter λ, we need to use some model validation technique (see [1, Chapter 13.2]) that assess the generalization

ability of the model, i.e. the ability of the model to predict the outcomes for previously unseen data values. Two main examples of model validation techniques are *out of sample vali dation* and *cross validation* [1, Chapter 13.2]:

Out of sample

In the out of sample validation, the whole set of data that we use to create the data fitting model is divided into two sets: a *training set* and a *test set*. To create the data fitting model, we only use the data in the training set. The model is then judged by the prediction error on the test set, which can be measured by the RMS value or by a norm of the residual vector.

Cross validation

Cross-validation is a model validation technique that extends the out of sample validation. The original data set is divided into ten sets which are called *folds*. We then create ten data fitting models so that we always use one of the folds as a test set and the rest of them as a training set (each of the folds will once be used as a test set). In this way, we get ten models, each of which is assessed by its performance on the corresponding test set. If the performance of these ten models on its test sets is similar, we expect the same or similar performance on new unseen data.

Choosing the regularization parameter The prediction error for the training data increases as λ increases but the test set prediction error can typically be reduced when choosing the appropriate value of the parameter λ. A good practice is to choose the biggest λ for which the test set prediction error is near its minimal value. This gives us a model of minimum sensitivity that makes a good prediction on the test set.

Notes:

1. Due to the regularization, we accept a worse value of the sum square fit $\|A\theta - \mathbf{y}\|_2$ in return for smaller values of θ.

2. Apart from Tikhonov regularization, there exist other methods of regu-

larization (see [2, Chapter 6.3.2]). As an example, we can mention *Lasso regression*, which minimizes the sum $\|A\theta - \mathbf{y}\|_2^2 + \lambda \|\theta\|_1$, or so called *Elastic net*, where there are even three criteria as it is a combination of both previous methods. This method minimizes the sum $\|A\theta - \mathbf{y}\|_2^2 + \lambda_1 \|\theta\|_1 + \lambda_2 \|\theta\|_2^2$.

5.2 Regularized least squares classification

In this section we will improve the least squares classifier which we applied on the MNIST data set in previous chapter. This section is based on [1, Chapter 15.4.2].

First, we will consider a classifier based on 28×28 pixel intensities plus one additional basis function with the constant value of one only (785 basis functions). The k-th classifier used to distinguish the digit k from the other digits has the form $\hat{f}^{(k)}(\mathbf{x}) = sign(\theta_1^{(k)} f_1(\mathbf{x}) + \cdots + \theta_p^{(k)} f_p(\mathbf{x}))$. However, the vector θ is now computed by minimizing the sum

$$\sum_{i=1}^{N} (\theta_1^{(k)} f_1(\mathbf{x}_i) + \cdots + \theta_p^{(k)} f_p(\mathbf{x}_i) - y_i^{(k)})^2 + \lambda \left\|\theta^{(k)}\right\|_2^2, \tag{33}$$

where again the index (k) denotes the vector θ from the k-th classifier, $y_i^{(k)} = 1$ if the training example $\mathbf{x}_i$ is an image of the digit k and $y_i^{(k)} = -1$ otherwise. The multi-class classifier is then defined as

$$\hat{f}(\mathbf{x}) = \underset{k=0,\ldots,9}{\operatorname{argmax}} (\theta_1^{(k)} f_1(\mathbf{x}) + \cdots + \theta_p^{(k)} f_p(\mathbf{x})). \tag{34}$$

Applying this classifier with the parameter $\lambda = 0.5$ and the simple 784 basis functions base on the pixel intensities plus one addition basis function with the constant value of one on the MNIST data set, we get the result with the error rate on the test set approx. 13.9%. Adding the 20 basis functions base on the k-means algorithm, the error rate decreases to approx. 10.4% (note that in this case, the regularization worsened the results by 0.4%. We can see that in this case the parameter $\lambda = 0.5$ is not appropriate).

Regularization is a powerful way to avoid over-fitting. If we use regularization, we can fit a model with more basis functions then would be appropriate if we used the ordinary least squares fitting model. We will now show one more example of application of the regularized least squares classification on the MNIST data set. We will extend the original set of 785 basis functions by some new basis functions.

Regularized least squares classification with randomly generated basis functions In the first case, we expand the original 785 basis functions with 1000 extra basis functions defined as follows. We generate 1000 vectors and scalars $\mathbf{r}_j \in \mathbb{R}^{784}, s_j \in \mathbb{R}$, with randomly generated entries ± 1, which we will use to define nonlinear basis functions $f_{785+j} = max\{0, \mathbf{r}_j^T \mathbf{x} + s_j\}, j = 1, \ldots, 1000$. We now compute the multi-class classifier by solving (34) with the new set of basis functions.

Applying this classifier on the MNIST data set, we get the results with the error rates 5.18% for $\lambda = 0.2$, 5.20% for $\lambda = 0.5$ and 5.21% for $\lambda = 1$.

6 Other applications of LSM

There is a huge variety of problems from various parts of mathematics which can be solved using the method of least squares. We will now show two more important examples.

6.1 Least norm problem

Let us consider a system of equations $A\mathbf{x} = \mathbf{b}$ where $A \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$. We can assume without loss of generality that the rows of A are linearly independent, i.e. $m \leq n$. In case $m = n$, the only solution to this equation is $\mathbf{x} = A^{-1}\mathbf{b}$. We will now consider the situation when $m < n$. Such a system of linear equations is called *under-determined* and in case it is solvable, then there exist infinitely many solutions to this system. If we want to select one of these infinitely many solutions, we need to add further criteria for the solution. By the *least norm problem* we mean the problem of finding a solution to this system with minimal norm (see [2, Chapter 6.2]). In case we take as the norm the Euclidean norm $\|\cdot\|_2$ the problem can be solved using least squares.

Let $\mathbf{x}_0$ be an arbitrary solution of $A\mathbf{x} = \mathbf{b}$ and let $Z \in \mathbb{R}^{n\times k}$ be a matrix whose columns are a basis for the nullspace of A. The general solution to the equation $A\mathbf{x} = \mathbf{b}$ can then be expressed as $\mathbf{x}_0 + Z\mathbf{u}$ where $u \in \mathbb{R}^k$. Our goal is to minimize the norm $\|\mathbf{x}_0 + Z\mathbf{u}\|$. For $\mathbf{b} = (0, \ldots, 0)^T$, the least norm solution of $A\mathbf{x} = \mathbf{b}$ is $\mathbf{x} = (0, \ldots, 0)^T$. Suppose $\mathbf{b} \neq (0, \ldots, 0)^T$.

Lemma 6.1 Let $Z \in \mathbb{R}^{n\times k}, n > k$ be a matrix whose columns are linearly independent, $\mathbf{x}_0 \in \mathbb{R}^n$. The vector $\mathbf{u} \in \mathbb{R}^k$ minimizes the norm $\|\mathbf{x}_0 + Z\mathbf{u}\|_2$ if and only if $\mathbf{u}$ is the least squares approximate solution to the equation $Z\mathbf{u} = -\mathbf{x}_0$.

Proof. The implication from right to left comes from the definition of the least squares problem (1).
Conversely, the norm $\|\mathbf{x}_0 + Z\mathbf{u}\|_2$ is minimal if and only if the distance between $-\mathbf{x}_0$ and $Z\mathbf{u}$ is minimal which holds if and only if $Z\mathbf{u}$ is the projection of the vector $-\mathbf{x}_0$ onto the vector space generated by the columns of the matrix Z which

is equivalent to the vector $\mathbf{u}$ being the least squares approximate solution to the equation $Z\mathbf{u} = -\mathbf{x}_0$. □

The least norm problem has a simple geometric interpretation. The set of solutions $X = \{\mathbf{x};\ A\mathbf{x} = \mathbf{b}\}$ is an affine subspace of $\mathbb{R}^n$ and our objective is to find $\mathbf{x} \in X$ so that the distance between $\mathbf{x}$ and 0 is minimal. Hence, the vector $\mathbf{x}$ is nothing but the orthogonal projection of the point 0 onto the affine subspace X.

In Chapter 1 where we were talking about existence and uniqueness of a solution to the least squares problem, we mentioned that if the columns of the matrix A from an over-determined system of equations $A\mathbf{x} = \mathbf{b}$ are linearly dependent, there exist infinitely many least squares solutions to this system as there is infinitely many ways how to express the projection $\mathbf{b}_{R(A)}$ as a linear combination of the columns of A. Hence, we can think about some additional constraints, one of which is to find a solution with minimal norm. This is an example of the least norm problem. As we already derived, a vector $\hat{\mathbf{x}}$ is a solution to the least squares problem if and only if it is the solution to the system of equations $A^TA\mathbf{x} = A^T\mathbf{b}$. In case the columns of the matrix A are linearly dependent, the matrix A^TA is singular. Hence, there exist a matrix $L \in \mathbb{R}^{l\times n}$, $l < n$ whose rows are linearly independent and a vector $\mathbf{t} \in \mathbb{R}^l$, so that $A^TA\hat{\mathbf{x}} = A^T\mathbf{b}$ holds if and only if $L\hat{\mathbf{x}} = \mathbf{t}$. The system of equations $L\mathbf{x} = \mathbf{t}$ is under-determined and so we can find a solution to this equations as a solution to the least norm problem.

6.2 Maximum likelihood estimation

This example is based on [2, Chapter 7.1.1]. We will look at the statistical parameter estimation. Let us consider a family of probability distributions on $\mathbb{R}^m$ with densities $p_{\mathbf{x}}(\cdot)$ indexed by a vector of parameters $\mathbf{x} \in \mathbb{R}^n$. If we consider $p_{\mathbf{x}}(\cdot)$ to be a function of $\mathbf{x}$, for a fixed $\mathbf{y} \in \mathbb{R}^m$, the function $p_{\mathbf{x}}(\mathbf{y})$ is called the *likelihood function*. It is more convenient to work with its logarithm, which is called the *log-likelihood function*, and denoted l:

$$l(\mathbf{x}) = \log p_{\mathbf{x}}(\mathbf{y}). \tag{35}$$

We will now consider the problem of estimating the values of the parameter vector $\mathbf{x}$, based on observing one sample $\mathbf{y}$ from the distribution. A widely used method is to estimate $\mathbf{x}$ as

$$\hat{\mathbf{x}}_{ml} = \operatorname*{argmax}_{\mathbf{x}}(p_{\mathbf{x}}(\mathbf{y})) = \operatorname*{argmax}_{\mathbf{x}}(l(\mathbf{x})). \tag{36}$$

This method is called *maximum likelihood (ML) estimation* as it chooses as the estimate values of the parameters which maximize the likelihood (or log-likelihood) function for the observed value $\mathbf{y}$.

Linear measurement with IID noise Let us consider a linear measurement model

$$y_i = \mathbf{a}_i^T \mathbf{x} + \mathbf{v}_i,\ i = 1, \ldots, m,$$

where $\mathbf{x} \in \mathbb{R}^n$ is a vector of parameters to be estimated, $y_i \in \mathbb{R}$ are the observed quantities and v_i the measurement noise or errors which are assumed to be independent, identically distributed (IID) with density p on $\mathbb{R}$. The likelihood function is then

$$p_{\mathbf{x}}(\mathbf{y}) = \prod_{i=1}^{m} p(y_i - \mathbf{a}_i^T \mathbf{x}),$$

and so the log-likelihood function is

$$l(\mathbf{x}) = \log p_{\mathbf{x}}(\mathbf{y}) = \sum_{i=1}^{m} \log p(y_i - \mathbf{a}_i^T \mathbf{x}). \tag{37}$$

The ML estimate is then any $\mathbf{x}$ which maximizes the log-likelihood function (37).

ML estimation for some common noise densities

- *Gaussian noise.* Considering v_i to be Gaussian with zero mean and variance σ^2, the density is $p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-z^2}{2\sigma^2}}$ and so the log-likelihood function is

  $$l(\mathbf{x}) = -\frac{m}{2} \log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \left\|A\mathbf{x} - \mathbf{y}\right\|_2^2,$$

  where A is a matrix with rows $\mathbf{a}_1^T, \ldots, \mathbf{a}_m^T$. Hence, the ML estimate of $\mathbf{x}$ is $\hat{\mathbf{x}}_{ml} = \operatorname*{argmin}_{\mathbf{x}} \left\|A\mathbf{x} - \mathbf{y}\right\|_2^2$, a solution to a least squares problem.

- *Laplacian noise.* For v_i being Laplacian, i.e. their densities are $p(z) = \frac{1}{2a}e^{-\frac{|z|}{a}}, a > 0$, the ML estimate is $\mathbf{x}_{ml} = \underset{\mathbf{x}}{\operatorname{argmin}} \|A\mathbf{x} - \mathbf{y}\|_1$, the solution to the $l_1 - norm$ approximation problem.

- *Uniform noise.* When v_i are uniformly distributed on $[-a, a]$, the density has the form $p(z) = \frac{1}{2a}$ on $[-a, a]$. Hence the ML estimate is any $\mathbf{x}$ satisfying $\|A\mathbf{x} - \mathbf{y}\|_\infty \leq a$.

Conclusion

One goal of this thesis was to provide the mathematical theory which is necessary to understand the method of least squares and its usage in data fitting and data classification problem. We have explained that a solution to the least squares problem can be find as a solution to corresponding system of normal equations. We have provided basic formulae that were needed for computing with orthogonal matrices. These were important for understanding the effective solution to the system of normal equations via the QR-decomposition and SVD. We also have showed how the least squares data fitting and least squares classification can be improved using regularization.

The second goal was to illustrate all the theory covered in this text on practical examples. We have described and implemented an algorithm base on least squares which aims to recognize handwritten digits. The error rates of this algorithm depends on the definition of the basis functions used for the data classification and on the regularization parameter. This algorithm provided the best results when used with the simple basis functions based on pixel intensity of each pixel of the images of the digits together with basis functions based on randomly generated vectors and the regularization parameter $\lambda = 0.2$ described in section (5.2). In this case it recognized the previously unseen digit from the MNIST data set with the error rate 5.18%. Apart from the digits recognition problem we have also showed two more examples of practical usage of the method of least squares. The first one was the application of least squares in the least norm problem where we explained how to find a solution with minimal norm to an under-determined system of equations. In the second example we used the method of least squares to estimate the parameters of a statistical model by maximizing the likelihood function for the observed values.

Bibliography

[1] Stephen Boyd and Lieven Vandenberghe, *Vectors, Matrices, and Least Squares*, ROUGH DRAFT October 1, 2016

[2] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004. ISBN 0 521 83378 7

[3] Jurjen Duintjer Tebbens, Iveta Hnětynková, Martin Plešinger, Zdeněk Strakoš, Petr Tichý, *Analýza metod pro maticové výpočty základní metody*, Matfyzpress, Praha, 2012. ISBN 978-80-7378-201-6

[4] Libor Barto and Jiří Tůma, *Lineární algebra*,
$http://www.karlin.mff.cuni.cz/\ sir/la/LinAlg/skripta_old.pdf$

[5] Mansfield Merriman, On the History of the Method of Least Squares, Annals of Mathematics, 1877