

# Image Analysis Coursework Report

CRSiD: tmb76

University of Cambridge

# Contents

<b>1</b>	<b>Fundamentals of Image Analysis</b>	<b>3</b>
1.1	Exercise 1.1 . . . . .	3
1.1.1	Segmentation . . . . .	3
1.1.2	Discussion . . . . .	3
1.1.3	Self Implementation . . . . .	3
<b>2</b>	<b>Inverse Problems and Multiresolution Analysis</b>	<b>4</b>
2.1	Exercise 2.1 . . . . .	4
2.1.1	$l_1$ and $l_2$ Minimisation . . . . .	4
2.1.2	Discussion . . . . .	7
2.2	Exercise 2.2 . . . . .	7
2.2.1	Getting undersampled signals . . . . .	7
2.2.2	The Iterative Soft Thresholding Algorithm . . . . .	7
2.2.3	Discussion . . . . .	7
2.3	Exercise 2.3 . . . . .	7
2.3.1	Daubechies Wavelet Transform . . . . .	7
2.3.2	Thresholding Coefficients . . . . .	7
2.3.3	Discussion . . . . .	7
<b>3</b>	<b>Solving Inverse Problems</b>	<b>9</b>
3.1	Exercise 3.1 . . . . .	9
3.2	Exercise 3.2 . . . . .	9
3.3	Exercise 3.3 . . . . .	9

# Introduction

# Chapter 1

## Fundamentals of Image Analysis

### 1.1 Exercise 1.1

#### 1.1.1 Segmentation

#### 1.1.2 Discussion

#### 1.1.3 Self Implementation

# Chapter 2

## Inverse Problems and Multiresolution Analysis

### 2.1 Exercise 2.1

In this exercise, the solving of ill-posed inverse problems is done for a simple 1D example of fitting a straight line to noisy data. This will be done using a minimisation algorithm of the form:

$$\min_u ||Au - f||_{l_n} \quad (2.1)$$

Where  $A$  is the parameter matrix,  $u$  is the measured input, and  $l_n$  describes the type of distance used to measure the error between the model and the data. Here,  $A$  is the vector  $[a, b]$ , where  $a$  is the slope and  $b$  is the intercept of the fitted line. And the two choices considered are  $l_1$  and  $l_2$  minimisation.

#### 2.1.1 $l_1$ and $l_2$ Minimisation

As defined,  $l_1$  minimisation is the minimisation of the sum of the absolute values of the residuals [4], while  $l_2$  minimisation is the minimisation of the sum of the squares of the residuals [2]. Switching to a more explicit notation of this problem, this gives:

$$||(ax + b) - f||_{l_1} = \sum_{i=1}^n |(ax_i + b) - f_i| \quad (2.2)$$

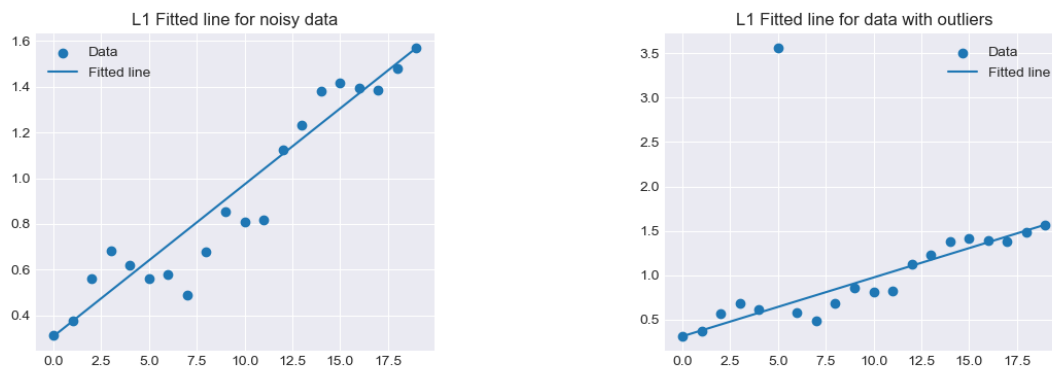
$$||(ax + b) - f||_{l_2} = \sqrt{\sum_{i=1}^n ((ax_i + b) - f_i)^2} \quad (2.3)$$

	Slope	Intercept
Noisy Data	0.0663	0.3092
Data w/ Outliers	0.0662	0.3120

Table 2.1: Table of the fitted parameters for the straight line using  $l_1$  minimisation.

The  $l_1$  minimisation problem is quite uncommon due to it not being differentiable, and resulting in the problem not having an analytical solution.

In this problem, some noisy data was obtained from two files, `y_line.txt` and `y_outlier_line.txt`, where the latter has the same data but with one outlier. The data was then fitted with a straight line using the two minimisation methods. For  $l_1$ , the `scipy.optimize.minimize` function was used using the Sequential Least Squares Programming (SLSQP) method. This is a way to solve the minimizing problem using linear programming, by replacing the original problem with a series of quadratic problems that are easier to solve [1]. Using this method, the fitted parameters were obtained and are shown in Table 2.1, and the plots of the fitted lines are shown in Figure 2.1.

Figure 2.1: Plots of the fitted straight line to the noisy (left), and outlier (right) data using  $l_1$  minimisation, assuming the x-coordinate of the data points is the index of the data point.

For  $l_2$  minimisation, the `scikit-learn` library was used to fit the data using the `LinearRegression` class, which uses the Ordinary Least Squares method to fit the data [3]. The fitted parameters were obtained and are shown in Table 2.2, and the plots of the fitted lines are shown in Figure 2.2.

	<b>Slope</b>	<b>Intercept</b>
<b>Noisy Data</b>	0.0665	0.2838
<b>Data w/ Outliers</b>	0.0462	0.6266

Table 2.2: Table of the fitted parameters for the straight line using  $l_2$  minimisation.

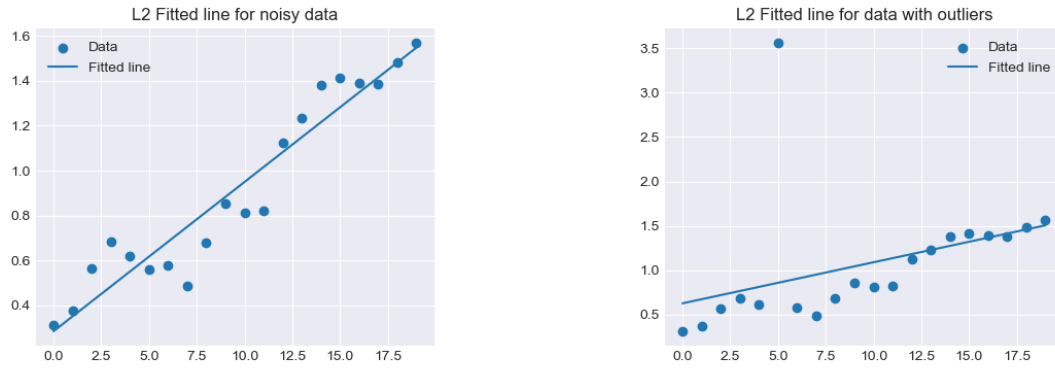


Figure 2.2: Plots of the fitted straight line to the noisy (left), and outlier (right) data using  $l_2$  minimisation, assuming the x-coordinate of the data points is the index of the data point.

### 2.1.2 Discussion

It is clear that  $l_1$  minimisation is more robust to outliers than  $l_2$  minimisation, as the  $l_1$  fitted line for the data with outliers is almost identical to the line fitted to the noisy data. Whereas the  $l_2$  fitted line for the data with outliers differs significantly from the line fitted to the noisy data. This can be explained from the definition of the two minimisation methods, where due to the squaring of the difference in  $l_2$  minimisation, the effect of outliers on the objective function ( $\|(ax + b) - f\|_{l_2}$ ) is amplified.

## 2.2 Exercise 2.2

In this exercise, the importance of random undersampling in compressed sensing theory. In order to do this, random and uniform undersampling will be compared in the reconstruction of a noisy signal, by solving:

$$\arg \min \left( \frac{1}{2} \|\mathcal{F}_\Omega \hat{x} - y\|_2^2 + \lambda |\hat{x}|_1 \right) \quad (2.4)$$

Using an iterative soft thresholding algorithm, with a data consistency constraint:

$$\hat{f}_{i+1}[j] = \begin{cases} \hat{f}_i[j] & \text{if } y[j] = 0 \\ y[j] & \text{otherwise} \end{cases} \quad (2.5)$$

### 2.2.1 Getting undersampled signals

First, a noisy signal is created. This is done by first creating a zero-filled vector of length 100 and replacing 10 entries by non-zero coefficients between 0 and 1. Gaussian noise, with mean 0 and standard deviation 0.05 is then added to all entries. This results in a noisy vector shown in Figure 2.3.

### 2.2.2 The Iterative Soft Thresholding Algorithm

### 2.2.3 Discussion

## 2.3 Exercise 2.3

### 2.3.1 Daubechies Wavelet Transform

### 2.3.2 Thresholding Coefficients

### 2.3.3 Discussion



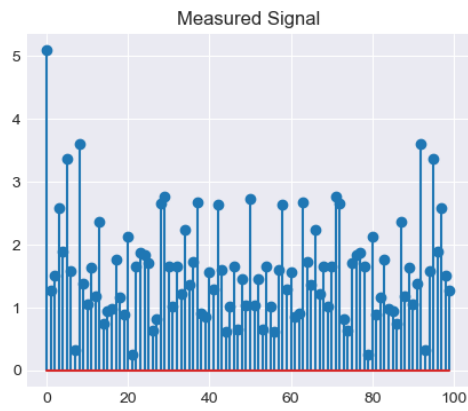


Figure 2.3: Plot of the noisy signal.

# Chapter 3

## Solving Inverse Problems

3.1 Exercise 3.1

3.2 Exercise 3.2

3.3 Exercise 3.3

# Bibliography

- [1] Math Stack Exchange. How does lsqp (sequential least squares programming) algorithm work? *Math Stack Exchange*, 2021.
- [2] Margalit and Rabinoff. The method of least squares. *Interactive Linear Algebra*, 2021.
- [3] Scikit-learn. Linear regression. *Scikit-learn Documentation*, 2021.
- [4] Wikipedia. Least absolute deviations. *Wikipedia, The Free Encyclopedia*, 2021.