

# Learning from data: Linear Regression

Christian Forssén<sup>1</sup>

Morten Hjorth-Jensen<sup>2,3</sup>

<sup>1</sup>Department of Physics, Chalmers University of Technology, Sweden

<sup>2</sup>Department of Physics, University of Oslo

<sup>3</sup>Department of Physics and Astronomy and National Superconducting Cyclotron Laboratory, Michigan State University

Sep 1, 2019

## 0.1 Why Linear Regression (aka Ordinary Least Squares and family)

Fitting a continuous function with linear parameterization in terms of the parameters  $\theta$ .

- Method of choice for fitting a continuous function!
- Gives an excellent introduction to central Machine Learning features with **understandable pedagogical** links to other methods like **Neural Networks, Support Vector Machines** etc
- Analytical expression for the fitting parameters  $\theta$
- Analytical expressions for statistical properties like mean values, variances, confidence intervals and more
- Analytical relation with probabilistic interpretations
- Easy to introduce basic concepts like bias-variance tradeoff, cross-validation, resampling and regularization techniques and many other ML topics
- Easy to code! And links well with classification problems and logistic regression and neural networks
- Allows for **easy** hands-on understanding of gradient descent methods
- and many more features

**Regression analysis, overarching aims.**

Regression modeling deals with the description of the sampling distribution of a given random variable  $y$  and how it varies as function of another variable or a set of such variables  $\mathbf{x} = [x_0, x_1, \dots, x_{n-1}]^T$ . The first variable is called the **dependent**, the **outcome** or the **response** variable while the set of variables  $\mathbf{x}$  is called the independent variable, or the predictor variable or the explanatory variable.

A regression model aims at finding a likelihood function  $p(\mathbf{y}|\mathbf{x})$ , that is the conditional distribution for  $\mathbf{y}$  with a given  $\mathbf{x}$ . The estimation of  $p(\mathbf{y}|\mathbf{x})$  is made using a data set with

- $n$  cases  $i = 0, 1, 2, \dots, n - 1$
- Response (target, dependent or outcome) variable  $y_i$  with  $i = 0, 1, 2, \dots, n - 1$
- $p$  so-called explanatory (independent or predictor) variables  $\mathbf{x}_i = [x_{i0}, x_{i1}, \dots, x_{ip-1}]$  with  $i = 0, 1, 2, \dots, n - 1$  and explanatory variables running from 0 to  $p - 1$ . See below for more explicit examples.

The goal of the regression analysis is to extract/exploit relationship between  $\mathbf{y}$  and  $\mathbf{x}$  in or to infer causal dependencies, approximations to the likelihood functions, functional relationships and to make predictions, making fits and many other things.

Consider an experiment in which  $p$  characteristics of  $n$  samples are measured. The data from this experiment, for various explanatory variables  $p$  are normally represented by a matrix  $\mathbf{X}$ .

The matrix  $\mathbf{X}$  is called the *design matrix*. Additional information of the samples is available in the form of  $\mathbf{y}$  (also as above). The variable  $\mathbf{y}$  is generally referred to as the *response variable*. The aim of regression analysis is to explain  $\mathbf{y}$  in terms of  $\mathbf{X}$  through a functional relationship like  $y_i = f(\mathbf{X}_{i,*})$ . When no prior knowledge on the form of  $f(\cdot)$  is available, it is common to assume a linear relationship between  $\mathbf{X}$  and  $\mathbf{y}$ . This assumption gives rise to the *linear regression model* where  $\boldsymbol{\theta} = [\theta_0, \dots, \theta_{p-1}]^T$  are the *regression parameters*.

Linear regression gives us a set of analytical equations for the parameters  $\theta_j$ .

**Example: Liquid-drop model for nuclear binding energies.**

In order to understand the relation among the predictors  $p$ , the set of data  $n$  and the target (outcome, output etc)  $\mathbf{y}$ , consider the model we discussed for describing nuclear binding energies.

There we assumed that we could parametrize the data using a polynomial approximation based on the liquid drop model. Assuming

$$BE(A) = a_0 + a_1 A + a_2 A^{2/3} + a_3 A^{-1/3} + a_4 A^{-1},$$

we have five predictors, that is the intercept, the  $A$  dependent term, the  $A^{2/3}$  term and the  $A^{-1/3}$  and  $A^{-1}$  terms. This gives  $p = 0, 1, 2, 3, 4$ . Furthermore we have  $n$  entries for each predictor. It means that our design matrix is a  $p \times n$  matrix  $\mathbf{X}$ .

## 0.2 General linear models

### Polynomial basis functions.

Before we proceed let us study a case from linear algebra where we aim at fitting a set of data  $\mathbf{y} = [y_0, y_1, \dots, y_{n-1}]$ . We could think of these data as a result of an experiment or a complicated numerical experiment. These data are functions of a series of variables  $\mathbf{x} = [x_0, x_1, \dots, x_{n-1}]$ , that is  $y_i = y(x_i)$  with  $i = 0, 1, 2, \dots, n-1$ . The variables  $x_i$  could represent physical quantities like time, temperature, position etc. We assume that  $y(x)$  is a smooth function.

Since obtaining these data points may not be trivial, we want to use these data to fit a function which can allow us to make predictions for values of  $y$  which are not in the present set. The perhaps simplest approach is to assume we can parametrize our function in terms of a polynomial of degree  $n-1$  with  $n$  points, that is

$$y = y(x) \rightarrow y(x_i) = \tilde{y}_i + \epsilon_i = \sum_{j=0}^{n-1} \theta_j x_i^j + \epsilon_i,$$

where  $\epsilon_i$  is the error in our approximation.

For every set of values  $y_i, x_i$  we have thus the corresponding set of equations

$$\begin{aligned} y_0 &= \theta_0 + \theta_1 x_0^1 + \theta_2 x_0^2 + \cdots + \theta_{n-1} x_0^{n-1} + \epsilon_0 \\ y_1 &= \theta_0 + \theta_1 x_1^1 + \theta_2 x_1^2 + \cdots + \theta_{n-1} x_1^{n-1} + \epsilon_1 \\ y_2 &= \theta_0 + \theta_1 x_2^1 + \theta_2 x_2^2 + \cdots + \theta_{n-1} x_2^{n-1} + \epsilon_2 \\ &\dots\dots\dots \\ y_{n-1} &= \theta_0 + \theta_1 x_{n-1}^1 + \theta_2 x_{n-1}^2 + \cdots + \theta_{n-1} x_{n-1}^{n-1} + \epsilon_{n-1}. \end{aligned}$$

Defining the vectors

$$\mathbf{y} = [y_0, y_1, y_2, \dots, y_{n-1}]^T,$$

and

$$\boldsymbol{\theta} = [\theta_0, \theta_1, \theta_2, \dots, \theta_{n-1}]^T,$$

and

$$\boldsymbol{\epsilon} = [\epsilon_0, \epsilon_1, \epsilon_2, \dots, \epsilon_{n-1}]^T,$$

and the design matrix

$$\mathbf{X} = \begin{bmatrix} 1 & x_0^1 & x_0^2 & \dots & \dots & x_0^{n-1} \\ 1 & x_1^1 & x_1^2 & \dots & \dots & x_1^{n-1} \\ 1 & x_2^1 & x_2^2 & \dots & \dots & x_2^{n-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n-1}^1 & x_{n-1}^2 & \dots & \dots & x_{n-1}^{n-1} \end{bmatrix}$$

we can rewrite our equations as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\epsilon}.$$

The above design matrix is called a [Vandermonde matrix](#).

### General basis functions.

We are obviously not limited to the above polynomial expansions. We could replace the various powers of  $x$  with elements of Fourier series or instead of  $x_i^j$  we could have  $\cos(jx_i)$  or  $\sin(jx_i)$ , or time series or other orthogonal functions. For every set of values  $y_i, x_i$  we can then generalize the equations to

$$\begin{aligned}
y_0 &= \theta_0 x_{00} + \theta_1 x_{01} + \theta_2 x_{02} + \cdots + \theta_{n-1} x_{0n-1} + \epsilon_0 \\
y_1 &= \theta_0 x_{10} + \theta_1 x_{11} + \theta_2 x_{12} + \cdots + \theta_{n-1} x_{1n-1} + \epsilon_1 \\
y_2 &= \theta_0 x_{20} + \theta_1 x_{21} + \theta_2 x_{22} + \cdots + \theta_{n-1} x_{2n-1} + \epsilon_2 \\
&\dots\dots\dots \\
y_i &= \theta_0 x_{i0} + \theta_1 x_{i1} + \theta_2 x_{i2} + \cdots + \theta_{n-1} x_{in-1} + \epsilon_i \\
&\dots\dots\dots \\
y_{n-1} &= \theta_0 x_{n-1,0} + \theta_1 x_{n-1,1} + \theta_2 x_{n-1,2} + \cdots + \theta_{n-1} x_{n-1,n-1} + \epsilon_{n-1}.
\end{aligned}$$

**Note that we have  $p = n$  here. The matrix is symmetric. This is generally not the case!**

We redefine in turn the matrix  $\mathbf{X}$  as

$$\mathbf{X} = \begin{bmatrix} x_{00} & x_{01} & x_{02} & \dots & \dots & x_{0,n-1} \\ x_{10} & x_{11} & x_{12} & \dots & \dots & x_{1,n-1} \\ x_{20} & x_{21} & x_{22} & \dots & \dots & x_{2,n-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n-1,0} & x_{n-1,1} & x_{n-1,2} & \dots & \dots & x_{n-1,n-1} \end{bmatrix}$$

and without loss of generality we rewrite again our equations as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\epsilon}.$$

The left-hand side of this equation is known. Our error vector  $\boldsymbol{\epsilon}$  and the parameter vector  $\boldsymbol{\theta}$  are our unknown quantities. How can we obtain the optimal set of  $\theta_i$  values?

We have defined the matrix  $\mathbf{X}$  via the equations

$$\begin{aligned}
y_0 &= \theta_0 x_{00} + \theta_1 x_{01} + \theta_2 x_{02} + \cdots + \theta_{n-1} x_{0n-1} + \epsilon_0 \\
y_1 &= \theta_0 x_{10} + \theta_1 x_{11} + \theta_2 x_{12} + \cdots + \theta_{n-1} x_{1n-1} + \epsilon_1 \\
y_2 &= \theta_0 x_{20} + \theta_1 x_{21} + \theta_2 x_{22} + \cdots + \theta_{n-1} x_{2n-1} + \epsilon_1 \\
&\dots\dots\dots \\
y_i &= \theta_0 x_{i0} + \theta_1 x_{i1} + \theta_2 x_{i2} + \cdots + \theta_{n-1} x_{in-1} + \epsilon_i \\
&\dots\dots\dots \\
y_{n-1} &= \theta_0 x_{n-1,0} + \theta_1 x_{n-1,1} + \theta_2 x_{n-1,2} + \cdots + \theta_{n-1} x_{n-1,n-1} + \epsilon_{n-1}.
\end{aligned}$$

As we noted above, we stayed with a system with the design matrix  $\mathbf{X} \in \mathbb{R}^{n \times n}$ , that is we have  $p = n$ . For reasons to come later (algorithmic arguments) we will hereafter define our matrix as  $\mathbf{X} \in \mathbb{R}^{n \times p}$ , with the predictors referring to the column numbers and the entries  $n$  being the row elements.

With the above we use the design matrix to define the approximation  $\tilde{\mathbf{y}}$  via the unknown quantity  $\boldsymbol{\theta}$  as

$$\tilde{\mathbf{y}} = \mathbf{X}\boldsymbol{\theta},$$

and in order to find the optimal parameters  $\theta_i$  instead of solving the above linear algebra problem, we define a function which gives a measure of the spread between the values  $y_i$  (which represent hopefully the exact values) and the parameterized values  $\tilde{y}_i$ , namely

$$C(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \tilde{y}_i)^2 = \frac{1}{n} \left\{ (\mathbf{y} - \tilde{\mathbf{y}})^T (\mathbf{y} - \tilde{\mathbf{y}}) \right\},$$

or using the matrix  $\mathbf{X}$  and in a more compact matrix-vector notation as

$$C(\boldsymbol{\theta}) = \frac{1}{n} \left\{ (\mathbf{y} - \mathbf{X}^T \boldsymbol{\theta})^T (\mathbf{y} - \mathbf{X}^T \boldsymbol{\theta}) \right\}.$$

This function is one possible way to define the so-called cost function.

It is also common to define the function  $Q$  as

$$C(\boldsymbol{\theta}) = \frac{1}{2n} \sum_{i=0}^{n-1} (y_i - \tilde{y}_i)^2,$$

since when taking the first derivative with respect to the unknown parameters  $\theta$ , the factor of 2 cancels out.

The function

$$C(\boldsymbol{\theta}) = \frac{1}{n} \left\{ (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) \right\},$$

can be linked to the variance of the quantity  $y_i$  if we interpret the latter as the mean value. When linking (see the discussion below) with the maximum likelihood approach below, we will indeed interpret  $y_i$  as a mean value

$$y_i = \langle y_i \rangle = \theta_0 x_{i,0} + \theta_1 x_{i,1} + \theta_2 x_{i,2} + \cdots + \theta_{n-1} x_{i,n-1} + \epsilon_i,$$

where  $\langle y_i \rangle$  is the mean value. Keep in mind also that till now we have treated  $y_i$  as the exact value. Normally, the response (dependent or outcome) variable  $y_i$  the outcome of a numerical experiment or another type of experiment and is thus only an approximation to the true value. It is then always accompanied by an error estimate, often limited to a statistical error estimate given by the standard deviation discussed earlier. In the discussion here we will treat  $y_i$  as our exact value for the response variable.

In order to find the parameters  $\theta_i$  we will then minimize the spread of  $C(\boldsymbol{\theta})$ , that is we are going to solve the problem

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{n} \left\{ (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) \right\}.$$

In practical terms it means we will require

$$\frac{\partial C(\boldsymbol{\theta})}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} \left[ \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \theta_0 x_{i,0} - \theta_1 x_{i,1} - \theta_2 x_{i,2} - \cdots - \theta_{n-1} x_{i,n-1})^2 \right] = 0,$$

which results in

$$\frac{\partial C(\boldsymbol{\theta})}{\partial \theta_j} = -\frac{2}{n} \left[ \sum_{i=0}^{n-1} x_{ij} (y_i - \theta_0 x_{i,0} - \theta_1 x_{i,1} - \theta_2 x_{i,2} - \cdots - \theta_{n-1} x_{i,n-1}) \right] = 0,$$

or in a matrix-vector form as

$$\frac{\partial C(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0 = \mathbf{X}^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}).$$

We can rewrite

$$\frac{\partial C(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0 = \mathbf{X}^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}),$$

as

$$\mathbf{X}^T \mathbf{y} = \mathbf{X}^T \mathbf{X} \boldsymbol{\theta},$$

and if the matrix  $\mathbf{X}^T \mathbf{X}$  is invertible we have the solution

$$\boldsymbol{\theta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$$

We note also that since our design matrix is defined as  $\mathbf{X} \in \mathbb{R}^{n \times p}$ , the product  $\mathbf{X}^T \mathbf{X} \in \mathbb{R}^{p \times p}$ . In the liquid drop model example from the Intro lecture, we had  $p = 5$  ( $p \ll n$ ) meaning that we end up with inverting a small  $5 \times 5$  matrix. This is a rather common situation, in many cases we end up with low-dimensional matrices to invert. The methods discussed here and for many other supervised learning algorithms like classification with logistic regression or support vector machines, exhibit dimensionalities which allow for the usage of direct linear algebra methods such as **LU** decomposition or **Singular Value Decomposition** (SVD) for finding the inverse of the matrix  $\mathbf{X}^T \mathbf{X}$ .

**Small question:** What kind of problems can we expect when inverting the matrix  $\mathbf{X}^T \mathbf{X}$ ?

**Some useful matrix and vector expressions.** The following matrix and vector relation will be useful here and for the rest of the course. Vectors are always written as boldfaced lower case letters and matrices as upper case boldfaced letters.

$$\begin{aligned} \frac{\partial (\mathbf{b}^T \mathbf{a})}{\partial \mathbf{a}} &= \mathbf{b}, \\ \frac{\partial (\mathbf{a}^T \mathbf{A} \mathbf{a})}{\partial \mathbf{a}} &= (\mathbf{A} + \mathbf{A}^T) \mathbf{a}, \\ \frac{\partial \text{tr}(\mathbf{B} \mathbf{A})}{\partial \mathbf{A}} &= \mathbf{B}^T, \\ \frac{\partial \log |\mathbf{A}|}{\partial \mathbf{A}} &= (\mathbf{A}^{-1})^T. \end{aligned}$$

The residuals  $\boldsymbol{\epsilon}$  are in turn given by

$$\boldsymbol{\epsilon} = \mathbf{y} - \tilde{\mathbf{y}} = \mathbf{y} - \mathbf{X}\boldsymbol{\theta},$$



and with

$$\mathbf{X}^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = 0,$$

we have

$$\mathbf{X}^T \boldsymbol{\epsilon} = \mathbf{X}^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = 0,$$

meaning that the solution for  $\boldsymbol{\theta}$  is the one which minimizes the residuals. Later we will link this with the maximum likelihood approach.

### 0.3 Adding error analysis and training set up

We can easily test our fit by computing various **cost functions** (or **training scores**). Several such cost functions are used in machine learning applications. First we have the **Mean-Squared Error** (MSE)

$$\text{MSE}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n (y_{\text{data},i} - y_{\text{model},i}(\boldsymbol{\theta}))^2,$$

where we have  $n$  training data and our model is a function of the parameter vector  $\boldsymbol{\theta}$ .

Furthermore, we have the **mean absolute error** (MAE) defined as.

$$\text{MAE}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n |y_{\text{data},i} - y_{\text{model},i}(\boldsymbol{\theta})|,$$

And the  $R^2$  score, also known as *coefficient of determination* is

$$R^2(\boldsymbol{\theta}) = 1 - \frac{\sum_{i=1}^n (y_{\text{data},i} - y_{\text{model},i}(\boldsymbol{\theta}))^2}{\sum_{i=1}^n (y_{\text{data},i} - \bar{y}_{\text{model}}(\boldsymbol{\theta}))^2},$$

where  $\bar{y}_{\text{model}}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n y_{\text{model},i}(\boldsymbol{\theta})$  is the mean of the model predictions.

**The  $\chi^2$  function.**

Normally, the response (dependent or outcome) variable  $y_i$  is the outcome of a numerical experiment or another type of experiment and is thus only an approximation to the true value. It is then always accompanied by an error estimate, often limited to a statistical error estimate given by the standard deviation discussed earlier. In the discussion here we will treat  $y_i$  as our exact value for the response variable.

Introducing the standard deviation  $\sigma_i$  for each measurement  $y_i$ , we define now the  $\chi^2$  function (omitting the  $1/n$  term) as

$$\chi^2(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=0}^{n-1} \frac{(y_i - \tilde{y}_i)^2}{\sigma_i^2} = \frac{1}{n} \left\{ (\mathbf{y} - \tilde{\mathbf{y}})^T \frac{1}{\boldsymbol{\Sigma}^2} (\mathbf{y} - \tilde{\mathbf{y}}) \right\},$$

where the matrix  $\boldsymbol{\Sigma}$  is a diagonal matrix with  $\sigma_i$  as matrix elements.

In order to find the parameters  $\theta_i$  we will then minimize the spread of  $\chi^2(\boldsymbol{\theta})$  by requiring

$$\frac{\partial \chi^2(\boldsymbol{\theta})}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} \left[ \frac{1}{n} \sum_{i=0}^{n-1} \left( \frac{y_i - \theta_0 x_{i,0} - \theta_1 x_{i,1} - \theta_2 x_{i,2} - \cdots - \theta_{n-1} x_{i,n-1}}{\sigma_i} \right)^2 \right] = 0,$$

which results in

$$\frac{\partial \chi^2(\boldsymbol{\theta})}{\partial \theta_j} = -\frac{2}{n} \left[ \sum_{i=0}^{n-1} \frac{x_{ij}}{\sigma_i} \left( \frac{y_i - \theta_0 x_{i,0} - \theta_1 x_{i,1} - \theta_2 x_{i,2} - \cdots - \theta_{n-1} x_{i,n-1}}{\sigma_i} \right) \right] = 0,$$

or in a matrix-vector form as

$$\frac{\partial \chi^2(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0 = \mathbf{A}^T (\mathbf{b} - \mathbf{A}\boldsymbol{\theta}).$$

where we have defined the matrix  $\mathbf{A} = \mathbf{X}/\boldsymbol{\Sigma}$  with matrix elements  $a_{ij} = x_{ij}/\sigma_i$  and the vector  $\mathbf{b}$  with elements  $b_i = y_i/\sigma_i$ .

We can rewrite

$$\frac{\partial \chi^2(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0 = \mathbf{A}^T (\mathbf{b} - \mathbf{A}\boldsymbol{\theta}),$$

as

$$\mathbf{A}^T \mathbf{b} = \mathbf{A}^T \mathbf{A} \boldsymbol{\theta},$$

and if the matrix  $\mathbf{A}^T \mathbf{A}$  is invertible we have the solution

$$\boldsymbol{\theta} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}.$$

If we then introduce the matrix

$$\mathbf{H} = (\mathbf{A}^T \mathbf{A})^{-1},$$

we have then the following expression for the parameters  $\theta_j$  (the matrix elements of  $\mathbf{H}$  are  $h_{ij}$ )

$$\theta_j = \sum_{k=0}^{p-1} h_{jk} \sum_{i=0}^{n-1} \frac{y_i}{\sigma_i} \frac{x_{ik}}{\sigma_i} = \sum_{k=0}^{p-1} h_{jk} \sum_{i=0}^{n-1} b_i a_{ik}$$

We state without proof the expression for the uncertainty in the parameters  $\theta_j$  as (we leave this as an exercise)

$$\sigma^2(\theta_j) = \sum_{i=0}^{n-1} \sigma_i^2 \left( \frac{\partial \theta_j}{\partial y_i} \right)^2,$$

resulting in

$$\sigma^2(\theta_j) = \left( \sum_{k=0}^{p-1} h_{jk} \sum_{i=0}^{n-1} a_{ik} \right) \left( \sum_{l=0}^{p-1} h_{jl} \sum_{m=0}^{n-1} a_{ml} \right) = h_{jj}!$$

The first step here is to approximate the function  $y$  with a first-order polynomial, that is we write

$$y = y(x) \rightarrow y(x_i) \approx \theta_0 + \theta_1 x_i.$$

By computing the derivatives of  $\chi^2$  with respect to  $\theta_0$  and  $\theta_1$  show that these are given by

$$\frac{\partial \chi^2(\boldsymbol{\theta})}{\partial \theta_0} = -2 \left[ \frac{1}{n} \sum_{i=0}^{n-1} \left( \frac{y_i - \theta_0 - \theta_1 x_i}{\sigma_i^2} \right) \right] = 0,$$

and

$$\frac{\partial \chi^2(\boldsymbol{\theta})}{\partial \theta_1} = -\frac{2}{n} \left[ \sum_{i=0}^{n-1} x_i \left( \frac{y_i - \theta_0 - \theta_1 x_i}{\sigma_i^2} \right) \right] = 0.$$