University of Illinois Fall 2018

ECE 544NA: Pattern Recognition
Lecture 2: August 30

Lecturer: Alexander Schwing Scribe: Yundi Fei

1 Overview

This document scribes Lecture 2: Linear Regression. The lecture aims to achieve the following goals:

- Getting to know linear regression
- Understanding how linear regression works
- Examples for linear regression

Section 2 introduces the baseline linear regression. Section 3 introduces 3 extensions to the baseline linear: higher dimensional problems, regularization, and higher order polynomials. Section 4 introduces the probabilistic interpretation of linear regression. Section 5 introduces linear regression for classification and its issue. Section 6 introduces the applications of linear regression. Section 7 answers the questions in quiz section on the slides.

2 Baseline Linear Regression

2.1 The Problem

Linear regression is a linear approach to the problem of finding the underlying system/model/model parameters given outcomes $y^{(i)} \in \mathbb{R}$ for covariates $x^{(i)} \in \mathbb{R}$. To put it in a more mathematical way, we first assume a linear model with parameters $w_1 \in \mathbb{R}$ and $w_2 \in \mathbb{R}$.

$$y = w_1 \cdot x + w_2 \tag{1}$$

The problem described in mathematical terms is then to find the parameters w_1, w_2 of model $y = w_1 \cdot x + w_2$, given a dataset of N pairs $(x; y)$, noted as $\mathcal{D} = \{(x^{(i)}; y^{(i)})\}_{i=1}^{N}$

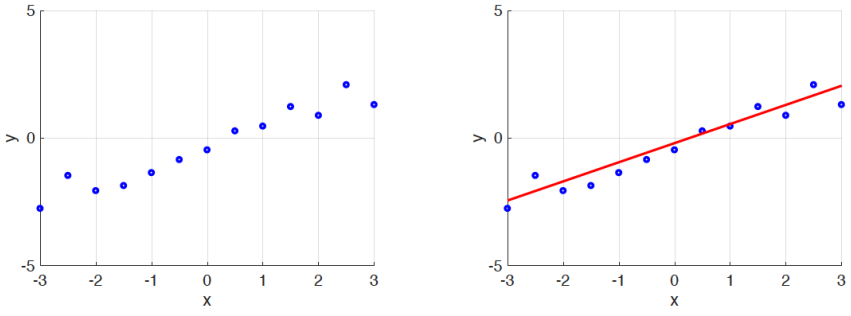

Figure 1: Illustration of Linear Regression

2.2 The Solution

2.2.1 The Program

The solution is to find parameters w_1, w_2 assuming model $y = w_1 \cdot x + w_2$, such that the squared error is small:

$$\arg\min_{w_1,w_2} \frac{1}{2} \sum_{i=1}^{N} \left(y^{(i)} - w_1 \cdot x^{(i)} - w_2 \right)^2 \tag{2}$$

Solving the program is to find the values for w_1, w_2 that minimize the value of cost/loss function. The difference between min and arg min is that min gives the smallest value and arg min gives the value of parameters that minimizes the value. $\frac{1}{2}$ is usually used for math convenience when we have squared error. The error $y^{(i)} - w_1 \cdot x^{(i)} - w_2$ is basically the vertical differentce between real value and predicted value shown in Figure 2.

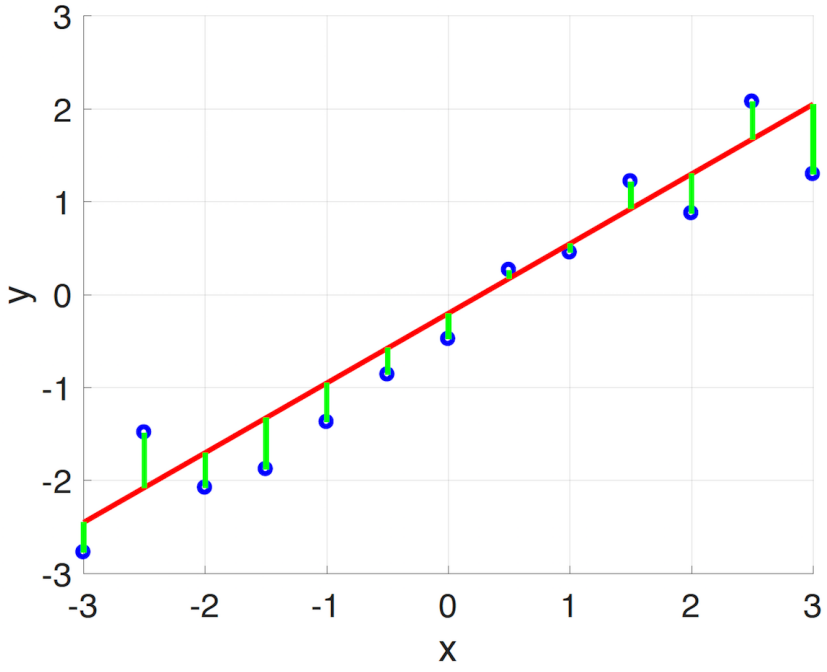

Figure 2: Error of Linear Regression Model

To simplify the notation of the program, we can use the following vector notation

$$\arg\min_{w_1,w_2} \frac{1}{2} \left\| \underbrace{\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}}_{\mathbf{Y} \in \mathbb{R}^N} - \underbrace{\begin{bmatrix} x^{(1)} & 1 \\ \vdots & \vdots \\ x^{(N)} & 1 \end{bmatrix}}_{\mathbf{X}^\top \in \mathbb{R}^{N \times 2}} \cdot \underbrace{\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}}_{\mathbf{w} \in \mathbb{R}^2} \right\|_2^2 \tag{3}$$

which can be furture simplified as

$$\arg\min_{\mathbf{w}} \underbrace{\frac{1}{2} \|\mathbf{Y} - \mathbf{X}^\top \mathbf{w}\|_2^2}_{\text{cost/loss function}} \tag{4}$$

2.2.2 Solving the Program

To solve the program, we take 3 steps:

1. Take derivative with regard to $\mathbf{w}$ of cost function
2. Set derivative with regard to $\mathbf{w}$ to zero
3. Solve for $\mathbf{w}$

After taking the derivative and setting derivative to zero, we have the equation:

$$\mathbf{X}\mathbf{X}^\top \mathbf{w}^* - \mathbf{X}\mathbf{Y} = 0 \tag{5}$$

Solving for equation 2 gives:

$$\mathbf{w}^* = \left(\mathbf{X}\mathbf{X}^\top\right)^{-1}\mathbf{X}\mathbf{Y} \tag{6}$$

3 Extension to Linear Regression

This lecture discusses 3 major extensions: higher dimensional problems ($\mathbf{x}^{(i)} \in \mathbb{R}^d$), regularization, and higher order polynomials. Each extensions are introduced with detail in the following subsections. After introducing all three extensions separately, the lecture also gives a combinational case.

3.1 Higher dimensional problems ($\mathbf{x}^{(i)} \in \mathbb{R}^d$, $y^{(i)} \in \mathbb{R}$)

The first extension is to extend baseline 1-dimensional features to $\mathbf{x}^{(i)}$ of d dimensions. The model is instead

$$y^{(i)} = w_0 + \sum_{k=1}^{d} \mathbf{x}_k^{(i)} w_k \tag{7}$$

To fit the linear regression model, program is

$$\arg\min_{\mathbf{w}} \frac{1}{2} \| \underbrace{\mathbf{Y}}_{\in \mathbb{R}^N} - \underbrace{\mathbf{X}^\top}_{\in \mathbb{R}^{N\times(d+1)}} \underbrace{\mathbf{w}}_{\in \mathbb{R}^{d+1}} \|_2^2 \tag{8}$$

Solution is

$$\mathbf{w}^* = \left(\mathbf{X}\mathbf{X}^\top\right)^{-1}\mathbf{X}\mathbf{Y} \tag{9}$$

The program and solution are same in notation compared to 1-d case. The only difference is that the dimension of $\mathbf{X}$ is generalized to be $N \times (d+1)$ instead of $N \times 2$ and the dimension of $\mathbf{w}$ is generalized to be $d+1$ instead of 2.

The Figure 3 shows a 2D example. The data points have 2-dimensional features x_1 and x_2. The model fitted to these data points is then a plane in 3 dimensional space.

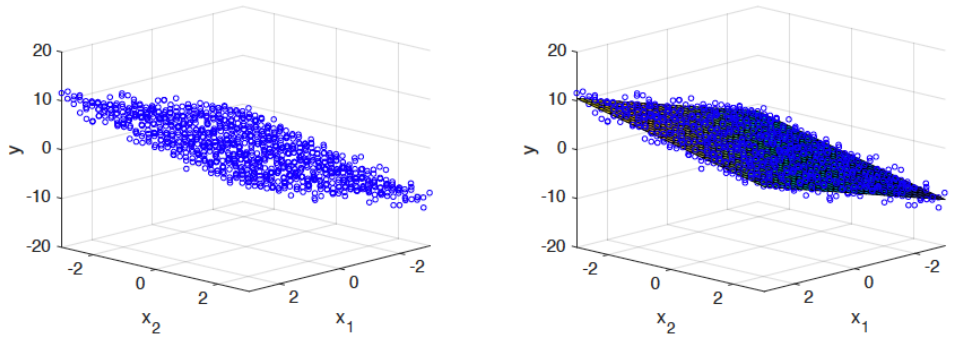

Figure 3: Example of higher dimensional (2D)

3.2 Regularization

A problem with this system is that it requires $(\mathbf{X}\mathbf{X}^\top)$ to be invertible. It's invertible if it has full rank and $N \geq d+1$. When $N < d+1$, which means we don't have enough unique data points, we can handle with regularization. Regularization help to make sure the parameters are not too large and the matrix is invertible.

To regularize, we have program to be:

$$\arg\min_{\mathbf{w}} \underbrace{\frac{1}{2}\|\mathbf{Y} - \mathbf{X}^\top\mathbf{w}\|_2^2 + \frac{C}{2}\|\mathbf{w}\|_2^2}_{\text{cost/loss function}} \tag{10}$$

C is the regularization constance/hyper parameter. If C is infinity, $\mathbf{w}$ is 0. We care more about $\mathbf{w}$ as C decreases.

Solution is then:

$$\mathbf{w}^* = \left(\mathbf{X}\mathbf{X}^\top + C\mathbf{I}\right)^{-1}\mathbf{X}\mathbf{Y} \tag{11}$$

Adding C solves the problem. If matrix $\mathbf{X}\mathbf{X}^\top$ is not invertible, its smallest eigenvalue is zero, then $\mathbf{X}\mathbf{X}^\top + C\mathbf{I}$'s smallest eigenvalue is some positive value. Therefore, $\mathbf{X}\mathbf{X}^\top + C\mathbf{I}$ is invertible.

3.3 Higher order polynomials ($x^{(i)} \in \mathbb{R}$, $y^{(i)} \in \mathbb{R}$)

We can also face higher order polynomials. For example, the model for order of 2 is

$$y^{(i)} = w_2 \cdot (x^{(i)})^2 + w_1 \cdot x^{(i)} + w_0 \tag{12}$$

Program for the case of order = 2 is

$$\arg\min_{w_0,w_1,w_2} \frac{1}{2}\left\| \underbrace{\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}}_{\mathbf{Y}\in\mathbb{R}^N} - \underbrace{\begin{bmatrix} (x^{(1)})^2 & x^{(1)} & 1 \\ \vdots & \vdots & \vdots \\ (x^{(N)})^2 & x^{(N)} & 1 \end{bmatrix}}_{\Phi^\top\in\mathbb{R}^{N\times M}} \cdot \underbrace{\begin{bmatrix} w_2 \\ w_1 \\ w_0 \end{bmatrix}}_{\mathbf{w}\in\mathbb{R}^M} \right\|_2^2 \tag{13}$$

Note that M = order +1, so that it's 3 in the case of order = 2.

Solution is then

$$\mathbf{w}^* = \left(\Phi\Phi^\top\right)^{-1}\Phi\mathbf{Y} \tag{14}$$

We need to be careful about using higher order model to fit the given data points. In the case shown in Figure 4, the higher order model is clearly overfitting compared to the simple line.

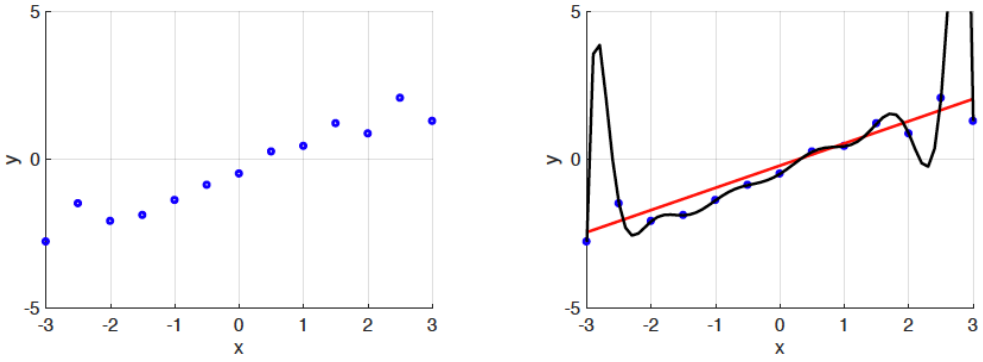

Figure 4: Example of overfiiting

3.4 Combination of higher dimensional problems and higher order polynomials

When combing higher dimensional and higher order, we have:

- $x^{(i)}$ is some data (e.g., images)
- $\phi(x^{(i)}) \in \mathbb{R}^M$ is a transformation into a feature vector

Combined model is:

$$y^{(i)} = \phi(x^{(i)})^\top \mathbf{w} \tag{15}$$

Therefore, we have program:

$$\arg\min_{\mathbf{w}} \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - \phi(x^{(i)})^\top\mathbf{w}\right)^2 \tag{16}$$

Solving the program gives the solution:

$$\mathbf{w}^* = \left(\Phi\Phi^\top\right)^{-1}\Phi\mathbf{Y} \text{ where } \Phi = [\phi(x^{(1)}), \cdots, \phi(x^{(N)})] \in \mathbb{R}^{M\times N} \tag{17}$$

4 Probabilistic Interpretation of Linear Regression

Previous two sections approaches linear regression in error view of $(y^{(i)} - \phi(x^{(i)})^\top\mathbf{w})^2$. Alternatively, we can have a probabilistic view of linear regression as illustrated in Figure 5. We take the model of Gaussian distribution:

$$p(y^{(i)}|x^{(i)}) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\big(-\frac{1}{2\sigma^2}(y^{(i)} - \phi(x^{(i)})^\top\mathbf{w})^2\big) \tag{18}$$

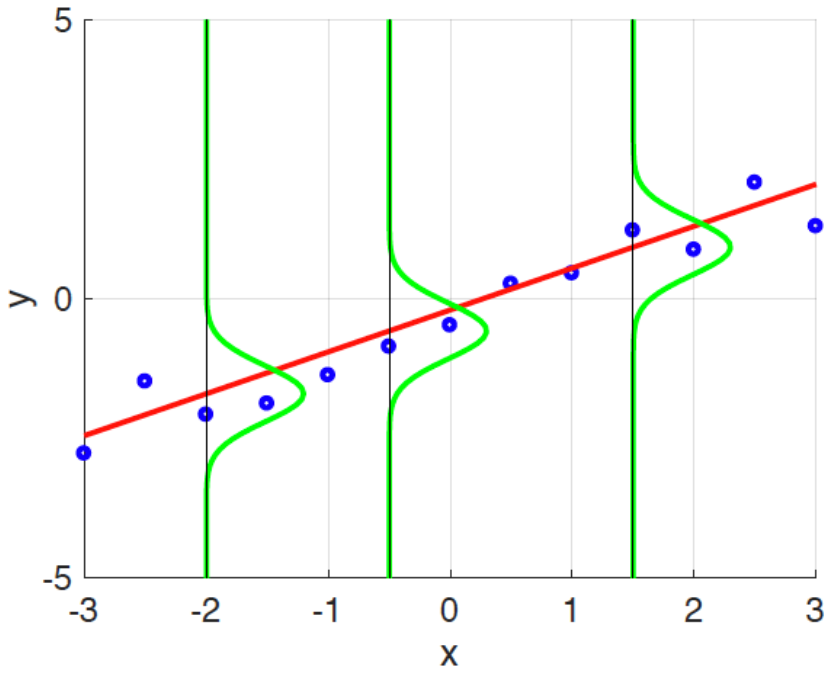

Figure 5: Gaussian distribution probabilistic view of linear regression

We need to maximize likelihood of given dataset $\mathcal{D} = \{(x^{(i)}; y^{(i)}\}$ assuming samples to be drawn independently from an identical distribution (i.i.d.). The likelihood is

$$p(\mathcal{D}) = \prod_{(x^{(i)},y^{(i)})\in\mathcal{D}} p(y^{(i)}|x^{(i)}) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\Big(-\frac{1}{2\sigma^2}\|\mathbf{Y} - \Phi^\top \mathbf{w}\|_2^2\Big) \tag{19}$$

To maximize the likelihood, we solve for:

$$\arg\max_{\mathbf{w}} p(\mathcal{D}) = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{Y} - \Phi^\top \mathbf{w}\|_2^2 \tag{20}$$

When we find arg max, we can take out constant/independent terms $\frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}}$ and $\frac{1}{2\sigma^2}$. We can also take out exp as it is monotonic function. We can remove $-$ and change from arg max to arg min

5 Linear Regression for Classification

5.1 Linear regression for classification model and examples

If we want to use linear regression for classification, we have $y^{(i)} \in \{-1, 1\}$. Our model is then $y = w_1 x + w_0$ threshold at $y = 0$.

5.1.1 Perfect classification

Figure 6 is an example of perfect classification, as all blue data points are classified as -1, and all black data points are classified as 1.

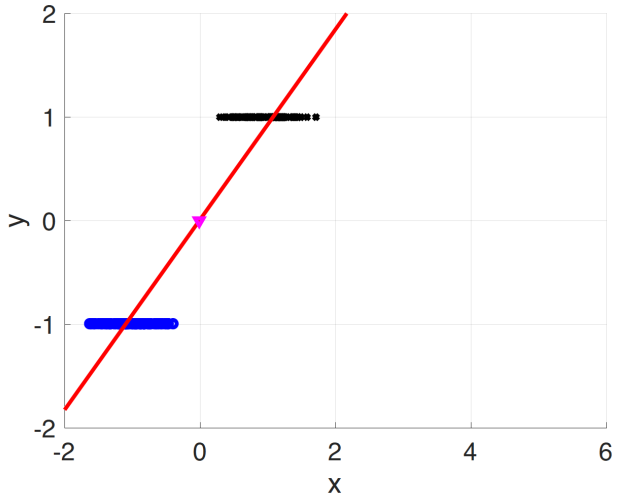

Figure 6: perfect classification

5.1.2 Imperfect classification

Adding more data on the far right leads to a different linear regression model, which does not give perfect classification as shown in Figure 7. In order to have the line with minimum squared error, the slope of the line change to fit additional black data points on the far right. With the change, the decision boundary represented as pink triangle shifts to the right and some black data points are now classified as -1 instead of 1.

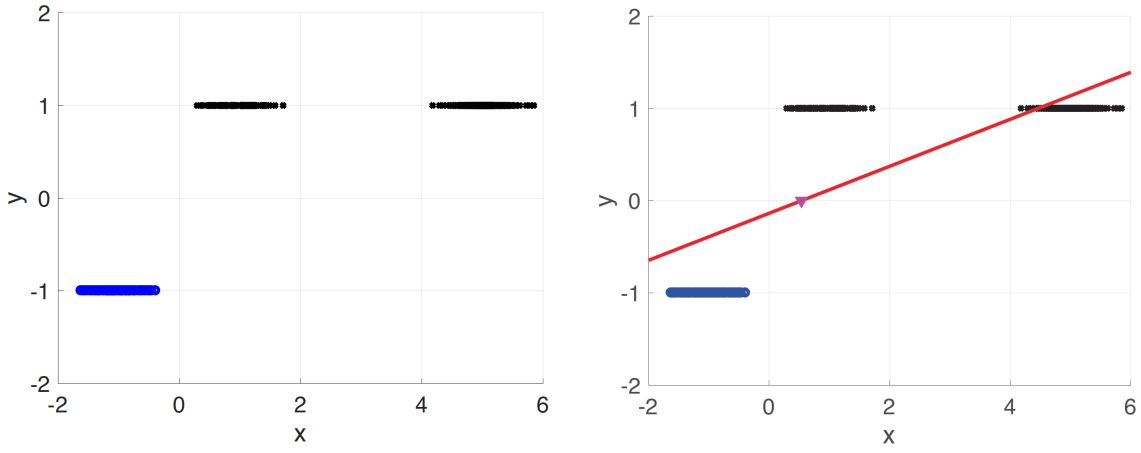

Figure 7: data points leading to a imperfect classification

5.1.3 Analysis: quadratic loss

The reason behind the shifting of decision boundary is the quadratic loss in linear regression model.Recall that $y^{(i)} \in \{-1, 1\}$, the loss function $\ell(y^{(i)}, \phi(x^{(i)})^\top \mathbf{w})$ is:

$$\begin{aligned}
\ell(y^{(i)}, \phi(x^{(i)})^\top \mathbf{w}) &= \frac{1}{2}(y^{(i)} - \phi(x^{(i)})^\top \mathbf{w})^2 \\
&= \frac{1}{2}(y^{(i)} y^{(i)} - y^{(i)} \phi(x^{(i)})^\top \mathbf{w})^2 \\
&= \frac{1}{2}(1 - y^{(i)} \underbrace{\underbrace{\phi(x^{(i)})^\top \mathbf{w}}_{F(x^{(i)}, \mathbf{w})}}_{F(x^{(i)}, \mathbf{w}, y^{(i)})})^2 \quad \text{because } (y^{(i)})^2 = 1
\end{aligned} \tag{21}$$

In the above equation, $F(x^{(i)}, \mathbf{w})$ computes the classification result, and $F(x^{(i)}, \mathbf{w}, y^{(i)})$ is the scoring function. The classification result is 1 if $F(x^{(i)}, \mathbf{w}) > 0$, and is -1 otherwise. Therefore, if the classification result matches actual result $y^{(i)}$, the scoring function gives positive; otherwise, the scoring function gives negative.

We can plot the loss function shown as Figure 8.

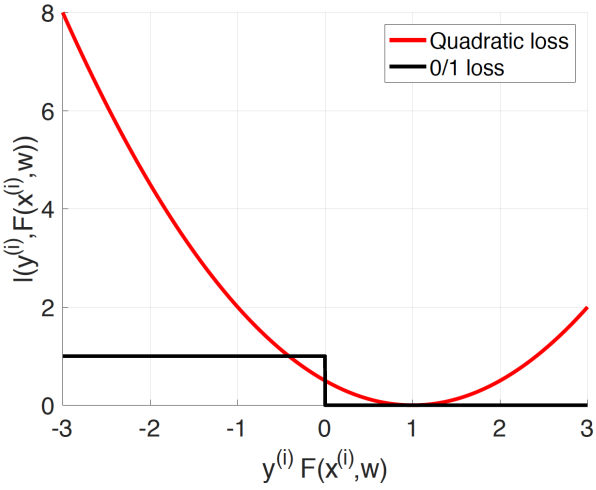

Figure 8: quadratic loss

The problem of quadratic loss is that the cost function penalizes samples that are 'very easy to classify'. Samples are said to be 'very easy to classify' if the absolute value of the scoring function is large. Consider 2 cases:

- Case 1: $F(x^{(i)}) = 1$ and $y^{(i)} = 1$. The loss is $\frac{1}{2}(1 - y^{(i)} F(x^{(i)}, \mathbf{w})^2 = \frac{1}{2}(1-1)^2 = 0$
- Case 2: $F(x^{(i)}) = 20$ and $y^{(i)} = 1$. The loss is $\frac{1}{2}(1 - y^{(i)} F(x^{(i)}, \mathbf{w})^2 = \frac{1}{2}(1-20)^2 = 180.5$

While both data points are classified correctly, data point in case 2 is easier to be classified yet penalized much more heavily.

6 Applications of Linear Regression

Some examples of applications are:

1. stock trading
2. sports betting
3. flight time prediction

7 Quiz Answers

1. Linear regression optimizes what cost function?
   Cost function for basic linear regression model:

   $$\frac{1}{2}\|\mathbf{Y} - \mathbf{X}^\top \mathbf{w}\|_2^2 \tag{22}$$

   For higher dimensional, regularization, and higher order polynomials, see the part after argmin in equation 8, 10, and 13.

2. How can we optimize this cost function?
   To optimize this cost function, we solve the program

   $$\arg\min_{\mathbf{w}} \underbrace{\frac{1}{2}\|\mathbf{Y} - \mathbf{X}^\top \mathbf{w}\|_2^2}_{\text{cost/loss function}} \tag{23}$$

   To solve the program, we take 3 steps:

   (a) Take derivative with regard to $\mathbf{w}$ of cost function
   (b) Set derivative with regard to $\mathbf{w}$ to zero
   (c) Solve for $\mathbf{w}$

   After taking the derivative and setting derivative to zero, we have $\mathbf{X}\mathbf{X}^\top \mathbf{w}^* - \mathbf{X}\mathbf{Y} = 0$. Solving for equation 2 gives: $\mathbf{w}^* = \left(\mathbf{X}\mathbf{X}^\top\right)^{-1}\mathbf{X}\mathbf{Y}$

3. What are issues of linear regression applied to classification?
   Quadratic loss used in the linear regression results in decision boundary shifting and penalizes samples that are very easy to classify. Read details about it in Section 5.

References