**Simple Linear Regression**

Table of Contents

[Notation 2](#_Toc100149487)

[Line of Best Fit 3](#_Toc100149488)

[Model Parameters 4](#_Toc100149489)

Consider that we have a graph for house prices based on size.

If we want to predict the price of a house of a particular size, we can use the **best fit line**, as shown above. This is an example of a **supervised** learning algorithm for a **regression** problem. The machine learning algorithm we will create will basically find the line of best fit.

## Notation

Throughout this course, we will be defining **data points** as , for the th data point. Thus, for this example, the s are the sizes and the s are the prices. The complete list of data points forms the **dataset** or the **training set** and consists of data points.

refers to the **output** for a particular data point. Since we will be dealing with **unstructured data**, will always be a single value. This is not true for , which could be a vector. For example, the price of a house could depend on multiple factors, such as the size, number of rooms, location, etc. This would cause to be a vector. Each element of the vector would be denoted as , , etc. We will not be dealing with these just yet though.

The algorithm we create will also give a particular output at a particular data point, . This is different from the actual output .

## Line of Best Fit

To find the line of best fit, we use the following equation:

For each data point, we have the **real output**, , and the **predicted output**, . The difference between these two values gives us the **error** for that data point. If we take the **sum** of the errors for each data point and divide it by , the total number of data points, we can find the **average error**. The reason we **square** the errors is just to get rid of negative values. This is called the **mean square error**. Our goal is to find the **minimum** possible value.

Now, why are we using instead of just ? Well, to find the minimum value, as we will later see, we will have to differentiate this equation. The extra will make things easier then. Additionally, it has no effect on our end goal, since the minimum value will still be the minimum value, even if we divide each value by .

Why the average though? Why not just directly take the sum? That should work too after all. The issue there is, if we have a line which fits some data points perfectly but gives a huge error for other data points, we might end up with a line which is poorly fitted but ends up giving the smallest sum. By averaging the value, we are taking this into account.

## Model Parameters

The equation for the line of best fit will have the format . Here, we will be using the notation , called the **hypothesis**. Thus, we need to find the parameters and which give us the line of best fit. We can consider these parameters to be a vector, . Thus, we can say that we need to find . The output of the machine learning algorithm will be the hypothesis .

and are called the **model parameters** and is called the **weight vector**. Using and , we will have a particular mean square error, . is called the **cost function**, or the criterion function or the error function or the object function.

Finally, we can also say that our end goal is to find . refers to the minimum value of the cost function. gives us the **argument** of the cost function when it is minimum, which is a specific value of . This value is the value of , the **optimum value**.