CS 6140: Machine Learning
College of Computer and Information Science
Northeastern University
Module 2 Leson 1
Instructor: Bilal Ahmed

Scribe: Bilal Ahmed & Virgil Pavlu

Linear Regression[1]

Consider the housing dataset where the objective is to predict the price of a house based on a number of features that include the average number of rooms, living area, pollution levels of the neighborhood, etc. In this example we are going to use a single feature: the average number of rooms to predict the value of the house. Our input is a single number $x \in \mathbb{R}$ and the output (label) is also a continuous number $y \in \mathbb{R}$, and our task is to find a function $f(x) : \mathbb{R} \to \mathbb{R}$ that takes as input the average number of rooms and outputs the value of the house (in tens of thousands of dollars). Here is a snippet of the data:

Average No. of Rooms	3	3	3	2	4	...
Price (10,000 \$)	40.0	33 .0	36.9	23.2	54.0	...

and here is how the data looks like,

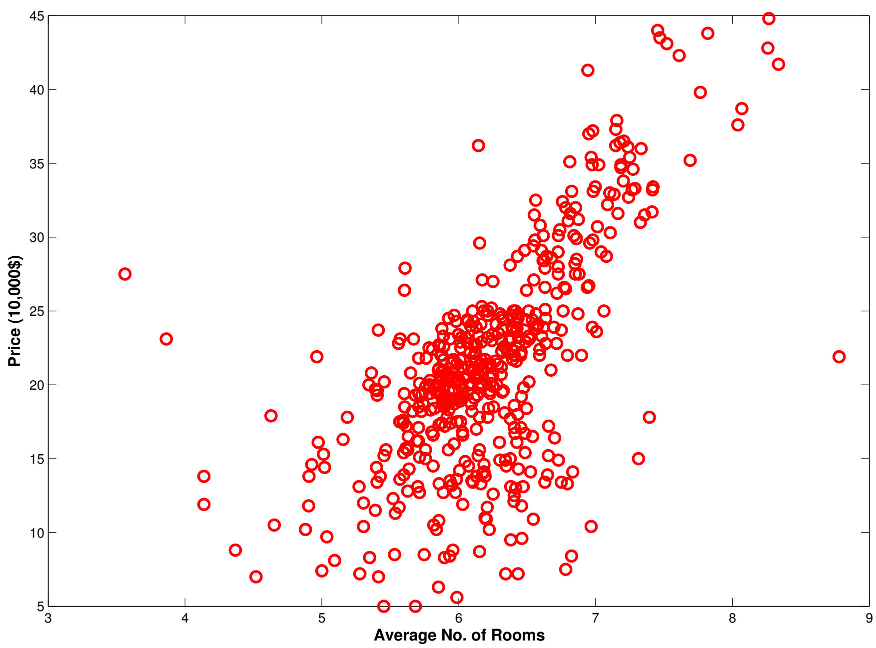

Figure 1: Plot of the average number of rooms (x-axis) and the price of the house (y-axis). Note that the input and the output are not normalized.

[1]Based on lecture notes by Andrew Ng. These lecture notes are intended for in-class use only.

Assume that the function that we are searching for is a straight line. In which case we can represent the function as:

$$y = f(x) = w_0 + w_1 x \tag{1}$$

where w_0 is the y-intercept of the line, and w_1 is the slope. Based on this formulation searching for the "best" line then translates into finding the optimal values of the parameters w_0 and w_1. It should be noted here that for a given training dataset having N observations, the x_i and y_i are fixed and we need to find the values of the parameters that satisfy the N equations:

$$\begin{aligned} y_1 =& w_0 + w_1 x_1 \\ y_2 =& w_0 + w_1 x_2 \\ & \dots \\ y_N =& w_0 + w_1 x_N \end{aligned}$$

If $N > 2$, then this system of equations is overdetermined and has no solution. Instead of looking for an exact solution that satisfies the N equations, we will search for an approximate solution, that satisfies these equations with some error. In order to find the approximate solution we need a way to decide the "goodness" of a given line (specific values for w_0 and w_1). For example, consider Figure-2, we have two candidate lines l_1 and l_2, which one should we choose? In other words how can we say that a given line is the optimal line with respect to the criterion we have defined for the approximate satisfiability of the set of equations given above?

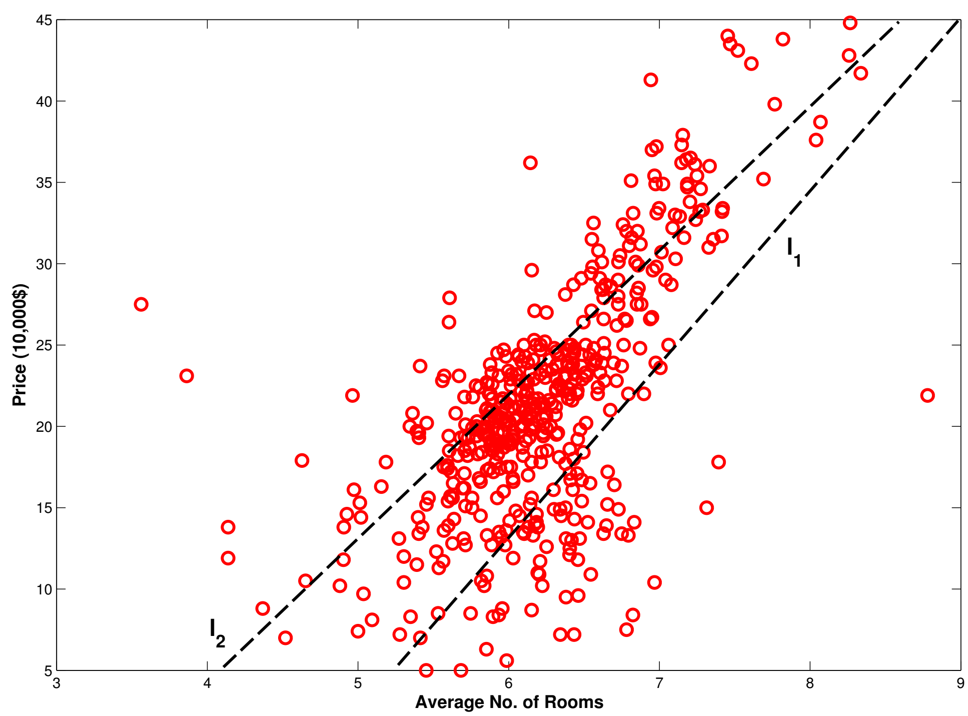

Figure 2: Two candidate lines, that can be used to predict the price of the house based on the average number of rooms. Which line is better?

1 Setup and Notation

So far in the example we have dealt with a single input feature and in most real-world cases we would have multiple features that define each instance. We are going to assume that our data is arranged in a matrix with each row representing an instance, and the columns representing individual features. We can visualize the data matrix as:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{Nm} \end{bmatrix}$$

where, X is also known as the "design matrix". There are N labels corresponding to each instance $\mathbf{x_i} \in \mathbb{R}^m$, arranged as a vector $\mathbf{y} \in \mathbb{R}^N$ as:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

the training data $\mathcal{D}$ can be described as consisting of $(\mathbf{x_t}, y_t)$; $\forall t \in \{1, 2, \dots, N\}$.

The regression function that we want to learn from the data can then be described analogously to Equation-1 as:

$$y = f_w(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \dots + +w_m x_m \tag{2}$$

where, $f(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}$ and $\mathbf{w} = [w_0, w_1, w_2, \dots, w_m]$ are the parameters of the regression function. The learning task is then to find the "optimal" weight vector $\mathbf{w} \in \mathbb{R}^{m+1}$, based on the given training data $\mathcal{D}$. When there is no risk of confusion, we will drop w from the f notation, and we will assume a dummy feature $x^0 = 1$ for all instances such that we can re-write the regression function as:

$$f(\mathbf{x}) = w_0 x_0 + w_1 x_1 + w_2 x_2 + \dots + w_m x_m = \sum_{i=1}^{m} w_i x_i = \mathbf{w}^T \mathbf{x} \tag{3}$$

Note: In the above equation x_i is the i^{th} feature, and to incorporate the augmented constant feature (which is always equal to one) we can augment the design matrix X with a column of 1s (as the first column) if $\mathbf{w} = [w_0, w_1, w_2, \dots, w_m]$.

Next, we need to define a criterion for assessing the "goodness" of fit to the training data for a given weight vector. Once the criterion is defined we can then minimize it to find the optimal set of weights giving us the best fit to the training data.