

Calculus in AI/Machine Learning

Qiyam Tung

November 10, 2011

1 Linear Regression

(Note this lesson borrowed slides and lessons from Andrew Ng's free machine learning class of 2011).

Artificial intelligence is a huge field with a very interesting history. Like any history, it has many roots, but one of them came from metamathematics. It is therefore not terribly surprising that many learning algorithms require a good understanding of mathematics. Not all topics are accessible to a high school calculus class, but we can at least discuss one topic with significant detail: classification.

Regression is a learning method of being able to predict the output given the input. Here are a few examples of linear regression.

- Predicting house prices as a function of its size (e.g. figure 1).
- Predicting consumption spending (a large number of input variables).
- Predicting the effect of factories of environments.

2 Lines Again

We have previously seen that lines can help us distinguish between two classes of objects. I used a contrived example of trying to distinguish between boys and girls based on weight and height. In the example, we saw how a line could distinguish between a boy and a girl. In our case it was obvious where the line should roughly be. However, there are problems with this approach

- You manually picked the line.
- The way of picking the line does not have a specific criteria.

Manually picking the line defeats the whole purpose of machine learning and without a criteria you cannot make this process automatic. In this lesson, we will try to learn how to solve both these problems.

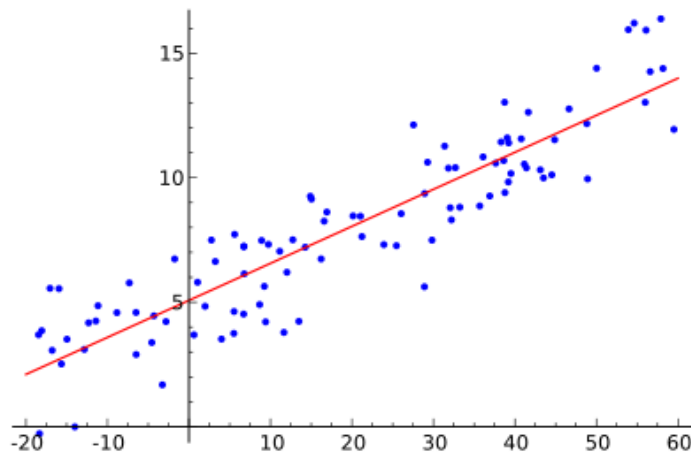


Figure 1: Regression is an extremely useful and important part of today's technology.

2.1 Estimating Lines

First, we need to brush up on lines. The most common form of a line equation is

$$y = mx + b \quad (1)$$

m and b are what we call the *parameters* of a line. They are called *parameters* because they decide the nature of the line. m is the slope, which changes the how slanted the line is and b is the y-intercept, which also determines the “height” of the line. Now, using the letters m and b are fine, but this gets confusing when your equation becomes more complicated, such as a quadratic:

$$y = ax^2 + bx + c \quad (2)$$

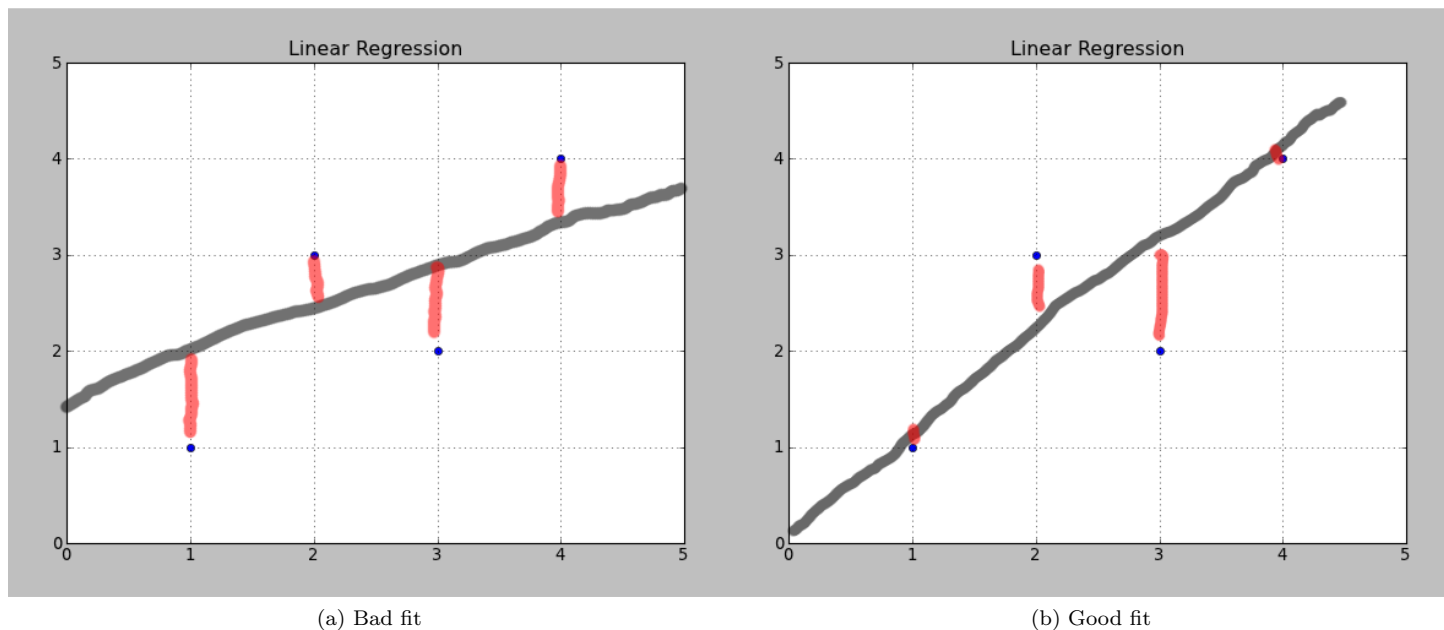


Figure 2: When we fit a line, we want to minimize the average distance of the points to the line we're fitting. The distance is indicated by the vertical lines from the point to the line.

Here, b means the coefficient for the linear term. c is now used to denote the y-intercept. To keep things consistent, then, we want to use symbols that stay consistent whether we're talking about linear, quadratic, cubic, or any kind of equation. We use the symbols θ to denote the parameter. For example, the linear equation now looks like

$$h_{\theta}(x) = \theta_0 + \theta_1 x \quad (3)$$

And a quadratic now becomes

$$h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \quad (4)$$

Now we see that θ_0 means the y-intercept and θ_1 is the coefficient for the linear term for both equations. With that in mind, let's do some practice.

2.2 Finding the parameters

For each problem, find the parameters for the line given a set of points. If the line doesn't exist, explain why not.

1. (1,1) (2,2)
2. (2,2) (4, 10)
3. (1,7)
4. (1,1) (2,3) (3,2) (4,4)

2.3 Fitting to Noisy Data

For problem 4, you may have noticed that there is no one line that goes through all three points. However, it is possible to find a line that *best fits* the points, or best fits the data. That is the idea behind linear regression.

Suppose you are given the points (1,1) (2,3) (3,2) (4,4) again (figure 2). It is easy to estimate the line by hand (the best fit is $y = x$). However, how would you deal with it when the lines are more noisy like in figure 1?

Here's the idea: we want to choose the parameters θ_0 and θ_1 such that $h_{\theta}(x)$ is as close to y as possible. Note that $h_{\theta}(x)$ is our line equation *estimate* whereas y is the value we have. With our best fit, our line predicts that $h_{\theta}(3)$ is 3 when our data shows that it is actually 2. In other words what we want to do is minimize the error

$$Error = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad (5)$$

What this says, in some sense, is to find the line the minimizes the average distance (or error) of all points to the estimated line. Here is a breakdown of the math and their equivalent English terms

- Average: $\frac{1}{m}$
- “Distance” or mean-squared-error: $(h_{\theta}(x^{(i)}) - y^{(i)})^2$
- Input: $x^{(1)}, x^{(2)}, \dots, x^{(m)}$
- Line: $h_{\theta}(x^{(i)})$
- Output: $y^{(1)}, y^{(2)}, \dots, y^{(m)}$

For our specific example, the equation would be

$$Error = \frac{1}{4} \sum_{i=1}^4 (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad (6)$$

$$Error = \frac{1}{4} ((h_{\theta}(x^{(1)}) - y^{(1)})^2 + (h_{\theta}(x^{(2)}) - y^{(2)})^2 + (h_{\theta}(x^{(3)}) - y^{(3)})^2 + (h_{\theta}(x^{(4)}) - y^{(4)})^2) \quad (7)$$

$$Error = \frac{1}{4} ((\theta_0 + \theta_1 x^{(1)} - y^{(1)})^2 + (\theta_0 + \theta_1 x^{(2)} - y^{(2)})^2 + (\theta_0 + \theta_1 x^{(3)} - y^{(3)})^2 + (\theta_0 + \theta_1 x^{(4)} - y^{(4)})^2) \quad (8)$$

$$Error = \frac{1}{4} ((\theta_0 + \theta_1 - 1)^2 + (\theta_0 + 2\theta_1 - 3)^2 + (\theta_0 + 3\theta_1 - 2)^2 + (\theta_0 + 4\theta_1 - 4)^2) \quad (9)$$

2.4 Practice Problems

Given the points (1,1) (2,3) (3,2) (4,4) , calculate the error $\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$ for the parameters

1. $\theta_0 = 2, \theta_1 = 0$

2. $\theta_0 = 0, \theta_1 = 1$

You should be able to see that the second set of parameters (the optimal set) will give you the smaller error.

3 Finding the Parameters

So, how do we find the line?

What we see in figure 3 is that if we only play with θ_1 (the slope), we see that as we move away from the best fit and decrease the slope, the error gets worse. Likewise, as we increase the slope, the error gets very bad very quickly. If you graph the error as a function of θ_1 , you'll see something like figure 4.

How do you minimize the function? In other words, how do you find the minimum y for this error function?

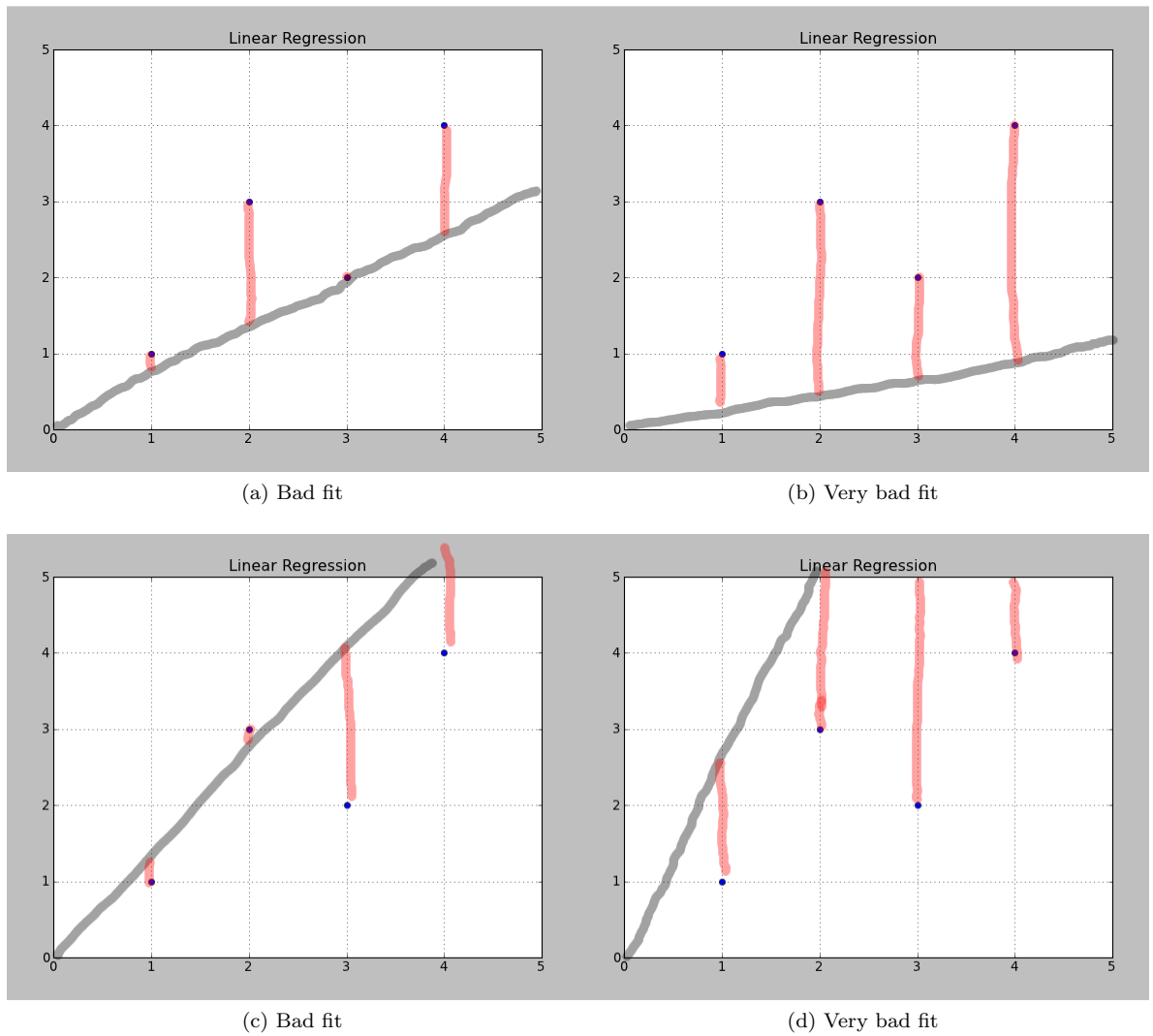


Figure 3: When we fit a line, we want to minimize the average distance of the points to the line we're fitting. The distance is indicated by the vertical lines from the point to the line.

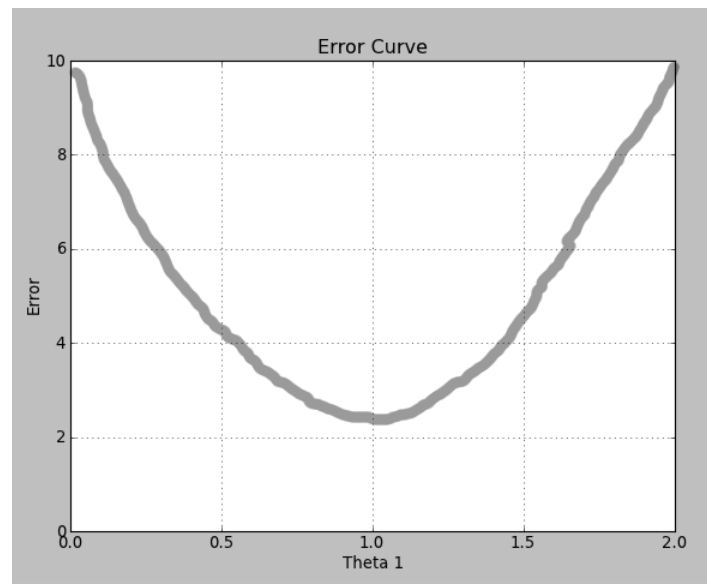


Figure 4: The error plotted as a function of theta.