

[— Linear Regression explained simply —]

 towardsdatascience.com/linear-regression-explained-d0a1068accb9

November 14, 2020

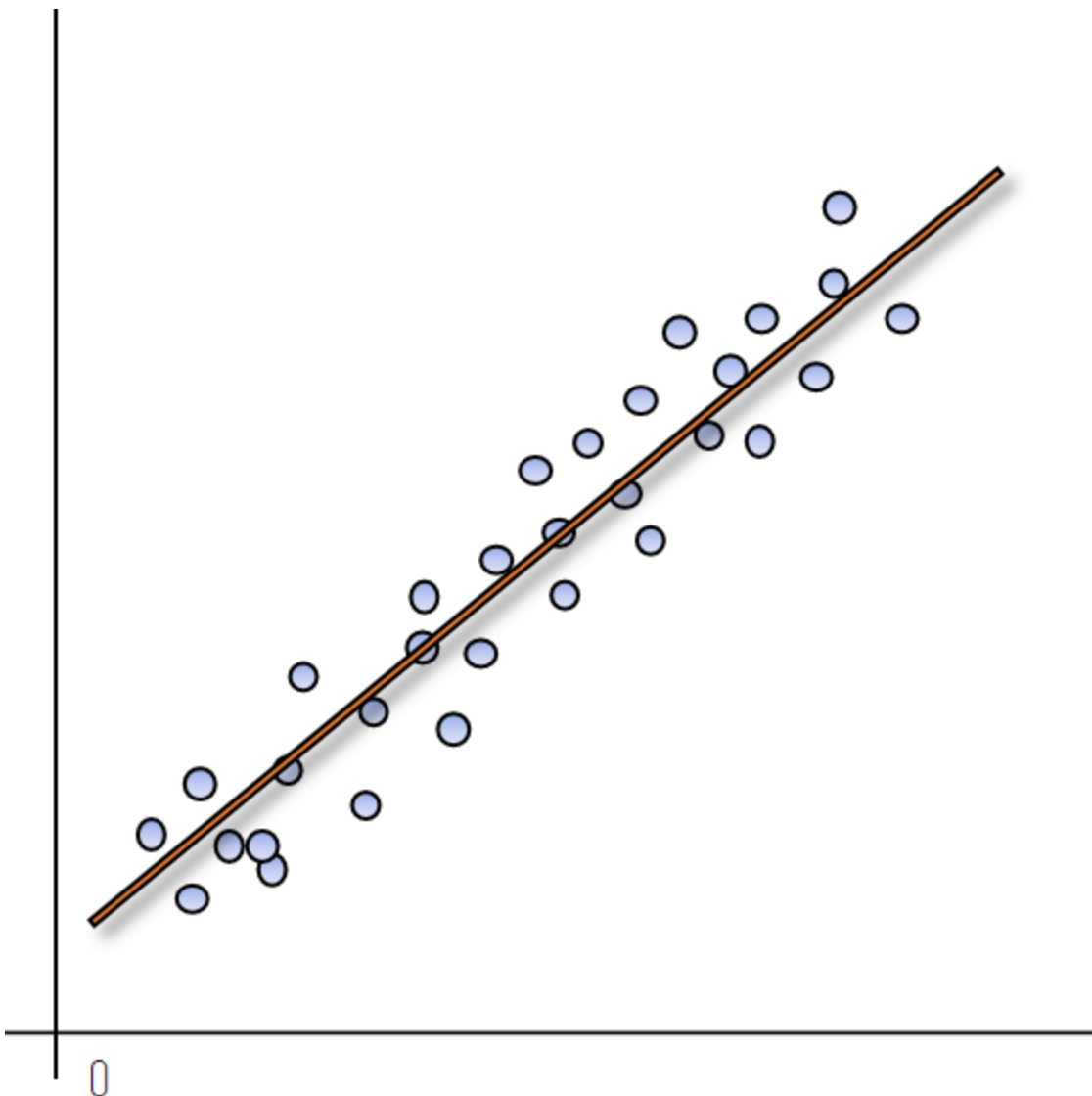
[Sign up for Medium and get an extra one](#)



James Thorn

Jan 26, 2020

Linear Regression Explained



In this post, I will explain Linear Regression in simple terms. It could be considered a Linear Regression for dummies post, however, I've never really liked that expression.

Before we start, here you have some additional resources to skyrocket your Machine Learning career:

Linear Regression in Machine Learning

In the Machine Learning world, **Linear Regression** is a kind of **parametric regression model** that makes a prediction by taking the weighted average of the input features of an observation or data point and adding a constant called *the bias term*.

This means that simple linear regression models are models that have a certain fixed number of parameters that depend on the number of input features, and they output a numeric prediction, like for example the price of a house.

The general formula for linear regression is the following:

$$\hat{y} = \Theta_0 + \Theta_1 x_1 + \dots + \Theta_n x_n$$

Linear regression formula

- is the value we are predicting.
- is the number of features of our data points.
- is the value of the feature.
- are the parameters of the model, where is the bias term. All the other parameters are the weights for the features of our data.

If we wanted to use linear regression to predict the price of a house, using 2 features; the surface of the house in squared meters and the number of bedrooms, the custom formula would look something like this:

$$price = \Theta_0 + \Theta_1 squared_meters + \Theta_2 number_of_bedrooms$$

Linear regression formula for calculating the price of a house with 2 features: the squared meters and the number of bedrooms

Okay, it seems pretty intuitive. Now, how do we calculate the values of that best fit our data? Very easy: Using our data to train the linear regression model.

Just to make sure that we are all on the same point, our training data is labelled data: this is data that contains the objective value that we want to calculate for new data points that don't have this value. In our house price example our training data would consist of a large amount of houses with their price, surface in squared meters, and number of bedrooms.

After we have trained the model, we could use it to predict the price of houses using their squared meters and number of bedrooms.

The steps for training the model are the following:

- 1st we have to choose a metric that tells us how well our model is performing by comparing the predictions made by the model for houses in the training set with their actual prices. These metrics are metrics like the (MSE) or (RMSE).
- We initialise the parameters of the model () to a certain value (usually randomly) and calculate this error for the whole training data.
- We iteratively modify these parameters in order to minimise this error. This is done with algorithms such as Gradient descent, which I will briefly explain now.

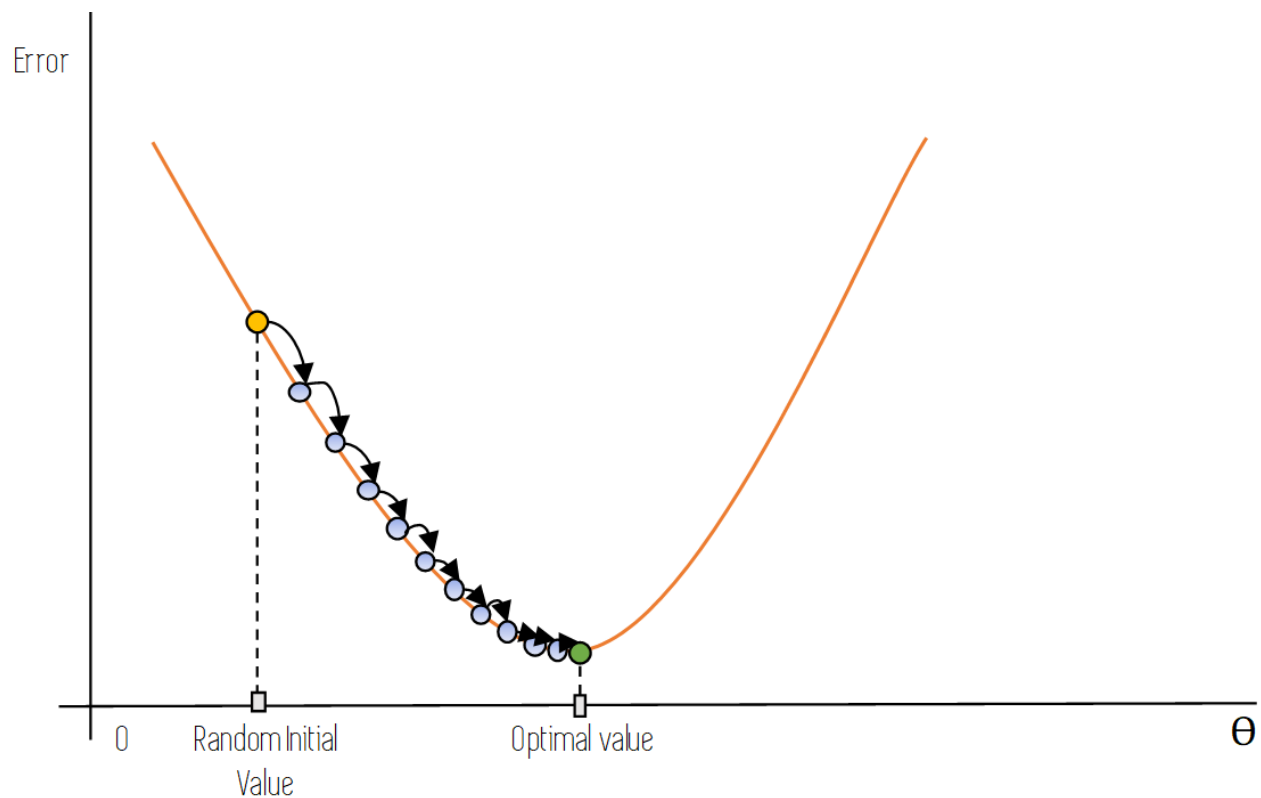
Training with Gradient Descent

is an optimisation algorithm that can be used in a wide variety of problems. The general idea of this method is to iteratively tweak the parameters of a model in order to reach the set of parameter values that minimises the error that such model makes in its predictions.

After the parameters of the model have been initialised randomly, each iteration of gradient descent goes as follows: with the given values of such parameters, we use the model to make a prediction for every instance of the training data, and compare that prediction to the actual target value.

Once we have computed this aggregated error (known as J), we measure the local gradient of this error with respect to the model parameters, and update these parameters by pushing them in the direction of descending gradient, thus making the cost function decrease.

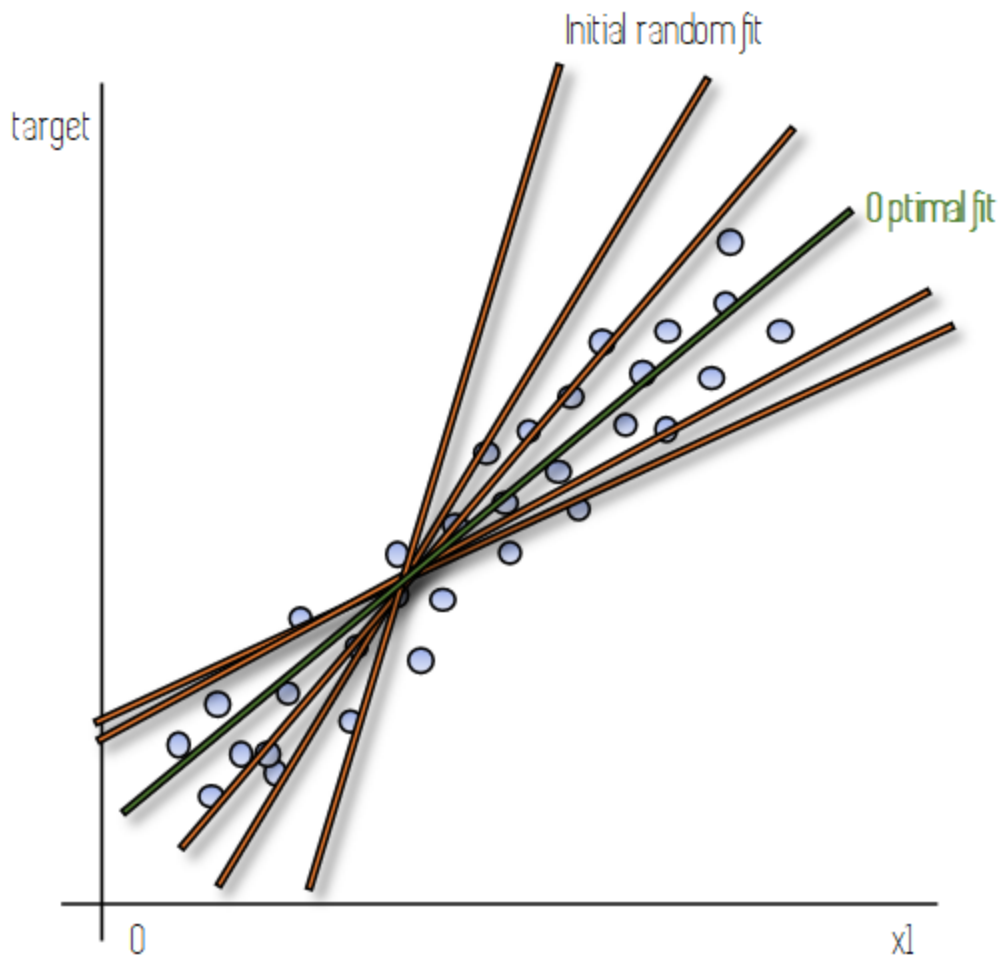
The following figure shows graphically how this is done: we start at the orange point, which is the initial random value of the model parameters. After one iteration of gradient descent, we move to the blue point which is directly right and down from the initial orange point: we have gone in the direction of descending gradient.



Gradient descent

Iteration after iteration, we travel along the orange error curve, until we reach the optimal value, located at the bottom of the curve and represented in the figure by the green point.

Imagine we had a linear model with only one feature (x_1) just so that we can plot it easily. In the following figure the blue points represent our data instances, for which we have the value of the target (for example the price of a house) and the value of the one feature (like for example the squared meters of the house).



Graphical representation of the different iterations of a linear regression model with one feature (x_1)

In practice, what happens when we train a model using gradient descent is that we start by fitting a line to our data (*the Initial random fit line*) that is not a very good representation of it. After each iteration of gradient descent, as the parameters get updated, this line changes its slope and where it cuts the y axis. This process is repeated until we reach a set of parameter values that are good enough (these are not always the optimal values) or until we complete a certain number of iterations.

These parameters are represented by the green *Optimal fit line*.

It's easy to visualise this for a model with only one feature, as the equation of the linear model is the same as the equation of a line that we learn in high school. For a higher number of features the same mechanics apply, however it is not so easy to visualise.

After we have completed the process and managed to train our model using this procedure, we can use it to make new predictions! Like shown in the following figure, using our optimal fit line, and knowing the squared meters of a house, we could use this line to make a prediction of how much it would cost.

Conclusion and Other resources

Linear regression is one of the most simple Machine Learning models. They are easy to understand, interpretable, and can give pretty good results. The goal of this post was to provide an easy way to understand linear regression in a non-mathematical manner for people who are not Machine Learning practitioners, so if you want to go deeper, or are looking for a more profound of mathematical explanation, take a look at the following video, it explains very well everything we have mentioned in this post.

That is all, I hope you liked the post. Feel free to [follow me on Twitter](#) at **@jaimezorno**. Also, you can take a look at my posts on Data Science and Machine Learning. Have a good read!

For more posts like this one, and stay tuned!

The information explained here was taken from the book in the following article, as long with some other resources.

The book to really start you on Machine Learning

Take your Machine Learning knowledge to the next level

towardsdatascience.com