

INTRODUCTION TO MACHINE LEARNING

August 14, 2020

FROM

KHIZR MOHAMMAD KHAN

**Contents**

[**List of Figures** 2](#_Toc48202023)

[**1. Logistic regression** 3](#_Toc48202024)

[**2. Regularization** 5](#_Toc48202025)

[**2.1 Bias** 5](#_Toc48202026)

[**2.2 Variance** 6](#_Toc48202027)

[**2.3 Bias – Variance Trade off (Balance between bias and variance)** 6](#_Toc48202028)

[**References** 8](#_Toc48202029)

# **List of Figures**

[Figure 1 binary classification with linear regression 2](#_Toc48201160)

[Figure 2 binary classification with linear regression with extreme case 2](#_Toc48201161)

[Figure 3 Sigmoid function 3](#_Toc48201162)

[Figure 4 underfit or high bias 5](#_Toc48201163)

[Figure 5 overfit or high variance 5](#_Toc48201164)

[Figure 6 optimal fit after bias variance trade-off 5](#_Toc48201165)

# **1. Logistic regression**

Logistic Regression, also known as Logit Regression or Logit Model, is a mathematical model used in statistics to estimate (guess) the probability of an event occurring having been given some previous data. Logistic Regression works with binary data, where either the event happens (1) or the event does not happen (0). Logistic regression is used to solve binary classification problems. It is called logistic regression because it uses log function which gives us the probability of the occurrence of the events. This function is also called the sigmoid function. Unlike Linear Regression we don’t use linear hypothesis function to classify our problem. This I want to explain with the help of figure.

Suppose we are doing some classification problem to see whether the tumor is malignant or benign based on the tumor size

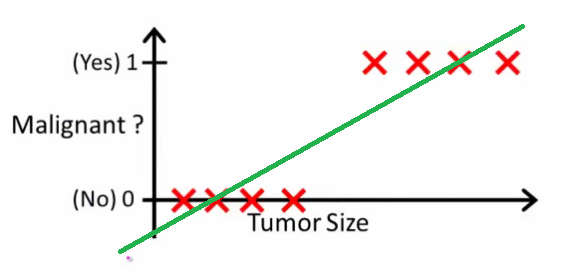


Figure 1 binary classification with linear regression

By looking the tumor size we can say that if the tumor size is greater than some threshold then it will be a malignant (1) tumor and if it is less than that threshold then it will be called a benign (0) tumor. Y axis represents the probability and X axis represents the size of the tumor. We set our threshold to be 0.5 and tumor size giving the probability greater than 0.5 will be classified as malignant and tumor size giving the probability less than 0.5 will be considered to be benign. So the linear regression seems to be working good on this dataset but it is not true for other examples. Let us consider now that there is some point which is far away these points as shown in the figure.

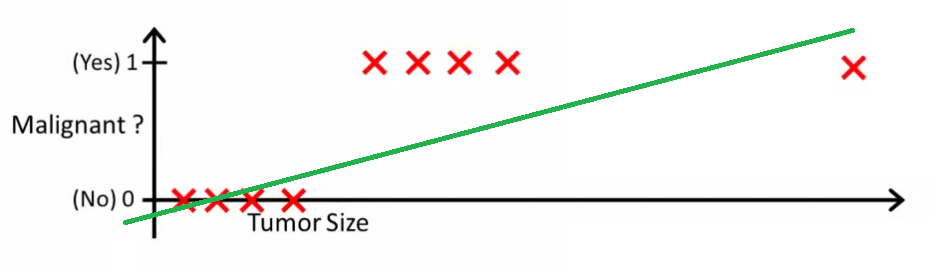


Figure 2 binary classification with linear regression with extreme case

In this particular example we see that our linear regression line will shift towards the right side and the threshold of 0.5 will not be good choice. To perform our classification task we use some different hypothesis function which is called the sigmoid function. It is given by the equation

g(z) =

z = θTX

hθ = g(z) =

The purpose of using the sigmoid function is that it will compress the probabilities between 0 and 1 rather than giving some absurd value for very large size tumors as you can see this in the above figure.

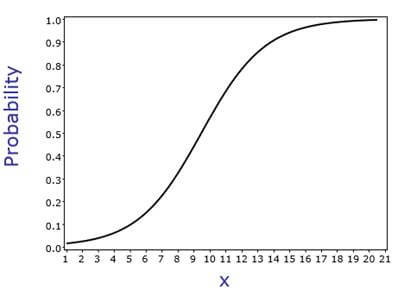


Figure 3 Sigmoid function

If you see that in logistic function if we take our input to be of very high magnitude then also the probability will remain between 1 and 0.

Similar to linear regression we try to optimize our parameters θ. To optimize θ we first derive our cost function and with help of cost function we will update our θ parameters. But the cost function is not the same as that of linear regression. I don’t want to go in that much details about how the cost function is derived, actually it comes from the principle of maximum likelihood in statistics.

hθ(x) + (1 – y) (1 - hθ (x))

Now to find the parameters θ we will use gradient descent algorithm and update the parameters θ. After differentiating our cost function we will see that our derivative term will come out to be same i.e.

hϴ(x) – y) x

And the procedure to update the values of θ will be same.

For optimizing two parameters

ϴ0 ϴ0

α. (X (ϴTX – Y)T )/m

ϴ1 ϴ1

# **2. Regularization**

Regularization is a technique used for tuning the function by adding an additional penalty term in the error function. The additional term controls the excessively fluctuating function such that the coefficients don't take extreme values.

There are two important terms which should be introduced before understanding the regularization.

1. Bias
2. Variance

## **2.1 Bias**

Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training data as well as unknown data. Such kind of situation is called underfitting or high bias.

## **2.2 Variance**

Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn’t seen before. As a result, such models perform very well on training data but has high error rates on unknown data. Such kind of situation is called overfitting or high variance.

## **2.3 Bias – Variance Trade off (Balance between bias and variance)**

If the algorithm is too simple (hypothesis with linear eq.) then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex (hypothesis with high degree eq.) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as Trade-off or Bias Variance Trade-off. This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can’t be more complex and less complex at the same time.

These things are easy to understand with the help of plots of high bias, high variance and optimum bias and variance.

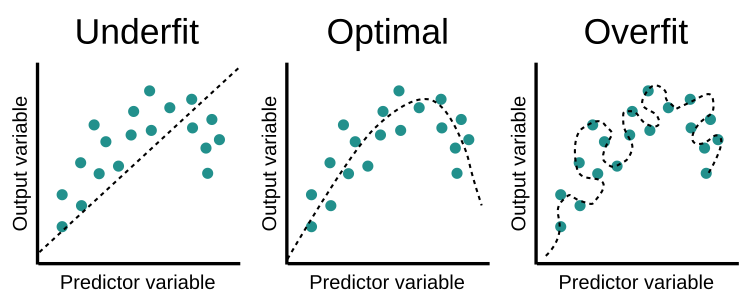


Figure 4 underfit or high bias

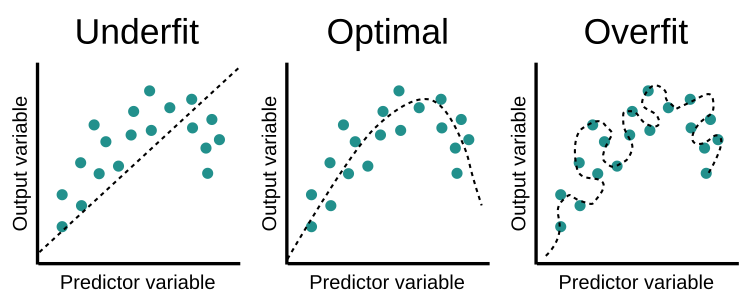


Figure 5 overfit or high variance

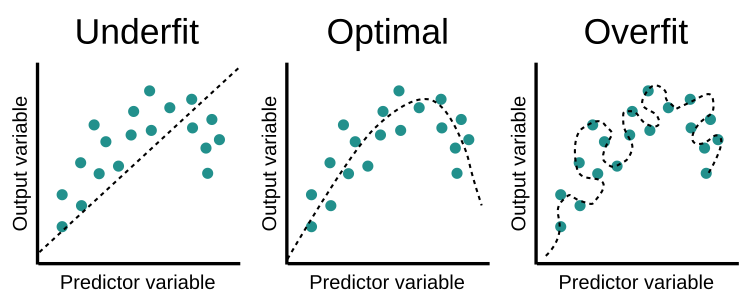


Figure 6 optimal fit after bias variance trade-off

To find the proper balance between the bias and variance we do slight modification in cost functions of our learning algorithms by penalizing our parameters which are coefficient of features in cost function in the following manner.

Suppose we take linear regression

J( ϴ) = 1/2m ∑( hϴ(xi) – yi)2 + λ/2m(ϴj2) where j ≥ 1

λ is called the regularization parameter

Similarly our ϴ update will also get modify where ϴ0 update will be the same because we are not penalizing our ϴ0 term and our derivative terms will be as follows

hϴ(x(i)) – y(i) )

hϴ(x(i)) – y(i) ) + λ/m(ϴj) where j ≥1

These derivative terms will then be put into our ϴ update algorithms.

# **References**

1. <https://stats.stackexchange.com/questions/22381/why-not-approach-classification-through-regression>

2. <https://www.jigsawacademy.com/logistic-regression-in-sas/>

3. <https://www.geeksforgeeks.org/ml-bias-variance-trade-off/>

4. <https://www.educative.io/edpresso/overfitting-and-underfitting>

5. <https://simple.wikipedia.org/wiki/Logistic_Regression>