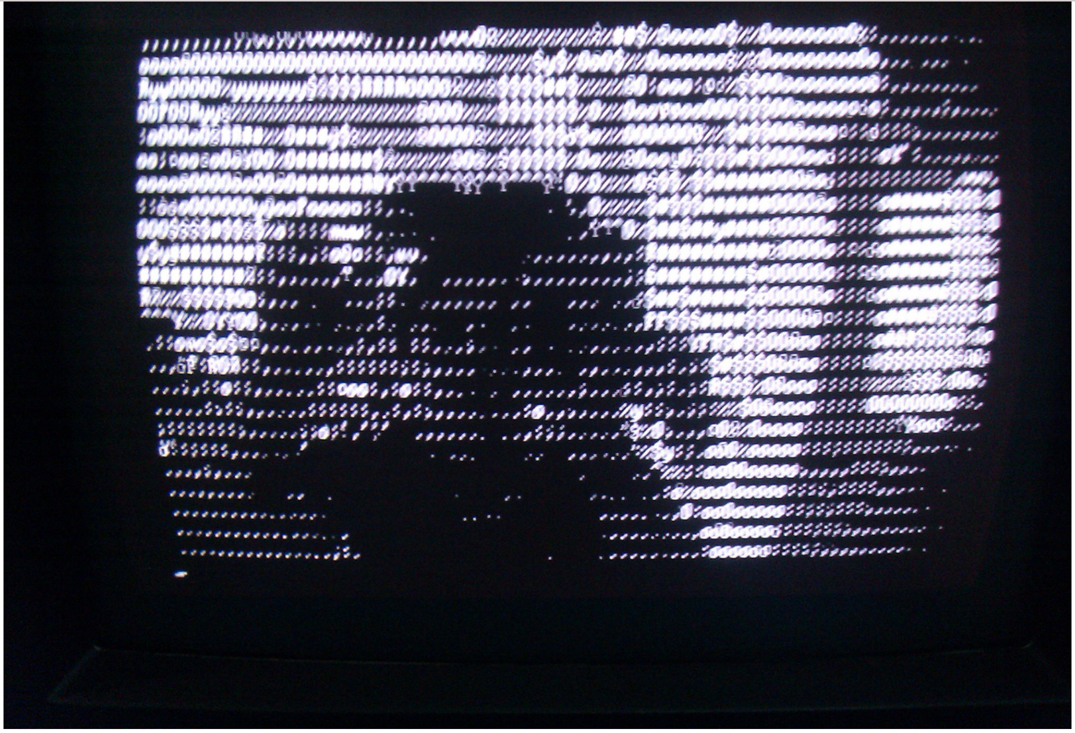
Published on March 11, 2020 In Mystery Vault (https://analyticsindiamag.com/category/mystery-vault/)

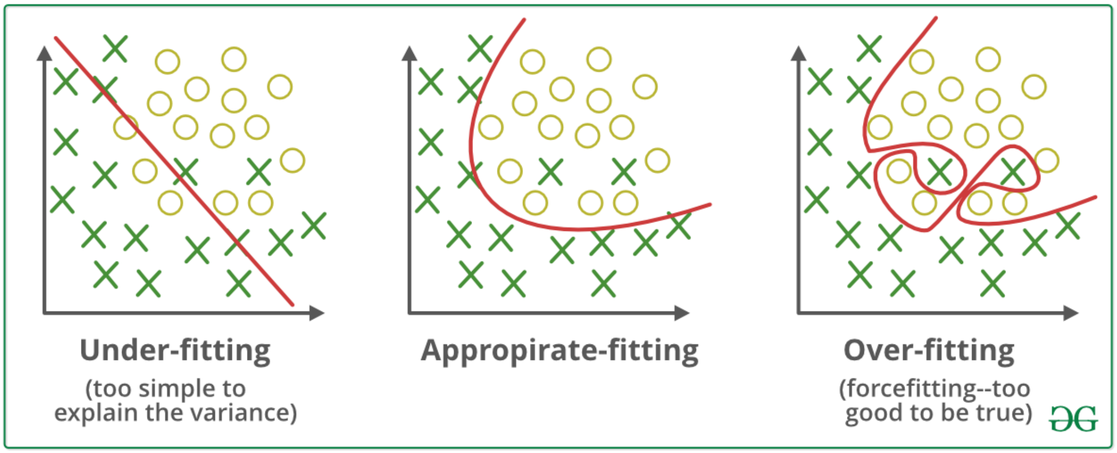
Regularization In Machine Learning – A Detailed Guide

By G R Chandrashekhar(https://analyticsindiamag.com/author/grchandraifmr-ac-in/)



Let us consider fitting a prediction model for data. One can use one of linear, quadratic, and other polynomial functional forms while fitting a regression model. Many times a linear regression model may underfit the data while a quadratic functional form may provide a better fit. To derive greater accuracy of fit if we go on to use polynomial functional forms instead of either linear or quadratic forms, it is likely that the model will fit the data very closely.

While this may appear like a great model for this dataset, if one were to change the dataset the same model may turn out to be a poor fit for the new data. What has very likely transpired is that the polynomial functional form used has fit the model so close to the original data that it is not generalizable across other similar data. This is a **problem of overfitting that one encounters in machine learning**.



Picture 1 – Overfitting in Machine Learning

A good model would be one that can pick the signal alone from the data or in other words a model that can learn the underlying generic pattern. Technically, overfitting occurs when a model fits the noise (some quirky aspects that are beyond normal) in addition to the signal. Noise can be either a stochastic type or deterministic type. The conventional meaning of noise that many are familiar with is referred to as stochastic noise.

Deterministic noise is a novel notion that is a function of the limitation of the model chosen to approximate the target. It is categorised as "noise" because it cannot be captured by the chosen model. For example, a straight line will be limited in its ability to approximate a sinusoidal wave. That aspect of the more complex target that cannot be captured by a simpler model becomes noise to the simpler model.

When noise (stochastic or deterministic (https://www.slideshare.net/sohail40/deterministic-vs-stochastic)) is part of a model, the model will extrapolate a non-existing pattern out of sample and that false pattern will take us away from the target function we are trying to approximate using machine learning. That becomes detrimental to out of sample performance or in other words overfitting harms generalization. Hence the need for regularization which is like a treatment for overfitting.

How does one address the overfitting problem?

One way could be to reduce the number of features or variables in a model. While this would increase the degrees of freedom of the model, there would be a loss of information due to the discarding of features. Thus the model would not have the benefit of all the information that would have been available otherwise. Regularization could be used in such cases that would allow us to keep all the features while working on reducing the magnitude or amplitude of the features available. Regularization also works to reduce the impact of higher-order polynomials in the model. Thus in a way, it provides a trade-off between accuracy and generalizability of a model.

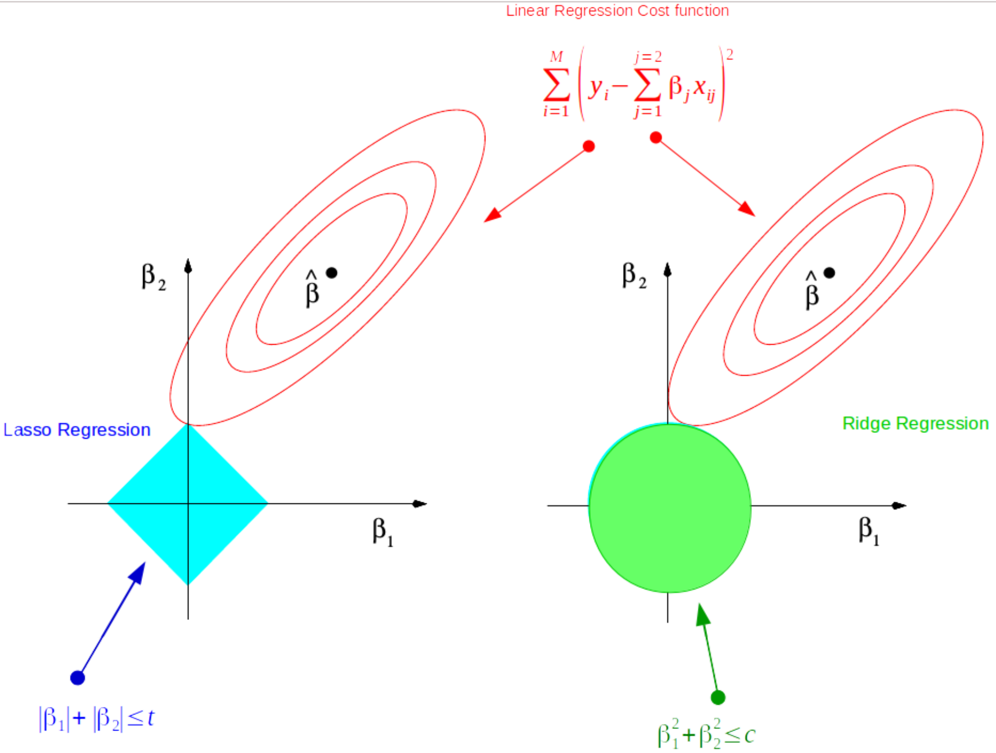
What is the challenge in regularization?

Just like cancer treatment should destruct the cancer cells alone and not affect the healthy cells in the body, the regularization approach should attack the noise alone and not affect the signal. Intuitively, when a regularization parameter is used, the learning model is constrained to choose from only a limited set of model parameters.

Instead of choosing parameters from a discrete grid, regularization chooses values from a continuum, thereby lending a smoothing effect. This smoothing effect provided by regularization is what helps the model to capture the signal well (signal is generally smooth) and filter out the noise (noise is never smooth) thereby doing the magic of fighting to overfitting successfully.

What types of regularizations are available for use in machine learning?

Two of the commonly used techniques are L1 or Lasso regularization and L2 or Ridge regularization. Both these techniques impose a penalty on the model to achieve dampening of the magnitude as mentioned earlier. In the case of L1, the sum of the absolute values of the weights is imposed as a penalty while in the case of L2, the sum of the squared values of weights is imposed as a penalty. There is a hybrid type of regularization called Elastic Net that is a combination of L1 and L2.


Linear Regression Cost function
$\sum_{i=1}^{M}\left(y_i-\sum_{j=1}^{j=2}\beta_j x_{ij}\right)^2$
β_2
$\hat{\beta}$
Lasso Regression
β_1
$|\beta_1|+|\beta_2|\leq t$
β_2
$\hat{\beta}$
Ridge Regression
β_1
$\beta_1^2+\beta_2^2\leq c$

Picture 2 – Lasso regularization and Ridge regularization

The next issue is to decide on the type of regularizer one is going to need in a model. The two types of regularizers work in slightly different ways. L1 is usually preferred when we are interested in fitting a linear model with fewer variables. L1 seems to encourage the coefficients of the variables to go towards zero because of the shape of the constraint which is an absolute value.

L1 is also useful when considering a categorical variable with many levels. L1 would make many of variable/feature weights go towards zero and thus leaving only the important ones in the model. This also helps in feature selection. L2 does not encourage convergence towards zero but is likely to make them closer to zero and prevent overfitting. Ridge or L2 is useful when there are a large number of variables with relatively smaller data samples, like in the case of genomic data.

Why did we need regularization in the first place?

Let us say that we had followed a statistical approach to building a prediction model. In a statistical approach, we would have normally followed a process of adding variables or factors one by one and tested the significance of each variable added and the overall goodness of fit of the model at each stage. In this process, we would also have also observed interaction effects, if any, between variables. In a machine learning approach, we tend to load all the variables or factors into the model and then observe the performance of the model.

In such an approach, the individual significance of variables, the interaction effects between them is not observed stage wise. Hence we do not know which variables are significant to be included and which should not be. In such a case regularization becomes handy to identify the variables or features that should remain in the model.

Machine learning is able to harness the power of a machine to build models with a large number of variables quicker than a statistical approach can, however, it lacks the discretion of a statistical approach in terms of identifying the right variables or features. Regularization comes in handy for this purpose.