Machine Learning Models Description

Linear Regression

Disadvantage

By its definition, linear regression only model relationships between dependent and independent variables that are linear. It assumes there is a straight-line relationship between them which is incorrect sometimes. Linear regression is very sensitive to the anomalies in the data (or outliers).

Take for example most of your data lies in the range 0-10. If due to any reason only one of the data item comes out of the range, say for example 15, this significantly influences the regression coefficients.

Another disadvantage is that if we have a number of parameters than the number of samples available then the model starts to model the noise rather than the relationship between the variables.

Linear Regression is a good supervised learning algorithm which is used to predictions problems, it find the target variable by finding a best suitable fit line between the independent and dependent variables. its main advantage is , the best fit line is the line with minimum error from all the points ,it has high efficiency but sometimes this high efficiency created disadvantage which is prone to overfitting of the data (i.e some noisy data also considered as useful data), and also it cant be used when the relation between dependent and independent variable is not linear.

Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.

Gradient descent is a method of updating a\_0 and a\_1 to reduce the cost function(MSE)

Linear regression is a very simple approach for supervised learning. Though it may seem somewhat dull compared to some of the more modern algorithms, linear regression is still a useful and widely used statistical learning method. Linear regression is used to predict a quantitative response Y from the predictor variable X.

Linear Regression is made with an assumption that there’s a linear relationship between X and Y.

Model evaluation metrics for regression

Evaluation metrics for classification problems, such as accuracy, are not useful for regression problems. Instead, we need evaluation metrics designed for comparing continuous values, here I’m using well Root Mean Squared Error, of course there are others but this is one of the favorites choice and we are going to go along with it.

Root Mean Squared Error is the square root of the mean of the squared errors(MSE), MSE by itself can be used for evaluation metric, but that it’s subject for another post.

Once we find the best θ1 and θ2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

Cost Function (J):

By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the θ1 and θ2 values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y)

Cost function(J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true y value (y).

Gradient Descent:

To update θ1 and θ2 values in order to reduce Cost function (minimizing RMSE value) and achieving the best fit line the model uses Gradient Descent. The idea is to start with random θ1 and θ2 values and then iteratively updating the values, reaching minimum cost.

Linear regression can provide valuable insight about the relationships between the target and feature columns in your data, revealing why your model returns the predictions that it does. The coefficients () are what the training procedure learns. Each model coefficient describes the expected change in the target variable associated with a unit change in the feature. The bias term indicates the "inherent" or "average" target value if all feature values were set to zero.

The coefficients often tell an interesting story of how much each feature matters in predicting target values. The magnitude (absolute value) of the coefficient for each feature indicates the strength of the feature's association to the target variable, holding all other features constant. The sign on the coefficient (positive or negative) gives the direction of the association.

When a GraphLab Create regression model is trained, the model.summary() output shows the largest positive and negative coefficients. For a trained model, we can access the coefficients as follows:

Random Forest Regression

The random forest model is a type of additive model that makes predictions by combining decisions from a sequence of base models. More formally we can write this class of models as:

g(x)=f0(x)+f1(x)+f2(x)+...

where the final model g is the sum of simple base models fi. Here, each base classifier is a simple decision tree. This broad technique of using multiple models to obtain better predictive performance is called model ensembling. In random forests, all the base models are constructed independently using a different subsample of the data.

Different kinds of models have different advantages. The random forest model is very good at handling tabular data with numerical features, or categorical features with fewer than hundreds of categories. Unlike linear models, random forests are able to capture non-linear interaction between the features and the target.

One important note is that tree based models are not designed to work with very sparse features. When dealing with sparse input data (e.g. categorical features with large dimension), we can either pre-process the sparse features to generate numerical statistics, or switch to a linear model, which is better suited for such scenario

Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks.

Decision Tree Regression

<https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96>

Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks.

Different kinds of models have different advantages. The decision tree model is very good at handling tabular data with numerical features, or categorical features with fewer than hundreds of categories. Unlike linear models, decision trees are able to capture non-linear interaction between the features and the target.

One important note is that tree based models are not designed to work with very sparse features. When dealing with sparse input data (e.g. categorical features with large dimension), we can either pre-process the sparse features to generate numerical statistics, or switch to a linear model, which is better suited for such scenarios.

Gradient Boosted Regression Trees

Cluster

<https://hdbscan.readthedocs.io/en/latest/performance_and_scalability.html>