Linear Regression-Simple and Multi variable with coding

Introduction

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog). There are two main types:

```
1.Simple Regression.
2.Multivariable Regression.
```

1.Simple Regression.

Simple linear regression uses traditional slope-intercept form, where m and b are the variables our algorithm will try to "learn" to produce the most accurate predictions. **x represents our input data and y represents our prediction.**

```
y=mx+b
```

2.Multivariable regression

A more complex, multi-variable linear equation might look like this, where w represents the coefficients, or weights, our model will try to learn.

```
f(x,y,z)=w1x+w2y+w3z
```

The variables **x,y,z** represent the **attributes, or distinct pieces of information,** we have about each observation.

For sales predictions, these attributes might include a company's advertising spend on radio, TV, and newspapers.

```
Sales=w1*Radio+w2*TV+w3*News
```

Double-click (or enter) to edit

Let's say we are given a dataset with the following columns (features): how much a company spends on Radio advertising each year and its annual Sales in terms of units sold. We are trying to develop an equation that will let us to predict units sold based on how much a company spends on radio advertising. The rows (observations) represent companies.

Company	Radio ($)	Sales
Amazon	37.8	22.1
Google	39.3	10.4
Facebook	45.9	18.3
Apple	41.3	18.5

Making predictions

Our prediction function outputs an estimate of sales given a company's radio advertising spend and our current values for Weight and Bias.

$$Sales = Weight \cdot Radio + Bias$$

Weight

the coefficient for the Radio independent variable. In machine learning we call coefficients weights.

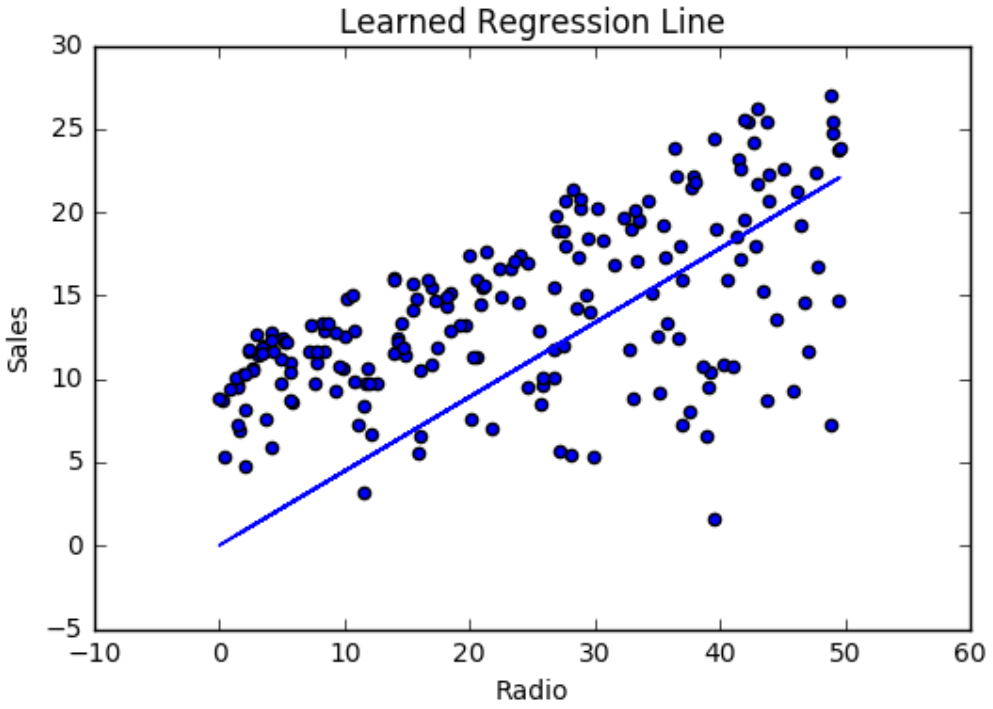
Radio

the independent variable. In machine learning we call these variables features.

Bias

the intercept where our line intercepts the y-axis. In machine learning we can call intercepts bias. Bias offsets all predictions that we make.

Our algorithm will try to learn the correct values for Weight and Bias. By the end of our training, our equation will approximate the line of best fit.



Double-click (or enter) to edit

Cost function

Different techniques can be used to prepare or train the linear regression equation from data, the most common of which is called Ordinary Least Squares. It is common to therefore refer to a model prepared this way as Ordinary Least Squares Linear Regression or just Least Squares Regression.

The prediction function is nice, but for our purposes we don't really need it. What we need is a cost function so we can start optimizing our weights.

Let's use **MSE (L2)** as our **cost function**. MSE measures **the average squared difference between an observation's actual and predicted values.** The output is a single number representing the cost, or score, associated with our current set of weights. **Our goal is to minimize MSE to improve the accuracy of our model.**

Math

Given our simple linear equation y=mx+b, we can calculate MSE as:

```
#MSE=1/N∑i=1n(yi−(mxi+b))2
```

N is the total number of observations (data points)

1/N∑ni=1 is the mean

yi is the actual value of an observation and mxi+b is our prediction

Loss Functions

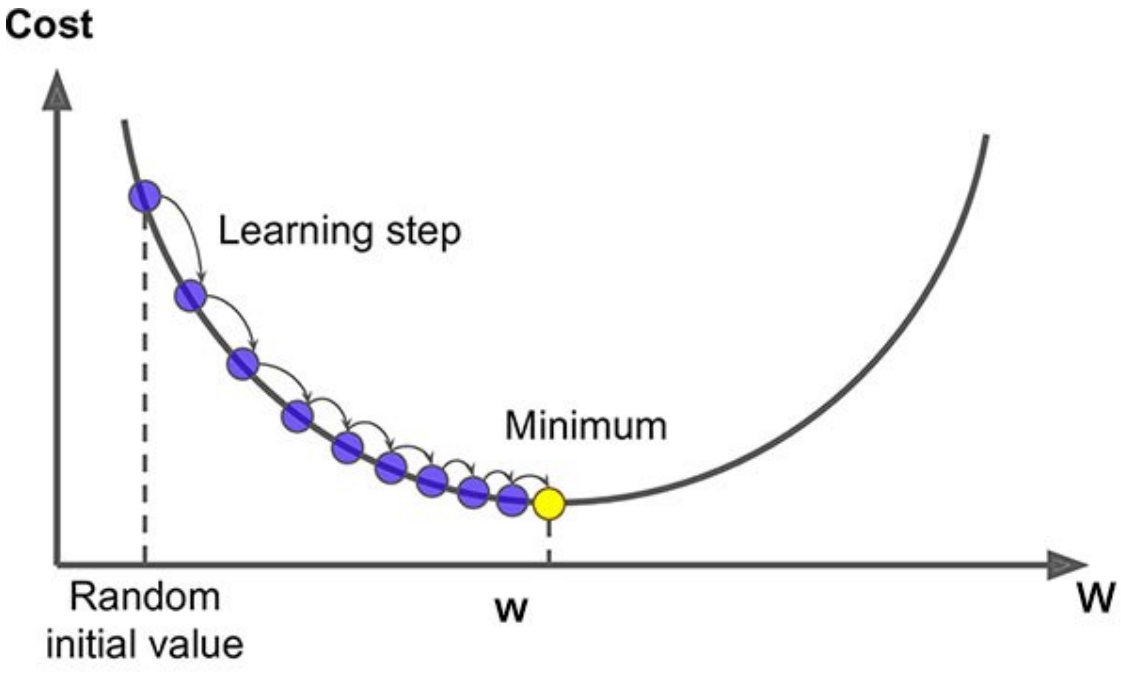
```
1.Cross-Entropy            [Classification]
2.Hinge                    [Classification]
3.Huber                    [Regression]
4.Kullback-Leibler         [Regression]

5.MAE (L1)                 [Regression]
        S=∑i=0n|yi−h(xi)|

6.MSE (L2)                 [Regression]
        S=∑i=0n(yi−h(xi))2
```

Gradient descent

To minimize MSE we **use Gradient Descent** to calculate the **gradient of our cost function**. Gradient descent consists of looking at the error that our weight currently gives us, using the derivative of the cost function to find the gradient (The slope of the cost function using our current weight), and then changing our weight to move in the direction opposite of the gradient. We need to move in the opposite direction of the gradient since the gradient points up the slope instead of down it, so we move in the opposite direction to try to decrease our error.



Training

Training a model is the process of iteratively improving your prediction equation by looping through the dataset multiple times, **each time updating the weight and bias values in the direction indicated by the slope of the cost function (gradient).** Training is complete when we reach an acceptable error threshold, or when subsequent training iterations fail to reduce our cost.

Before training we need to **initialize our weights (set default values), set our hyperparameters** (learning rate and number of iterations), and prepare to log our progress over each iteration.

Parameters

Hyperparameters

Model evaluation

If our model is working, we should see our cost decrease after every iteration.

Assumptions of Linear Regression

1.Linear relationship:

There exists a linear relationship between the independent variable, x, and the dependent variable, y.

How can you determine if the assumption is met?

The simple way to determine if this assumption is met or not is by creating a **scatter plot x vs y**. If the data points fall on a **straight line in the graph, there is a linear relationship** between the dependent and the independent variables, and the assumption holds.

What should you do if this assumption is violated?

If a linear relationship doesn't exist between the dependent and the independent variables, then apply a **non-linear transformation such as logarithmic, exponential, square root, or reciprocal** either to the dependent variable, independent variable, or both.

2.Independence: The residuals[Error:The difference between the observed value of the dependent variable (y) and the predicted value (ŷ) is called the residual (e)] are independent. In particular, there is no correlation between consecutive residuals in time series data.

How to determine if the assumption is met?

Conduct a Durbin-Watson (DW) statistic test. The **values should fall between 0-4. ** If **DW=2, no auto-correlation**; if **DW lies between 0 and 2, it means that there exists a positive correlation**. If **DW lies between 2 and 4, it means there is a negative correlation**. Another method is to plot a graph against residuals vs time and see patterns in residual values.

What should you do if this assumption is violated?

If the assumption is violated, consider the following options:

For **positive correlation**, consider adding lags to the dependent or the independent or both variables.

For negative correlation, check to see if none of the variables is over-differenced.

For seasonal correlation, consider adding a few seasonal variables to the model.

3.**Homoscedasticity**: The residuals have constant variance at every level of x.

How to determine if the assumption is met?

Create a scatter plot that shows residual vs fitted value. If the data points are spread across equally without a prominent pattern, it means the residuals have constant variance (homoscedasticity). Otherwise, if a funnel-shaped pattern is seen, it means the residuals are not distributed equally and depicts a non-constant variance (heteroscedasticity).

What should you do if this assumption is violated?

Transform the dependent variable

Redefine the dependent variable

Use weighted regression

4.**Normality**: The residuals of the model are normally distributed.

If one or more of these assumptions are violated, then the results of our linear regression may be unreliable or even misleading.

How to determine if the assumption is met?

Check the assumption using a **Q-Q (Quantile-Quantile) plot**. If the data points on the graph form a straight diagonal line, the assumption is met.

What should you do if this assumption is violated?

Verify if the outliers have an impact on the distribution. Make sure they are real values and not data-entry errors.

Apply non-linear transformation in the form of log, square root, or reciprocal to the dependent, independent, or both variables.

No Multicollinearity

The independent variables shouldn't be correlated. If multicollinearity exists between the independent variables, it is challenging to predict the outcome of the model. In essence, it is difficult to explain the relationship between the dependent and the independent variables. In other words, it is unclear which independent variables explain the dependent variable.

The standard errors tend to inflate with correlated variables, thus widening the confidence intervals leading to imprecise estimates.

How to determine if the assumption is met?

Use a scatter plot to visualise the correlation between the variables. Another way is to determine the VIF (Variance Inflation Factor). VIF<=4 implies no multicollinearity, whereas VIF>=10 implies serious multicollinearity.

Advantages of Linear Regression

1.Simple implementation.

2.Performance on linearly seperable datasets.

3.Overfitting can be reduced by regularization

Disadvantages of Linear Regression

1.Prone to underfitting

2.Sensitive to outliers

Multivariable regression

Let's say we are given data on TV, radio, and newspaper advertising spend for a list of companies, and our goal is to predict sales in terms of units sold.

Company	TV	Radio	News	Units
Amazon	230.1	37.8	69.1	22.1
Google	44.5	39.3	23.1	10.4
Facebook	17.2	45.9	34.7	18.3
Apple	151.5	41.3	13.2	18.5

Normalization

As the number of features grows, calculating gradient takes longer to compute. We can speed this up by "normalizing" our input data to ensure all values are within the same range. This is especially important for datasets with high standard deviations or differences in the ranges of the attributes. Our goal now will be to normalize our features so they are all in the range -1 to 1.

Making predictions

Our predict function outputs an estimate of sales given our current weights (coefficients) and a company's TV, radio, and newspaper spend. Our model will try to identify weight values that most reduce our cost function.

```
Sales=W1*TV+W2*Radio+W3*Newspaper
```

Cost function

Now we need a cost function to audit how our model is performing. The math is the same, except we swap the mx+b expression for W1x1+W2x2+W3x3. We also divide the expression by 2 to make derivative calculations simpler.

$$MSE = \frac{1}{2N}\sum_{i=1}^{n}(y_i - (W_1x_1 + W_2x_2 + W_3x_3))^2$$

Bias term

Our train function is the same as for simple linear regression, however we're going to make one final tweak before running: add a bias term to our feature matrix.

In our example, it's very unlikely that sales would be zero if companies stopped advertising. Possible reasons for this might include past advertising, existing customer relationships, retail locations, and

Advantages of Multivariable Regression

1.Works on any kind of dataset.

2.Works well on any non-liner problem.

Disadvantages of Multivariable Regression

We need to take good degree of the polynomial for good bias and variance tradeoff.