Optimizing Advertising Expenditure for Maximized Sales using Linear Regression with Gradient descent as a Optimizer

Zia Ur Rehman

October 2023

1 Problem Statement

In the realm of marketing, understanding the impact of advertising expenditures on overall product sales is crucial. Leveraging the "Advertising Data Set" sourced from Kaggle, comprising information on sales figures for various products based on advertising investments in TV, radio, and newspapers, the objective is to determine the optimal allocation of resources among different advertising mediums. By applying the gradient descent algorithm in conjunction with Linear Regression, we aim to identify the most influential medium and allocate appropriate weights to each advertising channel to maximize overall sales. The core objective is to uncover the relationship between advertising expenditures and sales, facilitating informed decision-making in resource allocation for marketing strategies. The problem demands the utilization of diverse optimization techniques, exploring the intricacies of the data, and uncovering potential non-linear relationships to enhance the accuracy and efficacy of the advertising strategy. Our primary focus is to investigate the influence of the input (advertising on TV, radio, and newspaper) on potential changes in sales. This analysis will help to optimize the advertising strategy, allocate resources effectively, and enhance overall sales performance.

2 Introduction

In contemporary advertising, businesses are continually seeking to optimize their marketing strategies to maximize sales and enhance return on investment (ROI). The provided dataset includes detailed information about various advertising campaigns, focusing on three key mediums: TV, radio, and newspapers. By employing the techniques of Linear Regression and the Gradient Descent optimization algorithm, the objective is to determine the most effective advertising mix that results in the highest possible sales.By analyzing this dataset,our objective function is the minimization of the error between the predicted sales

and the actual sales. Utilizing Linear Regression, we aim to model the relationship between advertising expenditure across different platforms (TV, radio, and newspapers) and the resultant sales. The Gradient Descent optimization algorithm enables the determination of the optimal parameters that best fit the data, thus allowing for the identification of the advertising strategy that yields the maximum sales and ROI. By iteratively adjusting the parameters, the algorithm converges on the optimal solution that minimizes the error and maximizes sales. In the next subsection, we explain the basics of these algorithms and formalization of our problem.

2.1 Linear Regression

Linear Regression is a technique used for modeling the relationship between a dependent variable and one or more independent variables by using the linear relationship. Here is the simple equation for using linear regression to represent the line. (equation add)Here y is the dependent variable,x is the independent variable,m shows the slope, and c represent the intercept.

$$y = mx + c \tag{1}$$

The main goal of LR is to find the best line that minimizes the difference between the observed value and model output. We formalized our problem in this way as represented in this equation:

$$Y' = WtvXtv + WradXrad + WnesXnew + b \tag{2}$$

Here Y represents the total sales Xtv,Xrad,Xnew represent the independent variables, and represented as advertising expenditures. And Wtv, Wrad, and Wnes represent the weights of each medium that we find by using the optimization technique. As an evaluation metric, we use Mean squared error for our analysis and this means squared error or loss function is also known as the objective function for our problem as represented in this equation. The objective function shows the difference between actual and predicted values and calculates the square root of all parameters. Here the main target is to find those parameters or weights that reduce the errors between predicted and actual,

2.2 Gradient descent

Gradient descent is an optimization algorithm used in machine learning for the optimization of model parameters by minimizing the cost functions or convex functions. The main idea behind this algorithm is to update the parameters in the opposite directions of gradients of the cost function with respect to parameters. According to our problem advertising effect on sales gradient descent find the best parameters for reducing the loss function and maximizing the sales and also analyze which advertising features have the highest contribution in sales profit. This was the mathematical explanation of our problem using gradient descent and according to this this was the cost function and the other property

of this function is a convex function (global minimum). Our main target is find the values of paramters or weights Wtv,Wrad and Wnew that reduce the mean squared errors For this purpose we implemented gradient descent in this way:

- 1. As a first step start from random value(weights and bias) and moves toward the optimal solution
- 2. calculate the estimated value of Y(predicted) for using the randmized weights.
- 3. Calculate mean squared error
- 4.Adjusting the values of weights by calculating the grading of error function

$$minJ(wi, xi) = \frac{1}{n}\sum_{i=1}^{n}(Yi-(WtvXtv+WradXrad+WnesXnew))^2 \quad (3)$$

$$b = b - \frac{\partial J(wi, xi)}{\partial b} \quad (4)$$

$$Wtv = Wtv - \frac{\partial J(wi, xi)}{\partial Wtv} \quad (5)$$

$$Wrad = Wrad - \frac{\partial J(wi, xi)}{\partial Wrad} \quad (6)$$

$$Wnew = Wnew - \frac{\partial J(wi, xi)}{\partial Wnew} \quad (7)$$

All these values Wtv, Wrad, Wnews, and b represent the weights and these equation shows the updation of weights A is the learning rate which represent the magnitude of update needed in these values to handle the correct global minimum. Here one important information that we will show is partial derivation of these weights and base values for this purpose this was the formula use for partial derivative:

$$\frac{\partial J(wi, xi)}{\partial b} = \frac{-2}{N}\sum_{i=1}^{N}(Y - Y') \quad (8)$$

$$\frac{\partial J(wi, xi)}{\partial Wrad} = \frac{-2}{N}\sum_{i=1}^{N}(Y - Y') * Xi \quad (9)$$

$$\frac{\partial J(wi, xi)}{\partial Wnews} = \frac{-2}{N}\sum_{i=1}^{N}(Y - Y') * Xi \quad (10)$$

$$\frac{\partial J(wi, xi)}{\partial Wtv} = \frac{-2}{N}\sum_{i=1}^{N}(Y - Y') * Xi \quad (11)$$

- 5. Repeat steps 1 to 4 for several iterations until the error stops reducing further or the change in cost is infinitesimally small.

  This algorithm will produce the best values for all weights of TV an other features while reducing the minimum cost point.

2.3 Metholodogy and Result Analysis

As our dataset (advertising) consists of 200 samples with three input features and one output feature(Sale) our target is to optimize the parameter of input features and analyze the sale optimization problem. A first we perform preprocessing (remove some values) and normalization of data (this can be done by subtraction means from features divided by standard deviation).In this way, we ensure all features are on the same scale. We use linear regression for our analysis as a first step to calculate mean squared error.Then we apply gradient descent as we already discussed in detail for the selection of best parameter values. This was the ouput of gradient descent

- Mean Squared Error (Linear Regression): 1.632219973467452
- Mean Squared Error (Gradient Descent): 0.645110668988854
- Final Estimate of b and theta : 5.596579039064977e-06 [0.75306255 0.53619132 -0.00403905]

As we see in Figure 1 and Figure 2 no of iteration increase the loss will decrease but we consider best Figure 2 because of low loss error with respect to Figure 1.

In the last, we compare different learning rates for analyzing the performance of the loss function with respect to a number of iterations. As you can see in Figure 7 we use three different learning rates (0.01,05,0.001) for analyzing the gradient descent performance. The yellow line indicates the constant convergence so it will not be considered but the blue curve indicates that the cost is highest in the initial phase and gradually with the number of epoch increase so it will less minimum time to converge to the global minimum.

2.4 Stochastic Gradient descent

stochastic Gradient is a variant of Gradient descent optimization algorithms in which we compute the gradient using only a single or small subset of training examples. All the procedures were the same as we discussed in the main gradient descent algorithm. This criteria of choosing a small subset lead to faster convergence in certain cases but this will depend on the learning rate and number of iterations.

2.4.1 Result Analysis

- Mean Squared Error (Linear Regression): 0.6231199304814649

	iteration	cost
0	0.0	0.428903
1	10.0	0.300880
2	20.0	0.224607
3	30.0	0.178762
4	40.0	0.150912
5	50.0	0.133778
6	60.0	0.123080
7	70.0	0.116289
8	80.0	0.111900
9	90.0	0.109007

Figure 1: No of iteration vs cost for Number of iteration

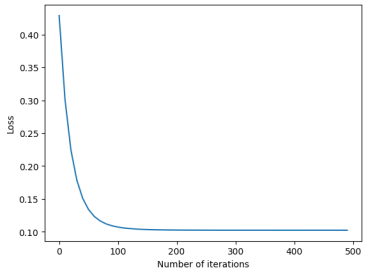

Figure 2: MSE Loss

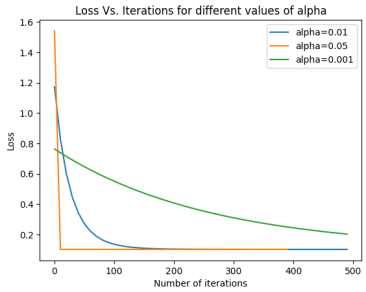

Figure 3: Comparison of different learning rates.

	iteration	cost
0	0.0	1.227512
1	10.0	0.829812
2	20.0	0.587660
3	30.0	0.434081
4	40.0	0.333388
5	50.0	0.265646
6	60.0	0.219170
7	70.0	0.186803
8	80.0	0.163995
9	90.0	0.147765

Figure 4: No of iteration vs cost for Number of iteration

- Mean Squared Error (Stochastic Gradient Descent): 0.10227908109852424
- Final Estimate of b (intercept): -7.26007299692408e-05 Final Estimate of theta (coefficients): [0.75291294 0.53340338 -0.00087803]

In the initially, we compute the mean square error of linear regression, but when we pass to over best-optimized values it will decrease the mean squared error. Moreover as can seen in Figure 4 the cost is high at the start of the iteration but gradually the cost decreases when we increase the number of epochs this will converge faster. We also try different learning rates with different numbers of iterations to analyze the convergence behavior so you can see in Figure 6 yellow line represents the learning rate of (0.005) it converges faster with respect to other learning rates but we consider best green line (0.0001) because of better converge (not static)

2.5 Minibatch Gradient

Minibatch is also a variant of gradient descent in which we use a small batch of the data point is used for computing the gradient of the cost function. The steps were all the same with respect to the main gradient descent as we discussed in the previous section. The only change is the dataset partition in which we distribute the dataset into small batches of equal or varying sizes and the training was done

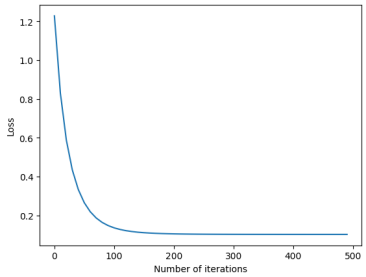

Figure 5: MSE

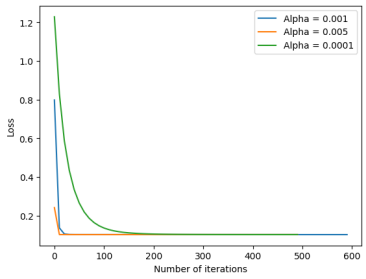

Figure 6: Comparison of different learning rates.

by iterating through each batch at a time.The main benefit of this technique is it takes vectorized operation for computing the gradient and updating the parameters.

- Mean Squared Error (Linear Regression): 1.6866469878042745
- Mean Squared Error (Minibatch Gradient Descent): 0.10279386236140299
- Final Estimate of b (intercept): 0.02267142548719684 Final Estimate of theta (coefficients): [0.73693582 0.53230544 0.00320827]

As we can see in Figure 7 in the start the cost is very high with respect to all other optimization algorithms but number of iterations increases it will decrease the loss, Moreover, we also analyze the different learning rates to analyze the performance of our optimizer,As you can see in the Figure 9 with the rate (0.01) convergence fastly and the yellow and green high convergence time.

2.5.1 Result Analysis

2.6 Newton Method

Nethon Method is an iterative algorithm for finding roots of real-valued functions, In the context of machine learning Newton method acts as a optimizer to

	iteration	cost
0	0.0	2.233053
1	10.0	1.720905
2	20.0	1.334977
3	30.0	1.043373
4	40.0	0.822473
5	50.0	0.654728
6	60.0	0.527051
7	70.0	0.429654
8	80.0	0.355197
9	90.0	0.298156

Figure 7: No of iteration vs cost for Number of iteration

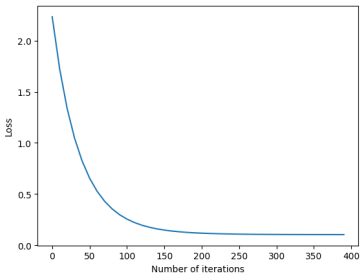

Figure 8: MSE

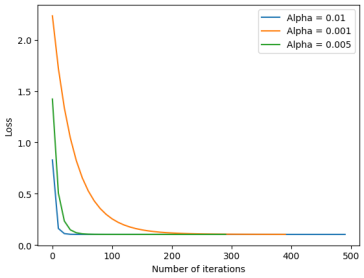

Figure 9: Comparison of different learning rates.

find the maximum or minimum of a function by iteratively updating the solution vectors. These were the main steps that we followed in Newton Method:

- 1. choose the initial guess for the solution vector in this case we represented as Wi and B0.
- 2. Calculate the gradient and Hessian matrix of the cost function at the current weights. The gradient shows the first derivative and hessian shows the second partial derivative of weights. According to our optimization problem cost function represents the difference between actual sales and predicted sales. We basically formulated in this way:

$$\nabla J(W) = \begin{bmatrix} \frac{\partial J(wi,xi)}{\partial Wtv} \\ \frac{\partial J(wi,xi)}{\partial Wrad} \\ \frac{\partial J(wi,xi)}{\partial Wnews} \end{bmatrix} \tag{12}$$

$$H(W) = \begin{bmatrix} \frac{\partial^2 J(wi,xi)}{\partial Wtv^2} & \frac{\partial^2 J(wi,xi)}{\partial Wtv * \partial Wrad} & \frac{\partial^2 J(wi,xi)}{\partial Wtv * \partial Wnew} \\ \frac{\partial^2 J(wi,xi)}{\partial Wrad * \partial Wtv} & \frac{\partial^2 J(wi,xi)}{\partial Wrad^2} & \frac{\partial^2 J(wi,xi)}{\partial Wrad * \partial Wnew} \\ \frac{\partial^2 J(wi,xi)}{\partial Wnew * \partial Wtv} & \frac{\partial^2 J(wi,xi)}{\partial Wnew * \partial Wrad} & \frac{\partial^2 J(wi,xi)}{\partial Wnew^2} \end{bmatrix} \tag{13}$$

- 3. Use these gradient an hessian matrix values to update the weights for this purpose we formalize our problem in this way.

$$W(n+1) = Wn - [H(Wn)]^{-} 1 * \partial J(wi, xi) \tag{14}$$

- 4. Repeat steps 2 and 3 until reach the local minimum of cost function

The main advantage of this optimization algorithm is rapid convergence near the maximum or minimum of a function especially when the second derivative is not zero and the Hessian matrix is well-defined. However, it may encounter some problems such as convergence of saddle point, and efficient updation required for solution vector(computationally expensive for high dimensional problem).Variation of these models such Quasi the newton method to solve the issue of computations.

2.6.1 Result Analysis

- Mean Squared Error (Newton-CG): 0.10227541501194284
- The best optimization algorithm is Newton-CG (LR=0.01) with MSE 0.1023 For the implementation of the Newton method we use the scipsy library as we can analyze from Figure 10 Newton method convergence is faster with respect to all other optimizers,its loss value start from 0.19 and firstly 200 iteration its behavior is static then start slowly convergence to minimum value of loss function with learning rate of 0.01.

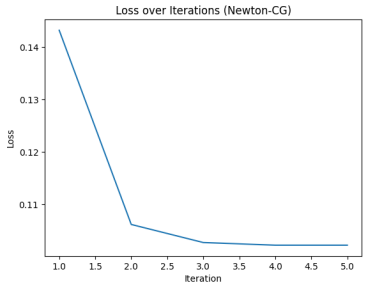

Figure 10: No of iteration vs cost for Number of iteration

	Feature	Simple Gradient	Minibatch GD	Stochastic GD	Newton-CG
0	TV	0.740632	0.740150	0.752011	0.753066
1	radio	0.537076	0.537277	0.532457	0.536482
2	newspaper	-0.004275	-0.003858	0.000333	-0.004331

Figure 11: Extract coefficients from different optimization methods

3 Analysis of features and Conclusion

We also implement one other method optimize_and_compare for analyzing the four different optimization algorithms to find the best parameters or weight values of Wtv, Wrad, Wnew and analyze the best optimizer. According to our analysis, Newton's method is the best optimizer (not considering the computational time). The second best optimizer method is stochastic. At the end, we also analyze the weights of all optimizers to check the performance of which features more impact on sales. So you can see in Figure 11 all these four optimizers consider investment in TV advertising to have the highest impact on sales with respect to radio and newspaper. Our performance is not according to the state of the art because we use only 200 instances so that's why the regression error is very high but as a future perspective we increase using a high number of instances and also these techniques applied in other domains or different datasets (using social media data to analyze the sales, analyze the customer behaviour impact on sales)