3/30/2021

Biomedical Imaging Informatics

Group-2

Vanusha Goyal
20MM45006
Chandrima Debnath
20MM92R03
Sandeep Changeriwal
20MM61R04
Rishabh Gupta
20MM61R07
Prabir Kumar Das
20MM62R02
Sayan Das
20MM45004
Koushik Kundu
20MM45002
Sumeet Kumar Rath
20MM61R13
Ausath G Anto
19MM60016
Chetan Shirvankar
20MM60009
Jaswanth G
20MM60015

AIM AND OBJECTIVES:

Our aim in this project is to apply the gradient descent iterative optimization algorithm to describe the relation between the relative volume of organ receiving a particular dose (y) and the following features:

1. Volume of organ
2. PTV receiving 60Gy radiation
3. PTV receiving 44Gy radiation

We aim to construct an algorithm using the training data provided in part A of the question and then using this algorithm, obtain y or the amount of radiation or dose received by the organ at risk in part B of the question.

INTRODUCTION:

Cancer is a disease in which abnormal cells divide uncontrollably and destroy the body tissue. The most widespread types of cancer are breast, prostate, lung, colon, skin, etc. There are numerous treatment options for cancer. The main ones are radiation therapy, targeted therapy, chemotherapy, surgery, hormonal therapy, etc. Often, the primary step of therapeutic measures is performing radical or conformal radiation therapy because of its ability to control cell growth and is followed by other treatments like surgery.

Radiation therapy employs ionizing radiation to control or kill malignant cells. Ionizing radiation functions by damaging the DNA of these malignant cells causing cellular death while sparing the normal cells. The radiation used can be either external (high energy x rays) or internal (radioactive source placed inside the body). Conformal radiotherapy makes use of a tool in the radiotherapy device to construct the radiation beams to match the shape of cancer. This makes sure that the enveloping healthy cells receive lesser or no radiation. Now, radiation can cause severe damage to the healthy tissues also, so to prevent that radiation dose and volume of the organ needs to be analyzed particularly.

Radiation Dose - It is the concentration of energy deposited in the tissue as a result of exposure to ionizing radiation. It mainly describes the intensity of the energy deposited in any small amount of tissue located anywhere in the body.Units for radiation is Gray (Gy) or milli-gray (mGy).

Various Target Volumes:

1. **GTV - GROSS TUMOR VOLUME**

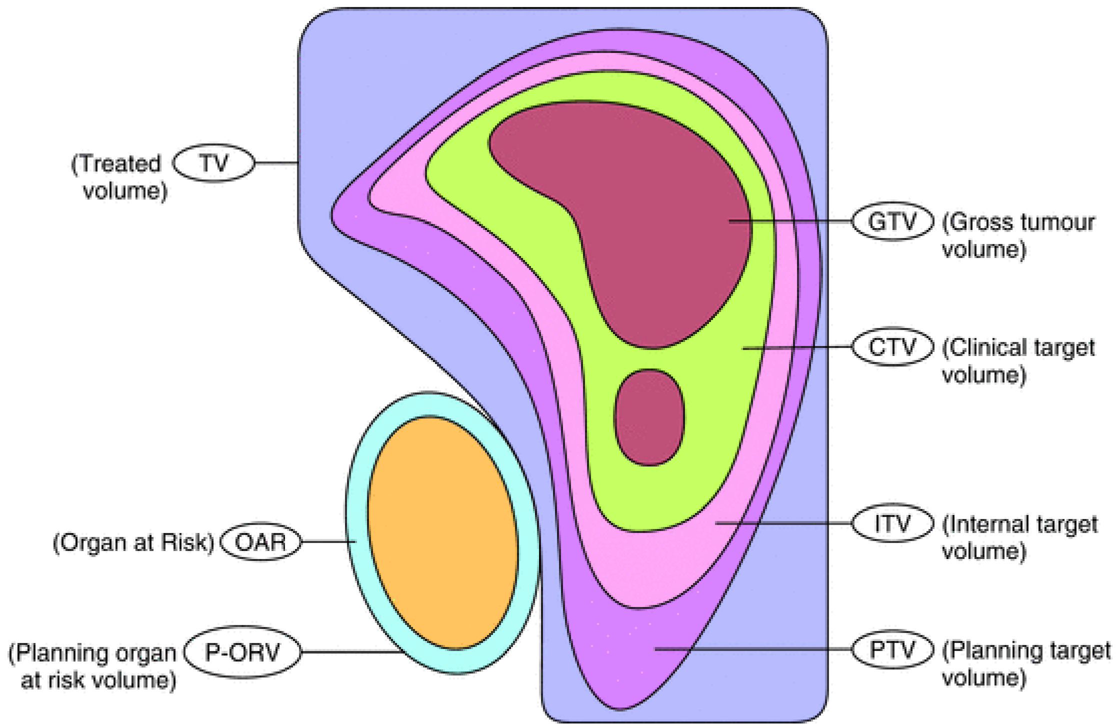
   It is the volume that can be seen by the eye either by the patient himself or with the help of imaging.

2. **CTV - CLINICAL TARGET VOLUME**

   It is the combination of GTV with the surrounding microscopic tumor infiltration. It is a tissue volume that contains the GTV and/or subclinical malignant disease at certain probability level.

3. **PTV - THE PLANNING TARGET VOLUME**

   It is a geometrical concept, introduced for treatment planning. The PTV surrounds the CTV with an additional margin to compensate for the different types of variations and uncertainties of the beam relative to the CTV. The PTV is used to select the appropriate beam sizes and beam arrangements to ensure that the prescribed dose is actually delivered to all parts of the CTV.



MATHEMATICAL FORMULATION:

Linear Regression with Multiple Features and Gradient Descent Optimization:

In this project we are told to use gradient descent optimization method on dataset involving one response (i.e., relative volume of organ receiving particular dose value, or y) and multiple variable (i.e., volume of organ, PTV receiving 60Gy radiation, and PTV receiving 44Gy radiation or xi).

We are using a popular machine learning technique known as Multiple Linear Regression. This algorithm is used for predictive analysis. In classical Statistics and Machine Learning, Regression means a technique to model the relationship between a dependent variable and a given set of independent variables. It can be divided into

1. Simple linear Regression
2. Multiple Linear Regression

Multiple Linear Regression attempts to model the relationship between two or more features and a response by fitting a linear equation to observed data.

x_i y

Features			Output
Organ volume	PTV60	PTV44	Relative volume of organ receiving dose of 900
0.186214	97.1	97.2	73.82
0.043601	109.1	389.9	96.31
0.020247	112.8	202.5	100
0.079899	139.7	317.9	100
0.027489	104.2	269.2	100
0.102716	120	202.5	90.04

← data

There is only one true model which fits our data perfectly. But unfortunately, we don't know that model. So, we have to predict that model using our Machine Learning Algorithm. We make a hypothesis, tuning it until we can check if it is close enough to the true model or it is a wrong one.

Hypothesis Function:

In Multivariate Linear Regression, the hypothesis function with multiple variables x and parameters θ is denoted below.

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$
$$Parameters: \theta = \{\theta_0, \theta_1, \theta_2, \ldots \ldots \ldots, \theta_n\}$$
$$Features: x = \{x_0, x_1, x_2, \ldots \ldots \ldots, x_n\}$$

We assume $x_0 = 1$ for convenience of notation. Simply, relative volume of organ receiving particular dose value (y) = f (volume of organ(x_1), PTV receiving 60Gy radiation(x_2), PTV receiving 44Gy radiation(x_3))

n = number of features

x^i = input (features) of i^{th} training example

x^i_j = value of feature j in i^{th} training example

When tuning the hypothesis, our model learns parameters θ which makes hypothesis function a 'good' predictor. A Good predictor means the hypothesis is closed enough to the true model. Here comes the concept of Cost function.

Cost Function:

We can measure the accuracy of our model using cost function. We define cost function using mean square error (MSE).

$$Hypothesis: h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$
$$Parameters: \theta = \{\theta_0, \theta_1, \theta_2, \ldots \ldots \ldots, \theta_n\}$$
$$Cost\ Function: J(\theta_0, \theta_1, \theta_2, \ldots \ldots \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

We square the difference between hypothesis and prediction to make the error positive. This Cost function should be differentiable so that we can apply Gradient Descent Algorithm to it. Essentially, the cost function J(θ) is the sum of the square error of each data. The larger the error, the worse the performance of the hypothesis. Therefore, we want to minimize the error, that is, minimize the J(θ).

Gradient Descent Algorithm:

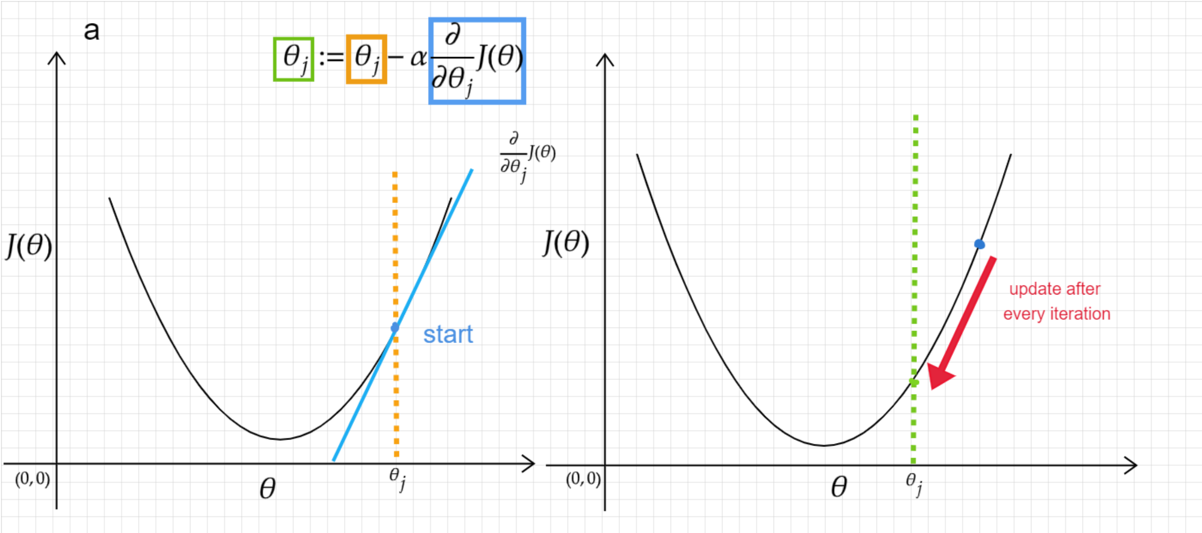
Gradient Descent is an optimization algorithm for finding a local minimum of a differentiable function. Gradient descent is simply used to find the values of a function's parameters (coefficients) that minimize a cost function as far as possible.

We use this algorithm to estimate the parameters value in such a way so that it minimizes the cost function. When we found the minimum error, our model learns the best value of parameters. Thus, we can predict more accurate values for new input data.

At each iteration, the parameters needed to be updated simultaneously.

$$Repeat\ until\ converge\ \{\ \theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta} J(\theta)\}$$

where j represents feature index number.



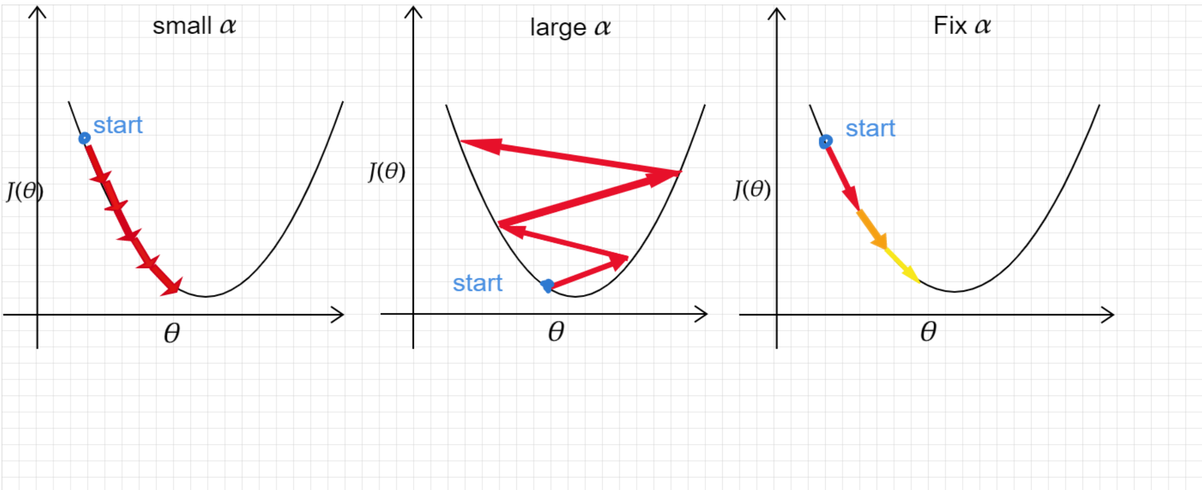
In figure (a), the starting point is at orange (θj, J(θj)). Calculate its partial differentiation, then multiplied it by a learning rate α and the updated result is at green (θj, J(θj)), as figure (b) shows.

Gradient Descent:

$$\theta_j = \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x^i$$

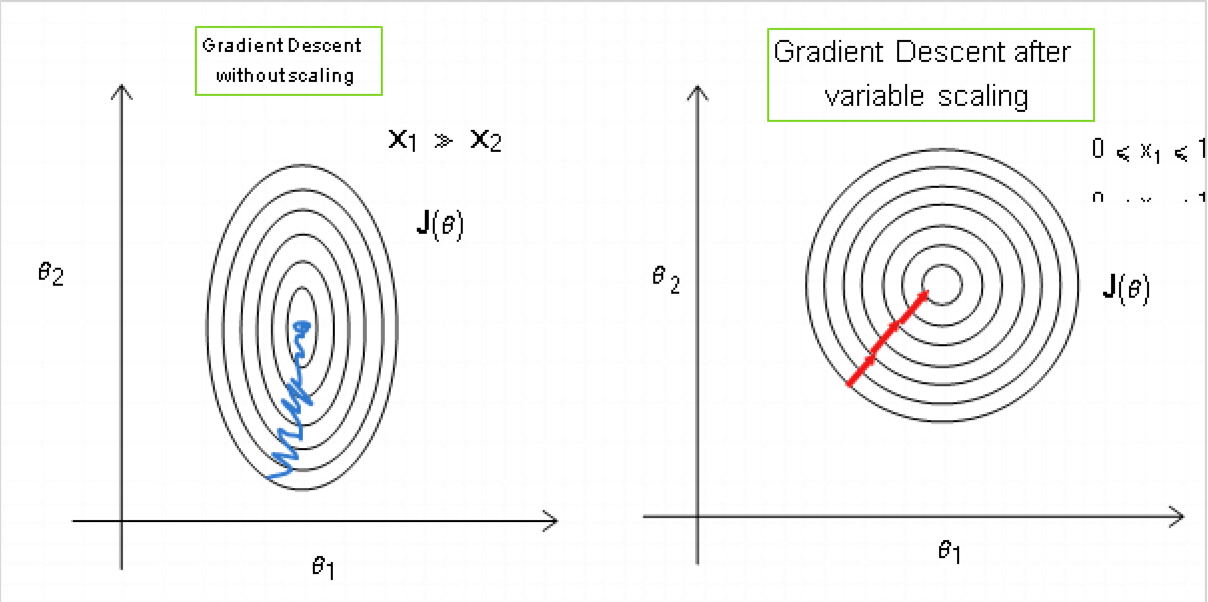
We use learning rate α to control how much we update at one iteration. If α is too small, it makes the gradient descent update too slow, whereas the update may overshoot the minimum and won't converge.

Note that we set a fixed learning rate α in the beginning since the gradient descent will update slowly and automatically until it reaches the minimum. Hence, there is no need to change the learning rate α at each iteration by ourselves.

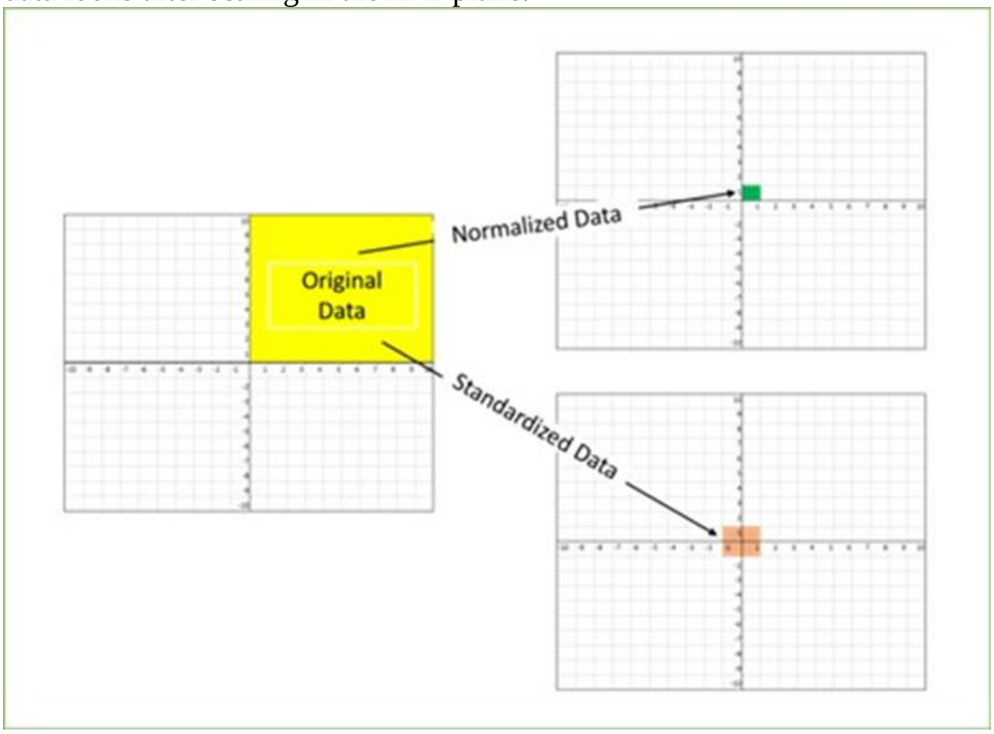


Feature Scaling:

Feature scaling is a method used to normalize the range of independent variables or features of data.



The most common techniques of feature scaling are Normalization and Standardization. Normalization is used when we want to bound our values between two numbers, typically, between [0,1] or [-1,1]. While Standardization transforms the data to have zero mean and a variance of 1, they make our data unitless. Refer to the below diagram, which shows how data looks after scaling in the X-Y plane.



We can speed up gradient descent by scaling because θ descends quickly on small ranges and slowly on large ranges, and oscillates inefficiently down to the optimum when the variables are very uneven.

Also, we can use Mean Normalization in Feature scaling.

$$x' = \frac{x - average(x)}{max(x) - min(x)}$$

where x is an original value, x' is the normalized value.

We can also use z-score normalization or standardization as we used here.

$$x' = \frac{x - mean(x)}{std(x)}$$

This will give us a gaussian distribution of the features which makes the algorithm computationally efficient.

Here, we will implement gradient descent optimization method to fit the dataset using linear regression model

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

wherein θ_0, θ_1, θ_2 and θ_3 need to be optimized to get a good fitting function for the given dataset. Hence, we define a predictive function ($h\theta(x)$)

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

$$h(\theta) = [\theta_0 \quad \theta_1 \quad \theta_2 \quad \theta_3] * \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Now, we will provide a guess value of $\theta_{old} = [\theta 0 \quad \theta 1 \quad \theta 2 \quad \theta 3]_{old}$ and perform an iteration till objective function minimizes to minimum value. Let us define an objective function as

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

To update the new value of $\theta_{new} = [\theta_0 \quad \theta_1 \quad \theta_2 \quad \theta_3]_{new}$, we will implement following steps

$$[\theta_{new}] = [\theta_{old}] - \alpha \begin{bmatrix} \frac{\partial}{\partial\theta_0}J(\theta) \\ \frac{\partial}{\partial\theta_1}J(\theta) \\ \frac{\partial}{\partial\theta_2}J(\theta) \\ \frac{\partial}{\partial\theta_3}J(\theta) \end{bmatrix}$$

$$\begin{bmatrix} \theta_{0,new} \\ \theta_{1,new} \\ \theta_{2,new} \\ \theta_{3,new} \end{bmatrix} = \begin{bmatrix} \theta_{0,old} - \alpha\frac{\partial}{\partial\theta_0}J(\theta) \\ \theta_{1,old} - \alpha\frac{\partial}{\partial\theta_1}J(\theta) \\ \theta_{2,old} - \alpha\frac{\partial}{\partial\theta_2}J(\theta) \\ \theta_{3,old} - \alpha\frac{\partial}{\partial\theta_3}J(\theta) \end{bmatrix}$$

$$\begin{bmatrix}\theta_{0,new}\\ \theta_{1,new}\\ \theta_{2,new}\\ \theta_{3,new}\end{bmatrix} = \begin{bmatrix}\theta_{0,old} - \frac{\alpha}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^i\\ \theta_{1,old} - \frac{\alpha}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^i\\ \theta_{2,old} - \frac{\alpha}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^i\\ \theta_{3,old} - \frac{\alpha}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^i\end{bmatrix}$$

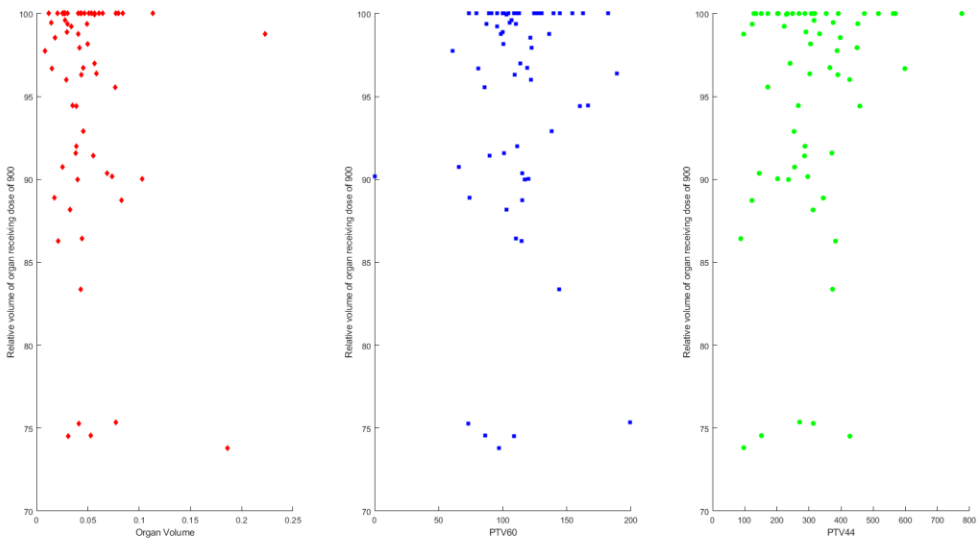
Next, $\theta_{\text{old}} = \theta_{\text{new}}$ and above iteration will continue till the termination condition reached (i.e., no of iteration or difference of cost).

RESULTS AND DISCUSSION:

A particular organ received a dose of particular volume and gradient descent method was used to analyze the relation between relative volume of organ receiving a particular dose with 3 different features.

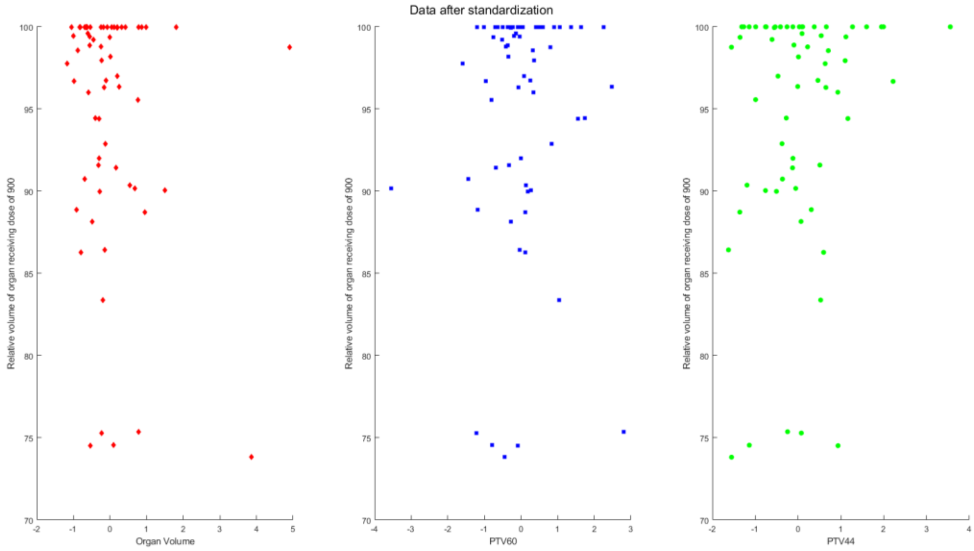
(A) We loaded and plotted the individual features to visualize our dataset. We can see the features are not in scale with each other.

```
%% Load data

data = xlsread('Group-2_data.xlsx');
x = data(:,1:3);
y = data(:,4);

%% Plot data

figure(1)
subplot(1,3,1)
scatter(x(:,1), y,'diamond','filled','MarkerFaceColor','r');
xlabel('Organ Volume')
ylabel('Relative volume of organ receiving dose of 900')
subplot(1,3,2)
scatter(x(:,2), y,'square','filled','MarkerFaceColor','b');
xlabel('PTV60')
ylabel('Relative volume of organ receiving dose of 900')
subplot(1,3,3)
scatter(x(:,3), y,'filled','MarkerFaceColor','g');
xlabel('PTV44')
ylabel('Relative volume of organ receiving dose of 900')
```



Hence, as mentioned above we normalized the data using z score normalization. The data is now plotted and we see the features are centered at mean 0.

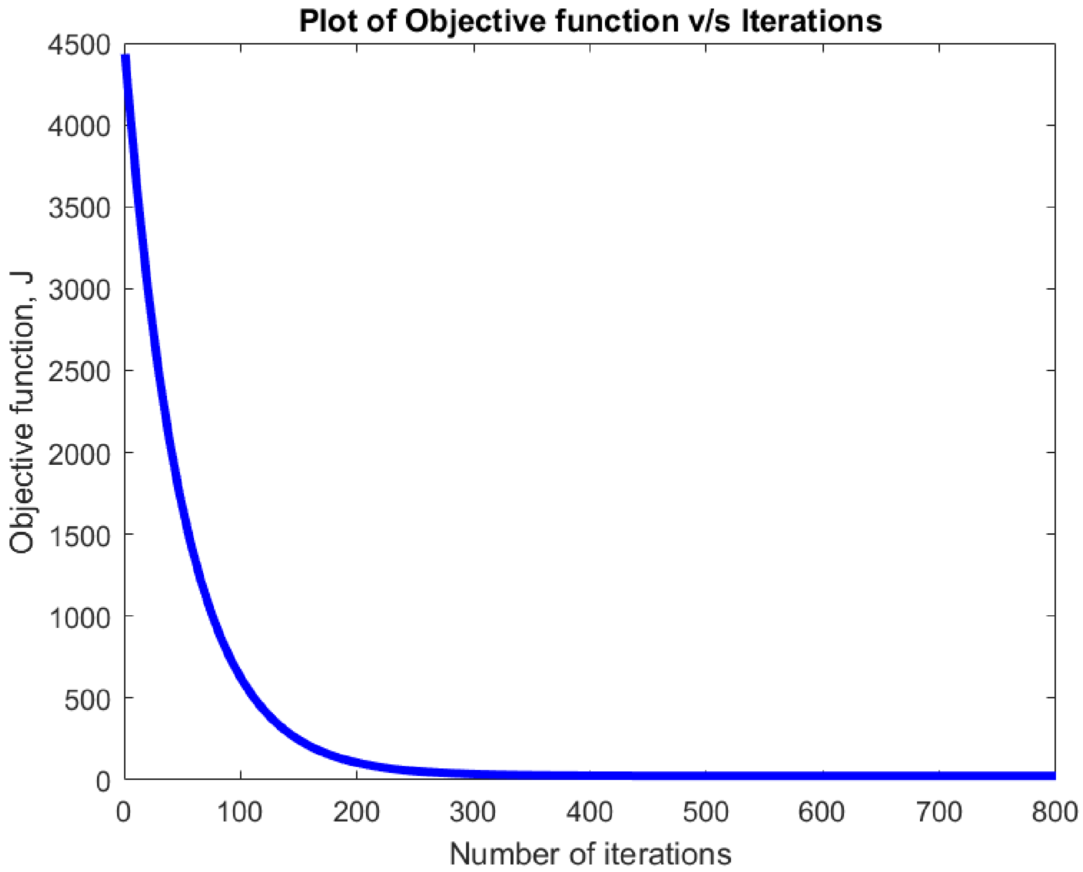
```
%% Normalisation

norm_mean = mean(x); % mean of dataset columns
norm_std = std(x); % standatd deviation of dataset columns
x = (x-norm_mean)./norm_std; % z-score normalisation
m = length(y); % lenght of dataset
x = [ones(m,1),x]; % adding x0 column

%% Plot Standardised data

figure(2)
subplot(1,3,1)
scatter(x(:,2), y,'diamond','filled','MarkerFaceColor','r');
xlabel('Organ Volume')
ylabel('Relative volume of organ receiving dose of 900')
subplot(1,3,2)
scatter(x(:,3), y,'square','filled','MarkerFaceColor','b');
xlabel('PTV60')
ylabel('Relative volume of organ receiving dose of 900')
subplot(1,3,3)
scatter(x(:,4), y,'filled','MarkerFaceColor','g');
xlabel('PTV44')
ylabel('Relative volume of organ receiving dose of 900')
sgtitle('Data after standardization')
```



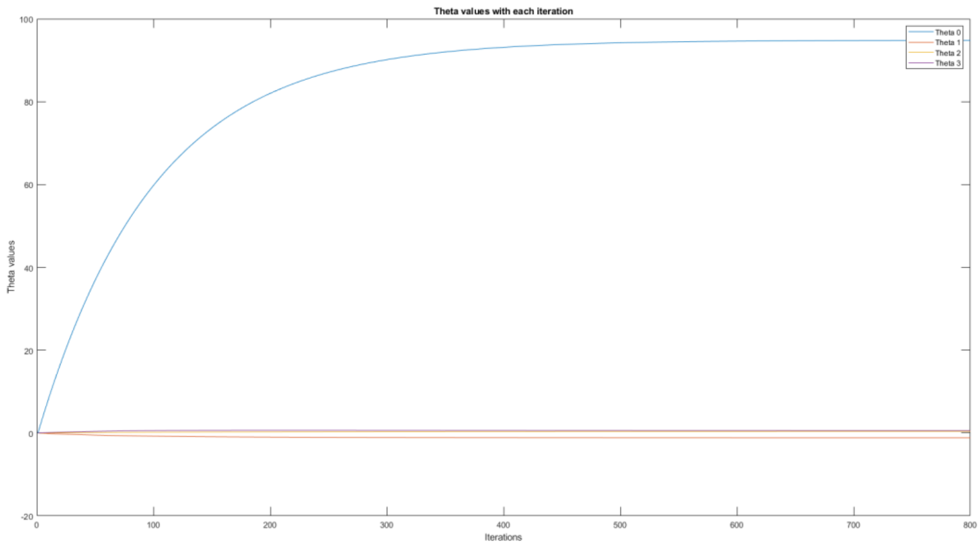
An iteration of 800 with alpha value as low as 0.01 confirmed the total convergence of objective function.

```
%% Parameters

alpha = 0.01; % Learning rate
theta = [0;0;0;0]; % Initialized theta values

% This program includes an optimization technique for no. of iterations
% The loop will terminate whichever condition satisfies first i.e...
% ...number of iterations or cost difference between iterations

iterations = 800; % preferred number of iterations
cost_difference = 0.01; % preferred cost difference to terminate
iteration
count = 1; % actual number of iteration happened

%% Gradient descent

cost = (0.5*m)*sum((x*theta-y).^2); % Cost function
cost_list(1) = cost; % Cost value for each iteration as an array
theta_list(1,:) = theta; % Theta value in each iteration as ix4 matrix

% Loop starts

while true
    theta = theta - (alpha/m).*(x.'*(x*theta-y)); % update theta
    cost_temp = (0.5*m)*sum((x*theta-y).^2); % calculate cost
    count = count+1;
    theta_list(count,:) = theta; % add theta to list
    cost_list(count) = cost_temp; % add cost to list
    if (cost - cost_temp) <= cost_difference
        break; % loop terminate condition based on cost difference
    end
    cost = cost_temp;
    if count >= iterations
        break; % loop terminate condition after desired iterations
    end
```

The relation of objective function with respect to number of iterations is plotted and shown as:



Moreover, the theta values namely theta 0, theta 1, theta 2, theta 3 were found and a converging pattern was observed using the code once gradient function was called.



Also, by seeing the training and testing data set the output y is also somehow is in the range of training data y.

(B) The output y can be seen after running code as:

```
%% Load test data

test_data = xlsread('Test_data.xlsx');

%% Standardize test data

test = (test_data-norm_mean)./norm_std;
test = [ones(length(test),1),test];
```

Now, we apply the theta we found using the training data to the standardized test data. We can find the desired result with that.

```
%% Apply theta to test data

result = test*theta;
final_result =
table(test_data(:,1),test_data(:,2),test_data(:,3),result,...
    'VariableNames', {'Organ Volume','PTV60','PTV44',...
    'Relative volume of the organ receiving the dose'});
```

We found the result of our test data as follows:

Organ Volume	PTV60	PTV44	Relative volume of the organ receiving the dose
0.0577689240000000	86.3000000000000	100.400000000000	93.3193422067384
0.0598035030000000	134.300000000000	234.100000000000	94.4547552607530
0.0391286810000000	93	247.900000000000	94.6628071025188
0.0508303980000000	111.100000000000	198.700000000000	94.2969208024558
0.0380401700000000	141.200000000000	328.600000000000	95.6737954401485
0.0422764230000000	154.300000000000	492	96.4132982671312
0.0605478140000000	133.100000000000	208.100000000000	94.3016757250464
0.0734149050000000	88.6000000000000	89.9000000000000	92.7865159841154
0.0362637360000000	89.1000000000000	273	94.8159771717215
0.0969637610000000	93.2000000000000	102.100000000000	92.1209433925459

CONCLUSION:

Our main aim in this project was to successfully applying iterative gradient descent optimization algorithm, and it was completed and gradient descent reaches to its global minima successfully. At the end of **300** iterations, we reached to the minimum cost value with learning rate ($\propto$) **0.01** and initial (θ) values are **$(1,1,1,1)^T$.**

The parameter (θ) which we obtained are **$(94.800366, -1.169533, 0.407477, 0.577144)^T$** . Using those optimized (θ) values, the relative volume of organ receiving the particular dose for the new patient data is successfully predicted with minimum error.

CITATION:

Mukherji A. (2018) Planning a Patient, Deciding on the Volumes and Fields and Plan Verification. In: Basics of Planning and Management of Patients during Radiation Therapy. Springer, Singapore. https://doi.org/10.1007/978-981-10-6659-7_9