LAB ACTIVITY 4: Regularized Linear and Logistic Regression

DANTING, Jules Raphael
DE VERA, Luis Paolo
LIM, Alyssa Raphaella
RAMOS, Brian
VALDEZ, Esteen Rae

1 INTRODUCTION

The main objectives for this experiment is to observe and understand the implementation of regularization in linear regression and logistic regression. In addition to apply it in practical predictions such as the number of passengers that survived the Titanic tragedy. The experiment is divided into three part to see the changes that regularization can do to the different types of regression and application. Regularization is used to reduce the overfitting of data which implies that it can constrain the optimization, which can reduce the non-linearity of the graph [1]. It can then simplify the model which can fit the training data more accurately. Overfitting occurs when the graph is precisely getting the data points but does not give an output that can predict that outcoming data well [2]. Regularization techniques will further reduce the errors due to underfitting and overfitting by reducing the cost function with the control variable lambda and its fitting parameters. In doing so normal equation is used to determine the parameters that best fit the regularized linear regression while it's the hessian and newton method for the regularized logistic regression.

2 PROCEDURE

Procedure 4.1 Plot the data
● Load the data files "ml4Linx.dat" and "ml4Liny.dat" into your program.
● These correspond to the "x" and "y" variables that you will start out with.
● Notice that in this data, the input "x" is a single feature, so you can plot y as a function of x on a 2-dimensional graph.

Procedure 4.2
● Using the Normal equation, find values for θ using the three regularization parameters below:
a. λ = 0 (this is the same case as non-regularized linear regression)
b. λ = 1
c. λ = 10

Procedure 4.3 Plot the polyomial fit for each value of λ.
● When you have found the answers for θ, verify them with the values in the solutions.
● In addition to listing the values for each element θ_j of the θ vector, we will also provide the 2-norm of θ so you can quickly check if your answer is correct.
● In Octave, you can calculate the L2-norm of a vector x using the command norm(x).

Question 1
● From looking at the previous graphs, what conclusions can you make about how the regularization parameter λ affects your model?

Procedure 4.4
● After loading the data, plot the points using different markers to distinguish between the two classifications. The commands in Matlab/Octave will be:

```
x = load('ml4Logx.dat');
y = load('ml4Logy.dat');
figure
% Find the indices for the 2 classes
pos = find(y);
neg = find(y == 0);
plot(x(pos, 1), x(pos, 2), '+')
hold on
plot(x(neg, 1), x(neg, 2), 'o')
```

Procedure 4.5
● Run Newton's Method using the three values of lambda below:
a. λ = 0 (this is the same case as non-regularized logistic regression)
b. λ = 1
c. λ = 10

Procedure 4.6
● Print out the value of J(θ) during each iteration.
● J(θ) should not be decreasing at any point during Newton's Method.
● If it is, check that you have defined J(θ) correctly.
● Also check your definitions of the gradient and Hessian to make sure there are no mistakes in the regularization parts.

Procedure 4.7
● After convergence, use your values of theta to find the decision boundary in the classification problem.
● The decision boundary is defined as the line where

$$P(y=1|x;\theta) = 0.5 \qquad \Longrightarrow \qquad \theta^T x = 0$$

Procedure 4.8
● Finally, because there are 28 elements θ, we will not provide an element-by-element comparison in the solutions.
● Instead, use norm(theta) to calculate the L2-norm of θ, and check it against the norm in the solutions.

3 RESULTS AND DISCUSSION

PART I REGULARIZED LINEAR REGRESSION

Procedure 4.1 Data Plot

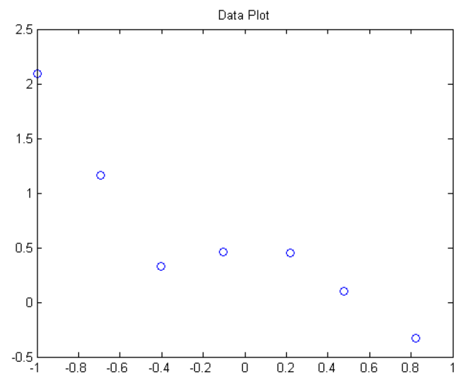

Fig. 1. Plot of Data

As seen in Figure 1, the data loaded to the variables x and y are plotted. Each value of lambda is consist of 6 elements.

Procedure 4.2 Theta Values for each lambda

Theta for lambda =0
0.4725
0.6814
-1.3801
-5.9777
2.4417
4.7371
Theta for lambda =1
0.3976
-0.4207
0.1296
-0.3975
0.1753
-0.3394
Theta for lambda =10
0.5205
-0.1825
0.0606
-0.1482
0.0743
-0.1280

The group used the normal equation with 5^{th} order in order to regularize the linear regression. With the use of the 6 training data in the given 3 values of lambda namely, lambda = 0, 1, and 10. Using the training data and applying the normal equation with each set of data, the group come up with the graphs shown in Figures 2 – 4.

Procedure 4.3 Plot with polynomial fit for each value of lambda

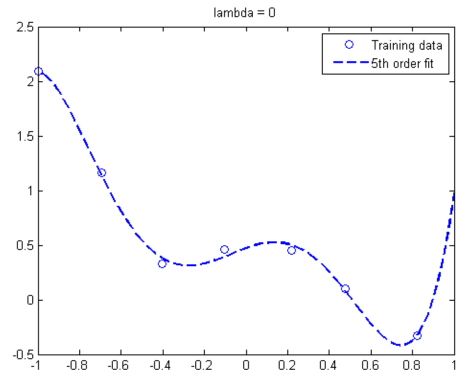

Fig. 2. Plot of Data using lambda = 0

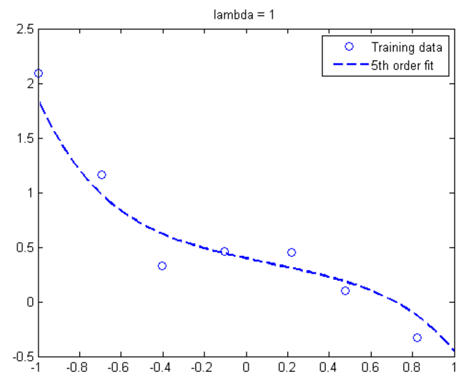

Fig. 3. Plot of Data using lambda = 1

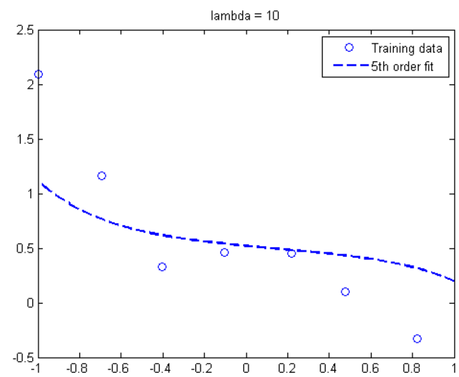

Fig. 4. Plot of Data using lambda = 10

As observed in Figures 2 – 4, while the value of the lambda increases, the 5^{th} order fit of the graph doesn't meet the data loaded in x and y, of the first part of the experiment. The lambda with the value of zero fits the data loaded in x and y variables perfectly.

MATLAB Code

```
%Experiment 4 - Regularization Linear Regression
clc;
close all;
x = load('ml4Linx.dat');
y = load('ml4Liny.dat');
m = length(y);

%Procedure 4.1 - Plot Data
```

```
    figure, plot(x,y,'o');
    title('Data Plot');
    x_orig = x;

    %Hypothesis has a 5th order polynomial
    x = [ones(m, 1), x, x.^2, x.^3, x.^4, x.^5];

    %Initialization of fitting parameters
    theta = zeros(6,1);

%Procedure 4.2
    %Regularization parameter
    lambda = 0; %Non Regularized Linear Regression
    % Closed form solution from normal equations
    L = lambda.*eye(6); % lambda .* identity matrix
6x6
    L(1) = 0; % diagonal matrix with a zero in the
upper left
    theta_L0 = ((x' * x + L)^-1) * (x' * y)
    %or theta_L0 = (x' * x + L)\x' * y %L = 6x6
zero matrix

    lambda = 1;
    L = lambda.*eye(6);
    L(1) = 0;
    theta_L1 = (x' * x + L)\x' * y %L = 6x6 zero
matrix

    lambda = 10;
    L = lambda.*eye(6);
    L(1) = 0;
    theta_L10 = (x' * x + L)\x' * y

%Procedure 4.3 - Plotting of the polynomial fit for
each lambda
    %calculate the L2-norm of verctor x
    norm_theta_L0 = norm(theta_L0);
    norm_theta_L1 = norm(theta_L1);
    norm_theta_L10 = norm(theta_L10);

    % Plot the linear fit for lambda = 0
    figure, plot(x_orig,y,'o');
    hold on;
    x_vals = (-1:0.025:1)';
    %Include other powers of x in feature vector x
    features = [ones(size(x_vals)), x_vals,
x_vals.^2, x_vals.^3,...
              x_vals.^4, x_vals.^5];
    plot(x_vals, features*theta_L0, '--',
'LineWidth', 2)
    title('lambda = 0')
    legend('Training data', '5th order fit')
    hold off

    % Plot the linear fit for lambda = 1
    figure; plot(x_orig,y,'o');
    hold on;
    x_vals = (-1:0.025:1)';
    %Include other powers of x in feature vector x
    features = [ones(size(x_vals)), x_vals,
x_vals.^2, x_vals.^3,...
              x_vals.^4, x_vals.^5];
    plot(x_vals, features*theta_L1, '--',
'LineWidth', 2)
    title('lambda = 1')
    legend('Training data', '5th order fit')
    hold off

    % Plot the linear fit for lambda = 10
    figure; plot(x_orig,y,'o');
    hold on;
    x_vals = (-1:0.025:1)';
    %Include other powers of x in feature vector x
    features = [ones(size(x_vals)), x_vals,
x_vals.^2, x_vals.^3,...
              x_vals.^4, x_vals.^5];
    plot(x_vals, features*theta_L10, '--',
'LineWidth', 2)
    title('lambda = 10')
    legend('Training data', '5th order fit')
    hold off
```

PART II REGULARIZED LOGISTIC REGRESSION

Procedure 4.4 Data Plot

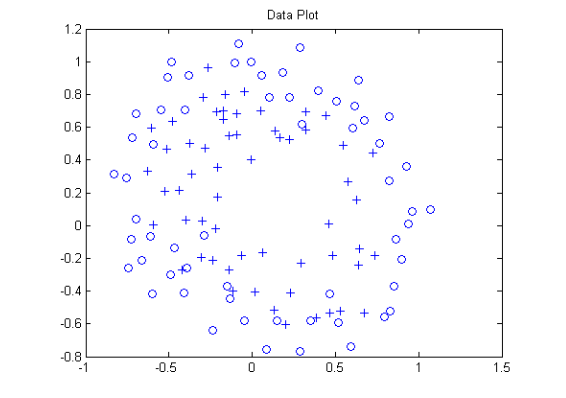

Fig. 5. Plot of Data

Procedure 4.5 & 4.6 J values using Newton's Method on each Lambda Values

J @ lambda = 0
0.6931
0.3491
0.3044
0.2869
0.2655
0.2496
0.2403
0.2288
0.2073
0.2016
0.2003
0.1999
0.1998
0.1998
0.1998

J @ lambda = 1
0.6931
0.5297
0.5247
0.5246
0.5246
0.5246
0.5246
0.5246
0.5246
0.5246
0.5246
0.5246
0.5246
0.5246
0.5246

J @ lambda = 10
0.6931
0.6476
0.6476
0.6476
0.6476
0.6476
0.6476
0.6476
0.6476
0.6476

0.6476
0.6476
0.6476
0.6476
0.6476

Same with the previous experiment with the normal equation, the group used the following data in the different values of lambda in order to regularize the linear regression. But, instead of using the normal equation, the group used the Newton's Equation. The group plot the given 28 data, as seen in Figures 6 – 8, where the group plot the decision boundary in each value of lambda.

Procedure 4.7 Plot Decision Boundary

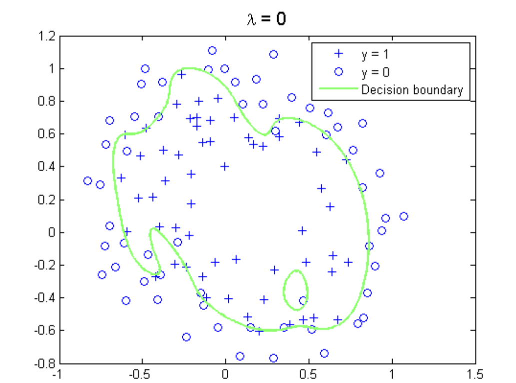

Fig. 6. Plot of Data with Decision Boundary using lambda = 0

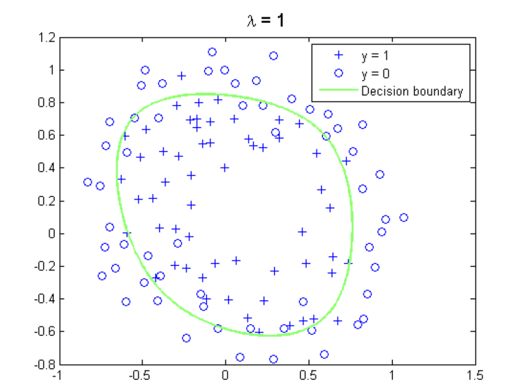

Fig. 7. Plot of Data with Decision Boundary using lambda = 0

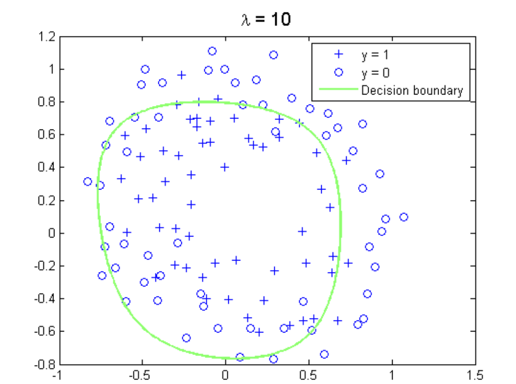

Fig. 8. Plot of Data with Decision Boundary using lambda = 10

As shown in Figures 6 – 8, while the value of the lambda increases, the fit of the curve also improves. In this experiment the lambda with the value of 10 shows the best fit-curve in this linear regression.

Procedure 4.8 L2 norm of Theta

norm_theta @ lambda 0 = 7.1727e+003
norm_theta @ lambda 1 = 4.2400
norm_theta @ lambda 10 = 0.9384

Using norm(theta) function, the group calculated the theta of each value of lambda.

MATLAB Code

```
%Experiment 4 - Regularization Logistic Regression
clc;
close all;
x = load('ml4Logx.dat');
y = load('ml4Logy.dat');
m = length(y);

%Procedure 4.4 - Plot Data

    figure
    pos = find(y);
    neg = find(y==0);
    plot(x(pos,1),x(pos,2),'+')
    hold on
    plot(x(neg,1),x(neg,2),'o')
    title('Data Plot')

    figure
    pos = find(y);
    neg = find(y==0);
    plot(x(pos,1),x(pos,2),'+')
    hold on
    plot(x(neg,1),x(neg,2),'o')

    %Use map_feature.m file - maps the original
inputs to the feature vector.
    x = map_feature(x(:,1), x(:,2));
    [m, n] = size(x);
    theta = zeros(n, 1); %Initialization of fitting
parameters

%Procedure 4.5 - Newton's Method using values of
lambda
    %Sigmoid Function
    g = inline('1.0 ./ (1.0 + exp(-z))');

    MAX_ITR = 15;
    J = zeros(MAX_ITR, 1);

    lambda = 1;
    for i = 1:MAX_ITR
        % Calculate the hypothesis function
        z = x * theta;
        h = g(z);

        % Logistic Regression Cost Function with a
regularization term
        % Cost Function: J(i) =(1/m)*sum(-y.*log(h)
- (1-y).*log(1-h))
        % Regularization term:
lambda/(2*m))*norm(theta([2:end]))^2;
        J(i) =(1/m)*sum(-y.*log(h) - (1-y).*log(1-
h))+ ...
        (lambda/(2*m))*norm(theta([2:end]))^2;

        % Calculate gradient and hessian.
        G = (lambda/m).*theta; G(1) = 0; % extra
term for gradient
        L = (lambda/m).*eye(n); L(1) = 0;% extra
term for Hessian
        grad = ((1/m).*x' * (h-y)) + G; %Gradient
with extra term
        H = ((1/m).*x' * diag(h) * diag(1-h) * x) +
L; %Hessian with extra term

        % Using update rule
        theta = theta - H\grad;
```

```
    end

%Procedure 4.6 - Print Value of J
    J

%Proecdure 4.7 - Plotting of Decision Boundary
    norm_theta = norm(theta)

    % Define the ranges of the grid
    u = linspace(-1, 1.5, 200);
    v = linspace(-1, 1.5, 200);
    % Initialize space for the values to be plotted
    z = zeros(length(u), length(v));
    % Evaluate z = theta*x over the grid
    for i = 1:length(u)
        for j = 1:length(v)
            z(i,j) = map_feature(u(i), v(j))*theta;
        end
    end
    % Because of the way that contour plotting
works
    % in Matlab, we need to transpose z, or
    % else the axis orientation will be flipped!
    z = z';
    % Plot z = 0 by specifying the range [0, 0]
    contour(u, v, z, [0, 0], 'LineWidth', 2)
    legend('y = 1', 'y = 0', 'Decision boundary')
    title(sprintf('\\lambda = %g', lambda),
'FontSize', 14)

    hold off
```

PART III Real World Application – Predicting Titanic Survivors

Features Used to Solve the Problem:
x-data = Age & Fare
y-data = Survived

Data Plot

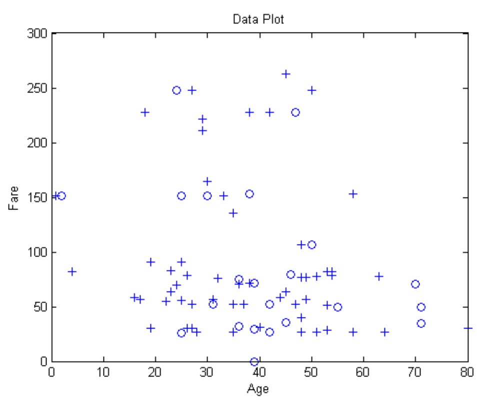

Fig. 6. Plot of 80 Data using Fare and Age

During the implementation process, there were warnings from the matlab indicating that the matrix is close to singular or badly scaled on the Newton's Method part of the code. This resulted to J values of NaN making the system unable to plot the decision boundary.

4 ANALYSIS & CONCLUSION

In this experiment, the group become familiar with regularization. The experiment is divided into 2 parts which is regularized linear regression and regularized logistic regression. An additional task is done which is using regularization in order to predict the survivors in the Titanic tragedy. Using the regularization technique, errors will be reduced but it will never be removed. Regularization can avoid underfitting and overfitting that can cause the predictions to be wrong or miscalculated. Normal equation is also used in the experiment to know the parameters that can be best fit the regularized regression.

5 REFERENCES

[1] (2014). Logistic Classifier Overfitting and Regularization. [Online]. Avaliable: http://www.codeproject.com/Articles/824680/Logistic-Classifier-Overfitting-and-Regularization
[2] Genevieve Orr. Neural Networks. [Online]. Available: https://www.willamette.edu/~gorr/classes/cs449/overfitting.html