Regularization and Titanic

LBYCP29 – Laboratory 4

Gervin Guevarra, Allen Koizumi, Nicholas Moreno, Charleston Uy

Department of Electronics and Communications Engineering

Gokongwei College of Engineering, De La Salle University

Manila, Philippines

*Abstract*—This laboratory report presents the implementation of regularized linear and logistic regression on a multi-feature data. The regularization further improves the accuracy of the fitted line, thus making it more suitable for machine analysis and prediction.

Keywords—Logistic Regression, binary classification

# Introduction

The previous lab works involve the use of linear regression and logistic regression. However, linear regression is not always good for approximating the data as it may underfit – a condition where the fitted line does not cover the majority of the points, hence, ineffective for prediction. One solution is to use a higher-order polynomial to fit more variations in the data. The trade-off is that it may cause overfitting, wherein too many points are being captured on the fitted line.

To prevent overfitting, regularization is used on the regression model. Cost function is now computed while taking into consideration the regularization parameter λ. Thus, the right regularization parameter must be determined while also minimizing the cost function, as the cost function increases with the increase on the fitting parameters. Finding the suitable regularization parameters requires the use of normal equations, which yields to Ɵ values for a given λ. These Ɵ values will be used for plotting the fitted line.

(1)

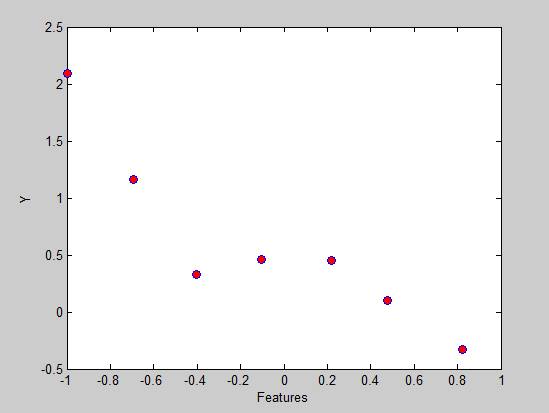
# Objectives

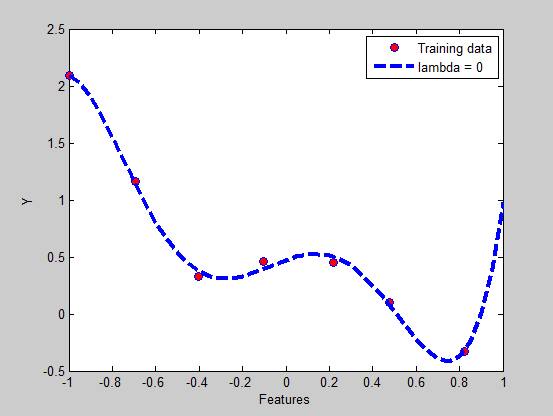
The experiment aims to achieve the following objectives

* To implement regularized linear regression and regularized logistic regression.
* To apply regularized logistic regression to predict which passengers survived the Titanic shipwreck tragedy.

# Data and Results

Part I. Regularized Linear Regression

Figure 1. Training data plot

Figure 2. Lamda = 0

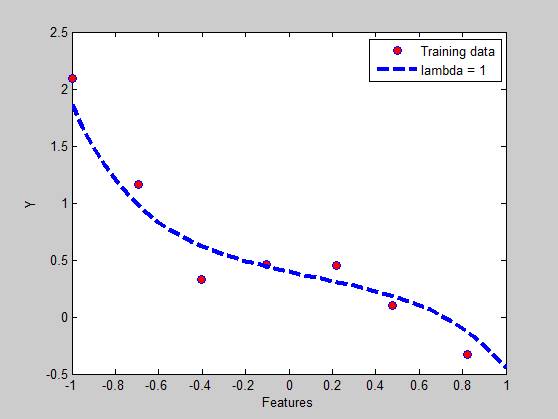


Figure 3. Lambda = 1

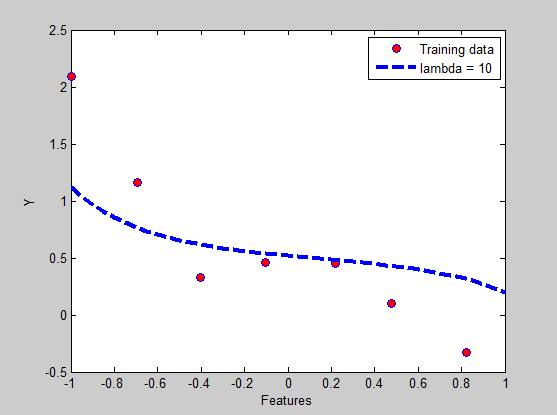


Figure 4. Lambda = 10

Table 1. Theta values and norm\_theta for Lambda = 0

|  |  |
| --- | --- |
| **Theta values (θ)** | **norm\_theta** |
| 0.4725 | 8.1687 |
| 0.6814 |  |
| -1.3801 |  |
| -5.9777 |  |
| 2.4417 |  |
| 4.7371 |  |

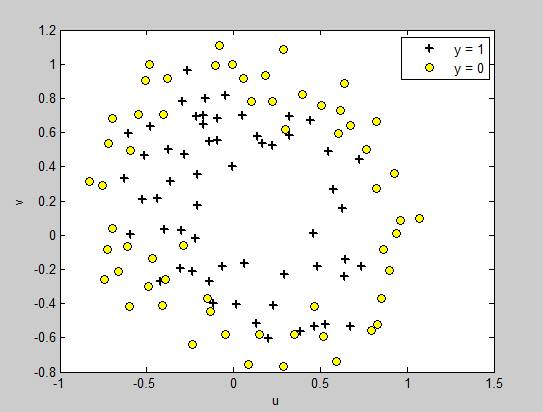
Table 2. Theta values and norm\_theta for Lambda = 1

|  |  |
| --- | --- |
| **Theta values (θ)** | **norm\_theta** |
| 0.3976 | 0.8098 |
| -0.4207 |  |
| -0.1296 |  |
| -0.3975 |  |
| 0.1753 |  |
| -0.3394 |  |

Table 3. Theta values and norm\_theta for Lambda = 10

|  |  |
| --- | --- |
| **Theta values (θ)** | **norm\_theta** |
| 0.5205 | 0.5931 |
| -0.1825 |  |
| 0.0606 |  |
| -0.1482 |  |
| 0.0743 |  |
| -0.1280 |  |

Part II. Regularized Logistic Regression

Figure 5. Regularization logistic regression training data

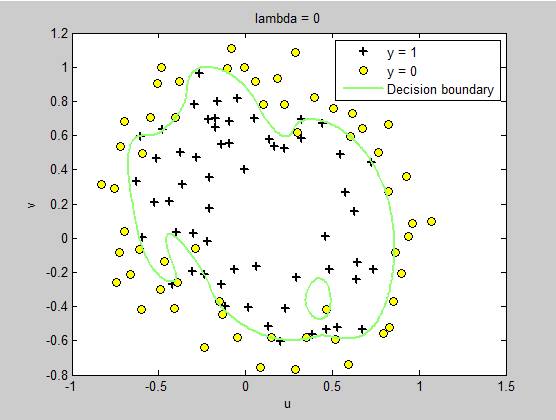


Figure 6. Lambda = 0

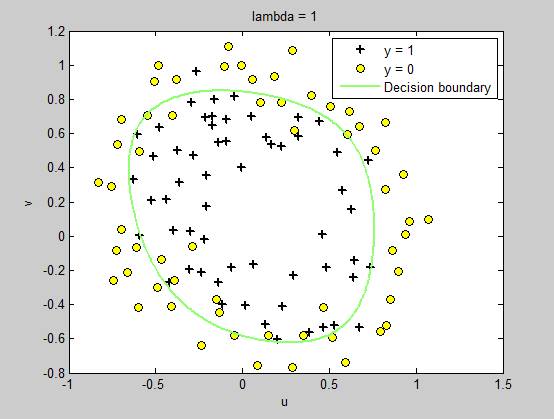


Figure 7. Lambda = 1

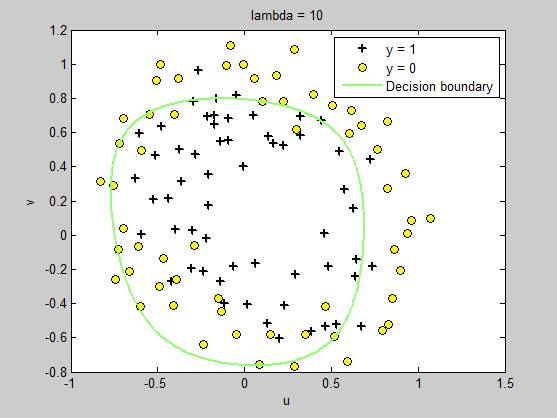


Figure 8 Lambda = 10

Table 4. J(θ) values and norm\_theta for Lambda = 0

|  |  |
| --- | --- |
| **J(θ)** | **norm\_theta** |
| 0.6931 | 7.1727e+03 |
| 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 |  |

Table 5. J(θ) values and norm\_theta for Lambda = 1

|  |  |
| --- | --- |
| **J(θ)** | **norm\_theta** |
| 0.6931 | 4.2400 |
| 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 |  |

Table 6. J(θ) values and norm\_theta for Lambda = 10

|  |  |
| --- | --- |
| **J(θ)** | **norm\_theta** |
| 0.6931 | 0.9384 |
| 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 |  |

Part III. Real-world application

*Values taken from titanic3.xls: age, fare and survived*

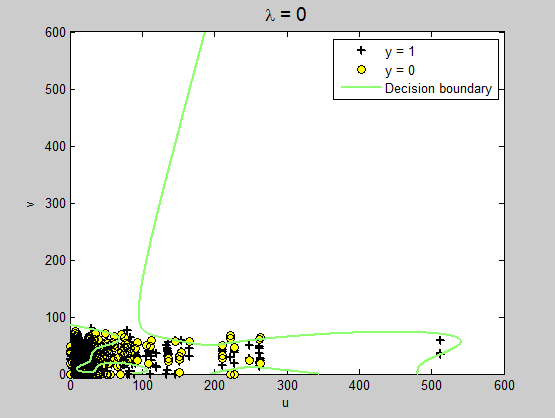


Figure 9. Titanic, Lambda = 0

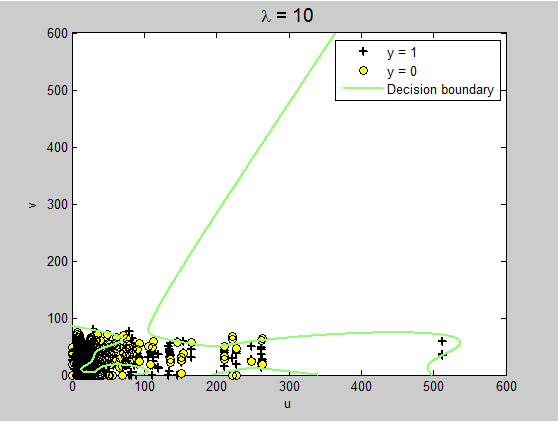


Figure 10. Titanic, Lambda = 10

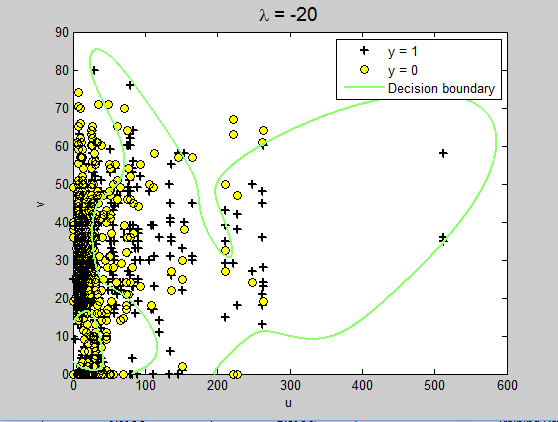


Figure 11, Titanic, Lambda = -20

Table 7. Titanic, Lambda = 0

|  |  |
| --- | --- |
| **Theta values (θ)** | **norm\_theta** |
| 0.6931 | 2.2458 |
| 0.6019 |  |
| 0.5992 |  |
| 0.5981 |  |
| 0.5978 |  |
| 0.5978 |  |
| 0.5978 |  |
| 0.5978 |  |
| 0.5978 |  |
| 0.5978 |  |

Table 8. Titanic, Lambda = 10

|  |  |
| --- | --- |
| **Theta values (θ)** | **norm\_theta** |
| 0.6931 | 2.1950 |
| 0.6023 |  |
| 0.5996 |  |
| 0.5986 |  |
| 0.5984 |  |
| 0.5983 |  |
| 0.5983 |  |
| 0.5983 |  |
| 0.5983 |  |
| 0.5983 |  |

Table 9. Titanic, Lambda = -20

|  |  |
| --- | --- |
| **Theta values (θ)** | **norm\_theta** |
| 0.6931 | 2.5403 |
| 0.6005 |  |
| 0.5972 |  |
| 0.5954 |  |
| 0.5947 |  |
| 0.5947 |  |
| 0.5946 |  |
| 0.5946 |  |
| 0.5946 |  |
| NaN |  |

# Analysis and Conclusion

In figure <1>, the data is approachable with regularized linear regression. As seen in figure <2>, the curve is specific to the data which looks correct but it’s not. The problem here is that it is too specific and is not good at showing a trend. Therefore the figure is overfitting. In figure <4>, the curve does not follow the data and does not show a good trend. The figure <3> shows the right type of fitting. Compared to figure <2>, the overfitting that figure <2> made is now reduced after increasing the lambda to 1. Looking at the results in tables <1, 2 and 3>, we notice that as lambda increases the normal theta decreases and by adjusting lambda we control how the data will fit.

In figure <5>, the data is approachable with regularized logistic regression. In figure <6>, we can observe that the decision boundary tries to precisely fit around the + and o. This is evident as we can see a region inside the boundary that avoided the o. In figure <8>, we can observe that the decision boundary is to large thus showing an inaccurate boundary. In figure <7>, we can notice that the decision boundary is more compact therefore separates the + and o uniformly. Considering the results in tables <6.1, 7.1 and 8.1>, we witness that as the normal theta decreases lambda increases which is similar to regular linear regression.

# Bibliography

|  |  |
| --- | --- |
| [1] | A. Ng, "Spervised Learning," n.d. [Online].  Available: http://cs229.stanford.edu/notes/cs229-notes1.pdf [Accessed 22 September 2015]. |
| [2] | A. Ng, "Machine Learning,"  n.d. [Online]. Available: https://class.coursera.org/ml-003/lecture.  [Accessed 22 September 2015]. |
| [3] | Institute for Digital Research and Edcation, "Stata Data Analysis and  Exampes", 2015 [Online]. Available: http://www.ats.ucla.edu/stat/stata/dae/logit.htm.  [Accessed 22 September 2015]. |

# Appendix

**Regularization Linear Regression**

Length = length(y);

figure;

plot(x, y, 'o', 'MarkerFacecolor', 'y', 'MarkerSize', 5);

x = [ones(Length, 1), x, x.^2, x.^3, x.^4, x.^5];

theta = zeros(size(x(1,:)))';

lambda = 0;

LADA = lambda.\*eye(6);

LADA(1) = 0;

theta = (x' \* x + LADA)\x' \* y

norm\_theta = norm(theta)

hold on;

x\_vals = (-1:0.05:1)';

Feat = [ones(size(x\_vals)), x\_vals, x\_vals.^2, x\_vals.^3,...

x\_vals.^4, x\_vals.^5];

plot(x\_vals, features\*theta, '--', 'LineWidth', 2)

legend('Training data', '5th order fit')

hold off

**Regularization Logistic Regression**

figure

position = find(y); negative = find(y == 0);

plot(x(position, 1), x(position, 2), 'k+','LineWidth', 2, 'MarkerSize', 7)

hold on

plot(x(negative, 1), x(negative, 2), 'ko', 'MarkerFaceColor', 'y', 'MarkerSize', 7)

x = map\_feature(x(:,1), x(:,2));

[m, n] = size(x);

theta = zeros(n, 1);

g = inline('1.0 ./ (1.0 + exp(-z))');

max = 15;

J = zeros(max, 1);

lambda = 0; *%for titanic lambda = -20*

for i = 1:max

z = x \* theta;

h = g(z);

J(i) =(1/m)\*sum(-y.\*log(h) - (1-y).\*log(1-h))+ (lambda/(2\*m))\*norm(theta([2:end]))^2;

G = (lambda/m).\*theta; G(1) = 0;

L = (lambda/m).\*eye(n); L(1) = 0;

grad = ((1/m).\*x' \* (h-y)) + G;

H = ((1/m).\*x' \* diag(h) \* diag(1-h) \* x) + L;

theta = theta - H\grad;

end

J

norm\_theta = norm(theta)

u = linspace(-1, 1.5, 200); *%for titanic linspace(0, 600, 1310);*

v = linspace(-1, 1.5, 200); *%for titanic linspace(0, 600, 1310);*

z = zeros(length(u), length(v));

for i = 1:length(u)

for j = 1:length(v)

z(i,j) = map\_feature(u(i), v(j))\*theta;

end

end

contour(u, v, z, [0, 0], 'LineWidth', 2)

legend('y = 1', 'y = 0', 'Decision boundary')

title(sprintf('\\lambda = %g', lambda), 'FontSize', 14)

xlabel('u');

ylabel('v');

hold off