Regularization

Aquino, Patrica Rose
Basallote, Lawrence Andrew
Carag, Stephanie
Jacinto, Dan Emanuel
Lunasco, Jan Osbert

***Abstract*—Regularization is a way to prevent overfitting and improve generalization by adding constraints or penalty to a specific model. In this paper, both linear and logistic regression will be shown to show the effect of regularization each with their own set of data.**

I. Introduction

In this experiment, the group will implement regularized linear regression and regularized logistic regression.The purpose in regularization is to minimize the cost function with respect to theta. The formula for regularization can be seen in equation [2]. By changing the values of λ, overfitting or underfitting can be avoided. Overfitting happens when the curve is very specific to the given data which must be avoided because it doesn't show a general trend. On the other hand, underfitting occur when the curve is too far from the given data. Regularization would be explained further on the latter part of the paper.

II. Procedure

A. Regularized linear regression

The first part of this exercise focuses on regularized linear regression and the normal equations.

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.

2) Hypothesis: From looking at the plot, it seems that fitting a straight line might be too simple of an approximation. Instead, we will try fitting a higher-order polynomial to the data to capture more of the variations in the points. Let's try a fifth-order polynomial. Our hypothesis will be

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_1 x^2 + \theta_1 x^3 + \theta_1 x^4 + \theta_1 x^5 \qquad (1)$$

This means that we have a hypothesis of six features, because x^0, x^1, , x^5 are now all features of our regression.Notice that even though we are producing a polynomial fit, we still have a linear regression problem because the hypothesis is linear in each feature. Since we are fitting a 5th-order polynomial to a data set of only 7 points, over-fitting is likely to occur.

3) Regularization: To guard against overfitting, we will use regularization in our model. Recall that in regularization problems, the goal is to minimize the following cost function with respect to .

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})^2 + \lambda\sum_{j=1}^{n}(\theta_j^2)\right] \qquad (2)$$

4) Regularization parameter: The regularization parameter λ is a control on your fitting parameters. As the magnitues of the fitting parameters increase, there will be an increasing penalty on the cost function. This penalty is dependent on the squares of the parameters as well as the magnitude of λ. Also, notice that the summation after λ does not include θ_0^2.

5) Normal Equations: Now we will find the best parameters of our model using the normal equations. Recall that the normal equations solution to regularized linear regression is

$$\theta = (X^T X + \lambda \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix})^{-1} X^T y \qquad (3)$$

The matrix following λ is an $(n+1) \times (n+1)$ diagonal matrix with a zero in the upper left and ones down the other diagonal entries. (Remember that n is the number of features, not counting the intecept term). The vector y and the matrix X have the same definition they had for unregularized regression:

$$\vec{y} = \begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^m \end{pmatrix} X = \begin{pmatrix} -(X^1)^T- \\ -(X^2)^T- \\ \vdots \\ -(X^m)^T- \end{pmatrix} \qquad (4)$$

6) Implementation Notes: Keep in mind that X is an m x $(n+1)$ matrix, because there are m training examples and n features, plus an x_0 = 1 intercept term. In the data provided for this exercise, you were only give the first power of x. You will need to include the other powers of x in your feature vector X, which means that the first column will contain all ones, the next column will contain the first powers, the next column will contain the second powers, and so on. You can do this in Octave with the command

```
% Features are all powers of x from x^0 to
x^5
x = [ones(m, 1), x, x.^2, x.^3, x.^4,
x.^5];
```

B. Regularized logistic regression

Implement regularized logistic regression using Newton's Method. Load the files 'ml4Logx.dat' and ml4Logy.dat' into your program. This dataset represents the training set of a logistic regression problem with two features. To avoid confusion later, we will refer to the two input features contained in 'ml4Logx.dat' as u and v. So in the 'ml4Logx.dat' file, the first column of numbers represents the feature u, which you will plot on the horizontal axis, and the second feature represents v, which you will plot on the vertical axis.

1) Logistic Hypothesis: We will now fit a regularized regression model to the data. Recall that in logistic regression, the hypothesis function is

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} = P(y = 1|x; \theta) \tag{5}$$

2) Map Feature: To save you the trouble of enumerating all the terms of x, we've included a Octave helper function named 'map_feature' that maps the original inputs to the feature vector. This function works for a single training example as well as for an entire training. To use this function, place 'map_feature.m' in your working directory and call

```
% calling the feature mapping function
x = map_feature(u,v)
```

This assumes that the two original features were stored in column vectors named 'u' and 'v.' (If you had only one training example, each column vector would be a scalar.) The function will output a new feature array stored in the variable 'x.' Of course, you can use any names you'd like for the arguments and the output. Just make sure your two arguments are column vectors of the same size.

3) Logistic Regression Cost Function: Recall that our objective is to minimize the cost function in regularized logistic regression:

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

$$+\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2 \tag{6}$$

Notice that this looks like the cost function for unregularized logistic regression, except that there is a regularization term at the end. We will now minimize this function using Newton's method.

4) Newton's method Update Rule: Recall that the Newton's Method update rule is

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1}\nabla_\theta J \tag{7}$$

This is the same rule that you used for unregularized logistic regression in previous exercise. But because you are now implementing regularization, the gradient and the Hessian will have different forms.

5) Formula Notes: Notice that if you substitute $= 0$ into these expressions, you will see the same formulas as unregularized logistic regression. Also, remember that in these formulas,

1. $x^{(i)}$ is your feature vector, which is a 28x1 vector in this exercise.
2. $\nabla_\theta J$ is a 28x1 vector.
3. $x^{(i)}(x^{(i)})^T$ and H are 28x28 matrices.
4. $y^{(i)}$ and $h_\theta(x^{(i)})$ are scalars.
5. The matrix following $\frac{\lambda}{m}$ in the Hessian formula is a 28x28 diagonal matrix with a zero in the upper left and ones on every other diagonal entry.

C. Application Objective

To apply regularized logistic regression to predict which passengers survived the Titanic shipwreck tragedy. Choose two features from the 'titanic3.xls' data and utilize it in Regularized Logistic Regression Implementation by repeating procedures 4.4 4.8. Give the necessary plots and analysis for each procedure.

III. Data and Results

A. Procedure 4.1

Plot the data

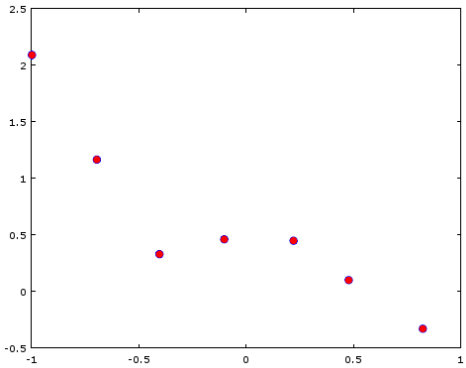

Fig. 1. Data Plot

B. Procedure 4.2

Using the Normal equation, find values for θ using the three regularization parameters below:

C. Procedure 4.3

Plot the polynomial fit for each value of λ.

D. Procedure 4.4

Plot the points using different markers to distinguish between the two classifications

E. Procedure 4.5

Run Newton's Method using the three values of lambda below:

F. Procedure 4.6

Print out the value of J() during each iteration.

G. Procedure 4.7

Use your values of theta to find the decision boundary in the classification problem.

H. Procedure 4.8

use norm(theta) to calculate the L2- norm of , and check it against the norm in the solutions.

IV. Answer to Questions

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

V. Conclusion

References