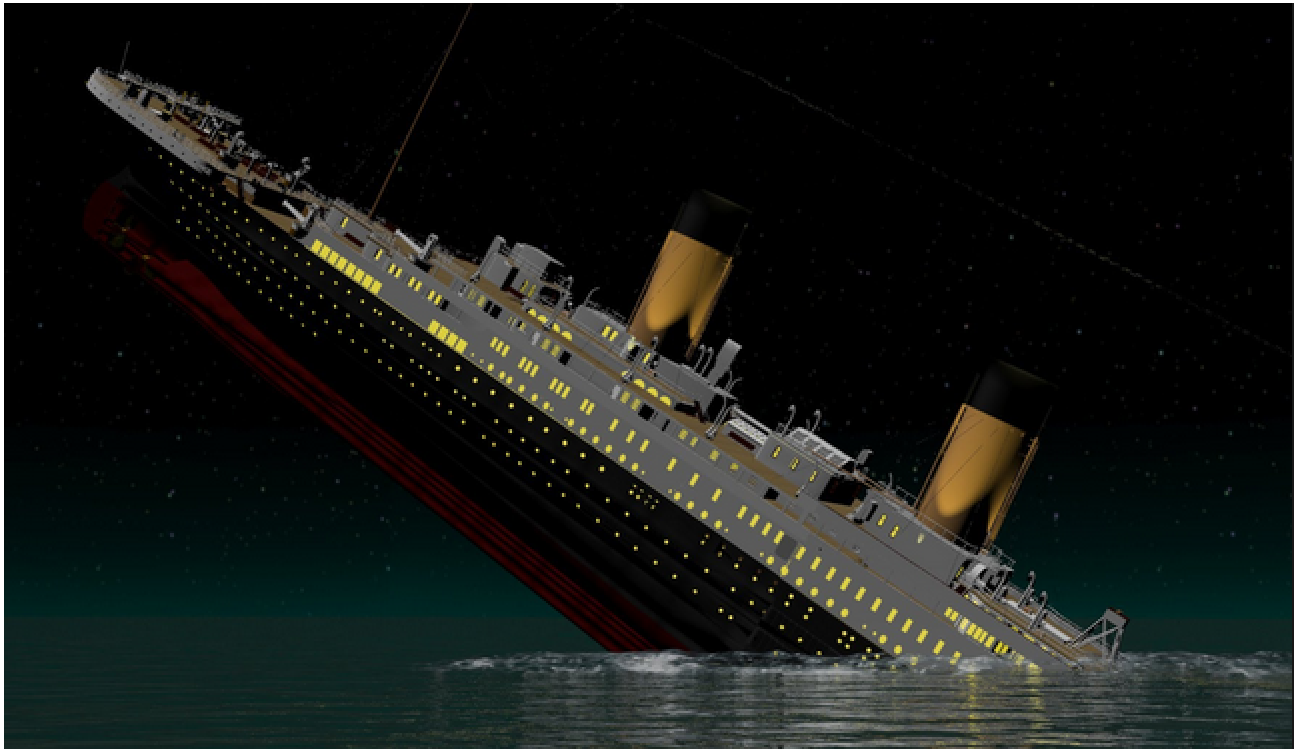
Regularization and Titanic



Objective

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

First Data Set

- The first data bundle contains two sets of data, one for linear regression and the other for logistic regression.
- It also includes a helper function named 'map_feature.m' which will be used for logistic regression.

Second Data Set

- The sinking of the RMS Titanic is one of the most infamous shipwrecks in history.
- On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing **1502** out of 2224 passengers and crew.
- One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew.
- Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

Variable Descriptions:

- survival Survival
  (0 = No; 1 = Yes)
- pclass Passenger Class
  (1 = 1st; 2 = 2nd; 3 = 3rd)
- name Name
- sex Sex
- age Age

Variable Descriptions:

- sibsp Number of Siblings/Spouses Aboard
- parch Number of Parents/Children Aboard
- ticket Ticket Number
- fare Passenger Fare
- cabin Cabin
- embarked Port of Embarkation

  (C = Cherbourg; Q = Queenstown; S = Southampton)

Special Notes

- Pclass is a proxy for socio-economic status (SES)

  1st ~ Upper; 2nd ~ Middle; 3rd ~ Lower

- Age is in Years; Fractional if Age less than One (1)

  If the Age is Estimated, it is in the form xx.5

- With respect to the family relation variables (i.e. sibsp and parch) some relations were ignored. The following are the definitions used for sibsp and parch.

Special Notes

- Sibling: Brother, Sister, Stepbrother, or Stepsister of Passenger Aboard Titanic
- Spouse: Husband or Wife of Passenger Aboard Titanic (Mistresses and Fiances Ignored)
- Parent: Mother or Father of Passenger Aboard Titanic
- Child: Son, Daughter, Stepson, or Stepdaughter of Passenger Aboard Titanic

Part I:
Regularized Linear Regression

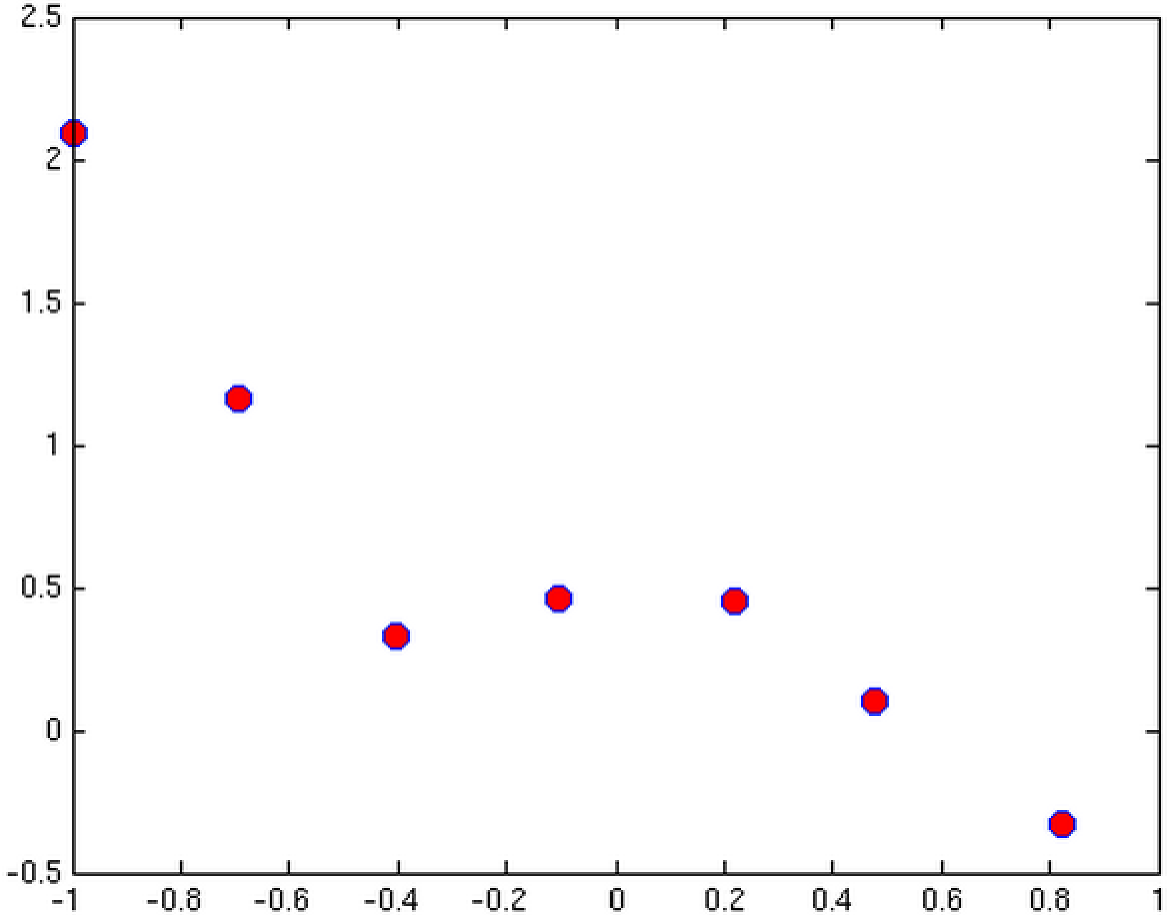
Regularized linear regression

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

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.

Linear Regression Data Plot



Hypothesis

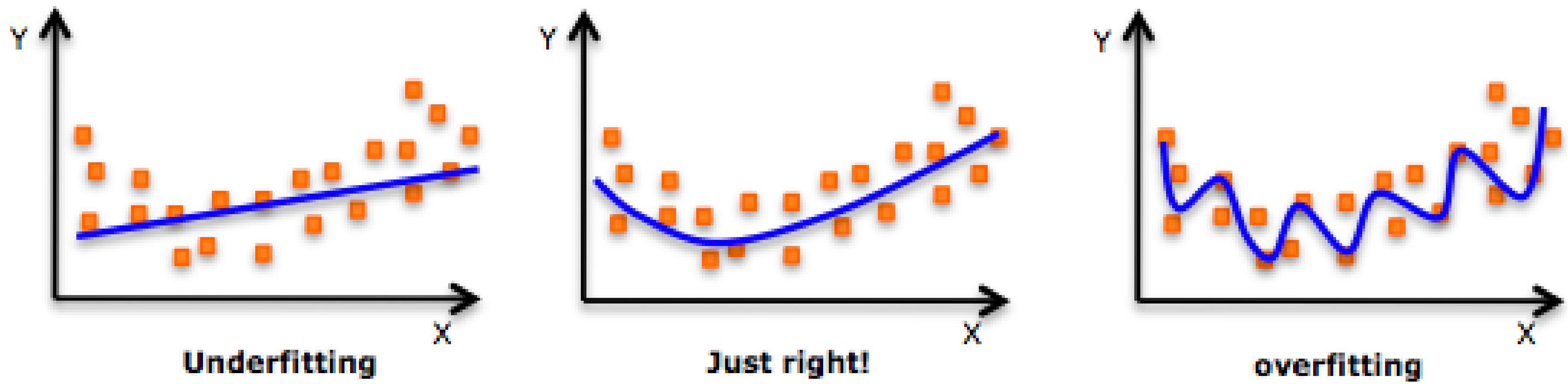
- From looking at the plot, it seems that fitting a straight line might be too simple of an approximation.
- Instead, we will try fitting a higher-order polynomial to the data to capture more of the variations in the points.
- Let's try a fifth-order polynomial. Our hypothesis will be

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5$$

Hypothesis

- 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.

Overfitting vs. Underfitting



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 θ.

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

where λ is the regularization parameter

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.

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

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

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 $\vec{y}$ and the matrix X have the same definition they had for unregularized regression:

$$\vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad X = \begin{bmatrix} —(x^{(1)})^T— \\ —(x^{(2)})^T— \\ \vdots \\ —(x^{(m)})^T— \end{bmatrix}$$

Procedure 4.2

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

  a. $\lambda = 0$ (this is the same case as non-regularized linear regression)

  b. $\lambda = 1$

  c. $\lambda = 10$

Implementation Notes

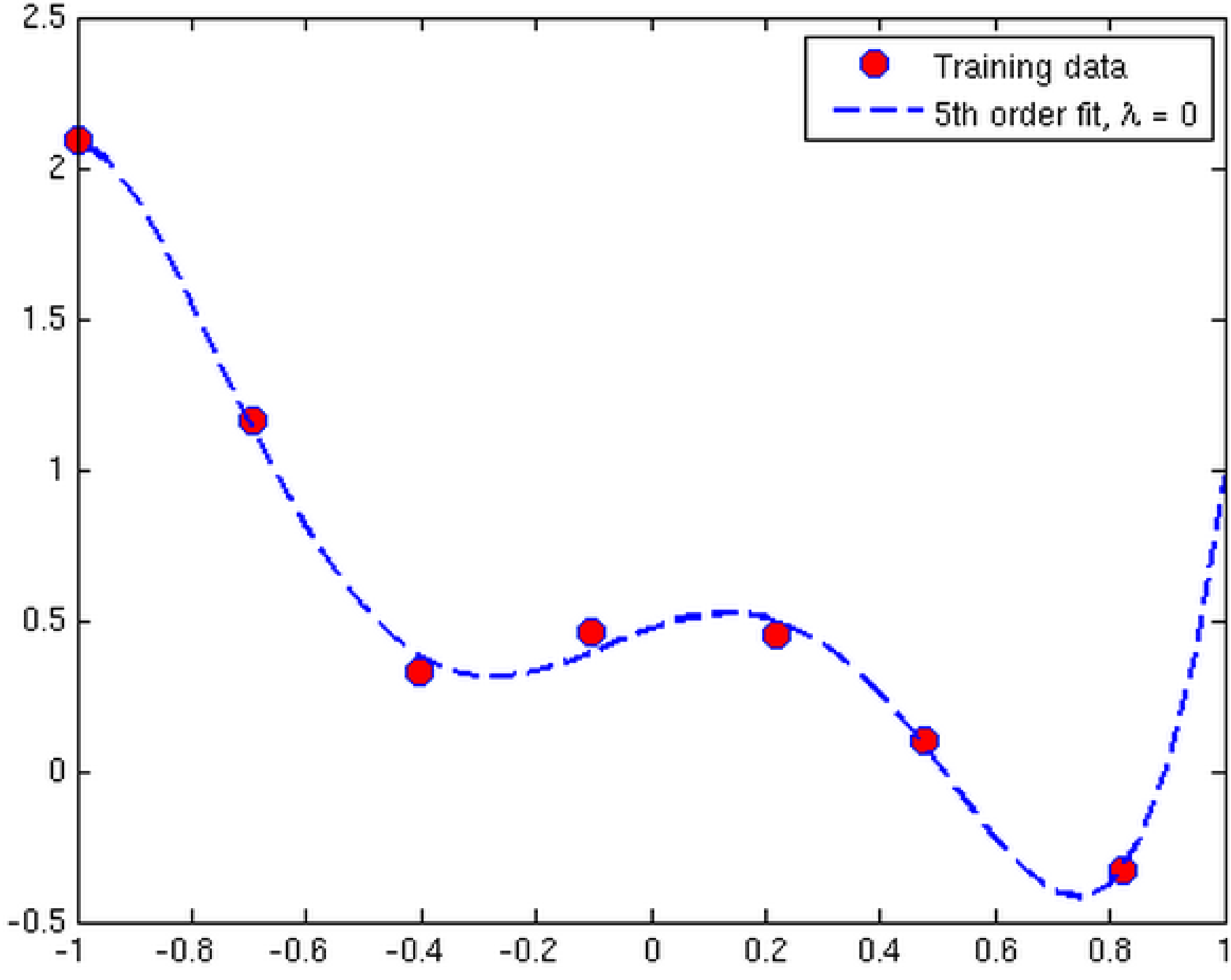
- Keep in mind that X is an m x (n+1) matrix, because there are m training examples and n features, plus an x_0 = 1 intercept term.
- In the data provided for this exercise, you were only give the first power of `x`.
- You will need to include the other powers of x in your feature vector X, which means 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 Matlab/Octave with the command

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

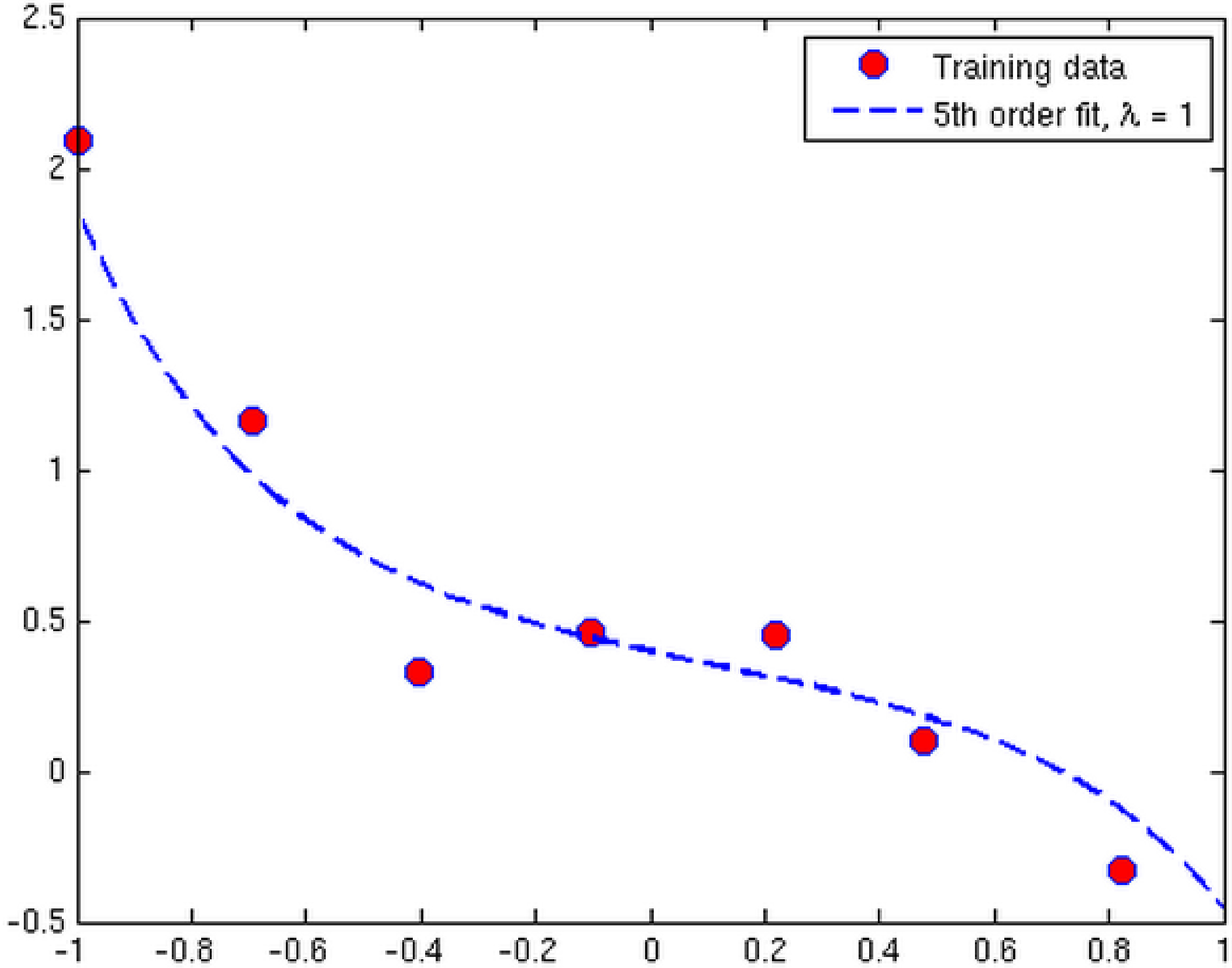
Procedure 4.3 Plot the polyomial fit for each value of λ.

- 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 L2-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).

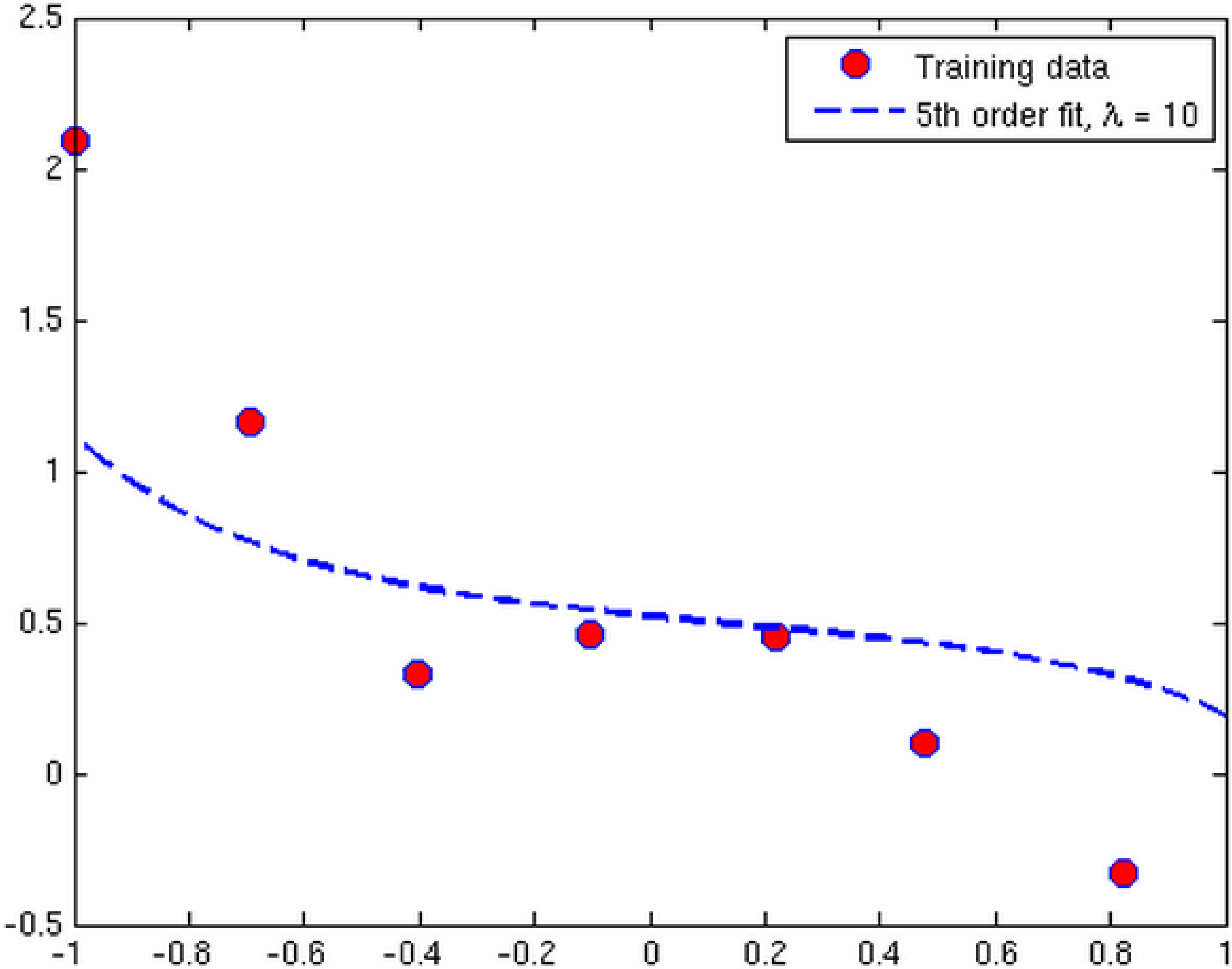
Sample Plot, $\lambda = 0$



Sample Plot, $\lambda = 1$



Sample Plot, λ = 10



Question 1

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

Part II
Regularized logistic regression

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.

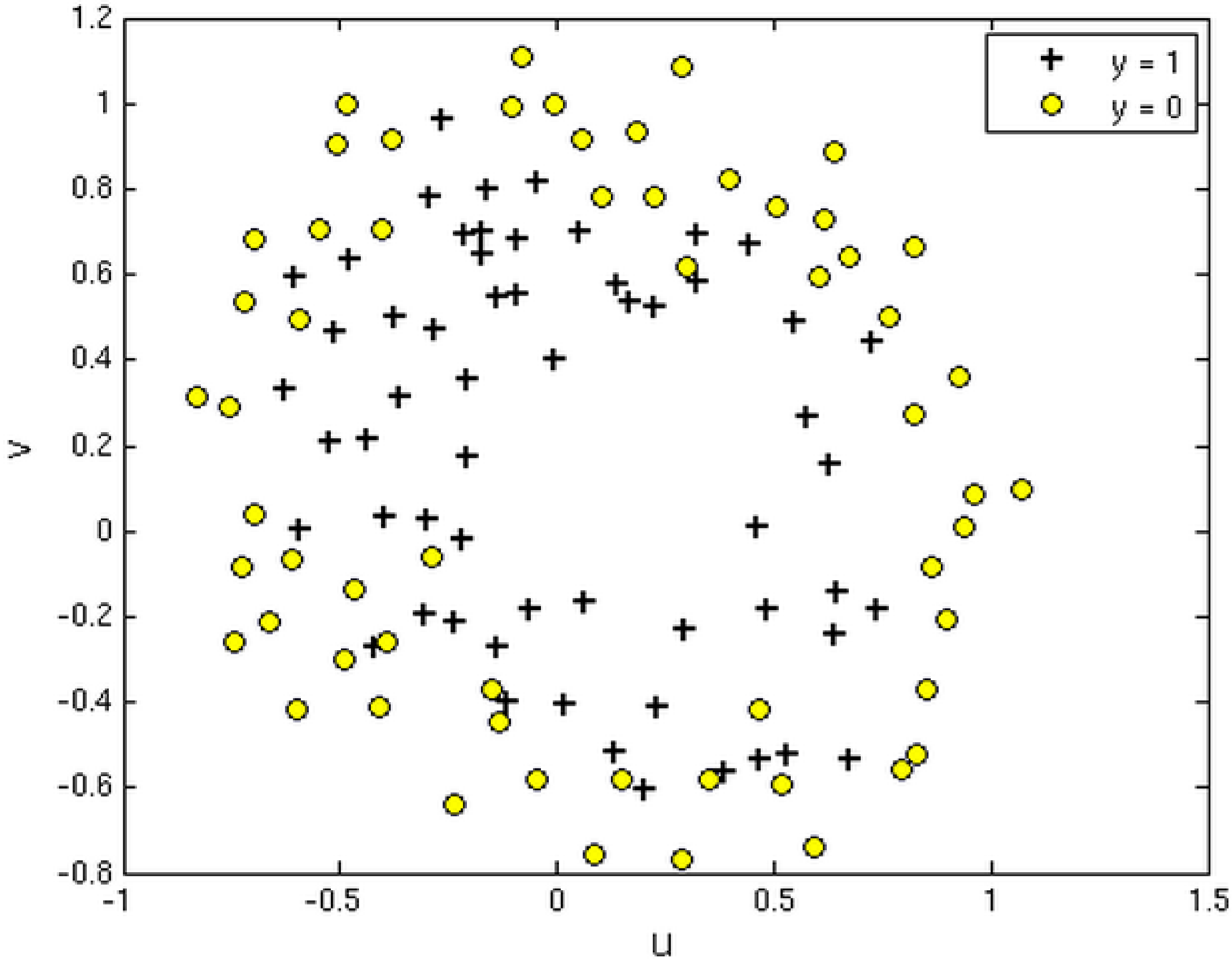
Procedure 4.4

- 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')
```

Sample Plot



Logistic Hypothesis

- We will now fit a regularized regression model to the data.
- Recall that in logistic regression, the hypothesis function is

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$= P(y = 1 | x; \theta)$$

Let's look at the $\theta^T x$ parameter in the sigmoid function $g(\theta^T x)$.

In this exercise, we will assign x to be all monomials (meaning polynomial terms) of u and v up to the sixth power:

$$x = \begin{bmatrix} 1 \\ u \\ v \\ u^2 \\ uv \\ v^2 \\ u^3 \\ \vdots \\ uv^5 \\ v^6 \end{bmatrix}$$

To clarify this notation: we have made a 28-feature vector x where $x_0 = 1, x_1 = u, x_2 = v, \ldots x_{28} = v^6$.

- Remember that u was the first column of numbers in your 'ml4Logx.dat' file and v was the second column.
- From now on, we will just refer to the entries of x as x_0, x_1, and so on instead of their values in terms of u and v.

map_feature.m

- To save you the trouble of enumerating all the terms of x, we've included a Matlab/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

```
x = map_feature(u, v)
```

map_feature.m

- 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.

Logistic Regression Cost Function

- Recall that our objective is to minimize the cost function in regularized logistic regression:

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

- 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.

Newton's method Update Rule

- Recall that the Newton's Method update rule is

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

- 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.

Gradient

$$\nabla_\theta J = \begin{bmatrix} \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_0^{(i)} \\ \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_1^{(i)} + \frac{\lambda}{m}\theta_1 \\ \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_2^{(i)} + \frac{\lambda}{m}\theta_2 \\ \vdots \\ \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_n^{(i)} + \frac{\lambda}{m}\theta_n \end{bmatrix}$$

Hessian

$$H = \frac{1}{m}\left[\sum_{i=1}^{m} h_\theta(x^{(i)})\left(1 - h_\theta(x^{(i)})\right) x^{(i)}\left(x^{(i)}\right)^T\right] + \frac{\lambda}{m}\begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

Formula Notes

- Notice that if you substitute $\lambda = 0$ 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.

Procedure 4.5

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

  a. $\lambda = 0$ (this is the same case as non-regularized logistic regression)

  b. $\lambda = 1$

  c. $\lambda = 10$

Procedure 4.6

- Print out the value of $J(\theta)$ during each iteration.
- $J(\theta)$ should not be decreasing at any point during Newton's Method.
- If it is, check that you have defined $J(\theta)$ 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

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

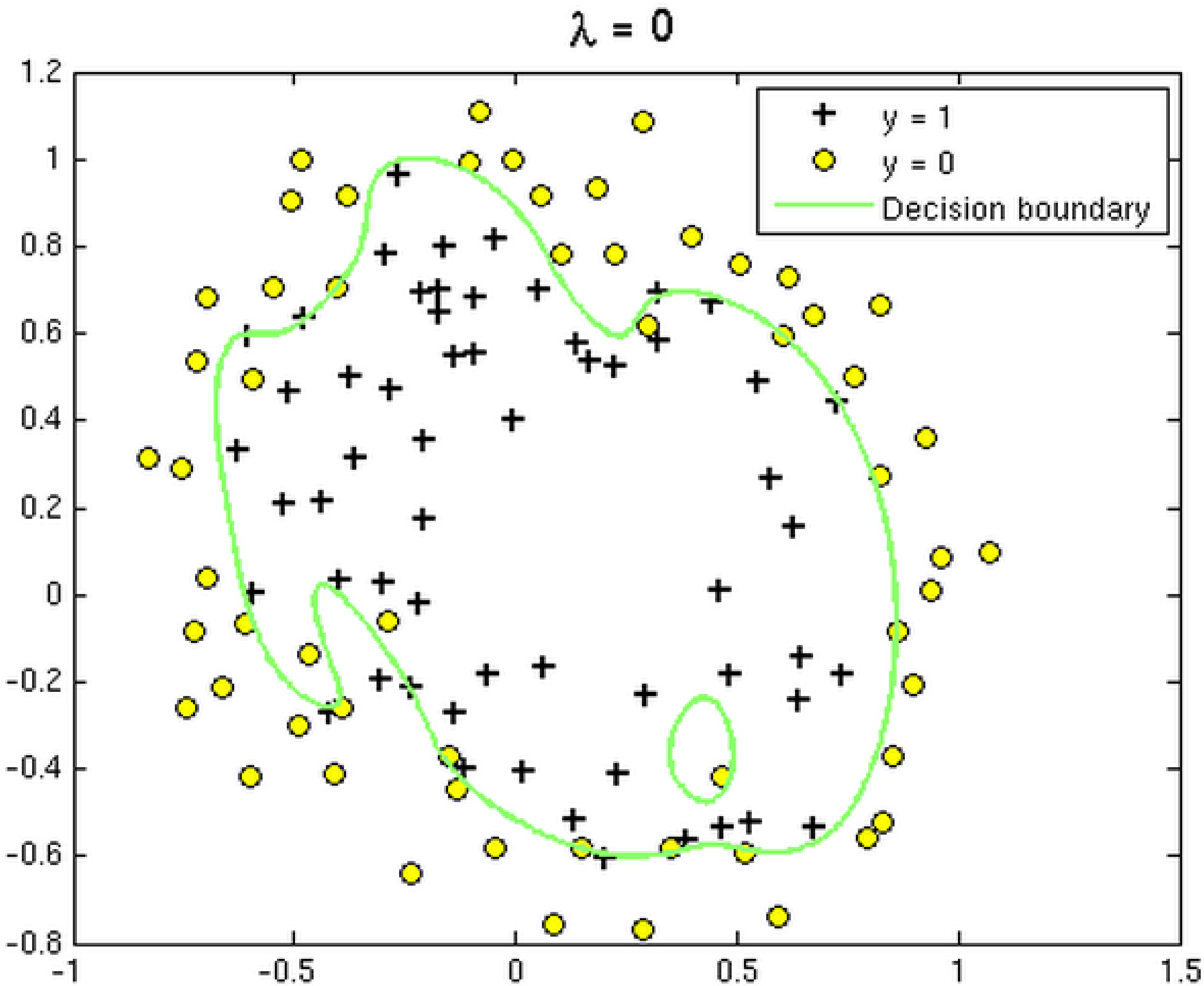
Notes

- Plotting the decision boundary here will be trickier than plotting the best-fit curve in linear regression.
- You will need to plot the $\theta^T x = 0$ line implicity, by plotting a contour.
- This can be done by evaluating$\theta^T x$ over a grid of points representing the original u and v inputs, and then plotting the line where $\theta^T x$ evaluates to zero.

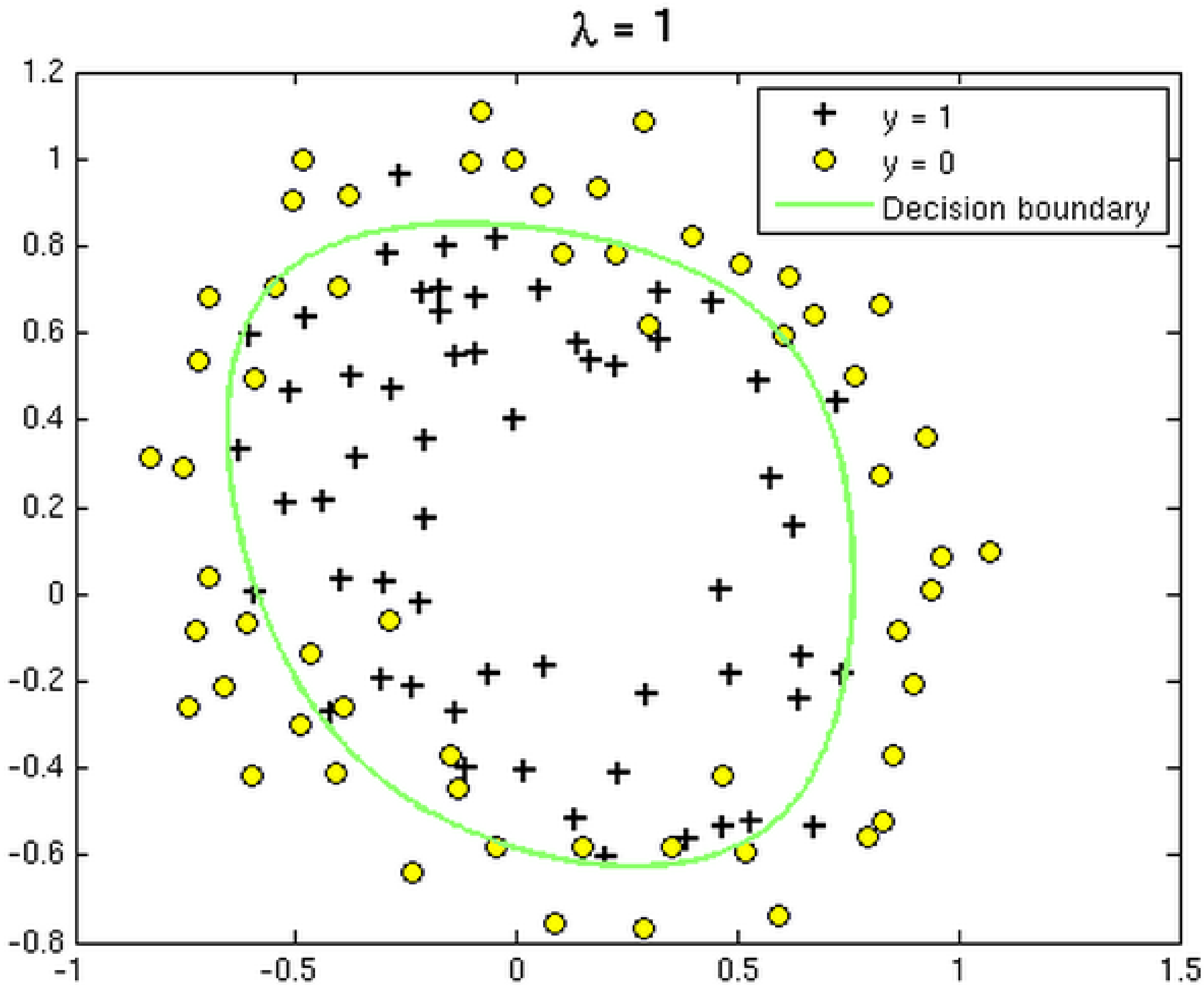
Implementation

```
% 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)
        % Notice the order of j, i here!
        z(j,i) = 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)
```

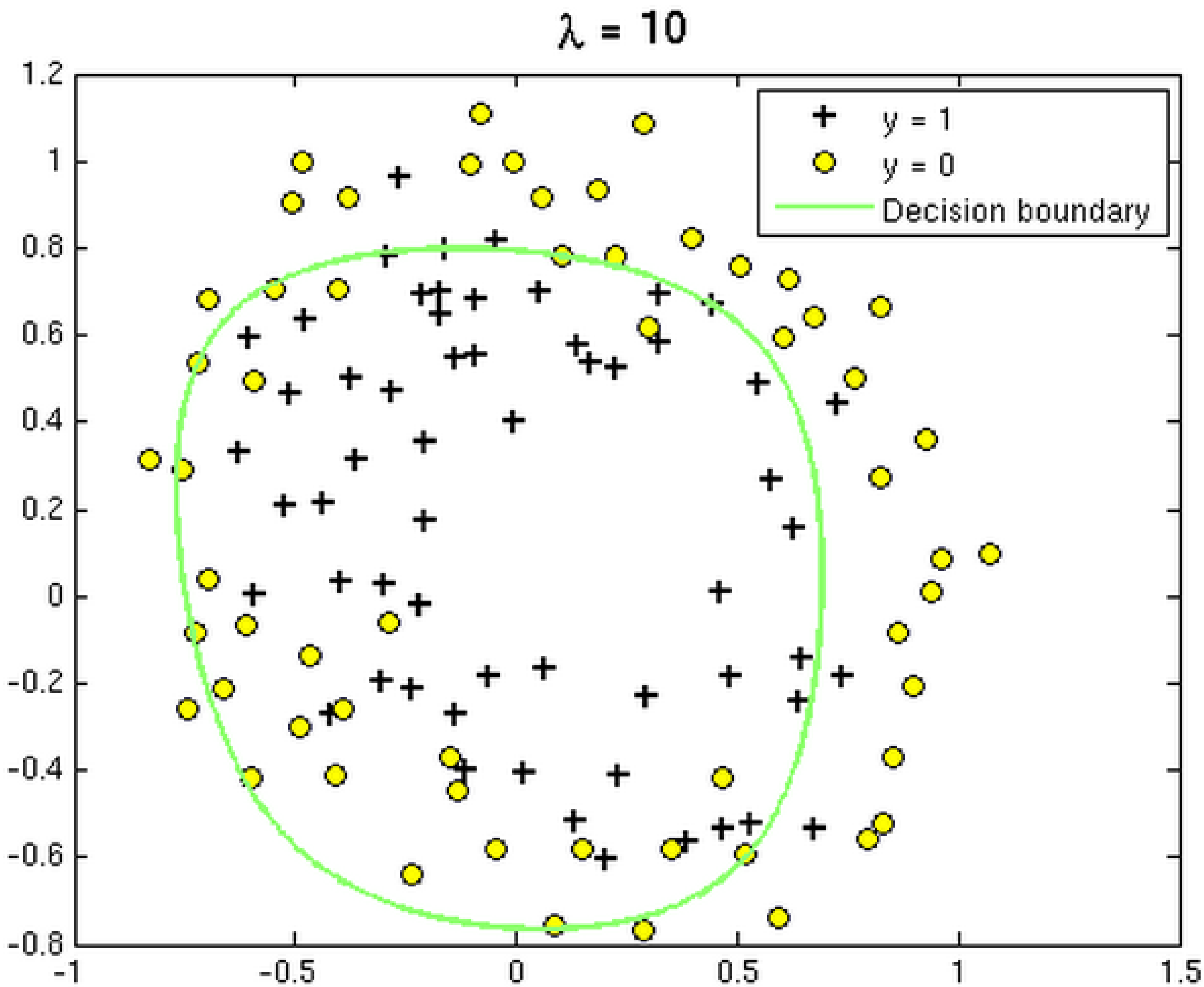
Sample Plot, $\lambda = 0$



Sample Plot, λ = 1



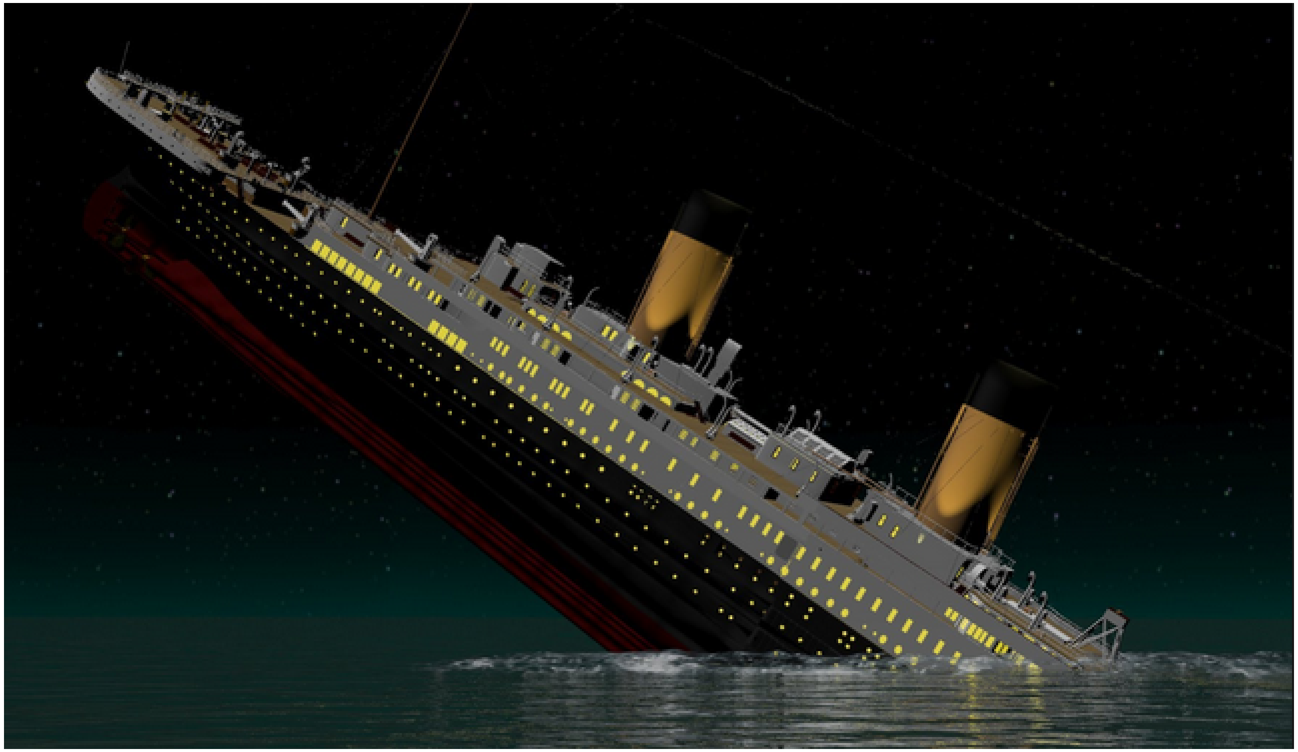
Sample Plot, λ = 10



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.

Part III
Real-world Application



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.

Reference

- Andrew Ng. Stanford University, CS 229 Machine LearningCourse Materials.

  http://cs229.stanford.edu/materials.html

- Titanic: Machine Learning from Disaster, https://www.kaggle.com/c/titanic

END

Appendix

- Wahat is Regularization?

  - introducing additional information or penalty to prevent over-fitting (or solve ill-posed problem)

  - can be restrictions for smoothness or bounds on the vector space norm

  - imposition of prior distributions on model parameters (Bayesian point of view)

Seatwork Questions

1

- The solution(s) to machine learning tasks are often called _______________ .

2

- In Linear Regression, given x = features, y = output data, m = length(x), and model parameter θ, how do you compute the gradient in octave?

```
gradient = ______________________
```

3

- In Linear Regression, given x = features, y = output data, m = length(x), and model parameter θ, how do you update θ in octave?

`theta = ____________________________`

4

- In Linear Regression, given x = features, y = output data, m = length(x), and model parameter θ, how do you compute the cost function values in octave?

```
for i = 1:length(theta0_vals)
   for j = 1:length(theta1_vals)
      t = [theta0_vals(i); theta1_vals(j)];

      J_vals(i,j) = ____________________________

    endfor
endfor
```

5

- How do you plot the contour of cost function, `J_vals`, with respect to parameter θ values - `theta0_vals` and `theta1_vals`?

```
contour(________________________);
```

6

- In Linear Regression exercise 1, how did you compute the predicted height given the model parameters θ and input age, i.e. 3.5 yrs. old.

```
height = ____________________
```

7

- What is the command in Octave for adding a column of ones to our feature vector x?

x = ______________________;

8

- How did we preprocess the raw feature data in Laboratory exercise 2?

```
x(:,2) = ______________________________
```

9

- In gradient descent, how did we compute the cost function in Octave?

```
for i = 1:length(alpha)
    theta = zeros(size(x(1,:)))';
    J = zeros(MAX_ITR, 1);
    for num_iterations = 1:MAX_ITR
        % Calculate the J term
        J(num_iterations) = ________________
        ...
    end
end
```

10

- Using the unscaled raw features, x_unscaled, and output data, y, how did we compute the model parameters, θ, using Normal equations in Octave?

```
theta_normal = ________________
```

11

- How do we compute the predicted price for a house with area 1650 and 3 bedrooms using the model parameters, θ, acquired from the Normal equation?

```
price_normal = ________________
```

12

- How do we compute the predicted price for a house with area 1650 and 3 bedrooms using the model parameters, θ, acquired from the Gradient descent (Lab exercise 2)?

`price_grad_desc = _______________`

13

- In Octave, how can you separate the positive class and the negative class in the data, i.e. pass has a label 1 nad fail with label 0?

```
pass = _____________ ;
fail = _____________ ;
```

14

- In Octave, one way to create a Sigmoid or logistic function is: (fill in the blank)

```
g = ______________________________;
z = linspace(-10,10,1000);
plot(z,g(z),'linewidth',5);
```

15

- In Linear regression we utilized Gradient descent for updating model parameters, in Logistic regression we used ____________________ .

16

- In logistic regression, how do you compute the gradient in Octave?

```
gradient = ______________________________
```

17

- In logistic regression, how do you compute the cost function, J, in Octave?

```
J_i = ______________________________
```

18

- In logistic regression, how do you update the model parameters, θ, in Octave?

```
theta = = ______________________________
```

19

- In Octave, how did you predict the probability that a student with a score of 20 on Exam 1 and a score of 80 on Exam 2 will not be admitted? (Given: g(z) as the octave function)

```
probability =
```