ROCHESTER INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING
CMPE 789 Machine Intelligence
HW #3

Name:____Kaiwen Zheng__________

Due 11:55pm, 09/20/2017, submit via Dropbox. Work alone. All questions pertain to Matlab/Octav.

1. (8 pts) Vector x=[3 -2 5 1 0]. What is $|x|_0$, $|x|_1$, $|x|_2^2$, $|x|_\infty$.

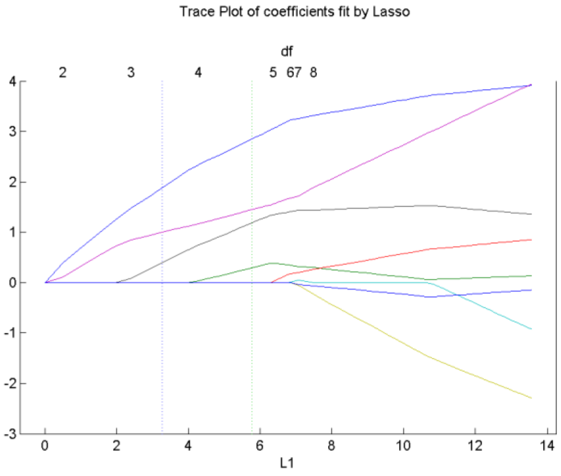
Ans:
$|x|_0$= 4
$|x|_{1,}$= 11
$|x|_2^2$= 39
$|x|_\infty$ = 5

2. (6 pts) Please check which statements are true regarding linear regression regularization

a) Best subset is usually preferred over lasso regression
b) If many features contribute to the solution ridge regression performs better than lasso regression
c) To remove many features from a solution, ridge regression is preferred

Ans: b

3. (3 pts) In the below Lasso coefficient plot, how many coefficients contribute at a L1 = 5?



Ans: 4 coefficients

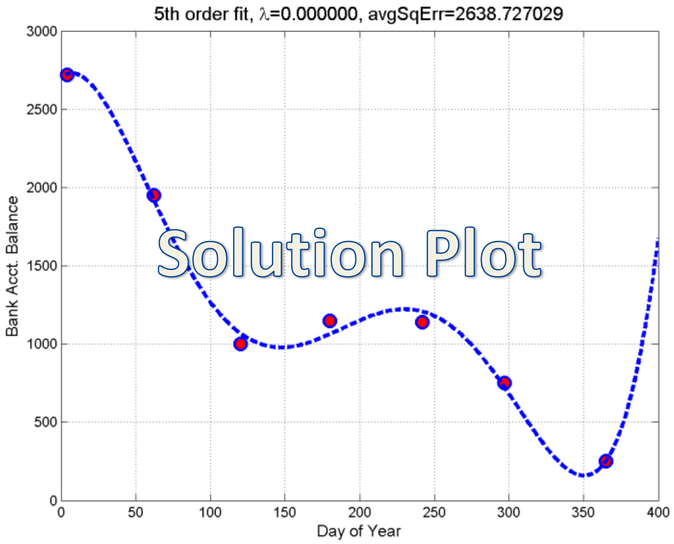
4. (6 pts) In the plot for the problem 3, some lines go up from zero, while others go down. Under what conditions will a line go up vs. down?

Ans: Line going up means that a specific coefficient has positive correlation with the cost function, the more influence it exerts, the higher the line goes. Line going down means that a specific coefficient has negative correlation with the cost function.

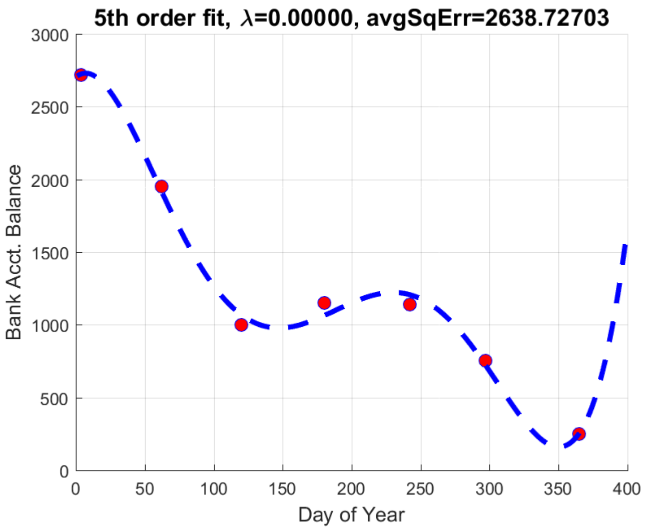
5. (12 pts) Starting with the data:

```
%day of year
x =[4 62 120 180 242 297 365]';
%bank balance
y =[2720 1950  1000 1150  1140  750 250]' ;
```

Use linear regression with a 5th order polynomial to produce a plot similar to below (without the watermark 'Solution Plot') Details: Your average squared error should be 2638.72709. Markersize needs to be 10, with face color of red and edge color of blue. Line fit needs to be a blue dashed line of width 3 evaluated over the range 1:400. Xlabel and ylabel need to have fontsize of 12 and title needs to have fontsize of 14. Title needs to show value of λ=0 (using Greek character) and avgSqErr displayed properly.



Ans (show code and include your plot at 50%size)



```
%day of year
x = [4 62 120 180 242 297 365]';
%bank balance
y = [2720 1950 1000 1150 1140 750 250]';

M = [ones(length(x),1) x x.^2 x.^3 x.^4 x.^5];
theta = ((M'*M)\M')*y;
avgSqErr = sum((y-M*theta).^2)./length(y);
err = num2str(avgSqErr,'%.5f');
str = strcat('5th order fit, \lambda=0.00000, avgSqErr=', err);
graphX = (1:400)';
M2 = [ones(length(graphX),1) graphX graphX.^2 graphX.^3 graphX.^4
graphX.^5];
graphY = M2*theta;
% plot
figure
scatter(x, y, 60,'MarkerEdgeColor','b','MarkerFaceColor','r')
hold on
plot(graphX, graphY,'b--','MarkerSize',10,'LineWidth',3)
% labels
title(str,'fontsize',14)
xlabel('Day of Year','fontsize',12);
ylabel('Bank Acct. Balance','fontsize',12);
grid on
print('cmpe677_hwk3_5_5th_order','-dpng')
```

6. (12 pts) Starting with the data:

```
%percent of way in semester
x =[0  0.2072  0.3494 0.4965  0.6485  0.7833  0.9400]';
%bank balance ($K)
y =[2.150 1.541  0.790 0.909  0.901  0.593 0.198]' ;
```

implement regularized linear regression, λ=0.001. The normal equation solution to regularized linear regression is:

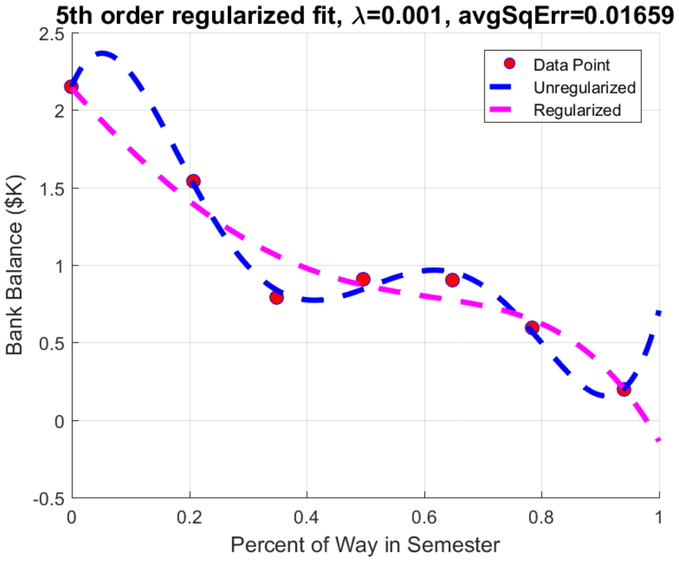
$$\theta = \left(xData^T xData + \lambda \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \right)^{-1} xData^T y$$

It is generally not a good idea to penalize the constant coefficient, so instead implement:

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

Use linear regression with a 5th order polynomial to produce a plot similar to the previous problem. Overlay a λ=0.001 regularized fit as a dashed line on top of a 5th order plot, color is magenta, linewidth=3, title needs to have λ and avgSqErr of the regularized model.

Ans: (code and plot)



```
%percent of way in semester
x = [0 0.2072 0.3494 0.4965 0.6485 0.7833 0.9400]';
%bank balance ($K)
y = [2.150 1.541 0.790 0.909 0.901 0.593 0.198]' ;
% 5th order linear regression
lambda = 0.001;
model = [ones(length(x),1) x x.^2 x.^3 x.^4 x.^5];
theta = ((model'*model)\model')*y;
theta2 = regularNormalEquation(model,y,lambda);
avgSqErr = sum((y-model*theta2).^2)./length(y);
str = strcat('5th order regularized fit, \lambda=', num2str(lambda), ',
avgSqErr=', num2str(avgSqErr,'%.5f'));
% plot
figure
graphX = (0:0.001:1)';
M1 = [ones(length(graphX),1) graphX graphX.^2 graphX.^3 graphX.^4
graphX.^5];
graphY1 = M1*theta;
```

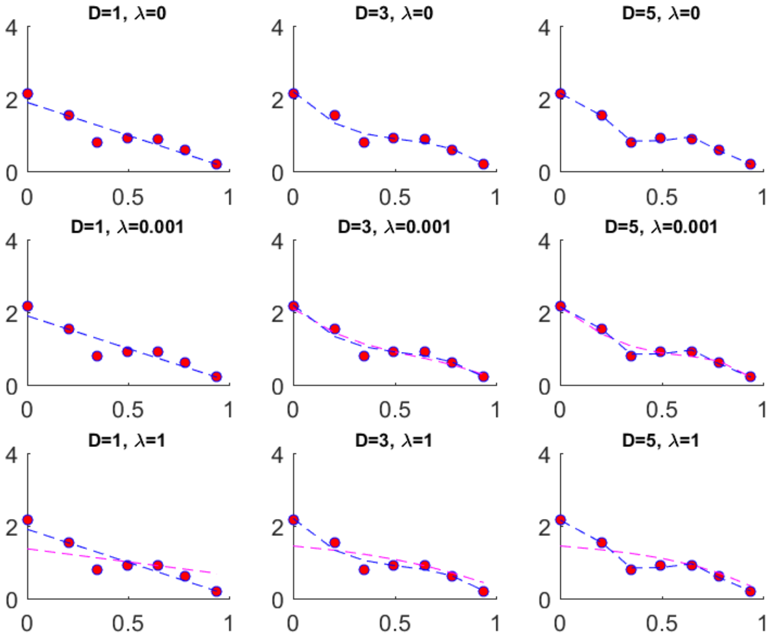
```
graphY2 = M1*theta2;
scatter(x, y, 60,'MarkerEdgeColor','b','MarkerFaceColor','r')
hold on
plot(graphX, graphY1,'b--','MarkerSize',10,'LineWidth',3)
plot(graphX, graphY2,'m--','MarkerSize',10,'LineWidth',3)
% labels
title(str,'fontsize',14)
xlabel('Percent of Way in Semester','fontsize',12);
ylabel('Bank Balance ($K)','fontsize',12);
legend('Data Point', 'Unregularized', 'Regularized')
grid on
print('cmpe677_hwk3_6_regularized','-dpng')
```

7. (13 pts) Starting with the data:

```
%percent of way in semester
x =[0  0.2072  0.3494 0.4965  0.6485  0.7833  0.9400]';
%bank balance ($K)
y =[2.150 1.541  0.790 0.909  0.901  0.593 0.198]' ;
```

generate a 3x3 subplot (9 subplots in one figure), where D=[1,3,5] and λ=[0 0.001 1]. Title for each is the value of D and λ, separated by a comma (ie: D=3, λ=0.001), fontsize=8. Do not use a xlabel or ylabel for this part.

Ans (show code and <u>75% plot</u>):



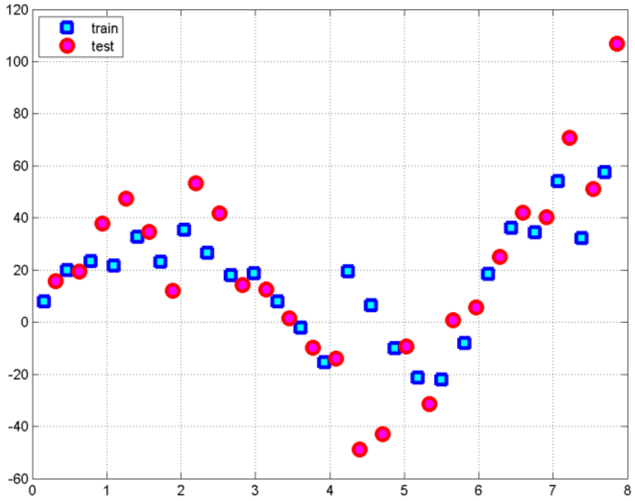
```
%percent of way in semester
x =[0 0.2072 0.3494 0.4965 0.6485 0.7833 0.9400]';
```

```
%bank balance ($K)
y =[2.150 1.541 0.790 0.909 0.901 0.593 0.198]';
lambda = [0 0.001 1];
D = [1 3 5];
D1 = [ones(length(x),1) x];
D3 = [ones(length(x),1) x x.^2 x.^3];
D5 = [ones(length(x),1) x x.^2 x.^3 x.^4 x.^5];
models = {D1, D3, D5};
count = 0;
index = 0;
figure
for lam = lambda
  count = 0;
  for d = D
    index = index + 1;
    count = count + 1;
    str = strcat('D=', num2str(d), ', \lambda=', num2str(lam));
    subplot(3,3,index)
    theta = regularNormalEquation(models{count},y,lam);
    theta2 = ((models{count}'*models{count})\models{count}')*y;
    scatter(x, y, 20,'MarkerEdgeColor','b','MarkerFaceColor','r')
    hold on
    plot(x,models{count}*theta,'m--')
    plot(x,models{count}*theta2,'b--')
    title(str,'fontsize',8)
  end
end
print('cmpe677_hwk3_7_3x3','-dpng')
```

8. (20 pts) Starting with the code:

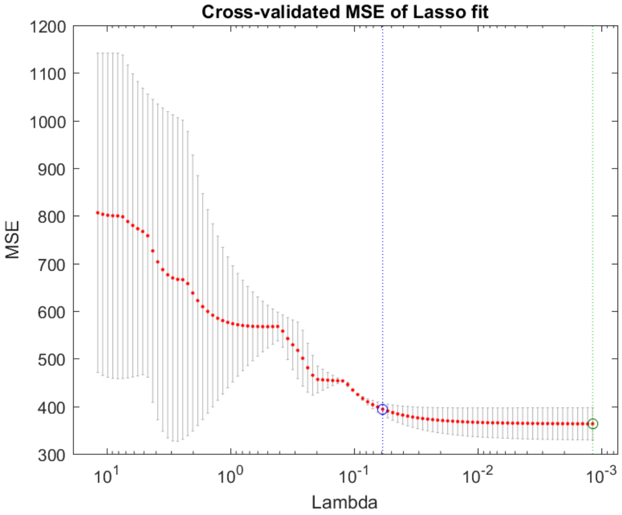
```
clear; close all
rng(2000);  %random number generator seed
mu = [0 0 ];
sigma = [4 1.5 ; 1.5 2];
r = mvnrnd(mu,sigma,50); %create two features, 50 samples of each
y = r(:,1);
x=(pi*(1:50)/20)';  %scale x for sin
y=10*sin(x).*(4+y); % add some curvature
y =y + x*4;  % gradually rise over time
hold off; plot(x,y,'x');

xtrain = x(1:2:end); ytrain = y(1:2:end);  %odd samples for train
xtest = x(2:2:end); ytest = y(2:2:end);    %even samles for test
figure(1);
hold off
plot(xtrain, ytrain, 'rs', 'MarkerSize', 10,'LineWidth',3,'markerfacecolor','c','markeredgecolor','b'); %
Plot the data
hold on
plot(xtest, ytest, 'ro', 'MarkerSize', 10,'LineWidth',3,'markerfacecolor','m','markeredgecolor','r'); %
Plot the data
grid on; legend('train','test');
print -dpng hwk3_problem8_data.png
```



Create a M matrix using [x^0 x^1 x^2 x^3] . Note- use all available train and test data here, variable ‘x’ in the above code. Using the lasso.m command (with 'cv',2 optional arguments), call lassoPlot.m.

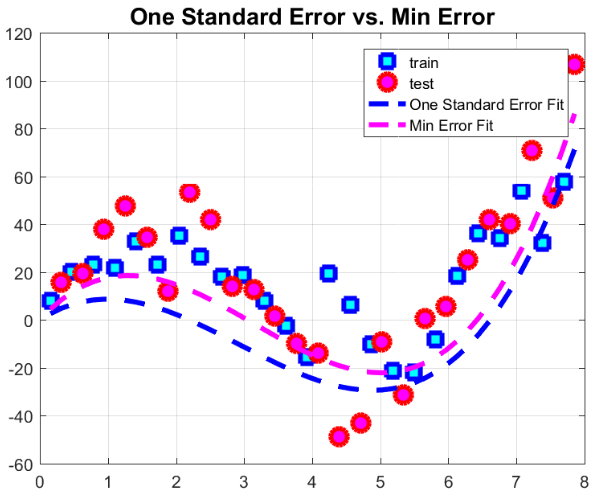
a) Attach the output plot from lassoPlot.m.



b) Using the data you just plotted with lassoPlot.m, do you think better results would come from more or less regularization?
Ans: We could use less regularization because looking at the graph, the MSE decreases as lambda decreases.

c) What are the weights associated with the one standard error point from min error?
Ans:
One standard error [0, 18.7937, -11.4026, 1.2943]
Min standard error [0, 31.2058, -15.0914, 1.5935]

d) Overlay two curves on top of the training/test plot points. One showing a line fit corresponding to the hypothesis created with the one standard error and another showing the line fit corresponding to the hypothesis created with minimum error weights?



e) Using the minimum MSE weights from lasso.m, what is the average mean squared error associated with the testing points?
Ans: Average squared error for testing points 445.9431

f) Using the minimum MSE weights from lasso.m, what is the average mean squared error associated with the training points?
Ans: Average squared error for training points 303.4353

9. (8 pts) Regularization is added to gradient descent by incurring a cost from the solved weights during each iteration. Before you can implement gradient descent, you need a way to track your performance, which computes J(θ). This cost was computed via a function called computeCost.m in hwk2. Copy computeCost.m from hwk2 to a m-file called computeCostReg.m. Update the cost function in computeCostReg.m from:

$$J(\theta) = \frac{1}{2n}\left[\sum_{i=1}^{n}\left(h\left(\theta, x^{(i)}\right) - y^{(i)}\right)^2\right]$$

To:

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

Note we do not penalize our θ_0 weight. Test your code with the following matlab segment.
Do not continue until your answer is: 2.1740e+09

```
clear ; close all;
% Load Data
data = load('..\hwk2\ex1data2.txt');
X = data(:, 1:2);
y = data(:, 3);
```

```
% Scale features and set them to zero mean with std=1
[Xnorm mu sigma] = featureNormalize(X);  % reuse this function from hwk2

% Add intercept term to X
Xdata = [ones(length(X),1) Xnorm];

% Init Theta and lambda
theta = ((Xdata'*Xdata)\Xdata')*y;  %well..this is the optimal solution
lambda=1;

%Run Compute Cost
disp(computeCostReg(Xdata,y,theta, lambda))
```

Ans (show computeCostReg.m):

```
function J = computeCostReg(xData, y, theta, lambda)

J = sum((xData*theta -y).^2) /(2*length(y)) +
(lambda/(2*length(y)))*sum(theta(2:end).^2);

end
```

10. (12 pts) Regularization can be added to gradient descent by modifying the theta update step from:

$$tempj = \theta_j - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h(\theta, x^{(i)}) - y^{(i)}\right)\left(x^{(i)}\right)$$

To:

$$tempj = \theta_j - \frac{\alpha}{n}\left\{\left[\sum_{i=1}^{n}\left(h(\theta, x^{(i)}) - y^{(i)}\right)\left(x^{(i)}\right)\right] + [\lambda\theta_j]\right\}$$

Copy gradientDescentMulti.m from Hwk2 to gradientDescentMultiReg.m. Update gradientDescentMultiReg.m such that it includes the above update step (Note- do not update the bias term) and calls computeCostReg.m from the problem 9.

Use the following matlab code:

```
clear ; close all;
data = load('..\hwk2\ex1data1.txt'); % Dataset from Andrew Ng, Machine Learning MOOC
X = data(:, 1);
y = data(:, 2);
M = [ones(length(X),1) X];
theta_init = zeros(2, 1); % initialize fitting parameters to zero

% Some gradient descent settings
iterations = 1500;
alpha = 0.01;
```

```
lambda = 0;

% run gradient descent
theta_unreg = gradientDescentMultiReg(M, y, theta_init, alpha, iterations,lambda);
lin_reg = ((M'*M)\M')*y;  % optimal solution
lambda = 1;
theta_reg = gradientDescentMultiReg(M, y, theta_init, alpha, iterations,lambda);

fprintf('Linear Regression: [%f,%f]\n',lin_reg);
fprintf('Gradient Descent: [%f,%f]\n',theta_unreg);
fprintf('Regularized Gradient Descent: [%f,%f]\n',theta_reg);
```

Do not continue until you get:
Linear Regression: [-3.895781,1.193034]
Gradient Descent: [-3.630291,1.166362]
Regularized Gradient Descent: [-3.624388,1.165623]

Change lambda = 100, what are the new regularized gradient descent weights?

Ans (show gradientDescentMultiReg.m code and ans with lambda=100):

```
function theta = gradientDescentMultiReg(Xdata, y, theta, alpha, 
num_iters, lam)
    n = length(y); % number of training examples
    temp = zeros(1,size(theta,1));

    for iter = 1:num_iters
        for i = 1:size(theta,1);
            temp(i) = theta(i)-(alpha/n)*sum((Xdata*theta-
y).*Xdata(:,i))+(alpha/n)*abs(theta(i)*lam);
        end
        theta = temp';
    end
end
```

Regularized Gradient Descent with lambda=100: [-0.659719, 0.879798]