COGS 109 Fall 2015
Due: 12/06/2015, 11:59Pm
Grade: ___out of 100 points

Homework assignment 6: Linear regression and perceptron

Instruction: Please read the instructions carefully and download the dataset at https://sites.google.com/site/ucsdcogs109fall2015/assignments

1. (**40 points**) (**Linear regression and error rates**) Here, you will learn how to do linear regression and compute error rates in matlab. Please download the data: "LinearRegression.mat". In this mat file, there are a 500 by 1 vector called Xtrain, which is one feature of 500 samples for training, and a 500 by 1 vector called Ytrain, which is the target value for X to train on. Additionally, there is a testing set, which called Xtest and Ytest, both are 250 by 1 vectors. In the following steps, we will try to find the associations between X and Y, using simple linear regression and polynomial regressions.
   (a) Using the training set, find the coefficients for simple linear regression, predicting Y using X only. Report the coefficients here.
   Hint:
   ```
   A1 = cat(2,ones(length(Xtrain),1),Xtrain);
   W1 = A1\Ytrain;
   ```

   (b) Using the training set, find the coefficients for linear regression with quadratic term, that is predicting Y using X and X^2. Report the coefficients here.
   Hint:
   ```
   A2 = cat(2,ones(length(Xtrain),1),Xtrain,Xtrain.^2);
   W2 = A2\Ytrain;
   ```

   (c) Using the training set, find the coefficients for linear regression with cubic term, that is predicting Y using X, X^2, and X^3. Report the coefficients here.
   Hint:

```
A3 = cat(2,ones(length(Xtrain),1),Xtrain,Xtrain.^2,Xtrain.^3);
W3 = A3\Ytrain;
```

(d) Generate predicted regression lines given the derived coefficients. Create a figure with 1 by 3 subplot. In each subplot, draw the scatter plot of your training data and overlay it with the predicted lines. Simple linear in subplot 1, linear regression with quadratic term in the 2nd, and linear regression with cubic term in the 3rd. Report your figure here. Please use different colors for scatter dots and predicted lines.

Hint: (Be careful about the dimension when do the matrix manipulation)

```
Xpred = linspace(min(Xtrain),max(Xtrain),500)'; % create an X series to draw lines
A1pred = cat(2, ones(length(Xpred),1),Xpred);
Ypred1 = A1pred*W1;
subplot(1,3,1);
scatter(Xtrain,Ytrain,20,'filled');
hold on;
plot(Xpred,Ypred1,'r','LineWidth',3);
```

(e) Calculate training error rate. Report training error rate for each of the three models here.

Hint:

```
Etrain1 = mean((Ytrain - A1*W1).^2);
```

(f) Apply the coefficient to the testing set. Calculate the testing error rates. Report testing error rate for each of the three models here.

Hint:

```
A1test = cat(2,ones(length(Xtest),1),Xtest);
Etest1 = mean((Ytest - A1test*W1).^2);
```

(g) Compare error rates calculated in the step (e) and (f). Point out which of the three

models is the best model. Explain why you pick that model.

2. (**30 points**) (**Multiple linear regression for predicting news popularity**) Here, we will use dataset downloaded from UCI machine learning repository. Please down load the data "OnLineNewsPopularity.mat". In this mat file, there are three different categories of features and one popularity measures of online news. We will build multiple regression models, one for each category of features, to predict the popularity of online news. In this dataset, there are 39605 online news.
   (a) First category of features is the Content of the online news, which has 17 features. The second category of features is the Pub_weekdays, which specify 7 publication days in the week. The final category is the Stats, which has 35 features regarding some summary about that news. Build multiple regression models to predict the popularity of online news, one for each category of features.
   Hint:
   ```
   A1 = cat(2,ones(length(Pub_Weekdays),1),Pub_Weekdays);
   W1 = A1\Popularity;
   ```
   (b) Use the original data points to generate the predicted popularity. Create a 1 by 3 subplot. In each subplot, plot the predicted popularity as a function of true popularity for each category of features. Report your figure here.
   ```
   subplot(1,3,1);
   scatter(Popularity,A1*W1,20,'filled');
   ```
   (c) Based on the plots created in step (b), which category of features is the best among the three to predict the online news? How this relates to training errors?

3. (**30 points**) (**Perceptron**) Here, we will use the dataset "SeparateMe.mat" to practice the perceptron algorithm. In this mat file, data has two features, x1 an x2. The perceptron would use those two features to find the best line that correctly classifies the label specified in a vector called target.
   (a) Initialize the weights for a perceptron "y = b + w1*x1 + w2*x2" as "y = -3 + 2*x1 +

1*x2".

(b) Draw a scatter plots of data points, each data point is colored given their corresponding true label specified in the target. Overlay the scatter plot with the decision boundary with initial weights in step (a). Report your figure here.

Hint:

```
scatter(x1(target==-1),x2(target==-1),10,'g','filled');
hold on
scatter(x1(target==1),x2(target==1),10,'r','filled');
x_test = -11:11; %define an arbitrary x sequence for drawing the line
y_test = (-w1*x_test-b)/w2;
plot(x_test,y_test,'k','linewidth',2);
```

(c) Given current decision boundary, assign the predicted labels and then determine whether it is a correct classification or not. Record those have incorrect classification. After calculation, create the same plot as step (b), but this time, mark those misclassified data points. Report your graph here.

Hint:

```
err_id=[];
for i = 1:N %loop through all points
        net=w1*x1(i)+w2*x2(i)+b;
        if net>=0 %set output to 1 if net >=0
                output(i) = 1;
        else    %set output to -1 if net <0
                output(i) = -1;
        end
        if output(i)==target(i)
                incorrect(i) = 0;
        else
                incorrect(i) = 1;
                err_id=[err_id i]; %add index of index of incorrect output to err_id
```

```
        end
    end
    if any(err_id)
        scatter(x1(target==-1),x2(target==-1),10,'g','filled');
        hold on
        scatter(x1(target==1),x2(target==1),10,'r','filled');
        x_test = -11:11; %define an arbitrary x sequence for drawing the line
        y_test = (-w1*x_test-b)/w2;
        plot(x_test,y_test,'k','linewidth',2);
        scatter(x1(err_id),x2(err_id) ,50,'k','linewidth',2);
```

(d) Update the weights for decision boundary given the error.

Hint:

```
w1=w1+(target(err_id(1))-output(err_id(1)))*x1(err_id(1));
w2=w2+(target(err_id(1))-output(err_id(1)))*x2(err_id(1));
b = b+(target(err_id(1))-output(err_id(1)));
```

(e) Create a 2 by 2 subplots. In each subplot, repeat the procedure from (c) to (d). Named the title of each subplot with the repeated times. Therefore, the four subplots of this 2 by 2 figure correspond to iteration 1, 2, 3, and 4, respectively.

(f) Repeat the procedure until there is no more classification error. Draw a figure as we did in the step (c). Report the figure here. How many iterations the perceptron went through until it converges?

Hint:

You can use `while any(err_id)`, or `if .... break` to stop the iterations given the converge criteria.