Homework 3
Applied Machine Learning
Fall 2017
CSCI-P 556/INFO-I 526

Ashay H. Sawant
ahsawant@umail.iu.edu

October 27, 2017

"All work herein is solely mine."

Problem 1 [100 points]

Simple Linear Regression [45 points]

1.1 Perform a simple linear regression with "mpg" as the response and "horsepower" as the predictor. What are the parameters $\theta = (\theta_0, \theta_1)$? Is the relationship between horsepower and mpg positive or negative? [20 pt]

Answer:

Parameters $\theta = (\theta_0, \theta_1)$ for simple linear regression with "mpg" as response and "horsepower" as the predictor are as follows:

$\theta_0 = 23.4453, \theta_1 = -6.075461$

I have trained my model with $\alpha = 0.1$ & Number of Iteration = 100.

Additionally, relationship between Horsepower and mpg is negative. As we can see in the plot below, we can estimate that as Horsepower increases MPG decreases.

1.2 Plot the output variable and the input variable. Display the least squares regression line. [5 pt]

Answer:

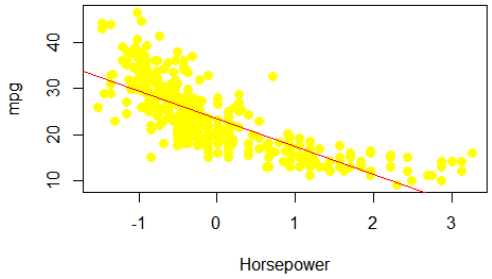

Figure 1: Horsepower vs Mpg with regression line

1.3 Use the model obtained in Q.1.1 to make predictions. What is the "mpg" value for "horsepower = 220"? [5 pt]

Answer:

According to univariate regression model trained in Q.1.1, predicted value of MPG when Horsepower = 220 is **5.21002**.

1.4 In a contour plot, show how $J(\theta)$ varies with changes in θ_0 and θ_1. Does $J(\theta)$ have a global minimum? [10 pt]
Answer:

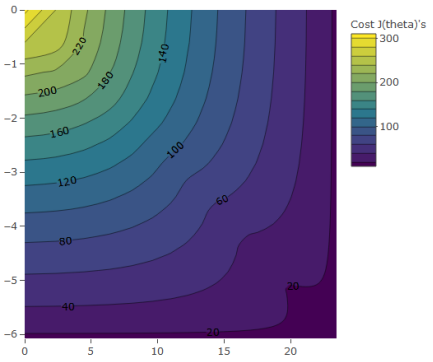

Figure 2: Changes in cost(J(θ)) with θ_0&θ_1

$J(\theta)$ has a global minimum.

1.5 The closed-form solution to linear regression is $\theta = (\Delta^T\Delta)^{-1}\Delta^T y$. Report the coefficients using this formula. [5 pt]
Answer:
Using the Normal Equation for linear regression, coefficients θ_0 and θ_1 are:
θ_0 = 39.9358610, θ_1 = -0.1578447.
Predicted value of MPG when Horsepower = 220 is **5.21002**.

Multivariate Linear Regression [55 points]

1.6 First, perform feature scaling (mean normalization) over the Auto data set to make gradient descent converge faster. Then, train a multivariate linear regression with "mpg" as the response and all other variables except name as the predictors. Report the parameters (θ's). What does the coefficient for the "year" variable suggest? [30 pt]

Answer:

Coefficients(θ's) of Multivariate Linear Regression are as follows: $\theta_0 = 23.0965457, \theta_1 = -0.8580478, \theta_2 = -0.8310335, \theta_3 = -1.1928443, \theta_4 = -2.4641686, \theta_5 = -0.3469268, \theta_6 = 2.5135884, \theta_7 = 1.0492937$
Here, coefficient of "year" is positive with value 2.513584. It suggests that, older the vehicle is, mpg value will be less. In other words, bigger the value of year, more will be the mpg. mpg and year shares positive relationship.

1.7 Use the model obtained in Q.1.6 to make predictions. What is the "mpg" value for $(x_1, \ldots, x_7) = (4, 300, 200, 3500, 11, 70, 2)$? [5 pt]

Answer:

As per the model trained with $\alpha = 0.003$ and Number of Iterations = 1400, predicted value for mpg is **15.56491**.

1.8 In this question, you are asked to test different learning rates. Run your gradient descent for 100 iterations at the chosen learning rates ($\alpha_1 = 3, \alpha_2 = 0.3, \alpha_3 = 0.03, \alpha_4 = 0.00003$). For each learning rate, make a plot that shows how $J(\theta)$ changes at each iteration. Discuss the plots? i.e., which one looks better? does it converge? [15pt]

Answer:

(a) When $\alpha = 3$

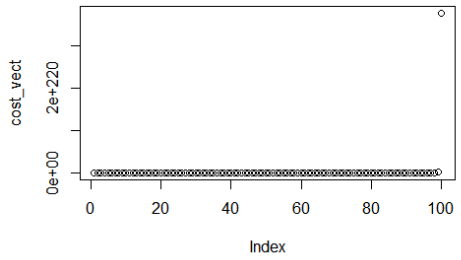

Figure 3: Jθ changes with iteration for $\alpha = 3$

As we can see in cost with iteration plot above, there is slight or no change in the Jθ. Additionally, at the 100^{th} iteration, value of Jθ raises to some large number. Hence, algorithm does not converge at $\alpha = 3$.

(b) When $\alpha = 0.3$

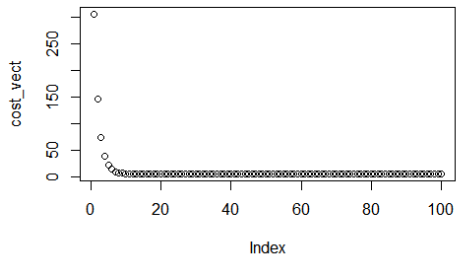

Figure 4: Jθ changes with iteration for $\alpha = 0.3$

As we can see in cost with iteration plot above, Cost(Jθ) decreases rapidly for first 15 iterations. Then for remaining iterations, there is slight change in Jθ to become stable. Hence, algorithm converge at $\alpha = 0.3$ for 100 iterations.

(c) When $\alpha = 0.03$

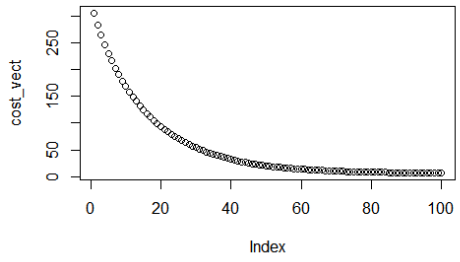

Figure 5: Jθ changes with iteration for $\alpha = 0.03$

As we can see in cost with iteration plot above, we can see subtle change in Cost(Jθ) with every iteration. Plot becomes stable at around 90^{th} iteration. Hence, algorithm converge at $\alpha = 0.03$ for 100 iterations.

(d) When $\alpha = 0.00003$

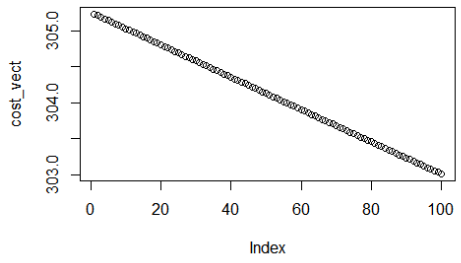

Figure 6: Jθ changes with iteration for $\alpha = 0.00003$

As we can see in cost with iteration plot above, we can see the linear change in Cost(Jθ) with every iteration. At 100^{th} iteration, cost is somewhere around 303. Hence, we can say that there is a scope to decrease the cost. So, algorithm does not converge at $\alpha = 0.00003$ for 100 iterations.

From the above analysis, we can conclude that both plot at $\alpha = 0.3$ and 0.03, the algorithm will yield better result than other two.

1.9 Calculate the coefficients using the normal equations. [5pt]
Answer:
Coefficients(θ's) of Multivariate Linear Regression using Normal Equation method are as follows:
$\theta_0 = 23.44591, \theta_1 = -0.8415907, \theta_2 = 2.081966, \theta_3 = -0.6524734, \theta_4 = -5.499057, \theta_5 = 0.2222864,$
$\theta_6 = 2.76565, \theta_7 = 1.148796$

Problem 2 [35 points]

1. Exercise 13:

   (a) (a) Using the rnorm() function, create a vector, x, containing 100 observations drawn from a N(0, 1) distribution. This represents a feature, X.
   R code:

   ```
   set.seed(1)
   x = rnorm(n = 100,mean = 0,sd = sqrt(1))
   ```

   (b) (b) Using the rnorm() function, create a vector, eps, containing 100 observations drawn from a N(0, 0.25) distribution i.e. a normal distribution with mean zero and variance 0.25.
   R code:

   ```
   eps = rnorm(n = 100,mean = 0,sd = sqrt(0.25))
   ```

   (c) (c) Using x and eps, generate a vector y according to the model
   Y = 1 + 0.5X + eps (3.39)
   What is the length of the vector y? What are the values of β_0 and β_1 in this linear model?
   R code:

   ```
   y = -1 + 0.5 * x + eps
   ```

   Length of vector y is **100**.
   Value of $\beta_0 = -1$ & $\beta_1 = 0.5$

   (d) (d) Create a scatterplot displaying the relationship between x and y. Comment on what you observe.
   R code:

   ```
   plot(x,y)
   ```

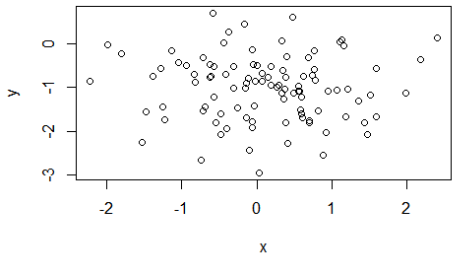

Figure 7: Scatter plot of X and Y

In the graph plot between x and y variable above, we can see the linear relationship between them. Y variable has positive slope.

(e) Fit a least squares linear model to predict y using x. Comment on the model obtained. How do $\bar{\beta_0}$ and $\bar{\beta_1}$ compare to β_0 and β_1?
R code:

```
linear_model = lm(y~x)
```

True coefficients are $\bar{\beta_0}$ = -1 and $\bar{\beta_1}$ = 0.5.
Coefficients after performing linear regression are $\beta_0 = -1.00447962 \& \beta_1 = -0.02919715$.
Hence, linear regression fits a model close to the true values of the coefficient.

(f) Display the least squares line on the scatterplot obtained in (d). Draw the population regression line on the plot, in a different color. Use the legend() command to create an appropriate legend.
R code:

```
plot(x,y)
abline(linear_model, col = "blue")
abline(-1, 0.5, col = "red")
legend("bottomright",c("Linear Model", "True Model"), col = c("blue", "red"), lty = c(1,1))
linear_model = lm(y~x)
```

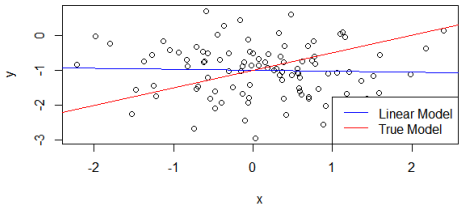

Figure 8: Graph plot of x and y with regression lines

(g) Now fit a polynomial regression model that predicts y using x and x^2. Is there evidence that the quadratic term improves the model fit? Explain your answer.
R code:

```
linear_model.quad = lm(y~x+I(x^2))
```

There is evidence that model fit has been increased. But this change has cost us slight increment is Residual Standard Error.

(h) Repeat (a) to (f) after modifying the data generation process in such a way that there is less noise in the data. The model (3.39) should remain the same. You can do this by decreasing the variance of the normal distribution used to generate the error term in (b). Describe your results.
R code:

```
set.seed(1)
x1 = rnorm(n = 100,mean = 0,sd = 1)
eps1 = rnorm(n = 100,mean = 0,sd = sqrt(0.005))
y1 = -1 + 0.5*x1 + eps1
plot(x1, y1)
linear_model1 = lm(y1~x1)
abline(linear_model1, col = "red")
```

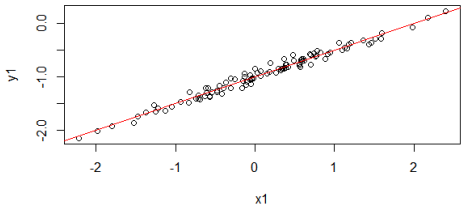

Figure 9: Graph plot of x1 and y1 with regression line

To decrease the noise in the data, let's decrease the value of variance in eps to 0.005.
As you can see in graph plot above, the data is saturated diagonally. Linear model fits more accurately than previous one. Hence, there will be less Squared error.

(i) Repeat (a) to (f) after modifying the data generation process in such a way that there is more noise in the data. The model (3.39) should remain the same. You can do this by increasing the variance of the normal distribution used to generate the error term in (b). Describe your results.
R code:

```
set.seed(1)
x2 = rnorm(n = 100,mean = 0,sd = 1)
eps2 = rnorm(n = 100,mean = 0,sd = sqrt(1.25))
y2 = -1 + 0.5*x2 + eps2
plot(x2, y2)
linear_model2 = lm(y2~x2)
abline(linear_model2, col = "red")
```

To increase the noise in the data, let's increase the value of variance in eps to 1.25.

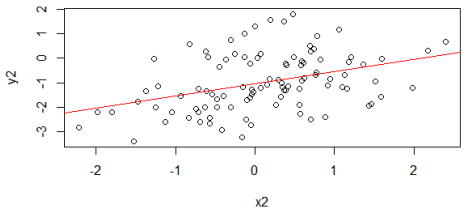

Figure 10: Graph plot of x2 and y2 with regression line

As you can see in graph plot above, the data is spread over the space. Linear model fits less accurately than previous one. Hence, there is considerable amount of Squared error.

(j) What are the confidence intervals for 0 and 1 based on the original data set, the noisier data set, and the less noisy data set? Comment on your results.
R code:

```
confint(linear_model)
confint(linear_model1)
confint(linear_model2)
```

```
> confint(linear_model)
                2.5 %     97.5 %
(Intercept) -1.1521917 -0.8567675
x           -0.1932659  0.1348716
> confint(linear_model)
                2.5 %     97.5 %
(Intercept) -1.1521917 -0.8567675
x           -0.1932659  0.1348716
> confint(linear_model1)
                2.5 %     97.5 %
(Intercept) -1.0162748 -0.9890557
x1           0.4848084  0.5150416
> confint(linear_model2)
                2.5 %     97.5 %
(Intercept) -1.2573275 -0.8269557
x2           0.2598003  0.7378286
```

Figure 11: Confint of 3 different models

As we can see in above output of confint for three models, intercept of regression line has positive relationship with variance of eps.

2. Exercise 14:

   (a) Perform the following commands in R:

```
set .seed (1)
x1=runif (100)
x2 =0.5* x1+rnorm (100) /10
y=2+2* x1 +0.3* x2+rnorm (100)
```

   The last line corresponds to creating a linear model in which y is a function of x1 and x2. Write out the form of the linear model. What are the regression coefficients?
   Answer:
   Form of the linear model: $Y = 2 + 2X_1 + 0.3X_2 + \epsilon$
   Regression coefficients: $\beta_0 = 2, \beta_1 = 2, \beta_2 = 0.3$

   (b) What is the correlation between x1 and x2? Create a scatterplot displaying the relationship between the variables.

```
cor(x1,x2)
```

   Co-relation between x_1 and x_2 is 0.8351212.

```
plot(x1,x2)
```

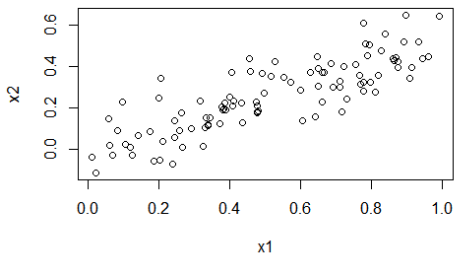

Figure 12: Scatter plot of x_1 and x_2

(c) Using this data, fit a least squares regression to predict y using x1 and x2. Describe the results obtained. What are $\bar{\beta}_0$, $\bar{\beta}_1$, and $\bar{\beta}2$? How do these relate to the true β_0, β_1, and β_2? Can you reject the null hypothesis $H_0 : \beta_1 = 0$? How about the null hypothesis $H_0 : \beta_2 = 0$?

```
linear_multi = lm(y~x1+x2)
summary(linear_multi)
linear_multi$coefficients
```

```
> summary(linear_multi)

Call:
lm(formula = y ~ x1 + x2)

Residuals:
    Min      1Q  Median      3Q     Max
-2.8311 -0.7273 -0.0537  0.6338  2.3359

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
x1            1.4396     0.7212   1.996   0.0487 *
x2            1.0097     1.1337   0.891   0.3754
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.056 on 97 degrees of freedom
Multiple R-squared:  0.2088,    Adjusted R-squared:  0.1925
F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

Figure 13: Summary of Multivariate Linear Regression

True coefficients are $\bar{\beta}_0 = 2$, $\bar{\beta}_1 = 2$ and $\bar{\beta}_2 = 0.3$.
Coefficients after performing linear regression are $\beta_0 = 2.130500$, $\beta_1 = 1.439555$ and $2\beta_2 = 1.009674$.
Predicted regression coefficients are somewhat close to the true coefficients. Standard error is pretty much high.
We can reject the null hypothesis for β_1 because its p-value is below 5%. But we cannot reject the null hypothesis for β_2 because its p-value is much above the 5%.

(d) Now fit a least squares regression to predict y using only x1. Comment on your results. Can you reject the null hypothesis $H_0 : \beta_1 = 0$?

```
linear_model_x1 = lm(y~x1)
summary(linear_model_x1)
```

P-value for t-statistics for β_1 is $2.66e^{-06}$ that is nearly equal to zero. Hence, we can ignore the Null Hypothesis for regression coefficient β_1.

(e) Now fit a least squares regression to predict y using only x2. Comment on your results. Can you reject the null hypothesis $H_0 : \beta_2 = 0$?

```
linear_model_x1 = lm(y~x1)
summary(linear_model_x1)
```

P-value for t-statistics for β_2 is $1.37e^{-05}$ that is nearly equal to zero. Hence, we can ignore the Null Hypothesis for regression coefficient β_2.

(f) Do the results obtained in (c) to (e) contradict each other? Explain your answer.
Answer:
We know that X_1 and X_2 have collinearity. Hence, when we consider them together for regression their individual effect hides. But when we perform regression separately, we can distinguish the individual effect.

(g) Now suppose we obtain one additional observation, which was unfortunately mismeasured.

```
x1=c(x1 , 0.1)
x2=c(x2 , 0.8)
y=c(y,6)
```

Re-fit the linear models from (c) to (e) using this new data. What effect does this new observation have on the each of the models? In each model, is this observation an outlier? A high-leverage point? Both? Explain your answers.
R code:

```
x1 = c(x1, 0.1)
x2 = c(x2, 0.8)
y = c(y, 6)
linear_regfit1 = lm(y~x1+x2)
linear_regfit2 = lm(y~x1)
linear_regfit3 = lm(y~x2)
```

Problem 3 [65 points]

Logistic Regression

3.1 Implement the sigmoid function and make a plot of it by testing different inputs. $\left(g(z) = \frac{1}{1+e^{-z}}\right)$ [5 pt]
Answer:

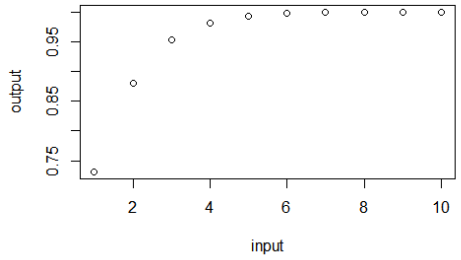

Figure 14: Sigmoid Function over inputs 1,2,..,10

3.2 Perform logistic regression on the new.Auto data set in order to predict "mpg01" using the input variables: cylinders, displacement, horsepower, weight. Report the parameters (θ's). [30 pt]
Answer:
Coefficients(θ's) of Multivariate Logistic Regression are as follows:
$\theta_0 = -0.2359074, \theta_1 = -0.7179372, \theta_2 = -0.7203217, \theta_3 = -0.5896819, \theta_4 = -0.8001831$

3.3 What is the error of the model over new.Auto data set? [10 pt]
Answer:
To calculate the error rate in model, I have first trained the model and then compute the mpg01 values for each data point. Next, I compared the predicted mpg01 values with actual mpg01 values to find the error rate.
According the this method, error rate of the model is **10.71%**.

3.4 Use the model obtained in Q.3.1 to make predictions. What is the "mpg01" value for $(x_1, \ldots, x_4) = (8, 340, 200, 3500)$ (first, scale the data point with the parameters obtained earlier while normalizing the features) [5 pt]
Answer:
Predicted mpg01 for above set of values is $\mathbf{0.01395648 \approx 0}$.

3.5 In this question, you are asked to test different learning rates. Run your gradient descent for 100 iterations at the chosen learning rates ($\alpha_1 = 3, \alpha_2 = 0.3, \alpha_3 = 0.03, \alpha_4 = 0.00003$). For each learning rate, make a plot that shows how $J(\theta)$ changes at each iteration. Discuss the plots? i.e., which one looks better (faster)? does it converge? [15pt]
Answer:

(a) When $\alpha = 3$

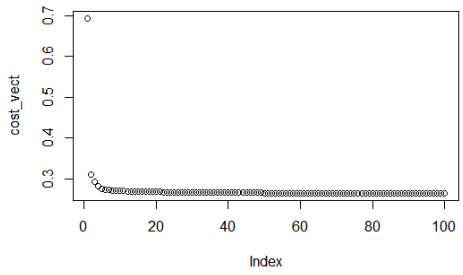

Figure 15: Jθ changes with iteration for $\alpha = 3$

As we can see in cost with iteration plot above, cost dropped significantly around 5th iteration. Further there is no change in the Jθ. Hence, algorithm converges at $\alpha = 3$.

(b) When $\alpha = 0.3$

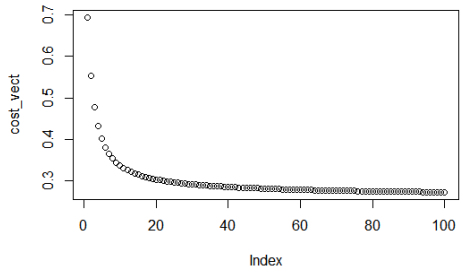

Figure 16: Jθ changes with iteration for $\alpha = 0.3$

As we can see in cost with iteration plot above, Cost(Jθ) decreases rapidly for first 20 iterations. Then for remaining iterations, there is slight change in Jθ to become stable. Hence, algorithm converge at $\alpha = 0.3$ for 100 iterations.

(c) When $\alpha = 0.03$

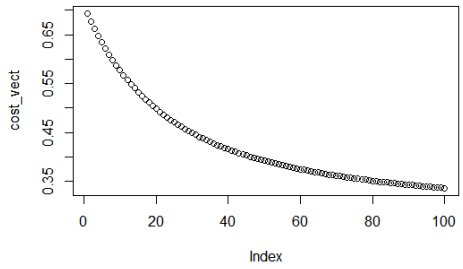

Figure 17: Jθ changes with iteration for $\alpha = 0.03$

As we can see in cost with iteration plot above, we can see subtle change in Cost(Jθ) with every iteration. Plot reaches cost 0.35 at 100th iteration. Hence, there is more scope to minimize the cost. Hence, algorithm does not converge at $\alpha = 0.03$ for 100 iterations.

(d) When $\alpha = 0.00003$

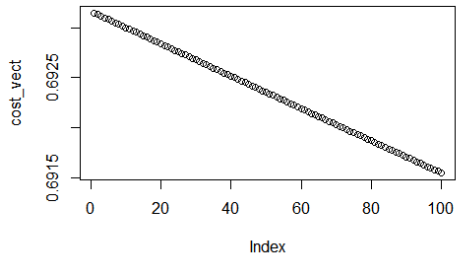

Figure 18: Jθ changes with iteration for $\alpha = 0.00003$

As we can see in cost with iteration plot above, we can see the linear change in Cost(Jθ) with every iteration. At 100^{th} iteration, cost is somewhere around 0.7. Hence, we can say that there is a scope to decrease the cost. So, algorithm does not converge at $\alpha = 0.00003$ for 100 iterations.

From the above analysis, we can conclude that both plot at $\alpha = 3$ and 0.3, the algorithm will yield better result than other two.