Matthew Cline
Machine Learning
Assignment 1: Regression
21 September 2017

Assignment 1 covered the the implementation of a linear and quadratic regression model. Gradient descent was also used as an optimization technique during the learning process. The following sections will cover the implementation of a single variable linear regression model, a single variable, quadratic regression model, and a multivariate linear regression model. Each section will also cover the methodology used in the testing of the model as well as the results. All of the data used in the experiments came from the data set provided to the class, covering the relationship between knowledge about various aspects of flu transmission and the perceived risk of contracting the flu.

1. Single Variable Linear Regression

The goal of the single variable linear regression model was to evaluate the correlation between the survey respondent's knowledge of flu transmission and their perceived risk of contracting the flu. The model was based on the linear relationship between these attributes represented in Equation 1.

$$h_0(x) = \theta_0 + \theta_1 x$$

Equation 1: Linear Regression

The fitness of the selected θ values was evaluated using a cost function. The cost function evaluated the performance of the model on a set of training data by finding the mean squared error between the values predicted by the model, and the labels associated with the data points. The cost function is represented in Equation 2.

$$J(\theta_0, \theta_1) = 1/2m * \sum_{i=0}^{m} (h_0(x^i) - y^i)^2$$

Equation 2: Cost Function

Gradient descent was then used to minimize the cost associated with the current values of θ. Each θ value was optimized iteratively by a factor of α multiplied by the partial derivative of the cost function with respect to each θ. The optimization can be seen in Equation 3.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Equation 3: Gradient Descent

The gradient descent algorithm was run iteratively until the change in the cost function associated with the current θ values versus the previous θ values approached zero or the maximum number of iterations had been reached. If the change in the cost function converged to zero, an acceptable set of θ values had been found, and the model was ready to be tested against the test data.

A total of three tests were ran against the model. Three different distributions of training data versus test data were used including: 20% training data, 50% training data, and 80% training data. The results captured include the number of iterations needed for the gradient descent algorithm, the optimal θ values discovered, and the value of the cost function when running the test data. The results are shown in Tables 1.

	20% Training	50% Training	80% Training
Iterations	67	69	75
$[\theta_0, \theta_1]$	[-0.057050647, -0.077152461]	[-0.04181292290866, -0.05842840462097]	[-0.09343715134972, -0.04873025285805]
Cost Function Test Data	0.1253980849036	0.13271077439231	0.10972014362612

Table 1: Single Variable Linear Regression Results

Since the value of α remained at a constant value of 0.1 throughout all three experiments the number of iterations needed for the gradient descent to converge was very similar across the board. The θ values in the 20% training set and the 50% training set were much more similar than the θ values found in the 80% training set. The data used in the experiments were not shuffled after being ingested, so there is a good possibility that the data at the beginning of the set was much more similar than the data in the second half of the set. That distribution could also explain why the 80% training set was much more effective at predicting the test data values. Another explanation for the effectiveness of the 80% training data set is simply being exposed to more of the data during training. The graph in Figure 1 shows the cost function converging to zero during the gradient descent of the 80% training data set.

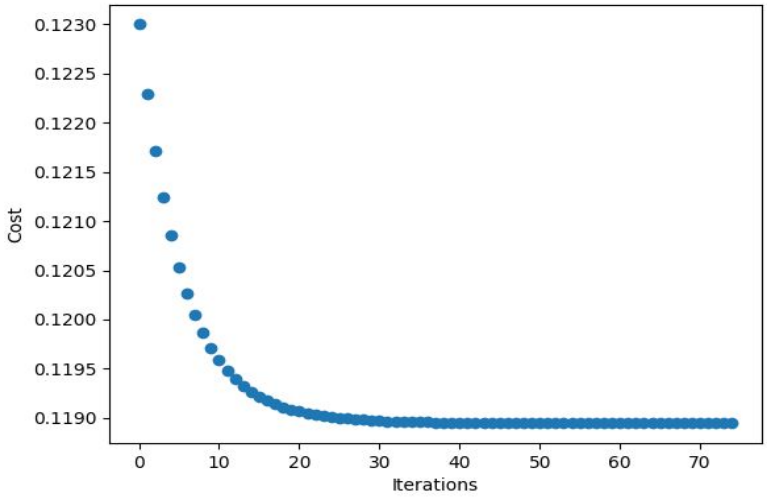

Figure 1: Gradient Descent Convergence

Figure 2 shows the best fit line that the optimal θ values found in the 80% training set create.

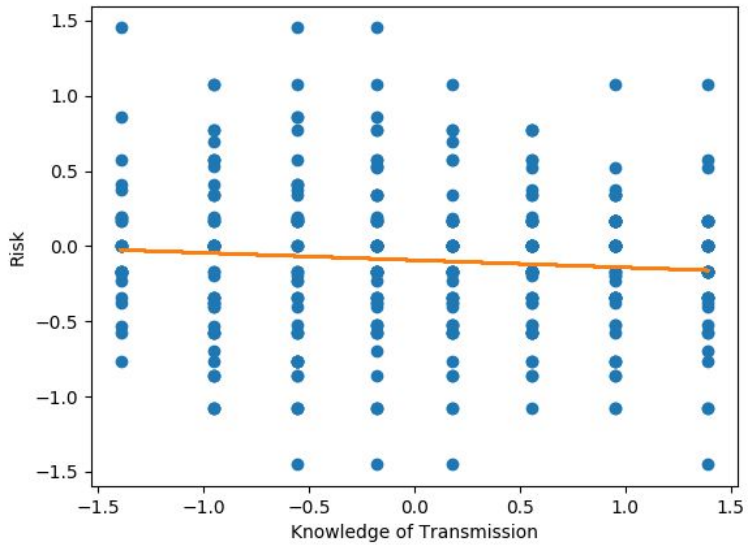

Figure 2: Linear Regression Model

2. Single Variable Quadratic Regression

The single variable quadratic regression model relies on most of the same techniques that the single variable linear regression model from Section 1 did. There are a couple of subtle differences however. The hypothesis function in Equation 1 was updated to a quadratic form shown in Equation 4.

$$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

Equation 4: Quadratic Hypothesis Function

The cost function and gradient descent functions remain the same with minor tweaks to the code in implementation to account for the additional terms in the hypothesis function. The same tests that were run in Section 1 were repeated with the quadratic model in order to keep the comparison as similar as possible. The results can be seen in Table 2.

	20% Training	50% Training	80% Training
Iterations	148	71	138
$[\theta_0, \theta_1, \theta_2]$	[0.010282183149, -0.069357456236, -0.091391377414]	[-0.02665562661055, -0.0580186635780, -0.02227728462526]	[-0.08636835671849, -0.04962548155071, -0.01099562660287]
Cost Function Test Data	0.12889203324525	0.13323938156505	0.11016777079484

Table 2: Quadratic Regression Results

In the quadratic regression testing, the 50% training set converged in the least number of gradient descent iterations, but also had the worst performance when evaluated against the test data. Like the single variable linear regression model, the most effective training data distribution was the 80% training and 20% test data. The quadratic model did not beat the linear model in this case though. The difference was less than 2% difference, but the linear model was able to learn faster and still perform more effectively. Figure 3 shows the best fit function that the optimal θ values provided in the quadratic regression testing. There is a very slight curve visible, but due to the even distribution of the data points it is very minimal and almost appears as a line.

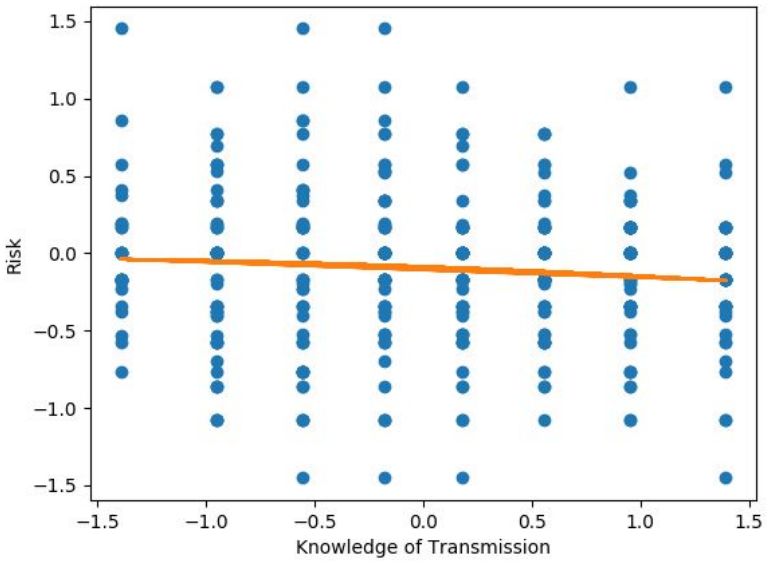

Figure 3: Quadratic Best Fit

3. Multivariate Linear Regression

The final model that was evaluated in the experiment was the multivariate linear regression model. For this model, an extra feature was added in an attempt to increase performance. In addition to the survey respondent's knowledge of flu transmission, their respiratory etiquette was taken into account. The revised hypothesis function can be seen in Equation 5 where x_1 is knowledge of transmission and x_2 is respiratory etiquette.

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Equation 5: Multivariate Hypothesis

Since the θ values are still the only unknowns, the cost function and the gradient descent functions still retain the same form as the other models, but with some slight changes to the code for implementation purposes. The same tests were run against the multivariate model with the 20% training set, the 50% training set, and the 80% training set. The results can be seen in Table 3.

	20% Training	50% Training	80% Training
Iterations		271	414
$[\theta_0, \theta_1, \theta_2]$		[0.02770784051620, -0.05192090377255, -0.01701127858740]	[-0.09704617418260, -0.04950457321331, 0.000950058275570]
Cost Function Test Data		0.1336736419872	0.10971408290955

Table 3: Multivariate Regression Results

The multivariate linear regression model beat the single variable linear regression model in the 80% training set, but only by 0.001%. The multivariate model also took much longer to converge during gradient descent due to the increased dimensionality of the feature set. The performance gains of the model do not justify the extra computation needed to achieve them. Figure 4 shows the three dimensional representation of the data points and the best fit plane approximated by the optimal θ values found during gradient descent.

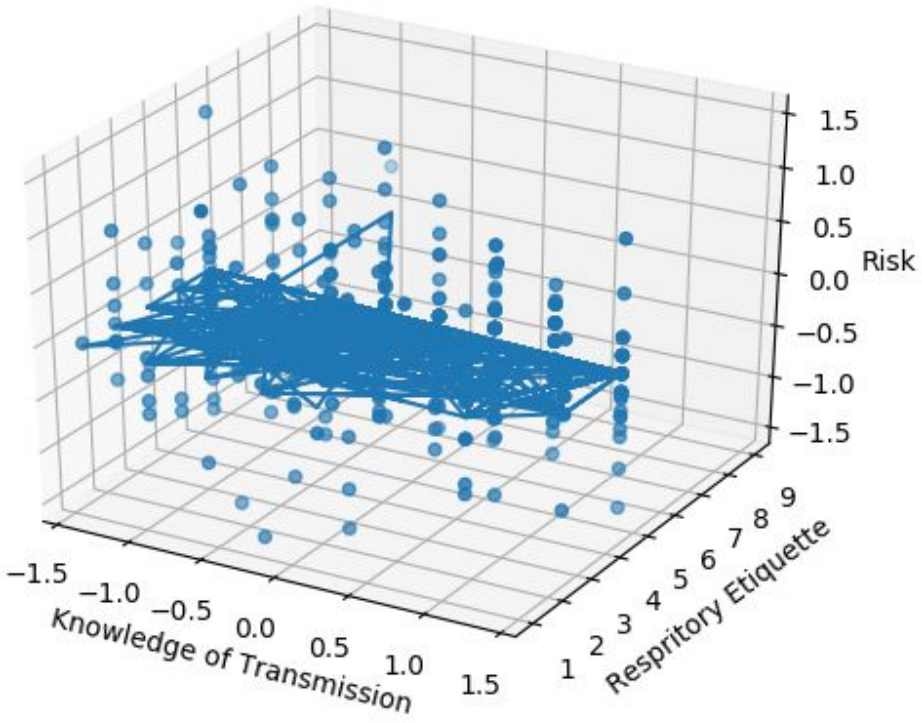

Figure 4: Multivariate Best Fit

Conclusion

The most efficient regression model for this particular data set was a simple single variable linear regression. The multivariate regression model beat the single variable model in accuracy, but the difference was within a rounding error. The extra computing power needed to perform the calculations of the more complex models would be wasteful on a similarly structured large data set. The data examined in this experiment did not show a strong correlation between the features used in the model, but due to the consistent distribution, the error values remained relatively low.