Machine Learning: Student Procrastination

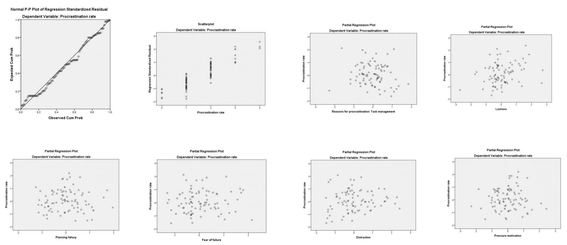
Cecilia Weingartner and Felix Weingartner

Introduction

The goal of this project was to explore machine learning algorithms utilizing a database containing information of student procrastination. So, what is machine learning(ML)? Well, machine learning is a lot of things, but a basic overview can be described as a field of study that gives computers the ability to learn without being explicitly programmed. Some refer to it as a from of artificial intelligence, data mining, pattern recognition or computational statistics. We will explore a few basic ML algorithms. Using the selected features to model the ML algorithm, we hope to predict the student's procrastination rate based on tasks management, laziness, fear of failure, distraction, planning fallacy, and pressure motivation.

Method

Starting with a survey of 1000 data entries on student procrastination, 100 were selected for the supervised machine learning. What this means is using the 100 randomly selected data as the training set and 60 randomly selected data as the testing set. Each data has a corresponding student's procrastination rate. Therefore, we will use the supervised ML algorithm. Using IBM SPSS, we have generated some significant results from the training set data. Using Chi-Square Test with p-values < 0.01, the feature list for tasks management, laziness, fear of failure, distraction planning fallacy, and pressure motivation from the sample data are significantly associated with student's procrastination rate. Therefore, we have chosen the features for this ML model.



Abstract

This project is to explore machine learning algorithms on how to predict student procrastination level. More than 1000 survey data results from student projects from last semester, Fall 2015, randomly selected 100 data for the training data set and 60 for the testing data set. Using the hypothesis function to predict the ranking of student procrastination ranking from none to always.

1. **Linear Model**

Multivariate Linear regession:

Hypothesis equation: $h_\theta(x) = \theta^T x = \sum_{i=1}^{n} \theta_i x_i$

The iterated gradient descent update:

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$$

α is the learning rate

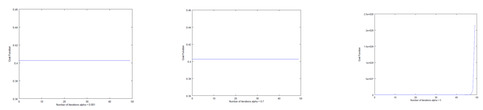
The cost function: $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

2. Simplify the illustration: using 1 feature for the model $h_\theta(x) = \theta_0 + \theta_1 x_1$

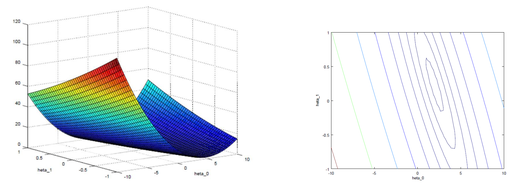
Calculate the starting $\theta = \left(x^T x\right)^{-1} x^T y$

After 50 iterations:
$\theta_{int} = [2.49, 0.2556]$
$\alpha = 5$: $\theta = [-2.0583 \times 10^{14}, -2.7373 \times 10^{13}]$
$\alpha = 0.7$: $\theta = [2.49, 0.2556]$
$\alpha = 0.001$: $\theta = [2.49, 0.2556]$

Therefore the results for the hypothesis equation:
$h_\theta(x) = 2.49 + 0.2556x$



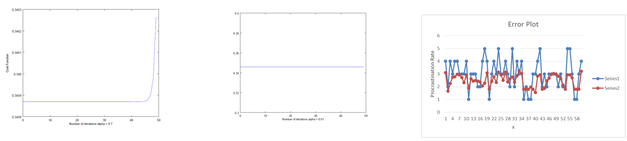
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h(x_{t,i}) - y)^2$$



3. Six features:
$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5 + \theta_6 x_6 = \theta^T x$

After 50 iterations:
$\theta_{int} = [2.49, -0.159148, 0.373563, 0.117043, 0.128663 - 0.030372, 0.034976]$
$\alpha = 0.7$: $\theta = [2.49, -0.165107, 0.368251, 0.111485, 0.123644, -0.036524, 0.029243]$
$\alpha = 0.6$: $\theta = [2.49, -0.159148, 0.373563, 0.117043, 0.128663 - 0.030372, 0.034976]$
$\alpha = 0.01$: $\theta = [2.49, -0.159148, 0.373563, 0.117043, 0.128663 - 0.030372, 0.034976]$

Therefore the results for the hypothesis equation:
$h_\theta(x) = 2.49 - 0.159148x_1 + 0.373563x_2 + 0.117043x_3 + 0.128663x_4 - 0.030372x_5 + 0.034976x_6$



Discussion

The results have indicated that the model does not need to use the gradient descent to find the optimization point. The initial calculation for the theta using the normalization equation of inverse matrix of x transpose times x and multiply x transpose and y will get the optimized value. However, if the size of the training set increases, it is harder to obtain the inverse of matrix. The cost for the computational time will be higher.

Since the output y can be classified as binary, the logistic model can be used as well.

The unsupervised learning method will be the next model to explore.

At the end, better feature should always to be considered in the future to implement a better machine learning algorithm.

Acknowledgements

Survey data collected from student projects in Fall 2015 (Math1040 and Math1030)
IBM SPSS
Octave