MSc in CSTE: CIDA option
Advanced Java & Advanced Python Assignment

Peter Sherar & Stuart Barnes
Cranfield University

September 19, 2023

Hand in date: 23/10/2023 (FT), 06/11/2023 (PT), 09:30am

1 Introduction

Linear regression is one of the most useful approaches for predicting a single quantitative (real-valued) variable given any number of real-valued predictors (or features). Given a list of input values x_i^j (observations), where j ranges from 1 to r where r is the number of features and i ranges from 1 to n where n is the number of observations, and output (outcome) values y_i, we have to find parameters $\theta_0, \dots, \theta_r$ such that the multi-linear function

$$y(x_i^1, \dots, x_i^r) = \theta_0 + \theta_1 x_i^1 + \ldots + \theta_r x_i^r$$

is as close as possible to the observed outcome y_i. We estimate the parameters by fitting the model to training data. Although it is almost never correct, a linear model often serves as a good and interpretable approximation to the unknown true function $f(X)$. We evaluate how well the vector $(\theta_0, \dots, \theta_r)$ approximates the data by defining a cost function. For multi-linear regression an appropriate cost function is the mean square error.

$$J(\theta_0, \dots, \theta_r) = \frac{1}{n}\sum_{i=1}^{n}\Big(y(x_i^1, \dots, x_i^r) - y_i\Big)^2 = \frac{1}{n}\sum_{i=1}^{n}\Big(\sum_{j=1}^{r}\theta_j x_i^j + \theta_0 - y_i\Big)^2$$

This is an optimisation problem with some parameters $(\theta_0, \dots, \theta_r)$ that we can tweak, and some cost function $J(\theta_0, \dots, \theta_r)$ we want to minimise. One way of solving this problem is to use a method called gradient descent. If we imagine $J(\theta_0, \dots, \theta_r)$ as a hyper-surface then what we are attempting to do is to find the lowest point on the surface. Although we don't know where the minimum point is we can move in a direction towards it by moving downwards. The direction in which $J(\theta_0, \ldots, \theta_r)$ decreases most is given by the gradient:

$$\nabla J(\theta_0, \ldots, \theta_r) = \langle \frac{\partial J(\theta_0, \ldots, \theta_r)}{\partial \theta_0}, \ldots, \frac{\partial J(\theta_0, \ldots, \theta_r)}{\partial \theta_r} \rangle$$

where

$$\frac{\partial J(\theta_0, \ldots, \theta_r)}{\partial \theta_0} = \frac{2}{n} \sum_{i=1}^{n} (y(x_i^1, \ldots, x_i^r) - y_i)$$

$$\frac{\partial J(\theta_0, \ldots, \theta_r)}{\partial \theta_1} = \frac{2}{n} \sum_{i=1}^{n} (y(x_i^1, \ldots, x_i^r) - y_i) x_i^1$$

$$\frac{\partial J(\theta_0, \ldots, \theta_r)}{\partial \theta_r} = \frac{2}{n} \sum_{i=1}^{n} (y(x_i^1, \ldots, x_i^r) - y_i) x_i^r$$

We update the $(\theta_0, \ldots, \theta_r)$ parameters in each step with

$$\theta_j = \theta_j - \eta \frac{\partial J(\theta_0, \ldots, \theta_r)}{\partial \theta_j}, \ldots j = 0, \ldots, r$$

Here η is a 'learning rate' parameter. If we set η to a small value we approach the minimum slowly, whilst a larger value implies bigger steps and so quicker movement downhill towards the minimum but which may overshoot missing it altogether.

2 Tasks

Write a Python/Java program which implements gradient descent algorithm for a nulti-linear regression problem based on r predictors and which takes in the value of η and the number of steps of the iteration. Test your implementation on the *advertising.csv* and *auto.csv* data sets as described below. Choose appropriate test and train dataset proportions for the computation. Evaluate the 'goodness of fit' by computing the cost function and using the r-squared test. This measures the proportion of the total variance in the output (y) that can be explained by the variation in x:

$$1 - \frac{\sum_{i=1}^{n} \left(y_i - y(x_i^1, \ldots, x_i^r)\right)^2}{n \cdot var(y)}$$

where $var(y)$ is the variance of the y values. Compare the analytic solution against the gradient descent method for varying number of iteration steps, 10, 50, 100, 1000, .. and varying values of η (0.1, 0.01, 0.001, 0.0001). Use a test for convergence based on the differences between successive values of θ_j. Take θ_j to be zero initially.

Note: Something to bear in mind. The ideal scenario is when the predictors are uncorrelated. Correlations amongst predictors cause problems. It is good to remember that a regression coefficient θ_j estimates the expected change in y per unit change in x_j , with all other predictors held fixed. But in reality predictors usually change together.

There are two datasets to test out the algorithm on.

The *advertising* dataset consists of some advertising and sales data that we have seen during the course. To improve sales of a particular product some data consisting of the sales of that product in 200 different markets, along with advertising budgets for the product in each of those markets for three different media: TV, radio, and newspaper. Typical questions one might ask when considering this dataset are:

Is there a relationship between advertising budget and sales?

How strong is the relationship between advertising budget and sales?

Which media contribute to sales?

How accurately can we predict future sales?

Is the relationship linear?

Is there interaction among the advertising media?

The *auto* dataset gives some information about cars, auto gas mileage, horsepower and other information. In particular, the variables are *mpg, cylinders, displacement, horsepower, weight, acceleration, year origin, name.*

1. For the *advertising* dataset:

   i/ use two predictors, TV and radio to predict sales:

   $$sales = \theta_0 + \theta_1 * TV + \theta_2 * radio$$

   In this case plot the resulting regression function obtained (which will be a plane) along with the dataset (TV, radio, sales).

   ii/ use all three predictors, TV, radio and newspaper to predict sales:

   $$sales = \theta_0 + \theta_1 * TV + \theta_2 * radio + \theta_3 * newspaper$$

2. Removing the additive assumption: interactions and nonlinearity

   In the analysis above we assumed that the effect on sales of increasing one advertising medium is independent of the amount spent on the other media. The linear model,

   $$sales = \theta_0 + \theta_1 * TV + \theta_2 * radio + \theta_3 * newspaper$$

states that the average effect on sales of a one-unit increase in TV is always θ_1, regardless of the amount spent on radio (i.e they are independent).

But suppose that spending money on radio advertising actually increases the effectiveness of TV advertising, so that the slope term for TV should increase as radio increases. In this case the model takes the form

$$sales = \theta_0 + \theta_1 * TV + \theta_2 * radio + \theta_3 * (radio * TV)$$

Investigate this model, how does it compare with the others.

3. For the *auto* dataset examine the non-linear effect of predictors by comparing the linear, quadratic and degree 5 dependence of *mpg* (target) on *horsepower* (predictor). Plot the resulting curves along with the dataset Which model do you think best fits the data and why?

3 Source Code and Report Requirements

Write a report to present and discuss your findings. The report should be no less than 2,000 words and must not exceed 3,000 words. The report can contain any number of figures/tables, however all figures/tables should be numbered and discussed. The report should include a description of the design of your solution explaining your choices. The source code and documentation should be included as an Appendix to the report.

4 Assignment Submission

The source code and documentation should be submitted electronically via the **Technical Work submission point** by 9:30am on the 23rd October (full-time students) or the 6th November (part-time students). The report should be submitted electronically via the **TurnItInUK submission point** by the prescribed deadline, for the assignment submission to be considered complete. This is a group assessment.

5 Marking

The assignment will be assessed based on the following marking scheme:

- 20% Introduction, methodology, conclusions
- 40% Source code, documentation
- 30% Analysis and discussion of the results
- 10% Report structure, presentation, references

6 References

1. https://www.cs.toronto.edu/~rgrosse/courses/csc311_f20/readings/notes_on_linear_regression.pdf
2. https://statisticsbyjim.com/regression/interpret-r-squared-regression/
3. https://towardsdatascience.com/a-quick-overview-of-optimization
-models-for-machine-learning-and-statistics-38e3a7d13138