Problem Set 4

D Keck

March 18, 2018

Objectives

- Implement linear regressions solvers: library, direct (normal equations),and gradient descent
- Understand and verify all output from library solver
- Compare predictions to neural network solver
- Feature selection for linear regression

Due April 4

Requirements

- Problem sets are done in groups of 3 (21 groups)
- Each group member must be skilled with the language chosen at the start of problem set
- Each student must do approximately $\frac{1}{3}$ of the development
- The deliverable is single PDF or HTML that meets the standard of a reproducible research report with professional quality communication. Deployable code will not be a deliverable for any problem set.
- All code must be displayed
- Identify and explain each answer. Don’t just print a number.
- Numeric output should be easy to read e.g., not 10 decimal places
- Graphics must be easy to read i.e., titles, axis labels, legends, colors, etc.
- The rows and columns of tables and the columns of matrices and data frames must be labeled
- These problem statements provide less detailed guidence, but you are expected to follow the data science process, especially exploratory data analysis and good communication of process and results.
- A maximum of 7 points will be awarded for minimalist correct answers. All 9 points will be awarded for nice explorations and descriptions that include correct answers.

Predict the Quality of Portuguese White Wines

Data Source (http://archive.ics.uci.edu/ml/datasets/Wine+Quality)

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.
Modeling wine preferences by data mining from physicochemical properties.
In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

Relavant Article (https://www.nytimes.com/2014/06/11/dining/tasting-portuguese-white-wines.html?_r=0)

You may use another data set from UCI or Kaggle if you prefer, but you must follow each step in this assignment

Question 1

- Download *winequality-white.csv*
- How many observations?
- How many explanatory and response variables?
- Is there any missing data?

Question 2

- Describe the structure and range of the data (suggest str() and summary())
- Plot histograms of each feature and response (suggest specifing the number of rows and columns for this plot)
- Comment on the correlation of the features (suggest corrplot() and pairs())
- Scale the explanatory variables (suggest looking at the histograms to choose between normal and uniform scaling)

Question 3

- Split the data into a training and test set. Put 25% of the data in the test set. (Suggest using sample.split() in caTools)
- What is the baseline *wine quality* prediction accuracy on the training set?
- Develop an lm() object using all of the explanatory variables
- Print the model information using summary()
- Print the model information criterion using AIC(), extractAIC(), and logLik()
- Predict the wine quality using the test set and compare the accuracy to the actual quality. Comment.
- Print the parameter estimates and their 95% confidence intervals in a single table. (Suggest using confint()), and cbind()

Question 4

Roll your own code to compute model parameters, $\widehat{\boldsymbol{\theta}}$ as well as the model information from the library solver

- First create the $\mathbf{X}$ matrix and the $\mathbf{y}$ vector for the training data (Remember to insert a column of 1's in the $\mathbf{X}$ matrix)
- Compute the parameter values (the coefficent estimates in the lm() object)
- Print the parameters from the lm() model and from your normal solver side by side. Comment. (suggest using head())
- Print the test set quality from the lm() model and from your normal solver side by side. Comment. (suggest using head())
- Print the rmse error between the predicted and actual test qualities

Question 5

- Now compute the parameters using the gradient descent solver using the same $\mathbf{X}$ and $\mathbf{y}$
- First, write a function to compute the scalar value of the cost function $J(\boldsymbol{\theta})$
- Clearly display your learning rate, α and your convergence criterion
- Print the estimated parameters from the lm() model, your normal solver, and your gradient descent solver side by side. Comment.
- Predict the wine quality using the gradient descent parameter using the test set and compare to the actual quality in the test set

Question 6

- Compare accuracies on the test set to those of a neural net model. Comment.
- Describe your final neural net model.

Question 7

Now re-compute all of the information from your lm() model using your *normal equation* model

- Compute error residuals, $\mathbf{e}$, and plot the histogram of residuals

- Print the summary() of the error vector, $\mathbf{e}$, and compare to lm() model output. Comment
- Plot histogram of residual errors to check approximate normality. If the errors were not nearly normal what might be the problem?
- Most residual errors are less than $|1|$, what does that mean ?

- Compute the residual standard error and the degrees freedom for the residual error

Question 8

Continue Question 7

- Create and print a table similar to that in lm() output for your theta values for estimates, $\widehat{\theta}$, compute standard error, T values, and P values.

Question 9

Continue Question 7

- Compute R^2
- Compute R^2_{ADJ}
- Compute AIC

- Compute the F statistic for the model
- Compute degrees of freedom 1 and 2 for the f diistribution
- Compute the P value for the F, overall model, statistic

Question 10

- Reduce the number of explanatory variables in your lm() model one by one to find the best model using the AIC criterion (tradeoff between maximum likelihood and number of parameters). (suggest using step(lm(),…))
- Increase the number explanatory variables from the intercept alone in your lm() model one by one to find the best model using the AIC criterion
- Note that step(lm()) uses extractAIC() not AIC()