CS385 : Assignment 3

Topics covered: Linear Regression, Gradient Descent Algorithm, RMSE

Deliverables: Your submission for this assignment should be an archive of two files, named `lin_reg.py` and `yourusername_report.pdf`.
`lin_reg.py` should contain functions list below.

Function name	Input	type	Output	type
dataLoad	Filename	String	X: dataset	array
dataNorm	X: the loaded dataset	Array with shape (506, 14)	X_norm: the normalized dataset	array
errCompute	X_Norm: dataset for computing the prediction error	Array with shape (506, 15)	Price of house	float
	Theta: parameters for linear model	Array with shape (14,1)		
gradientDescent	X: dataset for training and predicting	Array with shape (#, 15)	Learned parameters theta	array with shape (14,1)
	Theta: parameters for linear model	Array with shape (14,1)		
	Alpha: learning rate	Float		
	Num_iters: number of iterations	Int		
rmse	testY: the model predict value	Array with shape (#samples, 1)	The value of RMSE	float
	stdY: the predict in training data			

`Yourusername_report.pdf` should follow the outline described in Section 4 and you should put in all the required materials in Section 5. You should put two files in a directory called `cs385_yourusername_3` and zipping the directory into a zip file called `cs385_yourusername_3.zip` etc. In addition to the two files, your zip file should contain a folder named `output`, where your output files are stored. In addition, add one more folder named `data` where you will store your training datasets that you have split using the methods stated in Section 5. This will help me to run your code with the exact split of train and test dataset that you have made in your implementation. Failure to follow conventions will result in penalties in marks.

Objectives:

- To get familiarized implementing your second machine learning algorithm, which is Linear Regression. You are also expected to implement it by using *numpy*.
- To implement Gradient Descent algorithm to learn the parameters of linear regression function from the training data.
- To apply linear regression algorithm to predict the housing prices in Boston city.
- To get familiarized on tuning the performance of gradient descent algorithm.

1. Linear Regression

In statistics, linear regression is a linear approach for modeling the relationship between a scalar dependent variable **y** and one or more explanatory variables (or independent variables) denoted **X**. The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regressions. Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in many practical applications. Linear regression has been used in prediction, forecasting, and error reduction by fitting a predictive model to an observed data set of **y** and **X** values. After developing such a model, if an additional value of **X** is then given without its accompanying value of **y**, the fitted model can be used to make a prediction of the value of **y**.

In this assignment, you will design the linear regression algorithm to predict the housing prices in Boston using the dataset provided by the UCI Machine Learning repository[1].

2. Gradient Descent

When there are one or more inputs variables, you can use a process of optimizing the values of the coefficients by iteratively minimizing the error of the model on your training data. This operation is called Gradient Descent and works by starting with random values for each coefficient. The sum of the squared errors is calculated for each pair of input and output values. A learning rate is used as a scale factor and the coefficients are updated in the direction towards minimizing the error. The process is repeated until a minimum sum squared error is achieved or no further improvement is possible.

When using this method, you must select a learning rate (alpha) parameter that determines the size of the improvement step to take on each iteration of the procedure. Gradient descent is often taught using a linear regression model because it is relatively straightforward to understand. In practice, it is useful when you have a very large dataset either in the number of rows or the number of columns that may not fit into memory.

In this assignment, you will implement the gradient descent algorithm from the scratch to learn the parameters for your linear regression model.

3. Dataset

The Boston house price dataset contains information about the housing values in suburbs of Boston. This dataset was originally taken from the StatLib library which is maintained at Carnegie Mellon University and is now available on the UCI Machine Learning Repository. Each instance describes the properties of a Boston suburb and the task is to predict the house prices in thousands of dollars. There are 13 numerical input variables with varying scales describing the properties of suburbs. The readme file in the dataset provides details on the dataset. Read this file before implementing your algorithm to check whether you need to do any data preprocessing steps. To give a brief introduction, the Boston housing data consists of 506 entries that represent the aggregated data about 14 features for homes from various suburbs in Boston. The input attributes and the output attribute are as follows:

1. CRIM: per capita crime rate by town
2. ZN: proportion of residential land zoned for lots over 25000 sq.ft.
3. INDUS: proportion of non-retail business acres per town

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/housing/

4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. NOX: nitric oxides concentration (parts per 10 million)
6. RM: average number of rooms per dwelling
7. AGE: proportion of owner-occupied units built prior to 1940
8. DIS: weighted distances to five Boston employment centers
9. RAD: index of accessibility to radial highways
10. TAX: full-value property-tax rate per $10,000
11. PTRATIO: pupil-teacher ratio by town
12. B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT: % lower status of the population
14. MEDV (output): Median value of owner-occupied homes in $1000's

4. Documentation

For documentation, you can use LATEX or Word format. You need to write a report on this assignment by following instructions in Section 5. You should submit ONLY the "pdf" of the document and NO other file formats will be accepted.

5. Your tasks

- **a.** Write a function *loadData(filename)* to load housing data into an array *X*. (Hint: use split() for separating columns). *X.shape* should be <u>(506,14)</u>.
- **b.** Write a function *dataNorm(X)* for feature normalization.
  This time, try to use array operation instead of using for loop.
  You can use *numpy.min(X[:,0:13],axis=0)* and *numpy.max(X[:,0:13],axis=0)* to get the maximum and minimum for each feature. (hence the returned value is a vector with size 13). Then use *(X[:,0:13]-min)/(max-min)* to get the normalized data.
  The last step of this task is to insert a column of ones to *X_norm* as the first column, and append the output column as the last column.
  After the above steps, *X_norm.shape* will be <u>(506,15)</u>
  Check your normalized data *X_norm* by running *testNorm([X_norm])*

		Mean	Sum
Col0	1	1	506
Col1	CRIM	0.0405441	20.51531372
Col2	ZN	0.11363636	57.5
Col3	INDUS	0.39137752	198.03702346
Col4	CHAS	0.06916996	35
Col5	NOX	0.34916679	176.67839506
Col6	RM	0.52186901	264.0657214
Col7	AGE	0.67636355	342.23995881
Col8	DIS	0.24238128	122.64492721
Col9	RAD	0.37171335	188.08695652
Col10	TAX	0.42220831	213.63740458
Col11	PTRATIO	0.62292911	315.20212766
Col12	B	0.89856783	454.67532402
Col13	LSTAT	0.30140903	152.51296909
Col14	MEDV(output)	22.53280632	11401.6

c. Construct the linear regression equation to predict the housing prices in Boston. Explain this equation briefly and the number of parameters that need to be estimated. Write down your equation, brief explanation and the number of needed parameters in *Section 1 LR Equation* of your report.

d. Plot the input attributes (pick any 5 relevant attributes) and the output attribute in the given dataset to visualize how the output variable varies with respect to each of your input attribute. Paste your plot in *Section 2 Feature-Price Plot* of your report.

e. Construct the error function using the linear regression equation that you constructed in *task c*. Again, briefly explain this equation. Write down them in *Section 3 Error Function* of your report.

f. Use the error function (from task e) to write a function *errCompute()* for monitoring the convergence. This function takes in dataset *X_norm* and parameters *theta*(with shape (14,1)), returns the error *J*.

Run *errCompute(X_norm, np.zeros((X_norm.shape[1]-1,1))*, you will get result 296.0734585

g. Write *gradientDescent()* to implement the gradient descent learning algorithm for learning the parameters. This function takes in dataset *X_norm*, *theta*, learning rate *alpha*, and maximal iterations *num_iters*. It returns the learned *theta*.

Run *theta = gradientDescent(X_norm, np.zeros((14,1)),0.01,1500)*, then use the returned data to predict the price for *X_norm[0,0:14]*, i.e. *np.dot(X_norm[0,0:14],theta)*, you'll get a price 27.7865

A major part of this task requires you to find the optimal value of the learning rate (i.e., alpha) and the convergence criterion. How will you do this? Explain in *Section 4 Learning Rate*. (Also note, you may need to use this implementation in your future assignments)

h. Split the dataset into training and test set using k-fold cross-validation method. Choose the k value to be 5, 10 and 15. (You can use your code from your previous assignment)

i. For each of the folds in task h, run the gradient descend algorithm to compute the parameters. Using this parameters, plot the values of predicted_y and the actual y values from the training data. What do you observe? In addition, you need to evaluate on how accurate is your model predicts the output. In order to do this, you need to measure RMSE (root mean squared error). Explain briefly on RMSE and compute the RMSE for each of the fold in task h. Include the required plots and discussion in *Section 5 Experimental Result*.

5. Rubrics

This assignment is graded over total of 50 points. The breakdown is as follows:

- task a and task b Data Load and Normalization (10 points)
- task c: Proper construction of linear regression equation and sufficient explanation (5 points)
- task d: Proper plot and discussions (5 points)
- task e: Error function equation and descriptions (5 points)
- task f: Error function implementation (5 points)
- task g: Correct implementation of Gradient Descent algorithm and answers to task g (10 points)
- Plots of predicted_y and actual y values from dataset in task h and i (5 points)
- Provide a table of RMSE values for each value of k and explain (5 points)

Note: Submission requirements must be met i.e., reasonable comments, proper format of the report (Not copy and pasted from the assignment specifications!), detailed explanation (using necessary equations, tables, figures, charts, etc.), correct documentation and file formats for the submission, etc. If any of these requirements are not satisfactory, then zero marks for this assignment.