(new) Introduction to Machine Learning - Practicum 4 - Logistic Regression

Topics covered: Logistic Regression, Stochastic Gradient Descent Algorithm

Deliverables:

- Your submission for this practicum should be an archive of two files, named ***logistic_reg.py*** and ***<your_name>_report.pdf***
- ***logistic_reg.py*** should include functions listed in the following table:

Function name	Input	type	Output	type
`dataLoad`	Filename	String	X: dataset	array
`dataNorm`	X: the loaded dataset	2D array with shape (1372,5)	X_norm: the normalized dataset with dummy(1) X_0	2D array with shape (1372,6)
`errCompute`	X_norm: dataset for computing the cost	2D array with shape (1372,5)	Cost	float
	Theta: parameters for linear model	Array with shape (5,1)		
`stochasticGD`	X: dataset for training and predicting	2D array with shape (#,6)	Learned parameters theta	2D array with shape (5,1)
	Theta: parameters for linear model	2D array with shape (5,1)		
	Alpha: learning rate	Float		
	Num_iters: number of iterations	Int		

`Predict`	X_norm	2D array with shape (#,6)	The predict class for each sample	2D array with shape (#,1)
	Theta	Array (#,1)		

- ***<your_name>_report.pdf*** should follow the outline described in the *Documentation* section.
- You should put in all required materials as described in the *Tasks* section
- You should zip both files named ***<your_name>_ML_P04.zip***.
- In addition to the two files, your zip file should also contain the following:
    - a folder named ***output***, where your output files are stored.
    - a folder named ***data***, where you will store your training datasets that you have split using the methods stated in the *Tasks* section.

  This will be used 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.

Objectives:

- To get familiarized implementing a machine learning algorithm, which is Logistic Regression from the scratch (i.e., without using any machine learning API).
- To implement Stochastic Gradient Descent algorithm to learn the parameters of logistic regression function from the training data.
- To apply logistic regression algorithm to classify whether the bank notes is genuine or fake.
- To get familiarized on tuning the performance of stochastic gradient descent algorithm.
- To improve the capability to write simple technical report on describing a machine learning algorithm and its performance.

1. Logistic Regression

In statistics, logistic regression, or logit regression, or logit model is a regression model where the dependent variable is categorical. In this practicum, we will focus on the binary dependent variable—that is, where the output can take only two values, "0" and "1", which represent outcomes such as fake/original. In this practicum, you will design the logistic regression algorithm to classify the bank notes as genuine or fake using the dataset provided by the UCI Machine Learning repository https://archive.ics.uci.edu/ml/datasets/banknote+authentication#.

2. Stochastic Gradient Descent

Stochastic gradient descent, also known as incremental gradient descent, is a stochastic approximation of the gradient descent optimization and iterative method for minimizing an objective function that is written as a sum of differentiable functions. In other words, this algorithm tries to find minima or maxima by iteration. 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. In this practicum, you will implement the stochastic gradient descent algorithm from the scratch to learn the parameters for your logistic regression model.

3. Dataset

This dataset is about distinguishing genuine and forged banknotes. Data were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400 x 400 pixels. Due to the object lens and distance to the investigated object, gray-scale pictures with a resolution of about 660 dpi were gained. A Wavelet Transform tool was used to extract features from these images. The input and the output attributes are as follows:

1. Variance of Wavelet Transformed image (continuous)
2. Skewness of Wavelet Transformed image (continuous)
3. Kurtosis of Wavelet Transformed image (continuous)
4. Entropy of image (continuous)
5. Class (target): Presumably 0 for genuine and 1 for forged

Note: The dataset need to be randomly shuffled before you split the data.

4. Documentation

For documentation, you can use LATEX or Word format that has been provided in your first practicum. You need to write a detailed technical report on this practicum by explaining the algorithm, the techniques you have used to implement it, your dataset, evaluation metrics and how did you split your training and test data, etc. The results should be plotted using appropriate graphs and if needed use tables to report the results. More importantly, you need to acknowledge the learning materials in the reference section that you have used for this practicum. You are highly encouraged to use equations, tables, figures, charts, etc. wherever appropriate. Failing to do so will result in deduction of points for documentation. You should submit ONLY the “pdf” of the document and NO other file formats will be accepted.

5. Tasks

a. Write the function `dataLoad(filename)` to load banknotes data into an array `X`. `X.shape` should return `(1372,5)`.

b. Write a function `dataNorm(X)` for feature normalization. Don't forget to insert a dummy feature X_0 as the first column. The shape of the returned `X_norm` is `(1372,6)`. The mean and sum for each feature:

		Mean	Sum
Col0	1	1	1372
Col1	Variance	0.5391	739.664
Col2	Skewness	0.5873	805.78
Col3	Kurtosis	0.2879	395.03
Col4	Entropy	0.6689	917.75
Col5	Class(output)	0.4446	610

c. Construct the logistic regression equation to classify whether the bank note is genuine or fake. Write down your equation on a paper. 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 Logistic Regression Equation*** of your report.

d. Construct the error function(entropy loss) using the logistic regression equation that you have done in task c. Again, briefly explain this equation in ***Section 2 Error Function***.

e. Use the error function (task d) to write a function `errCompute()` for monitoring the convergence. This function takes in dataset `X_norm` and parameters `theta` (with shape (5,1)), returns the error `J`. Run `errCompute(X_norm, np.zeros((X_norm.shape[1]-1,1)))`, you should get 0.6931. You may find that the function `scipy.special.expit()` would be helping in implementing sigmoid.

f. Describe briefly on the stochastic gradient descent learning algorithm for learning the parameters of your logistic regression function in ***Section 3 SGD Algorithm***.

Implement the function `stochasticGD()`. This function takes in dataset `X_norm` (should be shuffled), `theta`, learning rate `alpha`, and maximal iterations `num_iters`. It returns the learned theta.

To test your code, a shuffled data set (***shuffled.data***, but it is a non-normalized data set) is provided. Use the normalized shuffled data set to run `theta = stochasticGD(X_shufnorm, np.zeros((X_shufnorm.shape[1]-1,1)), 0.01, 1372*20)`, you'll get the cost(return by `errCompute()`) 0.3151.

Use your learned theta to predict each sample in `X_shufnorm`. The predict value $\hat{y}$ for each sample is in file ***predict.data***. The other major part of this task is 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***.

g. Split the dataset into training and test set using 5 sets of train-and-test split method with 60 – 40% split.

h. For each of the set in task g, run the stochastic gradient descend algorithm to compute the parameters. Write down the value of the parameters that you have obtained using an appropriate table. What do you observe from this table? Using the parameters that you obtain from this step, evaluate on how accurate is your model that predicts the output variable. For all the test sets in task g, you need to add a table to show the accuracy of your algorithm in ***Section 5 Experimental Result***.

i. Plot a graph on the error function of your logistic function against iteration number. You can showcase this by taking any one of the set from the task g. What do you observe? Write your observation as the final sub-section in ***Section 5 Experimental Result***.

6. Rubrics

This practicum is graded over total of 100 points. The breakdown is as follows:

- Task a and task b, data load and normalization. (10 points)
- Task c: Proper construction of logistic regression equation and sufficient explanation (10 points)
- Task d: Error function equation and descriptions (10 points)
- Task e: Error function implementation (10 points)
- Task f: Correct implementation of Stochastic Gradient Descent algorithm and answers in your report section 3 and section 4 (10+10+10 points)
- Task g, h & i: Tables and Plots in your report section 5 (20+10 point)

Note: Submission requirements must be met i.e., reasonable comments, proper format of the report (Not copy and pasted from the practicum 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 practicum.